



ACTIVIDAD DE APRENDIZAJE 1 PARCIAL No. 1

METODOLOGIAS DE DESARROLLO EN SOFTWARE

METODOLOGIAS DE DESARROLLO EN SOFTWARE
ÁREA DE COMPUTACION
DEPARTAMENTO DE ITIN

Actividad de aprendizaje -C1.

Nombres del estudiante: Cristian Jesus Becerra Loaiza

Nivel: Tercer Nivel

NRC: 29022

Asignatura: Met. de Desarrollo en Software

Nombre del profesor: Ing. Jenny Alexandra Ruiz Robalino

METODOLOGIAS DE DESARROLLO EN SOFTWARE
ÁREA DE COMPUTACION
DEPARTAMENTO DE ITIN

Contenido

Actividad de aprendizaje 1.....	2
Descripción de la actividad.	2
Preguntas:	2
Referencias.....	4

Actividad de aprendizaje 1.

Descripción de la actividad.

En el capítulo 1 de Fundamentos de la Ingeniería de Software, se aborda una introducción al origen, evolución y diferencias con las ciencias de la computación y otras, acorde a un estudio de la ACM de 2020.

Resuelva las diez preguntas que se muestran a continuación, donde las definiciones, características y diferencias con las otras ciencias se presentan. Puede encontrar material de apoyo en el siguiente enlace:

Preguntas:

1. ¿Qué es ingeniería de software?

La ingeniería de software consiste en la aplicación de la ingeniería a los programas informáticos. No es solo cuestión de programar, sino también de hacerlo con calidad, organización y teniendo en mente que el software sea fácil de mantener, confiable y útil. (Potts, 1993; Bauer, 1968, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

2. ¿Cuáles son las actividades fundamentales de la ingeniería de software?

Las actividades principales que comprendí son cuatro: primero se especifica qué se quiere lograr con el software, después se diseña y programa, luego se valida para asegurar que funcione bien, y finalmente se da mantenimiento para que siga siendo útil con el tiempo. (Sommerville, 2011).

3. ¿Cuál es la diferencia entre ingeniería de software y ciencias de la computación?

La ciencia de la computación se enfoca más en la teoría y en los algoritmos, mientras que la ingeniería de software busca crear programas prácticos que resuelvan problemas reales con calidad. (ACM, AIS & IEEE-CS, 2020, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

4. ¿Qué es software?

El software son los programas que usamos en las computadoras junto con la documentación que los acompaña. Puede ser desarrollado para un cliente específico o para el mercado en general. (Sommerville, 2011).

5. ¿Cuáles son los atributos del buen software?

Un buen software no solo debe cumplir con lo que promete, también debe ser confiable, fácil de usar y mantener, además de rendir bien y no fallar con facilidad. (Sommerville, 2011).

6. ¿Cuáles son los principales retos que enfrenta la ingeniería de software?

Los principales retos son la diversidad de sistemas que se necesitan, la presión por entregarlos rápido y la exigencia de que sean seguros y confiables. También es un desafío adaptarse a metodologías nuevas como las ágiles. (Oktaba, 2018; Sommerville, 2011, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

7. ¿Cuáles son los costos de la ingeniería de software?

Gran parte del costo está en la construcción del software, pero también mucho en las pruebas. Como se especifica que el desarrollo de software puede alcanzar el 60% y su diferencia la alcanzaría las pruebas en lo que son costos. Sin embargo en proyectos personalizados a medida, el mantenimiento y las actualizaciones pueden costar incluso más que el desarrollo inicial. (Sommerville, 2011).

8. ¿Cuáles son los mejores métodos y técnicas de la ingeniería de software?

No existe un único método, depende del tipo de software. En la historia surgieron la programación estructurada y la orientación a objetos, y hoy predominan las metodologías ágiles e híbridas que combinan flexibilidad y formalidad. (Piattini, 2016; Sommerville, 2011, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

9. ¿Qué entiende por paradigmas de la ingeniería de software?

Los paradigmas son los enfoques de desarrollo de software que han ido evolucionando: desde codificar y corregir, hasta la programación estructurada, orientada a objetos y actualmente los métodos ágiles e híbridos. (Piattini, 2016; Hinojosa, 2019, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

10. ¿Cuál es la definición del modelo incremental versus otros paradigmas?

Características y diferencias con otras ingenierías.

El modelo incremental consiste en ir construyendo el software poco a poco, entregando versiones funcionales en cada etapa. Es más flexible que el modelo en cascada, que es rígido y no permite cambios. (Sommerville, 2011; Charette, 2005, citado en Fundamentos de la Ingeniería de Software, Cap. 1, PDF).

Referencias

- ACM, AIS & IEEE-CS. (2020). Computing Curricula 2020.
- Bauer, F. (1968). Software Engineering. NATO Software Engineering Conference.
- Charette, R. (2005). Why Software Fails. IEEE Spectrum.
- Hinojosa, L. (2019). Historia de la ingeniería del software.
- Oktaba, H. (2018). The Future of Software Engineering.
- Piattini, M. (2016). Ingeniería del Software.
- Potts, C. (1993). Software Engineering: Theory and Practice.
- Sommerville, I. (2011). Software Engineering (9th Edition). Pearson.