

Gestión de las Pruebas Funcionales

Beatriz Pérez Lamancha

Centro de Ensayos de Software
Universidad de la República, Montevideo, Uruguay
bperez@fing.edu.uy

Resumen

Se presenta una estrategia para la gestión de las pruebas funcionales de un producto de software. La estrategia define el alcance y la agenda de las pruebas a partir del análisis de riesgo del producto, combinando los casos de prueba con diseño previo y el testing exploratorio. Se definen los ciclos de prueba a partir del plan de desarrollo del producto y se elabora un plan general en función de las funcionalidades del producto, dicho plan se revisa y refina al comenzar cada ciclo de prueba. El testing exploratorio cumple un papel fundamental en la estrategia. Por un lado, ayuda a mitigar la posibilidad de equivocarse al realizar el análisis de riesgo del producto, dejando de lado funcionalidades importantes para el negocio. Por otro lado, complementa la prueba con diseño previo cuando no se dispone del tiempo suficiente en un ciclo de prueba, para generar los casos de prueba que cubran las funcionalidades requeridas.

1. Introducción

El objetivo de las pruebas funcionales es validar si el comportamiento observado del software cumple o no con sus especificaciones. La prueba exhaustiva del producto requiere ejercitar todos los caminos posibles del mismo, el tiempo requerido para esto, incluso para un programa pequeño es prohibitivo. Esto impacta directamente en la economía de las pruebas, ya que se deberán realizar suposiciones sobre el comportamiento del programa, y la forma en que se diseñan los casos de prueba para el mismo. El objetivo es maximizar la producción de las pruebas, esto es, maximizar el número de los errores encontrados por un número finito de los casos de prueba [1]. La técnica de selección

de los casos de prueba es uno de los principales supuestos a definir.

Existen distintos procesos y metodologías para las pruebas de software. La mayoría de ellos distingue las etapas de planificación, diseño y de ejecución de las pruebas. Durante la planificación de las pruebas se decide qué se probará y con qué profundidad. En la etapa de diseño de las pruebas funcionales, la especificación se analiza para derivar los casos de prueba y en la etapa de ejecución es donde se ejecutan los casos de prueba diseñados previamente, se compara el resultado real con el esperado y se reportan los resultados.

Se presenta una estrategia para la gestión de las pruebas funcionales que define el alcance y la agenda de las pruebas del proyecto en función del análisis de riesgo del producto, combinando las pruebas con diseño previo y el testing exploratorio. De esta forma se retroalimentan los casos de pruebas diseñados con los resultados del testing exploratorio. Esta estrategia de gestión es utilizada en el Centro de Ensayos de Software (CES) [2] para realizar servicios de prueba independiente.

En la sección 2 se presentan los principales conceptos referentes a las pruebas funcionales, en la sección 3 se presentan los principales enfoques existentes referidos al proceso de pruebas. En la sección 4 se describe la estrategia de planificación de las pruebas funcionales y por último, en la sección 5 se presentan las conclusiones.

2. Pruebas Funcionales

En esta sección se definen los principales términos referidos en este trabajo. Se define prueba funcional, prueba de regresión y testing exploratorio.

El objetivo de las pruebas funcionales es validar si el comportamiento observado del software cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario [3]. Las funciones son probadas ingresando las entradas y examinando las salidas. La estructura interna del programa raramente es considerada [4].

Para realizar pruebas funcionales, la especificación se analiza para derivar los casos de prueba. Técnicas como partición de equivalencia, análisis del valor límite, grafo causa-efecto y conjetura de errores son especialmente pertinentes para las pruebas funcionales. Se deben considerar condiciones inválidas e inesperadas de la entrada y tener en cuenta que la definición del resultado esperado es una parte vital de un caso de la prueba. El propósito de la prueba funcional es mostrar discrepancias con la especificación y no demostrar que el programa cumple su especificación [1].

Las Pruebas de Regresión tienen como objetivo verificar que no ocurrió una regresión en la calidad del producto luego de un cambio, asegurando que los cambios no introducen un comportamiento no deseado u errores adicionales. Implican la reejecución de alguna o todas las pruebas realizadas anteriormente [5].

El término "testing exploratorio" fue introducido por Cem Kaner, se trata de ejecutar las pruebas a medida que se piensa en ellas, sin gastar demasiado tiempo en preparar o explicar las pruebas, confiando en los instintos. El testing exploratorio se define como el aprendizaje, el diseño de casos de prueba y la ejecución de las pruebas en forma simultánea. En otras palabras, es cualquier prueba en la cual quien prueba controla activamente el diseño de las pruebas mientras las pruebas se ejecutan, y utiliza la información obtenida mientras prueba para diseñar nuevas y mejores pruebas [6]. En el testing exploratorio siempre se debe tomar nota de lo que se hizo y lo que sucedió [4]. Los resultados del testing exploratorio no son necesariamente diferentes de aquellos obtenidos de la prueba con diseño previo y ambos enfoques para las pruebas son compatibles [6].

El testing exploratorio puede ser realizado en cualquier situación donde no sea obvio cual es la próxima prueba que se debe realizar. También cuando se requiere obtener retroalimentación

rápida de cierto producto o funcionalidad, se necesita aprender el producto rápidamente, se quiere investigar y aislar un defecto en particular, se quiere investigar el estado de un riesgo particular, o se quiere evaluar la necesidad de diseñar pruebas para determinada área. Una estrategia básica para realizar testing exploratorio es tener un plan de ataque general, pero permitirse desviarse de él por periodos cortos de tiempo. Cem Kaner llama a esto el principio del "tour bus", las personas bajan del bus y conocen los alrededores. La clave es no perderse el tour entero [6]. En general, el testing solamente exploratorio puede funcionar para testers con mucha experiencia. Como ventaja se encuentra que es barato y rápido, como contra que no se tienen medidas de cubrimiento y no deja documentación [5]. En [8] se presenta la estrategia de testing exploratorio utilizada para probar un producto de software por el equipo del Centro de Ensayos de Software (CES).

3. Estrategia para la Gestión de las Pruebas Funcionales

En esta sección se presenta la estrategia para la gestión de las pruebas funcionales de un producto de software. Las principales características de la estrategia son que se basa en realizar un estudio de los riesgos del producto que permita definir el alcance para las pruebas, define los ciclos de prueba que se realizarán del producto en función del plan de desarrollo y usa un enfoque iterativo para la planificación de las pruebas, donde se define una planificación general a nivel macro de las funcionalidades a probar en cada ciclo, la cual es revisada y refinada al comenzar cada ciclo de prueba.

3.1. Prueba basada en los riesgos del producto

El enfoque basado en los riesgos tiene tres pasos: primero confeccionar una lista priorizada de riesgos, luego realizar pruebas que exploran cada riesgo, y por último, cuando un riesgo se mitiga y emergen nuevos, se debe ajustar el esfuerzo de la prueba [14]. Un riesgo es la probabilidad de que algo no deseado ocurra. La magnitud de un riesgo es proporcional a la

probabilidad y el impacto del problema. Cuanto más probable es que el problema suceda, y más alto el impacto, más alto es el riesgo asociado a ese problema. Existen dos enfoques heurísticos para el análisis de riesgo, uno "desde adentro hacia fuera" y otro "desde afuera hacia adentro". Son acercamientos complementarios, cada uno con sus fortalezas. El enfoque de adentro hacia fuera pregunta "¿qué riesgos se asocian a esta funcionalidad?", mientras que el enfoque de afuera hacia adentro es el opuesto "¿qué funcionalidades se asocian a esta clase de riesgo?".

El enfoque "desde afuera hacia adentro" estudia una lista de riesgos potenciales y se realiza la correspondencia con los detalles del producto. Con este enfoque, se consulta una lista predefinida de riesgos y se determina si aplican [14].

El enfoque usado en este artículo para la gestión de las pruebas es el "desde adentro hacia fuera" para realizar el estudio de riesgos del producto. Se estudia en detalle el producto y se identifican los riesgos asociados a cada parte del mismo, aquellas partes del sistema que en caso de fallar tienen las consecuencias más serias y aquellas que tienen mayor frecuencia de uso, ya que si una parte del sistema es usada frecuentemente y tiene un error, su uso frecuente probablemente hará aparecer la falla [10].

3.2. Ciclo de Prueba

Con cada nueva versión del producto se realizan alguna o todas las tareas asociadas a las pruebas, a esto se le llama un ciclo de prueba.

Durante el ciclo de vida de un producto, sin importar cual sea el proceso de desarrollo, se van generando distintas versiones de la aplicación. Las actividades de la prueba se realizan para una determinada versión del producto, sobre la cual se ejecutan las pruebas y se reportan los incidentes encontrados. Las pruebas que serán ejecutadas sobre una versión son planificadas con anticipación y deberían ser ejecutadas, a menos que las prioridades cambien.

En un ciclo de prueba se puede ejecutar una, alguna o todas las pruebas planificadas para el producto. Cada ciclo de prueba está asociado a una versión del producto a probar, cada nuevo

ciclo de prueba implica una nueva versión de uno o más componentes del sistema [5].

Uno de los principales desafíos desde el punto de vista de la prueba independiente es estimar cuantos ciclos de prueba se requieren, ya que no todas las versiones que genera desarrollo llegan a ser probadas por el equipo de prueba, entre dos ciclos de prueba podrían existir más de dos versiones del producto generadas por el equipo de desarrollo.

3.3. Alcance de las Pruebas Funcionales

El alcance de las pruebas indica qué funcionalidades se van a probar y cuales no. Para poder definir el alcance, se divide el sistema en módulos, componentes o subsistemas, no todos los componentes serán probados con la misma importancia y pueden existir componentes que queden fuera del alcance de las pruebas. Cada componente agrupa varias funcionalidades, se dividen las funcionalidades hasta un nivel en el que sea posible definir el alcance. Luego de esto, se analizan las funcionalidades, dando como resultado un Inventario de Pruebas. En la Tabla 1 se muestra un ejemplo de la priorización de las funcionalidades. El inventario es una lista de las funcionalidades del producto de software. Para cada funcionalidad se asigna:

- **Identificador:** Referencia única a la especificación de requerimientos, donde se encuentra la descripción detallada del mismo.
- **Nombre:** Nombre de la funcionalidad a probar.
- **Prioridad:** Indica la prioridad que tiene esa funcionalidad para las pruebas. Los valores posibles son: Alta, Media y Baja.

Estos valores se obtienen de utilizar el enfoque basado en los riesgos del producto de la sección 3.1.

Una vez que se tiene el inventario de pruebas, se define el alcance. Para esto se estima el tiempo en probar cada funcionalidad, utilizando la información de proyectos anteriores similares y la experiencia del equipo de pruebas. Conociendo esta información, el plan de desarrollo del producto y la fecha prevista para instalar el producto en el ambiente de producción, se define el alcance para las

pruebas. El inventario nos ayuda a recortar en forma ordenada el alcance, ya que es muy probable que las funcionalidades de prioridad baja sean las primeras en quedar fuera del alcance de las pruebas. La definición del alcance de las pruebas se realiza en reuniones donde participan los representantes del cliente, el gerente de proyecto, el líder de desarrollo y el líder de pruebas. Con un primer alcance para las pruebas definido, el próximo paso es la planificación de las pruebas funcionales en el tiempo, como se describe en la próxima sección.

Inventario de Pruebas	
Funcionalidad	Prioridad
Funcionalidad 1	Media
Funcionalidad 2	Alta
Funcionalidad 3	Baja
Funcionalidad 4	Alta
Funcionalidad 5	Media
Funcionalidad 6	Baja
Funcionalidad 7	Alta
Funcionalidad 8	Media
Funcionalidad 9	Media
Funcionalidad 10	Baja

Tabla 1 – Inventario de Pruebas

3.4. Planificación de las Pruebas Funcionales

Para definir el cronograma de las pruebas a lo largo del proyecto, es necesario conocer el plan de desarrollo, donde se encuentra el cronograma de las versiones del producto que se generarán. Se requiere conocer las fechas en que desarrollo tiene planificado liberar las versiones del producto al equipo de prueba y qué funcionalidades incorpora cada nueva versión. En la Tabla 2 se muestra un ejemplo de cómo podría ser un plan de desarrollo del producto cuyo inventario es el de la Tabla 1.

Plan de Desarrollo		
Versión	Fecha	Funcionalidades que incluye el ejecutable
1	Fecha 1	1,2,3,4
2	Fecha 2	5,6,7
3	Fecha 3	8,9,10

Tabla 2 – Plan de Desarrollo

A partir de esta información el siguiente paso es estimar cuántos ciclos de prueba de la aplicación se van a realizar. Siguiendo con el ejemplo, se podría definir para el producto de la Tabla 1, que se va a realizar un ciclo de prueba por cada versión del producto y que finalmente se realizará un último ciclo de prueba previo a la instalación en producción para asegurarse que los incidentes encontrados en el último ciclo de prueba fueron solucionados, como se muestra en la Figura 1. En este caso, las fechas “Fecha 4” y “Fecha 5” deben ser acordadas con el equipo de desarrollo y el cliente. Se negocia con los desarrolladores el tiempo que necesitan para realizar las correcciones y generar la nueva versión a probar.

Una vez definidos los ciclos, las siguientes preguntas a responder, son: ¿Qué probar en cada ciclo? ¿Con qué profundidad se realizarán las pruebas de cada ciclo?

A partir de las funcionalidades del Inventario de Prueba de la Tabla 1, se refina cada componente, definiendo las funcionalidades en detalle. Se realiza nuevamente el análisis de riesgo para las nuevas funcionalidades. Puede ocurrir que un grupo de funcionalidades al que se le asignó prioridad alta, al ser dividido en varias funcionalidades, algunas de ellas sean de prioridad alta, otras de prioridad media y otras queden fuera del alcance de las pruebas. En el ejemplo de la Tabla 1, podría ocurrir que la funcionalidad 2, que tiene prioridad “Alta”, al refinarla, resulte en cuatro funcionalidades, como se muestra en la Tabla 3.



Figura 1 – Ciclos de Prueba en el Tiempo

Inventario de Pruebas		
Funcionalidad Macro	Funcionalidad en Detalle	Prioridad
Funcionalidad 1	Funcionalidad 1	Media
Funcionalidad 2		Alta
	Funcionalidad 2.1	Alta
	Funcionalidad 2.2	Baja
	Funcionalidad 2.3	Media
	Funcionalidad 2.4	Alta
Funcionalidad 3	Funcionalidad 3	Baja
Funcionalidad 4	Funcionalidad 4	Alta
Funcionalidad 5	Funcionalidad 5	Media
Funcionalidad 6	Funcionalidad 6	Baja
Funcionalidad 7	Funcionalidad 7	Alta
Funcionalidad 8	Funcionalidad 8	Media
Funcionalidad 9	Funcionalidad 9	Media
Funcionalidad 10	Funcionalidad 10	Baja

Tabla 3 – Inventario de Pruebas refinado

Con el inventario refinado, se definen las prioridades para las pruebas de cada ciclo. Del Plan de Desarrollo de la Tabla 2, surge que para el ciclo 1 sólo se contará con las funcionalidades 1, 2, 3 y 4 del producto. Del inventario de pruebas de la Tabla 3 surge que el orden de prioridad para las pruebas de dichas funcionalidades es: Funcionalidad 2.1, Funcionalidad 2.4 y Funcionalidad 4 con prioridad Alta, Funcionalidad 1 y Funcionalidad 2.3 con prioridades Media y Funcionalidad 1y Funcionalidad 2.2 con prioridad Baja. Ordenado por prioridad las funcionalidades, se obtiene para cada ciclo qué es lo más importante a probar. En la Figura 2 se muestra el orden de las funcionalidades en cada ciclo.

Además de planificar el diseño de las pruebas, se debe planificar el testing exploratorio que se realizará en cada ciclo de prueba. En general, el tiempo con que se cuenta entre un ciclo de prueba y el siguiente no es suficiente como para diseñar todos los casos de prueba de las funcionalidades de ese ciclo. Una posibilidad es complementar el diseño de las pruebas con el testing exploratorio. De esta manera, usando el análisis de riesgo del producto, se podría

planificar el diseño de las pruebas de las funcionalidades de prioridad alta en cada ciclo y que las funcionalidades de prioridad media y baja, sean exploradas en sesiones de testing exploratorio.

3.5. Planificación de cada ciclo de prueba

Al comienzo del proyecto, se planifican las pruebas a realizar del producto en cada versión que se genere del mismo. Esta planificación debe ser revisada al comenzar cada nuevo ciclo de prueba, ya que los supuestos sobre los que se hizo la planificación probablemente hayan cambiado.

Esta técnica de planificación donde a medida que los ciclos de prueba se desarrollan pueden cambiar las prioridades de las pruebas y el Inventario de Prueba puede requerir ser ajustado, debido a cambios en las prioridades del negocio o a la confianza adquirida en el producto como resultado de la realización de las pruebas en ciclos anteriores se la conoce como Técnica de Priorización Dinámica [5].

También en cada ciclo se deben planificar las pruebas de regresión. Al obtener una nueva versión donde se corrigieron incidentes, se deben ejecutar nuevamente los casos de prueba que encontraron esos incidentes. Como a priori no se conoce cuales ni cuantos serán esos incidentes, al comenzar cada ciclo, deben ser agendadas las pruebas de regresión.

4. Conclusiones

La estrategia de gestión de las pruebas funcionales utilizada en este artículo es usada en el Centro de Ensayos de Software (CES) [2].

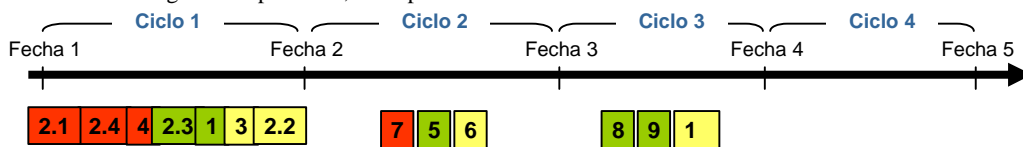


Figura 2 – Prioridad de las funcionalidades en cada ciclo

Los servicios que ofrece el CES incluyen servicios de prueba independiente, consultoría y capacitación. Se compone de dos laboratorios: el Laboratorio de Testing Funcional enfocado en la evaluación de productos desde el punto de vista funcional y el Laboratorio de Ensayos de Plataformas, donde se realizan pruebas de desempeño y se asiste a la industria para resolver problemas de funcionamiento en arquitecturas de hardware y software complejas. El CES tiene un proceso de pruebas definido para realizar sus proyectos de testing funcional independiente. En [11] se describe su uso en la prueba independiente de un producto y en [8] se describe el caso de estudio de la prueba independiente de un producto usando testing exploratorio. En ambos casos se siguió la estrategia para gestión de las pruebas descrita en este artículo, aunque el enfoque en ellos fue mostrar la metodología y técnicas de testing usadas y no la gestión de las pruebas.

A partir de la experiencia de realizar consultoría en pruebas funcionales, donde el personal del CES es contratado para ayudar a organizar el área de pruebas en una organización que se dedica al desarrollo de productos de software, donde se debe planificar las pruebas y formar un equipo propio de la organización especializado en pruebas, se puede concluir que la estrategia de gestión de las pruebas presentada en este artículo no es específica para un proyecto de prueba independiente sino que puede ser usada con éxito como parte del proyecto de desarrollo.

Como líneas de trabajo a futuro en la gestión de las pruebas funcionales, se encuentra lograr mediciones que ayuden en la estimación de las pruebas, según el tipo de producto. Para esto, se requiere tener una base de conocimiento de distintos proyectos de prueba de productos y conocer el tiempo requerido para el diseño de los casos de prueba y su ejecución por tipo de funcionalidad a probar. También resulta de interés conocer en promedio, cuánto tiempo se dedica a las pruebas de regresión en cada ciclo de prueba, esto ayudará en la estimación del tiempo requerido para las pruebas de regresión en cada ciclo.

Referencias

- [1] Myers G. "The art of software testing, 2nd edition", ISBN 0-471-46912-2, John Wiley & Sons Inc., 2004.
- [2] Centro de Ensayos de Software (CES) <http://www.ces.com.uy>, 2007
- [3] Beizer B. "Software testing techniques (2nd ed.)", ISBN:0-442-20672-0, Van Nostrand Reinhold Co, 1990.
- [4] Kaner C., Falk J., Nguyen H. "Testing Computer Software, 2nd Edition", ISBN: 0471358460, Wiley, 1999 .
- [5] Black R. "Managing the Testing Process, 2nd Edition". ISBN 0-471-22398-0, Editorial Wiley, 2002.
- [6] Bach J. "Exploratory Testing Explained", The Test Practitioner, 2002. <http://www.satisfice.com/articles/et-article.pdf>
- [7] Bach J. "What is Exploratory Testing? And How it differs from Scripted Testing" StickyMinds, Enero 2001.
- [8] Pérez B., Pittier A., Travieso M., Wodzislowski M., "Testing Exploratorio en la Práctica" VI JIISIC, Lima, Perú, 31 de Enero al 2 de Febrero de 2007. ISBN 978-9972-2885-1-7.
- [9] Bach J. "Risk-based Testing", Software Testing and Quality Engineering Magazine Vol. 1, No. 6, November-December 1999. <http://www.stqemagazine.com/featured.asp?stamp=1129125440>
- [10] Kit E. "Software Testing In The Real World : Improving The Process", ISBN 0201877562, Addison Wesley, 1995.
- [11] Pérez B., "Proceso de Testing Funcional Independiente (ProTest)", Tesis de Maestría en Informática, PEDECIBA Informática, Facultad de Ingeniería, Universidad de la Republica, Uruguay, ISSN: 0797-6410 - 06-11, 2006.