**Siete principios de pruebas.**

1. **Las pruebas muestran la presencia de defectos, no su ausencia.**

Las pruebas pueden mostrar que hay defectos, pero no pueden probar que no hay defectos. Las pruebas reducenla probabilidad de que queden defectos no descubiertos en el software, pero, incluso si no se encuentran defectos, las pruebasno es una prueba de corrección.

1. **Es imposible realizar pruebas exhaustivas**

Probar todo (todas las combinaciones de entradas y condiciones previas) no es factible excepto en casos triviales.

En lugar de intentar realizar pruebas exhaustivas, se deben utilizar el análisis de riesgos, las técnicas de prueba y las prioridades paraconcentrar los esfuerzos de prueba.

1. **Las pruebas tempranas ahorran tiempo y dinero**

Para encontrar defectos temprano, las actividades de prueba tanto estáticas como dinámicas deben iniciarse lo antes posible en elCiclo de vida del desarrollo de programas. Las pruebas tempranas a veces se denominan desplazamiento a la izquierda. Probando temprano en elEl ciclo de vida del desarrollo de software ayuda a reducir o eliminar cambios costosos (consulte la sección 3.1).

1. **Los defectos se agrupan.**

Una pequeña cantidad de módulos generalmente contiene la mayoría de los defectos descubiertos durante las pruebas previas al lanzamiento, oes responsable de la mayoría de las fallas operativas. Grupos de defectos predichos y los observados realesLos grupos de defectos en prueba u operación, son una entrada importante en un análisis de riesgo utilizado para enfocar el esfuerzo de prueba.(como se menciona en el principio 2)

1. **Cuidado con la paradoja de los pesticidas**

Si las mismas pruebas se repiten una y otra vez, eventualmente estas pruebas ya no encontrarán ningún defecto nuevo.Para detectar nuevos defectos, es posible que sea necesario cambiar las pruebas existentes y los datos de las pruebas, y es posible que sea necesario realizar nuevas pruebas.escrito. (Las pruebas ya no son efectivas para encontrar defectos, al igual que los pesticidas ya no son efectivos para matarinsectos después de un tiempo.) En algunos casos, como las pruebas de regresión automatizadas, la paradoja de los plaguicidas tiene unresultado beneficioso, que es el número relativamente bajo de defectos de regresión.

1. **Las pruebas dependen del contexto**

**Las** pruebas se realizan de manera diferente en diferentes contextos. Por ejemplo, el software de control industrial crítico para la seguridad esprobado de forma diferente a una aplicación móvil de comercio electrónico. Como otro ejemplo, se realizan pruebas en un proyecto ágilde forma diferente a las pruebas en un proyecto de ciclo de vida de desarrollo de software secuencial (consulte la sección 2.1).

1. **La ausencia de errores es una falacia**

Algunas organizaciones esperan que los evaluadores puedan ejecutar todas las pruebas posibles y encontrar todos los defectos posibles, perolos principios 2 y 1, respectivamente, nos dicen que esto es imposible. Además, es una falacia (es decir, una creencia errónea)esperar que el simple hecho de encontrar y corregir una gran cantidad de defectos asegure el éxito de un sistema. ParaPor ejemplo, probar minuciosamente todos los requisitos especificados y corregir todos los defectos encontrados aún podría producir unsistema que es difícil de usar, que no satisface las necesidades y expectativas de los usuarios, o que es inferioren comparación con otros sistemas de la competencia.

**Tecnicas de pruebas:**

**Partición de equivalencia:**

La participación de equivalencia divide los datos de particiones (también conocidas como clases de equivalencia) de tal manera que se espera que todos los miembros de una participación determinada se procesen de la misma manera.

* Los valores validos son los valores que deben ser aceptados por el componente y el sistema. Una partición de equivalencia que contienen los valores validos se denomina (“Participación de equivalencia valido”).
* Los valores no validos son los valores que deben ser rechazados por el componente o el sistema. Una partición de equivalencia que contienen los valores no validos se denomina (“Participación de equivalencia no valido”).
* Se pueden identificar particiones para cualquier elemento de datos relacionados con el objetivo de prueba, incluida las entradas, salidas, valores internos, valores relacionados con el tiempo (“Ejemplo: antes y después de un evento”) y para la interfaz parámetros (“por ejemplo componentes integrados que se prueban durante la prueba de integración”).
* Cualquier partición se puede dividir en subparticiones si es necesario.
* Cada valor debe pertenecer a una y solo una partición de equivalencia.
* Cuando se utilizan particiones de equivalencia no válidas en casos de prueba, deben probarse individualmente, es decir, no se combina con otras particiones de equivalencia no válidas, para garantizar que los fallos no queden enmascarados. Las fallas se pueden enmascarar cuando ocurren varias fallas al mismo tiempo, pero solo una es visible, haciendo que las otras fallas no sean detectadas.

Para lograr una cobertura del 100% con esta técnica, los casos de prueba deben cubrir todas las particiones identificadas (incluidas particiones inválidas) utilizando un mínimo de un valor de cada partición. La cobertura se mide como número de particiones de equivalencia probadas por al menos un valor, dividido por el número total de particiones identificadas particiones de equivalencia, normalmente expresadas como porcentaje. La partición de equivalencia es aplicable en todos niveles de prueba.

Pruebas Exploratorias

Una estructura lógica para realizar pruebas exploratorias es la de utilizar la criticidad de los módulos para la compañía, comenzando a probar lo más crítico, ya que, de esto, dependerá el negocio y los ingresos de esta.





