

La Gestión del Conocimiento en la Ingeniería de Requisitos

Hinojosa, Cecilia; Raura, Geovanny
cmhinojosa@espe.edu.ec, jgraura@espe.edu.ec
 Universidad de las Fuerzas Armadas ESPE

Resumen. La construcción y mantenimiento de software demandan un trabajo intenso en conocimiento, el personal técnico debe demostrar solvencia en Ingeniería de Software, técnicas, métodos, procesos organizacionales y otros aspectos que son altamente dinámicos y demandan un esfuerzo permanente de aprendizaje. Al ser la Gestión del Conocimiento una disciplina que, mediante un proceso sistemático, busca gestionar el conocimiento útil y válido; su estudio permitirá que de los ingenieros de requisitos, realicen un trabajo eficaz y eficiente; lo cual tendrá un alto impacto en la calidad del proceso y producto software.

El presente trabajo se plantea con el objetivo de crear una técnica para el análisis de requisitos basada en la gestión del conocimiento. Para el efecto se realizó un estudio comparativo de la ingeniería de requisitos y la gestión del conocimiento, llegándose a determinar que existe un alto grado de coincidencia en sus objetivos, ya que las dos buscan a obtener conocimientos válidos y útiles; de igual manera sus procesos son iterativos e incrementales con actividades similares. También se analizaron los atributos de calidad de los requisitos y conocimientos, identificando que hay un alto grado de coincidencia en los mismos. Con este contexto teórico se diseñó la técnica para el análisis de requisitos.

La validación de la técnica propuesta se realizó utilizando el caso de estudio como método de investigación en Ingeniería de Software, de manera documentada y sistemática se demostraron las fortalezas y debilidades de la propuesta. La técnica fue aplicada exitosamente por personal experimentado y novato, se demostró que la integración de elementos de la gestión del conocimiento en las técnicas de análisis de requisitos aportó a mejorar la calidad de los mismos; específicamente a la completitud, claridad, consistencia, pertinencia y exactitud.

Índice de Términos: Ingeniería de requisitos, Gestión del conocimiento, Análisis de requisitos, Ingeniería de Software.

I. INTRODUCCIÓN

La calidad de los requisitos puede determinar la diferencia entre el éxito o el fracaso de un proyecto,

independientemente del grado de complejidad del mismo. Para Capers Jones los requisitos deficientes son la principal causa del fracaso de proyectos de software. Luego de analizar cientos de organizaciones, llegó a determinar que el proceso de ingeniería de requisitos es deficiente en más del 75 por ciento de las mismas (Jones, 1996).

Para (Zapata, 2009), la ingeniería de requisitos (IR) es una fase crucial en el proceso de desarrollo, los diferentes mundos de donde provienen los involucrados deben armonizar y llegar a acuerdos sobre lo que necesitan y esperan del producto software resultante. Para obtener requisitos de alta calidad, es preciso seguir un proceso bien definido, así también utilizar en cada fase las técnicas que permitan asegurar que los requisitos son recopilados con precisión, revisados, documentados y aprobados (Info-Tech Group Research, 2007).

Para facilitar la comprensión de las necesidades y expectativas de los usuarios se creyó pertinente diseñar una técnica basada en la gestión del conocimiento, disciplina orientada a fomentar los procesos de comprensión, almacenamiento y difusión del conocimiento.

De esta manera se propone un conjunto de actividades, que con el soporte de matrices y otras herramientas facilitan el trabajo del personal técnico y contribuye a obtener requisitos de calidad.

El presente artículo se encuentra organizado en cinco secciones, en la primera se realiza una introducción de la problemática de la Ingeniería de Requisitos, la segunda presenta el marco teórico de las dos disciplinas, en la tercera se realiza un análisis comparativo, en la cuarta sección se propone una técnica para el análisis de requisitos basada en la gestión del conocimiento, guiada por directrices claras y formatos que guían su aplicación. Finalmente se presenta el trabajo futuro y las conclusiones.

II. MARCO TEÓRICO

A. Conocimiento

Para (Davenport & Prusak, 1998) el conocimiento es una mezcla de experiencia, valores, información y “saber hacer” que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Para (Biggs, 2007) “*el conocimiento es el objeto de la comprensión*”.

La Figura 1 representa de manera gráfica las relaciones y diferencias entre datos, información, conocimiento y sabiduría.

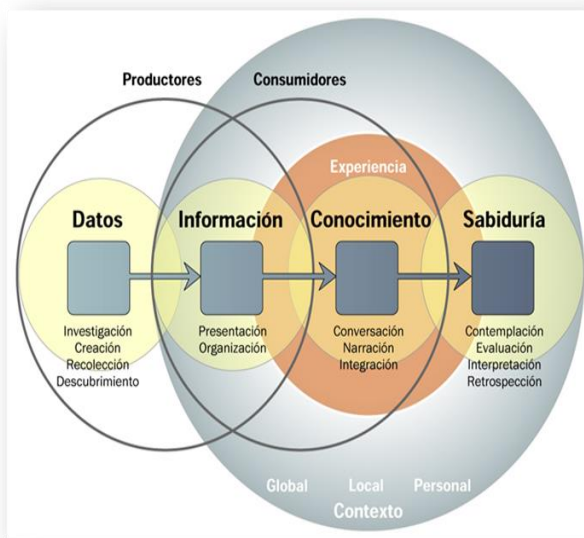


Figura 1: Relación entre datos, información, conocimiento y sabiduría. (Shedroff)

B. Características de Conocimiento

Validez: Para que un conocimiento sea válido debe ser exacto, consistente y cierto.

Utilidad: Los atributos de la utilidad de los conocimientos son: claridad, significado, relevancia e importancia. La utilidad de un conocimiento se puede evaluar únicamente en relación con quien es el conocedor y el problema que se aborda concretamente.

Claridad: Concierno a cuánto esfuerzo debe emplear el conocedor en la interpretación de alguna representación de los conocimientos. Si los

conocimientos se representan de manera incomprensible, entonces resultan poco útiles.

Significado: Naturalmente, lo que es claro para una persona puede resultar indescifrable para otra persona. Un elemento de conocimiento puede ser más significativo para una persona que para otra.

Pertinencia: Que sea correspondiente al problema a resolver.

Importancia: Actualidad, facilidad de uso, disponibilidad.

C. Tipos de Conocimiento

John Biggs (Biggs, 2011) identifica los siguientes tipos de conocimientos:

Declarativo / proposicional / de contenido: Se refiere al “saber qué” sobre las cosas, eventos o un tema, se expresa en sistemas de símbolos, generalmente de forma verbal. El conocimiento declarativo es de conocimiento público, con sujeción a reglas de evidencia que hacen que sea verificable, replicable y lógicamente consistente, está en las bibliotecas y los libros de texto y en Internet, es una clase teórica; el aprendiz interioriza el conocimiento pre-existente de manera significativa. Este conocimiento se incrementa con la investigación.

Procedimental: Se enfoca al “saber hacer” y se incrementa con la experiencia

Condiciona: Agrega los dos tipos anteriores de conocimiento, busca que la persona conocedora sepa qué, cuándo, por qué y en qué condiciones utilizar el conocimiento para la acción.

Funcional. Incorpora el conocimiento declarativo, procedimental y condicional. Se basa en actuaciones fundamentadas en la comprensión para resolver problemas. El conocimiento funcional requiere un sólido fundamento declarativo, saber cómo, cuándo, por qué y en qué condiciones de aplica para resolver problemas. El aprendiz participa activamente al aplicar el conocimiento en la práctica, utiliza la teoría para fundamentar sus decisiones sobre lo que debe hacer en su contexto. Se incrementa con la experiencia.

Conocimiento Tácito. Para (Leonard & Sensiper, 1998) el conocimiento tácito es la capacidad de la mente humana para dar sentido al conjunto de experiencias vividas y para conectar pautas desde el pasado al presente y al futuro. Es el conocimiento semiconsciente o inconsciente, presente en la mente

y en el ser de las personas. El conocimiento tácito se encuentra enraizado en la experiencia. Polanyi resume el fundamento del conocimiento tácito en la frase "sabemos más de lo que podemos decir". El conocimiento tácito también se puede diferenciar entre individual y colectivo, este último se construye en la comunidad a lo largo del tiempo, mediante las interacciones entre los individuos que la conforman.

Para (Nonaka & Takeuchi, 1995), desde la perspectiva epistemológica el conocimiento se clasifica en tácito y explícito.

Conocimiento tácito. Este tipo de conocimiento no es fácilmente visible ni expresable, está formado por las habilidades técnicas (know how) y por las dimensiones cognitivas: modelos mentales, costumbres, creencias y perspectivas arraizadas en las personas.

Conocimiento explícito. Es aquel que puede ser expresado mediante palabras, números, gráficos, se encuentra estructurado y esquematizado: facilitando su comunicación, difusión y transmisión, ya sea en forma de datos, fórmulas u otros. Es el conocimiento almacenado en medios físicos, también se conoce como "conocimiento codificado"; por lo que es fácil de transmitir y almacenar. Se presume que entre el 15 y 20 % del conocimiento que se tiene individualmente o en una empresa, suele resultar valioso y está capturado y codificado en forma de libros, bases de datos, videos y otros medios.

(Leonard & Sensiper, 1998) proponen un "continuo del conocimiento", en un extremo se encuentra el conocimiento completamente tácito, en el extremo opuesto se encuentra el conocimiento completamente explícito, es decir, el conocimiento codificado, estructurado y accesible a personas distintas de las que lo generaron. La mayoría del conocimiento se sitúa entre estos dos extremos. Los elementos explícitos son objetivos y racionales; mientras que los elementos tácitos son subjetivos y basados en la experiencia. En la Figura 2 se representa el continuo del conocimiento.



Figura 2: Continuo del conocimiento

(Nonaka & Takeuchi, 1995) indican que el conocimiento se crea mediante una espiral de transformación del mismo, este proceso integra las interacciones entre la dimensión epistemológica: conocimiento tácito y explícito; y la dimensión ontológica: individuo, grupo, organización e inter-organizaciones.

La interrelación entre las citadas dimensiones da lugar a un espiral ascendente, en el que los individuos, grupos, organizaciones amplían, extienden y transforman sus propios conocimientos. Este proceso consta de cuatro fases: socialización, externalización, combinación e internalización, las cuales se pueden apreciar en la Figura 3.

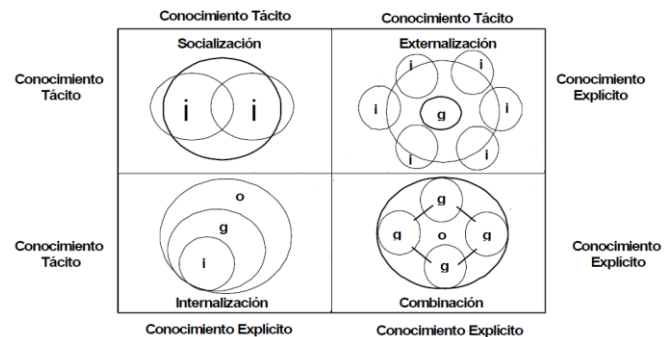


Figura 3: Fases para la creación de conocimiento (Nonaka & Takeuchi, 1995)

La espiral inicia, con el individuo que a través de la experiencia crea conocimiento tácito, éste se conceptualiza, mediante técnicas deductivas e inductivas convirtiéndolo en explícito individual. Mediante la recopilación, reflexión y síntesis lo comparte convirtiéndose en conocimiento explícito social. En la fase de internalización, se comparten las experiencias comunes, transformando el conocimiento explícito social en tácito individual, de esta manera, las organizaciones y las personas que las conforman amplían y transforman su conocimiento.

D. Gestión del Conocimiento

Para (Núñez Paula, 2002) la gestión del conocimiento no es una actividad natural o innata de los miembros de una organización, es la gerencia quien debe propiciar e inspirar para que la organización cultive y explote su conocimiento.

Un aporte de (Garvin, 1998) es la identificación de las fases de la gestión del conocimiento y enfatiza que su objetivo es transformar a la empresa en una organización inteligente.

Víctor Izunza (Izunza, 2010) indica que el conocimiento puede residir en cualquier lugar y precisa las actividades para gestionar el conocimiento.

(Soto y Barrios, 2006) Destacan la convergencia de recurso humanos, información y tecnología, como los recursos que posibilitan su ejecución.

En (Hinojosa, 2011), se recalca que la gestión del conocimiento es un proceso cíclico, que en cada iteración se incrementa el conocimiento.

Se puede definir entonces a la gestión del conocimiento como un proceso cíclico, mediante el cual se captura, analiza, organiza, almacena, difunde y crea nuevo conocimiento, con el fin de mejorar procesos, organizaciones o resolver problemas en cualquier ámbito. Incrementa la accesibilidad al conocimiento y a la información, y con ello potencializa procesos de aprendizaje y desarrollo personal, mediante la colaboración e investigación constantes.

E. Proceso de la Gestión del Conocimiento

El conjunto de actividades interactuantes que permiten que la GC cumpla con su objetivo, ha sido analizado por varios autores, quienes plantean propuestas similares, mas no han llegado a un consenso. En la Tabla 1 se presenta un compendio de varias propuestas.

Estas actividades de creación de conocimiento tienen lugar en y entre personas, en y entre las organizaciones. El conocimiento no sólo se encuentra dentro de documentos; están implícitos en las rutinas organizativas, procesos, prácticas y normas, que son la base de la capacidad productiva de la organización, también pueden estar en almacenes de datos y otros medios.

TABLA 1
Subprocesos de la gestión del conocimiento

Autor, año	Subprocesos					
Ruggles, 1997	Generación	Codificación	Transferencia			
Davenport & Prusak, 1998	Generación	Codificación	Uso	Transferencia		
Jackson, 1998	Recolección	Almacenamiento	Comunicación	Diseminación	Síntesis	
Garvin, 1998	Generar	Recoger	Asimilar	Aprovechar		
Laudon & Laudon, 1998	Creación	Captura y codificación	Distribución	Compartir		
Patel et al, 2000	Generación	Representación	Recuperación	Compartir		
Anumba et al, 2001	Obtener nuevo conocimiento	Ubicar y acceder	Transferir	Mantener	Modificar	
Hinojosa, 2011	Capturar	Analizar	Organizar	Almacenar	Difundir	

Fuente: Modificado de (Al-Ghassani, Carrillo, & Anumba, 2001)

En la Figura 4 se presenta, desde la visión de la autora, las fases de la gestión del conocimiento, en la que se considera que el ciclo de la gestión es: planificar, hacer, verificar, actuar; también conocido como el ciclo “P-H-V-A”.

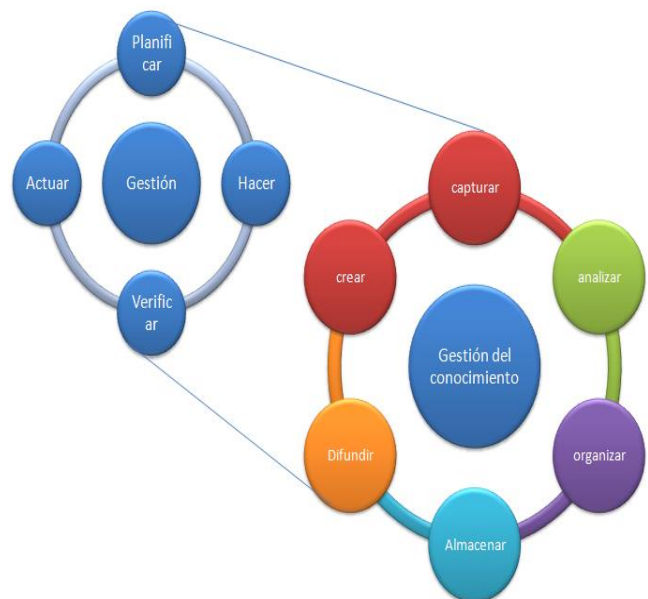


Figura 4: Fases para la gestión del conocimiento

F. Ingeniería de Requisitos

La Ingeniería de Requisitos es un proceso cooperativo, iterativo e incremental (Pohl, 1994), en el cual se descubren, analizan, documentan, comunican, validan y gestionan (Hull, Jackson, & Dick, 2011) las características o restricciones operativas y funcionales que se esperan del sistema, las cuales deben ser completas y acordadas (Pohl, 2010) entre los involucrados; de tal manera que sean la base para las posteriores actividades del desarrollo del sistema.

Su objetivo es recoger de manera correcta las necesidades de los interesados. La IR es la base para planificar el proyecto, para diseñar el software, para la entrega – recepción del producto; en resumen, se constituye en los cimientos para determinar la calidad del producto software.

G. Problemática de la Ingeniería de Requisitos

Está ampliamente documentado que la IR es la fase más crítica del proceso de desarrollo de software (Pandey, Suman, & Ramani, 2010). Para (Hofman & Lehner, 2001), “obtener los requisitos correctos podría ser la parte más importante y difícil de un proyecto de software”.

Según (Durán, 2012) “... es necesario destacar la falta de técnicas para el análisis de los requisitos no funcionales, que son prácticamente ignorados por las técnicas de modelado conceptual y relegados para las fases posteriores de desarrollo”.

El informe "Chaos Report 2010" publicado por la consultora Standish Group, indica que si bien hay un importante incremento en la tasa de éxito de los proyectos, respecto de los datos obtenidos en su primer estudio en 1995, los datos todavía son preocupantes. Cita que el 37% de los proyectos de desarrollo de software, se pueden considerar exitosos, ya que fueron entregados a tiempo, dentro del presupuesto, con las características necesarias y la funcionalidad pactada, el 42% se entregaron fuera de plazo, excedieron su presupuesto no cubrieron la totalidad de las características y funciones y el 21% de los proyectos fueron cancelados (Standish Group, 2009). En este mismo estudio se puntualiza

que uno de los principales factores para el fracaso de un proyecto de desarrollo de software radica en de mala calidad de los requerimientos.

En este escenario resulta evidente la necesidad de un mayor esfuerzo en la investigación de la IR, tendiente garantizar la calidad de los requisitos, especialmente por la complejidad que reviste el trabajo con los involucrados y por su heterogeneidad.

H. Proceso de Ingeniería de Requisitos

El “Software Engineering Body of Knowledge – SWEBOK” (IEEE Computer Society, 2004) determina que las fases genéricas del proceso de IR, son: Elicitación, Análisis, Especificación y Validación. Estas fases siguen una secuencia lógica y se ejecutan de manera iterativa. A continuación una breve descripción de las citadas fases.

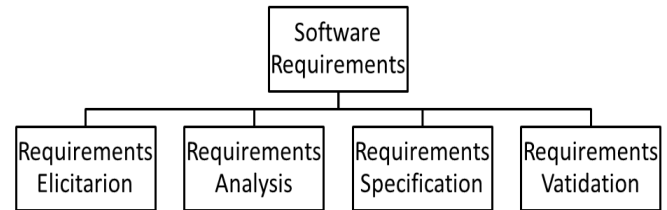


Figura 5: Fases del proceso de Requisitos del Software (IEEE Computer Society, 2004)

Elicitación de requisitos: Esta es una actividad fundamentalmente humana. También se la conoce como “captura de requisitos”, “descubrimiento de requisitos” o “adquisición de requisitos”. En esta fase se debe:

- a) Comprender el problema que resolverá el software.
- b) Identificar a los involucrados

Es importante que el especialista en requisitos tenga la capacidad de comunicarse eficazmente ya que debe mediar entre los involucrados y armonizar las necesidades que se originan en los diferentes niveles de la organización.

Análisis de requisitos: En la elicitación se recolectó información, parte de ella no constituyen verdaderos requisitos, en el análisis se deben realizar las actividades tendientes a asegurar la calidad de los mismos, lo cual constituye una tarea

compleja, ya que los requisitos deben responder a las diversas visiones de los actores, a los diferentes niveles de la organización y al entorno con el que interactuará y en el cual operará. Entre los objetivos que se persiguen en esta fase están:

- Determinar la información útil: Resulta muy beneficioso en esta fase clasificar la información, separando aquella que verdaderamente constituye una característica deseable del producto software que se desarrollará, de aquella que no lo es.
- Gestionar el conocimiento del dominio del problema: mediante un proceso claramente definido llegar a que todos los involucrados, esto es: clientes, usuarios y personal técnico tengan el mismo nivel de conocimiento del producto a desarrollar, tanto de la parte funcional, como de los aspectos no funcionales.
- Detectar conflictos en los requisitos: Debido a que la información proviene de distintas fuentes, se pueden presentar contradicciones o ambigüedades, las cuales deberán resolverse oportunamente aplicando las técnicas necesarias para este fin.
- Establecer las bases para el diseño: los modelos conceptuales utilizados en el análisis resultan una importante guía para el desarrollo de los modelos del sistema.

Especificación de requisitos: En esta fase se documentan los requisitos resultantes del análisis. El documento resultante es la especificación de requisitos del software, las buenas prácticas recomiendan seguir una norma para su desarrollo, por ejemplo la “IEEE Recommended practice for software requirements specifications” (IEEE, 1998).

Verificación y validación de requisitos: En esta fase los involucrados revisan los productos obtenidos, verifican que reflejen sus expectativas y necesidades, que estén expresadas claramente y definan el producto deseado, a fin de que se cuente con una base sólida para las siguientes fases del proceso de desarrollo.

I. Atributos de calidad de los requisitos

En (IEEE, 1998), se establecen los atributos de calidad de los requisitos del software, los cuales se

citan a continuación.

Correcto: Debe reflejar las necesidades reales de los involucrados.

No Ambiguo: Debe estar expresado de manera que tenga una única interpretación.

Completo: se refiere al conjunto de requisitos y debe incluir todos los requisitos significativos del software (relacionados con la funcionalidad, ejecución, diseño, atributos de calidad o interfaces externas).

Verificable: Debe garantizar que existe algún proceso no costoso por el cual una persona o una máquina puedan chequear que el software satisface cada requerimiento.

Consistente: Una ERS es consistente si y sólo si, ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto.

Clasificado: Los requisitos deben estar clasificados, ya sea por importancia, estabilidad u otros criterios.

Modificable: Debe garantizar que cualquier cambio puede realizarse de manera fácil, completa y consistente.

Trazable: Debe permitir identificar las referencias hacia atrás, esto es el origen y hacia adelante, cuando un requisito se desglosa o deriva de otro requisito. Estas referencias son especialmente importantes, para el mantenimiento del software y evitar la degradación del mismo.

Mantenible: Deben estar expresados de tal manera que sean utilizables durante las tareas de mantenimiento y uso, lo cual supone una correcta documentación.

III. ANÁLISIS COMPARATIVO

A. Análisis de los atributos de calidad

Uno de los factores críticos para la toma de decisiones y la solución de problemas en general, es la calidad de los conocimientos. La calidad de los conocimientos depende de las fuentes y de la manera en que se procesa la información. A la hora de determinar si un conocimiento es de calidad, se deben evaluar su utilidad y su validez.

De igual manera para desarrollar un producto software que cumpla con las expectativas de clientes y usuarios, es necesario contar con requisitos que cumplan con las características de calidad.

En la Tabla 2 se puede apreciar que existe un alto grado de concordancia entre los atributos de calidad de los conocimientos y los requisitos. Respecto de la validez, tanto conocimientos, como requisitos comparten los mismos atributos. En referencia a la utilidad, los conocimientos y los requisitos coinciden parcialmente, los atributos coincidentes son: exactitud, consistencia, certidumbre, organizado, fácil de usar.

TABLA 2

Atributos de calidad de los conocimientos y de los requisitos de software

Conocimiento		Requisito
Validez	Exacto	No ambiguo
	Consistente	Consistente
	Cierto / verdadero	Correcto / verificable
Utilidad	Organizado	Clasificado
	Fácil de usar	Trazable / modificable / mantenible
	Claro (facilidad de comprensión) Pertinente Disponible	Completo

Fuente: Autoría propia

Los conocimientos adicionalmente especifican: claridad, pertinencia y disponibilidad y para los requisitos se agrega completitud.

Claridad: Se refiere al esfuerzo que debe emplear el conocedor para comprender un conocimiento. Si se encuentra representado o expresado de manera incomprensible, el conocimiento resulta de poca utilidad.

Pertinencia: Si el conocimiento analizado no es relevante para la solución del problema, o se encuentra fuera de contexto, tampoco es de utilidad.

Disponibilidad: Si un conocimiento no se encuentra disponible, no puede ser usado.

Los atributos de calidad analizados, no solo son relevantes para los conocimientos, también son atributos cuyo análisis contribuye a mejorar la calidad de los requisitos. Si un requisito no es claro, pertinente y disponible, tampoco podrá considerarse como un requisito válido y útil.

B. Análisis de los procesos

Las dos disciplinas estudiadas son procesos cíclicos, iterativos e incrementales. Esto es, mediante la repetición se completan y mejoran los productos resultantes. En el caso de la IR, se obtienen los requisitos finales luego de varias iteraciones.

En la Tabla 3 se presentan de manera organizada los subprocesos o fases de las dos disciplinas. Para el efecto se han recopilado los términos utilizados por diversos autores.

TABLA 3

Fases de los procesos de la ingeniería de requisitos y gestión del conocimiento

	Gestión del Conocimiento	Ingeniería de Requisitos
1	Crear / Capturar / recolectar	Descubrir / elicitar / educir
2	Analizar / Organizar	Analizar
3	Almacenar / codificar / representar	Documentar / especificar
4	Difundir / distribuir / transferir	
5		Validar

Fuente: Autoría propia

Como se puede apreciar, tanto el proceso de gestión del conocimiento, como el proceso de ingeniería de requisitos tienen un alto grado de coincidencia. Las tres primeras fases son iguales, difieren en que la gestión del conocimiento incluye la difusión o transferencia del conocimiento y en el

proceso de ingeniería de requisitos propone de manera explícita la validación.

IV. TÉCNICA PARA EL ANÁLISIS DE REQUISITOS

La técnica propuesta para el análisis de requisitos basada en la gestión del conocimiento, busca alcanzar los siguientes objetivos específicos:

- Determinar si la información recolectada en la elicitación es válida y útil
- Gestionar el conocimiento del dominio del problema
- Establecer bases sólidas para el diseño del software

A. Actividades

Se ha diseñado un conjunto de actividades que permitan realizar un tamizaje de los requisitos candidatos, discriminar los verdaderos requisitos de los que no lo son. De esta manera será posible intervenir oportunamente, minimizar el retrabajo y disminuir los costos del proyecto.

TABLA 4
Actividades para el análisis de requisitos

Actividad	
1	Analizar la calidad de las fuentes de información
2	Analizar el aporte de los requisitos a los objetivos del sistema
3	Verificar que el conocimiento tácito se ha plasmado en explícito
4	Clasificar requisitos según afinidad
5	Contrastar la lista de requisitos resultantes de la elicitación con los mapas mentales o conceptuales
6	Difundir el conocimiento

El análisis de los diferentes elementos considerados en la técnica, propicia desde una perspectiva holística e integradora, la comprensión global del sistema, la reflexión de los objetivos del sistema y cómo cada requisito aporta al cumplimiento de los mismos. A continuación se detallan las actividades propuestas.

Analizar las fuentes de información: Esta actividad tiene por objetivo determinar si se consideraron las fuentes de información pertinentes, que se han considerado a todos los stakeholders, de tal manera que el conjunto de requisitos demuestren unidad y

coherencia. El formato sugerido se presenta en la Figura 6.

Informantes	Actores del sistema		
	<Actor 1>	<Actor 2>	<Actor n>
<nombre informante>			

Documentación relacionada	Ubicación	Argumentos para considerarla
<nombre documento>		

Otras fuentes	Ubicación	Argumentos para considerarla

ANÁLISIS

Novedades encontradas	Acciones a tomar
<novedad detectada>	<acción a tomar>

Figura 6: Formato análisis de fuentes de información

Analizar el aporte de los requisitos a los objetivos del sistema: Esta actividad tiene por objetivo, aportar a la reflexión de los objetivos específicos del sistema y evaluar en qué medida aporta cada requisito al cumplimiento de los mismos. Además permite una manera objetiva priorizar los requisitos. El formato sugerido se encuentra en la Figura 7.

Requisitos preliminares	Objetivos específicos del sistema			
	<Obj. 1>	<Obj. 1>	<Obj. 1>	Prioridad sugerida
<código del requisito>	<valor>	<valor>	<valor>	<∑ fila>

0 = No aplica 1 = Bajo 2 = Medio 3 = Alto

ANÁLISIS

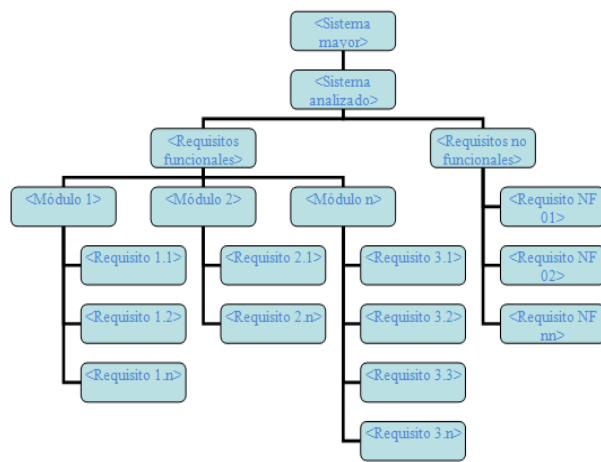
Novedades encontradas	Acciones a tomar
<novedad detectada>	<acción a tomar>

Figura 7: Formato análisis de fuentes de información

Verificar que el conocimiento tácito se ha plasmado en explícito: Con las actividades anteriores se ha cumplido con un primer nivel de

depuración de los requisitos, luego se debe pasar al siguiente nivel, el cual permitirá verificar que el conocimiento tácito se haya plasmado en conocimiento explícito, esto es que el conocimiento pertinente se encuentre documentado.

Para este efecto se ha considerado utilizar organizadores gráficos, ya que son una herramienta que aporta al aprendizaje, a la comprensión y a la transmisión de ideas y conocimientos. Mediante el uso de constructores se transforma el conocimiento tácito en explícito de una manera visual y simplificada. Un ejemplo de mapa conceptual se presenta en la Figura 8.



ANÁLISIS

Novidades encontradas	Acciones a tomar
<novedad detectada>	<acción a tomar>

Figura 8: Formato mapa conceptual

Clasificar requisitos por afinidad: Tomando como base los mapas mentales o conceptuales, se deben clasificar los requisitos por afinidad, al realizar este paso, se pueden identificar requisitos duplicados. Luego se procede a clasificar los requisitos por prioridad. De esta manera se puede confirmar con los “satkeholders”, aquellos que realmente se mantienen como requisitos y si fuera el caso desechar algunos. Se debe recordar que para el cumplimiento de cada requisito, se utilizan recursos y tiempo; por lo tanto se realizará un análisis costo beneficio si es necesario argumentar una decisión. De esta manera se obtiene la lista de requisitos depurados y clasificados por afinidad.

Difundir el conocimiento y retroalimentar: la lista de requisitos obtenida en la actividad anterior, se debe informar a todos los involucrados y obtener sus observaciones o aprobación.

V. Trabajos Futuros

La presente investigación se complementará con los siguientes trabajos:

La técnica debe ser validada de manera sistemática y organizada, a fin de determinar el aporte de la misma, y si fuera el caso, identificar elementos que contribuyan a su mejora. Para el efecto se recomienda utilizar el caso de estudio como método de investigación.

Desarrollar un software que apoye con la automatización de la técnica y la generación de una base de conocimiento. Esta base contribuirá a las organizaciones en sus procesos de mejora continua.

VI. CONCLUSIONES

Del En el presente trabajo se propuso una técnica para el análisis de requisitos de software basada en la gestión del conocimiento, para el efecto se investigaron los fundamentos y principios de la gestión del conocimiento y el proceso de ingeniería de requisitos.

Del estudio y análisis comparativo de la Ingeniería de Requisitos y la Gestión del Conocimiento, se concluye que las dos disciplinas tienen un alto grado de similitud en las fases del proceso. Así también las características de calidad de los conocimientos y los requisitos son coincidentes.

El objetivo principal de Análisis de Requisitos es asegurar la calidad de los requisitos y los objetivos secundarios son: precisar los límites del sistema, precisar la interacción entre el sistema y el entorno y trasladar los requisitos del usuario a requisitos del software. La técnica para el análisis de requisitos, mediante herramientas simples y de fácil uso, tal

como matrices y organizadores gráficos contribuyó a la comprensión del dominio del problema desde una visión holística e integradora.

El caso de estudio resultó el método de investigación adecuado para demostrar la validez de la técnica propuesta. De una manera sistemática permitió afrontar las características complejas del trabajo de la ingeniería de requisitos y mediante la cadena de evidencias cualitativas y cuantitativas se obtuvieron conclusiones relevantes para la investigación.

Tanto la pregunta principal, como las preguntas secundarias de investigación planteadas en el caso de estudio se confirmaron plenamente.

Se demostró que la integración de elementos de la gestión del conocimiento en las técnicas de análisis de requisitos aportó a mejorar la calidad de los mismos; los resultados obtenidos en los ámbitos académico y empresarial evidenciaron mejoras en la calidad de los requisitos obtenidos luego de la aplicación de la técnica planteada.

Se demostró que la técnica fue aplicada exitosamente por personal técnico, experimentado y novato.

Según la encuesta aplicada, la técnica resultó beneficiosa para las dos categorías de profesionales, sin embargo, los ingenieros junior le otorgaron una evaluación más alta: por lo tanto, se puede inferir que la técnica resulta más beneficiosa para ingenieros junior.

La técnica propuesta enfrenta eficientemente los problemas detectados en la empresa auspiciante. Los ingenieros que verificaron y validaron la técnica identificaron aspectos positivos, entre los que se pueden citar:

- Permite identificar desde las fases tempranas de desarrollo a todos los actores del sistema.
- Permite validar las fuentes de información
- Sustenta herramientas de organización clara y precisa.
- Tener requisitos más detallados
- Tener requisitos más claros
- Ayuda a identificar nuevos requisitos

- Permite tener una visión más detallada del alcance del sistema
- Aporta a comprender el dominio del problema
- Permite priorizar los requisitos de manera objetiva
- Muestra una manera práctica para determinar las prioridades, en base a los objetivos
- Determinar las prioridades
- El trabajo del análisis es más efectivo
- Se cuenta con documentación más específica
- Aporta a aplicar de forma correcta la Ingeniería de Software

VII. BIBLIOGRAFÍA

- Al-Ghassani, A., Carrillo, P., & Anumba, C. R. (2001). Software requirements for knowledge management in construction organizations. *17th Annual ARCOM Conference* (págs. 199-206). Salford: Association of Researchers in Construction.
- Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. Berkshire, England: McGraw-Hill Education.
- Davenport, T., & Prusak, L. (1998). *Working Knowledge: "How Organizations Manage What they know"*. Harvard Business School Press.
- Del Moral, A. (2007). *Gestión del Conocimiento*. Madrid: Thomson.
- Durán, A. (23 de 10 de 2012). *Tesis doctorales Universidad de Sevilla*. Obtenido de <http://www.buenastareas.com/ensayos/Tecnicas-Para-El-Analisis-De-Requerimientos/2542475.html>
- Garvin, D. (1998). Building a learning organization. *Harvard Business Review*, 78-91.
- Hinojosa, C. (2011). De las redes sociales a las redes del conocimiento. Madrid: Anaya .
- Hofman, H., & Lehner, F. (2001). Requirements Engineering as a Success Factor in Software Projects. *IEEE Software*, 58.66.
- Hsia, P., Davis, A., & Kung, D. (1993). Status report: requirements engineering. *IEEE Software*, 75-79.
- Hull, E., Jackson, K., & Dick, J. (2011). *Requirements Engineering*. London: Springer.
- IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. New York: IEEE.
- IEEE Computer Society. (2004). *Guide to the Software Engineering Body of Knowledge*. California: IEEE.
- Info-Tech Group Research. (20 de 02 de 2007). *Get Requirements Right the First Time*. Obtenido de Info-Tech Group Research: <http://www.infotech.com/research/get-requirements-right-the-first-time>

- Izunza, V. (10 de 2010). *Gestión del conocimiento, definiciones*. Recuperado el 10 de 2010, de <http://www.a3net.net/es/gescon/definiciones.htm>
- Jones, C. (1996). *Applied software measurement: assuring productivity and quality*. New York: McGraw Hill.
- Leonard, D., & Sensiper, S. (1998). The role of tacit knowledge in group innovation. *California Management Review*, 112-132.
- Loucopoulos, P., & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.
- Nonaka, I., & Takeuchi, H. (1995). *La organización creadora de conocimiento. Cómo las compañías japonesas crean la dinámica de la innovación*. México: Oxford University Press.
- Núñez Paula, I. A. (2002). Bases para la introducción de la gestión del conocimiento en Cuba. *CITMA*.
- Pandey, D., Suman, U., & Ramani, A. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management. *1*(978-1-4244-8093-7).
- Pohl, K. (1994). *The Three Dimensions of Requirements Engineering*. Recuperado el 20 de 04 de 2012, de http://www.sciencedirect.com/science?_ob=ArticleListURL&_method=list&_ArticleListID=1951381899&_sort=r&_st=13&view=c&_acct=C000228598&_version=1&_urlVersion=0&_userid=10&md5=d7e61ec8e990c4917bd50bd9116db4&searchtype=a
- Pohl, K. (2010). *Requirements Engineering*. London: Springer.
- Shedroff, N. (s.f.). *Experience Desing*. Recuperado el 25 de 02 de 2013, de <http://www.nathan.com/>
- Standish Group. (23 de 04 de 2009). *Standish Group Report*. (The Standish Group) Recuperado el 30 de 08 de 2011, de http://www1.standishgroup.com/newsroom/chaos_2009.php
- Zapata, C. (2009). Una propuesta de metaontología para la educación de requisitos. *18*(1).