

D0.5 Proceso de desarrollo para el caso de estudio a desarrollar

Metodología de desarrollo del software

Proceso unificado racional desde tres perspectivas

1. Perspectiva dinámica (fases del modelo a través del tiempo):
 - Fase de inicio: (se define el alcance del proyecto => Cliente)
 - Fase de Elaboración: (definición, análisis y diseño => Casos de uso)
 - Fase de desarrollo o construcción: (implementación del proyecto)
 - Fase de transición: (fin del proyecto y pruebas) => Usuarios finales)
2. Perspectiva estática (actividades del proceso o flujo de trabajo).
 - Modelado de negocio (caso de uso de la empresa)
 - Requisitos (actores, casos de uso, requerimientos del sistema)
 - Análisis y diseño (modelo de diseño, componentes, objetos, secuencias)
 - Implementación (componentes del sistema en subsistemas)
 - Pruebas (proceso iterativo)
 - Despliegue (liberación de producto y distribución)
3. Perspectiva práctica (buenas prácticas).
 - Gestión de requerimientos (documentar y analizar cambios)
 - Arquitectura basada en componentes
 - Software modelado visualmente (usar UML estático y dinámico)
 - Verificar la calidad del software (garantizar calidad y estándares)
 - Controlar los cambios del software (procedimientos y herramientas de configuración)

Soporte

- Gestión de cambio y configuraciones (cambios del sistema)
- Gestión del proyecto (administración del proyecto)
- Entorno (herramientas adecuadas al equipo de desarrollo)

Flujo de trabajo del proceso de desarrollo

1. **Fase inicial:**

- 1.1 Inicio del proyecto (Alcance del proyecto)
 - 1.1.1 Objetivo general
 - 1.1.2 Objetivos específicos
 - 1.1.3 Modelado de negocio (BPMN Business Process Model Notation) y/o diagrama del proceso de negocio con marco de responsabilidades (Diagrama de actividades UML)
 - 1.1.4 Concebir el proyecto: Riesgos, requisitos alto nivel y restricciones, modelo de desarrollo de software, plan de cada fase así como sus interacciones.
 - 1.1.5 Formalizar el proyecto (Acta constitutiva o Project Chapter)

2. Elaboración

- 2.1 Definir el sistema (detallar los requisitos)
 - 2.1.1 Especificación de requerimientos del cliente.
 - 2.1.2 Especificación de requerimientos del usuario.
 - 2.1.3 Especificación de requerimientos del Sistema: funcionales y no funcionales.
 - 2.1.4 Documentación de requisitos IEEE830 (casos de uso, requerimientos cliente, usuario y sistema, restricciones, calidad, ...)
 - 2.1.5 Apéndice: Modelado UML2.0 (caso de uso, clases, componentes, paquetes, secuencia, actividad, estado, despliegue, colaboración...)
- 2.2 Diseñar componentes del sistema y subsistemas
 - 2.2.1 Modelo de interacción: Análisis de caso de uso y su diagramas, y diagrama de secuencia.
 - 2.2.2 Modelo estructural: Diagrama de componentes, diagrama de paquetes, diagrama de clases y el diseño de la base de datos.
 - 2.2.3 Modelo comportamiento: Diagrama de procesamiento de pasos y datos, diagrama de eventos o estados.
- 2.3 Definir una arquitectura candidata
 - 2.3.1 Modelo de contexto: Modelo C4 para diagramar arquitectura de software (contexto, contenedor, componentes, código/clases).
 - 2.3.2 Definir el modelo de implementación (Patrones de Arquitectura de software, Modelo-vista-controlador, Capas, Cliente-Servidor, microservicios).