

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

Natalia Giselle Hernández Sánchez, Sergio Ramírez Camacho

Escuela Superior de Cómputo I.P.N México Ciudad de México

Tel. 57-29-6000 ext. 52000 y 52021. E-mail: hdeznatali@gmail.com, sramirez@live.com

Resumen - El Prototipo de editor de casos de uso para la construcción asistida de casos de prueba, llamado "PRISMA", es una herramienta que permite documentar casos de uso mediante el registro de entidades, atributos, reglas de negocio, interfaces, mensajes, actores y trayectorias. La herramienta permite generar casos de prueba funcionales con base en la documentación realizada, todo esto con la finalidad de disminuir el tiempo empleado en la etapa de pruebas de un sistema.

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

I. INTRODUCCIÓN

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], [2]: Análisis, Diseño, Implementación y Pruebas.

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	Análisis
25	Diseño
20	Implementación
40	Pruebas

Figura 1: Distribución de costos

Actualmente existen herramientas que permiten automatizar diferentes actividades del proceso de software, esto permite algunas mejoras en la calidad y productividad del software. El presente artículo expone las actividades realizadas y los resultados obtenidos de desarrollar una herramienta que permite asistir a la etapa de *Pruebas* a través de la generación semiautomática de casos de prueba funcionales basados en la documentación de análisis.

II. METODOLOGÍA

La herramienta PRISMA se dividió en dos incrementos, el primero es el Editor de casos de uso y el segundo es el Generador de casos de prueba. A continuación se describen las principales actividades realizadas para cada uno de ellos.

Incremento 1. Editor de casos de uso

Las actividades que se realizaron en el incremento 1 fueron las siguientes:

- I. Análisis de diferentes documentos de análisis
- II. Análisis de la información necesaria para crear una prueba automatizada

- III. Construcción de una prueba automatizada como base para la experimentación
- IV. Generación de una prueba automatizada con base en una fuente de información externa
- V. Diseño de la base de datos
- VI. Diseño de mecanismos para relacionar elementos

I. Análisis de diferentes documentos de análisis

Dado que las pruebas funcionales de un sistema pueden basarse en la especificación de casos de uso, se realizó un estudio de la documentación de casos de uso de diferentes sistemas, con el objetivo de conocer cómo se estructuraban y qué información requerían. El resultado de esta actividad permitió establecer la información con la que PRISMA operaría:

- Reglas de negocio
- Mensajes
- Actores
- Términos del glosario
- Entidades
- Pantallas

II. Análisis de la información necesaria para crear una prueba automatizada

Posteriormente se estudió la estructura y la información necesaria para construir un caso de prueba funcional, esperando que los resultados obtenidos presentaran una estrecha relación con los resultados del análisis de la estructura e información necesaria para un caso de uso.

Los resultados obtenidos fueron positivos, ya que gran parte de la información requerida para construir un caso de prueba funcional, coincidía con la información necesaria para la construcción de un caso de uso.

III. Construcción de una prueba automatizada como base para la experimentación

Con los resultados obtenidos anteriormente, se intentó realizar una prueba automatizada,

utilizando exclusivamente la información proporcionada por la especificación de casos de uso, el resultado de este ejercicio mostró que aunque la información que proporciona este documento es bastante útil, no es suficiente para la construcción de la prueba.

IV. Generación de una prueba automatizada con base en una fuente de información externa

Debido a que la prueba automatizada creada anteriormente fue construida directamente sobre una herramienta de pruebas, se realizó un ensayo para demostrar que era posible crearla de forma semiautomática desde una herramienta externa. Para ello se implementó un sistema simple, en el cual se solicitaba información referente a la prueba, para posteriormente utilizarla y generar la prueba automatizada. Los resultados fueron satisfactorios, se demostró que era posible generar pruebas desde una herramienta externa para una posible ejecución con alguna herramienta de pruebas.

V. Diseño de la base de datos

Con el análisis realizado anteriormente, se propuso como primer incremento la construcción de un editor de casos de uso que permitiera gestionar diferentes elementos contenidos en la documentación de análisis de un sistema. Con ello se diseñó una base de datos que permitiera alojar la información y que además modelara la gran cantidad de relaciones que tiene un caso de uso con diferentes elementos del documento análisis.

VI. Diseño de mecanismos para relacionar elementos

Como se mencionó anteriormente, un caso de uso se encuentra interrelacionado con una gran cantidad de elementos incluidos en el documento de análisis, por lo que resultó importante establecer mecanismos que elevaran la usabilidad al sistema. En general, este mecanismo consiste en poder seleccionar elementos previamente registrados en medio de

una redacción abierta. Es necesario utilizar una estructura definida para hacer una referencia, en la figura 2 se muestra un ejemplo de cómo hacer una referencia a una pantalla dentro de un paso.

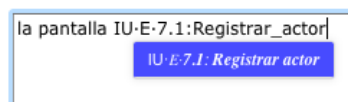


Figura 2: Referencia a un caso de uso

Incremento 2. Generador de casos de prueba

Las actividades que se realizaron en el incremento 2 fueron las siguientes:

- I. Análisis de las trayectorias a probar
- II. Creación de una prueba automatizada para una trayectoria
- III. Análisis de las equivalencias entre pasos de una trayectoria y componentes de una prueba
- IV. Diseño de mecanismos para la recolección de información faltante para los casos de prueba
- V. Diseño de un algoritmo para convertir pasos de un caso de uso a componentes de prueba

I. Análisis de las trayectorias a probar

Dado que el comportamiento de un caso de uso puede modelar un conjunto muy amplio de escenarios, se realizó un análisis de diferentes trayectorias, buscando comportamientos en común y altamente repetitivos.

El resultado de este análisis permitió definir un modelo general de trayectorias candidatas para la creación de casos de prueba funcionales. Posteriormente se analizaron nuevamente diferentes trayectorias, pero esta vez basando la búsqueda en el modelo obtenido del análisis anterior, esto con la finalidad de identificar claramente qué casos de uso realmente eran candidatos y con ello definir un vector de pruebas.

Con los resultados obtenidos se generó un diagrama que modela aquellas trayectorias que son candidatas, el diagrama se muestra en el **Anexo 1**.

II. Creación de una prueba automatizada para una trayectoria

Con un espacio de búsqueda reducido, se escogió la trayectoria de un caso de uso para utilizarlo como base para la generación de casos de prueba muestra. Se buscó que el caso de uso presentara la mayor cantidad de escenarios para así observar los componentes de prueba necesarios para cada uno de ellos. Este ejercicio reforzó el análisis realizado durante el primer incremento para determinar la información necesaria para construir la prueba automatizada y permitió definir claramente qué información no aportaba el editor para la construcción de la prueba.

III. Análisis de las equivalencias entre pasos de un caso de uso y componentes de prueba

Con la prueba creada, el siguiente paso consistió en analizar a detalle la relación existente entre los pasos de una trayectoria y los componentes de prueba. Para ello se estudió cada paso y se establecieron diferentes categorías, a las cuales a su vez se les asoció un conjunto de componentes de prueba, por ejemplo, para un paso que consiste en oprimir un botón, se le asoció una petición HTTP.

IV. Diseño de mecanismos para la recolección de información faltante para los casos de prueba

La siguiente tarea consistió en establecer la manera en que la información faltante sería solicitada al usuario, entre los datos que el editor de casos de uso no aportaba se encuentran por ejemplo los nombres de los input de los formularios.

Por otro lado, para el caso de los datos de entrada, se diseñaron mecanismos para su generación automática, con base en la especificación registrada en el editor de casos de uso.

Los mecanismos utilizados para solicitar información no resultaron triviales debido a que

gran parte de la información depende de la especificación del caso de uso, es por ello que su comportamiento es completamente dinámico.

V. Diseño de un algoritmo para convertir pasos de una trayectoria a componentes de prueba

Finalmente, con el análisis de las equivalencias entre pasos y componentes de prueba, se prosiguió a diseñar un algoritmo que permitiera transformar los pasos de una trayectoria a casos de prueba funcionales.

Inicialmente se propuso realizar una traducción lineal de los pasos, es decir, encontrar un paso y en ese mismo momento transformarlo a componentes de prueba, sin embargo con esa idea no resultó posible generar la prueba de forma automática pues la transformación dependía del contexto. Debido a esto, se implementó un algoritmo que permitiera realizar esta conversión.

III. RESULTADOS

A continuación se muestran los resultados más relevantes que se obtuvieron a lo largo del desarrollo del proyecto.

Prototipo

El desarrollo del proyecto permitió la construcción de PRISMA, un prototipo de editor de casos de uso que permite construir casos de prueba con base en la información de análisis. Algunas de las características de la herramienta son:

- Permite recabar los elementos que componen un documento de análisis: mensajes, reglas de negocio, actores, términos del glosario, pantallas, casos de uso y entidades.
- Permite indicar las relaciones entre los diferentes elementos mencionados en el punto anterior.
- Permite generar el documento de análisis en dos formatos distintos: pdf y docx.

- Permite modelar el proceso de revisión de los casos de uso entre colaboradores de un proyecto.
- Permite asistir a la generación semiautomática de casos de prueba con base en el documento de análisis.

Gestión de las actividades de los colaboradores

El diseño de la herramienta soporta tres perfiles de usuario diferentes:

- Administrador
- Líder de análisis
- Analista

Estos tres perfiles permiten que existan usuarios con diferentes responsabilidades y niveles de acceso. El Administrador es el encargado de gestionar los proyectos, así como el personal de la organización. El Líder de análisis y el Analista participan en el proceso de revisión de los casos de uso siendo cada uno colaborador de los proyectos.

Fórmula para el cálculo de los casos de prueba generados por el sistema

Un resultado interesante fue la fórmula que permite calcular el número de casos de prueba que pueden ser generados por PRISMA, el resultado del cálculo como se observa en la figura 3 depende de las reglas de negocio asociadas al caso de uso a probar, así como el tipo y la cantidad de entradas asociadas.

$$NCP = I + RNVC + RNDI (B + E + FL + FE) + RNLI (C + E + FL) + RNDO \cdot DO$$

Donde:

- NCP = Número de casos de prueba
- RNVC = Número de reglas de negocio de verificación de catálogos
- RNDI = Número de reglas de negocio de datos incorrectos
- B = Número de entradas de tipo booleano
- E = Número de entradas de tipo entero
- FL = Número de entradas de tipo flotante
- FE = Número de entradas de tipo fecha
- RNLI = Número de reglas de negocio de longitud inválida
- C = Número de entradas de tipo cadena
- RNDO = Número de reglas de negocio de datos obligatorios
- DO = Número de entradas obligatorias

Figura 3: Fórmula para calcular los casos de prueba generados

IV. CONCLUSIONES

Las pruebas de software con frecuencia representan un costo muy elevado en relación al invertido en las demás actividades del proceso de software. Debido a ello, se han realizado diversos intentos por automatizar su proceso, como lo son herramientas de software o metodologías de pruebas.

El desarrollo de la herramienta demostró que es posible automatizar las pruebas de software a través de mecanismos semiautomáticos que reutilizan la información de la documentación de análisis.

Debido a que la especificación de casos de uso es realizada por diferentes analistas, con frecuencia se encuentran diferentes redacciones para un mismo objetivo, lo cual complica bastante el entendimiento de la información para una computadora, por lo que analizar los pasos de un caso de uso y traducirlos a compo-

nentes de pruebas resulta altamente complejo.

Por otro lado, se identificó que tener la información de la documentación de análisis en una base de datos permite mantener la información organizada y centralizada. En general el contar con una base de datos de esta información abre las puertas a otros sistemas para automatizar algún proceso dentro del desarrollo de un software, en nuestro caso, por ejemplo, permitió automatizar la generación de casos de prueba funcionales.

REFERENCIAS

- [1] I. Sommerville, *Software Engineering*, ser. International Computer Science Series. Addison-Wesley, 1992.
- [2] R. Pressman, *Software Engineering: A Practitioner's Approach*, ser. McGraw-Hill series in computer science. Boston, 2005.

ANEXOS

Anexo I. Modelo de trayectorias

La figura 4 muestra un diagrama que modela los pasos más comunes de una trayectoria. Las transiciones con líneas punteadas son aquellas que no considera la herramienta.

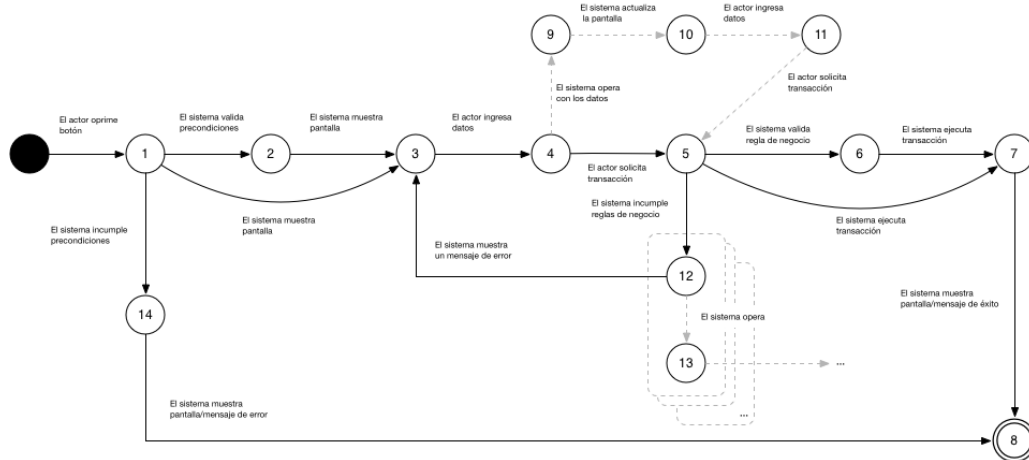


Figura 4: Modelo de trayectorias