**Material de Modulo 1 Parte 2**

Tiempo de lectura: 90 minutos

**Normas ISO de Calidad del Software: ISO/IEC 9126, ISO/IEC 25000 e ISO/IEC 25010**

**Introducción**

La calidad del software se ha convertido en un eje fundamental en el desarrollo y mantenimiento de sistemas informáticos. En un contexto de transformación digital y alta demanda de confiabilidad, usabilidad, seguridad y eficiencia, los desarrolladores, organizaciones y clientes requieren modelos estandarizados que permitan evaluar y garantizar la calidad del producto de software.

En este marco, han surgido diversas normas internacionales que guían la evaluación de calidad. Entre ellas, destacan la ISO/IEC 9126, la serie ISO/IEC 25000 (SQuaRE) y, más específicamente, la ISO/IEC 25010, que es la evolución directa del modelo conceptual de la 9126. En este documento, se desarrollarán de forma estructurada los contenidos y aportes de estas normas.

**La Norma ISO/IEC 9126: Origen, estructura y aplicación**

La norma ISO/IEC 9126, titulada "Software Engineering – Product Quality", fue publicada inicialmente en 1991 por la International Organization for Standardization (ISO) y la International Electrotechnical Commission (IEC). Su propósito principal fue proporcionar un modelo de referencia para evaluar la calidad del producto software, abordando tanto atributos internos como externos.

La norma surgió como respuesta a la necesidad de establecer una terminología común y un conjunto de criterios objetivos para evaluar la calidad del software, especialmente en una época donde el crecimiento del desarrollo informático requería métodos más rigurosos y sistemáticos.

**Estructura del modelo ISO/IEC 9126**

El modelo propuesto por ISO/IEC 9126 define la calidad del software como un conjunto de seis características principales, cada una de las cuales se subdivide en subcaracterísticas, tal como se muestra a continuación:

Característica principal Subcaracterísticas

Funcionalidad: Adecuación, exactitud, interoperabilidad, seguridad, cumplimiento funcional

Fiabilidad: Madurez, tolerancia a fallos, capacidad de recuperación

Usabilidad: Inteligibilidad, facilidad de aprendizaje, operatividad

Eficiencia: Comportamiento en el tiempo, utilización de recursos

Mantenibilidad: Capacidad de análisis, capacidad de cambio, estabilidad, facilidad de prueba

Portabilidad: Adaptabilidad, capacidad de instalación, coexistencia, reemplazabilidad

Cada subcaracterística podía ser evaluada mediante métricas cuantitativas y cualitativas definidas por el evaluador, adaptadas al contexto específico del producto software.

**Componentes adicionales**

La norma ISO/IEC 9126 estaba dividida en cuatro partes principales:

9126-1: Modelo de calidad.

9126-2: Métricas externas.

9126-3: Métricas internas.

9126-4: Métricas de calidad en uso.

Este enfoque integral permitía evaluar no solo el producto software desde una perspectiva técnica, sino también su interacción con el usuario final, a través de criterios como la efectividad, la productividad y la satisfacción.

**Limitaciones del modelo 9126**

* Aunque ampliamente utilizada, la ISO/IEC 9126 presentó ciertas limitaciones:
* Falta de actualización frente a nuevos contextos tecnológicos (software distribuido, servicios en la nube, aplicaciones móviles).
* Ambigüedad en la definición y medición de algunas subcaracterísticas.
* Escasa orientación práctica en cuanto a la implementación de métricas.

Estas limitaciones llevaron a la creación de una nueva familia de normas que actualizara, ampliara y reemplazara a la 9126: la serie ISO/IEC 25000 (SQuaRE).

**La Norma ISO/IEC 25000: SQuaRE (Software Product Quality Requirements and Evaluation)**

Propósito y enfoque general de la serie ISO/IEC 25000, conocida como SQuaRE, se introdujo como sucesora de ISO/IEC 9126 y de la norma ISO/IEC 14598 (Evaluación del software), con el objetivo de consolidar un modelo más completo, actual y aplicable en distintos contextos tecnológicos.

Su propósito es proporcionar un marco para la especificación de requisitos de calidad y para la evaluación de la calidad del producto de software, tanto en su diseño como durante su ejecución.

**Estructura de la serie ISO/IEC 25000**

* La serie SQuaRE está compuesta por múltiples normas organizadas en cinco divisiones temáticas:
* Requisitos de calidad del producto (ISO/IEC 2503x)
* Modelo de calidad del producto (ISO/IEC 2501x)
* Evaluación de calidad del producto (ISO/IEC 2504x)
* Métricas de calidad (ISO/IEC 2502x)
* Guía general y marco (ISO/IEC 25000)

La norma ISO/IEC 25000:2014, en particular, actúa como guía introductoria al modelo SQuaRE y establece definiciones clave, principios, y relaciones entre sus partes.

**Objetivos fundamentales del modelo SQuaRE**

* Mejorar la comprensión del concepto de calidad del producto software.
* Establecer criterios uniformes para definir y evaluar requisitos de calidad.
* Proporcionar soporte a los procesos de desarrollo, adquisición, evaluación y certificación de productos de software.
* Permitir la trazabilidad y comparabilidad entre productos mediante métricas normalizadas.

**La Norma ISO/IEC 25010: Modelo actualizado de calidad del software**

El contexto y objetivo de la norma ISO/IEC 25010:2011, titulada "Systems and software engineering —Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models", reemplaza a ISO/IEC 9126-1 y establece los nuevos modelos de calidad dentro de la familia 25000.

Esta norma introduce dos modelos fundamentales:

1. **Modelo de calidad del producto de software (ISO/IEC 25010)**

Este modelo define ocho características de calidad, que reemplazan y amplían las seis de la 9126:

Característica principal Subcaracterísticas incluidas

Adecuación funcional Completitud funcional, corrección funcional, adecuación funcional

Eficiencia del desempeño Tiempo de respuesta, utilización de recursos, capacidad (rendimiento bajo carga)

Compatibilidad Coexistencia, interoperabilidad

Usabilidad: Reconocibilidad, operabilidad, protección contra errores, estética, accesibilidad

Fiabilidad: Madurez, disponibilidad, tolerancia a fallos, capacidad de recuperación

Seguridad: Confidencialidad, integridad, no repudio, autenticación, autorización

Mantenibilidad: Modularidad, reusabilidad, analizabilidad, modificabilidad, capacidad de prueba

Portabilidad: Adaptabilidad, capacidad de instalación, reemplazabilidad

Este modelo refuerza la relevancia de nuevas características como la seguridad y la compatibilidad, que no estaban suficientemente diferenciadas en la norma 9126.

1. **Modelo de calidad en uso (Quality in Use)**

La calidad en uso hace referencia a la forma en que el software satisface las necesidades del usuario cuando se utiliza en un contexto real. Se define a través de las siguientes características:

Característica de calidad en uso Descripción

Efectividad: Grado en que los usuarios pueden alcanzar los objetivos propuestos

Eficiencia: Recursos utilizados en relación con los resultados obtenidos

Satisfacción del usuario: Grado de satisfacción respecto a la interacción con el producto

Seguridad en el uso: Minimización del riesgo para personas, negocios o medio ambiente

Cobertura del contexto: Capacidad del sistema para ser utilizado en distintos entornos o condiciones reales

Este enfoque amplía la perspectiva técnica tradicional y la integra con una visión centrada en el usuario.

**Comparación entre ISO/IEC 9126 y 25010**

|  |  |  |
| --- | --- | --- |
| Aspecto | ISO/IEC 9126 | ISO/IEC 25010 |
| Año de publicación | 1991 | 2011 |
| Características de calidad | 6 | 8 |
| Subcaracterísticas | 27 | 31 |
| Seguridad | Subcaracterística | Característica principal |
| Compatibilidad | No abordada | Característica principal |
| Calidad en uso | Limitada | Modelo completo |

**Relevancia práctica de las normas ISO/IEC 25000 y 25010**

Estas normas tienen múltiples aplicaciones en el ámbito profesional:

* Evaluación comparativa de productos software (benchmarking).
* Definición de métricas para pruebas, mantenimiento y validación.
* Elaboración de requisitos no funcionales durante la fase de análisis.
* Base para certificaciones de calidad y cumplimiento normativo.
* Soporte para procesos de mejora continua en equipos DevOps o metodologías ágiles.

Además, los modelos 25010 y 25000 se integran con otros marcos de referencia como ISO/IEC 12207 (procesos de ciclo de vida del software), CMMI, ITIL, y PMBOK, fortaleciendo la coherencia entre la gestión de proyectos y la calidad del producto. (de estas hablaremos en las próximas secciones)

**Resumen general de la lectura**

Mientras que la 9126 sentó las bases del modelo de calidad del producto, la serie 25000 y particularmente la ISO/IEC 25010 han proporcionado un marco más amplio, actual y aplicable a los entornos tecnológicos actuales. Estas normas son herramientas imprescindibles para ingenieros, gerentes de calidad, analistas de requisitos, testers y arquitectos de software comprometidos con la excelencia y la mejora continua.

La comprensión y aplicación de estas normas no solo permite desarrollar software con mayores estándares de calidad, sino también aumentar la confianza del usuario, reducir costos, mejorar la mantenibilidad y asegurar la viabilidad y éxito de los proyectos tecnológicos en el largo plazo.

En el contexto de la ingeniería del software, la calidad representa un componente esencial para garantizar que los productos desarrollados cumplan con los requisitos funcionales y no funcionales esperados por los usuarios. El desarrollo sistemático de software de calidad implica adoptar estándares internacionales que guíen su evaluación y mejora. Entre las normas más relevantes se encuentran la ISO/IEC 9126, la ISO/IEC 25000 y la ISO/IEC 25010, las cuales han evolucionado para ofrecer un marco robusto de evaluación de la calidad del software.

Referencias

[1] ISO/IEC 9126:1991, Software engineering — Product quality — Part 1: Quality model.

[2] ISO/IEC 25000:2014, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE.

[3] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[4] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo, "Recovering traceability links between code and documentation," IEEE Transactions on Software Engineering, vol. 28, no. 10, pp. 970-983, 2002.