ESTUDIANTE: Luis Quishpe

CARRERA: Software **PARALELO:** "A"

ASIGNATURA: Gestión de calidad del software **NIVEL:** 7 **FECHA: 14/10/2025**

DOCENTE: Ing. José Caiza, Mg.

TEMA: Resúmenes de las clases

Clase 1 - Conceptos de Calidad del software

Gestión Calidad del Software

La calidad es la aptitud para el uso. 1962

La calidad consiste en satisfacer un deseo 1993.

La calidad es el grado en el que un conjunto de características inherentes a un objeto cumple con los requisitos.

Construir software exitoso es difícil

- El 16.3% de los proyectos de software tienen éxito.
- El 52.7% de los proyectos software cuestan más, tardan más o hacen menos.
- El 31% son cancelados.

¿Qué es la calidad del producto?

La calidad es el conjunto de características de una entidad que le da esa entidad la capacidad de satisfacer necesidades expresas e implícitas(ISO 8402).

La calidad de software es el grado en que los atributos del software le permiten realizar su uso especifico.

Terminología

Error: es una acción cometida por un humano durante el desarrollo de software que produce un resultado incorrecto.

Defecto: el resultado de un error.

Falla: defectos no detectados y corregidos antes de la prueba.

Fallo: Manifestación de un defecto durante el testing o el uso del sistema.

Incidente: situación en la que se observa un comportamiento no esperado del sitema.

La calidad del software se entiende como la capacidad del producto para cumplir con las necesidades del usuario y los requisitos establecidos. No debe confundirse con la excelencia, ya que la calidad es relativa y depende del contexto y las expectativas del cliente. Según Juran, la calidad es la "aptitud para el uso", mientras que ISO 9000 la define como el grado en que un conjunto de características cumple los requisitos.

El software, a diferencia de los productos físicos, no se degrada con el tiempo, pero sus errores provienen del diseño y análisis. La ingeniería del software busca construir productos confiables, sin embargo, muchos proyectos fracasan por sobrecostos o incumplimiento de plazos. La calidad del proceso y del producto son dos dimensiones inseparables: un proceso bien definido genera productos más fiables.

En esta sección también se revisan los modelos de ciclo de vida del software: cascada, incremental, prototipado y espiral. Cada modelo busca garantizar la calidad a través de fases bien definidas, evaluación continua y reducción de riesgos.

Clase 2 - Calidad del producto de software

La calidad del producto de software puede definirse como el conjunto de características que le permiten satisfacer necesidades explícitas e implícitas. Existen diversos modelos que permiten estructurar y evaluar la calidad, entre ellos McCall, Boehm y FURPS.

El modelo de McCall clasifica la calidad en tres perspectivas: factores (vista del usuario), criterios (vista del producto) y métricas (medición cuantitativa). Sus principales factores son corrección, fiabilidad, eficiencia, integridad y usabilidad. Por otro lado, Boehm resalta la utilidad, facilidad de mantenimiento y portabilidad. FURPS, creado por Hewlett-Packard, añade dimensiones como funcionalidad, rendimiento y soporte.

Estos modelos buscan cuantificar la calidad mediante atributos medibles. Aunque empíricos, sirven como guías para mejorar la comunicación entre usuarios, desarrolladores y gestores, asegurando que el software cumpla con las expectativas de los stakeholders.

Clase 3 - Modelos de calidad del producto

Los modelos de calidad de producto han evolucionado hacia marcos estandarizados como el ISO/IEC 25000 y 25010. Estos establecen un lenguaje común para evaluar productos software mediante características y subcaracterísticas.

Relación entre propiedad a cuantificar, método de medición emc,mc

Si las mediciones son excelentes recién ahí puedo aplicar una subcaracterístia

25012 -> calidad de datos

25010- > Producto de software

Que implementaria deacuerdo a las caracteristicas o subcaracteristicas y como se mediria eso a travez de un software

El ISO/IEC 25010 define ocho características principales: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad. Cada una se evalúa mediante métricas cuantificables. Por ejemplo, la usabilidad se mide con subcaracterísticas como operabilidad, accesibilidad y estética de interfaz.

El propósito de estos modelos es establecer criterios de aceptación que aseguren la calidad interna (código, arquitectura), externa (comportamiento observable) y de uso (experiencia del usuario final). Estos estándares son esenciales para el aseguramiento y control de calidad.

Clase 4 - Estándares ISO/IEC 25000 y 25010

La familia ISO/IEC 25000, también conocida como SQuaRE (Software Product Quality Requirements and Evaluation), unifica las normas ISO 9126 y 14598. Su objetivo es proveer un marco común para definir, medir y evaluar la calidad del software.

El ISO/IEC 25010, componente central del SQuaRE, especifica un modelo de calidad basado en características y subcaracterísticas que permiten evaluar la conformidad de un producto con los requisitos del cliente. Este modelo es útil tanto para desarrolladores como para empresas que buscan certificar sus productos y diferenciarse en el mercado.

Implementar estos estándares facilita la detección temprana de defectos, la reducción de riesgos y la mejora continua de los procesos de desarrollo.

Clase 5 - Calidad en uso y evaluación

La calidad en uso representa la percepción del usuario sobre la efectividad, eficiencia y satisfacción al interactuar con el software en un contexto específico. Este modelo complementa la evaluación técnica del producto al considerar la experiencia del usuario final.

Según ISO/IEC 25010, la calidad en uso incluye cinco dimensiones: efectividad, eficiencia, satisfacción, libertad de riesgo y cobertura de contexto. Evalúa no solo el cumplimiento funcional sino también cómo el sistema contribuye a la productividad y bienestar del usuario.

Su aplicación permite priorizar mejoras centradas en la experiencia, detectando problemas de usabilidad, accesibilidad o rendimiento desde la perspectiva del usuario.

Clase 6 - Medición y métricas de calidad

La medición y las métricas son fundamentales para evaluar objetivamente la calidad del software. Según Fenton (1994), medir consiste en asignar números a atributos del mundo real siguiendo reglas claras. En ingeniería del software, las métricas permiten cuantificar atributos como mantenibilidad, fiabilidad o productividad.

Las escalas de medición pueden ser nominales, ordinales, de intervalo o de razón. La elección de la métrica depende de los objetivos del negocio y del tipo de evaluación. Por ejemplo, las métricas de usabilidad pueden basarse en tiempos de tarea o errores cometidos.

Medir la calidad implica establecer criterios, seleccionar métricas adecuadas y analizar resultados para tomar decisiones de mejora continua. Sin medición, no hay control ni optimización del proceso ni del producto.

Clase 7 – Métricas de Calidad interna

Métricas de calidad interna

Integridad funcional: es parte del desarrollo

Capacidad de cancelación de la operación del usuario

Idoneidad de la prueba

Las métricas de calidad interna se utilizan para evaluar las características del software desde el punto de vista del código fuente, la arquitectura y la estructura interna del sistema, antes de que esté en funcionamiento. Su propósito es detectar problemas de calidad de manera temprana durante el desarrollo, permitiendo realizar mejoras antes de que el software llegue al usuario final.

Estas métricas ayudan a medir atributos como la mantenibilidad, la complejidad, el modularidad, la legibilidad, el acoplamiento y la cohesión. Por ejemplo, una alta complejidad ciclomática puede indicar dificultad para mantener o probar un módulo, mientras que un bajo acoplamiento y alta cohesión suelen asociarse con una buena arquitectura.

Entre las métricas más utilizadas se encuentran:

- Complejidad ciclomática (McCabe): mide el número de caminos independientes en el código, indicando la dificultad de prueba y mantenimiento.
- Líneas de código (LOC): permite estimar el tamaño del software, aunque debe usarse junto con otras métricas.
- Densidad de defectos: calcula la cantidad de errores por cada mil líneas de código.
- Índice de mantenibilidad: combinas métricas como la complejidad y el volumen para evaluar qué tan fácil es mantener el código.

Las métricas internas se aplican principalmente en las fases de diseño e implementación, sirviendo como indicadores predictivos de la calidad externa (rendimiento, fiabilidad) y de la calidad en uso (satisfacción del usuario). En conclusión, son herramientas clave del aseguramiento de calidad, ya que proporcionan una base cuantitativa para controlar la calidad del producto antes de su entrega y mejorar continuamente los procesos de desarrollo.

Adjunto el repositorio

https://github.com/LUISALEXANDERQUISHPE/GestinCalidadSoftware.git