

Características de la calidad del software

Repasa la lista y piensa en tu producto y en sus características. Añade aspectos específicos para ajustar la lista a tu contexto y transfórmala a tu gusto.

Potencial. *¿El producto puede realizar funciones valiosas?*

- *Totalidad*: todas las funciones importantes deseadas por los usuarios finales están disponibles.
- *Precisión*: cualquier cálculo o resultado en el producto es correcto y se presenta con dígitos significativos.
- *Eficiencia*: el producto realiza sus funciones de un modo eficiente (sin hacer lo que no se espera que haga).
- *Interoperabilidad*: las distintas funcionalidades interactúan entre ellas de la mejor manera posible.
- *Concurrencia*: capacidad del producto de ejecutar tareas múltiples en paralelo y funcionar al mismo tiempo que otros procesos.
- *Agnosticismo de datos*: el producto soporta todos los formatos de datos posibles y puede gestionar el "ruido" de los mismos.
- *Extensibilidad*: capacidad, por parte de clientes o terceros, de añadir funcionalidades o cambiar el comportamiento del producto.

Fiabilidad. *¿Puedes confiar en el producto en situaciones de distinta índole y dificultad?*

- *Estabilidad*: el producto no debe sufrir caídas, excepciones no controladas ni errores en la secuencia de comandos.
- *Robustez*: el producto gestiona adecuadamente tanto los errores previstos como los no previstos.
- *Gestión del estrés*: ¿cómo hace frente el sistema a situaciones en las que se superan los límites existentes?
- *Recuperabilidad*: el producto puede recuperarse y seguir siendo usado tras un error crítico.
- *Integridad de los datos*: los distintos tipos de datos se mantienen íntegros a lo largo de todo el producto.
- *Inocuidad*: el producto no puede hacer daño a las personas o sus posesiones.
- *Recuperación frente a desastres*: ¿y si sucede algo realmente malo?
- *Integridad*: ¿el comportamiento del producto es consistente, predecible y confiable?

Usabilidad. *¿El producto es fácil de usar?*

- *Factibilidad*: el producto invita a descubrir sus propias posibilidades.
- *Carácter intuitivo*: es fácil entender y explicar lo que el producto puede hacer.
- *Minimalismo*: no hay redundancia en el contenido o apariencia del producto.
- *Facilidad de aprendizaje*: es rápido y fácil aprender a usar el producto.
- *De fácil memorización*: una vez has aprendido a hacer algo, no se te olvida.
- *De fácil descubrimiento*: la información y capacidades del producto pueden ser descubiertas mediante la exploración de la interfaz de usuario.
- *Operabilidad*: un usuario experimentado puede realizar acciones comunes con rapidez.
- *Interactividad*: los estados del producto y las posibilidades de interactuar con la aplicación (vía interfaz de usuario o APIs) son fáciles de entender.
- *Control*: el usuario debe sentirse con control sobre los procesos del software.
- *Claridad*: ¿está todo indicado explícitamente y con detalle, en un lenguaje comprensible que no deje lugar a dudas?
- *Errores*: hay mensajes de error informativos, es difícil cometer fallos y fácil recuperarse tras cometerlos.
- *Consistencia*: el comportamiento es el mismo a través del producto, y hay un único look & feel.
- *Configuración*: las opciones y los comportamientos por defecto pueden personalizarse, para una mayor flexibilidad.
- *Accesibilidad*: el producto puede ser utilizado por la mayor cantidad posible de personas y cumple los estándares de accesibilidad aplicables.
- *Documentación*: hay una ayuda que realmente ayuda y que encaja con las funcionalidades.

Carisma. *¿El producto "lo" tiene?*

- *Unicidad*: el producto es distinguible y tiene algo que ningún otro tiene.
- *Satisfacción*: ¿cómo te sientes tras usar el producto?
- *Profesional*: ¿tiene el producto un aire adecuado de profesionalidad y de encaje con su cometido?
- *Atractivo*: los distintos aspectos del producto, ¿son atrayentes a la vista y demás sentidos?
- *Curiosidad*: ¿se interesarán los usuarios por el producto e intentarán averiguar qué pueden hacer con él?
- *Hechizo*: los usuarios, ¿se enganchan, disfrutan con fluidez y están totalmente abstraídos mientras usan el producto?
- *Moda*: ¿el producto debería usar las tecnologías/ideas más modernas y potentes?
- *Expectación*: el producto supera tus expectativas e incluso satisface necesidades que no sabías que tenías.
- *Actitud*: ¿tienen el producto y su contenido la actitud adecuada y se comunican contigo mediante un lenguaje y tono correctos?
- *Dirección*: las (primeras) impresiones del producto, ¿impresionan?
- *Historia*: ¿existen historias fascinantes sobre la concepción, construcción o uso del producto?

Seguridad. *¿El producto se protege contra su uso indebido?*

- *Autenticación*: identificación de los usuarios por parte del producto.
- *Autorización*: gestión por parte del producto de lo que un usuario autenticado puede ver y hacer.
- *Privacidad*: capacidad de no revelar datos protegidos a usuarios no autorizados.
- *Agujeros de seguridad*: el producto no debería invitar al descubrimiento de sus vulnerabilidades mediante el uso de ingeniería social.

- *Secreto*: el producto no debe revelar información sobre sus sistemas internos bajo ninguna circunstancia.
- *Invulnerabilidad*: capacidad de resistir ataques.
- *Sin virus*: el producto no debe contener virus, ni comportarse como tal.
- *Resistencia a la piratería*: no hay posibilidad de copiar ni distribuir ilegalmente el software o su código.
- *Conformidad*: adecuación a los estándares de seguridad a los que se adhiere el producto.

Rendimiento. *¿El producto es suficientemente rápido?*

- *Capacidad*: los múltiples límites del producto, en distintas circunstancias (por ejemplo, lentitud en la red).
- *Utilización de recursos*: uso adecuado de memoria, almacenamiento y otros recursos.
- *Capacidad de respuesta*: velocidad a la que una acción es (percibida como) realizada.
- *Disponibilidad*: el sistema está disponible para su uso cuando debe estarlo.
- *Caudal*: capacidad del producto de procesar grandes volúmenes de cosas.
- *Resistencia*: ¿el producto puede aguantar carga durante un periodo prolongado?
- *Retroalimentación*: ¿es apropiada la retroalimentación por parte del sistema ante las acciones del usuario?
- *Escalabilidad*: ¿el producto reacciona adecuadamente cuando es escalado horizontal o verticalmente, tanto a mayor como a menor capacidad?

Capacidad tecnológica. *¿Es fácil de instalar, mantener y dar asistencia al producto?*

- *Requerimientos del sistema*: capacidad de funcionar en las configuraciones soportadas, así como de gestionar entornos distintos y ausencia de componentes.
- *Instalación*: el producto es instalable en las plataformas esperadas usando la cantidad de recursos adecuada.
- *Actualizaciones*: facilidad de actualizar el producto a una versión más reciente sin perder la configuración ni los ajustes de usuario.
- *Desinstalación*: ¿se eliminan todos los ficheros y recursos del programa (salvo los del usuario o sistema) cuando éste se desinstala?
- *Configuración*: ¿la instalación puede configurarse de diversas maneras para ajustarse a las necesidades del usuario?
- *Despliegue*: el producto puede ser desplegado por parte del departamento de TI a distintos tipos de usuarios (restringidos) y entornos.
- *Mantenibilidad*: ¿es fácil de mantener y dar soporte a los usuarios sobre el producto y sus artefactos?
- *Capacidad de prueba*: ¿cuán eficientemente puede el cliente testear el producto desplegado?

Compatibilidad. *¿Cómo interacciona el producto con otros programas y entornos?*

- *Compatibilidad de hardware*: el producto puede usarse con las distintas configuraciones del hardware indicado.
- *Compatibilidad de sistema operativo*: el producto funciona en las versiones indicadas de los sistemas operativos compatibles, comportándose tal y como se espera en ellos.
- *Compatibilidad de aplicación*: el producto y sus datos funcionan conjuntamente con las demás aplicaciones que el usuario desea.
- *Compatibilidad de configuración*: capacidad del producto de integrarse con las configuraciones del entorno.
- *Compatibilidad regresiva*: ¿el producto puede hacer todo lo que hacía su versión anterior?
- *Compatibilidad a futuro*: ¿será el producto capaz de usar artefactos o interfaces de versiones futuras?
- *Sostenibilidad*: efectos en el medio ambiente, por ejemplo: eficiencia energética, desconexiones, modos de ahorro de energía, teletrabajo.
- *Conformidad con los estándares*: el producto es conforme a los estándares, regulaciones, leyes o códigos éticos aplicables.

Características internas de la calidad del software

Estas características no son experimentadas directamente por los usuarios finales, pero pueden ser también importantes para conseguir el éxito del producto.

Soporte. *¿Se puede dar soporte al uso real del producto y sus problemas asociados?*

- *Identificadores*: ¿es fácil identificar partes del producto, sus versiones o errores específicos?
- *Diagnóstico*: ¿es posible obtener detalles de los problemas específicos de los usuarios?
- *Resolución de problemas*: ¿es fácil identificar errores (por ejemplo, en los ficheros de log) y obtener ayuda?
- *Depuración*: ¿se pueden observar los estados internos del programa cuando es necesario?
- *Versatilidad*: capacidad de usar el producto con más fines de los previstos inicialmente.

Capacidad de prueba. *¿Es fácil probar el producto?*

- *Trazabilidad*: el producto guarda registro de las acciones acorde a los niveles de traza adecuados y en un formato usable.
- *Control*: capacidad de establecer estados, objetos o variables de forma independiente.
- *Observación*: capacidad de observar cosas que deben ser probadas.
- *Monitorización*: ¿el producto puede dar pistas de qué está haciendo y cómo?
- *Aislamiento*: capacidad de probar una parte del producto por sí misma.
- *Estabilidad*: los cambios en el software están controlados y no son muy frecuentes.
- *Automatización*: ¿existen interfaces programáticas, ya sean públicas u ocultas, que puedan ser usadas?

- *Información*: el producto proporciona medios para que los testers puedan aprender del mismo lo que necesiten para realizar su tarea.
- *Auditoría*: ¿pueden el producto y sus derivados ser validados?

Mantenimiento. *¿Se puede mantener y ampliar el producto a bajo precio?*

- *Flexibilidad*: capacidad de cambiar el producto bajo petición.
- *Extensibilidad*: ¿será fácil añadir nuevas funcionalidades en el futuro?
- *Simplicidad*: el código no es más complejo de lo necesario y no dificulta el diseño, ejecución o evaluación de las pruebas.
- *Legibilidad*: el código está documentado adecuadamente y es fácil de leer y entender.
- *Transparencia*: ¿es fácil entender las estructuras subyacentes del producto?
- *Modularidad*: el código está separado en partes manejables.
- *Refactorizable*: ¿estás satisfecho con las pruebas unitarias del producto?
- *Análisis*: capacidad de encontrar las causas de los defectos u otras partes del código que sean de interés.

Portabilidad. *¿Es posible transferir el producto a otros entornos o lenguajes?*

- *Reusabilidad*: ¿se pueden reusar partes del producto en otros lugares?
- *Adaptabilidad*: ¿es fácil modificar el producto para que soporte un entorno distinto?
- *Compatibilidad*: ¿el producto cumple con interfaces comunes o estándares oficiales?
- *Internacionalización*: es fácil traducir el producto.
- *Localización*: ¿están las distintas partes del producto preparadas para satisfacer las necesidades de los países o culturas objetivo?
- *Robustez de la interfaz de usuario*: ¿el producto tendrá el mismo buen aspecto una vez traducido?



Rikard Edgren, Henrik Emilsson y Martin Jansson - thetesteye.com v1.1

*Este trabajo se encuentra bajo licencia Creative Commons Attribution-No Derivative
inspirado por CRUSSPIC STMPL (James Bach), ISO 9126-1, Wikipedia:Ilities y más...*

Traducido al castellano por Núria Cardona, Mauri Edo y Marcel Puchol