

LA TRAZABILIDAD EN EL PROCESO DE REQUERIMIENTOS DE SOFTWARE

Master Maria Marta Sandoval Carvajal, PMP
Universidad Nacional,
Escuela de Informática.
Heredia Costa Rica
msandova@una.ac.cr

Master Maria Adilia García Vargas
Universidad Nacional
Escuela de Informática
Heredia Costa Rica
mgarcia@una.ac.cr

ABSTRACT

La necesidad de mejorar el proceso relacionado con los requerimientos es evidente desde hace varios años. Uno de los aspectos que facilitan una mejora, es contar con la posibilidad de rastrear requerimientos a lo largo del desarrollo de un sistema.

Asimismo, la trazabilidad permite evaluar las implicaciones que puede tener el cambio de uno de los requisitos, en etapas tempranas o incluso en la etapa de implementación, de forma bidireccional, desde su origen hasta su implementación y viceversa.

A través de este artículo se presentan y discuten, varias definiciones fundamentales dadas por algunos autores acerca de diferentes temas conexos al concepto de la trazabilidad y su utilidad en el proceso de requerimientos.

Luego de esto, se presenta un caso de estudio que muestra cómo se puede dar seguimiento a un requisito desde la etapa de análisis, hasta la etapa de implementación. Esto dentro de un proceso de desarrollo de software. Para efectos de este ejemplo, se utilizará la metodología del Proceso Unificado (PU) apoyado por el software RUP™ (Rational Unified Process), específicamente la herramienta denominada Rational Requisite Pro™.

Palabras clave: trazabilidad, requerimiento, Proceso Unificado, RUP, UML.

1. INTRODUCCIÓN

A través del tiempo, muchos de los sistemas de software desarrollados, han fracasado debido a que en el momento en que se entregan, ya no cumplen los requisitos deseados por la empresa que contrató. En otros casos son utilizados solamente por poco tiempo, lo cual es una pérdida para la empresa, considerando la inversión realizada. En otros casos se ha tratado de ajustar el nuevo sistema a cambios que ha sufrido la empresa. Para esto se agregan una serie de soluciones poco efectivas, que han generado la desintegración de los diferentes subsistemas del proyecto provocando que se deje de utilizar el sistema, o peor aún, se llega a la conclusión que el sistema desarrollado no se puede ajustar a los nuevos requerimientos, concluyendo que es más rentable desecharlo y desarrollar uno nuevo que hacer adaptaciones al nuevo sistema.

Se pueden nombrar muchos de estos casos, los cuales tienen un problema en común: “Los sistemas no cumplen con los requerimientos”. Debido a esto, se le ha dado atención especial a esta, usualmente amplia fase del desarrollo de un sistema conocida como “la gestión de los requisitos”.

Por el especial interés que tienen los requisitos en este proceso, el presente trabajo se enfoca a algunas definiciones importantes introductorias sobre el tema, para luego, presentar el concepto de trazabilidad, el cual debe de ser utilizado desde la concepción de un sistema hasta la generación del código.

2. ¿QUÉ ES REQUISITO?

Son muchas las definiciones que se encuentran acerca de lo que es un requisito o requerimiento. En el SEWBOK [SEWBOK04], se define un requerimiento como la propiedad que resuelve un problema en el mundo real.

“... es una condición o capacidad a la que el sistema (siendo construido) debe conformar. puede ser definido como: Una capacidad del software necesaria por el usuario para resolver un problema o alcanzar un objetivo... Una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal”. [ZULOAGA96]

“Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes: **Necesario:** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso. **Conciso:** Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro. **Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión. **Consistente:** Un requerimiento es consistente si no es contradictorio con otro requerimiento. **No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector. **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.”

En síntesis, existen necesidades y solicitudes de parte de una Institución, empresa o un usuario, que deben ser resueltas por un sistema o por un componente del mismo, lo que conlleva a citar los requerimientos del software y los requerimientos del usuario.

El asunto parece muy sencillo aunque estas definiciones envuelven incógnitas tales como: ¿se interpretaron bien las necesidades? ¿El sistema resuelve esas necesidades? ¿El usuario queda satisfecho con el sistema?, etc.

Por tal motivo, se definió dentro de la Ingeniería de Sistemas, una sub-área denominada Administración de los Requerimientos. Algunos también lo conocen como Ingeniería de Requerimientos. [HERRERA00]

3. ¿QUÉ ES LA ADMINISTRACIÓN DE LOS REQUERIMIENTOS?

Al igual que en el caso de los requisitos, existen muchas definiciones de lo que es la Administración de los Requerimientos. Entre ellas podemos citar:

“... un enfoque sistemático para obtener, organizar y documentar los requisitos del sistema, así como mantener un acuerdo entre el cliente y el equipo del proyecto en los cambios a los requisitos. Debemos considerar que la clave para una efectiva administración de requisitos es:

Mantener un enunciado claro de los requisitos, junto con atributos para cada tipo de requisitos y su seguimiento con otros requisitos o elementos del proyecto.” [TORRES02]

“... una forma sistemática de obtener, organizar y documentar los requisitos de un sistema, y un proceso que establece y mantiene el acuerdo entre el cliente y el equipo de proyecto sobre los requisitos cambiantes.” [BAUFEST03].

“...RUP describe cómo: obtener los requisitos, organizarlos, documentar requisitos de funcionalidad y restricciones, rastrear y documentar decisiones, captar y comunicar requisitos del negocio” [GUERRERO03]

4. Tipos de requerimientos

Según la metodología que se utilice, existen varias clasificaciones de los tipos de requerimientos, entre los más usuales están los requerimientos de software y los requerimientos de hardware.

4.1 Requerimientos de software

Se refiere a lo que debe hacer y lo que no debe hacer el sistema, se pueden subdividir en requerimientos funcionales y requerimientos no funcionales.

4.2 Requerimientos de hardware

Dependen del volumen de operaciones, cantidad de usuarios y de las aplicaciones que deben de estar en uso cuando se implemente el sistema. Se pueden distinguir tipos de requerimientos de hardware, requerimientos de rendimiento, de interfaz, de ambiente.

5. CLASIFICACIÓN DE LOS REQUERIMIENTOS EN LA METODOLOGÍA DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (PU)

Existen varias clasificaciones de los requerimientos, una de ellas es la del autor Pressmann [PRESSMANN02] el cuál tomó como base el modelo FURPS (acrónimo en inglés dado a los factores de calidad que caracterizan a algunos requerimientos). Este modelo fue desarrollado por Hewlett - Packard para representar los requerimientos según la calidad y los atributos que deben tener. Este modelo se muestra en la Tabla 1. Clasificación de los requerimientos de software.

Tabla 1. Clasificación de los requerimientos de software

Factor de Calidad	Atributos
Funcionalidad	<ul style="list-style-type: none"> ✓ Características y capacidades del programa ✓ Generalidad de las funciones ✓ Seguridad del sistema
Facilidad de uso	<ul style="list-style-type: none"> ✓ Factores humanos ✓ Factores estéticos ✓ Consistencia de la interfaz ✓ Documentación
Confiabilidad	<ul style="list-style-type: none"> ✓ Frecuencia y severidad de las fallas ✓ Exactitud de las salidas ✓ Tiempo medio de fallos ✓ Capacidad de recuperación ante fallas ✓ Capacidad de predicción
Rendimiento	<ul style="list-style-type: none"> ✓ Velocidad del procesamiento ✓ Tiempo de respuesta ✓ Consumo de recursos ✓ Rendimiento efectivo total ✓ Eficacia
Capacidad de Soporte	<ul style="list-style-type: none"> ✓ Extensibilidad ✓ Adaptabilidad ✓ Capacidad de pruebas ✓ Capacidad de configuración ✓ Compatibilidad ✓ Requisitos de instalación

6. CLASIFICACIÓN DE REQUERIMIENTOS DE SISTEMAS DE SOFTWARE

Los requerimientos de un sistema pueden ser funcionales o no funcionales.

6.1 Requerimientos funcionales

Declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares y de cómo se comportará en situaciones particulares.

Pueden declarar lo que el sistema no debe hacer. Una falla de un requerimiento funcional degrada un sistema.

Estos requerimientos dependen del tipo de software, como por ejemplo para un sistema que imprime cheques semanales de pago, los requerimientos funcionales deben responder a preguntas como:

1. ¿Qué entrada es necesaria para que un cheque se imprima?
2. ¿Bajo que condiciones puede cambiar el monto del pago?
3. ¿Qué provoca la remoción de un empleado de la nómina de pago?

En un sistema de biblioteca, algunos tipos de requerimientos son los siguientes:

1. El usuario deberá tener la posibilidad de buscar en el conjunto inicial de la base de Datos o seleccionar un subconjunto de ella.
2. El sistema deberá proveer visores adecuados para que el usuario lea documentos en el almacén de documentos.
3. A cada pedido se le deberá asignar un identificador único que el usuario podrá copiar al área de almacenamiento permanente de la cuenta.

6.2 Requerimientos no funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entregará el sistema, sino a las propiedades emergentes de éste tales como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, etc.

Definen las restricciones del sistema como la capacidad de dispositivos de E/S y la representación de datos que se utiliza en las interfaces del sistema.

En general se refieren al sistema como un todo y no a rasgos particulares, por lo tanto, una falla de un requerimiento no funcional, podría inutilizar un sistema.

Ejemplos de este tipo de requerimiento son:

- 1- El sistema debe ser desarrollado sobre un computador X
- 2- Los estándares de desarrollo deben ser los que la organización tiene establecidos.

7. DIFICULTADES PARA DEFINIR LOS REQUERIMIENTOS

Además de que los requerimientos no son obvios, ellos provienen de muchas fuentes. Algunos de ellos son difíciles de expresar en palabras (el lenguaje es ambiguo).

Existen muchos tipos de requerimientos y diferentes niveles de detalle por lo que la cantidad de requerimientos en un proyecto puede ser difícil de manejar. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.

Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.

“Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas. Un requerimiento puede cambiar a lo largo del ciclo de desarrollo. Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto [MOREA00]”.

En la figura 2, el autor Zuloaga [ZULOAGA96] describe lo que sucede desde la captura de los requerimientos hasta el desarrollo de un sistema de software, en donde están involucrados los usuarios, los analistas y los programadores.

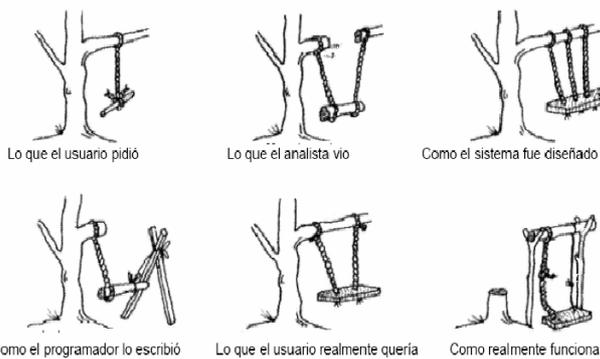


Figura 2. Diferentes interpretaciones desde los puntos de vista de los involucrados en el desarrollo de un sistema de software

En la figura 3, se puede observar un estudio realizado durante el año 1995, acerca de los resultados obtenidos en los proyectos de software el cual fue impactante dado por el autor Pressman [PRESSMAN02]:

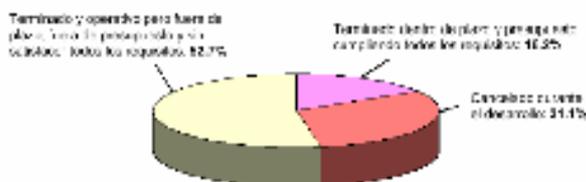


Figura 3. Estadísticas de sistemas de software en el año 1995

Según este estudio, el éxito de los proyectos se debe a una buena intervención de los usuarios, un buen apoyo de los directivos y a un enunciado claro de los requisitos. Por otro lado el fracaso se da cuando los usuarios no transmiten

algunos requisitos, a algunas especificaciones de requisitos que no fueron comprendidas y a falta de algunos requisitos.

Si sumamos a estos problemas que se han dado con los requisitos, el que no se tome en cuenta la relación de unos con otros y con otras partes del proceso, queda claro que es importante dar seguimiento al comportamiento de dichos requerimientos. Esto es particularmente necesario en el caso de un cambio en alguno de ellos o en el caso de añadir uno nuevo.

Para controlar el comportamiento y cambio en los requerimientos es necesario rastrearlos durante todo el desarrollo de un sistema, lo cual es posible utilizando la trazabilidad.

8. TRAZABILIDAD o RASTREABILIDAD

Debido a los acontecimientos citados anteriormente, con respecto a la dificultad que se ha tenido con el desarrollo de sistemas de software, el tema de la trazabilidad ha despertado un especial interés entre los Ingenieros de Software, los desarrolladores de estos sistemas y algunos usuarios.

Son muchas las definiciones que se dan acerca de este tema, pero casi todos tienen opiniones muy parecidas a las que se presentan a continuación:

“Una especificación de un requerimiento de software es trazable si

- (i) el origen de cada requerimiento está claro y
- (ii) si se facilita la referencia de cada requerimiento en el desarrollo futuro o en la documentación” [ANTON99]

“Es la capacidad de describir y de seguir la vida de un requisito, tanto en dirección hacia adelante y hacia atrás, es decir, desde sus orígenes, a través de su desarrollo y especificación, a su despliegue y uso subsecuentes, y a través de periodos de refinamiento y de la iteración en curso en cualesquiera de estas fases” [GOTEL94].

El autor Doorn [DOORN ET AL 07], en describe la trazabilidad desde el origen del requisito hasta la implementación de este requisito en el sistema y viceversa, es decir, hacia adelante y hacia atrás.

Esto implica que un requerimiento debe de ser rastreable desde que se define y durante todo el desarrollo del software, lo cual garantiza una adecuada administración del cambio con el fin de evaluar el impacto en el resto del sistema.

En el caso que se esté en la etapa de desarrollo de los requerimientos, se podrá evaluar como afectaría un cambio de un requisito en otro. Por otro lado, estando en etapa de implementación y en caso de que haya un cambio en los requerimientos, la trazabilidad permitirá hacer una evaluación en el diseño y en la implementación. Si el cambio se da cuando el sistema está implementado, la trazabilidad permitirá hacer una evaluación de cómo serán afectados los involucrados.

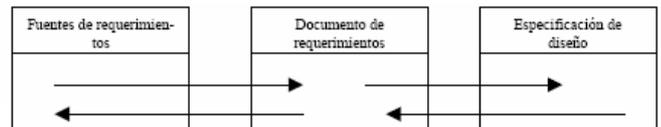


Figura 5. Relación que deben de tener los requerimientos en las diferentes etapas

En la figura 5, el autor Davis[DAVIS93] ilustra como el documento de requerimientos está muy ligado, en forma bidireccional, hacia las fuentes de requerimientos y la especificación de diseño. Así es como se establece una

estrecha relación y dependencia entre los requerimientos y el sistema. Esto se puede resumir de la siguiente manera:

“1. para diseñar o testear cualquier componente del sistema es necesario saber qué requerimientos satisface (aunque sea parcialmente)

2. Para probar el sistema software es necesario saber qué requerimientos están siendo validados en cada prueba”

Sommerville [SOM97] tiene una opinión similar a la de Davis y lo expresa de la siguiente manera: “La información de trazabilidad es la información que permite que se encuentren dependencias entre los requisitos, y entre los requisitos y los componentes del diseño del sistema y la documentación”

A la vez Sommerville [SOM97] expresa las razones del porqué utilizar trazabilidad de la siguiente manera:

“Se dispone de información en la evaluación de los cambios de requerimientos

-Son la base para el control de costos y calidad”

Para los autores mencionados anteriormente, la trazabilidad es de vital importancia ya que permite rastrear los requerimientos durante todo el desarrollo de un sistema. Además puede permitir a priori conocer cuáles serían las consecuencias de cambiar o eliminar un requerimiento en el sistema, lo que en estrategias carentes de trazabilidad, no se podía visualizar tan fácilmente.

Aplicar trazabilidad es un proceso delicado, ya que durante todo el desarrollo de un sistema, independientemente del modelo que se utilice, se deben de tomar en cuenta aquellos artefactos que permitan rastrear el requerimiento desde su nacimiento hasta su puesta en marcha y viceversa.

Por ejemplo, si se utiliza la metodología PU (Process Unified), apoyada por el software RUP (Rational Unified Process), se tomaría en cuenta los artefactos que se muestran en la figura 6, en la definición de los requerimientos [RATION00].

Requirements: Artifact Overview

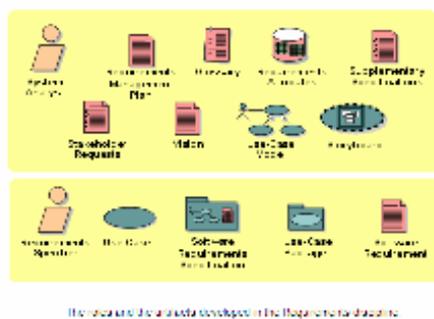


Figura 6. Los roles y los artefactos que se desarrollan en la disciplina de requerimientos

Existen herramientas desarrolladas con el fin de seguir la huella de los requerimientos, desde diferentes vistas. Tenemos por ejemplo el caso de la funcionalidad de Rational Requisite Pro. En la Figura No. 7, se puede observar una sesión típica de Requisite Pro conteniendo un caso desarrollado para la gestión de artículos deportivos [LÓPEZ03]. Se analizan las necesidades de los usuarios, las cuales se especifican mediante casos de uso. Luego se genera una matriz en la que se representa el requerimiento, la prioridad que tiene el caso de uso, el estado actual, el grado de dificultad y a quien ha sido asignado.

Requirements:	Prioridad	Estado	Dificultad	Asignado a
CUS1: Facturar Entrega Pedido	Baja	Propuesto	Media	
CUS2: Cobro Clientes	Baja	Propuesto	Media	
CUS3: Compra a Proveedores	Media	Aprobado	Media	
CUS4: Confeccionar Catálogo	Media	Aprobado	Media	
CUS5: Consultar Pedidos no Atendidos	Alta	Validado	Baja	José A. Mocholí, Eduardo Bueno
CUS6: Control Estadísticas	Media	Aprobado	Media	
CUS7: Consultar Catálogo	Media	Aprobado	Media	
CUS8: Entrevista Trabajo	Baja	Propuesto	Baja	
CUS9: Gestión Nóminas	Media	Aprobado	Media	
CUS10: Gestión de Personal	Baja	Propuesto	Media	
CUS11: Gestión de Regiones	Media	Aprobado	Media	
CUS12: Otorgar Incentivos	Baja	Propuesto	Baja	
CUS13: Política de Ventas	Baja	Propuesto	Baja	
CUS14: Reabastecer Almacén	Media	Aprobado	Media	
CUS15: Realizar Oferta	Media	Aprobado	Media	
CUS16: Redistribución de Personal	Media	Aprobado	Media	
CUS17: Atender Pedido	Alta	Incorporado	Media	José Antonio Mocholí, Eduardo Bueno Medina
CUS18: Cancelar Pedido Atendido	Alta	Incorporado	Media	
CUS19: Consultar Pedidos a Enviar	Alta	Incorporado	Baja	German Mira Rico
CUS20: Elaborar Pedido	Alta	Validado	Alta	German Mira y José A. Mocholí
CUS21: Elaborar Pedido On-line	Media	Aprobado	Alta	
CUS22: Gestión de Clientes	Media	Aprobado	Media	
CUS23: Incidencia Pedido	Media	Incorporado	Media	César López Rodríguez
CUS24: Introducir Recibos	Baja	Aprobado	Media	
CUS25: Pasar Pedido a Envío	Alta	Incorporado	Media	German Mira Rico
CUS26: Realizar Envío	Media	Aprobado	Media	
CUS27: Reposición de Stock	Media	Aprobado	Media	

Figura 7. Matriz de trazabilidad de los requerimientos de un sistema

En la figura 8, se puede observar la matriz de los involucrados en el sistema, la cuál brinda información acerca de cuál de los involucrados del sistema, hizo la solicitud de un determinado requerimiento, lo que permite saber a quién dirigirse en caso de tener alguna duda, problema o a quién debe de estar presente en caso de llevar a cabo alguna prueba de un requerimiento en particular. [LÓPEZ03]

Relationships: - direct only	CSW1: Dep. Departamento de Recursos Humanos	CSW2: Dep. Departamento de Marketing	CSW3: Dep. Departamento de Logística	CSW4: Gest. Gestión de Almacén	CSW5: Gest. Gestión de Ventas	CSW6: Gest. Gestión de Envíos	CSW7: Dep. Departamento de Contabilidad y Facturación
STK1: Logística							
STK2: Almacén							
STK3: Ventas							
STK4: Contabilidad / Facturación							
STK5: Marketing							
STK6: Envíos							
STK7: Recursos Humanos							

Figura 8. Matriz de trazabilidad de los requerimientos de un sistema

De igual manera, utilizando el Requisite Pro, se puede hacer una matriz de trazabilidad entre los actores y los casos de uso tal como se muestra en la figura 9, con el fin de tener conocimiento de cuál actor utiliza un caso de uso determinado, tal como lo muestra el autor López [LÓPEZ03], en el caso de estudio mencionado anteriormente.

Relationships: - direct only	CUS1: Facturar Entrega Pedido	CUS2: Cobro Clientes	CUS3: Compra a Proveedores	CUS4: Confeccionar Catálogo	CUS5: Consultar Pedidos no Atendidos	CUS6: Control Estadísticas	CUS7: Consultar Catálogo	CUS8: Entrevista Trabajo	CUS9: Gestión Nóminas	CUS10: Gestión de Personal	CUS11: Gestión de Regiones	CUS12: Otorgar Incentivos	CUS13: Política de Ventas	CUS14: Reabastecer Almacén	CUS15: Realizar Oferta	CUS16: Redistribución de Personal	CUS17: Atender Pedido	CUS18: Cancelar Pedido Atendido	CUS19: Consultar Pedidos a Enviar	CUS20: Elaborar Pedido	CUS21: Elaborar Pedido On-line	CUS22: Gestión de Clientes	CUS23: Incidencia Pedido	CUS24: Introducir Recibos	CUS25: Pasar Pedido a Envío	CUS26: Realizar Envío	CUS27: Reposición de Stock
ACT1: Ingeniero de Logística																											
ACT2: Jefe de Almacén																											
ACT3: Técnico de Almacén																											
ACT4: Representante de Ventas																											
ACT5: Jefe de Ventas																											
ACT6: Contable																											
ACT7: Empleado de Marketing																											
ACT8: Cliente Online																											
ACT9: Operadora																											
ACT10: Encargado de Transporte																											
ACT11: Empleado de Recursos Humanos																											
ACT12: Jefe de Recursos Humanos																											

Figura 9. Matriz de trazabilidad casos de uso y actores
 Existe una liga entre la especificación de cada caso de uso y estas matrices de trazabilidad, logrando así tener en cualquier momento los detalles para dar seguimiento a un requerimiento del sistema.

Requisite Pro también permite, hacer una traza de cuáles actores se relacionan en los diferentes casos de uso, tal como lo muestra la figura 10. Esta funcionalidad permite que, en caso de que haya alguna duda relacionada con un determinado caso de uso, se pueda saber a cuál actor referirse para resolver dicha duda. [LÓPEZ03]

Relationships: - direct only	CUS1: Facturar Entregas Pendido	CUS2: Cobro Clientes	CUS3: Compra a Proveedores	CUS4: Confeccionar Catálogo	CUS5: Consultar Pedidos no Atendidos	CUS6: Control Estadísticas	CUS7: Consultar Catálogo	CUS8: Entrevista Trabajo	CUS9: Gestión Nóminas	CUS10: Gestión de Personal	CUS11: Gestión de Regiones	CUS12: Oligar Incentivos	CUS13: Política de Ventas	CUS14: Reabastecer Almacén	CUS15: Realizar Oferta	CUS16: Redistribución de Personal	CUS17: Atender Pedido	CUS18: Cancelar Pedido Atendido	CUS19: Consultar Pedidos a Enviar	CUS20: Elaborar Pedido	CUS21: Elaborar Pedido On-line	CUS22: Gestión de Clientes	CUS23: Incidencia Pendido	CUS24: Introducir Recibos	CUS25: Pasar Pedido a Envío	CUS26: Realizar Envío	CUS27: Reposición de Stock
ACT1: Ingeniero de Logística																											
ACT2: Jefe de Almacén																											
ACT3: Técnico de Almacén																											
ACT4: Representante de Ventas																											
ACT5: Jefe de Ventas																											
ACT6: Contable																											
ACT7: Empleado de Marketing																											
ACT8: Cliente Online																											
ACT9: Operadora																											
ACT10: Encargado de Transporte																											
ACT11: Empleado de Recursos Humanos																											
ACT12: Jefe de Recursos Humanos																											

Figura 10. Matriz de trazabilidad de casos de uso y actores

El RUP tiene todas las herramientas necesarias para que se dé la trazabilidad de adelante hacia atrás y viceversa, a través de todas sus fases y procesos, [JACOB99].

9. CONCLUSIONES

La experiencia a través del tiempo ha demostrado que el éxito o fracaso del proceso de desarrollo de un sistema de software, está directamente relacionado con la Administración de los Requerimientos, también conocida por algunos como Ingeniería de Requerimientos.

La trazabilidad ha venido a constituir un gran apoyo, que debe formar parte en todas las etapas del desarrollo de un sistema, desde la captura de requerimientos hasta la implementación. Esto contribuirá al seguimiento de un requerimiento hacia adelante o hacia atrás, reflejando los efectos que puede tener, la inclusión o exclusión de un nuevo requisito.

La trazabilidad también ayuda en la integración de nuevos módulos reflejando las implicaciones que puedan tener en el sistema.

Rational Unified Process contiene las herramientas necesarias para hacer posible esta trazabilidad, por lo que la metodología del Proceso Unificado junto con las herramientas de Rational, son una buena opción para los Ingenieros de Sistemas a la hora de desarrollar un sistema de software.

Si bien es conocida la problemática del proceso de requerimientos, el desconocimiento sobre la importancia de la trazabilidad se manifiesta en lo poco que se sistematiza en esta área. La cultura del desarrollo de sistema cada vez más exige documentación adecuada, sin embargo el mantenimiento y seguimiento de esta documentación no se lleva de manera adecuada, la trazabilidad es una forma de cambiar esta cultura y minimizar los fracasos en los proyectos de desarrollo de software.

10. REFERENCIAS

- [DAVIS93] A. Davis, Software Requirements. Objects, Functions and States, Englewood Cliffs, Prentice Hall, New Jersey, 1993.
- [GOTEL 94] Gotel, O.C.Z., Finkelstein, A.C.W., "An Analysis of the Requirements Traceability Problem", International Conference on Requirements Engineering, ICRE'94, Los Alamitos, California, Abril, 1994, pp 94-101.
- [SOM97] Sommerville, I., Sawyer, P., Requirements Engineering, Chichester, John Wiley & Sons, 1997.
- [PRESSMAN02] Pressman R. Ingeniería de Software. Un Enfoque Práctico. Quinta Edición. McGraHill,2002
- [GOTEL94] Publicado por Patricio Letelier, Requeriments Traceability. DOI= <http://www.dsic.upv.es/~letelier/pub/p12.ppt>, 1994
- [JACOB99] I. Jacobson, G. Booch and J. Rumbaugh, The Unified Software Development Process, 1999
- [ZULOAGA96] Ing. Luis Zuloaga Rotta, Análisis de Requerimientos, Referencia: The Standish Group. DOI= http://ww.campus.dokeos.com/courses/IS50/document/Exam.Parcial/An%e1lisis_Requerimientos.ppt?cidReq=IS50.1996
- [LÓPEZ03] César López Rodríguez, Ejemplo de desarrollo software utilizando la metodología RUP, Desarrollo de un sistema para la gestión de artículos deportivos. DOI= <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/2003>
- [SEWBOK04] Guide to the Software Engineering Body of Knowledge, SEBOK®, A project of the IEEE Computer Society Professional Practices Committee 2004 Version
- [HERRERA00] Lizka Johany Herrera J.,johany@cwpa.net.pa., DOI=<http://www.monografias.com/trabajos6/resof/resof.shtml>, 2000
- [MOREA00] Lucas Morea, Webmaster, Ingeniería De Requerimientos Ingeniería De Software Monografias.com, 2000
- [TORRES02] José Luis Torres Pérez, Administración de Requisitos, Líder de Proyecto, Universidad Autónoma de Guadalajara, DOI= <http://www.ewh.ieee.org/r9/guadalajara/boletin/marzo02/admonreq.htm>, 2002
- [BAUFEST03] Baufest software engineering, Requisitos. DOI= <http://www.baufest.com/spanish/consulting/requerimientos.html>
- [GUERRERO03] Luis A. Guerrero, Análisis y diseño Orientado a Objetos, basado en Material de profesora Cecilia Bastarrica, Universidad de Chile. DOI=<http://www.dcc.uchile.cl/~luguerre/cc40b/rup.ppt>, 2004
- [ANTON99] Leandro Antonelli y Alejandro Oliveros, Traceability en la Etapa de Elicitación de

Requerimientos ,LIFIA, Facultad de Informática,
UNLP, Dep. de Computación, Facultad de
Ingeniería, UBA. DOI= http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER01/antonelli.pdf,1999

[DOORN ET AL 07]Jorge H. Doorn, Carmen Leonardi,
Marcela Ridaio, Universidad Central de Buenos
Aires. DOI= http://www.exa.unicen.edu.ar/catedras/ingrequi/index_archivos/2007_lunes8_inicio.ppt

[RATION00]Rational Software Corporation, Key Concepts,
http://venus.imp.mx/hilario/personal/RUP2001.03.00.01/manuals/intro/im_keymc.htm