

Prueba de software

Martin Solari
ORTsf

Prueba de software

- La prueba es una actividad de evaluación del producto.
- Definición (SWEBOK)
 - ‘Software testing consists of the *dynamic* verification of the behavior of a program on a *finite* set of test cases, suitably *selected* from the usually infinite executions domain, against the *expected* behavior.’

Prueba de software

- Evolución de la visión de la prueba
 - No se realiza solamente al final, sino que acompaña el desarrollo.
 - No es únicamente destructiva, es una actividad que aporta información sobre el producto para tomar decisiones.
- Objetivo: ‘luces delanteras del proyecto’ (Kaner et al.)



Prueba de software

- Definir objetivos y rol de la prueba
 - No ser la actividad que trunca la liberación.
 - No es la actividad que asegura la calidad.
 - No confundir la prueba con mejora del proceso.

Prueba de software

- Principales barreras de la prueba
 - Visión de la prueba como actividad no productiva
 - Dificulta la asignación de recursos.
 - Confianza sobre el producto
 - Dificulta la profundidad del análisis y visibilidad de los defectos.



Prueba de software

- Prueba vs. Revisión
 - La prueba es la evaluación del producto mediante su ejecución. En la revisión buscamos problemas en etapas tempranas mediante un análisis estático.
 - La prueba se realiza sobre el producto (software) mientras que la revisión se puede aplicar a otros productos intermedios (especificaciones, diseños).

Prueba de software

- Prueba vs. Métricas
 - La prueba es una actividad de evaluación que brinda información.
 - Información concreta: de problemas reales.
 - Información directa: sobre el producto de software.
 - Nos permite tomar acciones que mejoran la calidad.
 - Los indicadores numéricos pueden complementar esta información.

Prueba de software

- Distintas estrategias. Generar casos de prueba basados en:
 - Especificación
 - Estructura
 - Potenciales defectos
 - Experiencia del probador
 - Uso del producto



Prueba de software

- Prueba en distintos niveles de integración del producto:
 - Unitario
 - Integración
 - Sistema
- Alfa y beta
- Aceptación

Prueba de software

- Prueba de atributos específicos:
 - Instalación
 - Seguridad
 - Performance
 - Escalabilidad
- Usabilidad

Prueba de software

- Actividades principales
 - Diseño de la prueba
 - Ejecución de la prueba
 - Reporte de la prueba
- Evaluación y mejora del conjunto de pruebas
- Repetición de casos seleccionados

Prueba de software

- Diseño de la prueba
 - Es la actividad más importante, en la que se define la prueba mediante casos concretos. Es cuando aplicamos las estrategias: funcional, estructural, etc.
 - Enfocarnos en el diseño de los casos de prueba permite prevenir faltas.
 - Según el modelo de proceso (tradicional o ágil) el diseño se aplica distinto, pero el objetivo es el mismo.
 - Documentar los casos para que otro pueda aplicarlos
 - Conducir el desarrollo mediante la prueba, test-first

Prueba de software

Plantilla de caso de prueba

- **Identificación** [Grupo] [Requerimiento asociado]
- [Entorno] [Configuración]
- [Precondición] [Conjunto de datos]

- **Estimulo** (entradas)
- **Respuesta esperada** (salidas, estado programa, base de datos)

Mínimo: como ejercitar el programa y que se espera

Prueba de software

- Ejecución de la prueba
 - Muchas veces confundimos esta actividad con el diseño de la prueba.
 - En la ejecución la prueba los casos son aplicados por un tester concreto, en cierto entorno, sobre una determinada versión del programa, en un tiempo controlado.
 - Concepto: sesión de prueba.
 - En la prueba exploratoria no hay diseño previo y se centra en la sesión:
 - Experiencia del tester, intuición, percepción, planificación adaptativa de la prueba.

Prueba de software

Plantilla de sesión de prueba (ejecución de casos / exploratoria)

- Fecha, Tester, Tiempo de sesión
- **Componente probado**, Versión

- Por cada caso ejecutado: **identificación, resultado**
- [Derivaciones de casos de prueba] [Casos adicionales]

Mínimo: que se probó y cuando

Prueba de software

- Reporte de la prueba
 - La prueba es una actividad informativa del proceso, por eso tiene que comunicar los resultados.
 - 'Los tester son los abogados de los fallos que encuentran.'
 - El reporte de la prueba debe ser un buen instrumento de comunicación.
 - Lenguaje preciso, no personalizar los fallos.
 - Breve para facilitar su uso.
 - Centrado en los problemas importantes, priorizado.

Prueba de software

Plantilla de reporte de defecto

- **Identificación** [Categoría]
- **Severidad** [Prioridad]

- [Entorno] [Configuración] [Caso asociado]
- **Sumario**, Descripción, [Comentarios]
- [Archivo]

Mínimo: pasos para reproducir el defecto

Prueba de software

- Técnicas específicas de prueba:
 - Partición de entradas.
 - Análisis de valores límites.
 - Cobertura de sentencias, decisiones, predicados.
 - Análisis de cambios y pruebas de regresión.
 - Exploración.

Prueba de software

- Herramientas para gestionar la prueba:
 - Automatizar la prueba: *conductores*.
 - Evaluar el conjunto de casos de prueba: herramientas de *cobertura*.
 - Reportar y gestionar defectos: herramientas de *bug tracking*.



Prueba de software

- Prueba en el ciclo de vida
 - En un ciclo de vida iterativo la prueba debería estar presente en todas las iteraciones ya que es la forma de contar con información sobre el producto que desarrollamos.
 - El diseño de la prueba se puede comenzar en etapas tempranas y en paralelo a la construcción. Es una actividad preventiva.

Prueba de software

- Coordinación con el desarrollo y la liberación
 - La ejecución de la prueba necesita contar con el producto.
 - La información de la prueba debe producirse en ciertos momentos del desarrollo.

Prueba de software

- Recomendaciones
 - Asegurar recursos: **reservar el tiempo de la prueba.**
 - Sobrepasar la confianza inicial y los supuestos que tenemos sobre el producto: **profundizar el análisis para obtener información que no teníamos.**



Prueba de software

- Recomendaciones
 - Si el ciclo es iterativo realizar la prueba en todas las iteraciones (si no la ejecución, el diseño de la prueba)
 - Tratar el diseño de la prueba como una actividad preventiva.
 - Pensar en el volumen de los artefactos de prueba (cientos o miles de casos y defectos) para elegir herramientas adecuadas.
 - No intentar automatizar toda la prueba.