

# Introducción a Quality Assurance

## Definición de Calidad

Es importante conocer a qué apunta el concepto de “calidad” en el contexto del desarrollo de software.

La calidad, no solamente está ligada a la ausencia de errores sino también a conseguir de manera eficiente la satisfacción del usuario detrás de los objetivos comerciales.

Veamos algunas definiciones:

"La calidad del software es la totalidad de las características y características de un producto o servicio que se relacionan con su capacidad para satisfacer las necesidades declaradas o implícitas del usuario." Fuente: IEEE Std 610.12-1990 (Estándar para la Definición de Términos de Ingeniería de Software).

"La calidad del producto de software es la conformidad con los requisitos explícitos e implícitos del usuario, las necesidades del cliente, las normas establecidas y las características especificadas en la norma ISO/IEC 9126-1. La calidad también se refiere al grado en que el producto satisface las expectativas del usuario, proporciona un valor añadido y cumple con los estándares de la industria y las normas pertinentes. Fuente: ISO/IEC 9126-1 (Calidad del Producto de Software).

"La calidad es cumplir con los requisitos." Fuente: "Calidad es Gratis" – Philip B. Crosby (Libro), experto en gestión de calidad.

"La calidad de un producto es la medida en que satisface las expectativas." Fuente: "Principles of Software Engineering Management" – Tom Gilb (Libro), reconocido consultor y autor en el campo de la ingeniería de software y la gestión de proyectos

"La calidad es el valor para alguien que importa." Fuente: "Quality Software Management: Systems Thinking" – Gerald M. Weinberg (Libro).

"La calidad del software es la concordancia de los resultados del trabajo de ingeniería del software con las expectativas del cliente y del usuario." Fuente: "Quality Software Management: Systems Thinking" - Gerald M. Weinberg (Libro).

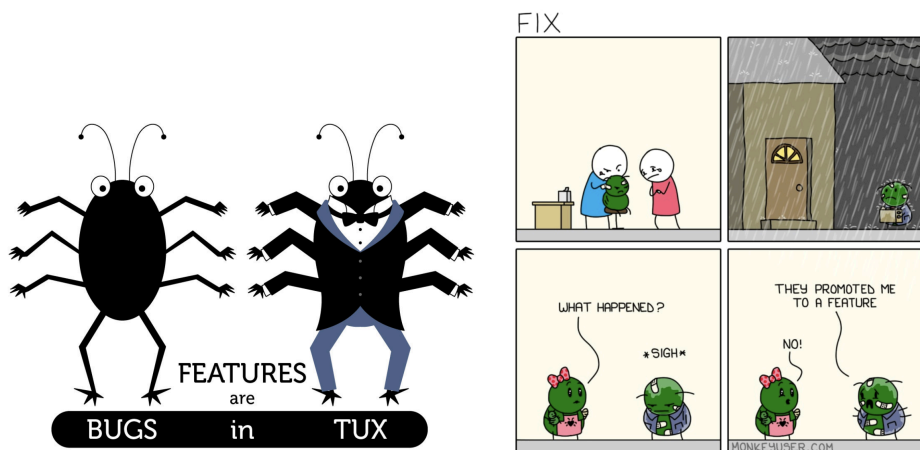
A continuación, te invitamos a ver el siguiente video sobre los fundamentos de la calidad del software.

👉 "Los fundamentos de la calidad del software" 🎥 [\[video\]](#)

💡 Tomar y registrar los **key concepts** y anotar dudas para trabajar con tu equipo de trabajo del día.

Repasamos nuevamente estas imágenes que aparecen en el video. Nos referimos a la historieta y al meme.

¿Cómo se relacionan con el testing y la calidad?



Fuentes: <https://dev.to/codemouse92/turning-bugs-into-features-48gl>

## ¿Qué es QA?

QA, o Aseguramiento de la Calidad, es una práctica esencial en el desarrollo de software que se enfoca en garantizar que los productos finales cumplan con los estándares de calidad establecidos. El objetivo principal de QA es identificar problemas y defectos en el software antes de que llegue a los usuarios finales.

## ¿Por qué es importante el QA?

En el mundo del desarrollo de software, la calidad es clave. Un producto con errores puede llevar a una mala experiencia del usuario, afectar la reputación de la empresa y generar costos adicionales en correcciones. El QA ayuda a prevenir estos problemas asegurando que el producto sea robusto, confiable y funcione como se espera.

## Rol del tester: ¿Que es un tester en la industria del software?

Es un profesional especializado en realizar pruebas y evaluaciones sistemáticas de software para identificar y reportar posibles defectos o problemas. La función principal del tester es garantizar la calidad del software antes de su lanzamiento al público.

La denominación exacta de los roles en el área de aseguramiento de calidad puede variar según la empresa y la industria. A continuación, te proporciono algunas definiciones generales de roles relacionados con testing y aseguramiento de calidad:

- **Tester QA (Control de Calidad):** : Un tester QA es un profesional especializado en la ejecución de pruebas y evaluaciones para garantizar la calidad del software. Este rol implica la identificación de defectos, la elaboración de casos de prueba y la colaboración con el equipo de desarrollo para mejorar la calidad del producto.

Referencia: Pressman, R. S. (2014). "Software Engineering: A Practitioner's Approach." McGraw-Hill Education.

- **Ingeniero de Pruebas (Test Engineer):** Se centra en el diseño, implementación y ejecución de casos de prueba. Este profesional trabaja para garantizar la funcionalidad, rendimiento y seguridad del software a lo largo del ciclo de vida del desarrollo. Referencia: Sommerville, I. (2011). "Software Engineering." Addison-Wesley.
- **Analista de Calidad de Software (Software Quality Analyst):** Un analista de calidad de software evalúa la calidad del software mediante la ejecución de pruebas y análisis de resultados. Este rol implica la identificación y documentación de problemas, así como la colaboración en la mejora continua de procesos.

Referencia: Pfleeger, S. L., & Atlee, J. M. (2006). "Software Engineering: Theory and Practice." Prentice Hall.

## **Rol de QA y Rol del Desarrollador: ¿todo junto o separado?**

A menudo, es posible encontrar espacios donde los roles no se encuentran diferenciados.

Aquí te proponemos tres razones que fundamentan la existencia de roles diferenciados:

1. En procesos de creación (como el desarrollo de software) suele ser difícil tomar distancia crítica sobre el producto creado.

Se torna complejo -para el creador (es decir, el desarrollador)- encontrar sus propias fallas. Por esa razón es más efectivo la diferenciación de roles, siendo los especialistas en QA quienes testean y prueban el producto creado.

2. El programador/desarrollador no puede simular el lugar de usuario final y probar todos los escenarios y combinaciones de uso posibles para su desarrollo. En primer lugar porque no es su función primordial, en segundo porque de ser así le insumiría mucho tiempo que podría destinar a desarrollar mejoras en las funcionalidades del producto.

3. El tester QA desarrolla una estrategia para testear el producto y luego lleva a cabo todas las pruebas posibles.

El crecimiento del área de desarrollo de software puso de manifiesto la necesidad de contar con un profesional que asumiera la responsabilidad de analizar y testear los productos creados. Es así, que el área de QA adquiere relevancia para la industria IT.

El programador desarrolla teniendo en cuenta los requerimientos y especificaciones solicitadas. Sin embargo, en cualquiera de las etapas de su trabajo (creación de un programa o cambios en productos ya en funcionamiento) existen variables que podrían generar errores en el funcionamiento del producto.

Por eso, es fundamental la presencia de profesionales que revisen (testeen) la funcionalidad desde la definición solicitada, hasta su implementación en el código.

Todo esto debe realizarse antes de colocar el producto en el mercado. Es decir, mientras antes se evalúen los posibles errores, mejor.

En la industria IT, se testea antes de que un producto o sistema avance a la etapa de producción detectando errores y comprobando que cumpla con los requisitos solicitados.

¿Qué tipos de problemas te imaginás que sucederían si no se hiciese testeo antes de la producción? ¿Cómo afecta eso al negocio -en caso de tratarse de una empresa- o a la reputación de un profesional independiente?

Al **testear**, se podrían encontrar con defectos que son productos de gaps o faltas de definiciones.

Analicemos su definición:

- Gap o Falta de definición: Algo que no funciona o funciona de manera errática. El escenario que da lugar a ese funcionamiento puede no haber sido considerado al momento de definir los requerimientos y/o especificaciones del sistema.

En resumen: El objetivo del **testing** es asegurar que los requerimientos funcionales y no funcionales sean analizados antes de que un sistema entre en servicio para que cualquier defecto encontrado, sea reportado al equipo de desarrollo y pueda corregirse. Los test a un sistema permiten medir aspectos relacionados a la calidad.

### **¿Cómo y cuándo interviene un tester en el desarrollo de un software?**

La intervención del tester en el desarrollo de software es esencial a lo largo de todo el ciclo de vida del proyecto. Su participación comienza desde las etapas iniciales y se extiende hasta la entrega final del producto.

Para comprender esto, es necesario revisar brevemente el ciclo de vida de producción de software (o SDLC, software development lifecycle, por sus siglas en inglés).

La producción de software se puede iniciar por una de dos vías:

- **Proyecto:** Existe un cliente particular con una necesidad específica para su negocio. Ej: Una peluquería que trabaja con turnos que desea además cobrar en el momento en el que se gestiona el turno.

El cliente entonces contacta a un proveedor de software y le detalla sus necesidades. El desarrollador (puede ser un individuo o una compañía, muchas veces llamadas software factories) toma nota de los requerimientos, hace las preguntas necesarias para entender mejor aquellas cuestiones que el cliente no sabe especificar - si desea que sea posible tomar turnos en feriados, por ejemplo - y luego realiza una cotización basándose en horas y cantidad de desarrolladores necesarios. Si el cliente lo acepta, se inicia el proyecto.

💡 **Entonces:** un proyecto es una solución particular para un cliente con necesidades específicas.

- **Producto:** La propuesta de software para un producto se inicia con la detección de una necesidad en el mercado. Por ejemplo, podemos decir que la compra remota es una necesidad que puede ser satisfecha a gran escala. Soluciones como Amazon, Ebay, Mercado Libre pretenden dar solución a esta necesidad.

💡 **Entonces:** un producto pretende resolver un problema a gran escala y sus usuarios habitualmente van directo a consumirlo.

El desarrollo de software es un proceso sistemático empleado para construir software de manera eficiente y efectiva. **El ciclo de vida del desarrollo de software** (SDLC) abarca diversas fases, cada una con objetivos y actividades específicas. Estos aspectos serán abordados en detalle posteriormente, profundizando de manera independiente en cada una de las etapas.