¿Porque es necesario el Testing?

Vamos a ir desglosando el porque es necesario desde distintos puntos de vista:

Si miramos el contexto de los sistemas hoy

* **Por la importancia económica del software**
  + El funcionamiento de maquinaria y equipamiento depende en gran medida del software
  + No es posible imaginar grandes sistemas, en el ámbito de las finanzas ni el control del tráfico automotor, entre otros, funcionando sin software.
* **Por la calidad del Software**
  + Cada vez más, la calidad software se ha convertido en un factor determinante del éxito de sistemas y productos técnicos o comerciales
* **Por las pruebas para la mejora de la calidad**
  + Las pruebas y revisiones aseguran la mejora de la calidad de productos de software así como de la calidad del proceso de desarrollo en sí.
* **Por los riesgos**
  + No todo el software tiene el mismo nivel de riesgo y no todos los problemas tienen el mismo impacto cuando ocurren.

Si miramos porque fallan los sistemas

Todos sabemos que las personas cometemos errores, ahora ¿Cómo inciden estos errores en los sistemas?

Vamos a definir un **Error (IEEE 610)**como: Acción humana que produce un resultado incorrecto. Ej. Un error de programación.

Este ‘Error’ va a producir un**Defecto (defect, bug)**en el software: Desperfecto en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas, por ejemplo una sentencia o una definición de datos incorrecta.

Si se localiza un defecto durante una ejecución puede causar un fallo en el componente o sistema. **Fallo (Failure)**es la manifestación física o funcional de un defecto. Si un defecto es encontrado durante la ejecución de una aplicación puede producir un fallo. Es la desviación de un componente o sistema respecto de la prestación, servicios o resultado esperado.

Si miramos la relación entre el testing y la calidad

La importancia del testing es que  nos ayuda a medir la calidad del software, nos da confianza en el software permitiéndonos reducir el nivel general de riesgo del sistema.

La calidad no solo cubre los aspectos del sistema/producto, también cubre los procesos y como estos pueden ser mejorados, ayuda a prevenir a que los errores no vuelvan a ocurrir. Por esto, el testing debe ser considerado como parte de una larga estrategia de aseguramiento de la calidad lo que incluye otras actividades (standars de desarrollo, capacitación, etc…)

**Ahora, ¿Cuánto testing es necesario?**

El testing exhaustivo es imposible. Ya que para poder detectar todas las posibles errores en un sistema, vamos a necesitar realizar todas los caminos posibles de combinación de entradas y lógica de ejecución. Testear todas las combinaciones de entradas y precondiciones no es factible, por eso para enfocar el testing nos debemos basar en Riesgos y Prioridades.

Utilizar el riesgo para determinar:

* que testear primero
* que testear más
* que tan profundo testear cada item
* Que no testear (por esta vez)

El cliente y los managers van a querer realizar una cantidad de testing que les provea un retorno de la inversión. El criterio de salida es típicamente usado para manejar el problema de cómo determinar cuando el testing de una fase en particular está completo

**¿Qué es Testing?**

Si vamos a la percepción común, el testing solo consiste en correr pruebas o ejecutar la aplicación y por supuesto, puede ser realizado por cualquier persona. Este percepción es falsa, ya que testing es mucho más que eso, es todo un proceso que incluye actividades antes, durante y después de la ejecución de pruebas. Además, puede tomar diferentes formas en base al objetivo de las pruebas. Por otro parte, el testing también contribuye en la búsqueda/prevención de defectos, proporciona información sobre la calidad de un sistema/aplicación para poder tener un grado determinado de confianza sobre el mismo.

Si entramos en definiciones más detalladas, podemos decir que el testing es:

* Un proceso de ejecución de pruebas para demostrar que un programa es libre de errores.
* El proceso de recopilación de información mediante observaciones y compararlas con las expectativas.
* Las pruebas de software es un proceso de ejecución de un programa con la intención de encontrar errores.
* Es una investigación técnica y empírica de un producto, hecho en nombre de los interesados, con la intención de revelar información relacionada con la calidad de un producto o servicio.
* Según IEEE: El proceso de funcionamiento de un sistema o componente bajo ciertas condiciones, observar o registrar los resultados, y hacer una evaluación de algún aspecto del sistema o componente.

**¿Pero cómo lo define la norma ISTQB?**

*El****proceso****que consiste en****todas las actividades del ciclo de vida****, tanto****estáticas como dinámicas****relacionadas con la****planificación, preparación y evaluación de productos****de software y productos relacionados con el trabajo para****determinar que cumplen los requisitos especificados****, para****demostrar que son aptos****para el propósito y para****detectar defectos****.*

 En primer lugar, echemos un vistazo a las pruebas como un proceso:

• **Proceso:**La prueba es un proceso más que una sola actividad. Hay una serie de actividades involucradas.

• **Todas las actividades del ciclo de vida:**se lleva a cabo durante todo el ciclo de desarrollo de software tratando de encontrar y corregir defectos desde el inicio del ciclo de vida a fin de reducir el coste total.

• **Tanto estáticos como dinámicos:** Así como hay pruebas donde se ejecuta el código del software para demostrar los resultados, también pueden probar y encontrarse defectos sin ejecutar código (revisión de documentos y el código).

• **Planificación, preparación, evaluación** se llevará a cabo antes y después de la ejecución de pruebas. Tenemos que administrar la planificación de las pruebas, el seguimiento y la presentación de informes. También tenemos que identificar lo que vamos a hacer, mediante la selección de las condiciones de prueba y el diseño de casos de prueba, así como evaluar el sistema bajo prueba los criterios de finalización, lo que ayuda a decidir si hemos terminado nuestras pruebas.

• **Productos de software y productos relacionados con el trabajo:** Probamos muchas otras cosas además de un código como los requisitos, especificaciones de diseño, manuales de usuario, etc.

La segunda parte de la definición se refiere a las razones del por qué lo hacemos:

• **Determinar que (los productos de software) cumplir con los requisitos especificados:** Si el producto cumple con sus especificaciones, podemos proporcionar esa información para ayudar a los stakeholders a ​​juzgar la calidad del producto y decidir si está listo para su uso.

• **Demostrar que (los productos de software) son aptos para el propósito:** Hace referencia a si el software hace lo suficiente para ayudar a los usuarios a llevar a cabo sus tareas o no.

• **Detectar defectos:** Esto nos ayuda a entender los riesgos asociados con poner el software en uso operacional, así como mejorar el desarrollo de procesos para cometer menos errores en el futuro.