

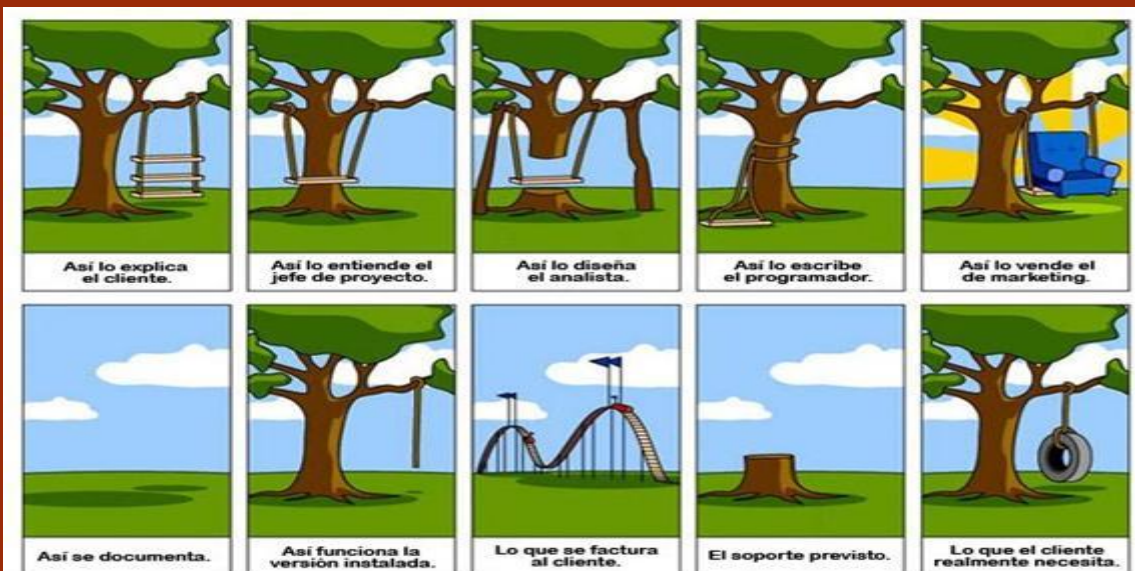
PRUEBAS DE REGRESIÓN Y DE ACEPTACIÓN

INTEGRANTES:

- Calle Mollo Ivan Marcos
- Choque Choque Wilson Henry
- Gonzales Breton Rebeca
- Soto Castellon Jose Fernando
- Via Meneces Juan Ariel

DOCENTE: Torrico Bascope Rosemary

MATERIA: Ingenieria de Software



Cochabamba – Bolivia



Prueba de Regresión y Aceptación:

Pruebas de regresión.- a cualquier tipo de pruebas de *software* que intentan descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa. Se evalúa el correcto funcionamiento del software desarrollado frente a evoluciones o cambios funcionales. El propósito de estas es asegurar que los casos de prueba que habían sido probados y fueron exitosos permanezcan así. Se recomienda que este tipo de pruebas sean automatizadas para reducir el tiempo y esfuerzo en la ejecución.

Cuando debe realizarse la prueba de regresión

Se llevan a cabo mediante la selección total o parcial de casos de prueba ejecutados previamente, que se ejecutan nuevamente para confirmar que las funcionalidades actuales funcionan sin problemas.

Se debe de equilibrar la necesidad de pasar el menor tiempo posible haciéndolo mientras maximiza la detección de nuevas fallas en el código anterior, probado anteriormente.

A lo largo de las pruebas de regresión pueden aparecer nuevos errores o regresiones de software. Una vez reconocidos, los casos de prueba que fallaron en la última ejecución de pruebas de regresión pueden tener sus defectos corregidos, y la funcionalidad se repara con éxito.

Por qué es necesario la prueba de regresión

Las pruebas de regresión siempre deben ser un componente importante del proceso de prueba. Aunque puede ser difícil de mantener, los beneficios de perseverar son significativos.

Las pruebas de regresión descubren nuevos defectos resultantes de los cambios en el producto antes de que se conviertan en un problema para el dueño del software. Recordar que la realización de pruebas de regresión tiene la capacidad de ahorrar tiempo y dinero a largo plazo.

Como hacer una prueba de regresión

Existen 4 métodos principales: El enfoque de volver a probar todos; una prueba de regresión basada en la selección; la priorización de los casos de prueba; y el método híbrido.

1. la aplicación del enfoque de volver a probar todo es un amanaera completa de garantizar que todos los casos de prueba en el programa sean monitoreados para su integridad y funcionamiento. Inconveniente de este método es que puede ser muy costoso, ya que requiere una gran inversión de tiempo y recursos.



2. Alternativa, el lugar de volver a ejecutar todo el conjunto de pruebas, la opción de selección de prueba de regresión permite a los usuarios seleccionar un parte específica del conjunto de pruebas que se ejecutara, tiene el potencial de conservar recursos de tiempo y dinero.
3. Mediante la priorización de casos de prueba. Este método prioriza los casos de prueba de acuerdo con su impacto comercial previsto, aspectos críticos y funcionalidades de uso frecuente.

Dentro de esta opción existen 2 tipos de priorización de casos de prueba: Priorización general, capacidad de priorizar casos de prueba que serán beneficiosos en versiones posteriores; priorización específica, priorizara casos de prueba con respecto a versión particular de software.

4. Técnica híbrida proporciona a los usuarios una opción flexible y es una combinación de la selección de prueba regresión y los métodos de priorización de casos de prueba.

Objetivo de las pruebas de regresión

Es la no regresión de la calidad luego de un cambio. Asegurar que los cambios no introduzcan un comportamiento no deseado.

Involucra realizar las pruebas, de forma que se pueda volver a probar con esas pruebas luego del cambio. Hay 3 tipos de pruebas de regresión:

- I. Hacer regresión de bugs arreglados cuando se reporta un bug y vuelve una nuva versión luego de arreglado. Objetivo probar que no fue arreglado.
- II. Hacer regresión de bugs viejos, que muestra que un cambio en el software causo que un bug arreglado se vuelva a romper
- III. Hacer regresión de efectos secundarios. Implica volver aprobar un aparte del producto. Objetivo es probar que el cambio ha causado que algo funcionaba ya no funcione.

Herramientas de prueba de regresión

Tipos de pruebas de regresión

Clasificación de ámbito

- Local, los cambios introducen nuevos errores.
- Desenmascarada, los cambios revelan errores previos.
- Remota, los cambios vinculan algunas partes del programa e introducen errores en ella.

Clasificación temporal



- Nueva característica, los cambios realizados con respecto a nuevas funcionalidades en la versión introducen errores en otras novedades en la misma versión el software.
- Característica preexistente, los cambios realizados con respecto a nuevas funcionalidades introducen errores en funcionalidad existente de previas versiones.

Ventajas

- Ayudan a minimizar ciertos riesgos de software
- Ayuda a rastrear los errores hasta la fuente
- Aumenta la efectividad del sistema

Desventajas

- Proceso costoso para la empresa
- Rediseño y prueba del software completo
- Efecto negativo en otra parte del código

Pruebas de aceptación.- son las últimas pruebas realizadas donde el cliente prueba el software y verifica que cumpla con las expectativas. Este tipo de pruebas generalmente son funcionales y se basan en los requisitos definidos por el cliente y deben hacerse antes de la salida a producción.

Las pruebas de aceptación son fundamentales por lo cual deben de incluirse obligatoriamente en el plan de pruebas de software.

Estas pruebas se realizan una vez que ya se ha probado que cada módulo funcione bien por separado, que el software realice las funciones esperadas y que todos los módulos se integren correctamente.

Cual es la base para definir las pruebas de aceptación del software

Según los estándares establecidos ISTQB, las pruebas de aceptación de software son diseñadas a partir de:

- Requerimientos del usuario
- Requerimientos de sistema
- Procesos de negocio

Las pruebas de aceptación, se enfocan en verificar si el sistema es “apto para el uso”. Se diseñan principalmente a partir de las especificaciones de requerimientos, casos de uso y de procesos de negocio.

En algunos casos, el cliente le pide al equipo, generalmente es el tester, que realice una demo y valide todos los requerimientos definidos.

Para el cliente o Product Owner apruebe las pruebas de aceptación no deben haber errores, como máximo algún error con crítica baja



Casos de prueba que debe incluir la aceptación de usuario

Con frecuencia, el diseño de casos de pruebas de aceptación de usuario se define buscando incluir los principales escenarios de ejecución de la aplicación.

De manera de confirmar, de manera temprana de los requerimientos han sido entendidos correctamente y que las pruebas a realizar sobre la funcionalidad son suficientes, conviene enviarla al cliente los casos de prueba definidos para pedir su opinión.

Si bien puede ocurrir que durante la sesión de pruebas de aceptación, el usuario define casos de prueba adicionales, debe hacerse todo lo posible que estos sean identificados de manera temprana.

Objetivo

Es permitir al usuario validar que el sistema cumple con el funcionamiento esperado y que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.





Existen 3 estrategias comunes para implementar una prueba de aceptación

- Aceptación formal
- Aceptación informal o prueba de versión alfa
- Prueba de versión beta

La estrategia que seleccione suele basarse en los requisitos contractuales, los estándares corporativos, de la empresa y el dominio de la aplicación.

Prueba de aceptación formal

Proceso altamente gestionado. Los casos de prueba seleccionados deben ser un subconjunto de los realizados en la prueba del sistema. Es importante no desviarse en absoluto de los casos de prueba escogidos. Pruebas de aceptación son totalmente automatizadas.

En algunas empresas, la empresa de desarrollo(grupo de prueba independiente), junto con los representantes de la empresa del usuario final, lleva a cabo la prueba de aceptación. En otras, la prueba de aceptación las lleva a cabo en su totalidad la empresa del usuario final, un grupo objetivo de personas escogidas por la empresa del usuario final.

Beneficios de esta forma de prueba:

- Las funciones y las características que se van a probar son conocidas
- Los detalles de las pruebas se conocen y se pueden medir
- Las pruebas se pueden automatizar, lo que permite realizar pruebas de regresión
- El progreso de las pruebas se puede medir y supervisar
- Los criterios de aceptabilidad son conocidos

Las desventajas incluyen:

- Requiere una planificación y recursos significativos
- Las pruebas pueden ser una re implementación de las pruebas del sistema
- Es posible que las pruebas no revelen defectos subjetivos en el software, ya que solo busca defectos que espera encontrar

Prueba de aceptación informal

Los procedimientos para realizar la prueba se definen tan rigurosamente como para las pruebas de aceptación formal. Las actividades empresariales y las funciones que se exploraran se identifican y documentan, pero no hay casos de prueba particulares que seguir. El verificador individual determina que hacer. Esta propuesta de prueba de aceptación no esta tan controlada como la prueba formal y es más subjetiva.

Las pruebas de aceptación informal las suele realizar la empresa del usuario final



Los beneficios de esta forma de prueba son:

- Las funciones y las características que van a probar son conocidas
- El progreso de las pruebas se puede medir y supervisar
- Los criterios de aceptación son conocidos
- Revela defectos mas subjetivos que la prueba de aceptación formal

Desventajas:

- Son necesarios recurso de gestión, planificación y recursos
- No se tiene control de los casos de prueba que se utilizan
- Los usuarios pueden adaptarse al funcionamiento del sistema y no apreciar los defectos
- Los usuarios pueden centrarse en la comparación del sistema nuevo con un sistema heredado, en vez de buscar defectos
- Los recursos de las pruebas de aceptación no se encuentran bajo el control del proyecto y piden limitarse

Prueba de versión beta

Es la menos controlada de las tres estrategias de prueba de aceptación. La prueba de versión beta, la cantidad de detalles y los datos y el enfoque adoptado dependen totalmente del verificador individual. Cada verificador es responsable de crear su propio entorno, seleccionar los datos y determinar las funciones, características o tareas que va a explorar. Cada verificador es responsable de identificar sus criterios para aceptar o no el sistema en su estado actual.

La prueba de versión beta la implementan los usuarios, a menudo con poca o ninguna gestión por parte de la empresa de desarrollo. La prueba de versión beta es la mas subjetiva de todas las estrategias de prueba de aceptación.

Beneficios:

- la prueba la implementan los usuarios
- hay grandes volúmenes de recursos de prueba potenciales
- los clientes que participan están cada vez mas satisfechos
- revela defectos mas subjetivos que las pruebas de aceptación formal e informal

Desventajas:

- Es posible que no pueda probar todas las funciones o características
- El progreso de la prueba es difícil de medir
- Los usuarios pueden adaptarse al funcionamiento del sistema y no apreciar o notificar los defectos
- Los usuarios pueden centrarse en la comparación del sistema nuevo con un sistema heredado, en vez de buscar defectos
- Los recurso de la prueba de aceptación no se encuentran bajo el control del proyecto y pueden limitarse
- Los criterios de aceptación no son conocidos
- Necesita más recursos de soporte para gestionar a los verificadores de beta.