



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Académico Profesional de Ingeniería de Sistemas

**Metodología para la ingeniería de requerimientos para
proyectos de desarrollo de software aplicado a
empresas clientes de GMD S.A.**

TESINA

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Oscar Freddy SONCO TENA

ASESOR

Lic. César LUZA MONTERO

Lima, Perú

2010



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Sonco, O. (2010). *Metodología para la ingeniería de requerimientos para proyectos de desarrollo de software aplicado a empresas clientes de GMD S.A.* Tesina para optar el título de Ingeniero de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

RESUMEN

**METODOLOGÍA PARA LA INGENIERÍA DE REQUERIMIENTOS
PARA PROYECTOS DE DESARROLLO DE SOFTWARE
APLICADO A EMPRESAS CLIENTES DE GMD S.A.**

Oscar Freddy Sonco Tena

Enero – 2010

Asesor : Lic. Cesar Luza Montero
Grado : Bachiller de Ingeniería de Sistemas

En el presente trabajo se describe un entorno metodológico para la Ingeniería de Requerimientos de Sistemas de información orientada a los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

La metodología propuesta está estructurada en un modelo de cinco etapas iterativas con una serie de tareas y técnicas recomendadas por cada etapa de la metodología. Se describe también la aplicación de la metodología al desarrollo de un prototipo para un Sistema de Servicio de Soporte de Sistemas en un proyecto de desarrollo de software aplicado a Empresas Clientes de GMD S.A. , describiéndose los pasos seguidos para ello y las conclusiones obtenidas.

Palabras claves: Ingeniería de Requerimientos, Sistemas de Información, Metodología.



ABSTRACT

**METHODOLOGY FOR THE ENGINEERING OF
REQUIREMENTS FOR PROJECTS OF SOFTWARE
DEVELOPMENT APPLIED TO BUSINESS
CUSTOMERS OF GMD SA**

Oscar Freddy Sonco Tena

January – 2010

Adviser : **Lic. Cesar Luza Montero**
Degree : **Bachelor of Systems Engineering**

In the present work it describes a methodologic environment for the Engineering Requirements of information systems oriented Software development projects applied to Business Customers of GMD S.A.

The propose methodology is structured in a model of five iterative stages with a series of tasks and techniques recommended by each stage. It also describes the application of the methodology to the development of a prototype for a System of Systems Support Services in a software development project applied to Business Customers of GMD S.A. , describing the steps followed for it and the obtained conclusions.

Keywords: Requirements Engineering, Information Systems, Methodology.



INDICE

INTRODUCCION	6
1. PLANTEAMIENTO DEL PROBLEMA.....	8
1.1 Fundamentación del problema.....	8
1.1.1 Descripción de la realidad problemática.....	8
1.1.2 Antecedentes del Problema.....	14
1.1.3 Formulación del Problema.....	17
DEFINICION DE PROBLEMAS	17
1.1.3.1 Problema General.....	17
1.1.3.2 Problema Especifico 1.....	18
1.1.3.3 Problema Especifico 2.....	18
DEFINICION DE OBJETIVOS.....	18
1.2 Objetivo General.....	18
1.2.1 Objetivo Especifico 1	18
1.2.2 Objetivo Específico 2	19
DELIMITACIONES DE LA INVESTIGACIÓN.....	19
1.3 Delimitaciones de la investigación	19
1.3.1 Delimitación Espacial.....	19
1.3.2 Delimitación Social	19
1.3.3 Delimitación Temporal.....	19
1.3.4 Delimitación Conceptual	19
JUSTIFICACIONES DE LA INVESTIGACIÓN.....	20
1.4 Justificación de la investigación	20
1.4.1 Justificación de carácter práctico	20
1.4.2 Justificación de carácter metodológico.....	20
1.4.3 Justificación de carácter teórico	20
MARCO TEORICO.....	21
2. MARCO TEORICO CONCEPTUAL	21
2.1 Base Teóricas.....	21
2.1.1 Requerimientos.....	21
2.1.2 Ingeniería de Requerimientos	31
2.2 Antecedentes de la Investigación.....	38
2.2.1 La Metodología DoRCU (Documentación de Requerimientos Centrada en el Usuario) para la Ingeniería de Requerimientos.....	38
2.2.2 Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información.....	39
2.2.3 System Quality Requirements Engineering (SQUARE) Methodology.....	40
2.3. Gestión del riesgo en la Ingeniería de Requisitos de un Proyecto Software	
2.3.1. Conceptos Básicos.....	41



2.3.2. Gestión de Riesgos.....	41
2.3.3. Etapas de la Gestión de Riesgos.....	43
3. ESTADO DEL ARTE.....	48
3.1 El modelo de Pohl.....	48
3.2 El modelo espiral.....	50
3.3 El modelo SWEBOK.....	52
3.4 Ingeniería de Software.....	55
3.5 UML.....	56
3.5.1 Diagramas de casos de uso.....	57
3.5.2 Relaciones entre casos de uso.....	58
3.5.3 Organización de casos de uso.....	60
3.5.4 Definición de términos básicos.....	60
4. APORTE TEÓRICO.....	62
4.1 Etapas propuestas.....	63
5. APORTE PRÁCTICO.....	68
5.1 Implementación de Etapas.....	70
5.2 La Empresa GMD S.A.....	79
5.2.1 Outsourcing de Tecnología.....	81
5.2.2 Software Factory.....	81
5.2.3 Outsourcing de Procesos.....	83
5.2.4 Outsourcing de Servicios de Aplicación.....	83
5.2.5 Servicios de Tecnología de GMD.....	85
5.3 Descripción de la aplicación de la Metodología MIRGMD.....	90
5.3.1 Etapa 1. Elicitación.....	90
5.3.2 Etapa 2. Análisis.....	97
5.3.3 Etapa 3. Especificación.....	99
5.3.4 Etapa 4. Verificación.....	100
5.3.5 Etapa 5. Administración de Requerimientos.....	101
5.3.6 Gestión de Riesgos.....	102
5.4 Descripción del Sistema a Desarrollar.....	105
5.5 Desarrollo del Prototipo.....	107
PROPUESTA METODOLÓGICA.....	113
6. CONCLUSIONES.....	161
7. RECOMENDACIONES.....	162
REFERENCIAS BIBLIOGRAFICAS.....	163
ANEXOS.....	168
ANEXO N° 1.....	168
ANEXO N° 2.....	196



CAPITULO 1: INTRODUCCION

El presente trabajo es una propuesta metodológica basada en modelos del proceso de la Ingeniería de Requerimientos y haciendo uso de varios métodos y técnicas usadas por diversos autores reconocidos en Ingeniería de Requerimientos.

Esta propuesta metodología está orientada a los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. con el fin que se obtengan eficacia y eficiencia en el desarrollo de un proyecto de software y a su vez este dentro del plazo, con costos previstos y con alto grado de satisfacción del cliente.

Encontrando en mi experiencia profesional que una mala gestión de los requerimientos conlleva al retrabajo, generando como consecuencia una entrega del producto final fuera de tiempo, también generando costos en recursos. Por lo tanto esto me motiva al desarrollo de este trabajo.

Haciendo un rápido recorrido sobre el trabajo de investigación, indicaremos que en el Capítulo 1, de este documento, describimos el Planteamiento del Problema, identificaremos los problemas y objetivos generales e específicos, limitaremos el trabajo de investigación, justificaremos el motivo de la investigación estableciendo su finalidad e importancia y finalmente formularemos las hipótesis y las variables.

En el Capítulo 2, de este documento, describiremos el Marco Teórico Conceptual, haremos las definiciones de las bases teóricas de los requerimientos e ingeniería de requerimientos, y finalmente presentaremos como antecedente de la investigación, a las Metodologías de Captura de requerimientos ya existentes.

En el Capítulo 3, de este documento, describiremos el Estado del arte, comentaremos los modelos de procesos para la ingeniería de requerimientos existentes, y finalmente haremos las definiciones de la Ingeniería de Software y el UML.



En el Capítulo 4, de este documento, describiremos el Aporte Teórico, definiendo nuestra propuesta de Metodología MIRGMD, con las etapas propuestas y estructurada para la misma.

En el Capítulo 5, de este documento, describiremos el Aporte Práctico aplicando nuestra propuesta de Metodología MIRGMD, con la implementación de las etapas propuestas, estableciendo los objetivos, tareas, entregables y técnicas definidas, la cual será desarrollada en un Sistema de Servicio de Soporte de Sistemas Integrado. Definiremos el rubro de la Empresa el cual desarrollara el Sistema, se realizara la descripción del Sistema y finalmente se presentara el desarrollo de un prototipo del Sistema de Servicio de Soporte de Sistemas Integrado.

En el Capítulo 6, de este documento, describiremos las Conclusiones las cuales se generaron luego de realizar e implementar el trabajo de investigación.

En el Capítulo 7, de este documento, describiremos las Recomendaciones que brinda el autor luego de realizar e implementar el trabajo de investigación.



1 PLANTEAMIENTO DEL PROBLEMA

1.1 Fundamentación del problema

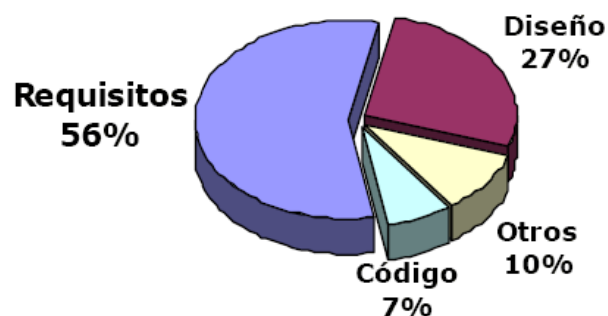
Descripción de la Realidad Problemática

Actualmente existe una creciente demanda de desarrollo de productos de Software. Este hecho inusual representa una gran oportunidad y a la vez un desafío debido a que nuestro país posee importantes capacidades para desarrollar software a medida (desarrollado para satisfacer una demanda particular). Para conseguir y asegurar mercados internacionales es necesario desarrollar productos de calidad, es decir, desarrollar el producto que el cliente/mercado desea, dentro del cronograma establecido y con los costos de desarrollo definidos. El primer paso en este camino lleva a cumplimentar la etapa de desarrollo de requerimientos. El desarrollo de requerimientos, así como la administración de los mismos, es importante ya que si se realizara de manera errónea se podría estar desarrollando el software incorrecto [Bohem:1994].

En la actualidad podemos encontrar problemas que persisten en el desarrollo de un software, entre ellos, un inadecuado entendimiento o captación de las necesidades de los usuarios, incapacidad de absorber cambios en los requerimientos e insatisfacciones de los clientes por inaceptable o bajo desempeño del software.

En un estudio de Dean Leffingwell sobre defectos en el desarrollo de software LEFFINGWELL(2003) se concluye que la mayor cantidad de defectos encontrados y el mayor esfuerzo por reparar esos defectos se encuentran en la etapa de requerimientos (figura 1.1 y 1.2).

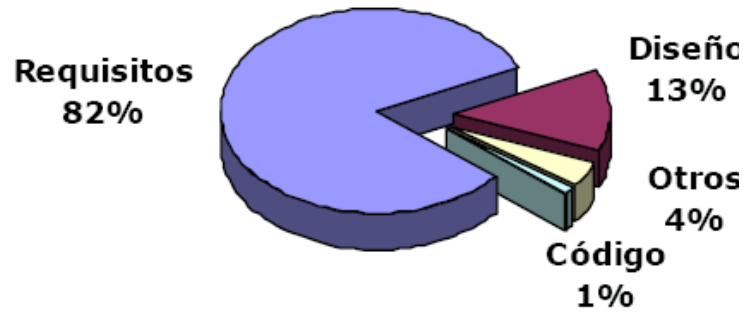
Figura 1.1 Distribución de defectos



Fuente: LEFFINGWELL(2003)



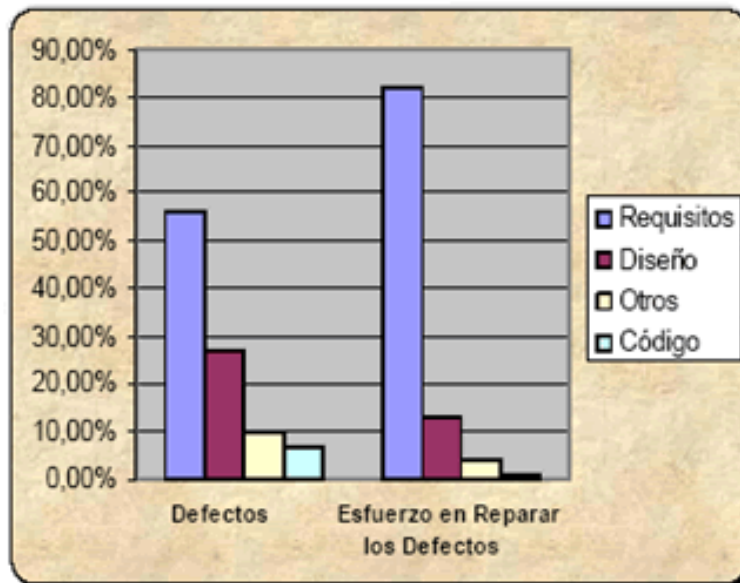
Figura. 1.2 Distribución del esfuerzo en la reparación de los defectos



Fuente: LEFFINGWELL(2003)

En el mismo estudio se muestra la distribución de defectos y distribución del esfuerzo en la reparación de los defectos expresado en porcentaje (figura 1.3).

Fig. 1.3 Distribución de defectos y Distribución del esfuerzo en la reparación de los defectos

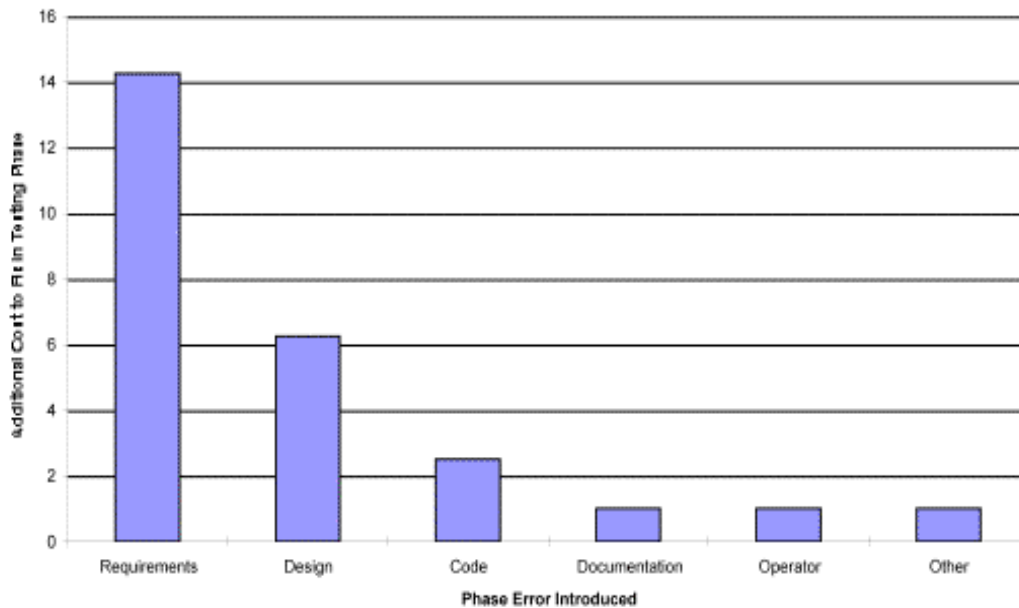


Fuentes: LEFFINGWELL(2003)

Por su parte, Linda Rosenberg, realizó una investigación en la NASA sobre Ingeniería de Requerimientos ROSENBERG(1998), cuyo resultado muestra que el costo adicional respecto a los requerimientos que se incurre en la etapa de testing del desarrollo de un proyecto.



Figura. 1.4 Costo para corregir errores en la fase de testing



Fuente: ROSENBERG (1998)

En la Fig. 1.5 se presenta el costo relativo de reparación de un error del producto en función de la etapa de desarrollo en la que se detecta. Se pone de manifiesto que cuanto más tarde en el ciclo de vida se detecta un error, más cuesta repararlo [Bohem:1994].

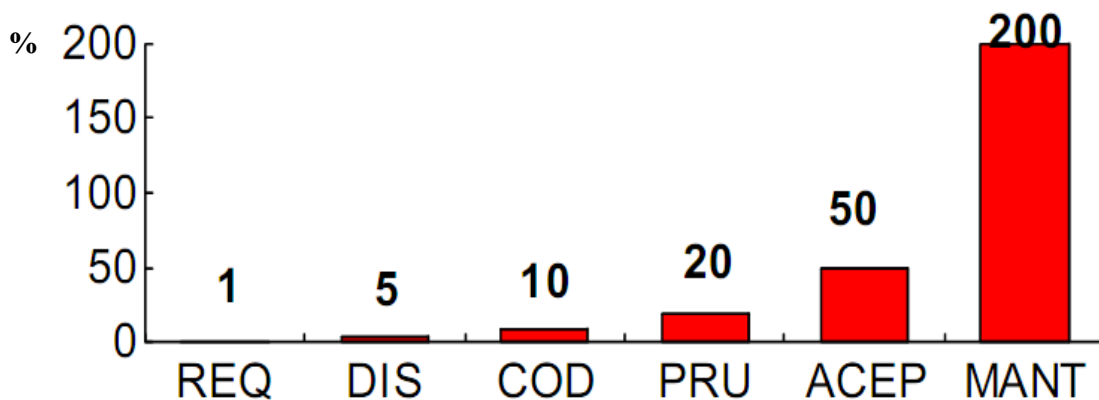


Figura 1.5 Costo de reparación de un error en función de la etapa en la que se detecta. Referencias: REQ: requerimiento, DIS: diseño, COD: codificación, PRU: prueba, ACEP: aceptación, MANT: mantenimiento [Bohem:1994]



El incremento de los costos puede devenir de la necesidad de corregir los errores originales pero también para reparar aquellos errores que se generan en etapas posteriores debido a problemas durante la etapa “Requerimiento”.

Como se visualiza en la Fig. 1.6, los problemas con los requerimientos constituyen una de las fuentes principales de inconvenientes (37%) [Standish:1995]

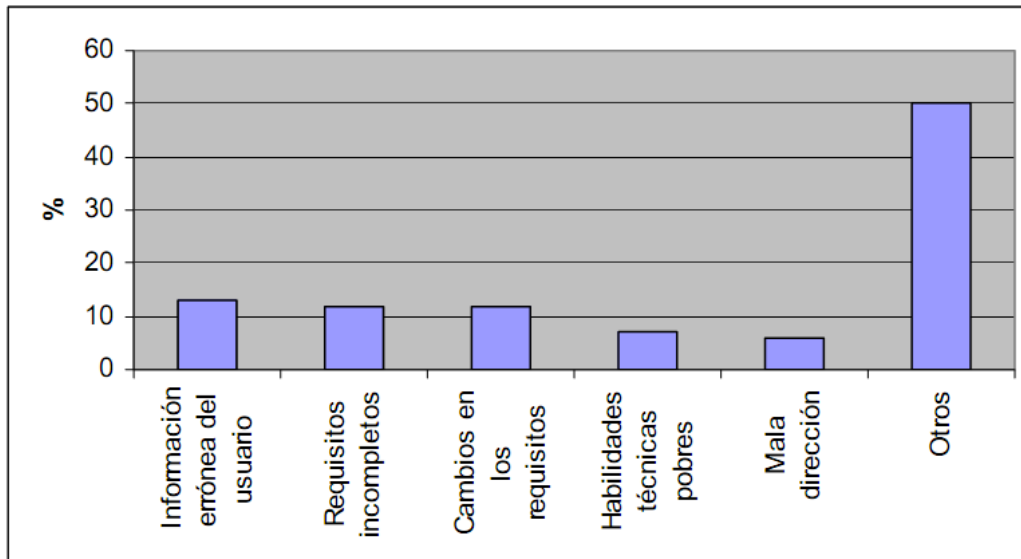


Figura 1.6 Factores del coste en proyectos de software reales [Standish:1994]

Los errores en la etapa de requerimientos tienen un importante impacto [Standish:1995]:

- El software resultante puede no satisfacer a los usuarios.
- Las interpretaciones no precisas de los requerimientos pueden causar desacuerdos entre clientes y desarrolladores.
- Puede gastarse tiempo y dinero construyendo el sistema incorrecto.

Entre los problemas asociados a la ingeniería de software, los requerimientos son los problemas más frecuentes. Se debe asignar gran esfuerzo en la recolección de requerimientos lo antes posible en el desarrollo del sistema. Puede suceder que: los requerimientos no reflejan las necesidades reales de los clientes y/o que los requerimientos son inconsistentes entre si y/o son incompletos. Estos problemas provienen de un mal entendimiento entre los clientes, las personas que desarrollan los requerimientos y los ingenieros de software que desarrollan o mantienen el sistema [COMP]. El principal problema en la recolección de requerimientos es de



comunicación. Por otro lado, es casi imposible probar formalmente que todos los requerimientos reflejan las necesidades reales de los usuarios. La naturaleza difusa de la recolección de requerimientos convierte a esta tarea en un eslabón débil de la cadena de actividades de desarrollo de software. Al mismo tiempo, la recolección de requerimientos es una de las fases más importantes ya que mediante ésta se determina cuál es la naturaleza del producto que se va a desarrollar. Todas las fases subsecuentes en el desarrollo están basadas en la recolección de requerimientos. Si los requerimientos son incorrectos, el software desarrollado puede representar un buen código pero no satisfará al cliente, y por tanto el trabajo será en vano [COMP].

El concepto de conocimiento del usuario fue introducido para generar una identificación efectiva de las necesidades de los mismos. Se definen dos tipos de conocimiento: conocimiento del usuario y conocimiento del diseñador [Sakol:2001].

El conocimiento del usuario es generado en el proceso de aprender y de resolver el problema en el contexto de uso. Los usuarios obtienen e interpretan la información embebida en los productos, como las funciones, atributos y métodos de uso. En un intento de utilizar lo mejor posible el producto, los usuarios intentan adaptar y modificar la manera del uso, las funciones y los atributos para adaptarlo a las situaciones fuera de consideración del diseño [Sakol:2001].

El conocimiento del diseñador, es considerado como aquel derivado de la construcción del producto. Baldwin and Clark (2000) definen al diseño como una descripción completa del producto; el resultado del proceso de producción es el producto en sí mismo [Sakol:2001].

Estos dos tipos de conocimiento componen lo que se conoce como “ciclo de vida del conocimiento” que se esquematiza en la Fig. 1.7. El problema de no conectar el conocimiento del usuario con el conocimiento adquirido en el diseño ocurre frecuentemente en la práctica y conduce a productos con más funciones de aquellas que los usuarios necesitarán. La elicitación del conocimiento del usuario proveerá un enlace desde el usuario hacia el desarrollador del producto [Sakol:2001].



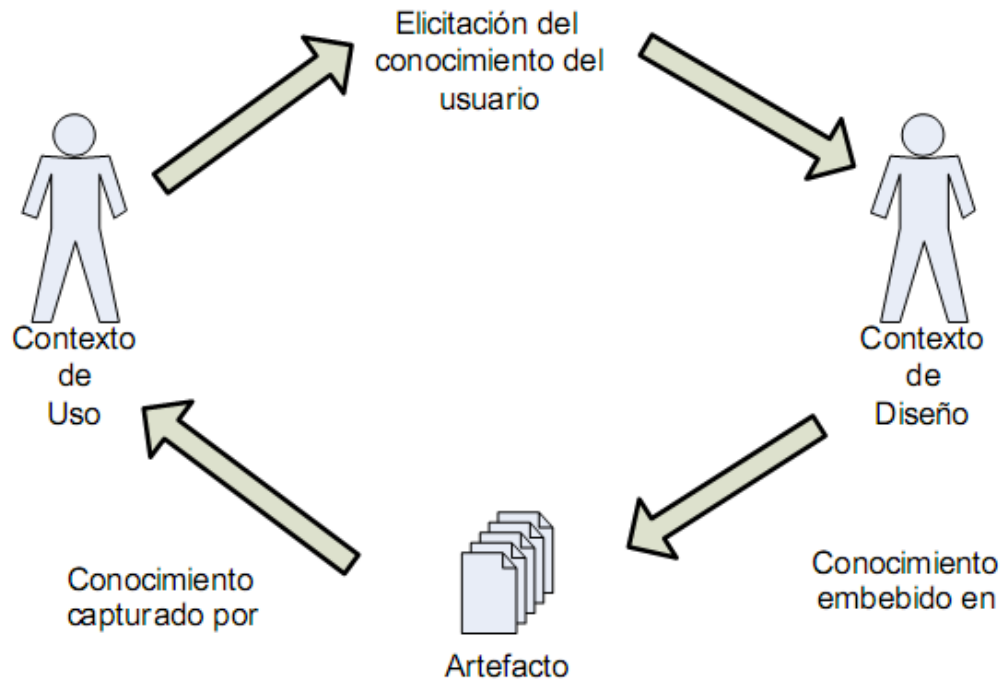


Figura 1.7 Ciclo de vida del conocimiento

La elicitación de requerimientos es el proceso de descubrir los requerimientos para un sistema a través de la comunicación con los clientes, usuarios del sistema y otras personas que tengan algún tipo de interés y conocimiento sobre el producto a desarrollar [Madigan].

Algunos de los beneficios más inmediatos y cuantificables de las métricas respecto a los requerimientos son [TickIT:2002]:

- Validación de la completitud de la definición de los requerimientos
- Identificación y medición de la conformidad de los requerimientos de software.
- Identificación y medición del logro de los objetivos de diseño del software
- Validación de la completitud de los objetivos de testing del software
- Identificación del criterio de aceptación del usuario por un producto de software completo.
- Provisión de una medida de confianza en los requisitos de calidad específicos del producto, por ejemplo, de confiabilidad y de funcionalidad.

Dentro de las métricas planteadas para el proceso de administración de cambios en los requerimientos, se puede mencionar [Borland:2004]:

- Cantidad de cambios o propuestas de cambios en los requerimientos para un proyecto.
- Frecuencia de cambios en los requerimientos
- Índice de la introducción de nuevos requerimientos
- Cantidad de cambios a requerimientos que están bajo línea base.
- Porcentaje de defectos encontrados cuya causa raíz sea errores en los requerimientos

Dentro de las métricas planteadas para el proceso de administración de requerimientos, se puede mencionar [Borland:2004]:

- Tendencia al lanzamiento de defectos en un cierto plazo.
- Tendencia a las propuestas de cambio que impliquen re-trabajo

Lo expuesto anteriormente sobre métricas muestra que existen distintas propuestas para el proceso de ingeniería de requerimientos, más específicamente para los cambios y administración de los mismos.

1.2 Antecedentes del Problema

Desde 1968 se ha invertido gran esfuerzo en determinar las causas y proponer soluciones para la crisis del software (término utilizado por primera vez en la conferencia organizada por la Comisión de Ciencia de la OTAN en Garmisch, Alemania, en octubre de 1968, para designar la gran cantidad de problemas que presentaba (y aún presenta) el desarrollo de software y el alto índice de fracasos en los proyectos de desarrollo).

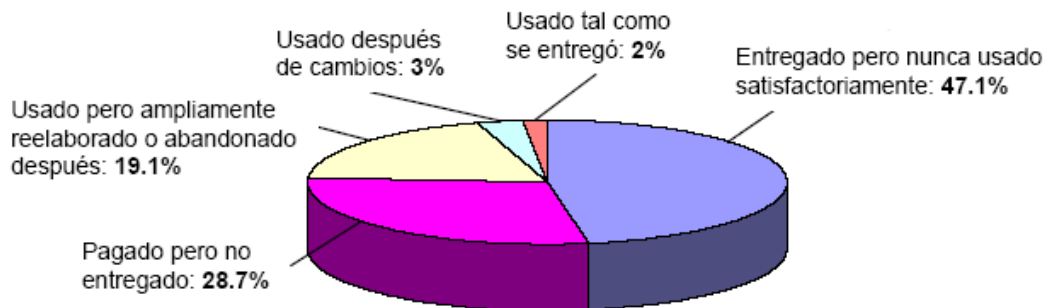
En 1979, la Oficina de Cuentas del gobierno norteamericano (*Government Account Office*, GAO) realizó un estudio GAO(1979) seleccionando 9 proyectos de desarrollo de software para el gobierno norteamericano cuyos contratos sumaban una cantidad total de 6.800.000 dólares.



De esta cantidad, sólo 119.000 dólares correspondían a un proyecto que se había utilizado tal como se había entregado. Dicho proyecto se trataba de un preprocesador de COBOL, por lo que era un problema relativamente simple cuyos requisitos eran comprendidos por clientes y desarrolladores y que además no cambiaron durante el desarrollo.

El resto de los 6.8 millones de dólares se distribuyeron como puede verse en la figura 1.8, en la que puede destacarse el enorme porcentaje de dinero invertido en proyectos cancelados o no satisfactorios.

Figura 1.8 Resultado del informe de GAO



Fuente: GAO(1979)

En 1995, el Grupo Standish realizó un estudio (el informe CHAOS) mucho más amplio y significativo que el del GAO cuyos resultados, a pesar de haber pasado más de 25 años, no reflejaban una mejoría sustancial TSG(1995).

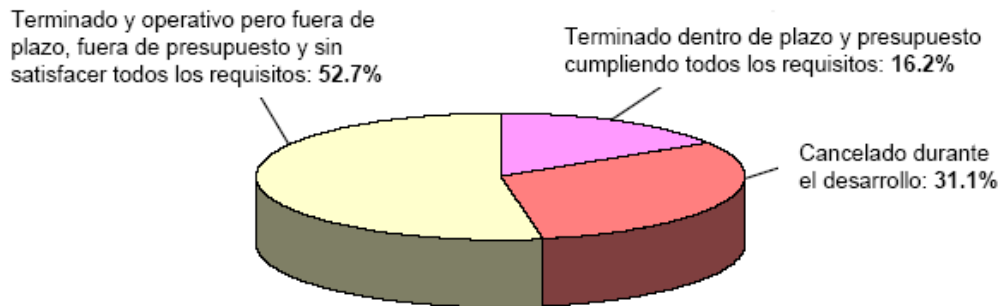
Los resultados generales, que pueden verse en la figura 1.9, al compararse con los de GAO(1979) presentan una mejora en los proyectos que se entregan cumpliendo todos sus requisitos, 2% frente al 16.2% (sólo el 9% en grandes compañías), pero empeoran ligeramente respecto a los que se abandonan durante el desarrollo, 28.7% frente a 31.1%.

Sin incluir al 16.2% de los proyectos terminados correctamente, la media del gasto final fue del 189% del presupuesto original, el tiempo necesario para su



realización del 222% del plazo original y se cumplieron una media del 61% de los requisitos iniciales, cifras que también empeoraban en el caso de grandes compañías.

Figura 1.9 Resultados del informe CHAOS



Fuente: TSG(1995)

Las encuestas realizadas a los directores de los proyectos que participaron en el estudio indicaron que, en su opinión, los tres principales factores de éxito eran:

1. Implicación de los usuarios
2. Apoyo de los directivos
3. Enunciado claro de los requisitos

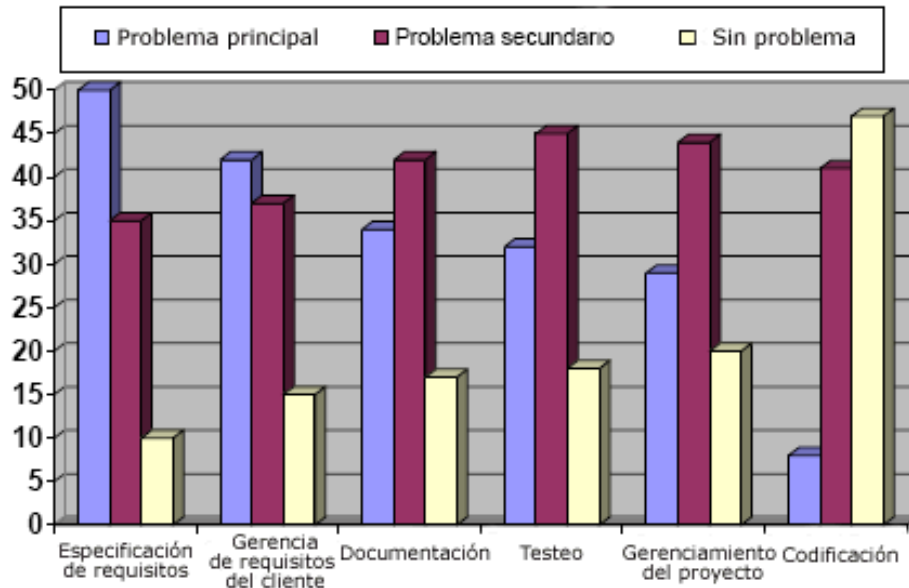
Mientras que los tres principales factores de fracaso eran:

1. Falta de información por parte de los usuarios
2. Especificaciones y requisitos incompletos
3. Especificaciones y requisitos cambiantes

En 1996, el proyecto ESPITI (*European Software Process Improvement Training Initiative*) ESPITI(1996) realizó una investigación sobre los principales problemas en el desarrollo de software a nivel europeo. Los resultados figura 1.10, muy similares a los obtenidos en el informe CHAOS, indicaron que los mayores problemas estaban también relacionados con la especificación, la gestión y la documentación de los requisitos.



Figura 1.10 Mayores problemas en el desarrollo de software



Fuente: ESPITI(1996)

En nuestro país en el 2004 la Oficina Nacional de Gobierno Electrónico e Informática - ONGEI de la Presidencia del Consejo de Ministros desarrollo una Metodología de Desarrollo de Sistemas de Información – MDSI, para las entidades integrantes del Sistema Nacional de Informática donde de incluye una etapa “MODELAMIENTO DE REQUERIMIENTOS” con la finalidad de obtener mejores resultados en el desarrollo de software.

1.1.3 FORMULACION DEL PROBLEMA

DEFINICION DE PROBLEMAS

1.1.3.1 Problema General

Las metodologías basadas en el proceso de la ingeniería de requerimientos en el desarrollo de software a pesar de su evolución aún presentan carencias, no todos abordan integralmente el proceso, ya que algunos no incorporan la gestión de los requerimientos, ni el manejo de los cambios, o no tiene en cuenta la inevitable interacción con el resto del ciclo de vida del proyecto y no plantean las mediciones a este proceso. Por otro lado, existen insuficiencias en la gestión de requisitos tanto al nivel de proyecto como al nivel de organización.



Entonces el Problema General encontrado en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. será el siguiente:

¿ En qué medida las Empresas Clientes de GMD S.A., podrá mejorar sus procesos de gestión de ingeniería de requerimientos, en los Proyectos de desarrollo de software, con el fin de que se obtengan eficacia y eficiencia en el desarrollo de un proyecto de software y a su vez este dentro del plazo solicitado, con los costos previstos y con un alto grado de satisfacción del cliente. ?

1.1.3.2 Problema Especifico 1

- a) ¿En que medida la documentación de los requerimientos del usuario final puede prevenir un inadecuado entendimiento o captación de las necesidades de los usuarios, en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. ?.

1.1.3.3 Problema Especifico 2

- a) ¿Cómo la elaboración de las etapas para la captura de requerimientos puede mejorar los tiempos de desarrollo y reducir los costos en los cambios de los requerimientos en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. ?.

DEFINICION DE OBJETIVOS

1.2 Objetivo general

Proponer una metodología para la ingeniería de requerimientos orientada a los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. , el cual podamos aplicar en la etapa de diseño, desarrollo y testeo del software para el usuario final.

1.2.1 Objetivo Específico 1

- a) Elaboración de una documentación en donde se especifique los requerimientos del desarrollo de software del usuario final en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. ?.



1.2.2 Objetivo Específico 2

- a) Elaboración de las etapas para la captura de requerimientos en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. , la cual nos permitirá mejorar los tiempos de desarrollo y reducir costos en los cambios de requerimientos.

DELIMITACIONES DE LA INVESTIGACIÓN

1.3 Delimitaciones de la Investigación

1.3.1 Delimitación Espacial

El ámbito físico geográfico es el Perú, debido a que esta metodología propuesta está enfocada en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

1.3.2 Delimitación Social

Socialmente esta orientada para todas las Empresas del sector informático involucradas en el desarrollo de software en nuestro país.

1.3.3 Delimitación Temporal

Temporalmente está planificado la Implementación de la Metodología de Ingeniería de Requerimientos, en el periodo de 1 año que abarca el Proyecto de desarrollo de Software.

1.3.4 Delimitación Conceptual

Conceptualmente está delimitado por la Ingeniería de Requerimientos, Ingeniería de Software y calidad de desarrollo de software.



JUSTIFICACIONES DE LA INVESTIGACIÓN

1.4 Justificación de la investigación

1.4.1 Justificación de carácter práctico

Desde el punto de vista práctico, el estudio sugerirá la aplicación de la Ingeniería de Requerimientos mediante una metodología que hace uso de diversas técnicas y métodos de requerimientos, permitiendo una mayor eficiencia en el desarrollo de un proyecto de software que este dentro del plazo y costos previstos, con alto grado de satisfacción del cliente, esto orientado a los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Se justifica la orientación a las Empresas Clientes de GMD S.A. basándome en una investigación realizada en el año 2003, para determinar el potencial exportador de las empresas de software y servicios informáticos, realizado por el Instituto Cuanto, CUANTO (2004), por encargo de la *Comisión para la Promoción de Exportaciones – PROMPEX* y la *Asociación Peruana de Productores de Software – APESOFT*, donde se tomo una muestra conformada por 150 compañías peruanas de software ubicadas en Lima Metropolitana. Las principales conclusiones derivadas del diagnóstico muestran a una industria peruana de software muy joven pero con gran potencial de crecimiento.

1.4.2 Justificación de carácter metodológico

La investigación es importante metodológicamente porque se propone una alternativa de mejora al proceso de desarrollo de software, plasmando el proceso de la Ingeniería de Requerimientos a través de una metodología a aplicarse en los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. en el país.

1.4.3 Justificación de carácter teórico

Desde el punto de vista teórico, el presente trabajo de investigación va permitir enriquecer la concepción teórica sobre el proceso de ingeniería de requerimientos.



CAPITULO 2: MARCO TEÓRICO

2 MARCO TEORICO CONCEPTUAL:

El avance de Internet y las comunicaciones de los últimos años ha provocado un interés creciente por el desarrollo de propuestas metodológicas que ofrezcan un marco de referencia adecuado cuando se desarrolla un sistema de información. Así, en los últimos años son varios los grupos que han trabajado y han propuesto metodologías que ofrecen procesos, modelos y técnicas adecuadas para trabajar con este tipo de sistemas. Sin embargo, si se analizan las diferentes propuestas, la gran mayoría de ellas enfocan su trabajo en la etapa de diseño del ciclo de vida, dando menor importancia a la gestión de requisitos, al testeo y a la gestión de calidad.

El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido. La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada captura de requisitos.

2.1 Base teóricas.

Para poder llegar a un entendimiento más acabado de lo que la ingeniería de requerimiento es y qué representa, antes debemos entender qué son los requerimientos, cómo están definidos y para qué sirven. Existen varias definiciones, entre las que podemos citar:

2.1.1 Requerimientos

Definición:

Normalmente, un tema de la Ingeniería de Software tiene diferentes significados. De las muchas definiciones que existen para requerimiento, ha continuación se presenta la definición que aparece en el glosario de la IEEE .



- **Requerimientos (1)** Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- **Requerimientos (2)** Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- **Requerimientos (3)** Una representación documentada de una condición o capacidad como en (1) o (2).
- **Requerimientos (4)** Son expresiones de las necesidades de stakeholders para alcanzar una meta particular.

Tipos de Requerimientos

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

- **Los requerimientos funcionales** definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.
- **Los requerimientos no funcionales** tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Clasificación de Requerimientos

Los requerimientos deben ser clasificados de acuerdo a un identificador, prioridad, criticismo, factibilidad, riesgo, fuente, y tipo.

a) Identificación. Cada requerimiento debe ser identificado de forma única (por ej., un número, etiqueta, mnemónico, etc.).

b) Prioridad. El cliente debe identificar la prioridad de cada requerimiento. Ésta puede ser establecida mediante un consenso entre los clientes potenciales. Se puede usar una escala 1:10, o bien, un esquema tal como alta, media, baja para identificar la prioridad de cada requerimiento.



c) Criticismo. El analista, junto con el cliente, debe definir el criticismo de cada requerimiento. Algunos requerimientos pueden tener una prioridad baja desde el punto de vista del cliente, pero ser esenciales para el éxito del sistema. Por ejemplo, un requerimiento para medir la temperatura ambiente externa puede ser esencial para otros requerimientos tal como mantener la temperatura interna de la cabina de un avión.

d) Factibilidad. El cliente y el analista deben identificar la factibilidad de incluir cada requerimiento particular en el sistema y clasificar cada requerimiento por tipos de factibilidad apropiados para el dominio del sistema. La factibilidad puede basarse en el estado actual de la tecnología (por ej., componentes disponibles comercialmente vs análisis original del sistema), el ambiente del cliente (por ej., la facilidad de adaptar un cambio), y los riesgos y costos asociados al nuevo requerimiento.

e) Riesgo. Las técnicas de análisis de riesgos pueden ser usadas para determinar un nivel de riesgo para los requerimientos del sistema. En término de sus consecuencias, los riesgos pueden estar relacionados con pérdidas financieras potenciales, impacto en el ambiente, temas de seguridad y salud, y leyes y estándares nacionales.

f) Origen. Cada requerimiento debería ser clasificado mediante una etiqueta que indique su origen. Puede haber múltiples fuentes que pueden ser todas ellas consideradas los creadores del requerimiento. Es útil identificar los creadores de cada requerimiento, de tal manera que si el requerimiento no es claro, hay conflictos, o requiere ser modificado o eliminado, será posible identificar los individuos u organizaciones a ser consultados.

g) Tipo. Los requerimientos pueden ser clasificados por uno o más de los siguientes tipos:



- | | |
|------------------------------|--|
| - Entrada | - Entrenamiento |
| - Salida | - Documentación |
| - Confiabilidad | - Interfaces externas |
| - Disponibilidad | - Pruebas |
| - Facilidad de mantenimiento | - Calidad |
| - Desempeño | - Políticas y regulaciones |
| - Rutas de acceso | - Compatibilidad con sistemas existentes |
| - Condiciones ambientales | - Conversión |
| - Ergonomía | - Capacidad de expansión |
| - Seguridad | - Instalación |
| - Facilidad de uso | |
| - Medios de transporte | |

Características de los Requerimientos

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes.

- **Comprensible:** La ingeniería de requisitos es básicamente un problema de descubrimiento y comunicación de las necesidades de clientes y usuarios, por lo que la propiedad más importante de una especificación es que sirva como canal de comunicación entre los participantes en el proceso de ingeniería de requisitos (ver figura 2.1).



Figura 2.1: La Ingeniería de Requerimiento como un proceso de comunicación



Fuente: Gause y Weinberg (1989)

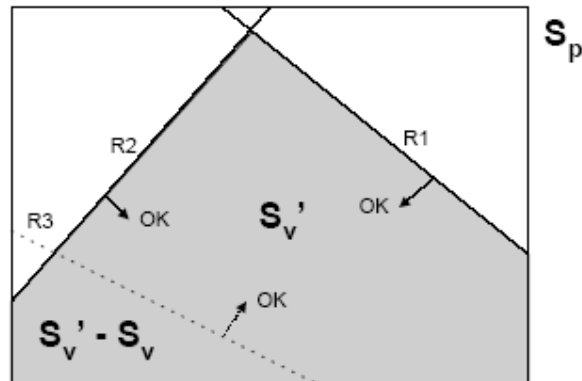
- **Necesario:** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- **Conciso:** Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- **Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

Considerando los requerimientos como restricciones, si no se especifican todos los requisitos se está definiendo un espacio incorrecto de sistemas válidos Sv' que incluiría sistemas no válidos.

Esta situación incrementa el riesgo de acabar desarrollando sistemas que no satisfagan las necesidades reales de clientes y usuarios, es decir, sistemas que pertenezcan a $Sv' - Sv$ (ver figura 2.2, en la que se supone que no se ha identificado el requisito R3.)



Figura 2.2: Requerimientos incompletos ($S_v \subset S_v'$)



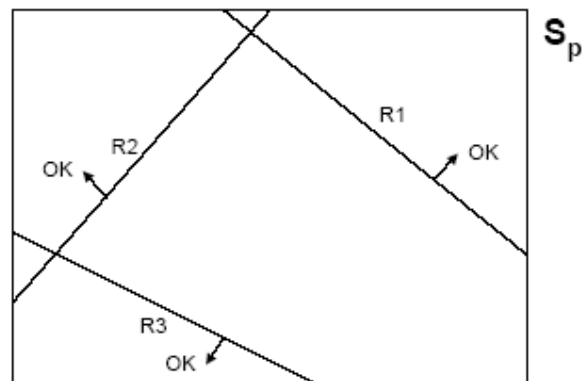
Fuente: Durán Toro, Amador (2004)

- **Consistente:** Un requerimiento es consistente si no es contradictorio con otro requerimiento.

En Wieringa(1996) se interpreta el término consistente como la propiedad de que pueda existir un sistema que satisfaga todos los requerimientos.

Evidentemente, si los requisitos son contradictorios es imposible que pueda existir un sistema que los satisfaga todos a la vez. Considerando los requisitos como restricciones, la consistencia se interpretaría como la existencia de un espacio no vacío de sistemas válidos, es decir $S_v \neq \emptyset$ (ver figura 2.3).

Figura 2.3: Requerimientos inconsistentes ($S_v = \emptyset$)

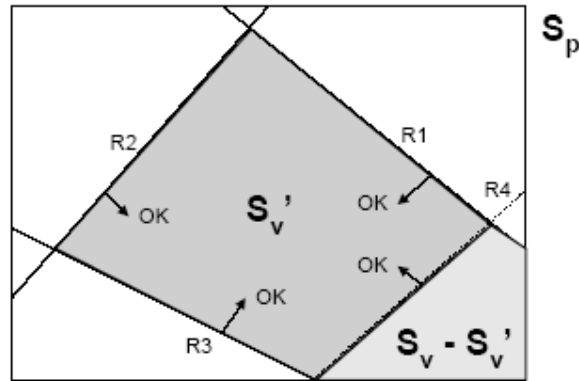


Fuente: Durán Toro, Amador (2004)



- **No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

Figura 2.4: Requerimientos innecesarios ($S_v' \subset S_v$)



Fuente: Durán Toro, Amador (2004)

- **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Considerando los requerimientos como restricciones, la verificabilidad y la no ambigüedad son necesarias para saber si un posible sistema s perteneciente al espacio de los sistemas posibles S_p cumple o no un determinado requisito R_i , es decir, si s pertenece a S_i o a $\overline{S_i}$

- **Modificable:** una especificación de requisitos es modificable si y sólo si su estructura y estilo de redacción permite que los cambios se puedan realizar fácil, completa y consistentemente Davis(1993), IEEE(1993).

Para conseguir esta propiedad, la especificación debe estar organizada coherentemente y debe contar con los índices y las tablas de referencias cruzadas oportunas, no debe ser redundante (debe estar *normalizada* según la terminología utilizada en IEEE(1996) y los requisitos deben expresarse individualmente y no de forma conjunta.



- **Rastreable:** una especificación de requisitos es rastreable si y sólo si para cada requerimiento contenido en ella se conoce su origen y puede referenciarse como origen en posteriores documentos durante el desarrollo, es decir, cada requerimiento puede rastrearse hacia atrás y hacia delante IEEE(1993), Davis(1993).

Dificultades para definir los requerimientos

- Los requerimientos no son obvios y vienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

Errores comunes en la construcción de Requerimientos

Algunos errores típicos que deben evitarse a la hora de construir un requerimiento bien formado son los siguientes:

a) Diseño e implementación. Hay una tendencia por parte de los analistas y clientes, quienes están definiendo los requerimientos, de incluir decisiones de diseño e implementación en los requerimientos. En este caso, la información debe ser documentada y comunicada en algún otro tipo de documento auxiliar de diseño e implementación



b) Sobre-especificación.

- 1) Requerimientos que describen un sistema comercial que puede ser comprado en lugar de construido (ésta no es una declaración de lo que el sistema debe hacer).
- 2) Requerimientos que establecen rangos de tolerancia para elementos de muy bajo nivel del sistema conceptual.
- 3) Requerimientos que implementan soluciones (los requerimientos describen una necesidad).

c) Sobre-restringido. Requerimientos con restricciones innecesarias (por ej., si un sistema debe trabajar con baterías recargables, un requerimientos derivado podría ser que el tiempo de recarga de la batería fuera menor a 3 hr. Si este tiempo es muy restrictivo, soluciones potenciales podrían ser desechadas).

d) Sin límites.

- 1) Requerimientos que establecen enunciados relativos (estos requerimientos no pueden ser verificados y podrían requerir ser reescritos. Por ej., el requerimiento “minimizar ruido” puede ser enunciado como “los rangos de ruido no deben exceder ...”).
- 2) Requerimientos abiertos (por ej., enunciados que contienen “etc.”).
- 3) Requerimientos que hacen declaraciones subjetivas o vagas (frecuentemente contienen términos como “amigable con el usuario”, “rápida respuesta”, o “costeable”).

e) Suposiciones.

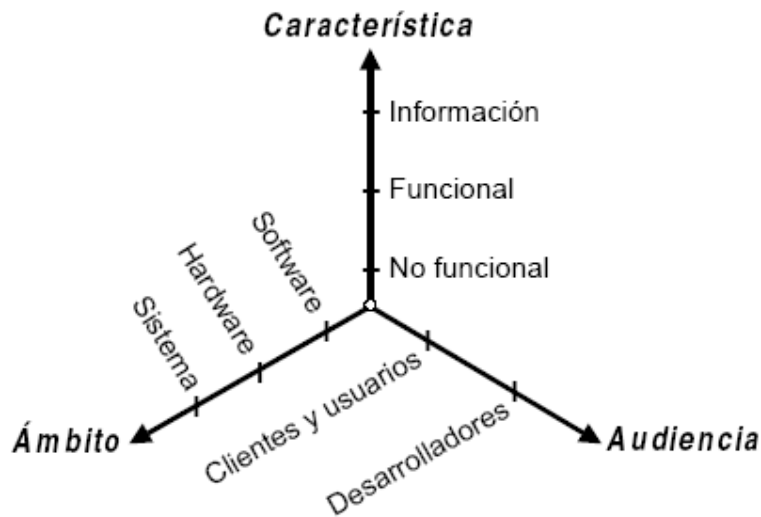
- 1) Requerimientos basados en suposiciones no documentadas (la suposición debe ser documentada tanto como el requerimiento).
- 2) Requerimientos basados en suposiciones de que un sistema o estándar en particular será terminado en determinada fecha (se debe documentar la suposición y una solución alternativa).



Las dimensiones de los requerimientos

La gran cantidad de calificativos que se aplican al término requerimiento muestran distintos aspectos ortogonales que ha menudo se consideran aisladamente. Para intentar clarificar la situación, se ha identificado tres dimensiones en las que se pueden clasificar los requerimiento (ver figura 2.5). Estas tres dimensiones son:

Figura 2.5 Dimensiones de los requisitos



Fuente: Pohl, Klaus (1997).

Ámbito: esta dimensión indica en qué ámbito se debe entender el requerimiento. En general, y siguiendo entre otras las propuestas de IEEE(1997), DoD(1994) y DAVIS, Alan(1993), un ámbito de *sistema* indica que el requerimiento debe cumplirse a nivel de sistema, entendiendo por sistema un conjunto de hardware y software. Si el ámbito es de *software* quiere decir que el requisito sólo afecta a la parte software de un sistema, mientras que si es el ámbito es de *hardware* sólo afecta a la parte hardware.

Para entender esta clasificación conviene recordar que DoD(1994) es una norma militar y que las normas IEEE(1997) están fuertemente influidas por dichas normas militares. En el contexto de los desarrollos para fines militares es frecuente tener que desarrollar sistemas en los que el hardware juega un papel tan importante como el



software. En el caso de sistemas sin una parte hardware significativa, es decir sin hardware *a medida*, los requerimientos de sistema y los requerimientos software coinciden.

Característica que define: esta dimensión clasifica los requisitos en función de la naturaleza de la característica del sistema deseada que se especifica. La clasificación más habitual suele ser la de requerimientos *funcionales* (qué funciones debe realizar el sistema) y *no funcionales* (otras características del sistema).

En Pohl, Klaus(1997) aparece una completa clasificación denominada RSM (*Requirements Specification Model*, Modelo de Especificación de Requerimientos), cuyas principales clases son: requerimientos funcionales, requerimientos de datos y requerimientos no funcionales.

Es conveniente destacar que al grupo de requisitos no funcionales no se le ha prestado la atención suficiente y que ya hay opiniones que lo consideran como un *cajón de sastre* excesivamente heterogéneo donde se han clasificado aquellos requisitos que resultan incómodos Bass (1998). Un ejemplo de esta situación es la escasa importancia que se les ha dado en las técnicas de modelado de sistemas, tanto estructuradas como orientadas a objetos o formales.

Audiencia: esta dimensión, fundamental indica la audiencia a la que está dirigido el requerimiento, es decir, las personas que deben ser capaces de entenderlo. En general, se pueden distinguir dos tipos de audiencia, los *clientes y usuarios*, que no tienen porqué tener formación en ingeniería del software, y los *desarrolladores* de software.

Cuando la audiencia está formada por clientes y usuarios, la forma más habitual de definir los requisitos es mediante lenguaje natural. En el caso de que la audiencia prevista esté formada por desarrolladores de software, los requerimientos suelen expresarse mediante un modelo, normalmente utilizando técnicas estructuradas, orientadas a objetos o formales.

2.1.2 Ingeniería de Requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es llamado Ingeniería de Requerimientos. La meta de la ingeniería de



requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa. A continuación se darán algunas definiciones para ingeniería de requerimientos.

Ingeniería de Requerimientos (1). “Es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema”.

Ingeniería de Requerimientos (2). “Es el proceso por el cual se transforman los requerimientos declarados por los clientes , ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones”.

Ingeniería de Requerimientos (3). “Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos”.

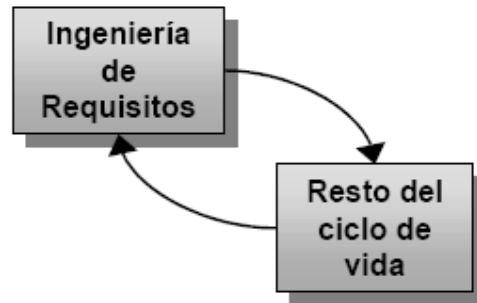
Ingeniería de Requerimientos (4). “Es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto”.

Se puede decir que la ingeniería de requerimientos puede considerarse como un proceso de *descubrimiento* y *comunicación* de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades.

Su situación dentro del ciclo de vida de desarrollo de software puede verse en la figura 2.6, aunque debe tenerse en cuenta que la ingeniería de requerimientos continúa durante todo el proceso de desarrollo Sawyer y Montoya(1999).



Figura 2.6: La Ingeniería de Requerimientos en el ciclo de vida de desarrollo del software



Fuente: Durán Toro, Amador (2004)

Importancia de la Ingeniería de Requerimientos

Los principales beneficios que se obtienen de la Ingeniería de Requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la RE.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).
- Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

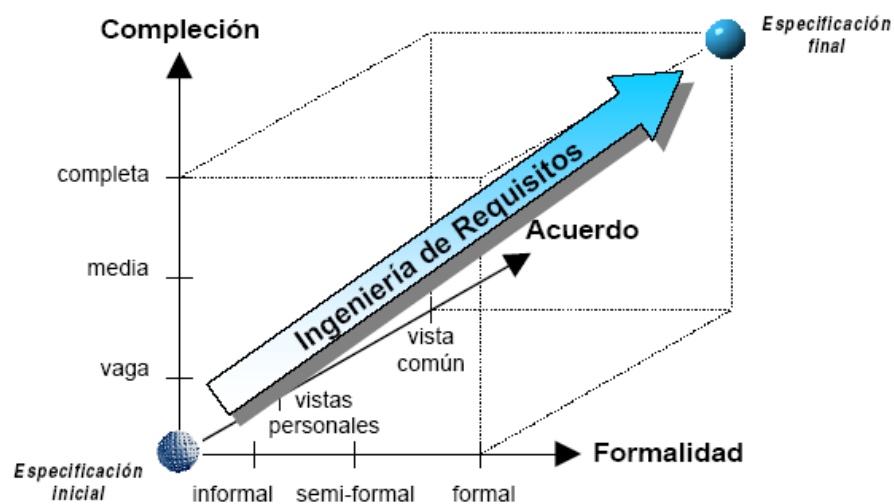


Las Dimensiones de la Ingeniería de Requerimientos

Una posible visión de la Ingeniería de Requerimientos es considerarla como un proceso de construcción de una especificación de requerimientos en el que se avanza desde especificaciones iniciales, que no poseen las propiedades oportunas, hasta especificaciones finales completas, formales y acordadas entre todos los participantes, Pohl, Klaus (1997). Por un lado están los factores psicológicos y cognitivos que afectan al grado de *compleción* del conocimiento sobre el sistema que se desea desarrollar, es decir, el llegar a conocer la totalidad de los requerimientos que debe satisfacer el sistema. Por otro lado, está el grado de *formalismo* de la representación del conocimiento sobre dichos requerimientos, teniendo en cuenta que un mayor grado de formalismo no implica necesariamente un mayor conocimiento Pohl, Klaus (1997).

Por último, como ya se comentó anteriormente, están los aspectos sociales, ya que al ser un proceso en el que participan personas con diferentes puntos de vista mediante algún tipo de *negociación* Boehm (1994). Estos factores pueden representarse como tres dimensiones Pohl(1997), de forma que durante el proceso de ingeniería de requisitos se avanza desde especificaciones incompletas, informales e individuales hacia la especificación ideal que sería completa, formal y acordada entre todos los participantes (ver figura 2.7).

Figura 2.7. Dimensiones de la Ingeniería de Requerimientos



Fuente: Pohl, Klaus (1997).



Así, las actividades de elicitación harían avanzar el proceso en las dimensiones de compleción y acuerdo, las actividades de análisis lo harían en compleción y formalidad y las actividades de validación en acuerdo y formalidad.

Personal involucrado en la Ingeniería de Requerimientos

Realmente, son muchas las personas involucradas en el desarrollo de los requerimientos de un sistema. Es importante saber que cada una de esas personas tienen diversos intereses y juegan roles específicos dentro de la planificación del proyecto; el conocimiento de cada papel desempeñado, asegura que se involucren a las personas correctas en las diferentes fases del ciclo de vida, y en las diferentes actividades de la IR.

No conocer estos intereses puede ocasionar una comunicación poco efectiva entre clientes y desarrolladores, que a la vez traería impactos negativos tanto en tiempo como en presupuesto. Los roles más importantes pueden clasificarse como sigue:

- **Usuario final:** Son las personas que usarán el sistema desarrollado. Ellos están relacionados con la usabilidad, la disponibilidad y la fiabilidad del sistema; están familiarizados con los procesos específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.
- **Usuario Líder:** Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requerimientos de las interfaces del sistema.
- **Personal de Mantenimiento:** Para proyectos que requieran un mantenimiento eventual, estas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías. Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.
- **Analistas y programadores:** Son los responsables del desarrollo del producto en sí; ellos interactúan directamente con el cliente.
- **Personal de pruebas:** Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas.



Son quienes van a validar si los requerimientos satisfacen las necesidades del cliente.

Otras personas que pueden estar involucradas, dependiendo de la magnitud del proyecto, pueden ser: administradores de proyecto, documentadores, diseñadores de base de datos, entre otros.

Puntos a considerar durante la Ingeniería de Requerimientos

Aunque la lista no está completa, se enumeran los puntos más importantes.

a) Objetivos del negocio y ambiente de trabajo

Aunque los objetivos del negocio están definidos frecuentemente en términos generales, son usados para descomponer el trabajo en tareas específicas. En ciertas situaciones IR se enfoca en la descripción de las tareas y en el análisis de sistemas similares. Esta información proporciona la base para especificar el sistema que será construido; aunque frecuentemente se añadan al sistema tareas que no encajan con el ambiente de trabajo planificado.

b) El nuevo sistema cambiará el ambiente de trabajo

Sin embargo, es muy difícil anticipar los efectos actuales sobre la organización. Los cambios no ocurren solamente cuando un nuevo software es implementado y puesto en producción; también ocurren cuando cambia el ambiente que lo rodea (nuevas soluciones a problemas, nuevo equipo para instalar, etc.). La necesidad de cambio es sustentada por el enorme costo de mantenimiento; aunque existen diversas razones que dificultan el mantenimiento del software, la falta de atención a la IR es la principal.

b) Frecuentemente la especificación inicial es también la especificación final

Lo que obstaculiza la comunicación y el proceso de aprendizaje de las personas involucradas; esta es una de las razones por las cuales existen sistemas inadecuados.



c) Punto de vista de los clientes

Muchos sistemas tienen diferentes tipos de clientes. Cada grupo de clientes tiene necesidades diferentes y, diferentes requerimientos tienen diferentes grados de importancia para ellos. Por otro lado, escasas veces tenemos que los clientes son los mismos usuarios; trayendo como consecuencia que los clientes soliciten procesos que causan conflictos con los solicitados por el usuario.

Diferentes puntos de vistas también pueden tener consecuencias negativas, tales como datos redundantes, inconsistentes y ambiguos.

El tamaño y complejidad de los requerimientos ocasiona desentendimiento, dificultad para enfocarse en un solo aspecto a la vez y dificultad para visualizar relaciones existentes entre requerimientos.

- **Barreras de comunicación**

La ingeniería de requerimientos depende de una intensa comunicación entre clientes y analistas de requerimientos; sin embargo, existen problemas que no pueden ser resueltos mediante la comunicación.

Para remediar esto, se deben abordar nuevas técnicas operacionales que ayuden a superar estas barreras y así ganar experiencia dentro del marco del sistema propuesto.

- **Evolución e integración del sistema**

Pocos sistemas son construidos desde cero. En la práctica, los proyectos se derivan de sistemas ya existentes. Por lo tanto, los analistas de requerimientos deben comprender esos sistemas, que por lo general son una integración de componentes de varios proveedores.

Para encontrar una solución a problemas de este tipo, es muy importante hacer planeamientos entre los requerimientos y la fase de diseño; esto minimizará la cantidad de fallas directas en el código.



- **Documentación de requerimientos**

Los documentos de ingeniería de requerimientos son largos. La mayoría están compuestos de cientos o miles de páginas; cada página contiene muchos detalles que pueden tener efectos profundos en el resto del sistema.

Normalmente, las personas se encuentran con dificultades para comprender documentos de este tamaño, sobre todo si lo leen cuidadosamente. Es casi imposible leer un documento de especificación de gran tamaño, pues difícilmente una persona puede memorizar los detalles del documento. Esto causa problemas y errores que no son detectados hasta después de haberse construido el sistema.

2.2 Antecedentes de la Investigación

Entre los aportes metodológicos para la ingeniería de requerimientos existen varios trabajos, citados a continuación:

2.2.1 La Metodología DoRCU (Documentación de Requerimientos Centrada en el Usuario) para la Ingeniería de Requerimientos.

Propuesta por Báez, M. Griselda y Barba Brunner, Silvia I. BAEZ(2001).

El cual consta de las siguientes etapas, Elicitación de requerimientos, análisis de Requerimientos, especificación de requerimientos, validación y certificación de los Requerimientos.

Los objetivos que se proponen para cada una de ellas son:

- **E licitación de Requerimientos.**

Esta es la etapa en donde se adquiere el conocimiento del trabajo del cliente / usuario, se busca comprender sus necesidades y se detallan las restricciones medioambientales. Como resultado de las acciones realizadas se tiene el conjunto de los requerimientos de todas las partes involucradas.



- **Análisis de Requerimientos.**

En esta etapa se estudian los requerimientos extraídos en la etapa previa a los efectos de poder detectar, entre otros, la presencia de áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes. El resultado de haber llevado a cabo las tareas que involucran estos términos puede, en más de una oportunidad, hacer que se deba regresar a la primera etapa, a los efectos de eliminar todas las inconsistencias y falencias que se han detectado. En esta etapa ya se realizan aproximaciones a un lenguaje técnico.

- **Especificación de Requerimientos**

Partiendo de lo elaborado en la etapa anterior tales como funciones, datos, requerimientos no funcionales, objetivos, restricciones de diseño/implementación o costos, e independientemente de la forma en que se realice, esta etapa es un proceso de descripción del requerimiento. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente.

- **Validación y Certificación de los Requerimientos.**

Esta etapa final se nutre de las anteriores y realiza la integración y validación final de lo obtenido en cada una de las etapas anteriores dando, como resultado final, el Documento de Requerimientos. Este documento no es uno solo sino que, como mínimo, existen dos que son isomórficos entre sí: uno destinado al cliente/usuario a los efectos de la certificación de los Requisitos y el otro técnico, orientado a nutrir las restantes etapas de la Ingeniería de Software. Y, al igual que en el caso anterior, su resultado puede ser la necesidad de retornar a la especificación e incluso a la elicitación; iterando entre etapas y sin perder contacto con el cliente/usuario.

2.2.2 Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información.

Propuesto por Durán Toro, Amador (2000)

En su trabajo describe un entorno metodológico para la ingeniería de requisitos de sistemas de información compuesto por:



- Un modelo de procesos iterativo en el que se identifican tres actividades principales: elicitación, análisis y validación de requerimientos.
- Una metodología para la elicitación de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, principalmente plantillas y patrones de requisitos, así como la posibilidad de introducir la reutilización en el proceso.
- Una metodología para el análisis de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, basadas en el estándar UML y con relaciones de rastreabilidad hacia los productos de la actividad anterior que facilita la reutilización de elementos complejos.
- Una metodología para la validación de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, basadas en la combinación de los walkthrough o recorridos con el prototipado de sistemas.
- Un prototipo de una herramienta CASE de ingeniería de requisitos como complemento a la metodología descrita.

2.2.3 System Quality Requirements Engineering (SQUARE) Methodology

Propuesto por Chen, Meter (2004)

Es una metodología de obtención y análisis de requisitos de seguridad que descompone el proceso en nueve pasos. Para cada uno describe su finalidad y en qué consiste, cuáles son las responsabilidades de los participantes, el input que toma, las técnicas que se pueden emplear y el resultado que produce.

Pasos de esta metodología son:

- Consenso sobre los conceptos y su definición (Consiste en la identificación de un conjunto de definiciones acordadas entre el cliente y el equipo SQUARE),



- Identificación de los objetivos de seguridad.
- Desarrollo de los artefactos necesarios para definir los requisitos de seguridad (diseños de la arquitectura, casos de uso, casos de uso incorrecto...).
- Valoración de riesgos.
- Selección de las técnicas para la obtención de los requisitos.
- Obtención de los requisitos.
- Clasificación de los requisitos.
- Priorización.
- Revisión de los requisitos.

2.3. Gestión del riesgo en la Ingeniería de Requisitos de un Proyecto Software

2.3.1. Conceptos Básicos :

Riesgo en un proyecto :

Es un evento o una condición incierta que, en caso de ocurrir, tiene un efecto positivo o negativo sobre los objetivos de un proyecto. Un riesgo tiene una causa y, si ocurre (evento de riesgo), una consecuencia (efecto).

Riesgos conocidos:

Son aquellos riesgos que han sido identificados y analizados durante la planificación del proyecto. Habitualmente se gestionan los riesgos con efecto negativo, es decir, aquellos que suponen una amenaza para el éxito del proyecto.

2.3.2. Gestión de Riesgos

La función de la gestión de riesgos del software es identificar, estudiar y eliminar las fuentes de riesgo antes de que empiecen a amenazar la finalización satisfactoria de un proyecto software.

De acuerdo con [Pressman, 2002], la administración o gestión de riesgos es un proceso iterativo que se aplica durante todo el proyecto y se desarrolla en cuatro etapas. Los resultados de la administración de riesgos deben ser documentados en un plan de administración de riesgos. La figura 2.8. refleja el procedimiento.



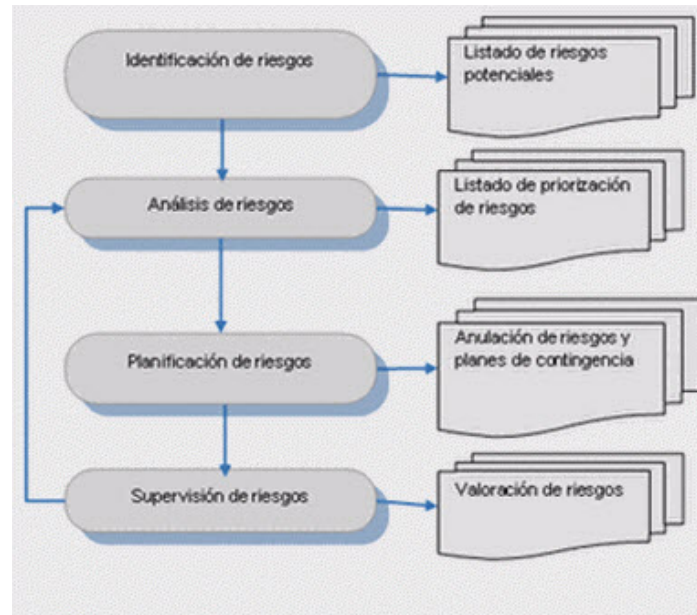


Figura 2.8. Procedimiento de Gestión del Riesgo

Un riesgo es aquel factor que influye negativamente en el éxito del proyecto. El riesgo en un proyecto de desarrollo de software incluye componentes técnicos y de conocimiento del mismo. Los temas de naturaleza organizacional constituyen los factores dominantes de los riesgos del proyecto, a la vez que son los que se tratan satisfactoriamente en menos de la tercera parte de los proyectos de desarrollo, entre ellos los conflictos entre departamentos, entre usuarios, el cambio del responsable ejecutivo del proyecto, volatilidad del personal, número de unidades de la organización implicadas y proyectos que involucran a múltiples proveedores.

Es premisa de esta propuesta que el riesgo se halla, de forma implícita, asociado a toda actividad. El riesgo acompaña a todo cambio porque implica elección e incertidumbre. Si a la vez que se inicia la actividad de elicitación de los requisitos del software a construir, se inicia la identificación de los riesgos asociados a los requisitos individuales y a grupos de ellos, será posible gestionarlos tempranamente para minimizarlos, evadirlos y controlarlos. El jefe o administrador de proyectos anticipa riesgos que pueden afectar al desarrollo o a la calidad de los requisitos y emprende acciones para evitarlos.



Esta actividad garantiza que, desde el inicio del proceso de desarrollo del software, se realicen las tareas encaminadas a garantizar la calidad del producto.

2.3.3. Etapas de la Gestión de Riesgos

Basado en el modelo dado por [Pressman, 2002], descrito anteriormente y representado en la figura 2.8, se define el procedimiento siguiente:

Paso 1: Identificación de riesgos. Problemas potenciales que pueden ocurrir en el proceso de Ingeniería de requisitos o en los requisitos, o en la Especificación de los Requisitos del Software, como de presupuesto, de personal, del usuario, de organización, técnicos, de comunicación u otros. Debe comenzar con el análisis de los riesgos genéricos, que constituyen una amenaza potencial para todos los proyectos de software, que puedan estar presentes en el proyecto en curso. Después se deben identificar los riesgos específicos, que implican un conocimiento profundo del proyecto, y están relacionados con el entorno de desarrollo, la tecnología, la experiencia y el tamaño del equipo.

Para los requisitos y la Especificación de los Requisitos del Software se debe comenzar esta tarea dando respuesta a la pregunta: ¿qué características especiales tiene este requisito, o grupo de ellos, que pueden estar amenazadas?

Un método probablemente efectivo para identificar riesgos es la creación de una lista de comprobación de elementos de riesgo. Esta lista debe centrarse en los riesgos relacionados con: tamaño del producto, impacto en el proyecto y en la organización, características del cliente, definición del proceso, el entorno de desarrollo, la tecnología a construir, el tamaño del equipo y la experiencia del personal.

En esta actividad del procedimiento se establece crear una lista de comprobación para los requisitos individuales y otra para la Especificación de los Requisitos. Los resultados se vierten en tablas para facilitar el análisis posterior. Los riesgos serán considerados a partir del comportamiento de sus características individuales y grupales (ambigüedad, claridad, completitud, consistencia, rastreabilidad, entre otras.). En dependencia de la intensidad de ese comportamiento de la característica de calidad del



requisito, podrá controlarse de forma efectiva el riesgo en el propio proceso de Ingeniería de requisitos, antes de que pase a la siguiente etapa del ciclo de vida del proyecto de desarrollo del software. De no gestionarse el riesgo en esta etapa inicial, puede hacerse difícil su control efectivo una vez iniciado el proceso de desarrollo. Desequilibrios en el comportamiento de las características de calidad de los requisitos se traducen en que la aparición de riesgos haga vulnerable el proyecto e impráctica la aplicación de cualquier plan de calidad, razón por la cual conviene su detección y tratamiento temprano. En esta situación, se recomienda la redefinición de los requisitos afectados (o la Especificación de los Requisitos), que es posible por iteraciones en este procedimiento.

Un ejemplo de la aplicación de este instrumento es:

Característica del requisito: ambigüedad. Pregunta: ¿es ambiguo el requisito?

Aplicada a todos los requisitos especificados, una tabla 1 puede mostrar el resultado.

Tabla 1 Identificación de riesgos.

ID requisito	Respuesta	Impacto	Prioridad	Componente
R1	SI	SI	Alta	Costo
R2	NO	NO	Baja	Tiempo

En este procedimiento se establece definir y listar los riesgos que pueden afectar al propio proceso de ingeniería de requisitos como aparece en las tablas 2, 3 y 4.

Tabla 2 Lista de riesgos y clasificación.

Riesgo	Tipo de riesgo	Descripción
Rotación de personal	Proyecto, producto y negocio	Personal con experiencia abandona el proyecto antes de que finalice
Cambios de requisitos	Proyecto y producto	Existencia de más cambios de requerimientos de los previstos inicialmente
Retrasos en la especificación	Proyecto y producto	Retrasos en las especificaciones de interfaces esenciales
Subestimación del tamaño	Proyecto y producto	El tamaño del requisito (la Especificación de los Requisitos, del proceso de Ingeniería de requisitos) se ha subestimado
Bajo rendimiento de la herramienta CASE	Producto	Las herramientas CASE que ayudan al proyecto no tienen el rendimiento y las funcionalidades esperadas



Tabla 3 Riesgos por requisitos.

ID Requisito	Tipo de Riesgo	Riesgos
R1	De personal	El personal no cuenta con los conocimientos requeridos para enfrentar la complejidad del requisito
		Miembros del equipo no disponibles en momentos críticos
	De requisitos	Cambios de requisitos que precisan modificaciones en el diseño
		Los clientes no comprenden el impacto de los cambios en los requerimientos
	De estimación	El tiempo requerido para desarrollar el proceso de ingeniería de requisitos está subestimado
De comunicación	El cliente no pueda participar en revisiones y en reuniones	

Tabla 4 Riesgos por tipos.

Tipo de riesgo	Posibles riesgos
Personal	Imposible contratar personal con los conocimientos requeridos.
Organizativos	La organización se reestructura y una nueva administración se responsabiliza del proyecto.
Herramientas	Las distintas herramientas CASE no están disponibles
Requerimientos	Cambios de requerimientos que precisan modificaciones en el diseño.
Estimación	El tamaño del sistema a desarrollar está subestimado.

Algunos riesgos que pueden afectar el desarrollo del proceso de Ingeniería de Requerimientos son:

- Sobrepasar los límites de los recursos asignados.
- Finalización fuera de plazos originales (a veces ni se finaliza).
- Pobre o descontrolada gestión de los requisitos.
- Incompatibilidad con el entorno.
- Riesgo más grave: que no se comprendan y no se satisfagan las necesidades de los usuarios.
- Problemas en la comunicación entre clientes y proveedores, entre usuarios, u otros grupos.

Paso 2: Evaluación de los riesgos. Determinar en qué indicador se verá reflejado que un problema se presente, se deben establecer puntos de referencia para cada riesgo, que



permita decidir si el riesgo, según su prioridad de atención, se sale del manejo aceptable. Obsérvese en la figura 2.9 un ejemplo de un punto de referencia.

Proyección de los riesgos : Consiste en determinar la probabilidad de que un riesgo ocurra y las consecuencias que puede tener, por ejemplo: incremento de costos, cancelación del proyecto, insatisfacción del cliente. Implica ordenar la lista de riesgos teniendo en cuenta la probabilidad de que ocurra y el impacto de cada riesgo. Se asigna el nivel de probabilidad, que puede ser alta, media o baja. Se valora el impacto (consecuencias) en cuanto al alcance (cuánto se afecta) y la duración (por cuánto tiempo se manifiesta).

En el análisis de riesgos se considera cada riesgo por separado y se valora en intervalos su probabilidad e impacto:

- Probabilidad del riesgo valorada como muy baja (<10%), baja (10-25%), media (25-50%), alta (50-75%) o muy alta (>75%)
- Efectos del riesgo valorados como catastrófico, serio, tolerable o insignificante

El resultado se registra en una tabla ordenada por la probabilidad o por el efecto del riesgo. Se decide del total, cuáles son los más importantes, considerados entonces como riesgos claves durante el proyecto -debe ser un número manejable.

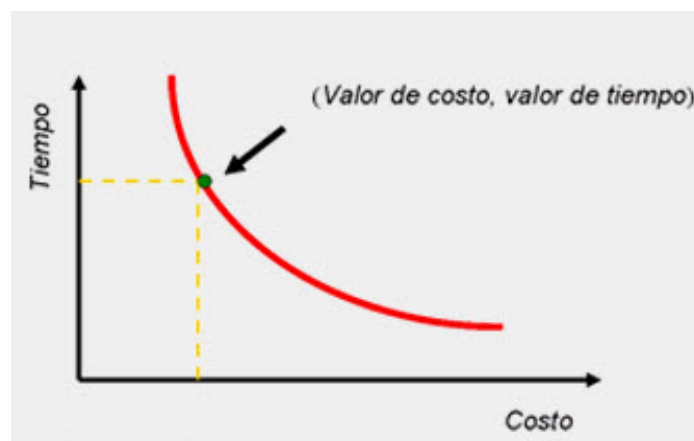


Figura 2.9. Ejemplo de punto de Referencia

Paso 3: Planificación de riesgos. Este paso tiene como objetivo desarrollar una estrategia para tratar los riesgos. Si el equipo de trabajo adopta un enfoque proactivo

frente al riesgo, evitarlo será siempre la mejor estrategia. Esto se consigue desarrollando los planes de reducción del riesgo y de contingencia.

En la planificación de riesgos se considera cada uno de los riesgos claves identificados y las estrategias para administrarlos, que vendrán dadas por el juicio y la experiencia del administrador del proyecto. Las estrategias de anulación intentan reducir la probabilidad de que surja el riesgo, las estrategias de disminución: intentan reducir el impacto del riesgo. Los planes de contingencia se elaboran para estar preparados por si el riesgo ocurre poder actuar con una estrategia determinada.

Paso 4: Supervisión de los riesgos. Consiste en hacer el plan de supervisión, dado el caso de que se acepte continuar con el proyecto. Indicar que acciones y decisiones se tomarán ante un problema que ya ha sido identificado, proyectado y evaluado.

La supervisión de riesgos valora cada uno de los riesgos identificados para decidir si es más o menos probable y cuándo han cambiado sus posibles efectos. Hay que controlar factores que pueden indicar cambios en la probabilidad y el impacto.

Gestión de Riesgos

- La tarea principal del administrador consiste en minimizar riesgos. Involucrar a todos los implicados/afectados y al equipo de desarrollo del proyecto en la identificación de los riesgos.
- El riesgo inherente en una actividad se mide en base a la incertidumbre que presenta el resultado de esa actividad. Las actividades con alto riesgo elevan los costos.
- Comunicar los riesgos a todos los niveles de la organización.
- El riesgo es proporcional al monto de la calidad de la información disponible. Cuanto menos información, mayor el riesgo.
- Monitorear constantemente los factores que propician la materialización de los riesgos y la efectividad de las acciones definidas encaminadas a su prevención y/o minimización.
- Definir plantillas o tablas de riesgos valorados para diferentes tipos de proyecto que puedan servir como punto de partida para un nuevo proyecto.



CAPÍTULO 3: ESTADO DEL ARTE

En esta apartado se comentan brevemente algunos de los modelos de procesos para ingeniería de requisitos existentes y que han servido de base para el modelo descrito en este trabajo.

3.1 El modelo de Pohl

El modelo de POHL (1997) es un modelo iterativo en el que se definen las cuatro actividades que pueden verse en la figura 3.1, en la que aunque el orden de realización de las actividades puede ser cualquiera, en POHL (1997) 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 3.1: Modelo de procesos de ingeniería de requisitos de Pohl



Fuente: POHL (1997)



Las características principales de las cuatro actividades de este modelo son las siguientes:

Elicitación de requisitos

Para Pohl, el objetivo de la elicitación es hacer explícito el conocimiento oculto sobre las necesidades de clientes y usuarios y el sistema a desarrollar de forma que todos los participantes en el problema sean capaces de entenderlo. Asumiendo las dimensiones de la ingeniería de requisitos propuestas también por Pohl esta actividad haría avanzar el proceso en la dimensión de compleción, ya que incrementa el conocimiento sobre el sistema a desarrollar.

Aunque implícitamente, en este modelo se asume que durante la realización de las actividades de elicitación 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

El objetivo de esta actividad es alcanzar acuerdos entre todos los participantes sobre los requisitos elicitados en la actividad anterior, avanzando en la dimensión de acuerdo del proceso. Para ello, Pohl propone 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

En esta actividad deben documentarse los requisitos elicitados y negociados, para lo que Pohl propone que no se utilice una sola notación sino tantas como sea necesario para que todos los participantes los entiendan, avanzando en la dimensión de formalidad del proceso.

Validación / verificación de requisitos

El propósito de esta actividad es comprobar que los requisitos documentados corresponden con las necesidades de los clientes y usuarios (*validación*) y comprobar que la especificación cumple los criterios de calidad oportunos (*verificación*), haciendo avanzar el proceso en las tres dimensiones descritas en [Pohl 1997].

3.2 El modelo espiral

El modelo espiral Sawyer(1997), Sommerville y Sawyer (1997), representado gráficamente en la figura 3.2, está basado en el modelo espiral de Boehm para la ingeniería de requisitos Boehm(1994) y el *Inquiry- Based Model* de Potts (1994). 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.

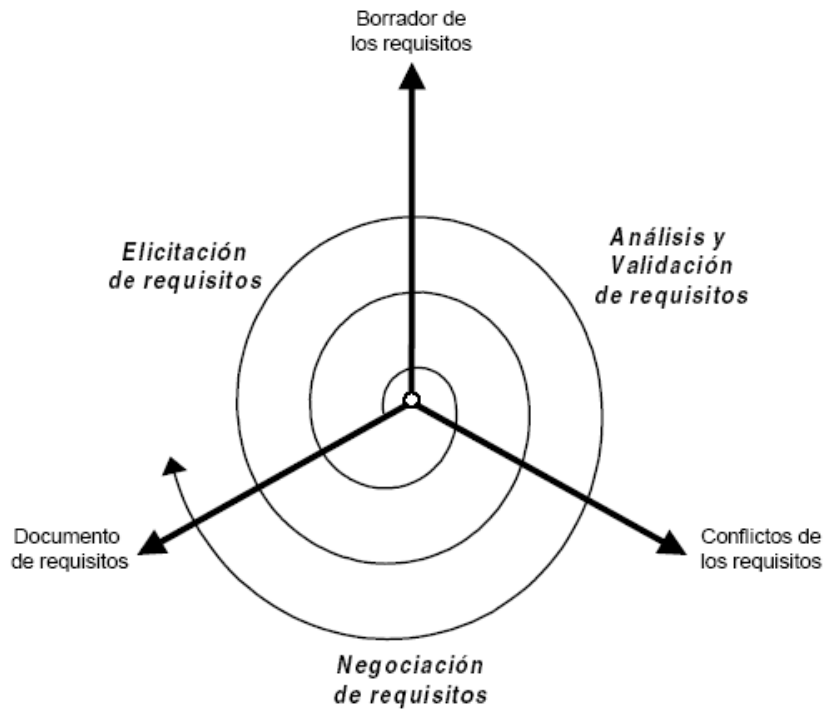
Elicitación de requisitos

En esta actividad, distintas fuentes de información como clientes, usuarios, expertos en el dominio, etc. son consultadas para entender el dominio del problema y



establecer los requisitos del sistema a desarrollar. Los requisitos elicidados puede que no sean completos y pueden estar expresados de forma vaga o no estructurada.

Figura 3.2: Modelo espiral de procesos de ingeniería de requisitos



Fuente: Sawyer (1997)

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 Sawyer(1997) 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 este trabajo, 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.

Negociación de requisitos

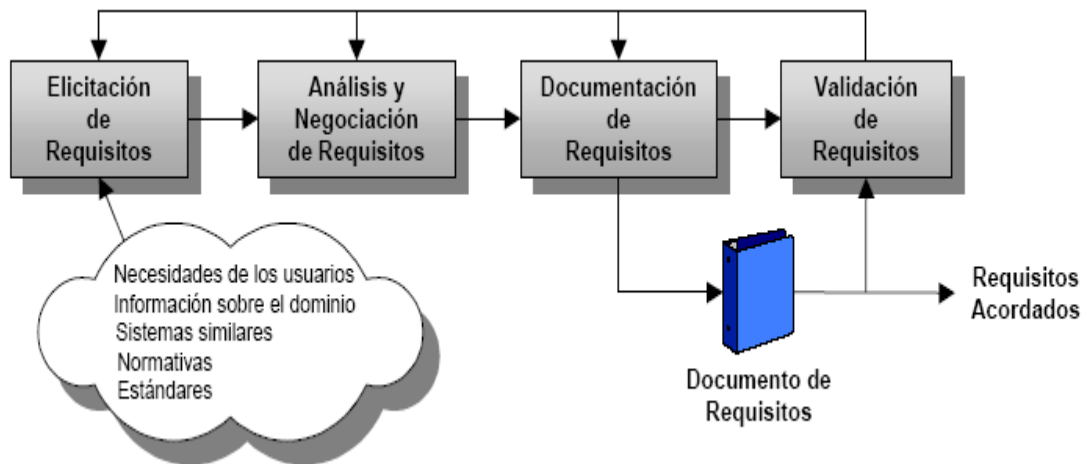
En esta actividad, los conflictos identificados durante el análisis deben resolverse llegando a acuerdos entre los participantes en el proceso, para lo que suele ser necesario elicitar nueva información.

3.3 El modelo SWEBOK

El proyecto SWEBOK (*Software Engineering Body of Knowledge*) es un proyecto conjunto del IEEE y de la ACM para producir un *cuero de conocimiento* sobre ingeniería de software que sienta las bases de dicha ingeniería como una profesión.

Dentro de las 10 áreas de conocimiento que han establecido, la novena corresponde a la ingeniería de requisitos Sawyer y Montoya(1999), dentro de cuya descripción se propone el modelo de procesos que puede verse en la figura 3.3, al que hay que añadir las actividades de gestión de requisitos que no se recogen en el gráfico.

Figura 3.3: Modelo SWEBOK de procesos de ingeniería de requisitos



Fuente: Sawyer y Montoya(1999).

Elicitación de requisitos

Esta actividad se considera como la primera que es necesario realizar para lograr entender los problemas que hay que resolver, y es fundamentalmente una actividad



humana. Para acometerla con ciertas garantías es necesario tener en cuenta los siguientes puntos:

- **Los objetivos:** pueden considerarse como los requisitos de alto nivel que deberá cumplir el sistema a desarrollar, por lo que es especialmente crítica su identificación en los comienzos del proceso.
- **Conocimiento del dominio:** es fundamental conocer el dominio del problema para poder inferir el conocimiento tácito que los participantes no suelen hacer explícito, además de para facilitar la comunicación.
- **Participantes (stakeholders):** es preciso identificar a todos los participantes que tengan algún tipo de interés en el desarrollo y tener en cuenta los puntos de vista de los distintos grupos.
- **Entorno operacional:** el sistema a desarrollar deberá funcionar en un entorno que es necesario conocer para poder identificar las necesidades de interoperabilidad con otros sistemas ya existentes.
- **Entorno organizacional:** además del entorno operacional, el sistema afectará al entorno organizacional, que es necesario conocer para evitar que surjan problemas con los procesos de negocio.

Análisis y negociación de requisitos

En esta actividad se pretende detectar y resolver los conflictos entre los requisitos, determinar los límites del sistema y cómo interactuará con su entorno y transformar los requisitos de usuario en requisitos software. En este modelo se propone que en esta actividad se clasifiquen los requisitos, se realicen modelos conceptuales y se negocien los conflictos detectados, tareas que se describen a continuación.

- Clasificación de requisitos
- Los autores consideran importante clasificar los requisitos para ayudar en las labores de negociación. Los criterios de clasificación son: capacidad / restricción, prioridad, coste / impacto, volatilidad / estabilidad y requisito de producto / requisito de proceso.



Modelado conceptual

El objetivo del modelado conceptual es ayudar a la comprensión del problema, aunque normalmente no puede evitarse iniciar el diseño de la solución. Existen múltiples técnicas de modelado conceptual, por lo que los autores proponen considerar factores como la naturaleza del problema, la experiencia del ingeniero de requisitos en el uso de una determinada técnica, si el cliente impone como requisito la utilización de una determinada metodología o la disponibilidad de metodologías y herramientas que soporten una determinada notación. Independientemente de la notación de modelado utilizada, los autores proponen realizar siempre un modelo de los límites del sistema para entender mejor el entorno y las interfaces necesarias.

Negociación de requisitos

La negociación de requisitos, también denominada resolución de conflictos, se ocupa de resolver los problemas que puedan surgir en los requisitos, bien porque haya peticiones por parte de clientes y usuarios que sean incompatibles, bien porque no se disponga de los recursos necesarios para la realización de ciertos aspectos del sistema, etc.

Para resolver estos conflictos es necesario consultar con todos los participantes afectados y registrar las decisiones tomadas y quién las tomó. Es necesario añadir que los autores asumen que los conflictos pueden aparecer no sólo durante el análisis, sino también durante la validación.

Documentación de requisitos

Los documentos de requisitos son el medio habitual para el registro y la comunicación de los requisitos. Los autores consideran deseable que los requisitos–C y los requisitos–D vayan en documentos separados.

Dentro de las actividades relacionadas con la documentación, los autores se remiten a la gestión de requisitos para aspectos como el control de versiones debido a los cambios en los requisitos.

Validación de requisitos

En esta actividad se debe comprobar los documentos de requisitos para detectar omisiones, conflictos y ambigüedades no detectadas en el análisis y también se debe



comprobar que los requisitos siguen las normas de calidad establecidas. En otras palabras, los autores combinan validación y verificación en una sola actividad.

Para realizar esta actividad se proponen revisiones técnicas de los requisitos basadas en listas de comprobación, el uso de prototipos, verificación formal de especificaciones, etc.

Gestión de requisitos

La gestión de requisitos, aunque no está reflejada en la figura 4.3, se realiza durante todas las actividades de ingeniería de requisitos. Su objetivo es gestionar los cambios y el mantenimiento de los requisitos para que representen el sistema que se va a desarrollar o que se ha desarrollado. Para conseguir estos objetivos, además de definir procedimientos para controlar los cambios y utilizar técnicas de gestión de configuración, los autores apuntan como factores importantes los atributos de los requisitos (identificadores, justificación, las fuentes del requisito, etc.) y la rastreabilidad para poder realizar análisis de impacto cuando se produce un cambio.

3.4 Ingeniería de software:

Según la definición del IEEE, citada por Lewis(1994) "**software** es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (SE del inglés *Software Engineering*) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la **Ingeniería de Software** es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" Cota(1994).

El **proceso de ingeniería de software** se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad" Jacobson (1999). El proceso **de desarrollo de**



software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" Jacobson (1999).

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Conviene hacer referencia a UML ya que será utilizado como base para la obtención de requerimientos funcionales.

3.5 UML (Unified Modeling Language)

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh (1999) Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE). UML ha puesto fin a las llamadas "guerras de métodos" que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una



herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común. Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

3.5.1 Diagramas de casos de uso

Los casos de uso tienen una representación gráfica en los denominados diagramas de casos de uso Booch(1999). En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Un ejemplo de este tipo de diagramas puede verse en la figura 3.6.

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

a) Actor:

Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.



Figura 3.4 Actor del caso de uso



b) Caso de uso

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Figura 3.5 Caso de uso



3.5.2 Relaciones entre casos de uso

A veces conviene establecer relaciones entre distintos casos de uso para simplificar su descripción. Las dos relaciones posibles y sus semánticas según Booch(1999) son las siguientes, cuya representación gráfica puede verse en el ejemplo de la figura 3.11.

a) Includes: Se dice que un caso de uso A *incluye* el caso de uso B, cuando B es una parte del caso de uso A, es decir, la secuencia de interacciones de B forma parte de la secuencia de interacciones de A.

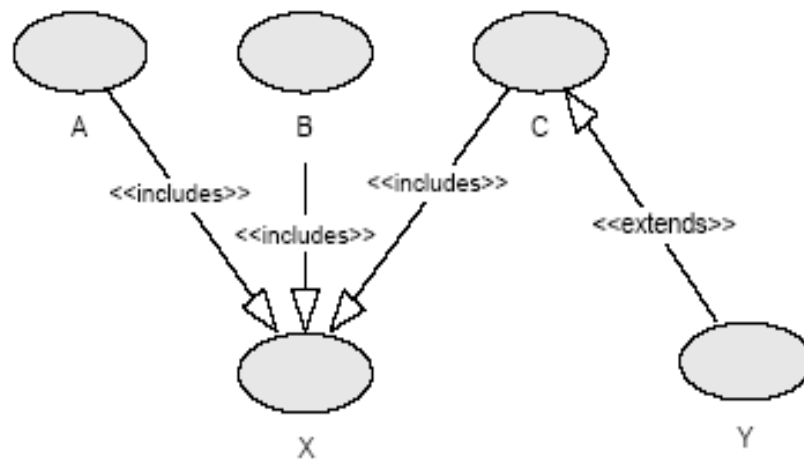
El caso de uso B se realiza siempre dentro del caso de uso A. Además, siempre que ocurre A ocurre también B, por lo que se dice que B es un caso de uso abstracto Jacobson(1999), Firesmith(1997).



Un caso de uso es abstracto si no puede ser realizado por sí mismo, por lo que sólo tiene significado cuando se utiliza para describir alguna funcionalidad que es común a otros casos de uso. Por otra parte, un caso de uso será *concreto* si puede ser iniciado por un actor y realizado por sí mismo.

Se suele utilizar esta relación cuando se detectan subsecuencias de interacciones comunes a varios casos de uso. Dichas subsecuencias comunes se sacan "factor común" de los casos de uso que las contienen y se les da forma de casos de uso que son incluidos por los casos de uso de los que se han "extraído". De esta forma se evita repetir las mismas subsecuencias de interacciones una y otra vez en varios casos de uso.

Figura 3.6: Representación gráfica de las relaciones includes y extens



Fuente: Boch (1999)

b) Extends: un caso de uso A extiende a otro caso de uso B cuando A es una subsecuencia de interacciones de B que ocurre en una determinada circunstancia.

En cierta forma, A completa la funcionalidad de B. El caso de uso A puede realizarse o no cuando se realiza el caso de uso B, según se den las circunstancias. Por otro lado, el caso de uso A puede ser un caso de uso abstracto o concreto, en cuyo caso puede ocurrir sin necesidad de que ocurra el caso de uso B.



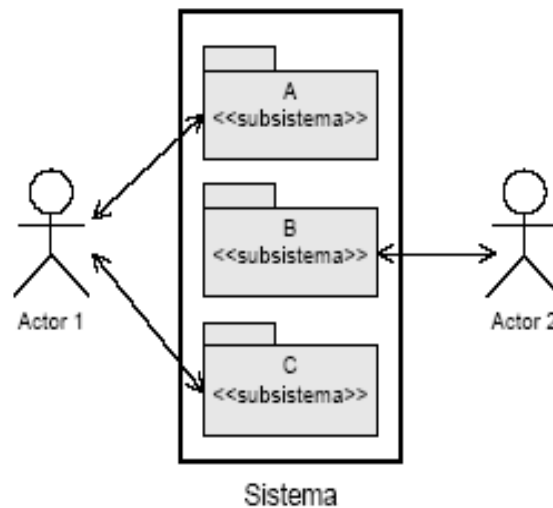
3.5.3 Organización de casos de uso

En la mayoría de sistemas, el número de casos de uso es lo suficientemente elevado como para que sea oportuno organizarlos de alguna forma en lugar de tener una lista plana por la que no es fácil navegar.

Una posible forma de organizar los casos de uso es recurrir a los paquetes descritos en la propuesta de UML Booch(1999). De esta forma, los casos de uso pueden organizarse en niveles, facilitando así su comprensión.

Cada paquete contiene a otros paquetes o a varios casos de uso. En el caso de que los casos de uso se agrupen por criterios funcionales, los paquetes que los agrupan pueden estereotiparse como subsistemas, tal como puede verse en el ejemplo de la figura 3.7.

Figura 3.7 Representación gráfica de los paquetes de casos de uso



3.5.4 Definición de términos básicos

- **Ckecklists.**- Listas de verificación que sirve de guía para el análisis o validación de requerimientos.
- **Domino del problema.**- Concerniente al entorno donde se da el problema.
- **End –User.**- Usuarios del sistema una vez que este haya sido entregado.



- **Etnografía.**-Técnica de las ciencias sociales.
- **Notación i*** .-propuesta por Eric Yu en la primera mitad de la década de los 90. i* permite expresar de forma clara y sencilla los objetivos de los actores que aparecen en los modelos y las dependencias entre ellos. Además, i* cuenta con una notación gráfica que permite tener una visión intuitiva y unificada del entorno modelado mostrando tales actores y dependencias.
- **Requerimientos Emergentes.**- Emergen cuando el sistema es puesto en uso, no puede ser especificado al inicio.
- **Requerimientos Funcionales.**- Definen las funciones que el sistema será capaz de realizar.
- **Requerimientos No Funcionales.**- Tienen que ver con características que de una u otra forma puedan limitar el sistema.
- **Requerimiento o Requisito.**- Necesidad o restricción que debe satisfacer el sistema.
- **Restricciones.**- Límites del sistema a construir.
- **Requisitos–C.**- Requerimientos del cliente.
- **Requisitos–D.**- Requerimientos del Desarrollador.
- **Stakeholders.**- Todas las personas relacionadas con el proyecto.
- **Usuario del sistema.**- Persona que interactúa con el sistema.
- **DoRCU .**- Documentación de Requerimientos Centrada en el Usuario.



CAPÍTULO 4: APORTE TEÓRICO

El presente estudio tiene por finalidad proponer una metodología para la Ingeniería de Requerimientos orientada al usuario. La ingeniería de requerimientos está formada por una serie de procesos bien diferenciados

Para el desarrollo de una metodología para la ingeniería de requerimientos aplicado a Empresas Clientes de GMD S.A. (MIRGMD) con su estructuración en etapas y tareas definidas para cada una de ellas, el autor se baso en el modelo de Pohl (1997), Modelo Espiral de Sawyer (1997) y el Modelo de SWEBOK, también en el trabajo de ingeniería de requerimientos de Duran Toro (2004), metodología DoRCU de Baez (2000) y en la experiencia del autor mismo.

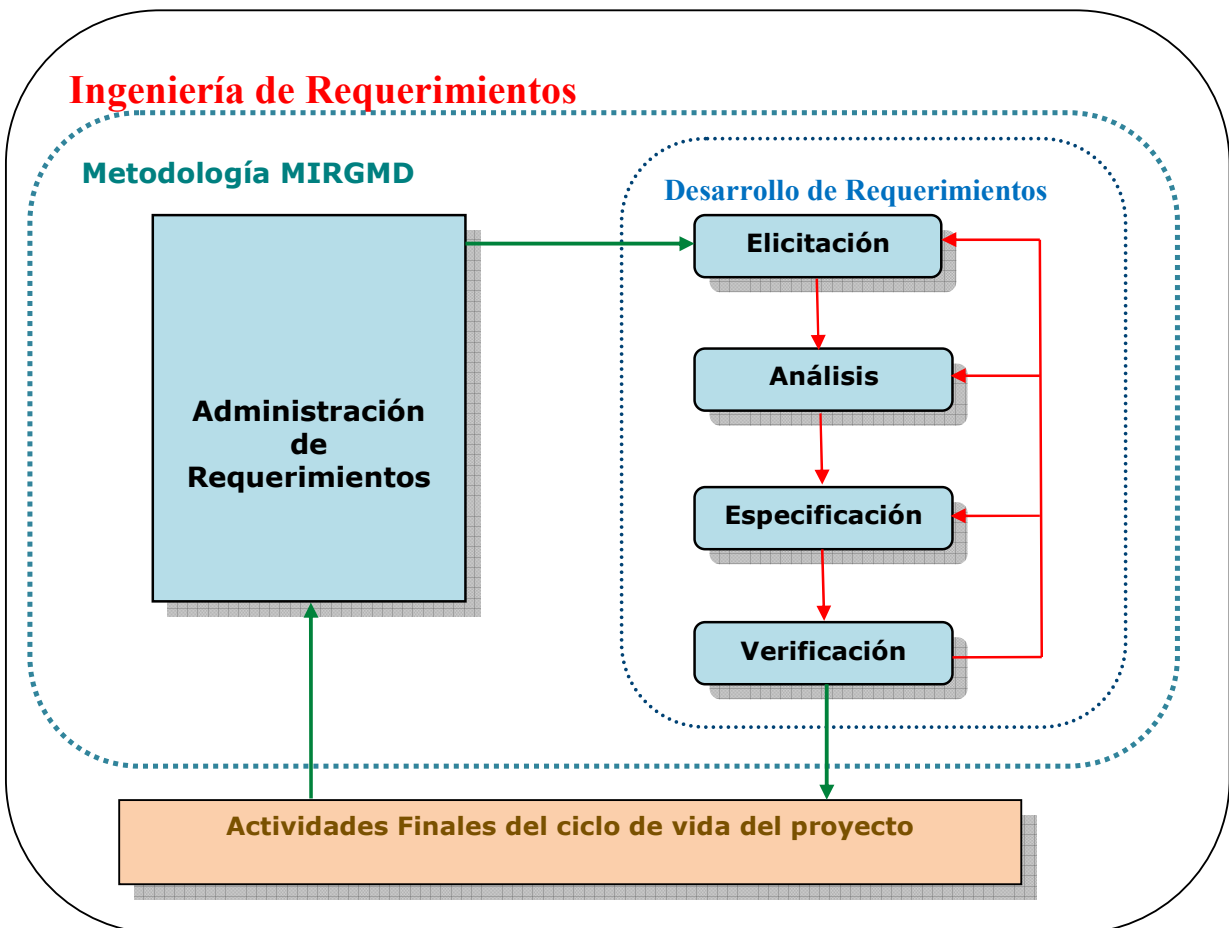


Figura 4.1: Procesos de la ingeniería de requerimientos



MIRGMD está estructurada en 5 etapas, Elicitación, Análisis, Especificación, Verificación y Administración de Requerimientos.

4.1 Etapas Propuestas

La Metodología propuesta completa se encuentra al final en el Anexo N° 1 de este trabajo, por lo que a continuación se presenta una breve descripción de las etapas en las cuales esta estructurada la metodología MIRGMD.

Etapas 1: Elicitación

Descripción

La Elicitación es el proceso mediante el que se identifican los ítems de información que determinan las características deseadas y las restricciones que deberá satisfacer el sistema software, que tendrán efectos satisfactorios para el usuario, en el ambiente donde se encuentra.

El proceso de elicitación de requerimientos trata de identificar la procedencia de los requerimientos y la manera en la que el ingeniero de software los puede recolectar. Es la primera etapa en la construcción de un entendimiento del problema que el software debe resolver.

Es fundamentalmente una actividad humana, y es en donde los stakeholders se identifican y comienzan a establecerse las relaciones entre el equipo desarrollador y el cliente. A este proceso también se lo conoce como “Captura de requerimientos”, “Descubrimiento de requerimientos” y “Adquisición de requerimientos”.

Este proceso, que merece una inversión por adelantado, debe ser llevado a cabo por un analista de negocios hábil que debe cumplir un rol crítico para la administración del proyecto forjando un enlace intrínseco entre las necesidades de los stakeholders, el sistema y los requerimientos de software.



Etapa 2: Análisis

Descripción

El Análisis de requerimientos es el proceso que trata, precisamente, de analizar la información recibida desde los usuarios, para distinguir las necesidades de tareas, los requerimientos funcionales, atributos de calidad, soluciones sugeridas, de información extraña. El análisis de requerimientos permite descubrir problemas en los requerimientos y encontrar soluciones mediante acuerdos entre los stakeholders.

Se define a la etapa de análisis de requerimientos como la actividad de transformar requerimientos informales en requerimientos técnicos mediante el aseguramiento de que los mismos reflejan los atributos de calidad de los requerimientos y que expresan las necesidades de los clientes. El análisis es una actividad iterativa. Los pasos del proceso deberán ser repetidos una cierta cantidad de veces, existe una constante comunicación entre la consulta de los clientes, los usuarios finales y los desarrolladores.

Es el proceso de razonamiento acerca de los requerimientos que han sido elicitados; involucra actividades como el examen de requerimientos en busca de conflictos o inconsistencias, la combinación de requerimientos relacionados y la identificación de requerimientos perdidos. Una vez recopilados los requisitos, el producto obtenido configura la base del análisis de requisitos.

Los riesgos asociados con cada requisito serán identificados y analizados. Se efectúan estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el costo del proyecto y en el plazo de entrega.

Etapa 3: Especificación

Descripción

La Especificación de requerimientos, es el proceso de grabado o el registro de los requerimientos en una o más formas, incluyendo el lenguaje natural y formal, representaciones simbólicas o gráficas. La especificación de los requerimientos es el paso en donde los resultados de la identificación de los requerimientos se “retratan”.



La especificación de requerimientos, es el modo habitual de guardar y comunicar requerimientos (en composición grupal). Como regla general, los requerimientos funcionales que describen qué es lo que el sistema y el software deben hacer deben estar separados de los requerimientos no funcionales que son los requerimientos de atributos de calidad especificados por el cliente, como son confiabilidad, seguridad y escalabilidad.

Los requerimientos, tradicionalmente, se representan en una forma puramente textual. Sin embargo, incrementalmente se está utilizando técnicas como construcción de modelos y prototipos, que demandan una descripción más detallada de los requerimientos.

Etapa 4: Verificación

Descripción

La Verificación de requerimientos, es el proceso de confirmación con el cliente o usuario del software que los requerimientos son válidos, correctos y completos. La validación es crítica para resaltar las disparidades entre las perspectivas de los stakeholders y para descubrir suposiciones que pueden quedar enmascaradas en la comunicación oral.

La verificación de los requerimientos permite para comprobar que el Documento de Requerimientos se ajusta a las necesidades de clientes/usuarios y otros implicados, es decir, es comprobar con los stakeholders que sus necesidades fueron adecuadamente interpretadas.

El proceso de validación de requerimientos se define como, un paso dentro de la ingeniería de requerimientos para establecer los criterios y técnicas para asegurarse que el software cumple con los requerimientos. El cliente y los desarrolladores deben llegar a un acuerdo sobre el criterio de aceptación y las técnicas o métricas a ser utilizadas durante el proceso de validación, como por ejemplo, la ejecución de un plan de testing para determinar que el criterio ha sido alcanzado.



Etapa 5: Administración de Requerimientos

Descripción

La Administración de requerimientos incluye todas las actividades para mantener la integridad, exactitud y difusión de los acuerdos de los requerimientos durante la vida del proyecto, ya que cambios a los requerimientos ocurren mientras los requerimientos están siendo elicitados, analizados, validados y después que el sistema es puesto en servicio.

No es suficiente para las organizaciones recolectar requerimientos desde múltiples stakeholders e incorporarlos independientemente dentro de un sistema. Es necesario administrar estos de manera simultánea, además alguien, o algún equipo, debe ser responsable de administrar esos requerimientos a lo largo del ciclo de vida de manera tal de mantener la visibilidad y control del proceso de entrega del software.

El acuerdo de los requerimientos es el puente que une el desarrollo de los requerimientos y la administración de los mismos. La administración de los requerimientos incluye todas las actividades que mantienen la integridad y exactitud de los requerimientos a medida que el proyecto progresa. Como se muestra en la Fig. 4.2 la administración de los requerimientos enfatiza:

- Control de los cambios a los requerimientos que están sobre una línea base.
- Mantener los planes del proyecto actualizados de acuerdo con los requerimientos.
- Control de versiones tanto de requerimientos individuales como del documento de requerimientos.
- Manejar las relaciones entre requerimientos, y los links y dependencias entre los requerimientos individuales y otros entregables del proyecto.
- Monitorear el estado de los requerimientos sobre una línea base.



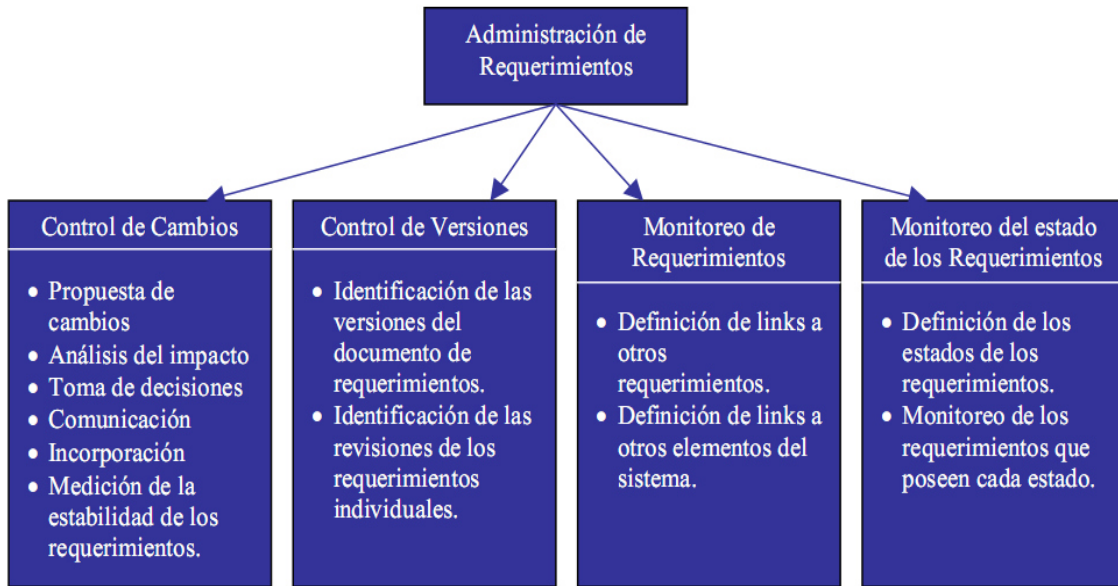


Figura 4.2: Actividades principales de la administración de requerimientos



CAPÍTULO 5: APORTE PRÁCTICO

Este trabajo es importante porque contribuirá a mejorar el producto software reduciendo o eliminando de esta manera los costos en tiempo y recursos generados por una mala gestión de requerimientos en los proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Permitiendo esto a su vez mejorar la calidad de ciclo de vida de desarrollo de software en las empresas del sector.

También es importante porque de esta manera brindara un aporte para el potencial exportador de las empresas de software en el Perú actualmente creciente según CUANTO (2004).de desarrollo de software en el país.

Para el desarrollo de una metodología para la ingeniería de requerimientos aplicado a Empresas Clientes de GMD S.A. (MIRGMD) con su estructuración en etapas y tareas definidas para cada una de ellas, el autor se baso en el modelo de Pohl (1997), Modelo Espiral de Sawyer (1997) y el Modelo de SWEBOK, también en el trabajo de ingeniería de requerimientos de Duran Toro (2004), metodología DoRCU de Baez (2000) y en la experiencia del autor mismo.

MIRGMD está estructurada en 5 etapas, Elicitación, Análisis, Especificación, Verificación y Administración de Requerimientos.



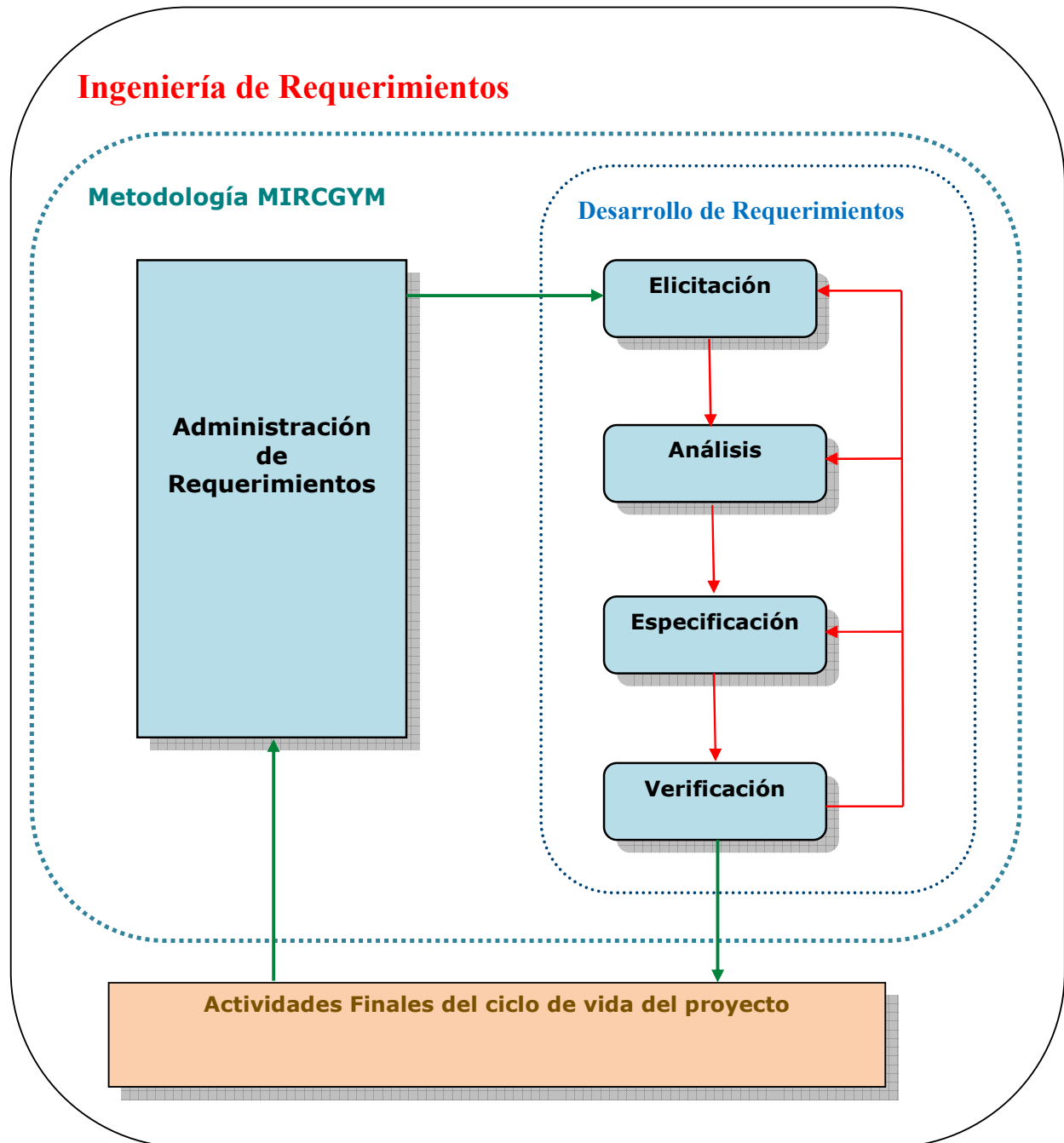


Figura 5.1: Procesos de la ingeniería de requerimientos MIRGMD



5.1 Implementación de Etapas

A continuación se implementará las etapas en las cuales está estructurada la metodología MIRGMD. Cada etapa contiene una descripción, objetivo, las tareas a realizar, los entregables a proporcionar y las técnicas recomendadas.

Etapas 1: Elicitación

Descripción

La Elicitación es el proceso mediante el que se identifican los ítems de información que determinan las características deseadas y las restricciones que deberá satisfacer el sistema software, que tendrán efectos satisfactorios para el usuario, en el ambiente donde se encuentra. La elicitación es la habilidad de trabajar en colaboración con los clientes y/o representantes de ellos para descubrir las necesidades actuales del producto y acordar la visión y las metas del proyecto propuesto. Ya que las personas que van a interactuar con el sistema, ya sea este empaquetado o a medida, no demandan exactamente las mismas funciones, parte del proceso de elicitación es la identificación temprana de las diferentes clases de usuarios y sus características.

El proceso de elicitación de requerimientos trata de identificar la procedencia de los requerimientos y la manera en la que el ingeniero de software los puede recolectar. Es la primera etapa en la construcción de un entendimiento del problema que el software debe resolver. Es fundamentalmente una actividad humana, y es en donde los stakeholders se identifican y comienzan a establecerse las relaciones entre el equipo desarrollador y el cliente. A este proceso también se lo conoce como “Captura de requerimientos”, “Descubrimiento de requerimientos” y “Adquisición de requerimientos”.

Este proceso, que merece una inversión por adelantado, debe ser llevado a cabo por un analista de negocios hábil que debe cumplir un rol crítico para la administración del proyecto forjando un enlace intrínseco entre las necesidades de los stakeholders, el sistema y los requerimientos de software. Dentro de las actividades principales definidas en la elicitación de requerimientos se encuentran: la identificación de los stakeholders, la elicitación de los requerimientos funcionales, la identificación de



restricciones, la definición de los escenarios y la identificación de los requerimientos de calidad.

Objetivo

Con esta etapa se pretende obtener una especificación preliminar detallada de las necesidades de los usuarios del software a desarrollar.

Tareas

- Obtener información sobre el problema.
- Obtener el dominio del problema, el dominio de la aplicación y sistemas existentes.
- Comprender el contexto del negocio.
- Comprender las necesidades y restricciones de los usuarios finales del Sistema.
- Preparar y realizar las sesiones de elicitación/ negociación.
- Identificar/ revisar los objetivos del sistema.
- Identificar/ Revisar los requerimientos funcionales.
- Identificar/revisar los requerimientos no funcionales.

Entregables

- Un borrador del documento de requerimientos conteniendo la descripción del problema, dominio del problema, objetivos, actores, requerimientos funcionales, no funcionales identificados y los conflictos encontrados.

Técnicas

- Entrevistas (Abiertas y Cerradas).
- Escenarios.
- Observación de tareas habituales.



- Reuniones (Tormenta de Ideas, JAD).
- Prototipos.
- Reutilización de requisitos (Familia de Sistemas).
- Análisis de Protocolos.
- Joint Application Design (JAD),.
- Casos de uso.
- Observaciones y Análisis Social.
- Plantillas y patrones lingüísticos.

Etapa 2: Análisis

Descripción

El Análisis de requerimientos es el proceso que trata, precisamente, de analizar la información recibida desde los usuarios, para distinguir las necesidades de tareas, los requerimientos funcionales, atributos de calidad, soluciones sugeridas, de información extraña. El análisis de requerimientos permite descubrir problemas en los requerimientos y encontrar soluciones mediante acuerdos entre los stakeholders.

Se define a la etapa de análisis de requerimientos como la actividad de transformar requerimientos informales en requerimientos técnicos mediante el aseguramiento de que los mismos reflejan los atributos de calidad de los requerimientos y que expresan las necesidades de los clientes. El análisis es una actividad iterativa. Los pasos del proceso deberán ser repetidos una cierta cantidad de veces, existe una constante comunicación entre la consulta de los clientes, los usuarios finales y los desarrolladores.

Es el proceso de razonamiento acerca de los requerimientos que han sido elicitados; involucra actividades como el examen de requerimientos en busca de conflictos o inconsistencias, la combinación de requerimientos relacionados y la identificación de requerimientos perdidos. Una vez recopilados los requisitos, el



producto obtenido configura la base del análisis de requisitos. Es corriente en clientes y usuarios solicitar más de lo que puede realizarse, consumiendo recursos de negocios limitados. También es relativamente común en clientes y usuarios el proponer requisitos contradictorios, argumentando que esa versión es esencial por necesidades especiales.

El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Los clientes, usuarios y el resto de intervinientes deberán clasificar sus requisitos y discutir los posibles conflictos según su prioridad. Los riesgos asociados con cada requisito serán identificados y analizados. Se efectúan estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el costo del proyecto y en el plazo de entrega.

Objetivo

El objetivo del análisis es descubrir problemas, incompletezas e inconsistencias en los requerimientos elicitados. El de la negociación es el de discutir los problemas encontrados para lograr un acuerdo entre todos los stakeholders.

Los requisitos se agrupan por categorías y se organizan en sub conjuntos, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad, y se clasifican en base a las necesidades de los clientes/usuarios.

Tareas

- Verificar las necesidades.
- Verificar si son consistentes y Completos.
- Verificar su factibilidad.
- Discusión de Requerimientos.
- Priorización de requerimientos.
- Concordancia de requerimientos.



Entregables

- Compromisos acordados en base a los requerimientos en conflicto luego de negociados.

Técnicas

- Checklist de Análisis(Listas de Verificación)
- Matriz de Interacción de requerimientos.

Etapa 3: Especificación

Descripción

La Especificación de requerimientos, es el proceso de grabado o el registro de los requerimientos en una o más formas, incluyendo el lenguaje natural y formal, representaciones simbólicas o gráficas. La especificación de los requerimientos es el paso en donde los resultados de la identificación de los requerimientos se “retratan”.

La especificación de requerimientos, es el modo habitual de guardar y comunicar requerimientos (en composición grupal). Como regla general, los requerimientos funcionales que describen qué es lo que el sistema y el software deben hacer deben estar separados de los requerimientos no funcionales que son los requerimientos de atributos de calidad especificados por el cliente, como son confiabilidad, seguridad y escalabilidad.

Los requerimientos, tradicionalmente, se representan en una forma puramente textual. Sin embargo, incrementalmente se está utilizando técnicas como construcción de modelos y prototipos, que demandan una descripción más detallada de los requerimientos.

Dado que los usuarios muchas veces no son capaces de pensar en todas las situaciones posibles en las cuales el software puede ser utilizado, es tarea de los desarrolladores documentar los requerimientos desde un punto de vista que admita el



testing. En este proceso se darán a conocer posibles situaciones que no se hubiesen tenido en cuenta la primera vez.

Objetivo

El objetivo de esta actividad es obtener un documento de los requerimientos en donde se registre el proceso de captura de requerimientos, y que defina de forma completa, precisa y verificable, los requisitos que debe cumplir el sistema, tanto funcionales como no funcionales.

El objetivo de la documentación es servir de base para la futura operativa del proyecto tanto para clientes como para desarrolladores. Debe abordar la descripción de lo que hay que desarrollar, no el cómo ni el cuándo, no debe incluir requisitos innecesarios, no solicitados por el cliente, ni incluir detalles sobre el diseño del sistema.

Tareas

- En esta etapa se caracteriza por enfocarse en la construcción del documento de requerimientos.

Entregables

- Documento de borrador final de los requerimientos incluyendo además los conflictos negociados si existiesen.

Técnicas

El documento de requerimientos debe ser adaptado del Standard IEEE/ANSI 830-1993 de acuerdo al tipo de sistema a desarrollar.

Etapa 4: Verificación

Descripción

La Verificación de requerimientos, es el proceso de confirmación con el cliente o usuario del software que los requerimientos son válidos, correctos y completos. La validación es crítica para resaltar las disparidades entre las perspectivas de los stakeholders y para descubrir suposiciones que pueden quedar enmascaradas en la



comunicación oral. La verificación de los requerimientos permite para comprobar que el Documento de Requerimientos se ajusta a las necesidades de clientes/usuarios y otros implicados, es decir, es comprobar con los stakeholders que sus necesidades fueron adecuadamente interpretadas.

El proceso de validación de requerimientos se define como, un paso dentro de la ingeniería de requerimientos para establecer los criterios y técnicas para asegurarse que el software cumple con los requerimientos. El cliente y los desarrolladores deben llegar a un acuerdo sobre el criterio de aceptación y las técnicas o métricas a ser utilizadas durante el proceso de validación, como por ejemplo, la ejecución de un plan de testing para determinar que el criterio ha sido alcanzado. Dentro del proceso de validación se pueden nombrar las siguientes actividades:

- Ejecutar una evaluación: para asegurar que los requerimientos de calidad han sido alcanzado apropiadamente, se han descubierto las inconsistencias entre los requerimientos, se han identificado las redundancias, y el impacto de los requerimientos derivados.
- Verificar la trazabilidad: verificar que todos los requerimientos se conectan con un requerimiento de más alto nivel, verificar que todos los requerimientos de alto nivel se conectan con un requerimiento formalizado e identificar requerimientos que no se encuentran anidados en otros, es decir que no poseen un “requerimiento padre”.
- Documentar los hallazgos: creación de reportes que contenga los requerimientos junto a una definición más acabada.
- Acuerdo: el documento formalizado de requerimientos será utilizado como un acuerdo entre el cliente y el desarrollador.
- Establecer una línea base de los requerimientos: poner bajo administración de la configuración al documento de requerimientos.

Objetivos

Los objetivos de esta actividad es permitir demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes.



En esta actividad se comprueba la consistencia, completitud, corrección, y precisión del documento, así como el descubrimiento de problemas en él antes de comprometer recursos en su implementación. La Verificación trabaja con un borrador final del documento de requerimientos, con requerimientos negociados y acordados. La Verificación de requerimientos es importante pues de ella depende que no existan elevados costos de mantenimiento para el software desarrollado.

Tareas

- Revisar el Documento de Requerimientos.
- Cerrar la versión de los requerimientos.

Entregables

- Documento de Requerimientos final validado.

Técnicas

- Inspección de requerimientos.
- Checklist de Inspecciones (Listas de verificación de inspecciones).
- Otras técnicas como desarrollo de Prototipos, teste de requerimientos, desarrollo del manual de usuario.

Etapas 5: Administración de Requerimientos

Descripción

La Administración de requerimientos incluye todas las actividades para mantener la integridad, exactitud y difusión de los acuerdos de los requerimientos durante la vida del proyecto, ya que cambios a los requerimientos ocurren mientras los requerimientos están siendo elicitados, analizados, validados y después que el sistema es puesto en servicio.



No es suficiente para las organizaciones recolectar requerimientos desde múltiples stakeholders e incorporarlos independientemente dentro de un sistema.

Es necesario administrar estos de manera simultánea, además alguien, o algún equipo, debe ser responsable de administrar esos requerimientos a lo largo del ciclo de vida de manera tal de mantener la visibilidad y control del proceso de entrega del software.

El acuerdo de los requerimientos es el puente que une el desarrollo de los requerimientos y la administración de los mismos. La administración de los requerimientos incluye todas las actividades que mantienen la integridad y exactitud de los requerimientos a medida que el proyecto progresa.

Objetivo

Esta etapa tiene como objetivo: identificar, controlar y seguir los requerimientos y sus cambios durante la vida del proyecto, esto con la ayuda de la herramienta OSRMT recomendada para esta metodología.

Ayuda al equipo de proyecto a rastrear los requisitos según las características de los mismos, el código fuente relacionado, dependencia entre requisitos, subsistemas e interfaces internas y externas; de forma que pueda identificarse con rapidez para entender cómo afectará una modificación diferentes aspectos del sistema a construir.

Tareas

- Identificación de problemas de los requerimientos.
- Análisis de los cambios propuestos.
- Implementación del cambio.

Entregables

- Nueva versión de los requerimientos.
- Nueva versión del documento de requerimientos.



Técnicas

- Tablas de rastreabilidad.
- Uso de la Herramienta Case OSRMT.

La finalidad de recomendar una herramienta Open Source como lo es OSRMT es que no requiere licencia el cual se enmarca en la orientación de esta metodología de captura de requerimientos, y nos permitirá reducir costos al no adquirir costosas licencias.

Para la elección de la herramienta case Open Source, el autor se baso en el informe de INCOSE[2006], donde se evalúan varias herramientas de gestión de ingeniería de requerimientos en base a ciertas características como rastreabilidad, asignación de prioridades y otros, concluyendo que OSRMT (Open Source Requirements Management Tool) cubre con la mayoría de características evaluadas, por tal motivo se considero a utilizarse para esta etapa. La última versión de OSRMTV. 1.5 con fecha (24/03/2007) es la que se usara para todo el ciclo de vida del software.

CASO DE ESTUDIO

La metodología MIRGMD fue utilizada en un proyecto de Ingeniería de Software con el fin de validar su aporte en la Ingeniería de Requerimientos aplicado a Empresas Clientes de GMD S.A. y para uno de sus Clientes el cual denominaremos “La Empresa Contratista”.

5.2. LA EMPRESA GMD S.A.

GMD es una empresa líder en la provisión de soluciones de tecnología de la información, cuenta con un trayectoria de más de 20 años que le ha permitido alcanzar una exitosa experiencia en la integración de tecnologías innovadoras, aportando durante este tiempo a sus clientes soluciones de tecnología de la información (IT) que mejoran su productividad, su relación con clientes y proveedores.

Éste éxito, se basa en una relación de aporte de valor, en donde usted se focaliza



en el giro de su negocio y GMD actúa como su socio especialista en Tecnología de la información; diseñando, implementando operando y/o administrando la solución tecnológica; y en muchas oportunidades haciéndose responsable de procesos integrales que pueden incluir infraestructura, recursos humanos, aplicaciones, supervisión y auditoría.

GMD entiende las necesidades y le provee una amplia gama de soluciones de negocios innovadoras, flexibles y escalables para los sectores: Industria y Comercio, Banca y Finanzas, Gobierno y Servicios Públicos que van desde la provisión de equipos de cómputo y comunicaciones, pasando por la integración de sistemas y soluciones de negocios, hasta la completa externalización de procesos y formación de sociedades comerciales.

Mercado Objetivo

El mercado objetivo de GMD S.A. es el sector corporativo, el cual reúne a las 200 empresas privadas e instituciones públicas más grandes del país de los sectores productivos.

Hace más de 8 años inició el proceso de expansión regional con la ejecución de importantes proyectos en los mercados de Colombia, Venezuela, Ecuador, Bolivia y Centro América.

Sector	Subsector
Industria y comercio	Minería, Energía, Industria, Bebidas y Alimentos, Comercio, Agroindustria, Pesca y Retail.
Banca y Finanzas	Banca, Seguros, AFP y Bursátil.
Gobierno	Educación, Salud, Emp. Públicas, Fuerzas Armadas, Ministerios, Gobierno Central, Regulación.
Servicios Públicos	Telecom.

Productos y Servicios

En GMD se entiende las necesidades del Cliente y se provee una amplia gama de soluciones de negocios innovadoras, flexibles y escalables que van desde la provisión de equipos de cómputo y comunicaciones, pasando por la integración de sistemas y soluciones de negocios, hasta la completa externalización de procesos.



5.2.1. Outsourcing de Tecnología

El servicio de outsourcing se ha convertido hoy en día en una herramienta muy eficaz que permite focalizar los recursos de su empresa en el giro de su negocio sin distracciones, así como alcanzar la modernidad, eficiencia y competitividad que le exige el mercado actual.

A través del outsourcing de tecnología GMD ayuda a conseguir los objetivos del negocio, encargándonos de sus necesidades de tecnología y garantizando en todo momento los más altos niveles de servicio, permitiendo que su negocio mejore en sus procesos internos y vuelva a dirigir sus recursos hacia las iniciativas estratégicas y las actividades que agregan valor a sus clientes.

Servicios

Mediante el outsourcing de Tecnología de GMD y a través del Data Center Services se provee servicios de alta especialización en tecnología, para satisfacer las necesidades de los Clientes

- Servicio de Hosting.
- Servicio de Housing.
- Servicios de Disaster Recovery.
- Servicio de Respaldo (Backup).
- Servicios de Almacenamiento.

5.2.2. Software Factory

La solución de Software Factory de GMD, es un modelo de servicios que permite ayudar a gestionar el Mantenimiento Correctivo, Evolutivo y Desarrollo de los sistemas de información. Se basa en Acuerdos de Niveles de Servicio (ANS) y Modelos de Estimación Claros, que garantiza la calidad del servicio, en términos de tiempo de respuesta, productividad y eficiencia.

La principal ventaja de este servicio, es delegar funciones y responsabilidades relacionadas a la tecnología de la información a un experto como GMD, además de



mejorar el manejo de riesgo. Sumado a esto se encuentran múltiples beneficios adicionales:

- **Planeación financiera mejorada:** Conocer de **antemano el costo exacto** le permitirá administrar mejor su presupuesto.
- **Foco de los esfuerzos y recursos de la empresa en sus negocios centrales:** Dejar en nuestros manos los problemas tecnológicos y concéntrese en el core business de su compañía.
- **Ciclos de implementación o instalación más cortos:** Gracias a nuestra experiencia tendrá la aplicación terminada en menos tiempo.
- **Acceso a conocimientos altamente calificados a un menor costo:** Uno de nuestros mayores activos es la experiencia ganada durante años con múltiples proyectos ligados al proceso productivo de software.
- **Disponibilidad de asesoría continua:** Usted contará con asistencia especializada durante todo el ciclo de vida del proyecto.
- **Cuotas reducidas de licencia y mantenimiento:** Nosotros proveemos las herramientas de apoyo al desarrollo software, lo que significa un ahorro que se traslada al costo final del proyecto.
- **Reducción de costos fijos:** Como proveedores de Software Factory, nuestros costos fijos se prorratan en múltiples proyectos, transfiriendo estas economías a nuestros clientes.
- **Mejoramiento Continuo:** Orientado a mejorar la productividad.
- **Contención y reducción total de defectos:** En las pruebas de aceptación, producción y reducción del esfuerzo de rehacer el trabajo.
- **Acuerdo de Niveles de Servicios (ANS):** Basados en objetivos de mejoras acordados.

Soluciones :

Por medio de la Software Factory de GMD se cuenta con los siguientes servicios:

- Fábrica de Software
- Desarrollo a Medida:
- Control de Calidad y Testing

Servicios:

- Control de versiones.



- Control de calidad del código fuente.
- Pruebas de funcionalidad.
- Control de pases a producción.

5.2.3. Outsourcing de Procesos

Por medio del Outsourcing de procesos de GMD usted puede mejorar su productividad, reducir costo e incrementar el valor de su empresa dejando en manos de los especialistas el diseño, implementación y operación de sus procesos, logrando convertir a su organización en un negocio de "Alto Rendimiento". Usted centra sus recursos y energía en el cumplimiento de los objetivos estratégicos y la actividad principal de su empresa y nosotros combinamos personas, procesos, metodología y tecnología para ayudarlo a conseguir máxima eficiencia y ventajas competitivas.

Con el Outsourcing de Proceso (BPO) de GMD, se obtiene beneficios como:

- Aumentar la rentabilidad de su negocio
- Aumentar la Calidad del Servicio
- Controlar los Procesos

Soluciones

A través de nuestro outsourcing de procesos usted puede acceder a una amplia gama de soluciones de negocio:

- Procesos Operativos
- Procesos de Soporte

5.2.4. Outsourcing de Servicios de Aplicación

Los servicios de Outsourcing de Aplicaciones, permite gestionar, mantener y mejorar las aplicaciones del cliente, con un enfoque de servicio integral y fuerte compromiso con los resultados, basado en la medición y el control; de manera, que pueda reducir el TCO de sus instalaciones SAP ya existentes.



GMD provee esta solución dentro de un esquema flexible que permite aumentar o disminuir el alcance y la capacidad de estos, en función de las necesidades del negocio.

Soluciones:

Se dispone de las siguientes soluciones:

Soporte BASIS Netweaver

- Soporte Basis remoto (Transportes, roles, monitoreo, periféricos y soporte basis).
- Upgrades a ECC6.0 (HP-UX, Windows).
- Upgrades BW (HP-UX).
- Upgrades SCM (HP-UX).
- Upgrades de base de datos SAP (DB2, Oracle, SQL, Maxdb).
- Upgrades Unicode (HP-UX ,Windows).
- Instalaciones Netweaver (BW, Portal, Cproject, Solution Manager4.0 SP3 y otros).
- Implementaciones avanzadas en Solution Manager (Monitoring, Reporting, Charms).

Soporte Funcional

- Soporte funcional SAP en los módulos: FI-CO-MM-PP-PM-QM-SD
- Mesa de Ayuda funcional SAP (primer nivel - agentes de servicio).
- Mesa de Ayuda funcional SAP (segundo nivel - Consultor base).
- ABAP factory y JAVA factory (a través del SWF de GMD).
- Implementaciones básicas y avanzadas de Solution Manager (implementation, methodology, Support desk, change request).
- Ejecución de talleres técnico operativos en clientes de la base instalada (módulos relevados y adecuados de acuerdo a la realidad del cliente).



5.2.5. Servicios de Tecnología de GMD

La línea de Servicios de Tecnología de GMD está conformada por un equipo de profesionales que reúnen experiencia, capacitación y los más altos niveles de certificación para atender sus necesidades en tecnologías de la información. La ejecución de los servicios de esta línea está basada en una combinación de metodologías de gestión de proyectos, propia y de los fabricantes asociados a GMD, y en nuestros procedimientos certificados ISO9001-2000. Respalados en esta sólida base, la línea de Servicios de Tecnología pone a su disposición una amplia gama de soluciones flexibles y confiables, las cuales están clasificadas como le mostramos a continuación:

Servicios :

- Servicios Microsoft
- Servicios Networking
- Servicios de Plataforma de Misión Crítica
- Computación Personal

Para la implementación de la Metodología MIRGMD en un Proyecto de Software Real, fue el Área de Software Factory de la Empresa GMD S.A. la que participo en el desarrollo de software usando la Metodología MIRGMD.

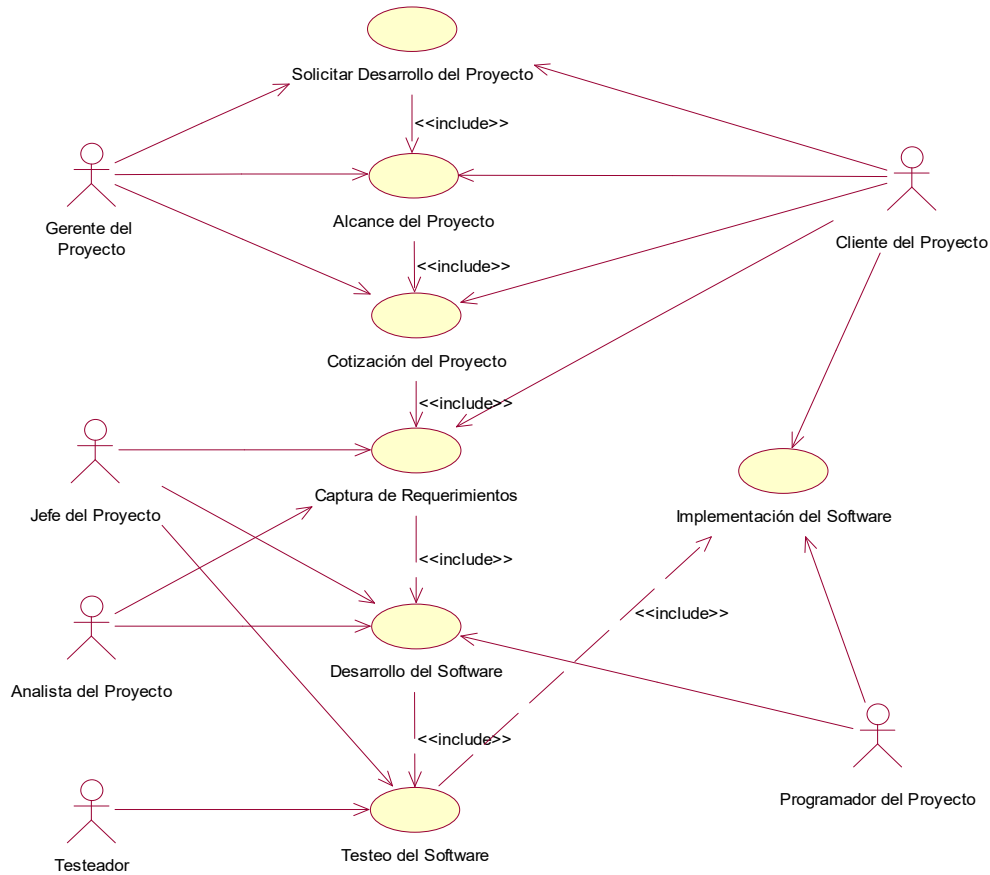
El Grupo Participante encargado de liderar la Implementación de la Metodología MIRGMD, son los siguientes:

- Ing. Lourdes Cavero: Gerente del Proyecto
- Ing. Pablo Arbulu: Jefe del proyecto.
- Bachiller. Oscar Sonco: Analista.
- Bachiller. Diego Enriquez: Programador.

Se describe a continuación los casos de uso utilizados para el proyecto gestionado por el Área de Software Factory de la Empresa GMD S.A. la que participo en el Proyecto de desarrollo de software a la Empresa Contratista, usando la Metodología MIRGMD.



Figura 5.2 Diagrama de casos de uso del Negocio



A. Caso de Uso : Solicitar Desarrollo del Proyecto

Precondición: Necesidad de atención a la Solicitud de Desarrollo.

Flujo Normal

1. El Gerente recibe la Solicitud del desarrollo de un Proyecto de Software de parte del Cliente
2. El Gerente realiza la revisión de la Solicitud del proyecto.

Post Condición: Solicitud Revisada.

B. Caso de Uso : Alcance del Proyecto

Precondición: Necesidad de Evaluar el alcance del Proyecto.



Flujo Normal

1. El Gerente evalúa en coordinación con el Cliente el alcance del Proyecto de Software a desarrollar.
2. El Gerente entrega el Alcance del Proyecto al Cliente para su aprobación.

Post Condición: Alcance del Proyecto aprobado.

C. Caso de Uso : Cotización del Proyecto

Precondición: Necesidad de Cotizar el Desarrollo del Proyecto

Flujo Normal

1. El Gerente realiza la Cotización del Proyecto de Software en Base al Alcance del Proyecto.
2. El Gerente envía la Cotización del Proyecto al Cliente.
3. El Gerente Ordena el Desarrollo del Proyecto de Software

Post Condición: Desarrollo de Proyecto de Software Cotizado.

D. Caso de Uso : Captura de Requerimientos

Precondición: Necesidad de realizar la Captura de Requerimientos

Flujo Normal

1. El Jefe del Proyecto y el Analista del Proyecto preparan las Sesiones y las entrevistas para las captura de requerimientos con el cliente.
2. Se realiza las sesiones y entrevistas para las capturas de requerimientos.
3. Se realiza la documentación de los resultados en las sesiones y entrevistas, esto incluye los requerimientos y documentos referentes al dominio.

Post Condición: Captura de Requerimientos realizados y elaboración de la Documentación de los Requerimientos.

E. Caso de Uso : Desarrollo del Software

Precondición: Necesidad de realizar el Desarrollo del Software.



Flujo Normal:

1. Revisión y Análisis del Documento de Requerimientos por parte del Jefe del Proyecto y el Analista del Proyecto.
2. Se realiza el Diseño del Software por parte del Analista del Proyecto.
3. Se realiza la Construcción del Software por parte del Programador del Proyecto.

Post Condición: Desarrollo del Proyecto del Software realizado

F. Caso de Uso : Testeo del Software

Precondición: Necesidad de realizar el Testeo del Software Final.

Flujo Normal:

1. El Testeador recibe el producto del Software Desarrollado. El Jefe del Proyecto realiza la entrega del Software al Testeador.
2. El Testeador realiza el testeo del Software Desarrollado de acuerdo a los Requerimientos del Cliente.
3. El Testeador realiza el Informe con las conclusiones del Testeo del Software Desarrollado al Jefe del Proyecto.

Flujo Alternativo:

1. Si encuentra errores en el funcionamiento del Software, se realiza el Informe de errores encontrados.

Post Condición: Software Testeado y elaboración del Informe de Conclusiones.

G. Caso de uso : Implementación del Software

Precondición: Necesidad de realizar la instalación del Software para el Cliente.

Flujo Normal:

1. El Programador del Proyecto instala el software desarrollado en el ambiente del Cliente.
2. El cliente verifica el correcto funcionamiento del Software instalado.
3. El Programador elabora el Informe de Conformidad dada por el Cliente.

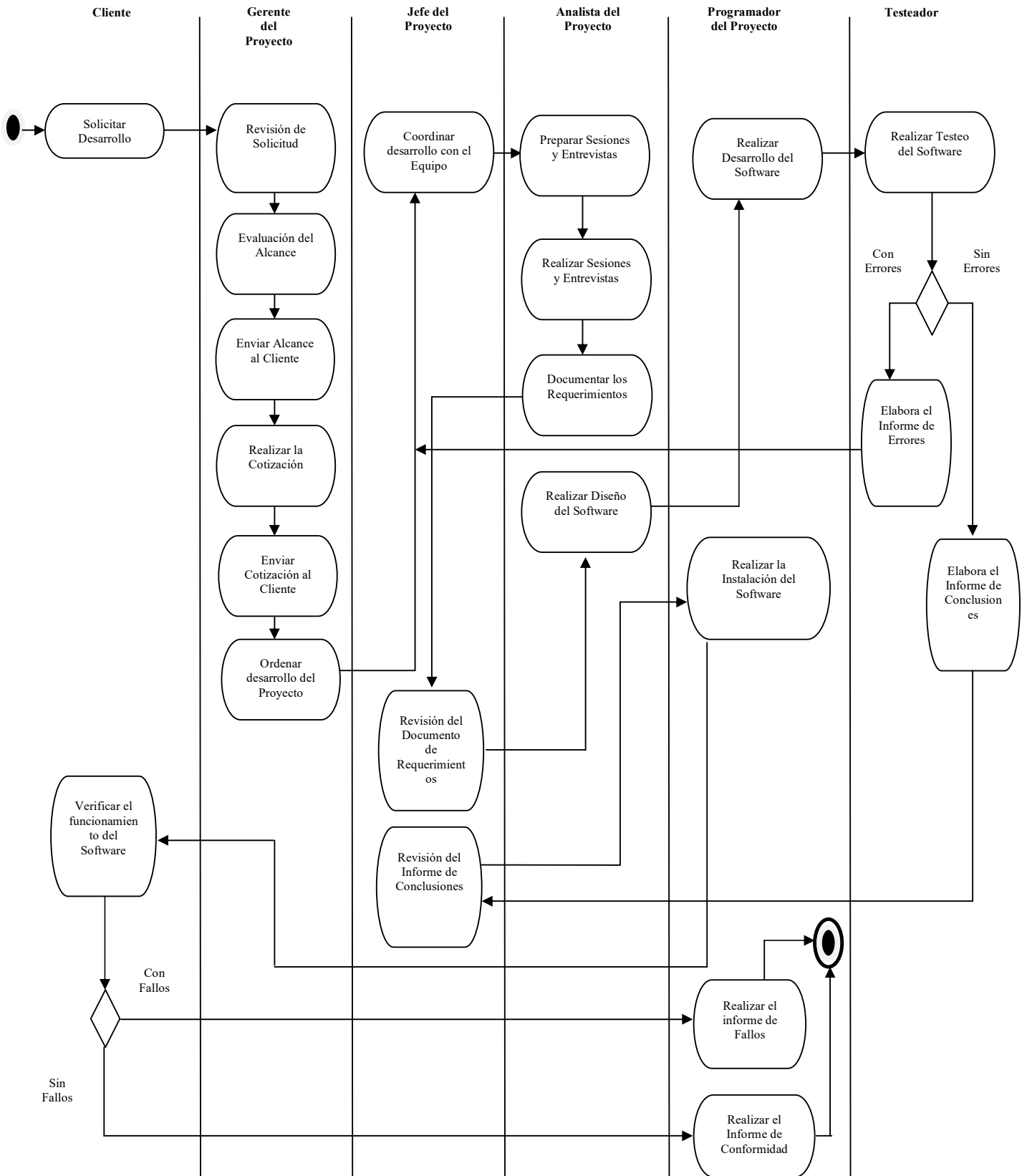
Flujo Alternativo:

1. Si se encuentra fallos en el Software, el Programador realiza el Informe de Fallos encontrados.

Post Condición: Software Instalado y elaboración del Informe de Instalación.



Figura 5.3 Diagrama de Actividades del Negocio



El cliente denominado “La Empresa Contratista”, solicito el requerimiento del Desarrollo de un Sistema basado en un entorno web que administre y gestione el Servicio del Soporte de Sistemas en las distintas Sedes de la Empresa permitiendo así un Control Centralizado y que a su vez también permita integrar las distintas áreas de Soporte de Sistemas de la Empresa.

El desarrollo del Sistema se realizo en las instalaciones del Cliente “ La Empresa Contratista ” en la Sede Principal de Lima.

5.3 Descripción de la aplicación de la Metodología MIRGMD

A continuación se describe la Metodología MIRGMD en cada una de las etapas aplicado en el caso específico del desarrollo de un Sistema de Servicio de Soporte de Sistemas Integrado.

5.3.1 Etapa 1. Elicitación

En esta etapa se trabajó con todos los participantes tanto de la empresa cliente y desarrolladora. Aunque la metodología propone varias técnicas de elicitación de requerimientos, para este caso específico relacionado con el desarrollo de la solución vía Internet se utilizó primeramente 2 sesiones JAD, previamente se necesito prepararnos para realizar dichas sesiones adecuadamente, realizando entrevistas previas y analizando la información obtenida en la primera tarea de la etapa de Elicitación, no se pudo realizar más sesiones debido a los horarios de trabajo del personal de la empresa cliente, se complemento esta etapa con entrevistas individuales abiertas y cerradas.

El resultado obtenidos durante la 1ra. sesión JAD fueron una descripción más exacta del problema y información del dominio.

Descripción del problema expresado por el Clientes

- a. Tiempo perdido entre las llamadas telefónicas realizadas por el usuario para reportar incidentes en los Sistemas de la Empresa.



- b. Tiempo perdido entre las llamadas telefónicas realizadas por el usuario para solicitar requerimientos en los Sistemas de la Empresa.
- c. Saturación de la línea telefónica de la Empresa ya que al reportarse un incidente masivo el número de las llamadas al Área de Sistemas, se elevan considerablemente.
- d. Tiempo de atenciones de incidentes al usuario demasiados extensos.
- e. Tiempo de atenciones de requerimientos al usuario demasiados extensos.
- f. Aumentos de Costos en el Equipo de Soporte de Sistemas.
- g. Ineficiencia y pobre desempeño del Servicio de Soporte de Sistemas brindado al usuario.
- h. Falta de Integración en los Servicios de Soporte de Sistemas en toda la Organización.
- i. Falta de conocimiento del Desempeño en tiempo real del Personal de Soporte de Sistemas

Los procesos actuales realizados son en su mayoría de carácter manual, habiéndose registrado problemas de orden administrativo y Organizativo ente el Personal de Soporte de Sistemas

Otros resultado obtenidos durante las sesiones JAD fue una lista de requerimientos y objetivos los cuales fueron registrados usando las plantillas recomendadas en la metodología.

Como resultado de la tarea de “Identificación/Revisión de los objetivos del sistema” se obtuvo los siguientes objetivos especificados usando las plantillas para objetivos.

Plantilla 1: OBJ -01 Gestión de Seguridad

OBJ-01	Gestión de Seguridad
Descripción	El Sistema deberá permitir gestionar los accesos a los usuarios y sus respectivos roles
Estabilidad	Alta
Comentarios	Ninguno



Plantilla 2: OBJ -02 Gestión de Incidentes

OBJ-02	Gestión de Incidentes
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los incidentes reportados.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 3: OBJ -03 Gestión de Problemas

OBJ-03	Gestión de Problemas
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los Problemas reportados.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 4: OBJ -04 Gestión de Requerimientos

OBJ-04	Gestión de Requerimientos
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los Requerimientos reportados.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 5: OBJ -05 Gestión de Búsquedas

OBJ-05	Gestión de Búsquedas
Descripción	El sistema deberá permitir que se realice las búsquedas de acuerdo al nombre del usuario, el numero de incidente / problema / requerimiento, el nombre del Analista de Soporte de Sistemas u otra característica valida.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 6: OBJ -06 Generación de Reportes de Evaluación

OBJ-06	Generación de Reportes de Evaluación
Descripción	El sistema deberá generar Reportes de Evaluación Estadísticos de los incidentes por periodos, requerimientos por periodos, desempeño del Personal, historiales de los usuarios, incidentes masivos y satisfacción del usuario en el Servicio.
Estabilidad	Alta
Comentarios	Ninguno

Luego de las tareas de “Identificar/ Revisar los requerimientos funcionales” y usando la técnica de casos de uso se obtuvo los siguientes resultados especificados en las siguientes plantillas:

Los actores identificados fueron los siguientes:



Plantilla 7: ACT-01 Administrador del sistema

ACT-01	Administrador del Sistema
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Administrador del Sistema del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno

Plantilla 8: ACT-02 Gerente del departamento de Sistemas

ACT-02	Gerente del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Gerente del departamento de Sistemas del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno

Plantilla 9: ACT-03 Jefe del departamento de Sistemas

ACT-03	Jefe del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Jefe del departamento de Sistemas del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno

Plantilla 10: ACT-04 Coordinador del departamento de Sistemas

ACT-04	Coordinador del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Coordinador del departamento de Sistemas del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno

Plantilla 11: ACT-05 Analista del departamento de Sistemas

ACT-05	Analista del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Analista del departamento de Sistemas del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno



Plantilla 12: ACT-06 Usuario Final

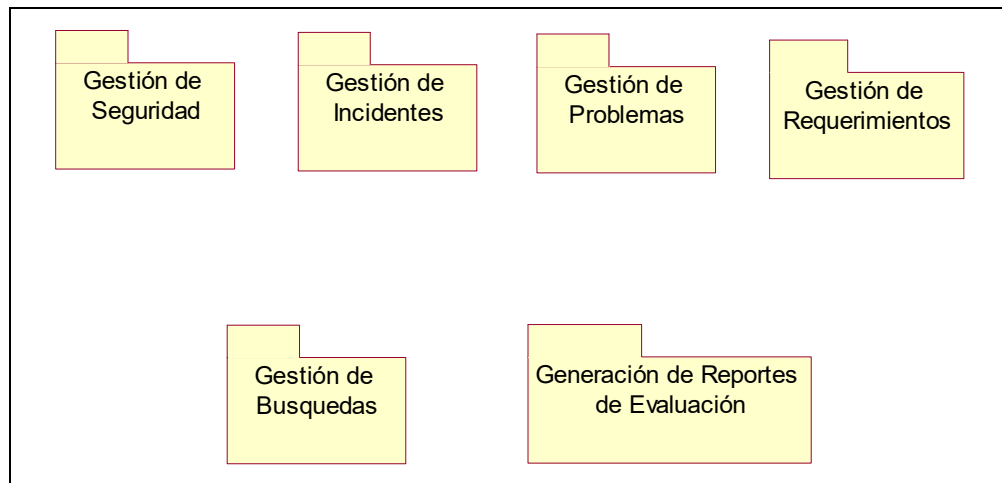
ACT-06	Usuario Final
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Usuario Final del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Plantilla 13: ACT-07 Gestor de Terceros

ACT-07	Gestor de Terceros
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Gestor de Terceros del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Se especificaron los siguientes subsistemas de acuerdo a funcionalidad del sistema por desarrollarse.

Figura 5.4 Diagrama de Subsistemas



Se obtuvieron los siguientes requerimientos funcionales, los cuales fueron identificados usando patrones lingüísticos descritos como RF-nro. para poder identificar los requerimientos.



- RF-01 Validar Usuario
- RF-02 Ingresar Usuario
- RF-03 Modificar Datos de usuario
- RF-04 Eliminar usuarios
- RF-05 Asignar accesos
- RF-06 Nuevo Incidente
- RF-07 Nuevo Requerimiento
- RF-08 Nuevo Problema
- RF-09 Nuevo Contacto
- RF-10 Nuevo Grupo
- RF-11 Nuevo Servicio
- RF-12 Nuevo Método de Reporte
- RF-13 Nueva Categoría
- RF-14 Buscar Incidente
- RF-15 Buscar Requerimiento
- RF-16 Buscar Problema
- RF-17 Buscar Contacto
- RF-18 Buscar Grupo
- RF-19 Buscar Servicio
- RF-20 Buscar Método de Reporte
- RF-21 Buscar Categoría
- RF-22 Editar Incidente
- RF-23 Editar Requerimiento
- RF-24 Editar Problema
- RF-25 Editar Contacto
- RF-26 Editar Grupo
- RF-27 Editar Servicio
- RF-28 Editar Método de Reporte
- RF-29 Editar Categoría
- RF-30 Eliminar Incidente
- RF-31 Eliminar Requerimiento
- RF-32 Eliminar Problema
- RF-33 Eliminar Contacto
- RF-34 Eliminar Grupo
- RF-35 Eliminar Servicio
- RF-36 Eliminar Método de Reporte
- RF-37 Eliminar Categoría
- RF-38 Actualizar Estado
- RF-39 Insertar Comentario
- RF-40 Transferencia de Ticket
- RF-41 Reportes de Evaluación de Incidentes
- RF-42 Reportes de Evaluación de Requerimientos
- RF-43 Reportes de Evaluación de Problemas
- RF-44 Reportes de Evaluación de Personal
- RF-45 Reportes de Evaluación de Usuario
- RF-46 Reportes de Evaluación de Masivos
- RF-47 Reportes de Evaluación de Satisfacción
- RF-49 Imprimir Reportes de Evaluación de Incidentes
- RF-50 Imprimir Reportes de Evaluación de Requerimientos
- RF-51 Imprimir Reportes de Evaluación de Problemas



RF-52 Imprimir Reportes de Evaluación de Personal
 RF-53 Imprimir Reportes de Evaluación de Usuario
 RF-54 Imprimir Reportes de Evaluación de Masivos
 RF-55 Imprimir Reportes de Evaluación de Satisfaccion

Casos de uso del subsistema Gestión de Incidentes, especificado en las plantillas

Plantilla 14: RF-06 Nuevo Incidente

RF-06	Nuevo Incidente	
Versión	1.0 (Mayo 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien genere un Nuevo Incidente	
Precondición	Usuario logueado como valido	
Secuencia Normal	Paso	Acción
	1	El Sistema muestra la barra de Menú Estándar
	2	El usuario selecciona la opción Archivo – Nuevo Incidente
	3	El Sistema genera un numero de Ticket
	4	El Usuario selecciona el usuario afectado
	5	El Usuario selecciona la Categoría
	6	El Usuario selecciona el estado del Ticket
	7	El Usuario selecciona el Analista de Soporte
	8	El Usuario selecciona el Grupo
	9	El Usuario selecciona el Servicio
	10	El Usuario selecciona el Método de Reporte
	11	El usuario ingresa el Resumen del Incidente
	12	El usuario ingresa la Descripción del Incidente
	13	El usuario guarda el ingreso en el Sistema
	14	El Sistema muestra el mensaje de confirmación
15	El Sistema muestra los datos del Nuevo Incidente Ingresado.	
Post-condición	El usuario ya genero un nuevo Ticket solicitado.	
Excepciones	Paso	Acción
	1	Si no se ingresa los campos requeridos el sistema muestra un mensaje de error
	2	Si no se selecciona la categoría correcta el Sistema muestra un mensaje de error.
Rendimiento	Paso	Cuota de tiempo
Frecuencia	50 veces / día	



esperada	
Importancia	Alta
Urgencia	Inmediatamente
Estado	En construcción
Estabilidad	Alta
Comentarios	Ninguno

Los requerimientos no funcionales obtenidos luego de la tarea de “Identificar/revisar los requerimientos no funcionales” se listan a continuación.

Los requerimientos no funcionales que fueron encontrados son generalmente de portabilidad, disponibilidad, seguridad, mantenibilidad y tamaño, a los cuales se le asignaron los identificadores (Patrones) RNF-nro. Estos fueron obtenidos basándose principalmente en las Métricas para Requerimientos no funcionales descritas en la Metodología MIRGMD.

Requerimientos No Funcionales de Portabilidad

Plantilla 15: RNF-01 Portabilidad

RNF-01	Portabilidad
Versión	1.0 (Mayo 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	El sistema deberá funcionar en un ambiente Web sobre plataforma Linux, usando un servidor Web Apache, también el Sistema deberá visualizarse sin ningún problema con los principales browsers.
Importancia	Alta
Urgencia	Inmediatamente
Estado	En construcción
Estabilidad	Alta
Comentarios	Ninguno

El detalle de todo lo obtenido y validado se encuentra en el documento de requerimientos. Anexo Nro. 2.

5.3.2. Etapa 2. Análisis

En esta segunda etapa del Análisis, se le realizo con el documento obtenido en la etapa anterior, se utilizó la técnica de “checklist de análisis” para evaluar cada



requerimiento, con la finalidad de descubrir problemas, conflictos, incompletez as e inconsistencias en los requerimientos elicitados en la etapa anterior.

Otra técnica que se utilizo como parte del análisis es la matriz de interacción de requerimientos con la finalidad de resaltar los conflictos y solapamientos, ver cuadro 5.1.

Cuadro 5.1. Matriz Interacción de requerimientos

Requerimiento	RF-01	RF-02	RF-03	RF-04	...	RF-54	RF-55
RF-01	0	1000	0	0		0	1
RF-02	0	0	0	0		1	0
RF-03	1000	0	1000	0		0	1000
RF-04	0	1000	1	0		0	1
...						0	1
RF-54	0	0	0	0		0	1
RF-55	0	0	0	1		0	0

Los requerimientos previamente identificados se colocaron como están indicados en la Matriz, del Cuadro 5.1, donde los requerimientos en conflicto estaban asignados con el número 1, los requerimientos que se solapaban con el número 1000 y los requerimientos independientes con el número 0, luego de aplicar la formula suma/100 mencionada en la metodología MIRGMD se obtuvo que 10 de los requerimientos se solapaban y 8 de los requerimientos entraban en conflicto. El conflicto de mayor importancia fue CFL-01.

Plantilla 16: Plantilla de Conflictos CFL-01 del Requisito RF – 05

CFL-01	Necesidad de especificar que Personal del Área de Soporte de Sistemas podrá visualizar los Reportes de Evaluación de Incidentes
Versión	1.1 (Mayo 2009)
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.)
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista)
Objs./Reqs. en conflicto	<ul style="list-style-type: none"> OBJ-01 Gestión de Seguridad RF-05 Asignar Accesos
Descripción	No se describe en ningún requisito cual de los usuarios van a tener el acceso para visualizar los Reportes de Evaluación de Incidentes.
Alternativas	Por definir
Solución	Pendiente de Solución
Importancia	Alta



Urgencia	Inmediatamente
Estado	Validado
Comentarios	Ninguno

La negociación de requerimientos se realizo con los Stakeholders involucrados usando reuniones de negociación, donde se encontraron soluciones a los conflictos, detallado en la plantilla de conflictos, ver plantilla CFL-01.

Para el caso de CFL-01 se presento una solución descrita en la columna alternativa de la plantilla la cual será implementada.

Plantilla 17: Plantilla de Conflictos CFL-01 del Requisito RF – 05

CFL-01	Necesidad de especificar que Personal del Área de Soporte de Sistemas podrá visualizar los Reportes de Evaluación de Incidentes
Versión	1.1 (Mayo 2009)
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.)
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista)
Objs./Reqs. en conflicto	<ul style="list-style-type: none"> OBJ-01 Gestión de Seguridad RF-05 Asignar Accesos
Descripción	No se describe en ningún requisito cual de los usuarios van a tener el acceso para visualizar los Reportes de Evaluación de Incidentes.
Alternativas	Solo el Administrador del Sistema y el Jefe de Soporte de Sistemas, tendrán acceso a los Reportes de Evaluación de Incidentes, y el resto de usuarios no tendrán el acceso.
Solución	Por implementar.
Importancia	Alta
Urgencia	Inmediatamente
Estado	Validado
Comentarios	Ninguno

5.3.3. Etapa 3. Especificación

Finalizada las tareas en la segunda etapa del Análisis, se continúa en esta tercera etapa con el desarrollo del documento de requerimientos final, la cual se estructuro de la siguiente forma.

1. Introducción.
 - 1.1 Propósito del Documento de Requerimientos.
 - 1.2 Alcance del Documento de Requerimientos.



- 1.3 Definiciones, acrónimos y abreviaciones.
- 2. Descripción general
 - 2.1 Descripción de la situación actual
 - 2.2 Propósito y alcance
 - 2.3 Definición del problema
 - 2.4 Participantes en el proyecto
 - 2.5 Restricciones Generales
- 3. Requerimientos Específicos
 - 3.2 Requerimientos funcionales
 - 3.3 Requerimientos No funcionales.
 - 3.5 Conflictos en los Requerimientos.
 - 3.4 Matriz de rastreabilidad Objetivos - requerimientos
- 4. Apéndices

Las versiones generadas están listadas en el cuadro 5.2.

Cuadro 5.2 Lista de Versiones del documento de requerimientos.

Numero	Fecha	Descripción	Autores
0	Abril, 2009	Versión 1.0	Oscar Sonco, Diego Enriquez
1	Mayo, 2009	Versión 1.1	Oscar Sonco, Diego Enriquez

5.3.4. Etapa 4. Verificación

En esta etapa se realizaron las preinspecciones y la Inspección de requerimientos recomendadas en la metodología MIRGMD, teniendo como entrada un borrador final del documento de requerimientos, aquí se verifico que los requerimientos estén de acuerdo en completeza y consistencia, conformidad a los estándares, conflictos de requerimientos, errores técnicos y requerimientos ambiguos. Para tal fin se uso la técnica del Checklist de Inspecciones donde los requerimientos fueron confrontados de acuerdo a:



- Entendibilidad – Legibilidad
- Redundancia
- Completeza
- Ambigüedad
- Consistencia
- Organización
- Conformidad a estándares
- Rastreabilidad

Para cumplir con el checklist de Rastreabilidad se implemento con la herramienta OSRMT descrita en la siguiente etapa de Administración de Requerimientos.

El desarrollo del prototipo permitió también una forma de validar los requerimientos debido a que el usuario podía contrastar sus requerimientos contra algo real que es el prototipo.

5.3.5. Etapa 5. Administración de Requerimientos

Los cambios en los requerimientos pueden continuar a pesar después de haber finalizado el Documento de Requerimientos, pueden venir nuevas necesidades de los clientes o de problemas operacionales con el sistema.

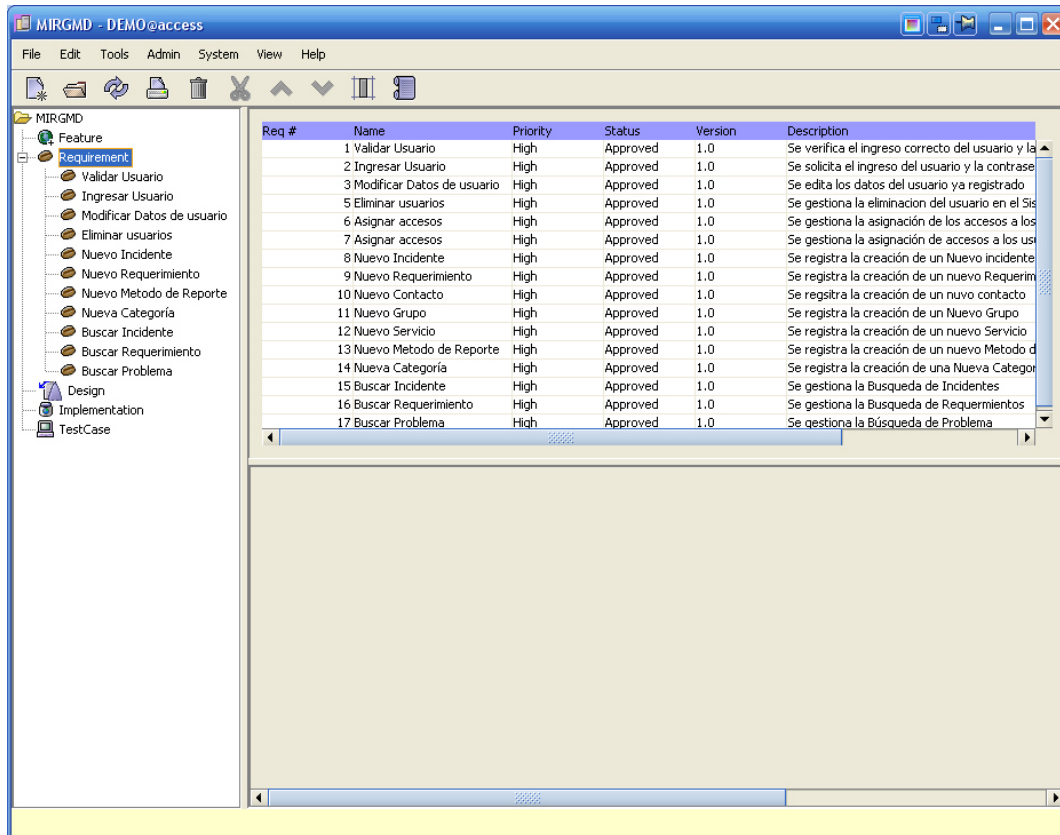
Para la gestión de los requerimientos del sistema por desarrollar se usara el OSRMT recomendada por esta metodología.

Después de realizado las iteraciones y culminado el prototipo, aparecieron cambios en algunos requerimientos. Para los cambios propuestos se utilizo el formulario de solicitud recomendado como parte de la identificación del cambio. Una vez analizado los cambios propuestos se procedió con la implementación.

El OSRMT como herramienta de Administración se uso para el registro de los requerimientos, el cual permitió llevar un control y asignación de prioridades, a su vez también se registraron los detalles y objetivos concernientes a cada uno de ellos.



Figura 5.5. Listado de los Requerimientos en OSRMT



5.3.6. Gestión de Riesgos

Aplicando las Etapas de la Gestión de Riesgos establecemos lo siguiente:

a) Identificación de riesgos.

Se realiza la identificación de los riesgos mediante las siguientes Tablas:

Característica del requisito: ambigüedad. Pregunta: ¿es ambiguo el requisito?

Tabla 1 Identificación de riesgos.

ID requisito	Respuesta	Impacto	Prioridad	Componente
RF-01 Validar Usuario	NO	SI	Alta	COSTO
RF-02 Ingresar Usuario	NO	SI	Alta	COSTO
RF-03 Modificar Datos de usuario	SI	NO	Baja	TIEMPO
RF-04 Eliminar usuarios	NO	NO	Baja	COSTO



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

RF-05 Asignar accesos	SI	SI	Alta	TIEMPO
RF-06 Nuevo Incidente	NO	SI	Alta	COSTO
RF-07 Nuevo Requerimiento	NO	SI	Alta	COSTO
RF-08 Nuevo Problema	NO	SI	Alta	COSTO
RF-09 Nuevo Contacto	SI	NO	Baja	TIEMPO
RF-10 Nuevo Grupo	SI	NO	Baja	COSTO
RF-11 Nuevo Servicio	SI	NO	Baja	COSTO
RF-12 Nuevo Método de Reporte	NO	NO	Baja	COSTO
RF-13 Nueva Categoría	SI	SI	Alta	TIEMPO

Riesgos que pueden afectar al propio proceso de Ingeniería de Requerimientos.

Tabla 2 Lista de riesgos y clasificación.

Riesgo	Tipo de riesgo	Descripción
Rotación de personal	Proyecto, producto y negocio	Personal con experiencia abandona el proyecto antes de que finalice
Cambios de requisitos	Proyecto y producto	Existencia de más cambios de requerimientos de los previstos inicialmente
Retrasos en la especificación	Proyecto y producto	Retrasos en las especificaciones de interfaces esenciales
Subestimación del tamaño	Proyecto y producto	El tamaño del requisito (la Especificación de los Requisitos, del proceso de Ingeniería de requisitos) se ha subestimado
Bajo rendimiento de la herramienta CASE	Producto	Las herramientas CASE que ayudan al proyecto no tienen el rendimiento y las funcionalidades esperadas

Tabla 3 Riesgos por requisitos.

ID Requisito	Tipo de Riesgo	Riesgos
RF-01 Validar Usuario	De comunicación	El cliente no pueda participar en revisiones y en reuniones
RF-02 Ingresar Usuario	De comunicación	El cliente no pueda participar en revisiones y en reuniones
RF-03 Modificar Datos de usuario	De comunicación	El cliente no pueda participar en revisiones y en reuniones
RF-04 Eliminar usuarios	De	El cliente no pueda participar en



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

	comunicación	revisiones y en reuniones
RF-05 Asignar accesos	De personal	El personal no cuenta con los conocimientos requeridos para enfrentar la complejidad del requisito
RF-06 Nuevo Incidente	De requisitos	Cambios de requisitos que precisan modificaciones en el diseño
RF-07 Nuevo Requerimiento	De requisitos	Cambios de requisitos que precisan modificaciones en el diseño
RF-08 Nuevo Problema	De requisitos	Cambios de requisitos que precisan modificaciones en el diseño
RF-09 Nuevo Contacto	De requisitos	Cambios de requisitos que precisan modificaciones en el diseño
RF-10 Nuevo Grupo	De estimación	El tiempo requerido para desarrollar el proceso de ingeniería de requisitos está subestimado
RF-11 Nuevo Servicio	De estimación	El tiempo requerido para desarrollar el proceso de ingeniería de requisitos está subestimado
RF-12 Nuevo Método de Reporte	De estimación	El tiempo requerido para desarrollar el proceso de ingeniería de requisitos está subestimado
RF-13 Nueva Categoría	De estimación	El tiempo requerido para desarrollar el proceso de ingeniería de requisitos está subestimado

b) Evaluación de los riesgos.

Realizamos la evaluación de todos los riesgos serios o catastróficos con su respectiva probabilidad.

Tabla 5 Riesgos ordenados por efectos.

Riesgo	Probabilidad	Efectos
Problemas financieros de la organización reducen el presupuesto del proyecto	Baja	Catastrófico
Imposible contratar personal con los conocimientos requeridos	Alta	Catastrófico
Personal clave enfermo o no disponible en momentos críticos	Media	Serio
Cambios de requerimientos que precisan modificaciones en la codificación	Media	Serio
El tiempo requerido para desarrollar el proceso de Ingeniería de requisitos está subestimado	Alta	Serio
Los clientes no comprenden el impacto de los cambios en los requerimientos	Media	Tolerable



c) Planificación de riesgos.

Aquí desarrollaremos las estrategias para tratar los riesgos.

Tabla 6 Estrategias por riesgos.

Riesgo	Estrategia
Problemas financieros de la organización	Preparar un documento breve para la dirección de la empresa que muestra que el proyecto hace contribuciones muy importantes a las metas del negocio
Problemas de reclutamiento	Organizar cursos de capacitación para el personal ya existente, investigar la posibilidad de contratar en otras regiones del país
Enfermedad del personal	reorganizar el equipo de tal forma que se solapen el trabajo y los miembros comprendan el trabajo de los demás
Cambios en los requisitos	Rastrear la información para valorar el impacto de los requerimientos, maximizar la información oculta en ellos
Tiempo de IR subestimado	Alertar al cliente de las dificultades potenciales y las posibilidades de retraso

d) Supervisión de los riesgos.

Se controlan factores que pueden indicar cambios en la probabilidad y el impacto.

Tabla 7 Indicadores potenciales por riesgos.

Tipo de riesgo	Indicadores potenciales
Tecnología	Entrega retrasada del hardware. Existencia de informes sobre problemas tecnológicos.
Personal	Baja moral del personal, malas relaciones entre miembros del equipo, plazas vacantes,
Organizacional	Rumores. Falta de iniciativa de la dirección.
Herramientas	Rechazo de los miembros del equipo a utilizar herramientas. Quejas sobre las CASE
Requisitos	Peticiones de muchos cambios en los requisitos. Quejas del cliente
Estimación	Fracaso en el cumplimiento de los tiempos planificados.

5.4 Descripción del Sistema a Desarrollar

El proyecto en el cual se realizó la implementación de la Metodología MIRGMD , tiene por objetivo el desarrollo de un producto para el Área de Sistemas llamado Servicio de Soporte de Sistemas Integrado – SSSI, el cual se desarrollara en un entorno Web, para las distintas Sedes de la Empresa Contratista, permitiendo así un Control



Centralizado y que a su vez también permita integrar las distintas áreas de Soporte de Sistemas de la Empresa.

Ver detalle en el Documento de requerimiento ANEXO N° 2.

Dicho sistema constara de seis módulos o subsistemas generales.

- El Módulo de Seguridad que permitirá definir a los usuarios y los accesos respectivos de acuerdo a sus roles y funciones en los cuales podrán registrar información, realizar búsquedas de información y Visualizar Reportes de Evaluación, de acuerdo a las reglas de seguridad establecidas.

- El Módulo de Incidente que permitirá que los usuarios realicen registros del incidente, ingresar su categoría, su método de reporte, su prioridad, la descripción del problema establecido y la actualización del Estado del Incidente hasta el Cierre del Incidente, de acuerdo a las reglas de seguridad establecidas.

- El Módulo de Requerimiento que permitirá que los usuarios realicen registros del requerimiento, ingresar su categoría, su método de reporte, su prioridad, la descripción del requerimiento que se solicita y la actualización del Estado del requerimiento hasta el cierre del requerimiento, de acuerdo a las reglas de seguridad establecidas.

- El Módulo de Problema que permitirá que los usuarios realicen registros del Problema, ingresar su categoría, su método de reporte, su prioridad, la descripción del problema que se va investigar y la actualización del Estado del



problema hasta el cierre del problema, de acuerdo a las reglas de seguridad establecidas.

- El Módulo de Búsquedas que permitirá que los usuarios realicen búsquedas de Incidentes, requerimientos, problemas, contactos, grupos, servicios, métodos de reporte y categorías , de acuerdo a las reglas de seguridad establecidas.
- El Módulo de Reportes de Evaluación que permitirá que los usuarios visualicen los reportes de evaluación de los incidentes, requerimientos, problemas, contactos, grupos, servicios, métodos de reporte y categorías, de acuerdo a las reglas de seguridad establecidas.

Para el caso de estudio se realizó el desarrollo de un prototipo del Sistema debido a que este trabajo no cubre el desarrollo del Sistema Completamente.

5.5 Desarrollo del Prototipo

Se presentará a continuación algunas de las impresiones de pantalla del prototipo desarrollado como parte del trabajo de investigación, los cuales también fueron validos para la implementación de la metodología MIRGMD.

Figura 5.5 Ventana de Ingreso al Sistema



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

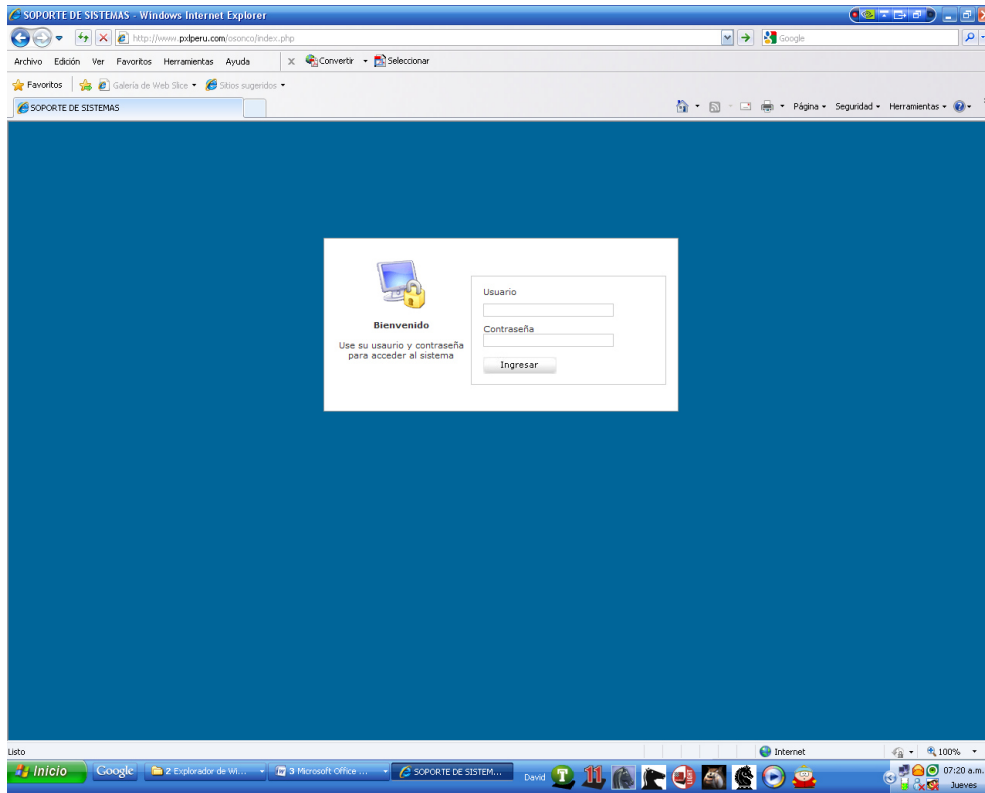


Figura 5.6 Ventana de Lista de Incidentes

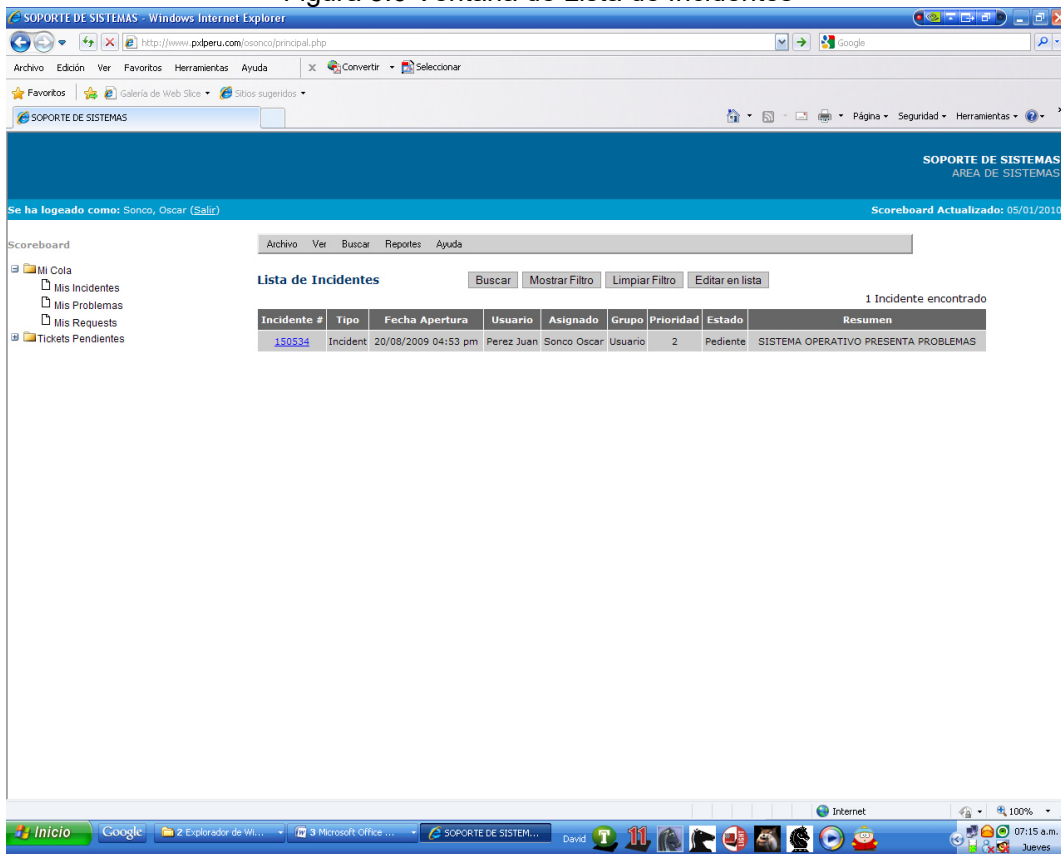


Figura 5.7 Ventana que registra un Nuevo Incidente

The screenshot shows a web browser window with the URL <http://www.pxperu.com/sonco/principal.php>. The page title is "SOORTE DE SISTEMAS" and the user is logged in as "Sonco, Oscar". The main content area is titled "Crear nuevo Incidente" and contains the following form fields:

- Usuario Afectado:** Text input field.
- Elemento Conf.:** Text input field.
- Categoría:** Text input field.
- Descripción:** Text input field.
- Estado:** Text input field.

Below these fields is a "Detalle" section with the following fields:

- Asignado:** Text input field.
- Grupo:** Text input field.
- Servicio:** Text input field.
- Tipo:** Dropdown menu with "Incidente" selected.
- Numero de Ticket:** Text input field.
- Método Reporte:** Text input field.
- Pais Afectado:** Text input field.
- Registrado por:** Text input field with "Sonco Oscar" displayed.
- Urgencia:** Dropdown menu.
- Impacto:** Dropdown menu.
- Prioridad:** Dropdown menu.
- Proveedor:** Text input field.
- Causa Raiz:** Text input field.

At the bottom, there is a "Resumen del Incidente" section with a "Resumen" text area and a "Tiempo total Act." timer set to "00:00:00". A "Descripción" text area and a "Timer" set to "00:00:03" are also present.

Figura 5.8 Ventana que registra un Nuevo Problema

The screenshot shows the same web browser window as Figure 5.7, but the main content area is titled "Crear nuevo Problema". The form fields are identical to those in Figure 5.7, but the "Tipo" dropdown menu is set to "Problema". The "Registrado por" field still shows "Sonco Oscar". The "Resumen del Problema" section at the bottom has a "Tiempo total Act." timer set to "00:00:00" and a "Timer" set to "00:00:03".



Figura 5.9 Ventana que registra un Nuevo Requerimiento

SOPORTE DE SISTEMAS - Windows Internet Explorer
 http://www.palperu.com/sonco/principal.php

Se ha logeado como: Sonco, Oscar (Salir) Scoreboard Actualizado: 05/01/2010

Crear nuevo Request Guardar Crear Change Order Cancelar Limpiar

Usuario Afectado Elemento Conf. Categoría Descripción Estado

Detalle

Asignado Grupo Servicio Tipo Numero de Ticket

Método Reporte País Afectado Registrado por Activo

Urgencia Impacto Prioridad

Proveedor Causa Raiz

Resumen del Request

Resumen Tiempo total Act.

Descripción Timer

Figura 5.10 Ventana que administra la Búsqueda de incidentes

SOPORTE DE SISTEMAS - Windows Internet Explorer
 http://www.palperu.com/sonco/principal.php

Se ha logeado como: Sonco, Oscar (Salir) Scoreboard Actualizado: 05/01/2010

Busqueda de Incidentes Buscar Mostrar Filtro Limpiar Filtro Editar en lista

Usuario Analista asignado Grupo Estado Categoría Activo

Registrado por Servicio Abierto desde Abierto hasta Ticket

Cerrado desde Cerrado hasta Resuelto desde Resuelto hasta

Resumen Descripción Estado de asignación

Incidente #	Tipo	Fecha Apertura	Usuario	Asignado	Grupo	Prioridad	Estado	Resumen
150334	Incident	20/08/2009 04:53 pm	Perez Juan	Sonco Oscar	Usuario	2	Pendiente	SISTEMA OPERATIVO PRESENTA PROBLEMAS

1 - 1 de 1



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Figura 5.11 Ventana que administra la Búsqueda de Problemas

Se ha logeado como: Sonco, Oscar (Salir) Scoreboard Actualizado: 05/01/2010

Scoreboard Archivo Ver Buscar Reportes Ayuda

Búsqueda de Problemas Buscar Mostrar Filtro Limpiar Filtro Editar en lista

Usuario: Analista asignado: Grupo: Estado: Categoría: Activo: Activo

Registrado por: Servicio: Abierto desde: Abierto hasta: Ticket:

Cerrado desde: Cerrado hasta: Resuelto desde: Resuelto hasta:

Resumen: Descripción: Estado de asignación: vacio

Problema #	Tipo	Fecha Apertura	Usuario	Asignado	Grupo	Prioridad	Estado	Resumen
160356	Problema	30/11/2009 02:34 pm	Rojas Jose	Sonco Oscar	Usuario	1	Pendiente	MESSANGER NO LEVANTA
170127	Problema	14/09/2009 10:16 am	Zavaleta Maria	Sonco Oscar	Usuario	2	Pendiente	VIRUS WIN32 TANATO.W

1 - 2 de 2

Figura 5.12 Ventana que administra la Búsqueda de Requerimientos

Se ha logeado como: Sonco, Oscar (Salir) Scoreboard Actualizado: 05/01/2010

Scoreboard Archivo Ver Buscar Reportes Ayuda

Búsqueda de Requets Buscar Mostrar Filtro Limpiar Filtro Editar en lista

Usuario: Analista asignado: Grupo: Estado: Categoría: Activo: Activo

Registrado por: Servicio: Abierto desde: Abierto hasta: Ticket:

Cerrado desde: Cerrado hasta: Resuelto desde: Resuelto hasta:

Resumen: Descripción: Estado de asignación: vacio

Request #	Tipo	Fecha Apertura	Usuario	Asignado	Grupo	Prioridad	Estado	Resumen
151873	Request	20/08/2009 05:55 pm	Perez Juan	Sonco Oscar	Usuario	3	Pendiente	TRASLADO DE EQUIPOS
149186	Request	14/08/2009 04:11 pm	Alvarez Mario	Sonco Oscar	Usuario	3	Pendiente	REASIGNACION DE EQUIPO
144398	Request	10/07/2009 04:53 pm	Salomon Carlos	Sonco Oscar	Gestión de Sistemas	3	Pendiente	SE REQUIERE UNA PC PARA ASIGNAR AL PUESTOO DE VIGILANCIA

1 - 3 de 3



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Figura 5.13 Ventana que visualiza la lista de Requerimientos Funcionales

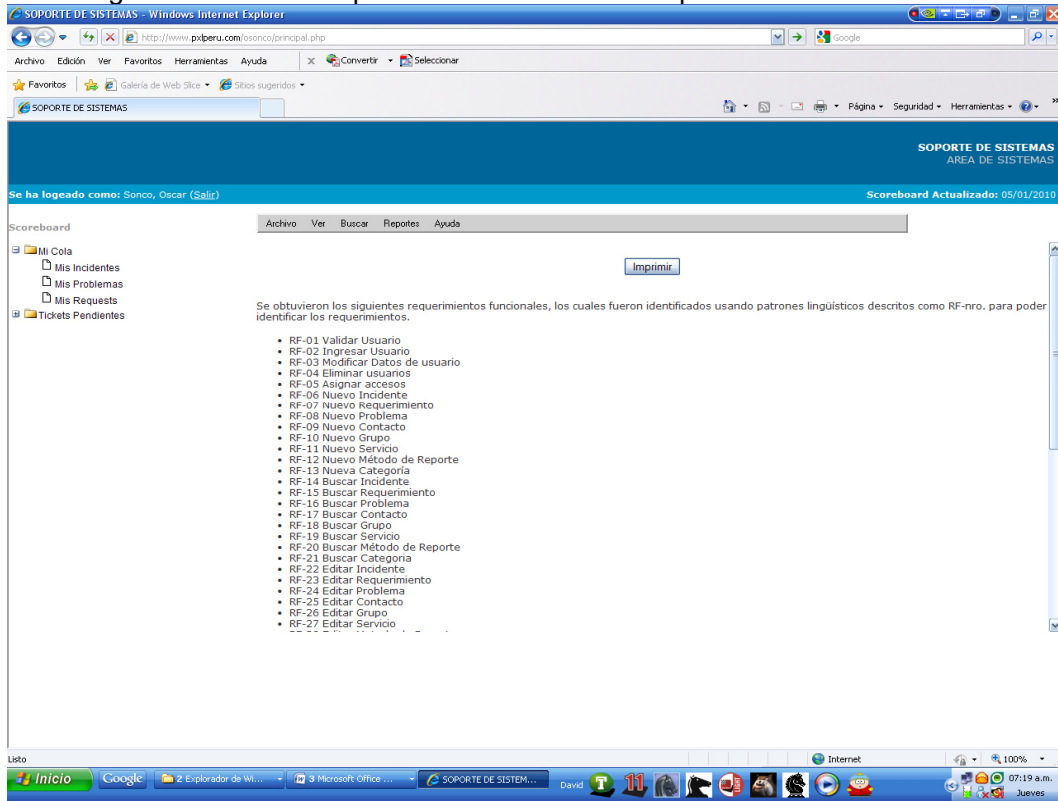
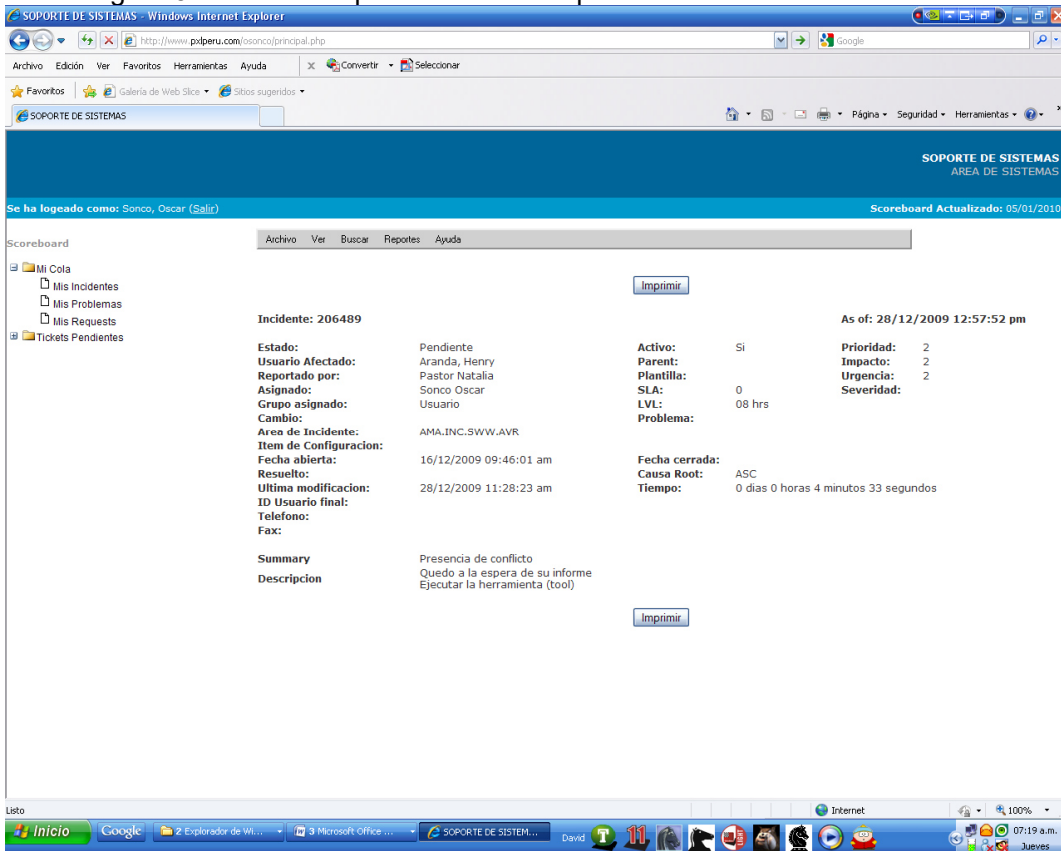


Figura 5.14 Ventana que visualiza el Reporte de evaluación de Incidentes



**METODOLÓGICA PARA LA INGENIERÍA DE REQUERIMIENTOS
ORIENTADA A PROYECTOS DE DESARROLLO DE SOFTWARE
APLICADO A EMPRESAS CLIENTES DE GMD S.A. - MIRGMD Versión 1.0**

HOJA DE INFORMACION GENERAL

CONTROL DOCUMENTAL:

PROCEDIMIENTO:	METODOLÓGICA PARA LA INGENIERÍA DE REQUERIMIENTOS ORIENTADA A PROYECTOS DE DESARROLLO DE SOFTWARE APLICADO A EMPRESAS CLIENTES DE GMD S.A. - MIRGMD Versión 1.0
AUTOR:	Oscar Freddy Sonco Tena
VERSIÓN:	1.0
FECHA EDICIÓN:	--/04/2009
NOMBRE DE ARCHIVO:	MIRGMD -1.0.doc
RESUMEN:	La metodología MIRGMD Versión 1.0, es una metodología desarrollada basada en diferentes enfoques propuestos por diversos autores. Está compuesta por las etapas de E licitación de requerimientos, Análisis de Requerimientos, Especificación de Requerimientos, Verificación de los Requerimientos y Administración de los Requerimientos.



INDICE DE LA METODOLOGIA

Conceptos Teóricos.....	116
1. Elicitación de requerimientos.....	117
1.1 Obtener información sobre el problema, el dominio del problema y sistemas existentes.....	117
1.2 Preparar y realizar las sesiones de elicitación/ negociación.....	118
1.3 Identificar/ revisar los objetivos del sistema.....	119
1.4 Identificar/ Revisar los requerimientos funcionales.....	120
1.5 Identificar/revisar los requerimientos no funcionales.....	121
2. Análisis y Negociación de Requerimientos.....	122
2.1 Análisis de requerimientos.....	122
2.1.1 Tarea 1: Chequear las necesidades.....	122
2.1.2 Tarea 2: Chequear si son consistentes y Completos.....	123
2.1.3 Tarea 3: Chequear su factibilidad.....	123
2.2 Negociación de requerimientos.....	124
2.2.1 Tarea 1: Discusión de Requerimientos.....	124
2.2.2 Tarea 2: Priorización de requerimientos.....	125
2.2.3 Tarea 3: Concordancia de requerimientos.....	125
3. Especificación de Requerimientos.....	126
4. Verificación de Requerimientos.....	128
4.1 Revisar el Documento de Requerimientos.....	128
4.2 Cerrar la versión de los requisitos.....	129
5. Administración de Requerimientos.....	130
5.1. Identificación de problemas de los requerimientos.....	131
5.2. Análisis de los cambios propuestos.....	131
5.3. Implementación del cambio.....	132
6. Técnicas Recomendadas.....	133
6.1 Técnicas de Elicitación de requerimientos.....	133
6.1.1 Entrevistas.....	133
6.1.2. Joint Application Development.....	136
6.1.3 Casos de uso.....	140
6.1.4. Prototipos.....	141
6.1.5 Observación y Análisis Social.....	142
6.1.6 Reuso de Requerimientos.....	143
6.1.7 Plantillas y patrones lingüísticos para elicitación de requisitos.....	143
6.1.8 Métricas Para Especificar Requerimientos no funcionales.....	151
6.2 Técnicas de Análisis de requerimientos.....	151
6.2.1 Checklist de Análisis.....	151
6.2.2 Interacciones de requerimientos.....	152



6.3 Técnicas de Validación de Requerimientos.....	153
6.3.1 Checklist de Inspecciones.....	153
6.3.2 Prototipos.....	154
6.3.3 Inspección de requerimientos.....	155
6.3.4 Desarrollo de manual de usuario.....	157
6.3.5 Desarrollar Teste requerimientos.....	157
6.4 Técnicas de Gerencia de Requerimientos.....	159
6.4.1 Tablas de rastreabilidad.....	159



CONCEPTOS TEORICOS

Dentro de la Metodología para Ingeniería de Requerimientos de Desarrollo de Sistemas de Información se manejarán los siguientes conceptos y/o términos:

Ingeniero de Requerimientos

Puede ser reemplazado por el Analista de Sistemas, es el responsable para elicitar y especificar los requerimientos del sistema.

Equipo de Usuarios

Pertencen al área usuaria directamente comprometida con el proyecto, pudiendo ser cualquier unidad organizacional de la Institución. Participan activamente durante todo el proceso de desarrollo del sistema de información.

Coordinador del Proyecto

Persona responsable del éxito del proyecto. Tiene la visión y la experiencia necesaria para coordinar los esfuerzos y organizar las actividades realizadas por los integrantes de un grupo de trabajo enfocados en el desarrollo de una solución

Experto del dominio o Líder Usuario

Persona que conoce al detalle la operatividad y funcionalidad del área usuaria solicitante.

Programador de Sistemas

Es el responsable de la creación de los prototipos así como del código del sistema. Las técnicas mencionadas en cada una de las etapas se describen en el punto 6.



1. Elicitación de requerimientos

Descripción y objetivo

La Elicitación es el proceso mediante el que se identifican los requerimientos que determinan las características deseadas y las restricciones que deberá cumplir el sistema software, que tendrán efectos satisfactorios para el usuario.

Esto es referido a la captura y descubrimiento de los requisitos que deberán ser realizados. Es una actividad más “humana” que técnica porque está dominada por factores humanos, sociales y organizacionales. Se identifica a los stakeholders (dueños del proceso) que poseen diferentes objetivos individuales, organizacionales y diversos background también se establecen las primeras relaciones entre ellos.

Los stakeholders del sistema pueden tener background técnico y no técnico y ser de diferentes disciplinas.

El objetivo es la obtención de una especificación preliminar detallada de las necesidades de los usuarios del software a desarrollar. En cuanto a esta etapa, la propuesta metodológica que se considera apropiada consta de las siguientes tareas:

1.1 Obtener información sobre el problema, el dominio del problema y sistemas existentes.

Objetivos

- Conocer el dominio del problema.
- Conocer la situación actual.

Descripción

Antes de mantener las reuniones con los clientes y usuarios e identificar los requisitos es fundamental conocer el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual.

Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios.



Por otro lado, mantener reuniones con clientes y usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo se vea deteriorada enormemente.

Esta tarea es opcional, ya que puede que no sea necesario realizarla si el equipo de desarrollo tiene experiencia en el dominio del problema y el sistema actual es conocido.

Productos internos

- Información recopilada: libros, artículos, folletos comerciales, desarrollos previos sobre el mismo dominio, etc.
- Modelos del sistema actual si existiera.

Productos entregables

- Información del dominio del problema y del problema en sí.

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 requerimientos como el estudio de documentación, Análisis Social, cuestionarios, etc.

1.2 Preparar y realizar las sesiones de elicitación/ negociación

Objetivos

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

Descripción

Teniendo en cuenta la información recopilada en la tarea anterior, en esta tarea se deben preparar y realizar las reuniones con los clientes y usuarios participantes con objeto de obtener sus necesidades y resolver posibles conflictos que se hayan detectado en iteraciones previas del proceso.



Esta tarea es especialmente crítica y ha de realizarse con especial cuidado, ya que generalmente el equipo de desarrollo no 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 no saben qué necesita saber el equipo de desarrollo para llevar a cabo su labor.

Productos internos

- Notas tomadas durante las reuniones, transcripciones o actas de reuniones, formularios, grabaciones en cinta o vídeo de las reuniones o cualquier otra documentación que se considere oportuna.

Productos entregables

- Participantes en el proyecto, en concreto los usuarios participantes, como parte del Documento de Requerimientos
- Objetivos, requerimientos o conflictos, que se hayan identificado durante las sesiones de elicitación, como parte del Documento de Requerimientos.

Técnicas recomendadas

- Entrevistas.
- Joint Application Design (JAD),
- Casos de uso.
- Prototipado.
- Observaciones y Análisis Social.
- Reuso de requerimientos.
- Plantillas y patrones lingüísticos.

1.3 Identificar/ revisar los objetivos del sistema

Objetivos

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



Descripción

A partir de la información obtenida en la tarea anterior, en esta tarea se deben identificar qué objetivos se esperan alcanzar una vez que el sistema software a desarrollar se encuentre en explotación o revisarlos en función de los conflictos identificados. Puede que los objetivos hayan sido proporcionados antes de comenzar el desarrollo.

Productos internos

- No hay productos internos

Productos entregables

- Objetivos del sistema como parte del Documento de requerimientos.

Técnicas recomendadas

- Plantilla para especificar los objetivos del sistema

1.4 Identificar/ Revisar los requerimientos funcionales

Objetivos

- Identificar los actores del sistema del software a desarrollar.
- Identificar los requisitos funcionales (casos de uso) que deberá cumplir el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos funcionales previamente identificados.

Descripción

A partir de la información obtenida en las tareas 1.1 y 1.2, y teniendo en cuenta los objetivos identificados en la tarea 1.3 y el resto de los requerimientos, en esta tarea se debe identificar, o revisar si existen conflictos, qué debe hacer el sistema a desarrollar con la información identificada en la tarea anterior. Inicialmente se identificarán los actores que interactuarán con el sistema, es decir aquellas personas u otros sistemas que serán los orígenes o destinos de la información que consumirá o producirá el sistema a desarrollar y que forman su entorno.



Se identificarán los casos de uso asociados a los actores, los pasos de cada caso de uso y posteriormente se detallarán los casos de uso con las posibles excepciones hasta definir todas las situaciones posibles.

Productos internos

- No hay productos internos en esta tarea.

Productos entregables

- Actores y Requisitos funcionales como parte del Documento de requerimientos.

Técnicas recomendadas

- Casos de uso.
- Plantilla para actores.
- Plantilla para los requisitos funcionales.

1.5 Identificar/revisar los requerimientos no funcionales

Objetivos

- Identificar los requisitos no funcionales del sistema software a desarrollar.

Descripción

A partir de la información obtenida en las tareas 1.1 y 1.2, y teniendo en cuenta los objetivos identificados en la tarea 1.3 y el resto de los requerimientos, en esta tarea se deben identificar, o revisar si existen conflictos, los requisitos no funcionales, normalmente de carácter técnico o legal. Para ayudar a identificar estos requerimientos se recomienda usar las métricas para requerimientos no funcionales:

Productos internos

- No hay productos internos en esta tarea.

Productos entregables

- Requisitos no funcionales del sistema como parte del Documento de Requerimientos.



Técnicas recomendadas

- Plantilla para requisitos no funcionales
- Métricas para especificar requerimientos no funcionales

Entregables de la fase

Un borrador del documento de requerimientos.

2. Análisis y Negociación de Requerimientos

2.1 Análisis de requerimientos

Descripción y Objetivo

El análisis de requerimientos es parte de la segunda etapa propuesta y es la etapa que provee feedback a los stakeholders para resolver los problemas a través del proceso de negociación. El análisis está entrelazado con la Elicitación, los problemas son descubiertos cuando los requerimientos son elicitados. El objetivo de análisis es descubrir problemas, incompletez e inconsistencias en los requerimientos elicitados. Las tareas del análisis de requerimiento son las siguientes:

2.1.1 Tarea 1: Chequear las necesidades

Objetivo

- Verificar que los requerimientos contribuyen a los objetivos del negocio de la organización.
- Conformidad al problema específico de ser atendido por el sistema.

Descripción

Esta consiste en la verificación que los requerimientos obtenidos estén de acuerdo a los objetivos del negocio de la organización o al problema específico a ser atendido por el sistema.

Productos internos

- Listado de validación de esta etapa.



Productos entregables

- No hay productos entregables en esta tarea.

Técnicas recomendadas:

- Checklist de Análisis

2.1.2 Tarea 2: Chequear si son consistentes y Completos

Objetivos

- Verificar que los requerimientos sean consistente.
- Verificar que los requerimientos sean completos.

Descripción

Los requerimientos deben ser chequeados de forma cruzada para ver si son completos y consistentes, consistencia significa que los requerimientos no debe ser contradictorios y completo significa que ningún servicio o restricción que es necesario este ausente.

Técnicas recomendadas:

- Checklist de Análisis.
- Matriz de Interacción de requerimientos.

2.1.3 Tarea 3: Chequear su factibilidad

Objetivos

- Evaluar el requerimiento de acuerdo al contexto del presupuesto y cronograma disponible.

Descripción

Los requerimientos deben ser chequeados para asegurar que ellos son factibles de implementar en el contexto del presupuesto y cronograma disponible para el desarrollo del sistema.



Productos internos

- Requerimientos factibles de implementación.

Productos entregables

- No hay productos entregables en esta tarea.

2.2 Negociación de requerimientos

Descripción y Objetivo

El proceso de negociación involucra las reuniones de negociación para tratar de encontrar acuerdos en conjunto en relación a los requerimientos en conflicto. Para realizar las reuniones de negociación se deben dividir estos en etapa de información, discusión y resolución, como se observa en la siguiente explicación:

- **Etapa de información:** Aquí se explica la naturaleza de los problemas asociados con un requerimiento.
- **Etapa de discusión:** Donde los stakeholders involucrados discuten como esos problemas pueden ser resueltos.
- **Etapa de resolución:** Donde acciones concernientes al requerimiento son consentidas. Estas acciones pueden ser eliminar el requerimiento, sugerir modificaciones específicas a los requerimientos o elicitar mayor información acerca del requerimiento

El objetivo de esta etapa es discutir los requerimientos en conflictos y alcanzar un compromiso de acuerdo entre todos los stakeholders.

Las tareas de negociación de requerimiento son:

2.2.1 Tarea 1: Discusión de Requerimientos

Objetivos

- Discutir requerimientos con problemas.

Descripción

Los requerimientos que han sido catalogados como problemáticos son discutidos, donde los stakeholders involucrados presentan sus puntos de vista.



Productos internos

- Listado de requerimientos en disputa.

Productos entregables

- No hay productos entregables en esta tarea.

Técnicas recomendadas

- Sesiones Jad
- Reuniones de negociación

2.2.2 Tarea 2: Priorización de requerimientos

Objetivos

- Priorización de requerimientos en conflicto.

Descripción

Requerimientos en disputa son priorizados para identificar requerimientos críticos, para ayudar al proceso de toma de decisiones.

Productos internos

- Informe de priorización de requerimientos.

Productos entregables

- No hay productos entregables en esta tarea.

Técnicas recomendadas

- Reuniones de negociación

2.2.3 Tarea 3: Concordancia de requerimientos

Son identificadas soluciones a los requerimientos en problemas, un conjunto de compromisos acordado. Generalmente, involucra hacer cambios a algunos de los requerimientos.



Productos entregables

- Compromisos acordados en base a los requerimientos en conflicto.

Técnicas recomendadas

- Sesiones Jad.
- Reuniones de negociación

3. Especificación de Requerimientos

Descripción y Objetivo

La especificación de requerimientos, conocida también como la documentación de requerimientos, es el modo habitual de guardar y comunicar requerimientos (en composición grupal).

El objetivo de esta actividad es obtener un documento de especificación que defina, de forma completa, precisa y verificable, los requisitos que debe cumplir el sistema, tanto funcionales como no funcionales. Debe abordar la descripción de lo que hay que desarrollar, no el cómo ni el cuándo. No debe incluir requisitos innecesarios, no solicitados por el cliente, ni incluir detalles sobre el diseño del sistema. El formato del documento de Requerimientos recomendados a seguir es el siguiente:

El Documento de Requerimientos

Documento formal utilizado para comunicar los requerimientos a clientes, ingenieros y gerentes.

Describe:

- Servicios y funciones que el sistema debe proveer.
- Restricciones bajo las cuales el sistema debe operar.
- Propiedades globales del sistema, restricciones sobre las propiedades emergentes del sistema.
- Definiciones de otros sistemas con el cual el sistema debe integrarse si existiese.
- Información sobre el dominio de la aplicación del sistema..
- Descripción del hardware sobre el cual el sistema se ejecutará.



El formato propuesto para el Documento de Requerimientos se ha tomado del Standard IEEE/ANSI 830-1993 el cual propone una estructura para documento de requerimientos de software.

El estándar IEEE es genérico, puede aplicarse a un amplio rango de documentos de requerimientos. En general no todas las partes del estándar son requeridas para todo el documento de requerimientos, cada organización debe adoptar el estándar dependiendo del tipo de sistema a desarrollar.

Formato de referencia para el documento de requerimientos.

1. Introducción.
 - 1.1 Propósito del documento de requerimientos.
 - 1.2 Alcance del producto.
 - 1.3 Definiciones, acrónimos y abreviaciones.
 - 1.4 Referencias.
 - 1.5 Resumen del documentos.
2. Descripción general.
 - 2.1 Perspectiva del producto.
 - 2.2 Funciones del producto.
 - 2.3 Características del usuario.
 - 2.4 Restricciones generales.
 - 2.5 Asunciones y dependencias.
3. Requerimientos Específicos
Requerimientos funcionales, no-funcionales y de interface.
4. Apéndice

Índice

El estándar IEEE es genérico, puede aplicarse a un amplio rango de documentos de requerimientos. En general no todas las partes del estándar son requeridas para todo el documento de requerimientos, Cada organización debe adoptar el estándar dependiendo del tipo de sistema a desarrollar.

Entregable de esta etapa

- Borrador final del documento de requerimientos.



4. Verificación de Requerimientos

Descripción y objetivo

La Verificación de los requerimientos es el proceso para comprobar que el documento de requerimiento se ajusta a las necesidades de clientes/usuarios y otros implicados, es decir, es comprobar con los stakeholders que sus necesidades fueron adecuadamente interpretadas.

Dentro del proceso de verificación también se desarrollan actividades de verificación de requerimientos, con el fin de comprobar que el documento de requerimientos se construye de acuerdo a los criterios y estándares establecidos, o sea, correctamente. La verificación no puede hacerse sin la participación y presencia de clientes, usuarios y demás implicados.

Los objetivos de esta actividad son la comprobación de la consistencia, completitud, corrección, precisión del documento de requerimientos, así como el descubrimiento de problemas en él antes de comprometer recursos en su implementación, también se comprueba que el documento tenga el formato adecuado y este de acuerdo a los estándares de la organización.

La Verificación trabaja con un borrador final del documento de requerimientos, con requerimientos negociados y acordados. Las tareas a realizar son las siguientes:

4.1 Revisar el Documento de Requerimientos.

Objetivos

- Verificar que los requerimientos sean consistentes, completos, correctos y precisos.
- Verificar que el documento de requerimientos estén de acuerdo a los estándares y libre de errores técnicos.

Descripción

Para llevar a cabo esta tarea se debe realizar una inspección detallada de requerimientos, para ello se recomienda usar un Checklist de que observar para dirigir el proceso de inspección de requerimientos, aunque previamente una Pre-Inspección



sería ideal ya que las inspecciones son caras porque ellas envuelven un número de personas consumiendo tiempo leyendo y chequeando el documento de requerimiento. Estos gastos pueden ser reducidos haciendo un chequeo de pre-inspección donde una persona chequea el documento y observa por problemas evidentes como omisión de requerimientos, no conformidad a estándares, errores tipográficos, etc.

Entre otros métodos de Verificación explicados en 6.3, se recomienda usar prototipos, teste de requerimientos, otra manera de validación también puede ser el desarrollo del manual de usuario ya que permite el análisis detallado de estos.

Productos Internos

- Estándares organizacionales.
- Documento de requerimientos.

Productos entregables

- Lista de problemas encontrados.
- Requerimientos validados.

Técnicas recomendadas

- Inspección de requerimientos.
- Checklist de Inspecciones.
- Otras técnicas como desarrollo de Prototipos, teste de requerimientos, desarrollo del manual de usuario

4.2 Cerrar la versión de los requisitos

Objetivos

- Obtener el documento de Requerimientos final del sistema.

Descripción

Esta tarea se lleva a cabo cuando ya no han aparecido nuevos conflictos durante el proceso de validación entonces se debe llegar a un acuerdo entre clientes y desarrolladores para cerrar la versión actual de los requerimientos, siempre teniendo en



cuenta que representa el conocimiento actual de los mismos y que, probablemente, sufrirá cambios en el futuro.

Productos entregables

- Documento de Requerimientos validado.

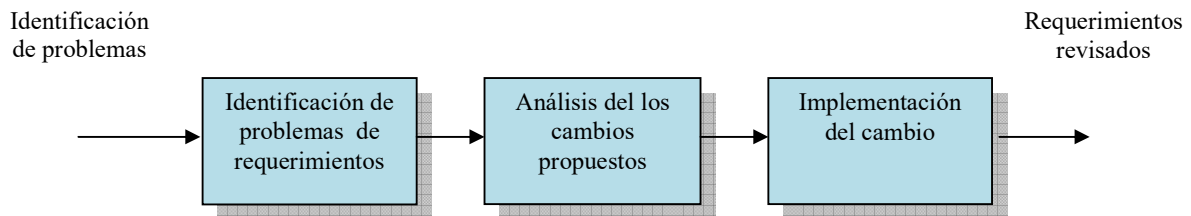
5. Administración de Requerimientos

Descripción y Objetivos

La Administración de Requerimientos es el conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento de ciclo de vida del software. Los cambios a los requerimientos ocurren mientras los requerimientos están siendo elicitados, analizados y validados y después que el sistema es puesto en servicio.

Para llevar esta actividad se propuso el uso de la herramienta OSRMT el cual permite realizar la rastreabilidad, importante para determinar los requerimientos afectados por los cambios, otra posibilidad es usar una tabla de rastreabilidad para implementar el seguimiento.

Esta etapa tiene como objetivo: identificar, controlar y seguir los requerimientos y sus cambios durante la vida del proyecto. Las tareas recomendadas a seguir están esquematizadas en el siguiente grafico:



5.1 Etapas de la Administración de cambio

Fuente: SOMMERVILLE(1997).



5.1. Identificación de problemas de los requerimientos.

Objetivos

- Identificar problemas en los requerimientos o nuevas necesidades.

Descripción

Puede venir del análisis de requerimientos, nuevas necesidades de los clientes o de problemas operacionales con el sistema. Los requerimientos son analizados utilizando información del problema y posteriormente son propuestos cambios a los requerimientos. Los cambios propuestos pueden ser registrados en un formulario de solicitud de cambios el cual es pasado a todas las personas involucradas en el análisis de cambio.

El formulario de solicitud de cambios debe incluir:

- Campos para documentar el análisis de cambio
- Campos de datos
- Campos de responsabilidad
- Campos de estatus
- Campos de comentarios

Productos Internos

- Formularios de solicitud de cambios.

Productos Entregables

- Cambios propuestos en los requerimientos.

Técnicas recomendadas

- Uso de la herramienta case OSRMT.

5.2. Análisis de los cambios propuestos

Objetivos

- Determinar los requerimientos afectados por el cambio.
- Determinar el impacto de los cambios en tiempo y recursos.



Descripción

En esta etapa se chequea cuantos requerimientos y si es necesario, componentes del sistema son afectados por el cambio, estimando cuanto podría costar, en tiempo y dinero hacer el cambio. Para ayudar a evaluar el impacto de los cambios en los requerimientos se puede utilizar las tablas de rastreabilidad de no disponer de una herramienta case para este fin.

Productos Internos

- Estimación de los Cambios propuestos en los requerimientos.

Productos Entregables

- Rastreabilidad de requerimientos afectados por los cambios.

Técnicas recomendadas

- Tablas de rastreabilidad.
- Uso de la herramienta case OSRMT.

5.3. Implementación del cambio

Objetivos

- Implementar los cambios de requerimientos.
- Producir un conjunto de enmiendas al documento de requerimientos. o una nueva versión del mismo.

Descripción

Es producido un conjunto de enmiendas al documento de requerimientos o una nueva versión del mismo. Este debe, de hecho, ser validado usando cualquier procedimiento normal de chequeo de calidad.

Actividades del Análisis del Cambio

- La solicitud de cambio es chequeada para validarla. Los Clientes pueden malentender requerimientos y sugerir innecesariamente cambios.
- Es descubierto aquellos requerimientos que son directamente afectados por el cambio.



- Información de rastreabilidad es usada para encontrar requerimientos dependientes afectados por el cambio.
- Son propuestos los verdaderos cambios que deben ser hechos a los requerimientos.
- Son estimados los costos de hacer los cambios.
- Son mantenidas negociaciones con los clientes para chequear si los costos de los cambios propuestos son aceptados.

Productos Internos

- Solicitudes de cambio de acuerdo al formato.

Productos Entregables

- Nueva versión de requerimientos.
- Nueva versión del documento de requerimientos.

6. Técnicas Recomendadas

6.1 Técnicas de Elicitación de requerimientos

Las técnicas más habituales en la elicitación de requisitos son las entrevistas, el Joint Application Development (JAD) o Desarrollo Conjunto de Aplicaciones, la utilización de escenario, más conocidos como casos de uso y reúso de requerimientos. A estas técnicas, que se describen en los siguientes apartados, se las suele apoyar con otras técnicas complementarias como la observación in situ y Análisis Social.

6.1.1 Entrevistas

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más *naturales* entre personas, donde el analista o ingeniero de requerimientos discute el sistema con diferentes stakeholders y crea un entendimiento de sus requerimientos. Los tipos de entrevistas a aplicarse son:

a) Entrevistas Cerradas: El ingeniero de requerimientos busca repuestas a un conjunto de preguntas predefinidas.



b) Entrevistas abiertas: No se tiene agenda predefinida, el ingeniero de requerimientos discute, de modo abierto, lo que los stakeholders quieren del sistema.

En las entrevistas se pueden identificar tres pasos: preparación, realización y análisis.

Paso 1: Preparación de entrevistas

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguientes tareas previas:

- **Estudiar el dominio del problema:** conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas, y para generar en los clientes y usuarios la confianza de que el ingeniero de requisitos *entiende* sus problemas.

Para conocer el dominio del problema se puede recurrir a técnicas de estudio de documentación, a bibliografía sobre el tema, documentación de proyectos similares realizados anteriormente, la *inmersión* dentro de la organización para la que se va a desarrollar o a periodos de *aprendizaje* por partes de los ingenieros de requisitos.

- **Seleccionar a las personas a las que se va a entrevistar:** minimizar el número de entrevistas a realizar, por fundamental seleccionar a las personas a entrevistar. Normalmente comienza por los directivos, que pueden ofrecer una continúa con los futuros usuarios, que pueden aportar más detallada, y con el personal técnico, que aporta entorno operacional de la organización.

Tal como se recomienda en, estudiar el perfil de los entrevistados, buscando con el entrevistador que ayuden a *romper el hielo*.

- **Determinar el objetivo y contenido de las entrevistas:** para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido. Previamente a su realización, se pueden enviar cuestionarios que los futuros entrevistados deben rellenar y devolver, y un pequeño documento de introducción al proyecto de desarrollo, de forma que el entrevistado conozca los temas que se van a tratar y el entrevistador recoja información para preparar la entrevista.



Es importante que los cuestionarios, si se usan, se preparen cuidadosamente teniendo en cuenta quién los va a responder y no incluir conceptos que se asuman conocidos cuando puedan no serlo.

- **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.

Paso 2: Realización de entrevistas

Dentro de la realización de las entrevistas se distinguen tres etapas, tal como se expone en:

Tarea 1: Apertura: el entrevistador debe presentarse e informar al entrevistado sobre la razón de la entrevista, qué se espera conseguir, cómo se utilizará la información, la mecánica de las preguntas, etc. Si se va a utilizar algún tipo de notación gráfica o matemática que el entrevistado no conozca debe explicarse antes de utilizarse. Es fundamental causar buena impresión en los primeros minutos.

Tarea 2: Desarrollo: la entrevista en si no debería durar más de dos horas, distribuyendo el tiempo en un 20% para el entrevistador y un 80% 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" o "¿Cómo se rellena un albarán?".
- 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 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.

Tarea 3. Terminación: al terminar la entrevista se debe recapitular para confirmar que no ha habido confusiones en la información recogida, agradecer al entrevistado su colaboración y citarle para una nueva entrevista si fuera necesario, dejando siempre abierta la posibilidad de volver a contactar para aclarar dudas que surjan al estudiar la información o al contrastarla con otros entrevistados.

Paso 3: Análisis de las entrevistas

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar 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.

6.1.2. Joint Application Development

La técnica denominada *JAD* (*Joint Application Development, Desarrollo Conjunto de Aplicaciones*), 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 periodo 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 técnica se base en cuatro principios: 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 *WYSIWYG* (*What You See Is What You Get*, lo que se ve es lo que se obtiene), por 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 elicitación y especificación de 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 cuando se aplica suele tener buenos resultados, especialmente para elicitación de 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

Tal como se expone en, se pueden distinguir seis clases de participantes o roles en el JAD:

- Jefe del JAD:** es el responsable de todo el proceso y asume el control durante las reuniones. Debe tener dotes de comunicación y liderazgo. Algunas habilidades importantes que debe tener son: entender y promover la dinámica de grupo, iniciar y centrar discusiones, reconocer cuándo la reunión se está desviando del tema y reconducirla, manejar las distintas personalidades y formas de ser de los participantes, evitar que decaiga la reunión aunque sea larga y difícil, etc.
- Analista:** es el responsable de la producción de los documentos que se deben generar durante las sesiones JAD. Debe tener la habilidad de organizar bien las ideas y expresarlas claramente por escrito. En el caso de que se utilizan herramientas software durante las sesiones, debe ser capaz de manejarlas eficientemente.



- c. **Patrocinador ejecutivo:** es el que tiene la decisión final de que se lleve a cabo el desarrollo. Debe proporcionar a los demás participantes información sobre la necesidad del nuevo sistema y los beneficios que se espera obtener de él.

- d. **Representantes de los usuarios:** durante el *JAD/Plan*, suelen ser directivos con una visión global del sistema. Durante el *JAD/Design* suelen incorporarse futuros usuarios finales.

- e. **Representantes de sistemas de información:** son personas 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.

- f. **Especialistas:** son personas que pueden 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.

Fases del JAD

Dentro de la técnica del JAD se distinguen tres fases:

Fase 1: 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.

Fase 2: 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:

a. Presentación: se presenta y 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.

b. Definir objetivos y requisitos: el jefe del JAD promueve la discusión para elicitación de 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.

c. 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*).

d. 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, para lo cual puede utilizarse plantillas.

e. 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.

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

a. 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.

b. Revisar la documentación: 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.

c. 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.

6.1.3 Casos de uso

Aunque inicialmente se desarrollaron como técnica para la definición de requisitos, algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requerimientos funcionales expresados como casos de uso) de un sistema.

Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades.



6.1.4. Prototipos

Un prototipo es una versión inicial de un sistema que puede ser utilizado para experimentación. Los prototipos son valiosos para la Elicitación de requerimientos porque los usuarios pueden experimentar con el sistema y resaltar sus fortalezas y debilidades. Se tiene algo concreto para criticar.

El desarrollo rápido de prototipos es esencial para que estén tempranamente disponibles en el proceso de elicitación. Los beneficios que se obtienen son que el prototipo permite a los usuarios experimentar y descubrir lo que ellos realmente necesitan para apoyar su trabajo, establece factibilidad y usabilidad antes que se incurran en altos costos de desarrollo, Puede ser usado para testear el sistema y desarrollo de documentación, y fuerza un estudio detallado de requerimientos, lo que puede revelar inconsistencias y omisiones.

Tipos de prototipos :

- a. **Prototipos Descartables.** Ayudan a elicitar y desarrollar los requerimientos del sistema. Los requerimientos que deben ser prototipados son aquellos que causan mayores dificultades a los clientes y que son más difíciles de entender. Requerimientos que son bien entendidos no necesitan ser implementados
- b. **Prototipos Evolutivos.** Permiten entregar rápidamente un sistema trabajable al cliente. En consecuencia. Los requerimientos que deben ser soportados por las versiones iniciales de este prototipo son aquellos que son bien entendidos y que pueden entregar funcionalidades útiles al usuario final. Solo después de su uso extensivo es que los requerimientos poco entendidos serán implementados.

Los abordajes para estos prototipos son:

- **Prototipos en el papel:** Desarrolla y utiliza un Modelo estructural de tamaño completo del sistema para experimentar el sistema
- **Prototipos “Mago de Oz”:** Una persona simula las repuestas del sistema en respuesta a algún input del usuario.



- **Prototipos Ejecutables** :Para el desarrollo de prototipos ejecutables se pueden emplear lenguajes de Cuarta Generación basados en sistemas de base de datos, lenguajes de programación Visual tales como Visual Basic, ObjectWorks, etc, también soluciones de prototipos basados en Internet sobre navegadores y lenguajes World Wide Web.

6.1.5 Observación y Análisis Social

Las personas frecuentemente encuentran difícil describir lo que ellos hacen porque eso es natural a ellos. Algunas veces, la mejor manera de entenderlo es observarlos en el trabajo. La aplicación de la Etnografía, técnica de las ciencias sociales que ha probado ser valiosa para entender los procesos reales de trabajo. Porque los procesos reales de trabajo frecuentemente difieren de los procesos formales prescritos. El etnógrafo debe gastar algún tiempo observando a las personas en el trabajo y crear una foto de cómo el trabajo es echo.

La etnografía asume que las personas son buenas en hacer sus trabajos, se fija en las maneras no-estándar de trabajar, gasta tiempo tratando de conocer a las personas y establecer una relación de confianza. Mantener nota detallada de todas las practicas de trabajo. A partir de estas analizarlas y bosquejar conclusiones.

Se puede combinar la observación con entrevistas abiertas formales – donde el etnógrafo habla con las personas fuera del proceso de trabajo, también se puede combinar con otras técnicas de Elicitación como el prototipado.

La etnografía se puede enfocar desde estas perspectivas:

a. Punto de Vista: Ejecutando el trabajo

Describe el contexto y la ubicación física del trabajo y como las personas usan los objetos para ejecutar tareas.

b. Perspectiva Social y Organizacional

Intenta expresar en palabras la experiencia de trabajo del día a día como vista por las diferentes personas involucradas, Cada individuo típicamente ve el trabajo de



una manera diferente, este punto de vista intenta organizar e integrar todas estas percepciones.

c. Punto de Vista: Workflow- Flujos de Trabajo

Presenta el trabajo como una serie de actividades de trabajo, con Información fluyendo desde una actividad a otra.

6.1.6 Reuso de Requerimientos

Esta técnica envuelve tomar los requerimientos que han sido desarrollados para un sistema y usarlos en un sistema diferente. El reuso de requerimientos ahorra tiempo y esfuerzo, debido a que los requerimientos re-usados ya han sido analizados y validados en otros sistemas

Actualmente, reuso de requerimientos en un proceso informal. Pero un reuso más sistemático podría conducir a grandes ahorros de dinero. Las posibilidades de reuso se encuentran cuando el requerimiento esta focalizado con proveer Información del dominio del aplicación, focalizado con el estilo de la presentación de la Información. El reuso conduce a una consistencia del estilo a través de las aplicaciones. Cuando los requerimientos reflejan políticas de la compañía tales como políticas de la compañía tales como las políticas de seguridad.

6.1.7 Plantillas y patrones lingüísticos para elicitación de requisitos

Las plantillas y patrones lingüísticos que se presentan en los siguientes apartados están pensados para utilizarse tanto durante las reuniones de elicitación con clientes y usuarios como para registrar y gestionar los requisitos.

Su objetivo es doble: por un lado intentar paliar la falta de propuestas concretas sobre la expresión de requisitos. Por otro lado, también pueden usarse como elementos de elicitación y negociación durante las reuniones con clientes y usuarios. De esta forma se consigue que durante las sesiones de elicitación se trabaje con una filosofía *WYSIWYG*, tal como se propone en las técnicas de JAD o brainstorming, ya que los participantes manejan directamente la documentación final, favoreciéndose así su implicación en el proceso.



Para algunos campos de las plantillas se han identificado frases "estándar" que son habituales en las especificaciones de requisitos y que se han parametrizado. Estas frases, son denominados *patrones lingüísticos*, o abreviadamente *patrones-L*, pueden usarse para rellenar los campos de las plantillas dándole valores a los parámetros con la información oportuna. Ambos aspectos, la estructuración de la información en forma de plantilla y la propuesta de frases "estándar", facilita la redacción de los requisitos, permitiendo a los participantes en las actividades de elicitación centrarse en expresar sus necesidades y no en cómo expresarlas. En la notación usada para describir los patrones-L, las palabras o frases entre < y > deben ser convenientemente reemplazadas, mientras que las palabras o frases que se encuentren entre { y } y separadas por comas representan opciones de las que se debe escoger una. En las siguientes secciones se describe las plantillas y los patrones-L.

Figura 1: Plantilla y patrones-L para objetivos

OBJ-<id>	< <i>nombre descriptivo</i> >
Versión	< <i>no de la versión actual</i> > (< <i>fecha de la versión actual</i> >)
Autores	<ul style="list-style-type: none"> • <<i>autor de la versión actual</i>> (<<i>organización del autor</i>>) • ...
Fuentes	<ul style="list-style-type: none"> • <<i>fuelle de la versión actual</i>> (<<i>organización de la fuente</i>>) • ...
Descripción	El sistema deberá < <i>objetivo a cumplir por el sistema</i> >
SubObjetivos	<ul style="list-style-type: none"> • OBJ-x <<i>nombre del subobjetivo</i>> • ...
Importancia	< <i>importancia del objetivo</i> >
Urgencia	< <i>urgencia del objetivo</i> >
Estado	< <i>estado del objetivo</i> >
Estabilidad	< <i>estabilidad del objetivo</i> >
Comentarios	< <i>comentarios adicionales sobre el objetivo</i> >

El significado de los campos que la componen, cuya mayoría está presente también en las plantillas para los requisitos, es el siguiente:



- **Identificador y nombre descriptivo:** siguiendo la propuesta, entre otros, cada objetivo debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con *OBJ*.
- **Versión:** para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.
- **Autores, Fuentes:** estos campos contienen el nombre y la organización de los autores (normalmente desarrolladores) y de las fuentes (clientes o usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del requisito.
- **Descripción:** este campo contiene un patrón-L que se debe completar con la descripción del objetivo.
- **Subobjetivos:** en este campo pueden indicarse los subobjetivos que dependen del objetivo que se está describiendo. En sistemas complejos puede ser necesario establecer una jerarquía de objetivos previa a la identificación de los requisitos. En caso de que esto no sea necesario, puede ignorarse este campo.
- **Importancia:** este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como *vital*, *importante* o *quedaría bien*. En el caso de que no se haya establecido aún la importancia, se puede indicar que está *por determinar (PD)*, equivalente al TBD (*To Be Determined*) empleado en las especificaciones escritas en inglés.
- **Urgencia:** este campo indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Como en el caso anterior, se puede asignar un valor numérico o una expresión enumerada como *inmediatamente*, *hay presión* o *puede esperar*, o *PD* en el caso de que aún no se haya determinado.
- **Estado:** este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar *en construcción* si se está elaborando, *pendiente de negociación* si tiene algún conflicto asociado pendiente de solución, *pendiente de validación* si no tiene ningún conflicto pendiente y está a la espera de validación o, por último, puede estar *validado* si ha sido validado por clientes y usuarios.



- **Estabilidad:** este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como *alta*, *media* o *baja* o *PD* en el caso de que aún no se haya determinado. La información sobre la estabilidad, bien a nivel de objetivos como en este caso, bien a nivel de requisitos, ayuda a los diseñadores a diseñar software que prevea de antemano la necesidad de posibles cambios futuros en aquellos aspectos relacionados con los elementos identificados como inestables durante la fase de ingeniería de requisitos, favoreciendo así el mantenimiento y la evolución del software.
- **Comentarios:** cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

Plantilla para requisitos funcionales

El significado de los campos específicos de esta plantilla es el siguiente :

- **Identificador y nombre descriptivo:** igual que en la plantilla anterior, excepto que los identificadores de los requisitos funcionales empiezan con *RF* y que el nombre descriptivo suele coincidir con el objetivo que los actores esperan alcanzar al realizar el caso de uso. No se debe confundir este objetivo con los objetivos del sistema. El objetivo que los actores esperan alcanzar al realizar un caso de uso es de más bajo nivel, por ejemplo *registrar un nuevo socio* o *consultar los pedidos pendientes*.

Figura 2: Plantilla y patrones-L para requisitos funcionales

RF-<i><id></i>	<nombre descriptivo>
Versión	<no de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-x <nombre del objetivo> • ...
Requisitos asociados	<ul style="list-style-type: none"> • Rx-y <nombre del requisito>



	• ...	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { durante la realización de los casos de uso <lista de casos de uso>, cuando <evento de activación> }	
Precondición	<precondición del caso de uso>	
Secuencia Normal	Paso	Acción
	P1	{El actor <actor>, El sistema} <acción/es realizada/s por actor/sistema>
	P2	Se realiza el caso de uso <caso de uso (RF-x)>
	P3	Si <condición>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>
	P4	Si <condición>, se realiza el caso de uso <caso de uso (RF-x)>
	...	
Postcondición	<postcondición del caso de uso>	
Excepciones	Paso	Acción
	Pi	Si <condición de excepción>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>, a continuación este caso de uso {continúa, termina}
	Pj	Si <condición de excepción>, se realiza el caso de uso <caso de uso (RF-x)>, a continuación este caso de uso {continúa, termina}
Rendimiento	Paso	Cuota de tiempo
	q	m <unidad de tiempo>
	...	
Frecuencia esperada	<no de veces> veces / <unidad de tiempo>	
Importancia	<importancia del requisito>	
Urgencia	<urgencia del requisito>	
Estado	<estado del requisito>	
Estabilidad	<estabilidad del requisito>	
Comentarios	<comentarios adicionales sobre el requisito>	

- **Descripción:** para los requisitos funcionales, este campo contiene un patrón-L que debe completarse de forma distinta en función de que el caso de uso sea abstracto o



concreto. Si el caso de uso es abstracto, deben indicarse los casos de uso en los que se debe realizar, es decir, aquellos desde los que es *incluido* o a los que *extiende*. Si, por el contrario, se trata de un caso de uso concreto, se debe indicar el *evento de activación* que provoca su realización.

- **Precondición:** en este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso.
- **Secuencia normal:** este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza una o más acciones, o se realiza (se *incluye*) otro caso de uso. Un paso puede tener una condición de realización, en cuyo caso si se realizara otro caso de uso se tendría una relación de *extensión*. Se asume que, después de realizar el último paso, el caso de uso termina. Otras propuestas similares. Para representar estructuras condicionales complejas se puede recurrir a añadir información aparte, por ejemplo una tabla de decisión, y referenciarla desde el paso o los pasos oportunos. En el caso de estructuras iterativas, su uso puede evitarse con un uso cuidadoso del lenguaje natural. Por ejemplo, para indicar que se procesan todos los artículos de un pedido se puede optar por frases como "*el sistema procesa todos los artículos del pedido introducidos por el usuario*", en lugar de estructuras como:

REPETIR

procesar artículo del pedido introducido por el usuario

HASTA que no haya más artículos

- **Postcondición:** en este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.
- **Excepciones:** este campo especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado.
- **Frecuencia esperada:** en este campo se indica la frecuencia esperada de realización del caso de uso, que aunque no es realmente un requisito, es una información interesante para los desarrolladores.
- **Plantilla para requisitos no funcionales**
Los requisitos no funcionales del sistema se pueden expresar usando la plantilla que puede verse en la figura 3. El único campo específico de esta plantilla es la



descripción, en la que se usa un patrón–L que debe completarse con la capacidad que deberá presentar el sistema, el significado del resto de los campos es el mismo que para las plantillas anteriores.

Figura 3: Plantilla y patrones–L para requisitos no funcionales

RNF–<id>	<nombre descriptivo>
Versión	<no de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Objetivos asociados	<ul style="list-style-type: none"> • OBJ–x <nombre del objetivo> • ...
Requisitos asociados	<ul style="list-style-type: none"> • Rx–y <nombre del requisito> • ...
Descripción	El sistema deberá <capacidad del sistema>
Importancia	<importancia del requisito>
Urgencia	<urgencia del requisito>
Estado	<estado del requisito>
Estabilidad	<estabilidad del requisito>
Comentarios	<comentarios adicionales sobre el requisito>

Plantilla para conflictos

El significado de los campos de la plantilla es el siguiente:

Figura 4: Plantilla y patrones–L para conflictos

CFL–<id>	<nombre descriptivo>
Versión	<no de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Objs./Reqs. en conflicto	<ul style="list-style-type: none"> • OBJ/Ryy--x <nombre del objetivo o requisito en conflicto> • ...
Descripción	<descripción del conflicto>



Alternativas	<ul style="list-style-type: none">• <descripción alternativa de solución> (<autores alternativa>)• ...
Solución	<descripción de la solución adoptada (si se ha acordado)>
Importancia	<importancia de la resolución del conflicto>
Urgencia	<urgencia de la resolución del conflicto>
Estado	<estado del resolución del conflicto>
Comentarios	<comentarios adicionales sobre el conflicto>

- **Identificador y nombre descriptivo:** al igual que el resto de la información correspondiente a los requisitos–C, cada conflicto debe poderse identificar de forma única y tener un nombre descriptivo. El prefijo propuesto para lograr una rápida identificación es *CFL*.
- **Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en las plantillas para objetivos y requisitos, aunque referidos al conflicto. En este caso especial, las fuentes son los participantes que deben participar en las posibles negociaciones necesarias para su resolución.
- **Objetivos y requisitos en conflicto:** este campo debe contener una lista con los objetivos y/o requisitos afectados por el conflicto.
- **Descripción:** este campo debe contener la descripción del conflicto.
- **Alternativas:** este campo debe contener una lista con las posibles alternativas de solución que se hayan identificado para solucionar el conflicto así como los autores de dichas alternativas.
- **Solución:** este campo debe contener la descripción de la solución negociada del conflicto, una vez que se haya acordado.
- **Importancia, Urgencia:** estos campos indican respectivamente la importancia y la urgencia de la resolución del conflicto.
- **Estado:** este campo indica el estado de resolución del conflicto, que podrá estar *en negociación* o bien *resuelto*.
- **Comentarios:** este campo tienen el mismo significado que en las plantillas descritas previamente.



6.1.8 Métricas Para Especificar Requerimientos no funcionales

Propiedad	Medida
Rapidez	<ul style="list-style-type: none">• Transacciones procesadas por segundo• Tiempo de respuesta al usuario y a eventos• Tiempo de actualización de la pantalla
Tamaño	<ul style="list-style-type: none">• KB's• Tamaño de Ram
Usabilidad	<ul style="list-style-type: none">• Tiempo de capacitación• Numero de ventanas de ayuda
Fiabilidad	<ul style="list-style-type: none">• Tiempo promedio entre fallas• Probabilidad de no disponibilidad• Tasa de ocurrencia de las fallas• Disponibilidad
Robustez	<ul style="list-style-type: none">• Tiempo de reinicio después de fallas• Porcentaje de eventos que provocan las fallas• Probabilidad de corrupción de los datos después de la fallas
Portabilidad	<ul style="list-style-type: none">• Porcentaje de declaraciones dependientes del objetivo.• Numero de sistemas objetivos

6.2 Técnicas de Análisis de requerimientos

6.2.1 Checklist de Análisis

- Diseño prematuro
¿El requerimiento incluye un diseño prematuro o informaciones de implementación?
- Requerimientos Combinados
¿La descripción de un requerimiento describe un único requerimiento o puede ser dividido en varios diferentes requerimientos?
- Requerimientos Innecesarios
¿Es el requerimiento un 'utensilio de oro'? Esto es, el requerimiento es una adición cosmética al sistema que no es realmente necesaria.



- Uso de hardware no-estándar
¿El requerimiento significa que ningún hardware o software no-estándar debe ser usado? Para tomar esta decisión, es necesario saber los requerimientos de la plataforma de cómputo.
- Conformidad con los objetivos del negocio
¿Es el requerimiento consistente con los objetivos del negocio definidos en la introducción del documento de requerimientos?
- Requerimientos ambiguos
¿Es el requerimiento ambiguo. Puede ser leído de diferentes modos por diferentes personas?, ¿Cuales son las posibles interpretaciones del requerimiento?
- Requerimiento realistas
¿Es el requerimiento sujeto a prueba. Esto es, está declarado de modo tal que los ingenieros de prueba(teste) puedan derivar un teste pueda ser mostrado como si el sistema satisface ese requerimiento?
- Requerimientos sujetos a prueba
¿Es el requerimiento sujeto a prueba, eso es, está declarado de modo tal que los ingenieros de prueba(teste) puedan derivar un teste que pueda ser mostrado como si el sistema satisface ese requerimiento?

6.2.2 Interacciones de requerimientos

Un importante objetivo del análisis de requerimiento es descubrir las interacciones entre requerimientos y resaltar los conflictos y solapamientos de requerimientos. Una matriz de interacción de requerimientos muestra cómo interactúan los requerimientos unos con otros. Los requerimientos son listados a lo largo de las filas y columnas de la matriz.

- Para requerimientos en conflicto, colocar 1
- Para requerimientos que se solapan, colocar 1000
- Para requerimientos que son independientes, colocar 0



Cuadro 1. Interacción de requerimientos

Requerimiento	R1	R2	R3	R4	R5
R1	0	1000	0	0	1
R2	0	0	0	0	0
R3	1000	0	1000	1	1000
R4	0	1000	1	0	1
R5	1	0	1	0	1

Se realiza la suma luego se divide entre /1000, de la operación resultante se considera:

Q(cociente) = Nro de solapamientos.

R(residuo) = Nro de conflictos.

6.3 Técnicas de Validación de Requerimientos

6.3.1 Checklist de Inspecciones

- Entendibilidad – Legibilidad
Pueden los lectores del documento entender lo que significa el requerimiento?
- Redundancia
Hay información necesaria repetida en el documento de requerimiento?
- Completeza
El inspector sabe de algún requerimiento omitido o hay alguna información ausente en las descripciones de los requerimientos?
- Ambigüedad
¿Están los requerimientos expresados con términos claramente definidos?
¿Pueden los lectores con diferente background tener diferentes interpretaciones de los requerimientos?
- Consistencia
¿Las descripciones de los diferentes requerimientos incluyen contradicciones? ¿Hay contradicciones entre los requerimientos y los requisitos globales del sistema?
- Organización
¿Esta el documento estructurado de un modo razonable? Están las descripciones de requerimientos organizadas tal que requerimientos relacionados estén agrupados?

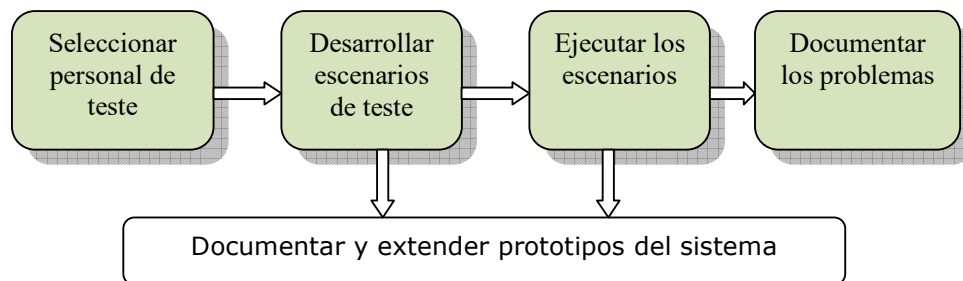


- Conformidad a estándares
¿El documento de requerimientos y cada uno de los requerimientos están de acuerdo a estándares definidos? Están las desviaciones de los estándares, justificados?
- Rastreabilidad
Están los requerimientos identificados sin ambigüedad, incluyen enlaces a requerimientos relacionados y las razones porque éstos han sido incluidos? ¿ Hay un enlace claro entre los requerimiento de software y los requerimientos más generales de Ingeniería de Sistemas?

6.3.2 Prototipos

Prototipos son efectivos para validación de requerimientos testan los requerimientos y ayudan a los stakeholders a descubrir problemas. Los prototipos de validación deben tener las propiedades de ser completos, razonables, eficientes y robustos. Deben ser posibles de usarlos en la misma manera como el sistema requerido. Debe proveerse documentación y entrenamiento al usuario.

Figura 6.1. Proceso de desarrollo de prototipos



Fuente: SOMERVILLE(1997)

Actividades para la validación de prototipos:

- **Seleccionar al personal de teste del prototipo :** El mejor personal de teste son los usuarios quienes tienen la experiencia y mente abierta acerca del uso del nuevo



sistema. Deben ser involucrados usuarios finales que hagan diferentes tareas tal que las diferentes áreas del sistema sean cubiertos.

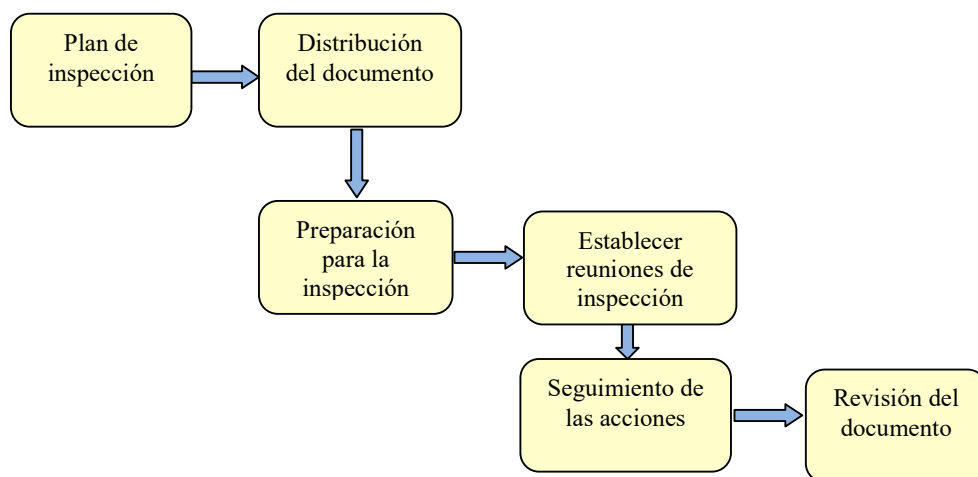
- **Desarrollar escenarios de testeo** : Es necesario planificar el diseño de un conjunto de escenarios de teste que provean una amplia cobertura de los requerimientos.
- **Ejecutar los escenarios** : Los usuarios del sistema trabajan, usualmente sobre sus problemas, para intentar que el sistema ejecute los escenarios planeados.
- **Documentar los problemas** : Es recomendable definir un formulario electrónico o de papel donde se reporte problemas cuando se encuentre alguno.

6.3.3 Inspección de requerimientos

Primero se debe elegir al equipo de inspección. El equipo de inspección debe involucrar un número de stakeholders con diferentes backgrounds, debido a que personas con diferentes background traen diferentes habilidades y conocimiento a la inspección. El equipo de inspección debe siempre involucrar al menos un especialista del dominio y un usuario final.

Después de elegido el equipo de inspección se sigue con las tareas propias del proceso de inspección. Un grupo de personas debe leer y analizar los requerimientos, observar los problemas, encontrar y discutirlos problemas y, acordar acciones para atenderlos. Se recomienda seguir los siguientes pasos de actividades para realizar la inspección de requerimientos.

Figura 6.2. Etapas de la Inspección de Requerimientos



Fuente: SOMERVILLE(1997)



a) Plan de Inspección

Se debe seleccionar el equipo de inspección y debe establecer una hora y lugar para la reunión de inspección.

b) Distribución de documentos

El documento de requerimientos se distribuye a los miembros del equipo de inspección.

c) Preparación del individuo para la inspección

Los miembros del equipo de inspección leen los requerimientos para encontrar conflictos, omisiones, inconsistencias, desviaciones de los estándares y otros problemas.

d) Establecer reuniones de inspección

Los comentarios y problemas encontrados por los miembros del equipo de inspección son discutidos y un conjunto de acciones para tratar los problemas es acordado.

e) Seguimiento de las acciones

El responsable de la inspección chequea que las acciones acordadas hayan sido ejecutadas.

f) Revisión del documento

El documento de requerimientos es revisado tal que refleje las acciones acordadas. En esta etapa, este puede ser aceptado o pasar por una nueva re-inspección.

Los tipos de problemas que suelen aparecer en la inspección y sugerencias.

- **Clarificación de Requerimientos** : El requerimiento puede estar mal expresado o accidentalmente omitida información que fue colectada durante la elicitación de requerimiento.
- **Información Omitida** : Alguna información esta ausente del documento de requerimiento. Es responsabilidad de los ingenieros de requerimientos. Quienes están revisando el documento, descubrir esta información de los stakeholders del sistema o de otras fuentes.



- **Conflicto de Requerimiento** : Hay conflictos significativos entre los requerimientos. Los stakeholders involucrados deben negociar para resolver el conflicto.
- **Requerimientos No Realistas** : Los requerimientos no parecen ser implementables con la tecnología disponible o dada otras restricciones sobre el sistema. Los stakeholders deben ser consultados para decidir cómo hacer los requerimientos más realistas.

6.3.4 Desarrollo de manual de usuario

Escribir un manual de usuario de los requerimientos fuerza un análisis detallado de éstos, así el manual puede ayudar a revelar problemas. La información del manual de usuario incluye descripción de las funcionalidades y como éstas son implementadas, que partes del sistema no han sido implementados, como salir de un problema, como instalar e inicializar el sistema.

6.3.5 Desarrollar Teste requerimientos

El teste de requerimientos nos ayuda a chequear si el requerimiento ha sido satisfecho en el sistema terminado, todo requerimiento debería tener el atributo deseable de ser testeable. Para registrar los testes se pueden usar los formularios de registro de teste.

Inventar testes de requerimientos es una manera efectiva de revelar problemas en los requerimientos como información omitida o ambigua. Si se encontrase dificultades en desarrollar casos de teste, implica que hay algún problema en el requerimiento que podría ser omisión o descripción no clara, Por lo tanto cada requerimiento funcional debe tener un teste asociado.

A continuación unas preguntas que nos ayudaran a definir casos de teste, cuyo propósito es validar el requerimiento no el sistema.

Preguntas para definir los casos de teste

- ¿Cual escenario de uso puede ser utilizado para chequear el requerimiento?
- Define el contexto del teste



- ¿El requerimiento, en sí mismo, incluye suficiente información para definir un teste? Si no que otros requerimientos deben ser examinados? Pueden haber dependencias importantes para la rastreabilidad.
- ¿Es posible testear el requerimiento usando un único teste o es necesario múltiples casos de teste?
- ¿Puede estar embutido más de un requerimiento en una única descripción'.
- ¿Puede el requerimiento ser re-escrito para hacer los casos de teste más obvios?

Formularios de registro de teste

Debe ser diseñado para cada requerimiento e incluye:

- **Identificación de requerimiento**
Al menos uno para cada requerimiento
- **Requerimientos relacionados**
Estos deben ser parafraseados, como que el teste puede también ser relevante a aquellos.
- **Descripción del teste**
Breve descripción del teste que puede ser aplicado y explicar porque este es un teste objetivo. Debe incluir inputs al sistema y los correspondientes outputs.
- **Problema con el requerimiento**
Una descripción de problemas que hicieron imposible o difícil la definición del teste
- **Comentarios y recomendaciones**
Consejos sobre cómo solucionar los problemas de requerimientos encontrados.

Recomendaciones para el diseño de testeo

No preocuparse con cuestiones de costos de teste, testes redundantes, definición de datos detallados de teste, etc.

Hacer los supuestos necesarios acerca de cómo el sistema satisface otros requerimientos y los modos en que el teste puede ejecutarse.

Siempre que es posible, diseñar testes que puedan ser utilizados como teste del sistema.

Éstos pueden ser aplicados después de la implementación como parte de la validación



del sistema. El reuso de testes disminuye los costos totales. Nos podríamos encontrar con requerimientos difíciles de testear, una clasificación a continuación.

Requerimientos del sistema

Requerimientos que aplican al sistema como un todo. En general, éstos son los requerimientos más difíciles de validar, independiente del método usado, pueden ser influenciados por cualquier de los requerimientos funcionales. Los testes no pueden testear características no-funcionales del sistema como un todo como por ejemplo usabilidad.

Requerimientos Especiales

Son requerimientos que no tienen un comportamiento específico. Por ej., un requerimiento puede declarar que las fallas del sistema nunca deben dañar su base de datos. Es imposible testear tal requerimiento exhaustivamente, la confianza se incrementa con el número de testes del sistema.

Algunos Requerimientos No – funcionales

Algunos requerimientos no-funcionales, ejemplo: Confiabilidad, solo puede ser testeados con un conjunto grande de testes. Diseñar eso no ayuda con la validación de requerimientos.

6.4 Técnicas de Gerencia de Requerimientos

6.4.1 Tablas de rastreabilidad

Las tablas de rastreabilidad muestran las relaciones entre requerimientos o entre requerimientos y componentes de diseño. Los requerimientos son listados a lo largo de los ejes horizontal y vertical y las relaciones entre requerimientos son marcadas en las celdas de la tabla.

Para etiquetar las filas y columnas de la tabla de deben usar los números de los requerimientos.



Tabla 1. Tabla de rastreabilidad

	R1	R2	R3	R4	R5
R1			*		*
R2				*	
R3				*	
R4	*				
R5		*			*

Si el número de requerimientos es relativamente pequeño (hasta 250), las tablas de rastreabilidad pueden ser implementadas usando una hoja electrónica. Cuando hay cientos o miles de requerimientos, las tablas de tornan grandes y pobremente pobladas, entonces una forma simplificada de la tabla de rastreabilidad puede ser usada Las listad de rastreabilidad.

Las Listas de rastreabilidad son listas simples de relaciones que pueden ser implementadas como texto o simples tablas.

Tabla 2. Tabla de rastreabilidad resumidas

Requerimientos	Depends on
R1	R3,R5
R2	R4
R3	R4
R4	R1
R5	R2,R5



6. CONCLUSIONES

- ❖ La Metodología para la captura de requerimientos MIRGMD, nos permitió implementar el proceso de la ingeniería de requerimientos de una manera simple y sistemática, orientada a los Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A. , en el desarrollo de un Sistema de Servicio de Soporte de Sistemas realizada para la “Empresa Contratista”.
- ❖ Aplicar la Metodología nos permitió obtener con una mayor eficiencia en la gran mayoría de los requerimientos básicamente mediante las sesiones JAD y entrevistas individuales, todo ello con el uso de las plantillas recomendadas para el registro de los requerimientos y problemas encontrados.
- ❖ Se tuvo buenos resultados en comparación a proyectos anteriores, en los cuales no se uso ninguna metodología de requerimientos MIRGMD, donde los conflictos y otros problemas afectaban negativamente en mayor medida el curso del desarrollo del proyecto generando por lo tanto mayor consumo de recursos y tiempo.
- ❖ El uso de la Herramienta Case OSRMT ayudo a llevar una adecuada gestión de los requerimientos obtenidos, permitiendo visualizar gráficamente la rastreabilidad, lo cual ayudo a ver qué requerimientos resultaban afectados por los cambios surgidos en uno u otro requerimiento y de esta manera finalmente poder estimar el impacto que produciría antes de implementar tales cambios.
- ❖ El tratamiento proactivo de los riesgos asociados a los requisitos del software nos permite adoptar, desarrollar e implementar adecuadamente las actividades de gestión de riesgos, en función de obtener productos de calidad que satisfagan las necesidades del cliente, manteniendo el equilibrio de plazo y costo del proyecto en virtud de lograr un mejor desempeño del proceso de Ingeniería de Requerimientos a los Proyectos de desarrollo de Software aplicado a Empresas Clientes de GMD S.A.



7. RECOMENDACIONES

- ❖ Se recomienda una vez decidida la aplicación de esta metodología para la Ingeniería de Requerimientos MIRGMD, planificar un tiempo prudencial para implementar cada una de las etapas y tareas de la misma, el tiempo a estimarse depende del tamaño del proyecto pero generalmente la etapa de requerimientos debiera cubrir el 15% de tiempo total estimado para el proyecto.
- ❖ Es recomendable si se es posible aplicar las sesiones JAD ya sea en la elicitación o negociación de requerimientos debido a que este permite ahorrar tiempos, al evitar que los clientes se contrasten por separado como ocurre con las entrevistas individuales o observaciones y análisis social.
- ❖ Se recomienda realizar las matrices de rastreabilidad con una herramienta case de soporte para esta metodología como es el caso del recomendado OSRMT de código abierto, pero también se podría usar otras herramientas case como: Doors, RequisitePro de software propietario.
- ❖ Se recomienda realizar prototipos como elemento de validación de los requerimientos.
- ❖ Se recomienda que el entrevistador conozca otros factores como la política organizacional al realizar las entrevistas.
- ❖ Se recomienda aplicar la gestión de requerimientos durante todo el ciclo de vida del proyecto debido a que nos permite llevar un adecuado seguimiento de los cambios y mantenimiento de los requerimientos.



REFERENCIAS BIBLIOGRAFICAS

BAEZ Griselda, M. y Barba Bruner, S.

2001 “Metodología DoRCU para Ingeniería de Requerimientos”, IV Workshop en Requisitos, Buenos Aires, Argentina, 22 y 23 Noviembre del 2001.

Disponible en: wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER01/baez.pdf

BASS, L y otros

1998 functional Requirements Is a Dysfunctional Term. En Software Architecture in Practice, páginas 76–77. Addison–Wesley.

BOEHM, B y otros

1994 Software Requirements as Negotiated Win Conditions. En Proceedings of the First International Conference on Requirements Engineering. Disponible en

<http://sunset.usc.edu/>

BOOCH, G y otros

1999 El Lenguaje Unificado de Modelado: Ed. Addison Wesley Iberoamericana.

Borland

2004. 1158 Project Retrospectives Miroslav Novak Systems Engineer Borland.

Borland Conference 2004



CHEN, Peter. Dean y otros

2004 System Quality Requirements Engineering (SQUARE) Methodology: Case Study on Asset Management System. Carnegie Mellon University.

Disponible en :
<http://www.sei.cmu.edu/publications/documents/04.reports/04sr015.html>

COMP

2003. Requirements Engineering. COMP 354 | Software Engineering I. Course Notes. Fall 2003

COTA, A.

1994 "Ingeniería de Software". Soluciones Avanzadas. pp. 5-13. Presentación. Rational Software. Presentación disponible en <http://www-306.ibm.com/software/rational/>

DAVIS, Alan. M.

1993. Software Requirements: Objects, Functions and States. Prentice-Hall, 2a edición.

Department of Defense of the United States of America (DoD).

1994 Military Standard 498: Software Development and Documentation.

DURAN Toro, Amador

2004 "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", Tesis Doctoral en Informática, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. Disponible en:
http://fondosdigitales.us.es/thesis/thesis_view?oid=63



FIRESMITH, D. G.

1997 Uses Cases: the Pros and Cons. Se hace referencia en Rotem-Gal-Oz,
Arnon 2003 Methodology for developing Use Cases for large systems,

GAUSE, Donald C. y Gerald, M.

1989 Weinberg. Exploring Requirements: Quality Before Design. Dorset House.

Disponible en <http://www.testing.com/writings/reviews/gause-exploring.html>

Institute of Electrical and Electronics Engineers(IEEE).

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

Institute of Electrical and Electronics Engineers(IEEE).

1996. IEEE Guide for Developing System Requirements Specifications.

Institute of Electrical and Electronics Engineers(IEEE).

1997 IEEE Software Engineering Standards Collection. Institute of Electrical
and Electronics Engineers.

JACOBSON, Ivan.

1999. "Applying UML in The Unified Process" UML World Conference in
New York 11 March 99.

LEFFINGWELL, Don Widrig.

2003 Managing Software Requirements: A Use Case Approach. Ed.

Addison Wesley, 2nd Edition.



PRESSMAN, ROGER S.

2002. Ingeniería de Software. Un enfoque práctico. Quinta edición.
McGraw-Hill. Madrid. 2002.

ROSENBERG, Linda

(1998) “Methodology for Writing High Quality Requirement Specifications and for Evaluating Existing Ones”, Software Assurance Technology Center. NASA Goddard Space Flight Center Greenbelt, MD, Septiembre 24, 1998.

Disponible en: <http://aaaproduct.gsfc.nasa.gov/TEAS/lr-web/lindarose.html>.

Sakol

2001. Teeravarunyou, Sakol. Object-Mediated User Knowledge Elicitation Method. A Methodology in Understanding User Knowledge, Institute of Design - Illinois Institute of Technology, Chicago, USA Sato, Keiichi, Institute of Design - Illinois Institute of Technology, Chicago, USA. The proceeding of the 5th Asian International Design Research Conference, Seoul, Korea, October 2001

SAWYER, P. y otros

1997 Requirements, Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1). Disponible en:

<http://www.comp.lancs.ac.uk/computing/research/cseg/projects/reaims/publications.html>



SAWYER, P. y Montoya, G.

1999 SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999.

Disponible en <http://www.swebok.org>

SOMMERVILLE, Ian y Sawyer, P.

1997 Requirements. Engineering: A Good Practice Guide. Wiley.

The International Council on Systems Engineering (INCOSE)

2006 INCOSE Requirements Management Tools Survey. INCOSE Tools Database Working Group (TDWG).

<http://www.incose.org/ProductsPubs/products/toolsdatabase.aspx>

The Standish Group TSG

1995. *The CHAOS Report*.

http://www.standishgroup.com/sample_research/unfinished_voyages_1.php

TickIT

2002. Getting The Measure Of TickIT. Guidance and information about the emerging ISO measurement standards for improving software processes and how they relate to ISO 9001:2000. 2002.

WIERINGA, R. J.

1996 Requirements Engineering: Frameworks for Understanding. JohnWiley & Sons.



ANEXOS

ANEXO N° 1

Servicio de Soporte de Sistemas integrado (SSSI)

**Documento de
Requerimientos del Sistema**

Versión 1.1

Mayo 2009

Aplicado a Empresas Clientes de GMD S.A.

Realizado para “La Empresa Contratista”



1. Introducción

1.1 Propósito del Documento de Requerimientos.

Este documento tiene como propósito, proporcionar los requerimientos para el Sistema de Servicio de Soporte de Sistemas Integrado (SSSI), además, que pueda servir como un Contrato entre “el cliente” representado como “La Empresa Contratista” y “el desarrollador” representado como “La Empresa GMD S.A.” quien lleva a cargo la elaboración del Proyecto, esto es debido a que en este Documento de Requerimientos se encuentran definidas las características del Sistema que se va a desarrollar.

1.2 Alcance del Documento de Requerimientos.

El Documento de Requerimientos contiene principalmente una definición de los requerimientos funcionales y no funcionales para el sistema a desarrollar.

1.3 Definiciones, acrónimos y abreviaciones.

Acrónimo o término	Definición
Browser o navegador	Aplicación utilizada para navegar por el internet, y que despliega las pantallas desde el servidor en la pantalla del computador del usuario.
Pagina Web	Documento Web con información (texto, imágenes, video, audio, etc.) que se presenta en una misma pantalla. Una página Web esta en un servidor Web, y es traído al computador del usuario para visualizarla.
Servidor	Computador y programa computacional que brinda los servicios solicitados por otro computador llamado cliente

2. Descripción general

2.1 Descripción de la situación actual

Actualmente en el Área de Sistemas del Cliente (La Empresa Contratista), las solicitudes de atención de incidentes, problemas y requerimientos se gestionan y coordinan de manera manual, es decir cuando se necesita de la atención de un Analista



de Soporte de Sistemas este llena un Reporte de Intervención, ingresando los campos siguientes:

1. Numero de Reporte
2. Tipo de Atención (Incidente, Requerimiento, Problema)
3. Usuario afectado
4. Categoría asignada
5. Analista asignado
6. Grupo Asignado
7. Método de Reporte
8. Prioridad de la Atención
9. Descripción del problema
10. Hora de inicio de la atención.
11. Hora final de la atención.
12. Actividades Realizadas

Luego de realizar el llenado del Reporte de intervención de manera manual el coordinador del departamento de Sistemas registra los Reportes de Intervención en una “Hoja de Excel” y realiza el seguimiento de la atención registrada, comunicándose con el Usuario afectado, para solicitar conformidad.

Si el Usuario afectado no otorga la conformidad al Coordinador del Sistemas, este se comunica con Analista de Soporte de Sistemas para que revise nuevamente el problema del usuario.

Si el Usuario afectado otorga la conformidad al Coordinador del Sistemas, este realiza el envío vía e-mail , de la “Hoja de Excel” con el registro de los Reportes de Intervención al Jefe del departamento de Sistemas para su respectiva validación y cierre de la atención.



El Jefe del departamento de Sistemas, realiza la validación y cierre del registro de Reportes de Intervención y con estos datos genera los reportes de evaluación de incidentes, reportes de evaluación de requerimientos, reportes de evaluación de problemas, reportes de satisfacción del usuario y reportes de caídas masivas en la Empresa, realizando cuadros estadístico y todo esto mediante una “Hoja de Excel”.

Una vez que el Jefe del departamento de Sistemas tiene los reportes de evaluación elaborados y validados, realiza el envío vía e-mail, de la “Hoja de Excel” con los reportes de evaluación al Gerente del departamento de Sistemas para su conocimiento y gestión de toma de decisiones y mejora continua del Servicio de Soporte de Sistemas, para la empresa.

Finalmente decimos que este Reporte de Intervención, tiene que ser validado por:

- El Usuario Afectado
- El Analista de Soporte.
- El Jefe del Departamento de Sistemas.
- El Gerente del Departamento de Sistemas.

2.2 Propósito y alcance

El Servicio de Soporte de Sistemas integrado (SSSI) a desarrollar, trabajara en una ambiente web teniendo como funcionalidades los siguientes:

El Servicio de Soporte de Sistemas integrado (SSSI), permitirá que el personal del departamento de Sistemas, gestione y administre de manera eficaz y eficiente , los registros de reportes de intervención de incidentes, requerimientos y problemas, solicitados por los usuarios, en las distintas áreas y sedes del cliente (Empresa Contratista).



El Servicio de Soporte de Sistemas integrado (SSSI), permitirá a los usuarios, de acuerdo a los niveles de acceso y las reglas de seguridad establecidas, poder visualizar el estado de su atención que solicitaron, así como verificar cual es el Analista de Soporte de Sistemas encargado de atenderla.

Con el Servicio de Soporte de Sistemas integrado (SSSI), se crearan mecanismos de forma automatizada que nos permita llevar un control preciso de todas las solicitudes de atención que se reciben, con la finalidad de generar, en un determinado lapso de tiempo, mediciones que permitan conocer la razón de las solicitudes de atenciones y las soluciones que se aplicaron.

El Servicio de Soporte de Sistemas integrado (SSSI), administrara las opciones de Búsquedas en la cual tanto el personal técnico, administrativo y gerencial podrán obtener información de los Incidentes, requerimientos, problemas, contactos, grupos, servicios, métodos de reporte y categorías , de acuerdo a las reglas de seguridad establecidas.

El Servicio de Soporte de Sistemas integrado (SSSI), administrara la elaboración de reportes evaluación, en los cuales la Gerencia del departamento de Sistemas, podrá realizar la gestión de la toma de decisiones y de mejora continua del Servicio de Soporte de Sistemas, para la empresa.

El Servicio de Soporte de Sistemas integrado (SSSI), también permitirá establecer una mejor comunicación con los proveedores del cliente. Aquí se registrara las solicitudes de los usuarios en los cuales se necesiten los servicios de los proveedores y poder administrar y gestionar dichas solicitudes.

2.3 Definición del problema

A continuación se describe el problema expresado en los siguientes puntos.

- Tiempo perdido entre las llamadas telefónicas realizadas por el usuario para reportar incidentes en los Sistemas de la Empresa.



- Tiempo perdido entre las llamadas telefónicas realizadas por el usuario para solicitar requerimientos en los Sistemas de la Empresa.
- Saturación de la línea telefónica de la Empresa ya que al reportarse un incidente masivo el número de las llamadas al departamento de Sistemas, se elevan considerablemente.
- Tiempo de atenciones de incidentes al usuario demasiados extensos.
- Tiempo de atenciones de requerimientos al usuario demasiados extensos.
- Aumentos de Costos en el Equipo de Soporte de Sistemas.
- Ineficiencia y pobre desempeño del Servicio de Soporte Sistemas brindado al usuario.
- Falta de integración en los Servicios de Soporte de Sistemas en toda la Organización.
- Falta de conocimiento del desempeño en tiempo real del Personal de Soporte de Sistemas

2.4 Participantes en el proyecto

El Grupo Participante de la Empresa GMD S.A. , encargado de liderar el desarrollo del proyecto, son los siguientes:

- Ing. Lourdes Cavero: Gerente del Proyecto
- Ing. Pablo Arbulu: Jefe del proyecto.
- Bachiller. Oscar Sonco: Analista.
- Bachiller. Diego Enriquez: Programador.

Los participantes de la empresa cliente, “La Empresa Contratista” son :

- Ing. Jorge Carassa, representante del departamento de Sistemas del cliente, ante nosotros.
- Ing. José Vélez, coordinador de servicios del departamento de Sistemas del Cliente.



- Ing. Rafael Arango, Jefe del departamento de Sistemas del Cliente.
- Ing. Juan Linares, Gerente del departamento de Sistemas del Cliente.
- Ing. José Vásquez, coordinador de servicios con proveedores del Cliente.

2.5 Restricciones Generales

El proyecto deberá ser desarrollado en un tiempo máximo de 16 semanas.

3. Requerimientos Específicos

Esta sección se divide en las siguientes sub-secciones en las que se describen los requerimientos del sistema.

3.1 Objetivos del sistema

Esta sección contiene una lista con los objetivos que se esperan alcanzar cuando el sistema de software a desarrollar esté en explotación, especificados mediante la plantilla para objetivos.

Plantilla 1: OBJ -01 Gestión de Seguridad

OBJ-01	Gestión de Seguridad
Descripción	El Sistema deberá permitir gestionar los accesos a los usuarios y sus respectivos roles
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 2: OBJ -02 Gestión de Incidentes

OBJ-02	Gestión de Incidentes
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los incidentes reportados.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 3: OBJ -03 Gestión de Problemas

OBJ-03	Gestión de Problemas
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los Problemas reportados.
Estabilidad	Alta
Comentarios	Ninguno



Plantilla 4: OBJ -04 Gestión de Requerimientos

OBJ-04	Gestión de Requerimientos
Descripción	El Sistema debe permitir la creación, registro, actualización y reporte de los Requerimientos reportados.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 5: OBJ -05 Gestión de Búsquedas

OBJ-05	Gestión de Búsquedas
Descripción	El sistema deberá permitir que se realice las búsquedas de acuerdo al nombre del usuario, el numero de incidente / problema / requerimiento, el nombre del Analista de Soporte de Sistemas u otra característica valida.
Estabilidad	Alta
Comentarios	Ninguno

Plantilla 6: OBJ -06 Generación de Reportes de Evaluación

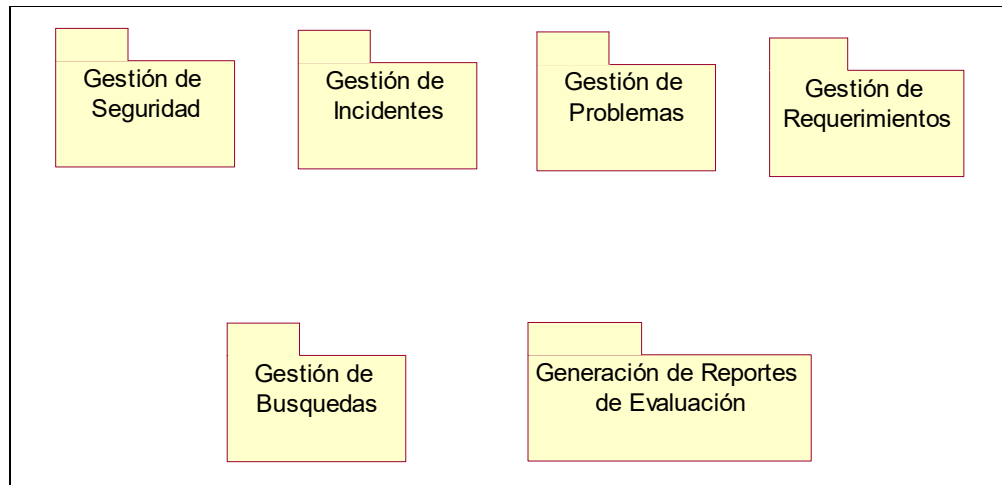
OBJ-06	Generación de Reportes de Evaluación
Descripción	El sistema deberá generar Reportes de Evaluación Estadísticos de los incidentes por periodos, requerimientos por periodos, desempeño del Personal, historiales de los usuarios, incidentes masivos y satisfacción del usuario en el Servicio.
Estabilidad	Alta
Comentarios	Ninguno

3.2 Requerimientos funcionales

Esta parte de documento se describe la lista de requerimientos funcionales identificados, dividiéndose en los siguientes apartados que se describen a continuación. Los requerimientos funcionales están clasificados de acuerdo a los subsistemas de la Figura 1.



Fig.1. Diagrama de subsistemas



3.2.1 Definición de actores

Esta sección contiene una lista con los actores que se hayan identificado, especificados mediante la plantilla para actores de casos de uso.

Plantilla 7: ACT-01 Administrador del sistema

ACT-01	Administrador del Sistema
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Administrador del Sistema del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno

Plantilla 8: ACT-02 Gerente del departamento de Sistemas

ACT-02	Gerente del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Gerente del departamento de Sistemas del Cliente denominado "La Empresa Contratista"
Comentarios	Ninguno



Plantilla 9: ACT-03 Jefe del departamento de Sistemas

ACT-03	Jefe del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Jefe del departamento de Sistemas del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Plantilla 10: ACT-04 Coordinador del departamento de Sistemas

ACT-04	Coordinador del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Coordinador del departamento de Sistemas del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Plantilla 11: ACT-05 Analista del departamento de Sistemas

ACT-05	Analista del departamento de Sistemas
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Analista del departamento de Sistemas del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Plantilla 12: ACT-06 Usuario Final

ACT-06	Usuario Final
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Usuario Final del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno

Plantilla 13: ACT-07 Gestor de Terceros

ACT-07	Gestor de Terceros
Versión	1.0 (01 de Abril del 2010)
Autores	Oscar Sonco (GMD S.A.)
Fuentes	-
Descripción	Este actor representa al Gestor de Terceros del Cliente denominado “La Empresa Contratista”
Comentarios	Ninguno



3.2.2 Casos de uso del sistema especificados en plantillas

Este apartado contiene los casos de uso que se han identificado, especificados mediante la plantilla para requisitos funcionales.

Casos de uso del subsistema Gestión de Seguridad

RF-01	Validación de Usuarios	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien ingresa al sistema.	
Precondición	Usuario ingresa a pantalla de bienvenida del sistema	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa su usuario y clave
	2	El sistema verifica si se encuentra registrado
	3	El sistema verifica privilegio asignado
	4	Sistema muestra menú de acuerdo al su privilegio
Postcondición	Usuario logueado	
Excepciones	Paso	Acción
	2	Si no se encuentra registrado, el sistema muestra pantalla de logueo
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	20 veces / día	
Importancia	importante	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Casos de uso del subsistema Gestión de Incidentes

RF-06	Ingresar Incidente	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Fuentes	<ul style="list-style-type: none"> • Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> • - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Analista de Sistemas realiza el ingreso de un nuevo incidente	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El Analista de Sistemas selecciona opción Archivo – Nuevo Incidente
	3	El Analista ingresa el usuario afectado
	4	El Analista selecciona la categoría
	5	El Analista ingresa el Soporte asignado
	6	El Analista selecciona el método de reporte
	7	El Analista selecciona la Prioridad
	8	El Analista ingresa la descripción del caso
Postcondición	Nuevo Incidente registrado en el Sistema	
Excepciones	Paso	Acción
	3	Si no se ingresa los campos requeridos el sistema muestra mensaje de falta llenar los campos.
	5	Si se ingresa los campos requeridos de manera incorrecta el sistema muestra mensaje de error
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	40 veces / día	
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

RF-22	Solicitud de Conformidad	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> • Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> • Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> • - 	



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Coordinador de Sistemas realiza la solicitud de conformidad	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El coordinador realiza el cambio de estado de Pendiente a Resuelto
	3	El coordinador selecciona el grupo solucionador
	4	El coordinador ingresa las observaciones del usuario dentro de los comentarios.
	5	El coordinador da el visto bueno y realiza el cambio de estado de Resuelto a Cerrado
	6	El Sistema muestra el mensaje de Incidente cerrado.
Postcondición	Cierre de Incidente validado por el Coordinador de Sistemas.	
Excepciones	Paso	Acción
	4	Si el usuario solicita que se revise otra vez el Incidente, el Coordinador se contacta con el Analista responsable.
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	40 veces / día	
Importancia	importante	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Casos de uso del subsistema Gestión de Problemas

RF-08	Ingresar Problema	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Analista de Sistemas realiza el ingreso de un nuevo Problema	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El Analista de Sistemas selecciona opción Archivo – Nuevo Problema
	3	El Analista ingresa el usuario afectado
	4	El Analista selecciona la categoría
	5	El Analista ingresa el Soporte asignado



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

	6	El Analista selecciona el método de reporte
	7	El Analista selecciona la Prioridad
	8	El Analista ingresa la descripción del caso
Postcondición	Nuevo Problema registrado en el Sistema	
Excepciones	Paso	Acción
	3	Si no se ingresa los campos requeridos el sistema muestra mensaje de falta llenar los campos.
	5	Si se ingresa los campos requeridos de manera incorrecta el sistema muestra mensaje de error
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	5 veces / día	
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

RF-38	Solicitud de Conformidad	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Coordinador de Sistemas realiza la solicitud de conformidad	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El coordinador realiza el cambio de estado de Pendiente a Resuelto
	3	El coordinador selecciona el grupo solucionador
	4	El coordinar ingresa las observaciones del usuario dentro de los comentarios.
	5	El coordinador da el visto bueno y realiza el cambio de estado de Resuelto a Cerrado
	6	El Sistema muestra el mensaje de Problema cerrado.
Postcondición	Cierre de Problema validado por el Coordinador de Sistemas.	
Excepciones	Paso	Acción
	4	Si el usuario solicita que se revise otra vez el Problema, el Coordinador se contacta con el Analista responsable.
Rendimiento	Paso	Cuota de tiempo
Frecuencia	5 veces / día	



esperada	
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

Casos de uso del subsistema Gestión de Requerimientos

RF-07	Ingresar Requerimiento	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Analista de Sistemas realiza el ingreso de un nuevo Requerimiento	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El Analista de Sistemas selecciona opción Archivo – Nuevo Requerimiento
	3	El Analista ingresa el usuario afectado
	4	El Analista selecciona la categoría
	5	El Analista ingresa el Soporte asignado
	6	El Analista selecciona el método de reporte
	7	El Analista selecciona la Prioridad
	8	El Analista ingresa la descripción del caso
Postcondición	Nuevo Requerimiento registrado en el Sistema	
Excepciones	Paso	Acción
	3	Si no se ingresa los campos requeridos el sistema muestra mensaje de falta llenar los campos.
	5	Si se ingresa los campos requeridos de manera incorrecta el sistema muestra mensaje de error
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	25 veces / día	
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	en construcción	



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

Estabilidad	Alta
Comentarios	Ninguno

RF-38	Solicitud de Conformidad	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Coordinador de Sistemas realiza la solicitud de conformidad	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El sistema muestra el menú de opciones
	2	El coordinador realiza el cambio de estado de Pendiente a Resuelto
	3	El coordinador selecciona el grupo solucionador
	4	El coordinar ingresa las observaciones del usuario dentro de los comentarios.
	5	El coordinador da el visto bueno y realiza el cambio de estado de Resuelto a Cerrado
	6	El Sistema muestra el mensaje de Requerimiento cerrado.
Postcondición	Cierre de Requerimiento validado por el Coordinador de Sistemas.	
Excepciones	Paso	Acción
	4	Si el usuario solicita que se revise otra vez el Requerimiento, el Coordinador se contacta con el Analista responsable.
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	25 veces / día	
Importancia	importante	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Casos de uso del subsistema Gestión de Búsquedas

RF- 14	Busqueda de Incidentes	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos	<ul style="list-style-type: none"> - 	



TESINA: Metodología para la Ingeniería de Requerimientos para Proyectos de desarrollo de software aplicado a Empresas Clientes de GMD S.A.

asociados		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Analista de Sistemas realiza una búsqueda de incidentes	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El Sistema muestra el menú de opciones
	2	El Analista de Sistemas selecciona la opción Buscar – Incidente
	3	El Analista de Sistemas ingresa el nombre del Usuario afectado.
	4	El Analista de Sistemas ingresa el nombre del Analista asignado.
	5	El Analista de Sistemas ingresa el numero de ticket del Incidente
	6	El Analista de Sistemas activa la búsqueda de incidentes solicitada
Postcondición	Búsquedas de incidentes elaborada.	
Excepciones	Paso	Acción
	3	Si no se ingresa los campos requeridos el sistema muestra mensaje de falta llenar los campos.
	6	Si se ingresa los campos requeridos de manera incorrecta el sistema muestra mensaje de error.
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	10 veces / día	
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	Ninguno	

Casos de uso del subsistema Generación de Reportes de Evaluación

RF- 47	Elaboración de Reportes de Evaluación	
Versión	1.0 (Abril 2009)	
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.) 	
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista) 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-2 	
Requisitos asociados	<ul style="list-style-type: none"> - 	
Descripción	El Sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Jefe de Sistemas, elabore los Reportes de Evaluación.	
Precondición	Usuario logueado	
Secuencia Normal	Paso	Acción
	1	El Sistema muestra el menú de opciones
	2	El Jefe de Sistemas selecciona los incidentes, problemas, requerimientos u otras solicitudes. El Jefe de Sistemas selecciona la opción Reportes – Resumen
	3	El Sistema muestra el reporte de los incidentes, problemas, requerimientos u otras solicitudes seleccionadas
	4	El Sistema brinda la opción de imprimir los Reportes de Evaluación que se visualizan.



Postcondición	Reportes de Evaluación generados	
Excepciones	Paso	Acción
	3	Si no se selecciona ninguna solicitud para los Reportes de Evaluación, el sistema mostrara el mensaje de seleccionar una solicitud como mínimo.
Rendimiento	Paso	Cuota de tiempo
Frecuencia esperada	5 veces / día	
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	Ninguno	

3.2.3 Diagramas de casos de uso

Esta sección contiene los diagramas de casos de uso del sistema que se va realizar.

Fig. 2. Diagrama de casos de uso del subsistema Gestión de Seguridad

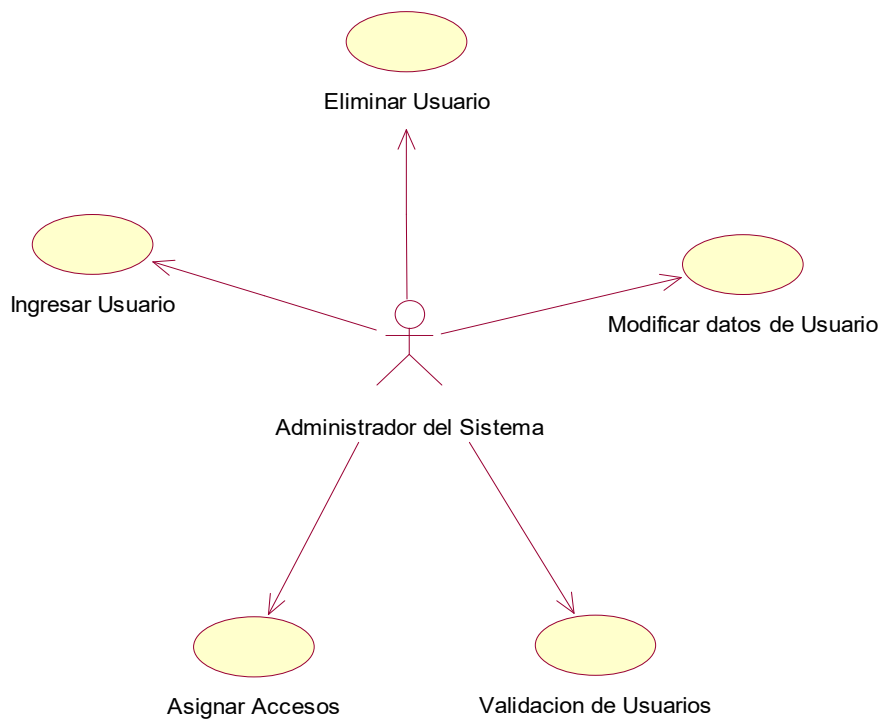


Fig. 3 Diagrama de casos de uso del subsistema Gestión de Incidentes

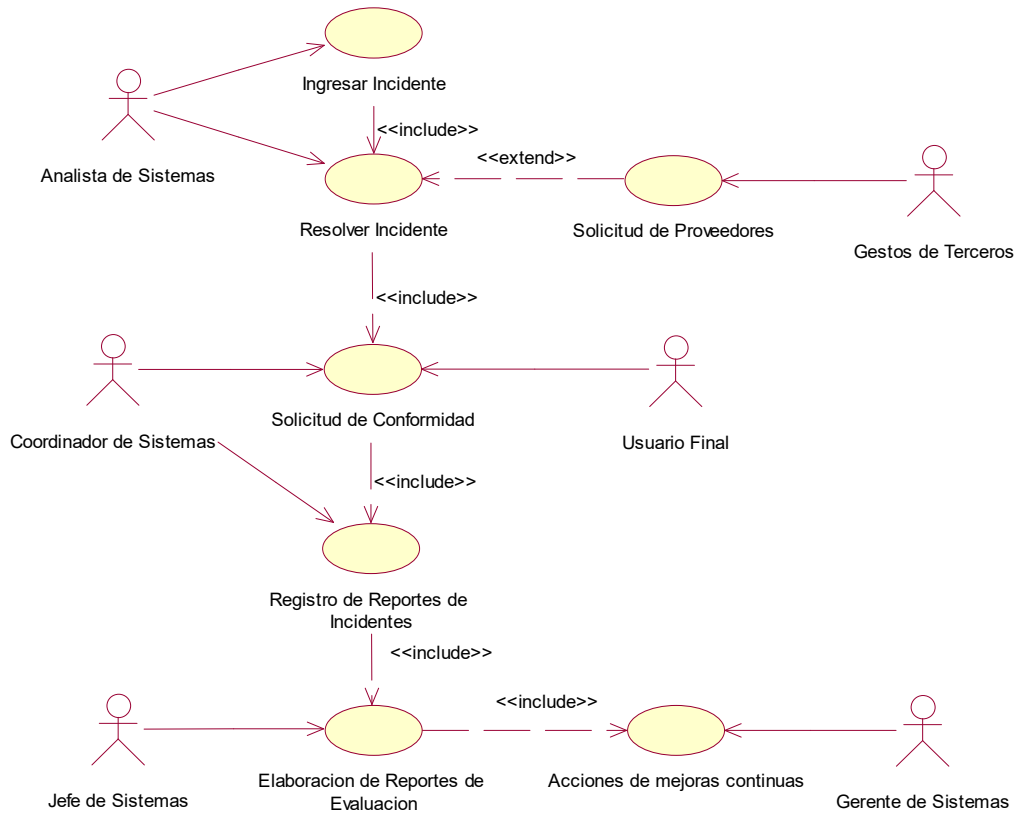


Fig. 4. Diagrama de casos de uso del Subsistema Gestión de Problemas

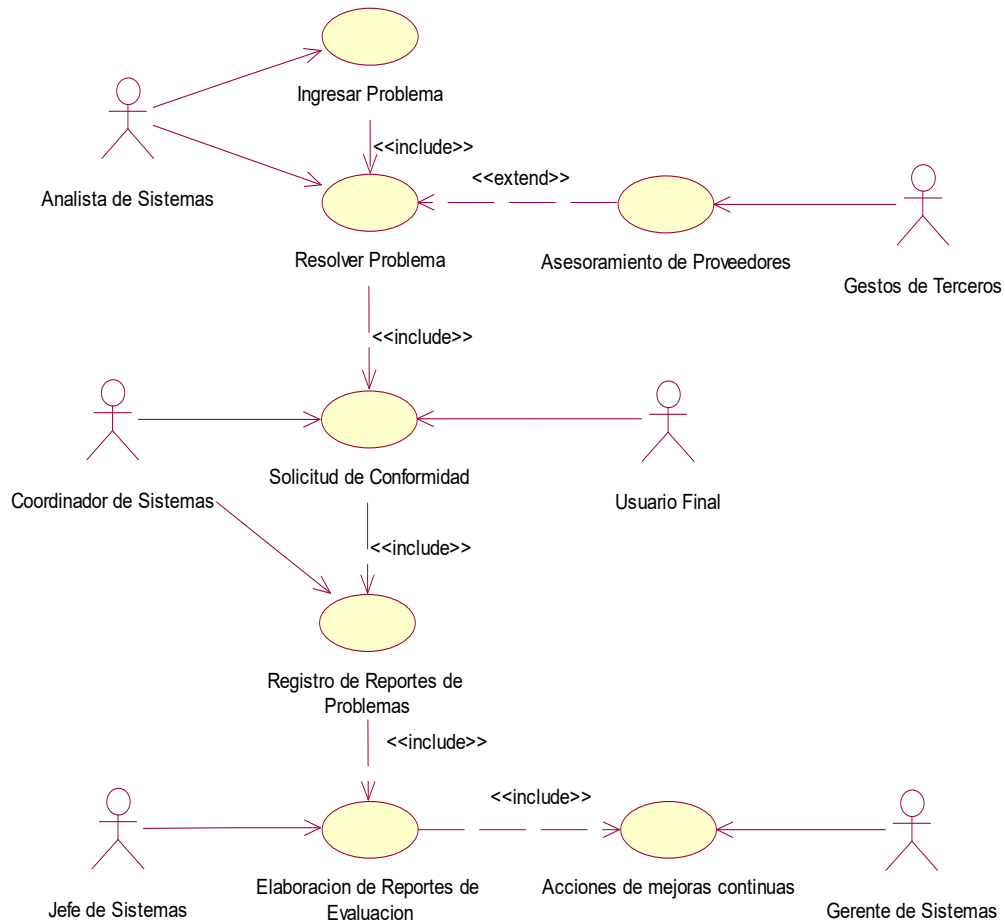


Fig. 5. Diagrama de casos de uso del subsistema Gestión de Requerimientos

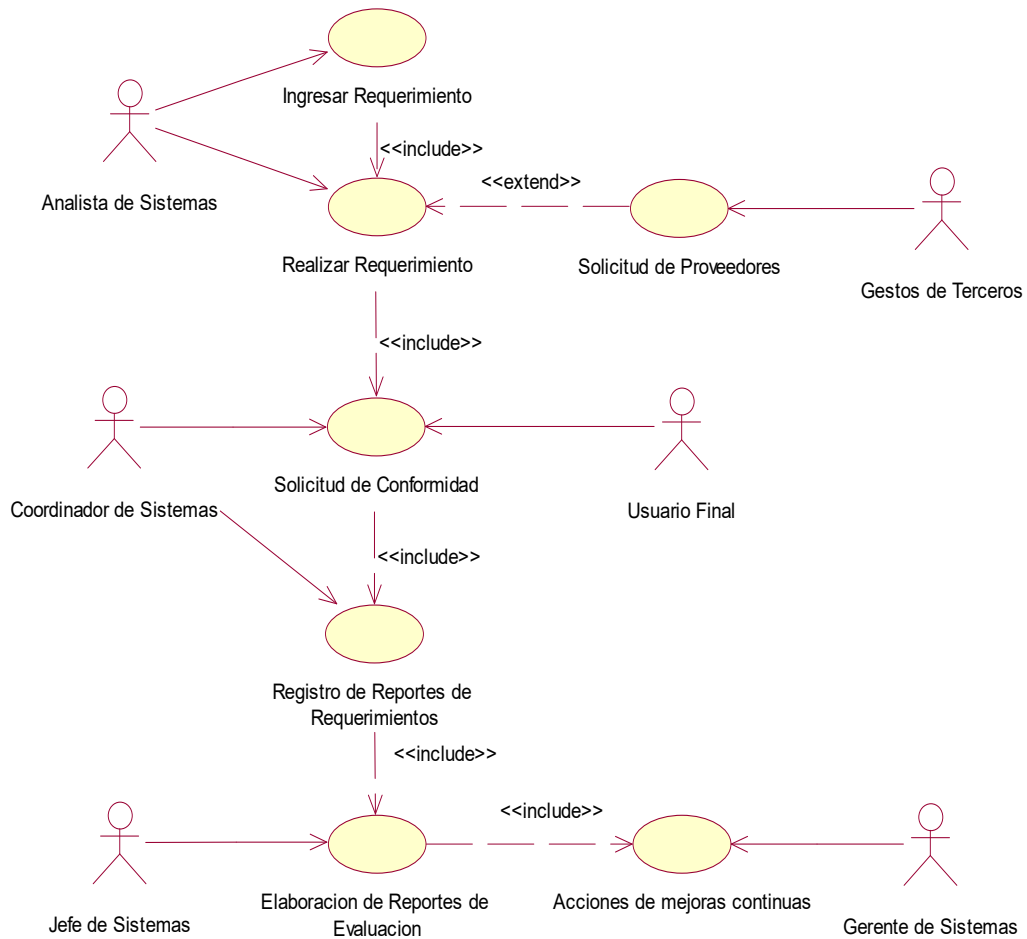


Fig. 6. Diagrama de casos de uso del subsistema Gestión de Búsquedas

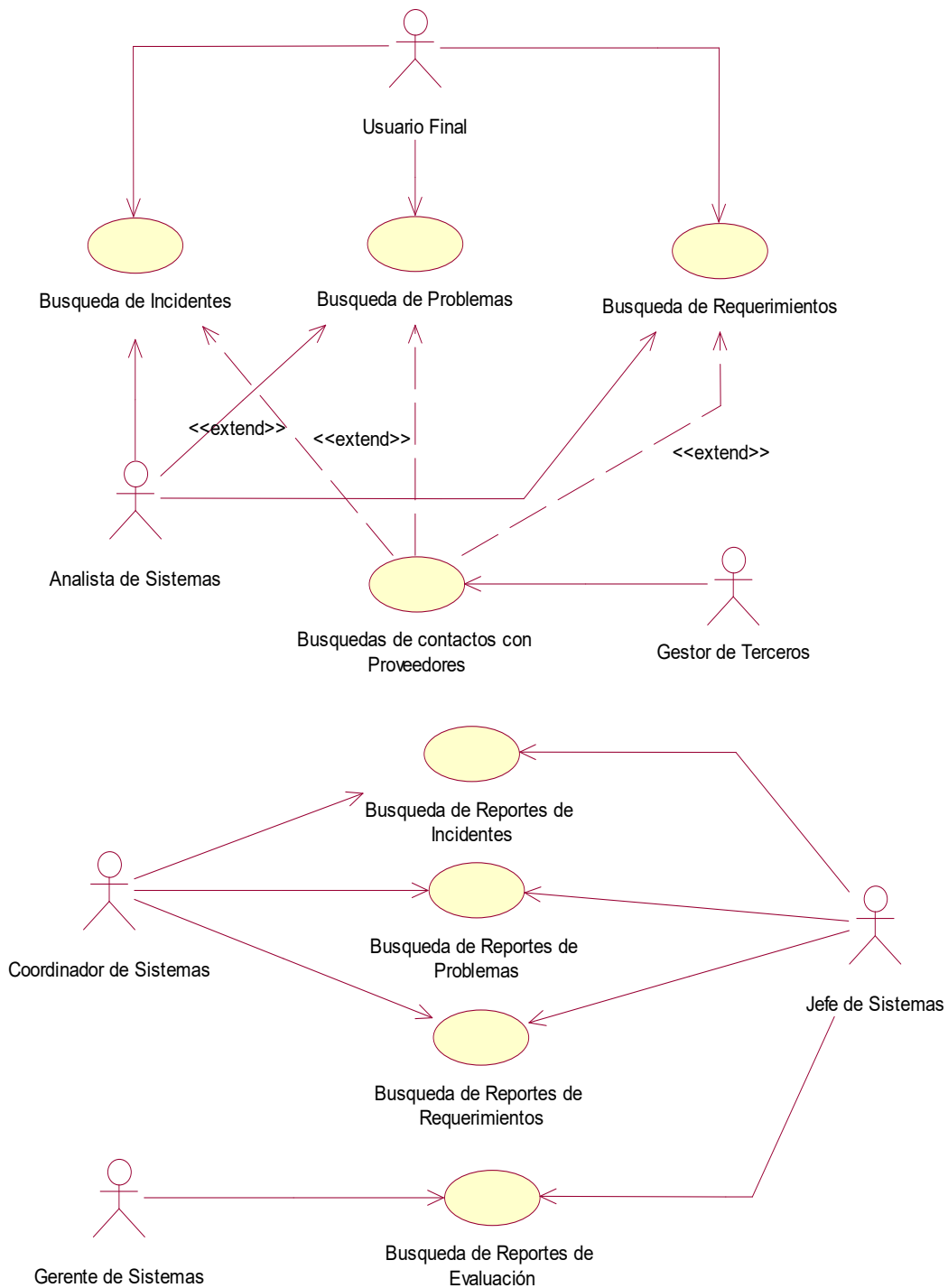
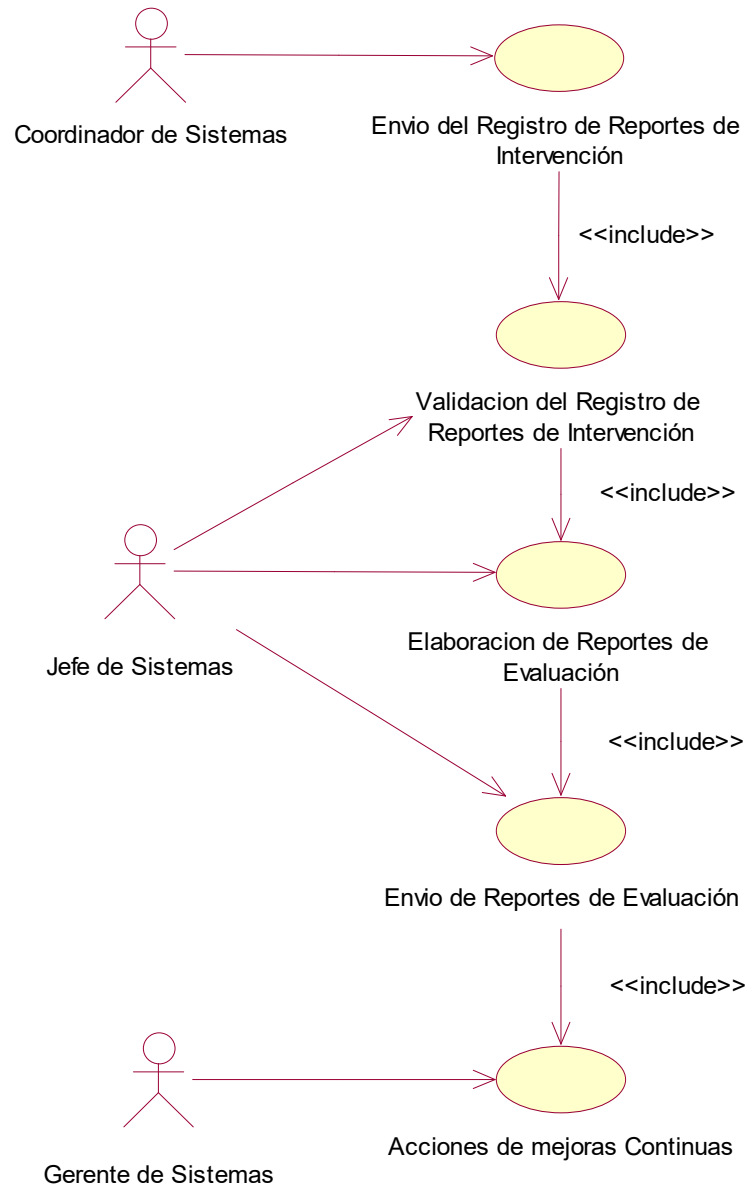


Fig. 7. Diagrama de casos de uso del subsistema Generación de Reportes de Evaluación



3.3 Requerimientos No funcionales.

Esta sub-sección contiene la lista los requerimientos no funcionales del sistema que se hayan identificado, especificados mediante la plantilla para requerimientos no funcionales.



3.3.1 Requerimientos No Funcionales de Portabilidad

RNF-01	Portabilidad
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	El sistema deberá funcionar en un ambiente Web sobre plataforma Linux, usando servidor Web Apache, también el sistema debe visualizarse sin ningún problema con los principales browsers
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno

3.3.2 Requerimientos No Funcionales de Copia de Seguridad

RNF-02	Copias de Seguridad
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno

3.3.3 Requerimientos No Funcionales de Disponibilidad

RNF-03	Disponibilidad
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	La disponibilidad debe ser continua con un nivel de servicio para los usuarios de 7 días x 24 horas
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno



3.3.4 Requerimientos No Funcionales de Seguridad

RNF-04	Seguridad
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	El sistema deberá reflejar patrones de seguridad de acuerdo a las especificaciones funcionales dadas
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno

3.3.5 Requerimientos No Funcionales de Mantenibilidad

RNF-05	Mantenibilidad
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	Se de estructurar el código de manera consistente y predecible
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno

3.3.6 Requerimientos No Funcionales de Tamaño

RNF-05	Tamaño
Versión	1.0 (Abril 2009)
Autores	-
Fuentes	-
Objetivos asociados	-
Requisitos asociados	-
Descripción	El sistema requiere de 128 MB de RAM como mínimo
Importancia	Vital
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	ninguno



3.4 Matriz de rastreabilidad Objetivos - Requerimientos

Esta sección contiene una la matriz objetivo–requerimiento, de forma que para cada objetivo se pueda conocer con qué requerimientos está asociado.

	OBJ-01	OBJ-02	OBJ-03	OBJ-04	OBJ-05	OBJ-06
RF-01	⊕					
RF-02	⊕					
RF-03	⊕					
RF-04	⊕					
RF-05	⊕					
RF-06		⊕				
RF-07				⊕		
RF-08			⊕			
RF-09		⊕	⊕	⊕		
RF-10		⊕	⊕	⊕		
RF-11		⊕	⊕	⊕		
RF-12		⊕	⊕	⊕		
RF-13		⊕	⊕	⊕		
RF-14					⊕	
RF-15					⊕	
RF-16					⊕	
RF-17					⊕	
RF-18					⊕	
RF-19					⊕	
RF-20					⊕	
RF-21					⊕	
RF-22		⊕				
RF-23				⊕		
RF-24			⊕			
RF-25		⊕	⊕	⊕		
RF-26		⊕	⊕	⊕		
RF-27		⊕	⊕	⊕		
RF-28		⊕	⊕	⊕		
RF-29		⊕	⊕	⊕		
RF-30		⊕				
RF-31				⊕		
RF-32			⊕			
RF-33		⊕	⊕	⊕		
RF-34		⊕	⊕	⊕		
RF-35		⊕	⊕	⊕		
RF-36		⊕	⊕	⊕		
RF-37		⊕	⊕	⊕		
RF-38		⊕	⊕	⊕		⊕



RF-39		⊕	⊕	⊕		⊕
RF-40		⊕	⊕	⊕		
RF-41						⊕
RF-42						⊕
RF-43						⊕
RF-44						⊕
RF-45						⊕
RF-46						⊕
RF-47						⊕
RF-48						⊕
RF-49						⊕
RF-50						⊕
RF-51						⊕
RF-52						⊕
RF-53						⊕
RF-54						⊕
RF-55						⊕

3.5 Conflictos en los Requerimientos.

Esta sección contiene los conflictos identificados y solucionados, especificados mediante la plantilla para conflictos.

Plantilla de Conflictos CFL-01 del Requisito RF – 05

CFL-01	Necesidad de especificar que Personal del Área de Soporte de Sistemas podrá visualizar los Reportes de Evaluación de Incidentes
Versión	1.1 (Mayo 2009)
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.)
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista)
Objs./Reqs. en conflicto	<ul style="list-style-type: none"> OBJ-01 Gestión de Seguridad RF-05 Asignar Accesos
Descripción	No se describe en ningún requisito cual de los usuarios van a tener el acceso para visualizar los Reportes de Evaluación de Incidentes.
Alternativas	El Gerente del departamento de Sistemas, el Jefe del departamento de Sistemas y el Administrador de Sistemas, tendrán acceso a los Reportes de Evaluación de Incidentes, y el resto de usuarios no tendrán el acceso.
Solución	Se especifico la alternativa propuesta.
Importancia	Alta
Urgencia	Inmediatamente
Estado	Validado
Comentarios	Ninguno



Plantilla de Conflictos CFL-02 del Requisito RF – 05

CFL-02	Necesidad de administrar el ingreso de nuevos contactos, servicios y categorías en el Sistema
Versión	1.1 (Mayo 2009)
Autores	<ul style="list-style-type: none"> Oscar Sonco Tena (GMD S.A.)
Fuentes	<ul style="list-style-type: none"> Jorge Carassa (Empresa Contratista)
Objs./Reqs. en conflicto	<ul style="list-style-type: none"> OBJ-01 Gestión de Seguridad RF-05 Asignar Accesos
Descripción	No se describe en ningún requisito cual de los usuarios va a administrar el ingreso de contactos, servicios y categorías nuevas al Sistema
Alternativas	El Coordinador del departamento de Sistemas y el Administrador de Sistemas, tendrán acceso al ingreso de nuevos contactos, servicios y categorías. El resto de usuarios no tendrán el acceso.
Solución	Se especifico la alternativa propuesta.
Importancia	Alta
Urgencia	Inmediatamente
Estado	Validado
Comentarios	Ninguno

3.6 Las versiones generadas del Documento de Requerimientos.

Numero	Fecha	Descripción	Autores
0	Abril, 2009	Versión 1.0	Oscar Sonco, Diego Enriquez
1	Mayo, 2009	Versión 1.1	Oscar Sonco, Diego Enriquez



ANEXO N° 2

INDICE DE CUADROS

Cuadro 5.1 Matriz Interacción de requerimientos.....	98
Cuadro 5.2 Lista de Versiones del documento de requerimientos.....	100

INDICE DE PLANTILLAS

Plantilla 1: OBJ -01 Gestión de Seguridad.....	174
Plantilla 2: OBJ -02 Gestión de Incidentes.....	174
Plantilla 3: OBJ -03 Gestión de problemas.....	174
Plantilla 4: OBJ -04 Gestión de requerimientos.....	175
Plantilla 5: OBJ -05 Gestión de búsquedas.....	175
Plantilla 6: OBJ -06 Generación de reportes de evaluación.....	175
Plantilla 7: ACT-01 Administrador del sistema.....	176
Plantilla 8: ACT-02 Gerente de sistemas.....	176
Plantilla 9: ACT-03 Jefe de sistemas.....	177
Plantilla 10: ACT-04 Coordinador de sistemas.....	177
Plantilla 11: ACT-05 Analista de Sistemas.....	177
Plantilla 12: ACT-06 Usuario Final.....	177
Plantilla 13: ACT-07 Gestor de Terceros.....	177
Plantilla 14: RF-06 Nuevo Incidente.....	96
Plantilla 15: RNF-01 Portabilidad.....	97
Plantilla 16: Plantilla de Conflictos CFL-01 del Requisito RF – 05.....	98
Plantilla 17: Plantilla de Conflictos CFL-01 del Requisito RF – 05.....	99

