

El diseño de las pruebas en un entorno específico

The design of tests in a specific environment

Violena Hernández Aguilar¹, Tayché Capote García¹, Roig Calzadilla Díaz¹

¹Universidad de las Ciencias Informáticas

{violena, tcapote, rcalzadilla}@uci.cu

Resumen

En la Universidad de las Ciencias Informáticas (UCI) existe un Laboratorio de Certificación de Calidad de Software (LCCS), donde se realizan las pruebas de liberación de los productos de exportación que se desarrollan en la Universidad, como son las aplicaciones Desktop y Web. En este Laboratorio se trabajaba con tres plantillas diferentes para el diseño de los Casos de Prueba, por lo que surge la necesidad de integrarlas en una sola, que abarcara lo más representativo de las mencionadas y que estandarizara el trabajo de los especialistas del LCCS y los grupos de calidad de las facultades y los proyectos. El método que se propone está basado en el de particiones equivalentes propuesto por el Proceso Unificado de Desarrollo (RUP en sus siglas en inglés), al que se le han añadido algunas consideraciones, teniendo en cuenta la experiencia adquirida por los autores de este trabajo, como especialistas del LCCS al frente de los procesos de prueba de liberación. La plantilla consta de 3 pasos fundamentales y usa como artefacto de entrada las especificaciones de Casos de Uso y Requerimientos.

Palabras clave: Caja negra, casos de uso, casos de uso de prueba, no conformidades, pruebas de software.

Abstract

The Software Quality Certification Laboratory of the University of the Informatics Sciences (UIS) carries out different kind of software testing to desktop and web applications. During the testing it is used three different templates to design uses cases test, for that reason is necessary to create a unique template with the most representative of the three. In this article is proposed a Rational Unified Process (RUP) method with any changes. The proposed template has three step and it's input artifacts are Uses Cases Specification and Requirement Specification.

Key words: Black box testing, no conformities, software testing uses case, uses case test.

Introducción

En la Universidad de las Ciencias Informáticas existe un Laboratorio de Certificación de Calidad de Software (LCCS), donde se realizan las pruebas de liberación de los productos de exportación que se desarrollan en la Universidad. Todos los artefactos que constituyen entregables al cliente son revisados en el LCCS, ya sean documentos o aplicaciones tanto Desktop como Web. En el caso de estas últimas, se necesita diseñar Casos de Uso de Prueba (CUP) para realizar la revisión. En el Laboratorio existían tres tipos de plantillas diferentes para este diseño, por lo que surge la necesidad de integrarlas en una sola, que abarcara lo más representativo de las mencionadas y que estandarizara el trabajo de los especialistas del LCCS y los grupos de calidad de las facultades y los proyectos.

Desarrollo

Las Pruebas de Liberación se encargan de la Verificación. Permite determinar si los requisitos están completos y correctos. Define las actividades a realizar por el adquirente, proveedor o tercera parte independiente para verificar la conformidad de los productos

y proyectos con sus especificaciones. Las actividades de este proceso son: implementación del proceso y verificación. [ISO/IEC 12207:1995]

Para el caso de entregables aplica Proceso de Revisión. Permite evaluar el estado del software en cada etapa del ciclo de vida. Las actividades de este proceso son: implementación del proceso, revisión de la administración del proyecto y revisiones técnicas. [ISO/IEC 12207:1995]

Cuando se diseñan CUP utilizando el método de caja negra, es decir si solo se quiere verificar que la aplicación o el sistema se comporta según los requerimientos funcionales establecidos por el cliente, se necesita como artefacto de entrada el documento donde se describen los requisitos y en caso de que se utilice la metodología Proceso Unificado de Desarrollo (RUP en sus siglas en inglés) como proceso de desarrollo de software, se requiere el documento de Especificación de Casos de Uso (CU).

RUP es una metodología guiada por CU, artefacto por el que se sigue toda la traza del sistema, pues estos dicen al cliente qué esperar (a través de la descripción textual se especifica la interacción cliente-sistema), al diseñador qué clases diseñar para luego implementar, al escritor técnico de qué trata el documento, y al probador qué debe probar. Por tal razón, como cada artefacto que lo sucede es consecuencia de los CU, es necesario que se elaboren lo más simple posible, pero a la vez que aporten de forma clara la información necesaria.

Cómo diseñar Casos de Uso

Cada diseñador del negocio tiene sus reglas para realizar la descripción de los CU, aunque generalmente tienen en cuenta los puntos que aparecen a continuación:

Tabla1. Aspectos a tener en cuenta al describir un CU.

Sección del Caso de Uso	Descripción
Nombre	Un nombre apropiado para el caso de uso.
Actor que inicia	El rol que inicia el caso de uso.
Breve Descripción	Una breve descripción sobre el papel y el propósito del caso de uso.
Pre-condición	Una descripción textual que define cualquier restricción del sistema del momento de inicio del caso de uso.
Flujo de Eventos	Descripción textual de lo que el sistema hace de forma general con respecto al caso de uso (no cómo los problemas específicos son solucionados por el sistema). La descripción debe ser comprensible al cliente.
Post-Condiciones	Una descripción textual que define cualquier restricción del sistema en el momento de finalización del caso de uso.
Flujo Alterno	A partir del flujo básico, se describen las posibles alternativas que pueden surgir a partir del flujo normal

Desde el punto de vista de la información necesaria para la elaboración del CUP el aspecto más importante es el **Flujo de Eventos** el cual consta de dos partes, el **flujo básico de eventos** y los **flujos alternativos de eventos**. El flujo básico de eventos debe cubrir lo que sucede normalmente cuando se realiza el CU. Los flujos alternativos de eventos cubren el comportamiento de una opción o excepción relativa al comportamiento normal. Se puede pensar en los flujos alternativos como desvíos del flujo básico de eventos.

Por tal razón es importante a la hora de describir el flujo de eventos, tener en cuenta algunos aspectos:

1. Escribir el CU como si se narrara un juego de pelota, es decir la interacción entre dos equipos, en este caso los equipos serían: **Actor** y **Sistema**, para que se ubique al lector en cuál es la acción de cada cual, y cómo responde el sistema ante un evento que desencadena el actor.
2. Por lo anteriormente dicho, sería bueno que cada acción comience con el equipo que la realiza, o sea:

1. El actor solicita la modificación de su nombre.	2. El sistema verifica que este actor esté registrado.
--	--

3. Cuando se describe el flujo básico no se debe pensar en que existen flujos alternativos, más adelante serán descritos en la sección **Flujos alternativos**.
4. En la sección flujos alternativos se describen todas las excepciones que existan, por muy evidentes que parezcan, pues los CU son una guía para los flujos de trabajos que suceden al suyo, e incluso para los stakeholder que no tienen por qué conocer exactamente todas las posibles alternativas del proceso.
5. No se deben mezclar elementos de interfaz en la descripción, pues se está describiendo el proceso no diseñando la interfaz.
6. Es importante ubicar al lector donde comienza y termina cada flujo (básico o alternativo). La descripción del CU está dividida por secciones, en caso de que no esté dividida de esta forma se toma el CU como una sección. Cada una de ellas cuenta con un flujo básico y flujos alternos, los cuales a su vez, constituyen los escenarios de estas secciones para diseñar los CUP.

Ejemplo de descripción de CU:

Tabla 2. Descripción detallada del CU “Gestionar Existencia de Libros”.

Nombre del Caso de Uso	Gestionar Existencia de Libros
Actores	Administrador.
Propósito	Permitir registrar, eliminar y modificar datos acerca de los libros.
Resumen	Es aquí donde se registran, eliminan, y se modifican los datos de los libros que entran al almacén.
Pre-condiciones	Administrador del sistema ya autenticado.
Post-condiciones	Se registra un medio básico, se actualizan los datos o se elimina el mismo.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El administrador del sistema necesita registrar, eliminar y modificar los datos de un libro.	El sistema ejecuta alguna de las siguientes acciones: a) Si el administrador decide registrar un medio básico, ir a la sección “Registrar Libro”. b) Si el administrador decide actualizar las

	<p>características de un libro, ir a la sección "Modificar Datos".</p> <p>c) Si el administrador decide eliminar un libro, ir a la sección "Eliminar Libro".</p>
Sección “Registrar Libro”	
Acciones del Actor	Respuesta del Sistema
2. El administrador del sistema entra los datos del libro para realizar su registro en la aplicación.	<p>2.1 El sistema verifica que los campos del Código, el Nombre y la Cantidad estén llenos.</p> <p>2.2 El sistema verifica que este libro no exista.</p> <p>2.3 El libro se almacena en el sistema.</p> <p>2.4 El sistema muestra un mensaje informándosele al administrador que ya ha sido efectuado el registro del libro y finaliza el caso de uso.</p>
Curso alternativo	
	<p>2.1 El sistema emite un mensaje para que llene los campos obligatorios.</p> <p>2.3 Si el libro existe el sistema muestra un mensaje informativo y finaliza el caso de uso.</p>
Sección “Actualizar Datos”	
Acciones del Actor	Respuesta del Sistema
2. El administrador del sistema selecciona el libro a modificar.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El administrador del sistema realiza las actualizaciones deseadas.	<p>3.1 El sistema verifica que los campos obligatorios estén llenos.</p> <p>3.2 El sistema actualiza la información y finaliza el caso de uso.</p>
Curso alternativo	
	3.1 El sistema emite un mensaje para que llene los campos obligatorios.
Sección “Eliminar Libro”	
Acciones del Actor	Respuesta del Sistema
2. El administrador del sistema selecciona el libro a eliminar.	<p>2.1 El sistema pide confirmación.</p> <p>2.2 El sistema elimina el libro.</p>
Curso alternativo	
	2.1 Si no se da la confirmación el sistema no elimina el libro

Cómo diseñar Casos de Uso de Prueba

A partir del momento en que el equipo de desarrollo ha elaborado los CU del sistema se pueden elaborar los CUP, que no son más que un conjunto de entradas con datos de prueba, con sus condiciones de ejecución y los resultados esperados, con el propósito de guiar la ejecución de la prueba.

El método que se propone para el diseño de CUP, está basado en el de particiones equivalentes propuesto por RUP, al que se le han añadido algunas consideraciones, teniendo en cuenta la experiencia adquirida por los autores de este trabajo, como especialistas del LCCS al frente de los procesos de prueba de liberación. La plantilla consta de 3 pasos fundamentales y usa como artefacto de entrada las especificaciones de Casos de Uso y Requerimientos.

1. Para cada CU, se genera un sistema completo de secciones

Cada sección del CUP coincidirá con las secciones definidas en la descripción del CU. Las secciones se componen por escenarios que son su flujo básico y flujos alternos. Esta información se representa en una tabla, que se denomina **Secciones a probar en el Caso de Uso**, donde se especifica la descripción de la funcionalidad de cada escenario y en el flujo central se pone el camino en el sistema a través del cual se puede acceder a la funcionalidad en cuestión, por lo que se necesita la aplicación para poder identificar el flujo central.

Tabla 3. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Registrar libro.	EC 1.1: Registro exitoso.	Se registra el libro introduciendo los datos correctamente.	<i>Pasos a desarrollar para probar la Funcionalidad que se indicó.</i>
	EC 1.2: Faltan datos obligatorios.	No se registra el libro porque no se introducen todos los datos o se hace incorrectamente.	<i>Pasos a desarrollar para probar la Funcionalidad que se indicó.</i>
	EC 1.3: El libro existe.	No se registra el libro pues ya existe uno registrado con sus mismos datos.	<i>Pasos a desarrollar para probar la Funcionalidad que se indicó.</i>

En este punto es válido señalar que en el caso de los flujos alternos, que generalmente representan excepciones que ocurren, se deben construir varios escenarios de pruebas que validen todas las alternativas:

- Si un dato de entrada se encuentra en un determinado rango se debe elaborar un escenario que verifique el comportamiento del sistema cuando el dato toma un valor dentro del rango (flujo básico), y dos escenarios más, uno que valide el comportamiento del sistema ante un valor menor al intervalo y otro que valide el comportamiento ante un valor mayor a dicho intervalo.
- Si existen datos de entrada que cumplen una determinada regla del negocio (ejemplo: la edad de una persona para sacar el carnet de identidad no puede ser menor a 16 años), es necesario además de los CU, entregar al equipo de prueba el documento donde se describen las reglas del negocio, o debe añadirse otro punto al documento Especificación de Casos de Uso, que podría llamarse “Requisitos especiales”, donde se expliquen las condiciones que deben cumplir los datos.

2. Elaborar la Matriz de Casos de Prueba

Luego de identificados los diferentes escenarios de prueba, se identifican las variables (la información necesaria para que el escenario se ejecute, que hace que dicho escenario se comporte de una forma o de otra, es decir, como un flujo básico o alternativo) y esta información se representa en una matriz, llamada Matriz de Casos de Prueba, donde se representa además el comportamiento esperado del sistema.

Tabla 4. Sección a revisar.

Id del escenario	Escenario	Código	Nombre	Cantidad	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Registro exitoso.	V	V	V	El sistema muestra un mensaje informándosele al administrador que ya ha sido efectuado el	<i>Se escribe el resultado que se obtiene al realizar la prueba.</i>
		V	V	V		
		V	V	V		
EC 1.2	Faltan datos obligatorios.	I-NA	I-NA	I-NA	El sistema emite un mensaje para que llene los campos obligatorios y debe reconocer la	Se escribe el resultado que se obtiene al realizar la prueba.
		I-NA	V	V		
		V	I-NA	V		
		V	V	I-NA		
EC 1.3	El libro existe.	V	V	V	El sistema muestra un mensaje informativo.	Se escribe el resultado que se obtiene al realizar la prueba.
		V	V	V		
		V	V	V		

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

3. Identificar Valores de Datos y realizar las Pruebas

Una vez que todos los CUP se han diseñado, se procede a aplicar la prueba, para lo cual es necesario sustituir los datos válidos e inválidos por juegos de datos reales. Después se prueban estos datos en la aplicación, y se verifica si se cumplieron los requisitos establecidos en los CU. En la siguiente tabla se muestra solo una parte de la anterior, como ejemplo de algunos campos llenos.

Tabla 5. Ejemplo de los juegos de datos a utilizar para las pruebas.

Id del escenario	Escenario	Código	Nombre	Cantidad	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Registro exitoso.	BN2345	El código Davinci	20	El sistema muestra un mensaje informándosele al administrador que ya ha sido	<i>Se escribe el resultado que se obtiene al realizar la prueba.</i>
EC 1.2	Faltan datos obligatorios.	BM5698	Corazón	gk - NA	El sistema emite un mensaje para que llene los campos obligatorios y debe reconocer la	Se escribe el resultado que se obtiene al realizar la prueba.
EC 1.3	El libro existe.	BN2345	El código Davinci	20	El sistema muestra un mensaje	Se escribe el resultado que se obtiene al realizar

4. Identificar las No Conformidades

Es necesario definir el concepto de No Conformidad (NC), que no es más que los problemas detectados en un artefacto según:

- Error con respecto a lo definido en artefactos anteriores y/o en lo pactado con el cliente.
- No concordancia con normas internacionales que deben ser cumplidas por el artefacto.
- Insatisfacción del cliente con el resultado final de un Elemento de Configuración según lo pactado con anterioridad en el proyecto.

La resolución de una NC siempre genera una Orden de Trabajo para el proyecto. Además debe definirse el nivel de importancia de la misma, por lo que se clasifican en Significativa y No Significativa.

Para revisión de documentación

Categoría	Nivel de Importancia	Descripción	Ejemplo
1	Significativa	Efectos de crítico alcance en el documento.	<p>Errores de redacción y ortografía.</p> <p>En el caso del documento de casos de uso el flujo básico y/o flujo alternativo no está bien descrito (no se entiende el proceso).</p> <p>No se representan todos los elementos en los diagramas de clases.</p> <p>No se usan los prototipos adecuados en el diagrama de clases de diseño en dependencia de la plataforma de desarrollo.</p> <p>No hay concordancia entre el diagrama de clases y el modelo de la BD.</p> <p>No se tienen en cuenta todos los nodos en el modelo de despliegue.</p> <p>No se representan todos los elementos en los diagramas de componentes.</p>
2	No significativa	Efectos no significativos en el documento.	Otros aspectos no mencionados en los ejemplos de las NC Significativas.

Para revisión de las funcionalidades de la aplicación

Categoría	Nivel de Importancia	Descripción	Ejemplo
1	Significativa	El sistema se detiene y no puede avanzar. Errores de validación.	Genera error que no permite llegar al final de un flujo. En un campo donde es necesario entrar números el sistema permita entrar caracteres.
2	No Significativa	Efectos no significativos en la funcionalidad y usabilidad del sistema.	En el caso de los portales palabras escritas en inglés cuando el sistema está descrito en español.

Además se definen dentro de estas clasificaciones, tipos de NC según el artefacto en el que se detecten:

Id	Tipo de NC	Tipo de Artefacto
1	Ortografía	Aplicación y Documentación
2	Implementación ≠ Documentado	Aplicación y Documentación
3	Redacción	Documentación
4	Formato	Documentación
5	Errores Técnicos	Documentación
6	No implementado	Aplicación
7	Validación	Aplicación
8	No se cumple objetivo de flujo básico	Aplicación
9	Seguridad	Aplicación
10	Diseño Gráfico	Aplicación

Cuando se detectan las NC se les asigna un valor, teniendo en cuenta el estado en el que están. La primera vez que se detecte estará Pendiente. En la siguiente tabla se explica cada uno de los valores que puede tener la NC.

Estado de la No Conformidad

Valor	Significado	Descripción
PD	Pendiente por el equipo de desarrollo.	Esperando por la respuesta del equipo de desarrollo, siempre que se encuentre una NC ese es el estado por defecto.
RA	Resuelta y aprobada.	El equipo de desarrollo le dio solución y se revisó por el equipo de pruebas que efectivamente fue así.
NP	No procede.	El equipo de desarrollo determina que la NC no es válida y explica las razones.

Conclusiones

- Un buen diseño de un CUP de caja negra depende de la información que brinde el CU.
- Apenas se cuente con la Especificación de un CU y/o la Especificación de Requisitos, se puede comenzar a diseñar los CUP que lo validan.
- La realización de las pruebas basadas en CU es un proceso que consta de tres pasos fundamentales: diseño, implementación del método y análisis de los resultados y puede tener varias iteraciones.

Referencias Bibliográficas

1. Ayuda en línea de la Suite del Rational 2003.
2. [Cockburn: 2000] Cockburn, Alistair, Writing effective use cases, 2000, Addison Wesley.
3. [ISO/IEC 12207:1995] Norma Internacional, Ciclo de Vida del Software.
4. [Pressman: 1995] Pressman, Roger S., Ingeniería de Software un enfoque práctico, 1995, 5^{ta} Edición, McGraw Hill.