



DESARROLLO DE SOFTWARE

Fundamentos de la calidad del software

- ▶ ASIGNATURA | Calidad de Software
- ▶ PROFESOR | Ing. Santiago Solís
- ▶ APELLIDOS Y NOMBRE | Guamaní Pérez Erick Fabricio

- ▶ EMAIL | _____ TELÉFONO | 0988526262
- ▶ TRABAJO GRUPAL N° | _____ FECHA DE ENTREGA | 20/07/2020

Contenido

1.	Fundamentos de calidad de software	3
1.1	Definición de calidad	3
1.2	Medición de los costos de mala calidad	3
1.3	Evolución del enfoque de calidad	4
1.4	Principios de la gestión de calidad total	4
1.5	Autores de calidad	4
2.	Enfoques de calidad	5
2.1	Funcionalidad	6
2.2	Confiabilidad	6
2.3	Usabilidad	7

1. Fundamentos de calidad de software

Según la ISO 2000 se la define como un grado en el conjunto de características inherentes cumple con los requisitos. En esta se hace especialmente énfasis en los requerimientos del cliente.

1.1 Definición de calidad

La calidad de software se define con el grado en que posee una combinación adecuada de determinadas características como son el rendimiento, la fiabilidad, y seguridad.

Esta se mide a través de su rendimiento, calidad, seguridad, las posibilidades de mantenimiento y de agregar actualizaciones. (Practicas ADSI, s.f.)

1.2 Medición de los costos de mala calidad

Los costes de los fallos, tanto internos como externos, son especialmente altos en el software que tiene una deuda técnica alta provocada por una baja mantenibilidad, baja fiabilidad y una inadecuada arquitectura de componentes. (Gandarillas, 2017)

Los conceptos de los costes aplicados al desarrollo de software son:

- Costes de las pruebas: Costes derivados de probar el software para evitar que los defectos lleguen a producción.
- Prevención: Costes derivados de mejorar los procesos de desarrollo como herramientas técnicas, arquitectura, métodos, practicas, etc. y de la formación de personas para minimizar la inyección de defectos en el software.
- Costes del fallo interno: Costes de búsqueda y corrección del defecto, así como la repetición de pruebas derivadas de a ver encontrado un defecto antes de liberar el software. No solo haya que tener en cuenta el coste de repetir la prueba que origino el fallo (retest) si no la posible regresión de pruebas y los costes asociados a un nuevo paso a producción que suelen ser la mayor cuantía.
- Costes del fallo en producción: En este escenario no solo hay que sumar los costes equivalentes al fallo interno, si no el daño producido al cliente, usuario o al propio negocio donde se utiliza el software. La

pérdida de confianza y las posibles penalizaciones o sanciones pueden ser los mayores cotes de este tipo de fallos.

1.3 Evolución del enfoque de calidad

Todo partió desde la revolución industrial en el punto de vista productivo, pero se forjó durante la primera guerra mundial, donde surgen los primeros inspectores de calidad, esto condujo a la creación de áreas organizativas de inspección separadas de la producción.

Pero no fue hasta la segunda guerra mundial donde nace la necesidad de un control estadístico de calidad por la enorme producción en masa donde el enfoque de control de calidad de un producto final a un control en todo el proceso de calidad del producto.

Después un análisis del control de calidad se enfoca más en producir productos de buena calidad y dejar de lado las producciones masivas dando más importancia a la calidad del producto en sí.

Y el último y actual enfoque está dirigido al cliente o usuario donde satisfacer las necesidades de este es lo más importante por medio de los controles de calidad ofrecer al cliente un producto que satisfaga las necesidades de este. (Romero, 2007)

1.4 Principios de la gestión de calidad total

Un servicio de calidad total es un enfoque organizacional global, que hace de la calidad de los servicios, según percibe el cliente, según su perceptibilidad y su calidad factual, donde la primera es la clave para que la gente compre y la segunda es de lograr la lealtad del cliente hacia el producto por medio de la marca u organización.

1.5 Autores de calidad

Los máximos autores que implantaron la calidad en el mundo llamados los cinco grandes que son:

- William Edwards Deming.
- Joseph M Juran.
- Armand V. Feigenbaum.
- Kaoru Ishikawa.

- Philip B. Crosby.

Otros han surgido después y son reconocidos a nivel mundial, pero los aportes de estas cinco personas fueron los que más impacto ocasionaron en el mundo.

Deming que con su idea de separa las cosas buenas y malas, además de integrar los métodos estadísticos para el control de calidad en Japón ayudo e impulso a este país a salir a delante a raíz de la postguerra.

Juran oriento el control estadístico de calidad a la necesidad de que se convirtiera en un instrumento de alta dirección. Se enfocaba en que todos los participantes deben tener en cuenta la calidad del producto durante todo el desarrollo del mismo.

Feigenbaum fue el fundador del control de calidad total (CTC) el cual se enfocaba en el desarrollo y mantenimiento de un producto de calidad, dando la pauta a la creación de departamentos gerenciales que este enfocados únicamente en el control de la calidad de los productos o servicios que ofrezca una empresa, siendo la calidad una tarea de todos en una organización.

Ishikawa tomo el concepto de control de calidad total y lo adopto al estilo japones, llamándolo “control de calidad en todo a empresa” haciendo referencia a que todos los trabajadores de una empresa deben estudiar, participar y practicar el control de calidad.

Crosby desarrolla toda una teoría basado fundamentalmente en lo que cuesta dinero son las cosas que no tienen calidad, partiendo de las acciones mal ejecutadas y la toma decisiones a raíz de esas malas acciones.

2. Enfoques de calidad

1920	Inspección de la calidad	Separación de las unidades buenas de las malas
1950	Control de la calidad	Detección y prevención de los defectos en el proceso de fabricación
1970	Aseguramiento de la calidad	Incorporación del control de calidad en todas las actividades de la organización
1980	Gestión de la calidad	Integrar los esfuerzos de todos hacia el logro de la calidad
1990	Gestión total de la calidad	Extensión del logro de la calidad a todas las actividades que realiza la organización

2.1 Funcionalidad

Se refiere a la capacidad del producto de software para suministrar un conjunto de funciones que satisfagan las necesidades implícitas o explícitas de los usuarios, al ser utilizado bajo condiciones específicas. Se relaciona directamente con aquello que el software hace para satisfacer necesidades, mientras que las demás características se refieren al cómo y al cuándo. Evalúa el cumplimiento de requerimientos, la exactitud de los resultados, la seguridad del producto y la interacción con otros sistemas.

Se clasifica en cinco subcategorías que abarcan los aspectos relevantes en relación con la funcionalidad de un producto software. (ISO 25 000, s.f.)

Apropiabilidad: Es la capacidad del software para suministrar un conjunto apropiado de funciones que cumplan tareas específicas y objetivos del usuario.

Exactitud: Capacidad del software para proveer los resultados correctos y los efectos pactados, con un adecuado grado de precisión.

Interoperabilidad: Capacidad del software para interactuar con uno o más sistemas específicos. Esto depende, en gran parte, de la necesidad de interacción con los demás sistemas.

Seguridad: Hace referencia a la capacidad del software para proteger los datos y la información, con el fin de que personas no autorizadas nunca puedan ingresar al sistema, leer o modificar los datos.

Conformidad en la funcionalidad: Capacidad del software para ajustarse a los estándares, convenciones y regulaciones relacionadas con la correcta implementación de especificaciones durante todo el proceso de análisis y construcción.

2.2 Confiabilidad

Esta característica principal se refiere a la capacidad del producto de software para conservar su nivel de desempeño bajo condiciones específicas durante un determinado período de tiempo.

La confiabilidad de un producto hace relación a la facultad del mismo para garantizar un buen manejo de errores en cualquier situación, lo cual implica varios aspectos, por ejemplo, que desde el diseño del software se hayan considerado todos los posibles errores funcionales y se les haya asignado una salida óptima (por ejemplo, mensajes de error que indiquen al usuario comportamientos anormales, entre otros), así como también, que se

haya pensado en aquellas situaciones de error que son externas al software, de tal forma que el producto pueda responder positivamente ante posibles fallas ocasionales, continuar su ejecución mediante planes de contingencia y recuperar las acciones inconsistentes que puedan afectar la integridad de los datos.

2.3 Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes sub características:

- **Capacidad para reconocer su adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Bibliografía

- Gandarillas, A. (13 de 06 de 2017). *Blog de Aurelio Gandarillas*. Obtenido de <https://aureliogandarillas.es/blog/2017/06/costes-calidad-software/#:~:text=Costes%20de%20detecci%C3%B3n%3A%20Costes%20de,antes%20de%20entregar%20el%20producto.>
- ISO 25 000. (s.f.). *ISO 25 000 Calidad de software y datos*. Obtenido de <https://iso25000.com/>
- Practicas ADSI. (s.f.). *Practicas ADSI*. Obtenido de <https://albertdel.wordpress.com/actividades/fundamentos-de-calidad-de-software/#:~:text=Control%20de%20calidad.&text=Un%20concepto%20clave%20del%20control,para%20minimizar%20los%20efectos%20producidos.>
- Romero, A. L. (10 de 08 de 2007). *gestiopolis*. Obtenido de <https://www.gestiopolis.com/evolucion-de-la-calidad-iso-9000-y-otros-conceptos-de-calidad/#cuadros-resumen>