**ISO/IEC 9126: Modelo de Calidad del Software**

La norma ISO/IEC 9126 se centra en la evaluación de la calidad del software desde diversas perspectivas, asegurando que cumpla con los requisitos funcionales y no funcionales de los usuarios. Esta norma ha sido fundamental en la industria del software y se divide en las siguientes características principales:

**1. Características del Modelo**

La ISO/IEC 9126 establece **seis características clave** para evaluar la calidad del software:

1. **Funcionalidad**:
   * **Adecuación**: Evalúa si el software cumple con las funciones especificadas.
   * **Exactitud**: Mide la precisión de las salidas en función de las entradas.
   * **Interoperabilidad**: Verifica la capacidad del software para interactuar con otros sistemas.
   * **Seguridad de acceso**: Asegura que los datos sensibles estén protegidos.
   * **Cumplimiento funcional**: Garantiza que las funciones cumplen con las normas legales y contractuales.
2. **Fiabilidad**:
   * **Madurez**: Determina la frecuencia de fallos en el software.
   * **Tolerancia a fallos**: Evalúa la capacidad del software para continuar operando tras un fallo.
   * **Capacidad de recuperación**: Mide qué tan rápido puede el software recuperar su funcionalidad tras un fallo.
   * **Cumplimiento de la fiabilidad**: Asegura que el software cumple con los requisitos establecidos en esta dimensión.
3. **Usabilidad**:
   * **Capacidad para ser entendido**: Mide si el usuario puede entender cómo funciona el software.
   * **Capacidad para ser aprendido**: Evalúa la facilidad de aprendizaje para usar el software.
   * **Capacidad para ser operado**: Mide la facilidad de operación del software por parte del usuario.
   * **Capacidad de atracción**: Considera la estética y el diseño atractivo del software.
   * **Cumplimiento de la usabilidad**: Verifica si cumple con las guías o estándares de diseño de interfaces.
4. **Eficiencia**:
   * **Comportamiento temporal**: Evalúa tiempos de respuesta y procesamiento del software.
   * **Utilización de recursos**: Mide el uso de recursos como CPU, memoria, etc.
   * **Cumplimiento de la eficiencia**: Asegura que cumple con los estándares de eficiencia acordados.
5. **Mantenibilidad**:
   * **Capacidad para ser analizado**: Evalúa qué tan fácil es diagnosticar problemas en el software.
   * **Capacidad para ser cambiado**: Determina qué tan sencillo es modificar el software.
   * **Estabilidad**: Mide el impacto de los cambios en la funcionalidad del software.
   * **Capacidad para ser probado**: Verifica la facilidad para realizar pruebas al software.
   * **Cumplimiento de la mantenibilidad**: Asegura que las prácticas de mantenimiento cumplen con estándares predefinidos.
6. **Portabilidad**:
   * **Adaptabilidad**: Mide qué tan fácil es trasladar el software a diferentes entornos.
   * **Instalabilidad**: Evalúa la facilidad de instalación del software.
   * **Coexistencia**: Analiza si el software puede operar junto a otros sistemas sin problemas.
   * **Capacidad para reemplazar**: Mide la facilidad con que puede ser reemplazado por otro software.
   * **Cumplimiento de la portabilidad**: Verifica que cumpla con las normas de portabilidad.

**2. Métricas de Calidad**

La norma se divide en varios enfoques de métricas:

* **Calidad interna**:
  + Analiza aspectos del código fuente (e.g., complejidad, redundancia).
  + Realiza pruebas de calidad antes de la implementación.
* **Calidad externa**:
  + Evalúa el comportamiento del software durante la ejecución (e.g., velocidad, errores en tiempo real).
* **Calidad en uso**:
  + Mide la satisfacción del usuario al interactuar con el software en un entorno real (e.g., eficacia, productividad, seguridad).

**3. Beneficios**

* **Estandarización**: Proporciona un marco común para evaluar la calidad del software.
* **Mejora continua**: Facilita la identificación de áreas de mejora.
* **Flexibilidad**: Adaptable a diferentes contextos y tipos de software.

**4. Ejemplo Práctico**

Un software de facturación puede ser evaluado con base en:

* Su **funcionalidad** (¿cumple con el cálculo de impuestos?).
* Su **fiabilidad** (¿cuántos errores se presentan durante el cierre de caja?).
* Su **usabilidad** (¿los empleados pueden aprender a usarlo rápidamente?).
* Su **eficiencia** (¿procesa facturas rápidamente?).
* Su **mantenibilidad** (¿es fácil agregar nuevos módulos, como contabilidad?).
* Su **portabilidad** (¿funciona igual en Windows y Linux?).

**ISO/IEC 25010: Modelo de Requisitos y Evaluación de Calidad de Software (SQuaRE)**

La norma **ISO/IEC 25010** es una evolución del modelo ISO/IEC 9126 y forma parte de la familia de estándares SQuaRE (*Software Quality Requirements and Evaluation*). Se centra en especificar y evaluar la calidad del software en dos dimensiones principales: **Calidad del Producto** y **Calidad en Uso**.

**1. Dimensiones de Evaluación**

**A. Calidad del Producto**

Esta dimensión mide las propiedades inherentes al software, como su código, arquitectura y funcionamiento general. Incluye las siguientes **8 características** principales:

1. **Adecuación Funcional**:
   * ¿El software cumple con las funciones requeridas? (e.g., calcular impuestos correctamente).
2. **Eficiencia de Desempeño**:
   * **Tiempo de respuesta**: Velocidad de ejecución.
   * **Uso de recursos**: Consumo de CPU, memoria, etc.
   * **Capacidad**: Manejo eficiente de la carga de trabajo (e.g., usuarios simultáneos).
3. **Compatibilidad**:
   * **Coexistencia**: Funcionamiento correcto junto a otros sistemas.
   * **Interoperabilidad**: Capacidad de interactuar con otras aplicaciones.
4. **Usabilidad**:
   * **Capacidad de aprendizaje**: Facilidad para aprender a usarlo.
   * **Accesibilidad**: Inclusión de usuarios con discapacidades (e.g., soporte de lectores de pantalla).
5. **Fiabilidad**:
   * **Madurez**: Estabilidad ante fallos.
   * **Disponibilidad**: Tiempo activo sin interrupciones.
   * **Tolerancia a fallos**: Recuperación tras errores críticos.
6. **Seguridad**:
   * **Confidencialidad**: Protección de datos.
   * **Integridad**: Evitar modificaciones no autorizadas.
   * **Autenticación**: Verificación de usuarios.
7. **Mantenibilidad**:
   * **Modularidad**: Facilidad para hacer cambios.
   * **Reusabilidad**: Uso de componentes en otros sistemas.
8. **Portabilidad**:
   * **Adaptabilidad**: Funcionamiento en diferentes entornos.
   * **Instalabilidad**: Facilidad de instalación.

**B. Calidad en Uso**

Evalúa el impacto del software en escenarios reales, midiendo la satisfacción del usuario final. Incluye las siguientes **5 características**:

1. **Eficacia**:
   * Capacidad del software para lograr los resultados esperados.
2. **Eficiencia**:
   * Uso óptimo de recursos por parte del usuario (e.g., completar tareas con pocos clics).
3. **Satisfacción**:
   * **Confianza**: Seguridad en que el software no fallará.
   * **Placer**: Experiencia positiva al usar el sistema.
4. **Seguridad en Uso**:
   * Minimización de riesgos durante su uso.
5. **Cobertura del Contexto**:
   * Adaptación del software a diferentes situaciones o necesidades de los usuarios.

**2. Métricas Asociadas**

El estándar ofrece guías detalladas para medir cada característica a través de métricas específicas, como:

* **Tiempos de respuesta promedio** para evaluar eficiencia.
* **Porcentaje de errores críticos por sesión** para fiabilidad.
* **Nivel de satisfacción del usuario** obtenido a través de encuestas.

**3. Beneficios**

* **Mejora de la calidad**: Permite identificar y corregir debilidades en el software.
* **Cumplimiento de requisitos**: Asegura que el software cumple tanto requisitos funcionales como no funcionales.
* **Optimización del desarrollo**: Orienta a los desarrolladores hacia objetivos claros.

**4. Ejemplo Práctico**

Un software de gestión hospitalaria puede ser evaluado según ISO/IEC 25010 de la siguiente manera:

* **Calidad del Producto**:
  + **Fiabilidad**: El sistema debe registrar y mostrar los datos de pacientes sin errores.
  + **Seguridad**: Proteger la información médica confidencial.
* **Calidad en Uso**:
  + **Eficacia**: Los médicos deben poder consultar historiales en menos de 5 segundos.
  + **Satisfacción**: Interfaces claras para reducir frustraciones en el personal.

**5. Diferencias con ISO/IEC 9126**

* Incluye dos nuevas características: **Seguridad** y **Compatibilidad**.
* Expande la dimensión de **Calidad en Uso**, con mayor énfasis en la experiencia del usuario final.
* Más adecuado para entornos modernos con múltiples dispositivos y sistemas integrados.

**CMMI (Capability Maturity Model Integration)**

El **CMMI** es un modelo de madurez y capacidad desarrollado para evaluar y mejorar los procesos organizacionales relacionados con el desarrollo, mantenimiento y operación de sistemas de software y otros proyectos. Su objetivo principal es proporcionar un marco que ayude a las organizaciones a lograr productos y servicios de alta calidad mediante la mejora continua de sus procesos.

**1. Principios Fundamentales**

* **Mejora de Procesos**: Se enfoca en la definición, implementación y mejora continua de procesos organizacionales.
* **Escalabilidad**: Aplicable a organizaciones de cualquier tamaño y sector.
* **Áreas Clave de Proceso (KPA)**: Categorías específicas que las organizaciones deben mejorar para avanzar en los niveles de madurez.

**2. Estructura de CMMI**

**A. Representaciones**

1. **Representación por etapas (Stages)**:
   * Define niveles de madurez que las organizaciones deben alcanzar secuencialmente.
   * Adecuada para organizaciones que buscan una mejora sistemática y estructurada.
2. **Representación continua (Continuous)**:
   * Enfatiza la mejora de áreas específicas de proceso sin un orden predefinido.
   * Permite flexibilidad según las necesidades organizacionales.

**B. Niveles de Madurez**

El modelo se compone de **cinco niveles de madurez**, que reflejan la capacidad de los procesos dentro de la organización:

1. **Nivel 1: Inicial**:
   * Procesos no estandarizados.
   * Dependencia de esfuerzos individuales.
   * Resultados impredecibles, frecuentes retrasos y sobrecostos.
2. **Nivel 2: Gestionado**:
   * Procesos básicos establecidos y planificados.
   * Uso de prácticas de gestión de proyectos como control de costos, tiempos y recursos.
3. **Nivel 3: Definido**:
   * Procesos estandarizados y documentados a nivel organizacional.
   * Uso de estándares y plantillas comunes.
4. **Nivel 4: Gestionado Cuantitativamente**:
   * Uso de métricas y datos para medir y gestionar la calidad y el rendimiento de los procesos.
   * Identificación de tendencias y predicción de problemas.
5. **Nivel 5: Optimizado**:
   * Mejora continua mediante análisis y ajustes.
   * Enfoque en la innovación y en adaptarse a los cambios del mercado o tecnología.

**3. Áreas Clave de Proceso (KPA)**

Las KPAs se agrupan en categorías dependiendo del nivel de madurez:

* **Gestión de Proyectos**: Gestión de requisitos, control de proyectos, aseguramiento de calidad.
* **Ingeniería de Procesos**: Desarrollo de requisitos, diseño, verificación y validación.
* **Soporte**: Gestión de configuración, medición y análisis.
* **Gestión Organizacional**: Gestión de recursos, planificación estratégica.

**4. Beneficios de Implementar CMMI**

* **Mejora de la Calidad**: Asegura que los productos y servicios cumplen estándares de alta calidad.
* **Reducción de Costos**: Minimiza retrabajos y errores al estandarizar procesos.
* **Optimización de Recursos**: Mejora la asignación y uso de recursos humanos y tecnológicos.
* **Mayor Satisfacción del Cliente**: Proporciona productos más confiables y alineados con las expectativas del cliente.

**5. Ejemplo Práctico**

**Contexto**: Una empresa de desarrollo de software implementa CMMI para mejorar la gestión de sus proyectos.

* **Nivel 1: Inicial**: Sus proyectos tienen retrasos frecuentes debido a falta de planificación.
* **Nivel 2: Gestionado**: Establece un plan de gestión de proyectos, incluyendo cronogramas y asignación de recursos.
* **Nivel 3: Definido**: Implementa estándares para documentación y pruebas, asegurando consistencia entre proyectos.
* **Nivel 4: Gestionado Cuantitativamente**: Introduce métricas para medir la eficiencia de los desarrollos y predecir riesgos.
* **Nivel 5: Optimizado**: Adopta metodologías ágiles e invierte en herramientas de automatización para mejorar la productividad.

**6. Comparación con otros estándares**

* A diferencia de ISO/IEC 9126 o 25010, CMMI se enfoca en la mejora de procesos organizacionales más que en características del producto final.
* Es más amplio, aplicable a múltiples industrias, mientras que los estándares ISO están más centrados en software.