

Pruebas de software

umh2818-TADS

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.

1 Historia

El objetivo de las pruebas es presentar información sobre la calidad del producto a las personas responsables de éste. 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^[1] en el que se desarrolla.

El ambiente ideal de las pruebas de testing 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 tal. 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.

2 Pruebas estáticas

Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación.

Puede referirse a la revisión de documentos, ya que no

se hace una ejecución de código. Esto se debe a que se pueden realizar “pruebas de escritorio” con el objetivo de seguir los flujos de la aplicación.

3 Pruebas dinámicas

Todas aquellas pruebas que para su ejecución requieren la ejecución de la aplicación.

Las pruebas dinámicas permiten el uso de técnicas de caja negra y caja blanca con mayor amplitud. Debido a la naturaleza dinámica de la ejecución de pruebas es posible medir con mayor precisión el comportamiento de la aplicación desarrollada.

4 Tipos de pruebas

Hay todo tipo de pruebas, pero nos centraremos en tres de ellas:

5 Pruebas de Compatibilidad

Se comprueba el funcionamiento del software desarrollado en muchas plataformas: sistemas operativos, navegadores, redes, hardware...entre otros

6 Pruebas de Regresión

Se evalúa el correcto funcionamiento del software desarrollado frente a evoluciones o cambios funcionales. El propósito de éstas es asegurar que los casos de prueba que ya habían sido probados y fueron exitosos permanezcan así. Se recomienda que este tipo de pruebas sean automatizadas para reducir el tiempo y esfuerzo en su ejecución.

7 Pruebas de Integración

Es el nivel de pruebas posterior a las pruebas modulares de los componentes de un sistema. Se centra principalmente en probar la comunicación entre los componentes de un mismo sistema, comunicación entre sistemas o entre hardware y software.

8 Tipos de pruebas por su ejecución

- Pruebas manuales
- Pruebas automáticas

9 Enfoques de pruebas

- Pruebas de Caja blanca
- Pruebas de Caja negra
- Testing aleatorio^[2]

10 Niveles de pruebas

- Pruebas unitarias
- Pruebas modulares
- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación de usuario UAT

11 Pruebas funcionales

- Pruebas funcionales
- Pruebas de humo
- Pruebas de regresión
- Pruebas de aceptación
- Alpha testing
- Beta testing

12 Pruebas no funcionales

- Pruebas no funcionales
- Pruebas de seguridad
- Pruebas de usabilidad
- Pruebas de rendimiento
- Pruebas de internacionalización y localización
- Pruebas de escalabilidad
- Pruebas de mantenibilidad
- Pruebas de instalabilidad
- Pruebas de portabilidad

13 Véase también

- Casos de prueba
- Desarrollo guiado por pruebas
- Oráculo

14 Referencias

- [1] <http://context-driven-testing.com/>
- [2] Barrientos, Pablo Andrés (25 de abril de 2014). *Enfoque para pruebas de unidad basado en la generación aleatoria de objetos*. p. 101. Consultado el 28 de abril de 2014.

15 Enlaces externos

- Context-driven Testing
- www.testingeducation.org
- ISO/IEC/IEEE 29119 Pruebas de Software (Grupo de Trabajo de AENOR AEN/CTN71/SC7/GT26)
- MÉTRICA v3 en el CSAE

16 Origen del texto y las imágenes, colaboradores y licencias

16.1 Texto

- **Pruebas de software** *Fuente:* https://es.wikipedia.org/wiki/Pruebas_de_software?oldid=89218965 *Colaboradores:* Robbot, Rosarino, Ascánder, Tano4595, Daniel G., Digigalos, Deleatur, Alhen, Yrbot, BOT-Superzerocool, Martingala, GermanX, Quiron, Er Komandante, Chlewbob, Roche, CEM-bot, Especiales, Jonpagecr, Isha, JAnDbot, Denarius, TXiKiBoT, Netito777, Idioma-bot, Developer, Biasoli, VolkovBot, Matdrodes, Synthebot, Gmarinp, El Pantera, SieBot, PaintBot, Loveless, Bigsus-bot, Alben9586, Switcher6746, Belb, Fadesga, Leonpolanco, Alejandro Lodes, UA31, AVBOT, MastiBot, MelancholieBot, Luckas-bot, Amirobot, Nallimbot, LordboT, ArthurBot, SuperBraulio13, Xqbot, Jkbw, Cvielma, BOTirithel, Halfdrag, RedBot, Pilarbarrio, Mr.Ajedrez, Pol29, Tarawa1943, Waeswaes, Wikiléptico, G Garro, EmausBot, Duhi, Elías, Douglasfugazi, CocuBot, Rezabot, Botero5, Invadibot, LlamaAI, Elvisor, QualityDave, Addbot, Tutitutti, Gonzalo.villarreal, Jarould, Miguel.DeLeonT, Edward9104, Hellboy 2k6 y Anónimos: 78

16.2 Imágenes

16.3 Licencia del contenido

- Creative Commons Attribution-Share Alike 3.0