



**FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS
INGENIERÍA EN SISTEMAS DE COMPUTACIÓN E INFORMÁTICA**

**Cómo Mejorar las Metodologías Actuales de Requerimientos Basados en los
Estándares IEEE 830, 1233 y 1362.**

**Trabajo de Titulación presentado en conformidad a los requisitos para optar por
el título de
Ingeniero en Sistemas de Computación e Informática**

Profesor Guía

Ing. Fernando Quevedo Espín

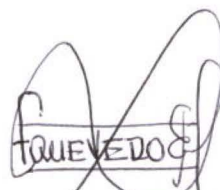
Autor

Grace Patricia Ayala Cifuentes

2009

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

A handwritten signature in black ink, appearing to read 'F. QUEVEDO E.', is written over a rectangular stamp that contains the name 'QUEVEDO' in capital letters.

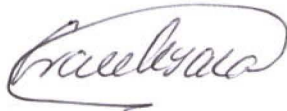
Fernando Quevedo Espín

Ingeniero

050168794-1

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.



Grace Patricia Ayala Cifuentes

171324942-1

AGRADECIMIENTO

Gracias a Dios que es el creador de todo y que creó este sueño que hoy termina en esta meta. La vida es un premio múltiple y hoy tengo el privilegio de un triunfo más, el que he logrado con constancia y esfuerzo, apoyándome en el amor de Dios, de mi familia, de mis compañeros, maestros y amigos. No hubiese sido posible sin el apoyo de todos los que amo y admiro; cada uno estuvo ahí en el momento preciso brindándome todo cuanto necesite en el camino el que hoy con gran orgullo culmino.

Un agradecimiento especial para mi Director de Tesis y amigo el Ing. Fernando Quevedo Espín por su perseverancia en la dirección y sus sabios consejos.

Grace

DEDICATORIA

Este trabajo es el esfuerzo conjunto de muchas personas. Por tanto, a todas ellas por su incondicional soporte, les dedico este gran logro de mi vida. A mis padres, que con su esfuerzo, cariño, dedicación y fortaleza, me han permitido convertirme en la persona que soy. A mis hijos Stephania y David que con madurez a pesar de sus cortos años entendieron mi tarea.

Grace

RESUMEN

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. Esta disciplina establece el proceso de definición de requerimientos como una sucesión de actividades mediante la cual lo que debe hacerse se "elicit", se modela y analiza. En este proceso se deben conciliar diferentes puntos de vista y utilizar una combinación de métodos, personas y herramientas. El resultado final constituye la documentación de los requerimientos. Éstos deben expresarse de forma clara y estructurada de manera que puedan ser entendidos tanto por expertos como por el usuario, quien deberá participar en la validación.

El desarrollo de software es un proceso exigente y la falta de una metodología estándar y práctica lo ha convertido en un trabajo individual y casi artístico donde cada programador tiene su propia forma de hacer las cosas.

El análisis de requerimientos genera el documento de especificación de requerimientos del software: que es la descripción completa del comportamiento y la lista de funcionalidades del programa a desarrollar.

El documento de especificación de requerimientos también debe incluir los requerimientos no funcionales que son aquellos que limitan tanto el diseño como la implementación, entre ellos tenemos la confiabilidad, la prioridad y el costo.

Es muy importante mencionar que el poder formular una especificación de requerimientos completa y consistente, es un paso muy importante para evitar cometer errores en la definición de los requerimientos, ya que los mismos pueden resultar muy caros de corregir una vez desarrollado el sistema. De ahí,

la vital importancia que tiene la ingeniería de requerimientos en generar una adecuada especificación que contemple claramente y sin ambigüedades los requerimientos del sistema a desarrollar, con el fin primordial de evitar que los proyectos fracasen debido a una mala elaboración de la definición y especificación de requerimientos.

El proceso de la Ingeniería de Requerimientos sirve para recopilar la información necesaria para establecer la funcionalidad que se quiere alcanzar con el sistema. Para ello, se debe de contar con buenos métodos y técnicas para hacerlo, además de una comunicación fluida y constante con el cliente, ya que los requerimientos deben reflejar las necesidades reales que el cliente quiere satisfacer. Las revisiones deben involucrar al cliente y al staff de contratistas para validar los requerimientos del sistema.

SUMMARY

The Engineering of Requirements fulfills a fundamental paper in process of production of software, since it focuses a fundamental area: The definition of which it wishes to produce. Its main task consists of generation of correct specifications that describe with clarity, without ambiguities, in solid and compact form, the behavior of the system. This discipline establishes the process of definition of requirements like a succession of activities by means of what it has to do "elicit", model and analyses. In this process have to conciliate different points of view and use a combination of methods, people and tools. The final result constitutes the documentation requirements. These have to express in clear and structured form so that they can be understood as much as by experts and user, the one who will have to take part in validation.

The development of software is a demanding process and the fault of a standard and practical methodology has converted it in an individual work and almost artistic where each programmer has its proper form to do the things. The analysis of requirements generates document of specification of software requirements: It is the complete description of behavior and list of functionalities of the program to develop.

The document specification of requirements also has to include non functional requirements which are those that limit so much the design like the implementation, between them have the reliability, scale and cost.

Is very important to mention that the power to formulate a specification of complete and solid requirements; is a very important step to avoid make errors in the definition of the requirements, since the same can result very expensive to correct once the system will be developed. Therefore, the vital importance of engineering requirements is to generate an adequate specification providing clearly and without ambiguities for the system requirements to develop, the primary objective of preventing the projects fail is due to wrong definition and specification requirements.

The process of engineering requirements is used to gather the information needed to implement the functionality to be achieved with the system. For that reason, it must have good methods and techniques to do it, plus a constant fluid communication with the customer, since requirements should reflect real needs that the client wants fulfill. Reviews should involve the client and contractor staff to validate the system requirements.

ÍNDICE

| CONTENIDO | Pág. |
|--|-------------|
| PORTADA | I |
| DECLARACIÓN DEL PROFESOR GUÍA | II |
| DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE | III |
| AGRADECIMIENTOS | IV |
| DEDICATORIA | V |
| RESUMEN | VI |
| SUMMARY | VIII |
| ÍNDICE DE CONTENIDOS | X |
| ÍNDICE DE FIGURAS | XIV |
| ÍNDICE DE TABLAS | XVI |
| INTRODUCCIÓN | 1 |
| | |
| CAPÍTULO I | |
| GENERALIDADES | 3 |
| 1.1. MOTIVACIÓN | 3 |
| 1.1.1. Necesidad de ingeniería de requerimientos | 4 |
| 1.2. INGENIERÍA DE REQUERIMIENTOS | 5 |
| 1.3. EL CONCEPTO DE REQUERIMIENTOS | 6 |
| 1.3.1. Las dimensiones de los requerimientos | 7 |
| 1.3.2. Los requerimientos como restricciones | 9 |
| 1.3.3. Propiedades deseables de los requerimientos | 10 |
| 1.4. EL ANÁLISIS DE REQUERIMIENTOS COMO DISCIPLINA DE INGENIERÍA | 11 |
| 1.5. MODELOS DE PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS | 12 |
| 1.6. SITUACIÓN DEL SOFTWARE EN EL ECUADOR | 15 |
| 1.6.1. Problemas en Requerimientos | 19 |
| 1.7. LOS COSTOS DE LA NO CALIDAD | 20 |
| 1.8. PLANTEAMIENTO DEL PROBLEMA | 21 |

CAPÍTULO II

| | |
|--|-----------|
| METODOLOGÍAS ACTUALES UTILIZADAS | 23 |
| 2.1. MODELOS TRADICIONALES | 23 |
| 2.1.1. Descomposición Funcional | 23 |
| 2.1.2. Análisis Estructurado | 23 |
| 2.1.3. Especificación Operacional | 24 |
| 2.1.4. Análisis Orientado a Objetos | 25 |
| 2.2. OTROS MODELOS | 26 |
| 2.3. METODOLOGÍA ANCORA | 27 |
| 2.4. METODOLOGÍA DORCU | 31 |
| 2.5. METODOLOGÍA ÁGIL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS | 42 |
| 2.5.1. Elicitación | 43 |
| 2.5.2. Especificación | 48 |
| 2.5.3. Gestión | 49 |
| 2.6. EVALUACIÓN DE LAS METODOLOGÍAS | 53 |
| 2.7. MATRIZ DE EVALUACIÓN DE LA METODOLOGÍA PROPUESTA | 57 |

CAPÍTULO III

| | |
|--|-----------|
| ESTÁNDARES IEEE PARA REQUERIMIENTOS | 62 |
| 3.1. INTRODUCCIÓN A LOS ESTÁNDARES IEEE | 62 |
| 3.2. ESTÁNDAR IEEE 830-1998: ESPECIFICACIONES DE LOS REQUERIMIENTOS DEL SOFTWARE | 63 |
| 3.2.1. Esquema de la Especificación de Requerimientos del Software definida en el IEEE 830-1998 | 63 |
| 3.3. ESTÁNDAR IEEE 1233-1998: ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA | 71 |
| 3.3.1. Esquema de la Especificación de Requerimientos del Sistema | 72 |
| 3.4. ESTÁNDAR IEEE 1362-1998: GUÍA PARA TECNOLOGÍA INFORMÁTICA DEFINICIÓN DE SISTEMA-CONCEPTO DE OPERACIONES Y DOCUMENTOS (ConOps) | 101 |

| | |
|--|-----|
| 3.4.1. Esquema de un documento de Concepto de Operaciones | 102 |
| 3.5. VENTAJAS DE USAR ESTÁNDARES EN LA DOCUMENTACIÓN DE REQUERIMIENTOS | 112 |
| 3.6. ANÁLISIS DE LAS OPORTUNIDADES DE MEJORA | 113 |

CAPÍTULO IV

| | |
|--|------------|
| METODOLOGÍA PROPUESTA | 118 |
| 4.1. ELICITACIÓN DE REQUERIMIENTOS | 119 |
| 4.1.1. Introducción | 119 |
| 4.1.2. La elicitación de requerimientos | 119 |
| 4.1.3. Problemas de la elicitación de requerimientos | 120 |
| 4.1.4. Propuesta metodológica para la elicitación de requerimientos | 122 |
| 4.1.4.1. Tarea 1: Obtener información sobre el dominio del problema y el sistema actual. | 123 |
| 4.1.4.2. Tarea 2: Preparar y realizar las sesiones de elicitación/negociación | 124 |
| 4.1.4.3. Tarea 3: Identificar/revisar los objetivos del sistema | 124 |
| 4.1.4.4. Tarea 4: Identificar/revisar los requerimientos de almacenamiento de información. | 125 |
| 4.1.4.5. Tarea 5: Identificar/revisar los requerimientos funcionales | 126 |
| 4.1.4.6. Tarea 6: Identificar/revisar los requerimientos no funcionales | 126 |
| 4.1.4.7. Tarea 7: Priorizar objetivos y requerimientos | 127 |
| 4.1.4.8. El documento de requerimientos del sistema | 127 |
| 4.2. ANÁLISIS DE REQUERIMIENTOS | 127 |
| 4.2.1. Introducción | 127 |
| 4.2.2. Propuesta metodológica para el análisis de requerimientos | 128 |
| 4.2.2.1. Tarea 1: Analizar los requerimientos de almacenamiento de información | 128 |
| 4.2.2.2. Tarea 2: Analizar los requerimientos funcionales | 129 |
| 4.2.2.3. Tarea 3: Analizar los requerimientos no funcionales | 129 |
| 4.2.2.4. Tarea 4: Desarrollar prototipos | 130 |
| 4.2.3. El documento de análisis del sistema | 131 |

| | |
|---|-----|
| 4.3. VALIDACIÓN DE REQUERIMIENTOS | 131 |
| 4.3.1. Introducción | 131 |
| 4.3.2. Validación y verificación de requerimientos | 132 |
| 4.3.3. El compromiso de la validación de requerimientos | 134 |
| 4.3.4. Propuesta metodológica para la validación de requerimientos | 135 |
| 4.3.5. Tarea 1: Validar los requerimientos de almacenamiento de información y funcionales | 135 |
| 4.3.6. Tarea 2: Validar los requerimientos no funcionales | 136 |
| 4.3.7. Tarea 3: Cerrar la versión de los requerimientos | 137 |
| 4.4. ESPECIFICACIÓN DE REQUERIMIENTOS | 137 |
| 4.5. COMPARACIÓN CON LAS METODOLOGÍAS EXISTENTES | 138 |
| 4.6. APLICACIÓN DE LA METODOLOGÍA PROPUESTA | 139 |
| 4.6.1. Elicitación de Requerimientos | 139 |
| 4.6.2. Análisis de Requerimientos | 146 |
| 4.6.2.1. Prototipos | 146 |
| 4.6.3. Validación de Requerimientos | 147 |
| 4.6.4. Especificación de Requerimientos | 150 |
| 4.6.5. Diseño | 158 |
| 4.6.5.1. Diagrama de arquitectura | 158 |
| 4.6.5.2. Diseño de interfaces | 159 |
| 4.6.6. Implementación | 161 |
| 4.6.6.1. Diagrama de Despliegue | 161 |
| 4.6.6.2. Ejemplo del código del sistema | 162 |
| | |
| CONCLUSIONES | 164 |
| | |
| RECOMENDACIONES | 165 |
| | |
| BIBLIOGRAFÍA | 166 |

ÍNDICE DE FIGURAS

| | Contenido | Pág. |
|------------|---|-------------|
| Fig. 1.1. | Dimensiones de los requerimientos | 8 |
| Fig. 1.2. | Los requerimientos como restricciones | 10 |
| Fig. 1.3. | Ejemplo de modelo de proceso de Ingeniería de Requerimientos | 13 |
| Fig. 1.4. | Modelo tradicional en cascada | 13 |
| Fig. 1.5. | Modelo en Espiral | 14 |
| Fig. 1.6. | Distribución del tiempo y costo | 16 |
| Fig. 1.7. | Fase de mayor consumo de recursos y principal origen de defectos | 17 |
| Fig. 1.8. | Mayores problemas que afectan el crecimiento de las empresas de Software del Ecuador | 18 |
| Fig. 1.9. | Principales problemas que enfrenta la industria de desarrollo de software | 19 |
| Fig. 1.10. | Principal razón de retrasos, baja calidad y rehacer trabajo | 21 |
| Fig. 2.1. | Ciclo de vida del Análisis de Requerimientos de Software en Áncora | 27 |
| Fig. 2.2. | Esquema de la Metodología Flexible propuesta (DoRCU) | 33 |
| Fig. 2.3. | Etapas de Elicitación de Requerimientos | 37 |
| Fig. 2.4. | Metodología ágil para el proceso de ingeniería de requerimientos | 42 |
| Fig. 3.1. | Estructura de una ERS | 64 |
| Fig. 3.2. | Contexto de desarrollo de SyRS | 79 |
| Fig. 3.3. | Proceso de desarrollo de la SyRS | 93 |
| Fig. 3.4. | Estructura de un documento de Concepto de Operaciones | 103 |

| | | |
|-----------|--|-----|
| Fig. 4.1. | Metodología Propuesta | 118 |
| Fig. 4.2. | Tareas de elicitación de requerimientos | 122 |
| Fig. 4.3. | Tareas de análisis de requerimientos | 128 |
| Fig. 4.4. | Tareas de validación de requerimientos | 135 |
| Fig. 4.5. | Comparación con las metodologías existentes | 138 |
| Fig. 4.6. | Casos de uso del sistema | 142 |
| Fig. 4.7. | Casos de uso del sistema (Administrador) | 152 |
| Fig. 4.8. | Casos de uso del sistema (Secretaria–Profesor) | 152 |

ÍNDICE DE TABLAS

| | Contenido | Pág. |
|-------------|---|-------------|
| Tabla 2.1. | Entradas por tipo de proyecto | 43 |
| Tabla 2.2. | Criterios para la distinción de proveedores de requerimientos | 44 |
| Tabla 2.3. | Estrategias para la recolección de requerimientos | 45 |
| Tabla 2.4. | Salida de la fase 1 para proyectos de desarrollo de software a la medida e idea de negocio. | 47 |
| Tabla 2.5. | Entradas para la fase de especificación de requerimientos | 48 |
| Tabla 2.6. | Salida fase de especificación de requerimientos | 49 |
| Tabla 2.7. | Entradas fase de gestión de requerimientos | 49 |
| Tabla 2.8. | Salida fase de gestión de requerimientos | 52 |
| Tabla 2.9. | Metodología del Modelo Tradicional | 53 |
| Tabla 2.10. | Metodología del Modelo en Espiral | 53 |
| Tabla 2.11. | Metodología Ancora | 54 |
| Tabla 2.12. | Metodología Dorcu | 55 |
| Tabla 2.13. | Metodología Ágil | 56 |
| Tabla 2.14. | Matriz de la evaluación de la metodología propuesta | 58 |
| Tabla 3.1. | Formato para registro del personal involucrado | 104 |
| Tabla 3.2. | Formato para registro de documentos referenciados | 105 |
| Tabla 3.3. | Formato para registro de tipos de usuarios | 107 |
| Tabla 3.4. | Formato para registro de tipos de usuarios | 111 |
| Tabla 4.1. | Problemas de la elicitación de requerimientos | 121 |

INTRODUCCIÓN

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de requisitos de la fase de ingeniería de requisitos del desarrollo de software.

En esta metodología se distinguen dos tipos de productos: los productos entregables y los productos no entregables o internos. Los productos entregables son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los no entregables son productos internos al desarrollo que no se entregan al cliente.

La estructura de este documento es la siguiente:

En el capítulo 1 se describe las generalidades de la ingeniería del software en especial las motivaciones por las cuales se optó por investigar este tema, describir la necesidad de una ingeniería de requerimientos y los principales estudios de fracasos de muchos sistemas.

El capítulo 2 hace una breve descripción de algunas metodologías existentes en la actualidad como los modelos tradicionales, otros modelos como los grafos, se revisa también la metodología Ancora y Dorcu cada una con sus respectivas etapas y ciclos de vida.

En el capítulo 3 se hace un estudio sobre los estándares IEEE y que son los siguientes:

- IEEE Std. 830 Prácticas recomendadas para las especificaciones de software.
- IEEE Std. 1233 Estándar para la especificación de requerimientos del sistema.
- IEEE Std. 1362 Guía para la especificación del documento de requerimientos "ConOps".

Cada uno de los cuales se describe con su respectivo formato y cláusulas que se deben cumplir para generar documentos que cumplan con estas normas.

El capítulo 4 describe la propuesta metodológica, en sus tres fases como son: la elicitación de requerimientos que contempla siete tareas, la fase de análisis de requerimientos que tiene cuatro tareas y finalmente la fase de validación de requerimientos que tiene tres tareas, y como resultado la especificación de requerimientos utilizando el formato IEEE 830.

Finalmente describe las conclusiones y recomendaciones que produjo el presente trabajo.

CAPÍTULO I

GENERALIDADES

1.1. MOTIVACIÓN

A través de los años se ha podido constatar que los requerimientos o requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que será uno de los principales mecanismos de control con los que se contará durante la etapa de desarrollo. Además la especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema y es contra lo que se va a estar verificando si se están cumpliendo las metas trazadas.

Es muy frecuente escuchar entre los conocedores del desarrollo de software, que un gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos. Dentro de esa mala administración se pueden encontrar factores como la falta de participación del usuario, requerimientos incompletos y el mal manejo del cambio a los requerimientos.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que se enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas.

1.1.1. Necesidad de ingeniería de requerimientos

El desarrollo de software es un proceso por el cual, dado un problema, se encuentra un programa (o un conjunto de programas) que lo resuelva eficientemente.

Una de las dificultades esenciales de este proceso consiste en que la descripción del problema a resolver, suele ser poco precisa o incompleta. Sin embargo, esta dificultad no cobró importancia hasta el final de la década del 60, dado que hasta ese momento las computadoras sólo eran usadas para resolver problemas científicos, los cuales estaban expresados en un lenguaje suficientemente preciso.

Con el abaratamiento de los costos, las computadoras comenzaron a ser utilizadas para problemas originados en otros ámbitos (en general problemas administrativos). El principal método utilizado entonces, era simplemente partir del problema, el cual estaba expresado de manera informal y poco detallada, y obtener un programa que, por definición, es muy detallado y está escrito en una notación formal. Este salto, fue una de las principales razones de la llamada "crisis del software", la cual produjo un decaimiento en la confiabilidad del software y por lo tanto un decaimiento de los métodos usados para su desarrollo. Otra dificultad de esta metodología, se presentaba a la hora de la corrección del programa. Para ello, se realizaban ensayos con conjuntos de datos para los cuales se conocía el resultado, y si éstos daban los resultados esperados se daba por terminada la tarea. En el caso frecuente en que los resultados no fueran correctos, se procedía a modificar el programa para intentar corregirlo. Esta tarea era sumamente difícil, debido a que no se sabía a priori si el resultado inesperado se debía a meros errores de programación o a una concepción inadecuada del problema.

Uno de los grandes problemas de la ingeniería del software ha sido y es que no ha sabido adaptarse consecuentemente a su propia definición. Esto es algo

que se puede considerar como una especie de traición a sí misma, a sus propios fundamentos. El enfoque sistemático y cuantificable ha tenido siempre como barreras las propias de las formas en las que el software se ha desarrollado y distribuido. El formato binario del software, la opacidad en los modelos de negocios, los secretos y barreras comerciales se encuentran entre las principales causas que han imposibilitado estudios cuantitativos y cualitativos a gran escala del software cuyos resultados pudieran ser verificados sistemáticamente por equipos de investigación independientes. Las "verdades" que han sido enunciadas son, con frecuencia, experiencias puntuales que han sido generalizadas y dadas por válidas ante la falta de alternativas. La propia forma de desarrollar, distribuir y comercializar software ha sido la que ha llevado a la ingeniería del software a la crisis.

1.2. INGENIERÍA DE REQUERIMIENTOS

El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos es entregar una especificación de requerimientos de software correcta y completa.

Algunos otros conceptos de ingeniería de requerimientos son:

"Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software". (Pressman, 2006: 155)

"La ingeniería de requerimientos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema". (Sommerville, 2005: 82)

En síntesis, el proceso de ingeniería de requerimientos se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto de software determinado, donde es muy importante tomar en cuenta que el aporte de la Ingeniería de Requerimientos vendrá a ayudar a determinar la viabilidad de llevar a cabo el software (si es factible o no), pasando posteriormente por un subproceso de obtención y análisis de requerimientos, su especificación formal, para finalizar con el subproceso de validación donde se verifica que los requerimientos realmente definen el sistema que quiere el cliente.

1.3. EL CONCEPTO DE REQUERIMIENTOS

Se presenta a continuación la definición existente en el glosario de la IEEE de lo que es un "Requerimiento":

1. "Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo". (Std 610.12-1900, IEEE: 62)
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". (Std 610.12-1900, IEEE: 62)

También, Ian Sommerville presenta una definición acerca de lo que es un "Requerimiento":

3. "Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste". (Sommerville, 2005: 108)

Analizando las definiciones anteriores, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Esta definición expresa la perspectiva clásica de los requerimientos como elementos de un producto, o criterios para acuerdos. Sin embargo, otros autores son más específicos frente a la relación de los requerimientos con relación al sistema que van a representar: "... Los requerimientos son una especificación de lo que debe ser implementado. Estos son descripciones de cómo el sistema se debe comportar, de las propiedades y atributos del mismo. Deben ser una restricción del proceso de desarrollo del sistema..."¹.

Esta definición está muy ligada a lo que constituye el desarrollo de un sistema. Otra definición, que justifica la necesidad de los requerimientos frente a las perspectivas del usuario y del sistema es: "... Un requerimiento es algo que el producto debe hacer o una cualidad que el producto debe tener. Un requerimiento existe ya sea porque el tipo de producto demanda ciertas necesidades o cualidades, o porque el cliente desea que ese requerimiento sea parte del producto entregado..."².

1.3.1. Las dimensiones de los requerimientos

La gran cantidad de calificativos que se aplican al término requerimiento muestran distintos aspectos ortogonales que a menudo se consideran aisladamente.

¹ SOMMERVILLE, Ian y SAWYER, Peter. Requirements engineering: A good practice guide. 3 ed. Chichester, Inglaterra: John Wiley & Sons Ltd., 2000.

² ROBERTSON, Suzanne y ROBERTSON, James. Mastering the requirements process. Londres: Addison - Wesley, 1999.

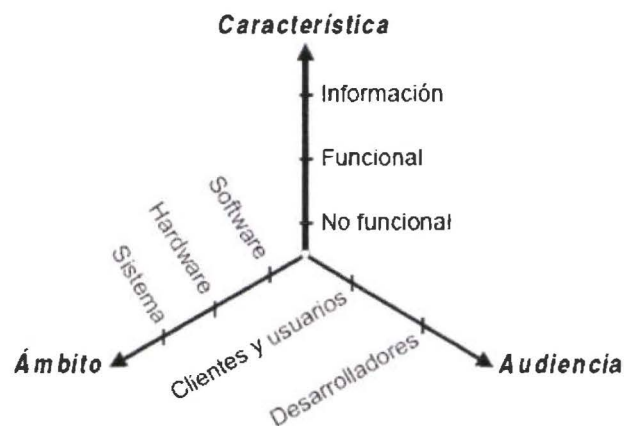


Fig. 1.1. Dimensiones de los requerimientos

Para intentar clarificar la situación, se ha identificado tres dimensiones en las que se pueden clasificar los requerimientos (ver figura 1.1). Estas tres dimensiones son:

- **Ámbito:** esta dimensión indica en qué ámbito se debe entender el requerimiento. En general, y siguiendo entre otras las propuestas de [IEEE 1997], un ámbito de sistema indica que el requisito 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.
- **Característica que define:** esta dimensión clasifica los requerimientos 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 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.

- **Audiencia:** esta dimensión, fundamental desde nuestro punto de vista, indica la audiencia a la que está dirigido el requisito, 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 requerimientos es mediante lenguaje natural. En el caso de que la audiencia prevista esté formada por desarrolladores de software, los requerimientos suelen expresarse mediante un modelo³, normalmente utilizando técnicas estructuradas, orientadas a objetos o formales.

1.3.2. Los requerimientos como restricciones

Los requerimientos también pueden interpretarse como restricciones que determinan el espacio de sistemas válidos [Gause y Weinberg 1989, Davis 1990] (ver Fig. 1.2).

Si S_p es el espacio de todos los posibles sistemas, cada requisito R_i establece una partición formada por S_i y \bar{S}_i , donde S_i representa al conjunto de sistemas que satisfacen el requisito R_i y \bar{S}_i representa al conjunto de sistemas que no lo satisfacen.

En esta situación, S_v , el conjunto de los sistemas válidos, es decir aquellos que satisfacen todos los requerimientos, se correspondería con la intersección de todos los conjuntos S_i . En otras palabras: $S_v = S_1 \cap S_2 \cap \dots \cap S_n$.

³ Principalmente los requerimientos de información y los funcionales. Como ya se comentó, los no funcionales no suelen contemplarse en las técnicas de modelado.

Desde este punto de vista, la ingeniería de requerimientos puede considerarse como el proceso iterativo de exploración [Gause y Weinberg 1989] para determinar las restricciones correctas, es decir los requerimientos, que determinen adecuadamente el espacio de sistemas válidos.

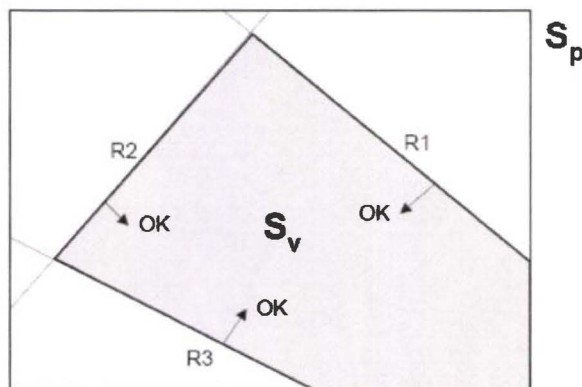


Fig. 1.2. Los requerimientos como restricciones

1.3.3. Propiedades deseables de los requerimientos

Un conjunto de requerimientos deben presentar una serie de propiedades tanto individualmente como en grupo.

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

- **No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
- **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

1.4. EL ANÁLISIS DE REQUERIMIENTOS COMO DISCIPLINA DE INGENIERÍA

La Ingeniería de Requerimientos es un término relativamente nuevo que cubre el conjunto global de las actividades relacionadas con el descubrimiento, la documentación y el mantenimiento del conjunto de requerimientos concernientes a un sistema interactivo.

De varias definiciones de la Ingeniería de Requerimientos la que al parecer, es la que proporciona el mejor enunciado es la propuesta por Zave⁴: La Ingeniería de Requerimientos es la rama de la Ingeniería del Software que se preocupa de los objetivos reales del mundo, de las funcionalidades y de las restricciones de los sistemas software. Ésta también se preocupa de la relación entre estos factores precisando especificaciones del comportamiento del software y acerca de su evolución con el tiempo.

Los objetivos principales de la Ingeniería de Requerimientos pueden resumirse en los siguientes puntos:

- Capturar un conjunto completo de requerimientos de los usuarios.
- Analizar detalladamente los requerimientos de los usuarios, encontrar todas las implicaciones de los mismos y comprenderlas.

⁴ Zave, P. (1995). Classification of research efforts in requirements engineering. In: Harrison, MD and Zave, P. (ed.) Proceedings 1995 International Symposium on Requirements Engineering. IEEE Computer Society Press, Los Alamitos CA, págs. 214-216.

- Especificar cómo los requerimientos se manifestarán en el diseño del sistema.
- Completar el análisis de los requerimientos con un conjunto de restricciones aceptable en términos temporales y económicos.

1.5. MODELOS DE PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS

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

En definitiva, los modelos son abstracciones simplificadas y estandarizadas de actividades repetitivas, generalmente producidos desde un punto de vista determinado, por lo que pueden existir diferentes modelos para un mismo proceso.

Sin embargo, en el caso del proceso de Ingeniería de Requerimientos y desde una perspectiva "intelectual", podemos decir que todos esos diversos modelos parten de una misma base, un modelo "madre" que llamaremos "modelo-abstracto".

Este tipo de modelo nos brinda una vista preliminar del proceso, una secuenciación aproximada y general de las actividades que luego deberemos realizar. Así, presentamos el siguiente ejemplo, en donde cada uno de los compartimientos cubre una sección particular del proceso.

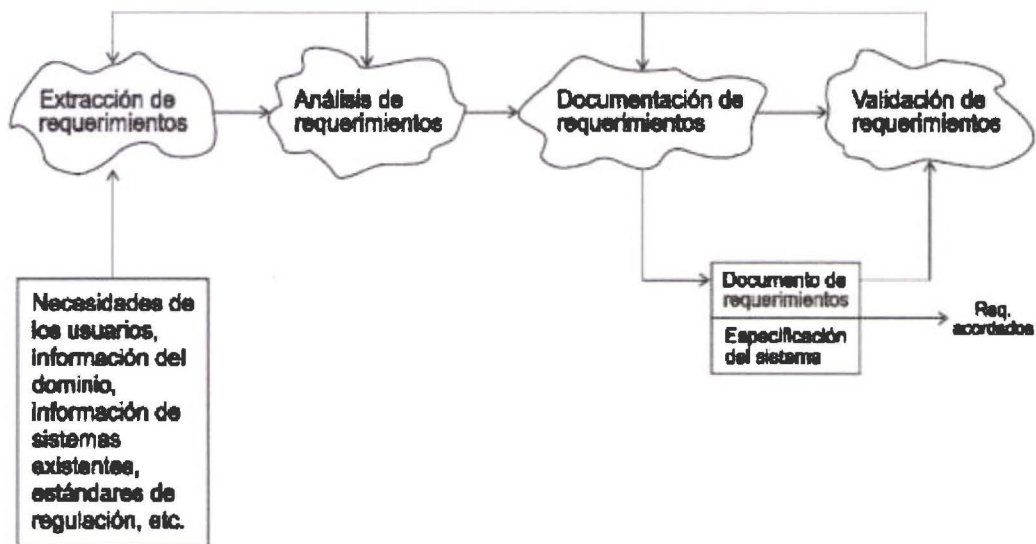


Fig. 1.3. Ejemplo de modelo de proceso de Ingeniería de Requerimientos

Las diversas necesidades de las diferentes organizaciones comienzan a surgir a partir de la aplicación de modelos más detallados. Así, tenemos dos modelos básicos que permiten estudiar el proceso de Ingeniería de Requerimientos y del cual se derivan numerosas variantes que dependerán del caso de estudio en cuestión.

a. Modelo tradicional en cascada

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

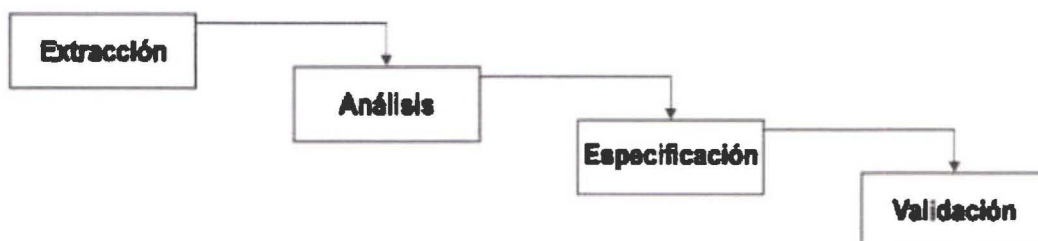


Fig. 1.4. Modelo tradicional en cascada

Si se ve a este modelo como una descripción general del proceso, es un modelo útil. Sin embargo, se debe entender que la realidad del proceso de Ingeniería de Requerimientos es mucho más compleja que lo que se vislumbra a partir del modelo en cascada: no existen fases claramente delimitadas ya que hay una retroalimentación constante entre las distintas etapas; los requerimientos del sistema van cambiando por circunstancias ajenas al proceso (como una ley nueva o un cambio de mercado que a su vez cambia las necesidades de la empresa) durante el desarrollo del mismo; se descubren problemas durante la validación que llevan a un cambio de requerimientos, etc.; y todo esto hará que más de una vez tengamos que volver "hacia atrás" en el proceso de Ingeniería de Requerimientos.

b. Modelo en espiral

Un modo alternativo de presentar modelos de actividad que toma en cuenta la retroalimentación entre etapas y la repetición de tareas, es el llamado Modelo en Espiral. [Kotonya G.; Sommerville I. 1998].

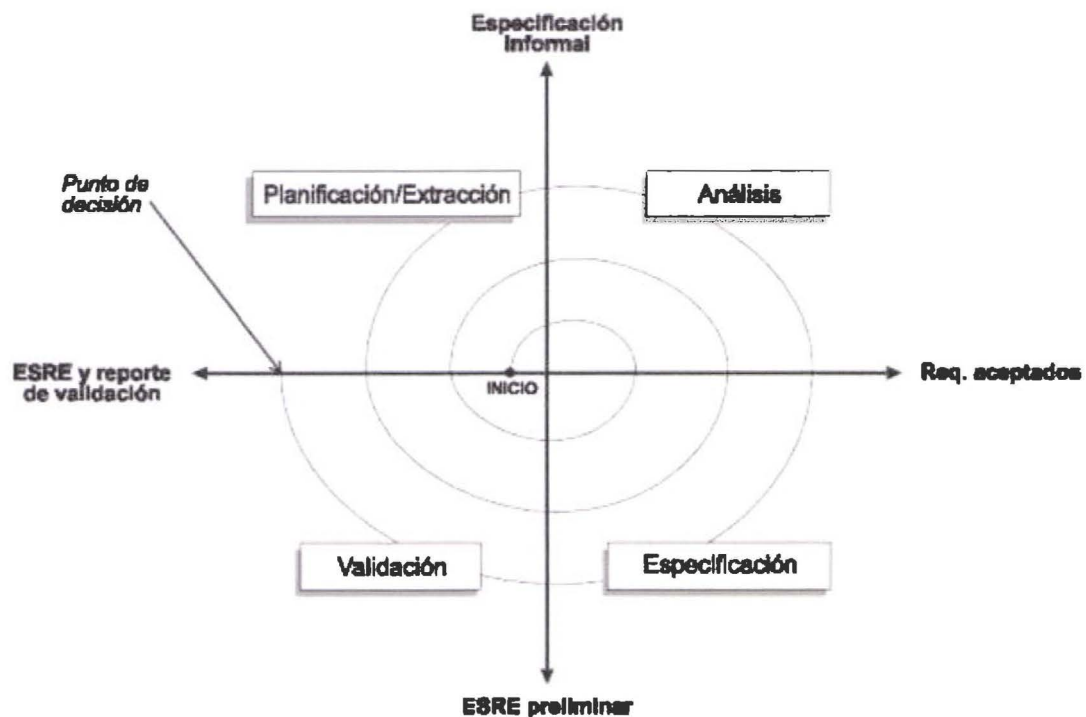


Fig. 1.5. Modelo en Espiral

En este diagrama, el uso de la espiral implica que las diferentes actividades de la ingeniería de requerimientos son repetidas hasta que se toma la decisión final, que es la aceptación del documento de especificación de requerimientos.

Es decir, si en el diseño preliminar se encuentran problemas, entonces recorreremos el ciclo nuevamente (extracción-análisis-especificación-validación) hasta que todos sean resueltos, que es lo mismo que decir que este ciclo continuará hasta que se pueda elaborar un documento aceptable.

Pero también existen factores externos que pueden determinar la finalización del ciclo, como por ejemplo la presión por cumplir con un determinado cronograma.

Luego del sucinto análisis de los dos modelos básicos antes mencionados, podemos concluir que dado el escenario de trabajo ("el analista se enfrenta a un dominio que desconoce y el cliente presenta un alto grado de incertidumbre con respecto al know-how de todos los procesos de su empresa") es más válida la aplicación del modelo en espiral para desarrollar el proceso de Ingeniería de Requerimientos. Y es que el modelo en espiral representa de manera más real cómo se irán desarrollando las actividades del proceso; esto es, debido al desconocimiento del tema, se genera un grado demasiado alto de incertidumbre que sólo puede disminuirse al repetir el ciclo de trabajo una y otra vez, permitiendo así ajustar todos los parámetros, cada vez en mayor detalle, hasta lograr un resultado satisfactorio.

1.6. SITUACIÓN DEL SOFTWARE EN EL ECUADOR

En el Ecuador desde hace pocos años se concientizó en la importancia de la ingeniería de requerimientos y los riesgos en que se incurre si ésta no se realiza o se la ejecuta en forma incompleta o incorrecta.

Actividades propias del desarrollo de software, como la adquisición de requerimientos del usuario, la especificación de requerimientos o la validación de los mismos, son algunos de los aspectos más críticos en el desarrollo y la producción del software. Estas actividades tienen por objetivo la determinación de las necesidades del sistema, la adquisición por parte del equipo de desarrollo de la información necesaria para desarrollar un producto de calidad, a la medida y en definitiva, la comunicación inicial entre el grupo de expertos y los clientes y usuarios. La ingeniería de requerimientos es un tema complejo y crucial para el éxito de todo el proyecto de ingeniería del software.

Las propuestas metodológicas para el desarrollo de software contemplan el tema de ingeniería de requerimientos con distinto grado de profundidad y proponen el uso de diferentes técnicas.

En estos últimos años, uno de los problemas de la industria del software fue el bajo nivel de calidad y de productividad; y los altos costos.

Estudios realizados en el año 2005 por egresados de la ESPOL sobre aspectos de la calidad y dificultades durante la gestión de proyectos establecen:

a) Distribución del tiempo y costo en las diferentes fases de desarrollo de software y de acuerdo al tamaño del proyecto:

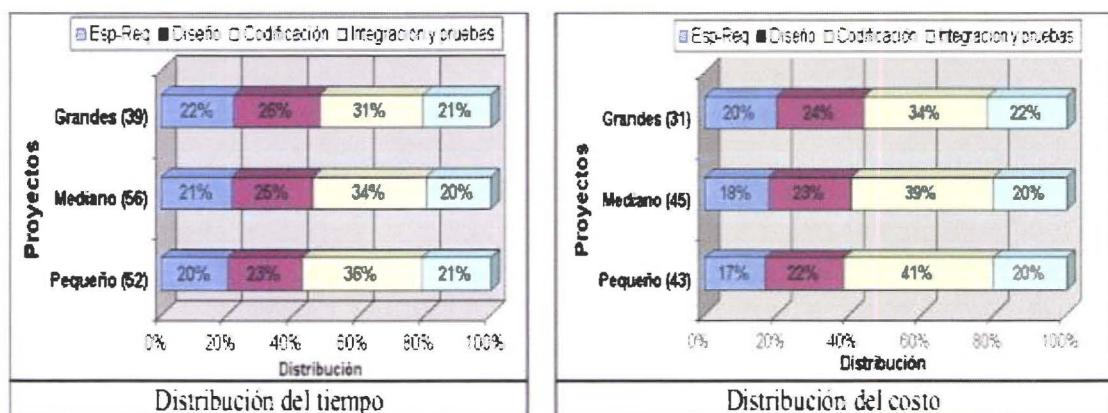


Fig. 1.6. Distribución del tiempo y costo

b) Fase de mayor consumo de recursos y principal origen de defectos de acuerdo al tamaño de los proyectos.

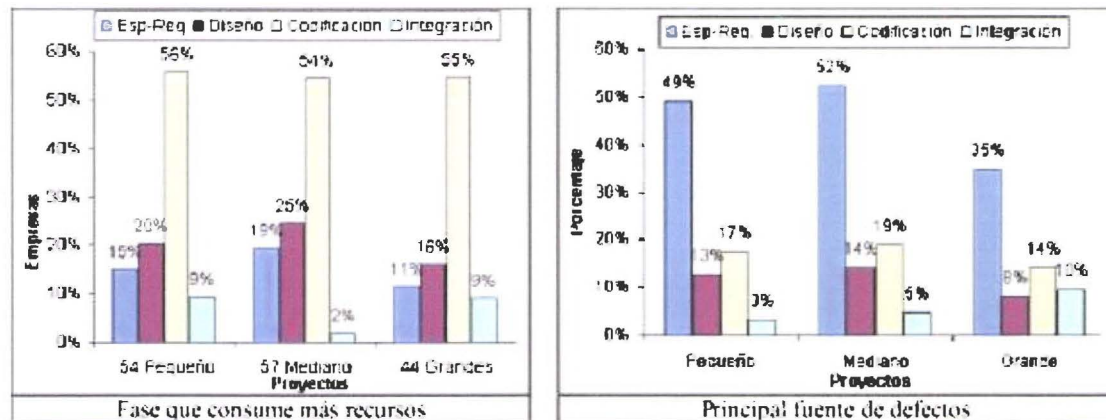


Fig. 1.7. Fase de mayor consumo de recursos y principal origen de defectos

Entonces, afirmar que la calidad proporcionará la solución puede parecer incorrecto, ya que solamente se ataca a uno de los problemas. Esta problemática representa cantidad de esfuerzo perdido en el desarrollo continuo en donde los productos, a menudo, son entregados con errores significativos que producen costos y posibles problemas y/o inconvenientes.

A través de un estudio, la AESOFT, buscaba generar una herramienta de gestión para el sector privado, gobierno y academia que permita incorporar planes de largo plazo en esta Industria clave para el desarrollo del país.

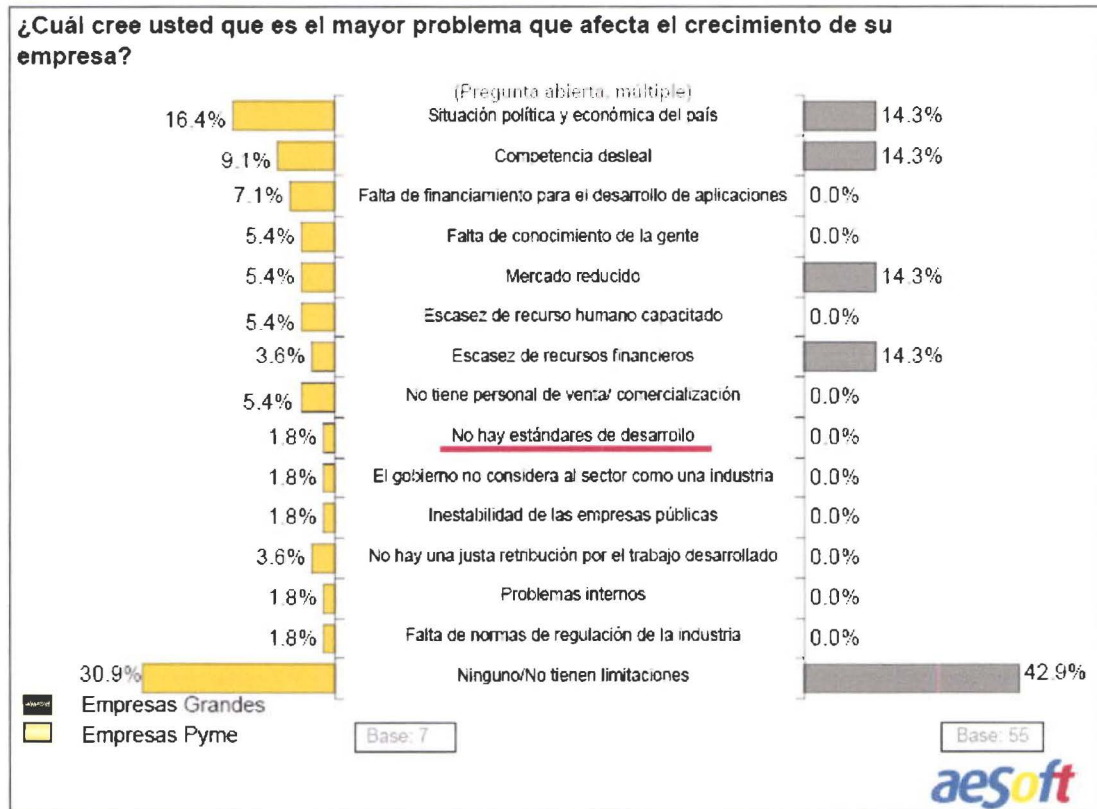


Fig. 1.8. Mayores problemas que afectan el crecimiento de las empresas de Software del Ecuador

Fuente: Primer Estudio de la Industria de Software del Ecuador. Aesoft 2005

Toda elección se desarrolla en el área de Software y es llevada a cabo teniendo en cuenta los propósitos que se pretenden alcanzar, el tiempo que demandaría la implantación del Modelo o Estándar seleccionado; y los costos y recursos asociados a dicha implantación.

Toda empresa forma parte de un mercado altamente competitivo y que exige altos niveles de calidad tanto en sus procesos como en sus productos.

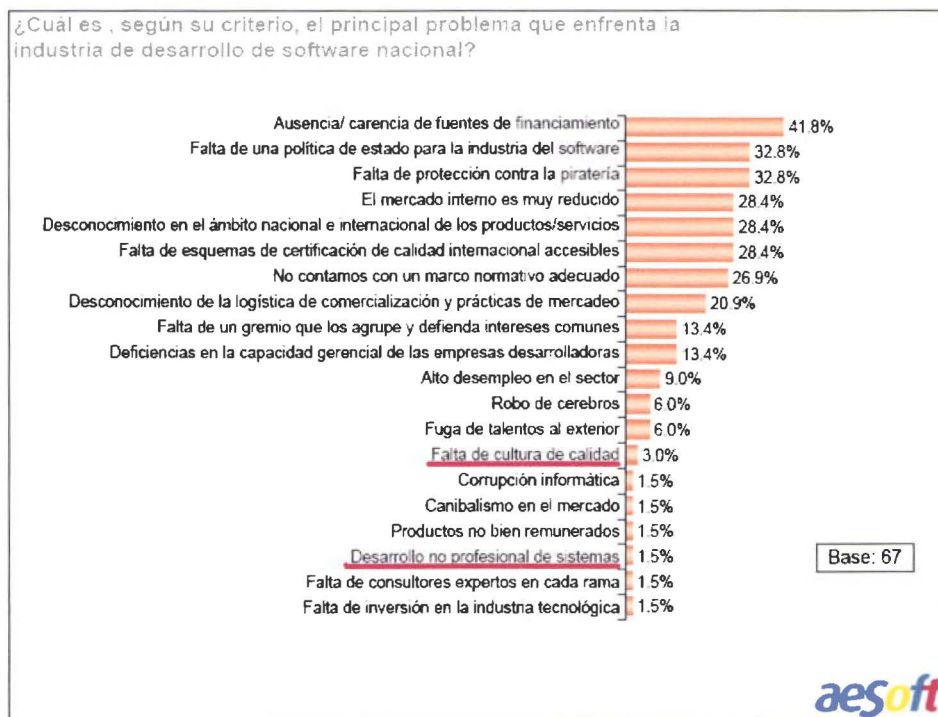


Fig. 1.9. Principales problemas que enfrenta la industria de desarrollo de software
Fuente: Primer Estudio de la Industria de Software del Ecuador. Aesoft 2005

1.6.1. Problemas en Requerimientos

Estudios realizados muestran que el 58.3%⁵ de los proyectos de software no establecen criterios para la aceptación de proveedores de requerimientos, lo que puede traer falencias para el desarrollo del proyecto por cuanto las fuentes de los requerimientos no tienen dominio de la problemática del sistema ni de sus procesos.

La falta de información sobre los factores que influyen la gestión de proyectos de software en las empresas del Ecuador, no permite identificar sus fortalezas y debilidades en la construcción y mantenimiento de software. Especialmente en lo que se refiere a la Ingeniería de Requerimientos, tomando en cuenta que los principales problemas en Requerimientos son:

- Los requerimientos no reflejan las necesidades reales de los clientes del sistema.

⁵ Revista Científica Guillermo de Ockham. Vol. 6, No. 1. Enero-Junio de 2008

- Los requerimientos son inconsistentes y/o incompletos.
- Es muy costoso el hacer cambios a los requerimientos después de que han sido acordados.
- Hay malos-entendidos entre clientes, analistas y desarrolladores.
- Los requerimientos definen que es lo que el sistema debe hacer y definen las restricciones del sistema.
- Los problemas en requerimientos llevan a una entrega retrasada y provoca cambios después de que el sistema entra en uso.
- La ingeniería de requerimientos abarca la elicitación, el análisis y la documentación de los requerimientos del sistema.
- El documento de requerimientos es la especificación definitiva de requerimientos para los clientes, ingenieros y administradores.
- El documento de requerimientos debe incluir un panorama general del sistema, glosario, definición de los requerimientos funcionales y las restricciones de operación.

1.7. LOS COSTOS DE LA NO CALIDAD

La calidad no cuesta. No es un regalo, pero es gratuita. Lo que cuesta dinero son las cosas que no tienen calidad —todas las acciones que resultan de no hacer bien las cosas a la primera vez⁶.

⁶ Philip B. Crosby: La Calidad no cuesta



Fig. 1.10. Principal razón de retrasos, baja calidad y rehacer trabajo

Los requerimientos deben definirse con claridad de modo que no puedan malinterpretarse. Entonces se toman medidas continuamente a fin de determinar el cumplimiento con dichos requerimientos. El no cumplir con los requerimientos significa ausencia de calidad. Los problemas de calidad se convierten en problemas de incumplimiento con los requerimientos.

1.8. PLANTEAMIENTO DEL PROBLEMA

La ingeniería de requerimientos es una disciplina donde se establece un conjunto de actividades que son utilizadas para elicitar, analizar, validar y administrar las necesidades del usuario, conocidas como requerimientos del sistema. Este proceso utiliza una combinación de técnicas, métodos y herramientas para obtener como resultado final el documento de especificación de los requerimientos. Dicho documento deberá contener una especificación completa, consistente y no ambigua del funcionamiento del sistema, utilizando lenguajes, diagramas y modelos de especificación de requerimientos. El documento de especificación de requerimientos, sirve para negociar y validar

los requerimientos de los usuarios, así también, como un contrato formal entre el cliente y el equipo de desarrollo. Este documento permite garantizar las características y funcionalidades del producto de software terminado.

En este trabajo de investigación, se iniciará realizando una evaluación de las diferentes metodologías existentes, una descripción de los estándares IEEE 830, 1233 y 1362 con sus ventajas y su aporte al mejoramiento de la calidad y estandarización de documentos de especificación de requerimientos; finalmente, identificarán y definirán las actividades de la ingeniería de requerimientos, las técnicas o métodos para la elicitación, análisis y validación de requerimientos, se definirán los diferentes tipos de requerimientos que se encuentran en un proyecto de software.

CAPÍTULO II

METODOLOGÍAS ACTUALES UTILIZADAS

2.1. MODELOS TRADICIONALES

Para entender mejor las técnicas que son utilizadas para llevar a cabo el análisis de requerimientos, hay que diferenciar 4 grandes perspectivas sobre las cuales se puede llevar a cabo este proceso.

2.1.1. Descomposición Funcional

Esta estrategia consiste en definir el comportamiento requerido (requerimientos) como una relación entre entradas y salidas de software. Se procede idealmente con una estructura **top-down** (arriba hacia abajo), identificando primero la funcionalidad del sistema como un todo. Después se procede a descomponer esta funcionalidad en un conjunto de funciones y subfuncionalidades. El resultado es una estructura jerárquica y de las funciones o funcionalidades y la definición de las interfaces funcionales. La ventaja de la descomposición funcional es que la especificación es escrita en el lenguaje y concepto de quienes implementan. Esto fomenta una buena comunicación de los requerimientos hacia los diseñadores y codificadores. La traducción al diseño y la codificación es sencilla debido a que la especificación de los requerimientos está escrita en términos del espacio de la solución que se necesita.

2.1.2. Análisis Estructurado

Este modelo está basado en la premisa que expone que las dificultades accidentales pueden ser enfrentadas con una aproximación sistemática del análisis del problema usando:

- Un modelo conceptual común para describir todos los problemas.

- Un conjunto de procedimientos que sugieran la dirección general del análisis y brinden un orden de pasos para el mismo.
- Una serie de pautas o soporte heurístico de decisiones acerca del problema y su especificación.
- Una serie de criterios para evaluar la calidad del producto.

Dentro de las prácticas comunes del análisis estructurado están los diagramas de flujo y los diccionarios de datos. Para tratar los problemas de comunicación y comprensión del espacio del problema, este tipo de herramientas utilizan un conjunto de estructuras conceptuales –una representación gráfica de la especificación en términos de estas estructuras- basándose en la hipótesis que la descomposición del problema en términos de datos que el sistema maneja será más clara y menos inclinada al cambio que otra basada en las funciones que el sistema debe realizar.

Esta técnica ha evolucionado y es ampliamente utilizada dentro del análisis de requerimientos pero también es criticada debido a sus falencias. Uno de los aspectos más criticados es que este tipo de análisis no provee suficiente asistencia ni guías. Los analistas tienen dificultad para decidir las partes del problema que deben ser modeladas y cómo deben ser modeladas. Por otro lado, mientras los pasos del proceso están definidos a grosso modo, las demás acciones que se deben seguir en el proceso son muy generales y difíciles de llevar a cabo, en especial si se aplican métodos heurísticos para obtener resultados. Esto conlleva a pensar que el análisis estructurado no facilita la formulación de un SRS (Software Requirement Specification) o documento de especificación de requerimientos de software que sea claro y con los atributos suficientes para satisfacer a todos los participantes del proyecto.

2.1.3. Especificación Operacional

Este modelo se enfoca principalmente en solucionar dos de los dilemas más importantes que rodean a los requerimientos. El primero, es que las personas

que están involucradas en el proceso de desarrollo no saben que desean construir, sino hasta que lo construyen. El segundo dilema, es el problema que se encuentra inmerso en el paso que implica pasar de una especificación de requerimientos a un diseño que satisfaga esa especificación. Entre más cercana esté la especificación del diseño, mejor y más fácil será la transición entre estas actividades, pero así mismo entre más cercanas son, las decisiones de diseño se convierten en decisiones prematuras.

Los elementos claves para una especificación operacional son:

- Un lenguaje de especificación formal.
- Un motor que permita obtener especificaciones correctas escritas en el lenguaje ya mencionado.

Esta aproximación también debería incluir un soporte automatizado para el análisis de las propiedades de la especificación formal y métodos para transformar dicha especificación en su correspondiente implementación.

2.1.4. Análisis Orientado a Objetos

Este modelo basa sus principios en realizar modelos de la información y el diseño orientado a objetos. Las técnicas del Análisis Orientado a Objetos (AOO) difieren del análisis estructurado en la forma en que se descomponen los problemas en sus partes, y los métodos a través de los cuales se descubren las relaciones entre dichas partes.

En este enfoque, el analista descompone el problema en un conjunto de objetos que interactúan entre sí, basados en las entidades y relaciones que existen en el dominio del problema. Un objeto encapsula un conjunto de datos, procesos y estados relacionados. En general, los componentes del análisis orientado a objetos son los objetos, sus datos y servicios que prestan, y las relaciones que tienen con otros objetos.

Este método representa muy bien el comportamiento del dominio de la aplicación que se desea realizar, pero no es soportado por un modelo conceptual que muestre el comportamiento del dominio del negocio.

Otro serio problema, es que la generalidad de esta aproximación se desvía más en el desarrollo de la aplicación y no permite concluir el objetivo específico de obtener un SRS completo.

2.2. OTROS MODELOS

Existen aproximaciones basadas en preceptos de las diferentes perspectivas, que satisfacen algunas de las actividades del análisis de requerimientos. Cysneiros, propone la utilización de grafos, en este caso dirigidos; en el análisis y descomposición de Requerimientos No Funcionales (RNF). En este caso, el autor sugiere que los requerimientos no funcionales representen metas a ser satisfechas. Cada meta debe ser descompuesta en subsecuentes submetas y estas a su vez descompuestas de igual manera hasta que se encuentren operacionalizaciones; estas son elementos que representan acciones o atributos que claramente identifican lo que es necesario para satisfacer la meta principal. Esta estructura permite el análisis de interdependencias y la detección de conflictos entre los requerimientos no funcionales.

De igual manera, se ha estudiado el uso de grafos y redes para mejorar la comprensión y comunicación de los requerimientos entre los stakeholders a través de modelos mentales y estructuras de pensamiento. Kudikyala, Lu y Torres; proponen en sus respectivos trabajos, clasificar requerimientos previamente recolectados y definidos en un documento de especificación; en grupos relacionados de acuerdo a la percepción de cada uno de los participantes involucrados.

Estos grupos de requerimientos son transformados en redes, y aplicando algoritmos para la búsqueda de caminos (Pathfinder Networks), se establecen

nuevas redes con un valor informativo y grupos de requerimientos interrelacionados (Cluster). De esta manera se realizan comparaciones entre las diferentes redes generadas y se buscan similitudes y no similitudes entre las mismas, para determinar conjuntos de requerimientos que sean conformes a los participantes del proyecto, y conjuntos de requerimientos que no correspondan a una correcta definición de requerimientos de acuerdo a las perspectivas de los stakeholders. En general, todas estas aproximaciones recrean un modelo que permite llevar a cabo alguna tarea específica de la ingeniería de requerimientos, aprovechando como recurso estructuras ampliamente conocidas como los grafos o redes.

Cabe recalcar que estas estructuras son tal vez la representación matemática más utilizada en los problemas de la vida real por su versatilidad para representar problemas.

2.3. METODOLOGÍA ANCORADA

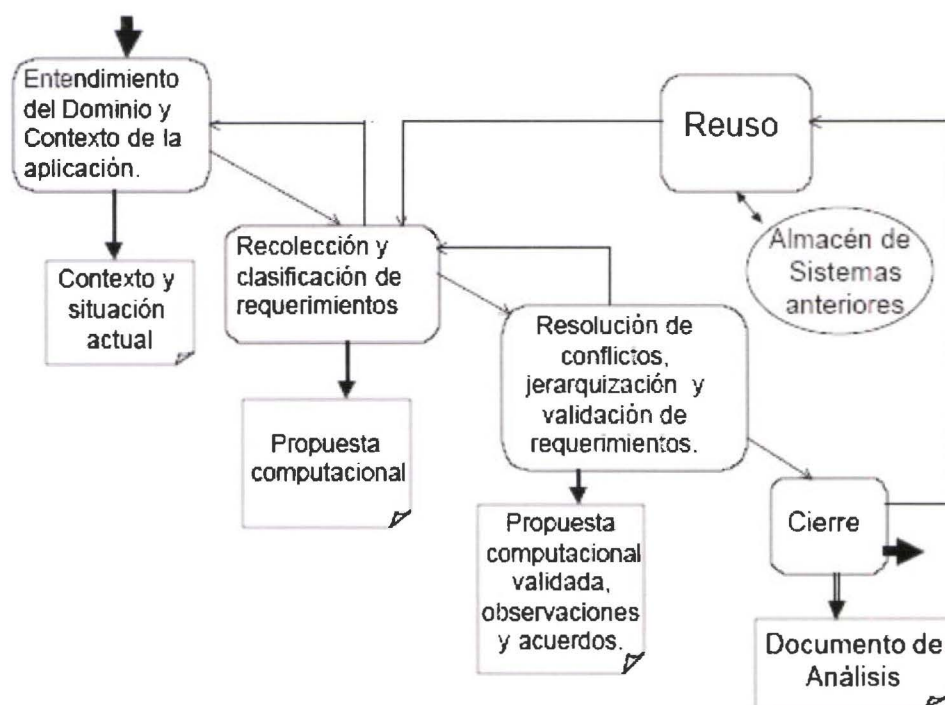


Fig. 2.1. Ciclo de vida del Análisis de Requerimientos de Software en Áncora

Las etapas que forman el ciclo de vida de Áncora se muestran en la Figura 2.1 comenzando con la flecha gruesa de entrada a la etapa de Entendimiento del Dominio y Contexto de la Aplicación y termina con la flecha gruesa de salida en la etapa de Cierre. Las flechas dobles indican los artefactos que arroja cada etapa, la flecha doble la interacción con el almacén magnético de sistemas anteriores y por último las flechas simples indican el paso de una etapa a otra, debe notarse que hay retroalimentación de unas etapas con otras, aunque no en todas.

Entendimiento del Dominio y Contexto de la Aplicación. En esta etapa se busca tener claros las metas de la empresa, el dominio de la aplicación, las actividades que se llevan a cabo en la situación actual del sistema para el cual se va a desarrollar el software de apoyo y la justificación del nuevo software. Para ello se llevarán a cabo las siguientes actividades que arrojan los artefactos correspondientes:

- **Leer material.** Con lo cual se podrá armar un primer glosario, esquema de la organización, lista tentativa de actividades que realizan los usuarios y guiones y diálogos de la situación teórica.
- **Entrevistas informales con el cliente.** Se obtendrán las metas de la empresa y se identificará la influencia del sistema en ellas.
- **Observar usuarios.** El analista podrá determinar cómo se realizan las tareas en el sistema actualmente y hará las anotaciones pertinentes.
- **Construir redes semánticas naturales del sistema.** Después de una reunión informal con los usuarios en la que se aplica cuatro veces la técnica de Red Semántica Natural (RSN) se tendrá un glosario más exacto y guiones y diálogos de la situación actual.
- **Construir y aplicar encuesta de actitud.** A partir de la red semántica natural del sistema se construye una encuesta de actitud que permitirá saber qué sentimientos hay entre los usuarios con respecto a la construcción del software e identificar los principales problemas existentes que permitirán redactar la justificación del nuevo software.

El orden de aplicación de cada actividad no es secuencial, se realizarán según las necesidades, usos y costumbres de la empresa.

Recolección, clasificación de Requerimientos. Esta etapa tiene como objetivo crear la propuesta computacional del sistema compuesta por:

- La realización de la propuesta computacional, que incluye: Guión de propuesta computacional, manual preliminar de operación (prototipo rápido), modelo de datos (Objetos Semánticos), lista de responsabilidades, lista de beneficios esperados.
- Creación de la Bitácora de desarrollo del sistema, la cual consiste en una tabla de cuatro columnas; cada una contiene: una función del sistema, la forma de comprobación por parte del usuario, el tiempo propuesto y tiempo real de implantación.
- Aplicación de puntos de función. Esto permitirá establecer las restricciones (requerimientos no funcionales) del software, refinar la propuesta computacional y dar una medida de complejidad al sistema.

Para obtener todos los artefactos que conforman la propuesta computacional, la primera actividad es construir el Guión y de ahí se puede proseguir con otros artefactos que, muy posiblemente, retroalimentarán al Guión y éste a su vez a otros artefactos. Durante la creación de la propuesta computacional se realizarán entrevistas informales a cliente y usuarios.

Reuso de requerimientos. El objetivo de esta etapa es brindar al analista elementos de otros sistemas de software que ya han sido probados y cuyas especificaciones de requerimientos se han almacenado en una Base de Datos de Reuso.

- **Abstraer y guardar los guiones de sistemas terminados.** Esta es una operación de generalización en la que, mediante transformaciones respectivas de verbos y sustantivos a primitivas y nombres genéricos y la representación de escenas y condiciones de entrada en forma clausular, se

obtendrá una representación abstracta de los guiones de la propuesta computacional.

- **Consultar y extraer elementos reutilizables.** Todos los artefactos de las especificaciones de requerimientos de sistemas semejantes que puedan reutilizarse, a saber: el guión o guiones de la propuesta computacional, manuales de operación, listas de restricciones y responsabilidades, bitácoras de desarrollo, cálculo de puntos de función y, si se uso el mismo paradigma de desarrollo elementos para la conexión con el análisis.

Resolución de conflictos, priorización y validación de requerimientos.

Como su nombre lo indica en esta etapa se busca resolver los conflictos que puedan existir entre los diferentes requerimientos, asignar prioridad a cada requerimiento y obtener la validación por parte del usuario de los requerimientos. Las actividades son dos.

- **Preparación de la Reunión de Reflexión y Diseño (RRD).** Una vez obtenidos los documentos de la situación actual y de la propuesta computacional, estos se reproducen y empastan, entregando una copia a cada involucrado (clientes, usuarios, analistas, diseñadores) en el software. El motivo es el que antes de llevar a cabo una reunión los involucrados puedan analizar los documentos poniendo sus anotaciones sobre el documento.
- **Reunión de Reflexión y Diseño.** Después de entregada la documentación se reúne a, por lo menos, un representante de cada grupo de interés y durante uno a dos días se lleva a cabo una reunión en donde todos los asistentes podrán verter sus opiniones y entre todos llegar a un contexto, además se aplicará la Técnica de Grupo Nominal para la jerarquización de requerimientos.

Al finalizar la reunión de reflexión y diseño se tendrán: Guiones, restricciones, prioridades, manual, responsabilidades y plan de prueba validados. Además de una lista de observaciones y acuerdos firmados de dicha reunión.

Cierre. En esta etapa se realizan una serie de actividades encaminadas a lograr una conexión con otras etapas del desarrollo del software. Las actividades son:

- Ratificar cálculo de Puntos de Función. Después de la reunión de reflexión y diseño, tal vez, será necesario hacer una revisión de los Puntos de Función.
- Envío de oficios, correo electrónico y faxes para aclarar puntos pendientes.
- Conexión con el modelo de desarrollo escogido. Se realizarán una serie de traducciones que llevarán al planteamiento de los artefactos del análisis.

2.4. METODOLOGÍA DORCU

DoRCU, Documentación de Requerimientos Centrada en el Usuario, es una metodología para la Ingeniería de Requerimientos caracterizada por su flexibilidad y orientación al usuario. Considera los mejores resultados de los enfoques examinados y se apoya en diversos métodos, técnicas y herramientas ya desarrollados por otros autores, pero sin comprometerse con los lineamientos de un paradigma en particular. Tiende, además, a que se unifique la terminología empleada en el campo de la Ingeniería de Requerimientos, eliminando de esta manera aparentes discrepancias que sólo son la consecuencia de confusiones semánticas que dificultan aún más el proceso de definición de requerimientos.

La metodología DoRCU (Documentación de Requerimientos Centrada en el Usuario), consta de las siguientes etapas:

- Elicitación de requerimientos
- Análisis de Requerimientos
- Especificación de Requerimientos
- Validación y Certificación de los Requerimientos

Y los objetivos que se proponen para cada una de ellas son:

- **Elicitación 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, requerimientos 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

Requerimientos 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.

Por consiguiente, la representación gráfica de la propuesta metodológica que se hace para la Ingeniería de Requerimientos es la que se puede ver en la figura 2.2.

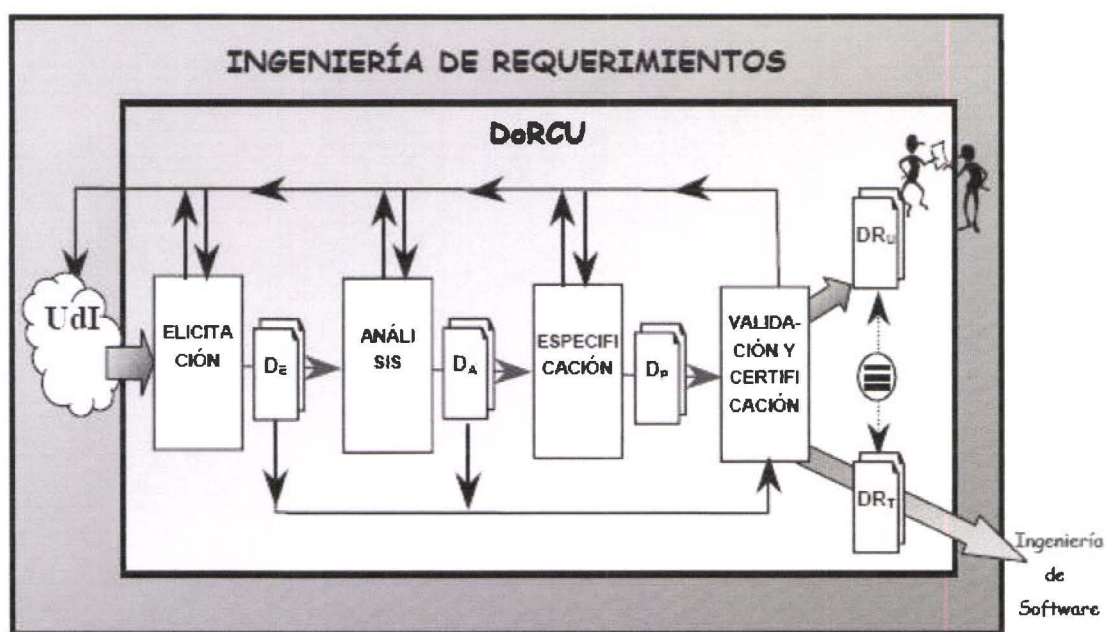


Fig. 2.2. Esquema de la Metodología Flexible propuesta (DoRCU)

Considerando el tronco metodológico planteado anteriormente, se detallan a continuación sus correspondientes subetapas, destacando en primer lugar que las subetapas de Elicitación de Requerimientos, exceptuando la a), son tomadas de la metodología propuesta por Christel⁷ debido a que se las considera apropiadas para los fines que se persiguen.

⁷ Christel M. G., Kang K. C. "Issues in Requirements Elicitation", Technical Report CMU/SEI-92-TR-12 ESC-TR-92-012 Software Engineering Institute. Pittsburgh. September 1992.

Etapa 1: Elicitación de requerimientos

En cuanto al proceso de elicitación de requerimientos, la propuesta metodológica que se considera apropiada consta de los siguientes pasos:

a) Formar el equipo multidisciplinario.

Considerando que la formación de la gente de sistemas, tratándose de problemas con alta incidencia del factor humano, no tiene la especialización necesaria como para diagnosticar el método de elicitación más apropiado para cada caso en particular, se aconseja que la recolección de requerimientos sea efectuada con el asesoramiento de profesionales especializados. Este asesoramiento puede extenderse incluso a un liderazgo activo de las sesiones de elicitación por parte de especialistas en ciencias de la comunicación o en ciencias del conocimiento.

b) Buscar hechos.

El primer paso en la elicitación de requerimientos está involucrado con el problema a ser encarado, y quién necesita ser involucrado en esta toma de decisión, tanto como quién se verá afectado por la formulación de los problemas y la eventual solución. Los resultados de esta actividad son: una declaración del contexto del problema, de los objetivos globales, límites e interfaces para el sistema original.

Este examen debe ser efectuado de manera tal que permita establecer, entre otros, cuál es el rol que desempeñará el sistema a desarrollar, sus objetivos y límites, las restricciones de arquitectura y la existencia o no de sistemas similares dentro de la organización.

c) Recolectar y clasificar requerimientos.

En esta etapa se obtienen: objetivos, necesidades y requerimientos de clientes y usuarios. Estas necesidades y requerimientos son verificadas comparándolas con los objetivos globales del sistema original expresados

durante el hallazgo de hechos. Es importante recolectar tanta información como sea posible.

Dependiendo de la manera en que el sistema se está desarrollando y los grupos que afectará, la etapa de recolección de requerimientos es una combinación de los enfoques composición y descomposición. Es importante en este momento, destacar los términos que son propios del lenguaje del Universo de Información.

Una vez recolectados los requerimientos, se debe proceder a clasificar los mismos en funcionales y no funcionales.

d) Evaluar y racionalizar.

Debe realizarse una valoración del riesgo, para encaminar las inquietudes técnicas, de costos y de tiempo. Debe examinarse la coherencia en la información reunida en subetapas previas, para determinar si los requerimientos verdaderos están escondidos o expresados explícitamente. Se realizan abstracciones para responder preguntas del tipo ¿Por qué usted necesita X?, y si esta pregunta tiene una respuesta concreta, entonces es un requerimiento, si no es un falso requerimiento.

Mediante el estudio comparativo de la información de requerimientos se ponen en evidencia las inconsistencias que pueden surgir entre los requerimientos extraídos.

Cabe destacar que tanto en la presente subetapa como en la anterior, se dan instancias de evaluación de factibilidad, negociables entre el cliente/usuario y el analista.

e) Dar prioridad.

En esta etapa, contando ya con requerimientos consistentes, se da un orden de prioridades, de manera tal que las necesidades de alta prioridad

pueden ser encaradas primero, lo que permite definir las y reexaminar los posibles cambios de los requerimientos, antes que los requerimientos de baja prioridad (que también pueden cambiar) sean implementados.

Durante el desarrollo del sistema, esto permite una disminución de los costos y ahorro de tiempo en procesamiento de los inevitables cambios de los requerimientos.

Los requerimientos deben tener prioridades basándose en las necesidades del usuario, el costo y la dependencia.

f) Integrar y validar.

Esta tarea se lleva a cabo de manera tal que sea posible obtener un conjunto de requerimientos, expresados en el lenguaje del usuario, de los cuales se pueda validar la consistencia con respecto a las metas organizacionales obtenidas en la primera etapa. Las tareas de integración deben ser ejecutadas principalmente por el analista de sistemas, y los resultados del proceso de elicitación comunicarlos a las otras comunidades involucradas. Esta validación de los requerimientos realizada por todas las partes afectadas, asegura que se alcanza lo deseado.

g) Documentar la etapa.

Elaborar la lista final de los términos del lenguaje del Universo de Información, y la de sentencias de los requerimientos obtenidos (D_E).

Como es de esperar, a los efectos de obtener buenos requerimientos, todos estos pasos deben iterar ante la menor inconsistencia detectada, aconsejándose que la iteración se realice recurriendo al cliente/usuario, tantas veces como sea necesario, para garantizar una correcta depuración del producto final de la etapa de elicitación.

Si se hiciera una representación gráfica de la propuesta metodológica para la etapa de elicitación de requerimientos, la misma quedaría bosquejada de la manera que se puede apreciar en la figura 2.3.

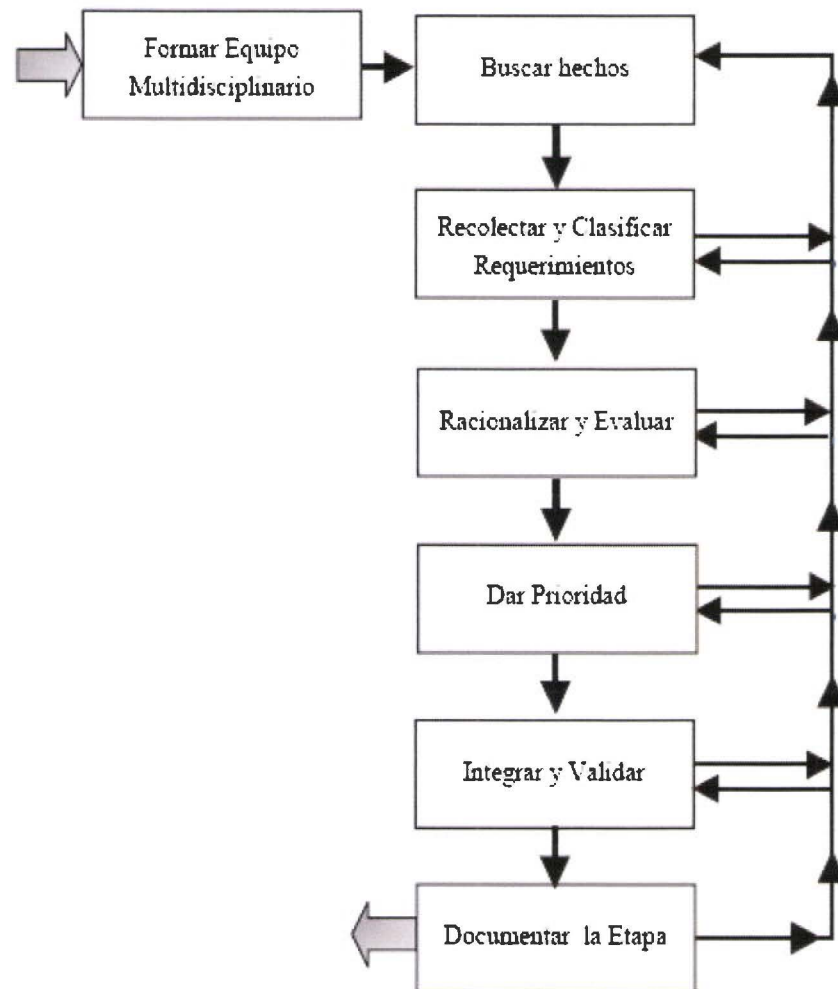


Fig. 2.3. Etapas de Elicitación de Requerimientos

Etapa 2: Análisis de Requerimientos

Los pasos a realizar durante esta etapa tienen como objetivo la obtención de buenos requerimientos⁸, para asegurar de esta manera que lo que se derivará a las etapas posteriores será de calidad tal que permita que disminuyan las fallas de los sistemas.

⁸ Hooks, Ivy. "Writing Good Requirements", Fourth INCOSE Symposium, 2000.

Las subetapas a contemplar durante esta etapa son:

a) Reducir ambigüedades en los requerimientos.

Los requerimientos obtenidos como resultado final de la etapa de elicitación, deben ser tratados a los efectos de llevarlos a una notación que permita reducir la ambigüedad del lenguaje del usuario. Por consiguiente, en esta subetapa se realizan las tareas que permiten eliminar los términos que tienen más de una acepción, unificando el léxico empleado en el Universo de Información.

b) Traducir a lenguaje técnico los requerimientos.

Los requerimientos, ya con menos ambigüedades, deben ser tratados a efecto de llevarlos a un lenguaje que se vaya aproximando al lenguaje técnico. Mediante esta traducción se busca aproximar los términos del usuario a los términos del sistema de software.

c) Plantear un modelo lógico.

Partiendo del lenguaje obtenido en la etapa anterior, transformarlo en una estructura preliminar, es decir, en un primer modelo lógico. De esta manera, en la presente subetapa se debe construir un modelo del problema ya sea en términos de diagramas de flujo o cualquier otro tipo de representación que se considere conveniente para el modelado y que permita, además, establecer un vínculo con la Etapa de Especificación.

d) Documentar la etapa.

Elaborar todo tipo de documento que se considere adecuado como soporte para la etapa siguiente. Este documento, dado el caso, puede resumirse a la colección de los modelos lógicos a que se ha arribado (D_A).

Etapa 3: Especificación de Requerimientos

Recién superadas las etapas anteriores debe comenzar a pensarse en la forma de describir los requerimientos. Para ello, se deben seguir las subetapas planteadas a continuación:

a) Determinar el tipo de requerimiento.

Considerando que existen diferentes tipos de requerimientos, determinar unívocamente a cuál de ellos pertenece el que se está tratando. Esto no significa que deba adoptarse una clasificación en particular, sino que aquí también queda de manifiesto la flexibilidad de la metodología, ya que cada analista de requerimientos puede utilizar la clasificación que considere como la más adecuada.

b) Elegir la herramienta de especificación acorde al tipo de requerimiento.

Una vez definido el tipo de requerimiento, seleccionar la herramienta de representación acorde a dicho tipo y al tipo de especificación que se desea realizar.

La única restricción al respecto es que la herramienta a seleccionar debe ser de índole formal o semiformal, ya que ellas son las únicas que permiten representar a los requerimientos sin ambigüedades.

c) Especificar de acuerdo a la herramienta seleccionada.

Representar el requerimiento sobre la base de la elección realizada en la etapa anterior. En caso de existir dificultades para su empleo, volver a la subetapa anterior para realizar una nueva selección o, incluso, a la primera ya que la dificultad de representación puede obedecer al intento de usar una herramienta para un requerimiento cuyo tipo ha sido mal definido, por lo cual se selecciona una inaplicable al caso.

d) Documentar la etapa.

Confeccionar el documento representativo de la etapa tomando como base a los modelos formales o semiformales que se han elaborado al realizar la especificación de los requerimientos. Incorporar al mismo toda extensión que se considere de utilidad para la etapa de Validación y Certificación de Requerimientos (D_P).

Etapa 4: Validación y Certificación de los Requerimientos

Todo el esfuerzo realizado durante las etapas anteriores puede darse por perdido si no es manifestado en forma correcta, ya que cualquier mala interpretación puede echar por tierra hasta el proceso de Ingeniería de Requerimientos más consistente, desvirtuando la naturaleza de lo que fue, hasta el momento de su declaración, un buen requerimiento. Tratando de minimizar aún más la posibilidad de error, se proponen las siguientes subetapas para su elaboración:

a) Seleccionar las fuentes de información a partir de las cuales validar el documento de especificación.

En esta etapa se procede a validar el documento de especificación D_P a partir de los documentos obtenidos de las etapas de elicitación (D_E) y análisis (D_A), seleccionando como fuente de información aquellos materiales que más aportan, por un lado a la claridad de su descripción y, por el otro, en cuanto a permitir la validación final entre los resultados de todas las etapas anteriores. El documento de especificación (D_P) validado se llamará, en adelante, documento de requerimientos técnico (DR_T).

b) Elegir o diseñar el modelo de documento acorde al grado de detalle requerido y al lector final.

Si bien muchos autores han propuesto modelos de documentación excelentes, es necesario decidirse por alguno de ellos. Dado el caso de que ninguno de los conocidos satisfaga las necesidades de documentación del analista de requerimientos, se deberá proceder a diseñar aquel que mejor se ajuste a sus necesidades.

c) Elegir la herramienta de documentación que mejor se aplica al modelo seleccionado.

Como no todos los modelos pueden ser plasmados con una misma herramienta, se debe seleccionar la que mejor se adecue al problema entre todas las alternativas posibles. Es así que en esta etapa se deberán tener

en cuenta, no sólo a los procesadores de texto sino también a herramientas de recursos gráficos que permitan la incorporación de diagramas y figuras, si se considera que lo antedicho ayuda a la interpretación que hará el usuario del Documento de Requerimientos.

d) Documentar respetando los estándares vigentes a la fecha de realización del documento de requerimientos.

Elaborar el documento de requerimientos orientado al usuario (DR_U) a partir del documento de requerimientos técnico (DR_T), realizando una traducción a un lenguaje entendible por aquél. Estos documentos deben ser elaborados respetando los estándares que existen a la fecha de su confección. Para ello, el personal de documentación debe estar al tanto de las normas IRAM e ISO y de las dictadas por instituciones como la IEEE. Como el DR_U tiene fines de certificación y contractuales, considerar como normas de redacción las disposiciones legales al momento de la confección.

e) Validar.

Verificar la correspondencia entre los documentos obtenidos de la etapa anterior, controlando que solo difieran en lo sintáctico y no en lo semántico, es decir que su contenido difiera solamente en el lenguaje utilizado para su definición, alcanzando de esta manera el isomorfismo entre DR_T y DR_U .

f) Certificar.

Proceder a la aprobación del DR_U por medio dno le estoy esperando que me diel conforme del cliente, y de esta manera dar por aprobado el Documento de Requerimientos Técnico DR_T , el que será utilizado por las restantes etapas de la Ingeniería de Software.

2.5. METODOLOGÍA ÁGIL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS

Basándose en el diagnóstico de estudios realizados por varios autores, la metodología ágil va dirigida a dos tipos de proyectos: desarrollo a la medida e ideas de negocio. La metodología se ha estructurado en tres fases (Ver figura 2.4.).

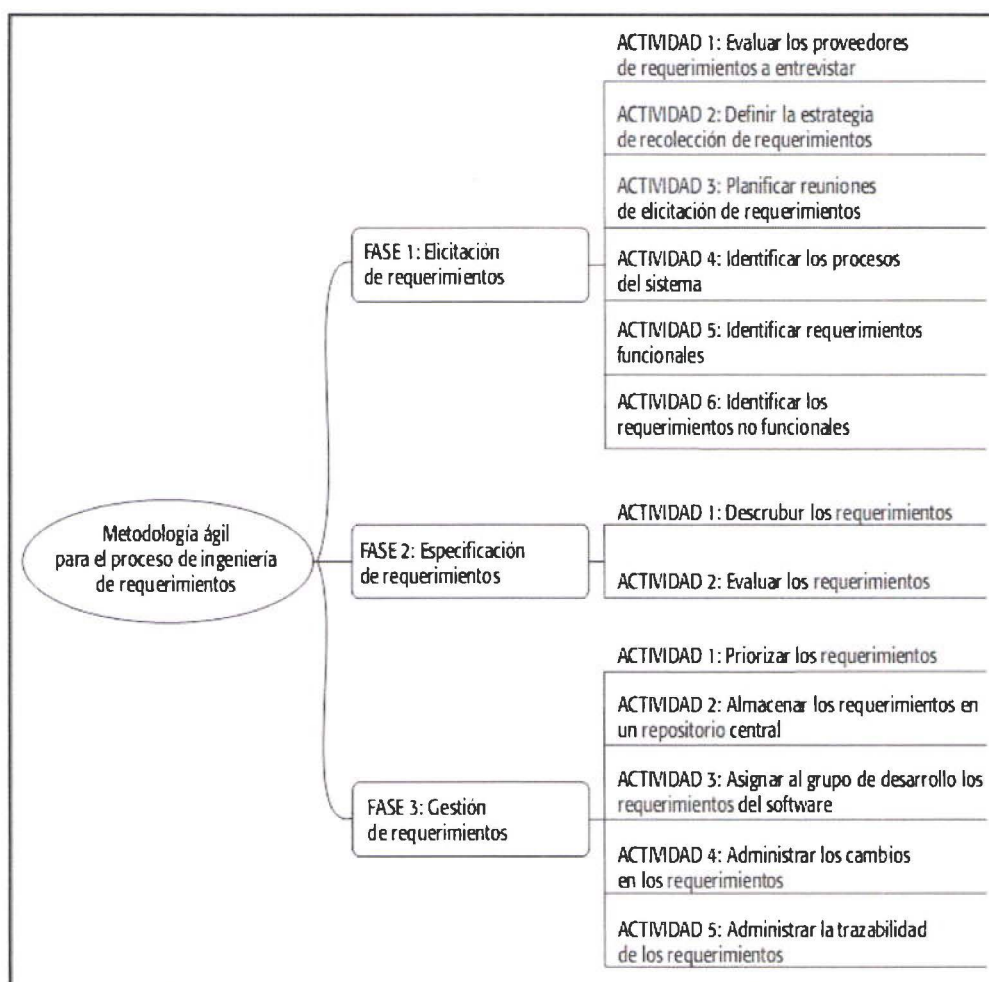


Fig. 2.4. Metodología ágil para el proceso de ingeniería de requerimientos

Fase 1: Elicitación de requerimientos

Su propósito es obtener un óptimo entendimiento de los requerimientos del software en los participantes –los usuarios finales y los desarrolladores–, e identificar los requerimientos funcionales y no funcionales.

Fase 2: Especificación de requerimientos

Su propósito es describir cada uno de los requerimientos del software, sus políticas y sus restricciones.

Fase 3: Gestión de requerimientos

Su propósito es administrar los cambios de los requerimientos, mantener las relaciones entre ellos e identificar las inconsistencias que puedan ocurrir en el transcurso del proyecto.

Cada fase incluye un conjunto de actividades a las cuales anteceden unas entradas propias del tipo de proyecto y les preceden unas salidas:

- Entradas.
- Actividades.
- Salidas.

2.5.1. Elicitación

Esta fase tiene tres entradas (Ver Tabla 2.1.) y seis actividades planteadas las cuales se detallan a continuación:

| TIPO DE PROYECTO | NOMBRE ENTRADA | FUENTE |
|------------------------------------|---|---|
| Proyecto de desarrollo a la medida | <ul style="list-style-type: none"> • Descripción del sistema actual • Descripción de la problemática del sistema actual • Diagramas de los procesos actuales del sistema | Fuentes de información |
| Proyecto de ideas de negocio | <ul style="list-style-type: none"> • Condiciones del mercado • Aplicaciones competencias • Estudios de mercado | <ul style="list-style-type: none"> • Personas que trabajan en el mismo negocio • Posibles consumidores • Búsquedas en Internet • Docentes universitarios de carreras afines • Documentación especializada en el tema |

Tabla 2.1. Entradas por tipo de proyecto

1. Actividad 1: Evaluar los proveedores de requerimientos a entrevistar

El objetivo es verificar que las personas a entrevistar o las fuentes de información cumplen con los criterios establecidos en la Tabla 2.1.

En los proyectos de desarrollo a la medida, normalmente es preferible iniciar con los líderes funcionales, ya que ellos tienen una visión global de los procesos y comprenden el dominio del problema. Luego se continúa con los futuros usuarios, es decir, usuarios finales, porque ellos pueden aportar información más detallada sobre el entorno operacional de la organización. En la Tabla 2.2. se pueden observar los criterios para distinguir los proveedores de requerimientos.

| TIPO DE PROYECTO | NOMBRE | DESCRIPCIÓN |
|------------------------------------|--|---|
| Proyecto de desarrollo a la medida | Usuario Líder | Son las personas que comprenden el dominio del problema en donde será empleado el software desarrollado |
| | Usuario Final | Son las personas que usarán el sistema desarrollado. Serán quienes utilicen las interfaces y los manuales de usuario |
| Proyecto de ideas de negocio | <ul style="list-style-type: none"> • Personas que trabajen en el mismo negocio • Posibles consumidores • Documentación especializada • en el tema • Otras aplicaciones competencias | Implica factores como: <ul style="list-style-type: none"> • Condiciones del área para competir • Condiciones del entorno global. • El sector |

Tabla 2.2. Criterios para la distinción de proveedores de requerimientos

2. Actividad 2: Definir la estrategia de recolección de requerimientos

Esta actividad tiene como objetivo verificar que la estrategia de recolección de requerimientos sea la recomendada, según el tipo de proyecto a desarrollar. La estrategia ayuda tanto a desarrolladores como a usuarios a conocer la problemática del entorno a mejorar e identificar las necesidades

de la organización cliente, mientras que para los emprendedores, les permite conocer el entorno donde aspiran consolidar su negocio.

Para los proyectos de desarrollo a la medida pueden aplicar estrategias tales como las entrevistas y el desarrollo de aplicaciones de agregación (JAD) (Ver Tabla 2.3.).

| TIPO DE PROYECTO | NOMBRE | DESCRIPCIÓN |
|------------------------------------|-------------------------------------|--|
| Proyecto de desarrollo a la medida | Entrevistas | Es la técnica de elicitación más utilizada por ser una de las formas de comunicación más naturales entre las personas. |
| | Joint Application Development (JAD) | En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse participes del desarrollo. |
| Proyecto de ideas de negocio | Brainstorming | Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. |

Tabla 2.3. Estrategias para la recolección de requerimientos

3. Actividad 3: Planificar reuniones de elicitación de requerimientos

Esta actividad tiene como objetivo:

- Identificar los usuarios del proyecto.
- Conocer los procesos de la organización cliente.
- Identificar las actividades necesarias.

Después de definir la estrategia de recolección de requerimientos, se planificarán reuniones de elicitación de requerimientos.

En los proyectos de desarrollo de software a la medida, esta actividad es fundamental ya que en estas reuniones los desarrolladores conocen las

necesidades de la organización cliente e identifican quiénes serán los usuarios del nuevo sistema.

4. Actividad 4: Identificar los procesos del sistema

Esta actividad tiene por objetivos:

- Identificar entradas y salidas de los procesos.
- Identificar las actividades que se realizan en cada proceso del sistema.
- Identificar el personal que interviene en los procesos del sistema.
- Realizar diagramas de los procesos del sistema.

Una vez se han planificado las reuniones de elicitación y se conocen las necesidades de la organización cliente, es fundamental conocer también los procesos del sistema que necesitan ser mejorados. Asimismo estos procesos ayudan a proveer información más específica para el planteamiento de los requerimientos funcionales y no funcionales.

5. Actividad 5: Identificar requerimientos funcionales

Esta actividad tiene por objetivos:

- Identificar los requerimientos funcionales que el software deberá cumplir.
- Resolver posibles conflictos.

En ella se debe identificar o revisar si existen conflictos y qué debe hacer el sistema a desarrollar con la información registrada en las actividades 3 y 4. Se toma cada proceso identificado y se definen requerimientos por cada proceso.

6. Actividad 6: Identificar los requerimientos no funcionales

Esta actividad tiene por objetivo identificar los requerimientos no funcionales del software a desarrollar.

Algunos tipos de requisitos que se suelen incluir en esta sección son los siguientes:

- Los requisitos de comunicaciones del sistema, los cuales son de carácter técnico relativos a las comunicaciones que deberá soportar el software a desarrollar. Por ejemplo: el sistema deberá utilizar el protocolo TCP/IP para las comunicaciones con otros sistemas.
- Los requisitos de interfaz de usuario, que especifican las características que deberá tener el sistema en su comunicación con el usuario.
- Los requisitos de fiabilidad que deben establecer los factores requeridos para la fiabilidad del software en tiempo de explotación. La fiabilidad mide la probabilidad del sistema de producir una respuesta satisfactoria a las demandas del usuario. Por ejemplo: la tasa de fallos del sistema no podrá ser superior a 2 fallos por semana.
- Los requisitos de entorno de desarrollo, los cuales especifican si el sistema debe desarrollarse con un producto específico.
- Los requisitos de portabilidad, los cuales definen qué características deberá tener el software para que sea fácil utilizarlo en otra máquina o bajo otro sistema operativo.

Ver en la Tabla 2.4. la salida que tiene la fase de elicitación de requerimientos.

| NOMBRE | DESCRIPCIÓN | DESTINO |
|--|---|-------------------------|
| Definición de requerimientos del software. | <ul style="list-style-type: none"> • Introducción: debe contener una breve descripción de las características del nuevo software a desarrollar y/o idea de negocio a llevar a cabo. • Procesos: debe contener los procesos del sistema. • Requerimientos funcionales: se nombran los requisitos funcionales que se hayan identificado en dicho proceso. • Requerimientos no funcionales: se nombran los requisitos no funcionales que se hayan identificado en dicho proceso. | Fuentes de información. |

Tabla 2.4. Salida de la fase 1 para proyectos de desarrollo de software a la medida e idea de negocio.

2.5.2. Especificación

Esta fase, que tiene tres entradas (Ver Tabla 2.5.), presenta las siguientes actividades:

| NOMBRE | FUENTE |
|---|------------------------------------|
| Documento de definición de requerimientos. | Analista Fuentes de información |
| Documento de priorización de requerimientos | Analista Fuentes de información |
| Definición de la versión | Analista Fuentes de información |

Tabla 2.5. Entradas para la fase de especificación de requerimientos

1. Actividad 1: Describir los requerimientos

Esta actividad tiene como objetivo:

- Definir los actores que participan en cada proceso.
- Definir la secuencia de interacciones de los procesos.
- Definir las excepciones asociadas a los procesos.

En esta actividad se pueden revisar las estrategias de especificación de requerimientos con el fin de describir los requerimientos en cuanto a sus actores, escenarios y excepciones, advirtiendo que ninguna estrategia de especificación de requerimientos existente es excluyente en esta fase de la metodología.

2. Actividad 2: Evaluar los requerimientos

Esta actividad tiene por objetivo verificar que los casos de uso, historias de usuarios o prototipos cumplan con los criterios establecidos. La práctica consiste en examinar la especificación para asegurarse que todos los requerimientos de software han sido escritos sin ambigüedades, sin inconsistencias, sin omisiones y errores, han sido detectados y corregidos; que los productos de trabajo sean conformes a los estándares establecidos para el proceso, para el proyecto y para el producto. Esta práctica de evaluación de requerimientos es indispensable en la fase de especificación

de los mismos, ya que una vez el requerimiento cumpla o no con los criterios establecidos de evaluación, es aceptado o no; finalmente esto conduce a corregir inconsistencias y a tomar decisiones inmediatas sin tener que esperar a la fase final del proyecto. Generalmente no se realiza para todos los requerimientos sino para una muestra de ellos.

Ver en la Tabla 2.6. la salida que provee la fase de especificación de requerimientos.

| NOMBRE | DESCRIPCIÓN | DESTINO |
|---|--|---------------------|
| Documento de Especificación de Requerimientos (DER) | <ul style="list-style-type: none"> • Introducción • Participante del proyecto • Requerimientos funcionales definidos • Definición de actores • Estrategia de especificación de requerimientos • Requerimientos no funcionales definidos. | Grupo de desarrollo |

Tabla 2.6. Salida fase de especificación de requerimientos

2.5.3. Gestión

Esta fase, que tiene tres entradas indicadas en la Tabla 2.7., plantea las siguientes actividades:

| NOMBRE | FUENTE |
|---|-------------------|
| Documento de especificación de requerimientos. | Analista |
| Definición de requerimientos | Analista |
| Asignación de las actividades para el grupo de desarrollo | Jefe del proyecto |

Tabla 2.7. Entradas fase de gestión de requerimientos

1. Actividad 1: Priorizar los requerimientos

Esta actividad tiene como objetivo:

- Asignar categorías los requerimientos en función de su criticidad, importancia, complejidad y riesgo.
- Definir las versiones entregables del proyecto.

Esta práctica permite realizar una organización de la construcción o desarrollo de los requerimientos de acuerdo con las necesidades del cliente o la importancia que tienen los requerimientos para él.

Es importante realizar el proceso de priorización de requerimientos debido a que se obtendrán claros objetivos de desarrollo, se definirá la entrada y salida de cada una de las etapas del proyecto, se podrá hacer una organización y distribución de actividades para el desarrollo incremental que permitirá por tanto negociar el cómo se liberará el producto.

2. Actividad 2: Almacenar los requerimientos en un repositorio central

Esta práctica tiene por objetivo almacenar los requerimientos definidos en un repositorio central (herramienta informática), pues conservarlos en plantillas no es muy recomendable por la facilidad de pérdida, además de un proceso desordenado de desarrollo y otros factores negativos que pueden afectar el progreso de este proceso.

3. Actividad 3: Asignar al grupo de desarrollo los requerimientos del software

Esta actividad tiene como objetivo:

- Asignar un responsable al desarrollo de cada requerimiento.
- Realizar chequeos a las actividades designadas a los integrantes del grupo de desarrollo.
- Reasignar actividades a nuevos responsables cuando la situación lo considere necesario.

El jefe del proyecto debe determinar los requerimientos establecidos a cada uno de los integrantes del grupo de desarrollo para ejecutar el proyecto, además es necesario llevar el control del proceso de desarrollo de dichos requerimientos a través de chequeos que permitirán asegurar que tal desarrollo estará a la fecha de entrega establecida por el jefe de proyecto evitando retrasos en el proyecto.

4. Actividad 4: Administrar los cambios en los requerimientos

Esta actividad tiene como objetivo:

- Verificar que el cambio haya sido autorizado.
- Evaluar el impacto del cambio del requerimiento con los usuarios relevantes del software.
- Mantener historial de los cambios en los requerimientos.

Los cambios siempre tienen un precio, incluso un cambio rechazado consume recursos que han sido requeridos para evaluar y decidir el rechazo. Cuando se evalúa cada cambio está latente la necesidad de cumplir con los objetivos del negocio, la visión del producto y el alcance del proyecto.

Se debe asignar quién puede cambiar los requisitos, quién debe revisarlos o aprobarlos. Se debe garantizar el conocimiento de quiénes son los responsables de los cambios propuestos para controlar los cambios de una manera disciplinada.

5. Actividad 5: Administrar la trazabilidad de los requerimientos

Esta fase tiene por objetivo:

- Identificar la relación del requerimiento con otros elementos del sistema.
- Identificar la relación del requerimiento con otros procesos del sistema.
- Realizar seguimiento del estado de los requerimientos.

En esta práctica también es importante resaltar el estado en el que se encuentra el requerimiento, por lo tanto se propone un ciclo de vida del requerimiento representado por los posibles estados en el que él se puede encontrar. Esta fase tiene la salida mostrada en la Tabla 2.8.

| NOMBRE | DESCRIPCIÓN | DESTINO |
|---------------------------------|--|---------------------|
| Priorización de requerimientos. | Permite mostrar una organización para el desarrollo de los requerimientos de acuerdo a las necesidades del cliente o a la importancia que tienen los requerimientos para él. | Grupo de desarrollo |

Tabla 2.8. Salida fase de gestión de requerimientos

Control en la metodología de ingeniería de requerimientos del software

Esta etapa de la metodología tiene como propósito describir los pasos para asegurar que las actividades son ejecutadas conforme al proceso que ha sido establecido.

En las fases de elicitación y especificación de requerimientos, el proceso de control contempla las siguientes tareas:

- Realizar revisión de las próximas actividades y sus objetivos.
- Elaboración de actas de reuniones de elicitación y especificación.
- Realizar revisión de los requerimientos.
- Resolver actividades pendientes.

En la fase de gestión, el proceso de control contempla las siguientes tareas:

- Realizar revisión sobre los requerimientos del software, es decir, revisiones periódicas hechas por el jefe del proyecto para:
 - Identificar los integrantes de la tarea.
 - Informar a los integrantes de las revisiones a realizar.
 - Realizar revisiones de diseño de requerimientos.
 - Almacenar los resultados de la tarea.
 - Comunicar los resultados.
- Revisar las actividades:
 - Verificar que las tareas se estén llevando a cabo según lo planeado.
 - Tomar medidas si se está desviando de lo planeado.
 - Generar informes con los resultados.
 - Realizar auditorías del proceso.

2.6. EVALUACIÓN DE LAS METODOLOGÍAS

Para la evaluación de las metodologías se utilizó cinco metodologías descritas con las etapas comprendidas en cada una, esto es:

| METODOLOGÍA | No. | POND. | TOT. |
|---------------------------|-----|-------|------|
| MODELO TRADICIONAL | 10 | 0.914 | 9.14 |
| Extracción | | | |
| Análisis | | | |
| Especificación | | | |
| Validación | | | |

Tabla 2.9. Metodología del Modelo Tradicional

| METODOLOGÍA | No. | POND. | TOT. |
|--|-----|-------|------|
| MODELO EN ESPIRAL | 5 | 0.686 | 3.43 |
| Planificación/Extracción | | | |
| Especificación informal | | | |
| Análisis | | | |
| Requerimientos Aceptados | | | |
| Especificación | | | |
| Especificación de requerimientos preliminar | | | |
| Validación | | | |
| Especificación de requerimientos y reporte de validación | | | |

Tabla 2.10. Metodología del Modelo en Espiral

| METODOLOGÍA | No. | POND. | TOT. |
|--|------------|--------------|-------------|
| METODOLOGÍA ANCORA | 1 | 0.142 | 0.142 |
| Entendimiento del Dominio y Contexto de la aplicación | | | |
| Leer material | | | |
| Entrevistas informales con el cliente | | | |
| Observar usuarios | | | |
| Construir redes semánticas naturales del sistema | | | |
| Construir y aplicar encuestas de actitud | | | |
| Recolección y clasificación de requerimientos | | | |
| Propuesta Computacional | | | |
| Guión | | | |
| Manual preliminar de Operación (Prototipo rápido) | | | |
| Modelo de datos | | | |
| Lista de responsabilidades | | | |
| Lista de beneficios esperados | | | |
| Creación de la bitácora de desarrollo del sistema | | | |
| Función del sistema | | | |
| Forma de comprobación por parte del usuario | | | |
| Tiempo real de implantación | | | |
| Aplicación de puntos de función | | | |
| Establecer restricciones | | | |
| Refinar la propuesta computacional | | | |
| Reuso de requerimientos | | | |
| Abstraer y guardar los guiones de sistemas terminados | | | |
| Consultar y extraer elementos reutilizables | | | |
| Resolución de conflictos, priorización y validación de req. | | | |
| Preparación de una reunión de reflexión y diseño | | | |
| Documento de la situación actual | | | |
| Documento de la propuesta computacional | | | |
| Reunión de reflexión y diseño | | | |
| Cierre | | | |
| Ratificar cálculo de puntos de función | | | |
| Aclarar puntos pendientes | | | |
| Conexión con el modelo de desarrollo escogido | | | |

Tabla 2.11. Metodología Ancora

| METODOLOGIA | No. | POND. | TOT. |
|---|------------|--------------|-------------|
| METODOLOGÍA DORCU | 1 | 0.142 | 0.142 |
| Elicitación de requerimientos | | | |
| Formar equipo multidisciplinario | | | |
| Buscar hechos | | | |
| Recolectar y clasificar requerimiento | | | |
| Evaluar y racionalizar | | | |
| Dar prioridad | | | |
| Integrar y validar | | | |
| Documentar la etapa | | | |
| Análisis de requerimientos | | | |
| Reducir ambigüedades de los requerimientos | | | |
| Traducir a lenguaje técnico los requerimientos | | | |
| Plantear un modelo lógico | | | |
| Documentar la etapa | | | |
| Validación y Certificación de requerimientos | | | |
| Seleccionar las fuentes de información a partir de las cuales validar el documento de especificación | | | |
| Elegir o diseñar el modelo de documento acorde al grado de detalle requerido y al lector final | | | |
| Elegir la herramienta de documentación que mejor se aplica al modelo seleccionado | | | |
| Documentar respetando los estándares vigentes a la fecha de realización del documento de requerimientos | | | |
| Validar | | | |
| Certificar | | | |

Tabla 2.12. Metodología Dorcu

| METODOLOGÍA | No. | POND. | TOT. |
|---|------------|--------------|-------------|
| METODOLOGÍA AGIL | 8 | 0.886 | 7.088 |
| Elicitación de requerimientos | | | |
| Evaluar los proveedores de requerimientos a entrevistar | | | |
| Definir la estrategia de recolección de requerimientos | | | |
| Planificar reuniones de elicitación de requerimientos | | | |
| Identificar los procesos del sistema | | | |
| Identificar requerimientos funcionales | | | |
| Identificar requerimientos no funcionales | | | |
| Especificación de requerimientos | | | |
| Describir los requerimientos | | | |
| Evaluar los requerimientos | | | |
| Gestión de requerimientos | | | |
| Priorizar los requerimientos | | | |
| Almacenar los requerimientos en un repositorio central | | | |
| Asignar al grupo de desarrollo los requerimientos | | | |
| Administrar los cambios en los requerimientos | | | |
| Administrar la trazabilidad de los requerimientos | | | |

Tabla 2.13. Metodología Ágil

En este cuestionario las personas que tenían que contestar, lo podían hacer por etapas o sub etapas, dependiendo de la utilización de cada una, en el desarrollo de los proyectos software.

Se revisó el proceso de ingeniería de requerimientos en algunas empresas de desarrollo de software tanto públicas como privadas, así como con alumnos que han realizado productos software como temas de tesis, dándose a conocer la necesidad que ellas tienen de implementar metodologías que soporten estas actividades. La gran mayoría asegura que no emplean metodologías de ingeniería de requerimientos, pero realmente realizan ciertas actividades de la forma correcta, o en otros casos no las realizan, lo que ocasiona problemáticas en cuanto a criterios para la aceptación de proveedores de requerimientos, criterios para la aceptación de requerimientos y ausencia de administración de la trazabilidad y de los cambios de los requerimientos.

Al realizar la evaluación se manifiesta que:

- No establecen criterios para la aceptación de proveedores de requerimientos, lo que puede traer falencias para el desarrollo del proyecto por cuanto las fuentes de los requerimientos no tienen dominio de la problemática del sistema ni de sus procesos, así como obtener requerimientos mal interpretados inconsistentes e incluso omitir requerimientos que pueden ser necesarios para la solución.
- No se establece criterios para la aceptación de requerimientos, factor que puede ocasionar la presencia de errores en el desarrollo de los requerimientos que redunden en retrasos en los cronogramas de trabajo, trabajo repetitivo o hasta rechazo del cliente.
- No realizan la administración de la trazabilidad de los requerimientos (factor esencial en la fase de gestión de requerimientos), y por ello las dificultades que pueden presentar en la ubicación de fallas en los requerimientos desde la documentación hasta el producto final. Tanto el historial como la administración de cambios son también factores ignorados. Esta situación conduce al desconocimiento del impacto que los cambios puedan generar, las razones de ellos, cómo se hicieron, cuándo fueron hechos y quién los autorizó, entre otros. Esta información histórica de los cambios es útil para la toma futura de decisiones.

A partir de este diagnóstico se creó la necesidad de disponer de una metodología del proceso de ingeniería de requerimientos, que representará, una facilidad de implementación, uso de pocos recursos y un apoyo en la ejecución de sus procesos de ingeniería de software, a través de una matriz de evaluación de metodologías.

2.7. MATRIZ DE EVALUACIÓN DE LA METODOLOGÍA PROPUESTA

Una vez descrito todo lo referente al proceso de Ingeniería de Requerimientos y los distintos modelos de metodologías existentes, permitió tener una idea de

cuales son algunas de las variables que se encuentran presentes en cada una de ellas y que permiten una adecuación a la propuesta.

Paralelo a esto se obtuvo información de distintas páginas web y material bibliográfico sobre las variables que se deben tomar en cuenta para la creación o selección de una metodología para el desarrollo de software y se ubicó un reportaje que engloba, con gran certeza las variables mencionadas y con las cuales se hizo una analogía para la creación de una metodología para la Ingeniería de Requerimientos.

| No. | FACTORES CLAVE DE ÉXITO | PUNTUACIÓN | | | | |
|-----|---|------------|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | La metodología debe adaptarse a los objetivos | | | | X | |
| 2 | La metodología debe cubrir el ciclo del proceso de Ingeniería de Requerimientos | | | | | X |
| 3 | La metodología debe incluir la realización de validaciones | | | | | X |
| 4 | La metodología debe ser la base de una comunicación efectiva | | | | X | |
| 5 | La metodología debe funcionar en un entorno dinámico orientado al usuario | | | | X | |
| 6 | La metodología debe especificar claramente los responsables de resultados | | | | X | |
| 7 | La metodología debe poder emplearse en un entorno amplio de proyectos software | | | | | X |
| 8 | La metodología debe estar soportada por herramientas case | | | | X | |
| 9 | La metodología debe soportar la eventual evolución de los requerimientos | | | | | X |
| 10 | La metodología debe soportarse en el uso de estándares (IEEE) | | | | | X |
| 11 | La metodología debe contener actividades conducentes a mejorar el proceso de ingeniería de requerimientos | | | | X | |

Tabla 2.14. Matriz de la evaluación de la metodología propuesta

1. **La metodología debe ajustarse a los objetivos:** Cada aproximación al desarrollo de software está basada en unos objetivos. Por ello la metodología debe recoger el aspecto filosófico de la aproximación deseada, es decir, que los objetivos generales del desarrollo deben estar implementados en la metodología.
2. **La metodología debe cubrir el ciclo del proceso de Ingeniería de Requerimientos:** Para ello la metodología ha de realizar las etapas de:
 - Elicitación de requerimientos
 - Análisis de requerimientos
 - Validación de Requerimientos
3. **La metodología debe incluir la realización de validación:** La metodología debe detectar y corregir errores cuanto antes. Uno de los problemas más frecuentes y costosos es la detección y corrección de problemas en la obtención de los requerimientos. Cuanto más tarde sea detectado el error más caro será corregirlo.
4. **La metodología debe ser la base de una comunicación efectiva:** Debe ser posible gestionar a los informáticos, y éstos deben ser capaces de trabajar conjuntamente con analistas, programadores, usuarios, es decir, con todos los stakeholders a través de una comunicación efectiva y con pasos bien definidos para realizar progresos visibles durante las actividades de Ingeniería de Requerimientos.
5. **La metodología debe funcionar en un entorno dinámico orientado al usuario:** A lo largo de todo el proceso se debe producir una transferencia de conocimientos hacia el usuario. La clave del éxito es que todas las partes implicadas han de intercambiar información libremente. La participación del usuario es vital importancia debido a que sus necesidades evolucionan constantemente. Para involucrar al usuario es aconsejable el empleo de técnicas lo más sencillas posibles.

6. **La metodología debe especificar claramente los responsables de resultados:** Debe especificar claramente quienes son los participantes de cada tarea a desarrollar, debe detallar de una manera clara los resultados de los que serán responsables.
7. **La metodología debe poder emplearse en un entorno amplio de proyectos software:** Una empresa deberá adoptar una metodología que sea útil para un gran número de sistemas que vaya a construir. Por esta razón no es práctico adoptar varias metodologías en una misma empresa. Tamaño, vida. Las metodologías deberán ser capaces de abordar sistemas de distintos tamaños y rangos de vida. Complejidad. La metodología debe servir para sistemas de distinta complejidad, es decir puede abarcar un departamento, varios de departamentos o varias empresas. Entorno. La metodología debe servir con independencia de la tecnología disponible en la empresa.
8. **La metodología debe estar soportada por herramientas CASE:** La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general. El uso de estas herramientas reduce el número de personas requeridas y la sobrecarga de comunicación, además de ayudar a producir especificaciones y diseños con menos errores, más fáciles de probar, modificar y usar.
9. **La metodología debe soportar la eventual evolución del sistema:** Normalmente durante su tiempo de vida los sistemas tienen muchas versiones y debe facilitar su mantenimiento.
10. **La metodología debe soportarse en el uso de estándares:** La metodología deberá soportarse en los estándares IEEE 830, 1233 y 1362

11. La metodología debe contener actividades conducentes a mejorar el proceso de ingeniería de requerimientos: Para mejorar el proceso es básico disponer de datos numéricos que evidencian la efectividad de la aplicación del proceso con respecto a cualquier producto software resultante del proceso. Para disponer de estos datos, la metodología debe contener un conjunto de mediciones de proceso para identificar la calidad y costo asociado a cada etapa del proceso. Sería ideal el uso de herramientas CASE.

CAPÍTULO III

ESTÁNDARES IEEE PARA REQUERIMIENTOS

3.1. INTRODUCCIÓN A LOS ESTÁNDARES IEEE

IEEE es el Instituto de Ingenieros en electricidad y electrónica (Institute of Electrical and Electronics Engineers).

Su misión es preservar, investigar y promover la información de las tecnologías eléctricas y electrónicas.

Surgió en 1963 con la fusión del AIEE (Instituto Americano de Ingenieros Eléctricos) y el Instituto de Ingenieros de Radio (IRE).

La IEEE Computer Society (www.computer.org) es una sociedad integrada en IEEE, formada en la actualidad por más de 100.000 miembros en todo el mundo. Su finalidad es avanzar en la teoría, práctica y aplicación de las tecnologías de la información. Realiza conferencias, publicaciones, cursos de formación, y desarrolla estándares.

IEEE ha desarrollado estándares para todas las áreas de Ingeniería del Software. Algunos de ellos, correspondientes a las principales áreas específicas de la Ingeniería del Software son:

- IEEE Std. 830 Prácticas recomendadas para las especificaciones de software.
- IEEE Std. 1063 Estándar para la documentación de usuario de software.
- IEEE Std. 1012 Estándar para la verificación y validación de software.
- IEEE Std. 1219 Estándar para el mantenimiento del software.
- IEEE Std. 1233 Estándar para la especificación de requerimientos del sistema.

- IEEE Std. 1362 Guía para la especificación del documento de requerimientos “ConOps”.

3.2. ESTÁNDAR IEEE 830-1998: ESPECIFICACIONES DE LOS REQUERIMIENTOS DEL SOFTWARE

Este documento tiene como fin el de marcar las pautas generales y las especificaciones que deberá seguir la aplicación a desarrollar, con el objetivo final de suplir las necesidades que el cliente ha planteado en las distintas reuniones realizadas.

Este documento está dirigido a los desarrolladores encargados de la codificación de la aplicación con el objetivo de que realicen las funciones de desarrollo a partir de las pautas marcadas en él y en los documentos que se realizan posteriormente.

Este documento servirá de canal de comunicación entre las distintas partes implicadas en el desarrollo de la aplicación y deberán tomar parte en su confección los miembros de cada una de las partes involucradas. Esta especificación está sujeta a revisiones, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas revisiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

3.2.1. Esquema de la Especificación de Requerimientos del Software definida en el IEEE 830-1998

La siguiente figura muestra la estructura de la Especificación de Requerimientos del Software propuesta por el estándar IEEE 830 [IEEE, 1998].

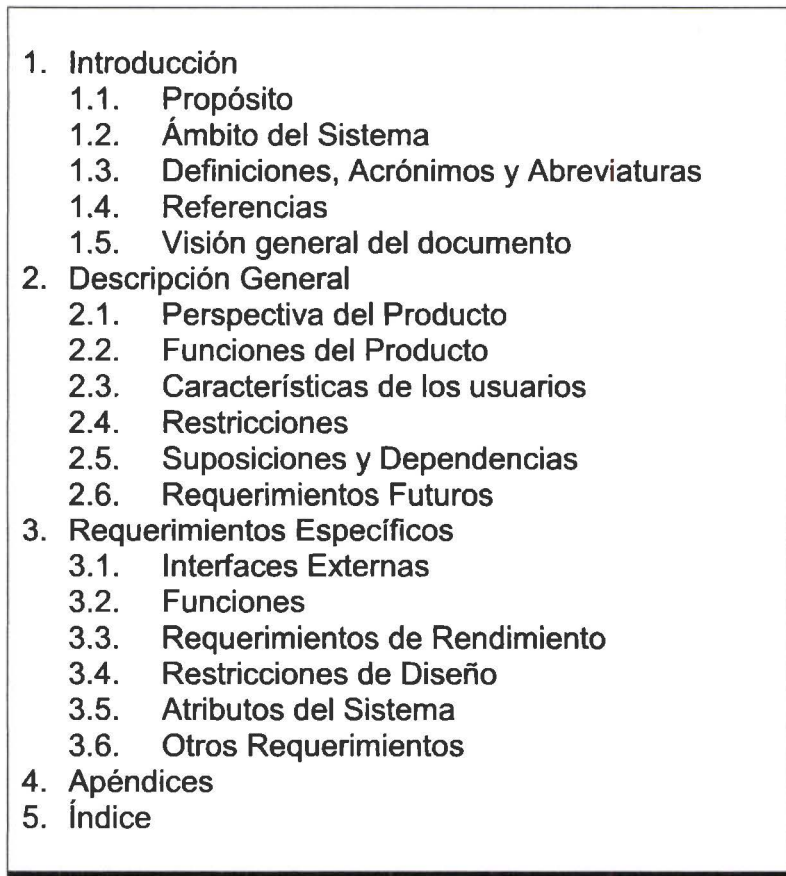


Fig. 3.1. Estructura de una ERS

A continuación se describirá brevemente cada uno de los apartados que se definen en el estándar estudiado.

1. Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requerimientos Software. Consta de varias subsecciones, las cuales son propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

1.1. Propósito

Se definirá el propósito del documento ERS y se especificará a quién va dirigido el documento.

1.2. Ámbito del Sistema

En esta subsección se pondrá nombre al futuro sistema, se explicará lo que el sistema hará y lo que no hará, se describirán los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema y se mantendrán referencias a los documentos de nivel superior que puedan existir.

1.3. Definiciones, Acrónimos y Abreviaturas.

Se definirán aquí todos los términos, acrónimos y abreviaturas utilizadas en el desarrollo de la ERS.

1.4. Referencias

Se presentará una lista completa de todos los documentos referenciados en la ERS.

1.5. Visión General del Producto

Esta subsección describirá brevemente los contenidos y la organización del resto de la ERS.

2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requerimientos. En esta sección no se describen los requerimientos, sino su contexto. Los detalles de los requerimientos se describen en la sección 3, detallándolos y haciendo más fácil su comprensión.

Normalmente podemos encontrar las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, suposiciones y futuros requerimientos.

2.1. Perspectiva del Producto

Esta subsección debe relacionar el futuro sistema con otros productos. Así pues, podríamos dividir ésta en pequeñas subsecciones indicando cada uno de los puntos a tener en cuenta:

- 2.1.1. Indicar si es un producto independiente o parte de un sistema mayor.
- 2.1.2. Interfaces de sistema
- 2.1.3. Interfaces de usuario
 - 2.1.3.1. Características lógicas del interfaz
 - 2.1.3.2. Cuestiones de optimización del interfaz de usuario
- 2.1.4. Interfaces hardware
- 2.1.5. Interfaces software
 - 2.1.5.1. Descripción del producto software utilizado
 - 2.1.5.2. Propósito del interfaz
 - 2.1.5.3. Definición del interfaz: contenido y formato
- 2.1.6. Interfaces de comunicaciones
- 2.1.7. Limitaciones de memoria
- 2.1.8. Operaciones
 - 2.1.8.1. Modos de operación de los distintos grupos de usuarios
 - 2.1.8.2. Periodos de operaciones interactivas y automáticas
 - 2.1.8.3. Funciones respaldo del procesamiento de datos
 - 2.1.8.4. Operaciones de backup y recuperación
- 2.1.9. Requerimientos para adaptarse a la ubicación
 - 2.1.9.1. Indicar cualquier dato o secuencia de inicialización específico de cualquier lugar, modo de operación.
 - 2.1.9.2. Características que deben ser modificadas para una instalación en particular.

2.2. Funciones del Producto

Esta subsección debe proporcionar un resumen de las funciones principales que el software debe llevar a cabo. Las funciones deben

estar organizadas de manera que el cliente o cualquier otra persona lo entiendan perfectamente. Para ello se pueden utilizar métodos textuales o gráficos, siempre que dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.

En la metodología estructurada se podrían utilizar los DFDs y en una metodología orientada a objetos, el funcionamiento y las relaciones del futuro sistema se modelarían a través de los Casos de Uso. En ellos se representa lo que el usuario ve del sistema, así pues facilitará en gran medida su comprensión, siempre y cuando en los diagramas se eviten las ambigüedades.

2.3. Características de los usuarios

Se indica aquí el tipo de usuario al que se dirige la aplicación, así como su experiencia técnica, nivel de conocimientos, etc.

2.4. Restricciones

Se debe indicar aquí cualquier tipo de limitación como pueden ser políticas de la empresa, limitaciones hardware, seguridad, protocolos de comunicación, interfaces con otras aplicaciones, estándares de la empresa en cuanto a interfaces, etc. Serán las limitaciones que se imponen sobre los desarrolladores del producto.

2.5. Suposiciones y Dependencias

En este apartado aparecerá cualquier factor, que si cambia puede afectar a los requerimientos. No son restricciones de diseño, por ejemplo, asumir que un determinado sistema operativo estará disponible, presuponer una cierta organización de las unidades de la empresa. Si cambian ciertos detalles puede ser necesario revisar los requerimientos.

2.6. Requerimientos Futuros

Se indican aquí posibles mejoras del sistema en el futuro. Estas mejoras deben estudiarse y analizarse una vez concluido y puesto en marcha el sistema. Son modificaciones a realizar en un futuro incierto.

3. Requerimientos Específicos

Esta sección de la especificación de requerimientos software contiene todos los requerimientos hasta un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que satisfaga dichos requerimientos, y que permita diseñar las pruebas que ratifiquen que el sistema cumple con las necesidades requeridas.

Los requerimientos que se aquí se indiquen deben describir comportamientos externos del sistema, observables por el usuario así como por parte de los operadores y otros sistemas.

Puesto que deben indicarse todos los requerimientos, esta sección es la más larga de la ERS y debe cumplir los principios descritos en los primeros apartados de este informe.

Estos principios son la corrección, no ambigüedad, completitud, consistencia, clasificación, verificabilidad, modificabilidad, explorabilidad y facilidad de mantenimiento.

Asimismo, éste documento debe ser perfectamente legible por el cliente y por personas de muy distinta formación. Otra de las cuestiones a tener en cuenta en esta sección es la identificación de cada uno de los requerimientos mediante algún código o sistema de numeración.

3.1. Interfaces Externas

En esta subsección se definirán los requerimientos que afecten a la interfaz de usuario e interfaz con otros sistemas (hardware y software), así como a interfaces de comunicaciones.

3.2. Funciones

En esta subsección se deben especificar todas aquellas acciones o funciones que deberá llevar a cabo el sistema a desarrollar. Las acciones que se indican como "el sistema deberá..." son las que deben incluirse en este apartado.

La estructuración de las funciones a desarrollar por el nuevo sistema no está del todo clara. Se debe tener en cuenta que en el estándar de IEEE 830 de 1983 se establecía que las funciones se deberían expresar como una jerarquía funcional, puesto que es la que mejor se adaptaba a los DFDs que proponía el análisis estructurado. Con la evolución de la programación y los nuevos métodos de análisis se puede observar como esta estructura no se adapta, por tanto es necesaria la modificación de los estándares.

El estándar IEEE 830, en sus últimas versiones, permite la organización de esta subsección de múltiples formas y simplemente sugiere alguna manera para hacerlo, dejando la oportunidad de utilizar cualquier otra justificando suficientemente la utilización de ésta.

Alguna de las formas sugeridas por el estándar son:

- Por tipo de usuario: Distintos usuarios poseen distintos requerimientos. Para cada clase de usuario que exista en la organización, se especifican los requerimientos funcionales que le afecten o tengan mayor relación con sus tareas.

- Por objetos: Los objetos son entidades del mundo real que son reflejadas en el sistema. Por tanto, para cada objeto se detallan sus atributos y sus funciones. Los objetos se pueden agrupar en clases. A pesar de realizar el análisis con objetos no obliga a que el diseño del sistema siga el paradigma Orientado a Objetos, aunque lo facilita en gran medida.
- Por objetivos: un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o subobjetivo requerido al sistema, se detallarán las funciones que permitan llevarlo a cabo.
- Por jerarquía funcional: La funcionalidad del sistema se especifica como una jerarquía de funciones que comparten entradas, salidas o datos del propio sistema. Para cada función y subfunción del sistema se detallará la entrada, el proceso en el que interviene y la salida. Normalmente este tipo de análisis implica que el diseño siga el paradigma de diseño estructurado. Por lo general éste sistema se utiliza cuando ninguno de los anteriores se puede aplicar.

Como se puede apreciar, el estándar propone una serie de plantillas según el tipo de sistema con el que nos enfrentemos. Pero en muchas ocasiones la elección se realiza por eliminación, o lo que es lo mismo, se escoge aquel que mejor se adapta a lo que se busca.

3.3. Requerimientos de Rendimiento

En esta subsección se incluyen los requerimientos relacionados con la carga que se espera que tenga que soportar el sistema (número de usuarios simultáneos, número de terminales...). Asimismo, se pueden incluir los requerimientos que afecten a la información que se vaya a guardar en la base de datos (cantidad de registros en una base de datos, frecuencia de uso...)

3.4. Restricciones de Diseño

Se incluyen aquí todas las restricciones que afecten al diseño de la aplicación, como pueden ser estándares internos de la organización, limitaciones hardware, etc.

3.5. Atributos del Sistema

Se detallarán atributos como la fiabilidad, mantenibilidad, seguridad, mecanismos de acceso restringido (password), usuarios autorizados a realizar ciertas tareas críticas...

3.6. Otros requerimientos

Aquellos requerimientos que no se hayan podido incluir en ninguna de las secciones anteriormente especificadas.

4. Apéndices

Se incluirá aquí cualquier tipo de información relacionada con la ERS, pero que no forme parte de la misma. Por ejemplo, se incluirían los resultados del análisis de costos, restricciones especiales acerca del lenguaje de programación...

5. Índice

Se proporciona un índice para poder tener un acceso rápido a la documentación presentada en la ERS.

3.3. ESTÁNDAR IEEE 1233-1998: ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA

El objetivo de la especificación es definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema que se desea construir. Esta documentación está sujeta a revisiones por el grupo de usuarios que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación por parte de la dirección y del grupo de usuarios. Una

vez aprobado, servirá de base al equipo para la construcción del nuevo sistema.

Esta guía nos da la pauta para el desarrollo de un conjunto de requerimientos que satisfarán una necesidad específica. En esta guía, a ese conjunto de requerimientos se le denomina Especificación de Requerimientos del Sistema (System Requirements Specification, SyRS). El desarrollo de una SyRS incluye la identificación, organización, presentación y modificación de los requerimientos.

Esta guía trata las condiciones necesarias para incorporar conceptos operacionales, restricciones de diseño, y requerimientos de la configuración del diseño en la especificación. Además, trata las características y cualidades necesarias de los requerimientos individuales y del conjunto de todos los requerimientos.

3.3.1. Esquema de la Especificación de Requerimientos del Sistema

4. (Cláusula 4) Una SyRS tradicionalmente ha sido vista como un documento que comunica los requerimientos del cliente a la comunidad técnica que especificarán y construirán el sistema. La colección de requerimientos que constituyen la especificación y su representación actúan como el puente entre los dos grupos y debe ser entendible tanto por el cliente como por la comunidad técnica. Una de las tareas más difíciles en la creación de un sistema, es aquella de comunicar a todos los subgrupos, especialmente en un solo documento. Este tipo de comunicación usualmente requiere diferentes formalismos y lenguajes.

4.1. Definición

La SyRS presenta los resultados de la definición de necesidades, los conceptos de operación y las tareas de análisis de sistema. Como tal, es una descripción de lo que los clientes del sistema esperan de

éste, el ambiente del sistema, el perfil de uso del sistema, sus parámetros de desempeño, y la calidad y efectividad deseados. Esto es, presenta las conclusiones del proceso de desarrollo de la SyRS tal como se describe en la cláusula 5.

Esta guía sugiere una distinción entre este conjunto estructurado de información y la manera en la cual es presentada a sus varias audiencias. La presentación de la SyRS debe tomar una forma que sea apropiada para su uso explícito. Esta forma puede ser un documento, modelos, prototipos, una representación en papel o cualquier combinación. De cualquier manera, se debe tener cuidado de asegurar que cada una de estas representaciones sea rastreable hacia una fuente común.

Esta guía hace una clara distinción entre los requerimientos del sistema (qué debe hacer el sistema) contenidos en la SyRS y los requerimientos del proceso (cómo se construye el sistema) que deben estar contenidos en los documentos del contrato como sería el Contrato de Trabajo.

4.2. Propiedades

El conjunto de requerimientos debería tener las siguientes propiedades:

- a) Conjunto Único. Cada requerimiento debe ser declarado sólo una vez.
- b) Normalizado. Los requerimientos no se deben traslapar (Por ejemplo, no se deben referir a otros requerimientos ni a las capacidades de otros requerimientos).
- c) Conjunto interdependiente. Se deben definir explícitamente las relaciones entre los requerimientos individuales para mostrar cómo los requerimientos están relacionados para formar el sistema completo.

- d) **Completo.** Una SyRS debe incluir todos los requerimientos dados por el cliente, así como aquellos requerimientos necesarios para la definición del sistema.
- e) **Consistente.** El contenido de una SyRS debe ser consistente y sin contradicciones en el nivel de detalle, estilo de la declaración de los requerimientos y en la presentación del material.
- f) **Acotado.** Los límites, alcance y el contexto de los requerimientos del sistema deben ser identificados.
- g) **Modificable.** La SyRS debería ser modificable. Requerimientos claros y sin traslapamientos contribuyen a lograrlo.
- h) **Configurable.** Las versiones deberían ser mantenidas a través del tiempo y a través de las instancias de la SyRS.
- i) **Granular.** Éste debe ser el nivel de abstracción para el sistema que está siendo definido

4.3. Propósito

El propósito de la SyRS es proveer una descripción tipo caja negra de lo que el sistema debe hacer, en términos de las interacciones del sistema o las interfaces con su ambiente externo. La SyRS debe describir completamente todas las entradas, salidas, y las relaciones requeridas entre salidas y entradas. La SyRS organiza y comunica los requerimientos del cliente y la comunidad técnica.

4.3.1. Organizando los requerimientos

El propósito de la SyRS puede ser logrado mas fácilmente organizando los requerimientos en categorías conceptuales. En la práctica, difícil identificar y separar los requerimientos de otros aspectos de la percepción del cliente del sistema que a menudo están incluidos en documentos que se definen como "requerimientos". Frecuentemente, los procedimientos tradicionales del usuario y las suposiciones de la comunidad técnica acerca de la implementación nublan la declaración fundamental de la necesidad.

Mientras se organizan las declaraciones no estructuradas del usuario en un conjunto estructurado de requerimientos, el analista debe identificar los requerimientos técnicos sin intentar enunciar las técnicas de implementación. El distraerse en temas de implementación antes de lograr entender claramente los requerimientos nos puede llevar a una declaración inadecuada de requerimientos así como a una implementación con fallas. Discernir entre requerimientos técnicos e implementaciones técnicas es un constante reto para los analistas.

La descripción del sistema debe ser establecida en términos operacionales y logísticos. Los temas a tratar incluyen capacidades operacionales, características físicas, parámetros de rendimiento y valores esperados, interfaces e interacciones con su ambiente, documentación requerida, requerimientos de seguridad y confianza; así como requerimientos personales que se esperan del sistema.

Estos requerimientos deberían ser comunicados de manera estructurada para asegurar que el cliente y la comunidad técnica puedan hacer lo siguiente:

- a) Identificar requerimientos derivados de otros requerimientos.
- b) Organizar requerimientos en diferentes niveles de detalle en un nivel apropiado.
- c) Verificar que el conjunto de requerimientos esté completo.
- d) Identificar inconsistencias entre los requerimientos.
- e) Claramente identificar las capacidades, condiciones, y restricciones de cada requerimiento.
- f) Desarrollar un entendimiento común con el cliente del propósito y los objetivos del conjunto de requerimientos.
- g) Identificar requerimientos que completarán la SyRS.

Es importante que la estructura que sea dada al conjunto de requerimientos por los analistas, y que las representaciones de la SyRS comuniquen los requerimientos de manera estructurada. La cláusula 6 nos provee una guía para definir explícitamente los requerimientos.

4.3.2. Comunicación con dos audiencias

La SyRS tiene dos audiencias principales y esencialmente sirve para documentar un acuerdo entre el cliente y la comunidad técnica.

4.3.2.1. Cliente

Cliente es un término colectivo que puede incluir al cliente del sistema propuesto, la persona que aceptará y firmará la entrega, y los encargados quienes serán los responsables de vigilar la implementación, operación y mantenimiento del sistema.

Todos los clientes tienen intereses e inquietudes que deben ser resueltos en la SyRS. Además, algunos clientes podrían no entender el proceso de cómo establecer los requerimientos o el proceso de creación del sistema. Aunque sean competentes en sus áreas de responsabilidad y en la aplicación para el cual el sistema será definido, generalmente no están familiarizados con el vocabulario ni con las técnicas de representación normalmente usadas para especificar requerimientos. Como uno de los objetivos principales del análisis de requerimientos del sistema es asegurar que la SyRS sea entendida, será necesario darle al cliente una representación de la SyRS completa, concisa y clara, en un lenguaje que el cliente entienda.

4.3.2.2. Comunidad Técnica

La SyRS debería también comunicar los requerimientos del cliente a la comunidad técnica. La comunidad técnica incluye analistas, estimadores, diseñadores, desarrolladores, auditores de aseguramiento de calidad, ingenieros, personal de integración, de pruebas, de mantenimiento y de manufactura. Para esta audiencia la representación de la SyRS debe ser técnicamente precisa y presentada en un formalismo adecuado para que ellos puedan diseñar, construir y probar el sistema requerido.

4.4. Uso recomendado

Los usos recomendados de la SyRS pueden variar conforme el ciclo de desarrollo progresa, son los siguientes:

- a) Durante el diseño del sistema, los requerimientos son asignados a subsistemas, hardware, software, operaciones y otros componentes importantes del sistema.
- b) La SyRS es utilizada para construir el sistema resultante. También es usada para escribir apropiadamente los planes de verificación del sistema. Si el sistema contiene hardware y software, entonces el plan de pruebas de hardware y el plan de pruebas de software son generados para los requerimientos del sistema.
- c) Durante la fase de implementación, los procedimientos de prueba serán definidos a partir de la SyRS.
- d) Durante el proceso de validación, los procedimientos de validación basados en la SyRS serán usados para darle al cliente las bases para aceptar el sistema.

Si se van a hacer cambios a la línea base de la SyRS, éstos deberán ser controlados a través de un proceso formal de gestión de cambios. Este proceso debe incluir una negociación apropiada entre

las partes afectadas por el cambio y debe activar procedimientos de control riesgos cuando sea pertinente (por ejemplo: calendario, costos).

4.5. Beneficios

Una SyRS apropiadamente escrita beneficia todas las subsecuentes fases del ciclo de vida de varias maneras. La SyRS documenta el conjunto completo de capacidades del sistema y nos provee de los siguientes beneficios:

- a) Asegura al cliente que la comunidad técnica entiende sus necesidades y que están respondiendo a ellas.
- b) Una oportunidad temprana para una mutua retroalimentación entre el cliente y la comunidad técnica.
- c) Un método para que el cliente y la comunidad técnica puedan identificar problemas y malentendidos mientras los costos de corregirlos son relativamente baratos.
- d) Una base para la calificación y calidad del sistema para establecer que el sistema cumple con las necesidades del cliente.
- e) Protección para la comunidad técnica, proporcionando una línea base para las capacidades del sistema y una base que determine cuando la construcción del sistema está completa.
- f) Soporte para el desarrollador en la planificación, diseño y desarrollo del programa.
- g) Ayuda en la evaluación de los efectos de los inevitables cambios en los requerimientos.
- h) Incrementa la protección en contra de los malentendidos entre el cliente y la comunidad técnica durante el progreso del desarrollo.

4.6. Dinamismo de los requerimientos del sistema

Los requerimientos son raramente estáticos. Aunque es deseable congelar un conjunto de requerimientos permanentemente, esto es

raramente posible. Los requerimientos que son propensos a evolucionar deben ser identificados y comunicados tanto a la comunidad técnica como al cliente.

Un subconjunto medular de requerimientos puede ser congelado al inicio. El impacto al proponer nuevos requerimientos debe ser evaluado para asegurar que la propuesta inicial de la línea base de requerimientos se mantenga, o que los cambios a dicha propuesta sean entendidos y aceptados por el cliente.

5. Descripción del proceso de desarrollo de la SyRS

Esta cláusula proporciona una descripción de los pasos en el proceso de desarrollo de la SyRS. El proceso de desarrollo de los requerimientos del sistema, en general, se relaciona con 3 agentes externos (cliente, ambiente y comunidad técnica) Cada uno de estos agentes es descrito en la figura siguiente:

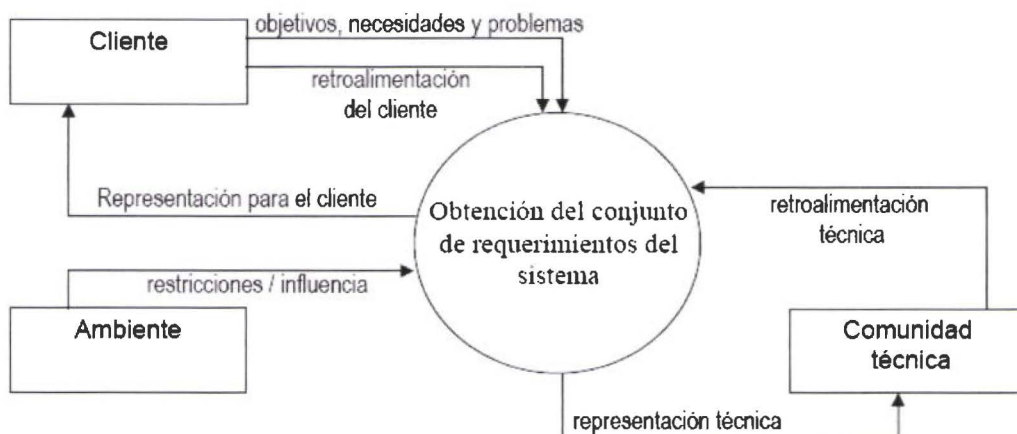


Fig. 3.2. Contexto de desarrollo de SyRS

5.1. Cliente

Los clientes son el elemento clave del contexto de la SyRS. Ellos son los principales conductores del sistema proporcionando sus objetivos, necesidades o problemas para el proceso de generación de la SyRS.

5.1.1. Requerimientos sin estructura

Previo al proceso de SyRS, el cliente tuvo la idea de un sistema, para mejorar un proceso o para resolver un problema. En este punto, cualquier concepto inicial para un sistema puede ser impreciso y sin estructura. Los requerimientos usualmente serán entremezclados con las ideas y sugerencias de un diseño potencial. Estos requerimientos sin estructura son muchas veces expresados en documentos iniciales parecidos a los siguientes:

- a) Concepto de operaciones. Este tipo de documento se enfoca en las metas, objetivos y capacidades deseadas del sistema potencial sin indicar como se implementará el sistema para lograr su objetivo.
- b) Concepto de sistema. Este tipo de documento incluye información sobre el concepto de operaciones, pero además va a incluir un diseño de la interfaz preliminar para el sistema y otros requerimientos explícitos.
- c) Especificaciones de mercadotecnia. Este tipo de documento incluye una lista de características para un nuevo sistema o sistemas e identificarán el alcance de sus características y su prioridad para proporcionar una ventaja en el mercado. Además, incluye un contexto definiendo cómo el nuevo sistema debe interactuar con los sistemas existentes. Un análisis de costo/beneficio y un calendario de entrega podrían ser provistos.
- d) Petición de la oferta (Request for proposal - RFP). En algunas situaciones un RFP será preparado. Ésta puede incluir uno o más de los documentos anteriores. El propósito será solicitar ofertas para construir el sistema, o para simplemente requerir asistencia para generar documentos iniciales del sistema.
- e) Interfaces externas del sistema. La definición de todas las interfaces externas del sistema, literalmente o por referencia, es una de las actividades más importantes a lograr antes de la generación de la SyRS. Una definición aprobada del universo

externo del sistema limita razonablemente o restringe lo que el sistema debe hacer internamente. Todos los elementos conocidos de cada interfaz definida individualmente deben ser descritos. Esta información puede ser incluida en la SyRS siempre y cuando no sea demasiado extensa. Aunque en la mayoría de los casos es mejor tener un documento de control de interfaces externas del sistema. Hay muchos tipos de posibles interfaces externas al sistema y un solo sistema puede tener varias interfaces de diferentes tipos. La siguiente lista provee algunos ejemplos:

- Operacional
- Computadora a computadora
- Eléctrica
- Intercambio de datos y protocolos
- Líneas de telecomunicaciones
- Dispositivo a sistema, sistema a dispositivo
- Computadora a sistema, sistema a computadora
- Sensible al ambiente y control de interfaces

5.1.2. Representación para el cliente

Retroalimenta al cliente, incluye representaciones de SyRS e intercambio técnico o comunicación de aclaraciones y/o confirmación de requerimientos.

5.1.3. Retroalimentación del cliente

Incluye información actualizada de los objetivos, problemas o necesidades del cliente; modificación de requerimientos concernientes a la comunicación de intercambio técnico; e identificar nuevos requerimientos.

5.2. Ambiente

Además del analista y del cliente, el ambiente puede implícita o explícitamente influenciar o poner restricciones en los requerimientos del sistema. El analista debe estar enterado de estas influencias en las capacidades del sistema. En caso de que el sistema sea sensible a las influencias del ambiente, el cliente y el analista especificarán las influencias que afectan los requerimientos del sistema. Las influencias ambientales pueden ser clasificadas en categorías traslapadas, como sigue:

- Política
- De mercado
- Estándares y políticas técnicas
- Cultural
- Organizacional
- Físicas

5.2.1. Influencia Política

Agencias gubernamentales internacionales, federales, estatales y locales tienen leyes y regulaciones que influyen en los requerimientos del sistema. Algunas agencias gubernamentales pueden tener organizaciones que regulan y obligan a cumplir sus leyes y regulaciones. Ejemplos de leyes gubernamentales son derechos reservados, patentes, y marcas registradas. Ejemplos de regulaciones gubernamentales son zonas de ubicación, riesgos ambientales, desechos, reciclaje, sistemas de seguridad y salud.

La influencia política cambia en función de limitaciones políticas, lo cual afecta los requerimientos de sistema, ya que un ambiente puede ser completamente diferente de otro. Por lo que es importante hacer una investigación del ambiente político donde será manufacturado y/o usado para asegurar que el sistema cumple con todas las leyes y regulaciones gubernamentales.

5.2.2. Influencia del mercado

Hay tres tipos de condiciones de mercado que influyen en el desarrollo de las especificaciones del sistema. La primera es cumplir las necesidades del cliente haciendo una investigación de mercado o desarrollando mercados que cumplan con la investigación técnica. Cumplir las necesidades del cliente en los sistemas afecta los requerimientos del sistema y se convierte en parte de los requerimientos del cliente.

La segunda influencia del mercado es la satisfacción de la demanda. Esta influencia debe ser considerada porque afecta la distribución y accesibilidad del sistema, lo cual es agregado a los requerimientos del sistema. Sin acceso sencillo al sistema, el éxito será limitado. Por lo que, es importante considerar los segmentos de mercado para los que el sistema está enfocado y considerar la información de mercado puede ser usado para derivar los requerimientos del sistema.

La tercera influencia del mercado es la competencia. Conociendo los sistemas de la competencia nos ayudará a definir los requerimientos. Para mantenerse competitivo, se debe considerar lo siguiente:

- a) Funcionalidad
- b) Precio
- c) Confiabilidad
- d) Durabilidad
- e) Funcionamiento
- f) Mantenimiento
- g) Seguridad del sistema

Analizar la competencia en el mercado es un proceso continuo que afectará los requerimientos de los sistemas nuevos y de los ya existentes. Los sistemas pueden evolucionar en sistemas

completamente diferentes que pueden tener pequeñas semejanzas a los conceptos originales del cliente.

5.2.3. Influencia de los estándares y reglamentos técnicos

Los requerimientos del sistema están influenciados directamente por los clientes quienes tienen que estar conforme a los estándares y reglamentos técnicos dictados por el gobierno o industrias. Las políticas técnicas y estándares asociados y los lineamientos ayudan a asegurar lo siguiente:

- a) consistencia del sistema
- b) seguridad del sistema
- c) confiabilidad y mantenimiento del sistema

Los estándares de seguridad industrial son generalmente impuestos para ayudar a prevenir riesgos y problemas legales potenciales. Cumplir con las reglas de seguridad debe estar claramente identificado en el documento de SyRS. El cliente y la comunidad técnica pueden requerir que el sistema pase ciertos criterios de confiabilidad como está prescrito en los estándares técnicos. Los requerimientos de confiabilidad y mantenimiento deben estar identificados en la SyRS. Estos requerimientos pueden venir en varias formas.

5.2.4. Influencia cultural

La cultura son los patrones de comportamiento humanos que son transmitidos de generación en generación. Es una experiencia adquirida que proviene de creencias religiosas, país de origen, grupo étnico, nivel socioeconómico, lenguaje, medios de comunicación, empleo y familia inmediata. Para entender la cultura de la región o de un segmento de mercado, deben conocerse los valores y creencias de la gente. La influencia cultural debe ser considerada

cuando se está desarrollando un sistema porque afectará los requerimientos de éste.

5.2.5. Influencia organizacional

Los requerimientos del sistema son influenciados por la organización en la cual los requerimientos son desarrollados. La influencia de la organización puede tomar la forma de mercadotecnia, políticas internas, reglamentos técnicos y estándares internos. Por ejemplo, cada compañía tiene su propia cultura, propósito, valores, objetivos, que pueden e influenciarán el sistema que desarrollan, manufacturan y/o entregan.

5.2.6. Influencia física

Incluyen las influencias naturales y humanas tales como temperatura, radiación, humedad, presión y químicos.

5.3. Comunidad técnica

La comunidad técnica está compuesta de aquellos involucrados en las actividades de diseño, implementación, integración, pruebas, manufactura, despliegue, operaciones y mantenimiento del sistema. Todos los elementos de la comunidad técnica deben estar involucrados en el proceso de desarrollo de la SyRS tan pronto como sea posible. La inclusión temprana de la comunidad técnica provee un mecanismo para que los desarrolladores de la SyRS reduzcan la posibilidad de que nuevos requerimientos y cambios a los requerimientos originales sean descubiertos después en el ciclo de vida del sistema.

5.3.1. Representación técnica

Representación del conjunto de los requerimientos, preparados para la comunidad técnica, puede incluir intercambios técnicos o comunicaciones que clarifican y/o conforman requerimientos.

5.3.2. Retroalimentación técnica

La comunidad técnica proporciona retroalimentación durante varias actividades que pueden causar modificaciones, adiciones y/o eliminaciones al conjunto de requerimientos. La SyRS es refinada debido a la necesidad de apoyar las fases subsecuentes del ciclo de vida del sistema. Por ejemplo, después de la fase de requerimientos, se desarrolla un plan de pruebas del sistema donde los requerimientos individuales son asignados a pruebas específicas. Este proceso puede revelar requerimientos que no pueden ser probados, resultando en la modificación de la SyRS.

Otra retroalimentación de la comunidad técnica puede proveer a los clientes de las características más recientes del sistema, tecnología de punta, así como recomendaciones de métodos avanzados de implementación.

6. Requerimientos bien formados

Esta cláusula explica las propiedades de un requerimiento bien formado. Provee un ejemplo de un requerimiento bien formado, y señala los errores comunes en los requerimientos.

6.1. Definición de un requerimiento bien formado

Un requerimiento bien formado es una declaración de la funcionalidad del sistema que puede ser validada, y que debe ser poseído por el sistema para resolver el problema de un cliente o lograr el objetivo del cliente, y que está calificado por condiciones medibles, y delimitado mediante restricciones. Esta definición ayuda en la clasificación de los requerimientos generales del cliente. Éstos pueden ser tomados de las necesidades del cliente y pueden ser derivados del análisis técnico. La definición provee un medio para distinguir entre requerimientos como capacidades y los atributos de estos requerimientos. Las restricciones pueden ser funcionales o no

funcionales. Un ejemplo de una restricción no funcional podría ser que el sistema sea pintado con un tono particular de azul solamente para propósitos decorativos.

Esta guía recomienda que el proceso de implementación de requerimientos del sistema, tal como demandar una metodología particular de diseño no sea incluido en la SyRS. Éstos deben ser capturados en otra documentación técnica de control del sistema como sería planes de calidad o contratos de trabajo.

6.1.1. Capacidades

Son los requerimientos fundamentales del sistema y representan las características o funciones del sistema requerido por el cliente. Una capacidad debe usualmente ser enunciada de tal manera que describa lo que el sistema debe hacer. La capacidad debe además ser enunciada de tal manera que sea independiente de la solución. Esto permitirá considerar diferentes maneras de alcanzar los objetivos o de proveer la característica o función.

Por ejemplo, capacidades de un sistema de trenes de alta velocidad entre L.A. y N.Y. puede incluir la capacidad de arrancar, acelerar, circular, desacelerar, parar, subir y bajar pasajeros. De cualquier manera, ni el tipo de computadora ni el sistema operativo se consideran capacidades del sistema de trenes de alta velocidad.

6.1.2. Condiciones

Son atributos y características medibles, cualitativa o cuantitativamente, que son estipulados para una capacidad. Permiten calificar una capacidad necesaria, y proporciona atributos que permiten que una capacidad sea formulada y enunciada de tal manera que pueda ser validada y verificada. Por ejemplo, en el sistema de trenes de alta velocidad antes mencionado, la capacidad

de circular puede ser circular en un rango de 0-300 Km/hr o en un rango óptimo de 200 Km/hr.

Tiene sentido incluir condiciones (atributos medibles) sólo si son aplicadas a algo que pueda ser medido tal como una capacidad. Por ejemplo, es insignificante tener un requerimiento de sistema que diga de 0-200 Km/hr en abstracto. Este rango puede describir una velocidad de circulación para una vía de alta velocidad, pero no la velocidad de un elevador.

6.1.3. Restricciones

Son requerimientos que son impuestos sobre la solución por circunstancias, por la fuerza o por compulsión. Limita absolutamente las opciones abiertas al diseñador de la solución imponiendo límites inamovibles.

Una lista de restricciones puede incluir una lista de interfaces a sistemas ya existentes (por ejemplo, formato, protocolos o contenidos) donde la interfaz no puede ser cambiada, limitaciones físicas (por ejemplo, cuando un controlador debe caber en cierto espacio en el ala de un avión), leyes de la naturaleza, leyes de un país, tiempo o presupuesto disponibles, prioridad (obligatorio u opcional), o plataformas existentes.

Las restricciones pueden ser identificadas como requerimientos por sí mismas, o como limitaciones de requerimientos individuales. Muchas restricciones, tal como el uso de determinada tecnología (por ejemplo, el sistema operativo) serán aplicables a todo el conjunto de capacidades. Otras restricciones son aplicables a una o a pocas capacidades.

6.1.4. Ejemplo

Capacidad: Trasladar personas entre L.A y N.Y.

Condición: Velocidad de circulación de 200 Km/hr

Restricción: Velocidad máxima de 300 Km/hr

6.2. Propiedades de un requerimiento

Cada requerimiento debe poseer las siguientes propiedades:

- a) Abstracto. Cada requerimiento debe ser independiente de su implementación.
- b) No ambiguo. Cada requerimiento debe ser enunciado de tal manera que pueda ser interpretado de una sola manera.
- c) Rastreado. Para cada requerimiento debe ser factible determinar una relación entre las declaraciones realizadas por el cliente y los requerimientos especificados en la SyRS, esto como evidencia del origen del requerimiento.
- d) Validable. Para cada requerimiento debe existir la manera de probar que el sistema satisface los requerimientos.

6.3. Clasificación

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 ejemplo, 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 ejemplo, componentes disponibles comercialmente vs análisis original del sistema), el ambiente del cliente (por ejemplo, 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
 - Salida
 - Confiabilidad
 - Disponibilidad
 - Facilidad de mantenimiento

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

6.4. Errores comunes

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 ejemplo, 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 horas. 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 ejemplo, el requerimiento “minimizar ruido” puede ser enunciado como “los rangos de ruido no deben exceder...”).
 - 2) Requerimientos abiertos (por ejemplo, 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).

7. Desarrollo de la SyRS

El desarrollo de SyRS es un proceso iterativo. Los cuatro subprocesos que incluye son los siguientes:

- Identificar requerimientos del usuario, el ambiente y la experiencia de la comunidad técnica;
- Construir requerimientos bien formados;
- Organizar los requerimientos en una SyRS;
- Presentar la SyRS en varias representaciones para diferentes audiencias.

El propósito de descomponer el proceso de desarrollo de la SyRS en subprocesos auxilia a un desarrollo completo y correcto de la SyRS. Los subprocesos presentes en la siguiente figura ocurren secuencialmente. Sin embargo, frecuentemente, existirá algún grado de traslape o iteración.

La aplicación iterativa de este proceso resulta en la subsecuente modificación de la SyRS. Las modificaciones son generalmente aplicadas a una línea base de la SyRS, y gestionada bajo procedimientos de control de cambios. Ver IEEE Std 1220-1998 para procedimientos de control de cambios.

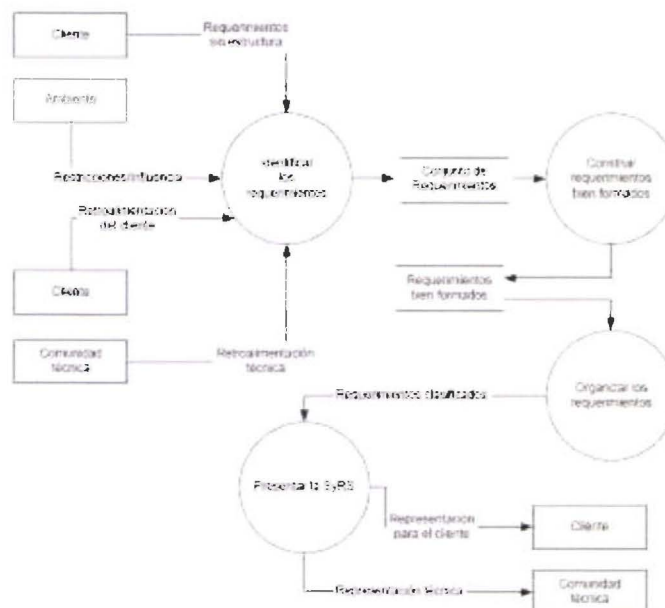


Fig. 3.3. Proceso de desarrollo de la SyRS

7.1. Identificar los requerimientos

Mientras se trabaja con los clientes, los analistas filtran las entradas y extraen un conjunto de requerimientos, establecen los requerimientos derivados necesarios y crean los requerimientos. Los requerimientos pueden ser extraídos desde los documentos iniciales hasta ejercicios analíticos conducidos con el cliente. La meta de este proceso iterativo es extraer todos los requerimientos del sistema, y asegurarse de que cada requerimiento es establecido sólo una vez y que no falta ninguno.

Existen una variedad de enfoques que pueden aplicarse durante la identificación de requerimientos.

En la práctica, cada organización tendrá sus propias técnicas para identificar los requerimientos e iniciar el proceso de crear una solución para el sistema. En algunas organizaciones, los clientes tomarán el proceso de identificación de forma interna. En este caso la identificación y especificación de requerimientos es dirigida por el cliente. En otras organizaciones, los clientes identificarán un conjunto preliminar de requerimientos y solicitarán asistencia de un analista dentro de su organización o a través de un contrato con un consultor externo o un integrador de sistemas.

El analista, ya sea interno o externo a la organización, trabajará junto con el cliente para identificar y estructurar los requerimientos. En algunas organizaciones el analista trabajará directamente con el cliente. En otras organizaciones, el analista no tendrá acceso directo al cliente y trabajará a través de uno o más intermediarios (legales, técnicos, administrativos) representantes del cliente.

Debido a la dinámica que involucra la identificación de requerimientos, es importante que el cliente y el analista estén de

acuerdo en el proceso. El analista requiere preparar un plan para guiar el proceso, definir representaciones de la SyRS que serán producidas para las diferentes audiencias, e identificar las herramientas y técnicas a ser usadas.

Un proceso de gestión de requerimientos debe ser usado para asegurar lo siguiente:

- a) El proceso está orientado a los objetivos y dirigido a la producción de un conjunto de requerimientos.
- b) El alcance del sistema es definido.
- c) Todos los requerimientos son detectados, evaluados y documentados.
- d) Los requerimientos son especificados como capacidades, y condiciones calificativas y restricciones son identificadas para cada capacidad.
- e) Los requerimientos son validados, o desechados si son inválidos, del conjunto de requerimientos.
- f) Se considera la consistencia del documento cuando participan varios autores en la producción de la SyRS.
- g) El conjunto de requerimientos bajo desarrollo es comprendido al nivel apropiado de detalle por todos los individuos participantes del proceso.

7.1.1. Técnicas para identificar requerimientos

Los requerimientos surgen como ideas o conceptos que pueden originarse como una respuesta a una amenaza percibida o competencia del mercado, de una imposición legal o regulación, del deseo de crear un nuevo o mejor sistema o proceso, de la necesidad de reemplazar un sistema existente o alguna otra necesidad percibida.

Hay muchas técnicas para identificar los requerimientos, incluyendo las siguientes:

- a) Talleres estructurados
- b) Sesiones de tormentas de ideas
- c) Entrevistas
- d) Cuestionarios
- e) Observación de campo
- f) Revisión de la documentación técnica
- g) Ingeniería inversa
- h) Simulaciones
- i) Prototipos

7.1.2. Interacción entre clientes y analistas

En una situación donde un analista ha sido contratado para trabajar con un cliente, será necesario establecer un proceso efectivo de interacción entre las dos partes. Para hacer esta interacción efectiva, cada parte necesita entender que ellos tienen el rol de enseñar a la otra parte y que deben trabajar juntos para definir los requerimientos.

7.1.2.1. Educación mutua

La educación debe ser un proceso de dos vías. Primero, el analista necesita aprender acerca del ambiente del cliente, del sistema actual (si existe) y de los requerimientos. Se requiere asignar tiempo y esfuerzo por ambas partes para este proceso educativo.

Segundo, los clientes también necesitan educación. Pueden necesitar educación del analista durante el proceso de identificación y especificación de los requerimientos. Además, el analista puede ser requerido para educar a sus clientes con respecto a los requerimientos por sí mismos y contribuir requerimientos desde su experiencia.

7.1.2.2. Definición conjunta de los requerimientos

Hay múltiples maneras en las cuales el cliente y el analista pueden interactuar en el proceso de definición de requerimientos. Por ejemplo, el analista puede simplemente conducir entrevistas para solicitar datos y entonces organizar y presentar los requerimientos para ser revisados por el cliente.

La experiencia del analista es muy importante, pero no debe influir ni menospreciar la participación del cliente de ninguna manera. Mientras trabajan con el cliente, su objetivo principal debe ser el de solicitar y organizar los requerimientos derivados del cliente. También deben agregar requerimientos desde su propia experiencia o de previas soluciones de sistemas predefinidos sólo cuando estos requerimientos han sido olvidados por el cliente.

Como otro ejemplo, personal del cliente puede trabajar con el analista en sesiones de trabajo en grupo. En estas sesiones puede haber una gran cantidad de ideas y definición interactiva de requerimientos. Estas sesiones son normalmente dirigidas por el analista. Los resultados de estas sesiones son documentados por los analistas.

Una forma más cooperativa puede involucrar al cliente directamente en la definición de los requerimientos. Personal del cliente puede participar en la definición de los requerimientos hasta el punto de que ellos también sean autores del documento.

Cualquiera que sea la técnica usada, el objetivo es definir los requerimientos mientras se consigue un consenso y un

nivel común de entendimiento. El cliente y el analista deben llegar al punto en el cual tienen un entendimiento común de los requerimientos y puedan representarlos consistentemente en forma de una SyRS, a satisfacción del cliente.

7.2. Construir requerimientos bien formados

El analista realiza esta subfase haciendo lo siguiente:

- a) Se asegura de que cada requerimiento es una declaración breve y definitiva de una necesidad.
- b) Define las condiciones apropiadas para cada requerimiento (medidas cuantitativas y cualitativas) y evite adjetivos tal como "resistente" o "aceptado por industria".
- c) Evite los errores comunes de especificación (ver sección 6.4).
- d) Se asegura de la legibilidad de los requerimientos, lo cual involucra lo siguiente:
 - 1) Palabras/frases/conceptos simples
 - 2) Una estructura y relaciones uniformes
 - 3) Definición de palabras, símbolos y notaciones únicas.
 - 4) El uso de un lenguaje y simbología gramaticalmente correctos.
- e) Se asegura de que los requerimientos puedan ser probados.

El siguiente ejemplo es un requerimiento bien formado:

Capacidad: Trasladar personas entre L.A y N.Y.

Condición: Velocidad de circulación de 200 Km/hr

Restricción: Velocidad máxima de 300 Km/hr

Requerimiento bien formado: El sistema debe trasladar personas entre L.A y N.Y. a una velocidad óptima de circulación de 200 Km/hr con un límite máximo de velocidad de 300 Km/hr.

7.3. Organizar los requerimientos

En este subproceso, el analista da estructura al conjunto de los requerimientos relacionándolos unos con otros de acuerdo a algún método comparativo definido. Algunas tareas de este subproceso pueden ser automatizadas.

Esta actividad se caracteriza por lo siguiente:

- Busca patrones para poder agrupar conjuntos de requerimientos
- Usa la experiencia y el juicio para usar técnicas apropiadas.
- Usa la creatividad y la intuición para generar diferentes alternativas y para priorizar los requerimientos de acuerdo a la información proporcionada por el cliente.
- Define las propiedades de los requerimientos
- Define los atributos de los requerimientos.

Los atributos de los requerimientos pueden ser asignados a cada requerimiento bien formado de la siguiente manera:

Identificador = 2.1.3.6

Prioridad = alta

Criticismo = bajo

Factibilidad = alta

Riesgo = medio

Fuente = cliente

Tipo = desempeño

Existen varios esquemas para organizar las especificaciones en forma de un conjunto ordenado. El esquema más común es organizar los requerimientos en una jerarquía de capacidades (servicios) donde las capacidades generales son descompuestas en requerimientos subordinados. Otro esquema es mediante el uso de referencias cruzadas para mostrar las relaciones entre los requerimientos de menor nivel. Sin importar el método que se use, la

SyRS debe mostrar la relación entre los requerimientos. Algunas relaciones entre los requerimientos incluyen las siguientes dependencias jerárquicas:

- a) Eventos
- b) Datos
- c) Objetos físicos o abstractos
- d) Funciones

7.4. Presentar la SyRS

En este subproceso, el analista trabajará con el cliente para identificar la mejor manera de comunicar los requerimientos a todos los individuos que necesiten entender, revisar, aceptar o usar la SyRS. Una sola representación no siempre es conveniente porque:

- a) El cliente y la comunidad técnica usualmente tienen diferentes culturas y lenguajes; así, los mismos requerimientos del sistema deberían ser presentados de manera diferente a los técnicos o a los clientes.
- b) La recuperación de información específica es difícil mediante algunas representaciones.
- c) La presentación de interacciones puede ser difícil de realizar en algunos métodos de representación.
- d) Relacionar información de un lugar con información en otro lugar puede ser difícil en algunas representaciones.

Por lo tanto, es importante que los analistas junto con el cliente identifiquen la mejor manera de comunicar los requerimientos a todos los individuos involucrados en el proceso de desarrollo de la SyRS. Para lograr lo anterior, deben usarse diferentes representaciones; estas representaciones no deben ser mantenidas por separado, sino que deben ser derivadas y generadas de la SyRS. Por ejemplo, se puede producir un resumen que contenga una descripción narrativa para el cliente. Para el individuo

responsable de aceptar los requerimientos por parte del cliente, se puede generar un documento más detallado que incluya modelos formales. Para el equipo de diseño, un conjunto completo de modelos formales pueden ser generados. Deben usarse herramientas automatizadas para mantener la SyRS y producir sus diferentes representaciones.

7.5. Métodos de representación

Los métodos de representación pueden incluir uno o una combinación de los siguientes:

1. Textual
 - a. Papel
 - b. Electrónico

2. Modelos
 - a. Físico
 - b. Simbólico
 - c. Grafico
 - d. Prototipo

3.4. ESTÁNDAR IEEE 1362-1998: GUÍA PARA TECNOLOGÍA INFORMÁTICA DEFINICIÓN DE SISTEMA-CONCEPTO DE OPERACIONES Y DOCUMENTOS (ConOps)

El Estándar IEEE 1362 ofrece un formato y contenidos para la confección de las descripciones de sistema en los desarrollos y modificaciones de sistemas intensivos de software.

El estándar no especifica técnicas exactas, sino que proporciona las líneas generales que deben respetarse. No es por tanto un modelo final, sino una guía de referencia sobre la que cada organización debe desarrollar sus propias

prácticas y procedimientos para preparar y actualizar su documentación con las descripciones de los sistemas.

Las partes esenciales de un ConOps son:

- Descripción del sistema existente.
- Justificación del desarrollo o de la modificación.
- Descripción del sistema propuesto.
- Los proyectos de tamaño pequeño requieren descripciones de sistema menos formales, pero no por su reducido tamaño debe ignorarse.

Si el proyecto de software forma parte de un proyecto mayor, la descripción del sistema de software puede ser un documento separado, o ir incluido en la descripción del sistema completo.

El estándar puede aplicarse a todos los tipos de sistemas de software: sólo software, intensivos de software o software/hardware/personas. Aunque los conceptos del estándar también podrían aplicarse a sistemas de hardware, esta no es su finalidad.

El estándar identifica los elementos que al menos debe incluir una Descripción del sistema. El usuario puede incorporar otros elementos, agregando cláusulas y sub-cláusulas.

3.4.1. Esquema de un documento de Concepto de Operaciones

La siguiente figura muestra la estructura de un documento de Concepto de Operaciones propuesta por el IEEE en su estándar 1362 [IEEE, 1998].

1. Alcance
 - 1.1. Identificación
 - 1.2. Visión general del documento
 - 1.3. Visión general del sistema
 - 1.4. Personal involucrado
2. Documentos referenciados
3. Situación Actual
 - 3.1. Antecedentes
 - 3.2. Políticas y restricciones operacionales
 - 3.3. Descripción del sistema o situación actual
 - 3.4. Tipos de usuarios
 - 3.5. Mantenimiento/Soporte
 - 3.6. Necesidad y naturaleza de los cambios
 - 3.7. Descripción de los cambios deseados
4. Sistema propuesto
 - 4.1. Antecedentes
 - 4.2. Políticas y restricciones operacionales
 - 4.3. Descripción del sistema propuesto
 - 4.4. Tipos de usuarios
 - 4.5. Mantenimiento/Soporte
 - 4.6. Escenarios operacionales
 - 4.7. Futuras evoluciones
 - 4.8. Cambios considerados pero no incluidos
5. Resumen de mejoras
6. Información adicional

Fig. 3.4. Estructura de un documento de Concepto de Operaciones

1. Alcance

El contenido de este apartado debe proporcionar una breve introducción del documento, su relación con otros posibles documentos, su finalidad y los destinatarios de la información que contiene.

1.1. Identificación

Identificación del sistema, proporcionando su nombre y abreviatura (si procede).

1.2. Visión general del documento

Explicación del propósito, audiencia y consideraciones de seguridad o privacidad del documento.

Generalmente, el propósito del documento suele ser uno de los siguientes:

- Comunicar las necesidades y explicaciones del cliente.
- Comunicar el entendimiento del proyecto.
- Obtener acuerdos entre las partes implicadas (personal representante de la parte cliente, personal del equipo de desarrollo, etc.).

1.3. Visión general del sistema

Resumen del propósito del sistema o subsistema propuesto (al cual se aplica la descripción del sistema).

1.4. Personal involucrado

Relación de personas involucradas en el desarrollo del sistema con información de contacto.

Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requerimientos, validaciones de seguimiento, etc.

| | |
|-------------------------|--|
| Nombre | |
| Rol | |
| Categoría profesional | |
| Responsabilidades | |
| Información de contacto | |
| Aprobación | |

Tabla 3.1. Formato para registro del personal involucrado

2. Documentos referenciados

| Nº | Título | Ruta | Versión | Fecha | Autor |
|----|--------|------|---------|-------|-------|
| | | | | | |
| | | | | | |
| | | | | | |

Tabla 3.2. Formato para registro de documentos referenciados

Relación de los documentos a los que se hace referencia en los requerimientos del sistema, indicando según proceda: el código del documento, título, ruta, revisión, fecha y origen.

3. Situación actual

Cuando el sistema propuesto tenga como finalidad la sustitución o ampliación de un sistema o situación existente, este apartado contendrá una descripción suficiente del actual. (aunque este último sea un sistema manual).

3.1. Antecedentes

Visión general del sistema o situación actual, incluyendo: misión, objetivos, alcance, modos de uso o aplicación, etc.

3.2. Políticas y restricciones operacionales

Relación de políticas o restricciones de cualquier índole impuestas sobre el sistema o situación actual. Algunos ejemplos pueden ser:

- Restricciones sobre el número de usuarios capaces de usar el sistema.
- Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)
- Restricciones de seguridad o relativas a protección de datos.
- Restricciones de software (uso obligado de una determinada base de datos, sistema operativo, etc.)

- Restricciones de recursos operacionales como espacio físico.

3.3. Descripción del sistema o situación actual

Descripción detallada del sistema o situación actual y de su funcionamiento, incluyendo los siguientes subapartados:

- Enumeración y descripción de funciones, características y capacidades del sistema o situación actual.

En el caso de considerarse apropiado puede aportarse información adicional como:

- Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema o situación actual.
- Interacción entre componentes del sistema.
- Descripción del entorno de operación y sus características.
- Interacción del sistema con otros sistemas externos.
- Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, etc.
- Atributos de calidad como: disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, etc.
- Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.
- Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, costo de las operaciones, etc.

Es importante que la descripción del sistema o situación actual sea lo más simple o clara posible, para que todos los lectores del documento puedan entenderlo completamente. Debe realizarse usando la terminología del usuario.

3.4. Tipos de usuarios

Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual el usuario interactúa con el sistema.

Factores como la responsabilidad, habilidades, competencias, etc. Distinguen a los distintos tipos de usuarios.

| | |
|----------------------------|--|
| Tipo de usuario | |
| Responsabilidad | |
| Formación | |
| Habilidades | |
| Actividades | |
| Interacción con el sistema | |

Tabla 3.3. Formato para registro de tipos de usuarios

3.5. Mantenimiento/Soporte

Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.

3.6. Necesidad y naturaleza de los cambios

Descripción de las carencias, defectos o debilidades del sistema o situación actual y que motivan al desarrollo de un nuevo sistema o a una modificación del existente.

De no existir un sistema anterior (ni tan siquiera manual), en este apartado y subapartados deben quedar reflejadas las justificaciones que llevan al cambio.

3.7. Descripción de los cambios deseados

Enumeración de las capacidades, funciones, procesos, etc. que deben generarse o modificarse para satisfacer las nuevas necesidades.

Los cambios deben basarse en el sistema descrito en el punto 3.3. Si no existe un sistema anterior, en este apartado se enumeran las capacidades requeridas del nuevo sistema.

De forma apropiada podrá hacerse referencia a:

- **Cambios en la capacidad.** Descripción de las funciones y características que deben añadirse, eliminarse o modificarse para conseguir los objetivos y requerimientos del nuevo sistema.
- **Cambios en el proceso del sistema.** Descripción de los cambios en el proceso o procesos de transformación de datos que darán lugar a nuevos resultados con los mismos datos, mismos resultados con nuevos datos o ambos.
- **Cambios de interfaces.** Descripción de los cambios en el sistema que provocan cambios en los interfaces y cambios en las interfaces que causan cambios en el sistema.
- **Cambios de personal.** Descripción de cambios en personal causados por nuevos requerimientos, cambios de tipos de usuarios o ambos.
- **Cambios de entorno.** Descripción de cambios en el entorno operacional que causan cambios en las funciones, procesos, interfaces o personal del sistema y/o cambios que deben realizarse en el entorno por causa de cambios de las funciones, procesos, interfaces o personal del sistema.
- **Cambios operacionales.** Descripción de cambios a procedimientos, métodos, rutinas de trabajo del usuario causadas por los cambios mencionados.
- **Cambios de soporte.** Cambios en los requerimientos de soporte causados por cambios en las funciones, procesos, interfaces o personal del sistema y/o cambios en las funciones , procesos, interfaces o personal del sistema causados por cambios de soporte.

- **Otros cambios.** Descripción de otros cambios que afectarán a los usuarios.

Deben identificarse las prioridades entre cambios deseados y nuevas características. Esta identificación puede realizarse clasificando cada uno de los cambios como esencial, deseado u opcional.

Esta información será útil en la toma de decisiones durante el desarrollo y en el caso de interrupciones o invasiones de agendas o presupuestos.

4. Sistema propuesto

Descripción del sistema propuesto. Debe ceñirse a los conceptos básicos que indican las nuevas capacidades operacionales sin entrar en especificaciones de diseño, a no ser que sean restricciones impuestas.

4.1. Antecedentes

Visión general del sistema propuesto, incluyendo misión, objetivos, alcance, modo de uso o aplicación, etc.

4.2. Políticas y restricciones operacionales

Relación de políticas o restricciones de cualquier índole aplicables al sistema propuesto. Algunos ejemplos pueden ser:

- Restricciones sobre el número de usuarios capaces de utilizar el sistema.
- Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)
- Restricciones de seguridad o relativas a protección de datos.
- Restricciones de software (uso obligado de una determinada base de datos, sistema operativo, etc.)
- Restricciones de recursos operacionales como espacio físico.

4.3. Descripción del sistema propuesto

Descripción detallada del sistema propuesto y de su funcionamiento, incluyendo los siguientes subapartados:

- Enumeración y descripción de funciones, características y capacidades del sistema propuesto.

Deben identificarse las prioridades entre funcionalidades y características. Esta identificación puede realizarse clasificando cada una de las funciones, características y capacidades como esencial, deseado u opcional.

Deben incluirse en la medida de lo posible:

- Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema o situación actual.
- Interacción entre componentes del sistema.
- Descripción del entorno de operación y sus características.
- Interacción del sistema con otros sistemas externos.
- Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, etc.
- Atributos de calidad como: disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, etc.
- Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.
- Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, costo de las operaciones, etc.

Es importante que los requerimientos del sistema propuesto sea lo más simple o clara posible, para que todos los lectores del documento puedan entenderlo completamente. Debe realizarse usando la terminología del usuario.

4.4. Tipos de usuarios

Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual el usuario interactúa con el sistema.

Factores como la responsabilidad, habilidades, competencias, etc. Distinguen a los distintos tipos de usuarios.

| | |
|----------------------------|--|
| Tipo de usuario | |
| Responsabilidad | |
| Formación | |
| Habilidades | |
| Actividades | |
| Interacción con el sistema | |

Tabla 3.4. Formato para registro de tipos de usuarios

4.5. Mantenimiento/Soporte

Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.

4.6. Escenarios operacionales

Relación de escenarios operacionales. Un escenario es una descripción paso a paso de cómo el sistema propuesto debe operar e interactuar con sus usuarios y sus interfaces externas, bajo unas determinadas circunstancias.

Los escenarios operacionales proporcionan información sobre cómo interactúan todas las partes del sistema propuesto, los usuarios y otras entidades externas. También puede utilizarse para describir lo que no debe hacer el sistema.

4.7. Futuras evoluciones

Descripción de las previsiones de evolución que se tienen previstas para el sistema (si se tiene).

Esta información resulta útil para el personal involucrado en el desarrollo del nuevo sistema para asimilar cambios futuros con el menor impacto.

4.8. Cambios considerados pero no incluidos

Identificación de cambios considerados pero no incluidos en el sistema propuesto y el motivo por el que no han sido incluidos.

5. Resumen de mejoras

Resumen de los beneficios proporcionados por el sistema propuesto. Este resumen puede incluir las siguientes secciones:

- Nuevas características y funcionalidades.
- Existentes características actualizadas.
- Características obsoletas, no útiles o confusas eliminadas.
- Mejoras de rendimiento como tiempo de respuesta, mejoras de calidad, etc.

6. Información adicional

Cualquier información adicional que facilite la comprensión del documento en sí.

3.5. VENTAJAS DE USAR ESTÁNDARES EN LA DOCUMENTACIÓN DE REQUERIMIENTOS

La principal ventaja de estandarizar la documentación, es que ésta nos servirá para:

- Comunicar de manera precisa los requerimientos, objetivos y presunciones del dominio
- Contrato legal, documento interno o a modo de memorando
- Base para estimación (tamaño, costo, tiempo) y planificación de proyecto
- Base para evaluación de producto final

- verificación y validación
- Debería tener suficiente información para decidir si el producto final es aceptable (satisface los requerimientos)
- Base para el control de cambios
 - Requerimientos cambian, software evoluciona, el entorno evoluciona

3.6. ANÁLISIS DE LAS OPORTUNIDADES DE MEJORA

Es muy importante mencionar que el poder formular una especificación de requerimientos completa y consistente, es un paso muy importante para evitar cometer errores en la definición de los requerimientos, ya que los mismos pueden resultar muy caros de corregir una vez desarrollado el sistema. De ahí, la vital importancia que tiene la ingeniería de requerimientos en generar una adecuada especificación que contemple claramente y sin ambigüedades los requerimientos del sistema a desarrollar, con el fin primordial de evitar que los proyectos fracasen debido a una mala elaboración de la definición y especificación de requerimientos.

Un factor estratégico para las empresas desarrolladoras de software debe ser la aplicación de modelos de mejoramiento de procesos, que una vez adoptados en los proyectos de software puedan disminuir los costos de producción y la inversión de recursos en el mantenimiento de los respectivos productos y servicios. Los modelos y metodologías actuales son extranjeros y ajenos a las condiciones y/o características propias, con servicios de capacitación muy formales y servicios de consultoría excesivamente costosos y no son fáciles de aplicar en organizaciones pequeñas.

Los principales puntos a considerarse para la mejora del proceso de ingeniería de requerimientos a través de los estándares IEEE 830, 1233 y 1362 están:

- **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 la Ingeniería de Requerimientos 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.

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 Ingeniería de Requerimientos es la principal.

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.

- **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.

Además, el documento a generar deberá tener las siguientes cualidades:

- **Completitud**
 - con respecto a los objetivos
 - con respecto a entradas: el comportamiento requerido del software ha sido especificado para todas las entradas posibles.
 - con respecto a estructura: no hay secciones rotuladas: “A completar...”
- **Pertinencia**
 - Cada requerimiento y presunción se necesita para la satisfacción de objetivo.
 - El documento no contiene elementos que no están relacionados con la definición de requerimientos (ej. decisiones de diseño o implementación)
- **Consistencia**
 - No hay contradicciones en la formulación de objetivos, requerimientos y presunciones
- **Medibilidad**

Los requerimientos han sido formulados de manera tal que su satisfacción pueda ser evaluada de manera no ambigua.
- **Precisión (No ambiguo)**
 - No hay vocabulario ambiguo: cada término está definido y es usado consistentemente.
 - No hay aseveraciones ambiguas: Objetivos, requerimientos y presunciones deben estar escritos de manera tal que no permiten interpretaciones distintas
 - No hay responsabilidades ambiguas: la separación de responsabilidades entre el mundo y el software debe estar indicado claramente.

- Factibilidad
 - Los objetivos y requerimientos deberían ser realizables dentro del presupuesto y cronogramas dispuestos.

CAPÍTULO IV

METODOLOGÍA PROPUESTA

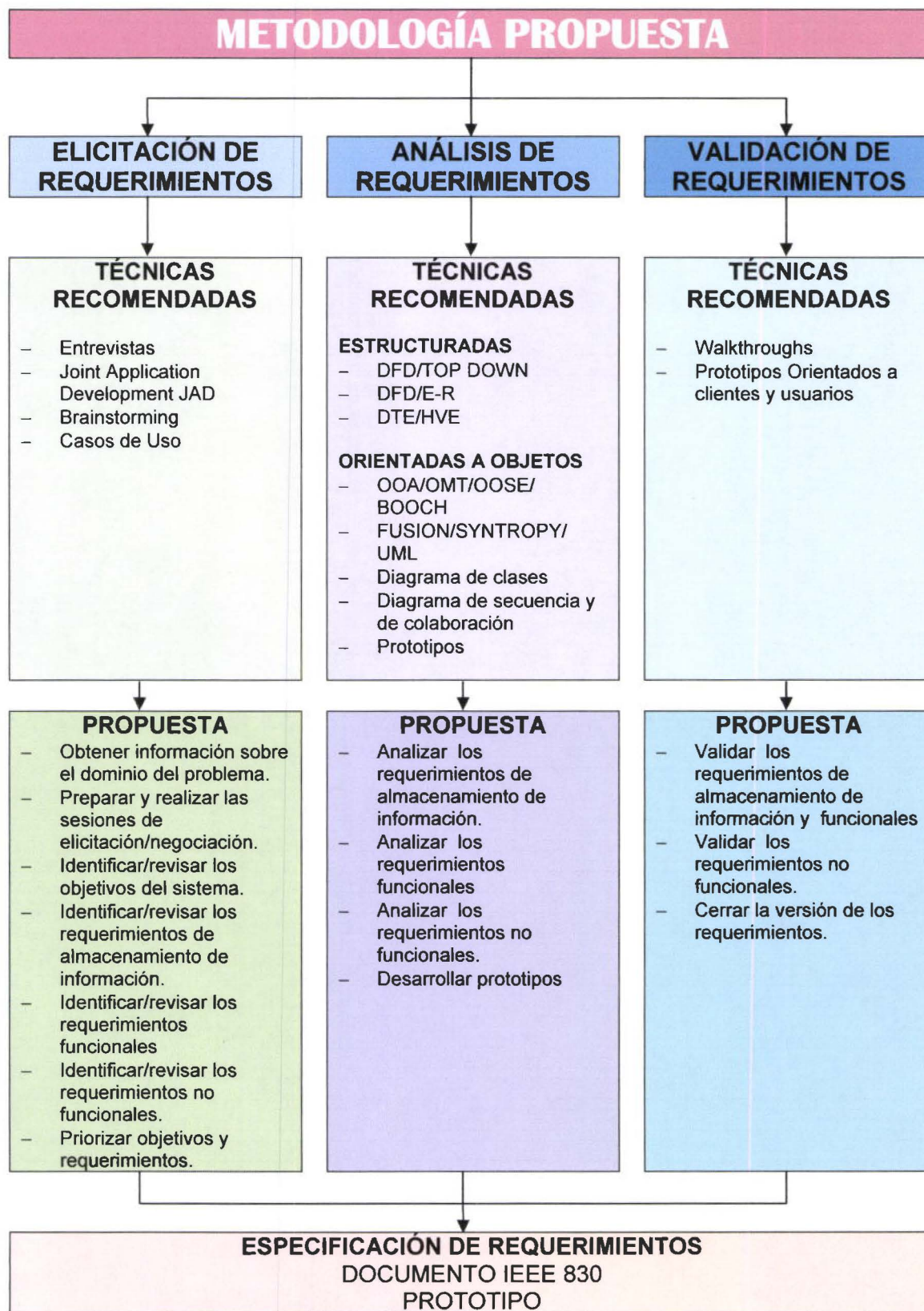


Fig. 4.1. Metodología Propuesta

4.1. ELICITACIÓN DE REQUERIMIENTOS

4.1.1. Introducción

Se considera la elicitación como la actividad de la ingeniería de requerimientos en la que se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registrar información sobre sus necesidades. Estas interacciones pueden realizarse para recopilar información o para resolver conflictos que se hayan detectado en la información recopilada.

4.1.2. La elicitación de requerimientos

La elicitación de requerimientos no se ha considerado parte del ciclo de vida de desarrollo de software hasta hace relativamente pocos años. Se solía asumir que el cliente proporcionaba los requerimientos, de forma que el ciclo de vida comenzaba siempre por el análisis de unos requerimientos ya dados, ya que las actividades de elicitación y validación no se consideraban necesarias. Cuando se ha detectado que los problemas en los requerimientos son uno de los principales factores de los fracasos de los proyectos software, es cuando se le ha comenzado a dar importancia al proceso de obtención de esos requerimientos.

La elicitación de requerimientos es considerarla como la actividad de la ingeniería de requerimientos en la que los ingenieros interactúan con el resto de participantes para obtener, registrar, y si es necesario negociar, los requerimientos que deberá satisfacer el sistema a desarrollar desde el punto de vista de clientes y usuarios.

Dentro del modelo de procesos iterativo, las actividades de elicitación pueden realizarse varias veces. En la primera iteración, la elicitación consistirá básicamente en obtener la mayor cantidad posible de información, asumiendo

que lo más probable es que dicha información sea incompleta, ambigua y contenga contradicciones.

En las siguientes iteraciones, la elicitación consistirá principalmente en la resolución de conflictos encontrados en la información elicitada durante las actividades de análisis o de validación.

4.1.3. Problemas de la elicitación de requerimientos

| PROBLEMA | DESCRIPCIÓN |
|-----------------|--|
| De Articulación | <ul style="list-style-type: none"> • Los clientes y usuarios pueden ser conscientes de sus necesidades pero no ser capaces de expresarlas apropiadamente. • Los clientes y usuarios pueden no ser conscientes de sus necesidades y puede que no entiendan cómo la tecnología puede ayudarles. • Los usuarios pueden no expresar sus necesidades por miedo a parecer incompetentes antes los demás o porque los desarrolladores juegan un papel excesivamente dominante en el proceso. • Los clientes pueden no llegar a tomar decisiones porque no pueden prever las consecuencias de su decisión o porque no entienden las alternativas que se les plantea. • Los desarrolladores no escuchan apropiadamente a los clientes y usuarios, bien porque creen haber entendido sus necesidades rápidamente, bien porque se dedican a pensar inmediatamente sobre aspectos de implementación y no se ponen en el lugar de clientes y usuarios. |
| De Comunicación | <ul style="list-style-type: none"> • Los clientes y usuarios y los desarrolladores tienen culturas y vocabularios diferentes, con la posibilidad de que los mismos términos tengan significados distintos. • Las preocupaciones sobre el sistema a desarrollar. Mientras los clientes y usuarios suelen preocuparse por aspectos de alto nivel como facilidad de uso o fiabilidad, los desarrolladores suelen preocuparse por aspectos de bajo nivel como utilización de recursos, algoritmos, etc. • El medio de comunicación que se utilice debe ser entendible por todos los participantes. Se suele utilizar lenguaje natural porque es el único medio de comunicación común a todos los participantes, a pesar de su inherente ambigüedad. La utilización de otro tipo de técnicas como diagramas o lenguajes artificiales puede presentar problemas de comprensión. |

| | |
|----------------------------|--|
| | <ul style="list-style-type: none"> • La comunicación puede verse afectada también por sus aspectos puramente sociales. El ingeniero de requerimientos debe ser capaz de comunicarse y tratar con todo tipo de personas y ser capaz de manejar conflictos personales y políticos. |
| De limitaciones cognitivas | <ul style="list-style-type: none"> • El ingeniero de requerimientos debe tener un conocimiento adecuado del dominio del problema y no hacer suposiciones sobre ello, al igual que los clientes y usuarios no deben hacer suposiciones sobre aspectos tecnológicos. • Cuando los problemas son grandes y complejos, algunas personas tienden a hacer simplificaciones no válidas, a ignorar las partes más complejas o a centrarse únicamente en los aspectos que más conocen o que más les afectan. |
| De conducta humana | <ul style="list-style-type: none"> • Puede haber conflictos y ambigüedades en los roles que cada persona debe jugar en el proceso de elicitación. Dentro del grupo de clientes y usuarios, algunos pueden pensar que, aunque conozcan ciertas necesidades o ciertos aspectos importantes, es responsabilidad de otros participantes más afectados el hacerlas explícitas, con lo que el resultado final es que nadie dice nada. • La suposición o el temor a que el sistema a desarrollar cambie su forma de trabajar o incluso ponga en peligro su puesto de trabajo, pueden provocar que algunos usuarios retengan información o incluso saboteen el desarrollo. |
| Técnicos | <ul style="list-style-type: none"> • El software tiene que resolver problemas cada vez más complejos, por lo que sus requerimientos son también cada vez más complejos. • Los requerimientos cambian en el tiempo, y a medida que los clientes y usuarios van conociendo sus propias necesidades y las posibilidades que les ofrece la tecnología se puede reconsiderar decisiones anteriores o descubrir nuevas necesidades. • El hecho de que el hardware y el software cambian rápidamente, haciendo asequibles requerimientos que antes eran inabordables por su complejidad o por su costo. • En algunos sistemas puede haber muchas fuentes de requerimientos, por lo que a más de consultar con los clientes y usuarios, se debe hacerlo con técnicos, personal de mantenimiento, consultar normativas, estándares, etc. • Es necesario tener también en cuenta que los sistemas novedosos requieren un esfuerzo mucho mayor de elicitación, y que las fuentes de información dependerán de la naturaleza del sistema a desarrollar. |

Tabla 4.1. Problemas de la elicitación de requerimientos

4.1.4. Propuesta metodológica para la elicitación de requerimientos

La propuesta metodológica para la elicitación de requerimientos, propone una serie de tareas a realizar y productos a obtener (tanto internos como externos o entregables) durante la realización de la actividad de elicitación de requerimientos. Esta metodología es una evolución de las ya existentes.

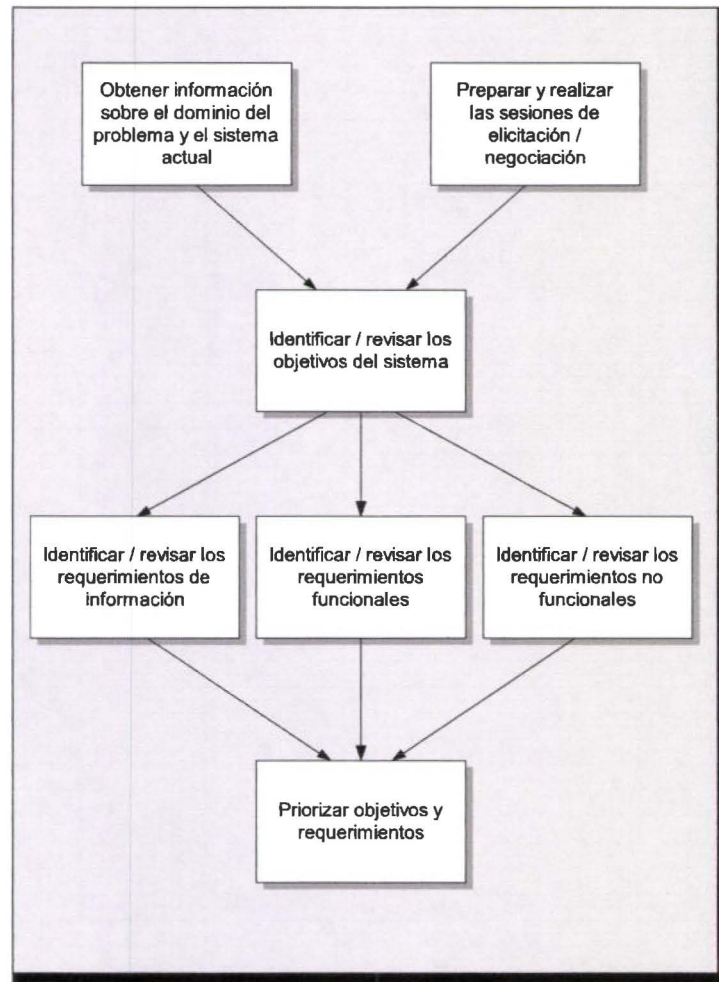


Fig. 4.2. Tareas de elicitación de requerimientos

En la propuesta se contemplan las siete tareas que pueden verse en la figura 4.2, en la que se propone un posible orden de realización orientativo.

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

El objetivo principal de esta tarea es conocer el dominio del problema lo mejor posible. Las razones para ello son varias:

1. El ingeniero de requerimientos debe conocer el lenguaje de clientes y usuarios para poder comunicarse con ellos. Cada dominio de problemas posee un vocabulario propio que es necesario conocer.
2. El ingeniero de requerimientos debe evitar utilizar sus propios esquemas y categorías mentales a la hora de obtener la información, ya que esto puede dificultar la comunicación. Debe aprender a pensar en los términos en los que lo hacen clientes y usuarios. Una herramienta muy útil para ello es realizar un estudio del sistema actual, en caso de que exista siguiendo el formato establecido el estándar IEEE 1362.
3. Mantener sesiones de elicitación sin conocer el dominio del problema puede provocar que los malentendidos o las preguntas continuas sobre el significado de los términos empleados por clientes y usuarios hagan que la confianza hacia el equipo de ingeniería de requerimientos se vea deteriorada enormemente, provocando cierta reticencia a participar e involucrarse en el proyecto.

Los productos resultantes de la realización de esta tarea son productos internos resultado de la búsqueda de información sobre el dominio del problema y la situación del sistema actual: modelos del sistema actual, documentación de la organización, resultados de entrevistas iniciales, resultados de cuestionarios exploratorios, documentación de desarrollos previos sobre el mismo dominio de problemas, información proveniente de expertos, etc.

Las técnicas que pueden utilizarse para realizar esta tarea son varias: recopilación de documentación, entrevistas, reuniones en grupo, cuestionarios,

inmersión en el negocio del cliente, aprendizaje del negocio, modelado del sistema actual, etc.

4.1.4.2. Tarea 2: Preparar y realizar las sesiones de elicitación/negociación

El objetivo de esta tarea es conocer las necesidades de clientes y usuarios y resolver los conflictos identificados en las actividades de análisis de iteraciones previas. Es la tarea más crítica, ya que en ella es donde existe una mayor interacción personal entre el equipo de ingeniería de requerimientos y los clientes y usuarios, por lo que una adecuada selección de los participantes es crucial.

Los productos resultantes de esta tarea, al igual que en la tarea anterior, son también internos y suelen componerse de notas tomadas durante las sesiones, transcripciones o actas de las sesiones, grabaciones de audio o vídeo, etc. Si se han resuelto conflictos durante las sesiones, se pueden considerar como productos dichos conflictos resueltos junto con la probable necesidad de cambios.

Las técnicas que pueden emplearse para la realización de esta tarea son las técnicas de elicitación que se utilizan frecuentemente.

4.1.4.3. Tarea 3: Identificar/revisar los objetivos del sistema

El objetivo de esta tarea es conocer por qué se acomete el desarrollo, y por lo tanto conocer qué objetivos se esperan alcanzar mediante el sistema software a desarrollar. En la primera iteración se realizará una primera identificación de los objetivos. En las iteraciones posteriores puede que sea necesario revisarlos si se han identificado conflictos que les afecten.

Esta información puede que haya sido dada previamente al comienzo del desarrollo, puede que se haya manifestado de forma explícita durante las sesiones de elicitación o puede que haya que deducirla de la información recopilada durante dichas sesiones.

La idea básica es ir obteniendo los requerimientos como un refinamiento de los objetivos, de forma que la existencia de un requisito esté siempre justificada como una necesidad para alcanzar uno o más objetivos.

4.1.4.4. Tarea 4: Identificar/revisar los requerimientos de almacenamiento de información.

El objetivo de las tareas 4, 5 y 6 es identificar, o revisar en función de posibles conflictos, a partir de la información obtenida en las tareas anteriores. La división en tres tareas se ha realizado por simplicidad, no porque se asuma que esa deba ser la secuencia de realización, ya que habitualmente se realizan en paralelo.

En esta tarea en concreto se deben identificar o revisar los requerimientos de almacenamiento de información que deberá satisfacer el sistema. Normalmente estos requerimientos son la respuesta a la pregunta ¿qué información, relevante para los objetivos de su negocio, deberá almacenar el sistema?.

Este tipo de requerimientos no suele incluirse como un grupo separado en las metodologías actuales, sin embargo en este trabajo, es importante identificarlos claramente.

Otros enfoques, no incluyen estos requerimientos porque pueden deducirse de los requerimientos funcionales, especialmente si se utilizan casos de uso. Aunque es cierto que la información que debe almacenar el sistema puede deducirse de los casos de uso, es conveniente hacerla explícita.

Por lo tanto, los productos de esta tarea serían los requerimientos de almacenamiento de información.

4.1.4.5. Tarea 5: Identificar/revisar los requerimientos funcionales

En esta tarea se deben identificar o revisar los requerimientos funcionales que deberá satisfacer el sistema, para lo que se ha optado por la utilización de los casos de uso. Estos requerimientos suelen obtenerse como respuesta a la pregunta ¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio? o ¿qué debe permitir el sistema hacer a los usuarios con la información almacenada?.

Además de los casos de uso, en esta tarea es necesario identificar y describir a los actores del sistema. Otro aspecto importante es la determinación del ámbito del sistema, es decir qué aspectos serán responsabilidad del sistema y qué aspectos se gestionarán manualmente o por otro procedimiento. La utilización de los diagramas de casos de uso, permite de forma sencilla especificar claramente qué queda dentro y qué queda fuera del ámbito del sistema.

Los productos de esta tarea son los diagramas de casos de uso y las especificaciones tanto de los actores como de los casos de uso expresados.

4.1.4.6. Tarea 6: Identificar/revisar los requerimientos no funcionales

En esta tarea se deben identificar o revisar los requerimientos no funcionales del sistema, normalmente relacionados con aspectos técnicos o legales: comunicaciones, interfaces con otros sistemas, fiabilidad, entorno de desarrollo, portabilidad, etc.

Los productos de esta tarea son los requerimientos no funcionales

4.1.4.7. Tarea 7: Priorizar objetivos y requerimientos

En esta última tarea se deben asignar prioridades a los objetivos y requerimientos estableciendo su importancia y su urgencia, de forma que en el caso de que se desarrolle incrementalmente se tengan los criterios suficientes para saber qué requerimientos deben implementarse en cada versión que se vaya entregando.

Los productos de esta tarea son los objetivos y requerimientos identificados/revisados en las tareas anteriores con su prioridad establecida.

4.1.4.8. El documento de requerimientos del sistema

En la propuesta metodológica, el producto final entregable de las actividades de elicitación de requerimientos son los documentos IEEE 830 y 1362 en sus versiones iniciales.

4.2. ANÁLISIS DE REQUERIMIENTOS

4.2.1. Introducción

El término análisis de requisitos se ha utilizado durante bastante tiempo para referirse al conjunto global de las actividades relacionadas con los requisitos en el desarrollo de software, al asumirse que eran proporcionados directamente por el cliente, por lo que no era labor del equipo de desarrollo la elicitación de dichos requisitos ni había necesidad de una validación por parte del cliente, ya que era él mismo el que los había producido.

Las técnicas de modelado conceptual son la principal herramienta del análisis de requerimientos. Tradicionalmente se han considerado las técnicas estructuradas y las orientadas a objetos como dos formas incompatibles de modelar sistemas de información.

4.2.2. Propuesta metodológica para el análisis de requerimientos

La propuesta metodológica para el análisis de requerimientos, se contemplan las cuatro tareas que pueden verse en la figura 4.3, en la que se propone un posible orden de realización orientativo en el que las tareas de analizar los tres tipos de requerimientos se realizarían en paralelo. En las siguientes secciones se describen cada una de ellas.

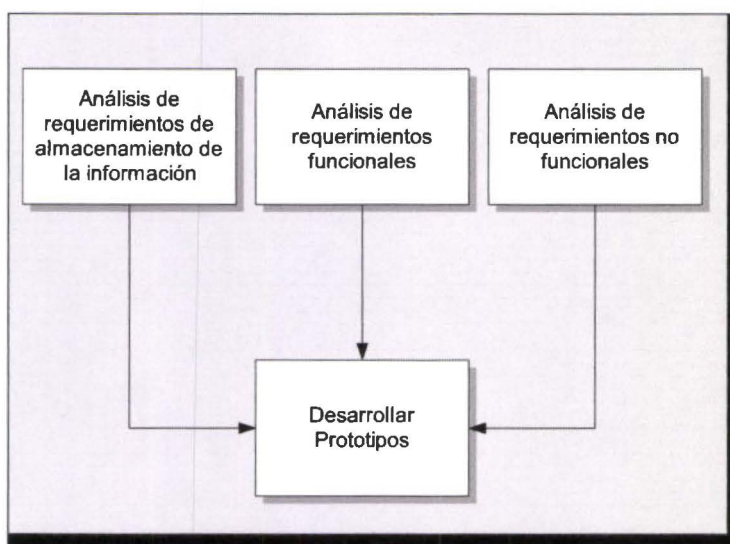


Fig. 4.3. Tareas de análisis de requerimientos

4.2.2.1. Tarea 1: Analizar los requerimientos de almacenamiento de información

El objetivo principal de esta tarea, al igual que el de la siguiente, es descubrir conflictos en los requerimientos profundizando en el conocimiento del problema y estableciendo las bases para un futuro diseño del sistema. En el caso concreto de esta tarea, el objetivo es descubrirlos en los requerimientos de almacenamiento de información.

La forma habitual de alcanzar estos objetivos es mediante la construcción de modelos abstractos. En esta propuesta se recomienda la utilización de técnicas de modelado orientadas a objetos, aunque también podrían utilizarse técnicas estructuradas, ya que las diferencias entre ambas no son insalvables.

Los productos resultantes de la realización de esta tarea son el modelo estático del sistema, compuesto por los diagramas de tipos y la descripción textual de los elementos que lo integran, y los posibles conflictos que se detecten al construir el modelo.

Las técnicas que pueden utilizarse para realizar esta tarea son los diagramas de tipos.

4.2.2.2. Tarea 2: Analizar los requerimientos funcionales

Esta tarea es similar a la anterior, con la diferencia de que se centra en los requerimientos funcionales, expresados mediante casos de uso, en lugar de hacerlo en los requerimientos de almacenamiento de información.

Para analizar los requerimientos funcionales lo habitual es construir modelos funcionales y, si se considera oportuno, modelos dinámicos. Por lo tanto, los productos resultantes de la realización de esta tarea son los modelos funcional y dinámico y los conflictos que se hayan detectado al construir ambos modelos.

4.2.2.3. Tarea 3: Analizar los requerimientos no funcionales

Esta tarea tiene también como objetivo descubrir conflictos en los requerimientos, en este caso en los requerimientos no funcionales. Sin embargo, a diferencia de las dos tareas anteriores, las escasas propuestas de modelado que integran este tipo de requerimientos aun no pueden considerarse satisfactorias.

La naturaleza heterogénea de este tipo de requerimientos hace que su análisis sea necesario realizarlo mediante una lectura detenida de su contenido, y combinando esta lectura con la experiencia, detectar posibles conflictos como la imposibilidad técnica de la implementación de ciertos requerimientos, la

necesidad de optar por unas características u otras (por ejemplo flexibilidad frente a eficiencia), etc.

Normalmente, estos requerimientos se tendrán en consideración durante el diseño de la arquitectura del sistema, lo que probablemente provocará iteraciones entre el resto del desarrollo y la fase de ingeniería de requerimientos.

Los productos resultantes de esta tarea son, por lo tanto, aquellos conflictos que se hayan detectado al realizar el análisis de los requerimientos no funcionales.

4.2.2.4. Tarea 4: Desarrollar prototipos

El objetivo de esta tarea es el desarrollo de prototipos que puedan utilizarse durante la elicitación o validación de los requerimientos, y que por lo tanto deben centrarse en la interfaz de usuario, independientemente de que su naturaleza sea de usar y tirar o evolutiva.

Las actuales herramientas de desarrollo rápido de aplicaciones (RAD, Rapid Application Development) permiten la construcción rápida de interfaces de usuario que no tienen porqué desecharse una vez utilizadas para obtener información de los clientes.

Otra posibilidad consiste en la prototipación de la funcionalidad de la aplicación mediante prototipos generados a partir de lenguajes de especificación formal como Z.

Estos prototipos funcionales generados automáticamente no suelen tener interfaces de usuario parecidas a las que puede tener el sistema final, por lo que su uso se restringe a la comprobación de algunas propiedades del modelo

por parte de los ingenieros de requerimientos, sin que puedan servir para la validación o elicitación con los clientes y usuarios.

4.2.3. El documento de análisis del sistema

En la propuesta metodológica, el Documento de Especificación de Requerimientos del Software IEEE 830-1998 y el Documento de Concepto de Operaciones propuesta por el IEEE 1362-1998, son los principales productos entregables, junto con el prototipo si se considera oportuna su realización, de las actividades de análisis de requerimientos. Los conflictos detectados pueden registrarse en un documento aparte. Los documentos en versiones más refinadas que las iniciales y el prototipo en su versión inicial.

4.3. VALIDACIÓN DE REQUERIMIENTOS

4.3.1. Introducción

Dentro del entorno metodológico propuesto, la validación de requerimientos se considera como la actividad de la ingeniería de requerimientos en la que clientes y usuarios, junto con la ayuda de los ingenieros de requerimientos, revisan los productos obtenidos durante las actividades anteriores para confirmar que realmente reflejan sus necesidades y que definen el producto deseado.

La validación de requerimientos es otra de las actividades de la ingeniería de requerimientos, que junto con la elicitación, han recibido tradicionalmente poca atención. La razón, al igual que en el caso de la elicitación, ha sido la suposición de que los requerimientos eran proporcionados directamente por el cliente, con lo que, implícitamente, se asumían validados.

La ingeniería de requerimientos actual asume como una necesidad la participación de los ingenieros de requerimientos tanto en las actividades de

elicitación, para ayudar a clientes y usuarios a encontrar y describir sus necesidades, como en las actividades de validación, para ayudar a clientes y usuarios a comprobar que los requerimientos elicitados y analizados resultantes del proceso describen realmente todas sus necesidades correctamente.

4.3.2. Validación y verificación de requerimientos

Los términos validación y verificación (V&V) suelen usarse a veces como sinónimos o aparecer unidos en la mayor parte de la bibliografía sobre ingeniería de software e ingeniería de requerimientos.

Generalmente, la V&V (sobre todo la validación) suele asociarse a los distintos tipos de técnicas de prueba, principalmente sobre el producto final.

La idea de considerar la V&V como actividades prácticamente inseparables se ve reforzada por el tratamiento que se les da en varios estándares, por ejemplo los del IEEE [IEEE 1997] como el 1012 (Standard for Software Verification and Validation Plans), el 1059 (Guide for Software Verification and Validation Plans) o dentro de las actividades definidas en el 1074 [IEEE 1995a, IEEE 1995b].

En el glosario de términos del IEEE [IEEE 1990] aparecen ambos términos en una única entrada:

“verificación y validación (1): el proceso de determinar si los requerimientos para un sistema o componente son completos y correctos, los productos de cada fase de desarrollo satisfacen los requerimientos o condiciones impuestas por la fase previa y el sistema o componente final es acorde con los requerimientos especificados”.

En la que se abordan tres aspectos fundamentales: la calidad de los requerimientos, la calidad de los productos intermedios y las pruebas de

aceptación del sistema. Curiosamente, en el mismo glosario aparecen también los términos V&V como entradas separadas que, sin embargo, no coinciden completamente con la definición anterior al dejar fuera los aspectos relativos a los requerimientos y referirse únicamente al sistema a desarrollar o a uno de sus componentes.

“validación (2): el proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si satisface los requerimientos especificados”.

“verificación (2): (a) el proceso de evaluar un sistema o componente para determinar si los productos de una fase de desarrollo dada satisfacen las condiciones impuestas al comienzo de la fase. (b) prueba formal de la corrección de un programa”.

En este trabajo, se entenderá por validación de requerimientos el conjunto de actividades encaminadas a llegar a un acuerdo entre todos los participantes en el que se ratifique que los requerimientos elicitados y analizados representan realmente las necesidades de clientes y usuarios y que, por lo tanto, deberían llevar a la construcción de software útil.

Por verificación se entenderá el conjunto de actividades relacionadas con la calidad de las especificaciones de requerimientos respecto a las propiedades deseables.

El mecanismo más importante para la verificación de los requerimientos que se ha contemplado en este trabajo es el análisis de requerimientos, cuyo principal objetivo es descubrir posibles conflictos en los requerimientos, Otras técnicas complementarias de verificación como las inspecciones, las revisiones técnicas o las auditorías no son tratadas en este trabajo.

4.3.3. El compromiso de la validación de requerimientos

La validación de los requisitos representa un compromiso entre los clientes y los desarrolladores. En [Wiegiers 1999] se exponen algunos de los problemas que puede producir un acuerdo si cada parte lo entiende según sus intereses.

Los clientes pueden considerar el acuerdo como un ritual necesario pero sin sentido, que es aceptado porque es la única forma de que comience el desarrollo. Esta actitud suele ir acompañada de cierto desinterés en la revisión de las especificaciones de requisitos y por lo tanto es probable que surjan problemas durante el desarrollo. Si el cliente no recibe el producto que esperaba, suele achacarlo a la falta de profesionalidad de los desarrolladores, en los que confió para que interpretaran correctamente sus necesidades.

Los gestores del proyecto pueden considerar el acuerdo como una forma de congelar los requisitos, y de esta forma rechazar cualquier solicitud de cambio posterior que conllevara la necesidad de cambios en la planificación o en la asignación de recursos, argumentando que el cliente debería haberlo pensado antes de validar los requisitos.

Para evitar posibles interpretaciones erróneas de lo que puede significar llegar a un acuerdo sobre un documento de requisitos, en [Wiegiers 1999] se propone considerar este acuerdo como un punto de partida en el proceso de desarrollo e incluir un texto similar al siguiente en los documentos de requisitos validados:

“Los abajo firmantes acordamos que este documento representa nuestro conocimiento actual de los requisitos del proyecto al día de hoy. Los futuros cambios de este documento se realizarán de acuerdo al procedimiento de cambio definido para el proyecto. Entendemos que los cambios aprobados podrán requerir que se renegocien los acuerdos actuales sobre costos, recursos asignados y fecha de entrega del proyecto”.

4.3.4. Propuesta metodológica para la validación de requerimientos

En la propuesta se contemplan tres tareas que pueden verse en la figura 4.4, en la que se propone un posible orden de realización orientativo en el que las dos tareas en las que se validan los requerimientos se realizarían en paralelo.

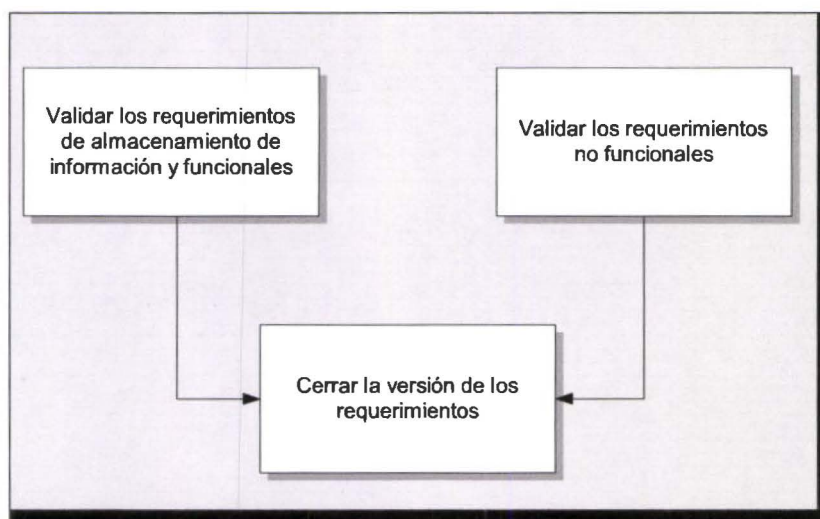


Fig. 4.4. Tareas de validación de requerimientos

4.3.5. Tarea 1: Validar los requerimientos de almacenamiento de información y funcionales

El objetivo de esta tarea es validar los requerimientos de almacenamiento de información y funcionales, es decir, asegurarse de que representan realmente las necesidades de clientes y usuarios.

Se han agrupado los dos tipos de requerimientos en una única tarea porque, su validación mediante walkthroughs asistidos con prototipos de interfaz de usuario permite una validación conjunta de forma sencilla guiada por los requerimientos funcionales descritos como casos de uso, teniendo siempre en cuenta las relaciones de rastreabilidad establecidas entre los requerimientos de almacenamiento de información y los requerimientos funcionales.

Los productos resultantes de esta tarea serían los requerimientos de almacenamiento de información, los requerimientos funcionales y el prototipo validados total o parcialmente. En el caso de que se descubran conflictos durante la validación, éstos serían también productos de esta tarea y deberían resolverse en nuevas sesiones de elicitación/negociación.

Como ya se ha comentado, las técnicas recomendadas para la realización de esta tarea son una combinación de walkthrough, de los casos de uso y de utilización de prototipos de interfaz de usuario para ayudar a los clientes y usuarios a revisar la documentación de requerimientos y, en el caso de que aparezcan conflictos, se pueden utilizar plantillas.

4.3.6. Tarea 2: Validar los requerimientos no funcionales

El objetivo de esta tarea es validar los requerimientos no funcionales. Sus productos, de forma similar a la tarea anterior, serían los requerimientos no funcionales validados total o parcialmente, y los posibles conflictos que pudieran aparecer.

A diferencia de la tarea anterior, la única técnica que parece aplicable para realizar esta tarea es una revisión por parte de los clientes y usuarios, ayudados por los ingenieros de requerimientos para aclarar las posibles dudas que surjan durante la revisión.

Al igual que en las actividades de análisis, en las que los requerimientos no funcionales no suelen contemplarse en las técnicas de modelado, en la validación tampoco parece existir una forma de integrarlos en un proceso de walkthrough o de prototipado.

4.3.7. Tarea 3: Cerrar la versión de los requerimientos

Si no han aparecido nuevos conflictos durante el proceso de validación, 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 [Wiegiers 1999].

El producto de esta actividad es, por lo tanto, una versión cerrada de los requerimientos y del prototipo.

4.4. ESPECIFICACIÓN DE REQUERIMIENTOS

Pressman escribe: "El término especificación significa distintas cosas para diferentes personas. Una especificación puede ser un documento escrito, un modelo gráfico, un modelo matemático, una colección de escenarios de uso, un prototipo o una combinación de lo anteriormente citado". (Pressman, 2002: 173).

La especificación de requerimientos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales, para la recolección, clasificación, identificación, priorización y especificación de los requerimientos del software.

La IEEE Std. 830-1998 normaliza la creación de las Especificaciones de Requerimientos de Software y es el documento resultante del proceso de Ingeniería de Requerimientos.

Para la generación del documento escrito IEEE 830 y aplicando las técnicas recomendadas se obtendrá un documento combinado con descripciones en

lenguaje natural y modelos gráficos como los diagramas UML, lo que generará una mejor alternativa para los desarrolladores y especialmente se convertirá en un documento contractual para las partes y entendible para todos los stakeholders involucrados en el sistema.

4.5. COMPARACIÓN CON LAS METODOLOGÍAS EXISTENTES

| | Entendimiento del dominio y contexto de la aplicación | Recolección y clasificación de requerimientos | Reuso de requerimientos | Resolución de conflictos, priorización y validación de requerimientos | Cierre |
|------------------------------|--|--|--|--|---------|
| METODOLOGÍA ANCORA | Entrevistas Encuestas Observación Redes Semánticas | Prototipado rápido Objetos semánticos | Consultas | Reunión de reflexión y diseño | Reunión |
| METODOLOGÍA DORCU | Elicitación de requerimientos ABIERTA No específica | Análisis de requerimientos ABIERTA No específica | Especificación de requerimientos ABIERTA No específica | Validación y certificación de los requerimientos ABIERTA No específica | |
| METODOLOGÍA AGIL | Elicitación de requerimientos Entrevistas JAD Brainstorming | Especificación de requerimientos Abierta No específica | Gestión de requerimientos Abierta No específica | | |
| METODOLOGÍA PROPUESTA | Elicitación de requerimientos Entrevistas JAD Brainstorming Casos de Uso | Análisis de requerimientos Diagramas UML Prototipos | Validación de requerimientos Walkthroughs Prototipos | Especificación de requerimientos IEEE 830 | |

Fig. 4.5. Comparación con las metodologías existentes

Como se indica en la figura 4.5, las metodologías evaluadas no indican las técnicas a utilizarse en todas las fases de la Ingeniería de Requerimientos, lo que sí es realizado en la metodología propuesta, así:

- Elicitación de requerimientos: Entrevistas, JAD, Brainstorming, Casos de uso.
- Análisis de requerimientos: Diagramas UML, Prototipos
- Validación de requerimientos: Walkthroughs, Prototipos

Además, se propone que el documento correspondiente a la especificación de requerimientos utilice el formato establecido en el IEEE 830, con la finalidad de

estandarizar el documento resultante y sea entendible para todos los stakeholders involucrados en el desarrollo del producto software.

4.6. APLICACIÓN DE LA METODOLOGÍA PROPUESTA

4.6.1. Elicitación de Requerimientos

Tarea 1: Obtener información sobre el dominio del problema

La Escuela Fiscal "San Pedro del Norte" se encuentra ubicada en la Parroquia Calderón, debido a su régimen, su funcionamiento se enmarca dentro de la Ley de Educación y su Reglamento General, además de sus Reglamentos Internos y Normas correspondientes.

La matriculación de los estudiantes la realiza cada Docente en sus respectivos grados y paralelos a los cuales han sido asignados por las Autoridades correspondientes.

Igualmente las notas son registradas por cada uno de los docentes, y al final del año lectivo entregan a Secretaría los respectivos archivos de respaldo.

Tarea 2: Preparar y realizar las sesiones de elicitación/negociación

Requerimientos obtenidos de acuerdo a las necesidades de los clientes/usuarios.

1. El Sistema debe almacenar la información de notas de los alumnos ingresados por los profesores.
2. El Sistema debe almacenar la información de matrículas de los alumnos ingresados por la secretaria.
3. El sistema estudiantil deberá registrar los accesos de los usuarios.
4. La navegación en el menú debe ser de fácil entendimiento.

5. El sistema debe garantizar alto desempeño, confiabilidad.
6. El sistema debe garantizar seguridad permitiendo el acceso solamente por medio de la clave personal.
7. Fácil mantenimiento para lo cual se generará la documentación completa del sistema.

Tarea 3: Identificar/revisar los objetivos del sistema

En la siguiente tabla se detallan los objetivos del sistema

| | |
|--------------------|--|
| OBJ-01 | Gestionar los estudiantes |
| Descripción | El sistema deberá gestionar los estudiantes pertenecientes a la Escuela "San Pedro del Norte": matrículas, notas, etc. |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

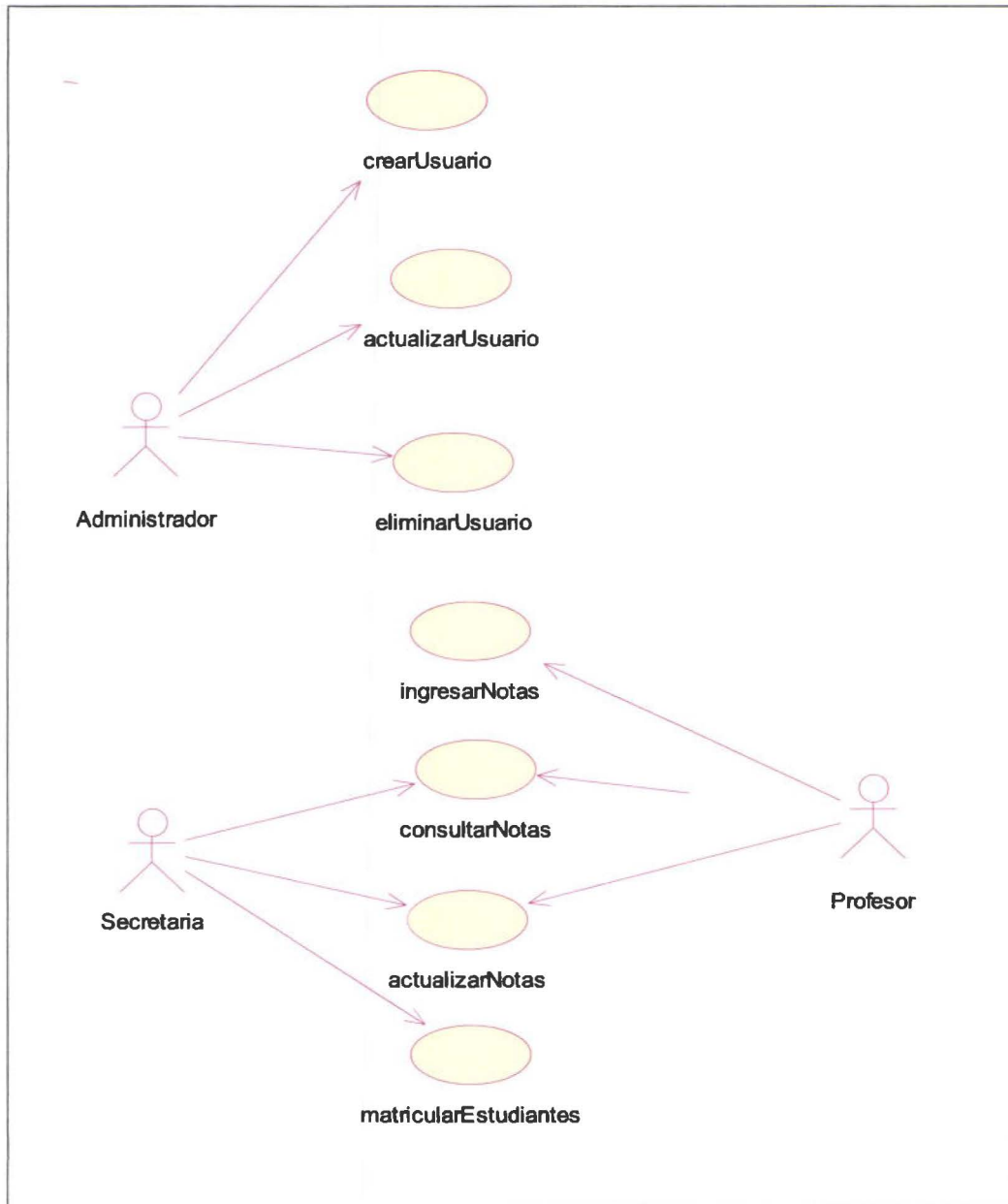
| | |
|--------------------|---|
| OBJ-02 | Gestionar los usuarios del sistema |
| Descripción | El sistema deberá gestionar los usuarios del sistema: perfil, login, password, etc. |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Tarea 4: Identificar/revisar los requerimientos de almacenamiento de información

| | |
|---------------------------------|---|
| RI-01 | Información sobre estudiantes |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Apellido Paterno - Apellido Materno - Nombres - Fechas de Nacimiento - Género |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|--|
| RI-02 | Información sobre notas |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a las notas de los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Curso - Período - Asignatura - Nota 1 - Nota 2 - Nota 3 |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|---|
| RI-03 | Información sobre usuarios |
| Objetivos asociados | OBJ-02 Gestionar los usuarios del sistema |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los usuarios. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Perfil - Login - Password - Nombres - Apellidos - E-mail - Teléfono |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

Tarea 5: Identificar/revisar los requerimientos funcionales**Requerimientos funcionales del sistema****Fig. 4.6. Casos de uso del sistema**

| | | |
|---------------------------------|--|--|
| RF-01 | Alta de estudiante | |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes | |
| Requerimientos asociados | RI-02 Información sobre estudiantes | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite su ingreso como estudiante. | |
| Precondición | El solicitante no es estudiante y tiene su documentación disponible | |
| Secuencia normal | Paso | Acción |
| | 1 | La secretaria solicita al sistema comenzar el proceso de alta de un estudiante |
| | 2 | El sistema solicita los siguientes datos del nuevo estudiante: Código (Número secuencial), Apellido Paterno, Apellido Materno, Nombres, Fecha de Nacimiento, Género. |
| | 3 | La Secretaria solicita los datos requeridos y la documentación al nuevo estudiante. |
| | 4 | La secretaria comprueba que los datos del nuevo estudiante coinciden con los de la documentación aportada. |
| | 5 | La Secretaria proporciona los datos requeridos y solicita al sistema que los almacene |
| | 6 | El sistema almacena los datos proporcionados e informa a la Secretaria de que el proceso ha terminado con éxito |
| Postcondición | El solicitante es alumno del establecimiento | |
| Excepciones | Paso | Acción |
| | 4 | Si la documentación aportada no es correcta, la secretaria cancela la operación, a continuación este caso de uso termina |
| | 5 | Si el sistema detecta que el nuevo estudiante ya está registrado, el sistema informa de la situación a la secretaria permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa. |
| | 5 | Si la secretaria solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina |
| Rendimiento | Paso | Cota de tiempo |
| | 4 | 5 Segundos |
| Frecuencia esperada | 100 veces/día | |
| Estabilidad | Alta | |
| Comentarios | La frecuencia será mucho mayor durante los meses determinados para matrículas, probablemente 200 veces/día | |

| | | |
|---------------------------------|--|---|
| RF-02 | Alta de usuario | |
| Objetivos asociados | OBJ-02 Gestionar los usuarios | |
| Requerimientos asociados | RI-03 Información sobre usuarios | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se solicite su ingreso como usuario. | |
| Precondición | El solicitante no es usuario y tiene su documentación disponible | |
| Secuencia normal | Paso | Acción |
| | 1 | El Administrador solicita al sistema comenzar el proceso de alta de un usuario |
| | 2 | El sistema solicita los siguientes datos del nuevo usuario: Perfil, Login, Password, Nombres, Apellidos, e-mail, teléfono |
| | 3 | El Administrador solicita los datos requeridos. |
| | 4 | El Administrador proporciona los datos requeridos y solicita al sistema que los almacene |

| | | |
|----------------------------|--|--|
| | 5 | El sistema almacena los datos proporcionados e informa al Administrador de que el proceso ha terminado con éxito |
| Postcondición | El solicitante es alumno del establecimiento | |
| Excepciones | Paso | Acción |
| | 5 | Si el sistema detecta que el nuevo usuario ya está registrado, el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa. |
| | 5 | Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina |
| Rendimiento | Paso | Cota de tiempo |
| | 4 | 5 Segundos |
| Frecuencia esperada | 3 veces/día | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Actores

| | |
|--------------------|--|
| ACT-01 | Administrador |
| Descripción | Este actor representa al Administrador del Sistema quien tendrá acceso a todas las opciones del sistema. |
| Comentarios | El Administrador será quien ejerza la Dirección de la Escuela. |

| | |
|--------------------|---|
| ACT-02 | Secretaria |
| Descripción | Este actor representa a la Secretaria |
| Comentarios | La Secretaria tendrá acceso al módulo de Matriculación y Crear Estudiantes. |

| | |
|--------------------|---|
| ACT-03 | Profesor |
| Descripción | Este actor representa a los profesor |
| Comentarios | Los Profesores son los encargados de ingresar las notas de los alumnos de sus respectivos grados. |

Tarea 6: Identificar/revisar los requerimientos no funcionales

| | |
|---------------------------------|---|
| RNF-01 | Copias de seguridad |
| Objetivos asociados | - |
| Requerimientos asociados | - |
| Descripción | El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados. |
| Comentarios | Ninguno |

| | |
|---------------------------------|--|
| RNF-02 | Portabilidad |
| Objetivos asociados | - |
| Requerimientos asociados | - |
| Descripción | El sistema deberá ser fácilmente portable a cualquier navegador. |
| Comentarios | Ninguno |

Tarea 7: Priorizar objetivos y requerimientos

| | |
|--------------------|--|
| OBJ-01 | Gestionar los estudiantes |
| Descripción | El sistema deberá gestionar los estudiantes pertenecientes a la Escuela "San Pedro del Norte": matrículas, notas, etc. |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|--------------------|---|
| OBJ-02 | Gestionar los usuarios del sistema |
| Descripción | El sistema deberá gestionar los usuarios del sistema: perfil, login, password, etc. |
| Estabilidad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|---|
| RI-01 | Información sobre estudiantes |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Apellido Paterno - Apellido Materno - Nombres - Fechas de Nacimiento - Género |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|--|
| RI-02 | Información sobre notas |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a las notas de los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Curso - Periodo - Asignatura - Nota 1 - Nota 2 - Nota 3 |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|---|
| RI-03 | Información sobre usuarios |
| Objetivos asociados | OBJ-02 Gestionar los usuarios del sistema |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los usuarios. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Perfil - Login - Password - Nombres - Apellidos - E-mail - Teléfono |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

4.6.2. Análisis de Requerimientos

4.6.2.1. Prototipos

Ingreso de notas

[|| Ingresar Notas](#) | [Consultar Notas](#) | | [Salir](#)

Seleccione el curso:

Seleccione el Período:

Seleccione la asignatura:

| | ESTUDIANTE | ASIGNATURA | NOTA1 | NOTA2 | NOTA3 | NOTA_FINAL |
|-------------|------------------------------|------------|-------|-------|-------|------------|
| Seleccionar | Landivar Caceres Karla Sofia | Dibujo | 10,00 | 9,00 | 8,75 | 9,25 |

Alumno: Landivar
Caceres Karla Sofia

Nota1:

Nota2:

Nota3:

Crear Usuario

Perfil:

| PERFIL | |
|-------------|---------------|
| Seleccionar | Administrador |
| Seleccionar | Profesor |
| Seleccionar | Secretaria |

Datos del Usuario:

Login:

Password:

Nombres:

Apellidos:

Email:

Telefono:

4.6.3. Validación de Requerimientos

Una vez validados los requerimientos se procede a formular la especificación de requerimientos. La validación se la realizó una reunión de walkthroughs, cuya acta de resultados es la siguiente:

En la Escuela Fiscal San Pedro del Norte ubicada en la Parroquia Calderón perteneciente al Distrito Metropolitano de Quito, hoy 25 de septiembre de 2009, siendo las 10H00 se instala la reunión para la validación de los requerimientos a través de la técnica walkthroughs con la asistencia de las siguientes personas:

- Autora : Grace Ayala
- Presentador : Grace Ayala
- Revisores : Alexandra Yáñez (Directora)
: Patricia Cifuentes (Secretaria)
: Profesores delegados

Nemesio Sánchez

Olmídes Frómata

Marilúz Viera

- Moderadora : Nataly Carrillo
- Registradora : Catalina Enríquez

Previo a esta reunión se les entregó a los revisores la documentación pertinente sobre la cual se iba a trabajar.

Para iniciar la reunión la Moderadora Nataly Carrillo indica a cada uno de los asistentes el rol que van a desempeñar cada uno en común acuerdo con la presentadora/autora.

Grace Ayala en su calidad de autora y presentadora de la reunión hace el respectivo recorrido con detalle sobre el producto requerido, dando lectura a los requerimientos definidos en las etapas anteriores, con la ayuda del prototipo creado.

La señora Alexandra Yáñez indica que el ingreso de notas se lo hará en fechas determinadas y con tiempos límites para ingresar o modificar las notas pasado ese tiempo el profesor deberá realizar una solicitud por escrito a la Dirección y ésta autorizará en su calidad de Directora y Administradora del sistema.

La Secretaria realiza la observación de que en la matriculación de los estudiantes se añada una opción que permita imprimir el carnet del estudiante.

La Lic. Mariluz Viera profesora delegada indica que sería mucho mejor si el sistema permitiera ingresar solo los parciales de cada trimestre y éste automáticamente calcularía los promedios correspondientes.

La Sra. Grace Ayala autora/presentadora indica que las sugerencias realizadas por la Sra. Directora serán consideradas en lo requerimientos y añadidas al

documento de especificación respectivo, mientras la observación realizada por la Profesora será considerada como un requerimiento futuro ya que en la actualidad la Institución no cuenta con un reglamento que unifique los aportes a ser considerados como parte del promedio trimestral, siendo en la actualidad discrecional por parte del profesor.

Una vez terminada la reunión la Sra. Catalina Enríquez registradora lee la presente acta a los participantes quienes aceptan las observaciones y conclusiones realizadas por cada uno de los participantes.

Siendo las 11H53 minutos se clausura la sesión y para constancia de lo actuado firman las partes.

Sra. Grace Ayala
AUTORA REQUERIMIENTOS

Sra. Alexandra Yáñez
DIRECTORA/CLIENTE

Sra. Patricia Cifuentes
SECRETARIA/CLIENTE

Sra. Nemesio Sánchez
PROFESOR/CLIENTE

Sra. Nataly Carrillo
MODERADORA

Sra. Catalina Enríquez
REGISTRADORA

4.6.4. Especificación de Requerimientos

Se utiliza el estándar IEEE-830, el mismo que se detalla a continuación:

1. Introducción

1.1. Propósito

Esta especificación tiene como propósito analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requerimientos que ha de satisfacer el nuevo sistema mediante las técnicas detalladas, el estudio de los problemas de las unidades afectadas y sus necesidades actuales. Además de identificar los requerimientos se deberán establecer prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

1.2. Ámbito del sistema

Los principales objetivos del sistema a desarrollar son la gestión de estudiantes, la gestión de usuarios y el control de notas de los estudiantes con que cuenta la Institución. El futuro sistema se denominará Mi Escuelita Fiscal.

1.3. Definiciones, Acrónimos y abreviaturas

Abreviaturas

| No. | Abreviatura | Significado |
|-----|-------------|--|
| 1 | RI | Requerimiento de almacenamiento de información |
| 2 | RF | Requerimiento Funcional |
| 3 | RNF | Requerimiento no Funcional |
| 4 | OBJ | Objetivo |
| 5 | ACT | Actor |

1.4. Referencias

| No. | Título | Ruta | Versión | Fecha | Autor |
|-----|--|------|---------|-------|-------|
| 1 | Ley de Educación | | | | |
| 2 | Reglamento General a la Ley de Educación | | | | |
| 3 | Reglamento Interno de la Escuela | | | | |

1.5. Visión General del documento

Este documento presenta una descripción general del sistema con el fin de conocer las funciones que debe soportar, los datos asociados, las restricciones impuestas y otros factores que pueda influir en la construcción del mismo.

2. Descripción General

2.1. Perspectiva del Producto

El sistema MI ESCUELITA FISCAL es un sistema independiente, ya que actualmente la escuela San Pedro del Norte lo hace de forma manual.

Por lo tanto el producto deberá realizar:

- Almacenar la información de notas de los alumnos ingresados por los profesores.
- Almacenar la información de matrículas de los alumnos ingresados por la secretaria.
- Registrar los accesos de los usuarios.
- Navegación en el menú de fácil entendimiento.
- Garantizar alto desempeño, confiabilidad.
- Garantizar la seguridad con una clave personal asignado.
- Fácil mantenimiento para lo cual se generará la documentación completa del sistema.

2.2. Funciones del Producto

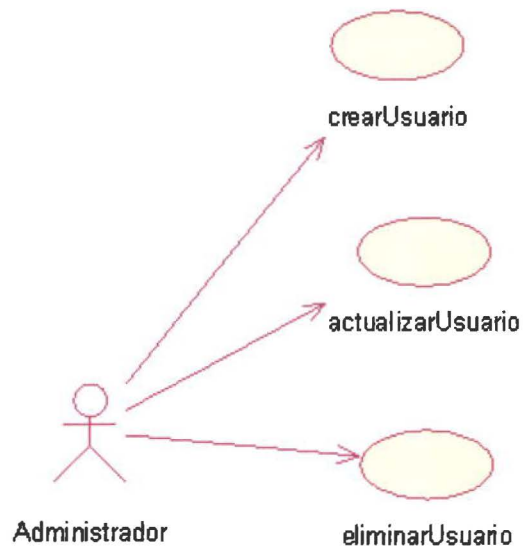


Fig. 4.7. Casos de uso del sistema (Administrador)

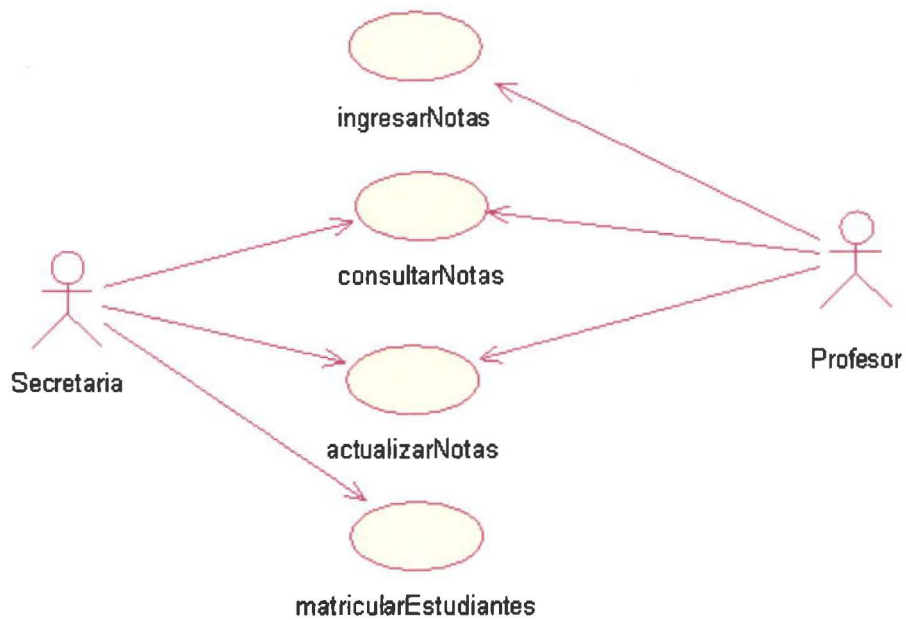


Fig. 4.8. Casos de uso del sistema (Secretaria – Profesor)

2.3. Características de los usuarios

| | |
|-------------------------|--|
| Nombre | Alexandra Yáñez |
| Rol | Directora |
| Categoría profesional | Licenciada en Ciencias de la Educación |
| Responsabilidades | Administradora del sistema |
| Información de contacto | 2598764 – Alexandra_yanez@yahoo.com |
| Aprobación | |

| | |
|-------------------------|---|
| Nombre | Patricia Cifuentes |
| Rol | Secretaría |
| Categoría profesional | Licenciada en Ciencias de la Educación |
| Responsabilidades | Usuario – Matricular estudiantes Actualizar notas Consultar notas |
| Información de contacto | 3285013 – pcifuentes@hotmail.com |
| Aprobación | |

Todos los Profesores serán usuarios del sistema para ingresar, actualizar y consultar las notas.

2.4. Restricciones

En el ingreso de notas se lo hará en fechas determinadas y con tiempos límites para ingresar o modificar las notas pasado ese tiempo el profesor deberá realizar una solicitud por escrito a la Dirección y ésta autorizará en su calidad de Directora y Administradora del sistema.

2.5. Suposiciones y dependencias

Se asume que los requerimientos en este documento son estables una vez que sean aprobados por la Dirección (CLIENTE). Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes intervinientes y será gestionada por el equipo de desarrollo.

2.6. Requerimientos Futuros

- El sistema deberá permitir que el profesor ingrese únicamente los parciales y los promedios calculará el mismo sistema.

3. Requerimientos Específicos

3.1. Interfaces Externas

Interfaces de usuario:

La interfaz de usuario debe ser orientada a ventanas tipo Windows.

Interfaces Hardware:

Ratón y teclado estándar.

3.2. Funciones

1. El Sistema debe almacenar la información de notas de los alumnos ingresados por los profesores.
2. El Sistema debe almacenar la información de matrículas de los alumnos ingresados por la secretaria.
3. El sistema estudiantil deberá registrar los accesos de los usuarios.
4. La navegación en el menú debe ser de fácil entendimiento.
5. El sistema debe garantizar alto desempeño, confiabilidad.
6. El sistema debe garantizar con seguridad con su clave personal asignado.
7. Fácil mantenimiento para lo cual se generará la documentación completa del sistema.

3.3. Objetivos del Sistema

| | |
|--------------------|--|
| OBJ-01 | Gestionar los estudiantes |
| Descripción | El sistema deberá gestionar los estudiantes pertenecientes a la Escuela "San Pedro del Norte": matrículas, notas, etc. |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|--------------------|---|
| OBJ-02 | Gestionar los usuarios del sistema |
| Descripción | El sistema deberá gestionar los usuarios del sistema: perfil, login, password, etc. |
| Estabilidad | Alta |
| Comentarios | Ninguno |

3.4. Requerimientos de Almacenamiento de Información

| | |
|---------------------------------|---|
| RI-01 | Información sobre estudiantes |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Apellido Paterno - Apellido Materno - Nombres - Fechas de Nacimiento - Género |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|--|
| RI-02 | Información sobre notas |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a las notas de los estudiantes. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Curso - Período - Asignatura - Nota 1 - Nota 2 - Nota 3 |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

| | |
|---------------------------------|---|
| RI-03 | Información sobre usuarios |
| Objetivos asociados | OBJ-02 Gestionar los usuarios del sistema |
| Requerimientos asociados | |
| Descripción | El sistema deberá almacenar la información correspondiente a los usuarios. En concreto: |
| Datos específicos | <ul style="list-style-type: none"> - Perfil - Login - Password - Nombres - Apellidos - E-mail - Teléfono |
| Intervalo temporal | Pasado y presente |
| Estabilidad | Alta |
| Prioridad | Alta |
| Comentarios | Ninguno |

3.5. Requerimientos Funcionales

| | | |
|---------------------------------|--|--|
| RF-01 | Alta de estudiante | |
| Objetivos asociados | OBJ-01 Gestionar los estudiantes | |
| Requerimientos asociados | RI-02 Información sobre estudiantes | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite su ingreso como estudiante. | |
| Precondición | El solicitante no es estudiante y tiene su documentación disponible | |
| Secuencia normal | Paso | Acción |
| | 1 | La secretaria solicita al sistema comenzar el proceso de alta de un estudiante |
| | 2 | El sistema solicita los siguientes datos del nuevo estudiante: Código (Número secuencial), Apellido Paterno, Apellido Materno, Nombres, Fecha de Nacimiento, Género. |
| | 3 | La Secretaria solicita los datos requeridos y la documentación al nuevo estudiante. |
| | 4 | La secretaria comprueba que los datos del nuevo estudiante coinciden con los de la documentación aportada. |
| | 5 | La Secretaria proporciona los datos requeridos y solicita al sistema que los almacene |
| | 6 | El sistema almacena los datos proporcionados e informa a la Secretaria de que el proceso ha terminado con éxito |
| Postcondición | El solicitante es alumno del establecimiento | |
| Excepciones | Paso | Acción |
| | 4 | Si la documentación aportada no es correcta, la secretaria cancela la operación, a continuación este caso de uso termina |
| | 5 | Si el sistema detecta que el nuevo estudiante ya está registrado, el sistema informa de la situación a la secretaria permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa. |
| | 5 | Si la secretaria solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina |
| Rendimiento | Paso | Cota de tiempo |
| | 4 | 5 Segundos |
| Frecuencia esperada | 100 veces/día | |
| Estabilidad | Alta | |
| Comentarios | La frecuencia será mucho mayor durante los meses determinados para matrículas, probablemente 200 veces/día | |

| | | |
|---------------------------------|--|--|
| RF-02 | Alta de usuario | |
| Objetivos asociados | OBJ-02 Gestionar los usuarios | |
| Requerimientos asociados | RI-03 Información sobre usuarios | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se solicite su ingreso como usuario. | |
| Precondición | El solicitante no es usuario y tiene su documentación disponible | |
| Secuencia normal | Paso | Acción |
| | 1 | El Administrador solicita al sistema comenzar el proceso de alta de un usuario |
| | 2 | El sistema solicita los siguientes datos del nuevo usuario: Perfil, Login, Password, Nombres, Apellidos, e-mail, |

| | | |
|----------------------------|--|--|
| | | teléfono |
| | 3 | El Administrador solicita los datos requeridos. |
| | 4 | El Administrador proporciona los datos requeridos y solicita al sistema que los almacene |
| | 5 | El sistema almacena los datos proporcionados e informa al Administrador de que el proceso ha terminado con éxito |
| Postcondición | El solicitante es alumno del establecimiento | |
| Excepciones | Paso | Acción |
| | 5 | Si el sistema detecta que el nuevo usuario ya está registrado, el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa. |
| | 5 | Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina |
| Rendimiento | Paso | Cota de tiempo |
| | 4 | 5 Segundos |
| Frecuencia esperada | 3 veces/día | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

3.6. Requerimientos No Funcionales

| | |
|---------------------------------|---|
| RNF-01 | Copias de seguridad |
| Objetivos asociados | – |
| Requerimientos asociados | – |
| Descripción | El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados. |
| Comentarios | Ninguno |

| | |
|---------------------------------|--|
| RNF-02 | Portabilidad |
| Objetivos asociados | – |
| Requerimientos asociados | – |
| Descripción | El sistema deberá ser fácilmente portable a cualquier navegador. |
| Comentarios | Ninguno |

3.7. Requerimientos de rendimiento

El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 5 segundos. El tiempo de respuesta a los listados dependerá de la tecnología de impresión.

3.8. Requerimientos de Desarrollo

El ciclo de vida será el de Prototipado Evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que permita incorporar de manera sencilla cambios y nuevas funcionalidades.

3.9. Restricciones de Diseño

Seguridad:

La seguridad de los datos será establecida por el Sistema Gestor de Base de Datos Relacional que se emplee.

Política de Respaldo:

No se ha definido.

Base de Datos:

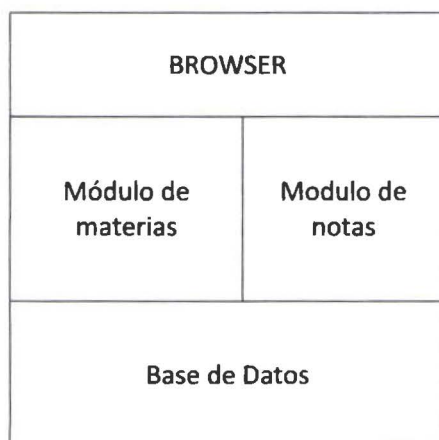
El Sistema Gestor de Base de Datos debe ser relacional y se accederán a los mismos usando la tecnología ODBC.

Política de Borrado:

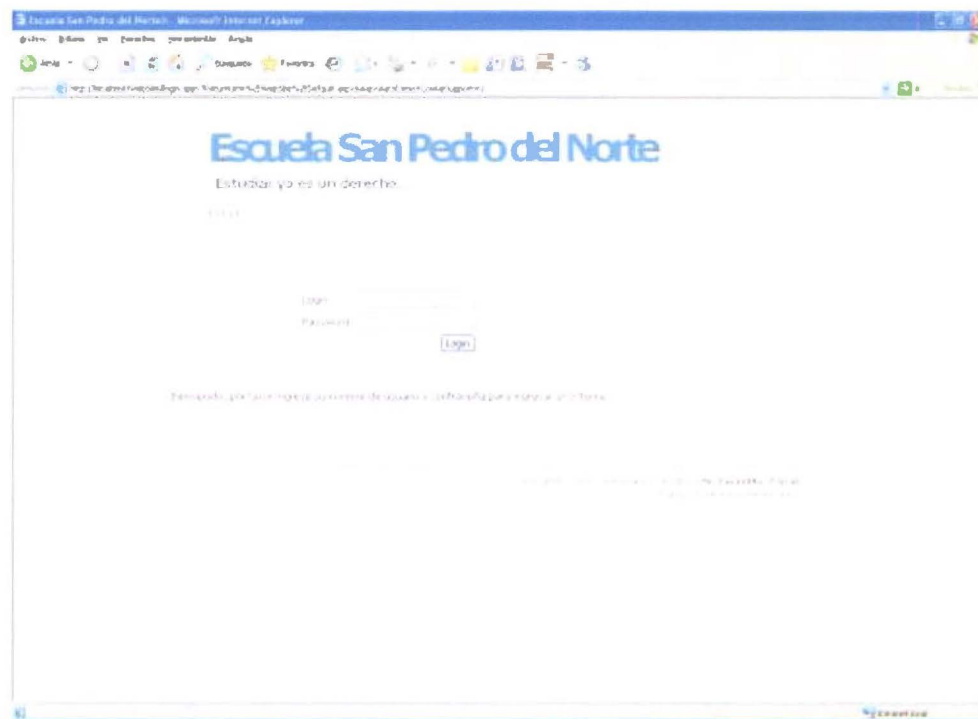
No se ha definido.

4.6.5. Diseño

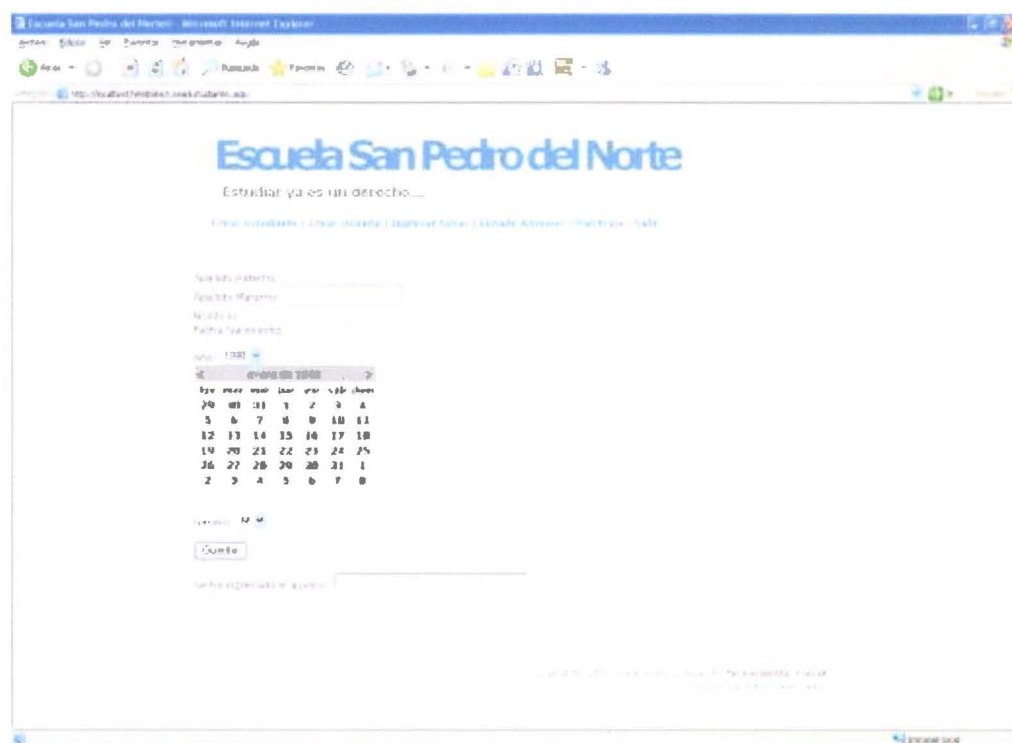
4.6.5.1. Diagrama de arquitectura



4.6.5.2. Diseño de interfaces



Pantalla de Inicio

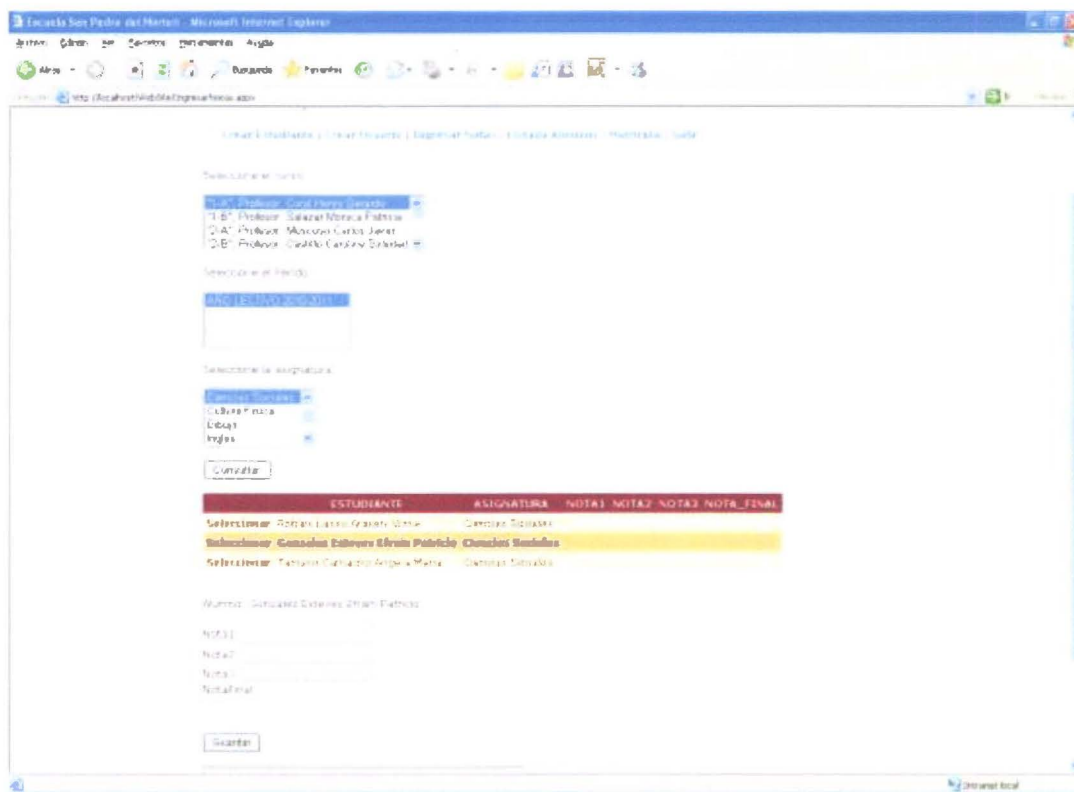


Pantalla Crear Estudiante

Pantalla Ingresar Notas

| APELLIDO PATERNO | APELLIDO MATERNO | NOMBRES | FECHA NACIMIENTO | GENERO |
|------------------|------------------|-------------------|------------------|--------|
| Alvarado | Salas | San Francisco | 2001-05-15 | M |
| Armas | Araya | Daniela Alejandra | 2002-05-25 | F |
| Armas | Araya | Estefanía | 2000-05-25 | F |
| Armas | Araya | Carlos Alberto | 2002-10-10 | M |
| Ortega | Rodriguez | Maria Belen | 2003-04-26 | F |
| Colombes | Santolalla | Marta Mariana | 2003-05-26 | F |
| Cordeiro | Ortega | Maria Alejandra | 2002-07-23 | F |
| Argente | Dea | Geovana Elizabeth | 2003-04-07 | F |
| Camacho | Camacho | Elmer Ramiro | 2003-06-26 | M |
| Camacho | Camacho | Yuleidy | 2001-02-10 | F |
| Martinez | Spence | Yulio Yulio | 2001-05-13 | M |
| More | More | Maria Cecilia | 1998-03-25 | F |
| More | More | Marta Cecilia | 2002-03-26 | F |
| More | More | Marta Roberto | 2000-09-26 | M |
| Ortega | Ortega | Jose David | 2001-04-26 | M |
| Roberto | Roberto | Aracely Araya | 2001-09-26 | F |
| Ortega | Ortega | Maria Ornela | 2000-04-07 | F |
| Tamayo | Ortega | Angela Maria | 2002-08-26 | F |
| Ortega | Ortega | Yuleidy Elizabeth | 2001-08-23 | F |
| Ortega | Rodriguez | Marta Cecilia | 1998-03-26 | F |
| Ortega | More | Marta Cecilia | 1998-03-26 | F |

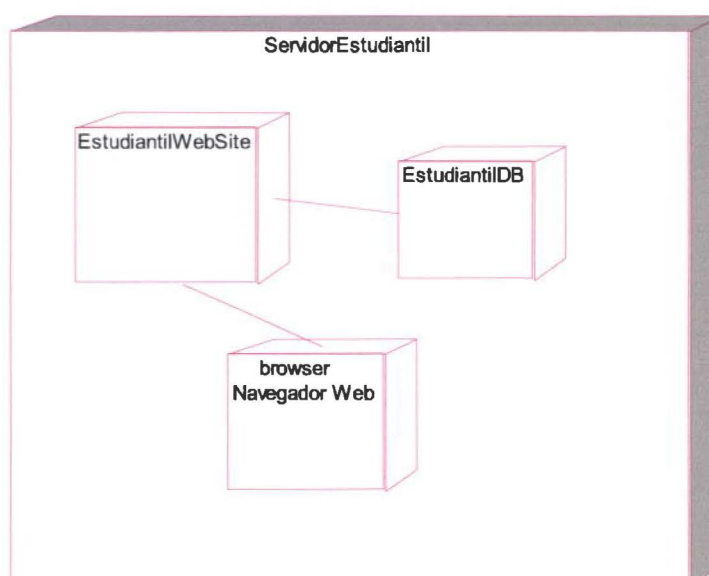
Pantalla Listado de Alumnos



Pantalla Consultar Nota

4.6.6. Implementación

4.6.6.1. Diagrama de Despliegue



4.6.6.2. Ejemplo del código del sistema

```

CREATE procedure [dbo].[usp_ingresar_estudiante]
    @EST_ID INT,
    @est_nombres varchar(64),
    @est_apellido_paterno varchar(64),
    @est_apellido_materno varchar(64),
    @est_fecha_nacimiento datetime,
    @est_genero char(1)
SELECT      ESTUDIANTE.EST_APELLIDO_PATERNO AS
[APELLIDO PATERNO], ESTUDIANTE.EST_APELLIDO_MATERNO AS [APELLIDO
MATERNO],
            ESTUDIANTE.EST_NOMBRES AS NOMBRES,
EVALUACION.EVA_NOTA1 AS [NOTA 1], EVALUACION.EVA_NOTA2 AS [NOTA 2],
            EVALUACION.EVA_NOTA3 AS [NOTA 3],
EVALUACION.EVA_NOTA_FINAL AS [NOTA FINAL]
FROM      EVALUACION CROSS JOIN
            ESTUDIANTE
as
begin
    if exists(select est_apellido_paterno, est_nombres from
Estudiante where est_apellido_paterno = @est_apellido_paterno and
est_nombres = @est_nombres)
        Begin
            update estudiante
                set Est_nombres = @est_nombres,
                    est_apellido_paterno = @est_apellido_paterno,
                        est_apellido_materno =
@est_apellido_materno,
                            est_fecha_nacimiento =
@est_fecha_nacimiento,
                                est_genero = @est_genero
                    where (est_apellido_paterno =
@est_apellido_paterno and est_nombres = @est_nombres)
                        --raiserror('estudiante ha sido actualizado'',
16, 1)
                select 'estudiante ha sido actualizado'
                return -1
        End

```

```
insert estudiante(est_apellido_paterno,  
est_apellido_materno, est_nombres, est_fecha_nacimiento, est_genero)  
values(@est_apellido_paterno, @est_apellido_materno,  
@est_nombres, @est_fecha_nacimiento, @est_genero)  
select 'estudiante se ha registrado', convert(smallint,  
@@identity)  
end
```


CONCLUSIONES

- La brecha entre la investigación académica sobre ingeniería de requerimientos y los problemas reales que tienen los desarrolladores es preocupante y grande, la visión de muchos de ellos es que todo lo relacionado con los requerimientos no es más que la generación de documentación que nadie lee, exigida por los estándares que han de seguir o por los procesos de calidad implantados en la organización donde laboran.
- Las ideas expuestas en este trabajo tienen como objetivo reducir la brecha existente, ya que algunas técnicas están evolucionando enormemente a través de su aplicación en numerosas prácticas académicas, y en la aplicación en pocos proyectos reales.
- Dentro de las técnicas de validación se ha prestado especial atención al walkthroughs y a los prototipos, especialmente a los de interfaz del usuario, que son los más interesantes en la fase de ingeniería de requerimientos y especialmente durante las fases de elicitación y validación en las que la comunicación con los clientes y usuarios es lo más importante.
- La metodología desarrollada demostró ser adecuada para el proceso de ingeniería de requerimientos de un sistema de software.
- Existen varias técnicas para procesos de ingeniería de requerimientos pero es necesario seleccionar la más apropiada para el proyecto que se quiere realizar.
- El proceso de ingeniería de requisitos es altamente relevante dentro del proceso de ingeniería de software porque determina las necesidades que un sistema debe satisfacer.

RECOMENDACIONES

- Impulsar la utilización de la metodología propuesta para el proceso de ingeniería de software en las empresas desarrolladoras del software.
- Asignar el tiempo y los recursos necesarios para realizar la fase de ingeniería de requerimientos en el desarrollo un proyecto de software.
- Utilizar herramientas que permitan la automatización del proceso de ingeniería de requerimientos.

BIBLIOGRAFÍA

LIBROS

- SOMERVILLE, IAN, Ingeniería del Software. Editorial Addison Wesley, Sexta Edición.
- LÓPEZ, HERNÁN, SOTAQUIRÁ, RICARDO, La Metodología de Sistemas “Blandos” De Chekland: El Heraldo de un Cambio Paradigmático en el Movimiento de Sistemas.
- MUÑOZ, JOSÉ, Aplicando la metodología de los sistemas blandos. Documentación del Curso Ingeniería de Sistemas, Diinf, Usach. Segundo semestre 2006.
- USACH, DIINF, Un Resultado Principal de la Investigación: La Metodología de Sistemas para Enfrentar Problemas No Estructurados, Documentación del Curso Ingeniería de Sistemas, Segundo semestre 2007.
- MARTÍNEZ, ANDRÉS, Una metodología para el diseño de sistemas de información, basada en el estudio de sistemas blandos, Revista Espacios Vol. 25 (2) 2004.
- TORRES, JOSÉ LUIS, Especificación de requerimientos en ingeniería de software.
- PRESSMAN, ROGER, Ingeniería del Software Un enfoque práctico. Editorial Mc Graw Hill, Cuarta edición.
- WIEGERS. Customer Rights and Responsibilities. Software Development, Diciembre 1999.
Disponible en <http://www.sdmagazine.com/breakrm/features/s9912f2.shtml>.

INTERNET

- http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER01/baez.pdf
- http://www.pol.una.py/archivos/IngeInfo/IngeSoftI/Ingenieria_Software_I.pdf
- <http://www.itba.edu.ar/capis/webcapis/RGMITBA/comunicacionesrgm/WICC-07-435-437.pdf>

- <http://lte.disc.ucn.cl/~jbekios/Pregrado/Ing-Software/ingSoftware.pdf>
- <http://www.uv.mx/mis/materias%5C2%20Reuso.pdf>
- <http://www.sel.unsl.edu.ar/licenciatura/ingsoft1/Apuntes/2007/practico1.pdf>
- http://www.intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf
- http://xmstudio.com.ar/Sistemas/uploads/calidaddesoftware/IEEE_830.pdf
- http://www.lisi.usb.ve/publicaciones/03%20evaluacion/evaluacion_09.pdf
- <http://www.dccia.ua.es/~eli/novatica03.pdf>
- http://www.tesisenxarxa.net/TESIS_UdL/AVAILABLE/TDX-0218107-133615//Tgsa1de5.pdf
- <http://www.itba.edu.ar/capis/rtis/rtis-7-2/Ingenieria-de-Software-Libre-y-Herramientas-Aplicadas.pdf>
- <http://www.inf.utfsm.cl/~visconti/papers/papersquidtxpan1999.pdf>
- <http://www.monografias.com/trabajos6/resof/resof.shtml>
- <http://www.monografiass.com/Ingenieria/more12.shtml>
- <http://www.data2max.com/ingenieria/658us672pmbwi.pdf>
- http://200.51.43.210/Programas/Satep0/PRG_0483_029_UACO_PAct.pdf
- <http://dis.unal.edu.co/profesores/ypinzon/2013326-206/docs/Presentacion0Medina.pdf>
- <http://www.monografias.com/trabajos12/ingreq/ingreq.shtml>
- <http://dis.unal.edu.co/profesores/ypinzon/2013326-206/docs/Presentacion1Medina.pdf>
- <http://www.freewebs.com/lfmedinac/poster2.pdf>
- <http://capturaderequerimientos.blogspot.com/2007/11/ingeniera-de-software.html>
- <http://www.freewebs.com/lfmedinac/presentacion-problema2.pdf>
- http://www.ucaecemdp.edu.ar/mdpcc/Carreras/De_Grado/DptoSistemas/Programas/Programas_Oficiales_Lic._en_Sistemas_mayo07/Ingenieria_de_Requerimientos_%7B6005-04S%7D_P04S_VIG_04_mayo07.pdf
- http://www.utm.mx/~caff/doc/Ing_de_Req_aplicada_a_la_UVi_de_la_UTM_ANIEI2003.pdf
- <http://www.inf.puc-rio.br/~wer98/artigos/79.html#3>