Glosario

Introducción y fundamentos de testing de software



B

BUG:

Término comúnmente utilizado para referirse a un defecto o error en el software que causa que un programa funcione de manera incorrecta o falle al realizar la función esperada. Los bugs pueden surgir de errores de programación, diseño, requisitos mal interpretados, o una combinación de estos factores.

C

CALIDAD:

Medida en que un producto o proceso cumple con las especificaciones y expectativas. En el contexto del software, la calidad se evalúa en términos de características como la fiabilidad, eficiencia, mantenibilidad y usabilidad.

CICLO DE VIDA DEL DESARROLLO:

Conjunto de fases que describen el proceso completo de desarrollo de software desde la concepción hasta el retiro.
Las fases típicas incluyen planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento. Modelos comunes del ciclo de vida incluyen el modelo en cascada, el modelo incremental, el modelo ágil y otros.

D

DEFECTO:

Anomalía en el software que puede provocar que el sistema o componente no realice la función deseada. Es una desviación del comportamiento esperado o especificado del software.

D

EFICIENCIA:

Capacidad del software para proporcionar el rendimiento adecuado en relación con la cantidad de recursos utilizados bajo condiciones específicas. Incluye aspectos como el tiempo de respuesta, la utilización de la memoria y el procesamiento.

ERROR:

Equivocación cometida por una persona durante la especificación, diseño o codificación de un software. Puede llevar a la introducción de defectos en el producto.

F

FALLO:

Manifestación de un defecto en el software durante su ejecución. Un fallo ocurre cuando un componente del sistema no realiza su función según lo esperado.

FIABILIDAD:

Capacidad del software para funcionar correctamente durante un periodo específico bajo condiciones determinadas. Se refiere a la frecuencia con la que el software experimenta fallos y su capacidad para recuperarse de ellos.

I

IEC (INTERNATIONAL ELECTROTECHNICAL COMMISSION):

Organización global que prepara y publica estándares internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas, incluyendo las tecnologías de la información y la ingeniería de software.

INSPECCIÓN:

Revisión detallada de un producto de trabajo realizada por un equipo para identificar defectos y problemas de calidad. Es una técnica formal de revisión en la que los participantes siguen un proceso específico para examinar el producto y documentar los hallazgos.

ISO (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION):

Organización internacional que desarrolla y publica estándares internacionales en una variedad de disciplinas, incluyendo la ingeniería de software y la gestión de calidad.

M

MANTENIBILIDAD:

Facilidad con la que el software puede ser modificado para corregir defectos, mejorar el rendimiento u otras características, o adaptarse a un entorno cambiado. Incluye la claridad del código, la modularidad y la documentación.

MODULARIDAD:

Grado en que un sistema o programa está compuesto de módulos componentes separados, pero interconectados, cada uno con una funcionalidad específica. La modularidad facilita la mantenibilidad, ya que permite realizar cambios en una parte del sistema sin afectar otras partes.

R

REQUISITOS:

Descripciones detalladas de las funciones, capacidades y atributos que debe tener un sistema o componente para satisfacer las necesidades del usuario y cumplir con las especificaciones. Incluyen tanto requisitos funcionales como no funcionales.

T

TESTING DE SOFTWARE:

Actividad que consiste en ejecutar un sistema o componente con la intención de encontrar errores. Incluye la planificación, preparación y evaluación de pruebas para asegurar que el software cumple con los requisitos especificados.

U

USABILIDAD

Facilidad con la que los usuarios pueden aprender a utilizar el software, operarlo de manera eficiente y ser satisfechos con su uso. Se centra en la interfaz de usuario y la experiencia del usuario en general.

R

VALIDACIÓN:

Proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si cumple con los requisitos especificados y satisface las necesidades del usuario. Responde a la pregunta “¿Estamos construyendo el producto correcto?”.

VERIFICACIÓN:

Proceso de evaluar productos intermedios de trabajo de un proyecto para asegurarse de que cumplen con los requisitos especificados. Se centra en la correcta implementación del producto de software. Responde a la pregunta “¿Estamos construyendo el producto correctamente?”.

Referencias Bibliográfica

Boehm, B. W. (1981). Software Engineering Economics. Prentice Hall.

Humphrey, W. S. (1989). Managing the Software Process. Addison-Wesley.

IEEE Computer Society. (2014). IEEE Standard for Software and System Test Documentation. IEEE Std 829-2008 (Revision of IEEE Std 829-1998).

International Organization for Standardization. (2011). ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. ISO.

Kan, S. H. (2002). Metrics and Models in Software Quality Engineering (2.a ed.). Addison-Wesley.

McConnell, S. (1996). Rapid Development: Taming Wild Software Schedules. Microsoft Press.

Pfleeger, S. L. (2001). Software Engineering: Theory and Practice (2.a ed.). Prentice Hall.

Sommerville, I. (2015). Software Engineering (10.a ed.). Pearson.

Wiegers, K. E. (2003). Software Requirements (2.a ed.). Microsoft Press.