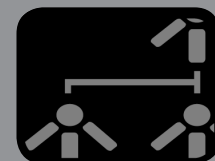


PARTICIPACIÓN DE LOS CASOS DE USO EN LA SOLUCIÓN DE LOS PROBLEMAS DE UNA ORGANIZACIÓN

USE CASES PERFORMANCE IN ORGANIZATIONAL PROBLEM SOLVING



AUTOR

CARLOS MARIO ZAPATA JARAMILLO
Ph.D.
*Universidad Nacional de Colombia
Profesor Asociado
Escuela de Sistemas
cmzapata@unal.edu.co
COLOMBIA

AUTOR

GLORIA LUCIA GIRALDO GÓMEZ
Ph.D.
*Universidad Nacional de Colombia
Profesora Asociada
Escuela de Sistemas
lggiraldog@unal.edu.co
COLOMBIA

AUTOR

DAVID ANDRÉS MORENO NIÑO
Ph.D.
Estudiante de Ingeniería de Sistemas e
Informática
*Universidad Nacional de Colombia
Escuela de Sistemas
damorennon@unal.edu.co
COLOMBIA

INSTITUCIÓN

*UNIVERSIDAD NACIONAL DE COLOMBIA
UNAL Medellín
Universidad
Carrera 80 No 65-223
Medellín, Antioquia
cmzapata@unal.edu.co
COLOMBIA

RECEPCIÓN: Abril 30 de 2010

ACEPTACIÓN: Mayo 23 de 2010

TEMÁTICA: Ingeniería del software

TIPO DE ARTÍCULO: Artículo de Reflexión

RESUMEN ANALÍTICO

Mediante las decisiones que toman, las organizaciones logran sus objetivos y superan los problemas que surgen. Según la metodología Kepner-Tregoe, se puede establecer una relación secuencial o de trazabilidad entre la definición de una solución, sus problemas y sus objetivos, con el fin de encontrar la más adecuada. Sin embargo, esta relación no se aplica al proceso de educación de requisitos en el método de desarrollo UN-Método y el análisis de la problemática de la organización y la valoración del impacto que la aplicación de software tiene sobre ella, la realiza el analista subjetivamente. Por ello, en este artículo se propone un método para determinar la participación de los casos de uso en la solución de los problemas de una organización, de modo que se aplique la relación de trazabilidad que se puede establecer entre los artefactos que se afectan al introducir una solución software a la organización, con el propósito de que se realice una valoración más objetiva de dicha aplicación de software.

PALABRAS CLAVES: Educación de requisitos, Valoración de soluciones, Casos de uso, Trazabilidad

ANALYTICAL SUMMARY

Organizations reach their goals and solve their problems by making decisions. According to Kepner-Tregoe methodology, goals and problems must be traced to the solution, in order to complete the most suitable solution. However, UN-Metodo elicitation requirement process does not have such relationship between problem analysis and solution assessment appliances; so, the analyst job involves subjectivity when a software solution is developed. For this reason, in this paper we propose a use cases performance in organizational problem solving method. Our main goal is allowing traceability relationship application to the appliances involved, in order to make a more-objective assessment of the solution.

KEYWORDS

Requirements elicitation, Solution assessment, Use cases, Traceability

INTRODUCCIÓN

Toda organización tiene objetivos que alcanzar, pero, cuando se presenta una condición de desequilibrio en sus procesos, debe realizar un análisis profundo de la problemática para tomar las decisiones que permitan encontrar la solución más adecuada. Kepner y Tregoe [8] proponen una metodología que establece una relación secuencial entre la definición de los problemas, las decisiones y el planteamiento de objetivos de una solución.

En el método de desarrollo de software UN-Método [1], en la etapa de educación de requisitos, se propone el análisis de los objetivos y de la problemática de la organización.

En dicho método, los objetivos se representan con el diagrama de KAOS [5] y la problemática con el diagrama causa-efecto [7]. El método pretende determinar la manera como los objetivos y los problemas influyen en los procesos de la organización, para luego proceder a analizar la posible solución informática, la cual se modela mediante los casos de uso. Así, el analista determinará qué problemas deberá atacar en su totalidad, cuáles podrá omitir o cuáles atacar parcialmente, sin alejarse del objetivo general que justifica la creación del software [1]. Sin embargo, aunque UN-Método procura el análisis del impacto de la aplicación de software sobre los problemas de la organización, la asignación del porcentaje de actuación de los casos de uso sobre los problemas se realiza subjetivamente, y no toman en consideración otros artefactos que también se afectan con la solución software, lo cual se reflejaría en un desfase a la hora de realizar la valoración final de la propuesta de solución, afectando, consecuentemente, la consistencia y trazabilidad de los modelos.

Por las razones antes anotadas, y como complemento a los trabajos de [14] y [1], en este artículo se propone un método para determinar el porcentaje de actuación que los casos de uso tienen sobre las causas de los problemas, de manera que se realice una valoración más objetiva del impacto de la aplicación de software. El método propuesto no sólo tiene en cuenta las ponderaciones de la problemática, sino que también se aplica la relación de trazabilidad que se puede establecer entre los procesos, los problemas que los afectan y la estructura jerárquica de los objetivos de la organización.

Este artículo se organiza de la siguiente manera: en la sección 1 se presenta el marco teórico, en donde se revisa tanto la relación que existe entre objetivos, problemas y procesos, como los conceptos involucrados en la valoración de la solución software. En la sección 2,

se presentan los antecedentes, en donde se hace una revisión de algunos trabajos que procuran una mejor trazabilidad entre los casos de uso y otros artefactos propios del modelado de soluciones informáticas. En la sección 3 se propone el método para determinar de una manera más objetiva la participación de los casos de uso en la solución de problemas de una organización, tomando en consideración dicha relación de trazabilidad. Finalmente, en la sección 4 se presentan las conclusiones y el trabajo futuro.

CONTENIDO

1. MARCO TEÓRICO

1.1 TRAZABILIDAD ENTRE LOS PROCESOS ORGANIZACIONALES, LOS PROBLEMAS Y LOS OBJETIVOS.

Toda organización tiene metas que alcanzar y, de acuerdo con su nivel de abstracción, se pueden clasificar en: objetivos de alto nivel o estratégicos, de bajo nivel o técnicos, funcionales, que corresponden a los procesos que realizan y no funcionales, que se asocian con la calidad de dichos procesos [11].

Para que exista un problema, se debe presentar una condición de desequilibrio total o parcial en los procesos del sistema analizado. Además, su impacto debe ser lo suficientemente significativo como para que el analista intervenga. En contraste, un objetivo se compone de elementos que enuncian un estado al que se debe llevar un proceso, mediante la toma de una serie de decisiones [8]. Por lo tanto, para tomar la decisión más acertada, se debe realizar un análisis profundo de los objetivos, teniendo en cuenta su clasificación e importancia, de la valoración del impacto del problema y de las acciones que permitan revertirlo, para alcanzar todos los objetivos de la organización y, finalmente, de otras situaciones adversas que puedan surgir ante la implementación de las nuevas medidas a tomar.

En la metodología Kepner-Tregoe se establece, entonces, una relación secuencial o de trazabilidad entre la definición de una solución, el análisis de los problemas, la toma de decisiones sobre los procesos de la organización que se afectan y el establecimiento de los objetivos de dicha solución [8], sin olvidar que el analista es el encargado de determinarla, obedeciendo a la jerarquía de objetivos.

En el método de desarrollo de software UN-Método [1], se destaca la importancia de trabajar con los objetivos identificados para la organización y se realizan paulatinas derivaciones a los objetivos para obtener subobjetivos, hasta llegar a las expectativas y requisitos, tal y como

se exige en la estructura del diagrama de objetivos de KAOS [5]. De este modo, se trata de evitar la información irrelevante, separando la estable de la volátil, buscando explicar al interesado los requisitos necesarios para solucionar una situación problemática, con la menor ambigüedad posible [12]. Dicha problemática se analiza, también, respecto a la manera como afectan los problemas a los procesos que se llevan a cabo en la organización y que se representan de manera jerarquizada y estructurada en el diagrama causa-efecto o Espina de Pescado [7]. Así, el analista determinará qué problemas deberá atacar en su totalidad, cuáles podrá omitir o cuáles atacar parcialmente, sin alejarse del objetivo general que justifica la creación del software [1].

1.2 EL PROCESO DE EDUCIÓN DE REQUISITOS.

El desarrollo de software suele ser una tarea compleja y en la que la ingeniería poco influye. Por ello, si se realiza una retrospectiva, se encuentran más errores que éxitos [1]. Sin embargo, con el surgimiento de la Ingeniería de Software, aparecen diversas corrientes dedicadas a la planificación estructurada y al modelado de los problemas, cuyo propósito es introducir cierta organización a dicho proceso de desarrollo, permitiendo, a su vez, la capacidad de mantenimiento y trazabilidad en sus productos.

El desarrollo de software, generalmente, inicia con una serie de encuentros entre el analista y el interesado, en los cuales se establece un vocabulario común para describir las funciones de los actores de la organización, la forma de proceder, la jerarquía de objetivos, entre otros [1]. Las reuniones periódicas con el interesado continúan para identificar el dominio del problema, los procesos que se afectan y los objetivos que los justifican (plasmándolos en los diferentes diagramas y artefactos) y determinar, así, los aspectos más relevantes que conforman y justifican la solución. Sin embargo, es posible que, después de esa ardua tarea, se concluya que una aplicación de software no aporta una mejora tan significativa como se esperaba, ya que ésta puede requerir un cambio previo tanto en la estructura, como en los procesos de la organización o que, incluso, no se requiera tal aplicación de software [1]. Una aplicación de software se modela con un conjunto de artefactos, y, por ello, se hace necesario articularlos de tal manera que los resultados sean consistentes, ya que, de lo contrario, el analista tendrá que concentrarse en corregir errores cuyo origen está en las fases iniciales del desarrollo de software y, de este modo, se desperdicia tiempo y esfuerzo [16].

En vista de que la identificación de los requisitos se realiza de manera subjetiva, en muchas ocasiones, esto conlleva a que la especificación analizada de la solución difiera de la situación problemática real, lo

que puede provocar inconsistencias en el sistema desarrollado. Entre los factores que pueden ocasionar estas imprecisiones en los requisitos se destacan los siguientes [2]:

- Interesados que tienen una visión parcial del negocio, conllevando a que los analistas no tengan una idea clara ni consistente del mismo.
- Analistas centrados en especificar los requisitos a través de casos de uso, objetivos o discursos de interesado con los que detallan las funcionalidades, pero que pasan por alto parte de la información de las relaciones entre requisitos o de estos con su entorno.
- Cada artefacto modela una porción específica del dominio con sus propias características, lo que aumenta la posibilidad de que se modelen elementos comunes. Es aquí donde aparece la necesidad de consistencia [16].

1.3 DIAGRAMAS DE CASOS DE USO

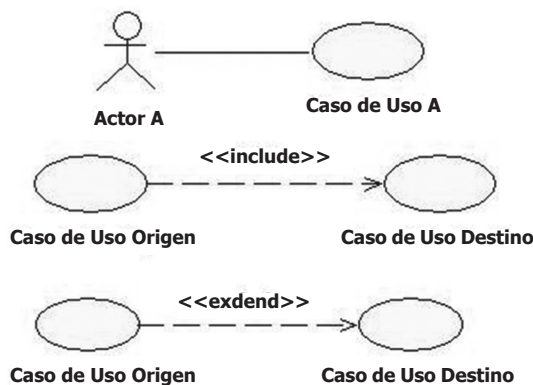
Los diagramas de casos de uso pertenecen al estándar de UML y se emplean para modelar aspectos funcionales del sistema, ya que facilitan la delimitación y el establecimiento de las relaciones entre los actores y el sistema [13]. Un caso de uso se puede definir como una secuencia de acciones, incluyendo acciones alternas, que el sistema puede realizar y que producen un resultado concreto para un actor que interactúa con el sistema [3]. Los siguientes elementos corresponden a la especificación del diagrama de casos de uso [4]:

- Actores: roles que los usuarios, ya sean humanos u otros sistemas, desempeñan respecto del sistema para comunicarse con el mismo.
- Escenario: secuencia específica de acciones que refleja un comportamiento.
- Relaciones: interacciones entre casos de uso y entre actores. Las relaciones pueden ser de cuatro tipos:
 - Asociación: se presenta entre los actores y los casos de uso.
 - Inclusión: ocurre cuando un comportamiento descrito por el caso de uso destino se incluye también en una instancia del caso de uso origen.
 - Extensión: se presenta cuando el caso de uso a extender invoca el caso de uso base de acuerdo con ciertas condiciones.
 - Herencia: ocurre cuando un actor o un caso de uso origen heredan la especificación del actor o caso de uso destino, respectivamente, y probablemente la modifiquen.

Las principales ventajas de utilizar el diagrama de casos de uso son [15]:

- permiten el buen desarrollo del sistema gracias a la captura de los requisitos desde la perspectiva del usuario.
- ofrecen una buena base para la verificación y validación de los requisitos del sistema.
- ayudan a gestionar la complejidad de proyectos grandes, descomponiéndolos en funciones más simples.

Figura 1: Diagrama de casos de uso para modelar aspectos funcionales del sistema.



2. ANTECEDENTES

2.1 ALGUNAS CONSIDERACIONES SOBRE LOS CASOS DE USO

Aunque el éxito de los casos de uso radica en que constituyen una técnica intuitiva [6], la obtención y la especificación de casos de uso realmente útiles y la falta de consenso sobre cómo tratarlos, son las razones que hacen necesario establecer una guía para la identificación, descripción y organización de ellos.

Los requisitos, y por tanto los casos de uso, se deberían organizar de manera jerárquica y cada nivel de casos de uso debe refinar los del nivel inmediatamente superior. Además, la jerarquía de casos de uso no debe ser el resultado de una descomposición funcional, sino que se debe realizar de manera iterativa e incremental [10].

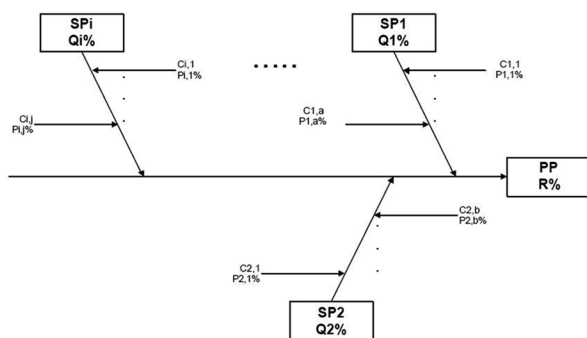
Otro aspecto a tener en cuenta es la ubicación del modelado de casos de uso dentro del proceso de desarrollo. Se entiende el modelado de casos de uso como un paso previo al modelado conceptual. En contraste, no es posible crear casos de uso adecuados y útiles sin comprender el dominio, y, por tanto, el modelado de casos de uso y el modelado conceptual deberían ser actividades paralelas [9].

Aunque los trabajos aquí revisados manifiestan la preocupación por darle un trato más consistente a las relaciones de los casos de uso con otros artefactos necesarios para modelar una aplicación de software (como el diagrama de objetivos), no tienen en cuenta los problemas de la organización, ni mucho menos la manera en cómo la solución informática modelada con los casos de uso los atiende.

2.2 ACTUACIÓN DE LOS CASOS DE USO SOBRE LOS PROBLEMAS ORGANIZACIONALES

En [14] y, posteriormente, en [1], se propone un método para la valoración de la aplicación de software desde las causas que se cubren por medio de los casos de uso de la solución informática. Para ello, se emplea como base el diagrama Causa-Efecto [7], pero, en este caso particular, con pesos ponderados para cada una de las causas y subproblemas del dominio, cumpliendo con lo siguiente (véase la figura 2):

Figura 2: Esquema general del diagrama Causa-efecto para la representación de problemas.



$$\sum_{i=1}^n Q_i * \left(\sum_{j=1}^m P_{i,j} \right) = 100\% \tag{1}$$

$$\forall i \sum_{j=1}^m P_{i,j} = 100\% \tag{2}$$

En estas ecuaciones, Q_i representa el porcentaje de relevancia del problema SP_i sobre el problema principal; P_{ij} es el porcentaje de relevancia de la causa C_{ij} en el problema SP_i . El método de valoración consiste en asignar un porcentaje de actuación A_{ij} que una función de un caso de uso tiene sobre una causa determinada. En este caso, si \tilde{n} es el conjunto de funciones asociadas

con un caso de uso y ACU el porcentaje de atención del caso de uso sobre el problema, se debe cumplir también que:

$$\%ACU = \sum_{k=1}^{\tilde{n}} \left(\sum_{i=1}^n Q_i * \left(\sum_{j=1}^m P_{i,j} * A_{i,j} \right) \right) \tag{3}$$

$$\forall i,j \sum_{k=1}^{\tilde{n}} A_{i,j} \leq 100\% \tag{4}$$

La aplicación de este método supone la elaboración de una tabla que incluye el listado de las funciones de los casos de uso, con sus respectivos porcentajes de cumplimiento, junto con las ponderaciones del diagrama causa-efecto (véase la tabla 1).

Tabla 1: Asignación de porcentajes para la valoración total de aplicaciones de software.

CASO DE USO	CAUSA	P_{ij}	Q_{ij}	A_{ij}	ACU
CdeU	SP5	0.1	0.1	0.1	0.01
...
TOTAL					X

Sin embargo, este método indica que no todos los problemas requieren soluciones ligadas con la informática, pero, de todas formas, se podrían estudiar desde el punto de vista de la Ingeniería de Sistemas [1]. Además, sería necesario realizar la valoración sobre el conjunto seleccionado de soluciones, para presentarle al interesado con el fin de mirar su impacto sobre la organización.

Aunque en UN-Método se realiza un análisis del impacto de la aplicación de software sobre los problemas de la organización, desde los casos de uso, la valoración del porcentaje de actuación de la solución informática aún la realiza el analista de manera subjetiva. Además, se dejan de lado artefactos como los procesos y los objetivos de la organización, de gran influencia en la educación de requisitos, representando un posible desfase respecto del impacto real sobre la problemática, en el momento de realizar la valoración total de la aplicación de software.

3. DESARROLLO

3.1 ESTIMACIÓN EN UN-MÉTODO DEL PORCENTAJE DE ACTUACIÓN DE LOS CASOS DE USO SOBRE LOS PROBLEMAS

Tras superar la etapa de la educación de requisitos, la aplicación de software resultante habrá de tener un impacto sobre la organización, el cual [14] y [1] calculan desde el punto de vista de las causas de los problemas que soluciona total o parcialmente, como se muestra a continuación. Supóngase la "nueva tabla explicativa de procesos" resultante en el tercer entregable de un determinado proyecto realizado con UN-Método y que se muestra en la Tabla 2. Por simplicidad, solamente se muestran las columnas relevantes para la demostración y los códigos de los enunciados de los respectivos artefactos. En ese orden de ideas, P_i es el código del proceso i que se lleva a cabo en una organización; OB_i y RE_i son los respectivos objetivos y requisitos asociados con el proceso i ; C_i es la causa o problema que afecta al respectivo proceso. Adicionalmente, se añade la columna "casos de uso" para especificar aquellos procesos que se considera se pueden automatizar en los casos de uso $CdeU_i$ que atienden dichas causas.

De este modo, el porcentaje de actuación de los casos de uso sobre las causas que atienden, se calcula empleando la tabla 1 que proponen [14] y [1], determinando el porcentaje A_{ij} como se muestra en la tabla 3. Esta manera de asignar porcentajes de cumplimiento la realiza el analista intuitivamente basándose, la mayoría de las ocasiones, en la cantidad de ocurrencias de una determinada causa en la nueva tabla explicativa de procesos.

Tabla 2: Nueva tabla explicativa de procesos para presentar los procesos automatizables.

PROCESOS	OBJETIVOS	PROBLEMAS	CASO DE USO
P1	OB1		
P2	OB2	C3	CdeU1
P3	OB5,RE4	C3	CdeU1
P4	RE1, RE9	C1	CdeU1
P5	RE1, RE5	C1	
P6	OD3, RC2	C5	CdeU2
P7	RE2, RE6	C5	
P8	OB4		CdeU3

P9	OB6,RE7	C2	CdeU3
P10	RE10	C2	
P11	RE3		
P12	RE8	C2	

Tabla 3: Asignación tentativa de porcentajes de actuación para cada una de las causas.

PROBLEMA	A_{ij} %
C1	0
C2	33.3
C3	100
C4	100
C5	50

Como se observa en la tabla 2, la causa C1 no se asocia con ningún proceso automatizable, por lo que no se le asigna porcentaje de actuación. Las causas C3 y C4 se encuentran presentes en procesos que se automatizan completamente con casos de uso, por lo que se les asigna 100% de actuación. Pero, ahora, como sólo uno de los procesos (P9) que contiene la causa C2 se asocia con un caso de uso (CdeU3), dicha causa se atiende de manera parcial; así, se supone que su porcentaje de actuación sería de tan sólo una tercera parte, es decir, se le asignaría el 33.3%, si los objetivos de la organización tuvieran igual importancia. De forma análoga, se podría deducir que la causa C5 se atendería al 50%, pues uno de los procesos con los cuales se asocia (P7) no se incluye como un caso de uso en la solución informática. La valoración total para esta metodología se presenta en la tabla 4. Supóngase, además los porcentajes de incidencia P_{ij} y Q_{ij} , los cuales deben estar presentes en el diagrama Causa-efecto, como se indica en la Figura 2.

Tabla 4: Valoración total de la aplicación de software para determinar su impacto sobre la problemática.

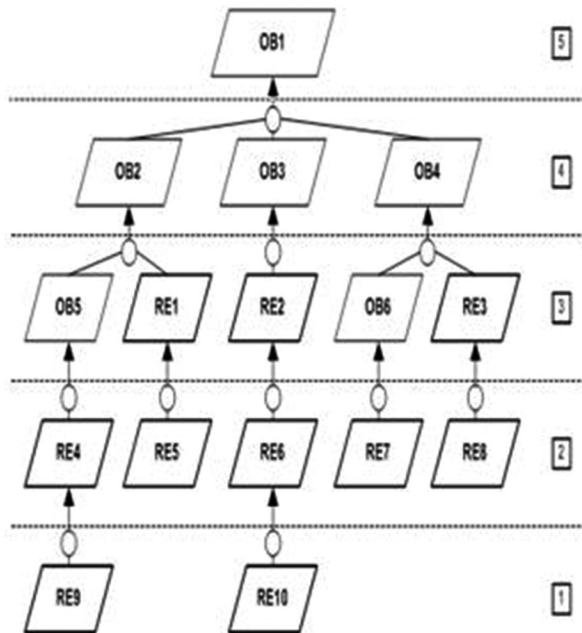
CASO DE USO	CAUSA	P_{ij}	Q_{ij}	A_{ij}	ACU
CdeU1	C3	0,25	0,43	1	0,1075
CdeU1	C4	0,2	0,375	1	0,075
CdeU2	C5	0,16	0,52	0,5	0,0416
CdeU3	C2	0,2	0,28	0,333	0,0186
TOTAL					0,2427 = 24,3%

En este método, la valoración total puede presentar un desfase respecto del impacto real sobre la problemática, puesto que, si bien se emplea un razonamiento matemático para definir tal efecto, el porcentaje de actuación A_{ij} para las causas que se atienden parcialmente (como C2 y C5), lo asigna el analista según su criterio, es decir, se hace subjetivamente. Además, para este cálculo, no se tienen en cuenta otros elementos vitales en la educación de requisitos, como los objetivos y los procesos de la organización (alterando la consistencia y trazabilidad de este método), quienes, finalmente, también se afectan con la introducción de la aplicación informática en la organización. Así, como se argumentó en la sección 2.2, se puede establecer una relación entre la definición de una solución, el análisis de los problemas, las decisiones sobre los procesos y el establecimiento de los objetivos de la misma [8].

3.2 MÉTODO PARA DETERMINAR LA PARTICIPACIÓN DE LOS CASOS DE USO EN LA SOLUCIÓN DE PROBLEMAS DE UNA ORGANIZACIÓN.

A continuación, se propone un método para determinar el porcentaje de actuación A_{ij} sobre los problemas que se atienden parcialmente con los casos de uso. Para ello, supóngase que los objetivos de la tabla 2, se jerarquizan de acuerdo con los pesos del diagrama de KAOS [5] de la Figura 3. Por simplicidad, se omiten los enunciados de los objetivos y los requisitos:

Figura 3: Diagrama KAOS para representar los objetivos de la organización.



Ahora, teniendo en cuenta la manera como influyen los procesos que una causa determinada afecta y la jerarquía de los objetivos asociados, el nuevo porcentaje de actuación A_{ij} de los casos de uso sobre las causas que atienden de manera parcial, se calcula realizando la sumatoria de los pesos de los objetivos involucrados, tras realizar el análisis de la tabla 5.

Tabla 5: Determinación de los pesos de los objetivos que se afectan con la causa C2.

CAUSA	CASO DE USO	PROCESOS AFECTADOS	OBJETIVOS	PESOS	SUB-TOTAL
	CdeU3	P9	OB6, RE7	3+2	5
C2		P10	RE10	1	1
		P12	RE8	2	2
			TOTAL		8

Como se mencionó anteriormente, el caso de uso CdeU3 atiende únicamente la causa C2, implicando que solamente uno de los tres procesos a los que afecta (el proceso P9) es automatizable. Por lo tanto, el total de los pesos correspondientes a los respectivos procesos afectados es de 8, de los cuales sólo 5 se atienden, luego (por regla de tres simple), el porcentaje de actuación A_{ij} de la causa C2 es del 62,5%, y no del 33.3% como se había estipulado inicialmente. Cabe destacar que, para este caso particular, con la asignación original se desprecia casi la mitad del porcentaje de actuación del caso de uso.

Análogamente, el análisis para la causa C5 se muestra en tabla 6, indicando que de los 12 pesos afectados, 7 se atienden. De este modo, su porcentaje de atención A_{ij} es del 58,3%, y no del 50%.

Tabla 6: Determinación de los pesos de los objetivos que se afectan por la causa C5.

CAUSA	CASO DE USO	PROCESOS AFECTADOS	OBJETIVOS	PESOS	SUB-TOTAL
	CdeU2	P6	OB3, RE2	4+3	7
C5		P7	OB2, RE6	3+2	5
			TOTAL		12

A partir de la valoración final, con los nuevos porcentajes de atención, se aprecia que, si se tiene en cuenta el impacto de los casos de uso únicamente sobre la

problemática específica, se estaría despreciando cierto porcentaje de actuación, que puede ser importante a la hora de determinar impacto total.

Tabla 7: Valoración final del impacto de la pieza de software con los nuevos porcentajes de cumplimiento.

CASO DE USO	CAUSA	P_{ij}	Q_{ij}	A_{ij}	ACU
CdeU1	C3	0,25	0,43	1	0,1075
CdeU1	C4	0,2	0,375	1	0,075
CdeU2	C5	0,16	0,52	0,583	0,0485
CdeU3	C2	0,2	0,28	0,625	0,035
TOTAL					0,2660 = 26,6%

Para este caso particular, la valoración final es del 26,6%, es decir, 2,3% más que la valoración originalmente realizada (24,3%), que se muestra en la Tabla 4. Se encuentra este valor dado que los objetivos de las organizaciones no se consideran de igual importancia, sino que hay unos más estratégicos que otros, por lo general aquellos que se encuentren más altos en la jerarquía que plantea el diagrama de objetivos.

4. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presentó un método para establecer una relación mediante la cual se integran ciertos artefactos que se construyen durante el proceso de educación de requisitos, como los procesos que se afectan con las causas de los diferentes problemas y los objetivos por los cuales se llevan a cabo (teniendo en cuenta su jerarquía). Este método permite que se realice un análisis más realista del impacto de la solución informática sobre los problemas que se atacan parcialmente.

Al mismo tiempo, se elimina la intervención subjetiva del analista a la hora de hacer la valoración del porcentaje de actuación de los casos de uso sobre la solución de los problemas que afectan a la organización, mejorando, en ese sentido, la consistencia y trazabilidad de los requisitos que se capturan por medio de UN-Método.

Como trabajo futuro a desarrollar en relación con este tema, se encuentran las siguientes líneas de investigación:

- Desarrollar una metodología que establezca un criterio para tratar de cuantificar el grado de automatización de un proceso, con el fin de determinar la certeza de asignar el 100% de actuación de un caso de uso sobre un proceso problemático.
- Proponer un método formal para definir los pesos de los objetivos presentes en el diagrama de KAOS, de manera que no necesariamente se considere el aumento lineal, a medida que se asciende en la jerarquía de objetivos.
- Desarrollar un método más objetivo para establecer el porcentaje de gravedad de los problemas, es decir, para determinar el grado de importancia de la causa que se atiende, evitando que el analista se base únicamente en su propio criterio.
- Proponer un método que permita establecer equivalencias entre los casos de uso y los requisitos que ellos cubren, de tal manera que se pueda determinar una relación más directa entre los mismos, a la hora de elegir la solución informática que se va a implementar.

5. REFERENCIAS BIBLIOGRAFICAS

- [1] ARANGO, Fernando y ZAPATA, Carlos M. UN-MÉTODO para la Elicitación de Requisitos de Software. Medellín: Carlos M. Zapata (Ed.), 2006. 100p.
- [2] BERROCAL, Javier, GARCÍA, José Manuel, MURILLO, Juan Manuel. Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio. En: II TALLER DE PROCESOS DE NEGOCIO E INGENIERÍA DE SERVICIOS (II: 2009: San Sebastián). Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, San Sebastián: 2009.
- [3] BOOCH, G., RUMBAUGH, J., JACOBSON, I. The Unified Modeling Language User Guide. New York: Addison-Wesley Professional, 1999. 512p.
- [4] FOWLER, M. A brief guide to the Standard Object Modeling Language. Redwood City: Addison-Wesley, 2004. 185p.
- [5] DARDENNE, Anne, LAMSWEERDE, Axel, FICKAS, Stephen. Goal-Directed Requirements Acquisition. Science of Computer Programming, Vol. 20, Issue 1-2. Ámsterdam: Elsevier North-Holland, Inc. 1993.
- [6] GARCÍA, Jesús, ORTÍN, M. José, MOROS, Begoña, NICOLÁS, Joaquín, TOVAL, José. De los procesos del negocio a los casos de uso. Técnica Administrativa (revista electrónica), Vol. 6, No. 32. Buenos Aires: Ciencia y Técnica Administrativa C&TA, 2007.

- [7] ISHIKAWA, Kaoru. Guide to quality control. Tokyo: Asian Productivity Organization, 1986. 225p.
- [8] KEPNER, Ch. and TREGOE, B. The New Rational Manager: An Updated Edition for a New World. Princeton: Princeton Research Press, 1997. 254p.
- [9] KORSON, T. Constructing Useful Use Cases [en línea]. Consultado el 5 de marzo de 2010. Disponible en: <http://softwarearchitects.com/publications/korson/usecase3>. 1999.
- [10] KORSON, T. Misuse of Use Cases [en línea]. Consultado el 5 de marzo de 2010. Disponible en: <http://softwarearchitects.com/publications/korson/korson9803om.htm>. 1998.
- [11] LAMSWEERDE, Axel. Requirements engineering in the year 2000: a research perspective. En: PROCEEDING OF THE 22ND INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, (XXII: 2000: Limerick), 5-19.
- [12] LAMSWEERDE, Axel. (2001). Goal-oriented Requirements Engineering: A guided tour. En: 5TH IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING. (V: 2001: Toronto), 249-262.
- [13] OMG. Object Management Group. Unified modeling language specification [en línea]. Consultado el 15 de marzo de 2010. Disponible en: <http://www.omg.org/UML/>
- [14] ZAPATA, Carlos y ARANGO, Fernando. Alineación entre metas organizacionales y elicitación de requisitos del software. Revista Dyna Vol. 71, No. 143, pp. 101 - 110. Medellín: Universidad Nacional de Colombia, Noviembre de 2004.
- [15] ZAPATA, Carlos M., TAMAYO, Paula, ARANGO, Fernando. Conversión de esquemas preconceptuales a diagrama de casos de uso empleando atom3. Revista Dyna Vol. 74, No. 153, pp. 237-251. Medellín: Universidad Nacional de Colombia, Noviembre de 2007.
- [16] ZAPATA, Carlos M., VILLEGAS, Sandra M., ARANGO, Fernando. Reglas de consistencia entre modelos de requisitos de UN-Método. Revista Universidad EAFIT Vol. 42, No. 141, pp. 40-59. Medellín: Universidad EAFIT, 2006.