**¿Qué son las pruebas y por qué deberíamos hacerlas?:**

Un error puede hacer que un usuario deje de usar una aplicación, ya que arruina la experiencia de usuario. Hay mejores prácticas para evitar introducir errores.

Las pruebas son una manera de explorar, experimentar y entenderlo, mientras menos se entienda el producto, más errores va a tener. Las pruebas son el proceso de evaluar un producto, aprendiendo a través de la exploración y experimentación: cuestionar, estudiar, observar e inferir, checar salidas de datos. Etc.

Aunque un software pase todas las pruebas no quiere decir que no tenga errores, porque cuando las personas usan nuevos dispositivos o salen nuevas versiones de software, ya no se puede estar a la par de esas tecnologías, así que hay que actualizar el software. Ni importa lo bueno que sea el plan de pruebas siempre se van detalles, esto paso con la presentación del reconocimiento facial de Apple y en una presentación en vivo de Windows 98, además hay riesgos legales implícitos, como en el caso de Uber tras un gran robo de datos.

Las pruebas se tienen que hacer por diversas cuestiones, pero muchas empresas no consideran las pruebas parte del desarrollo, así que hay tres razones para hacer las pruebas

* Resultado no esperado
* Mucho costo (muchos errores duplican el trabajo)
* Implicaciones legales o de estándares tecnológicos (quien lo usa, donde lo usa)

**Proceso de pruebas del software y los estándares internacionales:**

Cuando se desarrolla software hay que tener:

* Recursos: aunque se quieran hacer pruebas pero no se está preparados, después esas carencias se vuelven defectos
* Metodología: se establece la estrategia de como se van a llevar a cabo las pruebas
* Herramientas: permite agilizar el proceso

Hay etapas para diseño, programación, pruebas y liberación, pero la mayor parte de los errores salen durante la etapa de diseño y análisis, hay pruebas desde la concepción de los requisitos hasta su puesta final en producción.

Escala de tiempo

Descripción generada automáticamenteLas pruebas revisan la calidad el producto o sea si los datos de salida del producto están cumpliendo las especificaciones, o la calidad del proceso o sea si se sigue paso a paso para evitar omitir información.

* Definir la calidad del producto: Si estas creando el software correctamente.
* Calidad del proceso: Si alguien de tu equipo no te especifica lo que quieres no vas a poder crear un producto que cubra sus necesidades.

Una empresa para cumplir las expectativas del cliente tiene certificaciones, estándares y metodologías para individuos, proceso, la empresa en general, servicios/productos (software/hardware), y para tipo de industria. Una certificación no asegura que se hagan todas las pruebas, solo que siempre se hacen da la misma manera, así que hay que complementar las certificaciones. Los estándares son: ISTQB, IEEE, TPI

**Ciclo de vida del software:**

Diagrama

Descripción generada automáticamenteA lo largo del ciclo de vida del software se realizan distintas pruebas para garantizar que este cumpla con los requerimientos para los que fue diseñado y de la misma forma se encuentren procesos de mejora y optimización a medida que se desarrolla el software.

* Es necesario hacer pruebas en todas las fases del desarrollo de software ya que un error encontrado en una etapa tardía puede generar costos muy elevados.
* Errores detectados lo antes posible reducen los costos y son mucho más fáciles de corregir.
* El ciclo de vida permite que los errores se detecten lo antes posible y por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

El ciclo de vida básico de un software consta de los siguientes procedimientos:

1. Definición de objetivos: En esta fase se define el alcance general del software y su papel dentro de una estrategia global o dentro del ecosistema que va a funcionar.
2. Análisis de los requisitos y su viabilidad: Se trata de recopilar la mayor cantidad de información posible para evaluar la viabilidad del producto, encontrar posibles restricciones y analizar todos los requisitos del cliente.
3. Diseño:
   1. Alto nivel: Se trata de realizar un diseño básico que valide la arquitectura de la aplicación.
   2. Bajo nivel: Es una definición detallada de la estructura de la aplicación basada en el diseño general.
4. Programación: Es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
5. Pruebas de verificación: Aunque en todas las fases anteriores se hacen pruebas en esta fase se cubren: pruebas de componentes, integrales y de sistema.
6. Prueba beta (o validación): Se hace para garantizar que el software cumple con las especificaciones originales o también se hacen las pruebas de aceptación.
7. Implementación: Se realiza una prueba del sistema implementado para encontrar posibles fallas en la Implementación.
8. Mantenimiento: Se hace para todos los procedimientos correctivos (mantenimiento correctivo) y a las actualizaciones secundarias del software (mantenimiento continuo), junto con la actualización de las pruebas.

Si hacemos las pruebas durante cada fase del ciclo de vida del software tendremos al final del ciclo un producto validado y robusto de acuerdo con las necesidades del cliente.

**Proceso de pruebas del software: Calidad y Defectos**

La calidad es la percepción de lo analizado, lo deseado y lo que se va a entregar, así que la calidad la define el cliente, no solo se encuentra dentro del producto sino que también hace parte del proceso, es necesario saber que es la calidad del cliente.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteLa IEEE lo define como el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario. Los estándares de la IEE forman parte de la documentación, y esta tiene que especificar que quiere el cliente.

Las pruebas no solo se enfocan a programar, sino que también se puede hacer en la fase de análisis con un bench marking.

* Verificación: es ir en cada una de las etapas revisando lo que el cliente pidió
* Validación: va al final del proceso, se valida que en conjunto de todo cumple

Al hacer pruebas puede suceder una de las siguientes situaciones

* Anomalía: cualquier insatisfactoria condición, no es reproducible
* Defecto: problema que puedes reproducir muchas veces
* Fallos: situaciones no asociadas al software, como un corte de internet
* Error: acción humana incorrecta

Lo que hay que resolver realmente es el error humano.

**Principios del testing moderno:**

Principios creados por Alan Page y Brent Jensen, evolución natural del testing ágil, ya que El tester y el desarrollador deberían trabajar en equipo. 7 principios del testing moderno:

1. Nuestra prioridad es mejorar el negocio: El producto que se va a entregar al cliente permitirá hacer funcionar el negocio. Si en algún momento no quieres hacerlo, estás poniendo en riesgo ese negocio porque si el producto no se vende o no es aceptado la empresa puede cerrar o puedes perder el trabajo.
2. Nosotros aceleramos el equipo y usamos modelos como Lean Thinking y Teoría de las Restricciones para ayudar a identificar, priorizar y mitigar cuellos de botella en el sistema: Cuando queremos hacer algo, lo queremos hacer perfecto y eso puede ser demasiado. Deberías construir en base a procesos cortos para poder encontrar los defectos de una manera más rápida.
3. Nosotros somos la fuerza para la mejora continua, ayudando al equipo a adaptarse y optimizar para tener éxito, en lugar de proporcionar una red de seguridad para detectar fallas: El cliente puede entender que el producto se va a liberar por fases, es importante que nosotros enfoquemos nuestras pruebas en cada una de esas fases. No tiene que ser todo al inicio y al final, debe haber una distribución que nos permita manejar el riesgo del software
4. Nos preocupamos profundamente acerca de la cultura de calidad en nuestro equipo, y asesoramos, lideramos y nutrimos el equipo para llevarlos a una cultura de calidad más madura: Al inicio los testers eran personas desarrollando software y un día con tantos defectos y trabajo, separaron los roles para que así hubiese una persona dedicada a realizar las pruebas. El tester puede hacer recomendaciones de herramientas, mejorar el proceso o volverse un coach.
5. Nosotros creemos que el cliente es el único capaz de juzgar y evaluar la calidad de nuestro producto: Si el cliente está satisfecho con lo entregado y cumple las expectativas entonces has alcanzado la calidad deseada.
6. Nosotros usamos datos de manera extensa y profunda para entender los casos de uso del cliente y entonces cerrar huecos entre hipótesis del producto e impacto del negocio.
7. Expandimos las habilidades de testing y el conocimiento en todo el equipo; entendemos que esto reduce o elimina la necesidad de un especialista dedicado al testing.

El tester debe dominar varias áreas, necesita entender y tener toda la visión del producto y negocio. Saber sobre herramientas que optimicen el trabajo.

**Especialidades del testing:**