



lunes, 07 de octubre de 2024

Desarrollo de Medios Digitales – Actividad 8 – Repaso sobre Ingeniería de software.

Instrucciones: Lea los capítulos 1, 2 y 3 del libro “Ingeniería de software” de Sommerville y realice un resumen de los conceptos más relevantes que encuentre.

Modalidad de Entrega: Genere la entrega de su documento al espacio de Teams indicado:

[Canal General > Archivos > Actividad 8](#)

Fecha límite: miércoles, 9 de octubre. 11:55am.

1. Resumen y conceptos importantes del capítulo 1 “Introducción a la ingeniería de software”

*El **capítulo 1** de "Ingeniería de Software" de Sommerville introduce a los lectores en el concepto de la ingeniería de software profesional, destacando su relevancia en el mundo moderno. Se enfatiza que la ingeniería de software no solo se ocupa de la creación de programas, sino también de la documentación y el mantenimiento a largo plazo de los sistemas.*

Conceptos principales:

1. **Ingeniería de software:** Se define como una disciplina que abarca todos los aspectos del desarrollo y producción de software, desde la especificación inicial hasta el mantenimiento post-desarrollo.

2. **Productos de software:** Se clasifican en **productos genéricos**, como sistemas comerciales, y **productos personalizados**, diseñados específicamente para un cliente. Ambos deben cumplir con requisitos clave como la confiabilidad, eficiencia y seguridad.
3. **Atributos de un buen software:** Estos incluyen la **mantenibilidad**, la **confiabilidad**, la **eficiencia** y la **aceptabilidad**. Es esencial que el software se adapte a las necesidades cambiantes de los usuarios.
4. **Actividades fundamentales:** Se destacan la especificación, el desarrollo, la validación y la evolución del software como procesos cruciales para cualquier sistema.
5. **Conflictos éticos y profesionales:** Los ingenieros de software deben tener en cuenta las implicaciones sociales y éticas de su trabajo, actuando con responsabilidad tanto hacia sus clientes como hacia la sociedad en general .

2. Resumen y conceptos importantes del capítulo 2 "Procesos de software"

El **capítulo 2**, "Procesos de Software", presenta la idea central de que un proceso de software es una serie de actividades organizadas que culminan en la creación de un producto de software. Se destacan varias actividades esenciales, como la especificación, el desarrollo, la validación y la evolución del software.

Conceptos principales:

1. **Modelos de procesos de software:**
 - **Modelo en cascada (Waterfall):** Se trata de un enfoque secuencial donde las actividades de desarrollo, especificación, diseño y pruebas se realizan en fases separadas.
 - **Desarrollo incremental:** Aquí el sistema se desarrolla en etapas o versiones, agregando funcionalidades con cada incremento.
 - **Ingeniería de software orientada a la reutilización:** El software se construye reutilizando componentes ya existentes, en lugar de desarrollarlo desde cero.
2. **Actividades del proceso:**
 - **Especificación:** Define tanto las funcionalidades como las restricciones del software.
 - **Diseño e implementación:** El software se desarrolla de acuerdo con las especificaciones.
 - **Validación:** Garantiza que el software cumple con las necesidades del cliente.
 - **Evolución:** A medida que cambian las necesidades del cliente, el software debe ser actualizado y adaptado.
3. **Enfrentar el cambio:** Los procesos deben ser flexibles para adaptarse a los cambios, que pueden surgir durante el desarrollo o incluso después de que el

software se haya entregado. Las estrategias como el desarrollo iterativo y el prototipado ayudan a gestionar mejor estos cambios.

4. **Proceso Unificado Racional (RUP):** RUP es un marco moderno que estructura el desarrollo de software en fases (concepción, elaboración, construcción y transición), permitiendo un enfoque iterativo y adaptable, integrando buenas prácticas a lo largo del ciclo de vida del software

3. Resumen y conceptos importantes del capítulo 3 “Desarrollo ágil de software”

El capítulo 3, "Desarrollo ágil de software", explora los principios y métodos detrás del desarrollo ágil, un enfoque que ha ganado popularidad por su capacidad para adaptarse a entornos de rápido cambio y la necesidad de entregar software de manera rápida y continua.

Conceptos principales:

1. Métodos ágiles:

Los métodos ágiles se basan en un desarrollo incremental y continuo, lo que permite que los equipos entreguen versiones operativas del software de manera frecuente. Esto facilita la retroalimentación rápida y constante por parte de los usuarios.

2. El Manifiesto Ágil:

El manifiesto destaca valores como la prioridad de los individuos y las interacciones sobre los procesos y herramientas, el software funcionando sobre la documentación extensa, la colaboración con el cliente sobre la negociación contractual y la capacidad de respuesta al cambio sobre seguir un plan estricto.

3. Programación Extrema (XP):

La programación extrema es uno de los métodos ágiles más conocidos y se enfoca en prácticas como las pruebas automatizadas antes de escribir código, la simplicidad en el diseño y la entrega continua de software. Involucra al cliente en el equipo de desarrollo para obtener retroalimentación continua.

4. Scrum:

Scrum es un marco ágil que organiza el desarrollo en sprints cortos, típicamente de 2 a 4 semanas, donde el equipo trabaja en un conjunto específico de tareas priorizadas. Al final de cada sprint, se presenta una versión funcional del software al cliente para su evaluación y ajuste.

5. Escalabilidad de los métodos ágiles:

Aunque los métodos ágiles son ideales para pequeños equipos y proyectos, enfrentan desafíos cuando se implementan en proyectos de gran escala debido a la necesidad de más diseño frontal y documentación para coordinar grandes equipos.

Estos conceptos hacen que el desarrollo ágil sea flexible y adaptado a entornos donde los requerimientos cambian rápidamente, pero también puede presentar dificultades para proyectos grandes o sistemas críticos donde se requiere una mayor previsión.