Prototipo de editor de casos de uso para la construcción asistida de casos de prueba

Trabajo Terminal No. 2014-B064

*Alumnos: Hernández Sánchez Natalia Giselle, Ramírez Camacho Sergio**
Directores: López Rabadán José Jaime, Maldonado Castillo Idalia
e-mail: sramirezc1101@alumno.ipn.mx

Resumen - Se propone construir un prototipo de editor de casos de uso que permita generar el documento de análisis de un sistema de información con base en una estructura definida, esta estructura será reutilizada por el prototipo para asistir en la generación de casos de prueba que podrán ser ejecutados manualmente con una herramienta externa. El construir una herramienta con estas características podría reducir el tiempo y costo de la realización de un sistema.

Palabras clave - Documentación de Sistemas, Ingeniería de Pruebas, Ingeniería de Software, Proceso de Software.

1. Introducción

Las pruebas funcionales de un sistema de información requieren del estudio exhaustivo de la documentación de análisis de casos de uso, que describe el comportamiento del sistema, así como todas sus validaciones. La ejecución de pruebas funcionales requiere del diseño de casos de prueba, la definición de las bases de datos de prueba, los datos de entrada de prueba y la interpretación de los resultados generados. Derivado de lo anterior el tiempo y costo requeridos para probar un sistema son muy altos [1], [2].

Actualmente existen herramientas que requieren del diseño manual de los casos de prueba, de la participación de un analista que conozca el negocio y de un experto que realice la configuración adecuada. El número de casos de prueba a diseñar es directamente proporcional al número de validaciones y combinaciones que considere el caso de uso. Se propone elaborar un prototipo de editor de casos de uso que permita obtener los elementos para generar la documentación de análisis de un sistema, esta información será reutilizada por el prototipo para asistir a la construcción de casos de prueba con el objetivo de contribuir al *proceso de software*.

Para comprender el funcionamiento del prototipo del editor de casos de uso es necesario conocer la definición de un proceso de software, así como las etapas que comprende y los costos implicados en cada una de ellas.

Proceso de software

Un *proceso de software* es un conjunto coherente de actividades que se llevan a cabo para la producción de un software. Los procesos de software son complejos y los intentos por automatizarlos han tenido un éxito limitado debido a su inmensa diversidad. Aunque existen muchos procesos diferentes, algunas etapas son comunes entre ellos [1], [3]:

- **Análisis**. Consiste en especificar qué hará el software.
- **Diseño**. Consiste en especificar cómo debe implementarse el software, con base en el análisis realizado.
- **Implementación**. Consiste en codificar el software.
- **Pruebas**. Consiste en verificar[1] y validar[2] que el software se ajusta a su especificación.

Costos del proceso de software

Si se considera que el costo total del desarrollo de un sistema de software es de 100 unidades de costo, la Figura 1 muestra cómo se gastan estas en las diferentes actividades del proceso [1].

15	25	20	40
Análisis	Diseño	Implementación	Pruebas

Figura 1 Distribución de costos de las actividades del proceso del software

[1] Verificar. Se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica (¿Construimos el producto correctamente?) [3].

[2] Validar. Se refiere al conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente (¿Construimos el producto correcto?) [3].

La etapa de *Pruebas* representa un costo muy elevado en relación al invertido en las demás etapas, normalmente el costo de realizar pruebas es alrededor del 40 % del valor total del proyecto, esto se debe en parte a que existe un número muy grande de casos de prueba y estos son diseñados y ejecutados manualmente [1], [2]. Para diseñar casos de prueba es necesario un análisis para determinar cuáles son los comportamientos que se desean comprobar. Para cualquier sistema existe un número muy grande de casos de prueba, por lo que es imprescindible hacer una selección adecuada de ellos, Fewster y Graham describen 4 atributos fundamentales que nos permiten conocer la calidad de un caso de prueba, el primero es la eficacia al detectar errores, el segundo se refiere a poder aplicarlo a más de una circunstancia, el tercero sugiere que el costo económico de diseñarlo y ejecutarlo sea bajo y finalmente, el cuarto se refiere a que el mantenimiento no debe requerir mucho esfuerzo frente a un cambio en el software [2].

Etapa de Análisis

La etapa de *Análisis* consiste en identificar las necesidades del cliente y la funcionalidad que tendrá el sistema con base en los procesos del negocio para describir el comportamiento que tendrá el mismo, además de definir la información que será almacenada en una estructura de datos.

Las necesidades del cliente se transforman en requerimientos funcionales y no funcionales, estos describen el comportamiento del sistema, parte de este comportamiento es modelado mediante casos de uso. El producto de la etapa de *Análisis* es un documento donde se incluye la definición de los casos de uso, la cual requiere la descripción de entidades, especificación de los atributos, reglas de negocio, interfaces, mensajes y perfiles de usuario que intervienen en el sistema.

La etapa de Análisis es muy importante ya que en ella se define el comportamiento del sistema y los errores u omisiones generados durante esta etapa se propagarán a todas las demás etapas del proceso de construcción del software [1], [4].

Etapa de Pruebas

La etapa de Pruebas consiste en verificar y validar que el software se ajusta a los requerimientos funcionales y no funcionales del sistema.

Una parte de la etapa de *Pruebas* consiste en la validación de la funcionalidad del software, existen muchos tipos de pruebas y cada uno de ellos tiene un propósito específico, sin embargo en este trabajo se abordarán únicamente las pruebas funcionales, las cuales se enfocan en comprobar que el software desarrollado funcione de acuerdo a lo descrito en los casos de uso. Este tipo de pruebas pertenecen al conjunto de pruebas de caja negra, las cuales validan que a partir de los datos de entrada se obtengan las salidas esperadas, para realizar estas pruebas se utilizan casos de prueba [5], [6], [7].

2. Objetivo

Objetivo general

Desarrollar un prototipo que permita la edición de casos de uso y la generación semiautomática de casos de prueba funcionales de un sistema de información que opere a través de la web, con la finalidad de asistir en las etapas de *Análisis* y *Pruebas* de software.

Objetivos específicos

- Desarrollar un módulo para la captura de información que permita registrar entidades, especificación de los atributos, reglas de negocio, interfaces, mensajes, perfiles de usuario y trayectorias de casos de uso.
- Generar el documento de análisis de un sistema, considerando el paradigma orientado a objetos con base en la información obtenida en el módulo de captura.
- Generar propuestas de casos de prueba funcionales de catálogos simples en un formato estándar para su uso con una herramienta de ejecución de pruebas con base en la información recolectada durante el análisis.

3. Justificación

Problema

En la etapa de *Análisis* es complejo construir el documento de análisis debido a que en este se define con qué información trabajará el sistema y la estructura donde se va a organizar esta, así como describir cada una de las relaciones de los elementos definidos. En este documento también se busca detallar todos los posibles comportamientos que tendrá el sistema, lo que se vuelve complicado debido a la cantidad de validaciones y a todas sus combinaciones.

Las herramientas que existen actualmente para generar un documento de análisis están limitadas debido a que no permiten realizar las siguientes tareas en conjunto:

- Organizar y relacionar las entidades de un sistema con los casos de uso.
- Proveer un mecanismo para organizar y relacionar las reglas de negocio utilizadas en la documentación de casos de uso.
- Proveer un mecanismo para organizar y relacionar los mensajes utilizados en la documentación de casos de uso.
- Entrelazar los casos de uso con las reglas de negocio, mensajes y entidades.
- Documentar el comportamiento que tendrá el sistema a través de las trayectorias principales y alternativas.

Dentro de la etapa de *Pruebas* es muy costoso en tiempo y recursos desarrollar las pruebas funcionales de un sistema debido a la cantidad de escenarios y validaciones que se deben de probar. Además las personas responsables de realizar las pruebas funcionales de un sistema requieren forzosamente conocer el negocio, validaciones y comportamientos del sistema a través del documento de análisis o del estudio del sistema implementado, y no existen mecanismos automatizados para utilizar el conocimiento vertido en el documento de análisis para la creación de pruebas funcionales. Algunas de las desventajas de las herramientas de pruebas que existen se deben a que:

- No reutilizan la información de los datos de entrada que se definieron en el documento de análisis.
- No reutilizan la información de las salidas esperadas que se describieron en el documento de análisis.
- No permiten reutilizar todas las validaciones descritas en las trayectorias de los casos de uso.
- No proveen un mecanismo que transforme las trayectorias de los casos de uso en casos de prueba.

Propuesta

Se propone construir un prototipo de editor de casos de uso que permita organizar entidades, especificación de los atributos, reglas de negocio, interfaces, mensajes y perfiles de usuario, además de relacionarlos con los casos de uso.

El prototipo generará casos de prueba a partir de la información recabada, estos casos de prueba modelarán escenarios de prueba y validarán el comportamiento del sistema.

Actualmente no existe una herramienta que asista en la generación de casos de prueba que utilice el conocimiento vertido en el documento de análisis. El realizar una herramienta con estas características facilitará la documentación y pruebas de un sistema y con ello posiblemente reducir el tiempo y costo implicados en el proceso de software.

La construcción de este prototipo requiere de la aplicación de los conocimientos adquiridos en la carrera de Ingeniería en Sistemas Computacionales, las aptitudes adquiridas en las unidades de aprendizaje de Algoritmia y Programación Estructurada, Estructura de Datos, Teoría Computacional, Bases de Datos, Programación Orientada a Objetos, Análisis y Diseño Orientado a Objetos, Ingeniería de Software, Compiladores, entre otras.

4. Productos o resultados esperados

Al finalizar este proyecto los productos o resultados esperados son:

1. **Software**. Prototipo que permita la edición de casos de uso y la generación semiautomática de casos de prueba funcionales de un sistema de información. El software tendrá los siguientes módulos:

   - *Editor de casos de uso.* Módulo que permitirá registrar entidades, especificación de los atributos, reglas de negocio, interfaces, mensajes, perfiles de usuario y trayectorias principales y alternativas que describan los casos de uso, para posteriormente guardarlos de manera organizada en una estructura de información.

   - *Generador de pruebas.* Módulo que asistirá en la generación de casos de prueba a partir de la estructura de información generada por el *Editor de casos de uso*. Este módulo únicamente generará los casos de pruebas funcionales, sin considerar su ejecución.

El prototipo de editor de casos de uso será construido para funcionar con un caso de estudio propuesto que posea las siguientes características:

- Sistema que opere a través de la web.
- Sistema descrito mediante casos de uso.
- Sistema de catálogos simples que permitan las operaciones de altas, bajas, cambios y consultas.
- Sistema de formularios simples que utilicen elementos como campos de texto, campos de fecha, áreas de texto, radiobotones, casillas de verificación, listas desplegables, selector de archivos y botones.

- Sistema que no utilice frameworks visuales.
- Sistema que no haga peticiones asíncronas.

Este prototipo generará casos de prueba que validen lo siguiente:

- Reglas de negocio referentes a la existencia de la información en la base de datos.
- Reglas de negocio de operaciones aritméticas.
- Reglas de negocio relacionadas con la unicidad de elementos en la base de datos, dado un atributo.
- Reglas de negocio que describan datos obligatorios, longitud y tipos de dato.
- Perfiles de usuario mediante el inicio de sesión.
- Coherencia de mensajes entre el sistema y el documento de análisis generado por el prototipo.

2. **Documentación de análisis**. Documento que describa el desarrollo de la etapa de *Análisis* del prototipo de editor de casos de uso.

3. **Manual de usuario**. Documento que sirva de guía al usuario para utilizar el prototipo de editor de casos de uso.

5. Metodología

Para el desarrollo del prototipo se utilizará un modelo de proceso incremental, se planea realizar el proyecto en dos incrementos:

1. Editor de casos de uso
2. Generador de pruebas

Cada uno de los incrementos incluye las etapas de *Análisis*, *Diseño*, *Implementación* y *Pruebas*, al inicio de cada incremento se proponen reuniones de trabajo para detallar y corregir los requerimientos del sistema.

A lo largo de cada incremento se propone una serie de entregables y revisiones periódicas con la finalidad de corregir a tiempo los errores que puedan surgir.

6. Cronograma

Nombre de la alumna: Hernández Sánchez Natalia Giselle TT No.: 2014-B064
Título del TT: Prototipo de editor de casos de uso para la construcción asistida de casos de prueba

	Tareas a realizar	Actividad	2015								
			mar	abr	may	jun	jul	ago	sep	oct	nov
TT I - Editor de casos de uso	• Editor de atributos • Editor de entidades • Editor de reglas de negocio • Generador de la estructura de información • Generador del documento de análisis.	Requerimientos									
		Análisis									
		Diseño									
		Implementación									
		Pruebas									
		Elaborar reporte técnico									
		Presentación									
TT II - Generador de pruebas	• Generador de casos de prueba	Requerimientos									
		Análisis									
		Diseño									
		Implementación									
		Pruebas									
		Elaborar reporte técnico									
		Presentación									

Nombre del alumno: Ramírez Camacho Sergio TT No.: 2014-B064
Título del TT: Prototipo de editor de casos de uso para la construcción asistida de casos de prueba

	Tareas a realizar	Actividad	2015								
			mar	abr	may	jun	jul	ago	sep	oct	nov
TT I - Editor de casos de uso	• Editor de perfiles de usuario • Editor de mensajes, editor de interfaces • Editor de casos de uso • Generador de la estructura de información • Generador del documento de análisis.	Requerimientos									
		Análisis									
		Diseño									
		Implementación									
		Pruebas									
		Elaborar reporte técnico									
		Presentación									
TT II - Generador de pruebas	• Generador de casos de prueba	Requerimientos									
		Análisis									
		Diseño									
		Implementación									
		Pruebas									
		Elaborar reporte técnico									
		Presentación									

7. Referencias

[1] I. Sommerville, *Ingeniería del software*, 7a ed. Madrid: Pearson, 2005.

[2] M. Fewster D. Graham, *Software Test Automation: effective use of test execution tools*, New York: Addison-Wesley, 1999.

[3] R. S. Pressman, Ingeniería del software: un enfoque práctico, 7a ed. México, D.F.: Mc Graw Hill, 2010.

[4] K. E. Kendall, J. E. Kendall, *Análisis y diseño de sistemas*, 6a ed. Edo. de México: Prentice Hall, 2005.

[5] P. C. Jorgensen, *Software testing: a craftsman's approach,* 2nd ed. United States of America: CRC Press, 2002.

[6] R. V. Binder, "*Testing: a brief introduction" en Testing Object-oriented Systems: Models, Patterns and tools*, Canada: Addison-Wesley, 2004, pp. 11-41.

[7] C. Kaner, J. Falk, H. Q. Nguyen, *Testing computer software*, United States of America: Dreamtech Press, 2000.

8. Alumnos y directores

Natalia Giselle Hernández Sánchez.- Alumna de la carrera de Ing. en Sistemas Computacionales en ESCOM, Especialidad Sistemas, Boleta: 2012630214, Tel.: 55 43 67 65 94, email: hdeznatali@gmail.com.

Firma:_____

Sergio Ramírez Camacho.- Alumno de la carrera de Ing. en Sistemas Computacionales en ESCOM, Especialidad Sistemas, Boleta: 2012630361, Tel.: 55 30 40 14 39, email: sramirezc1101@alumno.ipn.mx.

Firma:_____

M. en C. José Jaime López Rabadán.- Profesor de la ESCOM, egresado de la Universidad Autónoma Metropolitana, Maestría en Ciencias de la Computación del CINVESTAV. Áreas de interés: Sistemas de Información e Ingeniería de Software. Email: rabadanlorj@gmail.com.

Firma:_____

M. en C. Maldonado Castillo Idalia.- Profesora de la ESCOM, egresada de la Ing. En Sistemas Computacionales de la Escuela Superior de Cómputo del IPN, Maestría en Ciencias de la Computación de la University of Saskatchewan, Canadá. Áreas de interés: Sistemas de Información, Clasificación y Procesamiento de Imágenes, Ingeniería de Software. Email: idaliamaldonado@gmail.com.

Firma:_____