

Los Casos de Uso del Rational Unified Process (RUP)

Balverdi Cruz Luis Harley¹, Cachay Valdivia Jimy, Ing. Daniel Lévano Rodríguez Mg.²

RESUMEN

El uso del lenguaje de modelamiento unificado (UML) como una notación que ayuda en el proceso de desarrollo del software. Es por eso el empleo de los casos de uso, como parte del UML. El propósito de los casos de uso es describir en lenguaje amigable para la funcionalidad completa de un sistema a desarrollar y su empleo se realiza en el proceso de especificación de requisitos del sistema. Existen, muchas formas de aplicar los casos de uso y no son pocas las veces en que su empleo es inadecuado. Algunas de las causas son: mala interpretación del estándar UML y la secuencia incorrecta de actividades para la creación de casos de uso. Representa la forma que un actor opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan "operaciones o casos de uso". Como resultado de las actividades de este flujo de trabajo, se desarrolla la visión, el modelo de los casos de uso, los casos de uso que en conjunto describen la herramienta a desarrollar. En concreto, en el diagrama no muestra el orden en el que se llevan a cabo los pasos para lograr los objetivos de cada caso de uso. Los casos de uso solamente se utilizan para los requisitos funcionales de un sistema.

Palabras clave: Actor, caso de uso y UML.

ABSTRACT

The use of Unified language model (UML) as a standard for software construction has spread in recent years. That is why the use of use cases, it is as part of the UML standard. The purpose of use cases is to describe the full functionality of a system to develop in natural language and its use is in the process of system requirements specification. Unfortunately exists, many forms of implementing the use cases and not few the times that its use is inappropriate. Some of the causes are misinterpretation of the UML standard and incorrect sequence of activities for the creation of use cases. Represents the shape of a "actor" client operates with system in development, as well as the shape, type, and order in how elements interact 'operations or use cases. As a result of the activities of this workflow, develops vision, model use cases use cases which together describe the tool to develop. Also built a glossary and a prototype of the user interface. In particular, the diagram does not show the order in which the steps are performed to achieve the objectives of each use case.

Key words: Actor, use case y UML.

¹ Autor - Estudiante de la Universidad Peruana Unión
- luxobalverdi@gmail.com
- 968923696

² Asesor

1. INTRODUCCIÓN

Uno de los primeros procesos que se realizan en un proyecto de construcción de software es la especificación de requisitos. Los objetivos de este proceso son identificar, validar y documentar los requisitos de software; es decir determinar las características que deberán tener el sistema o las restricciones que deberá cumplir para que sea aceptado por el cliente y los futuros usuarios del sistema de software. El modelo de casos de uso debe servir como medio de comunicación y soporte, entre clientes, usuarios y desarrolladores; proveyendo funcionalidad “el cómo” a ser implementada.

El producto final de este proceso es el documento de especificación de requisitos de software y en este se señala, con el detalle adecuado, lo que el usuario necesita del sistema de software. Es por ello, que el documento de requisitos de software se considera como un contrato entre el cliente y el equipo de desarrollo del sistema. El modelo se compone de casos de uso y actores. Cada caso de uso en el modelo es escrito en detalle, mostrando paso a paso la forma como los actores interactúan con el sistema y como responde el sistema.

Los objetivos es establecer y mantener la conformidad de las

necesidades de los clientes y usuarios acerca de lo que el sistema debe hacer, proporcionar a los desarrolladores una mejor comprensión de los requerimientos del sistema, definir las fronteras del sistema. Proporcionar la base del planeamiento de los contenidos técnicos de las iteraciones, proporcionar la base de la estimación de costos y tiempo de desarrollo del sistema, definir una interfaz de usuario para el sistema centrada en las necesidades y metas de los usuarios.

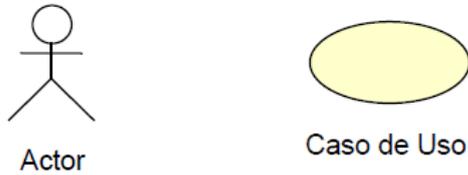
2. UML Y LA ESPECIFICACIÓN DE REQUISITOS

Yourdon [1] Concuerta que para determinar la funcionalidad de un sistema a desarrollar UML señala el uso de dos elementos: el actor y el caso de uso. El actor representa una entidad externa que interactúa con el sistema, las entidades externas podrían ser personas u otros sistemas. Es importante resaltar que los actores son abstracciones de papeles o roles y no necesariamente tienen una correspondencia directa con personas. El caso de uso hace referencia al sistema a construir, detallando su comportamiento, el cual se traduce en resultados que pueden ser observados por el actor. Los casos de uso describen las cosas que los actores quieren que el

¹ Autor - Estudiante de la Universidad Peruana Unión
- luxobalverdi@gmail.com
- 968923696

² Asesor

sistema haga, por lo que un caso de uso debería ser una tarea completa desde la perspectiva del actor.



Grafica1.- Representación gráfica de actor y caso de uso[1]

UML especifica que para representar gráficamente la relación entre un actor y caso de uso se debe trazar una línea que los una, a la que se le denomina “relación de comunicación”. Además, UML señala que los casos de uso pueden tener relaciones entre sí. Los tipos de relaciones que pueden ser: “include” “extends” y “generalization”

¿QUÉ ES UN INCLUDE?

Bittner [2] Explica que son términos muy simples, cuando relacionamos dos casos de uso con un “include”, estamos diciendo que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido). Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien, pues no podría cumplir su objetivo. Para una venta en caja, la venta no puede considerarse completa si no se realiza el proceso para cobrarla en ese momento.

¿QUÉ ES UN EXTENDS?

Una de las diferencias básicas es que en el caso del “extend” hay situaciones en que el caso de uso de extensión no es indispensable que ocurra, y cuando lo hace ofrece un valor extra (extiende) al objetivo original del caso de uso base.

¿QUÉ ES UNA GENERALIZATION?

La Generalización especifica que un caso de uso hereda las características y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

ACTIVIDADES PARA LA ESPECIFICACIÓN DE REQUISITOS CON CASOS DE USO

Brown [3] Expresa que los resultados de la especificación de requisitos son dos productos: el catálogo de requisitos y el documento de especificación de requisitos de software. El primero de ellos contiene la lista de requisitos de software clasificada por tipo y prioridad; y el segundo de ellos, especifica el comportamiento del sistema a un grado de detalle mayor al del catálogo de requisitos.

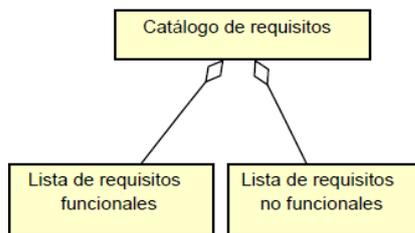
¹ Autor - Estudiante de la Universidad Peruana Unión
- luxobalverdi@gmail.com
- 968923696

² Asesor

IDENTIFICAR Y CLASIFICAR REQUISITOS

Bruegge [4] Asume que esta actividad es el punto de partida para las siguientes actividades del proceso de obtención de requisitos y se refiere a la identificación de los requisitos del sistema de software a desarrollar. En esta actividad, deberemos responder a los siguientes cuestionamientos: ¿Qué le permitirá hacer, el sistema de software al usuario? ¿El cliente o usuario me solicita alguna restricción para construir el sistema de software?

Luego de ser contestadas las preguntas, se clasificarán en dos grupos: requisitos Funcionales y requisitos no funcionales.



Grafica2.- Especificación de requisitos[1]

IDENTIFICAR CASOS DE USO

El caso de uso es el que especifica todos los escenarios posibles para una parte de funcionalidad dada, es decir todos los escenarios similares se agrupan en un solo caso de uso.

IDENTIFICAR ACTORES

Para encontrar actores del sistema se puede buscar en las categorías de personas.

Para un sistema de biblioteca, los actores podrían ser: bibliotecario.

En el caso de un sistema de ventas, los actores podrían ser: el cliente.

IDENTIFICAR ESCENARIOS

Un escenario es una descripción concreta, enfocada e informal de una sola característica del sistema desde el punto de vista de un solo actor, es decir, un escenario muestra la secuencia de pasos que se produce cuando un actor interactúa con el sistema en una situación específica y un tiempo determinado.

ERRORES EN LA IDENTIFICACIÓN DE CASOS DE USO

Schneider [5] Señala que los casos de uso deben mostrar lo que el usuario necesita del sistema y no mostrar las funciones u opciones del menú que permitirán realizar lo solicitado.

¹ Autor - Estudiante de la Universidad Peruana Unión
- lux0.balverdi@gmail.com
- 968923696

² Asesor

ERRORES EN LA IDENTIFICACIÓN DE ACTORES

Los errores introducidos en esta etapa se deben principalmente a no comprender quienes son los actores del sistema.

En algunos casos se incluyen actores que realmente no lo son por ejemplo, en un sistema en el que se realizan pedidos de productos, se considera al cliente como un actor. Realmente quien ingresa los pedidos en el sistema es el vendedor y no el cliente, por lo tanto el vendedor sería el actor del sistema[2].

APORTE SOCIAL

Los casos de uso se utilizan para tener una mejor comprensión del Software a realizar, nos muestra además un panorama amplio, interactivo y objetivo del funcionamiento del Software, de esa manera nosotros necesitamos tener las cosas bien definidas, planteadas y fundamentales para así contribuir con las organizaciones valores y efectividad en la labor a desempeñar.

APORTE ESPIRITUAL

El requerimiento principal es creer en Dios. En la vida cotidiana cada persona debe realizar diferentes acciones que lo lleven a estar más en comunión con nuestro padre, cuando nuestra vida esta aferrada a Dios podremos

desempeñarse bien en el área que estamos laborando porque él nos brindó sabiduría e inteligencia; sobre todo nuestro objetivo principal es tener la vida eterna.

3. CONCLUSIÓN

En este artículo se muestran las actividades que se deben realizar para la especificación de requisitos utilizando casos de uso. Estas actividades han permitido minimizar los errores en la aplicación del estándar UML y lo que es importante finalizar el proceso con un documento de especificación de requisitos libre de errores y útil para las etapas de análisis y diseño. El uso de las relaciones de casos de uso es opcional y no es necesaria para realizar un documento de especificación de requisitos adecuado y detallado.

El presente trabajo es un inicio para el establecimiento de patrones en la aplicación de casos de uso, de manera que nos puedan ayudar a mejorar.

¹ Autor - Estudiante de la Universidad Peruana Unión
- luxobalverdi@gmail.com
- 968923696

² Asesor

REFERENCIAS

1. Yourdon E., Analisis Estructurado Moderno, Prentice Hall hispanoamericana, México, 1989.
 2. Bittner, K., Why Use Cases Are Not functions, <http://www.therationaledge.com>, USA, 2000.
 3. Brown, W.J., AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis, John Wiley & Sons, USA, 1998.
 4. Bruegge B., Allen, S., Ingeniería de Software Orientado a Objetos, Addison-Wesley, USA, 2002.
- Ministerio de Administraciones Públicas, Analisis del Sistema de Informacion-Metrica Version 3, Espana, 2000.
5. Schneider, G., Winters, J.P., Applying Use Cases, Second Edition, Addison-Wesley, Massachusetts, USA, 2001.

¹ Autor - Estudiante de la Universidad Peruana Unión
- luxobalverdi@gmail.com
- 968923696

² Asesor