

Programación Orientada a Objetos
Calidad del Software y Factores de Calidad
(apuntes del profesor)
Juan Antonio Romero del Castillo (aromero@uco.es)
Departamento de Informática y Análisis Numérico
Universidad de Córdoba

7 de noviembre de 2024

Índice general

1. Software de Calidad	1
1.1. Introducción	1
1.1.1. Operación del producto	2
1.1.2. Transición del producto	2
1.1.3. Revisión del producto	3
1.2. Otros factores importantes	4
1.2.1. Seguridad	4
1.2.2. Accesibilidad	4
1.2.3. Oportunidad	4
1.2.4. Economía	4
1.2.5. El compromiso entre factores de calidad	4
Bibliografía	5

Capítulo 1

Software de Calidad

1.1. Introducción

El software de calidad es el principal objetivo la programación orientada a objetos (POO) y de la abstracción. Pero, ¿qué es software de calidad?. ¿Se puede medir la calidad en el software?.

En un primer enfoque podemos clasificar las cualidades de calidad del software desde dos puntos de vista distintos pero relacionados: calidad funcional y calidad estructural.

La calidad funcional tiene que ver con el grado de acercamiento del funcionamiento del software a lo que se determinó en sus especificaciones y requisitos funcionales. Se trata de una característica que puede apreciarse durante la ejecución del software por parte del usuario, por ello se denominan también factor de calidad externo. Afecta de forma directa al usuario del software.

La calidad estructural tiene que ver con el cumplimiento de características internas (por eso también se llama factor de calidad interno) no-funcionales del software que dan soporte características internas del software también muy deseables como son la mantenibilidad, la facilidad de extensión, la reutilización, etc. Afecta de forma más directa al programador del software (y de forma indirecta usuario final).

Hay que resaltar que ambos tipos dependen del desarrollador y que, al final, afectan directa o indirectamente al usuario final.

En cualquier caso el concepto de calidad del software no es fácil de definir. Existen incluso planteamientos que defienden que es imposible determinar la calidad de un software. A pesar de ello sí existen factores de calidad del software que se asumen en general como deseables y que son interesantes que, se use el modelo de desarrollo software que se use, estén presentes en el software resultante.

Los **factores de calidad** que nosotros estudiaremos están basados en el trabajo de McCall y colaboradores de 1977, el denominado modelo de McCall basado en los denominados factores de calidad. Es el modelo más extendido y base de muchos otros. Dicho modelo se basa en tres ejes: Operación, revisión y transición.

1.1.1. Operación del producto

Product operation. Características funcionales y operacionales.

Corrección

Correctness. Es un factor externo. Es la capacidad del producto para realizar de forma adecuada aquello para lo que fue creado, tal como se definió en los documentos de especificación y requerimientos.

Robustez

Reliability. Es un factor externo. Es la capacidad del producto de comportarse con exactitud, de manera satisfactoria y precisa en todas las situaciones. Por tanto, manejar correctamente situaciones imprevistas, de error, o fuera de lo normal. Esta característica es imprescindible en infraestructuras críticas y sistemas de gran responsabilidad, sistemas en tiempo real, etc.

Eficiencia

Efficiency. Es un factor externo e interno. Se trata de conseguir que el programa, además de realizar correctamente aquello para lo que ha sido creado, lo realice además de la mejor forma posible. Algunos factores que afectan a la eficiencia son: el espacio en memoria utilizado por el programa y el tiempo de ejecución (los dos más importantes en la mayoría de los casos), el espacio en disco, la memoria temporal, número de accesos a determinado hardware (de comunicaciones por ejemplo), etc. Importante en sistemas en tiempo real.

Integridad

Integrity. Es un factor externo. El producto no debe corromperse ni comprometerse ni ser vulnerable en ninguna de sus partes ni siquiera por el hecho de su utilización masiva o por una gran acumulación de datos. Un sistema que no es integro puede ver comprometida su seguridad con facilidad.

Facilidad de Uso

Usability. Es un factor externo. Facilidad al introducir datos, interpretar datos, comprender errores, interpretar resultados, etc. Para usuarios con diferentes formaciones y aptitudes. Instalación, operación y supervisión. Capacidad multilingue. A veces es clave para que el software tenga éxito.

1.1.2. Transición del producto

Product transition. Capacidad de adaptación a nuevos entornos.

Portabilidad

Portability. Es un factor externo e interno. Es la capacidad o facilidad del producto de ejecutarse en otro hardware diferente o en otro sistema operativo diferente. Aquí es muy importante que el programa no haga uso de características de bajo nivel del hardware o que aisle la parte dependiente del hardware para que al portarlo solo sea necesario modificar dicha parte.

Reutilidad/Reusabilidad

Reusability. Es un factor interno. Es la capacidad del producto de ser reutilizado en su totalidad o en parte por otros productos, con el objetivo de ahorrar tiempo en soluciones redundantes ya hechas con anterioridad. De esta forma, un programa debe agrupar en módulos aislados los aspectos dependientes de la aplicación particular, mientras que las utilidades que puedan generalizarse deben mantenerse preparadas para ser utilizadas por otros productos y así ahorrar costes de producción. Diseño por capas de diferente nivel de abstracción.

Compatibilidad

Interoperability. Es un factor externo. Es la facilidad que tienen los programas de combinarse entre sí. Es la posibilidad de utilizar los datos de salida de un programa como entrada de otro programa. La clave es la estandarización y el consenso entre organismos, instituciones y empresas, en primera instancia, y entre programadores en última instancia.

1.1.3. Revisión del producto

Product revision. Aptitudes ante la modificación y el mantenimiento del producto.

Mantenibilidad

Maintainability. Es un factor interno. Capacidad de encontrar y corregir un defecto en el software.

Extensibilidad

Flexibility. Es un factor interno. Es la facilidad de adaptar el producto a cambios en la especificación de requisitos. Cuanto mayor es el programa suele costar más esfuerzo, y si no se aplica un buen diseño, y sobre todo simple, este factor puede verse muy perjudicado. El software siempre está evolucionando luego este factor es importante siempre.

Testable

Testability. Es un factor interno. Habilidad y de poder validarse el software. Facilitar el testing es fundamental en el desarrollo de cualquier software.

1.2. Otros factores importantes

Son factores que avanzan en importancia tanto como para ser considerados por separado.

1.2.1. Seguridad

Es un factor externo. Es la capacidad del producto de proteger sus componentes de usos no autorizados y de situaciones excepcionales de pérdida de información. Para ello debe prever mecanismos de control de acceso, encriptación, autenticación, diferentes roles, etc. Y también, desde el otro punto de vista copias de seguridad, procesos rutinarios de comprobación de integridad, etc.

1.2.2. Accesibilidad

Es un factor externo.

Accesibilidad General: Tener acceso, paso o entrada a un lugar o actividad sin limitación alguna por razón de deficiencia, discapacidad, o minusvalía.

Accesibilidad Informática: consiste en el acceso a la información sin limitación alguna por razones de deficiencia, incapacidad o minusvalía. Por tanto, un programa accesible es aquél que permita el acceso a la información sin limitación alguna por razón de deficiencia, incapacidad o minusvalía con el fin de poder elaborar, reproducir, manipular, etc. dicha información, así como al acceso sin ningún tipo de limitación por dichas causas a las herramientas y opciones de dicho programa.

1.2.3. Oportunidad

Capacidad de un sistema de ser lanzado cuando el usuario/mercado lo necesita, o antes. Ej: ideas “tontas” y “simples” han hecho millonario a muchos...

1.2.4. Economía

Los costes del producto. A veces, lo más importante de todo.

1.2.5. El compromiso entre factores de calidad

El compromiso entre ellos es necesario en muchas ocasiones y muchas veces no se puede cumplir alguno de los factores al cien por cien. Nuestra experiencia nos hace alcanzar ese compromiso para cumplir nuestras expectativas.

Nuestra tarea mientras nos formamos en programación es aprender métodos y herramientas que hagan que el software que desarrollamos tenga una alta calidad

y por tanto debemos aprender como mejorar cada uno de los factores de calidad anteriores cuando programamos. Esa será una de las tareas fundamentales durante este curso.