



Universidad de las Ciencias Informáticas
Facultad 3

MIR-SWG: MODELO DE INGENIERÍA DE REQUISITOS PARA SOFTWARE DE GESTIÓN EN LA FACULTAD 3

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERO EN CIENCIAS INFORMÁTICAS

AUTOR(AS):

MARISLEYDIS SOCAS ALVEZ
LISSETTE RODRÍGUEZ VERDECIA

TUTOR(A): ING. KARINA PÉREZ TERUEL
CO-TUTOR: LIC. MANUEL VÁZQUEZ ACOSTA

CIUDAD DE LA HABANA

JUNIO, 2007

*LA MAYORÍA DE LAS IDEAS FUNDAMENTALES DE LA CIENCIA SON
ESENCIALMENTE SENCILLAS Y, POR REGLA GENERAL PUEDEN SER
EXPRESADAS EN UN LENGUAJE COMPENSIBLE PARA TODOS.*

ALBERT EINSTEIN (1879-1955)

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año

_____.

Autor: Marisleydis Socas Alvez

Autor: Lissette Rodríguez Verdecia.

Tutora: Ing. Karina Pérez Teruel

DATOS DE CONTACTO

Ing. Karina Pérez Teruel

Centro de trabajo: Universidad de Ciencias Informáticas

Categoría docente: Instructora.

Asignatura que imparte: Ingeniería de Software I y II

Aspirante al grado científico de máster en ciencias.

Línea de investigación: Ingeniería de Requisitos.

Participación en eventos: UCIENCIA 2005, UCIENCIA 2006, Informática 07, Congreso de Reconocimiento de patrones 2005.

AGRADECIMIENTOS

Hemos querido agradecer a todos los que de una manera u otra nos han brindado su apoyo, y de manera muy especial:

A nuestra Tutora Karina Pérez, por su ayuda incondicional y por sus ideas siempre oportunas.

Al Profe Manuel Vázquez, por guiarnos en la concepción metodológica de la tesis.

A todos los especialistas, que nos dedicaron su tiempo para validar la aplicabilidad del modelo.

DEDICATORIA

Hoy, cuando ha llegado el gran día y me siento colmada de felicidad, llegan a mi mente los recuerdos de estos cinco años que he vivido con amor, entre la nostalgia de estar lejos de mi familia y la gran oportunidad que me dio la Revolución, de estudiar en la UCI.

Hoy, cuando ha llegado el gran día, pienso en todos los que me han ayudado a cumplir mi sueño y digo entonces, menos mal que existen ustedes.

He querido dedicar este trabajo a quienes de alguna manera les pertenece.

A nuestro Comandante en Jefe, Fidel Castro, por brindarme la oportunidad de formar parte de su idea.

A mi madre, por ser mi madre y mi amiga y porque a pesar de la distancia siempre está a mi lado, transmitiéndome su fuerza, su espíritu y su amor.

A mi padre, porque estoy orgullosa de él y se que el también lo está de mi; por su confianza y porque siempre tendré una deuda con él (no haber estudiado medicina).

A mi hermano Rodo, porque es lo que más quiero en mi vida.

A mi familia, porque siempre han estado al tanto de mis estudios, mi futuro y mi vida.

A todos mis amigos y en especial a Adita y Darian, porque siempre han estado junto a mí.

A Lisy, porque más que amiga ha sido una hermana para mi y porque esta tesis, para ambas, cierra una etapa importante en nuestras vidas y constituye el sello de nuestra amistad.

MARISLEYDIS SOCAS ALVEZ

DEDICATORIA

A mi madre, por sobre todas las cosas, va dedicado mi esfuerzo y todos mis años de estudio, por estar siempre a mi lado, presente en cada decisión y celebrar mis triunfos como si fueran suyos.

A mi padre, que a pesar de la distancia que nos separa, está cerca por su apoyo y preocupación constantes.

A mi abuelita Aby, por darme el privilegio de contar con dos madres y estar siempre pendiente de mi vida.

A mi hermanito Armandito, por hacerme feliz cada día con sus ocurrencias.

A Armando, que también ha sabido ocupar el lugar de un padre.

A quien por amor estoy unida, mi novio Ariesty, que comparte conmigo en cada momento, y lo siento también parte de mi vida.

A mi familia en general, por apoyarme y estar siempre pendiente de mí y de esta carrera que tanto les ha gustado.

A mis compañeros de estudio, y en especial a quienes la palabra amigas, le quedaría corta, Rosa y Perla, que han sido siempre incondicionales conmigo.

A Mary, que más que una amiga, es una hermana, mi compañera de tesis, que sabe comprenderme y poner nuestra amistad por encima de todo.

A todos los que de una forma u otra también han sido parte de este logro.

LISSETTE RODRÍGUEZ VERDECIA

RESUMEN

La Ingeniería de Requisitos (IR) abarca la primera fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de ella es garantizar que los requisitos del sistema a desarrollar sean consistentes con las necesidades de la organización donde se utilizará el mismo y con las expectativas y necesidades de los clientes y/o usuarios.

En el trabajo se propone un modelo para la Ingeniería de Requisitos, en los proyectos de desarrollo de Software de Gestión (SWG) de la facultad 3. Teniendo en cuenta que este tipo de software utiliza grandes cantidades de información almacenadas en base de datos con objeto de facilitar las transacciones comerciales o la toma de decisiones, se propone que el modelo esté orientado a un sistema de procesos y a elementos que apoyan este sistema, como: técnicas, plantillas y herramientas. El funcionamiento de dicho modelo está basado en una espiral, cuyo principal aporte es brindar un desempeño iterativo e incremental de los procesos que comprende la Ingeniería de Requisitos, facilitando así un mejor procesamiento de la información en el SWG que se implemente.

Al aplicar este modelo en la primera fase del proceso de ingeniería del software en un SWG, se refina la declaración general del ámbito del software en una especificación final de los requisitos que se convierte en el fundamento de las siguientes actividades que comprende la ingeniería del software.

El modelo propuesto representa una teoría a seguir sobre el funcionamiento de los procesos de la Ingeniería de Requisitos y los resultados a obtener.

Palabras Clave: Ingeniería del Software, Ingeniería de Requisitos (IR), Software de Gestión (SWG).

ABSTRACT

The Requirements' Engineering (RE) covers the first phase inside the development of a data processing system. One of its most important challenges is to guarantee that the requirements of the system to develop are consistent with the needs of the organization where it will be used and with the expectations and needs of the clients and/or users.

In the work a model for the Requirements' Engineering (RE) is proposed, in the Management Software development projects (MSW) of the faculty 3. Keeping in mind that this type of software utilizes large quantities of information stored in database, with purpose is facilitating the commercial transactions or decisions taking, is proposed that the model be oriented to a system of processes and to elements that support this system, such as: technical, staffs and tools. The operation of the previously mentioned model is based on a spiral, whose main one contribution is to offer an iterative and incremental performance of the processes that understands the Requirements' Engineering, facilitating thus a better prosecution of the information in the MSW witch is implemented by itself.

Applying this model in the first phase of the process of the software's engineering in a MSW, the general statement of the environment of the software in a final specification of the requirements is refined; witch becomes the base of the following activities that understands the engineering of the software.

The model proposed represents a theory to continue on the operation of the processes of the Engineering of Requirements and the results to obtain.

Keywords: Software's Engineering, Requirements' Engineering (RE), Management Software (MSW).

ÍNDICE DE ILUSTRACIONES

Figuras

Figura 1-1: Proceso de IR.	11
Figura 1-2: Modelo en Cascada. Tomada de (Dávila, 2001)	16
Figura 1-3: Modelo en Espiral. Tomada de (Dávila, 2001).....	17
Figura 1-4: Actividades de la Gestión de Requisitos (García Crespo, y otros, 2005) ..	24
Figura 2-1: Representación visual del Modelo de IR definido.....	35
Figura 2-2: Representación visual del Sistema de Procesos.....	36
Figura 2-3: Representación visual de los elementos de apoyo.	37
Figura 2-4: Representación en espiral de los procesos de la IR.	38
Figura 2-5: Organización del equipo de IR.	40
Figura 2-6: Estrategia de comunicación en la IR.....	45
Figura 2-7: Proceso de Obtención de Requisitos.....	48
Figura 2-8: Proceso de Análisis.	50
Figura 2-9: Proceso de Especificación y Descripción.	52
Figura 2-10: Proceso de Validación de Requisitos.....	54
Figura 2-11: Proceso de Control de Cambios en los requisitos.....	56

Tablas

Tabla 1-1: Resumen de las técnicas usadas en la IR.	30
Tabla 3-1: Control de versiones de los requisitos.	65
Tabla 3-2 : Matriz de Seguimiento de requisitos.	66
Tabla 3-3: Tabla de seguimiento del estado de los requisitos.	66
Tabla 3-4: Resultados del cuestionario de validación	69

TABLA DE CONTENIDO

AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN.....	V
ABSTRACT.....	VI
INTRODUCCIÓN.....	1
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	8
1.1 Conceptos fundamentales.....	8
1.2 Estado de la Ingeniería de Requisitos en los proyectos de desarrollo de SWG de la facultad 3.	13
1.3 Tipos de Requisitos de Software.....	14
1.4 Modelos aplicables a la IR en el ciclo de vida del software.....	16
1.5 Actividades de la Ingeniería de Requisitos	19
1.6 Técnicas de la Ingeniería de Requisitos.	24
1.7 Herramientas de la Ingeniería de Requisitos.	31
1.8 Conclusiones del capítulo.....	34
2. CAPÍTULO 2: MODELO DEFINIDO. SISTEMA DE PROCESOS	35
2.1 Definición del Modelo de Ingeniería de Requisitos.....	35
2.2 Definición del equipo de desarrollo en el proceso de IR y las responsabilidades de cada rol.	39
2.3 Estrategia de comunicación en el proceso de IR.....	44
2.4 Desarrollo de Requisitos.....	46
2.5 Administración de Requisitos. Actividades.	54
2.6 Conclusiones del capítulo.....	59
3. CAPÍTULO 3: MODELO DEFINIDO. ELEMENTOS DE APOYO	60
3.1 Definición de las Técnicas a usar en cada etapa.	60
3.2 Definición de las Técnicas a usar en la Administración.....	64
3.3 Definición de Herramientas.	67
3.4 Determinación del valor práctico del modelo propuesto	69
3.5 Conclusiones del capítulo.....	70
4. CONCLUSIONES.....	71
5. RECOMENDACIONES.....	72

6.	BIBLIOGRAFÍA	73
7.	ANEXOS	80
8.	GLOSARIO DE TÉRMINOS Y SIGLAS	87

INTRODUCCIÓN

El incesante desarrollo de las tecnologías de la información y las comunicaciones (TIC) ofrece grandes posibilidades para acelerar e incrementar el desarrollo económico de un país.

Cuba se ha planteado, como uno de los objetivos más importantes para este milenio, la informatización de la sociedad, que se define como el proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad.

En aras de preparar los recursos humanos especializados en el uso de las TIC se creó a mediados del 2002 la Universidad de las Ciencias Informáticas (UCI), como universidad de nuevo tipo, que combina el estudio con la producción y la investigación.

La UCI está llamada a convertirse en una significativa fuente de ingresos al país y para el logro de este objetivo empresas cubanas vinculadas a la universidad trabajan en la materialización de proyectos productivos asociados al programa cubano de informatización.

En tal dirección, la facultad 3 de dicha universidad, labora entre otros polos de producción, en proyectos de desarrollo de software de gestión.

Se reconocen como SWG, aquellos programas basados en múltiples aplicaciones (denominados comúnmente suites), independientes o modulares, que gestionan información y no la centralizan en una única base de datos.

El SWG maneja habitualmente programas o módulos de facturación y contabilidad, puede intervenir además en el control de muchas actividades de negocios como ventas,

entregas, pagos, producción, administración de inventarios y calidad de administración.

Los objetivos principales del SWG son:

- Optimizar procesos empresariales.
- Acceder a toda la información de forma confiable, precisa y oportuna (integridad de datos).
- Eliminar datos y operaciones innecesarias o redundantes.
- Reducir el tiempo y el coste de los procesos, mediante procesos de reingeniería.

Características del SWG desarrollado en la Facultad 3.

1. Manejo de alto nivel de transacciones de manera confiable.
2. Multimoneda.
3. Multiusuario.
4. Bases de datos configurables.
5. Desarrollo en módulos.
6. Posibilidad de crear múltiples documentos de salida (factura, presupuesto, orden de compra, entre otros).
7. Construcción de sólidos perfiles de usuario que permiten una fácil administración del sistema y su mantenimiento.
8. Estadísticas y gráficos acumulables.
9. Histórico de operaciones.
10. Integración contable.

El SWG desarrollado en la facultad 3, necesita un modelo de IR que adaptado a sus características garantice la calidad del mismo.

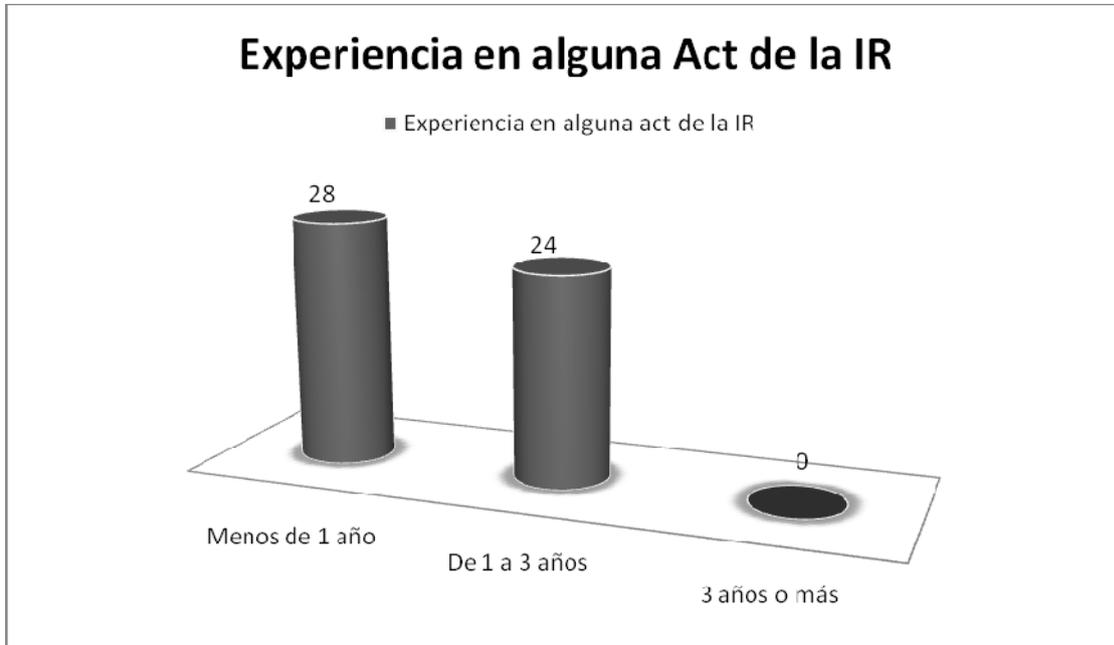
Para obtener un modelo que guíe el proceso de IR e identifique cuáles son las necesidades del usuario y las condiciones que el SWG debe cumplir, se hace un estudio de la problemática existente en los proyectos de desarrollo de SWG en la facultad 3 de la

UCI, donde se realizaron 56 entrevistas en 5 proyectos de este tipo, las cuales arrojaron los siguientes resultados. La entrevista realizada se puede ver en el Anexo1.

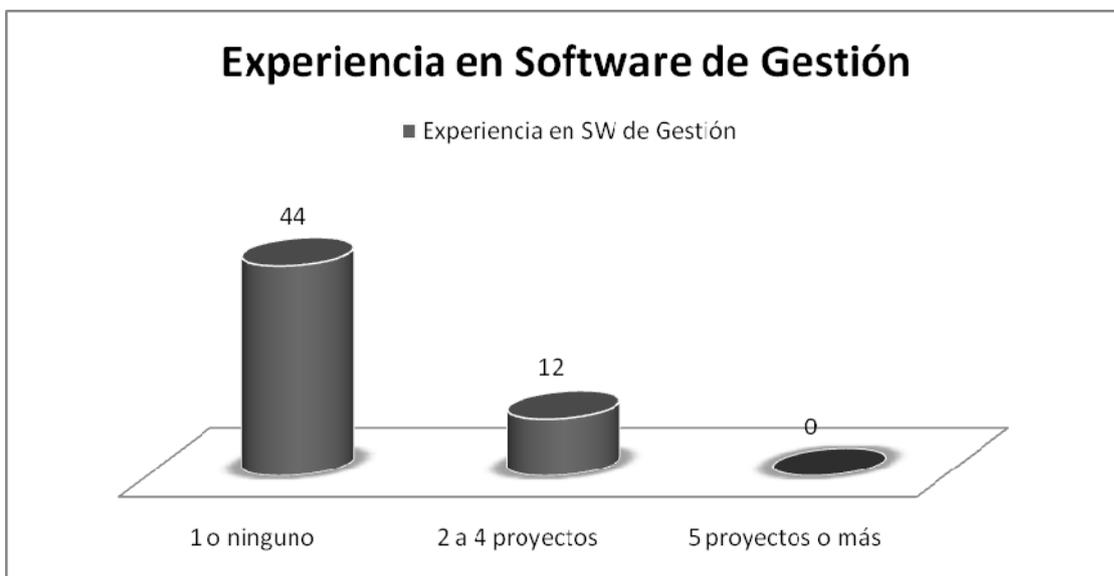


Gráfica 1: Cantidad de entrevistados que han realizado actividades del proceso de IR.

Como se puede apreciar, de los 56 entrevistados, el número máximo realizando alguna actividad, es 40. Se pueden considerar como las actividades más críticas en este proceso las siguientes: Derivar requisitos, clasificar requisitos, Dar seguimiento, gestionar y validar. Esto demuestra que el proceso de IR en estos proyectos no se desarrolla con la calidad requerida y que el software que se está desarrollando no cumple al máximo las expectativas del cliente. Las actividades que se consideran como críticas, son unas de las más importantes en el proceso de producción del software.



Gráfica 2: Experiencia de los entrevistados en la realización de alguna de las actividades del proceso de IR.



Gráfica 3: Experiencia de los entrevistados en la realización de alguna de las actividades del proceso de IR en proyectos de desarrollo de SWG.

Las gráficas 2 y 3 muestran la inexperiencia que existe en la realización de las actividades de la IR en proyectos de desarrollo de SWG, pues el mayor número de personas entrevistadas se encuentran en el rango de menos de un año realizando alguna actividad y ninguno supera los 3 años de experiencia. De igual manera se comporta el número de entrevistados, respecto a la participación en proyectos de desarrollo de SWG, donde la gran mayoría que se encuentra en el rango de 1 o ningún proyecto supera a una minoría que se encuentra en el rango de 2 a 4 proyectos y ninguno ha participado en al menos 5 proyectos.

Los gráficos ilustran de forma general, las condiciones actuales de los proyectos de desarrollo de SWG en la facultad 3. Tal situación ha provocado generalmente un proceso de IR con poca calidad, resumiendo que:

- Existen cinco actividades críticas en el proceso de IR.
- Los requisitos obtenidos no cumplen las características requeridas.
- Desconocimiento del equipo de desarrollo de dichas características.
- Los requisitos no cumplen con todas las expectativas del cliente.
- Inexperiencia de los integrantes del equipo de desarrollo en la realización de las actividades de la IR y en la participación en proyectos de desarrollo de SWG.

Definición del Problema:

A partir de la situación problemática existente se ha planteado el siguiente **problema:** Insuficiencias en la aplicación de modelos de IR en los proyectos de desarrollo de SWG en la facultad 3.

Se ha definido como **objeto de estudio** La Ingeniería de Software.

Del objeto de estudio analizado se ha definido como **campo de acción** La IR en los proyectos de desarrollo de SWG de la facultad 3.

Objetivo General: Proponer un Modelo para la IR en los proyectos de desarrollo de SWG de la facultad 3.

Objetivos Específicos:

1. Definir un sistema de procesos.
2. Definir técnicas, plantillas y herramientas como elementos de apoyo al sistema de procesos.
3. Validar el Modelo propuesto.

Hipótesis: Si se aplica un Modelo de IR mejorará la calidad de este proceso, en los proyectos de desarrollo de SWG de la facultad 3.

Tareas Investigativas:

1. Realizar un estudio del estado del arte sobre la IR y las características que hacen posible que el SWG desarrollado en la facultad 3, necesite un proceso único de este tipo.
2. Desarrollar un Modelo para la IR en los proyectos de desarrollo de SWG de la facultad 3, a partir de los modelos existentes.
3. Escoger un método de validación para el modelo propuesto.

Variables de la Investigación.

Variable Independiente: modelo de IR en SWG.

Variables dependientes: calidad del proceso de IR.

Se utiliza como estrategia de investigación la Investigación explicativa o experimental. Se hace uso de los métodos teóricos: histórico-lógico, hipotético-deductivo, analítico-sintético, de los métodos empíricos: la observación y de los métodos particulares: la entrevista.

Aportes prácticos esperados del trabajo.

La investigación debe dar la posibilidad de contar con un Modelo que guíe el proceso de IR en los proyectos de desarrollo de SWG de la facultad 3, brindando las actividades a seguir en cada uno de los procesos definidos, los documentos o artefactos a obtener, las plantillas a aplicar y las técnicas y herramientas sugeridas a utilizar.

Estructura de la tesis.

Para el desarrollo del tema de investigación se proponen tres capítulos, los cuales están estructurados de la siguiente manera:

En el capítulo 1 se hace un análisis crítico del estado actual de los proyectos de desarrollo de SWG en la facultad 3 y de investigaciones anteriores, entorno a las actividades, modelos, técnicas y herramientas usadas en la IR, teniendo en cuenta las características específicas que hacen que el SWG necesite un proceso único de IR.

En el capítulo 2, se conforma el sistema de procesos del modelo definido, se especifican las actividades, objetivos y resultados de cada una de las etapas de la IR, siguiendo el funcionamiento del modelo espiral.

En el capítulo 3, se relacionan elementos de apoyo como técnicas, plantillas y herramientas que dan soporte a los procesos descritos en el capítulo anterior. Además se valida la aplicabilidad del modelo definido.

1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el siguiente capítulo se realiza un análisis crítico del estado actual de los proyectos de desarrollo de SWG en la facultad 3 y de investigaciones anteriores relacionadas con la IR entorno a las siguientes temáticas:

- Actividades
- Modelos
- Técnicas y herramientas

1.1 Conceptos fundamentales

Para analizar la IR, se considera importante tener claros algunos conceptos necesarios para entender en qué consiste la misma.

"Software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo" (Lewis, 1994)

"Un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software" (Lewis, 1994) que en palabras más específicas, se considera que:

"la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" (Cota, 1994)

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" (Jacobson, 1998)

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requisitos de software, estos requisitos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (Jacobson, 1998)

En el proceso de desarrollo de un sistema, el equipo de desarrollo se enfrenta al problema de la identificación de los requisitos y tal es así que en uno de los párrafos más citados en la bibliografía de Ingeniería de Software se expresa:

"La parte más dura en la construcción de un sistema software es decidir cómo construirlo... ninguna parte del trabajo mutila el resultado del sistema si está hecho mal. Ninguna parte es más dificultosa para rectificarlo después", Según Fred Brooks en (Pressman, 2005)

La afirmación anterior expresa la necesidad de identificar y especificar correctamente los requisitos de un sistema software en el proceso de construcción del mismo.

Se entiende por requisito: a la cualidad necesaria de un sistema, una declaración que identifique una capacidad, una característica, o un factor de calidad de un sistema que proporcione valor y utilidad al cliente o usuario. Los requisitos son importantes porque proporcionan la base para todo el trabajo de desarrollo que le sigue al software (Ralph, 2004)

Un requisito existe ya sea porque el tipo de producto demanda ciertas funciones o cualidades, o porque el cliente quiere que ese requisito sea parte del producto final. Así que si no se tienen los requisitos correctos, no se puede diseñar o construir el producto correcto y, consecuentemente, el producto no permitirá a los usuarios finales realizar su trabajo. Y esto está confirmado por estudios que demuestran que más del 60% de los errores de diseño se originan durante las etapas de requisitos y análisis (Dávila, 2001)

Los requisitos se pueden dividir en requisitos funcionales y no-funcionales. Los funcionales definen qué hace el sistema (describen todas las entradas y salidas), es decir,

las funciones del sistema. Los no-funcionales definen los atributos que le indican al sistema cómo realizar su trabajo (eficiencia, hardware, software, interfaz, usabilidad, etc.); es el cómo, cuándo y cuánto de qué (Dávila, 2001)

Después de haber analizado qué significan los requisitos del software y los conceptos fundamentales y necesarios para entender qué es la IR, se presentan definiciones que ayudan a comprender esta disciplina.

“La Ingeniería de Requisitos es la disciplina encargada de establecer los servicios que un sistema debe suministrar y las restricciones bajo las cuales debe operar” (Somerville, 1996)

“El proceso de Ingeniería de Requisitos puede ser descrito en 5 pasos distintos: Identificación de Requisitos, Análisis de Requisitos y Negociación, Especificación de Requisitos, Modelizado del Sistema, Validación de Requisitos y Gestión de Requisitos” (Somerville, y otros, 1997)

Entonces, "Ingeniería de Requisitos" se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requisitos para un producto determinado. El uso del término ingeniería implica el uso de técnicas sistemáticas y repetibles que aseguren que los requisitos del sistema sean completos y consistentes.

Se puede definir como proceso, al conjunto organizado de actividades que transforman entradas en salidas. En el caso de un sistema software la definición de proceso de IR sería como muestra la figura siguiente:

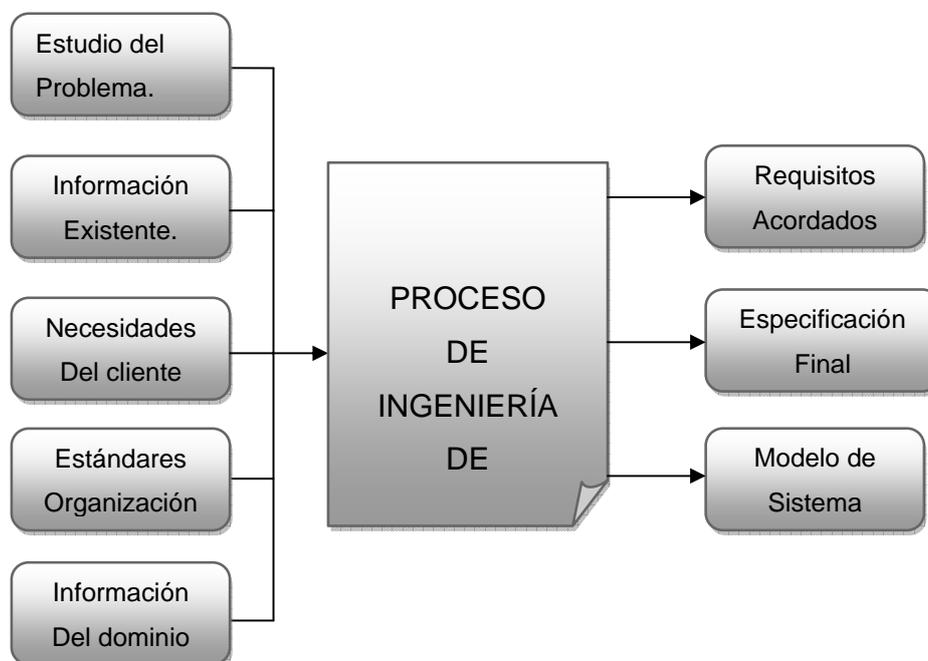


Figura 1-1: Proceso de IR.

Estudio del problema: Es la descripción de forma sencilla y clara de los procesos que ocurren dentro del entorno donde se enmarca el problema,

Información existente: Se trata de información relativa a la funcionalidad de los sistemas que deben ser sustituidos por el sistema software en desarrollo.

Necesidades del cliente: Descripción de lo que los usuarios del sistema en estudio necesitan como soporte de su trabajo.

Estándares de la organización: Son los estándares utilizados en la organización relativos a las prácticas de desarrollo, de aseguramiento de la calidad, de controles; con lo que el sistema en estudio debe ser conforme.

Información sobre el dominio: Información general sobre el dominio de aplicación del sistema.

Requisitos acordados: Una descripción de los requisitos del sistema, escrita de modo comprensible por el cliente, y que han sido acordados con él.

Especificación del sistema: Es el documento final de especificación de los requisitos que brinda una descripción más detallada de las funcionalidades del sistema.

Modelo del sistema: Un conjunto de modelos, tales como diagramas de flujos de datos, modelos de casos de uso y modelos de procesos, que describen el sistema desde diferentes perspectivas.

La Ingeniería de Requisitos cumple un papel primordial en el proceso de producción de un software, porque enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que faciliten el mecanismo apropiado para comprender lo que quiere el cliente, describiendo con claridad, de forma consistente y compacta el comportamiento del sistema, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transforme en un sistema operacional (Pressman, 2005)

En resumen de los conceptos y definiciones que anteriormente se dan se comprende que un aspecto importante es el análisis del problema y la comprensión por parte de los desarrolladores del entorno en el cual se desarrolla el cliente, para poder identificar, especificar y documentar las necesidades de dicho cliente y posteriormente gestionar sus cambios. En este proceso de descubrimiento de las necesidades del cliente se utilizan técnicas, herramientas y la opinión de muchas personas, por lo que se caracteriza como un proceso centrado en las personas. Uno de los aspectos más importantes de ella es la comunicación, característica ésta que vuelve el proceso complejo por la alta presencia del factor humano y es la responsable de que la disciplina contenga aspectos sociales y culturales y no sólo de índole técnica.

1.2 Estado de la Ingeniería de Requisitos en los proyectos de desarrollo de SWG de la facultad 3.

Se realizaron 56 entrevistas en 5 proyectos de desarrollo de SWG de la facultad 3. A partir de ellas se han obtenido los siguientes resultados:

- Los miembros del equipo de desarrollo de los proyectos, en su mayoría, dominan el concepto de IR, aunque hay que señalar que no con profundidad, solo rasgos elementales.
- Todos los arquitectos de los proyectos asignan a los requisitos un nivel de prioridad Alto dentro del proceso de desarrollo de software.
- En el 50% de los proyectos entrevistados existe un documento de especificación de requisitos, pero hay que señalar que no existe una plantilla estándar para su confección.
- En la mayoría de los proyectos no se utilizan herramientas para el tratamiento de los requisitos y una minoría utiliza algunas funcionalidades del Requisite Pro.
- Las técnicas para el tratamiento de los requisitos más usadas son las entrevistas, los talleres y la tormenta de ideas.
- Solo en 2 de los proyectos entrevistados se conoce el modelo que se emplea para el tratamiento de los requisitos, en un caso Modelo en Cascada y en el otro el modelo Prototipado.
- En todos los proyectos entrevistados se utiliza RUP como metodología de desarrollo de software.
- Las actividades que más se realizan dentro del proceso de ingeniería de requisitos de un software son la obtención y el refinamiento de los requisitos, a menor escala se realizan las actividades de asignar requisitos por subsistemas de diseño y la gestión de los requisitos. En ningún caso se realiza la actividad de derivar requisitos.
- Una parte del equipo de desarrollo desconoce las características que determinan la calidad de los requisitos de un proyecto. En el 50% de los proyectos la lista de los requisitos actualmente no abarca todas las necesidades del cliente.

- La mayoría de los entrevistados llevan menos de 1 año realizando alguna actividad del proceso de IR y solo han realizado alguna de estas actividades a lo sumo en 1 proyecto de desarrollo de SWG.

1.3 Tipos de Requisitos de Software.

Los requisitos del sistema son las condiciones, funcionalidades o restricciones que debe tener el software que se implementa, los mismos deben cumplir con las exigencias del cliente o usuario final.

Existen dos tipos de Requisitos en un sistema, los requisitos funcionales y los requisitos no funcionales, estos últimos se dividen además en diferentes categorías.

1.3.1 Requisitos Funcionales.

Los requisitos funcionales describen lo que el sistema o el software deben hacer. La funcionalidad es la capacidad útil proporcionada por uno o más componentes de un sistema. A veces se le llaman a los requisitos funcionales conductual u operacional porque ellos especifican las entradas (los estímulos) al sistema, los rendimientos (las contestaciones) del sistema, y las relaciones conductuales entre ellos. En algunos casos, los requisitos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer. (Ralph, 2004)

1.3.2 Requisitos No Funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. (Ralph, 2004)

Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto.

Requisitos de Software: debe mencionarse el software del que se debe disponer, después de implementado el sistema.

Requisitos de Hardware: se deben enunciar los elementos de hardware que se necesitan para que el software cumpla sus funcionalidades.

Restricciones en el diseño y la implementación: especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Requisitos de apariencia o interfaz externa: este tipo de requisito describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. También pueden ser necesidades de cumplir con normas estándares, o con los estándares de la empresa para la cual se esté desarrollando el software.

Requisitos de Seguridad: este es el tipo de requisito más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:

- **Confidencialidad:** La información manejada por el sistema esta protegida de acceso no autorizado y divulgación.
- **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
- **Disponibilidad:** Significa que los usuarios autorizados se les garantizará el acceso a la información.

Requisitos de Usabilidad: describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

Requisitos de Soporte: abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como lograr su mejoramiento progresivo y evolución en el tiempo.

1.4 Modelos aplicables a la IR en el ciclo de vida del software.

Un modelo es una simplificación de la realidad que incluye aquellos elementos que tienen una gran influencia y omite aquellos elementos que no son relevantes para el nivel de abstracción dado. (Dávila, 2001)

En resumen, los modelos son abstracciones simplificadas y estandarizadas de actividades repetitivas, generalmente producidos desde un punto de vista determinado, por lo que pueden existir diferentes modelos para un mismo proceso. Sin embargo, en el caso del proceso de IR y desde una perspectiva "intelectual", podemos decir que todos esos diversos modelos parten de una misma base, un modelo "madre" que llamaremos "modelo-abstracto" (Dávila, 2001)

1.4.1 Modelo en Cascada.

Este modelo sugiere que los resultados de una tarea del proceso llevan a la siguiente, y así sucesivamente. En el ejemplo presentado, la extracción lleva al análisis, el análisis desencadena la documentación, y la documentación inicia la validación.

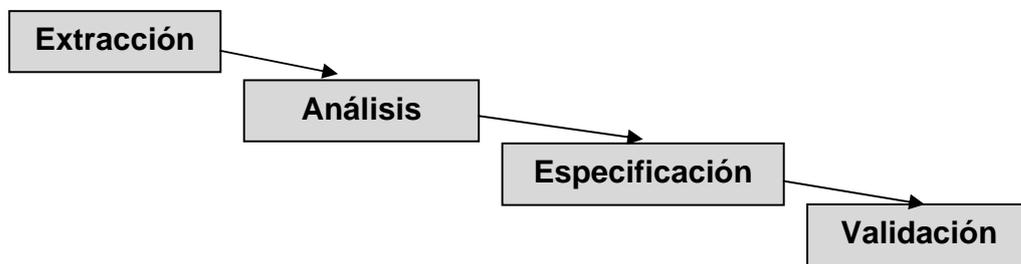


Figura 1-2: Modelo en Cascada. Tomada de (Dávila, 2001)

1.4.2 **Modelo en Espiral.**

Un modo alternativo de presentar modelos de actividad que toma en cuenta la retroalimentación entre etapas y la repetición de tareas, es el llamado Modelo en Espiral. (Sommerville, y otros, 1998)

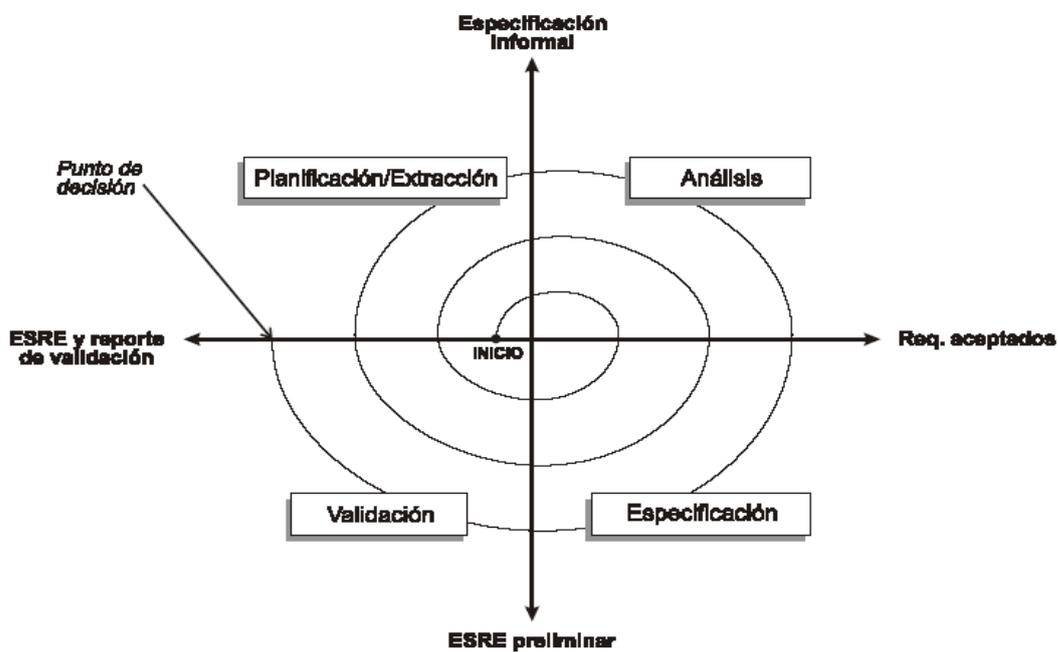


Figura 1-3: Modelo en Espiral. Tomada de (Dávila, 2001)

En este diagrama, el uso de la espiral implica que las diferentes actividades de la Ingeniería de requisitos son repetidas hasta que se toma la decisión final, que es la aceptación del documento de especificación de requisitos. (Dávila, 2001)

Antes de cada nueva iteración se procede a evaluar la anterior. Se valora hasta qué punto se han alcanzado los objetivos marcados y se han respetado las limitaciones impuestas. También se revisan los planes elaborados para el próximo ciclo y los riesgos detectados. Aumentando el control se pretende reducir los posibles inconvenientes que pudieran surgir durante cada ciclo.

La monitorización del proceso sirve también para identificar el momento en que la espiral debe terminar. Aunque las condiciones de cada ciclo llevan implícitas los objetivos finales del proceso, es necesaria una evaluación continua para asegurar el cumplimiento de las condiciones de finalización del trabajo.

1.4.3 **Modelo Prototipado.**

Un prototipo es un modelo a escala reducida de la solución final y sirve para verificar que las especificaciones han sido construidas de acuerdo a los requisitos del sistema. La función de un prototipo es obtener un modelo del sistema que se va a desarrollar.

Este modelo simula el comportamiento externo, aquél al que se enfrentará el usuario, del nuevo sistema. Es por ello que el prototipo se desarrolla siguiendo las indicaciones del usuario final y es el complemento de los requisitos. La principal ventaja que se obtiene al usar este tipo de elementos es que se dispone en una fase muy temprana de unos requisitos completos, y de un modelo que facilita la construcción del sistema, en cuanto a la interfaz de usuario. (Sommerville, 1996)

1.4.4 **Sumario**

El Modelo en Cascada puede ser muy útil para obtener una visión general del proceso, pero, lo que se hace en un proceso de IR es mucho más complejo que lo que se muestra en la figura, no hay una definición clara entre las fases, pues una depende de la otra. Es preciso aclarar que los requisitos del sistema pueden cambiar durante el ciclo de vida del software y esto significa que haya que volver hacia atrás cuando se este realizando el proceso de IR.

En el Modelo en Espiral, si se presenta un problema en la definición inicial que se haga de los requisitos, se recorre el ciclo nuevamente hasta que sea resuelto completamente, y este ciclo no se termina hasta que no se llegue a un documento que sea aceptable por el usuario. En este modelo se obtienen puntos de decisión cada vez que se completan las actividades de la IR, donde se analiza si el proceso culminado cumple las expectativas y calidad suficiente. Este Modelo es el aplicable al desarrollo de un software de gestión,

pues da la posibilidad de la retroalimentación, la evaluación continua del proceso y de lograr un documento de especificación de requisitos con la calidad suficiente. Muestra la idea más real de lo que está sucediendo en el desarrollo de las actividades del proceso de IR, brindando la posibilidad de repetir el ciclo cuantas veces sea necesario para lograr así un buen resultado.

El Modelo Prototipado sólo se debe usar como base para el trabajo posterior, sería un error tomarlo como primera versión del software que se está construyendo.

1.5 Actividades de la Ingeniería de Requisitos

Este epígrafe tratará las actividades de acuerdo a las necesidades y características del SWG. Por ello se han definido 5 actividades bases en este proceso: Obtención, Análisis, Especificación y Descripción, Validación y Gestión.

1.5.1 Obtención de Requisitos

La obtención de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. (Díez, 2001)

Antes que los requisitos puedan ser analizados, modelados o especificados, deben ser recogidos a través de un proceso de obtención. La técnica de obtención más usada es llevar a cabo una reunión o entrevista preliminar. (Pressman, 2005)

El proceso de obtención de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la IR ha trabajado desde años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa, las mismas se tratan con más detalle en el epígrafe 1.5.

El resultado alcanzado como consecuencia de la obtención de requisitos variará dependiendo del tamaño del sistema o producto a construir. Para grandes sistemas, el producto obtenido debe incluir (Pressman, 2005)

- Una relación de necesidades y características;
- Un informe conciso del alcance del sistema o producto;
- Una lista de clientes, usuarios y otros intervinientes que deben participar en esta actividad de obtención de requisitos;
- Una descripción del entorno técnico del problema;
- Una relación de requisitos (perfectamente agrupados por funcionalidad) y las restricciones del dominio aplicables a cada uno;
- Un conjunto de escenarios que permiten profundizar en el uso del sistema o producto bajo diferentes condiciones operativas; y
- Cualquier prototipo desarrollado para definir mejor los requisitos.

1.5.2 Análisis de Requisitos

Sobre la base de la obtención realizada previamente, comienza esta fase que se presenta sumamente compleja en un proyecto donde el dominio es desconocido en la cual se apunta a descubrir problemas con los requisitos del sistema identificados hasta el momento.

El Análisis de requisitos permite especificar las características operacionales del software (función, datos y rendimiento), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software. (Pressman, 2005)

Los requisitos se agrupan por categorías y se organizan en subconjuntos, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad, y se clasifican en base a las necesidades de los clientes o usuarios. (Pressman, 2005)

1.5.3 Especificación y Descripción de Requisitos.

Una especificación puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de escenarios de uso, un prototipo o una colección de lo anteriormente citado. (Pressman, 2005)

En esta actividad se documentan los requisitos acordados con el cliente, en un nivel apropiado de detalle. Para ello existen un gran número de técnicas propuestas. Se describen brevemente las más relevantes para este trabajo.

Documento de Especificación de Requisitos: la correcta especificación de requisitos de software (ERS) determina en gran medida el éxito de un proyecto de desarrollo de software. Basado en el estándar IEEE Std. 830:1998: Práctica Recomendada para la ERS, las especificaciones de los requisitos que se incluyan en un documento de este tipo deben tener las siguientes características (IEEE):

Correctas: Una ERS es correcta si, y solo si, cada requisito definido debe ser satisfecho por el sistema software.

No ambiguas: Una ERS es no ambigua si, y solo si, cada requisito definido en la misma tiene una sola interpretación. Como mínimo, ello exige que cada característica del producto final sea descrita utilizando un único término. En los casos en que el término usado en un contexto particular pueda tener múltiples significados, debe ser incluido en un glosario, donde se aclare su significado específico.

Completas: Una ERS es completa si, y solo si, incluye los siguientes elementos:

- Todos los requisitos significativos, sean relativos a funcionalidad, rendimiento, restricciones de diseño o interfaces externas.
- Definición de la respuesta del sistema software, todas las clases y tipos de entradas de datos en cualquier situación que pueda presentarse. Hay que tener presente que esto supone que deben especificarse tanto las respuestas a entradas válidas como las que no lo son.
- Referencia de todas las figuras, diagramas, tablas, etc.,... y definición de todos los términos y unidades de medida.

Consistentes: La consistencia se refiere a la conformidad de las ERS con documentos de más alto nivel. Si no hay esta conformidad es que las ERS no son consistentes.

Clasificadas por su relevancia o importancia: Los requisitos relativos a los sistemas software no son todos igualmente importantes. Algunos pueden ser imprescindibles, otros pueden ser deseables.

Cada requisito en las ERS debe ser identificado para clarificar estas diferencias y hacerlas explícitas. Una posible clasificación se refiere, por ejemplo, al grado de necesidad:

- Esencial: Imprescindible
- Condicional: Pueden darse condiciones para que dicho requisito este o no presente.
- Opcional: Implica una clase de requisitos que pueden o no existir.

Verificables: Una ERS es verificable si y solo si cada uno de los requisitos definidos es verificable. A su vez, un requisito es verificable si, y solo si, existe algún procedimiento por el que una persona o máquina puede chequear que el producto software cumple los requisitos. En general un requisito ambiguo es no verificable.

Modificable: Una ERS es modificable si, y solo si, su estructura y estilo son tales que cualquier cambio de los requisitos puede hacerse fácil, completa y consistentemente manteniendo la misma estructura y estilo.

Trazables: Las ERS son trazables si el origen de cada requisito es claro y si facilita la referencia del mismo para futuros desarrollos o modificaciones.

1.5.4 Validación de Requisitos.

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Lowe, y otros, 1999).

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto. (Pressman, 2005)

La validación representa un punto de control interno y externo; interno, porque se debe verificar internamente lo que se está haciendo, y externo, porque se debe validar con el cliente.

En resumen, la validación de especificaciones significa asegurarse de que el documento de requisitos represente una descripción clara del sistema, y es una verificación final de que los requisitos cubren las necesidades de los usuarios.

Esta etapa puede confundirse con la de análisis, pero la diferencia es clara: mientras que en el análisis se trabaja sobre el boceto del documento de requisitos, en la validación se utiliza el documento final.

1.5.5 **Gestión de Requisitos.**

Un proyecto de desarrollo de SWG conlleva a que el producto software viva en un mundo dinámico que obliga a realizar cambios que pueden afectar a los sistemas o módulos en funcionamiento como a los que están en vías de desarrollo, si estos cambios no se gestionan de manera adecuada puede generar: que el producto no se entregue en el tiempo establecido, que se cancele el proyecto por no cumplir con los requisitos o que el cambio de requisitos consuma más recursos (persona, máquinas etc.), afectando de manera general la calidad de este proceso.

Los requisitos del sistema cambian y el deseo de cambiarlos persiste a lo largo de la vida del sistema. La Gestión de Requisitos es el conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento. (Pressman, 2005)

Básicamente, consiste en gestionar los cambios a los requisitos acordados, las relaciones entre ellos, las dependencias entre la ERS y otros documentos producidos por el proceso de desarrollo de software. Esta actividad asegura la consistencia entre los requisitos y el sistema construido (o en construcción). Consume grandes cantidades de tiempo y esfuerzo.

El control de cambios implica: propuesta de cambio, análisis de impactos, comunicación, incorporación; dentro del control de versiones tenemos: identificar las versiones de los documentos; dentro del seguimiento de los requisitos debemos: definir la relación con los otros requisitos, definir la relación con otros elementos del sistema; y, dentro del seguimiento del estado de los requisitos debemos: definir el estado de todos y cada uno de los requisitos. Para comprender mejor las actividades de la gestión de requisitos antes explicadas se muestra la siguiente figura:



Figura 1-4: Actividades de la Gestión de Requisitos (García Crespo, y otros, 2005)

1.6 Técnicas de la Ingeniería de Requisitos.

Existen numerosas técnicas para realizar cada una de las actividades de la IR, en este epígrafe se tratan las más relevantes para este trabajo.

1.6.1 Técnicas para la Obtención de Requisitos.

Entrevistas y Cuestionarios: resulta una técnica muy aceptada dentro de la IR y su uso está ampliamente extendido. La entrevista, sin embargo, no es una técnica sencilla de aplicar. Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Bajo este aspecto la preparación de la entrevista representa un papel esencial. (Escalona, y otros, 2002)

Para realizar las entrevistas, conviene llevar preparado un cuestionario. En términos generales, un cuestionario consiste en un conjunto de preguntas presentadas a una persona para su respuesta. La forma de la pregunta puede influir en las respuestas, por lo que hay que planearlas cuidadosamente. Las preguntas suelen distinguirse en dos categorías: abiertas y cerradas. Las preguntas abiertas permiten que los encuestados respondan con su propia terminología. Generalmente estas son más reveladoras, ya que los interrogados no están limitados en sus respuestas. Son especialmente útiles en la etapa exploratoria de la investigación. (Dávila, 2001)

Las preguntas cerradas predeterminan todas las posibles respuestas y el interrogado elige entre las opciones presentadas.

A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Permiten al analista tomar conocimiento del mismo y comprender los objetivos de la solución buscada.

JAD: (Joint Application Development, Desarrollo conjunto de aplicaciones) resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la entrevista se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado. Pero requiere un grupo de participantes bien integrados y organizados. (Escalona, y otros, 2002)

Mapas Conceptuales: un modelo conceptual es una representación de conceptos del dominio del problema. (Fowler, 1996) Permite mostrar conceptos, asociaciones entre conceptos y atributos de conceptos. La creación del modelo también ayuda a comprender la terminología del dominio y comunica cuáles son los términos importantes y las relaciones existentes entre ellos.

Concepto: categoría de idea o cosas. La intención del concepto es la descripción de sus atributos, operaciones y significado. (Craig, 1999)

Son muy usados dentro de la IR pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sugestivos y pueden llegar a ser ambiguos en casos complejos si no se acompaña de una descripción textual. (Escalona, y otros, 2002)

Esta técnica es muy buena para obtener un idea general de cómo funciona el negocio, las relaciones que se establecen en él y el vocabulario que se emplea. También es de ayuda para incluir nuevos conceptos al glosario de términos.

Tormenta de ideas (Brainstorming): este es un modelo que se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requisitos para el sistema. No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando sobre la base de distintos criterios como, por ejemplo, "caro", "impracticable", "imposible", etc. (Arango, y otros, 2004)

Casos de Uso: los casos de uso permiten mostrar el entorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede

interactuar con varios actores. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada. (Escalona, y otros, 2002)

Comparación de Terminologías: uno de los problemas que surge durante la obtención de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste) (Pan, y otros, 2001).

1.6.2 Técnicas para la Especificación y Descripción.

Lenguaje Natural: resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. Pero, a pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando (Escalona, y otros, 2002)

Lenguajes Formales: esta técnica es el extremo opuesto al lenguaje natural, es la utilización de lenguajes formales para describir los requisitos de un sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción formal, han sido aplicadas en el mundo de la IR desde hace años. Sin embargo, resultan muy complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario, es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas. (Escalona, y otros, 2002)

Plantillas o patrones: esta técnica tiene como objetivo describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece. Sin embargo, si el nivel de detalle elegido es demasiado estructurado, el trabajo de rellenar las plantillas y mantenerlas, puede ser demasiado tedioso. (A. Durán, 2002)

Escenarios: la técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. Esta representación puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo. El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales de sistema. (Lowe, y otros, 1999)

1.6.3 Técnicas para la Validación.

Prototipos: durante la actividad de obtención de requisitos, puede ocurrir que algunos requisitos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requisitos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Entonces, para validar los requisitos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado sobre la base de los requisitos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva (Dávila, 2001)

Revisión Técnica Formal: se examina la especificación del sistema buscando errores en el contenido o en la interpretación, áreas donde se necesitan aclaraciones, información

incompleta, inconsistencias, requisitos contradictorios, o requisitos imposibles o inalcanzables. (Pressman, 2005)

Auditorías: consiste en la revisión completa de la documentación generada y chequeo de los resultados contra una lista de verificación definida al comienzo del proceso para medir la calidad del desarrollo del mismo. Se revisan también los procedimientos empleados en la preparación de las distintas técnicas utilizadas.

Matrices de Trazabilidad: esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito. De esta forma se podrán detectar inconsistencias u objetivos no cubiertos (Escalona, y otros, 2002). Es muy importante que todos los objetivos estén cubiertos pues estos responden a las necesidades del cliente y se verifica cuan acertado está el proceso de obtención de requisitos, es una manera de implementar la auditoria.

1.6.4 Sumario

Las técnicas que se describen en este epígrafe no son las únicas que existen, pero sí las que más se usan.

En la tabla 1.1 se muestra un resumen de las técnicas que se utilizan para realizar cada una de las actividades de la IR.

Técnicas y Herramientas	Obtención	Análisis	Especificación y Descripción	Validación
Entrevistas y Cuestionarios	X			
JAD	X			
Comparación de Terminología	X			
Tormenta de Ideas	X	X		
Mapa Conceptual	X	X	X	
Casos de Uso	X	X	X	X
Lenguaje Natural			X	
Lenguajes Formales			X	
Plantillas o Patrones			X	
Escenarios			X	
Prototipos				X
Revisión Técnica F.				X
Auditorías				X
Matrices de Trazabilidad				X

Tabla 1-1: Resumen de las técnicas usadas en la IR.

Como se puede apreciar no existe una única técnica estandarizada para realizar la IR en el desarrollo de un software, estas técnicas se utilizan en conveniencia de los desarrolladores y del objetivo que quieran lograr con su aplicación. También es muy común encontrar la forma de combinar algunas de estas técnicas para obtener un resultado satisfactorio.

Se debe tener en cuenta que un SWG requiere un manejo de alto nivel de transacciones, además de implementar varios módulos sobre los cuales se sustenta la aplicación en general, por ello se deben usar técnicas que reflejen bien las necesidades del usuario y/o cliente en los primeros encuentros, como la entrevista, los cuestionarios y casos de uso. El éxito de los resultados que se obtengan además de depender de las técnicas que se utilicen, también depende en gran medida de la experiencia del equipo de análisis y desarrollo y de los propios clientes o usuarios que en ellas participen.

1.7 Herramientas de la Ingeniería de Requisitos.

La IR cuenta con una gama interesante de herramientas software que apoyan sus procesos.

1.7.1 Integral Requisite Analyzer (IRqA)

Herramienta desarrollado por TCP Sistemas e Ingeniería (Díez, 2001) que soporta los procesos de IR (básicamente obtención y análisis de requisitos) Consiste de un entorno gráfico y textual que permite generar una documentación sencilla que los clientes pueden revisar en las primeras fases de construcción de una solución. IRqA además, posee módulos para la definición y gestión de pruebas de aceptación, estimación de costos basados en casos de uso, gestión de versiones y navegación por los dominios del negocio. El inconveniente con esta herramienta es su escasa flexibilidad para adaptarse a metodologías operativamente diferentes para la cual fue desarrollada.

1.7.2 Rational Requisite Pro

Es una poderosa Herramienta CASE para la Administración de Requisitos (Rational Corporation), según el procedimiento establecido en el Proceso Unificado para la obtención de un Modelo de Casos de Uso. Rational Requisite Pro presenta compatibilidad con toda la suite de herramientas desarrollado por Rational Corporation y con otras aplicaciones de escritorio como las producidas por Microsoft.

Permite la creación y gestión de una base de datos de requisitos a través de un procesador de palabras, acceso a través de interfaces Web, plantillas para la descripción

de requisitos y rastreo de requisitos en fases posteriores del proceso de desarrollo de software. Es robusta y funcional.

1.7.3 **Controla**

Herramienta diseñada para pequeñas compañías, que permite la identificación de los requisitos cerca del stakeholder (cliente/usuario del sistema), sus detalles, la administración de los cambios y el control de versiones. Permite la definición de requisitos funcionales y no funcionales, especificación del sistema a través de casos de usos y establecer relaciones de dependencia entre los requisitos para los diferentes grupos de artefactos a través de las Matrices de Rastreabilidad. (Vieira, 2006)

1.7.4 **Borland Caliber RM**

Es un sistema de gestión de requisitos diseñado para facilitar la colaboración, el análisis de efectos y la comunicación en la definición y gestión de los requisitos. Automatiza la gestión de requisitos para facilitar la asignación de prioridades a tareas y actualizaciones del proyecto, la simplificación de la comunicación y la gestión del ámbito del proyecto. (Borland CaliberRM, 2005)

Es una herramienta muy fácil de usar, totalmente configurable e integrable con otras herramientas, posee un entorno distribuido y colaborativo, notifica en tiempo real los cambios de los requisitos a los responsables, ofrece múltiples vistas sobre los requisitos y genera documentación automática.

1.7.5 **Doors**

Esta herramienta permite el análisis, comparación y clasificación de requisitos, identificación de inconsistencias, despliega estadísticas y métricas a través de gráficas. Los documentos están escritos en lenguaje claro, lo que proporciona una comprensión inmediata de cada requisito. Las plantillas presentan la información de manera estandarizada.

Permite el acceso de un gran número de usuarios concurrentes en la red, manteniendo en línea un gran número de requisitos así como su información asociada. Ayuda al usuario a

procesar las solicitudes de cambios de requisitos en línea. Esta herramienta proporciona rastreabilidad para aquellas relaciones entre requisitos que poseen gran tamaño. Permite generar enlaces a reportes de muchos niveles, para desplegarlos en la misma vista. (Doors, 2005)

1.7.6 Sumario

Existen numerosas herramientas que se pueden utilizar en cada una de las actividades del proceso de IR. Y, como toda herramienta, es necesario practicar numerosas veces su manejo para lograr el mejor provecho de ellas.

Rational Requisite Pro es la herramienta sugerida para darle soporte a los requisitos en un una aplicación de gestión en la cual se manejan gran cantidad de requisitos funcionales y no funcionales, y que deben ser tratados y seguidos en todo el ciclo de vida del software. Esta herramienta es la adecuada cuando se trabaja con la metodología de desarrollo RUP. Permite su utilización en todas las actividades de la IR y ofrece plantillas para la especificación de los mismos.

1.8 Conclusiones del capítulo.

- La IR consiste en un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software, y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada.
- La IR del SWG es un proceso de obtención, análisis, definición, especificación y validación de las necesidades del cliente; donde se identifica el propósito del sistema y el contexto en el que será usado.
- Los requisitos constituyen el enlace entre las necesidades reales de los clientes, usuarios y otros participantes vinculados al sistema.
- Los requisitos funcionales y no funcionales son identificados por la obtención de información facilitada por el cliente, son analizados y refinados para verificar su claridad, completitud y consistencia. La especificación de estos es validada tanto por el ingeniero de software como por los clientes y/o usuarios.
- Tanto el desarrollador como el cliente interactúan en el transcurso de todas las actividades, esta comunicación es necesaria para obtener una solución software que resuelva correctamente el problema.
- El uso sistemático de técnicas, lenguajes y herramientas dan soporte a la obtención, análisis, documentación y seguimiento de los requisitos del sistema en todo el ciclo de vida del software.

2. CAPÍTULO 2: MODELO DEFINIDO. SISTEMA DE PROCESOS

En este capítulo se describe el sistema de procesos del modelo de Ingeniería de Requisitos definido para los proyectos de desarrollo de SWG en la facultad 3; el cual es una representación simplificada y flexible, que describe y explica los elementos y características más importantes que componen la IR. Representa una teoría a seguir sobre el funcionamiento de este proceso y los resultados a obtener.

2.1 Definición del Modelo de Ingeniería de Requisitos.

Atendiendo a las necesidades y características fundamentales del SWG desarrollado en la facultad 3, se propone la aplicación de un Modelo de IR, que centrado en una espiral explica el funcionamiento de este proceso.

El Modelo consta de un sistema de procesos y elementos que apoyan dicho sistema.

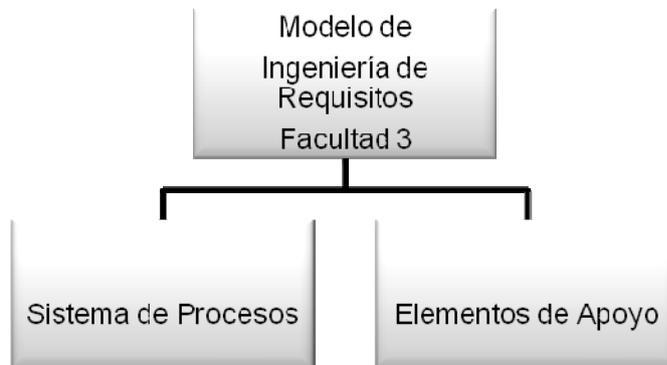


Figura 2-1: Representación visual del Modelo de IR definido.

Se propone que el Sistema de Procesos se divida en el Desarrollo de Requisitos y la Administración de Requisitos.

El Desarrollo de Requisitos involucra las etapas de Obtención, Análisis, Especificación y Descripción, y Validación de los requisitos del proyecto software. Estas etapas abarcan

todas las actividades involucradas para reunir, evaluar, documentar y comprobar los requisitos.

La Administración de Requisitos, tiene como objetivo mantener la integridad, exactitud y actualidad de los requisitos de acuerdo con el avance del proyecto. Para ello se definen los procesos: control de cambios de los requisitos, control de versiones y seguimiento de los requisitos dentro del proyecto software.

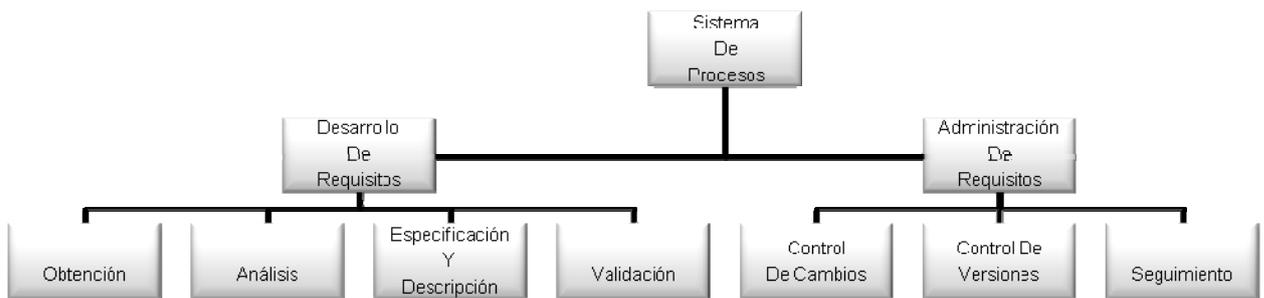


Figura 2-2: Representación visual del Sistema de Procesos.

Los elementos de apoyo dan soporte al desarrollo de las actividades propuestas en el sistema de procesos. Pueden ser: técnicas para la realización de las actividades, plantillas y herramientas para la automatización de dichas actividades dentro de los procesos.

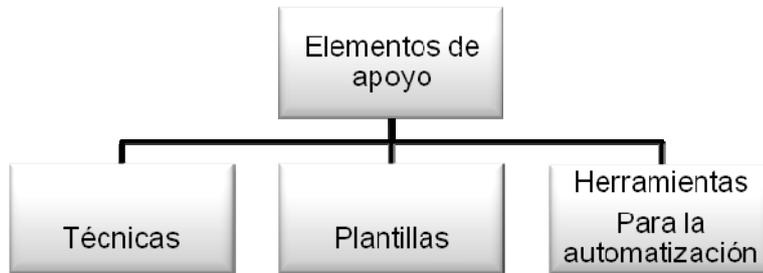


Figura 2-3: Representación visual de los elementos de apoyo.

La espiral que se muestra en la figura 2.4, visualiza dentro del Sistema de Procesos, el Desarrollo de Requisitos. Muestra el ciclo de vida de los procesos Obtención, Análisis, Especificación y Descripción y Validación en un SWG, dividido en cuatro etapas, y los documentos que se deben obtener como resultado de cada proceso.

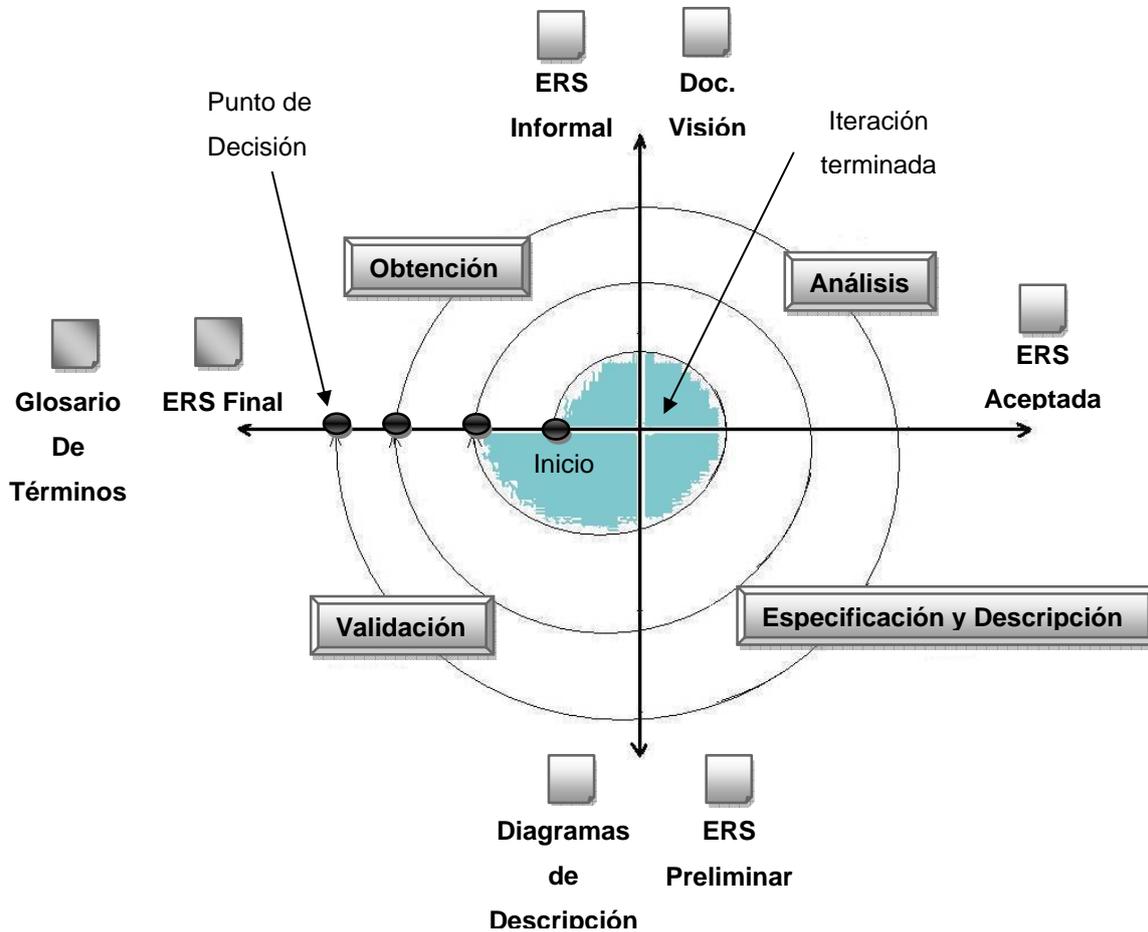


Figura 2-4: Representación en espiral de los procesos de la IR.

El modelo es iterativo e incremental, por lo que es compatible con cualquier metodología de desarrollo de software que tenga esta misma característica, y con cualquier actividad adicional que se le pueda incorporar. Su principal ventaja está dada por la continua evaluación de riesgos, potenciando la reusabilidad y la eliminación a tiempo de errores.

Cada una de las etapas (Obtención, Análisis, Especificación y Descripción y Validación) es representada por un cuadrante del diagrama cartesiano que representa el modelo. La forma espiral indica que el ciclo de actividades se puede repetir hasta que se tome la decisión final de aceptar un documento de especificación de requisitos completo, permitiendo iteraciones, vuelta atrás y finalizaciones rápidas. Al dividir el proceso en

iteraciones, al final de cada una existe un punto de decisión. Es necesario monitorear muy bien el proceso para saber en qué momento debe terminar la espiral.

Los puntos de decisión indican el momento de valorar la continuidad de la espiral con una nueva iteración o culminar la existente. Si se decide continuar, se debe completar la iteración anterior con una revisión, valorar hasta que punto se cumplieron los objetivos propuestos y elaborar un plan para la nueva iteración. Si se culmina, es porque ya se tiene un documento de ERS bien definido y que refleja todas las expectativas del cliente. Este documento debe ser aprobado por el cliente y por el equipo de desarrollo, y constituye el artefacto fundamental a obtener en la realización de las actividades de todo el proceso de IR.

2.2 Definición del equipo de desarrollo en el proceso de IR y las responsabilidades de cada rol.

Durante el proceso de IR de cualquier proyecto de desarrollo de software de gestión, participa un equipo de personas que ocupan diferentes roles. La comunicación entre los miembros del equipo de desarrollo y de estos con los clientes o usuarios, es uno de los aspectos más importantes a tener en cuenta en este proceso. En la figura 2.5 se representa cómo está estructurado el equipo de desarrollo en el proceso de IR.



Figura 2-5: Organización del equipo de IR.

El equipo de IR está integrado por:

- Jefe del Proyecto
- Analista Principal
- Analista de sistema
- Experto en el dominio del problema
- Especificador de Requisitos
- Diseñador de interfaz de usuario
- Responsable del seguimiento y control de los requisitos
- Grupo de Control de Calidad
- Cliente y/o Usuario.

Estos roles son desempeñados en la mayoría de los procesos de la IR. En el epígrafe 2.2.1 se detalla cuáles participan en cada proceso.

Jefe del Proyecto:

Su misión es planificar, motivar, organizar y controlar a los profesionales que realizan el trabajo de software. Dirige y coordina todo el aseguramiento técnico y logístico del proyecto así como todas las tramitaciones legales. La persona que desempeña esta responsabilidad debe ser un especialista en Informática con amplios conocimientos de Gestión de Software. Puede solicitar cambios de los requisitos.

Analista Principal:

Además de cumplir las funciones del analista de sistema, se desempeña como Jefe del Equipo de IR.

Analista de sistema:

Identificar los requisitos reales del sistema planeado, trabajar eficazmente con los clientes y usuarios para manejar un nuevo requisito y gestionar los cambios de los requisitos existentes. Estar alerta a las nuevas tecnologías que pueden ayudar al desarrollo de este proceso. Facilitar el proyecto la reusabilidad de los requisitos. Ayudar al proyecto y a sus clientes a prever un camino de crecimiento de la primera versión del producto. Aconsejar al equipo de proyecto y a los clientes de métodos, técnicas y herramientas automatizadas que están disponibles para el mejor manejo de los requisitos y actividades de esta área y tener dominio del área donde el software va a ser usado.

Experto en el dominio del problema:

Es un profesional con experiencia en las prácticas y terminologías empleadas por la organización para la cual se desarrollará el sistema. Es el encargado de proporcionar la entrada inicial al proceso de IR, el documento de ERS Informal.

Especificador de Requisitos:

Es el encargado de traducir los requisitos obtenidos en el análisis a un documento de especificación preciso, completo y verificable, que refleje las necesidades del cliente. Detalla la especificación de una parte de la funcionalidad del sistema describiendo los

aspectos de Requisitos de uno o algunos casos de uso y otros requisitos de soporte al software.

Diseñador de interfaz de usuario:

Lleva y coordina el prototipo y diseño de la interfaz del usuario, para capturar los requisitos en la interfaz de usuario, incluso los requisitos de usabilidad, construir el prototipo de la interfaz de usuario e implicar a otros involucrados en la interfaz de usuario, tales como usuarios finales, en la revisión de usabilidad. Además, juega un papel importante en la etapa de validación de los requisitos.

Responsable del seguimiento y control de los requisitos:

Identifica y controla quién realizó el cambio y el motivo del mismo, pues no todo el personal involucrado en el proyecto puede realizar un cambio. Debe conocer las últimas versiones de requisitos y darle seguimiento a los mismos durante el proceso restante de desarrollo del software.

Grupo de Control de Calidad:

Se asegura de que todo el sistema funcione apropiadamente y de acuerdo con lo planeado. Generalmente basa su trabajo en estándares de calidad de software (CMMI, ISO). Además debe controlar la calidad en todo el proceso de cambios, versiones y seguimiento de los requisitos.

Cliente:

El principal contacto con la empresa, y quien finalmente pagará por el proyecto de software que se desarrolla. Por lo general, es quien contacta al equipo de desarrollo para que éste genere el sistema. Forma parte del equipo de Obtención de Requisitos en la primera etapa de este proceso, y es el encargado de suministrar la información referente a sus necesidades y expectativas con respecto al sistema a implementar. Participa también como principal miembro del equipo de validación de los requisitos. Debe validar y aprobar el documento final de los requisitos. Puede solicitar cambios de los requisitos.

Usuario:

La persona que interactúa directamente con el sistema desarrollado. A menudo, no es la misma persona que el cliente. Puede ser un operador, una secretaria, etc.

2.2.1 Especificación de los roles en cada proceso de la IR.

Equipo de Obtención

- Jefe del proyecto
- Analista Principal
- Analista de sistema
- Experto en el dominio del problema
- Cliente y/o Usuario

Equipo de Análisis

- Jefe del proyecto
- Analista Principal
- Analista de sistema
- Experto en el dominio del problema

Equipo de Especificación y Descripción

- Jefe del proyecto
- Analista Principal
- Analista de sistema
- Especificador de Requisitos

Equipo de Validación

- Jefe del proyecto
- Analista Principal
- Analista de sistema
- Diseñador de interfaz de usuario

- Cliente y/o Usuario

Equipo de Administración

- Jefe del proyecto
- Analista Principal
- Grupo de Control de Calidad
- Responsable del seguimiento y control de los requisitos

2.3 Estrategia de comunicación en el proceso de IR.

La comunicación es un aspecto fundamental en el proceso de elaboración de una propuesta software que resuelva las necesidades de los clientes o usuarios con su consiguiente aceptación y verificación. A continuación se define una estrategia de comunicación en los proyectos de desarrollo de SWG de la facultad 3 durante el proceso de IR.

El proceso de comunicación puede resumirse como sigue:

- Los clientes y/o usuarios tienen una necesidad o un problema a resolver.
- Los clientes contactan con el equipo de desarrollo para resolver dicho problema.
- Esta necesidad de los clientes, es básicamente, la ERS de un software de gestión.
- El equipo de desarrollo a partir de la información que le brindan los clientes y/o usuarios, estudia el entorno donde se enmarca el problema y teniendo en cuenta las características del SWG y las expectativas de los clientes, propone una solución.
- Dicha solución debe ser entendible y válida por los clientes y/o usuarios.



Figura 2-6: Estrategia de comunicación en la IR.

A lo anterior se debe agregar que la comunidad de usuarios potenciales del SWG que se desarrolla puede ser muy diversa, con diferencias notables, tanto formativas como de expectativas, con relación a las funciones y módulos que el sistema a construir debe soportar; funcionalidades que deben ser especificadas por los desarrolladores y que finalmente se deben diseñar e implementar.

También debe tenerse en cuenta que los clientes y/o usuarios y los desarrolladores tienen culturas y vocabularios diferentes, con la posibilidad de que los mismos términos tengan significados diferentes en los distintos vocabularios, o que su significado se vea enormemente afectado por el contexto.

Las preocupaciones sobre el sistema a desarrollar son distintas. Mientras los clientes suelen preocuparse por aspectos de alto nivel como facilidad de uso o fiabilidad, los desarrolladores suelen preocuparse por aspectos de bajo nivel como utilización de recursos, algoritmos, etc.

Es importante no olvidar que el principal interés de los clientes no es un sistema software en sí mismo, sino los efectos positivos resultantes de la introducción del sistema en su organización.

El medio de comunicación que se utilice debe ser entendible por todos los participantes. Se suele utilizar lenguaje natural porque es el único medio de comunicación común a todos los participantes, a pesar de su inherente ambigüedad.

El éxito de la participación del cliente puede aumentar el éxito de la implementación y posible aceptación del sistema final.

El cliente es un factor crítico en el éxito de la entrega del software. Si se entrega un producto sin realizar antes alguna reunión con el cliente, puede darse la posibilidad de que este descubra que el sistema solicitado no reúna sus necesidades, implicando realizar nuevos cambios o modificaciones.

2.4 Desarrollo de Requisitos.

2.4.1 Obtención

La etapa de obtención de requisitos abarca la primera y más importante fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de la obtención de requisitos es garantizar que los requisitos del sistema sean consistentes con las necesidades de la organización donde se utilizará y con las necesidades de los clientes y/o usuarios.

Objetivos:

- Conocer el dominio del problema.
- Descubrir las necesidades de los usuarios llegando a un consenso entre los propios clientes o usuarios.

Actividades:

- Formar un equipo multidisciplinario que sea capaz de diagnosticar el método más apropiado para obtener y recolectar la mayor cantidad de información posible, que garantice la calidad de la actividad y una buena descripción del entorno y el sistema.

- Conocer el entorno donde se enmarca el problema, describiendo de forma sencilla y clara los procesos que ocurren dentro de este y especificando qué procesos de negocio serán automatizados después.
- Recolectar la mayor cantidad de información suministrada por los clientes y/o usuarios. De dicha información se obtienen los requisitos funcionales y no funcionales que debe tener el software que se implemente. En esta etapa se obtienen: objetivos, necesidades y requisitos de clientes y/o usuarios.

Resultados:

- Al finalizar esta etapa se obtiene una especificación informal de los requisitos.
- En esta etapa del proceso es esencial que la especificación esté orientada al usuario. Se debe escribir de modo que el cliente la comprenda y no debe incluir referencias al diseño del software.
- Se obtiene además el Documento Visión del proyecto.

Documento Visión del Proyecto: Se muestra en el Anexo 3.y define, el punto de vista de los involucrados acerca del producto a ser desarrollado, especificando las necesidades claves y características de los mismos. Contiene un bosquejo de los principales requisitos, suministra la base contractual para el posterior detalle de los requisitos técnicos.

A esta etapa se le debe dedicar el mayor tiempo posible, porque es la base para la realización de los demás procesos de la IR y de todo el desarrollo posterior del software. En la figura 2.7 se representan las actividades que dan lugar a un documento de especificación de requisitos informal.

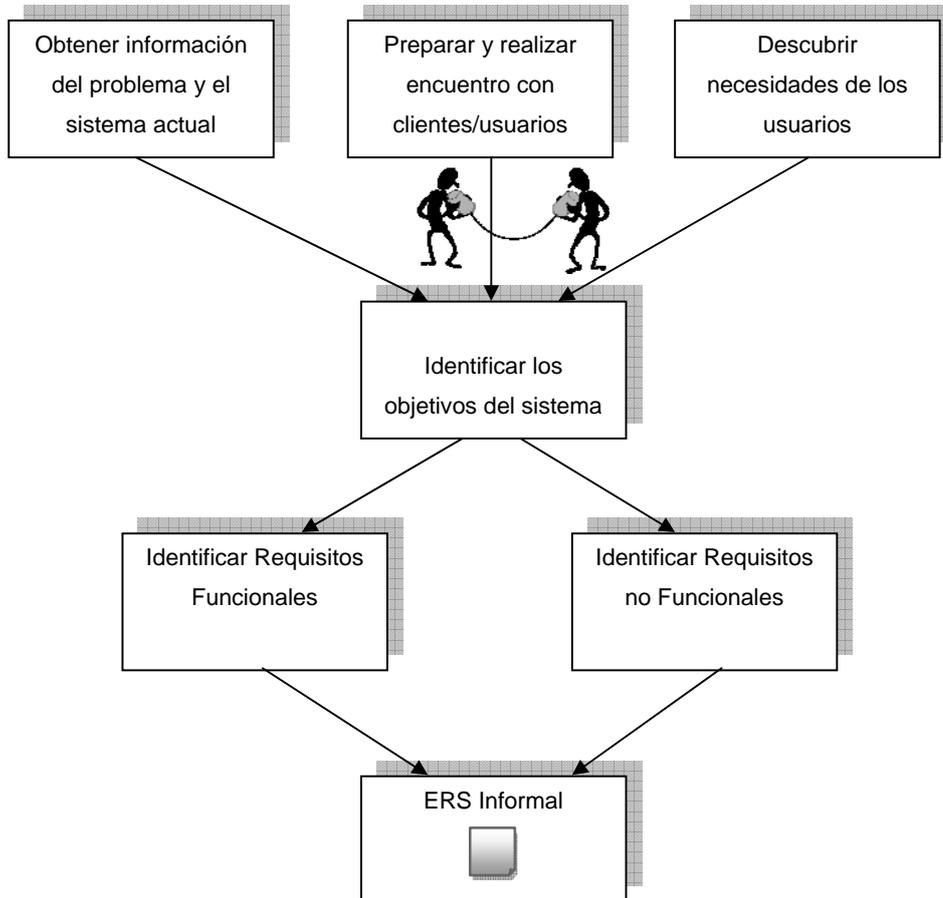


Figura 2-7: Proceso de Obtención de Requisitos.

2.4.2 Análisis

El análisis se define como el proceso de estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software. Una vez obtenidos los requisitos, se refinan, estructuran y clasifican en una lista, determinando el límite que debe alcanzar el software y su relación con el entorno.

Objetivos:

- Descubrir la presencia de requisitos contradictorios.
- Profundizar en el conocimiento del problema.
- Realizar la aproximación de los requisitos a un lenguaje técnico.

Actividades:

- Estudiar los requisitos extraídos en la etapa de obtención.
- Determinar el tipo de requisito al que pertenece el que se está tratando.
- Definir los requisitos funcionales: Definir los requisitos a partir de la información obtenida de las técnicas empleadas.
- Definir los requisitos no funcionales: Definir los requisitos del software con otros elementos como los usuarios, el hardware u otras aplicaciones software.
- Reducir ambigüedades en los requisitos obtenidos como resultado de la etapa de obtención. Estos deben ser llevados a una notación que permita reducir la ambigüedad del lenguaje del usuario.
- Traducir al lenguaje técnico los requisitos con menos ambigüedades. Mediante esta traducción se busca aproximar los términos del usuario a los términos del sistema de software.
- Integrar los requisitos en un documento de especificación.
- Asignar prioridades a los requisitos.

Resultados:

- El análisis proporciona una comprensión precisa de los requisitos en un nivel de detalle que generalmente no interesa al cliente y/o usuario, haciendo uso de un lenguaje formal y entendible por el desarrollador para detallar aspectos relativos a los requisitos del sistema.
- Al finalizar esta etapa se obtiene un documento de especificación de requisitos aceptado por clientes, usuarios y desarrolladores.

En la figura 2.8 se representan las actividades que dan lugar a un documento de especificación de requisitos aceptado.

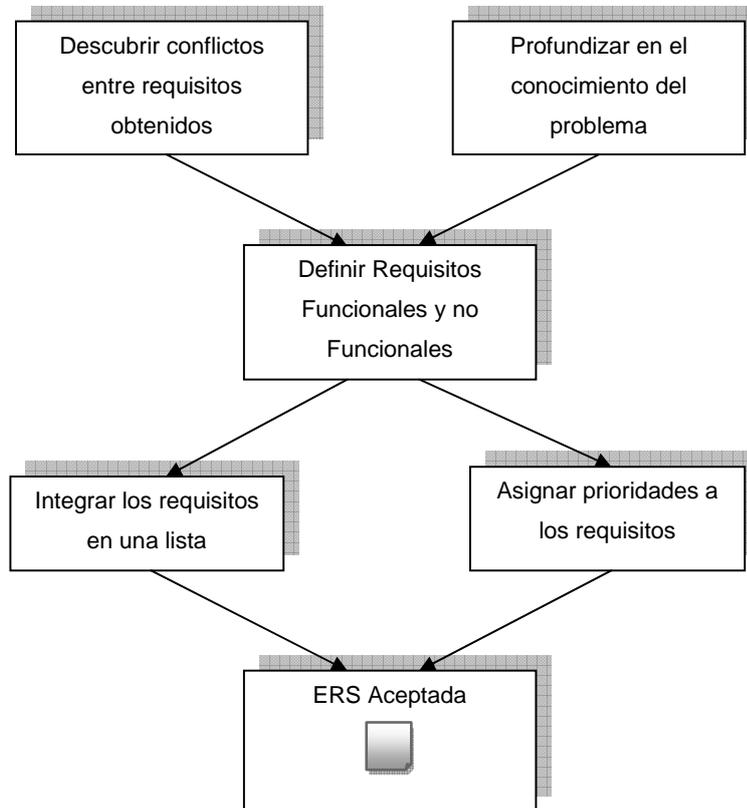


Figura 2-8: Proceso de Análisis.

2.4.3 Especificación y Descripción

Esta etapa se va realizando conjuntamente con el análisis, pero se puede decir que la Especificación consiste en formalizar el análisis realizado previamente, aplicando técnicas y/o estándares de documentación, como la notación UML.

Para cualquier tipo de aplicación, se debe hacer una especificación completa de los requisitos en un documento formal. En el caso de un SWG se propone utilizar tres tipos de técnicas para la especificación y descripción de los requisitos; gráficas, textuales y plantillas, que reflejen el flujo de información que se establece entre el cliente y/o usuario y el sistema.

Objetivo:

- Traducir la información recolectada durante la actividad de análisis a un documento que define, de forma completa, precisa y verificable, los requisitos, el comportamiento u otras características del sistema o componentes de este.

Actividades:

- Realizar una especificación de requisitos de software que incluya información veraz, comunicando dicha información de forma eficaz y que describa correctamente todos los requisitos del software de acuerdo con las características que debe tener la ERS.
- Producir una documentación comprensible por todos los implicados en el proceso de IR.
- Describir los requisitos con diagramas de actividades.

Características que debe tener la ERS.

- No ambigua: Todo requisito posee una sola interpretación.
- Completa: Todo que el software debe hacer debe estar incluido en la ERS.
- Correcta: Todo requisito en la ERS contribuye a satisfacer una necesidad real.
- Comprensible: Todo tipo de lectores deben entender la ERS.
- Verificable: Para cada requisito expresado en la ERS existe un procedimiento de prueba finito para demostrar que el futuro sistema lo satisface.
- Realizable: Si dados los actuales recursos la ERS se puede implementar.
- Concisa: La ERS debe ser lo más breve posible sin afectar a los atributos de calidad.
- Trazable: Cada requisito se puede referenciar de forma unívoca.
- Modificable: Los cambios deben ser fáciles de introducir.
- Precisa: La ERS hace uso de valores numéricos para precisar las características del sistema (aplicable a requisitos no funcionales).

Resultados:

- En esta etapa se finaliza con una especificación preliminar de los requisitos.
- Se obtienen además los diagramas que describen los requisitos.

Crear una especificación preliminar obliga a un análisis detallado del sistema que, por lo general, conduce al descubrimiento de errores e inconsistencias en la especificación previa e informal de los requisitos. Esta detección de errores es la que dota de robustez a la especificación preliminar, y los errores detectados deben conducir a cambios en dicha especificación. En la figura 2.9 se representan las actividades que dan lugar a un documento de especificación de requisitos preliminar.

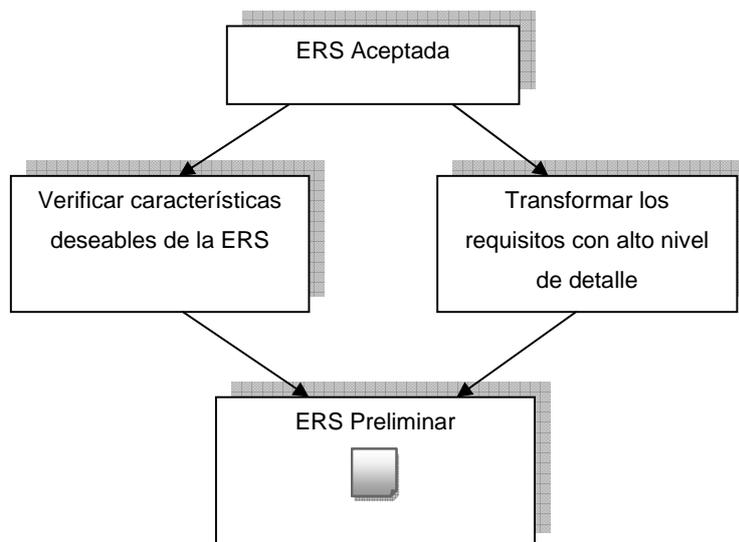


Figura 2-9: Proceso de Especificación y Descripción.

2.4.4 Validación

Objetivos:

- Validar lo obtenido en cada una de las etapas anteriores en un documento de requisitos.
- Descubrir problemas en el documento de requisitos antes de comprometer recursos a su implementación.

- Asegurar que los requisitos describan el producto deseado.

Actividades:

- Seleccionar las fuentes de información a partir de las cuales validar el documento de especificación, teniendo en cuenta los documentos obtenidos en etapas anteriores, seleccionando aquellos que más aporten a la claridad de la descripción y a la validación final.
- Documentar teniendo en cuenta los estándares vigentes para el documento final de los requisitos.
- Verificar la correspondencia entre los documentos obtenidos en etapas anteriores, analizando que sólo se diferencien en el lenguaje utilizado para su definición.
- Aprobar el documento final de los requisitos por los clientes, usuarios y desarrolladores, el cual será utilizado en las etapas siguientes de la Ingeniería de Software.

Resultados:

- Al finalizar esta etapa se obtiene la Especificación final de los requisitos del sistema. Este es el documento oficial para el equipo responsable del desarrollo de los requisitos del sistema, clientes y usuarios finales.
- En esta etapa también se obtiene un Glosario de Términos.

Glosario de Términos: Se muestra en el Anexo 4. Es el documento que contiene las definiciones de los términos principales de los procesos a automatizar, que se enmarcan en el entorno del cliente. Este documento contribuye a una mejor comunicación y entendimiento entre clientes y desarrolladores.

Es recomendable que esta etapa se realice en conjunto con los clientes o usuarios finales del producto; y que estos dominen las características que hacen que los requisitos tengan calidad, o sea, que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que

sean rastreables. En la figura 2.10 se representan las actividades que dan lugar el documento final de especificación de los requisitos.

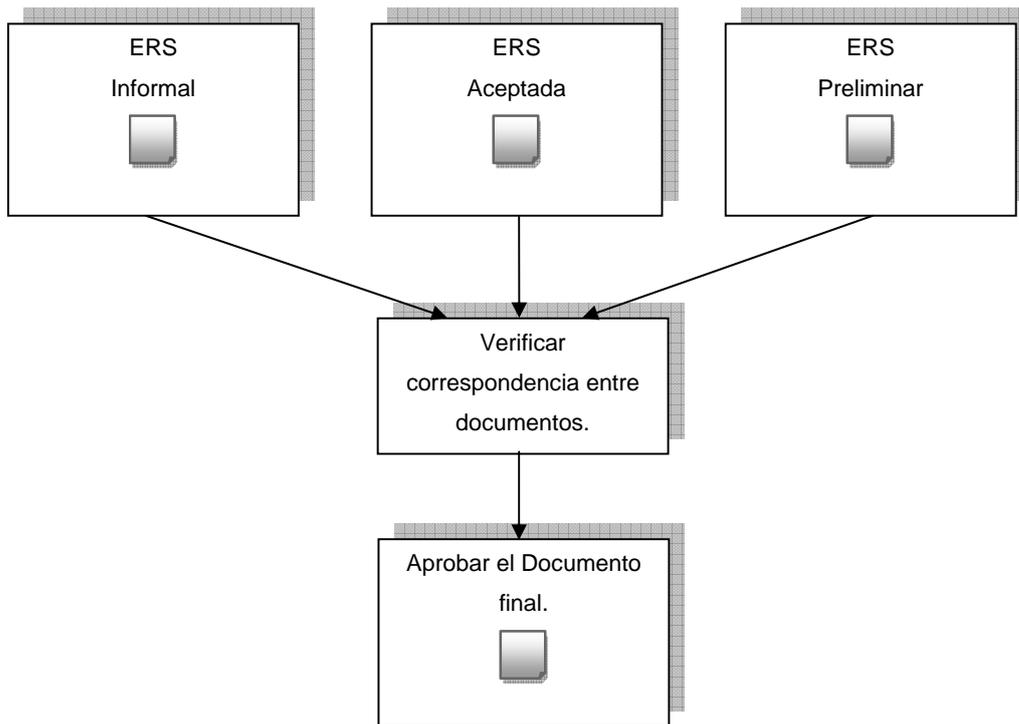


Figura 2-10: Proceso de Validación de Requisitos.

2.5 Administración de Requisitos. Actividades.

La administración de los requisitos consiste en identificar, controlar y seguir los requisitos durante todo el proceso de desarrollo del software, teniendo en cuenta los cambios que surgen en el propio desarrollo. Es el proceso que se encarga de verificar continuamente la validez del documento final de los requisitos obtenido en las etapas anteriores. Las actividades que garantizan la administración de los requisitos consumen gran cantidad de tiempo y personal.

2.5.1 Control de Cambios

Objetivos:

- Gestionar los cambios de los requisitos acordados.
- Asegurar la consistencia entre los requisitos y el sistema a construir o en construcción.

Actividades:

- Establecer la línea base de los requisitos, como un elemento necesario para el control de cambios, que se podrá usar para capturar nuevos cambios.
- Realizar la revisión formal de los requisitos antes de entrar en la Línea Base y solo con su aprobación serán incluidos en esta.
- Someter el requisito al procedimiento de control de cambios, una vez entrado en la línea base.

La línea base de los requisitos es el conjunto de requisitos funcionales y no funcionales que el equipo del proyecto se ha comprometido a implementar. Es una versión aprobada por clientes y desarrolladores de la especificación de requisitos del software.

Resultados:

- Continuamente se obtienen, en esta actividad, los cambios en los atributos de los requisitos afectados.

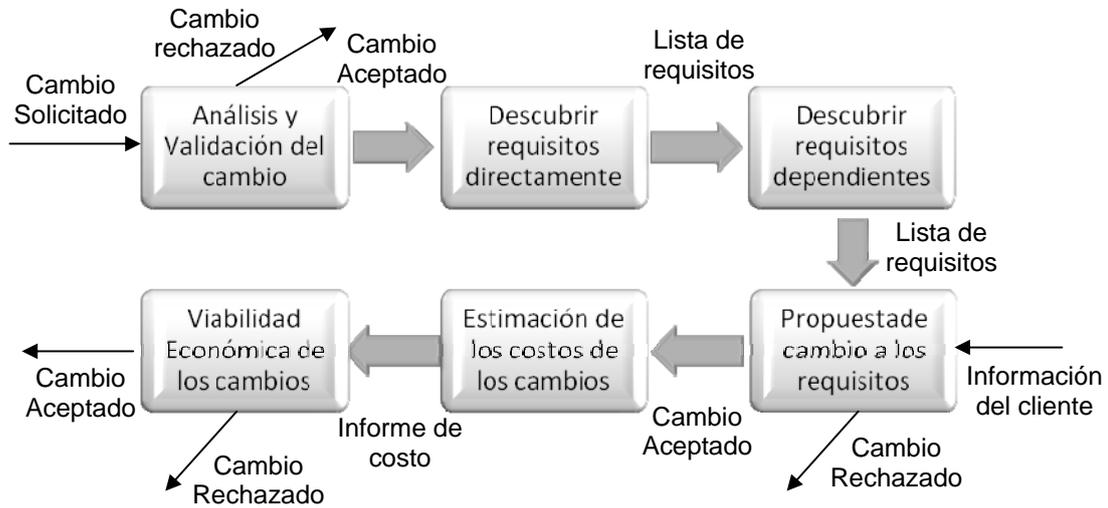


Figura 2-11: Proceso de Control de Cambios en los requisitos.

2.5.2 Control de Versiones

Objetivos:

- Identificar las versiones de los documentos de ERS generados en las cuatro etapas del proceso de IR.

Actividades:

- Prevenir cambios no autorizados.
- Guardar revisiones de los documentos de ERS.
- Recuperar versiones previas de los documentos de ERS.
- Prevenir la modificación simultánea de los requisitos.

Es adecuado que el documento de requisitos que se esté elaborando previo a entrar en la línea base esté sometido a un procedimiento de control de versión (para distinguir versiones borradores de versión aprobada).

Cada requisito debe estar identificado, y como hay que reorganizar y añadir nuevos requisitos, es necesario conocer la última versión de los mismos siendo de vital importancia, para evitar trabajar con requisitos no actualizados. Se debe asegurar que el personal responsable conozca las últimas versiones. Las versiones de los requisitos deben estar debidamente identificadas en las líneas bases, siendo esta actividad esencial en la administración de requisitos.

Resultado:

- En esta actividad se obtiene una versión actualizada del documento de ERS.

2.5.3 Seguimiento

Objetivo:

- Definir la relación de cada requisito con otros requisitos.
- Definir la relación de cada requisito con otros elementos del sistema.
- Definir el estado de todos y cada uno de los requisitos.

Actividades:

- Analizar el impacto que se origina respecto a otros requisitos cuando se modifica uno de los requisitos documentados.
- Registrar en la matriz de seguimiento la relación de cada requisito con el aspecto específico del sistema al que se encuentra asociado.
- Ubicar los requisitos funcionales y no funcionales de acuerdo a su estado en la matriz de seguimiento del estado de los requisitos.

Resultado:

- Se obtiene la ubicación y el estado actual de cada requisito con respecto al sistema.

El seguimiento del estado de los requisitos es un aspecto importante en la Administración de los requisitos, para esto se establecen una serie de categorías que forman parte de la documentación establecida como línea base.

Los estados en que se pueden encontrar los requisitos son los siguientes: propuesto, aprobado, diseñado, verificado, implementado, entregado, anulado (planificado y retirado de la línea base) y rechazado (presentado, pero no aprobado).

2.6 Conclusiones del capítulo.

- Mediante el sistema de procesos definido se logró dividir la IR en dos partes fundamentales, el Desarrollo de Requisitos y la Administración de Requisitos.
- Se describieron los procesos que tienen lugar en cada una de estas partes, brindando sus objetivos, actividades y resultados.
- Se definieron como procesos que componen el Desarrollo de Requisitos, Obtención, Análisis, Especificación y Descripción, y Validación. Dentro de la Administración se incluyen Control de cambio, Control de Versiones y Seguimiento.
- El modelo espiral definido para el Desarrollo de Requisitos logró organizar el proceso de IR de forma iterativa e incremental, lo cual brinda la posibilidad que este sea compatible con cualquier metodología de desarrollo de software que tenga estas características, abarcar todas las actividades involucradas para reunir, evaluar, documentar y comprobar los requisitos y poder mostrar los artefactos y documentación a generar en cada etapa.
- Se hizo una propuesta referente a la organización del equipo de desarrollo de software y las responsabilidades de cada rol.
- Se ofrece una estrategia de comunicación entre clientes y/o usuarios y el equipo de desarrollo, que constituye un elemento fundamental para lograr el éxito del proceso de IR.

3. CAPÍTULO 3: MODELO DEFINIDO. ELEMENTOS DE APOYO

En este capítulo se describen las técnicas y herramientas que dan soporte al sistema de procesos del modelo de IR definido para los proyectos de desarrollo de SWG en la facultad 3.

3.1 Definición de las Técnicas a usar en cada etapa.

Para el desarrollo de los procesos que comprende la IR se propone el uso de técnicas gráficas, textuales y marcos o plantillas. En los siguientes epígrafes se detalla cuándo usar cada una de ellas.

3.1.1 Obtención

Para comenzar la Obtención de Requisitos, se deben establecer un conjunto de reuniones con el cliente. Una de las técnicas más idónea para comenzar esta actividad es la realización de una entrevista por parte del equipo de desarrollo a los clientes y/o usuarios. Es conveniente comenzar con preguntas de contexto libre como se muestra en el Anexo 5, para entender el problema, personas interesadas en la solución, naturaleza de ésta, y efectividad de la reunión. Luego se propone que se aplique una entrevista formal que garantice la obtención detallada de información y logre adquirir las necesidades y expectativas del cliente.

Estas reuniones ayudarán a verificar si el equipo está desarrollando los respectivos requisitos y tomar las medidas correctivas sino es el caso. Una de las formas más efectivas de mostrar que se han recibido y comprendido sus requisitos, es mostrarles en la siguiente reunión los cambios efectuados. De esta manera, el cliente debe conocer las consecuencias de su participación por el incremento o modificación de los requisitos que suscitan, así como el incremento de costos, tiempo de duración y personal.

Es recomendable combinar la entrevista con la creación de modelos conceptuales y la técnica de casos de uso.

Con la creación de un modelo conceptual se obtiene una representación de los conceptos más importantes del contexto del sistema y las relaciones existentes entre ellos.

Cómo técnica gráfica se propone utilizar los diagramas del Lenguaje Unificado de Modelado (UML) de la siguiente manera.

Los requisitos obtenidos en las entrevistas, se agrupan en un diagrama de casos de uso. Este diagrama representa las funcionalidades del SWG que se desarrolla y la secuencia de interacciones que se producen entre el sistema y los actores del mismo. Estos actores son elementos externos como personas u otros sistemas.

Estas técnicas son una buena forma para obtener un idea general de cómo funciona el negocio y como debe funcionar el sistema. Capturar vocabulario y conceptos en etapas iniciales del proceso de IR, también es de ayuda para incluir nuevos conceptos al glosario de términos.

3.1.2 Análisis:

En el análisis se debe profundizar en la técnica de modelos conceptuales, refinando los modelos conceptuales construidos durante la obtención, de forma tal que sean entendibles por los desarrolladores y sirvan de base para etapas posteriores.

En esta etapa, para representar un concepto se recomienda usar la metodología orientada a objetos, utilizando las clases como forma de representar el concepto, dado que una clase representa un concepto del dominio del problema. Para especificar el modelo conceptual se debe utilizar el diagrama de clases, en el cual se muestran las relaciones existentes entre las clases, es decir, los conceptos que se manejan en el negocio y como se relacionan entre sí.

El modelo conceptual puede especificarse informalmente utilizando un lenguaje natural puro, semiformalmente a partir de un lenguaje natural restringido o con la ayuda de diagramas, o formalmente con formalismos lógicos y/o algebraicos.

3.1.3 Especificación y Descripción

Para sistemas de gestión, con procesos bien definidos, se recomienda la especificación de los requisitos en un documento escrito, con descripciones en lenguaje natural y acompañado de representaciones gráficas que sean comprensibles por los clientes y/o usuarios.

Con el objetivo de comprender el entorno en el cual se encuentra el sistema, se propone como técnica gráfica, la realización de un diagrama de actividad, en el cual se describe su funcionamiento interno y su relación con el ambiente.

El diagrama de actividad, como se muestra en el Anexo 6, es un mapa de procedimientos, que muestra el flujo de actividades: se toman decisiones (bifurcaciones) de acuerdo a las condiciones (condición de guarda), para luego pasar a la siguiente actividad o estado (transición). Este modelo también permite representar actividades que ocurren en paralelo, o aquellos casos en los que una única actividad desencadena más de una tarea (división de control), o cuando se unen dos o más actividades para formar una tercera (unión de control).

Como técnica textual se propone la descripción de los diagramas de casos de usos realizados en la obtención, mediante la plantilla que propone RUP, ver Anexo 7.

También se deben especificar los requisitos en un lenguaje natural antes de llevarlos a un documento formal, para comprender mejor el significado de cada requisito. Mediante esta técnica, se especifican con más detalle los componentes definidos en los gráficos.

Para realizar la especificación en lenguaje natural se deben tener en cuenta los siguientes puntos:

De forma general:

- Expresarlos de forma individual (esquemáticamente)
- Organizarlos de forma jerárquica (a distintos niveles de detalle)
- Enumerarlos (para facilitar su gestión)
- Expresarlos de forma clara y concreta, evitando imprecisiones y ambigüedades (Uso de puntos suspensivos, etcétera...)
- Deben ser concisos (sin rodeos ni figuras retóricas)
- Deben ser completos y consistentes
- Deben indicar lo que se espera que haga el sistema (¿qué?), su justificación (¿por qué ha de ser así? ¿Quién lo propuso?) y, en su caso, los criterios de aceptación que sean aplicables (¿cómo se verifica su cumplimiento?).

Para los requisitos funcionales:

- Deben estar redactados de tal forma que sean comprensibles por los clientes y/o usuarios, sin conocimientos técnicos avanzados.
- Deben especificar el comportamiento externo del sistema y evitar, establecer características de su diseño.

Para los requisitos no funcionales:

- Deben especificarse cuantitativamente siempre que sea posible, para verificar su cumplimiento.

Se propone el uso de una plantilla, como se muestra en el Anexo 8, basada y conforme con el estándar IEEE Std. 830-1998 como documento formal para realizar la Especificación de Requisitos de Software.

3.1.4 Validación:

En la Validación se trabaja buscando errores en el contenido o en la interpretación de los requisitos, en áreas donde se necesitan aclaraciones. Verificando con no exista

información incompleta, inconsistencias, requisitos contradictorios o requisitos imposibles o inalcanzables.

Se propone usar como mecanismo de validación de requisitos la Revisión Técnica Formal que propone Roger S. Pressman (Pressman, 2005), chequeando cada requisito con un cuestionario, el mismo se muestra en el Anexo 9.

También es muy útil realizar un prototipo de interfaz de usuario, sobre la base de la ERS Final, para comprobar si responde con eficiencia a las necesidades del cliente y/o usuario. La creación de prototipos permite que los clientes y/o usuarios tengan una idea más clara del producto que van a recibir. Usualmente pueden manifestarse como bocetos de papel o pinturas, imágenes de una herramienta de dibujo o un prototipo ejecutable interactivo.

Los prototipos de interfaz de usuario deben estar orientados a los clientes y/o usuarios y los prototipos que se generen a partir de las especificaciones formales y que no contengan aspectos relacionados con la interfaz de usuario, deben estar orientados a los desarrolladores, cuyo uso principal sería la comprobación de las definiciones realizadas en la etapa de análisis.

3.2 Definición de las Técnicas a usar en la Administración.

La IR comprende también la administración en las siguientes etapas de desarrollo del software, y para ello también se proponen técnicas que ayudan a entender y comprender el futuro camino de los requisitos.

3.2.1 Control de Versiones

Las versiones de los requisitos pueden generar incremento de costes y tiempo de entrega, por lo que se deben registrar dichas variaciones como: fecha del cambio, persona quién lo originó, motivo del cambio y el incremento y costes que genera dicho cambio, para esto se propone el uso de la siguiente tabla:

No. Requisito: R01			Descripción del requisito:				
Fecha	Persona Autorizada:	Área Funcional:	Descripción del cambio:	Razón del cambio:	Versión:	Inc. Tiempo:	Inc. Costes:

Tabla 3-1: Control de versiones de los requisitos.

3.2.2 Seguimiento

Se recomienda utilizar como modo para el seguimiento de los requisitos la matriz de trazabilidad, recomendada por Roger S. Pressman (Pressman, 2005), lo que significa realizar el seguimiento de los requisitos desde su inicio hasta la implementación de los mismos.

A cada requisito se le asigna un identificador que puede tomar la forma:

< Tipo de requisito>< Número de requisito>

El tipo de requisito toma los valores: RF si es un Requisitos Funcional, y RNF si es un Requisito No Funcional. De esta forma, un requisito identificado como RF04 indica que se trata de un requisito funcional y que tiene asignado el número 4 dentro de los requisitos relacionados. Una vez identificados los requisitos se debe conformar la matriz de seguimiento que muestra la relación de los requisitos con uno o más aspectos del sistema o su entorno.

Cada fila representa un requisito específico y cada columna un aspecto específico del sistema o de su entorno.

Requisito	Aspecto específico del sistema o de su entorno.						
	A01	A02	A03	A04			Aii
RF01	✓		✓				
RF02	✓			✓			✓
RNF03		✓	✓				
RF04	✓		✓	✓			
Rnn		✓		✓			✓

Tabla 0-1 : Matriz de Seguimiento de requisitos.

Para dar seguimiento al estado de los requisitos se propone el uso de la siguiente tabla:

Categorías	Requisitos				
	R01	R02	...	Rn	Totales
Propuesto	✓	✓			
Aprobado	✓	✓			
Diseñado	✓	-			
Verificado	✓	-			
Implementado	✓	-			
Entregado	-	-			
Anulado	-	✓			
Rechazado	-				

Tabla 0-2: Tabla de seguimiento del estado de los requisitos.

3.3 Definición de Herramientas.

Teniendo en cuenta las características del SWG, donde predomina un amplio manejo de datos y un nivel de transacciones elevado, se ha definido como herramienta Case a utilizar en el modelo de IR descrito, el Rational Rose, debido a que esta herramienta suplir las necesidades que los proyectos de desarrollo de SWG demandan.

3.3.1 Rational Rose

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto.

Es una solución integrada que facilita el modelado visual mediante UML con diferentes tipos de diagramas que permiten ver el sistema desde diferentes perspectivas.

Permite especificar, analizar y diseñar el sistema antes de codificarlo. Además mantiene la consistencia de los modelos del sistema software, así como el chequeo de la sintaxis UML, genera documentación automáticamente, código a partir de los modelos y se puede usar si se aplica ingeniería inversa (crear modelo a partir código). Esta herramienta proporciona un ambiente adecuado para la representación de los diagramas de actividad que se emplean durante el proceso de descripción de los requisitos.

UML es el lenguaje de modelado, mediante el cual los desarrolladores pueden crear diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Facilita la representación gráfica del gran flujo de información que se establece entre el usuario y el sistema cuando se implementa un software de gestión; esto se lleva a cabo mediante un conjunto de símbolos y diagramas, cada uno de estos diagramas tiene fines distintos dentro del proceso de desarrollo.

Por lo general, un SWG cuenta con diversas personas implicadas, las cuales tienen enfoques particulares en diversos aspectos del sistema, por lo que es necesario crear diferentes perspectivas del mismo, que lo examinen desde distintos puntos de vista y esta posibilidad es brindada también por UML.

3.3.2 Rational Requisite Pro

Se propone como herramienta para la administración de requisitos Rational Requisite Pro, como parte de la Suite de Rational.

Es un producto potente y fácil de utilizar para la gestión de requisitos y casos de uso que propicia una mejor comunicación, mejoras en el trabajo en equipo y reduce el riesgo de los proyectos.

Combina la interfaz conocida y fácil de utilizar de los documentos de Microsoft Word con funciones de base de datos para conseguir la máxima eficacia en análisis y consulta de requisitos.

Proporciona un repositorio central con acceso seguro, plantillas de proyecto y documentos, organización de los requisitos en paquetes y por vistas. Muestra y filtra información por tipo de requisito y por valores de atributos de los requisitos. Requisite Pro también incluye poderosas funciones para la administración como la matriz de trazabilidad gráfica empleada como técnica para el seguimiento de los requisitos.

Brinda a los equipos la posibilidad de comprender el impacto de los cambios en tiempo real. Garantiza que todos los componentes del equipo estarán informados de los requisitos más actuales para asegurar la coherencia, a través de e-mail enviados a los miembros del equipo de desarrollo.

Adicionalmente, por su integración con la mayoría de las herramientas de desarrollo de software, Rational Requisite Pro promueve la reusabilidad de los artefactos y facilita compartir la información, optimizando aún más la colaboración del equipo. Se complementa con Rational Rose (Análisis y Diseño), Rational Soda (Documentación de Modelo) y Rational Suite TestStudio (Herramientas para la automatización de las pruebas de software), Rational ClearCase (Herramienta para el control y administración de

versiones), y Rational Unified Process (Asistente de Metodología y Procedimientos de Desarrollo de Software).

3.4 Determinación del valor práctico del modelo propuesto

Al culminar el Modelo de Ingeniería de Requisitos para los proyectos de desarrollo de software de gestión de la facultad 3 (MIR-SWG) se circuló entre especialistas en el tema un resumen de las propuestas principales dentro del modelo.

En el cuestionario se evaluó la necesidad del modelo, la posibilidad real de implantación y la importancia del modelo para el software de gestión de la UCI. Además, se evaluó el tiempo que demoraría la implantación del modelo según el criterio de los especialistas.

El cuestionario circulado entre los especialistas aparece en el Anexo 10.

Para evaluar el valor práctico del modelo se fijaron un grupo de factores necesidad, posibilidad de implantación, importancia, etc.

El Modelo fue evaluado por tres Ingenieros Informáticos con más de tres años de experiencia, uno de ellos, Máster en Ciencias.

Los criterios de los especialistas reflejados en por ciento aparecen totalizados en la siguiente tabla.

Pregunta \ Evaluación	Si	No	No Se
¿Es necesario un modelo que guíe la Ingeniería de Requisitos en los proyectos de desarrollo de software de gestión de la facultad 3?	100%		
¿Es posible el uso de MIR-SWG en los proyectos de desarrollo de software de gestión de la facultad 3?	100%		
¿Considera importante MIR-SWG para los proyectos de desarrollo de software de gestión de la facultad 3?	100%		

Tabla 0-3: Resultados del cuestionario de validación

3.5 Conclusiones del capítulo.

- La importancia de conocer en profundidad las necesidades de los usuarios ha llevado a la aplicación de una serie de técnicas que facilitan la obtención, análisis, especificación, comprensión y validación los de requisitos, de manera previa a la construcción de un sistema.
- La aplicación de técnicas adecuadas en la obtención de requisitos en los primeros momentos del ciclo de desarrollo, supone una mejora considerable en términos de calidad y reducción de costes en la construcción del producto final.
- La administración de requisitos desde los inicios de un proyecto mediante una herramienta con funcionalidades como el Requisite Pro, es la clave del éxito para lograr que el mismo se entregue en tiempo y presupuesto esperados.

4. CONCLUSIONES

- Se definió un sistema que abarca los procesos fundamentales de la IR.
- Se definieron técnicas, plantillas y herramientas como elementos de apoyo al sistema de procesos.
- Se validó la aplicabilidad del modelo MIR-SWG.

Del modelo MIR-SWG se puede destacar que:

- Contribuye al entendimiento de la Ingeniería de Requisitos detallando procesos bien definidos en el Desarrollo y Administración de Requisitos.
- Por ser un modelo flexible, se le puede incorporar cualquier actividad adicional y su principal ventaja está en la continua evaluación de riesgos, potenciando la reusabilidad y la eliminación a tiempo de errores.
- El uso de técnicas, plantillas y herramientas, proporciona el soporte a los procesos definidos y una mejor comprensión por parte de clientes y/o usuarios y del equipo de desarrollo.

5. RECOMENDACIONES

- Aplicar el modelo MIR-SWG a los proyectos de desarrollo de software de gestión de la facultad 3 y extenderlo en una etapa posterior a los proyectos de nuestra universidad.
- Estudiar la herramienta MindManager para una futura integración al modelo MIR-SWG, como soporte para el modelado conceptual, con el objetivo de organizar ideas y tener una perspectiva general del software que se desarrolla.
- Estudiar Rational RSA, la nueva herramienta que propone IBM, para el modelado con UML 2.0, pues introduce construcciones nuevas de diagramas de actividad para modelar requisitos.

6. BIBLIOGRAFÍA

Acero M., Fernando A. "Gestión Libre, Una oportunidad para todos". 2002.
<http://es.tldp.org/Presentaciones/200211hispalinux/acero/ponencia.html>

Águila C., Isabel M. "Marco metodológico de la ingeniería de los requisitos". Universidad de Almería. Febrero 2005

Anaya, Víctor; Letelier, Patricio. "SmarTTrace: Una Herramienta para Trazabilidad de Requisitos en Proyectos basados en UML". Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, España, 2002.

Anaya, Víctor; Letelier, Patricio. "Trazabilidad de Requisitos Adaptada a las Necesidades del Proyecto: Un Caso de Estudio Usando Alternativamente RUP y XP". Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, España, 2003. Consultado: 8 de noviembre de 2006. Disponible en: <http://issi.dsic.upv.es/publications/archives/f-1055508123529/Ideas2003.pdf>

Arango, J.; Zapata, Carlos " Alineación entre Metas Organizacionales y Elicitación de Requisitos del Software." Universidad Nacional de Colombia. Red de Revistas Científicas de América Latina y el Caribe. Vol. 71, número 143, Medellín, Colombia, Noviembre, 2004

Báez, M. Griselda; Barba B., Silvia I. "Metodología DoRCU para la Ingeniería de Requerimientos". Instituto Superior Politécnico "José Antonio Echeverría", La Habana, 2002. Consultado: 13 de febrero de 2007. Disponible en: <http://www.inf.puc-rio.br/wer01/Mod-Req-1.pdf>

Barreiro Alonso, Enrique. "El Proceso de Desarrollo". Universidad de Vigo, Departamento de Informática. Consultado: 02 de febrero de 2007. Disponible en: <http://trevinca.ei.uvigo.es/~ebalonso/asignaturas/esx/guiones/esxClase4.pdf>.

Borland CaliberRM. "Collaborative requirements management system", 2005 Consultado: 13 de febrero de 2007. Disponible en: <http://www.quinaria.cl/Borland%20CaliberRM.htm>

Cota, A. "Ingeniería de Software."s.l. : Soluciones Avanzadas. 1994.

Dávila, Nicolás Davyt. "Ingeniería de Requerimientos una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto." Artículo Técnico. 2001. Consultado: 23 de marzo de 2007. Disponible en: <http://webs.montevideo.com.uy/nicolasd>

Díez, A. "IRQA y el Desarrollo de Proyectos: Experiencias Prácticas. Jornadas de Ingeniería de Requisitos Aplicada". 2001.

Doors. "ALS: Software Lifecicle Optimazation". 2005. Consultado: 16 de enero. Disponible en:

<http://www.als-es.com/home.php?location=herramientas/entorno-desarrollo/doors>.

Durán Toro, A.; Bernárdez Jiménez, B. " Metodología para la Elicitación de Requisitos de Sistemas Software". Versión 2.3. Departamento de Lenguajes y Sistemas Informáticos Facultad de Informática y Estadística. Universidad de Sevilla. Octubre 2002.

Escalona, María José; Koch, Nora. "Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo". Universidad de Sevilla Lenguajes y Sistemas Informáticos España. Diciembre, 2002. Consultado: 16 de enero. Disponible en:

<http://lsiweb.lsi.us.es/docs/informes/LSI-2002-4.pdf>

Estrada, Hugo; Martínez, Alicia; Pastor, Oscar. "Generación de Especificaciones de Requisitos de Software a partir de Modelos de Negocios: un enfoque basado en metas". Universidad Politécnica de Valencia, España, 2002.

Fernández Sánchez, Leidy. "Gestión del riesgo en la fase de ingeniería de requisitos de un proyecto software." Empresa de Telecomunicaciones de Cuba S. A. Cuba. 2005. Consultado: 11 de noviembre de 2006. Disponible en:

<http://www.monografias.com/trabajos41/riesgo-etapa-requisitos/riesgo-etapa-requisitos.shtml>

Fowler, M. 1996. "Analysis Patterns, Reusable Object Models. Reading." MA. Addison Wesley : s.n., 1996.

García Crespo, Angel; Montenegro Sánchez, Marilú. "Representación visual de la Gestión de Requisitos en la Gestión de Proyectos Informáticos." Universidad Carlos III de Madrid, España. Departamento de Informática 2005. Consultado:15 de enero de 2007. Disponible en: <http://www.monografias.com/trabajos41/requisitos-software/requisitos-software.shtml>

González, Rubén; Pérez, Sergio. "LESE-2 Introducción a Rational Rose, Funcionalidad General". Facultad de informática de Barcelona.

Bruno González, Miguel A. Laguna. "MDA e Ingeniería de Requisitos para Líneas de Producto". Departamento de Informática, Universidad de Valladolid, 2005.

IEEE. Std. 830-1998 "Especificación de los Requisitos del Software." Consultado: 10 de marzo de 2007 Disponible en: http://www.cali.gov.co/publico2/documentos/informatica/IEEE830_ext.pdf

Insfrán, Emilio; Tejadillos, Elena. "Transformación de Especificación de Requisitos en Esquemas Conceptuales usando Diagramas de Interacción." Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, España, 2003

Introducción a las aplicaciones de gestión. Consultado: 19 de noviembre de 2006. Disponible en: http://www.hospedajeydominios.com/mambo/documentacion-manual_html-pagina-gestion0.html

Jacobson, I. "Applying UML in The Unified Process". 1998. Consultado: 2 de febrero de 2007. Disponible en: <http://www.rational.com/uml como UMLconf.zip>.

Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado". España : Pearson, 1999.

Lewis, G. "What is Software Engineering?", 1994.

Lowe, D. y Hall, W. "Hypermedia and the Web. An Engineering approach." s.l. : John Wiley & Son, 1999.

Lutowski, Rick. "Software Requirements. Encapsulation Quality and Reuse" . London, 2005

Mcdonald L., Bárbara A. "Definición de Perfiles en Herramientas de Gestión de Requisitos". Facultad de Informática, Universidad Politécnica de Madrid. Septiembre, 2005. Consultado: 19 de noviembre de 2006. Disponible en:
<http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Mcdonald.pdf>

Merrick, Peter; Barrow, Patrick. "Testing the predictive ability of a requirements pattern language". Universtiy of East Anglia, London, Abril 2004. Consultado: 2 de febrero de 2007. Disponible en. <http://www.springerlink.com/content/b61ny0yqlh5m6pl6/>

Modelos en el ciclo de vida del Software. Consultado 25 de noviembre de 2006. Disponible en: <http://alarcos.infcr.uclm.es/doc/ISOFTWAREI/Tema03.pdf>.

Modelo Espiral. Consultado 25 de noviembre de 2006. Disponible en:
<http://www.idc.usb.ve/~vtheok/cursos/ci3711/apuntes/990114/Info/Modelo%20Espiral.htm>

Moreno G., María N; García P., Francisco J. "Medición de la calidad del software en el ámbito de la especificación de requisitos". Universidad de Salamanca Departamento de Informática y Automática, 2000

Pan, D.; Zhu, D.; Johnson, K. "Requirements Engineering Techniques. Internal Report." Department of Computer Science. University of Calgary. Canada. : s.n., 2001.

Pavón, Juan. "Rational Rose. Ingeniería del Software 2". Facultad de Informática Dep. Sistemas Informáticos y Programación, Universidad Complutense Madrid, 2002.

Payá, Álvaro. "Seminario de Rational Rose". I. T. Sistemas. Tercer curso. Departamento de Informática y Sistemas. Facultad de Informática. Universidad de Murcia. Diciembre de 2003.

Pelechano, Vicente ; Pastor, Oscar ; Insfrán, Emilio. "OO-Method: Una apuesta por la integración de técnicas formales y semi-formales en la ingeniería de requisitos." Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, España, 2003

Pressman, K. "Ingeniería del Software, un enfoque práctico." Ciudad de la Habana : Felix Varela, 2005.

Quintero, J. B.; Anaya de Páez, Raquel. "Un estudio comparativo de herramientas para el modelado con UML". Revista Universidad Eafit, Enero- Marzo, vol 41, número 137, Medellín Colombia, 2005.

Ralph, R. Young. "The Requirements Engineering Handbook." London : s.n., 2004.

Rational Corporation. Consultado : 12 de febrero de 2007. Disponible en: <http://www.rational.com/products/regpro/index.jsp>

Rojas P., Cristián. Comunicación: "El aspecto humano de la Ingeniería de Requisitos". Departamento de Ciencias de la Computación Universidad de Chile. 2006.

Ruiz, Fran. "Documento de Especificación de Requisitos". Octubre 2006. Consultado 12 de febrero de 2007. Disponible en: <http://webdiis.unizar.es/asignaturas/LabProg/material/semER.pdf>

Sawyer, Pete; Kotonya, Gerald. "SWEBOK: Software Requirements Engineering Knowledge Area Description". Computing Department, Lancaster University, 2001

Sánchez, Juan. VRU Un Método para validar requisitos y genera interfaces de usuario multiplataforma. Departamento de Sistemas Informáticos y computación. Universidad Politécnica de Valencia, España.2002. Consultado: 25 de noviembre de 2006. Disponible en: http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C2002_WER.pdf

Solarte S., Mario F. "AMIR-ST: Propuesta de una Aproximación Metodológica para la Ingeniería de Requisitos de Sistemas Telemáticos". Universidad del Cauca, Grupo de Ingeniería Telemática. 2004. Consultado 10 de marzo de 2007. Disponible en: http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r52_art5_r.pdf

Somerville, I. "Software Engineering." Quinta edición. s.l. : Addison Wesley, 1996.

Somerville, I.; Sawyer, P. "Requirements Engineering." Wiley : s.n., 1997.

Sommerville, I. y Kontoya, G. "Requirements Engineering. Processes and techniques." USA. J. Wiley : s.n., 1998.

Sumano L., María A. "Análisis de Requerimientos de Software. Estado del Arte". Instituto Politécnico Nacional Centro de Investigación en Computación. Diciembre 1999.

Consultado: 17 de febrero de 2007. Disponible en:

<http://www.geocities.com/diegolp/ingsof/requerimientos.pdf>

Student Manual IBM Corporation. "Rational UniversityEssentials of Rational RequisitePro" Junio, 2003.

Valera, Fernando. "Soporte a Pruebas con IRqA". Madrid. Noviembre, 2006. Consultado: 10 de marzo de 2007. Disponible en:

<http://www.irqaonline.com/public/docs/events/ExpoQA%20Nov%2006.pdf>

Vieira, Clayton. "Controla: Herramienta de apoyo al proceso de desarrollo de software en las pequeñas compañías." Revista Ingeniería Informática. Abril de 2006. Consultado: 23 de marzo de 2007. Disponible en:

<http://www.inf.udec.cl/revista/ediciones/edicion12/articulo%2012-6.pdf>.

Villanueva, Isabel; Sánchez, J.; Pastor, Ó. "Elicitación de requisitos en sistemas de gestión orientados a procesos". Departamento Sistemas Informáticos y Computación.

Universidad Politécnica de Valencia. España. 2002. Consultado: 24 de enero de 2007.
Disponible en:
http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER05/isabel_villanueva.pdf

7. ANEXOS

Anexo 1

Entrevista de Apoyo a la Investigación.

Fecha: _____

Nombre del Proyecto:	
Nombre del Entrevistado:	
Rol:	Fecha

Resumen:
De los Proyectos de Desarrollo de Software de Gestión en los que ha participado(s):

1. ¿Conoce usted a Ingeniería de Requisitos?

___ Si ___ No

2. ¿Cuál es el rol del ingeniero de requisitos o de quién que corresponde en su definición?

___ El ingeniero de requisitos es encargado de establecer los requisitos que un sistema debe suministrar y las especificaciones de los cuales debe operar.

___ Posibilidad de cumplimiento y comunicación de las necesidades del equipo de desarrollo y la gestión de los requerimientos de los requerimientos.

___ Conlucio de actividades que se siguen con el objetivo de reducir la incertidumbre y mantener un documento de requisitos. El cual es un proceso iterativo y continuo.

3. ¿Qué nivel de prioridad le asigna los requisitos dentro del proceso de desarrollo de software?

___ Alto
___ Medio
___ Bajo

4. ¿De qué manera se maneja el proceso de Ingeniería de Requisitos dentro de su proyecto?
(Marque con una X, o más de una.)

___ Identificación/validación
___ Orientación de requisitos
___ Refinar requisitos
___ Validar requisitos
___ Especificar requisitos
___ Priorizar requisitos
___ Derivar requisitos
___ Clasificar requisitos
___ Aligned requirements management
___ Der seguimiento a los requisitos
___ Gestión de riesgos
___ Roles y responsabilidades
___ Modelar requisitos

5. ¿Qué herramientas o herramientas (o) actividades (s)?

___ menos de 1 año ___ de 1 a 5 años ___ 3 o más años

6. ¿En cuántos proyectos de Gestión ha realizado esta(s) actividad(es)?

___ 1 o más proyectos ___ 2-4 proyectos ___ 5 proyectos o más

7. ¿Existe un documento de especificación de Requisitos del Proyecto?

___ Si ___ No

8. ¿Los Requisitos obtenidos cumplen con los siguientes características?

Complejidad de los requisitos ___ Si ___ No
No Ambigüedad ___ Si ___ No
Consistencia ___ Si ___ No
Verificabilidad ___ Si ___ No
Trasversalidad ___ Si ___ No
Modificabilidad ___ Si ___ No

9. ¿Utiliza alguna herramienta para la gestión de los requisitos dentro del proyecto?

___ Si
___ No

10. En caso afirmativo cuál.

___ IBM Rational Requisite Pro
___ Borland Capella RM
___ Telelogic DOORS
___ Integral Requisite Analyzer (IRA)
___ Requirements Tools System
___ TRUView
___ XTe-Requirements Trace
___ Requirements Mgmt Database
___ TigerPRO
___ Trace4
___ Otros: ¿Cuál? _____

11. ¿Qué técnica(s) se usa para obtener los requisitos de los usuarios y clientes?

___ Entrevistas
___ Talleres
___ JAD
___ Model conceptual
___ Tormenta de ideas
___ Otros de uso
___ Ninguna
___ Otros: ¿Cuál? _____

12. ¿Qué modelo se usa para el tratamiento de los requisitos en su proyecto?

___ Modelo en cascada
___ Modelo en espiral
___ Modelo Prototipo
___ Ninguno

13. ¿Qué Metodología de desarrollo de software utiliza?

___ Métrica V.I.
___ Rational Unified Process (RUP)
___ Dynamic Systems Development Method (DSDM)
___ Extreme Programming (XP)
___ Ninguna

Para Cliente:

1. ¿Este usted satisfecho con los requisitos obtenidos?

___ Alto
___ Medio
___ Bajo

Anexo 2

Estándar IEEE Std. 830:1998.

Tabla de Contenidos

- 1. Introducción
 - 1.1 Propósito
 - 1.2 Alcance
 - 1.3 Definiciones, siglas, y abreviaciones
 - 1.4 Referencias
 - 1.5 Apreciación global
 - 2. Descripción global
 - 2.1 Perspectiva del producto
 - 2.2 Funciones del producto
 - 2.3 Características del usuario
 - 2.4 Restricciones
 - 2.5 Atención y dependencias
 - 3. Los requisitos específicos.
 - 3.1 Requisitos Funcionales.
 - 3.1.1 Requisitos Funcional 1
 - 3.1.1.1 Introducción.
 - 3.1.1.2 Entradas
 - 3.1.1.3 Procesamiento
 - 3.1.1.4 Salidas
 - 3.1.2 Requisitos Funcional 2
 -
 - 3.1.n Requisitos Funcional n.
 - 3.2 Requisitos No Funcionales.
- Apéndices
- Índice.

Anexo 3

Documento Visión

Anexo 4.

Glosario de Términos

Anexo 5

Propuesta de preguntas.

Preguntas centradas en el cliente, objetivos globales y beneficios.

- ¿Quién solicita el trabajo?
- ¿Quién utilizará la solución?
- ¿Cuál será el beneficio económico de una buena solución?
- ¿Existen otras alternativas a esta solución?

Preguntas sobre el problema y la solución.

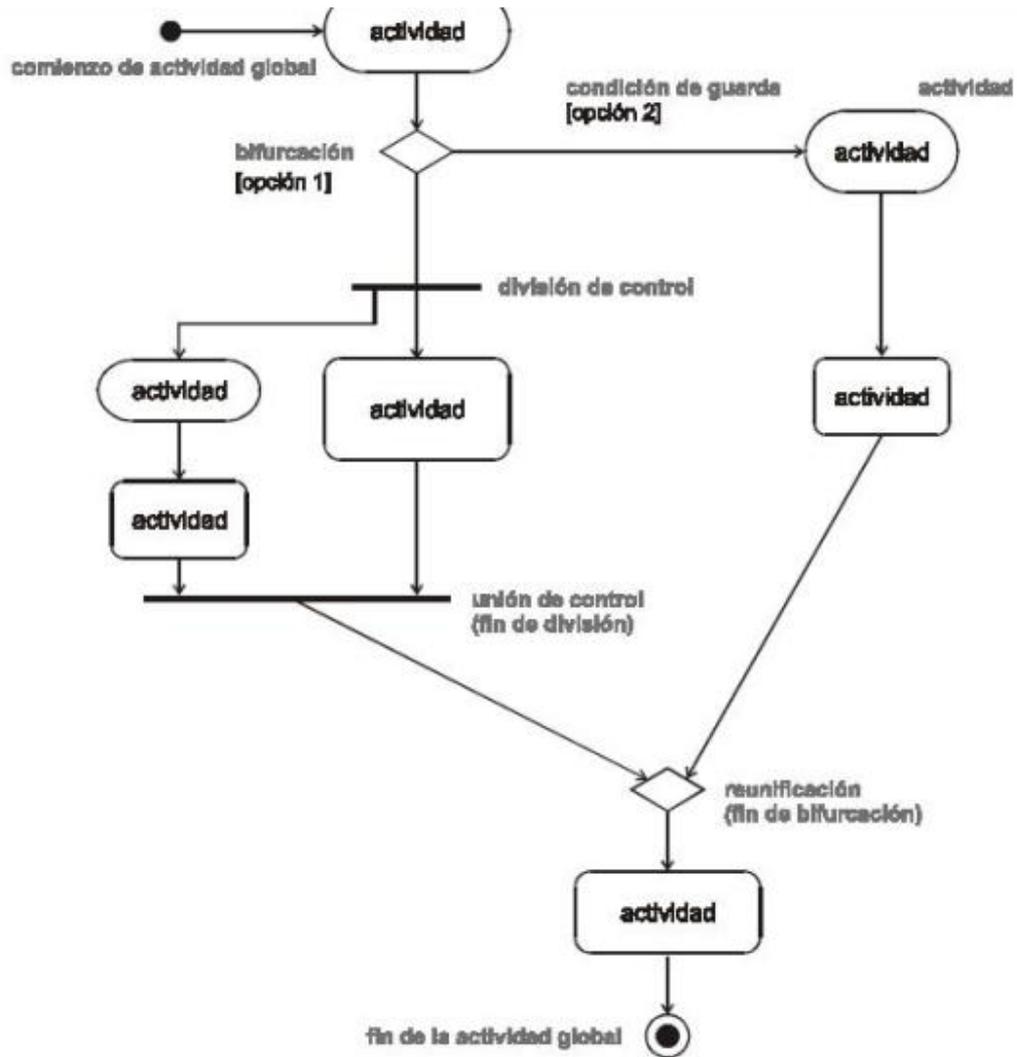
- ¿Qué entiende el cliente por una solución “correcta”?
- ¿Qué problemas afrontará esta solución?
- ¿En qué entorno se va a implantar la solución?
- ¿Existen restricciones o aspectos de rendimiento importantes?

Preguntas sobre la efectividad de la reunión.

- ¿Es usted la persona adecuada para responder a estas preguntas? ¿Sus respuestas son “oficiales”?
- ¿Son relevantes mis preguntas para su problema?
- ¿Hay alguien más que pueda proporcionar información adicional?
- ¿Hay algo más que debería preguntar?

Anexo 6

Diagrama de Actividad



Anexo 7

Plantilla de Descripción de Casos de Uso

Anexo 8

Documento de ERS según Estándar IEEE

Anexo 9

Cuestionario para la Validación de Requisitos

- ¿Está el requisito claramente definido? ¿Puede interpretarse mal?
- ¿Está identificado el origen del requisito (Por ejemplo: persona, norma, documento)? ¿El planteamiento final del requisito ha sido contrastado con la fuente original?
- ¿El requisito está delimitado en términos cuantitativos?
- ¿Qué otros requisitos hacen referencia al requisito estudiado? ¿Están claramente identificados por medio de una matriz de referencias cruzadas o por cualquier otro mecanismo?
- ¿El requisito incumple alguna restricción definida?
- ¿El requisito es verificable? Si es así, ¿Podemos efectuar pruebas (Algunas veces llamadas criterios de validación) para verificar los requisitos?
- ¿Se puede seguir el requisito en el modelo del sistema que hemos desarrollado?
- ¿Se puede localizar el requisito en el conjunto de objetivos del sistema/producto?
- ¿Está el requisito asociado con los rendimientos del sistema o con su comportamiento y han sido establecidas claramente sus características operacionales? ¿El requisito está implícitamente definido?

Anexo 10

Cuestionario para determinar la aplicabilidad del modelo MIR-SWG

1. Es necesario un modelo que guíe la Ingeniería de Requisitos en los proyectos de desarrollo de software de gestión.

Si No No se

2. Considera posible la implantación de MIR-SWG en los proyectos de desarrollo de software de gestión de la UCI.

Si No No se

3. Cuánto tiempo considera UD. demoraría el adecuado funcionamiento de MIR-SWG en los proyectos de desarrollo de software de gestión de la UCI.

Más de 5 años entre 5 y 3 años menos de 3 años

4. Evalúe la importancia de implantar MIR-SWG en los proyectos de desarrollo de software de gestión de la UCI.

No es importante Medianamente importante Muy Importante

8. GLOSARIO DE TÉRMINOS Y SIGLAS

Términos

Diagrama de clases: Es un diagrama utilizado para el análisis y diseño de las aplicaciones. Presenta las clases del sistema con sus relaciones estructurales y de herencia.

Dominio: Área de conocimiento o actividad que se caracteriza por un conjunto de conceptos y una terminología que entienden los profesionales del área.

Metodología orientada a objetos: es una forma especial de programar, más cercana a como se expresarían las cosas en la vida real que otros tipos de programación.

Stakeholder: es un interesado en que el proyecto resulte, es alguien a quien beneficiará el éxito del sistema, y el fracaso del sistema perjudicará. Cualquier persona o grupo que se verá, directa o indirectamente, afectado por el sistema a ser desarrollado.

Siglas

(CASE): Computer-Aided Software Engineering. CASE es una filosofía que se orienta a la mejor comprensión de los modelos de empresa, sus actividades y el desarrollo de los sistemas de información. Esta filosofía involucra además el uso de programas que permiten: construir los modelos que describen la empresa; describir el medio en el que se realizan las actividades; llevar a cabo la planificación; el desarrollo del Sistema Informático, desde la planificación, pasando por el análisis y diseño de sistemas, hasta la generación del código de los programas y la documentación.

(ERS) Especificación de Requisitos de Software.

(IR) Ingeniería de Requisitos

(IRqA) Integral Requisite Analyzer

(JAD) Joint Application Development / Desarrollo conjunto de aplicaciones.

(RUP) Rational Unified Process

(SWG) Software de Gestión

(TIC) Tecnologías de la Información y las Comunicaciones.

(UCI) Universidad de las Ciencias Informáticas.

(UML): Lenguaje Unificado de Modelado.

(WYSIWYG) What You See Is What You Get / lo que ve es lo que obtiene.

