Wikiprint Book

Title: Proceso de Construcción de Aplicaciones de Software Libre

Subject: Mayaguaray - iPruebas~

Version: 1

Date: 01/07/24 13:32:19

Table of Contents

Proceso de Construcción de Aplicaciones de Software Libre

Fase de Pruebas

En esta fase se elaboran y aplican pruebas funcionales y no-funcionales a cada versión del software, así como pruebas de regresión y de instalación, con lo cual se facilita la detección temprana de errores e/o incompatibilidades en el código. Estas pruebas deben ser elaboradas y aplicadas por probadores de software, quienes se recomiendan deben ser personas distintas a quienes codifican la aplicación.

Cada versión del software que se obtiene pasa por la fase de pruebas, de esta manera los planes de prueba evolucionan con cada iteración, pues en cada una de éstas se agregan casos de prueba para verificar el comportamiento de las versiones del software y el cumplimiento de atributos de calidad.

En esta fase no se elaboran pruebas de integración, dado que las pruebas funcionales permiten a su vez verificar la integración entre los componentes o módulos del software, razón por la cual no se considera necesario realizar las pruebas de integración, lo cual permite agilizar el proceso de desarrollo del software.

A continuación se indican las actividades y tareas que componen esta fase.

Actividad: Aplicación de pruebas funcionales

Tarea: Elaborar el plan de pruebas funcionales correspondientes a las funcionalidades desarrolladas en la iteración actual.

Recomendaciones:

- Las pruebas funcionales se utilizan para verificar que el software ejecute sus funciones según lo establecido en la especificación de requerimientos funcionales. Por esta razón, la elaboración de las pruebas funcionales se basa en la descripción textual de los casos de uso definidos para el software.
- Un plan de pruebas funcionales se compone de casos de prueba con los cuales se verifica el comportamiento de las funcionalidades del software, en términos de los escenarios indicados en la descripción textual de los casos de uso asociados a dichas funcionalidades, es decir, en términos de flujos básicos, flujos alternativos y requisitos especiales.
- Los datos utilizados en los casos de prueba deben ser de tipo válidos e inválidos, a fin de verificar el comportamiento del software ante estos tipos de datos. Por lo general, el comportamiento del software ante datos de entrada inválidos es especificado en los flujos alternativos de los casos de uso especificados para el software.
- Un escenario de un caso de uso puede tener asociado varios casos de prueba, conforme a la cantidad y/o complejidad de los datos de entrada al software indicados en el caso de uso.
- Para agilizar la elaboración y aplicación de las pruebas funcionales se sugiere, de ser posible, abarcar varios escenarios en un mismo caso de prueba, por ejemplo, se podrían abarcar varios flujos alternativos en un mismo caso de prueba.
- Un caso de prueba debe contener: a) Escenario bajo el cual se realiza el caso de prueba, es decir, debe indicar si la prueba se realiza para el flujo básico, para flujos alternativos o para requisitos especiales del caso de uso respectivo. b) Objetivo del caso de prueba, en éste se indica el propósito que se persigue al ingresar u omitir datos de entrada al software para un escenario específico. c) Pre-condición, es un requisito que debe cumplir el software para poder ejecutar el caso de prueba, por lo general este requisito es la misma condición de entrada que se especifica en la descripción textual del caso de uso respectivo. d) Pasos para realizar el caso de prueba, acá se indican los pasos que debe llevar a cabo el probador en el software para ejecutar la prueba, por ejemplo: 1) Entrar a la función “Registrar proyecto”. 2) Ingresar los datos solicitados a excepción del título del proyecto, el cual debe dejarse en blanco. 3) Pulsar la opción “Guardar”. e) Salida esperada, es decir, el resultado que debe emitir el software, según el escenario de prueba, indicado en la descripción del caso de uso respectivo. f) Salida obtenida, acá se indica si el comportamiento del software al aplicar el caso de prueba respectivo es igual o diferente a la salida esperada. Una salida obtenida que sea diferente a la salida esperada se considera como un error o falla en el software.
- Para agilizar el proceso de elaboración del plan de pruebas funcionales se recomienda no indicar en el mismo, de manera explicita, los datos a ingresar para ejecutar cada caso de prueba. En este sentido, para que el Probador tenga conocimiento acerca de los datos que debe ingresar al software, para ejecutar los casos de prueba, basta con mencionar en el plan el objetivo que se persigue con dichos casos de prueba.

Herramientas:

- Existen varias técnicas de prueba, en el caso de esta metodología se utiliza la técnica de diseño de pruebas Caja Negra (black-box testing) para elaborar casos de pruebas funcionales.
- El plan de pruebas puede registrarse en la plantilla Pruebas funcionales'

Productos:

- Plan de pruebas funcionales.

Responsables:

- Probadores.

Tarea: Aplicar el plan de pruebas funcionales correspondientes a las funcionalidades desarrolladas en la iteración actual.

Recomendaciones:

- En vista de que las pruebas funcionales pueden utilizarse como pruebas de regresión se recomienda utilizar herramientas que permitan grabar la ejecución de las pruebas funcionales, de modo que éstas puedan ser reproducidas posteriormente para verificar comportamiento del software después de haber modificado funcionalidades, o haber corregido errores.
- Las pruebas funcionales al igual que las no funcionales pueden ser aplicadas en forma manual, de forma automatizada o apoyándose en el uso de herramientas de prueba.
- La automatización de las pruebas es una actividad que requiere tiempo y esfuerzo, por lo cual ésta se recomienda siempre que se puedan reutilizar los casos de prueba automatizados, y/o en ocasiones en que resulte complicado ejecutar pruebas de forma manual, por ejemplo, cuando se requiere hacer pruebas no funcionales para verificar comportamiento y tiempos de respuesta del software cuando existen muchos usuarios conectados al mismo tiempo. En este caso, por lo general, el número de probadores con los que se puede contar no es suficiente para realizar estos tipos de prueba de forma manual, por lo que la automatización de dichas pruebas es la mejor opción (Oliveira y Gouveia, 2006).
- Los errores que se encuentren al aplicar las pruebas funcionales deben reportarse al Líder de Proyecto, a fin de que éste gestione la corrección de los mismos. Para reportar un error se debe indicar el caso de prueba en el cual ocurre el error, así como una captura de la pantalla donde se indica el error en el software.

Herramientas:

- Existen diversas herramientas para aplicar pruebas funcionales, entre estas se encuentran: Selenium (permiten grabar las pruebas), Jmeter, Watir (para pruebas de aplicaciones web en Ruby), SOLEX (permiten grabar las pruebas), entre otras.
- El reporte de errores se puede hacer por vía correo electrónico, a través de mecanismos de reporte o directamente a través de un sistema para gestión de errores.

Productos:

- Reporte de pruebas funcionales.

Responsables:

- Probadores.

Actividad: Aplicación de pruebas no funcionales

Tarea: Elaborar el plan de pruebas no funcionales correspondientes a las funcionalidades desarrolladas en la iteración actual.

Recomendaciones:

- Las pruebas no funcionales se utilizan para verificar en el software el cumplimiento de los atributos de calidad (requerimientos no-funcionales) establecidos para el mismo. Existen varios tipos de pruebas no funcionales, entre éstas se encuentran: pruebas de seguridad, pruebas de rendimiento, pruebas de usabilidad y pruebas de portabilidad. Entre las pruebas no funcionales mas aplicadas se encuentran las pruebas de rendimiento, con las cuales se estudia el comportamiento del software ante cargas máximas o mínimas de entrada de datos, de actividades o de almacenamiento.
- Dado los diferentes tipos de pruebas no funcionales que se pueden aplicar, y teniendo en consideración la importancia que adquieren las pruebas de rendimiento para la gran mayoría de los software, en esta metodología solo se especificará, con cierto detalle, la elaboración de planes de pruebas de rendimiento. En este tipo de planes se indica el ambiente de prueba a utilizar y las variables del software a estudiar. En lo referente al ambiente de prueba se indican los recursos lógicos y físicos a utilizar en las pruebas. Los recursos lógicos se refieren a las herramientas para la realización de pruebas, por ejemplo, herramientas automatizadas. Los recursos físicos se corresponden a las características del equipo (hardware) a utilizar para realizar las pruebas, por ejemplo, tipo de computador y su velocidad, tipo de memoria, características de disco duro, etc. Las variables del software que se estudian con las pruebas de rendimiento constituyen atributos de calidad, como por ejemplo: tiempos de respuesta,

capacidad de almacenamiento de datos, entre otras.

Herramientas:

- El plan de pruebas puede registrarse en la plantilla Pruebas no funcionales

Productos:

- Plan de pruebas no funcionales.

Responsables:

- Probadores.

Tarea: Aplicar el plan de pruebas no funcionales correspondientes a las funcionalidades desarrolladas en la iteración actual.

Recomendaciones:

- La aplicación de pruebas de rendimiento de forma manual es poco factible, dado que se requiere de una gran cantidad de probadores utilizando el software en un mismo período de tiempo, por lo cual se recomienda el uso de herramientas de prueba y/o la automatización de éstas.
- Los resultados de las pruebas de rendimiento deben indicar cuando el software falla para determinados valores de las variables que se estudian en estas pruebas. Por ejemplo, en una prueba de rendimiento en la cual se estudia la variable “número de usuarios conectados al mismo tiempo”, se debe poder indicar como resultado de la prueba el número de usuarios que soporta el software conectados al mismo tiempo antes de presentar fallas de conectividad o de operación.
- Los responsables de aplicar las pruebas de rendimiento deben contar con conocimientos y habilidades en el área de programación, dado que estas pruebas implica la simulación de un número de usuarios interactuando con las funcionalidades del software.
- Los errores del software que se encuentren al aplicar las pruebas no funcionales deben reportarse al Líder de Proyecto, a fin de que éste gestione la corrección de los mismos. Para reportar un error se debe mostrar una captura de la pantalla donde se indica el error en el software.
- Posteriormente a la corrección de los errores que se reporten durante este tipo de pruebas se deben repetir éstas para verificar que el software opere correctamente.

Herramientas:

- Para aplicar pruebas de rendimiento se pueden utilizar herramientas como: JMeter (realizada en Java), JCrawler, SOLEX (también se puede usar para aplicar pruebas de regresión), entre otras.
- Para aplicar pruebas de seguridad se pueden utilizar herramientas como: Powerfuzzer, Nessus, Netcat, John the Ripper y OpenSSH.
- Para reportar los resultados de las pruebas no funcionales se pueden utilizar las herramientas planteadas para el reporte de errores en pruebas funcionales.

Productos:

- Reporte de pruebas no funcionales.

Responsables:

- Probadores

Colaboradores:

- Programadores.

Actividad: Aplicación de pruebas de regresión

Tarea: Aplicar pruebas de regresión a la versión del software obtenida en la iteración actual.

Recomendaciones:

- Las pruebas de regresión se utilizan, generalmente, después de la corrección de errores o modificaciones en el código que puedan generar alteraciones considerables en el software, dado que al corregir errores o modificar el código se pueden introducir errores en el software que no existían antes de tales correcciones o modificaciones. En este sentido, el objetivo de las pruebas de regresión es verificar si se han introducido errores en el software después de la corrección de errores o de modificaciones al código, para lo cual se realizan pruebas de tipo funcional (Williams, Succi y Marchesi, 2003). En este sentido, las pruebas de regresión se definen en sí como pruebas funcionales, de allí la recomendación de utilizar herramientas que permitan grabar las pruebas funcionales aplicadas al software, con el fin de poder reproducirlas como pruebas de regresión.
- Existen tres tipos de pruebas de regresión: pruebas para todas las funcionalidades del software, pruebas sólo para las funcionalidades que han sido modificadas y pruebas para las funcionalidades o piezas de código que se puedan ver afectadas por las modificaciones realizadas en el software.
- Para reportar los errores encontrados en las pruebas de regresión se sugiere utilizar el mismo procedimiento indicado para el caso de reporte de errores en pruebas funcionales.

Herramientas:

- Para reportar los errores encontrados en las pruebas de regresión se pueden utilizar las herramientas planteadas para el reporte de errores en pruebas funcionales.

Productos:

- Reporte de pruebas de regresión.

Responsables:

- Probadores.

Actividad: Aplicación de pruebas de instalación/desinstalación

Tarea: Probar el proceso de instalación/desinstalación de la aplicación de software en los hardware y software para los cuales ésta pueda operar.

Recomendaciones:

- Las pruebas de instalación/desinstalación básicamente radican en ejecutar los pasos definidos para tales acciones en el manual de usuarios. Con estas pruebas se busca verificar que el proceso de instalación/desinstalación de la aplicación, ya sea éste manual o automatizado, no presente fallas.
- Los errores que se encuentren en el proceso de instalación/desinstalación deben reportarse al Líder de Proyecto, a fin de que éste gestione la corrección de los mismos. Para reportar un error se debe presentar una captura de la pantalla donde se indica el error en el software.
- Posteriormente a la corrección de los errores que se reporten durante este tipo de pruebas se deben repetir éstas para verificar que el proceso de instalación/desinstalación se lleve a cabo sin presentar fallas.

Herramientas:

- Para reportar las fallas del proceso de instalación/desinstalación se pueden utilizar las herramientas planteadas para el reporte de errores en pruebas funcionales.

Productos:

- Reporte de pruebas de instalación/desinstalación.

Responsables:

- Probadores.

volver a metodología