

Unidad 5

Guía Práctica N° 6

Guía de preguntas generales

Consignas:

- Para poder contestar correctamente, deben tener disponible la bibliografía de esta unidad:
 - (Pressman, 7ma. Edición, 2010) "Ingeniería del Software: Un Enfoque Práctico" Parte 3: Capítulo 14 y 16.
 - (Sommerville, 9na Edición, 2011) "Ingeniería del Software". Parte 4: Capítulo 24.

Ejercicios:

1. Describa cómo evaluaría la calidad de una universidad antes de inscribirse. ¿Cuáles factores serían importantes? ¿Cuáles tendrían importancia crítica?
2. Explique por qué un proceso de software de *alta calidad* debería conducir a productos de alta calidad de software. Discuta los posibles problemas con este sistema de gestión de calidad.
3. ¿Es lo mismo *calidad* y *seguridad*? Explique su respuesta.
4. Un colega, que es muy buen programador, elabora software con un bajo número de defectos, pero siempre pasa por alto los estándares de calidad de la organización. ¿Cómo deberían reaccionar sus administradores ante este comportamiento?
5. La calidad tiene un costo: lleva tiempo y recursos lograrla. Explique de manera clara el COSTO DE LA CALIDAD.
6. Comparando con el punto anterior, indique y explique el COSTO DE FALLA.
7. ¿Es posible evaluar la calidad del software si el cliente cambia continuamente lo que se supone que debe hacerse?
8. La *calidad* y *confiabilidad* son conceptos relacionados, pero difieren en lo fundamental por varias razones. Analice las diferencias.
9. ¿Un programa puede corregirse y aún así ser confiable? Explique su respuesta
10. ¿Un programa puede corregirse y tener buena calidad? Explique lo que responda.
11. ¿Por qué es frecuente que haya tensiones entre el grupo de ingeniería de software y el del aseguramiento de la calidad? ¿Es saludable eso?

12. Además de contar los errores y defectos, ¿hay otras características cuantificables de software que impliquen calidad? ¿Cuáles son y cómo podrían medirse directamente?
13. Considere dos sistemas cuya seguridad sea crítica y que estén controlados por computadora. Liste al menos tres peligros que se relacionen directamente con fallas del software.
14. ¿Qué efecto puede tener en la calidad de un producto de software no alcanzar lo que se denomina 'Calidad de Concordancia' (o 'Calidad de Conformidad')?
15. La Calidad de Software se basa en que deben existir 'características medibles', y por otra parte, en que todos los productos de trabajo tienen características medibles. De ejemplos de características medibles para una: a) 'Especificación de Requisitos', b) 'Especificación de Casos de Uso', c) 'Modelo de Casos de Usos', d) 'Diseño de Interfaz de Usuario', e) 'Diseño de Programa', y f) 'Diseño de Arquitectura de Software'.

Defina, si lo considera necesario, el paradigma de modelado base a ser utilizado como referencia en cada caso.

16. ¿A qué nos referimos cuando hablamos de costos de fallas internas y externas del software? ¿Cuáles fallas son más críticas?
17. Analice los aspectos que debe cubrir un producto de software para ser considerado de 'calidad' de acuerdo a la definición de McCall (Modelo de McCall). ¿Es posible alcanzarlo completamente? Justifique.
18. Analice los aspectos que debe cubrir un producto de software para ser considerado de 'calidad' de acuerdo a la definición de Boehm (Modelo de Boehm). ¿Es posible alcanzarlo completamente? Justifique.
19. Comparando el modelo de Boehm con el modelo de Mc Call, ¿puede aseverar que el modelo de Mc Call es más completo que el modelo de Boehm?. Justifique.
20. ¿Qué impacto tiene el proceso de desarrollo aplicado en el Aseguramiento de la Calidad?
21. ¿Podemos decir que un producto de software que tiene un *defecto*, indefectiblemente *fallará*?
22. ¿Cuál es la diferencia fundamental entre *validar* un producto de software y *verificar* el mismo producto?
23. ¿Cuáles de los enfoques (validar o verificar) puede aplicarse sobre la mayor cantidad de productos de software?.