



UNIVERSIDAD TECNOLÓGICA DE DURANGO

Desarrollo y Gestión de Software (DGS)

Desarrollo Móvil Integral

Investigación VII: Pruebas de software en aplicaciones móviles

Alumnos: Diana Itzel Ayala Herrera
De la Cruz Cenicerros Carlos Emmanuel
González Chávez Jorge Israel
Irving Ulises Rodríguez Romero

Docente: I.S.C Raúl Iván Herrera González

Tabla de contenido

Tabla de contenido.....	2
Tabla de ilustraciones	2
Introducción.....	3
Tipos de pruebas de software	4
Tipos de pruebas de software o tipos de testing	4
Esquema de pruebas en modo debug y release.....	7
Esquema de pruebas en modo debug.....	7
Esquema de pruebas en modo debug.....	8
Conclusión	9
Referencias	10

Tabla de ilustraciones

Ilustración 1.- Pruebas funcionales.....	4
Ilustración 2.- Pruebas no funcionales.....	5
Ilustración 3.- Pruebas estructurales	6
Ilustración 4.- Pruebas de manejo de cambios	6
Ilustración 5.- Debug software	7
Ilustración 6.- Software Release	8

Introducción

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder. Es una actividad más en el proceso de control de calidad.

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

El objetivo de las pruebas es presentar información sobre la calidad del producto a las personas responsables de este. Las pruebas de calidad presentan los siguientes objetivos: encontrar defectos o bugs, aumentar la confianza en el nivel de calidad, facilitar información para la toma de decisiones, evitar la aparición de defectos.

Teniendo esta afirmación en mente, la información que puede ser requerida es de lo más variada. Esto hace que el proceso de testing sea completamente dependiente del contexto en el que se desarrolla.

El ambiente ideal de las pruebas es aquel que es independiente del desarrollo del software, de esta manera se logra objetividad en las pruebas.

A pesar de lo que muchos promueven, no existen las "mejores prácticas" como tales. Toda práctica puede ser ideal para una situación, pero completamente inútil o incluso perjudicial en otra.

Por esto, las actividades técnicas, documentación, enfoques y demás elementos que condicionarán las pruebas a realizar deben ser seleccionadas y utilizadas de la manera más eficiente según contexto del proyecto.

Tipos de pruebas de software

Las pruebas de software (Software Testing) comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo.

Los tipos de pruebas de software se usan para mejorar la calidad de lo que hacemos mientras estamos desarrollando software. Esto nos ayuda a que el usuario se lleve la mejor experiencia y que evite pasar por malos ratos al momento de usar un producto, una página o realizando algún tipo de transacción.

Tipos de pruebas de software o tipos de testing

Existen diferentes tipos de pruebas de calidad de software que nos ayudan a mejorar lo que estamos desarrollando y pueden ser manuales o automáticas. Algunos de los tipos pruebas son:

1. Pruebas funcionales

No, no significa que solo vamos a testear una función en nuestro código.

Con este tipo de pruebas testeamos la funcionalidad de nuestro sistema o software, podemos hacernos preguntas sobre cómo funciona, qué debe estar haciendo, cómo están interactuando los usuarios. También conocidas como pruebas de caja negra.

Estas pruebas nos ayudan a revisar todas las funciones del sistema y componentes con los que el usuario tiene una interacción directa.

Las pruebas funcionales se dividen en:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de sanidad
- Pruebas de regresión
- Pruebas de aceptación
- Pruebas de Sistema

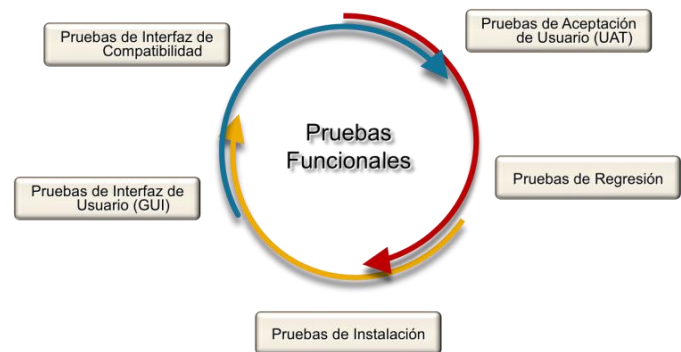


Ilustración 1.- Pruebas funcionales

2. Pruebas no-funcionales

Puede que tengamos un sistema funcionando, pero el usuario está experimentando otro tipo de problemas que no son detectados por las pruebas anteriores.

Factores como lentitud, problemas en la combinación de colores provocando poca legibilidad o claridad, usabilidad, seguridad, son las que testeamos en con este tipo de pruebas.

Así que las pruebas no funcionales se enfocan en elementos de experiencia de usuario, diseño, seguridad y, en general, de calidad del sistema.

Algunas pruebas no funcionales existen son:

- Pruebas de rendimiento
- Pruebas de portabilidad
- Pruebas de usabilidad
- Pruebas de carga
- Pruebas de estrés



Ilustración 2.- Pruebas no funcionales

3. Pruebas estructurales

Mejor conocidas como pruebas de caja blanca porque conocemos todo nuestro stack de tecnología.

Acá estamos interesados en conocer cómo funciona nuestro sistema, no con la perspectiva del usuario sino como desarrollador, por eso con este tipo de pruebas es necesario conocer cómo está internamente implementado nuestro código, cómo funciona “por dentro”.

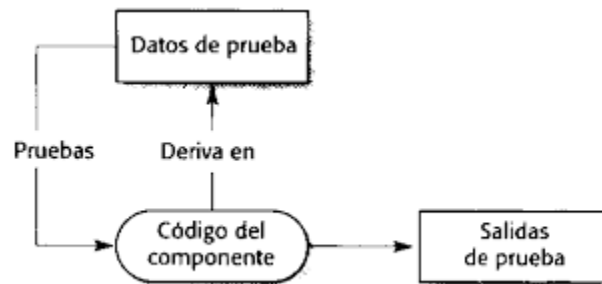


Ilustración 3.- Pruebas estructurales

4. Pruebas de manejo de cambios

En esta parte probamos nuevamente un componente o un módulo para verificar que no haya sido afectado por actualizaciones realizadas en otras partes de nuestro software y así evitamos que los usuarios puedan percatarse del error.

Es probar lo ya probado en cada actualización lanzada.

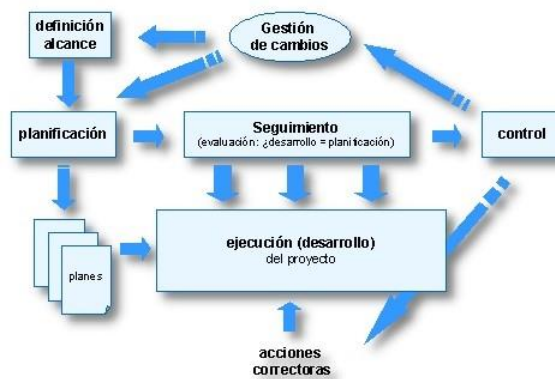


Ilustración 4.- Pruebas de manejo de cambios

Esquema de pruebas en modo debug y release

Debug and Release son solo tags para diferentes configuraciones de soluciones . Puede agregar otros si lo desea. Un proyecto en el que una vez trabajé tenía uno llamado “Debug Internal” que se usaba para activar las funciones de edición interna de la aplicación. Puede ver esto si va al Configuration Manager... (está en el menú Build). Puede encontrar más información en MSDN Library en el cuadro de diálogo Configuration Manager .

Cada configuración de solución consiste en un conjunto de configuraciones de proyectos . De nuevo, estas son solo tags, esta vez para una colección de configuraciones para su proyecto.

Esquema de pruebas en modo debug

La depuración de programas es el proceso de identificar y corregir errores de programación.¹ En inglés se conoce como debugging, porque se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.

Si bien existen técnicas para la revisión sistemática del código fuente y se cuenta con medios computacionales para la detección de errores (depuradores) y facilidades integradas en los sistemas lower CASE y en los ambientes de desarrollo integrado, sigue siendo en buena medida una actividad manual, que desafía la paciencia, la imaginación y la intuición de programadores. Muchas veces se requiere incluir en el código fuente instrucciones auxiliares que permitan el seguimiento de la ejecución del programa, presentando los valores de variables y direcciones de memoria y ralentizando la salida de datos ("modo de depuración"). Dentro de un proceso formal de aseguramiento de la calidad, puede ser asimilado al concepto de "prueba unitaria".



Ilustración 5.- Debug software

Esquema de pruebas en modo debug

El release de un software es la distribución del mismo, su documentación y materiales de soporte. Release y versión pueden considerarse sinónimos en ocasiones. De todas maneras, si hilamos fino, una "release" es una versión de un software, pero no todas las versiones del software son release.

La versión release es una versión de lanzamiento, es decir, que el software se hace público. En ocasiones una versión del software puede ser una release candidate o candidata a lanzamiento; es decir, es una versión previa al lanzamiento definitivo de dicho software.

En cambio, una versión de un software no necesariamente es una release.

La distinción puede verse cuando se traduce la palabra "release" que significa: lanzamiento, estreno, liberación. Evidentemente no todas las versiones de un software son liberadas o lanzadas. De hecho, las correcciones menores en las versiones de un software no suelen considerarse releases.

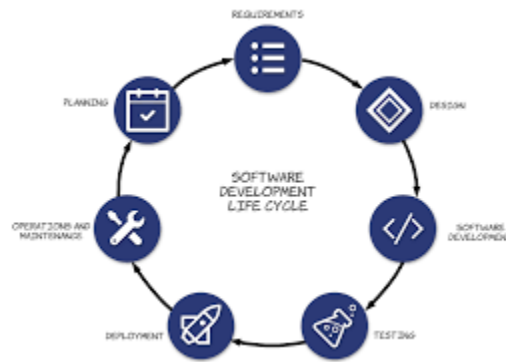


Ilustración 6.- Software Release

Conclusión

El testing de software es una de las actividades más importantes y fundamentales en el desarrollo de un proyecto, ya que posibilita los procesos, métodos de trabajo y herramientas necesarias para garantizar la calidad de cualquier desarrollo.

Llevar a cabo la realización de las pruebas nos permite asegurar el funcionamiento correcto de cada una de las partes funcionales de los elementos que conforman el software.

Para realizar el testing es necesario conocer las diversas pruebas que existen dentro del desarrollo del software y el orden en el cual son aplicadas cada una de estas. Ya que cada una de estas pruebas se encargan de probar ciertos aspectos en específicos como es el caso de las pruebas unitarias las cuales se encargan de comprobar el correcto funcionamiento de las funciones. Otro ejemplo son las pruebas de integración las cuales son ejecutadas después de las pruebas unitarias las cuales se encargan de asegurar el correcto funcionamiento de las funciones en conjunto.

Durante la investigación se conocieron dos conceptos muy importantes dentro del desarrollo los cuales son los diversos esquemas o métodos en los cuales se puede llevar a cabo la creación del software. Estos dos conceptos hacen referencia al debug y al release.

El debug hace referencia a un entorno en el cual se buscan los diversos errores que pueden provocar un fallo dentro del software. Dentro de este entorno se realizan acciones llamadas depuraciones que cuentan con el propósito de limpiar dichos problemas con objetivo de eliminar los errores.

El release se refiere a un entorno de la versión final en la cual se encuentran los desarrolladores trabajando. Es una versión en la cual se realizan las pruebas para posteriormente pasar a producción.

Referencias

- Laballos, D. (18 de 11 de 2021). *DiegoLaballos*. Obtenido de <https://diegolaballos.com/podcast/40-que-es-y-para-que-sirve-el-testing-en-programacion/>
- MDN. (18 de 11 de 2021). *Mozilla Developer Network*. Obtenido de <https://social.msdn.microsoft.com/Forums/es-ES/f232736d-adb3-4d25-b5bc-1d334c3b6d02/diferencia-entre-debug-y-release?forum=vbes>
- Platzi. (18 de 11 de 2021). *Platzi*. Obtenido de <https://platzi.com/discusiones/1375-fundamentos-csharp/54235-que-diferencia-existe-entre-compile-un-proyecto-en-debug-o-release/>
- Quastak. (18 de 11 de 2021). *Quastak*. Obtenido de <https://qastack.mx/programming/367884/what-is-the-difference-between-debug-and-release-in-visual-studio>