

ISO 25010 y el desarrollo de software



Nicolás F. Ormeño Rojas
May 15, 2019 · 5 min read



Photo by [NESA by Makers](#) on [Unsplash](#)

Sin duda alguna, la calidad es la piedra angular en todo, no sólo en el desarrollo de sistemas. No obstante, en esta ocasión revisaremos específicamente el desarrollo de software, y aquí es donde entra ISO/IEC 25000 (SQuaRE — System and Software Quality Requirements and Evaluation), la cual es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

Cuando hablamos de ISO 25010, estamos hablando de un modelo de calidad compuesto de 8 características que se relacionan con las propiedades estáticas del software y las propiedades dinámicas del sistema informático. El modelo es aplicable tanto a los sistemas informáticos como a los productos de software.





Photo by [abstracta](https://www.abstracta.us)

Funcionalidad (Functional)

El sistema satisface las necesidades declaradas cuando se utilizan en condiciones específicas. Básicamente es que el sistema haga lo queremos que haga, o en otras palabras, es funcionalmente adecuado si cumple con todos los requisitos, los cubre correctamente y solo hace las cosas que son necesarias y adecuadas para completar las tareas.

Algunas pruebas para saber si cumplimos o no con la funcionalidad son las pruebas de regresión, pruebas del sistema, pruebas unitarias, pruebas de procedimientos, etc.

Rendimiento (Performance)

Sabemos que un sistema con alto rendimiento es rápido, escalable y estable incluso cuando hay una gran cantidad de usuarios concurrentes, por lo que debemos evaluar cómo se comporta el sistema en ciertas situaciones con diferentes tamaños de carga, es decir, que tenga un rendimiento de acuerdo a lo esperado, y con comportamiento, nos referimos a comportamiento en el tiempo de carga, uso de recursos, eficiencia y capacidad.

Algunas pruebas de rendimiento que podemos realizar, son de carga, resistencia, peak, etc.

Compatibilidad (Compatibility)

Nuestro software debe ser compatible con su entorno, por lo que este debe ser compatible con el hardware y software que lo rodea.

Por un lado debe existir coexistencia, el cual se refiere a la capacidad de desempeñarse de manera eficiente al compartir un entorno y recursos comunes con otros, sin un impacto perjudicial en sí mismo o en otro. También debe existir interoperabilidad, lo que significa la posibilidad de intercambiar información y usar dicha información con otros sistemas. Debido a esto, es muy importante que desarrollemos basado en estándares, ya que estos mismos nos ayudan a sopesar estos problemas.

Las pruebas que se pueden realizar para garantizar estos atributos de calidad son las pruebas de conformidad, compatibilidad, interoperabilidad y conversión.

Usabilidad (Usability)

Debe ser usable por nuestros usuarios objetivos. Un sistema es altamente utilizable si los usuarios tienden a no cometer muchos errores, aprenden a usarlo rápidamente, realizan tareas de manera oportuna y están satisfechos con él en general. Es por esto que nuestro

sistema debe ser operativo y poseer protección contra errores de usuario.

Aquí podemos realizar pruebas de UX y de accesibilidad (como localización e internacionalización), entre otras.

Confiabilidad (Reliability)

Algo sencillo para que el usuario confíe en nuestro sistema es que nuestro sistema no se caiga (o al menos el usuario no lo note), por lo que debemos trabajar en la probabilidad de una operación específica sin fallas durante un período de tiempo específico en un entorno específico.

Aquí nos fijamos en la madurez (grado en el que satisface las necesidades de confiabilidad), disponibilidad (cuán accesible es), capacidad de recuperación (qué tan bien puede regresar a su estado anterior deseado después de una interrupción o falla) y tolerancia a fallas (qué tan bien puede operar y hacer frente a la presencia de fallas de hardware o software).

Algunas pruebas son las de respaldo, recuperación y recuperación de desastres.

Seguridad (Security)

Sin duda alguna, seguridad es un término que últimamente está cada vez más en boca de todos. La información es muy importante, es por esto que debemos asegurarnos de proteger la información y los datos de nuestros usuarios. ¿Nos imaginamos que sucede si sale a la luz pública que la información de los usuarios de un banco ha sido filtrada?

Debemos tener en mente la confidencialidad (acceso a los datos), integridad (grado de protección), no repudio (saber cuándo y dónde se llevó a cabo una acción), responsabilidad y autenticidad (afirmar que el usuario es quién dice ser).

Aquí podemos realizar pruebas que analizan estos atributos de seguridad son pruebas de penetración, pruebas de vulnerabilidad, ethical hacking, y static analysis.

Mantenibilidad (Maintainability)

Nuestro sistema debe ser fácil de mantener, incluso analizarlo, cambiarlo y probarlo (evitar código spaghetti). Debido a aquello deber pensar en desarrollar de forma modular, en la reutilización, la capacidad de análisis, la modificabilidad y la capacidad de prueba.

Las pruebas asociadas con la capacidad de mantenimiento incluyen la revisión de pares o el análisis estático.

Portabilidad (Portability)

¿Podemos cambiar nuestro entorno sin mayores cambios? Con portabilidad nos referimos a la facilidad con la que un sistema se puede mover de un entorno a otro, por ejemplo la posibilidad de que una aplicación móvil pase una versión de sistema operativo a otro.

Se realizan pruebas para ver qué tan bien se puede instalar o desinstalar en un entorno específico. La capacidad de reemplazo es la capacidad de la aplicación para ser utilizada en lugar de otra para el mismo propósito y en el mismo entorno. La adaptabilidad se refiere a su capacidad para adaptarse a diferentes entornos.



Photo by iso25000.com

Conclusión

Debemos tener claro que la calidad, o el grado de calidad es subjetivo, pero seguir normas, estándares, etc. es algo que nos ayuda a crear.

¿Porqué decimos que la calidad es subjetiva? Esto es debido a que cada uno de nuestros usuarios puede fijarse en distintos elementos, por ejemplo, para un usuario, puede ser importante que sea fácil de entender, y que no le importe si se ejecuta un poco más lento, pero puede que haya otros usuarios que ni siquiera consideren el uso de una aplicación si no se ejecuta lo suficientemente rápido .

Es por esto que, seguir las normas y estándares nos permite tener un piso mínimo de entrega.

QA

Software Development

Iso 25010

Medium

[About](#) [Help](#) [Legal](#)

Get the Medium app

Download on the
App Store

GET IT ON
Google Play