**TEMA 1**

**Luca Brandan y Benjamín Núñez 7-3**

* **El análisis funcional en el desarrollo de software es un proceso clave que se centra en definir y documentar las funciones y requisitos que debe cumplir un sistema o aplicación. Su objetivo principal es identificar y entender que es lo que el sistema debe hacer para satisfacer las necesidades del usuario o del negocio. El análisis funcional es esencial para garantizar que el desarrollo del software este alineado con las expectativas del cliente y las necesidades del negocio, lo que a su vez ayuda a entregar un producto final de alta calidad. La importancia del análisis funcional da “Claridad en los requisitos” lo cual permite obtener una visión clara y comprensible de lo que se espera del sistema así evitando futuros problemas. Siguiendo con “Base para el diseño” proporciona una base solida sobre cual se pueden diseñar y desarrollar soluciones técnicas. “Comunicación efectiva” y “Control de calidad” facilitando la comunicación entre los usuarios finales y los demás miembros del equipo al igual que el “Control de Calidad” se utiliza como referencia para la creación de pruebas y validaciones.**
* **Los requerimientos funcionales (FR) representan las funcionalidades o características principales que debe poseer un “Software para cumplir con el propósito previsto, estos requisitos especifican en lo que debe hacer el sistema, describiendo las interacciones entre software y usuario, así como el comportamiento del software bajo diversas condiciones, caracterizando la especificidad, la verificabilidad, centrado en el usuario y su cambiable.**

**Los requerimientos no funcionales (NFR) complementan a los (FR) al especificar como un sistema de software debe realizar ciertas funciones, definen las cualidades, características y limitaciones del sistema más que sus características específicas, los requisitos no funcionales establecen los estándares para el rendimiento, la seguridad y la usabilidad del sistema. Sus características son tal que Cualitativos, Global, estabilidad y son medibles.**

* **Se define a el proceso de elicitación de requerimientos como la fase principal de un ciclo de desarrollo de software enfocada en recopilar y analizar los requerimientos y objetivos deseados para el sistema desde diferentes puntos de vista El proceso consta de varios pasos:**
* **Identificacion de stakeholders**
* **Planificación de la elicitacion**
* **Recopilar de información**
* **Análisis de requerimientos**
* **Validación**
* **Documentación**

**Y Las técnicas para la elicitación de requerimientos son las siguientes:**

* **Entrevistas**
* **Encuestas y Cuestionarios**
* **Talleres y Sesiones de Grupo**
* **Observacion**
* **Analisis de documentos**
* **Prototipos**
* **Casos de uso**
* **Historias de usuario**
* **Analisis de procesos de negocio**
* **Un caso de uso es un tipo de técnica para entender y describir requisitos. En el análisis funcional, los casos de uso son requisitos, describen requisitos funcionales, Describen como el sistema debe comportarse desde el punto de vista del usuario**
* **La importancia de la priorización de requerimientos es crucial en un proyecto de software por que**
* **Optimiza los recursos**
* **Maximiza el valor del negocio**
* **Gestion de riesgo**
* **Flexibilidad en el Alcance**
* **Satisface al cliente**

**Tema 2: Metodologias agiles**

* **Las metodologías agiles y las metodologías tradicionales representan dos enfoques diferentes para la gestión de proyectos**

**Metodologías agiles: es un enfoque que se centra en ser flexible la colaboración y entrega continua de valor. Estas metodologías enfatizan la adaptación a los cambios y la mejora continua**

**Metodologías tradicionales: se basan en un enfoque lineal y secuencial para el desarrollo de software, estas siguen un proceso rigido en el que cada fase debe contemplarse antes de pasar a la siguiente**

* **1.individuos e interacciones sobre procesos y herramientas**

**2. Software funcionando sobre documentación extensiva**

**3. Colaboración con el cliente sobre negociación contractual**

**4. respuesta ante el cambio sobre seguir un plan**

**8. ¿Qué es Scrum y describe tres de sus elementos principales?**

**Scrum es un marco de trabajo dentro de las metodologías ágiles que se utiliza para gestionar proyectos complejos, especialmente en el desarrollo de software. Se basa en ciclos iterativos y adaptativos, llamados sprints, para entregar productos de forma incremental. Scrum promueve la colaboración en equipo, la adaptabilidad y la entrega continua de valor.**

**- Sprint:Es un ciclo de trabajo corto y predefinido, generalmente de 1 a 4 semanas, en el que el equipo de Scrum desarrolla una parte funcional del producto. Al final de cada sprint, se presenta un incremento del producto.**

**- Product Backlog: Es una lista priorizada de todas las tareas, características y requisitos que se necesitan para el producto. El Product Owneres responsable de gestionar y priorizar esta lista.**

**- Scrum Master: Es el facilitador del equipo Scrum. Su función es asegurarse de que el equipo siga los principios y prácticas de Scrum, eliminando obstáculos que puedan interferir con el progreso y facilitando la comunicación dentro del equipo.**

**9. ¿Qué es un "sprint" en el contexto de las metodologías ágiles y cuál es su propósito?**

**Un sprint es un ciclo de trabajo iterativo y de duración fija (generalmente entre 1 y 4 semanas) en las metodologías ágiles, particularmente en Scrum. Durante el sprint, el equipo trabaja en un conjunto definido de tareas o "historias de usuario" extraídas del Product Backlog con el objetivo de entregar un incremento funcional del producto.**

**Propósito del sprint:**

**- Permitir la entrega continua de valor en partes pequeñas, incrementales y funcionales.**

**- Facilitar la inspección y adaptación mediante retrospectivas y revisiones al final de cada sprint.**

**- Mejorar la capacidad del equipo para responder a cambios y adaptarse a las necesidades del cliente.**

**10.¿Qué es una "historia de usuario" y cómo se utiliza en el desarrollo ágil de software?**

**Una historia de usuario es una descripción simple y breve de una funcionalidad desde la perspectiva del usuario final o cliente. Se utiliza en el desarrollo ágil de software para capturar los requisitos y necesidades del cliente de manera comprensible.**

**Cada historia de usuario sigue una estructura común:**

**- Como [tipo de usuario], quiero [realizar alguna acción] para [obtener un beneficio o valor].**

**Las historias de usuario se priorizan en el Product Backlog y se seleccionan para ser trabajadas durante los sprints. Sirven como unidad básica de trabajo para el equipo ágil, facilitando la entrega de valor al cliente de forma continua.**

**11. ¿Qué es el ciclo de vida del software y cuáles son sus fases principales?**

**El ciclo de vida del software es el conjunto de fases por las que pasa un software desde su concepción hasta su retiro. Incluye el desarrollo, implementación, mantenimiento y actualización del software. El objetivo es gestionar y organizar el desarrollo del software de manera eficiente.**

**Fases principales del ciclo de vida del software:**

**1.Análisis de requisitos: Se identifican las necesidades y expectativas del cliente.**

**2. Diseño: Se definen las arquitecturas y especificaciones técnicas del software.**

**3. Desarrollo: Se escribe el código y se desarrolla el producto.**

**4. Pruebas: Se verifica y valida el software para asegurar su funcionamiento correcto.**

**5. Implementación: Se despliega el software en el entorno de producción.**

**6.Mantenimiento: Se corrigen errores, se hacen mejoras y se mantiene el software actualizado.**

**12. ¿Cuál es la diferencia entre el modelo en cascada y el modelo iterativo de desarrollo de software?**

**Modelo en cascada: Es un enfoque lineal y secuencial del desarrollo de software donde cada fase debe completarse antes de que comience la siguiente. No hay lugar para retrocesos, y los cambios son difíciles de implementar una vez que se ha avanzado en el ciclo.**

**- Modelo iterativo: Es un enfoque en el que el software se desarrolla a través de iteraciones (ciclos). Cada iteración produce una versión parcial o completa del producto que se evalúa, mejora y se vuelve a trabajar en las siguientes iteraciones. Esto permite adaptarse a cambios y mejora continua a lo largo del desarrollo.**

**Diferencia clave: El modelo en cascada es más rígido y secuencial, mientras que el modelo iterativo es más flexible y permite mejoras continuas durante el proceso.**

**13. Describe la fase de pruebas en el ciclo de vida del software. ¿Por qué es crucial esta fase?**

**La fase de pruebas en el ciclo de vida del software es el proceso mediante el cual se verifica y valida que el software funciona correctamente y cumple con los requisitos establecidos. Durante esta fase, se detectan errores, se evalúa la calidad del software y se asegura que el producto sea confiable y seguro antes de su entrega.**

**Por qué es crucial:**

**- Detecta errores antes de que el software sea implementado, reduciendo los costos de reparación de fallos.**

**- Asegura que el software cumple con los requisitos del cliente y satisface sus expectativas.**

**- Garantiza la estabilidad y seguridad del sistema, evitando fallos que puedan afectar al usuario final o al negocio.**

**14. ¿Qué es el mantenimiento del software y por qué es importante considerarlo como parte del ciclo de vida?\*\***

**El mantenimiento del software es el proceso de realizar modificaciones y mejoras en el software después de su implementación. Esto incluye corrección de errores, mejoras de rendimiento, actualización de características y adaptaciones a nuevas tecnologías o requisitos.**

**Es importante porque:**

**- El software debe adaptarse a los cambios en el entorno tecnológico o en las necesidades del cliente.**

**- El mantenimiento prolonga la vida útil del software, asegurando que siga siendo funcional y útil.**

**- Es crucial para corregir problemas que no fueron detectados durante la fase de pruebas o que surgieron después de la implementación.**

**15. ¿Qué es la fase de análisis de requisitos en el ciclo de vida del software y por qué es fundamental para el éxito del proyecto?**

**La fase de análisis de requisitos es la etapa del ciclo de vida del software donde se recopilan, analizan y documentan las necesidades del cliente o usuario final. En esta fase se define lo que el sistema debe hacer, cómo debe comportarse y qué restricciones debe cumplir.**

**Es fundamental porque:**

**- Proporciona una base clara y detallada sobre lo que se espera del software.**

**- Ayuda a evitar malentendidos y errores durante el desarrollo, lo que reduce costos y tiempos de corrección.**

**- Es el punto de partida para las fases posteriores de diseño y desarrollo, garantizando que el equipo trabaje alineado con las expectativas del cliente.**