Metodología clase 5/08

Fecha: 5/08

💡***Ética en la ingenieria de software***

La etica en la ingenieria de software debe tener una confidencialidad a la cual debe respetar la info del cliente. Debe tener una competencia, no aceptar proyectos duera de nuestra experiencia. Propiedad intelectual debe de respetar derechos de autor y licencias. Se debe hacer un uso adecuado de computadoras para evitar acciones malintencionadas.

▼ Conclusiones:

La ingenieria de softwarre permite el desarrollo eficiente y profesional.

Aplicar metodología adecuadas mejora la calidad y confiabilidad.

La ética y la responsabilidad son claves en el desarrolo del software.

😜Explicación detallada con ejemplos

1. Confidencialidad en la ingeniería de software:

La confidencialidad implica proteger la información sensible del cliente y no divulgarla sin autorización.

Ejemplo: Un desarrollador que trabaja con datos financieros de una empresa debe firmar acuerdos de confidencialidad y no compartir información sobre algoritmos propietarios, incluso en conversaciones informales con colegas de otras compañías.

2. Competencia profesional:

Consiste en aceptar solo proyectos dentro de nuestra área de experiencia y capacidad técnica.

Ejemplo: Un desarrollador especializado en aplicaciones web rechaza un proyecto de sistemas embebidos críticos para equipos médicos, ya que no tiene la experiencia necesaria, y recomienda a otro profesional más capacitado para esa tarea.

3. Respeto a la propiedad intelectual:

Implica respetar los derechos de autor, patentes y licencias de software.

Ejemplo: Un equipo de desarrollo utiliza bibliotecas de código abierto en su proyecto, pero se asegura de cumplir con las condiciones de la licencia (como MIT o GPL), dando crédito apropiado y manteniendo las notificaciones de copyright.

4. Uso responsable de recursos computacionales:

Consiste en evitar el uso indebido de los sistemas informáticos.

Ejemplo: Un ingeniero detecta una vulnerabilidad en el sistema de un cliente pero, en lugar de explotarla, documenta el problema y notifica al cliente para que pueda corregirlo, siguiendo protocolos éticos de divulgación responsable.

Importancia de estos principios éticos

Estos principios éticos no solo protegen a los clientes y usuarios, sino que también construyen confianza en la profesión y contribuyen a la creación de software más seguro y confiable. En un mundo cada vez más dependiente de la tecnología, la ética en la ingeniería de software se vuelve fundamental para proteger el bienestar de la sociedad.

🤔¿Que es un proceso de software?

Es un conjunto de actividades organizadas para la produccion de software profesional.

▼ **Actividades esenciales**

🧩Especificación: Define requisitos y restricciones.

🧩Diseño e implementación: desarrollo del software.

🧩Validación: pruebas y aseguramineto de calidad.

🧩Evolución: adaptacion a cambios.

Ejemplos de actividades en cada fase:

Especificación:

- Entrevistas con stakeholders para recopilar requisitos
- Creación de documentos de especificación de requisitos (SRS)
- Desarrollo de casos de uso y historias de usuario

Diseño e implementación:

- Creación de diagramas de arquitectura del sistema
- Diseño de bases de datos y modelos de datos
- Programación y desarrollo de código fuente

Validación:

- Pruebas unitarias y de integración
- Revisiones de código entre pares
- Pruebas de aceptación con usuarios finales

Evolución:

- Implementación de nuevas funcionalidades solicitadas
- Corrección de errores y optimización de rendimiento
- Actualización de tecnologías y plataformas

💡Productos, roles y condiciones

- Producto: Documento de requerimiento, código, pruebas manuales, unit test, diagramas, uml, etc.

- Roles: programador, gerente de proyecto, analista, tester (QA), diseñador UX/UI, etc.

▼ Pre-condiciones (ejemplos):

Requisitos aprobados del diseño.

Pruebas unitarias antes de pruebas de integración.

Diagrama de base de datos antes de crear código fuente.

▼ Post-Condiciones:

Validación del cliente tras prueba de integración aprobadas.

Despliegue en servidor para aprobación del cliente.

💡Modelos de procesos de software

Diferentes enfoques para estructurar el desarrollo de proyectos de software.

▼ Modelos

Cascadas.

Incremental

Basado de reutilización de componentes.

Espiral

RUP

1. Modelo en Cascada:

Es un enfoque lineal y secuencial donde cada fase debe completarse antes de pasar a la siguiente. Las etapas típicas son: requisitos, diseño, implementación, verificación y mantenimiento. No permite retroceder fácilmente a fases anteriores.

2. Modelo Incremental:

Divide el proyecto en incrementos o mini-proyectos, cada uno siguiendo una secuencia de requisitos, diseño, implementación y pruebas. Permite entregar funcionalidad útil al cliente de forma progresiva mientras se desarrollan las demás partes.

3.Modelo Basado en Reutilización de Componentes:

Consiste en construir software a partir de componentes previamente desarrollados. Se enfoca en ahorrar tiempo y mejorar la calidad mediante la integración de módulos existentes, como librerías o frameworks.

4.Modelo Espiral:

Es un modelo de desarrollo iterativo que se centra en la evaluación y gestión de riesgos. Cada ciclo incluye planificación, análisis de riesgos, diseño, implementación y evaluación, permitiendo adaptar el sistema según nuevos requerimientos.

5.RUP (Rational Unified Process):

Es un proceso de desarrollo estructurado en fases: inicio, elaboración, construcción y transición. Utiliza casos de uso como base y promueve buenas prácticas como la documentación detallada, control de versiones y pruebas constantes.