

Universidad de las Ciencias Informáticas
Facultad No. 7



**Título: Ingeniería de requisitos: elicitación,
análisis y negociación, y especificación.**

Trabajo de diploma para optar por el título de
Ingeniero en ciencias informáticas

Autoras: Lilianne Cantillo Romero
Karelys Mesa Pérez

Tutores: Ing. Marla Rosa del Castillo
Ing. Rolando Inerarity Striker

Co-tutora: Rosalía Cué Delgado

Ciudad de La Habana, Junio del 2007

“Pero ninguna idea triunfa así, fácilmente. Para que una idea triunfe hay que empezar a pensarla bien, hay que predicarla, hay que defenderla, hay que persuadir a mucha gente, y entonces al final la idea triunfa.”

Fidel Castro.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 29 días del mes de junio del año 2007.

Lilianne Cantillo Romero

Karelys Mesa Pérez

Firma de la Autora

Firma de la Autora

Marla Rosa del Castillo

Rolando Inerarity Striker

Firma de la Tutora

Firma del Tutor

DATOS DE CONTACTO

Ing. Marla del Castillo Martín: Graduada de Ing. en SAD en el año 1984. Ha presentado ponencias en eventos y congresos nacionales e internacionales y ha sido tutor de tesis universitarias. Ha dirigido proyectos y ha recibido premios relevantes en forum de ciencia y técnica. Ha recibido cursos de postgrado y diplomados.

Correo: marla@softel.cu.

Ing. Rolando Inerarity Striker: Graduado de Ing. en SAD en el año 1985. Ha impartido asignaturas en la UCI como profesor adjunto de Estructura de Datos, Lenguaje C ++. Posee categoría docente de Profesor Auxiliar Adjunto y diplomado en Dirección de Empresas. Ha presentado ponencias en eventos y ha sido tutor de tesis universitarias. Ha dirigido proyectos, ha sido director de empresa y de centros de investigación. Ha recibido premios relevantes en forum de ciencia y técnica, obteniendo los mejores resultados científicos de la Academia de Ciencias.

Correo: inerarity@softel.cu.

Ing. Rosalía Cué Delgado: Graduada de Ingeniería en Sistema Automatizados en el año 1995. Profesora Auxiliar Adjunto Facultad No. 7 en la disciplina de Ingeniería de Software. Posee 11 años de experiencia en el desarrollo de software desempeñando diferentes roles. En la actualidad se desempeña como Especialista Principal en Ingeniería de Requerimientos de la dirección de desarrollo de la empresa Softel.

Correo: rosalia@softel.cu.

AGRADECIMIENTOS

De Lilianne:

A nuestra Revolución por darme la dicha de estudiar en nuestra primera universidad de la Batalla de Ideas, a todo este claustro de profesores consagrado que día a día aportó un grano mas de conocimientos en mi vida profesional y social.

A mis padres porque aún en la distancia siempre estuvieron en la primera línea de combate junto a mí, a mi madre por su ternura infinita y sus regaños a tiempo, a mi tía Leidys que me ha brindado todo su apoyo y cariño hasta en los últimos momentos de mi carrera.

A mi abuelita Linda donde quiera que esté. A mis amigas por apoyarme en todo y estar siempre cuando las necesité, a mi compañera de tesis.

A mi tutora Marla por apoyarnos siempre y a Rosalía que si ella esta investigación no hubiese sido posible.

Al Willy por quererlo tanto y quererlo matar de vez en cuando, porque me ha demostrado que aún con todos sus defectos el amor si existe en tiempos de cólera.

De Karelys:

A nuestro Comandante en Jefe Fidel Castro Ruz por habernos dado la posibilidad de haber estudiado en una “Universidad del Futuro”.

A mi familia que ha sabido guiarme y apoyarme y siempre ha confiado en mí. En especial a mi hermano que con paciencia pudo ayudarme con sus conocimientos a pesar de haber estado realizando su tesis simultánea a la mía y a mi madre por constituir todo por lo que lucho, por su inmenso amor y por brindarme la fuerza que me faltara. A mis amigas Yilén, Darlen, Lucy y Yaimí por estar siempre a mi lado. A mi tutora Marla y a Rosalía por lograr que finalmente se realizara este trabajo con la calidad requerida gracias a sus conocimientos y a su apoyo y a mi compañera de tesis.

DEDICATORIA

De Lilianne:

“A mis padres que tanto esperaron este momento”.

De Karelys:

“Primeramente a mi familia, a mis compañeras y amigas, a las personas que nunca tuvieron la oportunidad de estudiar porque su situación se lo impedía, a los niños y jóvenes que no conocen los pupitres de una escuela y que se hunden en la ignorancia, a los que luchan por la paz, el amor y la unidad de los pueblos, al guía incansable e indestructible de nuestro pueblo Fidel Castro Ruz”.

A todos los que luchan por aprender...

RESUMEN

Actualmente los diferentes proyectos productivos de la Universidad de las Ciencias Informáticas, no cuentan con un procedimiento para la ingeniería de requisitos, en los pasos de elicitación, análisis y negociación, y especificación de los mismos. La misma ha sido tratada en la mayoría de las bibliografías muy superficialmente; y recibido, en general; muy poco tratamiento por parte de las metodologías de desarrollo de software. Por lo que el principal objetivo de este trabajo, es proponer un procedimiento para la elicitación, análisis y negociación, y especificación de requisitos.

Para la realización de la presente investigación, se hizo un estudio de diferentes modelos de procesos y metodologías. Se investigó en diversos textos, documentación y sitios de Internet acerca del proceso de ingeniería de requisitos. Además, se realizaron entrevistas a una muestra de líderes de los diferentes proyectos productivos de la universidad donde se comprobó que hay un déficit del uso de procedimientos y metodologías aplicables a proyectos, aunque se usan algunas metodologías y técnicas o la unión de varias de ellas, provocando que muchas veces sean más trabajosos los pasos a realizar.

Esta investigación, permitió corroborar la necesidad de poder contar en los diversos proyectos productivos, que se desarrollen en la universidad, con un procedimiento para la elicitación, análisis y negociación, y especificación de los requisitos de una manera correcta. Posibilitando que la producción software, no solo se preste atención al diseño, sino que brinde una solución adecuada al problema planteado, con clientes satisfechos, y sin pérdidas económicas y de tiempo.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Elicitación de requisitos	6
1.1.1 Problemas de la elicitación de requisitos	7
1.2 Análisis de Requisitos	11
1.2.1 Los problemas del análisis de requisitos	11
1.3 Negociación	12
1.3.1 Factores que intervienen en la negociación	14
1.3.2 Resultados de la negociación	14
1.3.3 Como conseguir la predisposición	14
1.3.4 Normas para negociar con éxito	15
1.3.5 Objeciones del cliente	15
1.4 Especificación de requisitos	16
1.4.1 Características de la especificación de requisitos	17
1.4.2 Problemas de la especificación de requisitos	18
1.5 Modelado de Sistema	19
1.6 Validación de requisitos	19
1.7 Gestión de requisitos	19
1.8 Modelos de proceso de la ingeniería de requisitos	20
1.8.1 Modelo de Proceso de Duran	20
1.8.2 Modelo de Pohl	21
1.8.3 Modelo Espiral	22
1.9 Metodologías de desarrollo del software	24
1.9.1 Extreme Programming	24
1.9.2 Microsoft Solutions Framework	25
1.9.3 Proceso de la ingeniería de requisitos según RUP	25
CAPÍTULO 2: UTILIZACIÓN DE LA INGENIERÍA DE REQUISITOS, ANÁLISIS Y NEGOCIACIÓN, Y ESPECIFICACIÓN	27
2.1 Cómo se comporta la elicitación, análisis y negociación, y especificación de requisitos en los proyectos productivos	27
2.2 Resultados de la entrevista	28
2.2.2 Metodologías usadas	28
2.2.3 Procedimiento para la elicitación de requisitos	29
2.2.4 Técnicas para la elicitación de requisitos	30

2.2.5 Plantillas para las reuniones de elicitación o captación de requisitos.	31
2.2.6 Documentos que sean guardados como resultado de la elicitación de requisitos.	32
2.2.7 Técnicas usadas para la especificación de requisitos.	32
2.2.8 Procedimiento para el análisis de requisitos.	33
2.2.9 Proceso de elicitación, análisis y negociación, y especificación.	34

CAPÍTULO 3: PROCEDIMIENTO PARA LA ELICITACIÓN, ANÁLISIS Y NEGOCIACIÓN, Y ESPECIFICACIÓN DE REQUISITOS.35

3.1 Técnicas.....	35
3.1. 1 Entrevistas	35
3.1.2 El Desarrollo Conjunto de Aplicaciones (JAD).....	40
3.1.3 Tormenta de Ideas	43
3.1.4 Casos de Uso.	45
3.1.5 Prototipado de interfaz de usuario.....	46
3.1.6 Método de negociación de conflictos Win Win.	47
3.1.7 Táctica del embudo.....	49
3.1.8 Punto de Función por Caso de Uso.	49
3.2 Actividades recomendadas en el procedimiento propuesto para obtener los productos descritos. ...	50
3.2.1 Actividad 1: Obtener información sobre el dominio del problema y el sistema actual.	51
3.2.2 Actividad 2: Preparar y realizar reuniones de elicitación/negociación.	53
3.2.3 Actividad 3: Identificar/revisar los objetivos del sistema	55
3.2.4 Actividad 4: Identificar los requisitos	57
3.2.5 Actividad 5: Revisar los requisitos.....	58
3.2.6 Actividad 6: Eliminar los conflictos entre requisitos.	60
3.2.7 Actividad 7: Estimar el costo y el esfuerzo de cada uno de requisitos funcionales.	61
3.2.8 Actividad 8: Asignar Prioridades a los requisitos funcionales.	62
3.3 Documento de Especificación de requisitos.	63
3.3.1 Portada	64
3.3.2 Índice	65
3.3.3 Introducción	65
3.3.4 Descripción del sistema actual.....	65
3.3.5 Descripción técnica del entorno del sistema	65
3.3.6 Participantes en el proyecto	65
3.3.7 Listado de características y necesidades	66
3.3.8 Listado de clientes y su descripción	66
3.3.9 Definición de los actores	66
3.3.10 Objetivos del sistema.....	66
3.3.11 Catálogo de requisitos del sistema.....	66
3.3.12 Restricciones del dominio y Reglas del negocio.....	67
3.3.13 Origen y razón de cada requisito.....	67

3.3.14 Costos del proyecto	67
3.3.15 Características del software	67
3.3.16 Glosario de términos	67
3.4 Descripción técnica del entorno del sistema	67
3.5 Definición de actores	68
3.6 Especificación de casos de uso.	68
3.7 Informe de requisitos en conflicto	68
3.8 Documentación de ayuda para el cliente	68
3.9 Prototipo	68
3.10 Modelos que ayuden a la comunicación	68
3.11 Agenda de la reunión	69
3.12 Esfuerzo del caso de uso	69
RECOMENDACIONES	71
REFERENCIAS BIBLIOGRÁFICAS	72
BIBLIOGRAFÍA	74
ANEXOS	77
GLOSARIO	87

INTRODUCCIÓN

“La parte más dura en la construcción de un sistema de 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”. [1]

En la actualidad, no existe un método seguro que asegure que se ha especificado un sistema que recoge las necesidades del cliente y satisface sus expectativas, solamente esto se puede asegurar si se implementa un sólido proceso de ingeniería de requisitos.

La ingeniería de requisitos es el mecanismo apropiado para comprender lo que quiere el cliente, analizando sus necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. [2]

El proceso de ingeniería de requisitos puede ser descrito en 6 pasos: [3]

1. Elicitación de requisitos: Preguntar a los clientes, usuarios y todos los involucrados en los objetivos del sistema y que sean expertos, cómo investigar los sistemas o productos que se ajustan a las necesidades del negocio y finalmente cómo el sistema o producto va a ser utilizado.
2. Análisis y negociación de los requisitos: Una vez recopilados los requisitos, el producto obtenido configura la base del análisis de requisitos. Se agrupan por categorías y se organizan en subconjuntos. Se estudia cada requisito en relación con el resto, se examinan en cuanto a su consistencia, completitud, ambigüedad y se clasifican en base a las necesidades del usuario/cliente.
3. Especificación de requisitos: Se sugiere que debe desarrollarse una plantilla estándar y usarse en la especificación del sistema (Es el producto final, sobre los requisitos del sistema, obtenido por los ingenieros). Argumentando que así, se conseguirían requisitos que fueran presentados de forma consistente y comprensible.

Ingeniería de requisitos: elicitación, análisis y negociación, y especificación

4. Modelado del sistema: Su construcción es importante para evaluar los componentes del sistema y la relación entre ellos, para determinar como están reflejados los requisitos y valorar como se ha concebido la estética en el sistema.
5. Validación de requisitos: Examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin omisiones, que los errores detectados hayan sido corregidos, y el resultado del trabajo se ajuste a los estándares establecidos para el proceso, proyecto y producto.
6. Gestión de requisitos: Es un conjunto de actividades que ayuda al equipo de trabajo a identificar, controlar y seguir los requisitos y cambios en cualquier momento.

El alcance de esta investigación se ubica en los tres primeros pasos: elicitación, análisis y negociación, y especificación de requisitos.

Debido al alto riesgo de fracasar en los proyectos del software, a partir de 1968 se comenzó a pensar en una metodología de la ingeniería para la construcción de un software de mayor calidad. Varias instituciones realizaron estudios con diversos proyectos, llegando a la conclusión que los principales factores de motivación en un proyecto del software eran: la implicación de los usuarios, el apoyo de los directivos, y el enunciado claro de los requisitos. Mientras que los tres principales factores de fracaso eran: la falta de información por parte de los usuarios, las especificaciones y requisitos incompletos y las especificaciones y requisitos cambiantes.

En 1996, el proyecto ESPITI (European Software Process Improvement Training Initiative) realizó una investigación sobre los principales problemas en el desarrollo de software a nivel europeo. Los resultados, indicaron que los mayores problemas estaban también relacionados con la especificación, la gestión y la documentación de los requisitos.

Estos informes, ponen de manifiesto el hecho de que, a pesar de que las herramientas para construir software han evolucionado enormemente, se sigue produciendo software, que no es satisfactorio para los clientes y usuarios. Esto indica que los principales problemas que han dado origen a la crisis del software residen en las primeras etapas del desarrollo, cuando hay que decidir las características del producto a desarrollar.

Otro hecho comprobado es que el costo de un cambio en los requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al costo que hubiera representado el mismo cambio durante las fases iniciales de desarrollo. [4] Por lo que no es de extrañar, que aquellos proyectos en los que no se determinan correctamente los requisitos y cambian frecuentemente durante el desarrollo, superen con creces su presupuesto inicial.

Todas estas circunstancias han convencido a la gran parte de la comunidad de la ingeniería del software de la necesidad, cada vez mayor, de una ingeniería de requisitos.

Es por esto que es necesario para el éxito de un proyecto de software comprender cuales son las necesidades del cliente para lo cual es importante tener bien definido un conjunto de buenas prácticas que permitan una mejor comprensión de sus necesidades para con ello poder realizar una mejor especificación de los requisitos que debe cumplir el software a construir.

En nuestra universidad no se cuenta con un procedimiento que ayude en el proceso de la ingeniería de requisitos, a realizar la elicitación, análisis y negociación, y especificación de los mismos lo cual es el principal objetivo de este trabajo.

Dada la **situación problemática** anteriormente expuesta y la necesidad de definir un procedimiento para llevar a cabo la elicitación, análisis y negociación, y especificación de requisitos de un software de forma óptima para los proyectos productivos de nuestra universidad con vista a desarrollar software de mayor calidad, los esfuerzos puestos en este trabajo estarán encaminados a resolver el siguiente **problema**: ¿Cómo desarrollar un procedimiento para una mejor elicitación, análisis y negociación, y especificación de requisitos en el proceso de desarrollo de software?

El **objeto de estudio** es la ingeniería de requisitos dentro el proceso de la Ingeniería de Software.

El **campo de acción** la elicitación, análisis y negociación, y especificación de requisitos.

La investigación de este trabajo de diploma tiene como **objetivo general**: Desarrollar un procedimiento para una mejor elicitación, análisis y negociación, y especificación de requisitos en el proceso de desarrollo de software.

Definiendo como **objetivos específicos**:

Analizar el estado actual de la ingeniería de requisitos, analizar como se comporta la misma en los diferentes proyectos productivos de la misma y definir un conjunto de técnicas de recopilación de las informaciones útiles para la elicitación, análisis y negociación, y especificación de los requisitos.

Para dar cumplimiento a estos objetivos se plantean las siguientes **tareas investigativas**:

- 1- Analizar el proceso de ingeniería de requisitos y las principales metodologías usadas.
- 2- Definir los conceptos principales de elicitación, análisis negociación y especificación de requisitos.
- 3- Analizar la tecnología orientada a objeto en la ingeniería de requisitos.
- 4- Analizar el tratamiento de la ingeniería de requisitos en los diferentes modelos de proceso.
- 5- Investigar el proceso de la ingeniería de requisitos en los diferentes proyectos productivos de la universidad.
- 6- Realizar entrevistas a los diferentes proyectos productivos de la universidad.
- 7- Investigar y definir un conjunto de técnicas para la elicitación, análisis y negociación, y especificación que son más útiles en proyectos de software.

La idea a defender es que: Si se logra encontrar un procedimiento para una mejor elicitación, análisis y negociación, y especificación de los requisitos, se logrará realizar productos de software que satisfagan las necesidades de los clientes y por consiguiente se logrará el éxito del proyecto.

El presente documento fue estructurado en tres capítulos. A continuación se dará una breve descripción de los contenidos que se abordarán en los mismos:

Capítulo 1: Este capítulo contiene toda la fundamentación teórica en la que se basa para llevar a cabo el desarrollo del presente trabajo de diploma, se hace un estudio de diferentes modelos de proceso ya existentes viendo como se trata en los mismos el tema de la ingeniería de requisitos, y las metodologías más utilizadas, principalmente RUP.

Capítulo 2: Ya en este capítulo se hace un estudio basado en diferentes entrevistas realizadas a una muestra de proyectos de la UCI de cómo se lleva a cabo este proceso de ingeniería de requisitos, en los mismos atendiendo a sus tres primeros pasos, o sea, la elicitación, análisis y negociación, y especificación de los requisitos.

Capítulo 3: En este capítulo, se hace la propuesta de un procedimiento para la elicitación, análisis y negociación, y especificación de los requisitos especificando las tareas a realizar, las actividades por tarea, los trabajadores involucrados, productos de entrada y salida, etc.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Cuando se inicia un proyecto de software y antes de que este pueda ser construido, el “sistema” en el que residirá se debe comprender, para ello se deben definir los objetivos generales del sistema: se debe identificar el papel del hardware, del software, de las personas y otros elementos del sistema; y los requerimientos funcionales deben ser identificados, analizados, especificados, modelados, validados, y gestionados, estas actividades son las bases de la ingeniería del sistema [5].

Por lo anterior, se puede afirmar que la ingeniería de sistemas trabaja para asignar un papel al software y establecer los vínculos que enlazan al software con otros elementos de un sistema basado en computadoras.

El resultado del proceso de ingeniería de sistemas, es la especificación de un producto de software. Esta constituye un gran desafío, pues siempre se tiene la duda o interrogante de si la misma cumple con las necesidades del cliente y satisface sus expectativas, esto solo lo se puede garantizar si se ejecuta un correcto proceso de ingeniería de requisitos.

La ingeniería de requisitos facilita el mecanismo para comprender lo que quiere el cliente, analizando sus necesidades y negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional, el proceso de ingeniería de requisitos puede ser descrito en 5 pasos: Elicitación de requisitos, Análisis de requisitos y Negociación, Especificación de requisitos, Modelado de sistemas, Validación de requisitos y Gestión de requisitos. [6].

1.1 Elicitación de requisitos.

Es la primera actividad del modelo de procesos de la ingeniería de requisitos, en ella se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registrar información sobre sus necesidades. Estas interacciones se realizan para recopilar información y detectar conflictos en la información recopilada.

Antes de identificar los requisitos que el proyecto de software puede cumplir es necesario conocer el ambiente y los procesos que se desarrollan dentro de la organización donde se va a aplicar.

Dentro de los modelos de procesos iterativos presentados anteriormente en este capítulo, las actividades de elicitación pueden realizarse varias veces. En la primera iteración de la elicitación consistirá en recopilar y obtener toda la información posible y necesaria para modelar la organización en estudio. Se propone en esta etapa la obtención del Modelo de Negocio o Dominio, donde se describen los procesos de negocios de la organización, especificando sus datos, tareas, roles, agentes y reglas, información a partir de la cual se identifican los primeros requisitos candidatos a ser cubiertos por el proyecto a desarrollar.

En las siguientes iteraciones, la elicitación consistirá en la resolución de conflictos encontrados en la información elicitada durante las actividades de análisis o de validación. La resolución de estos conflictos se llevará a cabo, normalmente, mediante algún tipo de *negociación* entre los participantes.

Pudiera parecer fácil realizar esta investigación, pero no sucede así pues existe un conjunto de problemas que hacen que la elicitación de requisitos un proceso complicado, en los próximos epígrafes se abundará acerca del tema.

1.1.1 Problemas de la elicitación de requisitos.

La mayoría de los problemas del desarrollo de software están relacionados con la ingeniería de requisitos, y dentro de ésta, con la elicitación de dichos requisitos.

Aunque se cuente con excelentes lenguajes de especificación de requisitos y se consiga que los clientes y usuarios validen una determinada especificación, si no se han elicitado los requisitos *correctos*, todo el trabajo de desarrollo terminará con un producto técnicamente correcto pero inútil, que no va a satisfacer las necesidades que dieron origen a su desarrollo.

La elicitación de requisitos se enfrenta a múltiples problemas, para una mejor comprensión se agrupan en 5 grandes categorías que son: problemas de articulación, de comunicación, de conocimiento, de conducta humana y técnicos. Según en [7]. Los cuales se trataran a continuación.

Problemas de articulación: Los problemas de articulación están relacionados con la expresión de sus necesidades por parte de clientes y usuarios y la comprensión de dichas necesidades por parte de los desarrolladores. Algunos de estos problemas son:

- Los clientes y usuarios pueden ser conscientes de sus necesidades pero no ser capaces de expresarlas apropiadamente. Es lo que en sociología se denomina el *problema de decir-hacer* y en filosofía se denomina *conocimiento tácito*: las personas saben cómo hacer muchas cosas que no saben describir [8].
- Los clientes y usuarios pueden no ser conscientes de sus necesidades y puede que no entiendan cómo la tecnología puede ayudarles. Algunos usuarios pueden no expresar sus necesidades por miedo a parecer incompetentes antes los demás o porque los desarrolladores juegan un papel excesivamente dominante en el proceso, provocando que la falta de conocimiento tecnológico de los usuarios les haga sentir en inferioridad de condiciones [9].
- Los clientes pueden no llegar a tomar decisiones porque no pueden prever las consecuencias de su decisión o porque no entienden las alternativas que se les plantea. En otras ocasiones no se toman decisiones porque no hay una sola persona que tenga una visión global, por lo que puede haber varios puntos de vista que tengan que integrarse [10].
- Algunos desarrolladores no escuchan apropiadamente a los clientes y usuarios, bien porque creen haber entendido sus necesidades rápidamente, bien porque se dedican a pensar inmediatamente sobre aspectos de implementación y no se ponen en el lugar de clientes y usuarios.

Problemas de comunicación: Algunas de las dificultades en la comunicación entre clientes, usuarios y desarrolladores son las siguientes:

- Los clientes y usuarios y los desarrolladores tienen culturas y vocabularios diferentes, con la posibilidad de que los mismos términos tengan significados distintos en los distintos

vocabularios, o que su significado se vea enormemente afectado por el contexto, ya que en estos momentos del desarrollo la información está fuertemente *contextualizada* [11].

- No sólo la cultura y el vocabulario son distintos, las preocupaciones sobre el sistema a desarrollar también suelen serlo. Mientras los clientes y usuarios 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. En palabras de A. M. Davis [12]: *"Nunca debe perderse de vista porqué se desarrolla el software: para satisfacer necesidades reales, para resolver problemas reales. La única forma de resolver las necesidades reales es comunicarse con aquellos que tienen dichas necesidades. El cliente o usuario es la persona más importante involucrada en el proyecto."*
- 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. La utilización de otro tipo de técnicas como diagramas o lenguajes artificiales puede presentar problemas de comprensión.
- La comunicación puede verse afectada también por sus aspectos puramente sociales. El ingeniero de requisitos debe ser capaz de comunicarse y tratar con todo tipo de personas y ser capaz de manejar conflictos personales y políticos [13], [14]. En [15] se exponen los resultados de utilizar reuniones mediante vídeo conferencia o mediante discusiones por ordenador para evitar los problemas de posibles enfrentamientos cara a cara entre los participantes.

Problemas en el conocimiento: Los problemas de las limitaciones cognitivas del ser humano aparecen, como no podría ser de otra manera, en las actividades de elicitación de requisitos. Algunos de estos problemas son:

- El ingeniero de requisitos debe tener un conocimiento adecuado del dominio del problema y no hacer suposiciones sobre ello, al igual que los clientes y usuarios no deben hacer suposiciones sobre aspectos tecnológicos. Muchas veces, las ideas preconcebidas sobre una posible solución afectan a la forma en que se establece la definición del problema. En el caso del ingeniero de requisitos, es fundamental que conozca el dominio del problema y el entorno tanto organizacional como operacional en el que deberá funcionar el sistema a desarrollar [16]

- El Ingeniero de requisitos tiene que ser capaz de determinar las fuentes de información, las cuales no siempre están definidas de forma explícita.
- Cuando los problemas son grandes y complejos, algunas personas tienden a hacer simplificaciones no válidas, a ignorar las partes más complejas o a centrarse únicamente en los aspectos que más conocen o que más les afectan. El ingeniero de requisitos debe ser capaz de manejar problemas complejos y asegurarse de que todos los temas importantes son tratados.

Problemas de conducta humana: La naturaleza social de la elicitación de requisitos provoca que surjan problemas de conducta humana, algunos de los cuales son los siguientes:

- Puede haber conflictos y ambigüedades en los roles que cada persona debe jugar en el proceso de elicitación. Dentro del grupo de clientes y usuarios, algunos pueden pensar que, aunque conozcan ciertas necesidades o ciertos aspectos importantes, es responsabilidad de otros participantes más afectados el hacerlas explícitas, con lo que el resultado final es que nadie dice nada.
- Otra situación similar, puede producirse si algunos clientes y usuarios piensan que los desarrolladores les harán todas las preguntas necesarias sobre el dominio del problema y los desarrolladores piensan que los clientes y usuarios les proporcionarán toda la información necesaria sin necesidad de preguntar por su parte, con lo que pueden quedar aspectos sin tratar. La suposición o el temor a que el sistema a desarrollar cambie su forma de trabajar o incluso ponga en peligro su puesto de trabajo, puede provocar que algunos usuarios retengan información o incluso saboteen el desarrollo, por ejemplo proporcionando información falsa. Es fundamental que el ingeniero de requisitos conozca el entorno organizacional para poder detectar y tratar este tipo de problemas [17].

Problemas técnicos: Otros problemas de la elicitación pueden considerarse como *técnicos*, entre ellos se pueden citar:

- El software tiene que resolver problemas cada vez más complejos, por lo que sus requisitos son también cada vez más complejos y contemplan detalles cada vez más específicos del dominio del problema.

- Los requisitos cambian en el tiempo, ya que a medida que los clientes y usuarios van conociendo sus propias necesidades y las posibilidades que les ofrece la tecnología puede surgir la necesidad de reconsiderar decisiones anteriores o descubrir nuevas necesidades. Esta es una de las principales causas de la naturaleza iterativa de la ingeniería de requisitos.
- Otra fuente de cambios es el hecho de que el hardware y el software cambian rápidamente, haciendo asequibles requisitos que antes eran inabordables por su complejidad o por su costo.

1.2 Análisis de Requisitos

Este es el segundo paso de la ingeniería de requisitos en el cual se estudia la información extraída en el primer paso de elicitación de requisitos, el cual es muy importante a la hora de identificar la presencia de áreas no detectadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes aquí se clasifican y se organizan en subconjuntos, se estudia cada requisito con relación al resto, examinan los requisitos en cuanto a sus consistencia, completitud y ambigüedad.

La tarea de análisis de los requisitos es un proceso de descubrimiento y refinamiento, El ámbito del programa, establecido inicialmente durante la ingeniería del sistema, es refinado en detalle.

El análisis y negociación de los requisitos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Puesto que el contenido de comunicación es muy alto, abundan los cambios por mala interpretación o falta de información. El dilema con el que se enfrenta un ingeniero de software puede ser comprendido repitiendo la sentencia de un cliente anónimo: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que lo que creíste oír sea lo que yo quise decir".

1.2.1 Los problemas del análisis de requisitos.

El primer problema que se presenta es la captura de los requisitos del usuario. Para empezar, es necesario recoger los requisitos de los usuarios o clientes de una manera sistemática y organizada.

Una vez conseguidos los requisitos, se pasa a la fase de análisis. En ella, lo que se hace es analizar los requisitos obtenidos de los usuarios con el fin de comprenderlos, y a partir de ellos desarrollar una especificación de la aplicación, que deberá ser completa y consistente, y deberá estar expresada de una manera al menos semiformal, no simplemente textual. En este proceso, se encuentran habitualmente gran

cantidad de problemas en los requisitos, lo cual llevará de vuelta a los usuarios con el fin de mejorar la calidad de los requisitos: pero se debe abordarles sabiendo lo que se quiere conseguir, qué aspectos de los requisitos obtenidos inicialmente se interesa aclarar, y el por qué.

1.3 Negociación

En los diferentes pasos de la ingeniería de requisitos, tanto en la elicitación como en el análisis se lleva a cabo una negociación con el cliente. Es muy importante que esta tenga éxito puesto que se debe entender qué quiere en realidad el cliente.

Según una definición informal, negociar es el hecho de tratar ciertos asuntos para lograr mejores resultados. La pregunta inmediata es: ¿Para Quién?, si la respuesta es tan obvia como la pregunta: Obtener los mejores resultados para ambas partes.

La negociación se puede definir como la relación que establecen dos o más personas en relación con un asunto determinado con vista a acercar posiciones y poder llegar a un acuerdo que sea beneficioso para todos ellos.

Al negociar, se debe estar preparado para ceder en algunos puntos frente a las reclamaciones de la otra parte. Si se fijan las posiciones antes de empezar la negociación, esta será un fracaso.

Una idea que no se cansan de repetir los expertos, es que jamás se debe acudir a una negociación sin antes haberla planificado detenidamente. Y no hay nada maquiavélico en ello. La planificación es la única forma de conocer exactamente cuáles son sus necesidades y sus límites. Además, le ayudará a descubrir los posibles objetivos y peticiones de la otra parte.

También se debe tener en cuenta, que planificar la negociación no significa que vaya a suceder todo tal y como se tiene previsto. Siempre hay que ir dispuesto a adaptarse a las circunstancias y a proponer un aplazamiento del acuerdo si las mismas te obligan a replantearte el objetivo.

A continuación se muestran algunos puntos importantes que se deben tener en cuenta para realizar una buena negociación:

- Conoce a tu oponente. Averigua con quién te vas a enfrentar y qué tipo de negociador es.
- Elige la estrategia más adecuada. En cada negociación se debe seguir una estrategia diferente, teniendo en cuenta la información obtenida de la otra parte, la situación de poder, los objetivos y las metas. Estos factores determinan las tácticas y argumentos a preparar. Por ejemplo, al estar en desventaja, se utilizará un estilo de negociación más suave y cooperante.
- Busca todas las alternativas posibles. Se debe preparar la negociación de la forma más creativa posible, buscando siempre el mayor número de variables o factores negociables que conduzcan a tu objetivo final.
- El entrenamiento. Si te sientes inseguro, ensaya tu postura con un compañero. La simulación te ayudará a reconocer tus puntos débiles y buscar alternativas para superar los posibles momentos de tensión.

En las negociaciones difíciles, las situaciones de bloqueo son muy frecuentes, pero casi siempre existe una salida:

- Busca objetivos comunes. Si el impasse se ha producido por un punto concreto, déjalo para el final y continúa con la negociación. Si hay más puntos de fricción, intenta aislar los desacuerdos.
- Una pausa para replantear la situación. Toma un respiro para pensar detenidamente cómo podéis cambiar las condiciones sin alterar la suma final. Por ejemplo, en lugar de negociar sobre valores fijos, puedes hacerlo sobre porcentajes.
- Mantén sobre la mesa tu mejor oferta. Aunque la otra parte te presione, no debes retirar tu mejor oferta. Intenta resaltarla para que vea las ventajas del acuerdo.
- Recurre a un intermediario. Una tercera persona neutral puede hacer que se empiecen a suavizar las posturas.
- Cambia de aires. Otra opción es que continúe la negociación en un lugar más distendido; por ejemplo durante el almuerzo.
- Aplaza el encuentro para otra ocasión. Cuando resulte imposible salir del bloqueo, lo mejor es que se deje el acuerdo final para otro momento. Recuerda que esta solución siempre es más positiva.

que dinamitar cualquier oportunidad de acuerdo o llegar a un trato poco satisfactorio para cualquiera de los dos.

1.3.1 Factores que intervienen en la negociación

- ✓ Tiempo.
- ✓ Predisposición (actitud).
- ✓ Conocimiento del interlocutor.
- ✓ Oportunidad de la entrevista (momento).
- ✓ Referencias que se deben ofrecer.
- ✓ Grado de profesionalidad (experiencia).
- ✓ Personas que intervienen en la negociación.
- ✓ Medios económicos.

1.3.2 Resultados de la negociación

- ✓ Cliente gana yo pierdo
- ✓ Cliente pierde yo pierdo
- ✓ Cliente gana yo gano...la mejor/satisfacción mutua

1.3.3 Como conseguir la predisposición

Existen algunos aspectos que conviene recordar a la hora de obtener la predisposición en el cliente. Entre las más importantes se encuentran:

- ✓ Poder.
- ✓ Empatía.
- ✓ Simpatía.
- ✓ Observar.
- ✓ Preguntar.
- ✓ Escuchar de forma activa.
- ✓ Resumir.
- ✓ Tomar notas.
- ✓ Conviene recordar que la predisposición es aquella fase que persigue.

En resumen, lo que se debe hacer es “Quitar hierro a las diferencias y resaltar los puntos en común”.

1.3.4 Normas para negociar con éxito

No utilizar palabras como nunca, imposible, jamás, etc...Utilizar: tal vez, o ¿cuáles son las alternativas?

1. Se debe comenzar a negociar antes de que la otra parte se dé cuenta del proceso. Recoger datos y pruebas de la otra parte y de las alternativas que ofrece es una gran ventaja antes de empezar a negociar. Es decir, información es poder.
2. Las propuestas por escrito iniciales deben estar bien presentadas y siempre con el objetivo de dejar claro el posicionamiento inicial.
3. Negociar es el proceso que fomenta la colaboración ¡Adelántalo!
4. No alardear del éxito obtenido, que sea la otra parte la que piense que ha alcanzado el mejor acuerdo.
5. Negociar con aquellos que puedan tomar la decisión aunque sin olvidar las que influyen en las decisiones.
6. No ponerle las cosas fáciles al contrario. Las personas se sienten más satisfechas con aquellos acuerdos que les han costado mucho esfuerzo.
7. Actuar siempre con cierto margen de maniobra, es decir, "espacio" para ceder a cambio de más.
8. Utilizar la Técnica de preguntas cuando dudas de la veracidad de una situación.
9. No se le da credibilidad a todo lo que se oye por casualidad.
10. No olvidar que Negociación es un " proceso continuo" con el cliente.
11. Utilizar un único canal por dónde pasen todas las comunicaciones: "Hablar con muchas personas de la otra parte, pero siempre manteniendo informado a tu interlocutor principal.
12. No hacer una pregunta si sabes que la respuesta no conviene.
13. Recapitule a menudo y resuma antes de terminar la reunión con el cliente. No dejar los acuerdos a medias.
14. No haga concesiones a menos que consiga algo a cambio.
15. No trate de ser un sabiondo.
16. No insista sobre aquellos temas que presenten dificultades para conseguir un acuerdo de colaboración mutua con el cliente.

1.3.5 Objeciones del cliente

La objeción es cualquier duda o silencio del cliente que retrase el cierre de un acuerdo. Existen varios tipos de objeciones, entre ellos se mencionan a continuación:

- ✓ Siempre el problema es nuestro.
- ✓ Suponiendo...
- ✓ Boomerang...
- ✓ Implicación.
- ✓ Si, pero...
- ✓ Actitud positiva-silencio.

Reglas para el tratamiento de objeciones:

1. No olvidar que, una objeción no es necesariamente negativa, ya que podría ser una demostración de interés.
2. El buen profesional mantendrá la calma y buscará a través de la técnica de preguntas el "sentido real" de la objeción. Por ejemplo ¿qué quiere Vd. decir por falta de servicio durante períodos promocionales?
3. Es preciso desviar las objeciones que salgan a principio de una entrevista hasta el final de la misma. Por ejemplo, Sr. cliente, si me permite trataré su comentario en seguida, pero primero... sugiero que se analicen una serie de "oportunidades"
4. Nunca es conveniente contradecir abiertamente al cliente.
5. Al aparecer dudas ante la objeción, se debe volver a la necesidad más importante del cliente.
6. Es importante "vaciar" al cliente de objeciones. Una objeción "guardada" es un obstáculo para la venta. Por ejemplo, Sr. Cliente ¿tiene Vd. alguna duda? ¿Desea usted añadir algo más?
7. El desconocimiento de las necesidades "profesionales" y "personales" del cliente es causa directa de un exceso de objeciones por parte del cliente.
8. Frente a una objeción "real" y donde el cliente lleva la razón, es importante reconocer su validez y volver a la necesidad prioritaria.

1.4 Especificación de requisitos

El término de especificación puede tener diversos significados para diferentes personas. Una especificación puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de casos de uso, un prototipo o una combinación de todo lo citado anteriormente.

La “Especificación de los Requisitos del Software” es el producto final sobre los requisitos del sistema obtenido por el ingeniero, describe la función y las características de un sistema de computación y las restricciones que gobiernan su desarrollo. La especificación delimita cada elemento del sistema y describe la información que entra y sale del mismo.

Una especificación es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema [18]

La especificación de requisitos es la documentación de requisitos esenciales (funciones, rendimiento, diseño, restricciones y atributos) del software y de sus interfaces y surge como resultado de la tarea de análisis y negociación, esta especificación debe ser evaluada y aprobada por el usuario.

1.4.1 Características de la especificación de requisitos

- Debe incluir información veraz
- Debe comunicar la información de forma eficaz
- Debe expresar qué hacer y no cómo hacerlo
- Describir los requisitos software
- Excluir cosas innecesarias
- No debe incluir detalles del diseño, de la verificación o de la dirección, salvo las restricciones de diseño que afecten a los requisitos.

-Características deseables

- Fácil de modificar ya que debe tener una organización coherente y no ser redundante
- Facilidad para identificar el origen y las consecuencias de cada requisito
- Facilidad para su uso en explotación y mantenimiento y un personal no relacionado con el desarrollo se encargue del mantenimiento
- Es una fuente de conocimientos

La especificación de requisitos proporciona a los clientes describir de manera precisa que es lo que quieren obtener y les brinda a los desarrolladores comprender qué es lo que quiere el cliente. Tiene como

objetivos de la especificación de requisitos establecer una base de acuerdo entre clientes y desarrolladores de lo que debe hacer el software, proporciona una base para estimaciones y un “contrato” para validación y verificación, define un documento base para futuras versiones o ampliaciones estableciendo con ello un punto de inicio para la comprensión del proceso de desarrollo.

1.4.2 Problemas de la especificación de requisitos

En el momento en que se requiere más información o cuando surgen discrepancias, se buscan más detalles en las especificaciones. Es entonces donde muchas veces aparecen problemas porque las especificaciones no son adecuadas y, en vez de aclarar la intención del diseñador, crean complicaciones adicionales. Algunos de los problemas encontrados son la falta de organización, el uso de generalidades, omisiones o ambigüedades, la inclusión de aspectos que no aplican, el requisito de hacer pruebas que han sido descontinuadas y el especificar materiales, método y los resultados. A continuación, se abunda sobre estos problemas encontrados.

Cuando las especificaciones están mal organizadas estas tienden a ser voluminosas y es muy difícil encontrar lo deseado. Además, algunas mezclan diferentes tipo de información en un mismo párrafo lo que contribuye a aumentar la confusión, por eso es de gran importancia que el documento de especificación de requisitos esté bien organizado en capítulos.

Otro problema encontrado es que muchas especificaciones definen generalidades y no especifican claramente el trabajo a ejecutar, por lo que a veces se observa que algunas indican que el trabajo se debe ejecutar "a satisfacción del ingeniero". Esto abre espacio, para que dos personas distintas que actúen como ingeniero del proyecto, exijan requisitos diferentes. Esto ocurre también en especificaciones que indican que el trabajo se deberá ejecutar "según lo determine el ingeniero".

También se pueden ver en la práctica muchas especificaciones con omisiones y ambigüedades. Esto ocurre, cuando se usan plantillas para especificaciones como parte de las especificaciones de un proyecto sin haber sido ajustadas para las peculiaridades del proyecto particular.

Otro problema común cuando se utilizan plantillas de especificaciones es que estas contienen requisitos que no se aplican en situaciones particulares.

En los párrafos anteriores se han descrito algunos problemas encontrados en las especificaciones de los proyectos de construcción. ¿Cómo se pueden evitar? Primeramente se debe tener más cuidado en la preparación de la especificación de requisitos. Esto requiere asignar un presupuesto adecuado para su elaboración. Se recomienda utilizar una plantilla de especificaciones que se ajusten a los detalles de cada proyecto. Este ajuste se debe hacer luego de conocer bien el proyecto y con la mentalidad de evitar problemas como los antes citados. Se recomienda que se tenga un departamento, o al menos un ingeniero, a cargo de las especificaciones.

1.5 Modelado de Sistema

Para especificar completamente lo que se va a desarrollar, se necesita un modelo de sistema que sirva para evaluar los componentes del sistema y sus relaciones entre sí, determinando como se reflejan los requisitos, y valorar como se ha concebido la estética del sistema.

Todos los sistemas basados en computadora pueden modelarse como una transformación de la información empleando una arquitectura del tipo entrada-proceso.

1.6 Validación de requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencia sin omisiones que lo 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.

1.7 Gestión de requisitos

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de desarrollo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento. Muchas de las actividades de la gestión de requisitos son tomadas de la gestión de configuración del software.

1.8 Modelos de proceso de la ingeniería de requisitos

A continuación, se exponen los principales modelos de proceso encontrados en diferentes bibliografías.

1.8.1 Modelo de Proceso de Duran

Este modelo consta de tres actividades fundamentales: elicitación, análisis y validación, y su principal característica es la interactividad. Ver figura 1.1

El proceso de elicitar y negociar requisitos, analizarlos y validarlos es iterativo por naturaleza, ya que prácticamente es imposible obtener todos los requisitos y que éstos tengan las propiedades antes discutidas sin tener que volver atrás en algún momento del proceso. Sin embargo, no por ello se debe dejar de intentar elicitar, y posteriormente analizar y validar, la mayor cantidad posible de requisitos en cada iteración, intentando alcanzar la situación ideal en el que el proceso sería lineal.

Al igual que en cualquier otra fase de la ingeniería de software, en la ingeniería de requisitos se incluyen también aspectos de gestión de recursos, verificación, control de calidad, control de configuración, etc.

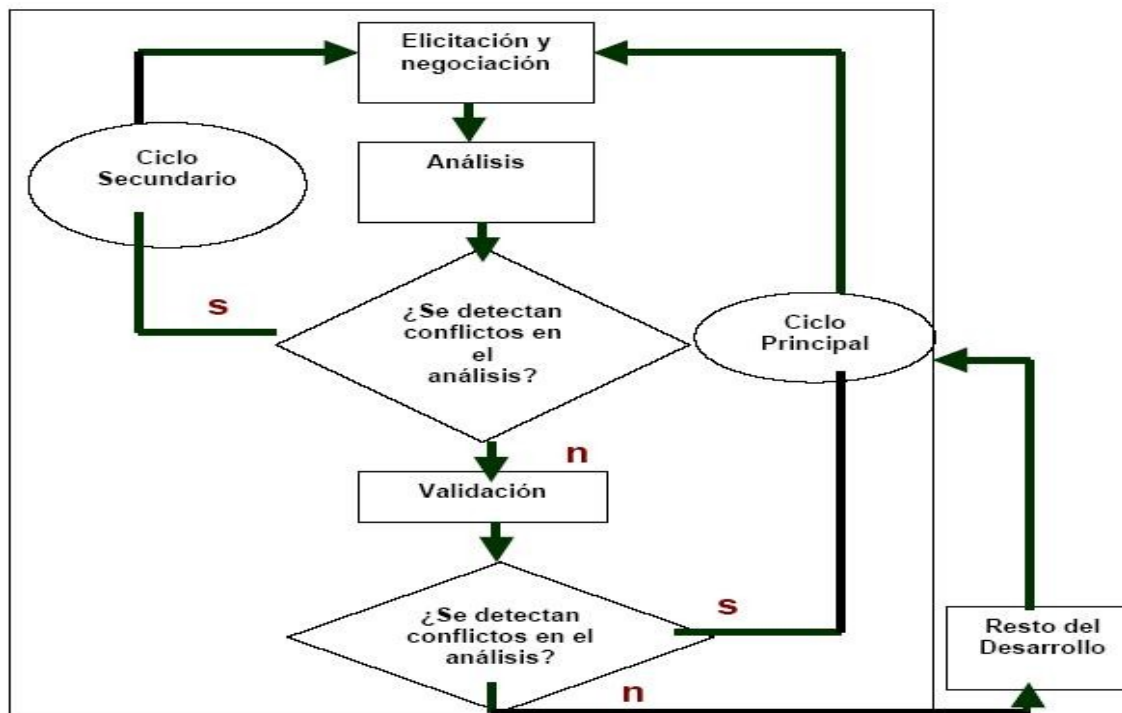


Figura 1.1 Modelo de Proceso de Duran

1.8.2 Modelo de Pohl

El modelo de Pohl [19] es un modelo iterativo en el que se definen las cuatro actividades que pueden verse en la figura 1.2, el orden de realización de las actividades puede ser cualquiera, en [20] se asume una secuencia en la que los requisitos son elicitados, a continuación son negociados entre los participantes, se integran con el resto de la documentación y finalmente se validan y verifican para asegurar que corresponden con las necesidades reales de los clientes y usuarios y que no presentan conflictos con los demás requisitos.



Figura 1.2 Modelo de procesos de ingeniería de requisitos de Pohl

Este modelo presenta cuatro características principales que se explican a continuación:

Elicitación de requisitos: El objetivo de la elicitación es mostrar el conocimiento oculto sobre las necesidades de clientes y usuarios y el sistema a desarrollar para que sean de fáciles para la comprensión de todos los participantes en el problema. En este modelo durante el desarrollo de las actividades es necesario identificar a las fuentes de información, conocer lo mejor posible el dominio del

problema, reutilizar especificaciones de requisitos similares en la medida de lo posible y utilizar las técnicas habituales de elicitación como son las entrevistas, casos de uso, cuestionarios, prototipos, etc.

Negociación de requisitos

Tiene como objetivo lograr acuerdos entre todos los participantes sobre los requisitos elicitados, avanzando en la dimensión de acuerdo del proceso. Para ello hay que tener en cuenta cuatro factores:

- ✓ Hacer explícitos los conflictos y evitar los conflictos emocionales entre los participantes, de forma que quede claro qué es lo que se negocia y que dicha negociación no se vea afectada por motivos personales.
- ✓ Hacer explícitos para cada conflicto las alternativas, las argumentaciones y las razones subyacentes que los provocan, de forma que la negociación pueda basarse en las raíces del conflicto.
- ✓ Asegurarse de que se toman las decisiones correctas, de forma que la mayoría de los participantes estén de acuerdo en los resultados de la negociación y no se sientan desplazados del proceso.
- ✓ Asegurarse de involucrar a las personas adecuadas en el momento adecuado, para evitar tener que volver a replantear las negociaciones porque alguno de los participantes afectados no participó en las negociaciones oportunas.

Especificación /documentación de requisitos.

El objetivo es claro: documentar los requisitos elicitados y negociados. Pohl [21] propone utilizar las notaciones que sean necesarias para que todos los participantes la entiendan. De este modo, según su propuesta, se avanza en la dimensión de formalidad del proceso.

Validación/Verificación de requisitos.

El objetivo es comprobar que los requisitos documentados corresponden a las necesidades de los clientes y usuarios (validación) y comprobar que la especificación cumple los criterios de calidad oportunos (verificación).

1.8.3 Modelo Espiral

El modelo espiral [22], está basado en el modelo espiral de Boehm para la ingeniería de requisitos [23] y el Inquiry-Based Model de Potts [24]. En este modelo se asume una naturaleza iterativa del proceso y la

dificultad de establecer un punto de terminación del mismo, dado que los requisitos nunca llegarán a ser perfectos. A parte de las tres actividades que se describen a continuación, el modelo asume que existe una cuarta, la gestión de requisitos, que se realiza durante todo el proceso y que se encarga de gestionar la obtención incremental de los requisitos y los inevitables cambios a los que están sujetos. Ver figura 1.3

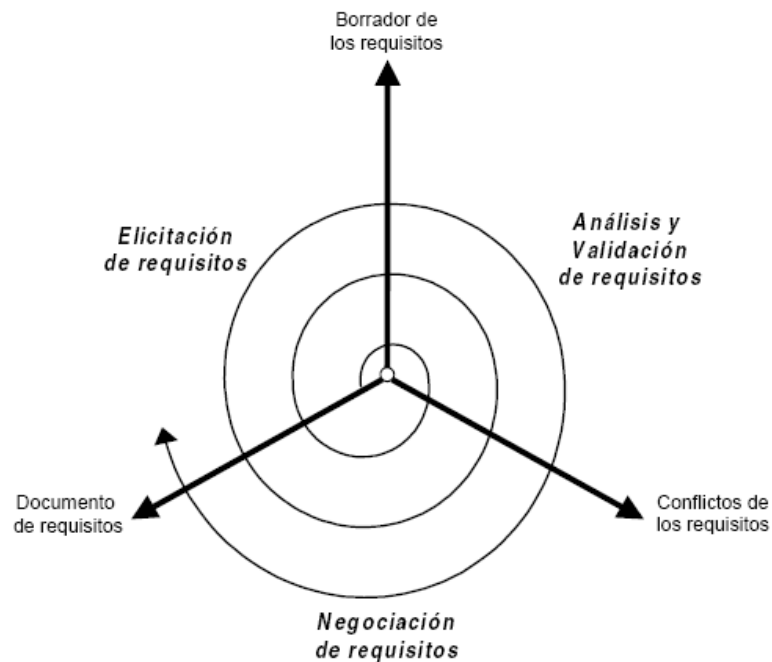


Figura 1.3 Modelo de procesos de ingeniería de requisitos de espiral.

Elicitación de requisitos: En esta actividad, se consultan las diferentes fuentes de información como clientes, usuarios, expertos en dominio, etc., con el propósito de entender el dominio del problema y establecer los requisitos del sistema a desarrollar, obviamente, es factible que los requisitos elicitados no sean completos y que se expresen en forma vaga o no estructurada.

Análisis y validación de requisitos: Los requisitos elicitados se integran y analizan, lo que suele provocar la identificación de requisitos que faltan, inconsistencias y conflictos entre los requisitos. Aunque en [25] se denomina a esta actividad análisis y validación de requisitos, sin embargo no se hace referencia a ningún tipo de actividad de validación tal como se entiende el término en esta tesis, es decir, la confirmación por parte de clientes y usuarios de que los requisitos reflejan realmente sus necesidades y especifican el sistema que ellos desean ver figura 3.

Negociación de requisitos: Su objetivo es solucionar todos los conflictos que se hayan presentado durante el análisis, llegando a acuerdos entre los participantes en el proceso. Normalmente hay que elicitar nuevamente.

1.9 Metodologías de desarrollo del software.

En la Ingeniería del software se propone el uso de metodologías que definan y especifiquen el proceso de desarrollo de un producto de software. No se tiene definido una metodología de software. Sino que dependiendo de las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. A continuación se muestran algunas de las más usadas.

1.9.1 Extreme Programming.

Extreme Programming (XP) es una metodología ágil que su principal misión es potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

El ciclo de vida de XP consta de seis fases:

1. Exploración: Los clientes van a decir las historias de usuario que son de interés para la primera entrega del producto, y el equipo de desarrollo se familiariza con las herramientas, tecnologías, lenguaje y prácticas que se utilizarán en el proyecto.
2. Planificación de la Entrega: Se establece la prioridad a cada historia de usuario y se realiza una estimación de esfuerzo necesario para cada una de ellas.
3. Iteraciones: Incluye varias iteraciones sobre el sistema antes de ser entregado
4. Producción: Se realizan pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
5. Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.
6. Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema.

1.9.2 Microsoft Solutions Framework.

Microsoft Solutions Framework es la metodología empleada por Microsoft para el desarrollo de software. MSF es un conjunto de principios, modelos, disciplinas, conceptos, lineamientos y prácticas probadas, que de acuerdo al contexto de proyecto al tamaño del equipo, frecuencia de entregas, etc. serán más o menos recomendables de aplicar.

Propone varias fases como:

Fases que propone MSF:

1. Estrategia y alcance: Se inicia la preparación previa al desarrollo del proyecto.
2. Planificación y Prueba de Concepto: Se refiere a toda la planificación del proceso, además del diseño de la arquitectura.
3. Estabilización: La solución implantada en la maqueta se pasa a un entorno real de explotación, restringido en número de usuarios y en condiciones tales que se pueda llevar un control efectivo de la situación.
4. Despliegue: Se llevarán a cabo en esta fase los planes diseñados en la anterior, principalmente el de despliegue y el de formación.

1.9.3 Proceso de la ingeniería de requisitos según RUP

RUP no es más que un proceso de desarrollo de software, o sea, una forma disciplinada de asignar tareas y responsabilidades en una empresa o proyecto de software. Sus objetivos se basan en asegurar que se produzca un software con toda la calidad requerida en un plazo y presupuestos predecibles, dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Además, es un producto desarrollado y mantenido por Rational, el mismo es actualizado constantemente puesto que tiene en cuenta las mejores prácticas de acuerdo con la experiencia, aumenta la productividad de los desarrolladores mediante el acceso a bases de conocimiento, plantillas y herramientas. Se centra en la producción y en el mantenimiento de modelos del sistema más que en producir documentos. Constituye una guía de cómo usar UML de la forma más efectiva, y contiene herramientas de apoyo durante todo el proceso como el modelamiento visual, programación, pruebas, etc.

En cuanto a la ingeniería de requisitos se puede decir que RUP por su parte describe como obtener los requisitos, organizarlos, documentarlos en cuanto a su funcionalidad y restricciones, y describe además su

forma de captar y comunicar los requisitos del negocio. Los casos de uso y los escenarios indicados por el proceso han probado ser una buena forma de captar requisitos y guiar el diseño, la implementación y las pruebas.

RUP divide su proceso de desarrollo en ciclos obteniéndose un producto final en cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción y Transición, cada una de ellas se ejecutan en una o más iteraciones. Este proceso cuenta con nueve disciplinas en las cuales se ejecutan diferentes flujos de trabajo, para cada uno de ellos se describen las actividades y los roles.

El segundo flujo de trabajo es el de Requerimientos o Requisitos el cual constituye uno de los más importantes del proceso y se desarrolla a lo largo de todo el ciclo de vida del software. Los desarrolladores y clientes deben acordar que es lo que el sistema debe hacer, revelando los requisitos, documentando su funcionalidad y restricciones, identificando los casos de uso los cuales describen la funcionalidad, los actores, y especificando de forma complementaria los requisitos no funcionales, todo esto en un documento al cual se le llama Documento Visión y Especificaciones adicionales.

Se lleva a cabo un análisis de los requisitos, se define el mismo como el conjunto de actividades de la Ingeniería del Software que se ocupa de recoger las necesidades de clientes y usuarios (requisitos) sobre un sistema dado, y de traducirlas a especificaciones técnicas del sistema.

Concluyendo, RUP se basa en casos de uso para obtener la especificación de requisitos y describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) para la realización de los modelos. Describe los requisitos de la aplicación desde el punto de vista del usuario y define los requisitos técnicos sin meterse con detalles de implementación.

En este capítulo, después de haber estudiado la ingeniería de requisitos, se puede concluir que por ser una disciplina muy reciente, no tiene todavía definido claramente sus terminologías, los procesos y los productos que en ella se elaboran, por ello es de vital importancia comprender los procesos de los modelos de desarrollo existentes, para llegar a entender los problemas de la ingeniería de requisitos y poder desarrollar algún procedimiento para su desarrollo. Una vez estudiados y analizados estos modelos de los mismos se puede decir que tienen múltiples características comunes a partir de las cuales se diseñará la propuesta.

En el próximo capítulo se hará un estudio de cómo se lleva a cabo este proceso de elicitación, análisis y negociación, y especificación en una muestra de los diferentes proyectos de la universidad.

CAPÍTULO 2: UTILIZACIÓN DE LA INGENIERÍA DE REQUISITOS, ANÁLISIS Y NEGOCIACIÓN, Y ESPECIFICACIÓN.

En este capítulo se tratará el tema de cómo se realiza en la Universidad de las Ciencias Informáticas la ingeniería de requisitos, centrándose principalmente en los tres primeros pasos de la ingeniería de requisitos: elicitación, análisis y negociación, y especificación.

Para tener una mejor visión de cómo se desarrolla este tema en los proyectos productivos, se utilizó una técnica de recogida de información, la entrevista. Este análisis, se basó principalmente en entrevistas realizadas a una muestra de analistas y líderes de proyecto de la UCI, basada en preguntas que tienen como objetivo conocer si se utiliza algún procedimiento para la elicitación, análisis y negociación, y especificación de requisitos. O si usan algún tipo de técnica específica para estos pasos. Ver Anexo 1.

Se utilizó la entrevista porque es uno de los métodos que más resultados tiene a la hora de comunicarse con las personas, sus preguntas son fáciles y eficientes a la hora de recoger una información específica.

Se aplicó una entrevista estructurada, se realizan preguntas que ya han sido creadas con una previamente, en el capítulo siguiente se profundiza acerca de esta técnica.

Este trabajo se desarrolló, gracias al apoyo y colaboración de los vicedecanos de las facultades que proporcionaron toda la información posible para la localización de los analistas y líderes de proyectos productivos. Se realizaron 27 encuestas en las facultades 1, 7, 8 y 10.

Para la realización de la diferentes tablas que se mostrarán en el capítulo, se utilizó como herramienta Microsoft Office InfoPath 2003, esta racionaliza el proceso de reutilización, utilización y uso compartido de información mediante formularios ricos y dinámicos. Permite recopilar información con mayor eficacia y precisión, administra la información de un modo más flexible, también se utilizó la herramienta Microsoft Excel, para generar los datos del InfoPath creando así las gráficas utilizadas en esta investigación.

2.1 Cómo se comporta la elicitación, análisis y negociación, y especificación de requisitos en los proyectos productivos.

Para que se realice un buen proceso de elicitación, análisis y negociación, y especificación de los requisitos en los proyectos productivos de la Universidad de las Ciencias Informáticas, se necesita una serie de pasos a tener en cuenta y así lograr una mejor calidad que resolverá las necesidades del cliente y contribuirá a su mejor desempeño. En los proyectos entrevistados, se aprecia que no todos aplican un

procedimiento específico, sino que más bien usan algunas partes de diferentes metodologías o utilizan sus experiencias anteriores, en esta misma actividad de proyectos ya terminados. Con el desarrollo de esta investigación se muestra como se comporta, y en qué medida se utilizan o no.

2.2 Resultados de la entrevista.

En este epígrafe se muestran los resultados obtenidos en la realización de la entrevista de los proyectos productivos de la Universidad de las Ciencias Informáticas apoyándose principalmente de gráficas que proporcionarán una mejor representación de las preguntas que se hicieron en este material.

2.2.1 Tamaño del proyecto.

A través de las entrevistas realizadas se llegó a la conclusión de que la mayoría de los proyectos productivos de la Universidad de las Ciencias Informáticas son grandes o medianos y la minoría son pequeños. Esta es una característica muy importante, ya que entre más grandes sean los proyectos, será más complicado el proceso de elicitar, analizar y negociar y especificar los requisitos. Para una mejor comprensión se muestra una gráfica donde se aprecian estos datos. Ver figura 2.1

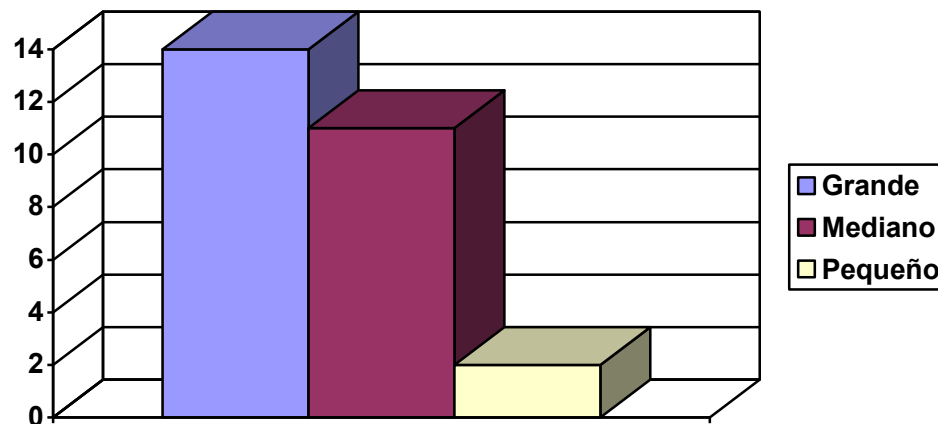


Figura 2.1: Tamaño del Proyecto

2.2.2 Metodologías usadas.

Con el uso de una metodología de desarrollo de software, se le proporciona al equipo de desarrollo la manera más eficiente de trabajar siguiendo un orden lógico. Lo que sí queda claro es que la mala

aplicación de una metodología o de ninguna, trae consecuencias no previstas tanto para el equipo de desarrollo como para el cliente.

Existen diversas metodologías y los proyectos adoptan las que de acuerdo a sus características principales le sean más útiles para su mejor desarrollo.

Una de las más utilizadas en la UCI es RUP con un 47 % de de los 27 proyectos entrevistados que la usan, 11 de ellos grandes y 9 medianos esto justifica su utilización, y la Metodología XP, principalmente usada en los proyectos pequeños. Ver figura 2.2

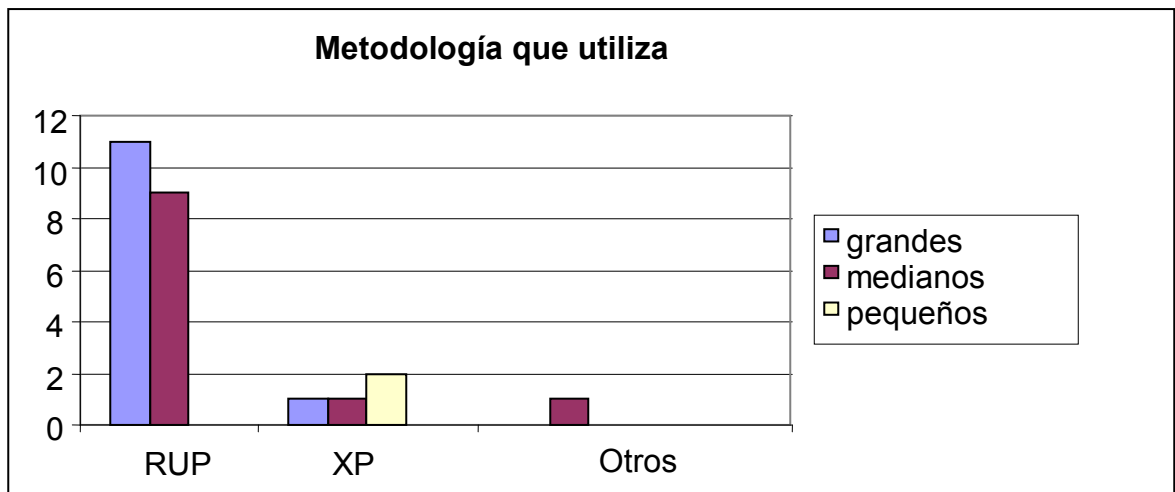


Figura 2.2: Metodología que utiliza

2.2.3 Procedimiento para la elicitación de requisitos.

La etapa de elicitación de requisitos abarca la primera, y quizás más importante fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de la elicitación de requisitos, es garantizar que los del sistema sean consistentes con las necesidades, de la organización donde se utilizará el mismo y con las futuras necesidades de sus usuarios.

Para ello, es necesario el desarrollo de un procedimiento que apoye en esta actividad. Muchos de los proyectos entrevistados usan un procedimiento o metodología, pero solo una minoría lo usa

correctamente, aunque vale aclarar que no aplican una en específico. Sino la unión de conocimientos de algunas de las metodologías más utilizadas, adquiridos por la experiencia en trabajo con proyectos productivos, y un 39 % no lo usa ningún momento. Ver figura 2.3

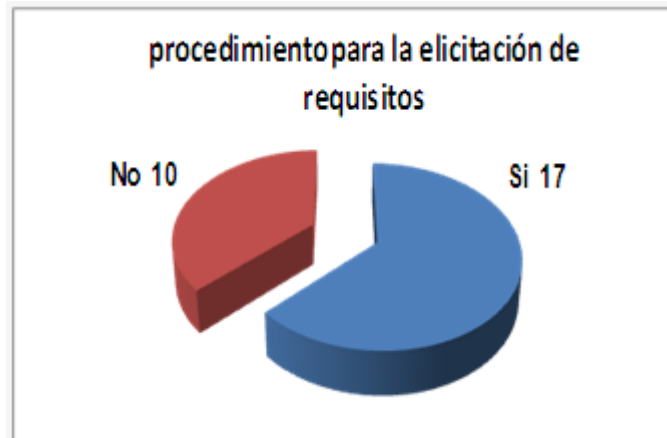


Figura 2.3 Procedimiento para la elicitación

2.2.4 Técnicas para la elicitación de requisitos.

Existen diversas técnicas para este paso de la ingeniería de requisitos, que serán utilizadas de acuerdo con las necesidades o características que posea el proyecto, a continuación se muestran las técnicas más usadas:

- ✓ Entrevistas Es una de las más conocidas debido a su gran ventaja: una mejor comunicación con el cliente.
- ✓ Join Application Development (JAD): Es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo.
- ✓ Tormenta de ideas: Es una técnica muy utilizada cuando se necesita evitar conflictos de requisitos, y puede ser utilizada como una técnica dentro de JAD.
- ✓ Casos de uso. Los casos de uso son una técnica para la especificación de requisitos funcionales. Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables.

En la figura 2.4 se muestra como se utilizan estas técnicas en cada proyecto entrevistado.

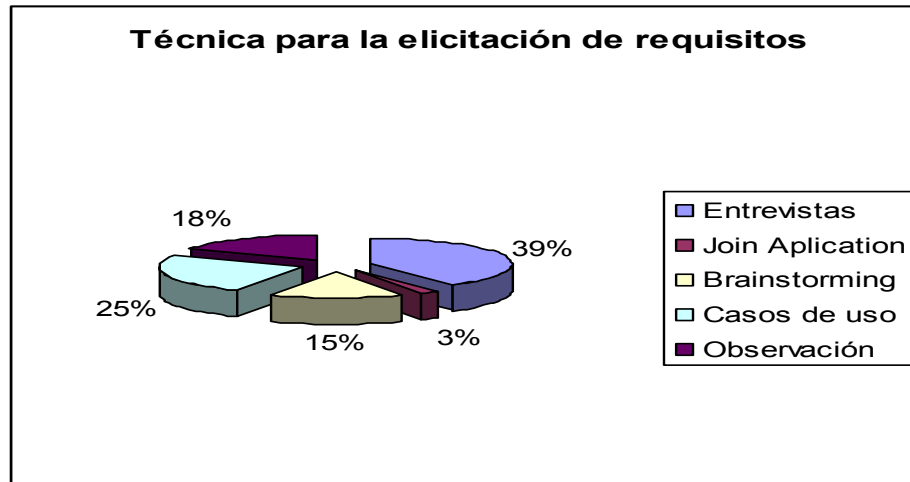


Figura 2.4 Técnicas para la elicitación de requisitos

2.2.5 Plantillas para las reuniones de elicitación o captación de requisitos.

Las plantillas son de gran utilidad ya que van guiando al ingeniero de software en la recogida de los requisitos. Se nota como un 22 % utiliza plantillas, y un 78 % que no la utiliza, como se puede ver en la figura 2.5. Muchos alegan por desconocimiento, que no hay necesidad de usarlas, entre otras opiniones, pero cuando se observan los resultados se aprecia que los proyectos que utilizan plantillas tienen más probabilidades de terminar las tareas en el tiempo previsto. Y tienen mejor organizado toda la parte de recogida de requisitos.

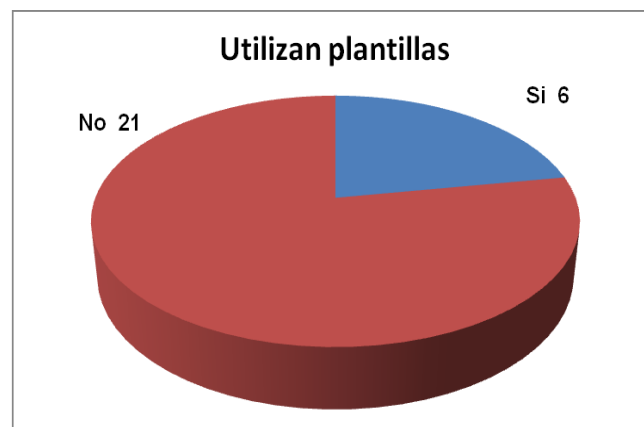


Figura 2.5 Plantillas de requisitos

2.2.6 Documentos que sean guardados como resultado de la elicitación de requisitos.

El Documento de Especificación de Requisitos garantiza establecer una base de acuerdo entre clientes y desarrolladores qué debe hacer el software, y define un documento base para futuras versiones o ampliaciones estableciendo un punto de inicio para la comprensión del proceso de desarrollo. Realizan un documento el 70 % los entrevistados, aunque no siempre se llama de la misma forma, ni tienen las mismas características, y un 30 % no lo utiliza. Ver figura 2.6

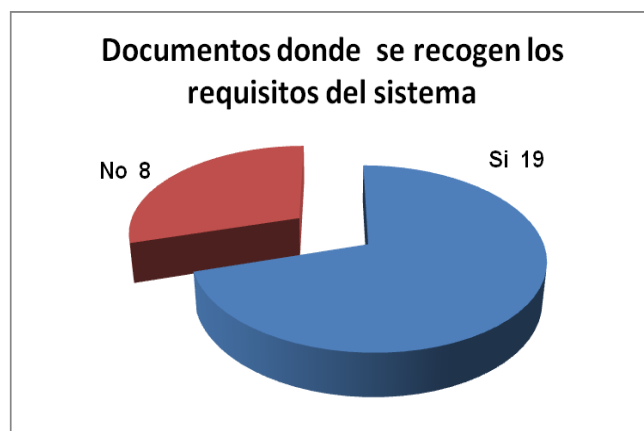


Figura 2.6 Documentos donde se recogen los requisitos del sistema.

2.2.7 Técnicas usadas para la especificación de requisitos.

Ninguna técnica es efectiva por si misma, requiere ser combinada con otras para lograr una correcta especificación de requisitos. Prácticamente en todos los proyectos se noto un notable uso de técnicas y la combinación de estas como se puede ver en la figura 2.7. Para una mejor comprensión se definen algunas de las más usadas:

- ✓ Especificación de casos de uso.
- ✓ Historia de usuario
- ✓ Especificación de Requisitos No Funcionales.

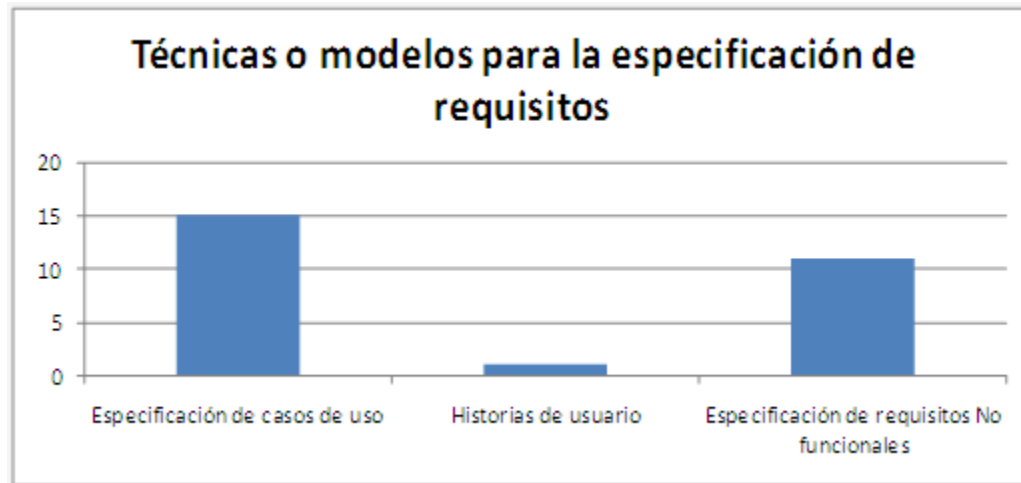


Figura 2.7 Técnicas o modelos para la especificación de requisitos.

2.2.8 Procedimiento para el análisis de requisitos.

Un procedimiento de análisis de requisitos, es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear, durante la actividad de la fase de ingeniería de requisitos del ciclo de vida del producto. Cuando se hace un uso apropiado de este, se logra una mejor calidad en el producto de software a desarrollar, por lo que es de vital importancia utilizar un buen procedimiento que satisfaga todas las necesidades. Se observa que solo un 33% de los proyectos usa algún tipo de procedimiento aunque vale aclarar que no tienen este paso muy claro, sino que utilizan algunas técnicas, modelos o plantillas pero que no están bien definidos. Y un 67 % no usa, y los principales motivos son:

- ✓ Desconocimiento de técnicas.
- ✓ No lo creen necesario.
- ✓ No tienen predefinido plantillas.

Ver figura 2.8 donde se muestra la cantidad de proyectos que utilizan o no un procedimiento para el análisis de requisitos.



Figura 2.8 Procedimiento para el análisis de requisitos

2.2.9 Proceso de elicitación, análisis y negociación, y especificación.

En el proceso de elicitación, análisis y negociación, y especificación; aunque existen grandes volúmenes de documentación y bibliografía acerca del tema, se necesita organizarla, y utilizar de cada una lo mejor, adaptándolo a las necesidades que pueda tener un producto. Muchas veces por evitar el trabajo de reunir la información necesaria, se obvian estos pasos tan importantes, o simplemente se hace de la forma que se piensa que pueda ser correcta. Sin guiarse por ciertos parámetros que logran una mejor calidad, debido a que suele tomarse mucho tiempo el estudio del procedimiento correcto y además se necesita de mucha experiencia en el tema.

En la Universidad de las Ciencias Informáticas se pudo apreciar con las entrevistas que no se realiza en la mayoría de las ocasiones un proceso correcto en los proyectos productivos.

En este capítulo, se hizo un análisis del comportamiento del proceso de elicitación, análisis y negociación, y especificación en los diferentes proyectos productivos de la universidad, a través de gráficas y conceptos a tener en cuenta, donde se comprobó que hay un déficit del uso de procedimientos y metodologías aplicables a proyectos, aunque se usan algunas metodologías y técnicas o la unión de varias de ellas. También, se vió la necesidad de hacer un procedimiento que proporcionara tareas, técnicas, documentos y plantillas que apoyen y guíen este proceso de vital importancia en un software, el cual se propone en el siguiente capítulo.

CAPÍTULO 3: PROCEDIMIENTO PARA LA ELICITACIÓN, ANÁLISIS Y NEGOCIACIÓN, Y ESPECIFICACIÓN DE REQUISITOS.

Durante la investigación, se obtuvo información sobre la importancia de la determinación de los problemas y necesidades de los clientes, para con ello poder realizar la especificación de los requisitos. Durante la investigación es posible encontrarse ante diferentes problemas de comunicación con el cliente. Es por ello que se debe analizar cada requisito para verificar su consistencia, que no sean ambiguos y asegurarse que estén completos. En caso de encontrarse ante una dificultad de ambigüedad o consistencia se deben entonces, negociar con los clientes los requisitos afectados.

Todo este proceso, posee una fuerte carga de comunicación y negociación, donde se puede afirmar que no siempre el ingeniero en informática posee la preparación académica suficiente, de forma tal que pueda guiar con éxito este proceso. Por ello en la descripción de cada actividad, se recomienda la técnica que se puede utilizar, no siendo esto un traje a la medida, pues cada ingeniero aplicaría la técnica que mejor resultado la haya dado aún cuando no sea la recomendada. A continuación se describen las técnicas propuestas.

3.1 Técnicas

En el procedimiento propuesto se recomiendan varias técnicas, que permitirán obtener el producto de las actividades que se han descrito anteriormente. Se tendrá en cuenta en qué ocasión es más eficiente la aplicación de una u otra.

Las técnicas más comunes en la elicitación de requisitos son: las Entrevistas, el Desarrollo Conjunto de Aplicaciones, la Tormenta de Ideas y la utilización de Escenarios o Casos de Uso.

A estas técnicas se le suman otras, que sirven como complemento para una mejor elicitación como la observación in situ, el estudio de documentación, los cuestionarios, la inmersión en el negocio del cliente. Es válido señalar que en un proyecto pueden ser aplicadas todas las que sean necesarias pues no son excluyentes unas de otras.

3.1. 1 Entrevistas

La entrevista es una de las técnicas mas utilizadas ya que son prácticamente imprescindibles para la comunicación de las personas implicadas en el proyecto. Esta cuenta con tres fases: preparación,

realización y análisis. Los desarrolladores usan principalmente esta técnica, cuando necesitan datos que solo el cliente conoce.

-Preparación de la entrevista.

Antes de realizar una entrevista, se deben tener en cuenta un conjunto de actividades de apoyo, para lograr mejores resultados.

- Planificar reuniones con el cliente. Para ello es de gran importancia conocer el dominio del problema para entender las necesidades de dicha comunidad, logrando una mayor confianza a los clientes y usuarios de que se les entiende sus problemas, y hay buena comunicación con ellos.
- Analizar las técnicas a utilizar en todo el proceso. Algunas de las técnicas más aplicadas por el ingeniero de requisitos son: revisar la documentación, bibliografía ya existente de proyectos similares, indagar acerca de la organización.
- Hay que seleccionar a las personas a las que se va a entrevistar y tratar de realizar solo las entrevistas necesarias, fundamentalmente a los directivos del centro que pueden dar una noción global del tema, luego a los futuros usuarios que son los que mas van a utilizar el software y al personal técnico, que aporta detalles sobre el entorno operacional de la organización.
- Preparar el documento de la entrevista y su contenido, para ellos es de vital importancia determinar el objetivo y contenido de las entrevistas, para hacer una mejor entrevista con menos tiempo. Previamente se pueden enviar cuestionarios para rellenar y devolver, con un documento a modo de introducir el proyecto de desarrollo para que los futuros entrevistados tengan una idea de lo que se les va a preguntar y el entrevistador posea más datos para preparar la entrevista. Es importante, que si se usan cuestionarios sean preparados, teniendo en cuenta quien los va a responder.
- Planificar las entrevistas: la fecha, hora, lugar y duración de las entrevista deben fijarse teniendo en cuenta siempre la agenda del entrevistado. En general, se deben buscar sitios agradables donde no se produzcan interrupciones y que resulten naturales a los entrevistados.

- Realización de entrevistas.

Dentro de la realización de la entrevista se producen tres etapas:

1 Etapa Inicio: el entrevistador se presenta e informa al entrevistado la razón de la entrevista, que objetivos quiere conseguir que tipos de preguntas se le van a realizar. Es de gran importancia lograr una buena comunicación con la persona entrevistada.

2 Etapa Desarrollo: En la entrevista debe distribuirse el tiempo de forma tal que la mayor parte sea para el entrevistado. Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo. Durante esta fase se pueden emplear distintas técnicas:

- Preguntas abiertas: también denominadas de libre contexto, estas preguntas no pueden responderse con un "sí" o un "no", permiten una mayor comunicación y evitan la sensación de interrogatorio. Por ejemplo, "¿Qué se hace para registrar un pedido?", "Dígame qué se debe hacer cuando un cliente pide una factura". Estas preguntas se suelen utilizar al comienzo de la entrevista, pasando posteriormente a preguntas más concretas.

En general, se debe evitar la tendencia a anticipar una respuesta a las preguntas que se formulan Utilizar palabras apropiadas: se deben evitar tecnicismos que no conozca el entrevistado y palabras o frases que puedan perturbar emocionalmente la comunicación.

- Mostrar interés en todo momento: es fundamental cuidar la comunicación no verbal [26] durante la entrevista: tono de voz, movimiento, expresión facial, etc. Por ejemplo, para animar a alguien a hablar puede asentirse con la cabeza, decir "ya entiendo", "sí", repetir algunas respuestas dadas, hacer pausas, poner una postura de atención, etc.

Debe evitarse bostezar, reclinarse en el sillón, mirar hacia otro lado, etc.

3 Etapa de Terminación: es de gran importancia, se debe volver hacer una revisión de la entrevista para confirmar que no hubo ninguna confusión en la información recogida, se agradece al entrevistado por su colaboración siempre dejando la posibilidad de volver a contactarlo en caso de que se necesita aclarar alguna duda o conflicto en la información.

- **Análisis de las entrevistas.**

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, la información, contrastarla con otras entrevistas o fuentes de información, etc. Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

- **Pasos importantes para el desarrollo de una entrevista**

Para la realización de una entrevista se planean una serie de preguntas que el cliente debe ir contestando, suelen ser muy naturales y, cuando son bien realizadas, son muy eficientes. Se requiere obtener el conocimiento en forma directa, rápida y sencilla.

-**Aspectos importantes para realizar una entrevista:**

-Realizar una preparación de la entrevista.

-Utilizar grabadora en caso de que pueda.

-Es recomendable que sean dos ingenieros con habilidades de analistas de sistemas quienes participen en el proceso, uno para que lleve el peso de la entrevista y otro que realice una labor crítica manteniendo la entrevista en el curso adecuado, como para concretar puntos claves.

-Realizarlas en un espacio físico confortable con el menor número de interrupciones.

Entrevista enfocada:

Tiene como objetivo conocer los temas que componen el dominio, el resultado es un conocimiento general del dominio. Es similar a una conversación cotidiana. La idea es obtener la mayor cantidad de información sobre el tema para realizar otro tipo de entrevistas posteriormente.

Entrevista estructurada:

Tiene como objetivo obtener la estructura de conceptos, partes del modelo, detalles de las reglas o explicaciones sobre aspectos detallados del conocimiento obtenido en entrevistas previas o bien sobre conocimiento obtenido de otra forma. Se realizan preguntas preparadas. Es una buena forma de atacar problemas específicos una vez que se tienen los elementos iniciales.

- Propuestas de Modelos para la Realización de Entrevistas

A continuación se da una visión de las partes principales de una entrevista para la elicitación de los requisitos y los tipos de entrevistas que se pueden hacer dependiendo del personal al que se le vaya a realizar.

➤ Entrevistas a la Dirección.

Se debe tener bien definidos los objetivos en este tipo de entrevista para así poder elaborar preguntas más específicas, principalmente los que aparecen a continuación.

Objetivos

- objetivos deseados con la entrevista
- estructura de la organización.
- interfaces con otros proyectos
- delimitar en lo posible el campo de estudio

Técnica: informal, periódica

Resultados: Visión del proyecto

- objetivos principales
- lista de puestos de trabajo
- campo de estudio
- restricciones: medios, calendario, legislación, etc.

➤ Entrevistas a puestos de trabajo

Objetivos:

- operaciones efectuadas (Lista de Tareas)
- eventos periódicos
- datos y documentos
- reglas y restricciones
- informaciones manipuladas
- qué puestos intervienen
- también mensajes electrónicos, telefónicos, fax,...
- lenguaje de la empresa

Técnica: Se debe intentar estructurar la información recibida, mediante fichas, representación gráfica...

Fichas de entrevista

El contenido de una ficha de entrevista a un puesto de trabajo será:

- Identificación: Nombre y Apellidos, Departamento al que pertenece, Empleo
- Operaciones que realiza y descripción: Documentos enviados y recibidos desde el puesto (incluidos los “documentos” orales) y descripción. Nombre, origen y destino, periodicidad, volumen, conservación/destrucción
- Otras preguntas que consideres necesarias para la comprensión.

3.1.2 El Desarrollo Conjunto de Aplicaciones (JAD).

La técnica denominada Desarrollo Conjunto de Aplicaciones (JAD), desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un período de 2 a 4 días. En estas reuniones, se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo. Esta se basa algunos principios de la dinámica de grupo: el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación en la que durante las reuniones se trabaja directamente sobre los documentos a generar.

El JAD tiene dos grandes pasos, el JAD/Plan cuyo objetivo es elicitar y especificar requisitos, y el JAD/Design, en el que se aborda el diseño del software. En este documento sólo se verá con detalle el primero de ellos. Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia. Aunque con ella se pueden obtener buenos resultados, especialmente para elicitar requisitos en el campo de los sistemas de información. En comparación con las entrevistas individuales, presenta las siguientes ventajas:

- Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- Todo el grupo, incluyendo los clientes y los futuros usuarios revisa la documentación generada, no sólo los ingenieros de requisitos.
- Implica más a los clientes y usuarios en el desarrollo.

- Participantes del JAD.

Se pueden distinguir seis clases de participantes o roles en JAD:

- Jefe del JAD: es el responsable de todo el proceso y asume el control durante las reuniones.
- Analista: es el responsable de la producción de los documentos que se deben generar durante las sesiones JAD.
- Patrocinador ejecutivo: es el que tiene la decisión final de que se lleve a cabo el desarrollo.
- Representantes de los usuarios: durante el *JAD/Plan*, suelen ser directivos con una visión global del sistema.
- Representantes de sistemas de información: son expertos en sistemas de información que deben ayudar a los usuarios a comprender qué es o no factible con la tecnología actual y el esfuerzo que implica.
- Especialistas: proporcionar información detallada sobre aspectos muy concretos, tanto del punto de vista de los usuarios porque conocen muy bien el funcionamiento de una parte de la organización, como desde el punto de vista de los desarrolladores porque conocen perfectamente ciertos aspectos técnicos de la instalación hardware de la organización.

- Dentro de la técnica del JAD se distinguen tres fases:

1 Etapa Adaptación: es responsabilidad del jefe del JAD, ayudado por uno o dos analistas, adaptar la técnica del JAD para cada proyecto. La adaptación debe comenzar por definir el proyecto a alto nivel, para lo cual pueden ser necesarias entrevistas previas con algunos clientes y usuarios. También, suele ser necesario recabar información sobre la organización para familiarizarse con el dominio del problema, por ejemplo utilizando técnicas complementarias como el estudio de documentación o la observación in situ. Una vez obtenida una primera idea de los objetivos del proyecto, es necesario seleccionar a los participantes, citarles para las reuniones y proporcionarles una lista con los temas que se van a tratar en las reuniones para que las puedan preparar.

El jefe del JAD debe decidir la duración y el número de sesiones a celebrar, definir el formato de la documentación sobre la que se trabajará y preparar transparencias introductorias y todo el material audiovisual que considere oportuno.

2 Etapa Celebración de las sesiones JAD: durante las sesiones, los participantes exponen sus ideas y se discuten, analizan y refinan hasta alcanzar un acuerdo. Los pasos que se recomienda seguir para este proceso son los siguientes:

- ✓ **Presentación:** se da la bienvenida a todos los participantes por parte del patrocinador ejecutivo y del jefe del JAD. El patrocinador ejecutivo expone brevemente las necesidades que han llevado al desarrollo y los beneficios que se esperan obtener. El jefe del JAD explica la mecánica de las sesiones y la planificación prevista.
- ✓ **Definir objetivos y requisitos:** el jefe del JAD promueve la discusión para elicitar los objetivos o requisitos de alto nivel mediante preguntas como: "¿Por qué se construye el sistema?", "¿Qué beneficios se esperan del nuevo sistema?", "¿Cómo puede beneficiar a la organización en el futuro?", "¿Qué restricciones de recursos disponibles, normas o leyes afectan al proyecto?", "¿Es importante la seguridad de los datos?". A medida que se van elicitando requisitos, el analista los escribe en transparencias o en algún otro medio que permita que permanezcan visibles durante la discusión.
- ✓ **Delimitar el ámbito del sistema:** una vez obtenido un número importante de requisitos, es necesario organizarlos y llegar a un acuerdo sobre el ámbito del nuevo sistema. En el caso de los sistemas de información, es útil identificar a los usuarios potenciales (actores) y determinar qué tareas les ayudará a realizar (casos de uso).
- ✓ **Documentar temas abiertos:** aquellas cuestiones que hayan surgido durante la sesión que no se han podido resolver, deben documentarse para las siguientes sesiones y ser asignadas a una persona responsable de su solución para una fecha determinada.
- ✓ **Concluir la sesión:** el jefe del JAD concluye la sesión revisando con los demás participantes la información elicitada y las decisiones tomadas. Se da la oportunidad a todos los participantes de expresar cualquier consideración adicional, fomentando por parte del jefe del JAD el sentimiento de propiedad y compromiso de todos los participantes sobre los requisitos elicitados.

3 Etapa Conclusión: una vez terminadas las sesiones es necesario transformar las transparencias, notas y demás documentación generada en documentos formales. Se distinguen tres pasos:

- ✓ **Completar la documentación:** los analistas recopilan la documentación generada durante las sesiones en documentos conformes a las normas o estándares vigentes en la organización para la que se desarrolla el proyecto.

- ✓ Revisar la documentación generada: se envía a todos los participantes para que la comenten. Si los comentarios son lo suficientemente importantes, se convoca otra reunión para discutirlos.
- ✓ Validar la documentación: una vez revisados todos los comentarios: el jefe del JAD envía el documento al patrocinador ejecutivo para su aprobación. Una vez aprobado el documento se envían copias definitivas a cada uno de los participantes.

3.1.3 Tormenta de Ideas

La tormenta de ideas es una técnica muy utilizada cuando se necesita evitar conflictos de requisitos, y puede ser utilizada como una técnica dentro de JAD.

- Tormenta de ideas

Esta técnica es más utilizada cuando se necesitan ideas originales acerca del tema de elicitación. Brinda un mayor conocimiento de lo que se quiere elicitar. Se reúnen un grupo de personas, que pueden ser los socios potenciales, clientes o personas que tengan que ver con el proyecto. Se les informa el propósito de la reunión, que en este caso consiste en aportar ideas para una mejor elicitación de los requisitos. Hay que evitar que participen personas autoritarias o que puedan inhibir a los participantes. El éxito de esta técnica depende principalmente de la libertad para expresar las ideas.

Los grupos generalmente no deben superar las seis personas. Cuanto más heterogéneo (en edades, sexo, actividad, etc.) mejor.

Reglas que deben darse a conocer:

- Se prohíbe toda crítica.
- Toda idea es bienvenida.
- Cuanto más fantástica, original o curiosa mejor. Muchas ideas que provocaron cambios profundos en cuanto a innovaciones y tecnologías provienen de estas técnicas. Hay que dejar de lado el pensamiento "racional" y utilizar el pensamiento "lateral" o desenvuelto.
- Todos deben aportar la mayor cantidad de ideas.
- Toda idea es del grupo y no de la persona que la exprese.
- Hay que establecer un límite de tiempo.

Se designa una persona para escribir las ideas que surjan. Quién actúa como moderador puede utilizar alguna lista de nombres que sirvan de "disparadores" para que los participantes comiencen a soltar ideas.

Por ejemplo, requisitos, elicitación, proyecto, ideas etc....

Cada participante se toma unos diez minutos para anotar en una ficha o tarjeta las ideas que se les ocurren. Luego se expresan y las ideas se anotan tratando de clasificarlas por rubro para un mejor ordenamiento.

¿Cómo elegir la mejor?

Decidir es optar por una alternativa para dejar de lado las demás. Aquí se corre el riesgo de descartar ideas que pueden ser muy bien recibidas por el mercado o elegir aquella que no tenga éxito.

En la tormenta de ideas se distinguen las siguientes etapas:

1 Etapa Preparación: se selecciona a los participantes y al jefe de la sesión, se prepara la sala donde se llevará a cabo la sesión. Los participantes en una sesión de la tormenta de ideas para la elicitación de requisitos son normalmente clientes, usuarios, ingenieros de requisitos, desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.

2 Etapa Generación: el jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de una idea principal se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema, bien por un orden establecido por el jefe de la sesión o aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea suficiente para pasar a la siguiente etapa. Durante esta etapa se deben observar las siguientes reglas:

1. Se prohíbe la crítica de ideas, de forma que los participantes se sientan libres de formular cualquier idea.
2. Se fomentan las ideas más avanzadas, que aunque no sean factibles, estimulan a los demás participantes a explorar nuevas soluciones más creativas.
3. Se debe generar un gran número de ideas, ya que cuantas más ideas se presenten más probable será que se generen mejores ideas.
4. Se debe alentar a los participantes a combinar o completar las ideas de otros participantes. Para ello, es necesario, al igual que en la técnica del JAD, que todas las ideas generadas estén visibles para todos los participantes en todo momento.

3 Etapa Consolidación: en esta fase se deben organizar y evaluar las ideas generadas durante la fase anterior. Se suelen seguir tres pasos:

1. Revisar ideas: se revisan las ideas generadas para clarificarlas. En caso de que haya ideas similares se unen en un solo enunciado
2. Descartar ideas: aquellas ideas que los participantes consideren excesivamente avanzadas se descartan.
3. Priorizar ideas: se priorizan las ideas restantes, identificando las absolutamente esenciales, las que estarían bien pero que no son esenciales y las que podrían ser apropiadas para una próxima versión del sistema a desarrollar.
4. Documentación: después de la sesión, el jefe produce la documentación oportuna con las ideas priorizadas y comentarios generados durante la consolidación.

- Preguntas a tener en cuenta para una tormenta de ideas

Durante la preparación de la tormenta de ideas no se puede olvidar el verdadero objetivo por el que se realiza la reunión, para ello es de vital importancia apoyarse de diversas preguntas, aquí se muestran algunas de las que pueden tener en cuenta:

¿Por qué se desarrolla el sistema?

¿Cuáles son los objetivos a cumplir?

¿Por qué se desarrolla el sistema?

¿Quiénes serán los usuarios del sistema?

¿Qué funcionalidades serán claves para satisfacer a los usuarios?

¿Qué estilo de interfaz gráfico debe seguirse?

¿Cómo recibirán asistencia?

3.1.4 Casos de Uso.

Los casos de uso son una técnica para la especificación de requisitos funcionales. Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables. Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo. Los casos de uso permiten hacer una mejor elicitación de requisitos y son de fácil

comprensión para los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios.

3.1.5 Prototipado de interfaz de usuario.

Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software.

Es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar su uso, probarlo en situaciones reales y explorar su uso.

Los prototipos se realizan con medios muy reducidos, pueden ser sitios Web rápidamente esbozados y de poca funcionalidad o incluso realizados con recortes de papel o incluso un complejo software, sin embargo todos ellos pueden llegar a ser muy útiles.

Una manera rápida de prototipos es crear imágenes en cualquier programa de retoque fotográfico o incluso PowerPoint y en caso necesario darles funcionalidad mediante un editor html. Sin embargo lo mejor para centrarse en las estructuras de la información y olvidarse de los aspectos estéticos es utilizar cualquier sencilla herramienta de dibujo para crear simples esqueletos.

En caso necesario los prototipos, permiten llegar a un nivel muy detallado en las especificaciones de diseño del futuro sitio y concretar sus elementos. De esta manera el trabajo posterior de los programadores es mucho más sencillo y la planificación temporal del proyecto puede cumplirse más fácilmente.

¿Por qué un prototipo?

Porque son útiles para comunicar, discutir y definir las ideas entre los diseñadores y las partes responsables.

Los prototipos responden a preguntas y apoyan el trabajo de los diseñadores probando ideas, clarificando requisitos o definiendo alternativas.

Permiten al desarrollador del software tener una idea de lo que va a crear y darle una visión al usuario del mismo puesto que son útiles para comunicar, discutir y definir las ideas entre ambas partes. O sea, se

lleva a cabo un diseño rápido centrándose en la representación de aquellos aspectos del software que serán visibles para el cliente o usuario final a través de la construcción de prototipos, los mismos son evaluados por los clientes o usuarios para una retroalimentación, lo cual permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Una vez realizados varios prototipos éstos se comparan para escoger el más usable. El prototipo final servirá como un mecanismo para la definición de requisitos. Su principal objetivo es la captura de modo declarativo e independiente de todos los requisitos necesarios para la interacción entre el usuario y la aplicación.

3.1.6 Método de negociación de conflictos Win Win.

Este modelo, no es más que la elicitación y negociación mutuamente satisfactoria y compartida de factores críticos de éxito para cada uno de los participantes de un proyecto desarrollo de software. Los participantes son clientes, analistas, usuarios, programadores, arquitectos, etc. Los cuales son importantes en la negociación para entender y resolver conflictos de requerimientos mediante la elicitación y negociación de artefactos en función de las condiciones de ganancia, para entender las restricciones, la transferencia del conocimiento, el aprendizaje cooperativo, todo esto visto como un proceso iterativo.

El modelo utiliza la teoría “W”, que pretende que todo involucrado salga ganador. De esta forma, asiste a los involucrados en el desarrollo a identificar y negociar distintos aspectos, reconciliando conflictos entre las opciones de ganancias para todos.

Teoría W:

- Establezca un set de precondiciones Win – Win
 - Entender lo que la gente quiere
 - Aterrizar las expectativas
 - Compara las tareas de las personas con sus condiciones ganadoras
 - Proveer un ambiente colaborativo.
- Estructurar un procesos de software Win Win
 - Establecer un proceso realista.
 - Tener un plan para controlar el proyecto.
 - Identificar o manejar los riesgos del Win – lose o lose-lose.

- Involucrar a las personas.
- Estructurar Win Win para el producto (software)
 - El producto debe corresponder con las condiciones ganadoras.

El Win Win cuenta con varios ciclos que se describirán a continuación:

Ciclo 0:

- Se experimenta inicialmente con la enseñanza del Modelo Espiral Win Win en el curso de Ingeniería de Software.
- Se exploran condiciones y se determina la viabilidad del sistema.
- Se identifican actores.
- Se identifica y se resuelve el principal riesgo.

Ciclo1:

- Guías para el desarrollo de los documentos LCO (Life Cycle Objectives) y LCA (Life Cycle Architecture)
- Se asegura un proceso común de desarrollo mediante prototipos.
- Se identifican los participantes, roles y sus conceptos sobre los límites del sistema (Win Win conditions)

Ciclo2:

- Se elabora la arquitectura para cada aplicación.
- Prototipos vs. Negociación Win Win.
- Métodos orientados a objetos, diagramas de clases, flujogramas, diagramas de transición, trazabilidad.

Ciclo3

- Chequear lista de riesgos

La teoría Win Win hace posible “pensar en grande” acerca de los proyectos. El proceso de negociación permite tener una visión rápida a los desarrolladores, clientes y usuarios del producto final.

3.1.7 Táctica del embudo

Durante la etapa de DETECCIÓN DE NECESIDADES, se utilizará la "Táctica del embudo" (ver figura 3.1) con el fin de dirigir y llevar la iniciativa de la entrevista.

Objetivos

- Detectar necesidades reales del cliente.
- Orientar la conversación hacia sus intereses.
- Facilitar la obtención de información sobre el cliente.

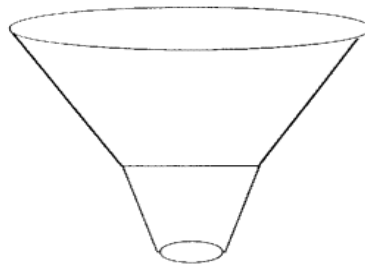


Figura 3.1 Táctica del embudo

Preguntas de carácter general. El cliente se acostumbra a responder

Preguntas relacionadas con el negocio, más concretas.

Preguntas de cierre. En definitiva, se trata de no "ir al grano" directamente, sino de una manera progresiva, venciendo primero las barreras de comunicación.

3.1.8 Punto de Función por Caso de Uso.

Técnica que permite realizar la estimación de costos y el esfuerzo del proyecto.

Existe una relación natural entre los puntos de función y los casos de uso. Los puntos de función permiten estimar el tamaño del software a partir de sus requerimientos, mientras que los casos de uso permiten documentar los requerimientos del software. Ambos tratan de ser independientes de las tecnologías utilizadas para la implementación. En etapas tempranas del ciclo de vida, se identifican los actores y los casos de uso del sistema, y se documenta cada uno de ellos mediante una breve descripción. Aplicando

el análisis de puntos de función a estos casos de uso, se podrá obtener una estimación inicial del tamaño y a partir de ella del esfuerzo. Esta estimación es bastante imprecisa debido principalmente a la escasa información que se tiene sobre el software al principio de un proyecto, pero permitirá obtener una idea del esfuerzo necesario para llevar adelante el mismo, y podrá ser refinada a medida que se obtenga más información.

Posteriormente se amplía la documentación de cada caso de uso, describiendo los escenarios que se producen dentro del mismo. Un escenario relata la secuencia de pasos que efectúan los actores y el sistema durante la ejecución del caso de uso. Si se aplica nuevamente el análisis de puntos de función sobre estos casos de uso detallados, la estimación del tamaño y esfuerzo será más precisa que la anterior.

Con el conocimiento de las técnicas presentadas y la combinación de estas con actividades encaminadas a guiar el proceso de elicitar, analizar y negociar y especificar, se obtendrá un producto con la calidad esperada por el cliente y que cumpla con todas sus necesidades, por lo que a continuación se proponen una serie de actividades que servirán de base para este proceso de vital importancia.

3.2 Actividades recomendadas en el procedimiento propuesto para obtener los productos descritos.

Para la elaboración de este procedimiento se tomó como punto de partida el proceso de ingeniería de requisitos que Pressman definió, donde se centraron en los tres primeros pasos, elicitación, análisis y negociación, y especificación de los requisitos, en la que se desarrollan las siguientes actividades:

Actividad 1: Obtener información sobre el dominio del problema y el sistema actual.

Actividad 2: Preparar y realizar las reuniones de elicitación/negociación.

Actividad 3: Identificar/revisar los objetivos del sistema.

Actividad 4: Identificar los requisitos.

Actividad 5: Revisar los requisitos.

Actividad 6: Eliminar los conflictos entre los requisitos.

Actividad 7: Estimar el costo y el esfuerzo de cada uno de requisitos funcionales

Actividad 8: Asignar Prioridades a los requisitos funcionales.

Para una mejor realización de estas actividades es recomendable que se hagan en ese orden, para lograr un mayor resultado en el proceso de elicitación, análisis y negociación, y especificación. Ver figura 3.2

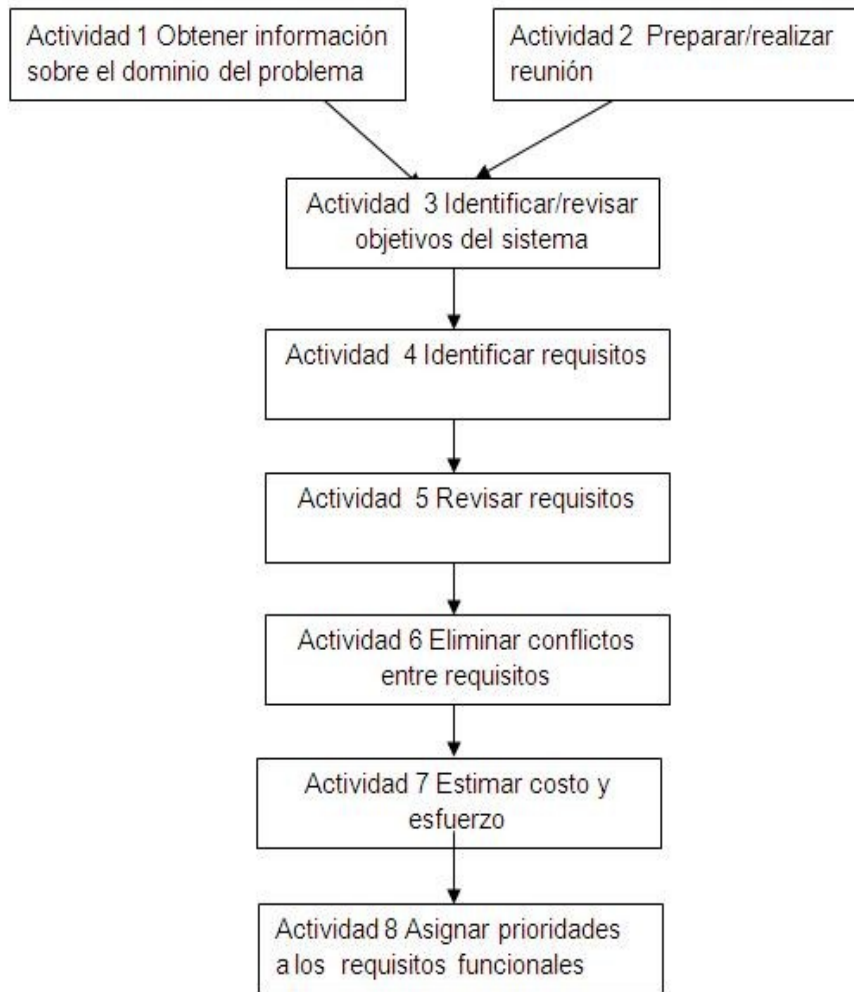


Figura 3.2 Mapa del Procedimiento de elicitación, análisis y negociación, y elicitación de requisitos.

3.2.1 Actividad 1: Obtener información sobre el dominio del problema y el sistema actual.

-Objetivos

- ✓ Conocer el dominio del problema.
- ✓ Conocer la situación actual.
- ✓ Identificar clientes y usuarios para la obtención de información.

-Responsable

Analista del sistema, Jefe de proyecto, clientes.

- Inicio.

A esta actividad se le da comienzo cuando se inicia un nuevo proyecto y comienzan a realizarse las primeras reuniones con los clientes y usuarios.

-Final.

Se culmina esta tarea cuando el equipo de desarrollo ha obtenido el conocimiento necesario para entender el dominio del problema y se han identificado a los clientes y usuarios que brindarán toda la información necesaria para la ejecución del proyecto.

- Sub_Actividades.

- ✓ Realizar las primeras reuniones con los clientes.
- ✓ Identificar el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual para con ello poder planificar las reuniones de trabajo así como el equipo de desarrollo que ejecutará el proyecto.

-Artefactos de entrada.

- ✓ No posee ningún producto de entrada. Se recomienda realizar un estudio de vigilancia tecnológica en iguales dominios para obtener información adicional.

-Artefactos de salida.

Las posibles soluciones y la oportunidad de negocio. Un modelo de dominio inicial. Y el Documento de Especificación de Requisitos con los epígrafes:

- ✓ Descripción del sistema actual, Ver figura 3.3.
- ✓ Descripción técnica del entorno del sistema.


Nombre del Proyecto:	
Descripción del Sistema Actual	

Figura 3.3 Descripción del Sistema Actual

-Técnicas recomendadas

- ✓ Obtener información de fuentes externas al negocio del cliente: folletos, informes sobre el sector, publicaciones, consultas con expertos, etc.

En el caso de que se trate de un dominio muy específico puede ser necesario recurrir a fuentes internas al propio negocio del cliente, en cuyo caso pueden utilizarse las técnicas auxiliares de elicitación de requisitos, como el estudio de la documentación, observación in si, entre otras.

- ✓ Modelado del negocio o dominio inicial.

3.2.2 Actividad 2: Preparar y realizar reuniones de elicitación/negociación.

-Objetivos

- ✓ Identificar a los usuarios participantes.
- ✓ Conocer las necesidades de clientes y usuarios.

-Responsable

Jefe de proyecto, analista del sistema, clientes.

- Inicio.

Se comienza esta actividad realizando nuevas reuniones con los clientes para determinar sus necesidades y los posibles conflictos detectados en iteraciones anteriores.

-Final.

Esta actividad es terminada cuando el equipo de desarrollo conoce y ha eliminado los posibles conflictos que existan en el problema determinado o en las posibles soluciones que se hayan detectado. El equipo

de desarrollo ya conoce los detalles específicos de la organización para la que se va a desarrollar el sistema y, por otra parte, los clientes y posibles usuarios tienen el conocimiento de lo que necesita saber el equipo de desarrollo para llevar a cabo su labor.

-Sub_actividades.

- ✓ Identificar los posibles conflictos que se hayan detectado entre el problema y las posibles soluciones.
- ✓ Preparar las reuniones con los clientes.

-Artefactos de entrada.

El Documento de Especificación de Requisitos y notas tomadas durante las reuniones, transcripciones o actas de reuniones, formularios, grabaciones en cinta o video de las reuniones o cualquier otra documentación que se considere oportuna.

-Artefactos de salida

Del Documento de Especificación de Requisitos se realizarán los epígrafes:

- ✓ Participantes en el proyecto. Ver figura 3.4.
- ✓ Listado de características y necesidades. Ver figura 3.5
- ✓ Listado de clientes y su descripción. Ver figura 3.6
- ✓ Definición de actores.


Nombre del Proyecto:		
Participantes en el proyecto		
Nombre del participante	Rol que desempeña	

Figura 3.4 Participantes en el proyecto


Nombre del Proyecto:		
Listado de características y necesidades		
Nombre de la característica o necesidad	Descripción de la característica o la necesidad	

Figura 3.5 Listado de características y necesidades


Nombre del Proyecto:		
Listado de los clientes		
Nombre del cliente	Descripción del cliente	

Figura 3.6 Listado de los clientes

-Técnicas recomendadas.

- Especificación de casos de uso para la elicitación de requisitos, la entrevista y tormenta de ideas.
- Técnicas de negociación como Win Win.

3.2.3 Actividad 3: Identificar/revisar los objetivos del sistema

- Objetivos

- ✓ Identificar los objetivos que se esperan alcanzar mediante el sistema software a desarrollar.
- ✓ Revisar, en el caso de que exista conflictos con los objetivos previamente identificados.

-Responsable

Analista del sistema, cliente responsable.

- Inicio.

A partir de los objetivos, requisitos o conflictos, que se hayan identificado claramente durante las sesiones de elicitación en la actividad anterior, se identifica qué objetivos se esperan alcanzar.

- Final.

Cuando ya se hayan identificado todos los objetivos del sistema.

-Sub_actividades.

- ✓ Revisar los objetivos.
- ✓ Obtener la aprobación del cliente sobre sus objetivos y alcance del sistema

- Artefactos de Entrada

Documento de Especificación de requisitos y actas de reuniones con los clientes

- Artefactos de salida.

Del documento de Especificación de Requisitos se realizarán los epígrafes:

- ✓ Objetivos del sistema. Ver figura 3.7.


Nombre del Proyecto:		
Objetivos del sistema		
Nombre del objetivo	Descripción objetivo	

Figura 3.7 Objetivos del sistema.

-Técnicas recomendadas

- ✓ Plantilla para especificar los objetivos del sistema.

3.2.4 Actividad 4: Identificar los requisitos

- Objetivos

- ✓ Identificar los requisitos que debe cumplir el sistema software a desarrollar.
- ✓ Revisar, en el caso de que existan conflictos, en los requisitos previamente elicitados.

- Inicio.

Se inicia teniendo en cuenta los objetivos identificados en la actividad de: "Identificar/revisar los objetivos del sistema", y los demás requisitos.

-Responsable

Analista del sistema.

- Final.

Obtención de todos los datos relevantes, conceptos generales que el cliente debe almacenar en el sistema.

-Sub_actividades.

- ✓ Planificar reuniones con el cliente
- ✓ Preparar documentación para la reunión con el cliente
- ✓ Distribuir, si es necesario la información a los participantes de la reunión.
- ✓ Realizar Modelo de Dominio o Negocio.

- Artefactos de entrada.

Documento de Especificación de Requisitos y actas de reuniones con los clientes.

- Artefactos de salida.

- ✓ Especificación de casos de uso.

-Técnicas recomendadas

- ✓ Casos de uso.

-Productos entregables

Terminación del Documento de Especificación de Requisitos. Con el glosario de términos y la especificación de los casos de uso.

3.2.5 Actividad 5: Revisar los requisitos.

-Objetivos

- ✓ Analizar los requisitos identificados para revisar que no sean ambiguos, estén consistentes y completos. Para ello se deberá analizar requisito por requisito (comparándolos entre ellos). Cualquier conflicto entre ellos que sea encontrado deberá negociarse con el cliente.

-Responsable

Analista del sistema, Jefe de proyecto, clientes.

-Inicio.

Esta actividad se inicia cuando se han especificados diferentes casos de uso que permite realizar un análisis de la especificación para valorar su consistencias, no ambigüedad y completitud.

-Final.

Cuando se verifica toda la especificación y se aprueba por ambas partes.

-Sub_Actividades.

- ✓ Revisar la consistencia de los requisitos o sea que estos no sean contradictorios ni entren en conflicto lo cual se puede analizar de tres formas:
 - Cerciorarse de que no se describa el mismo objeto real utilizando distintos términos.
 - Asegurarse también que no existan requisitos con definiciones contradictorias.
 - Eliminar si existen los conflictos lógicos temporales entre requisitos.
- ✓ Revisar la ambigüedad de los requisitos, teniendo e cuenta que un requisito es no ambiguo si:
 - Tiene una única interpretación.

- Es descrito utilizando términos únicos.
- ✓ Revisar la completitud de los requisitos en el Documento de Especificación de Requisitos, o sea que todos los requisitos del software estén recogidos en dicho documento.
- ✓ Revisar que los requisitos especificados no tengan un nivel de detalle técnico inapropiado e esta etapa.
- ✓ Revisar que existe para cada requisito un origen y registrarlo en el Documento de Especificación de requisitos.
- ✓ Revisar que cada requisito se puede implementar en el entorno técnico propuesto.
- ✓ Revisar que cada requisito puede ser probado.
- ✓ Registrar los conflictos detectados para una posterior negociación con el cliente.

-Artefactos de entrada.

- Documento de Especificación de Requisitos.
- Especificación de casos de uso.

-Artefactos de salida.

- ✓ Informe de requisitos en conflicto

Del Documento de Especificación de requisitos el epígrafe:

- ✓ Origen y razón de cada requisito. Ver figura 3.8


Nombre del Proyecto:		
ID y nombre del requisito.	Origen y razón de cada requisito	Solicitante

Figura 3.8 Origen y razón de cada requisito

-Técnicas recomendadas

- ✓ Prototipado.

3.2.6 Actividad 6: Eliminar los conflictos entre requisitos.

-Objetivos

- ✓ Negociar los posibles conflictos detectados para encontrarles una solución donde queden ambas partes de acuerdo.

-Responsable

Jefe de proyecto, analista del sistema, clientes.

-Inicio.

Se comienza esta actividad realizando nuevas reuniones con los clientes para darle solución a los conflictos detectados en la revisión.

-Final.

Esta actividad es terminada cuando el equipo de desarrollo ha negociado las soluciones con el cliente y se llega a un acuerdo común.

-Sub_actividades.

- ✓ Preparar las reuniones con los clientes.
- ✓ Negociar los requisitos que no pueden ser implementados, que son contradictorios o que no están dentro del alcance del sistema.
- ✓ Elaborar materiales de ayuda que le brinden al cliente la información necesaria para entender los conflictos. (Ejemplo: Elaborar prototipos con los requisitos en conflictos).

-Artefactos de entrada.

- ✓ Informe de requisitos en conflicto.
- ✓ Documento Especificación de Requisitos.

- ✓ Especificación de casos de uso

-Artefactos de salida

- ✓ Documentación de ayuda para el cliente.
- ✓ Prototipo.
- ✓ Modelos que ayuden a la comunicación
- ✓ Agenda de la reunión

-Técnicas recomendadas.

- ✓ Reuniones con el cliente.
- ✓ Tormenta de ideas.
- ✓ Técnicas de negociación como Win Win
- ✓ Táctica del embudo.
- ✓ Prototipado.

3.2.7 Actividad 7: Estimar el costo y el esfuerzo de cada uno de requisitos funcionales.

-Objetivos

- ✓ Valorar los costos y el esfuerzo para elaborar el calendario del proyecto.

-Responsable

Analista del sistema.

- Inicio.

A partir de los requisitos especificados en el Documento de Especificación de Requisitos se decide estimar el costo y el esfuerzo de los mismos.

- Final.

Cuando ya se haya calculado el costo y el esfuerzo de cada requisito especificado.

-Sub_actividades.

- ✓ Calcular el costo de cada requisito.
- ✓ Calcular el esfuerzo de cada requisito.
- ✓ Registrar los resultados en el Documento de Especificación de Requisitos.

-Artefactos de Entrada

Documento de Especificación de Requisitos.

Especificación de los casos de uso

- Artefactos de salida.

- ✓ Especificación de casos de uso.
- ✓ Esfuerzo del caso de uso

Del documento de Especificación de Requisitos el epígrafe:

- ✓ Costos y esfuerzo del proyecto.

-Técnicas recomendadas

- ✓ Punto de Función por Caso de Uso.

3.2.8 Actividad 8: Asignar Prioridades a los requisitos funcionales.

- Objetivos

- ✓ Asignarle a cada requisito identificado la prioridad que el mismo posee desde la perspectiva del cliente.
- ✓ Es importante conocer y documentar lo que el cliente considera más importante pues con esto se puede determinar lo que realmente necesita y brindarle esa funcionalidad desde la primera iteración.

-Inicio.

La tarea se inicia cuando el analista decide darle orden por prioridad a los diferentes requisitos funcionales.

-Responsable

Analista del sistema, Jefe de proyecto, clientes.

- Final.

Finaliza cuando el analista ha determinado la prioridad de los requisitos.

-Sub_actividades.

- ✓ Organizar los diferentes requisitos funcionales en el Documento de Especificación de Requisitos por grado de requisito de mayor a menor.

- Artefactos de entrada.

Documento de Especificación de Requisitos.

- Artefactos de salida.

Documento de Especificación de Requisitos con el epígrafe de Características del Software donde se registren todos los requisitos funcionales ordenados por prioridad.

-Productos entregables

Documento de Especificación de Requisitos

3.3 Documento de Especificación de requisitos.

El Documento de Especificación de Requisitos va a tener la siguiente estructura:

- Nombre del Proyecto.
- Revisión.
- Versión.
- Índice.
- Introducción.
- Descripción del sistema actual.
- Descripción técnica del entorno del sistema.
- Participantes en el proyecto.
- Objetivos del sistema.

- Listado de características y necesidades.
- Lista de clientes con su descripción.
- Lista de las personas involucradas y sus responsabilidades.
- Relación de requisitos funcionales agrupados por funcionalidad.
- Relación de requisitos no funcionales.
- Requisitos de almacenamiento.
- Restricciones del dominio y Reglas del Negocio.
- Requisitos de interfaz, documentación, legales, seguridad.
- Origen y razón de cada requisito.
- Costos del proyecto.
- Características del software.

En las siguientes secciones se describe con detalle cada sección del Documento de especificación de requisitos. Ver Anexo 2 Documento de Especificación de requisitos.

3.3.1 Portada

La portada del Documento de Especificación de Requisitos debe tener el formato que aparece en el Anexo 2.

Los elementos que deben aparecer son los siguientes:

- Nombre del proyecto: el nombre del proyecto al que pertenece el Documento de Especificación de Requisitos
- Revisión: Es el número de veces que el documento ha sido revisado.
- Versión: la versión del Documento de Especificación de Requisitos que se entrega al cliente. La versión se compone de dos números X e Y. El primero indica la versión, y se debe incrementar cada vez que se hace una nueva entrega formal al cliente. Cuando se incrementa el primer número, el segundo debe volver a comenzar en cero. El segundo número indica cambios dentro de la misma versión aún no entregada, y se debe incrementar cada vez que se publica una versión con cambios respecto a la última que se publicó y que no se vaya a entregar formalmente todavía. Este tipo de versiones pueden ser internas al equipo de desarrollo o ser entregadas al cliente como una forma de orientación.

- Portada: En la portada aparecerá la fecha de publicación de la versión, el nombre de la empresa o equipo de desarrollo y nombre del cliente o empresa.

3.3.2 Índice

El índice del Documento de Especificación de Requisitos debe indicar la página en la que comienza cada sección, subsección o apartado del documento. En la medida de lo posible, se sangrarán las entradas del índice para ayudar a comprender la estructura del documento.

3.3.3 Introducción

Esta sección debe contener una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se acomete, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

3.3.4 Descripción del sistema actual

Esta parte debe contener una descripción del sistema actual en el caso de que se haya acometido su estudio. Esta se desarrolla en la actividad de “Obtener información sobre el dominio del problema y el sistema actual”.

3.3.5 Descripción técnica del entorno del sistema

Debe tener una detallada descripción técnica de todo el sistema con todos los datos de interés, conceptos etc. Este epígrafe se realiza en la actividad de “Obtener información sobre el dominio del problema y el sistema actual”.

3.3.6 Participantes en el proyecto

Esta sección debe contener una lista con todos los participantes en el proyecto, tanto desarrolladores como clientes y usuarios. Para cada participante se debe indicar su nombre, el papel que desempeña en el proyecto, la organización a la que pertenece y cualquier otra información adicional que se considere oportuna. Este epígrafe se realiza en la actividad de “Identificar participantes y usuarios del proyecto”.

3.3.7 Listado de características y necesidades

Se realizará un listado que tendrá las características que el sistema deberá tener y las necesidades que surjan. Esta parte del Documento de Especificación de Requisitos se realiza en la actividad de “Identificar participantes y usuarios del proyecto”.

3.3.8 Listado de clientes y su descripción

Esta sección tendrá un listado de todos los clientes que tengan que ver con el producto y una descripción de cada uno de ellos. Y se realiza en la actividad de “Identificar participantes y usuarios del proyecto”.

3.3.9 Definición de los actores

Este apartado debe contener una lista con los actores que se hayan identificado, especificados mediante la plantilla para actores de casos de uso. Se realiza en la actividad de “Identificar participantes y usuarios del proyecto”

3.3.10 Objetivos del sistema

En esta parte debe estar contenida una lista de los objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación, especificados mediante la plantilla para objetivos. Se realiza en la actividad de “Identificar y revisar los objetivos del sistema”.

3.3.11 Catálogo de requisitos del sistema

En esta sección se van a describir los requisitos del sistema, cada uno de sus grandes grupos de requisitos, no funcionales y funcionales, podrán dividirse para ayudar a la legibilidad del documento. El desarrollo de este epígrafe del documento se realiza en la actividad de Identificar requisitos.

- Requisitos funcionales agrupados por funcionalidad

Va a estar contenido dentro del catálogo de requisitos como un sub_epígrafe de este, contiene una plantilla que agrupa los requisitos funcionales agrupados por su funcionalidad, lo que apoya más en la organización del documento de especificación de requisitos.

- Requisitos no funcionales

Es otro sub_epígrafe, se agruparán todos los requisitos no funcionales y su clasificación, de hardware, software, seguridad, soporte del sistema que se hayan identificado, especificados mediante la plantilla para requisitos no funcionales.

3.3.12 Restricciones del dominio y Reglas del negocio

Va a contener todas las restricciones que tenga el dominio, agrupadas en una lista y todas las reglas que el negocio debe cumplir para su mejor funcionamiento. Se realiza en la actividad de “Identificar requisitos”.

3.3.13 Origen y razón de cada requisito.

Esta sección deberá contener el listado de todo los requisitos especificados en el Catálogo de requisitos con su ID y nombre y al lado la razón de ser y lo que le dio origen a cada uno de ellos, y el nombre o nombres de las personas que lo solicitan. Se realiza en la actividad de “Revisar los requisitos”.

3.3.14 Costos del proyecto

En esta sección se registrarán los costos del software en el sistema y en el calendario de acuerdo con los resultados obtenidos por las técnicas recomendadas o por cualquier otra técnica que se haya querido utilizar. Se realiza en la actividad de “Estimar el costo y el esfuerzo de cada uno de requisitos funcionales”.

3.3.15 Características del software

En esta sección se registrará un listado de los requisitos funcionales identificados organizados por prioridad de mayor a menor. Se realiza en la actividad de “Asignar prioridades a los requisitos funcionales”.

3.3.16 Glosario de términos

Esta sección, deberá contener una lista ordenada alfabéticamente de los términos específicos del dominio del problema, acrónimos y abreviaturas que aparezcan en el documento y que se considere que su significado deba ser aclarado. Cada término deberá acompañarse de su significado. Se realiza después que se termina el desarrollo de todas las actividades programadas.

3.4 Descripción técnica del entorno del sistema.

En este documento se realiza una detallada descripción técnica del sistema que se va a implementar para tener una mejor visión del mismo. Se realiza en la actividad de “Identificar el dominio del problema y el sistema actual”.

3.5 Definición de actores.

En este documento deben quedar recogidas detalladamente las descripciones de cada uno de los actores que interactúen con el software. Se realiza en la actividad “Identificar el dominio del problema y el sistema actual”.

3.6 Especificación de casos de uso.

Este artefacto de salida se obtiene en la actividad de “Identificar los requisitos”, para el mismo utilizaremos la plantilla propuesta por RUP. Ver plantilla propuesta en Anexo 3.

3.7 Informe de requisitos en conflicto.

Este informe se debe realizar en la actividad de “Revisar requisitos”, en el mismo deben quedar registrados todos los requisitos en los que haya conflictos para una posterior negociación con el cliente

3.8 Documentación de ayuda para el cliente.

La misma se refiere a toda aquella documentación que prepare el analista para la negociación de los conflictos encontrados en los requisitos en la actividad de “Revisar requisitos”, se debe preparar en correspondencia al tipo de persona con el que se vaya a negociar puesto que algunas personas necesitan de más materiales de apoyo que otras dependiendo del coeficiente intelectual que tengan y de la preparación o desenvolvimiento. Se realiza en la actividad de “Eliminar los conflictos entre requisitos”.

3.9 Prototipo.

Los mismos se utilizan para darle ideas al cliente de lo que este quiere y de esta forma poder intercambiar con él criterios, ideas, poder llegar a algún acuerdo en cuanto a algún conflicto entre requisitos, etc. Se utilizan en la actividad de “Eliminar los conflictos entre requisitos”.

3.10 Modelos que ayuden a la comunicación

El analista debe realizar varios modelos pues estos ayudan a ver el problema desde diferentes puntos de vista, y de esta manera poder llegar a un acuerdo del conflicto detectado. Se utilizan en la actividad de “Eliminar los conflictos entre requisitos”.

3.11 Agenda de la reunión

El analista debe llevar una agenda de reunión donde debe preparar la entrevista con el cliente en vista a tener una mejor preparación a la hora de negociar con el mismo. Esto se realiza en la actividad de “Eliminar los conflictos entre requisitos”.

3.12 Esfuerzo del caso de uso

Este documento debe registrar el esfuerzo de cada caso de uso. El mismo se realiza en la actividad de “Estimar el costo y el esfuerzo de cada uno de requisitos funcionales”.

En este capítulo se describió el procedimiento propuesto para la elicitación, análisis y negociación, y especificación de requisitos, así como las diferentes técnicas que pueden utilizarse o combinarse para la realización de este. Se explica el proceso de realización de los artefactos: Documento de Especificación de Requisitos. Además se menciona y explican brevemente otros artefactos de salida que se entregan al cliente y no se registran dentro del documento antes mencionado.

CONCLUSIONES

A partir de toda la investigación realizada sobre la ingeniería de requisitos, profundizando más en la elicitación, análisis y negociación, y especificación de los mismos se arribó a la conclusión de que la carencia de un procedimiento para la realización de la captura y análisis de requisitos es uno de los grandes problemas que afectan la obtención de software con una mayor calidad.

Por esta razón se estudiaron los diferentes conceptos, modelos de procesos, metodologías y técnicas de la ingeniería de requisitos para profundizar en estos temas. Finalmente se propone el diseño de un procedimiento para llevar a cabo este proceso en los proyectos productivos de la Facultad 7.

Al finalizar el presente trabajo de diploma se han cumplido los objetivos planteados, obteniendo el diseño de un procedimiento en el que se aplican los resultados de la investigación llevada a cabo, y que favorecerá el incremento de la calidad en los proyectos productivos la Facultad 7 de la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Una vez vencidos los objetivos de este trabajo, y teniendo en cuenta las experiencias obtenidas durante la realización del mismo, se recomienda:

- ✓ Evaluar el procedimiento propuesto con la finalidad, de tener resultados prácticos del mismo.
- ✓ Utilizar el procedimiento propuesto para la elicitación, análisis y negociación, y especificación de los requisitos en los diferentes proyectos productivos de la Facultad 7 y hacerlo extensivo al resto de la universidad en los casos que sea posible.
- ✓ Utilizar los conceptos, modelos de procesos, metodologías y técnicas abordadas en este trabajo para lograr que los grupos de desarrollo tengan conocimiento acerca del tratamiento que se le da a la ingeniería de requisitos y puedan hacer uso de los mismos.
- ✓ Integrar este procedimiento a los que ya existen para el aseguramiento de la calidad del software.

REFERENCIAS BIBLIOGRÁFICAS

- [1] F. P. Brooks, Jr. The Mythical Man–Month: Essays on Software Engineering Anniversary Edition. Addison–Wesley, 1995.
- [2] R. S. Pressman. Ingeniería del Software. Un Enfoque Práctico. McGraw–Hill, 4a edición, 1997.
- [3] Idem 2
- [4] A. M. Davis. Software Requirements: Objects, Functions and States. Prentice–Hall, 2a edición, 1993.
- [5] Idem 2
- [6] Idem 2
- [7] S. Raghavan, G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI–94–EM– 10, Software Engineering Institute, Carnegie Mellon University, 1994. Disponible en <http://www.sei.cmu.edu>.
- [8] J. A. Goguen. Requirements Engineering as the Reconciliation of Social and Technical Issues. En Requirements Engineering: Social and Technical Issues, páginas 165–199. Academic Press, 1994. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [9] L. A. Macaulay. Seven–Layer Model of the Role of the Facilitator in Requirements Engineering. Requirements Engineering Journal, 4(1), 1999.
- [10] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [11] J. A. Goguen. Requirements Engineering as the Reconciliation of Social and Technical Issues. En Requirements Engineering: Social and Technical Issues, páginas 165–199. Academic Press, 1994. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [12] F. Davis. La comunicación no verbal, volumen 616 de El Libro de Bolsillo. Alianza Editorial, 1985.
- [13] Idem 8
- [14] L. A. Macaulay. The Role of the Facilitator in Requirements Engineering. En Fifth International Conference on Requirements Engineering, 1998.
- [15] D. E. H. Damian, A. Eberlein, M. L. G. Shaw, y B. R. Gaines. Using Different Communication Media in Requirements Negotiation. IEEE Software, 17(3):28–36, Mayo/Junio 2000.

[16][Sawyer y Kontoya 1999]

[17] Idem 16

[18] IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI. Standard 830–1993, Institute of Electrical and Electronics Engineers, 1993.

[19] K. Pohl. Requirements Engineering: An Overview. Encyclopedia of Computer Science and Technology, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.

[20] Idem 19

[21] Idem 19

[22] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. Software Process – Improvement and Practice, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.

[23] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En Proceedings of the First International Conference on Requirements Engineering, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPMRequirements93.ps>.

[24] C. Potts, K. Takahashi, y A. Anton. Inquiry–Based Requirements Analysis. IEEE Software, 11(2), 1994.

[25] Idem 22

[26] Idem 12

BIBLIOGRAFÍA

- ✓ Análisis de Requerimientos. 2005 Microsoft Ibérica S.R.L. Extraído el 10 de marzo del 1997 de <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/analisis.asp>
- ✓ Análisis de Sistemas.ppt. Extraído el 25 de febrero del 1997 de <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema06.pdf>
- ✓ Análisis del Sistema de Información.ppt. Extraído el 10 de marzo del 1997 de <http://www.csi.map.es/csi/metrica3/asiproc.pdf>
- ✓ Balzer, R., and N. Goodman, Principles of Good Software Specification, Proc, on Specifications of Reliable Software, IEEE, 1979, pp.58-67.
- ✓ Bendahan M. Proceso de desarrollo de software. 31 de Enero del 2004 Extraído el 4 de abril del 1997 de <http://www.ilustrados.com/publicaciones/EpZVVyZAFuLYtfSRKp.php>
- ✓ Booch, G., Rumbaugh, J., Jacobson, I. El Lenguaje Unificado de Modelado. Addison-Wesley. 1999.
- ✓ Booch G., Rumbaugh, J. Jacobson, I. El Proceso Unificado de Desarrollo Software. Addison Wesley. 1999.
- ✓ Booch, G., Rumbaugh, J., Jacobson, I. El Proceso Unificado de Desarrollo de Software. La Habana. Cuba 2004.
- ✓ Booch G., Rumbaugh, J. Jacobson, I. Manual de Referencia. Addison Wesley. 1999.
- ✓ Dorling, A.; SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. IEEE.1998.
- ✓ Douglass, B. P. Doing Hard Time. Developing Real-time Systems with UML, Objects, Frameworks and Patterns. Addison Wesley. 1999.
- ✓ Durán A, Bernárdez B. Metodología para el Análisisde Requisitos de Sistemas Software. Sevilla, noviembre de 2000. Extraído el 2 de enero del 1997 de http://tejo.usal.es/~fgarcia/docencia/isoftware/doc/metodologia_analisis.pdf
- ✓ Durán A, Bernárdez B. Metodología para la Elicitación de Requisitos de Sistemas Software, Versión 2.3, Informe Técnico LSI–2000–10 (revisado), Universidad de Sevilla

- ✓ Durán A, Bernárdez B. Metodología para la Elicitación de Requisitos de Sistemas Software. Sevilla, abril de 2002. Extraído el 8 de junio del 1997 de http://dmi.uib.es/~dmiamp/ESI/Analisi/metodologia_elicitacion.pdf
- ✓ Elicitación de requisitos. Grupo de Ingeniería del Software. Marzo de 2006. Extraído el 5 de abril del 1997 de <http://www.lsi.us.es/docencia/get.php?id=1876>
- ✓ Freeman, P., "Requirements Analysis and Specification", Proc. Intl. Computer Technology Conf., ASME, San Francisco, August, 1980
- ✓ Fernandez A. Comportamiento del sistema. 2001-03-20. Extraído el 13 de febrero del 1997 de <http://www-gris.det.uvigo.es/~avilas/UML/node24.html>
- ✓ Identificación de necesidades y definición de requisitos. 2002. Extraído el 2 de febrero del 1997 de <http://www.gsync.inf.uc3m.es/docencia/iu/tema%20requisitos.ppt#342,86,Identificación> de necesidades y definición de requisitos.
- ✓ INCE, D. C.: Ingeniería de Software. Addison-Wesley Iberoamericana, 1993
- ✓ Ingeniería De Requerimientos. 1997 Monografias.com S.A. Extraído el 8 de junio del 1997 de <http://www.monografias.com/trabajos6/resof/resof.shtml>
- ✓ Insfrán E, Tejadillos E, Martí S, Burbano M. Transformación de Especificación de Requisitos en Esquemas Conceptuales usando Diagramas de Interacción.pdf. Extraído el 1 de marzo del 1997 de <http://www.dsic.upv.es/~einsfran/papers/38-wer2002.pdf>
- ✓ Laguna M. Requisitos. Extraído el 10 de marzo del 1997 de <http://www.infor.uva.es/~mlaguna/is1/apuntes/3-requisitos.pdf>
- ✓ Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado ". Prentice Hall, 2004
- ✓ López C. Ejemplo de desarrollo software utilizando la metodología RUP. Extraído el 15 de mayo del 2007 de <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/index.html>
- ✓ Paulk, M. y otros. The Capability Maturity Model: Guidelines for Improving the software Process. Addison Wesley. 1995.
- ✓ Pressman R.S. Ingeniería del Software: un enfoque práctico. Mc Graw Hill. 2001. 5ª Edición.
- ✓ Procesos de la Ingeniería de Requisitos.pdf. Extraído el 4 de febrero del 1997 de <http://www.infor.uva.es/~chernan/Ingenieria/Teoria/Tema3C.pdf>
- ✓ Requerimiento (Sistemas). Wikipedia. 27 may 2007. Extraído el 25 de febrero del 1997 de [http://es.wikipedia.org/wiki/Requerimiento_\(sistemas\)](http://es.wikipedia.org/wiki/Requerimiento_(sistemas))

- ✓ Roger S. Pressman, "Ingeniería del Software: Un enfoque práctico", Segunda edición, Editorial McGraw Hill, 1990
- ✓ Rumbaugh J. et al. Object-Oriented Modeling and Design. Prentice Hall. 1991
- ✓ Schuller, J. UML in 24 hours. SAMS. 1999.
- ✓ Sommerville, I. Ingeniería del software. Pearson, 2005 (7ª ed.)
- ✓ Sommerville, I. Ingeniería de Software. 2ª Edición. Addison-Wesley Iberoamericana 1988.
- ✓ Sommerville, I.: Software Engineering 4th Edition. Addison-Wesley, 1992.
- ✓ Sommerville, I. y Sawyer, P., Requirements Engineering, Wiley. 1997.
- ✓ Tema 7 Procesos de Ingeniería.ppt.11 Noviembre 2005. Extraído el 10 de marzo del 1997 de <http://gva1.dec.usc.es/~antonio/docencia/EspProyectos/6a%20clase%20-%20Tema%207%20Procesos%20de%20ingenieria.pdf>
- ✓ Thayer, R. y Dorfman M., Software Requirements Engineering, IEEE, 1997.

ANEXOS

Anexo 1- Entrevista realizada a los proyectos productivos de la Universidad de Ciencias Informáticas

Entrevistas a proyectos productivos vinculados al desarrollo de aplicaciones.

Tema: Procedimiento para la elicitación, análisis y negociación, y especificación de Requisitos.

Proyecto: _____

Nombre y apellidos: _____

1-Tamaño del Proyecto: a) Grande ____ b) Mediano ____ c) Pequeño ____

2- Metodología que utiliza: a) RUP ____ b) XP ____ c) MDA ____ d) Otra: _____

3- Utiliza alguna metodología o procedimiento específico para la elicitación de requisitos?

Si No

4-Señale las técnicas que utiliza para la elicitación de requisitos

Técnicas	Utiliza
Entrevistas	
Join Application Development(JAD)	
Brainstorming o tormenta de ideas	
Casos de uso	
Observación	

5 – En caso de no utilizar estas técnicas, cual utiliza? _____

6 - Utiliza algún tipo de plantilla para las reuniones de elicitación o para registrar requisitos

Si _____ No _____ Cuáles utiliza? _____ -

7- Como resultado de este proceso de elicitación de ingeniería de requisitos le queda algún documento donde queden recogido los requisitos del sistema

Si _____ No _____ Diga cual? _____

8-Señale las técnicas o modelos que utiliza para la especificación de requisitos.

Técnicas	Utiliza
Diagrama de Casos de Uso	
Especificación de Casos de Uso	
Historia de Usuario	
Especificación de Requisitos No Funcionales	
Otros:	

9-Utiliza alguna metodología o procedimiento específico para el análisis y negociación de requisitos.

Si	No

10- Mencione las estrategias que utiliza para la negociación con el cliente.

Utiliza	Ninguna

Anexo 2 Documento de Especificación de Requisitos.

Universidad de las Ciencias Informáticas



Documento de Especificación de Requisitos

- ✓ Nombre del Proyecto
- ✓ Revisión.
- ✓ Versión.
- ✓ Índice.
- ✓ Introducción.
- ✓ Descripción del sistema actual.
- ✓ Descripción técnica del entorno del sistema.
- ✓ Participantes en el proyecto.
- ✓ Objetivos del sistema.
- ✓ Listado de características y necesidades.
- ✓ Lista de clientes con su descripción.
- ✓ Lista de las personas involucradas y sus responsabilidades.
- ✓ Relación de requisitos funcionales agrupados por funcionalidad.
- ✓ Relación de requisitos no funcionales.
- ✓ Requisitos de almacenamiento.
- ✓ Restricciones del dominio y Reglas del Negocio.
- ✓ Requisitos de interfaz, documentación, legales, seguridad.
- ✓ Origen y razón de cada requisito.
- ✓ Costos del proyecto.
- ✓ Características del software.

Anexo 3-Plantilla para la Especificación de Casos de Uso.

<Nombre del proyecto>
Especificación de Caso de Uso:
<Nombre del caso de uso>

Versión 0.9

[Nota: La siguiente plantilla se ha desarrollado para su uso con Rational Unified Process. El texto que se encuentra entre corchetes y presentado en estilo itálico azul se ha incluido para proporcionar una guía para el autor y se debería borrar antes de la entrega del documento.]

[Hay que sustituir el texto resaltado con marcador amarillo por su equivalente en el proyecto de desarrollo y eliminar el resaltado]

[La versión del documento se actualizará según la iteración y la fase del proyecto]

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15/01/2005	1.0	Versión preliminar como propuesta de desarrollo.	

Tabla de Contenidos

1.	<Nombre del caso de uso>	84
1.1	Descripción	84
2.	Flujo de Eventos	84
3.	Requerimientos Especiales	85
3.1	< 1er Requerimiento Especial >	85
4.	Precondiciones	85
4.1	<Una Precondición>	85
4.2	<Otra precondición>	85
5.	Poscondiciones	85
5.1	<Una poscondición>	85
5.2	<Otra poscondición>	85
6.	Puntos de Extensión	85
6.1	<Un punto de extensión>	86
6.2	<Otro punto de extensión>	86

Especificación de caso de uso: <Nombre del caso de uso>

<Nombre del caso de uso>

Descripción

[Breve descripción en líneas generales de la funcionalidad del caso de uso, de los actores que intervienen y del entorno de invocación]

[La descripción debe contener de forma breve el rol y el propósito del caso de uso. Un párrafo simple puede ser suficiente para esta descripción.]

Flujo de Eventos

Flujo Básico	Flujos Alternativos

Flujo Básico

[Este caso de uso comienza cuando un actor realice algo. Un actor siempre inicia un caso de uso. El caso de uso debe describir que hace el actor y que realiza el sistema en respuesta. Este deberá redactarse en forma de dialogo entre el actor y el sistema]

El caso de uso debe describir que sucede dentro del sistema, pero no como ni porque, si existe intercambio de información se será específico sobre lo que es pasado, por ejemplo no es correcto decir que el actor introduce la información del cliente; es mucho mejor decir que el actor introduce el nombre y la dirección del cliente, un glosario de términos puede ser definido y aquí especificar lo que cada información significa

Flujo Alternativo

[Alternativas más complejas deberán ser descritas en una sección separada. En esta sección podrá especificar alternativas en el comportamiento del sistema, podrá ser tan extensa como sea necesario para describir los eventos asociados con un comportamiento alternativo.] Durante la ejecución de un caso de uso, suelen aparecer errores o excepciones. Por ejemplo, mientras se ingresa un pedido, el cliente puede solicitar un producto que está discontinuado. El sistema deberá en este caso informar esta situación al empleado que ingresa el pedido. Esas desviaciones del curso normal del caso de uso se llaman alternativas. Las alternativas tienen las siguientes características:

Representan un error o excepción en el curso normal del caso de uso.

No tienen sentido por sí mismas, fuera del contexto del caso de uso en el que ocurren.

Las alternativas simples pudieran estar presente dentro del texto del caso de uso, si la descripción solo se lleva unas cuantas oraciones, descríbelas directamente dentro de la sección de flujo de eventos, si la alternativa tiene un flujo

complejo, emplea una sección separada para describirla.

Puedes emplear gráficos para mostrar de una forma más clara el flujo de evento.

Si bien en la bibliografía las alternativas se documentan al final del caso de uso, la experiencia demuestra que resulta útil documentar los casos en tablas, mostrando el curso principal en la primera columna, y las alternativas en una segunda columna.]

Requerimientos Especiales

[Un requerimiento Especial es típicamente un requerimiento no funcional el cual es específico a un determinado caso de uso el cual no es fácil o naturalmente especificado en el texto del flujo de eventos del caso de uso. Por ejemplo; aquí pueden ser incluidos requerimientos legales o regulatorios, atributos de calidad del sistema construidos incluyendo usabilidad, confiabilidad, requerimientos del desempeño o de sustentabilidad. Adicionalmente otros requerimientos tales como el del sistema operativo y el ambiente, compatibilidad y restricciones del diseño podrán ser detallados en esta sección.]

< 1er Requerimiento Especial >

Precondiciones

[Una pre-condición (de un caso de uso) es el estado en el que tiene que encontrarse el sistema antes de que el caso de uso sea ejecutado.]

[Las precondiciones se pueden eliminar si no son relevantes]

<Una Precondición>

<Otra precondición>.....

Poscondiciones

[Las poscondiciones se pueden eliminar si no son relevantes]

<Una poscondición>

<Otra poscondición>.....

Puntos de Extensión

Muchas veces, la funcionalidad de un caso de uso incluye un conjunto de pasos que ocurren sólo en algunas oportunidades. La extensión consiste en interrumpir el caso de uso y pasar a ejecutar otro caso de uso.

Las extensiones tienen las siguientes características:

- *Representan una parte de la funcionalidad del caso que no siempre ocurre.*
- *Son un caso de uso en sí mismas.*
- *No necesariamente provienen de un error o excepción. En su libro, Jacobson ejemplifica los casos de uso con ir a cenar a un restaurante. Para él, tomar café después de cenar es un ejemplo de una extensión.*

La pregunta que surge claramente es ¿cuál es la diferencia entre una alternativa y una extensión? La respuesta puede derivarse de las características de cada uno:

- *Una extensión es un caso de uso en sí mismo, mientras que una alternativa no.*
- *Una alternativa es un error o excepción, mientras que una extensión puede no serlo.*

De todas formas, en la práctica aparecen dudas con respecto a la conveniencia de considerar algo optativo en un caso como una alternativa o una extensión, sobre todo porque no queda claro si algo puede ser visto como un caso de uso en sí mismo o no. Como regla aproximada en este caso podemos pensar que si algo opcional debe ser expresado con más de un paso, seguramente es una extensión y no una alternativa.

[Los puntos de extensión se pueden eliminar si no son relevantes]

<Un punto de extensión>

[Definición de la localización del punto de extensión en el flujo de eventos.]

<Otro punto de extensión>.....

[Definición de la localización del punto de extensión en el flujo de eventos.]

GLOSARIO

Artefacto: pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tenida en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento.

Metodología: la rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adoptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

Modelos: idealización de la realidad utilizada para plantear un problema, normalmente de manera simplificada en términos relativos y planteada desde un punto de vista matemático, aunque también puede tratarse de un modelo físico. Es una representación conceptual o física a escala de un proceso o sistema (fenómeno), con el fin de analizar su naturaleza, desarrollar o comprobar hipótesis o supuestos y permitir una mejor comprensión del fenómeno real al cual el modelo representa.

Procedimiento: es el modo de ejecutar determinadas acciones que suelen realizarse de la misma forma, con una serie común de pasos claramente definidos, que permiten realizar una ocupación o trabajo correctamente.

Proceso de Ingeniería de Requisitos: conjunto de actividades que son seguidas con el objetivo de descubrir, modelar, validar y mantener un documento de requisitos. Este proceso debe lidiar con diferentes puntos de vista, usar una combinación de técnicas, herramientas y personas. Todo este proceso acontece en un universo de discurso con actores reales, por lo que se puede considerar un proceso centrado en las personas.

Proyecto: es un esfuerzo temporal emprendido para crear un producto o un servicio único. Así, el resultado final buscado puede diferir con la misión de la organización que la emprende, ya que el proyecto tiene determinado específicamente un plazo y el esfuerzo es temporal.

Requisito:

- condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo
- Condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal.
- Representación en forma de documento de una condición o capacidad como las expresadas anteriormente.

Técnica: es un procedimiento o conjunto de procedimientos que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología, del arte o en cualquier otra actividad.