

Contents

Clase 1: Introducción a la Calidad del Software - ¿Qué es y por qué importa?	1
--	---

Clase 1: Introducción a la Calidad del Software - ¿Qué es y por qué importa?

Objetivos de la Clase:

- Entender la importancia fundamental de la calidad del software en el contexto actual del desarrollo de software.
- Definir la calidad del software desde diversas perspectivas (usuario final, desarrollador, cliente).
- Identificar los principales atributos y factores que contribuyen a la calidad del software.
- Introducir la idea de modelos y estándares de calidad como herramientas para mejorar el proceso de desarrollo.

Contenido Teórico Detallado:

1. ¿Por qué es importante la Calidad del Software?

- **Impacto en el Negocio:** Un software de baja calidad puede acarrear graves consecuencias para un negocio, incluyendo:
 - Pérdida de ingresos debido a fallas del sistema o errores.
 - Daño a la reputación de la marca y pérdida de confianza del cliente.
 - Aumento de los costos de mantenimiento y soporte.
 - Posibles demandas legales y sanciones regulatorias.
- **Impacto en el Usuario:** Para el usuario final, un software de baja calidad se traduce en:
 - Frustración y pérdida de tiempo debido a errores y mal funcionamiento.
 - Riesgos de seguridad y privacidad de sus datos.
 - Dificultad para realizar tareas y alcanzar sus objetivos.
- **Impacto en el Desarrollo:** En el equipo de desarrollo, la falta de calidad implica:
 - Mayor tiempo dedicado a la depuración y corrección de errores.
 - Aumento de la complejidad del código y la deuda técnica.
 - Disminución de la productividad y moral del equipo.

2. Definición de Calidad del Software: Múltiples Perspectivas

- **Perspectiva del Usuario Final:** La calidad se define por la facilidad de uso, la funcionalidad que satisface sus necesidades, la confiabilidad y la rapidez de respuesta del software. En resumen, ¿el software resuelve su problema de manera efectiva y eficiente?
- **Perspectiva del Desarrollador:** La calidad se centra en la corrección del código, la eficiencia del rendimiento, la mantenibilidad, la capacidad de adaptación a futuros cambios y la adherencia a los estándares de codificación.
- **Perspectiva del Cliente (Pagador):** La calidad se evalúa en términos de cumplimiento de los requisitos especificados en el contrato, el costo total del proyecto (incluyendo mantenimiento), el tiempo de entrega y el retorno de la inversión (ROI).
- **Definición formal (ISO/IEC 25010):** "Grado en el que un sistema, producto o servicio satisface las necesidades especificadas y las necesidades implícitas de sus partes interesadas".

3. Atributos y Factores de la Calidad del Software

- **Funcionalidad:** El software debe realizar las funciones para las que fue diseñado de forma correcta y completa.
- **Confiabilidad:** El software debe funcionar de manera consistente y sin fallas durante un período determinado. Incluye atributos como la disponibilidad, la tolerancia a fallos y la recuperabilidad.
- **Usabilidad:** El software debe ser fácil de usar, aprender y comprender para los usuarios. La interfaz debe ser intuitiva y proporcionar una buena experiencia de usuario.
- **Eficiencia:** El software debe utilizar los recursos del sistema (memoria, CPU, ancho de banda) de manera eficiente.

- **Mantenibilidad:** El software debe ser fácil de modificar, corregir y adaptar a nuevos requisitos. Un código bien estructurado y documentado facilita la mantenibilidad.
- **Portabilidad:** El software debe poder ejecutarse en diferentes plataformas y entornos sin necesidad de grandes modificaciones.
- **Seguridad:** El software debe proteger los datos y recursos del sistema contra accesos no autorizados y amenazas externas.
- **Integridad:** El software debe asegurar que la información sea precisa y completa.

4. Modelos y Estándares de Calidad (Introducción)

- **¿Por qué necesitamos Modelos y Estándares?** Los modelos y estándares proporcionan un marco de referencia para evaluar y mejorar la calidad del software. Ayudan a definir criterios objetivos, establecer procesos de desarrollo y asegurar la consistencia en la calidad.
- **Ejemplos:**
 - **ISO 9126 (Reemplazada por ISO/IEC 25010):** Define un conjunto de características de calidad y métricas para evaluar la calidad del software. (Se abordará en más detalle en clases futuras).
 - **CMMI (Capability Maturity Model Integration):** Un modelo de mejora de procesos que ayuda a las organizaciones a mejorar sus procesos de desarrollo de software y alcanzar niveles más altos de calidad. (Se abordará en más detalle en clases futuras).

Ejemplos/Casos de Estudio:

- **Caso 1: El desastre del Therac-25:** Un acelerador lineal de radioterapia que causó sobredosis masivas de radiación debido a errores de software. Este caso ilustra la importancia crítica de la calidad del software en sistemas que afectan la vida humana. (Buscar información en internet sobre "Therac-25 incident").
- **Caso 2: El fallo del software en el Boeing 737 MAX:** Un error en el sistema MCAS (Maneuvering Characteristics Augmentation System) causó dos accidentes aéreos fatales. Este caso destaca el impacto de la calidad del software en la seguridad de los sistemas complejos. (Buscar información en internet sobre "Boeing 737 MAX MCAS").

Problemas Prácticos/Ejercicios con Soluciones:

1. **Ejercicio:** Considera una aplicación móvil para gestionar citas médicas. Describe cómo definirías la calidad de esta aplicación desde la perspectiva del usuario final y desde la perspectiva del desarrollador.
 - **Solución (Usuario Final):** La calidad se definiría por la facilidad para programar y cancelar citas, la precisión de la información mostrada (horarios, médicos, ubicaciones), la confiabilidad de las notificaciones (recordatorios de citas), la seguridad de los datos personales y la rapidez de la aplicación.
 - **Solución (Desarrollador):** La calidad se definiría por la limpieza y eficiencia del código, la adherencia a los estándares de codificación, la facilidad de mantenimiento y actualización de la aplicación, la seguridad de la base de datos y la escalabilidad para soportar un número creciente de usuarios.
2. **Ejercicio:** Identifica tres atributos de calidad que consideras críticos para un software de banca online y explica por qué son importantes.
 - **Solución:**
 - **Seguridad:** Es fundamental para proteger la información financiera de los usuarios contra accesos no autorizados y fraudes.
 - **Confiabilidad:** Es crucial que el software funcione de manera consistente y sin fallas, ya que los usuarios confían en él para realizar transacciones importantes.
 - **Usabilidad:** Una interfaz intuitiva y fácil de usar permite a los usuarios realizar operaciones bancarias de forma rápida y sencilla.

Materiales Complementarios Recomendados:

- **Artículos:**
 - "What is Software Quality?" - Artículo de la ASQ (American Society for Quality).
 - Buscar artículos en IEEE Xplore o ACM Digital Library sobre la importancia de la calidad del software.
- **Libros:**
 - "Software Engineering" por Ian Sommerville (capítulo sobre calidad del software).
 - "The Art of Software Testing" por Glenford J. Myers, Corey Sandler, Tom Badgett.

Esta clase proporciona una base sólida para comprender la importancia de la calidad del software y los factores que influyen en ella. En las próximas clases, profundizaremos en los modelos y estándares de calidad, así como en las técnicas y herramientas para asegurar la calidad del software a lo largo del ciclo de vida del desarrollo.