Guía 4.1

Metodología de desarrollo de sistemas I

1. ¿Qué es y para que se utiliza la Ingeniería de Requisitos?

La Ingeniería de Requisitos se convierte en pieza clave para poder medir la calidad de un sistema informático al poder iniciar la definición de la batería de pruebas que el sistema debe pasar, garantizando que éstas satisfacen los requisitos establecidos y por lo tanto el sistema es válido y funcionalmente es correcto.

1. ¿Qué es un requisito de software? ¿Cuáles son las características deseables?

Los requisitos software son la descripción de las características y las funcionalidades del sistema “target”. Los requisitos nos comunican las expectativas de los consumidores de productos software. Los requisitos pueden ser obvios o estar ocultos, conocidos o desconocidos, esperados o inesperados, des del punto de vista del cliente.

Las características deseables en un producto de software son:

Corrección. Que cumpla con su objetivo.

Usabilidad. Que sea fácil de aprender.

Seguridad. Que sea resistente a ataques externo.

Flexibilidad. Que pueda ser modificado por los desarrolladores.

Portabilidad. Que pueda ser utilizado en diversos equipos.

1. ¿Qué es la Elicitación?

La Elicitación se usa, para referirse al traspaso de información de un punto a otro, en forma fluida. La información puede fluir desde un software a otro, de un computador a una persona o de persona a persona. La información posee elicitación si está fluyendo entre los programas se pueden ver y compartir con otros contactos sin interrupciones ni dificultades.

1. Investigue y defina las siguientes técnicas de Elicitación: Entrevistas, Brainstorming, Prototipos, Escenarios, Observación

La **entrevista** es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada.

**Brainstorming**: Consiste en reuniones con cuatro a diez personas donde como primer paso sugieren toda clase de ideas sin juzgar su validez y después de recopilar todas las ideas se realiza un análisis detallado de cada propuesta. Esta técnica se puede utilizar para identificar un primer conjunto de requisitos en aquellos casos donde no están muy claras las necesidades que hay que cubrir, o cuando se está creando un sistema que habilitará un servicio nuevo para la organización.

Los **prototipos** suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Esta técnica es particularmente útil cuando:

El área de la aplicación no está bien definida (posiblemente por ser algo muy novedoso).

El costo del rechazo de la aplicación por los usuarios es muy alto.

Es necesario evaluar previamente el impacto del sistema en los usuarios y en la organización. Los prototipos de sistema permiten a los usuarios experimentar para ver cómo éste ayuda a su trabajo.

**Escenarios**: Estos se utilizan para documentar el comportamiento del sistema cuando se le presentan eventos específicos. Cada evento de interacción distinto, o la selección de un servicio del sistema, se documentan como un escenario de eventos distinto. Los escenarios de eventos incluyen una descripción del flujo de datos y las acciones del sistema, y documenta las excepciones que puedan surgir.

**Observación:** Por medio de esta técnica el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método permite observar la forma en que se llevan a cabo los procesos y, por otro, verificar que realmente se sigan todos los pasos especificados.

1. Definir y explicar los tipos de Requisitos conocidos.

**Requerimientos de usuario:** Son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

**Requerimiento de sistema:** Estos requerimientos establecen con detalle las funciones, servicios y restricciones operativas del sistema.

**Requerimientos funcionales:** Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares. O también pueden declarar explícitamente lo que el sistema no debe hacer.

**Requerimientos no funcionales:** Son **restricciones** de los servicios o **funciones** ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares.

Los requerimientos Funcionales a su vez se clasifican de la siguiente manera:

REQUERIMIENTOS DE PRODUCTO, **ORGANIZACIONALES y EXTERNOS**

1. ¿Qué diferencias existen entre los requisitos funcionales y no funcionales?

Los requisitos funcionales definen la funcionalidad de un software, es decir, qué pueden hacer, mientras que los requisitos no funcionales definen, otras cosas que no son requeridas por el usuario, pero requeridas por un proveedor de servicios o el software en sí, como el registro, es un requisito no funcional para respaldar un aplicación, no utilizada directamente por el usuario, pero esencial para solucionar cualquier problema en un entorno de producción.

1. Indique 3 ejemplos de cada tipo de requisito.

Requerimientos funcionales legales o regulatorios

Requerimientos de seguridad

Requerimientos de interfaces externas (Hardware y Software)

REQUISITOS NO FUNCIONALES:

Algunos ejemplos de requisitos no funcionales típicos son los siguientes:

• Rendimiento

• Disponibilidad

• Durabilidad

• Estabilidad

• Funcionalidad

• Accesibilidad

1. Relacione Requisitos con Casos de uso.

Los Requisitos del sistema están escritos desde la perspectiva del sistema y no en la interacción del usuario, representan las características en estado puro.

Los Casos de Uso están escritos como una serie de interacciones entre el usuario y el sistema. Hacen hincapié en el contexto orientado al usuario. Las características que utiliza cada usuario identificado en el sistema. Son la forma de capturar los requisitos del sistema desde el punto de vista del usuario.

1. ¿Qué es un caso de uso? ¿Para qué se utiliza?

Un caso de uso es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable. Los casos de uso proporcionan una estructura para expresar requisitos funcionales en el contexto de procesos empresariales y de sistema.

1. Explique cómo utilizaría los casos de uso en cada fase del UP

Fase de Inicio

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores o involucrados del proyecto en el cual tenemos que, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

Fase de Elaboración

En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de Desarrollo o Construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo con las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Fase de Transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

1. ¿Qué representa un actor en un caso de uso?

Un actor representa una entidad externa (persona) que interactúa o estimula con el sistema.

1. ¿Los actores deben ser sólo personas? Justifique

No, No necesariamente deben ser personas, pueden ser también otro sistema.

1. ¿Qué diferencia hay entre actores primarios y secundarios?

La diferencia entre estos es que el actor primario es un cliente que utiliza las funciones principales del sistema. Mientras que el actor primario es un administrador/mantenimiento que hace ajustes o mantiene el sistema.

1. ¿Qué es un escenario?

Un escenario es una serie de eventos/acciones entre el actor y el sistema.

1. ¿Qué es una colaboración?

Una colaboración se representa en UML mediante varios diagramas que muestran el contexto (las clases y sus relaciones) y la interacción entre los participantes en la colaboración para conseguir la funcionalidad deseada.

1. ¿Cómo se describe un caso de uso? Detalle sus partes y qué se realiza en cada una de ellas
2. ¿Qué diferencia hay entre el escenario principal y el curso alternativo en los casos de uso?
3. ¿Qué son las pre y post condiciones en la descripción de los casos de uso?

Las precondiciones son validaciones que hace el caso de uso al comienzo del mismo o en algún paso siguiente. Las postcondiciones y el resultado del caso de uso son una misma cosa.

1. ¿Qué representan las relaciones de extensión entre casos de uso? ¿Para qué se utilizan?

La relación de extensión sirve para modelar: la parte opcional del sistema, un subflujo que sólo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado. Este tipo de relación produce confusión y no debería utilizarse en exceso.

1. ¿Qué representan las relaciones de inclusión entre casos de uso? ¿Para qué se utilizan?

La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso, también puede utilizarse para estructurar un caso de uso describiendo sus subfunciones.