

Contents

Clase 6: Modelos de Calidad del Software: ISO 9126/ISO 25000 y CMMI	1
---	---

““

Clase 6: Modelos de Calidad del Software: ISO 9126/ISO 25000 y CMMI

1. Objetivos de la Clase:

- Comprender el propósito y la estructura de los modelos de calidad del software.
- Analizar en detalle el modelo ISO 9126 (ahora ISO 25000) y sus características.
- Analizar en detalle el modelo CMMI (Capability Maturity Model Integration) y sus niveles.
- Comparar y contrastar los modelos ISO 9126/ISO 25000 y CMMI, identificando sus fortalezas y debilidades.
- Aplicar los conceptos de los modelos de calidad a escenarios prácticos de desarrollo de software.

2. Contenido Teórico Detallado:

2.1. Introducción a los Modelos de Calidad del Software:

Un modelo de calidad del software es un marco de referencia que define un conjunto de características, atributos y métricas que se utilizan para evaluar y mejorar la calidad del software. Estos modelos proporcionan una estructura para comprender la calidad del software y para identificar áreas de mejora.

Propósito de los modelos de calidad:

- **Evaluación:** Proporcionar un marco para evaluar la calidad del software.
- **Mejora:** Identificar áreas de mejora en el proceso de desarrollo de software.
- **Comunicación:** Facilitar la comunicación sobre la calidad del software entre las partes interesadas.
- **Comparación:** Permitir la comparación de la calidad del software entre diferentes proyectos u organizaciones.

2.2. ISO 9126 / ISO 25000 (SQuaRE - Software Quality Requirements and Evaluation):

- **Historia y Evolución:** Originalmente ISO 9126, ahora reemplazada por la serie ISO 25000 (SQuaRE). SQuaRE proporciona un marco más amplio y detallado para la calidad del software.
- **Características de Calidad (ISO 9126):** ISO 9126 define seis características de calidad principales:
 - **Funcionalidad:** Grado en que el software proporciona las funciones que satisfacen las necesidades explícitas e implícitas del usuario. Subcaracterísticas: Adecuación, exactitud, interoperabilidad, seguridad, cumplimiento funcional.
 - **Confiabilidad:** Capacidad del software para mantener un nivel especificado de rendimiento bajo condiciones establecidas durante un período de tiempo determinado. Subcaracterísticas: Madurez, tolerancia a fallos, recuperabilidad.
 - **Usabilidad:** Facilidad con la que los usuarios pueden aprender a usar, operar, preparar entradas y comprender las salidas del software. Subcaracterísticas: Facilidad de aprendizaje, facilidad de uso, atractivo, cumplimiento de la usabilidad, accesibilidad.
 - **Eficiencia:** Rendimiento relativo a la cantidad de recursos utilizados. Subcaracterísticas: Comportamiento temporal, utilización de recursos.
 - **Mantenibilidad:** Facilidad con la que el software puede ser modificado, corregido, adaptado o mejorado. Subcaracterísticas: Facilidad de análisis, facilidad de cambio, estabilidad, facilidad de prueba.
 - **Portabilidad:** Capacidad del software para ser transferido de un entorno a otro. Subcaracterísticas: Adaptabilidad, facilidad de instalación, coexistencia, facilidad de reemplazo.
- **Componentes de ISO 25000:** La serie ISO 25000 expande y refina el concepto de ISO 9126, proporcionando un modelo de calidad más completo. Incluye divisiones para la gestión de la calidad, el modelo de calidad, la medición de la calidad y los requisitos de calidad.

2.3. CMMI (Capability Maturity Model Integration):

- **Propósito:** Un modelo de mejora de procesos que proporciona un marco para mejorar la capacidad de una organización para desarrollar y mantener productos y servicios de calidad.
- **Representaciones:** CMMI tiene dos representaciones:
 - **Escalonada (Staged):** Proporciona una ruta de mejora predefinida a través de cinco niveles de madurez.
 - **Continua:** Permite a las organizaciones seleccionar las áreas de proceso que son más importantes para sus objetivos comerciales y mejorar esas áreas de forma incremental.
- **Niveles de Madurez (Staged):**
 - **Nivel 1: Inicial:** El proceso es impredecible, mal controlado y reactivo.
 - **Nivel 2: Gestionado:** El proceso se planifica, ejecuta, mide y controla a nivel de proyecto.
 - **Nivel 3: Definido:** El proceso está estandarizado y documentado a nivel de organización.
 - **Nivel 4: Cuantitativamente Gestionado:** El proceso se mide y controla utilizando técnicas estadísticas y cuantitativas.
 - **Nivel 5: Optimizado:** El proceso se mejora continuamente basándose en la comprensión de las causas de variación inherentes al proceso.

2.4. Comparación y Contraste:

Característica	ISO 9126/ISO 25000	CMMI	:-----	:-----
-----	:-----			

|| **Enfoque** | Calidad del producto | Mejora del proceso | | **Propósito** | Evaluar y especificar la calidad del software | Mejorar la capacidad de una organización para desarrollar y mantener productos y servicios de calidad | | **Naturaleza** | Estándar de calidad | Modelo de mejora de procesos | | **Aplicación** | Seleccionar software, especificar requisitos de calidad, evaluar productos de software | Mejorar los procesos de desarrollo de software, evaluar la madurez de una organización | | **Niveles/Fases** | Características y subcaracterísticas de calidad | Niveles de madurez (Escalonada) o categorías de proceso (Continua) | | **Limitaciones** | No proporciona una guía detallada sobre cómo lograr la calidad | Se centra en los procesos, no necesariamente en la calidad intrínseca del producto. Implementar CMMI puede ser costoso y requiere un compromiso significativo de la organización. | | **Fortalezas** | Ampliamente reconocido y utilizado, proporciona un vocabulario común para la calidad | Proporciona una hoja de ruta para la mejora continua, ayuda a las organizaciones a ser más predecibles y eficientes. |

3. Ejemplos y Casos de Estudio:

- **Caso de Estudio 1:** Una empresa de desarrollo de software está buscando una herramienta de gestión de proyectos. Utiliza ISO 25000 para definir los requisitos de calidad de la herramienta y evaluar diferentes opciones. Prioriza la usabilidad y la eficiencia debido a la gran cantidad de usuarios que utilizarán la herramienta diariamente.
- **Caso de Estudio 2:** Una organización de desarrollo de software está buscando mejorar su proceso de desarrollo de software. Adopta el modelo CMMI (representación escalonada) y se somete a una evaluación. Actualmente se encuentra en el Nivel 2 (Gestionado) y está trabajando para alcanzar el Nivel 3 (Definido). Se enfoca en la estandarización de los procesos y la documentación.
- **Ejemplo:** Un proyecto de desarrollo de una aplicación móvil que maneja datos sensibles de usuarios. En el contexto de ISO 25000, la **seguridad** (una subcaracterística de la funcionalidad) es primordial. Se deben implementar medidas de seguridad robustas, como el cifrado de datos, la autenticación de usuarios y la autorización de acceso.

4. Problemas Prácticos o Ejercicios con Soluciones:

- **Ejercicio 1:** Una empresa está desarrollando un sistema de gestión de inventario. ¿Cuáles son las tres características de calidad de ISO 25000 que serían más importantes para este sistema y por qué?
 - **Solución:**
 - * **Funcionalidad:** El sistema debe proporcionar las funciones necesarias para gestionar el inventario de forma precisa y eficiente.
 - * **Confiabilidad:** El sistema debe ser fiable y evitar la pérdida de datos o errores en el cálculo del inventario.

- * **Eficiencia:** El sistema debe responder rápidamente a las consultas y operaciones de los usuarios, minimizando el tiempo de espera y el uso de recursos.
- **Ejercicio 2:** Una organización se encuentra en el Nivel 1 de CMMI. ¿Cuáles son los pasos que debe seguir para alcanzar el Nivel 2?
 - **Solución:**
 - * **Establecer una gestión de proyectos básica:** Definir los objetivos del proyecto, planificar las actividades, asignar recursos y establecer un cronograma.
 - * **Gestionar los requisitos:** Capturar, analizar y documentar los requisitos del cliente.
 - * **Gestionar la configuración:** Controlar las versiones del software y los documentos del proyecto.
 - * **Establecer una gestión de la calidad:** Planificar y ejecutar actividades de control de calidad, como revisiones de código y pruebas.
 - * **Medir y analizar el progreso:** Recopilar datos sobre el progreso del proyecto y analizar las métricas para identificar problemas y áreas de mejora.

5. Materiales Complementarios Recomendados:

- **Libros:**
 - "Software Engineering" by Ian Sommerville
 - "The CMMI for Development: Guidelines for Process Integration and Product Improvement" by Chrissis, Konrad, and Shrum
- **Artículos:**
 - Buscar artículos académicos y profesionales sobre ISO 25000 y CMMI en bases de datos como IEEE Xplore, ACM Digital Library y Google Scholar.
- **Sitios Web:**
 - ISO: <https://www.iso.org/>
 - CMMI Institute: <https://cmmiinstitute.com/> ““