Tema 10

La calidad del producto software: SQuaRE

Ingeniería del Software Avanzada

Índice

[Esquema 3](#_Toc535214619)

[Ideas clave 4](#_Toc535214620)

[10.1. Introducción y objetivos 4](#_Toc535214621)

[10.2. La calidad del software 5](#_Toc535214622)

[10.3. Modelos de calidad del producto software 9](#_Toc535214623)

[10.4. Familia ISO/IEC 25000 13](#_Toc535214624)

[10.5. ISO/IEC 25010: modelo y características de calidad 17](#_Toc535214625)

[10.6. ISO/IEC 25012: la calidad de los datos 24](#_Toc535214626)

[10.7. ISO/IEC 25020: medición de la calidad del software 26](#_Toc535214627)

[10.8. ISO/IEC 25022, 25023 y 25024: métricas de calidad del software y de los datos 27](#_Toc535214628)

[10.9. ISO/IEC 25040: evaluación del software 31](#_Toc535214629)

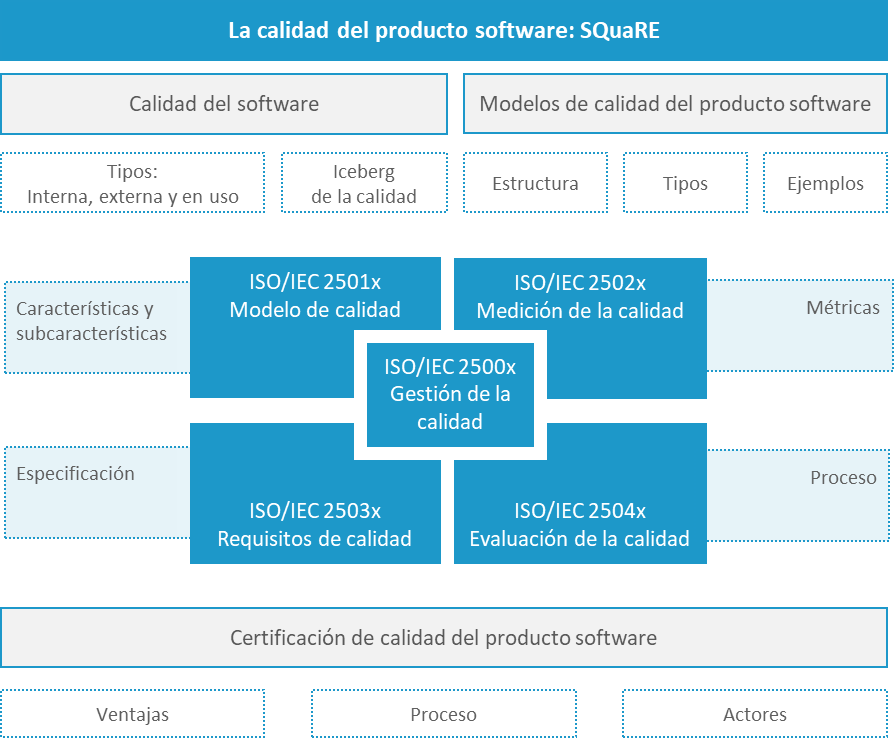
[10.10. El proceso de certificación 33](#_Toc535214630)

[10.11. Referencias bibliográficas 36](#_Toc535214631)

[A fondo 41](#_Toc535214632)

[Test 44](#_Toc535214633)

Esquema



Ideas clave

10.1. Introducción y objetivos

El concepto de **calidad del software** es más complejo de lo que puede parecer a primera vista. Se trata de un concepto abstracto (XBOSOFT, 2018), cuya presencia es difícil de definir, pero su ausencia es fácilmente detectable.

Existen *frameworks* y modelos que ayudan a determinar el conjunto de características y parámetros que es necesario evaluar desde diferentes puntos de vista. En este tema analizaremos el conjunto de **normas de la familia ISO/IEC 25000** que pretenden ayudar y guiar en este sentido.

Mediante el estudio de este tema alcanzaremos los siguientes objetivos:

* **Analizar el concepto de calidad de software** y los elementos que la condicionan.
* **Conocer los tres enfoques generales para analizar la calidad de un producto**, desde el punto de vista interno, externo y en uso, y su relación con la calidad del proceso.
* **Conocer los elementos que definen, generalmente, los modelos** de calidad de software y su clasificación.
* **Analizar el conjunto de normas de la familia ISO/IEC 25000** y sus elementos principales.
* **Conocer las fases del proceso de certificación de la calidad del software** según ISO/IEC 25000 y los actores que intervienen.

10.2. La calidad del software

Según la **Real Academia Española de la Lengua**, la **calidad** es la «propiedad, o conjunto de propiedades inherentes a algo, que permiten juzgar su valor» (RAE, 2018).

Una definición más técnica contenida en la **norma ISO 9000:2005** nos dice que se trata del «grado en que un conjunto de características inherentes de un objeto cumple con sus requisitos» (ISO, 2015). El hecho de que las características sean «inherentes» implica que existen *de facto* en el objeto, y por tanto pueden ser medidas, mientras que un requisito es una «necesidad o expectativa que se declara sobre ese objeto», generalmente con carácter obligatorio.

Cuando hablamos de la **calidad del producto software**, o calidad del software, la norma **ISO/IEC 25000:2014** indica que es la «capacidad del producto software para satisfacer las necesidades expresadas o implícitas, cuando se utiliza en condiciones específicas» (ISO, 2014a). Vemos, por tanto, que **la calidad depende de los atributos del objeto**, que deben ser observables, y siempre está relacionada con las **expectativas del usuario y unas condiciones de utilización concretas**. Estas necesidades o expectativas son precisamente los requisitos del producto.

**Es más fácil identificar los sistemas de baja calidad,** como aquellos que no cumplen la función que se espera de ellos, que no se utilizan porque resultan demasiado complejos, o que no pueden mantenerse cuando el equipo que lo desarrolló desaparece. Por el contrario, **la identificación de la calidad es más compleja**, pues precisa de la definición de métricas concretas que permitan evaluar en qué medida las expectativas sobre el producto son alcanzadas.

En la norma **ISO/IEC 25010** se presentan **tres enfoques del concepto de calidad** que se relacionan en un ciclo de vida de calidad, de manera que cada nivel tiene influencia en los siguientes, como se observa en la figura 1. Mientras que la calidad del propio proceso de desarrollo influye claramente los atributos internos y externos del producto, estos tienen un impacto final en las características del producto, una vez que es utilizado en diferentes contextos de operación.

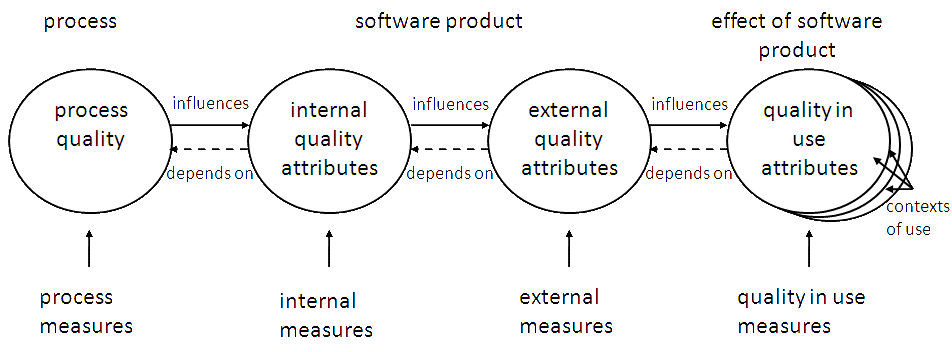


Figura 1. Relación entre la calidad del proceso, la calidad interna, la calidad externa y la calidad en uso del producto software. Mientras que los tres primeros elementos son controlables por el proveedor o fabricante del producto, la calidad en uso depende de la percepción que el usuario tiene durante su utilización.  
Fuente: Norma ISO/IEC 9126

Del mismo modo, las metas que se fijen para los atributos de calidad del producto van a determinar las metas fijadas y las características deseadas para el proceso de desarrollo que dará como resultado ese producto con ese nivel de calidad exigido.

* **Calidad interna**: es observable a partir de características intrínsecas del propio producto, como el código fuente. Aquí encontramos atributos como la estructura del código, la mantenibilidad, complejidad, acoplamiento, escalabilidad o reusabilidad. Todos estos aspectos solo pueden ser evaluados mediante inspecciones y pruebas estáticas del software, analizando la propia estructura del código fuente, sus componentes y el diseño y arquitectura del sistema.
* **Calidad externa**: es observable a partir del comportamiento del producto y puede ser evaluada mediante pruebas de software. Encontramos aspectos como la corrección, eficiencia y costes de mantenimiento. Aquí es donde las pruebas de software tienen utilidad, pues permiten ejecutar el producto y comprobar si se verifican determinados requisitos de calidad externa.
* **Calidad en uso**: es observable durante la utilización del producto por parte del usuario, en un entorno de pruebas o de producción.

Cuando construimos software,

intentamos que la mejora del proceso repercuta en la calidad interna,

que esta, a su vez, lo haga en la calidad externa y que, finalmente, la calidad en uso para cada uno de los contextos de uso sea buena.

Las dos primeras categorías son las que, de alguna manera, puede evaluar directamente el fabricante, sin la intervención del usuario.

Sin embargo, nos encontramos con un dilema que viene representado por el **«iceberg de la calidad del software»,** representado en la figura 2, que marca una clara diferencia entre las pruebas del software (*software testing*) y el aseguramiento de la calidad (*quallity assurance*), o entre lo «visible» y lo «invisible» (Bloom, 2011).

Mientras que las pruebas tradicionales están dirigidas a verificar la calidad del software desde el punto de vista de su funcionamiento, muchas veces solo pueden llegar a determinar la presencia de errores, o ausencia de calidad, pero no a garantizar un software libre de errores, o a detectar posibles causas más profundas relacionadas con atributos de calidad interna.

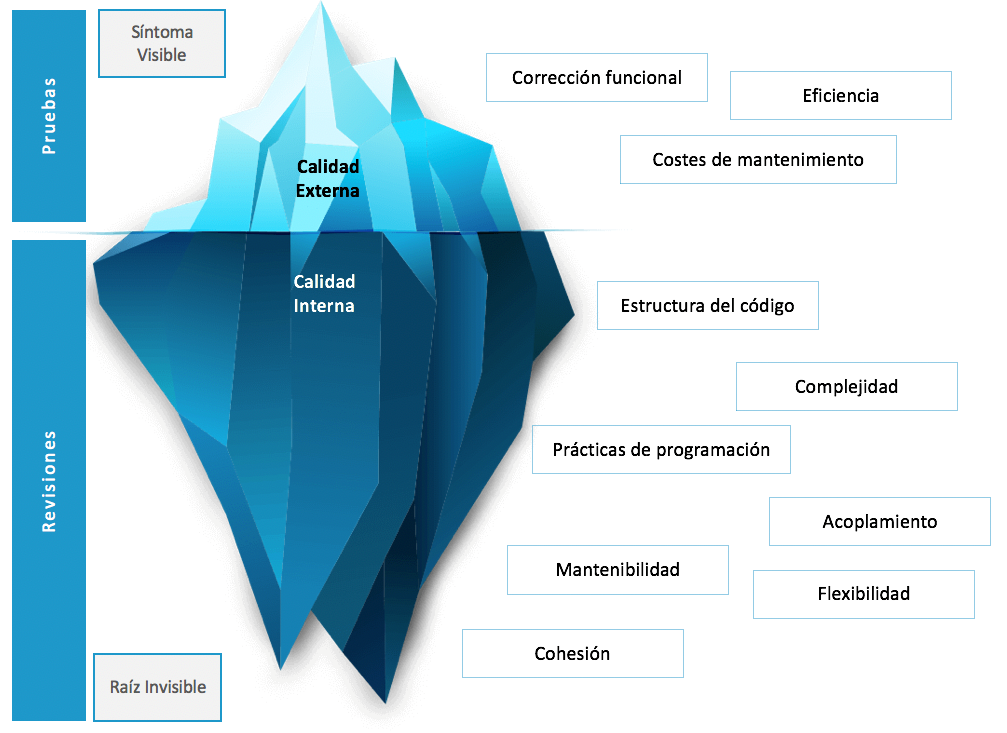


Figura 2. El iceberg de la calidad del software. Fuente: Elaboración propia a partir de McConnell (1993).  
Créditos de la imagen: kisspng

Actualmente, todos **los aspectos relacionados con la calidad del proceso de desarrollo de software** y **la calidad del propio producto** están reclamando cada vez más atención en el mundo de la ingeniería del software. Son dos aspectos que están claramente interconectados; l**a mejora de los procesos de construcción de software afecta a la calidad del resultado, mientras que las mejoras introducidas en el propio producto pueden facilitar los procesos que soportan su desarrollo.**

Las principales iniciativas han venido centrándose en la mejora de los procesos, pero no debemos olvidar que, finalmente, de lo que se trata es de mejorar la calidad del producto construido. Algunos autores vienen señalando que cualquier estándar o proceso de certificación debería estar centrado principalmente en el producto y no tanto en el proceso. «Las evaluaciones deberían basarse en evidencias directas sobre los atributos del producto, y no en evidencia circunstancial sobre el proceso» (Maibaum y Wassyng, 2008, p. 91).

Otros autores señalan que existe poca evidencia de que la adopción de procesos estandarizados garantice en sí misma la obtención de buenos productos. Los procesos estandarizados garantizan la uniformidad de los resultados obtenidos, pero se corre el peligro de institucionalizar prácticas perjudiciales, que no estén en realidad contribuyendo al objetivo final.

Por todos estos motivos, se ha venido detectando la **necesidad de contar con normas y marcos metodológicos que faciliten la medición de la calidad del producto software**, su análisis y la elaboración de los resultados en informes. La nueva familia de estándares **ISO/IEC 25000** de reciente aparición se centra precisamente en este interés, y también es denominada **SQuaRE** (iso25000.com, 2018a): *System and Software Quality Requirements ad Evaluation* (Requisitos y evaluación de la calidad del software y el sistema).

10.3. Modelos de calidad del producto software

En esta sección vamos a revisar algunas de las principales normas y modelos que se ocupan de valorar el producto software teniendo en cuenta un conjunto de atributos que lo caracterizan. La mayoría de los modelos de calidad del software se estructuran como se muestra en la figura 3.

* Los **factores de calidad**, o atributos externos, son las características que determinan la calidad desde el punto de vista del usuario. Aglutina las expectativas que tiene el usuario sobre los beneficios que obtendrá del producto.
* Los **criterios de calidad**, o atributos internos, son características del propio producto asociadas a cada factor.
* Las **métricas asociadas a cada criterio de calidad** son medidas cuantitativas concretas que permiten valorar el grado de presencia de un atributo.

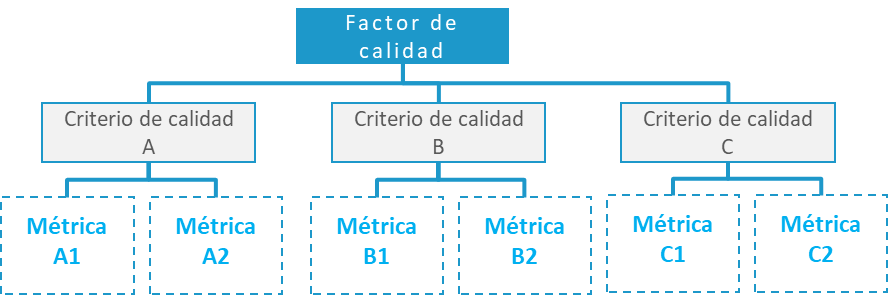


Figura 3. Estructura de la calidad del software.  
Fuente: Elaboración propia a partir de Callejas–Cuervo, Alarcón–Aldana y Álvarez–Carreño (2017, p. 238)

Los diferentes **modelos de calidad del producto software** se pueden clasificar según tengan un enfoque de modelo de calidad fijo, a medida o mixto (Calero, Piattini y Moraga, 2010).

* **Los modelos fijos** ofrecen un catálogo de factores de calidad para emplear como base de la evaluación. Generalmente ofrece una jerarquía amplia de factores, criterios y métricas, de los que se puede seleccionar los elementos más adecuados para un proceso de evaluación concreto. A esta categoría pertenecen los modelos de McCall (McCall, Richards y Walters, 1977b, 1977a), Boehm (Boehm, 1978), y FURPS (Grady y Caswell, 1987), con un enfoque más industrial.
* **Los modelos a medida** se definen *ad hoc* para cada proyecto, partiendo de los objetivos concretos que se desea alcanzar.
* **Los modelos mixtos** intentan combinar las ventajas de los modelos anteriores. Ofrecen un conjunto de factores de calidad de alto nivel, reutilizables en todos los proyectos, pero refinados para cada caso particular. A esta categoría pertenece el estándar ISO/IEC 9126-1 que revisaremos un poco más adelante.

La tabla 1 recoge las principales ventajas e inconvenientes de las dos primeras categorías, lo cual da una idea de los potenciales beneficios de un modelo mixto. En el trabajo de Callejas y Cuervo et al. (2017) se puede encontrar una revisión de algunos de los modelos de calidad más conocidos, junto con una recopilación de diferentes implementaciones a nivel mundial.

|  |  |  |
| --- | --- | --- |
| **TIPO** | **VENTAJAS** | **INCONVENIENTES** |
| **Fijo** | Ofrecen una vista común que facilita la comparación entre proyectos. | Falta de flexibilidad, al asumir que un conjunto de factores es válido para todo proyecto. |
| **A medida** | Adaptabilidad total a las características del producto. | Falta de reutilización y elevado coste de desarrollo del modelo. |

Tabla 1. Ventajas e inconvenientes de los diferentes tipos de modelos de calidad de software

En este apartado revisaremos únicamente algunos de los estándares internacionales más directamente relacionados con SQuaRE.

ISO/IEC 9126:2001

Esta norma establece un modelo de calidad para el producto software. La primera versión apareció en 1991, aunque fue sustituida en 2001 por dos estándares relacionados, el ISO/IEC 9126 para la calidad del software y el ISO/IEC 14598 para la evaluación de productos de software.

Está formada por cuatro partes fundamentales, representadas en la figura 4. Tanto esta norma como su sucesora, ISO 25000 (SQuaRE), **contemplan la calidad del producto desde tres perspectivas**:

* **Una interna**: relacionada con la calidad del código.
* **Otra externa**: relacionada con las características en ejecución.
* **Otra en uso**: más relacionada con la perspectiva del usuario, que ya vimos en la figura 1.

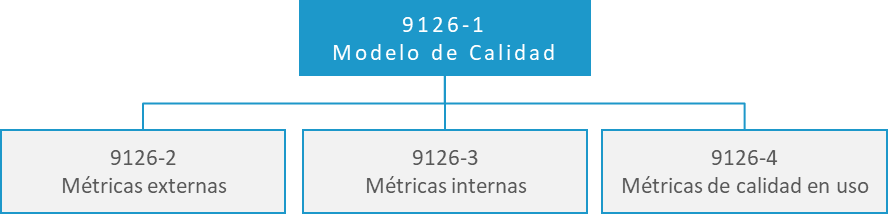


Figura 4. Las cuatro partes de la norma ISO 9126

Se trata de un modelo mixto por definir «una jerarquía de factores de calidad clasificados en características, subcaracterísticas y atributos» (Calero et al., 2010, p. 294).

Se ofrece un conjunto de factores de partida compuesto por seis características comunes a las vistas interna y externa —funcionalidad, fiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad— y veintisiete subcaracterísticas.

ISO/IEC 14598

Esta norma está directamente relacionada con ISO 9126 y se encarga de establecer los procesos para realizar la evaluación de la calidad del software. Su primera versión salió a la luz en 1999. Las normas ISO 9126 e ISO 14598, a pesar de estar relacionadas entre sí, tienen múltiples inconsistencias. Por este motivo, se ha desarrollado recientemente el conjunto de normas ISO/IEC 25000, que aglutinan bajo una misma familia tanto el modelo de calidad del producto software, como las guías para realizar el proceso de evaluación.

CISQ

El *Consortium for IT Software Quality* (CISQ) es una organización fundada por el OMG (*Object Management Group*) y el SEI (*Software Engineering Institute*) para desarrollar estándares que permitan automatizar la medida del tamaño del software y la calidad estructural del código fuente (CISQ, 2018). A la hora de analizar la calidad del código podemos considerar que CISQ se basa en SQuaRE, pues elige cuatro de las ocho características propuestas en el modelo de calidad de software (ver figura 7) y propone maneras de automatizar su medida. Estas características son la fiabilidad, la eficiencia de desempeño, la seguridad y la mantenibilidad, por ser en opinión del OMG y el SEI las más importante y las más susceptibles de poder medirse de manera automática con herramientas específicas (Garzás, 2013).

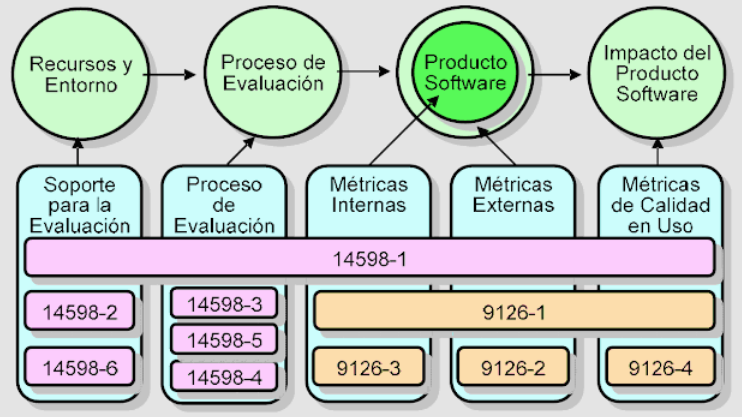


Figura 5. Relación entre los estándares ISO/IEC 9126 e ISO/IEC 14598. Fuente: C alero et al. (2010, p. 295)

10.4. Familia ISO/IEC 25000

Todos los aspectos relacionados con la calidad del proceso de desarrollo de software y la calidad del propio producto están reclamando cada vez más atención en el mundo de la ingeniería del software. La nueva familia de estándares ISO/IEC 25000, de reciente aparición, se centra precisamente en este último aspecto, el de **evaluar la calidad del producto software**.

Toma el sobrenombre de **SQuaRE** (iso25000.com, 2018a): *System and Software Quality Requirements and Evaluation* (Requisitos y evaluación de la Calidad del software y del sistema).

**ISO/IEC 25000** aparece para sustituir y evolucionar un conjunto de normas anteriores, basadas en **ISO/IEC 9126** (*Software Engineering – Product Quality*), que ofrecía un modelo de calidad del producto software, y en **ISO/IEC 14598** (*Information Technology – Software Product Evaluation*), que se centra en la evaluación del producto software. **El objetivo de SQuaRE es la construcción de un marco común de referencia para evaluar la calidad del producto software.**

A su vez, ISO/IEC 25000 se descompone en **cinco grandes divisiones**, como se aprecia en la figura 6. Cada una de ellas está constituida por diferentes normas que tratan aspectos específicos, y que se resumen en la tabla 2.

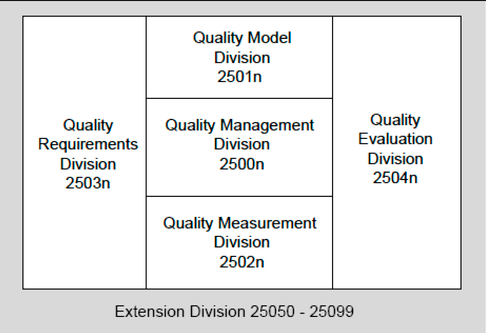


Figura 6. Organización de SQuaRE en familias de normas, también llamadas divisiones. Fuente: ISO (2017)

|  |  |  |
| --- | --- | --- |
| **NORMA ISO/IEC** | **DENOMINACIÓN** | **DESCRIPCIÓN** |
| **2500x** | **División de gestión de la calidad** | Contiene términos y definiciones comunes que se emplean por el resto de las normas de la familia. |
| 25000:2014 | *Guide to SQuaRE* | Contiene el modelo de la arquitectura de SQuaRE, terminología, un resumen de las partes y modelos de referencia. |
| 25001:2014 | *Planning and management* | Contiene orientaciones para gestionar la evaluación de los requisitos del producto de software. |
| **2501x** | **División de modelo de calidad** | Ofrece modelos de calidad detallados, definiendo características para evaluar la calidad del producto. |
| 25010:2011 | *System and software quality models* | Contiene el modelo de calidad del producto, y define las características que deben evaluarse. |
| 25011:2017 | *Service quality models* | Define un modelo específico para representar la calidad de los servicios que desarrollamos. |
| 25012:2008 | *Data quality model* | Define un modelo general para la calidad de los datos que se almacenan en el sistema. |
| **2502x** | **División de medición de calidad** | Ofrece orientaciones para la medición de la calidad del producto, definiciones de medidas de calidad y guías prácticas de aplicación. |
| 25020:2007 | *Measurement reference model and guide* | Explicación introductoria y modelo de referencia para la medición de calidad. También ofrece una guía de selección y desarrollo de medidas. |
| 25021:2012 | *Quality measure elements* | Define un conjunto de métricas base y derivadas que pueden utilizarse a lo largo de todo el ciclo de vida. |
| 25022:2016 | *Measurement of quality in use* | Define métricas específicas para evaluar la calidad del producto en uso. |
| 25023:2016 | *Measurement of system and software product quality* | Define métricas específicas para medir la calidad de productos y sistemas. |
| 25024:2015 | *Measurement of data quality* | Define métricas específicas para evaluar la calidad de los datos. |
| **2503x** | **División de requisitos de calidad** | Ayudan a mejorar la especificación de requisitos desde el punto de vista de su calidad. |
| 25030:2007 | *Quality requirements* | Proporciona recomendaciones para realizar la especificación de requisitos de calidad del producto. |
| **2504x** | **División de evaluación de la calidad** | Ofrece un conjunto de guías y recomendaciones para realizar la evaluación del producto software. |
| 25040:2011 | *Evaluation process* | Modelo de referencia general para la evaluación. Considera entradas al proceso, restricciones y recursos necesarios. |
| 25041:2012 | *Evaluation guide for developers, acquirers and independent evaluators* | Guía la implementación del proceso de evaluación de manera práctica para desarrolladores, responsables de adquisiciones y evaluadores independientes. |

Tabla 2. Diferentes componentes del conjunto de normas ISO/IEC 25000.  
Fuente: Elaboración propia a partir de ISO (2018) y de iso25000.com (2018)

Además de estas divisiones, se contempla la futura extensión de SQuaRE con nuevas normas —de la 25050 a la 25099 para dominios de aplicación específicos— y recoger, así, sus particularidades concretas. Por ejemplo:

* **ISO/IEC 25051**: *Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing*. Contiene requisitos de calidad y requisitos para realizar las pruebas de productos RUSP (ISO, 2014b). Se trata de productos disponibles para cualquier usuario, con o sin coste, listos para ser utilizados sin realizar actividades de desarrollo adicionales. Se caracterizan porque el usuario no tiene influencia en la especificación de sus características. En la norma se considera dentro de RUSP la descripción del producto (incluyendo la información del sitio web), la documentación de usuario (necesaria para instalar y utilizar el producto y configurar el entorno donde se va a ejecutar), y el propio software que se ofrece a través de algún mecanismo de distribución (CD–ROM, descarga de Internet, etc.). Se incluyen en esta categoría productos como los procesadores de texto, hojas de cálculo, aplicaciones móviles y aplicaciones web, pero se deja fuera el software de código abierto.
* **ISO/IEC 25062:2006***: Common Industry Format (CIF) for usability test reports*. Está dirigido a expertos en usabilidad para guiarles en la generación de informes de pruebas de usabilidad (ISO, 2006). El CIF estandariza el tipo de información que se captura durante las pruebas con usuarios. Contempla variables como la demografía, la descripción de la tarea, el contexto de la prueba y el equipamiento necesario para realizarla. Hasta el momento no existía un formato estándar para ofrecer informes de usabilidad, y cada compañía adoptaba sus propios formatos.

Es importante recordar que **SQuaRE describe qué se debe evaluar, pero no especifica cómo hacerlo.** No detalla los umbrales que deben emplearse junto a las métricas ni la manera de agruparlas para asignar una única puntuación de calidad a un producto (Rodríguez & Piattini, 2014).

De todas las normas anteriores podemos destacar la **ISO/IEC 25010**, por definir el modelo general de calidad del producto y el conjunto de atributos que deberían evaluarse, la **ISO/IEC 25012**, por hacer lo mismo para el caso particular de los datos, y la **ISO/IEC 25040**, puesto que proporciona un modelo de referencia general para el proceso de evaluación.

10.5. ISO/IEC 25010: modelo y características de calidad

La norma **ISO/IEC 25010:2011 define los modelos fundamentales que proporcionan un conjunto de características susceptibles de ser evaluadas para determinar la calidad del producto software.**

Por un lado, se define un **modelo de calidad en uso** compuesto por cinco característicasy, por otro, **un modelo de calidad del producto** definido por otras ocho características. Ambos modelos se complementan con el modelo de calidad de los datos que proporciona la norma ISO/IEC 2012:2008, que veremos luego.

Modelo de calidad en uso

La **calidad en uso** es el «grado en el cuál un producto o sistema puede ser utilizado por usuarios específicos cumpliendo sus necesidades para alcanzar objetivos específicos con efectividad, eficiencia, libres de riesgos y satisfacción, en contextos específicos de utilización» (ISO, 2011, sección 4.1).

Aquí la norma identifica un total de cinco características, algunas de las cuales se desagregan en propiedades específicas, como se recoge en la tabla 3.

|  |  |
| --- | --- |
| **Característica**  Subcaracterística | **Explicación** |
| **Efectividad** | Es la precisión y con qué medida los usuarios alcanzan sus objetivos. |
| **Eficiencia** | Cantidad de recursos gastados con relación a la efectividad obtenida. |
| **Satisfacción** | Grado en que las necesidades del usuario se satisfacen al emplearse el producto en un contexto determinado. |
| Utilidad | Grado en que el usuario queda satisfecho al alcanzar sus objetivos con el uso del producto. |
| Confianza | Grado en que el usuario espera que el producto se comportará según lo esperado. |
| Placer | Grado en que el usuario utiliza el producto con agrado. |
| Confort | Grado en que el usuario está físicamente satisfecho. |
| **Ausencia de riesgos** | Grado en que el producto mitiga potenciales riesgos económicos, sobre la salud o la vida humana, o el entorno. |
| Mitigación de riesgo económico | Grado en que el producto mitiga riesgos relacionados con el estado financiero, la operación eficiente, la propiedad comercial o la reputación. |
| Mitigación de riesgo de salud y seguridad | Grado en que el producto mitiga riesgos potenciales para las personas en los contextos de uso contemplados. |
| Mitigación de riesgo medioambiental | Grado en que el producto mitiga potenciales riesgos para la propiedad o el medio ambiente en los contextos de uso contemplados. |
| **Cobertura de contexto** | Grado en que el producto puede utilizarse tanto en contextos especificados como en otros no contemplados inicialmente. |
| Completitud de contexto | Grado en que el producto puede utilizarse de manera efectiva, eficiente, sin riesgos y con satisfacción en todos los contextos de uso especificados. |
| Flexibilidad | Grado en que el producto puede utilizarse de manera efectiva, eficiente, sin riesgos y con satisfacción en contextos diferentes a aquellos especificados en los requisitos. Puede alcanzarse adaptando el producto, y permite que el producto aproveche circunstancias, oportunidades o preferencias no anticipadas inicialmente. |

Tabla 3. Principales características del modelo de calidad en uso según ISO/IEC 2010:2011.   
Fuente: Elaboración propia a partir de ISO (2011)

Modelo de calidad del producto

ISO/IEC 2510 define un total de ocho características de calidad para el producto software (frente a las seis que definía ISO/IEC 9126) y otras treinta y una subcaracterísticas.

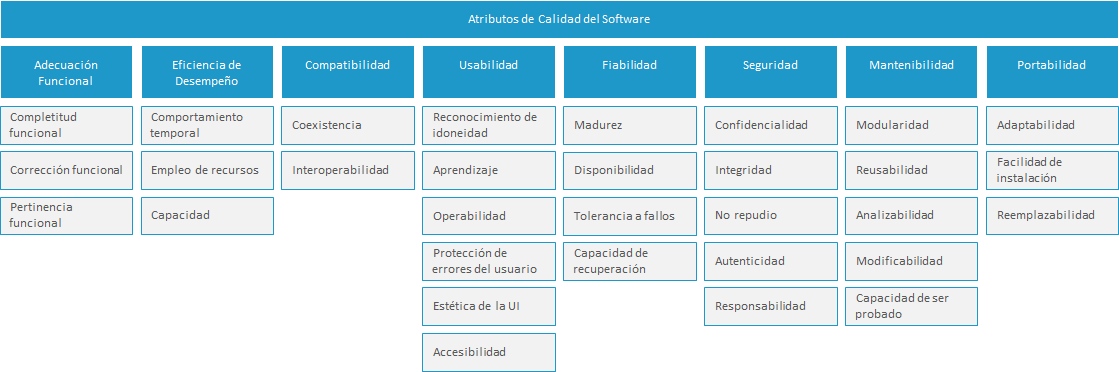


Figura 7. Atributos de calidad según el modelo de calidad de producto de la norma ISO/IEC 25010:2011. Fuente: Elaboración propia a partir de ISO (2011)

* **Adecuación funcional (*functional suitability*)**. Es la capacidad del producto para proporcionar las funcionalidades que satisfacen necesidades declaradas e implícitas. Contiene las siguientes subcaracterísticas:
  + **Completitud funcional**: grado en que las funcionalidades implementadas cubren todas las necesidades y objetivos especificados por el usuario.
  + **Corrección funcional**: capacidad del producto para proporcionar resultados correctos según el nivel de precisión requerida.
  + **Pertinencia funcional**: capacidad del producto para proporcionar funcionalidades apropiadas según las tareas y objetivos del usuario.
* **Eficiencia de desempeño** (*performance eficiency*). Representa el desempeño de sistema relativo a la cantidad de recursos empleados en determinadas condiciones de funcionamiento. Contiene las siguientes subcaracterísticas:
  + **Comportamiento temporal**: mide los tiempos de respuesta y procesamiento del sistema cuando realiza sus funcionales según un banco de pruebas (*benchmark*) predefinido.
  + **Utilización de recursos**: cantidades y tipos de recursos empleados en el funcionamiento del software bajo determinadas condiciones.
  + **Capacidad**: grado en que los límites máximos de un parámetro de un producto o sistema cumple con los requisitos.
* **Compatibilidad (*compatibility*).** Capacidad de dos o más sistemas, o componentes software, para intercambiar información y/o realizar sus funciones compartiendo el mismo entorno. Contiene las siguientes subcaracterísticas:
  + **Coexistencia**: capacidad de dos productos para coexistir en un entorno común, compartiendo recursos de manera que uno no perjudique al otro.
  + **Interoperabilidad**: capacidad de dos o más sistemas o componentes para intercambiar información y procesar la información recibida.
* **Usabilidad (*usability*)**. Capacidad del producto para ser utilizado, aprendido y comprendido resultando atractivo para el usuario, al ser empleado en determinadas condiciones. Contiene las siguientes subcaracterísticas:
  + **Capacidad para reconocer su adecuación**: permite al usuario detectar si el software es adecuado para sus necesidades.
  + **Capacidad de aprendizaje**: permite al usuario aprender su funcionamiento.
  + **Capacidad para ser usado**: permite al usuario manejarlo con facilidad.
  + **Protección contra errores de usuario:** previene que el usuario pueda cometer un error de manera involuntaria.
  + **Estética de la interfaz de usuario**: es la capacidad del sistema para agradar al usuario y proporcionarle una interacción satisfactoria.
  + **Accesibilidad:** es la capacidad del producto para ser utilizado por usuarios con características especiales o discapacidades.
* **Fiabilidad (*reliability*)**. Capacidad del producto para desempeñar su función cuando se utiliza en unas determinadas condiciones. Contiene las siguientes subcaracterísticas:
  + **Madurez**: es la capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
  + **Disponibilidad**: capacidad del producto para estar disponible en el momento que se requiere.
  + **Tolerancia a fallos**: capacidad para operar según lo que se espera a pesar de la ocurrencia de fallos en el hardware o en el software.
  + **Capacidad de recuperación**: capacidad del sistema para recuperarse de un fallo, restaurando la información y el estado deseado.
* **Seguridad (*security*)**. Capacidad para proteger los datos frente al intento de lectura o modificación de agentes no autorizados. Contiene las siguientes subcaracterísticas:
  + **Confidencialidad:** protección contra el acceso accidental o deliberado a la lectura de la información almacenada.
  + **Integridad**: capacidad para prevenir la modificación no autorizada de los datos del propio programa.
  + **No repudio**: capacidad para demostrar el origen de las acciones y eventos, de manera que no puedan ser negados con posterioridad.
  + **Responsabilidad**: capacidad para rastrear las acciones de un agente.
  + **Autenticidad**: capacidad para demostrar la identidad de un usuario o de un recurso del sistema.
* **Mantenibilidad (*mantainability*).** Capacidad del producto para ser modificado en función de las nuevas necesidades de manera eficiente. Se contemplan razones evolutivas, correctivas o perfectivas para realizar este mantenimiento. Contiene las siguientes subcaracterísticas:
  + **Modularidad**: es la capacidad del sistema para permitir la modificación o sustitución de un componente con un impacto mínimo en los demás.
  + **Reusabilidad:** capacidad de un elemento para ser empleado como integrante en diferentes desarrollos.
  + **Analizabilidad:** hace referencia a la facilidad para evaluar el impacto de un cambio sobre el sistema, o diagnosticar deficiencias e identificar su origen.
  + **Capacidad para ser modificado: e**s la capacidad del producto para ser modificado sin introducir defectos adicionales o afectar su desempeño.
  + **Testabilidad**, o capacidad para ser probado: es la facilidad para establecer criterios de prueba y realizar las pruebas correspondientes para determinar que se cumplen estos criterios.
* **Portabilidad** **(*portability*).** Capacidad del producto para ser transferido a un entorno operacional o de utilización diferente. Contiene las siguientes subcaracterísticas:
  + **Adaptabilidad**: es la facilidad para adaptar un mismo producto a diferentes entornos operacionales o de uso.
  + **Capacidad para ser instalado:** es la facilidad para instalar o desinstalar el producto en un determinado entorno.
  + **Capacidad para ser reemplazado:** es la capacidad de sustituir un producto para satisfacer el mismo propósito en el mismo entorno.

10.6. ISO/IEC 25012: la calidad de los datos

La calidad de los datos viene determinada por el grado en que satisfacen sus propios requisitos. **El modelo de calidad de datos está compuesto por un total de quince características que se clasifican en dos grandes categorías**:

* **Calidad de datos inherente**: tiene que ver con el grado en que las características de los datos tienen potencial intrínseco para satisfacer las necesidades del sistema.
* **Calidad de datos dependiente del sistema**: grado en que la calidad de los datos es alcanzada y preservada durante su utilización en un sistema informático. Depende del dominio tecnológico de uso y de las capacidades de los componentes del sistema.

Las diferentes características que contempla esta norma para la calidad de los datos aparecen recogidas en la tabla 4.

|  |  |
| --- | --- |
| **Características que determinan la calidad inherente** | |
| **Exactitud** | Grado en que los datos representan correctamente el verdadero valor del atributo que representan. |
| **Completitud** | Grado en que los datos asociados a la entidad representada tienen valores para todos los atributos esperados. |
| **Consistencia** | Grado en que los datos no se contradicen entre sí y son coherentes con otros datos. |
| **Credibilidad** | Grado en que los datos almacenan valores verdaderos y creíbles. |
| **Actualidad** | Grado en que los datos representan el verdadero valor de la entidad representada en un momento dado. |
| **Características que determinan la calidad dependiente del sistema** | |
| **Accesibilidad** | Grado en que resulta fácil consultar los datos en un contexto de uso y, particularmente, cuando las personas que los necesitan presentan algún tipo de necesidad especial. |
| **Conformidad** | Grado en que los atributos de los datos se adhieren a convenciones, normativas o estándares vigentes. |
| **Confidencialidad** | Grado en que se garantiza que solo acceden a los datos los usuarios autorizados. |
| **Eficiencia** | Grado en que el procesamiento de los datos es posible según el nivel de rendimiento esperado. |
| **Precisión** | Grado en que los datos proporcionan el nivel de exactitud necesario para el uso específico. |
| **Trazabilidad** | Grado en que la manera de acceder a los datos o sus modificaciones puede ser determinada o auditada. |
| **Comprensibilidad** | Grado en que los valores de los datos pueden ser interpretados y expresados mediante el lenguaje, simbología o unidades apropiados. |
| **Disponibilidad** | Grado en el que es posible acceder a los datos según se necesitan. |
| **Portabilidad** | Grado en el que es posible transferir los datos de un sistema a otro. |
| **Recuperabilidad** | Grado en que es posible mantener un cierto nivel de calidad en los datos a través del tiempo, e incluso en caso de fallo. |

Tabla 4. Características de calidad de los datos.  
Fuente: Elaboración propia a partir del Modelo de Calidad de Datos de ISO/IEC 2012:2008 ISO (2008)

10.7. ISO/IEC 25020: medición de la calidad del software

La norma ISO/IEC 25020:2016 presenta un modelo de referencia para la medida de la calidad del producto software. **Describe la relación entre el modelo de calidad, sus características, los atributos del producto y sus correspondientes medidas de calidad, funciones y métodos de medición** (ISO, 2007). La norma también proporciona ejemplos de criterios para seleccionar las medidas de calidad.

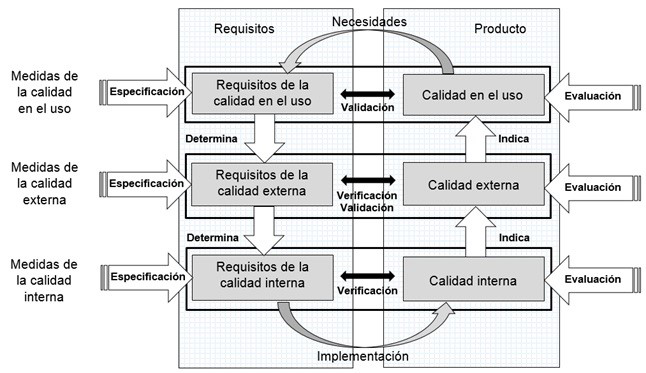


Figura 8. Relación entre los requisitos y la calidad del producto, y los diferentes niveles de evaluación.  
Fuente: Subcomité7 (2016)

La figura 9 representa las relaciones entre las diferentes normas que componen la división de medición de la calidad.

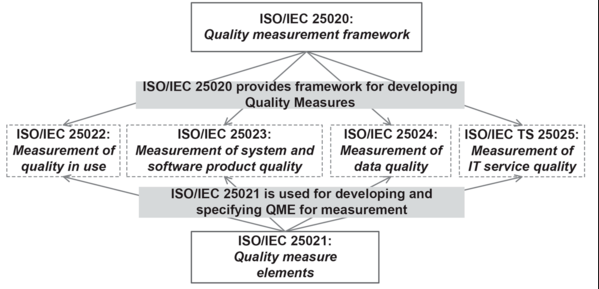


Figura 9. Estructura de la división de Medición de la Calidad (2502x) de SQuaRE.  
Fuente: Norma ISO/IEC 25020 (ISO, 2007)

10.8. ISO/IEC 25022, 25023 y 25024: métricas de calidad del software y de los datos

En la norma ISO/IEC 25010 se señala que para cada una de las características se debe determinar la capacidad asociada definiendo el conjunto de propiedades y métricas asociadas. En el caso de la calidad en uso, la calidad interna y externa del producto, y la calidad de los datos, las normas ISO/IEC 25022, 25023 y 25024 proporcionan respectivamente un conjunto de métricas que pueden ser utilizadas.

* **Las medidas de calidad interna** proporcionan una visión de caja blanca del sistema. Son apropiadas **durante las etapas de desarrollo del software** (análisis de requisitos, análisis del diseño y del código fuente).
* **Las medidas de calidad externa** ofrecen una visión de caja negra del producto cuando se ejecuta en un determinado entorno de hardware y sistema operativo. Son apropiadas durante las **fases de pruebas y en alguna etapa de operación**.
* **Las medidas de calidad en uso** permiten comprobar en qué medida el sistema satisface las necesidades del usuario, y deben ser adquiridas **cuando el sistema está en operación**.

Las métricas ofrecidas en estas normas pueden ser modificadas o sustituidas por otras, siempre que se especifique la relación de la métrica con la propiedad que se quiere evaluar dentro del modelo de calidad definido en ISO/IEC 25010. En todo caso, **estas métricas deben definirse y especificarse para cada proceso de evaluación concreto, siguiendo el modelo recogido en la tabla 5**.

|  |  |
| --- | --- |
| **Campo** | **Descripción** |
| **Subcaracterística** | Subcaracterística de calidad asociada según el modelo ISO/IEC 25010. |
| **Nombre de la métrica** | Nombre asignado a la métrica de calidad. |
| **Fase de ciclo de vida de calidad** | Calidad interna, calidad externa o calidad en uso. |
| **Propósito de la métrica** | Motivo por el cual se ha seleccionado la métrica. |
| **Método de aplicación** | Manera en que se debe aplicar la métrica. |
| **Fórmula y cálculo de datos** | Establece los datos o parámetros a medir y su significado, y la fórmula de medición específica que los combina en un valor de medida. |
| **Valor esperado** | Proporciona un rango de valores preferibles y recomendados. |
| **Tipo de medida** | Especifica el tipo de medida seleccionada: tamaño (de una función, de una clase), tiempo (intervalo de tiempo), contar (número de cambios, de fallos…). |

Tabla 5. Campos necesarios para la definición de una métrica de calidad de producto software.  
Fuente: ISO/IEC 25020 (ISO, 2007)

A modo de ejemplo se recogen en las siguientes tablas algunas métricas de calidad interna, externa y en uso.

|  |  |
| --- | --- |
| **Campo** | **Descripción** |
| **Subcaracterística** | Completitud funcional. |
| **Nombre de la métrica** | **Completitud de la implementación funcional.** |
| **Fase de ciclo de vida de calidad** | Interna/Externa. |
| **Propósito de la métrica** | Determinar lo completa que es la implementación de acuerdo con la especificación de requisitos. |
| **Método de aplicación** | Contar el número de funciones indicada en la especificación de requisitos, y el número de funcionales que faltan o son incorrectas. |
| **Fórmula y cálculo de datos** | X = A/B, siendo:   * A el número de funciones incorrectas o no implementadas. * B el número de funciones establecidas en la especificación de requisitos. |
| **Valor esperado** | 0 £ X £ 1, siendo preferibles valores próximos a 0. |
| **Tipo de medida** | * X de tipo contable/contable. * A y B de tipo contable. |

Tabla 6. Ejemplo de métrica relacionada con la característica de adecuación funcional: **Completitud de la implementación funcional**. Fuente: Elaboración propia a partir de ISO/IEC 25023 (ISO, 2016b)

|  |  |
| --- | --- |
| **Campo** | **Descripción** |
| **Subcaracterística** | Corrección funcional. |
| **Nombre de la métrica** | **Precisión computacional**. |
| **Fase de ciclo de vida de calidad** | Interna/Externa. |
| **Propósito de la métrica** | Determinar con qué frecuencia se producen resultados inexactos. |
| **Método de aplicación** | Contar el número de resultados inexactos y el tiempo de operación asociado. |
| **Fórmula y cálculo de datos** | *X = A/T*, siendo:   * *A* el número de cálculos inexactos encontrados. * *T* el tiempo de operación. |
| **Valor esperado** | El más cercano a 0 es el preferible. |
| **Tipo de medida** | * *X* de tipo contable/tiempo. * *A* de tipo contable. * *T* de tipo tiempo. |

Tabla 7. Ejemplo de métrica relacionada con la característica de adecuación funcional: **Precisión computacional**. Fuente: Elaboración propia a partir de ISO/IEC 25023(ISO, 2016b)

|  |  |
| --- | --- |
| **Campo** | **Descripción** |
| **Subcaracterística** | Efectividad. |
| **Nombre de la métrica** | **Completitud de la tarea**. |
| **Fase de ciclo de vida de calidad** | Uso. |
| **Propósito de la métrica** | Determinar qué cantidad de tareas son completadas correctamente. |
| **Método de aplicación** | Contar el número de tareas intentadas y el número de tareas completadas. |
| **Fórmula y cálculo de datos** | *X = A/B,* siendo:   * *A* el número de tareas completadas. * *B* el número de tareas intentadas. |
| **Valor esperado** | 0 £ *X £* 1, siendo preferibles valores próximos a 1. |
| **Tipo de medida** | * *X* de tipo contable/contable. * *A* y *B* de tipo contable. |

Tabla 8. Ejemplo de métrica relacionada con la característica en uso de efectividad: **Completitud de la tarea**.  
Fuente: Elaboración propia a partir de ISO/IEC 25022 (ISO, 2016a)

Existen infinidad de métricas que tradicionalmente se han venido empleando en la evaluación de la calidad del software desde diferentes puntos de vista. En el trabajo de Constanzo, Casas y Marcos (2014) encontrarás una recopilación de ellas, agrupadas según su utilidad a distintos enfoques del proceso de desarrollo: desarrollo orientado a objetos, desarrollo basado en componentes, desarrollo orientado a aspectos y métodos ágiles.

10.9. ISO/IEC 25040: evaluación del software

La norma **ISO/IEC 25040:2011** contiene requisitos y recomendaciones para la evaluación de la calidad del producto software. Puesto que está dirigida a todo aquel responsable de la evaluación del producto, resulta apropiada para desarrolladores, adquisidores y también evaluadores independientes de productos.

Define un modelo de proceso de evaluación formado por **cinco actividades principales**, como se muestra en la figura 10.

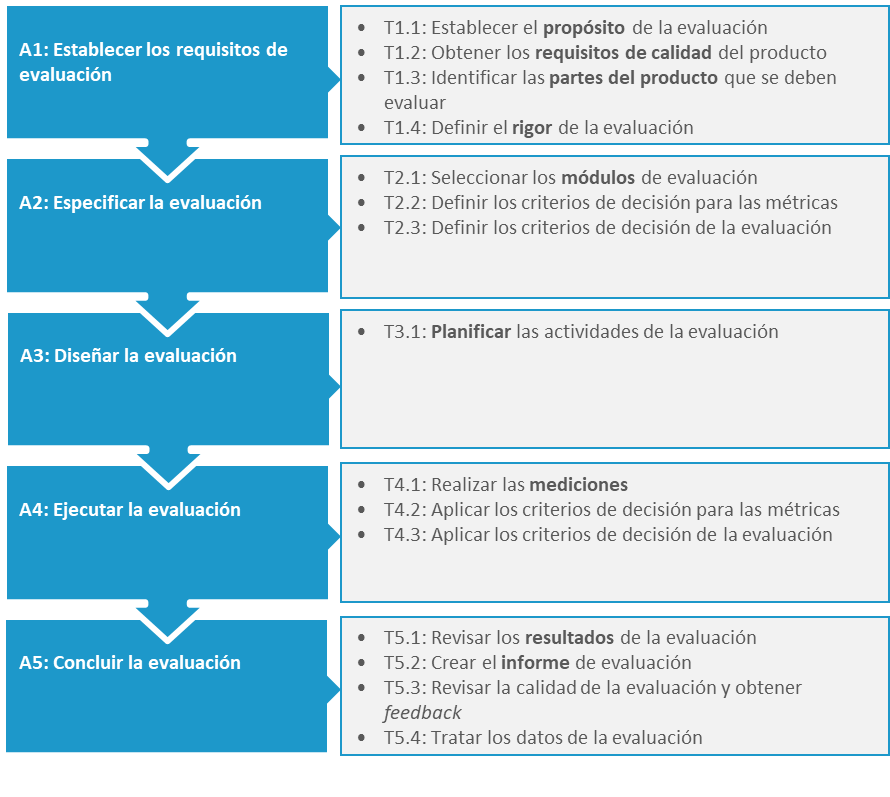


Figura 10. Actividades y tareas definidas para el proceso de evaluación según ISO/IEC 25040:2011

1. **Durante la primera actividad se definen los requisitos de la evaluación**. Para ello, se comienza definiendo el **propósito de la evaluación** (decidir si se compra un producto, asegurar su calidad, comparar la calidad de un producto con otros semejantes, etc.). Para obtener los requisitos de calidad es necesario **conocer los puntos de vista** de las partes interesadas (usuarios, proveedores, compradores…) y aplicar algún modelo de calidad. También es necesario **definir qué partes se van a evaluar**, así como **el rigor de la evaluación**, que dependerá de su propósito y del uso previsto para el software. En función de este rigor será posible **elegir las técnicas** a utilizar y los **resultados esperados** de la evaluación.
2. A continuación, **se especifican los módulos de evaluación**, que están compuestos por métricas, herramientas y técnicas de medición. Para **definir las métricas** apropiadas se puede utilizar la norma **ISO/IEC 25020**. Los criterios de decisión para las métricas son umbrales numéricos que se pueden relacionar con los requisitos de calidad y con los criterios de evaluación. Los criterios de evaluación permiten evaluar las características de calidad teniendo en cuenta las métricas definidas.
3. La siguiente actividad consiste en **planificar las actividades de evaluación**. Para ello, se tendrá en cuenta la disponibilidad de recursos (materiales y humanos), el presupuesto y las herramientas necesarias.
4. La ejecución de la evaluación comienza por la **obtención de las medidas teniendo en cuenta las métricas seleccionadas**. Sobre estos valores se aplican los criterios de decisión definidos en la actividad 2 y, finalmente, se aplican los criterios de decisión a nivel de característica y subcaracterística de calidad. El resultado es una **valoración de la calidad del producto analizado**.
5. Por último, se concluye la **evaluación de los resultados**, donde participan tanto el evaluador como el cliente de la evaluación. Se elabora un **informe** que contiene los requisitos de la evaluación, los resultados y cualquier otra información, como limitaciones y restricciones del proceso. A la luz de todo ello se revisa el propio proceso y su validez, con el fin de obtener retroalimentación para la mejora de futuras evaluaciones.

10.10. El proceso de certificación

**La certificación de calidad de un producto mediante SQuaRE permite por un lado a las empresas que fabrican software conocer la calidad de sus productos y, por otro, a las empresas compradoras les ayuda a decidirse por la solución más adecuada.**

Los principales beneficios que ofrece la obtención de una certificación son los siguientes (AENOR, 2018a; Fernández, Rodríguez, & Piattini, 2013):

* Permite a los fabricantes de software **diferenciarse de sus competidores**.
* Permite **establecer acuerdos de servicio** con un proveedor, definiendo parámetros de calidad que deben cumplirse al entregar el producto.
* Facilita la **detección de defectos** en un producto y su mejora antes de la entrega, ahorrando costes en la fase de mantenimiento posterior.
* Facilita la **evaluación y control del rendimiento** del producto.

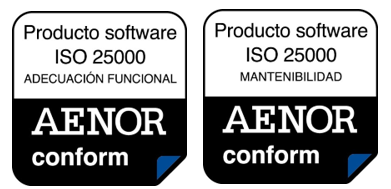


Figura 11. Logotipo de la marca AENOR conform que certifica la adecuación funcional (izquierda) y la mantenibilidad (derecha) de un producto software, según ISO/IEC 25000 (SQuaRE). Fuente: AENOR (2018b)

El proceso de evaluación y certificación conforme a la norma ISO/IEC25000 consta de seis pasos, como se aprecia en la figura 12:

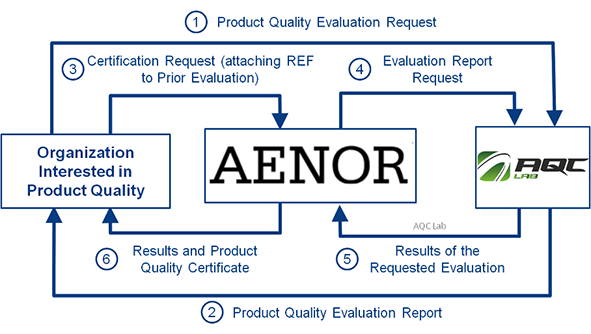


Figura 12. Proceso de evaluación y certificación del producto software.  
Fuente: ISO25000.com (2018b)

1. **La organización interesada solicita una evaluación a un laboratorio acreditado.** Deben indicar qué características del producto desean evaluar. El laboratorio envía un contrato de evaluación especificando las condiciones de servicio. Si es aceptado, la organización envía el software al laboratorio, que realiza las pruebas empleando un *framework* (modelo, proceso y herramientas) basado en SQuaRE y acreditado por ENAC (Entidad Nacional de Acreditación).
2. **El resultado de la evaluación se envía a la organización solicitante**. Si es desfavorable, la organización puede recurrir a la ayuda de un consultor de calidad que le ayude a mejorar su producto (ver ecosistema de la figura 13), hasta que consigue superar la evaluación. El proceso finaliza cuando se obtiene una evaluación favorable por parte del laboratorio acreditado.
3. **La organización solicita el certificado a la entidad de certificación** (en este caso AENOR), indicando la referencia de la evaluación realizada por el laboratorio acreditado.
4. **La entidad de certificación solicita el resultado de la evaluación** al laboratorio.
5. **El laboratorio proporciona la información** a la entidad de certificación.
6. **Finalmente, la organización certificadora visita al solicitante** para comprobar el producto y sus características, y emite un informe junto con el certificado que reconoce la calidad del producto evaluado.

Dentro del ecosistema que se representa en la figura 13, merece destacar la importancia de los **consultores de calidad**, que ayudan a las empresas en la mejora de sus productos para poder superar un proceso de certificación, y las compañías que desarrollan las herramientas que permiten realizar las mediciones necesarias.

En este contexto podemos señalar las **herramientas desarrolladas** por **Kiuwan** (Kiuwann, 2018) o **SonarQube** (SonarQube, 2018).

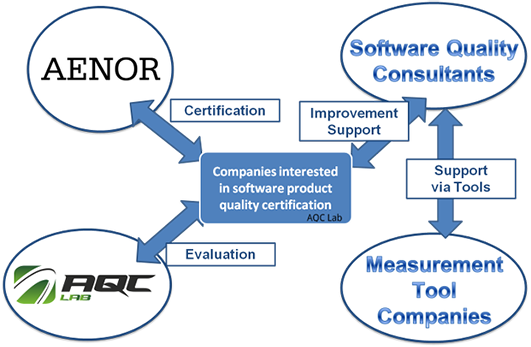


Figura 13. Ecosistema de la evaluación y certificación del producto software. Fuente: ISO25000.com (2018c)

En el caso concreto del laboratorio de evaluación **AQC Lab** (Alarcos Quality Center) se utilizan **tres elementos para realizar las evaluaciones de calidad** (Rodríguez & Piattini, 2014):

1. **El proceso de evaluación**, que adopta las actividades definidas en ISO/IEC 25040 y las completa con roles y actividades desarrolladas internamente. Este proceso ofrece como resultado un informe que indica el nivel de calidad alcanzado por el producto en cuanto a su mantenibilidad, su adecuación funcional, o la calidad de los datos.
2. **El modelo de calidad**, que define las características y métricas necesarias para evaluar el producto. Aunque se evalúan muchas de las características presentadas en ISO/IEC 25010, el proceso se centra inicialmente en la mantenibilidad, al ser la fase de mantenimiento la más cara del ciclo de vida y ser esta una de las evaluaciones más demandadas por los clientes.
3. **El entorno de evaluación**,que permite automatizar en gran medida la evaluación. Utiliza herramientas que obtienen información del producto, combinan los valores obtenidos y finalmente presentan la información de una manera accesible.

El proceso de evaluación y certificación de la calidad de los datos es muy similar al anteriormente descrito para la calidad del producto software (Caballero, Rodríguez, & Fernández, 2017).

10.11. Referencias bibliográficas

AENOR. (2018a). El Certificado AENOR de Calidad del Producto Software ISO 25000. Recuperado de <https://www.aenor.com/asociacindocumentos/info-apoyo/calidad/17-%20Ventajas%20del%20certificado%20IS0%2025000%20jun-18.pdf>

AENOR. (2018b). Sistemas de Gestión de la Calidad del Producto Software ISO 25000. Recuperado de <https://www.aenor.com/certificacion/tecnologias-de-la-informacion/producto-software>

Bloom, J. (2011). Titanic Dilemma: The Seen Versus the Unseen [Artículo web]. Recuperado de <https://www.castsoftware.com/blog/titanic-dilemma-the-seen-versus-the-unseen>

Boehm, B. W. (1978). *Characteristics of software quality*. Amsterdam: North-Holland Pub. Co.

Caballero, I., Rodríguez, M. y Fernández, C. M. (2017). Calidad de datos digitales certificada. *AENOR: Revista de la normalización y la certificación*, 331, 36-39.

Calero, C., Piattini, M. G. y Moraga, M. Á. (2010). *Calidad del producto y proceso software*. Madrid: Editorial Ra-Ma.

Callejas-Cuervo, M., Alarcón-Aldana, A. C. y Álvarez-Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado, 13*(1), 236-250. <https://doi.org/10.18041/entramado.2017v13n1.25125>

CISQ. (2018). Consortium for IT Software Quality [Web]. Recuperado de 2018, de <https://it-cisq.org/>

Fernández, C. M., Rodríguez, M. y Piattini, M. (2013). ISO/IEC 25000: Calidad del producto software. *AENOR: Revista de la normalización y la certificación*, 288, 30-35.

Garzás, J. (2013, noviembre 7). El otro estándar de calidad del producto software, el CISQ [Blog]. Recuperado de <http://www.javiergarzas.com/2013/11/estandar-de-calidad-del-producto-software-cisq.html>

Grady, R. B., & Caswell, D. L. (1987). *Software Metrics: Establishing a Company-Wide Program* (1st Ed.). Englewood Cliffs, N.J: Prentice Hall.

ISO. (2006). ISO/IEC 25062:2006 - SQuaRE: Common Industry Format (CIF) for usability test reports. Recuperado de <https://www.iso.org/standard/43046.html>

ISO. (2007). ISO/IEC 25020:2007 - SQuaRE: Measurement reference model and guide. Recuperado de <https://www.iso.org/standard/35744.html>

ISO. (2008). ISO/IEC 25012:2008 - SQuaRE: Data quality model. Recuperado de <https://www.iso.org/standard/35736.html>

ISO. (2011). ISO/IEC 25010:2011 - SQuaRE: System and software quality models. Recuperado de <https://www.iso.org/standard/35733.html>

ISO. (2014a). ISO/IEC 25000:2014 - SQuaRE: Guide to SQuaRE. Recuperado de <https://www.iso.org/standard/64764.html>

ISO. (2014b). ISO/IEC 25051:2014 - SQuaRE: Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing. Recuperado de <https://www.iso.org/standard/61579.html>

ISO. (2015). ISO 9000:2015 - Quality management systems - Fundamentals and vocabulary. Recuperado de <https://www.iso.org/standard/45481.html>

ISO. (2016a). ISO/IEC 25022:2016 - SQuaRE: Measurement of quality in use. Recuperado de <https://www.iso.org/standard/35746.html>

ISO. (2016b). ISO/IEC 25023:2016 - SQuaRE: Measurement of system and software product quality. Recuperado de <https://www.iso.org/standard/35747.html>

ISO. (2018). Standards catalogue: 35.080 (Software). Recuperado de <https://www.iso.org/ics/35.080/x/>

iso25000.com. (2018a). Portal ISO 25000. Recuperado de <http://iso25000.com/>

iso25000.com. (2018b). Software Product Evaluation and Certification Process. Recuperado de <http://iso25000.com/index.php/en/product-evaluation/process>

iso25000.com. (2018c). Software Product Quality Evaluation and Certification Ecosystem. Recuperado de <http://iso25000.com/index.php/en/product-evaluation/ecosystem>

Kiuwann. (2018). Seguridad de Aplicaciones 360º [Web]. Recuperado de <https://www.kiuwan.com/es/>

Maibaum, T., & Wassyng, A. (2008). A Product-Focused Approach to Software Certification. *Computer, 41*(2), 91-93. <https://doi.org/10.1109/MC.2008.37>

McCall, J. A., Richards, P. K. & Walters, G. F. (1977a). *Factors in Software Quality. Volume 2. Metric Data Collection and Validation*. Fort Belvoir, VA: Defense Technical Information Center. <https://doi.org/10.21236/ADA049015>

McCall, J. A., Richards, P. K., & Walters, G. F. (1977b). *Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality.* Fort Belvoir, VA: Defense Technical Information Center. <https://doi.org/10.21236/ADA049014>

McConnell, S. (1993). *Code complete* (1st Ed.). Redmond, Wash: Microsoft Press.

RAE. (2018). Diccionario de la lengua española: definición de «calidad». Recuperado de <http://dle.rae.es/?w=calidad>

Rodríguez, M. & Piattini, M. (2014). Software Product Quality Evaluation Using ISO/IEC 25000. *ERCIM News*. *Special theme: Software Quality*, 99, 39-40.

Solo, M. A., Casas, S.I. y Marcos, C.A. (2014). Comparación de modelos de calidad, factores y métricas en el ámbito de la ingeniería de software. Informe Científico Técnico UNPA, Vol. 6, Nº. 1, 1–36. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/5123569.pdf>

SonarQube. (2018). Continuous Inspection [Web]. Recuperado de <https://www.sonarqube.org/>

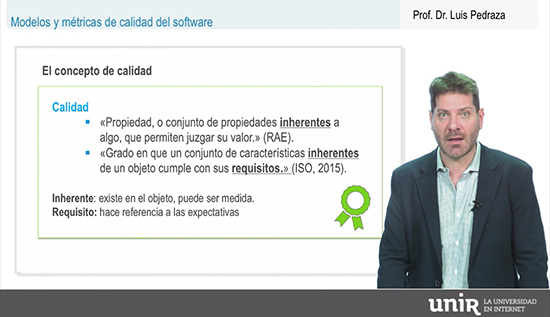
Subcomité7. (2016). NC ISO/IEC 25020:2016 SQuaRE – Modelo de referencia y guía para las mediciones. Recuperado de [https://subcomite7.cubava.cu/2016/07/11/nc-isoiec-250202016-square-modelo-de-referencia-y-guia-para-las-mediciones/#.W9tN\_npKjUJ](#.W9tN_npKjUJ)

XBOSOFT. (2018). How to Start with Software Quality Metrics [Web]. Recuperado de <https://xbosoft.com/software-quality-metrics/>

A fondo

Lección magistral: Modelos y métricas de calidad del software

En esta lección magistral revisaremos el concepto de calidad del software, resaltando la dificultad que presenta su medición y análisis. También se explicarán los diferentes niveles en los que se puede analizar la calidad del producto software: interno (dependiente de las características del propio código), externo (que depende además del entorno de ejecución y su utilización) y en uso (más relacionado con la percepción que tiene el usuario final durante su utilización). Además, revisaremos los modelos de calidad que propone el estándar SQuaRE y mostraremos algún ejemplo de métricas que permiten evaluar la calidad en uso de un producto.



Accede a la lección magistral a través del aula virtual.

ISO 25000: Software Product Quality

En esta página web encontrarás una amplia descripción del conjunto de normas ISO/IEC 25000, junto con noticias de actualidad e información sobre el proceso de evaluación y certificación de productos software y laboratorios acreditados.



Accede a la web a través del aula virtual o desde la siguiente dirección web:

https://iso25000.com

ISO/IEC 25000: Calidad del producto software

Fernández, C. M., Rodríguez, M. y Piattini, M. (2013). ISO/IEC 25000: Calidad del producto software*. AENOR: Revista de la normalización y la certificación*, 288, 30-35.

En este artículo encontrarás una introducción general a la importancia de la calidad del producto software, así como al proceso de certificación de calidad del producto software en el caso de AENOR. También se menciona la experiencia piloto realizada en 2013 con tres empresas participantes, donde se constató la mejora de producción que supone el proceso y un ahorro de costes de mantenimiento de aproximadamente un 40 %.

Accede al documento a través del aula virtual o desde la siguiente dirección web:

https://www.aenor.com/Certificacion\_Documentos/Folletos/calidad\_producto\_software\_ISO25000.pdf

Página Web de AQC Lab

AQC Lab es un *spin–off* de la Universidad de Castilla La Mancha especializado en la evaluación de la calidad del software. Además, es el primer laboratorio acreditado a nivel mundial para la evaluación de la calidad de los datos y del producto software. En su página web encontrarás información sobre los diferentes procesos de evaluación que realizan, actualmente centrados en la mantenibilidad, la adecuación funcional y la calidad de datos.



Accede a la web a través del aula virtual o desde la siguiente dirección web:

http://www.aqclab.es

Webinar: ISO 25000 Calidad del producto software

En este *webinar* organizado por ArgenTesting en octubre de 2018, Moisés Rodríguez, director de AQC Lab, habla de la importancia de la calidad del producto software y la manera de evaluarlo bajo la norma ISO/IEC 25000.



Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

<https://youtu.be/CWbftDKbyyo>

Accede a otros *webinars* organizados por ArgenTesting a través del aula virtual o desde la siguiente dirección web:

<https://argentesting.com/webinars/>

Auditoría compatibilidad producto software

Sánchez–Carrasco, M. (2015). *Auditoría compatibilidad producto software* (Trabajo de Fin de Grado). Universidad Complutense de Madrid, Madrid, España.

Este trabajo de fin de grado analiza la calidad del producto software desde el punto de vista de su portabilidad. En el capítulo 3 encontrarás un conjunto de métricas externas e internas de portabilidad definidas en las normas ISO 9126-2 e ISO 9126-3, y algunas adicionales definidas por el propio autor. En el capítulo 4 se presenta un caso práctico donde el proceso de evaluación se aplica a las páginas web de cuatro universidades españolas.

Accede al documento a través del aula virtual o desde la siguiente dirección web:

<https://eprints.ucm.es/33446/>

Test

1. ¿Qué tres enfoques del concepto de calidad se contemplan en las normas ISO/IEC 9126 y en la ISO/IEC 25000?

A. Calidad inherente, calidad intrínseca y calidad medible.

B. Calidad interna, calidad del proceso y calidad externa.

C. Calidad interna, calidad externa y calidad en uso.

D. Ninguna de las respuestas anteriores es correcta.

1. ¿Sobre qué normas se apoya la nueva familia ISO/IEC 25000 para la calidad del software?

A. ISO 9000 e ISO 9600.

B. ISO/IEC 14598 e ISO/IEC 9000.

C. ISO/IEC 14598 e ISO/IEC 9126.

D. Ninguna de las respuestas anteriores es correcta.

1. ¿Qué cuatro partes tiene la norma ISO/IEC 9126:2001?

A. Una para definir el modelo de calidad y otras tres relacionadas con las métricas de calidad interna, externa y en uso.

B. Una para definir el modelo de calidad y otras tres para definir las fases del proceso de medida de la calidad.

C. Contiene cuatro partes en las que se establecen procesos para realizar la evaluación de la calidad del software.

D. Las mismas partes que la norma ISO/IEC 25000.

1. ¿En qué división de SQuaRE define el modelo de calidad del software?

A. 2501x.

B. 2502x.

C. 2503x.

D. 2504x.

1. ¿Cuál de las siguientes es una característica del modelo de calidad en uso según la norma SQuaRE?

A. Mantenibilidad.

B. Satisfacción.

C. Portabilidad.

D. Todas las respuestas anteriores son correctas.

1. ¿Qué elementos define el modelo de calidad del producto en la norma ISO/IEC 25010?

A. Un conjunto de características y un conjunto de métricas de calidad asociadas.

B. Un conjunto de actividades recomendadas para realizar el proceso de medida de la calidad del software.

C. Un conjunto de características de calidad que se descomponen en subcaracterísticas más concretas.

D. Todas las respuestas anteriores son correctas.

1. ¿Cuál de las siguientes definiciones se ajusta mejor al concepto de «compatibilidad» según el modelo de calidad de SQuaRE?

A. Capacidad del producto para ser utilizado, aprendido y comprendido resultando atractivo para el usuario, al ser empleado en determinadas condiciones.

B. Capacidad del producto para desempeñar su función cuando se utiliza en unas determinadas condiciones.

C. Capacidad del producto para ser modificado en función de las nuevas necesidades de manera eficiente.

D. Capacidad de dos o más sistemas o componentes software para intercambiar información y/o realizar sus funciones compartiendo el mismo entorno.

1. ¿A qué se refiere la calidad de datos inherente dentro del modelo de calidad de datos definido en ISO/IEC25012?

A. A la calidad de los datos que dependen del propio potencial intrínseco para satisfacer las necesidades del sistema.

B. Al grado en que se alcanza un nivel de calidad de datos durante su utilización en un sistema concreto.

C. Las dos respuestas anteriores son correctas.

D. Ninguna de las respuestas anteriores es correcta.

1. ¿Qué actores principales intervienen como mínimo en un proceso de certificación según la norma ISO/IEC 25000?

A. La organización solicitante, el laboratorio acreditado y el consultor de calidad.

B. La organización solicitante, el laboratorio acreditado y la entidad de certificación.

C. El laboratorio acreditado, la entidad de certificación y el consultor de calidad.

D. Ninguna de las respuestas anteriores es correcta.

1. ¿Cuál de las siguientes es una subcaracterística de la característica de calidad de producto «compatibilidad» según SQuaRE?

A. La coexistencia.

B. La interoperabilidad.

C. Las respuestas A y B son correctas.

D. Ninguna de las respuestas anteriores es correcta.