P8. Fase de Validación de Requerimientos

La **validación de requerimientos** es una etapa fundamental en el desarrollo de software, ya que garantiza que los requisitos definidos sean correctos, completos y alineados con las necesidades del cliente y los objetivos del proyecto. Su propósito principal es evitar errores o malentendidos que puedan generar problemas en fases posteriores del desarrollo.

Objetivos de la Validación de Requerimientos

- **Confirmar la exactitud de los requerimientos**: Asegurar que reflejan las necesidades reales del usuario.
- **Verificar la viabilidad**: Evaluar si los requerimientos son alcanzables desde el punto de vista técnico y económico.
- **Eliminar ambigüedades**: Identificar y corregir inconsistencias, omisiones o malentendidos en la especificación de requerimientos.
- **Asegurar la consistencia y completitud**: Revisar que los requerimientos no se contradigan entre sí y que cubran todas las funcionalidades esperadas.

Métodos de Validación de Requerimientos

Para asegurar que los requerimientos son correctos y completos, se pueden aplicar diversas técnicas, como:

1. **Revisión de requerimientos**: Un grupo de expertos (stakeholders, analistas, desarrolladores) revisa y analiza los requerimientos para detectar errores o inconsistencias.
2. **Prototipado**: Se crean prototipos o modelos funcionales para validar con los usuarios si los requerimientos cumplen con sus expectativas.
3. **Casos de uso y escenarios**: Se definen ejemplos prácticos y flujos de trabajo para comprobar si los requerimientos se adaptan a las necesidades del usuario.
4. **Matrices de trazabilidad**: Se usan para relacionar los requerimientos con sus respectivas validaciones y garantizar que todos sean cubiertos en el desarrollo.
5. **Entrevistas y encuestas con usuarios**: Se recaba feedback directo de los interesados para asegurar que los requerimientos sean claros y completos.

Importancia de la Validación de Requerimientos

Una validación efectiva evita errores costosos en fases avanzadas del desarrollo, reduce la necesidad de cambios posteriores y mejora la satisfacción del cliente. Además, ayuda a prevenir fallas en el software, asegurando que se cumplan los objetivos del negocio.

En conclusión, la validación de requerimientos es un paso crítico en el proceso de desarrollo de software, ya que garantiza la calidad, viabilidad y alineación de los requerimientos con las expectativas del usuario y las capacidades del equipo de desarrollo.

P9. La **verificación de validez** es un proceso clave en la **validación de requerimientos** dentro del desarrollo de software. Su objetivo es garantizar que los requerimientos definidos sean

correctos, completos, consistentes y realizables antes de proceder con el diseño y desarrollo del sistema.

🔍 **Aspectos clave en la verificación de validez**

1. **Corrección**: Se verifica que los requerimientos reflejan correctamente las necesidades del cliente.
2. **Consistencia**: Se evalúa que no haya contradicciones entre los requerimientos.
3. **Completitud**: Se revisa que todos los requerimientos necesarios estén documentados.
4. **Viabilidad**: Se asegura que los requerimientos sean técnicamente y económicamente factibles.
5. **Trazabilidad**: Se comprueba que cada requerimiento pueda ser rastreado a lo largo del desarrollo.

✅ **Métodos para la verificación de validez**

- **Revisión por pares**: Expertos revisan los requerimientos para detectar errores.
- **Prototipado**: Se crean modelos iniciales para evaluar la viabilidad.
- **Casos de uso y escenarios**: Se simulan situaciones de uso real del software.
- **Listas de verificación**: Se utilizan checklists para asegurar que todos los criterios se cumplen.

P10. Requerimientos Duraderos en Ingeniería de Software

Los **requerimientos duraderos** (también llamados **requerimientos estables o persistentes**) son aquellos que **se mantienen constantes a lo largo del tiempo** en el desarrollo de un sistema, independientemente de los cambios tecnológicos, normativos o de negocio.

🔍 **Características de los Requerimientos Duraderos**

✓ **Independencia tecnológica**: No dependen de una plataforma, lenguaje o tecnología específica.
✓ **Base del negocio**: Reflejan principios fundamentales del dominio del negocio.
✓ **Poco afectados por cambios**: Son menos propensos a modificaciones por actualizaciones en hardware o software.
✓ **Generalmente no funcionales**: Suelen estar relacionados con la seguridad, confiabilidad, accesibilidad, etc.

📌 **Ejemplos de Requerimientos Duraderos**

1. **Seguridad**: "El sistema debe garantizar la autenticación segura de los usuarios."
2. **Regulaciones y normativas**: "El sistema debe cumplir con la ley de protección de datos vigente."

3. **Accesibilidad**: "La plataforma debe ser accesible para personas con discapacidades visuales."
4. **Integridad de datos**: "El sistema debe mantener la integridad de los datos en todas las transacciones."
5. **Usabilidad**: "El software debe ser intuitivo para usuarios sin formación técnica."

🔄 Requerimientos Duraderos vs. Requerimientos Volátiles

Característica	Requerimientos Duraderos	Requerimientos Volátiles
Cambio con el tiempo	Poco o nada	Frecuente
Dependencia tecnológica	No	Sí
Ejemplo	Seguridad, cumplimiento legal	Interfaz de usuario, compatibilidad con nuevas tecnologías
Impacto en el sistema	Estructural	Adaptativo

P11. Requerimientos Volátiles en Ingeniería de Software

Los **requerimientos volátiles** son aquellos que pueden **cambiar, evolucionar o desaparecer** a lo largo del ciclo de vida del software. Estos cambios pueden deberse a nuevas necesidades del usuario, avances tecnológicos, cambios en el negocio o normativas actualizadas.

🔍 Características de los Requerimientos Volátiles

✓ **Propensos a cambios**: Se modifican o desaparecen con el tiempo.
✓ **Dependientes del contexto**: Afectados por cambios en la tecnología, negocio o regulaciones.
✓ **Mayor riesgo en el desarrollo**: Pueden causar retrasos o costos adicionales si no se gestionan bien.
✓ **Frecuentemente funcionales**: Suelen estar relacionados con la interfaz de usuario, funcionalidades específicas, etc.

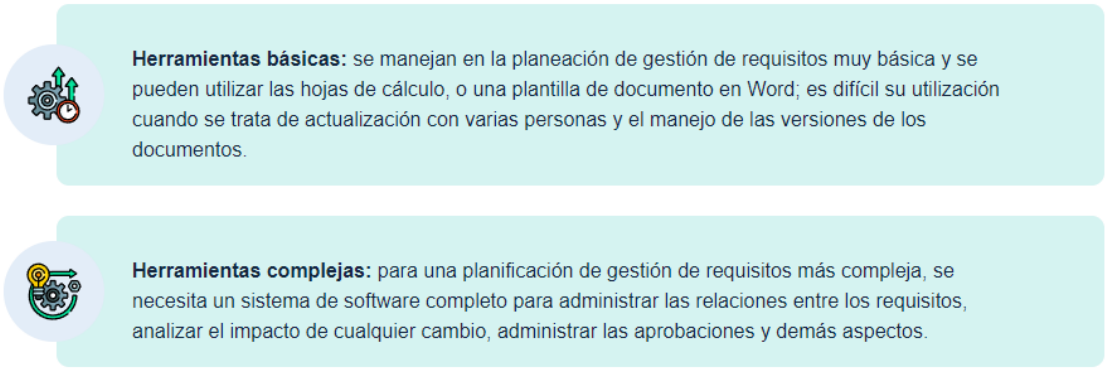
📌 Tipos de Requerimientos Volátiles

1. **Evolutivos**: Se basan en el negocio y evolucionan con el tiempo.
    - *Ejemplo:* "El sistema debe soportar pagos con criptomonedas en el futuro."
2. **Emergentes**: Surgen a medida que avanza el desarrollo del software.
    - *Ejemplo:* "Se requiere una nueva funcionalidad para la gestión de reportes en tiempo real."
3. **Consecuentes**: Aparecen como resultado de la implementación de otros requerimientos.

- *Ejemplo:* "Debido a la integración con un nuevo sistema, se requiere autenticación en dos pasos."

4. **Compatibilidad**: Se derivan de cambios tecnológicos o de hardware.
    - *Ejemplo:* "El sistema debe ser compatible con la última versión de iOS y Android."
5. **Regulatorios**: Cambian según nuevas normativas o leyes.
    - *Ejemplo:* "El software debe adaptarse a la nueva ley de protección de datos."

P12. La gestión de requisitos es fundamental en el desarrollo de software, ya que permite documentar, analizar, priorizar y rastrear los requisitos a lo largo del ciclo de vida del proyecto. Existen diversas herramientas que ayudan a facilitar esta tarea.



Video: Instalación del software REM

https://www.youtube.com/watch?v=67ZLZQtBFBg

Video: Manejo de software REM

https://www.youtube.com/watch?v=p-3qfU1OZb8

Sigue la presentación…

P.25 En esta introducción se presenta una explicación sobre la importancia de la **validación de requerimientos** en el desarrollo de software y otros proyectos. Destaca que antes de escribir código o iniciar una obra, es fundamental identificar si los requerimientos realmente son necesarios para el cliente y el sistema de información.

🔷 **Punto clave**: La validación de requerimientos asegura que el proyecto cumpla con las necesidades reales y evita retrabajos.

🔷 **Dato adicional**: Según Sommerville (2011), esta revisión puede ser **formal o informal**, lo que permite adaptarla a diferentes metodologías de desarrollo.

P26. En la **validación de requerimientos**, se verifica que estos sean correctos, completos y útiles para el desarrollo del sistema. Algunos aspectos clave que se validan incluyen:

🔍 **1. Corrección**

- ¿El requerimiento refleja fielmente la necesidad del usuario?
- ¿Es técnicamente posible de implementar?

📌 **2. Compleción**

- ¿Se han identificado todos los requerimientos esenciales?
- ¿No hay ambigüedades o lagunas en la especificación?

📝 **3. Consistencia**

- ¿Los requerimientos no se contradicen entre sí?
- ¿Son compatibles con las restricciones del sistema?

📊 **4. Viabilidad**

- ¿Son realistas dentro de los recursos disponibles (tiempo, costos, tecnología)?
- ¿Se pueden implementar con la tecnología actual?

🔄 **5. Verificabilidad**

- ¿Cada requerimiento puede ser probado o medido?
- ¿Existen criterios claros para evaluar su cumplimiento?

👥 **6. Alineación con las necesidades del cliente**

- ¿El requerimiento realmente aporta valor al usuario final?
- ¿Se ha validado con los interesados para asegurarse de su relevancia?

P27. Construcción de prototipos

Podría ser una sencilla representación gráfica del sistema en papel o la representación de interfaces del sistema usando herramientas de ofimática, existen varias herramientas gratuitas y de pago que facilitan la construcción de prototipos, algunas de ellas solo requieren un registro de correo electrónico para acceder a sus funcionalidades directamente desde el navegador, y otras requieren la descarga e instalación en el equipo donde se planean construir los prototipos.

P29. Generación de casos pruebas

La validación de requisitos por casos de prueba requiere el desarrollo de cuatros fases: i) **planeación de la prueba**. Aquí se define la estrategia a utilizar, el alcance de la prueba a realizar y los tiempos requeridos para el desarrollo de la prueba; ii) **diseño de los casos de prueba**; iii) **ejecución de los casos de prueba** y iv) **elaboración del informe final de la prueba** donde se describen los aspectos más importantes y hallazgos de la ejecución de la prueba.

Video: Aprende a utilizar balsamiq Mockups paso a paso con ejemplos

https://www.youtube.com/watch?v=6vXjEd6BWgo&t=144s