

¿Qué es probar?

Probar software es una manera de “asegurar la calidad” del producto y “reducir el riesgo” de fallas cuando esté operando y con esto aumentar la confianza del cliente en el producto, y por consiguiente en la marca. Es el proceso que garantiza la correcta ejecución de las pruebas, Probar algo significa revisar que todo esté bien, que funciona según lo esperado.

Hay dos errores comunes en la definición de pruebas de software.

1. Las pruebas de software solo consisten en ejecutar el software y comprobar los resultados.
2. Las pruebas de software se enfocan en verificar que el software cumpla con la verificación de requerimientos.

Las pruebas de software puede dividirse en dos tipos:

- Pruebas dinámicas Implica ejecutar el software para revisar resultados.
- Pruebas estáticas: Revisión de documentos de requerimientos.

La Verificación y validación son dos cosas diferentes es importante preguntarse en cada una:

Verificación: ¿Construimos BIEN el software según los requerimientos?

Validación ¿Construimos el software ADECUADO basado en requerimientos CORRECTOS?

Definición de Pruebas de Software. La podemos dividir en 2 partes:

El proceso y los objetivos

Respecto al proceso: probar es un proceso no una actividad, involucra todo el ciclo de vida de desarrollo. Abarca pruebas estáticas y dinámicas. Se planifica antes, durante y después de la ejecución de las pruebas. Las pruebas se deben preparar, se verifica el resultado, se prueba el código y los productos de trabajo relacionados.

Respecto a los objetivos, se verifica y se valida el cumplimiento de los requisitos.

Al detectar defectos comprendemos riesgos, mejoramos la calidad de los productos analizamos la causa raíz y mejoramos los procesos de desarrollo.

Es sumamente importante:

- Planificación

- Análisis
- Diseño
- Implementación y ejecución
- Generación del reporte

Además:

1. Evaluar los productos de trabajo que se generan en cada tarea significativa del ciclo de vida de desarrollo de software.
 - Producto de trabajo "Work Product"
2. Verificar que los requerimientos sean satisfechos
3. Validar el objeto de prueba: Es importante saber si esta completo y si funciona como los usuarios lo esperan. Validar: asegurar que se entregó lo que el usuario solicitó.
4. Generar confianza, esto se da si verifica y se valida correctamente
5. Prevenir defectos: Al encontrar y corregir defectos durante las pruebas se disminuye el riesgo de fallas cuando el software ya se esté ejecutando u operando.

Un defecto se introduce en el código y es la consecuencia de un error, un defecto puede generar otro, por ende se puede evitar este error cometido por una persona corrigiendo el primer error y así se previene el segundo.
6. Encontrar fallas y defectos: Es importante encontrar tantos defectos como sea posible en las pruebas, ya que esto aumenta la confianza en que el software no fallará tanto cuando este operando. No es posible garantizar que el producto se entregará 100% libre de defectos y fallas
7. Entregar la información para tomar decisiones
8. Reducción del nivel de software de baja calidad

Los objetivos de las pruebas varían dependiente del sistema que se vaya a probar y en qué fase del ciclo de desarrollo se está o el nivel del prueba.

Pruebas y Depuración

Probar es distinto a depurar, ya que depurar es el proceso de desarrollo que encuentra, analiza y repara el defecto y probar es verificar que el software funcione como se espera.

El ciclo de depuración implica que el desarrollador depura el código y realiza sus pruebas, luego del ciclo de depuración, un probador independiente ejecuta las pruebas de confirmación.

Las pruebas de verificaciones se realizan como su nombre lo dice para verificar que el cambio que ha sido realizado no haya afectado otras secciones del software.

¿Por qué es necesario probar?

1. Al ser humanos, cometemos errores y esto introducen defectos que pueden generar fallas.

2. Reduce el riesgo y aumenta la confianza.
3. Hay defectos que son difíciles de identificar ya que se originan en suposiciones incorrectas.
4. Algunas pruebas son obligatorias debido a estándares legales.

Las pruebas son rentables porque detectan defectos. Y por ende se pueden depurar ya sea una actividad que no es de prueba y así da mayor calidad.

La evaluación directa de la calidad de un objeto de prueba a lo largo de diversas fases del Ciclo de Vida del Desarrollo de Software se logra mediante la realización de pruebas. Estas evaluaciones forman parte de una gestión de proyectos más amplia y contribuyen a las decisiones que implican avanzar a la siguiente etapa del Ciclo de Vida del Desarrollo de Software, como, por ejemplo, la decisión de realizar una entrega.

Además las pruebas le proporcionan a los usuarios la seguridad de que se ha comprendido las necesidades que requieren y que se tendrán en cuenta en su medida ya sea por disposición o costo; en cada ciclo de vida del desarrollo o en su mayoría.

Pruebas y Aseguramiento de Calidad (QA)

QA se enfoca en la prevención y mejora continua de los procesos para garantizar la calidad desde el inicio, mientras que QC se concentra en la identificación y corrección de defectos en el producto final. Ambos son esenciales para un sistema de gestión de calidad efectivo pero en sí son diferentes.

Principios de la Prueba

1. Las pruebas muestran la presencia, no la ausencia de defectos.
2. Las pruebas exhaustivas son imposibles.
3. Las pruebas tempranas ahorran tiempo y dinero.
4. Los defectos se agrupan.
5. Las pruebas se desgastan.
6. Las pruebas dependen del contexto.
7. Falacia de la ausencia de defectos.

Las pruebas son contextualmente dependientes, pero en líneas generales, se identifican conjuntos comunes de actividades esenciales para alcanzar los objetivos de prueba, conformando así un proceso de prueba. La adaptabilidad de este proceso a una situación específica se determina mediante la planificación de la prueba, donde se decide qué actividades se incluirán, cómo se implementarán y cuándo ocurrirán, según factores diversos.

Actividades y Tareas de la Prueba:

La Planificación significa establecer metas y elegir la mejor manera de alcanzarlas dentro de las limitaciones.

El Monitoreo es estar al tanto de lo que estás haciendo y compararlo con un plan y Control es tomar acciones si es necesario.

Análisis de prueba significa examinar lo que se está probando. Evaluar las cosas que se pueden verificar, definir condiciones de pruebas y priorizar lo que es más importante. También mirar riesgos y defectos.

El diseño de la prueba, se da al planificar cómo se va a hacer las pruebas. Esto implica pensar en las condiciones de prueba, que son como las reglas o situaciones que queremos revisar. Luego, creamos casos de prueba y otros materiales de prueba, como cartas de prueba.

Al diseñar la prueba, es necesario pensar en qué datos se necesitan, cómo debe ser el entorno de prueba y qué herramientas se necesitan.

Para implementar la prueba es necesario ejecutarla por ejemplo los datos que están en prueba. Estos se pueden organizar en procedimientos de prueba. Es importante crear scripts de pruebas manuales automatizados, así priorizando y organizándolo en el cronograma de ejecución para que sea eficiente.