

Testing de Calidad de Software

1. Ética Profesional
2. Calidad de Software
3. ¿Qué tareas cumple un gestor de calidad?
4. Error – Defecto – Falla
5. QA vs. QC

El estándar [ISTQB](#) (*International Software Testing Qualifications Board*) tiene en cuenta estas cuestiones y establece un código deontológico para el tester, que resumo en los siguientes puntos:

- 1 **Actuar coherentemente, no solo con los intereses de la empresa, sino del cliente y de la sociedad.** Esto hace que si se detecta un bug realmente crítico para el cliente, pudiendo afectar a terceros, no se debe ceder a ninguna presión para dejar de informar sobre el mismo.
- 2 **Asegurar que los productos probados cumplen con los estándares más altos,** dentro de las posibilidades. Aquí entraríamos en la falta moral de no probar correctamente un producto deficiente a sabiendas que se necesitará contratar un mantenimiento correctivo para tratar de tapar los múltiples agujeros.
- 3 **Mantener la integridad e independencia en el juicio profesional.**
- 4 **Promover un método ético para la gestión de las pruebas,** no entrando en discriminaciones, presión injustificada a los compañeros, ...
- 5 **Promover la integridad y la reputación de la profesión,** de nuevo, en conformidad con el interés público.
- 6 **Ser justos con los compañeros,** ser una ayuda para ellos y promover la cooperación con los desarrolladores. Aquí entra la inteligencia emocional, el trasladar los bugs de forma “constructiva”, aportando a la calidad del producto y no caer en la crítica “destructiva”, que por otra parte no suele servir para conseguir los objetivos marcados sino que crea más problemas.
- 7 **Participar en el aprendizaje permanente en relación con la práctica de la profesión.** Asistir a cursos, charlas, leer libros, dar conferencias, escribir un blog, participar en las redes sociales y en foros sobre la profesión... todo lo que te ayude a avanzar y formarte un (buen) criterio. Y yo diría más, no solo hay que saber sobre temas técnicos, sino tratar de cultivar capacidades como el trabajo en equipo, el liderazgo, técnicas de negociación... y cultivar la empatía y la inteligencia emocional para ser el mejor aliado en tu proyecto y no un problema más.

Calidad en el proceso del desarrollo del software.

El software se ha convertido en la actualidad en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada día, los procesos mas importantes de las organizaciones y su supervivencia dependen del funcionamiento del software.

Según Pressman (2005), es la concordancia del software producido con los requerimientos explícitamente establecidos y con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario. Otra definición que contempla Vega, Rivera & García (2008) en su libro. Y que es propuesta por la organización internacional de estándares (ISO/IEC DEC 9126): “La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas”.

La calidad del software se puede observar en una característica o atributo. Como un atributo, la calidad se refiere a características mensurables, es decir cosas que se pueden comparar para conocer estándares, como longitud, color, propiedades eléctricas y maleabilidad. Sin embargo, el software que es una entidad intelectual, tiene la complejidad de caracterizar los objetos físicos. No obstante, existen mediciones que nos permiten evaluar las características de un programa. Dichas propiedades incluyen complejidad psicosomática, número de puntos de función, líneas de código, etcétera. Cuando se examina un elemento sus características mensurables se pueden encontrar dos tipos de calidad:

Calidad de diseño; la calidad de diseño se refiere a las características que los diseñadores especifican para un elemento.

Calidad de concordancia; la calidad de concordancia es el grado en el que las especificaciones de diseño se aplican durante la fabricación.

En el desarrollo de software la calidad del diseño incluye requisitos, especificaciones y el diseño del sistema. La calidad de concordancia es un tema enfocado principalmente a la implementación. Si el diseño y el sistema resultante satisfacen los requisitos y metas de desempeño, la calidad de concordancia es alta. Glass (1998), argumenta que es conveniente generar una relación más intuitiva.

Satisfacción del usuario = producto manejable + buena calidad + entrega dentro del presupuesto y tiempo.

Glass (1998), afirma que la calidad es importante, pero si el usuario no está satisfecho, nada más importa en realidad. De igual forma afirma que la calidad de un producto es una función de cuánto cambia el mundo para mejorar. Esta visión de la calidad afirma que si un software proporciona beneficio sustancial a sus usuarios finales, éstos están dispuestos a tolerar problemas ocasionales en aspectos como la confiabilidad y el desempeño.

Control de calidad

El control de la variación puede equipararse con el control de calidad. Esto involucra la serie de inspecciones, revisiones y pruebas empleadas a lo largo del proceso del software para garantizar que cada producto del trabajo satisfaga los requisitos que se le han asignado. El control de calidad incluye un bucle de retroalimentación con el proceso que creó el producto del trabajo. La combinación de mediciones retroalimentación permite afinar el proceso cuando los productos de trabajo creados fracasan en cuanto a satisfacer sus especificaciones. Un concepto clave del control de calidad es que todos los productos de trabajo tienen especificaciones definidas mensurables con las cuales se puede comparar la salida de cada proceso. Dicho bucle es esencial para minimizar los efectos producidos.

Garantía de la calidad

La garantía de la calidad consiste en un conjunto de funciones de auditoría e información que evalúan la efectividad y qué tan completa son las actividades de control de calidad. La meta del aseguramiento de la calidad es presentarle al gestor los datos necesarios para que esté informado acerca de la calidad del producto y por consiguiente que comprenda y confíe en que la calidad del producto está satisfaciendo las metas y objetivos. Si se identifican problemas en el proceso de aseguramiento de calidad, es responsabilidad del gestor aportarlos y aplicar los recursos necesarios para resolver los conflictos de calidad.

Costo de la calidad

El costo de la calidad incluye todos los costos que se generan o que demandan el desarrollo de las actividades relacionadas con la calidad. Los estudios de costo de la calidad se llevan a cabo para ofrecer una línea base e identificar oportunidades que reduzcan el costo de calidad y proporcionan una base que sirva de comparación. La base de normalización casi siempre es monetaria, ya que se tienen los datos necesarios para evaluar dónde se encuentran las oportunidades para mejorar los procesos, se puede evaluar el efecto de los cambios en términos monetarios. Los costos de calidad se dividen en:

- 1) Costos asociados con prevención; estos costos incluyen la planificación de la calidad, revisiones técnicas formales, equipo de pruebas y entrenamiento.
- 2) Evaluación y fallas; estos costos incluyen actividades que permiten comprender mejor la condición del producto a través de cada proceso. Algunos ejemplos de costos de valuación incluye ni inspección en el proceso y procesos, calibración y mantenimiento de equipo además de las pruebas correspondientes. Los costos de fallas son aquellos que desaparecerán si no hubiese defectos antes de enviar el producto a los clientes. Estos costos se subdividen en costos de fallas internas y externas.

Se incurren los costos de fallas internas cuando se detecta un defecto en el producto antes del envío, dichos costos incluyen reelaboración, reparación y análisis el modo de falla. Los costos de fallas externas se asocian con defectos detectados después de que el producto ha sido enviado al cliente algunos ejemplos de estos son la resolución de las quejas, devolución y reemplazo del producto, soporte de ayuda en línea y trabajo de garantía.

Garantía de la calidad del software

El control y la garantía de la calidad son actividades esenciales en cualquier negocio que elabora productos de consumo. En la actualidad, toda compañía tiene mecanismos que garantizan la calidad en sus productos. De hecho, las afirmaciones explícitas de la preocupación de una compañía por la calidad se ha convertido en una práctica de mercadotecnia durante las décadas pasadas. La historia de la garantía de la calidad en el desarrollo de software avanza paralelamente a la de la calidad en la fabricación del hardware. Durante los primeros días de la computación (décadas de 1950 y 1960), la calidad era responsabilidad exclusiva del programador. Los estándares de garantía de la calidad para el software se introdujeron en los contratos militares durante el decenio de 1970 y se han extendido rápidamente en el desarrollo del software en el mundo de los negocios.

Si se extiende la definición de garantía de la calidad del software podemos decir que es un patrón de acciones sistemático y planificado que se requiere para garantizar la alta calidad en el software. Numerosos y diversos participantes tienen responsabilidad en la garantía de la calidad del software

como ingenieros de software, gestores del proyecto, clientes, vendedores y los individuos que participan en el grupo de desarrollo. Las revisiones del software son un filtro para el proceso de software. Esto es, que las revisiones se aplican en varios puntos durante la ingeniería del software y sirven para descubrir errores y defectos que pueden eliminarse. Las revisiones del software purifican las actividades que se han denominado análisis, diseño y codificación. Freedman & Weinberg (1990), abordan el modo siguiente y la necesidad de las revisiones:

" el trabajo técnico necesita revisarse por la misma razón que los lápices necesitan gomas, errar es de humanos. La segunda razón por la que se necesitan las revisiones técnicas es que aunque la gente sea buena al captar algunos de sus propios errores, las grandes clases de errores escapan de su creador con más facilidad de lo que se le escapan a alguien más".

El objetivo principal de las revisiones técnicas formales es descubrir los errores durante el proceso, de modo que no se conviertan en defectos después de liberar el software. El beneficio de las revisiones técnicas formales es el descubrimiento temprano de los errores de modo que ya no se propaguen a la etapa siguiente en el proceso de desarrollo de software.

De acuerdo a los estudios realizados por Vega et al (2008), algunas normativas de calidad en los sistemas de información y que ayudan a la realización, además de aplicar mejores prácticas en las organizaciones son:

ISO 9000, gestión y aseguramiento de calidad (conceptos y directrices generales).

Recomendaciones externas para aseguramiento de la calidad (ISO 9001, ISO 9002, ISO 9003).

Recomendaciones externas internas para aseguramiento de la calidad (ISO 9004).

Malcom Baldrige national quality award.

Software Engineering Institute (SEI).

Capability Maturity Model (CMM).

Six Sigma.

Los sistemas ISO de garantía de calidad fueron creados para ayudar a las organizaciones a garantizar que sus productos y servicios satisfacen las expectativas de los clientes al cumplir las especificaciones. El estándar ISO 9000 describe un sistema que garantiza la calidad en términos genéricos y que se puede aplicar a cualquier negocio sin importar los productos o servicios ofrecidos. ISO 9000 requiere que los sistemas de operaciones de calidad y una compañía se sometan a revisión de auditores de una tercera entidad, el cual tiene conocimiento del estándar y de su funcionamiento. Antes del registro exitoso, los auditores extienden a la compañía un certificado de la organización que representan. Entrevistas de auditoría semianuales garantizan la concordancia continua con el estándar.

El estándar de garantía de la calidad que se aplica en la ingeniería del software es el ISO 9001: 2000. Este estándar contiene 20 requisitos que deben estar presentes para generar un sistema eficiente de garantía de la calidad. Puesto que el estándar 9001: 2000 es aplicable a todas las disciplinas de ingeniería, se ha desarrollado un conjunto especial de directrices que permiten interpretar el estándar para emplearlo en el proceso de software. Los requisitos que especifica el estándar abordan tópicos como responsabilidad de la gestión, sistema de calidad, revisión de contrato, control de diseño, control de documentos y datos, identificación y seguimiento del producto, control de proceso, inspección y pruebas, acciones correctivas y preventivas, control de registros de calidad, auditorías de calidad interna, entrenamiento, servicio y técnicas estadísticas.

Una organización de software obtendrá el registro ISO 9001:2000 si establece políticas y procedimientos para abordar cada uno de los requisitos anotados además, ser capaz de demostrar que se siguen dichas políticas y procedimientos.

Entre las políticas y procedimientos que se deben de demostrar en una auditoría están las siguientes:

a) Establecer los elementos de un sistema de gestión de calidad

Desarrollar, implementar y mejorar el sistema.

Definir una política enfatice la importancia del sistema.

b) Documentar el sistema de calidad

Describir el proceso.

Producir un manual operativo.

Desarrollar métodos para controlar los documentos.

Establecer métodos para la conservación de registros.

c) Soporte del control y la garantía de calidad

Promover la importancia de la calidad entre todos los participantes.

Enfocarse en la satisfacción del cliente.

Definir un plan de calidad que aborde objetivos, responsabilidades de autoridad.

Definir mecanismos de comunicación entre los participantes.

d) Establecer mecanismos de revisión para el sistema de gestión de calidad

Identificar métodos de revisión y mecanismos de retroalimentación.

Definir procedimientos de seguimiento.

Identificar recursos de calidad que incluyan personal, entrenamiento, elementos de infraestructura.

e) Establecer mecanismos de control

Para planeación.

Para requisitos del cliente.

Para actividades técnicas, por ejemplo análisis diseño y pruebas.

Para supervisión y gestión del proyecto.

f) Definir métodos para corrección

Valorar los datos y métricas de calidad.

Definir enfoques para procesos continuos y de mejora de la calidad.

Según Dunn y Ullman(1982), "el aseguramiento de la calidad del software es el mapeo de los preceptos gerenciales y las disciplinas de diseño de la garantía de calidad en el espacio gerencial y tecnológico aplicable del ingeniería del software".

La habilidad para garantizar la calidad es la medida de una disciplina de ingeniería madura. Cuando el mapeo se logra de manera exitosa el resultado es la aplicación de la ingeniería de software en un nivel de madurez.

Gestor de Calidad.

Algunas de las actividades que debe ejercer un Gestor de Calidad para contribuir a la mejora de su empresa :

- 1 *El Gestor de Calidad debe **reportar directamente al “big boss” de la empresa** (en una PYME se reportaría al gerente, por ejemplo).*
- 2 *El Gestor de Calidad debe **participar en la planificación de las diferentes área de negocio**, y por tanto, la calidad debe estar presente y debe ser un factor directo a considerar en estas planificaciones.*
- 3 *El Gestor de Calidad debe **participar en el diseño de los productos o servicios**, considerando el impacto de los cambios del diseño en la calidad.*
- 4 *El Gestor de Calidad debe **inspeccionar y probar el producto o servicio** a lo largo de las etapas de su producción.*
- 5 *El Gestor de Calidad deben tener voz y voto en la **selección y evaluación del desempeño de los proveedores**, hasta el punto de realizar encuestas, visitas y auditorías a proveedores.*
- 6 *El Gestor de Calidad debe llevar a cabo el **control estadístico de los procesos y su calibración**, además de los **planes de inspección y control** (y **seguimiento de acciones correctivas**).*

Error – Defecto – Falla. Concepto.

A modo de aclarar conceptos debemos definir cada tipo de mal funcionamiento que puede ser encontrado en un software, los casos habituales son:

1. Error: Lo comete un humano durante el Proceso de Desarrollo.
2. Defecto: Es la consecuencia de un error.
3. Falla: Es un defecto no detectado y que no está eliminado.
4. Fallo: Defecto que se manifiesta durante la ejecución.
5. Incidente: Comportamiento no esperado.

Evidentemente la mejor forma de que no ocurran fallos es evitar escribir con errores que produzcan defectos. Para aumentar la calidad de un producto de Ingeniería de Software debemos realizar un control estático; que en palabras simples no es que otra cosa que otro un tester revise el sistema y retroalimente al equipo de trabajo del área de desarrollo.

Está comprobado que la calidad del Producto software aumenta inmediatamente si los desarrolladores saben que serán revisados por otras personas, en el llamado Control estático.

Muchas personas relacionan la calidad del Producto con que pase las pruebas - Control dinámico - antes de entregarlo, lo cual es un error de interpretación; el control estático es preventivo y es más eficiente pues obliga inmediatamente a implementar buenas prácticas.

Tomando como ejemplo un software “Calculadora”, podríamos detallar que:

- Error, se refiere a la diferencia entre la salida actual de un software y la correcta. También puede ser una acción que introduce en el software un defecto o falla. Por ejemplo: en la calculadora digitamos $2+2$ y nos arroja como resultado un 5.

- Defecto, es una condición que causa que un software falle al realizar una función requerida (sinónimo de bug) Por ejemplo: en la calculadora digitamos $2+2$ y nos arroja como resultado un mensaje de error.

- Falla, es la incapacidad de un software para realizar una función requerida de acuerdo con sus especificaciones (son producidas por defectos) Por ejemplo: la calculadora no tiene previsto sumar.

Comparativa entre control y garantía de calidad.

- ⑩ **Control de calidad** – Don las técnicas y actividades operacionales que se utilizan para cumplirlos requisitos de calidad.
- ⑩ **Garantía o Seguro de calidad** – Son todas las actividades planificadas y sistemáticas implementadas para proporcionar la confianza adecuada de que una entidad cumplirá los requisitos de calidad.

QC vs QA

Control de calidad
Se encarga de:

Producto
Reactivo
Encontrar Defectos
Testing
Inspección de Control
Revisión de Puntos

Garantía de calidad
Se encarga de:

Procesos
Proactivo
Prevenir Defectos
Auditoría de Calidad
Definición de Procesos
Selección de Herramientas
Entrenamiento

QA (Quality Assurance)	vs	QC (Quality Control)
Se diseñan y definen todos los parámetros de aceptación de un paquete de Software		Se controla el comportamiento del producto final
Es un sistema de PREVENCIÓN de fallos que predice casi todo sobre la seguridad, funcionamiento, normas de calidad y legalidad de un paquete de SOFTWARE y genera medidas correctivas para controlar y evitar que los productos o servicios defectuosos lleguen a la fase de producción		Es un sistema de CORRECCIÓN de fallos e introducción de mejoras
El departamento QA trabaja junto a desarrollo, ingenieros, managers y el cliente		El departamento de QC, trabaja junto a QA
El departamento QA está presente desde la fase del diseño del producto		El departamento de QC entra en acción cuando el producto está finalizado.
El QA está orientado al proceso		QC está orientado al producto
QA asegura que todos los desarrolladores siguen el mismo estándar de calidad, dentro de una gran corporación		QC asegura que el funcionamiento del producto, es el esperado
QA se diseña y ejecuta antes de tener un producto finalizado		QC se ejecuta durante la puesta en pre-producción
Se ejecutan las pruebas funcionales, unitarias, de integración y las pruebas de regresión, en otras palabras, se efectúa todo lo que es el "WHITE BOX TESTING" (pruebas a nivel de código fuente)		Se ejecutan las pruebas funcionales, pruebas de estrés, de rendimiento y de seguridad, etc., en otras palabras, se efectúa todo lo que es el "BLACK BOX TESTING" (pruebas a nivel de servicio)

Formas de desarrollar planes de testing

Sobre como hacer un proyecto ... desarrollo Iterativo y desarrollo Incremental.

Había una vez un ferretero llamado Carlos, quien compartía su negocio a partes iguales con su hermano Roberto. Ellos habían heredado la ferretería por parte de sus padres y un día decidieron compartir la compra de un terreno para edificar cada uno su casa y así ahorrar el pago del alquiler hogareño.

Carlos, construyó una casa pequeña, de un sólo ambiente y con pocas funcionalidades, aunque contaba con lo que necesitaba (por lo menos en esos días) para que su vida fuera cómoda y sin sobresaltos.

Roberto, en cambio, ya estaba casado y tenía dos hijos, con lo cual, el dinero le alcanzó para construir una casa de tres ambientes y dos baños, pero con menor confort.

En el caso de Carlos, la casa estaba pintada y sus techos eran de cemento. Roberto, por su parte, no había podido terminar de cementar las paredes, sus techos eran de chapa y el garaje se inundaba con las fuertes lluvias ya que las canaletas daban todas hacia ese lugar. Carlos?, el no tenía garaje, dejaba el auto en la calle.

Con el tiempo, Carlos también se casó y formó una familia. Era tiempo de agrandar la casa. Ambos decidieron solicitar un préstamo al banco, poniendo como garantía la ferretería.

En el caso de Carlos, su arreglo correspondían a crear nuevas habitaciones y hacer que éstas encajen con el diseño de lo que ya había construido tiempo atrás, en cambio, Roberto, se encargó de corregir los desvíos de su casa, terminar las paredes y cambiar los techos.

Los dos terminaron con hermosas casas. Roberto, construyó su casa con un desarrollo Iterativo y Carlos con un desarrollo Incremental.