EL MODELO DE NEGOCIO BASADO EN UML : ELEMENTO CLAVE PARA LOGRAR UNA ADECUADA ADMINISTRACION DE REQUERIMIENTOS

Msc. Orlando Rivera Jurado

Ingeniería de Sistemas – Universidad Católica Boliviana "San Pablo"

La Paz – Bolivia

oriver@ucb.edu.bo

Abstract— **El modelado del negocio es uno de los aspectos mas importantes que permite conjugar el desarrollo de un proyecto informático con las metas y objetivos de las organizaciones. Si se realiza de tal forma en que el modelo quede consensuado entre los grupos involucrados, las posibilidades de éxito del proyecto aumentarán en forma muy importante. El modelado de procesos de negocios, es una de las representaciones más eficientes para comunicarnos con los usuarios de todos los niveles, para lograr una adecuada administración de los requerimientos.**

Keywords—**Modelo de negocio, UML, Requerimientos funcionales, Casos de uso, Instancias de casos de uso.**

I. INTRODUCCIÓN

Un proceso de negocio es un conjunto estructurado de actividades, diseñado para lograr un objetivo. Los procesos describen cómo es realizado el trabajo en la empresa y se caracterizan por ser observables, medibles, mejorables y repetitivos. Por otro parte, un sistema informático es un conjunto de componentes organizados que cumplen funciones específicas, empleando tecnología informática. Este concepto se relaciona con el conjunto de funciones de información que están siendo requeridas por las actividades de un proceso de negocio. Es decir, los procesos de negocios requieren funciones de información, y los sistemas informáticos entregan funcionalidades operativas, que apoyan a las actividades de los procesos.

El análisis de los modelos de procesos de negocios está enfocado principalmente a la completitud de éstos, es decir, qué elementos de la realidad a ser modelada pueden ser representados. El concepto de modelo considera aspectos inherentes a los procesos de negocios y aspectos de la relación proceso de negocio y sistemas informáticos.

Lo anterior, conforma la base conceptual, que permite identificar los elementos que necesitan ser representados en los modelos de procesos de negocios que consideren explícitamente su relación con sistemas informáticos.

El modelado del negocio es la técnica por excelencia para conciliar el desarrollo con las metas y objetivos de las empresas e instituciones. Si se realiza de tal forma en que el modelo quede consensuado entre los grupos interesados (*stakeholders*), las posibilidades de éxito del proyecto aumentarán en forma muy importante. El modelado de negocios, y más específicamente el modelado de procesos de negocio, es la forma idónea para comunicarnos con los usuarios de todos los niveles.

II. UML Y EL PROCESO DE DESARROLLO

UML es un lenguaje unificado de modelado, que gracias a su semántica y sintaxis basada en una notación grafica, permite la representación conceptual y física de un sistema. Proporciona un vocabulario y reglas con el propósito de facilitar la comunicación, de modo tal que el analista pueda describir un modelo en UML y que el desarrollador pueda interpretarlo sin ambigüedades. Lo que ha permitido el desarrollo de sistemas más robustos y extensibles, y un aspecto importante: lograr una adecuada forma de comunicación entre las diversas entidades de cualquier organización. Si bien UML ha sido creado para modelar sistemas complejos que agrupan software, es lo suficientemente expresivo como para modelar sistemas no informáticos, como flujos de trabajo en una empresa (*workflow*), diseño de la estructura de una organización y también en el diseño de hardware.

Aunque UML es independiente del proceso de desarrollo, sus creadores han propuesto una metodología, denominada el Proceso Unificado de Desarrollo, que fundamenta la construcción de un sistema basado en componentes interconectados a través de interfaces. Además, el Proceso Unificado utiliza la notación UML para expresar gráficamente todos los esquemas de un sistema. Los aspectos que lo definen son: iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura (MDA).

Es iterativo e incremental porque el desarrollo de un sistema relativamente complejo involucra un gran esfuerzo que puede tener una duración considerable. Por lo tanto, lo más práctico es dividir un proyecto en fases. Cada recorrido por las fases se denomina iteración, la cual corrige defectos de la anterior e incorpora nueva funcionalidad, se denomina también *dirigido por el riesgo*, lo que significa que cada nueva versión se ataca y reducen los riesgos más significativos para lograr el éxito del proyecto.

Dirigido por casos de uso puesto que en base a los casos uso, se crean una serie de modelos de análisis, diseño que permiten establecer el comportamiento de lo "que se hace" y lo "que se hará". Permiten la comunicación entre las personas involucradas en el proyecto, además, estos modelos validan la arquitectura y las pruebas en base a los componentes desarrollados.

Centrado en la Arquitectura *e*s decir el desarrollo de un proyecto de software debe incluir los elementos estructurales y funcionales del sistema. Cada uno de estos están representados por un modelo en particular, así como en la arquitectura de la construcción, se consideran varios aspectos: estructura de la edificación, conducciones eléctricas, instalaciones sanitarias, etc.

III. EL APORTE DE UML 2.0

Las versiones de UML desde UML 1.0 tenía como objetivo principal la respuesta a las necesidades clásicas de la industria del Software, UML 2.0 plantea cubrir otros campos como el **modelado de negocio**, sistemas en tiempo real, sistemas basados en componentes, etc. El mecanismo estándar para extender las capacidades de UML a cada uno de estos campos específicos es el de los perfiles y este mecanismo se ve mejorado en la nueva versión. En general, se ha mejorado la escalabilidad y la integracion.

IV. LA ADMINISTRACIÓN DE REQUERIMIENTOS

Alguna vez no le ocurrió la circunstancia de comprar algún objeto que al cabo de un tiempo termine sin ser utilizado?. Probablemente se deba a su complicado uso o simplemente porque no colma nuestro interés, o no satisface una real necesidad. Sea cual fuera la razón, la verdad es que hicimos un gasto improductivo e innecesario.

Con el software ocurre con frecuencia algo semejante, no es extraño el hecho de contratar a una empresa para el desarrollo de software, sólo para darse cuenta del desperdicio de dinero y tiempo, pues no obtuvieron lo que de ellos esperaban. Una de las razones principales por las cuales se da esta situación, y de hecho, una de las causas principales por las cuales los proyectos fracasan, se debe a una mala administración de requerimientos. Esto generalmente se da por falta de habilidad en el personal responsable o por el empleo de técnicas inapropiadas. La administración de requerimientos, cubre actividades como la recopilación, documentación, validación y control de los requerimientos y sus eventuales cambios. UML, como estándar integrador de las prácticas de desarrollo de software ofrece los Casos de Uso como una técnica muy adecuada para la licitación de requerimientos, de forma independiente al método de análisis empleado.

V. REQUISITOS FUNCIONES Y NO FUNCIONALES

Hay que considerar que los requisitos pueden ser de carácter funcional o no funcional, en el primer caso se especifican las acciones que el sistema debe ser capaz de realizar, generalmente expresados en términos de entradas y salidas. Los requisitos no funcionales están orientados a las características del sistema como la plataforma operativa, el uso del lenguaje, tiempos de respuesta, rendimiento, interfaces de usuario, .. etc.
Los requisitos funcionales se manejan cuando los flujos de trabajo (*workflow*) de los requisitos y del análisis están en proceso de desarrollo, a diferencia de los requisitos no funcionales pueden esperar hasta la fase diseño, puesto que en esta etapa es donde se definen mejor estos requisitos.

VII. LOS REQUERIMIENTOS FUNCIONALES

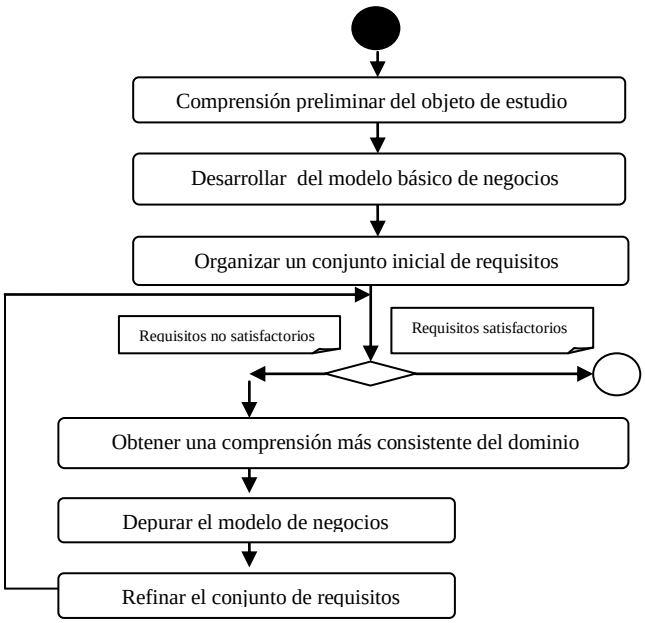
Normalmente los Casos de Uso son iniciados por algún actor (primario), se inicia con algún evento, que podría ser tan simple como elegir una opción en el sistema, y continúa como una serie de eventos o interacciones entre actores y sistema, hasta que el objetivo del caso de uso se cumple. Por lo tanto los casos de uso describen la funcionalidad del sistema para alcanzar los objetivos más importantes.

Lo que estamos obteniendo así es una especificación de requerimientos funcionales mediante un análisis descendente, es decir, primero obtenemos los objetivos que hay que cumplir en el sistema descritos con el nombre del caso de uso (ej: Realizar una venta) y después buscamos cuáles son las funciones o requerimientos funcionales precisos y ordenados, para que el actor cumpla dicho objetivo al utilizar sistema. Esto nos lleva a identificar requerimientos realmente relevantes para alcanzar la misión del sistema. A continuación se resume una recopilación de los requerimientos funcionales adecuados del sistema mediante casos de uso:

1. Especificar la misión del sistema
2. Identificar quiénes utilizarán el sistema (actores primarios)
3. Averiguar cuáles objetivos desean cumplir los actores al usar el sistema (casos de uso)
4. Identificar los pasos o eventos de cada caso de uso (especificación del caso de uso)

VIII. LOS PASOS INICIALES

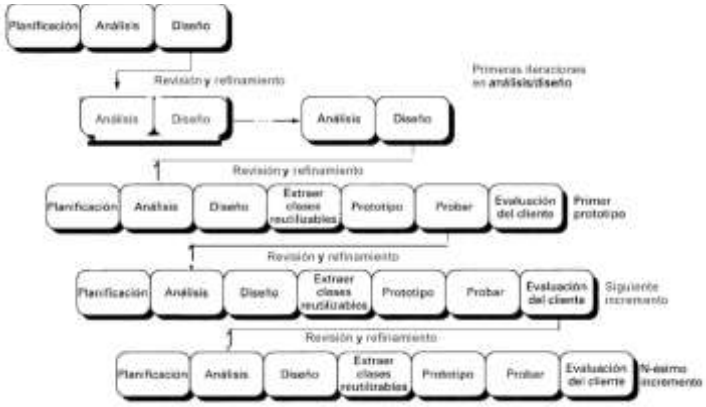
El proceso unificado es iterativo, en el desarrollo de cualquier proyecto (*worklow de requisitos*), el primer paso es obtener una comprensión del dominio u objeto de estudio, una vez comprendido el dominio de aplicación, se construye un modelo de negocios, este modelo se emplea para determinar los requisitos del cliente.



El modelo de negocios inicial, es una descripción de los procesos del dominio de aplicación especificando los datos, las actividades, roles y las reglas de negocio. Para su adecuada construcción se debe comprender con detalle los diversos procesos involucrados en el dominio de estudio, para lo cual se pueden emplear una serie de técnicas que permitan obtener información necesaria y suficiente que consoliden el modelo de negocios (especialmente las entrevistas), es en este punto donde UML provee una de sus herramientas más importantes: el diagrama de casos de uso, que representa la interacción entre el sistema y los usuarios (actores) a partir de requisitos funcionales, es decir se está mencionando lo que tiene que hacer un sistema. Esta técnica permite capturar información de cómo un sistema o negocio trabaja actualmente, y de cómo se desea que trabaje. No pertenece al enfoque orientado a objetos, es una técnica para el modelado de escenarios (un escenario es una instancia de un Caso de Uso) en los cuales el sistema debe operar. Se determinan observando y precisando, las secuencias de interacción desde el punto de vista del usuario. Es una muy buena forma de dirigirse hacia el análisis orientado a objetos.

En esta etapa es importante mantener el nivel de abstracción que logre una mejor comprensión del usuario, simplificando la representación de cada caso de uso que describa los procesos más importantes del dominio. Se ve en la práctica que quienes se inician en el desarrollo de sistemas comienzan a capturar requerimientos en base a la elaboración de un excesiva cantidad de casos de uso, provocando dos problemas: el usuario que debe entender este modelo se ve ante una maraña de casos de usos, que en lugar de orientarlo terminan por confundirlo más. Se debe recordar además que esta etapa debe ser la mejor entendida por el usuario, para lograr las retroalimentaciones necesarias que permitan refinar este modelo, hasta lograr que los requerimientos del negocio sean satisfactorios. Un segundo problema es el hecho de entrar en detalles prematuros, que resultan más costosos de corregirlos, cuando un requerimiento no ha sido comprendido o ha sido omitido, es como dejar correr una excesiva cantidad de agua solo para tomar una muestra.

En este sentido se recomienda efectuar el ejercicio de la abstracción combinado con la iteración de los casos de uso, es decir simplificar y generalizar procesos, en una primera factorización (refinación) de casos de uso de alto nivel. Se debe considerar que una abstracción es padre e hijo a la vez, porque primero se concibe a partir del análisis de ciertos elementos, pero luego sirve como origen para la concreción de nuevos elementos derivados.



Sergio Orozco Consultor de Sistemas de Milestone Consulting, afirma:

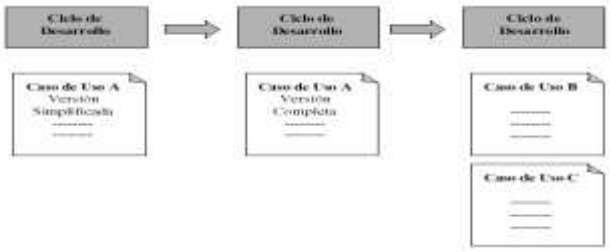
"Por experiencia sabemos que, en los modelos de UML, el buen uso de pocos elementos da mejores resultados que el uso de muchos elementos mal aplicados. La esencia de los modelos radica en la simplicidad, ya que facilita el análisis, entendimiento y comunicación entre los participantes del desarrollo. Existen elementos adicionales para modelar los casos de uso, pero podemos asegurar que la sencillez es parte importante del modelado, y esto implica que en ocasiones, y

sobre todo cuando el analista no tiene tanta experiencia en UML, es mejor limitarnos a los elementos más básicos."

A partir de esta primera representación de casos de uso básico se debe proceder a determinar los requisitos del cliente en base al modelo inicial de negocios (los casos de uso básicos o de alto nivel) que también son sujetos a cambios en función a una mejor comprensión del dominio y del modelo de negocios, por la interacción continua con el usuario. Los requisitos son cambiantes dependiendo de muchos factores, por lo que un requisito que inicialmente pueda haber sido considerado importante puede no serlo, después de una mejor retroalimentación. La herramienta que permite hacer una descripción más adecuada de los requisitos es la ***descripción de los casos de uso***, que establece una pormenorización del caso de uso, a través de una representación del escenario de los procesos contenidos dentro del caso de uso inicial. Se considera a los casos de uso como una técnica para el modelado de escenarios en los cuales el sistema debe operar, estos cuando están lo suficientemente refinados se convierten en un elemento valioso para efectuar el subsecuente modelado conceptual y de comportamiento.

Según lo descrito anteriormente no se han establecido compromisos con la solución, es decir se han descrito los casos de uso a un nivel abstracto (esencial o de **alto nivel**), independiente de la tecnología y de la implementación. Los casos de uso definidos a alto nivel son siempre esenciales por naturaleza, debido a su brevedad y abstracción. Por el contrario, un caso de uso **real** describe concretamente el proceso en términos del diseño, la solución específica que se va a llevar a cabo. Se ajusta a un tipo de interfaz específica, y se desciende a detalles como interfaces y objetos.

A continuación se esquematiza la evolución de los casos de uso (refinamiento)



IX. ESPECIFICACIONES DE CASOS DE USO (INSTANCIA DEL C.U)

La mayoría de los proyectos de software que fallan tienen como causa principal una mala administración de requerimientos. Un ejemplo en este sentido suele ser un bajo nivel de entendimiento de los requerimientos entre usuarios y desarrolladores. Aún y cuando el equipo de desarrollo cree comprender lo que el cliente le está solicitando, puede no ser así. El caso extremo es que en las primeras etapas ni siquiera el cliente está totalmente consciente de lo que quiere o necesita. Ahí es donde el analista debe facilitarle al usuario expresar sus necesidades para validarlas posteriormente mediante mecanismos eficientes de comunicación que ambos entiendan. Un ejemplo excelente de estos mecanismos son las especificaciones de casos de uso.

Partiendo de la premisa que ya se identificaron los actores y casos de uso apropiados del sistema lo que corresponde es detallar los pasos necesarios para cumplir con dicho caso de uso. Para especificar **cada caso de uso** se deben tomar en cuenta los siguientes elementos:

1. **Interacciones:** Mencionar la participación tanto de actores primarios como de secundarios desde que inicia el caso de uso hasta su culminación
2. **Eventos:** Indicar cada uno de los eventos que ocurren durante del caso de uso.
3. **Nivel de detalle:** Los casos de uso y sus especificaciones son la licitación que establecemos con el cliente, por lo que debemos especificarlos al detalle. Mientras más conozcamos de la funcionalidad del sistema, más precisas serán las estimaciones de nuestro plan de trabajo.
4. **Escenarios:** Un caso de uso muestra diferentes escenarios posibles y no una sola forma de ejecutarlo. Se debe detallar cada uno de esos escenarios, mediante un flujo principal y sus diferentes flujos alternos y excepcionales.
5. **Claridad en la especificación:** En general se debe ser lo más claro posible en la explicación de los casos de uso utilizando la terminología adecuada del objeto de estudio, sin entrar en detalles técnicos a los que no está acostumbrado.

Algunas veces los analistas evitan el esfuerzo requerido para realizar la "documentación" que implica especificar los casos de uso, pero vale la pena el esfuerzo, por los resultados que se han de obtener. Si no se especifica claramente qué es lo que requiere el cliente entonces la incertidumbre en cuanto a tiempo y costo será algo que no podremos evitar. En este sentido dependerá del contexto del proyecto el nivel de detalle a realizar, y por lo tanto la cantidad de "documentación" generada para especificar los casos de uso. Sólo hay que tomar en cuenta que entre menos precisión y detalle haya, mayor será el riesgo de no tener un proyecto funcional.

X. ABSTRACCION EN LAS INSTANCIAS DE LOS CASOS DE USO

Inicialmente prepare su escenario de alto nivel (secuencia de acciones e interacciones entre actores y sistema) para un caso ideal (evitando las restricciones), esta abstracción le permite manejar mejor sus requerimientos, dejando a un lado momentáneamente el "y que ocurre si…", lo que permitirá una mayor claridad en esta primera aproximación a los requerimientos del usuario.

En la etapa de refinado final, se deben incluir las acciones alternativas (restricciones). En consecuencia la idea que se persigue es la de crear casos de uso que tengan la necesaria abstracción para su mejor comprensión, pero en donde se efectúa un detalle completo en términos funcionales, es en la descripción de los casos de uso y de manera particular a partir de la instancia de los casos de uso: los escenarios.

En conclusión, la especificación del caso de uso, es la razón por la cual el desarrollador debe evitar el uso de excesivo de casos de uso en forma de esquemas. Una de las grandes ventajas de un lenguaje grafico es su poder expresivo, pero si este se complica con una excesiva cantidad de símbolos, el usuario en lugar de entender lo que el analista presenta para su retroalimentación o su aceptación, terminara rechazando el mismo por su complejidad (Un grafico vale por mil palabras, pero mil gráficos pueden a veces no representar una coherente idea).

XI .HERRAMIENTAS PARA APOYAR EL MODELO DE NEGOCIOS

El modelado de los procesos de negocio se ha visto facilitado por nuevas herramientas que permiten una mejor definición de los requisitos funcionales, esto debido a que muchas veces en el proceso de recolección de requisitos, los usuarios no están seguros como funcionan los procesos de negocio en los que están involucrados, en consecuencia resulta muy difícil definir los requerimientos a cabalidad del sistema, es en estas circunstancias que es necesario contar con medios que permitan comprender mejor los procesos del negocio.

Hoy en día, y de acuerdo con el estado de la industria del modelado de negocios, es posible identificar a UML y BPMN (Business Process Modeling Notación) como los estándares más importantes. Aunque ambos son aceptados en el ámbito de los negocios, el modelado con UML es dominante en la industria del software. Sin embargo la simbiosis de estos modelos puede permitir pasar desde las especificaciones de procesos de negocio hasta modelos de software más coherentes y concretos. Según Sthepen White en su libro *Modeling and Reference Guide:*

"El mundo de los procesos de negocio ha cambiado en los últimos años. Un proceso de este tipo abarca múltiples participantes, y la coordinación puede ser compleja. Antes de ***BPMN*** *no había una técnica de modelado estándar desarrollado para encargarse de estos asuntos, ha sido desarrollado para proveer a los usuarios de una notación de uso libre. Esto beneficiará a los usuarios de la misma forma que UML benefició el mundo de la ingeniería de software".*

UML toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que BPMN toma un perfil orientado a procesos en el modelado de sistemas, y tiene un enfoque en procesos de negocio, UML se enfoca al diseño de software y por lo tanto ambas notaciones son totalmente compatibles entre sí.

De la misma forma las extensiones UML para el modelado de negocio aportan elementos muy importantes ya que proporcionan algunas vistas de la arquitectura de negocio que son más difíciles de observar usando únicamente **BPMN**. Por ejemplo, la visualización de las responsabilidades, la manipulación de las entidades y la comprensión de los estados asociados a las entidades del negocio.

XII. CONCLUSIONES

Si bien algunos autores recomiendan el incorporar diagramas de comportamiento de UML que complementan el complejo proceso de la captura de requisitos, además de las especificaciones de casos de uso, previa a la definición del modelo conceptual, considero importante tomar en cuenta las alternativas de complementar el enfoque guiado por casos de uso con la notación de modelado de negocios, creo que este enfoque permitirá el desarrollo de una mejor elaboración del análisis de requerimientos, labor esencial en el proceso de desarrollo de sistemas, permitiendo en última instancia lo que todo diseñador de sistemas desea, ver que el sistema una vez concluido sea de satisfacción para el dominio de usuarios a quienes está dirigido.

Por otra parte si bien el empleo de UML se adecua mas a la construcción de sistemas grandes y altamente complejos, que requieren continuas refinaciones en el modelo de negocio y en el conceptual, hasta lograr un modelo que este acorde a los requerimientos del usuario, no se debe olvidar que el modelo de negocios con técnicas semi-formales como las contenidas en este documento deberían aplicarse aun en proyectos pequeños o medianos, porque aun siendo el conjunto de requerimientos pequeño o mediano, esto no garantiza una adecuada comprensión de los mismos.

REFERENCIAS

[1] Larman, G. "UML y Patrones", Prentice Hall, 1999
[2] Perdita Stevens "Utilizacion de UML en Ingenieria del Software con Objetos y componentes", Addisson Wweslwy, 2002
[3] Raul Alarcon, "Diseño O.O con UML", Grupo EIDOS, 2002
[4] Jacobson, G.Booch, J.Raumbaugh, "El proceso Unificado de Desarrollo", Addison Wesley, 2000
[5] Dan Pilone, Neil Pitam, "UML 2.0 in a Nutshell", O'Really , 2005
[6] Robert A.Maksimmchu, Eric J.Naiburg , "UML for Mere Mortals",Addison Weslwy, 2004
[7] Kendall Scoot, "Fast Track UML 2.0", Apress, 2004
[8] Kim Hamilton, Russel Miles, "Learning UML 2.0",O'Really, 2006
[9] Paul Kimmel , "UML Demystified", McGrawHill, 2005
[10] Gunnar Övergaard, Karin Palmkvist , Addison Weslwy,"Use Cases Patterns and Blueprints", 2004
[11] Sthepen White , "Modeling and Reference Guide BPMN"