



Universidad del cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Programa de Ingeniería de Sistemas

Ingeniería de software I

Taller 2- Preguntas sobre el producto software

Nombre: Jorge Ortiz, fecha 17 de noviembre de 2021

Teniendo en cuenta el capítulo 1 del libro de Pressman (Guía de la Materia) [1], titulado "El Producto", intenta

responder de forma razonada a las preguntas que se formulan a continuación:

Preguntas generales.

Considerando el software desde el punto de vista del producto, se trata de justificar las respuestas Verdadero o Falso de las siguientes preguntas:

1. El software es un producto y se puede manufacturar utilizando la misma tecnología empleada para otros artefactos de la ingeniería.

R./ Falso. El software se produce con el intelecto, aunque claro que las manos tocan el teclado, pero uno podría dictar los comandos y se obtendría el mismo resultado.

2. El software termina fallando con el tiempo, aunque funcione y no tenga defectos al principio.

R./ Verdadero. El autor nos dice "El software no se desgasta ... pero sí se deteriora".

Adicional a esto hay un dicho entre programadores que dice que programar es el arte de ingresar errores (bugs) al software. El software no es un producto finalizado, sino en continuo progreso, aunque funcione perfectamente, haga lo que tenga que hacer e incluso haga "magia" (como algunos clientes piensan que sucede), con el tiempo las necesidades evolucionarán, el ambiente evolucionará, los usuarios evolucionarán, todos ellos de una manera imposible de predecir por el mejor de los programadores y todos estos factores sumados o harán que falle el software o lo harán actualizar. Si es la última opción, entonces, como ya antes dije, se introducirán errores al software irremediablemente. Un ejemplo de esto es Windows, que hasta el momento no ha podido solucionar el problema de las impresoras en red con Windows 10, con cada actualización que hacen para arreglar el problema se cargan otras funcionalidades.

3. Las aplicaciones de software modernas son tan complejas que es difícil desarrollar nombres de categorías mutuamente excluyentes.



Universidad del cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Programa de Ingeniería de Sistemas

Ingeniería de software I

R./ Verdadero. Hay 7 categorías de software según el autor (de sistemas, de aplicación, de ingeniería y ciencias, incrustado, de línea de productos, aplicaciones web, de inteligencia artificial). Un compilador podría estar incrustado y por ende ser de sistemas e incrustado a la vez. Uno puede crear una suite de inteligencia artificial, entonces pertenecería a la categoría de software de línea de productos y también a la de inteligencia artificial. Las aplicaciones web, tienen categoría propia pero uno podría incluir muchas (o todas) a la categoría de aplicaciones. Por último, es evidente que las categorías de ingeniería y ciencias, y la de inteligencia artificial se solapan continuamente. Y si uno quisiera crear categorías mutuamente excluyentes en algún momento dejarán de ser excluyentes, aunque sea por un solo ejemplo.

Preguntas tipo V/F

1. Una declaración general de objetivos es todo lo que se necesita para empezar a desarrollar una aplicación software.
R./ Falso. Eso solo puede llevar a muchos dolores de cabeza y pérdida de recursos (tiempo y por ende dinero). Por algo primero se levantan primero los requerimientos primero, requerimientos tan detallados como sea necesario.
2. Cuanta más gente se incorpore a un proyecto empezado, más rápidamente se acabará.
R./ Falso. Eso tiene el efecto contrario, retrasa el proyecto. Más gente nueva significa quitarle tiempo a la gente que ya conoce el proyecto para explicarles a los nuevos, y como los nuevos son humanos, no robots, cometerán muchos errores al empezar.
3. La especificación de requisitos puede cambiar sin ningún tipo de problemas.
R./ Falso. Está bien un cambio leve, incluso a veces es necesario, no solamente un capricho del cliente. Pero los requisitos no podrían cambiar de hacer una calculadora para Android, a un sistema autónomo para naves espaciales.
4. La especificación de requisitos no puede cambiar nunca.
R./ Falso. Relacionada con el hecho anterior, sería un extremo, y todos los extremos son malos. Los requisitos pueden cambiar, bien sea porque el cliente en este caso sí tiene la razón, o porque no se definieron muy bien los requisitos iniciales o quedaron unas zonas grises.
5. Tener un modelo de proceso de desarrollo definido en una organización es suficiente para garantizar la aplicación de técnicas de IS (Ingeniería del Software).
R. / Falso. Es necesario pero no suficiente. Se necesita que sea el modelo adecuado, se requiere la experticia en la aplicación de dicho modelo y además también está el hecho de que quienes aplican las técnicas de IS son humanos, y pueden pasarse los modelos por alto, bien sea por ignorancia o por voluntad propia.



Universidad del cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Programa de Ingeniería de Sistemas

Ingeniería de software I

6. Contar con herramientas CASE modernas es más importante que contar con el hardware más nuevo para lograr una buena calidad y productividad del software.
R./ Verdadero. Aunque no hay que exagerar con la antigüedad del hardware, porque ahí se convierte en un problema para la productividad. Eso sin perder de vista que hay que aspirar a mantener la backwards compatibility pero es difícil y costoso, por no decir imposible mantener siempre el hardware actualizado a lo más nuevo (ni se diga ahora con los problemas de abastecimiento de toda la cadena abastecimiento de chips y de hardware en general)
7. Lo único que se entrega al terminar el proyecto es el programa funcionando.
R./ Falso. También se necesita la documentación, como mínimo un manual, si es que el código es privativo.