

Relación de problemas 2 (1, 2 ,3 y 4)

Santiago Muñoz Castro

1. ¿Cuáles crees que son los tres principales problemas en el desarrollo de software?

El principal problema es que haya un mal funcionamiento del software o que deje al usuario insatisfecho porque no cumpla con las expectativas propuestas por este. La dificultad que conlleva mantener el volumen de software existente es otro problema que obliga a realizar un constante mantenimiento del sistema. El último problema puede darse debido a que surjan nuevas tecnologías o nuevos competidores que obliguen a la adaptación constante y por ello a un aumento de la complejidad del desarrollo.

2. ¿Por qué crees que es importante la calidad en el proceso de desarrollo de software?

Una mala calidad conlleva a un mal funcionamiento del software, hay que tener un buen control de la calidad durante todo el proceso de desarrollo, revisando todo lo que se va obteniendo junto con pruebas de código. De esta forma nos aseguramos de que al final del proceso de desarrollo el número de errores sea los más mínimo y que el funcionamiento sea correcto.

El control de calidad toma tanta importancia debido a que controla todos los aspectos del software, estos son, la funcionalidad, el rendimiento, la compatibilidad, la usabilidad, la fiabilidad, la seguridad, la mantenibilidad y la portabilidad.

3. ¿En qué se diferencian el modelo en cascada de los modelos iterativos e incrementales?

La principal diferencia es que en el modelo en cascada se puede saltar entre etapas con total libertad, es decir se puede estar en la etapa de mantenimiento y pasar a la de diseño directamente, en cambio, en el modelo iterativo no se pueden realizar estos saltos, sino que se tiene que seguir el ciclo evaluación, requerimiento, análisis e implementación y prueba, pudiendo obtener en cada una el producto ejecutable.

4. ¿Por qué crees que la documentación forma parte del software?

Porque en este se va a tener todo el registro de todo lo que ocurrió en la implementación del software. Es muy importante para retroalimentar al resto de programadores, por ejemplo, si hay una función muy compleja es mejor tener explicada porque ha surgido y que va a hacer para que en un futuro, cuando alguien tenga que volver al punto de inicio de la función (puede ser el mismo que la programo) no tenga que perder tiempo en entender por qué y el funcionamiento de esta.