

MODELO DE CASOS DE USO - UN EJE PARA EL PROCESO DE DESARROLLO DE SOFTWARE

María Inés Lund, Laura Aballay, Cintia Ferrarini, María Romagnano
Instituto de Informática – Fac. Cs. Exactas F. y Naturales – Universidad Nacional de San Juan

CONTEXTO

El proyecto se encuentra inserto dentro del Gabinete de Ingeniería de Software del Instituto de Informática y está relacionado con las cátedras Diseño de Software y Sistemas de Información II de las carreras de informática de la FCEF N de la Universidad Nacional de San Juan.

RESUMEN

El propósito de este proyecto es efectuar una recopilación, estudio y análisis de los usos existentes de los casos de uso, de autores reconocidos en el ambiente internacional de la Ingeniería de Software y del desarrollo orientado a objetos. Se analizarán los usos o extensiones de uso que se les dan a los modelos de casos de uso, para lograr una descripción completa y bien definida (formalización) de cada uno de ellos, permitiendo que los modelos de casos de uso sean un eje que acompañe y ayude a todo el proceso de desarrollo de software.

Un modelo de caso de uso es una notación que permite especificar y documentar requisitos, planificar el proyecto de software, tales como fechas de entregas, tamaño del software, también ayudan al equipo de diseño, al diseño de interfaz de usuario y de pruebas del sistema.

(No más de 200 palabras exponiendo sintéticamente los alcances de la línea de I/D)

Palabras clave: casos de uso, análisis, modelos de casos de uso, requisitos, proceso de desarrollo de software.

1. INTRODUCCION

Los proyectos de desarrollo de software suelen fallar, por entregas tardías, mayor presupuesto del original, a causa de la falta de concordancia entre las necesidades reales de los usuarios finales del sistema y los requisitos elicitados [Kotonya&Sommerville 2002]. En general esto no se debe a que el grupo que trabaja sea ineficiente o aplique escasas técnicas de ingeniería, sino como consecuencia de que los requisitos definidos para el sistema no son lo suficientemente claros.

Los requisitos del sistema definen lo que se requiere que haga el sistema y las circunstancias bajo las cuales debe operar. Es decir, definen lo que el sistema se compromete a proveer y el conjunto de restricciones operativas.

Existe una audiencia variada para los documentos de requisitos; esta audiencia o lectores de los documentos son todos los interesados (stakeholders) del sistema, que tienen distintas intenciones y diferentes necesidades con respecto al documento de requisitos, entre los cuales se pueden encontrar:

- Los clientes del sistema: son quienes proporcionan las especificaciones del sistema y quienes chequean si sus necesidades se encuentran correctamente plasmadas. Ellos son los que solicitan cambios en los requisitos.

- Los administradores del proyecto de software: usan el documento de requisitos para planear el tamaño del sistema y planificar el proceso de desarrollo de software.
- Los ingenieros de sistemas: usan los requisitos para entender las necesidades del cliente/usuario y así desarrollar un sistema acorde a esas necesidades.
- Los ingenieros de pruebas del sistema: usan los requisitos para elaborar y llevar a cabo las pruebas del sistema.
- Los ingenieros del mantenimiento del sistema: usan los requisitos para entender el sistema y las relaciones entre las partes. [Kotonya&Sommerville 2002 pp. 6-16].

Una buena especificación debe incluir requisitos que sean técnicamente factibles, medibles, rastreables, no ambiguos y consistentes.

Muchas personas escriben casos de uso, para especificar requisitos, tanto para sistemas de software como para procesos de negocio. Pareciera que escribir casos de uso es una tarea sencilla, pero ¿Cuánto escribir?, ¿Qué detallar?, ¿Cómo escribirlo?. No es lo mismo decir que un caso de uso refleja correctamente la realidad que intenta representar, a decir que se ve bien.

En forma simple, se puede decir que un caso de uso es una porción de funcionalidad del sistema, desde la perspectiva del usuario, para obtener un resultado de valor. Tiene una notación gráfica y está acompañado de una descripción de lo que hace.

El conjunto de casos de uso especifican todas las formas existentes de usar el sistema, y reciben el nombre de Modelo de Casos de Uso.

En la World Wide Web, se encuentra gran cantidad de información relacionada a los casos de uso, y autores con diferentes perspectivas de aplicar o de emplear los casos de uso.

Estas diferentes 'vistas' o 'perspectivas' de los casos de uso pueden confundir al lector ya que, a pesar de tener el mismo origen (Jacobson,1967), se les fue dando diferentes usos, aplicaciones y extensiones de acuerdo a las necesidades y criterios. Esto también surge, tal como lo afirman Kotonya & Sommerville, y explicado anteriormente porque los lectores (stakeholders) de la especificación de requisitos son diferentes y con necesidades de información distinta.

Haciendo una breve reseña, los casos de uso surgen de los *escenarios* propuestos por Jacobson (1967). Estos representan distintos caminos posibles de ejecución de una funcionalidad, y de las propuestas de *conversaciones* de Rebecca Wirf-Brock (1988), en las que se distingue el diálogo entre el actor primario y el sistema, reflejando -en dos columnas- las acciones de cada uno de ellos, generalmente escritas para preparar el diseño de la interfaz del usuario, con gran detalle. Constantine y Lockwood, impulsores de los métodos ágiles -centrados en el usuario- para el desarrollo de software, adoptan este formato de conversación en lo que ellos llaman casos de uso esenciales, en donde todos los detalles de movimientos del usuario relacionados a la interfaz son omitidos, resultando descripciones mucho más cortas [Constantine&Lockwood, 2000].

En los 80, los casos de uso fueron incorporados en la comunidad de programación orientada a objetos, en donde llegaron para llenar un vacío existente en el proceso de requisitos. Alistair Cockburn (1994) introdujo el modelo conceptual de 'actores' y 'objetivos'. También en 1994, Booch, Rumbaugh y Jacobson, crean el Lenguaje de Modelado Unificado (UML) para los procesos de desarrollo de sistemas de información, el que en noviembre de 1997 se transforma en un estándar. Ivar Jacobson y otros en el libro *Object-Oriented Software Engineering: A Use Case Driven Approach*, refieren a los casos de uso para *representar lo que los usuarios deberían poder hacer con el sistema. Cada caso de uso es un curso de eventos completo, en el sistema, desde la perspectiva del usuario e iniciado por un actor. Es una forma específica de usar el sistema para realizar alguna porción de funcionalidad. Un caso de uso es entonces una secuencia especial de transacciones relacionadas, realizadas por un actor y el sistema, en un diálogo* [Jacobson et al.1998, pp157-159].

Algunos autores, reconocidos en el ambiente de la ingeniería de software orientada a objetos, también dan sus propias definiciones de casos de uso, siempre en consenso con la definición de Jacobson.

Si bien los casos de uso tienen una notación visual gráfica, deben estar acompañados de una descripción, generalmente en formato de texto u oraciones estructuradas, aunque también pueden usarse diagramas de flujo, diagramas de secuencia, Redes de Petri, o lenguajes de programación. Normalmente, los casos de uso sirven como medio de comunicación entre personas, sin que ellas deban tener algún entrenamiento en especial sobre 'cómo leerlos'. Por lo tanto, el texto simple suele ser la mejor opción.

Aprender a leer un caso de uso es aparentemente sencillo, aprender a construir adecuadamente casos de uso es una tarea ardua, y de ello depende una buena especificación de requisitos de software.

El modelo de casos de uso emplea el concepto de actores, para representar los roles que los usuarios pueden jugar en el sistema y los casos de uso para representar lo que los usuarios pueden hacer con el sistema.

Los casos de uso son el único concepto incluido en el enfoque de Jacobson en su libro de Ingeniería de Software Orientada a Objetos (OOSE) para modelar requisitos, pero cualquier persona sin conocimientos sobre orientación de objetos podría leer y entender el modelo.

El modelo de requisitos [Jacobson et al.1998, pp154-155] es la especificación de requisitos con la perspectiva del usuario, cuyo objetivo es delimitar el sistema y definir su funcionalidad. Para esto se desarrolla un gráfico conceptual del sistema, usando objetos del dominio del problema y también descripciones específicas de interfaz del sistema, si éstas son significativas para él. Se describe al sistema como un conjunto de casos de uso que son ejecutados por un conjunto de actores. Los actores se constituyen fuera del sistema y los casos de uso toman lugar dentro del sistema.

El modelo de requisitos debería funcionar como un contrato entre los desarrolladores y el cliente que solicita el sistema. Es la imagen para el desarrollador, de lo que quiere el cliente.

El modelo de requisitos luego es estructurado a través del modelo de análisis, realizado por un modelo de diseño, implementado por un modelo de implementación y probado por un modelo de pruebas. Estos modelos no solo son verificados sobre el modelo de requisitos, sino que son desarrollados directamente desde él.

2. LINEAS DE INVESTIGACION y DESARROLLO

- Identificar las características o elementos mínimos que debiera contener la descripción de los casos de uso, para poder dar trazabilidad de los requisitos del Documento de Requisitos del Sistema (SRD) o Documento de Especificación de Requisitos (ERS) a los casos de uso. Esto es, poder identificar cada requisito del usuario y saber qué caso de uso solucionará su necesidad. , aún en proceso de estudio.
- Encontrar indicadores de evaluación, en el modelo y la descripción de casos de uso, que permitan determinar si el diseño resultante (diagrama de clases) cumple con las necesidades reales ó da solución al problema planteado, y si está correctamente definido. En esta línea se está trabajando con información proveniente de las producciones de los alumnos avanzados de la cátedra Diseño de Software y Sistemas de Información II, de Licenciatura en sistemas de Información y Licenciatura en Ciencias de la Computación de la FCEF.N.

3. RESULTADOS PARCIALES OBTENIDOS

- Recopilación en Internet sobre el Modelo de Casos de Uso y perspectivas de uso en el desarrollo de Software.

- Análisis de la información relevada sobre casos de uso y su aplicación en distintas etapas del desarrollo de software.
- Estudio comparativo entre la concepción del Modelo de Casos de Uso y sus objetos entre los autores Larman, Amour-Miller, Weitzenfeld, Jacobson, Booch y Rumbaugh.
- Análisis de los ítems de evaluación relevantes en la construcción de un Modelo de Casos de Uso.
- Extracción e identificación de Indicadores de valoración del Modelo de Casos de Uso.
- Pruebas de valoración con esos indicadores.

Divulgación:

2008. Conferencia sobre “UML – Modelo de Casos de Uso”, destinada a alumnos y profesores de la Universidad del Cauca, Colombia, en el marco del proyecto Fortalecimiento de Redes Interuniversitarias II – de SPU. Disertante: María Inés Lund.

2008. “Experiencia Preliminar Incubación de Empresas Locales desarrolladoras de Software”, trabajo aceptado en el III Congreso Internacional: "El Aporte de la Universidad al Desarrollo Sostenible" – Univ. Católica de Cuyo – Instituto de Desarrollo Sostenible. San Juan. Autores: Pedro Zárate, Sergio Zapata, María Inés Lund, Emilio Ormeño.

2009. “Indicadores para la Evaluación de Diseños de Sistemas a través del Modelo de Casos de Uso en la Educación Superior”. Trabajo aceptado en el ICECE'2009 - International Conference on Engineering and Computer Education. Autores: María Inés Lund, Cintia Ferrarini, Emilio Ormeño y Sergio Zapata.

2009. “A CMT-based Modeling Language for Courseware Design” en el CSCWD 2009. esponsorado por la IEEE. Autores: Emilio Ormeño, Sergio Ochoa, Francisco Ibañez, M. Inés Lund, Susana Ruiz, Laura Aballay, Víctor Rosales.

4. FORMACION DE RECURSOS HUMANOS

Año 2008: Aprobación del Curso de Posgrado –“USO DE LA ESTADISTICA EN LA INGENIERIA DE SOFTWARE”. Instituto de Informática. Res N° 56/07-CD-FCEFN. Destinado a profesionales egresados en carreras relacionadas a la Ingeniería de Software. Laura Aballay, María Romagnano y María Inés Lund.

Año 2009: Aprobación del Curso de Capacitación en lenguaje Java. Cintia Ferrarini.

En ejecución Tesis de Maestría en Informática:

Metodología LuCoA para el Aprendizaje Lúdico Colaborativo en la Educación Superior. Cintia Ferrarini Oliver. Directores: Cesar Collazos, Mónica González de Doña.

Especificación de Requisitos y Uso de los Casos de Uso. María Inés Lund. Directores: Emilio Gustavo Ormeño y Fernando Pinciroli.

Mtedología para Modelar Procesos de Negocio. Maria G. Domagnano. Directores: Adriana De Luca y Emilio G. Ormeño

Asesoramiento de Tesis de Licenciatura:

2008. “Sistema de Gestión de solicitudes y proceso de adquisición de Obras Bibliográficas” – Desiderata. Autores: J.M.Blanco, A.Orellana Vassallo.

2008. “Propuesta de Desarrollo de Software adaptado a Pequeñas Organizaciones Locales de Software” - Aplicación a un caso de estudio: SIGEPRI. Autores: M.J. Sarmiento, V.E.Lucero.

2009. “Evaluación de Técnicas de Testing Funcional aplicando Ingeniería de Software Experimental”. Autores: A.Soto, Sánchez.

En ejecución: “Evaluación experimental de Técnicas de Elicitación Distribuida de Requisitos”. Alumnos: L.Montaña, M.Reus.

“Ciudad Digital: Sistema Web de Administración de Programas Sociales para el municipio de 25 de Mayo”. Alumno: Héctor Aravena.

Incorporación de un alumno con cargo auxiliar de 2da categoría, al cual se lo está formando en la temática general del proyecto, y capacitando en la búsqueda y evaluación de información en Internet, en adhesión con la competencia informacional “Aprender a buscar y evaluar información” propuesta en [Pinto 2009], en donde se pretende básicamente definir las principales secuencias y estrategias para llevar a cabo una búsqueda precisa y pertinente, para evaluarlos.

5. BIBLIOGRAFIA

Referencias:

[Armour&Miller 2001] Advanced use case modeling. Addison-Wesley © 2001. Frank Armour, Granville Miller.

[Booch et al. 1999] El lenguaje unificado de modelado. Addison-Wesley. 1999. Grady Booch, James Rumbaugh, Ivar Jacobson

[Cockburn 2005] Writing Effective Use Cases. Addison-Wesley. © 2001, Ed. 2005. Alistair Cockburn

[Constantine&Lockwood 2000] Artículo: “Structure and Style in Use Case for User Interface Design”. Larry L. Constantine, Lucy A.D. Lockwood. © 2000. <http://foruse.com/articles/structurestyle2.htm>

[Jacobson et al.1998] Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley. © 1992, Ed.1998. Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Overgaard.

[Jacobson et al. 2000] El proceso unificado de desarrollo de software. Pearson. 2000. Ivar Jacobson, Grady Booch, James Rumbaugh

[Kotonya&Sommerville 2002] Requirements engineering: processes and techniques. John Wiley. © 1998, Ed. 2002. Gerald Kotonya, Ian Sommerville.

Bibliografía de referencia:

[Larman 1999] UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice-Hall & Pearson. 1999. Craig Larman

[Leffingwell&Widrig 2003] Managing Software Requirements. Second Edition. A Use Case Approach. Ed. 2003. Dean Leffingwel y Don Widrig.

[Muller 1997] Modelado de Objetos con UML. Gestión 2000. 1997. Pierre-Alain Muller.

[Pinto, 2009] Pinto, María. Proyecto Alfin-EEES. Año 2009. www.mariapinto.es/alfineees/buscar.htm

[Pressman 2002] Ingeniería del software: un enfoque práctico. McGraw-Hill. 2002. Roger S. Pressman

[SWEBOK 2004] "SWEBOK" - Guide to the Software Engineering Body of Knowledge. 2004 version. - IEEE Computer Society -. www.swebok.org

Bibliografía de Internet:

1. "Use cases, ten years later". Alistair Cockburn. 2002. <http://alistair.cockburn.us/index.php/>

2. Métodos Ágiles. <http://www.foruse.com>

3. "Use cases - yesterday, today, tomorrow". Ivar Jacobson. 2003.

http://www.therationaledge.com/content/mar_03/f_useCases_ij.jsp

4. “UML – Resource Page”. <http://www.uml.org/>

5. “Casos de Uso. Un método práctico para explorar requerimientos”. Santiago Ceria. 2002. http://www-2.dc.uba.ar/materias/isoft1/2002_2/apuntes/CasosDeUso.pdf

6. “Relaciones entre Casos de Uso en el Unified Modeling Language”. Roxana S. Giandin, Claudia F. Pons. 2000. http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r11_art4_c.pdf

7. "Estudio de la Necesidad y Propuestas para Formalizar Esquemas de Trazabilidad". Angelina Espinoza Limón. 2004. <http://lucio.ls.fi.upm.es/doctorado/Trabajos20042005/Espinoza.pdf>
8. "Indicadores organizacionales para comparación de herramientas case en venezuela". Teresita Rojas, Luis E. Mendoza, María A. Pérez. 2000.
[http://www.lisi.usb.ve/prof/lmendoza/Documentos/PS-6117%20\(Laboratorio\)/Indicadores%20Organizacionales%20para%20Comparaci%C3%B3n%20de%20Herramientas%20CASE%20en%20Venezuela.pdf](http://www.lisi.usb.ve/prof/lmendoza/Documentos/PS-6117%20(Laboratorio)/Indicadores%20Organizacionales%20para%20Comparaci%C3%B3n%20de%20Herramientas%20CASE%20en%20Venezuela.pdf)