



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: INVESTIGACIÓN DE LAS PRÁCTICAS DE LA
INGENIERÍA DE REQUISITOS EN LAS EMPRESAS DE
DESARROLLO DE SOFTWARE DE LA CIUDAD DE QUITO Y
GENERACIÓN DE UNA PROPUESTA METODOLÓGICA PARA
LA MEJORA DE DICHAS PRÁCTICAS**

AUTORES:

**WILSON JAVIER SIMBAÑA SARANSIG
JUAN GABRIEL SIMBAÑA QUINSASAMÍN**

DIRECTOR: ING. HINOJOSA, CECILIA

COORDIRECTOR: ING. RON, MARIO

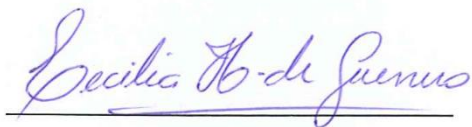
SANGOLQUÍ

2015


CERTIFICADO

Certificamos que el presente trabajo titulado “Investigación de las prácticas de la ingeniería de requisitos en las empresas de desarrollo de software de la ciudad de Quito y generación de una propuesta metodológica para la mejora de dichas prácticas” fue realizado en su totalidad por el Sr. Wilson Javier Simbaña Saransig y el Sr. Juan Gabriel Simbaña Quinsasamin, como requisito parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, Mayo de 2015



ING. CECILIA HINOJOSA
DIRECTOR



ING. MARIO RON
CODIRECTOR

AUTORÍA DE RESPONSABILIDAD

Nosotros, Wilson Javier Simbaña Saransig y Juan Gabriel Simbaña Quinsasamin, declaramos que el presente trabajo es de nuestra autoría; que no ha sido previamente presentado para ningún grado, calificación personal y que hemos consultado las referencias bibliográficas que se incluyen en el documento.

La Universidad de las Fuerzas Armadas - ESPE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la ley de propiedad intelectual, por su reglamento y normativa institucional vigente.

Sangolquí, Mayo de 2015



Wilson Javier Simbaña Saransig



Juan Gabriel Simbaña Quinsasamin

AUTORIZACIÓN

Nosotros, Wilson Javier Simbaña Saransig y Juan Gabriel Simbaña Quinsasamin, autorizamos a la Universidad de las Fuerzas Armadas - ESPE, a que publique en el repositorio digital de la biblioteca Alejandro Segovia el presente trabajo de fin de carrera, así como también los materiales y documentos relacionados con la misma.

Sangolquí, Mayo de 2015



Wilson Javier Simbaña Saransig



Juan Gabriel Simbaña Quinsasamin

DEDICATORIA

Dedico este trabajo de fin de carrera principalmente a Dios. A mis padres y hermanos quienes me brindan su confianza y apoyo infinito. A mi novia por el apoyo brindado en la consecución de este trabajo y a mi directora de trabajo de fin de carrera, por su valiosa colaboración.

Espero que éste trabajo, esté a la altura de lo que merecen.

WILSON JAVIER SIMBAÑA SARANSIG

DEDICATORIA

Dedico el presente trabajo a Dios, a mi madre que ha sido el pilar fundamental en mi vida por todo su apoyo, dedicación y confianza.

A mi hermano que siempre ha estado de forma incondicional. A toda mi familia que ha estado a mi lado.

A mis directores de trabajo de fin de carrera por su colaboración, enseñanza y paciencia.

JUAN GABRIEL SIMBAÑA QUINSASAMIN

AGRADECIMIENTO

Agradezco a Dios por permitirme llegar a este momento tan especial en mi vida, a mis padres Carlos y Zoila que con su amor, trabajo y sacrificio me forjaron como la persona que soy en la actualidad, muchos de mis logros incluido este, se los debo a ustedes y a mis hermanos por su apoyo en todo momento.

Ingeniera Cecilia Hinojosa mi agradecimiento por su valiosa colaboración durante la ejecución del trabajo realizado.

Finalmente un eterno agradecimiento a la Universidad de las Fuerzas Armadas E.S.P.E., por prepararme para un futuro competitivo y fórmame como profesional.

WILSON JAVIER SIMBAÑA SARANSIG

AGRADECIMIENTO

Agradezco a Dios por brindarme salud y sabiduría para concluir esta importante etapa de mi vida.

Le doy gracias a mi madre que ha sido padre y madre, gracias por ser el mejor ejemplo que he tenido en mi vida, por ser una persona luchadora y que a pesar de la adversidad siempre ha logrado salir adelante, por siempre contar su apoyo y recibir sus palabras de aliento; a mi hermano que siempre ha estado a mi lado y gracias a ellos hacer posible mi formación profesional.

A mis hermanas y sobrinos que me han brindado su confianza y siempre han estado en los buenos y malos momentos de forma incondicional.

A mi esposa Janeth por su paciencia en este tiempo, por apoyarme y darme la fuerza para culminar esta meta.

A mis directores de trabajo de fin de carrera, por la colaboración recibida en todo el proceso del presente trabajo.

JUAN GABRIEL SIMBAÑA QUINSASAMIN

ÍNDICE DE CONTENIDO

CERTIFICADO.....	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	viii
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS.....	xii
RESUMEN.....	xiii
ABSTRACT	xiv
CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 JUSTIFICACIÓN	2
1.3 OBJETIVOS	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos Específicos	3
1.4 ALCANCE.....	3
CAPÍTULO 2: MARCO TEÓRICO	5
2.1 INGENIERÍA DE REQUISITOS.....	5
2.1.1 Características de los requisitos.....	5
2.1.2 Clasificación de requisitos.....	6
2.2 PROCESOS DE LA INGENIERÍA REQUISITOS	13
2.2.1 Documentación.....	14
2.2.2 Elicitación.....	17
2.2.3 Negociación	27

2.2.4	Validación de Requisitos	31
2.2.5	Gestión de los Requisitos	33
2.3	MODELOS, ESTANDARES Y NORMAS DE CALIDAD.....	40
2.3.1	Modelo Integrado de Capacidad de Madurez – CMMI.....	40
2.3.2	ISO/IEC 25030:2007 Software Engineering	48
2.3.3	Estándar IEEE 830 -Especificación de Requisitos Software.....	53
CAPÍTULO 3: PLAN DE INVESTIGACIÓN DE CAMPO.....		60
3.1	Objetivo General.....	60
3.2	Objetivos Específicos	60
3.3	Justificación	60
3.4	Metodología de la investigación de Campo.....	60
3.4.1	Tipos de investigación	60
3.4.2	Hipótesis	62
3.4.3	Unidad de análisis.....	62
3.4.4	Población	62
3.4.5	Muestra	62
3.5	Elaboración del Instrumento de investigación.....	63
CAPÍTULO 4: ANÁLISIS DE RESULTADOS.....		64
4.1	ANÁLISIS DE RESULTADOS POR PREGUNTA.....	64
4.1.1	Pregunta N° 1	64
4.1.2	Pregunta N° 2	65
4.1.3	Pregunta N° 3	66
4.1.4	Pregunta N° 4	66
4.1.5	Pregunta N° 5	67
4.1.6	Pregunta N° 6	68
4.1.7	Pregunta N° 7	70
4.1.8	Pregunta N° 8	71
4.2	ACOTACIONES A LA APLICACIÓN DE LA ENCUESTA GUIADA:	76
CAPÍTULO 5: PROPUESTA METODOLÓGICA PARA EL PROCESO DE		
	INGENIERÍA DE REQUISITOS	77

5.1	OBJETIVO DE LA METODOLOGÍA	78
5.2	FASES ACTIVIDADES Y TÉCNICAS RECOMENDADAS	78
5.2.1	Elicitación.....	80
5.2.2	Análisis	82
5.2.3	Especificación.....	83
5.2.4	Gestión.....	84
CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES		85
6.1	CONCLUSIONES	85
6.2	RECOMENDACIONES.....	86
BIBLIOGRAFÍA		88
GLOSARIO		91

ÍNDICE DE FIGURAS

Figura 1. Tipos de requisitos no funcionales	7
Figura 2. Diagrama de casos de uso.....	13
Figura 3. Proceso de ingeniería de requisitos.....	14
Figura 4. Relación entre las fases del proceso de ingeniería de requisitos	15
Figura 5. Identificación de fuentes relevantes.....	18
Figura 6. Fuentes de requisitos existentes.....	19
Figura 7. Idoneidad de las técnicas para las sub-actividades	25
Figura 8. Mapa mental ingeniería de requisitos.....	27
Figura 9. Matriz de Interacción.....	30
Figura 10. Progreso del esfuerzo necesario para reparar requisitos defectuosos	32
Figura 11. Matriz de trazabilidad para varias relaciones	37
Figura 12. Clasificación Kano	40
Figura 13. La historia de los CMMs	42
Figura 14. Proceso de cambio en requisitos gestionado por requirement management ..	46
Figura 15. Organización de la serie de estándares de SQuaRE	49
Figura 16. Modelo de calidad interno y externo	50
Figura 17. Modelo de calidad en el uso.	51
Figura 18. Categorización de los requisitos	52
Figura 19. Porcentaje de utilización de cada fase del proceso de IR.	70
Figura 20. Porcentaje de técnicas utilizadas para elicitación de requisitos.....	71
Figura 21. Porcentaje de técnicas utilizadas para el análisis de requisitos.	72
Figura 22. Porcentaje de técnicas utilizadas para la negociación de requisitos.....	72
Figura 23. Porcentaje de técnicas utilizadas para documentar requisitos.	73
Figura 24. Porcentaje de técnicas utilizadas para validar requisitos en las empresas	74
Figura 25. Porcentaje de técnicas utilizadas para gestionar requisitos.	74
Figura 26. Porcentaje de utilización de cada fase del proceso de IR pregunta 7	75
Figura 27. Porcentaje de utilización de cada fase del proceso de IR pregunta 8.	76
Figura 28. Diagrama del proceso de ingeniería de requisitos propuesta	79

ÍNDICE DE TABLAS

Tabla 1. Métricas para especificar requisitos no funcionales	9
Tabla 2. Ej. de uso de información de trazabilidad durante el desarrollo del sistema	35
Tabla 3. Tiempo de vida la empresa o departamento de desarrollo de software	64
Tabla 4. Cantidad de personal técnico en el área de desarrollo de software.....	65
Tabla 5. Proyectos realizados según el tiempo de desarrollo.....	66
Tabla 6. Número de personal especializado en ingeniería de requisitos.....	67
Tabla 7. Número de personal especializado en ingeniería de software.	68
Tabla 8. Número de empresas con certificación CMMI	69
Tabla 9. Número de empresas con certificación ISO	69

RESUMEN

El presente trabajo presenta un estudio de investigación sobre las prácticas de la ingeniería de requisitos en las empresas de desarrollo de software de la ciudad de Quito. El estudio se realizó para obtener una idea de la medida en que las prácticas de ingeniería de requisitos han sido adoptadas por las organizaciones en Quito, cómo estas organizaciones describen sus requisitos y la evaluación de sus prácticas de ingeniería de requisitos. El estudio se realizó en 51 empresas elegidas de forma aleatoria, y la encuesta guiada fue diseñada para utilizar la estadística descriptiva sobre los datos recogidos. El estudio ha puesto de manifiesto el estado actual de la ingeniería de requisitos en la industria del software de Quito. A partir del resultado, se puede concluir que la industria del software carece de empleo de buenas prácticas en la ingeniería de requisitos. Esto indica que el área de la ingeniería de requisitos debería aplicarse con mayor regularidad, con el fin de mejorar la calidad del software y la productividad dentro del marco de tiempo programado.

PALABRAS CLAVES:

- PRÁCTICAS DE INGENIERÍA DE REQUISITOS
- INVESTIGACIÓN EXPLORATORIA
- ENCUESTA GUIADA
- EMPRESAS DE DESARROLLO DE SOFTWARE,
- QUITO – ECUADOR.

ABSTRACT

This paper presents an investigation of requirement engineering for development companies at Quito. The study was made for get an idea of measures in the requirement engineering practice adopted for these organizations, as they describe requirements and evaluations in their practices in requirement engineering. The study was made in 51 random companies, and a guided inquiry was designed for use engineering statistics on the data collected. This study shows the actual status of software industry at requirement engineering in the city of Quito. With these results, it can be conclude that software industry lacks from good practices at requirement engineering. This concludes that requirement engineering should be applied with more regularity, in order to improve the software quality and the productivity under the schedules.

KEYWORDS:

- REQUIREMENTS ENGINEERING PRACTICE
- EXPLORATORY RESEARCH
- QUESTIONARY GUIDED
- SOFTWARE DEVELOPMENT COMPANIES
- QUITO – ECUADOR.

CAPÍTULO 1: INTRODUCCIÓN

En la actualidad la ingeniería de requisitos ha cobrado una mayor importancia, tanto para su estudio, como para su uso en la industria de desarrollo de software, debido a que se espera que los productos de software proporcionen funciones cada vez más centradas en el usuario, con mayor calidad y seguridad. Por ser una de las actividades que se realizan en etapas tempranas del proceso, su calidad tiene un alto impacto en el producto resultante. Por tal motivo es importante, conocer cómo se utilizan estas técnicas, métodos, herramientas, notaciones, en la industria del software, de manera general y particularmente en la ciudad de Quito, en donde se encuentran radicadas un número importante de empresas dedicadas a esta actividad, con un diagnóstico objetivo en el cual se identifiquen las causas raíz de los problemas y dificultades que tienen las empresas para aplicar las propuestas de la ingeniería de requisitos, será posible diseñar mejores métodos, centrarse en una formación apropiada y propiciar el desarrollo eficiente de software de alta calidad.

La industria de software en el Ecuador no dispone de estudios que permitan orientar acciones concretas de mejora, por lo cual se ha creído conveniente realizar la investigación de las prácticas de la ingeniería de requisitos a una muestra significativa de las empresas de desarrollo de software, radicadas en la ciudad de Quito; considerando que en esta ciudad se concentran el 49% de las empresas de este estratégico sector. (CIU, 2012)¹

1.1 PLANTEAMIENTO DEL PROBLEMA

Según datos proporcionados por la Superintendencia de Compañías, a noviembre de 2013, Quito cuenta con 388 empresas dedicadas a:

“Actividades de diseño de la estructura y el contenido de los elementos siguientes (y/o escritura del código informático necesario para su creación y aplicación): programas de sistemas operativos (incluidas actualizaciones y parches de corrección), aplicaciones informáticas (incluidas actualizaciones y parches de corrección), bases de datos y páginas web.

¹Clasificación Internacional Industrial Uniforme.

Adaptación de programas informáticos a las necesidades de los clientes, es decir, modificación y configuración de una aplicación existente para que pueda funcionar adecuadamente con los sistemas de información de que dispone el cliente.

Actividades de planificación y diseño de sistemas informáticos que integran equipo y programas informáticos y tecnología de las comunicaciones.” (Superintendencia de Compañías, 2013)

Del total de compañías en la ciudad, 80 empresas se dedican exclusivamente al desarrollo de software. El sector de desarrollo de software ha sido reconocido como un eje estratégico por el gobierno ecuatoriano, quien a través del Ministerio de Industrias y Productividad, busca “...el fortalecimiento de la cadena de valor del sector software y la promoción e inserción de los productos ecuatorianos en mercados locales e internacionales, lo cual aporta al Cambio de la Matriz Productiva que impulsa el Gobierno Nacional.” (MIPRO, 2013), por lo que es necesario fortalecer esta área.

El Ecuador no cuenta con un estudio de las prácticas utilizadas en el proceso de desarrollo, así como de las prácticas empleadas en la ingeniería de requisitos. Sin un estudio que permita evidenciar lo que se aplica en la cotidianidad de la industria de software, no será posible proponer acciones tendientes a fortalecer este proceso que tiene una alta incidencia en la calidad del producto resultante, y consecuentemente en la calidad de la industria de software. Por lo que el presente proyecto plantea aportar con el estudio requerido y en base a los resultados obtenidos generar una propuesta metodológica para mejorar las prácticas que comúnmente se usan en ingeniería de requisitos.

1.2 JUSTIFICACIÓN

La investigación tendrá como objetivo generar una propuesta metodológica que contribuya a mejorar las prácticas de ingeniería de requisitos en la industria de software en el Ecuador, considerando una muestra significativa de organizaciones y empresas dedicadas al desarrollo de software. Para el efecto, se realizará una investigación aplicada, exploratoria y cuantitativa, que permita conocer la situación real, entorno a las prácticas de la ingeniería de requisitos en la ciudad de Quito.

Con un conocimiento profundo de la realidad, se planteará una propuesta metodológica para mejorar las prácticas que comúnmente se usan en ingeniería de requisitos, con lo cual se espera contribuir de manera directa a mejorar la calidad de los productos de software ecuatorianos y su industria.

1.3 OBJETIVOS

1.3.1 Objetivo General

Investigar las prácticas de la ingeniería de requisitos en las empresas de desarrollo de software en la ciudad de Quito y generar una propuesta metodológica para la mejora de dichas prácticas.

1.3.2 Objetivos Específicos

- Evaluar la situación actual de las prácticas en ingeniería de requisitos en las empresas de desarrollo de software de la ciudad de Quito.
- Estudiar los diferentes modelos de calidad enfocados en la ingeniería de requisitos y adaptar las mejores prácticas al medio local.
- Plantear la propuesta metodológica tendiente a mejorar la aplicación de buenas prácticas en ingeniería de requisitos, en la industria de desarrollo de software.
- Aplicar la guía metodológica para ingeniería de requisitos en un caso práctico.

1.4 ALCANCE

Desarrollar el marco teórico: Se detallarán las diferentes propuestas actuales en la práctica de ingeniería de requisitos.

Diseñar los instrumentos de investigación: Será considerado el siguiente instrumento para el análisis de campo: encuesta guiada.

Aplicar encuesta guiada: Aplicar el instrumento de investigación a una muestra significativa de las empresas dedicadas al desarrollo de software en la ciudad de Quito.

Analizar e interpretar los resultados: Se analizarán los resultados obtenidos de la encuesta anteriormente planteada y se obtendrán conclusiones las cuales servirán para la toma de decisiones en la propuesta de mejora de las prácticas de ingeniería de requisitos.

Elaborar el informe del diagnóstico: Con los resultados obtenidos en el análisis e interpretación de la situación en las empresas desarrolladoras de software se realizará el informe de diagnóstico sobre la situación actual de las prácticas de ingeniería de requisitos en la ciudad de Quito.

Diseñar una propuesta metodológica tendiente a mejorar el proceso de ingeniería de requisitos: En base a los resultados obtenidos, se generará una propuesta metodológica tendiente a optimizar las prácticas de ingeniería de requisitos en las empresas dedicadas al desarrollo de software, de nuestro medio.

Realizar un prototipo de obtención y especificación de requisitos: Se desarrollará un prototipo de la obtención requisitos, especificados en base a las recomendaciones planteadas en la propuesta metodológica.

CAPÍTULO 2: MARCO TEÓRICO

2.1 INGENIERÍA DE REQUISITOS

Existen diferentes definiciones de ingeniería de requisitos según distintos autores, por ejemplo las siguientes:

“La ingeniería de requisitos, se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y administración de los requisitos para un producto determinado.” (Ortas, 2001).

“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” (IEEE Std 610, 1990)

“Un proceso sistemático de desarrollo de requisitos a través de un proceso iterativo cooperativo de análisis del problema, la documentación de las observaciones resultantes en una variedad de formatos de representación y la comprobación de la exactitud de la comprensión obtenida” (Loucopoulos & Karakostas, 1995)

A las definiciones anteriores se puede acotar que la ingeniería de requisitos es el proceso de recopilar, analizar, validar, priorizar, especificar, documentar las necesidades del cliente, almacenarlos en un repositorio central y administrar sus cambios. No existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe aplicar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, al nivel de experiencia y habilidad de las personas encargadas de la ingeniería de requisitos.

2.1.1 Características de los requisitos.

Un conjunto de requisitos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes según (Boehm, 2007):

- **Necesario:** Un requisito 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 requisito 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 requisito está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- **Consistente:** Un requisito es consistente si no es contradictorio con otro requisito.
- **No ambiguo:** Un requisito no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
- **Verificable:** Un requisito es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Dificultades para definir los requisitos (Pressman, 2006):

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

2.1.2 Clasificación de requisitos

El proceso de ingeniería de requisitos trata varios tipos de requisitos, estos se pueden clasificar en 3 grupos:

- Según la característica del requisito: requisitos funcionales, no funcionales y de información.
- Según el ámbito del requisito: requisitos de hardware, software y sistemas.
- Según la audiencia del requisito: requisitos de usuarios y desarrolladores.

2.1.2.1 Según la característica del requisito

Como parte de esta clasificación se consideran a los siguientes tipos:

- **Requisitos Funcionales:** Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en estas situaciones. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (Sommerville, Ingeniera de Software, 2005).

Ejemplo: El usuario deberá tener la posibilidad de buscar en la base de datos, sin la necesidad de escribir la palabra completa.

1. **Requisitos No Funcionales:** Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no funcionales a menudo se aplican al sistema en su totalidad. (Sommerville, Ingeniera de Software, 2005)

Ejemplo: El sistema no debe tardar más de 2 segundos al enlistar la lista de clientes existentes.

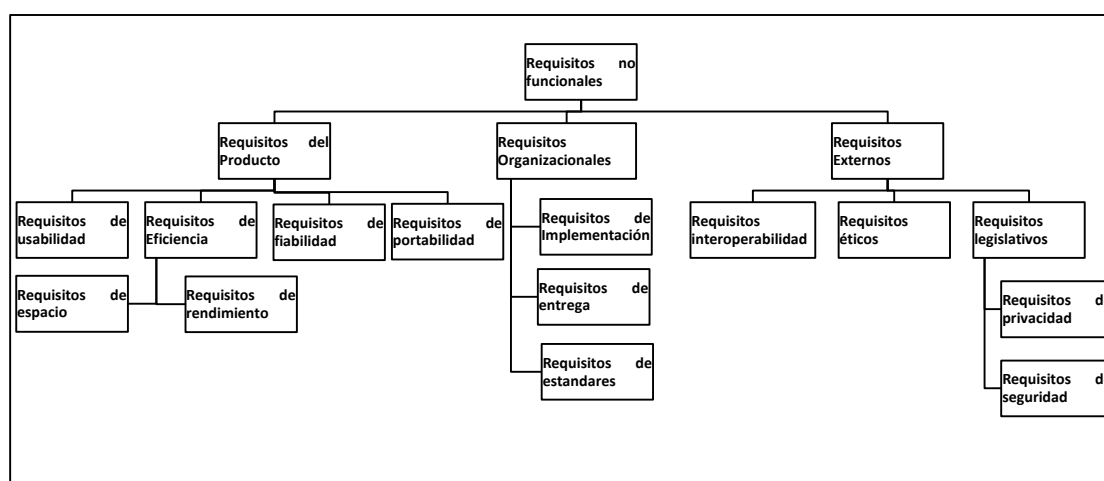


Figura 1. Tipos de requisitos no funcionales

Fuente: (Sommerville, Ingeniera de Software, 2005)

La **Figura 1** muestra la clasificación de los requisitos no funcionales. Puede verse en este diagrama que los requisitos no funcionales pueden venir de las características requeridas del software (requisitos del producto), de la organización que desarrolla el software (requisitos organizacionales) o de fuentes externas.

Los tipos de requisitos no funcionales son:

- **Requisitos del producto.** Estos requisitos especifican el comportamiento del producto. Algunos ejemplos son los requisitos de rendimiento en la rapidez de ejecución del sistema y cuánta memoria se requiere; porcentaje de fiabilidad que fijan la tasa de fallos para que el sistema sea aceptable; los requisitos de portabilidad, y los requisitos de usabilidad. (Sommerville, Ingeniera de Software, 2005)

Ejemplo: La interfaz de usuario del Sistema se utilizará JSF y AJAX.

- **Requisitos organizacionales.** Estos requisitos se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. Algunos ejemplos son los estándares en los procesos que deben utilizarse; los requisitos de implementación, como los lenguajes de programación o el método de diseño a utilizar, y los requisitos de entrega que especifican cuándo se entregará el producto y su documentación. (Sommerville, Ingeniera de Software, 2005)

Ejemplo: El proceso de desarrollo del sistema y los documentos a entregar deberán ajustarse al proceso y a los productos a entregar definidos en XYZ-STAN-95.

- **Requisitos externos.** Se incluye todos los requisitos que se derivan de los factores externos al sistema y de su proceso de desarrollo. Éstos pueden incluir los requisitos de interoperabilidad que definen la manera en que el sistema interactúa con sistemas de otras organizaciones, requisitos legales que deben seguirse para asegurar que el sistema funcione dentro de la ley, y los requisitos éticos. Estos últimos son puestos en un sistema para asegurar que será aceptado por sus usuarios y por el público en general. (Sommerville, Ingeniera de Software, 2005)

Ejemplo: El sistema debe adquirir un certificado SSL (https) para ser publicado en el internet.

Siempre que sea posible, se deben redactar los requisitos no funcionales de manera cuantitativa para que se puedan probar de un modo objetivo. La tabla 1 muestra varias métricas posibles que pueden usarse para especificar las propiedades no funcionales del sistema. Se pueden medir estas características cuando se prueba el sistema para comprobar si cumple sus requisitos no funcionales.

Tabla 1

Métricas para especificar requisitos no funcionales

Requisito no funcional	Ejemplo
Rapidez	Transacciones procesadas por segundo. Tiempo de respuesta al usuario y a eventos Tiempo de actualización de la pantalla
Tamaño	KB Número de chips de RAM
Facilidad de uso	Tiempo de formación Números de cuadros de ayuda
Fiabilidad	Tiempo medio entre fallos Probabilidad de no disponibilidad Tasa de ocurrencia de fallos Disponibilidad
Robustez	Tiempo de reinicio después de fallos Porcentaje de eventos que provocan fallos Probabilidad de corrupción de los datos después de fallos
Portabilidad	Porcentaje de declaraciones dependientes del objeto Número de sistemas objeto

Fuente: (Sommerville, *Ingeniera de Software*, 2005)

2. Requisitos de información: Se trata, aquí, de recopilar todos los datos con los que trabaja la organización y que soportan información. Hay que distinguir muy claramente lo que es documento(es soporte de información). (Laguna, 2012)

Ejemplo: El sistema guardará información sobre los socios del club, en concreto CI, nombre, apellido, teléfono, email.

2.1.2.2 Según el ámbito del requisito

Los requisitos según el ámbito en el que se desenvuelve el software se divide en:

Requisitos de hardware: Los requisitos de hardware son las características que

debe tener el hardware de una computadora para poder soportar y/o ejecutar una aplicación. Los requisitos pueden ser:

- Requisitos mínimos de hardware, que son las características mínimas (mínimo costo y mínima antigüedad) indispensables para ejecutar la aplicación correctamente.
- Requisitos recomendados de hardware, que son las características más apropiadas que debe tener el hardware para poder ejecutar una aplicación específica. Por ejemplo:
 - Requisitos de placa madre y de CPU.
 - Requisitos de memoria.
 - Requisitos de espacio disponible en disco duro.
 - Requisitos de monitor y adaptador de video.
 - Requisitos de unidades de CD, DVD, impresora, módem, tarjeta de red, puertos disponibles.

Ejemplo: Para un correcto funcionamiento del software los requisitos recomendados son:

- Procesador de 32 bits (x86) o 64 bits (x64) a 1 gigahercio (GHz) o superior.
- Memoria RAM de 1 gigabyte (GB) (32 bits) o memoria RAM de 2 GB (64 bits).
- Espacio disponible en disco rígido de 16 GB (32 bits) o 20 GB (64 bits).
- Dispositivo gráfico DirectX 9 con controlador WDDM 1.0 o superior.

Requisitos de software: Los requisitos de software son las características que debe tener el software instalado en una computadora para poder soportar y/o ejecutar una aplicación. Los requisitos de software pueden ser:

- Requisitos de sistema operativo.
- Requisitos de aplicaciones específicas instaladas.
- Requisitos de ciertas aplicaciones no instaladas en el mismo sistema.

- Requisitos de determinadas configuraciones en el sistema operativo o en ciertas aplicaciones.

Ejemplo: Requisitos para un correcto funcionamiento del software:

- Sistema operativo: Windows 7 profesional (o superior).
- Debe estar instalado: Flash Player 10 o superior.
- Debe estar instalada la máquina virtual JAVA 7 o superior.

Requisitos de sistema: Describen el comportamiento externo del sistema y sus restricciones operativas. No trata sobre el diseño e implementación del software. Sin embargo, detallar los requisitos en sistemas complejos no es una tarea fácil, existen varios motivos para esto, por ejemplo:

- Se puede requerir el diseño de una arquitectura inicial a partir de esta se organizarán los diferentes subsistemas que componen el software.
- En algunos casos el software deben inter-operar con otros ya existentes. Esto restringe el diseño, y estas restricciones imponen requisitos en el sistema nuevo. (Sommerville, Ingeniera de Software, 2005)

Ejemplo: El sistema debe enviar un email automático a los socios del club con la información de cada evento, los días lunes de cada semana a las 8 de la mañana.

2.1.2.3 Según la audiencia del requisito

Según esta clasificación de los requisitos se pueden dividir en:

Requisitos del usuario: Los requisitos del usuario están definidos en lenguaje natural que indican los servicios y restricciones del sistema. Estos requisitos deben describir los requisitos funcionales y no funcionales de tal forma que sean comprensibles por los usuarios del sistema sin tener conocimiento. Deben especificar el comportamiento externo del sistema y deben evitar especificar las características del diseño del sistema. Por consiguiente, si se están redactando requisitos del usuario, no se debe utilizar un lenguaje técnico. Se debe redactar de una forma sencilla, con formularios, tablas y diagramas que el usuario pueda entender. (Sommerville, Ingeniera de Software, 2005)

Existen diversos problemas cuando se redactan con un lenguaje natural en un documento de texto. (Sommerville, Ingeniera de Software, 2005)

- **Falta de claridad.** Algunas veces es difícil utilizar el lenguaje de forma precisa y no ambigua sin hacer el documento poco conciso y difícil de leer.
- **Confusión de requisitos.** No se distinguen claramente los requisitos funcionales y no funcionales, las metas del sistema y la información para el diseño.
- **Conjunción de requisitos.** Diversos requisitos diferentes se pueden expresar de forma conjunta como un único requisito.

Los requisitos del usuario que incluyen demasiada información restringen la libertad del desarrollador del sistema para proporcionar soluciones innovadoras a los problemas del usuario y son difíciles de comprender. Los requisitos del usuario deben simplemente enfocarse a los recursos principales que el sistema debe proporcionar. (Sommerville, Ingeniera de Software, 2005)

Para minimizar los malentendidos al redactar los requisitos del usuario, se recomienda seguir algunas pautas sencillas:

- Crear un formato estándar y asegurar que todos los requisitos se adapten al formato. Estandarizar el formato para que los requisitos sean más fáciles de verificar. Se debe incluir la información del usuario que propuso el requisito, para que en un futuro se sepa a quién consultar si se tiene que cambiar el requisito
- Se debe distinguir entre los requisitos deseables y los obligatorios. Los requisitos obligatorios son los requisitos a los que el sistema debe dar soporte y normalmente los requisitos deseables no son fundamentales y se los puede realizar a futuro.
- Resaltar el texto (con negrita, cursiva o color) para distinguir las partes clave del requisito.
- Evitar, hasta donde sea posible, el uso de términos informáticos. Sin embargo, inevitablemente se incluirán términos técnicos detallados en los requisitos del usuario.

Ejemplo: El sistema debe parametrizar las variables, de manera que no requiera de ayuda del departamento de sistemas en caso de necesitar ingresar, eliminar o actualizar las diferentes variables.

Requisitos de desarrollador: Los requisitos de desarrollador habitualmente son modelos conceptuales, como diagramas de clases de UML, ontologías² de manera que se mapee correctamente las necesidades de la organización con las funcionalidades a cubrir por el sistema. Estas metodologías propician la unión del modelado organizacional con las especificaciones de software mejorando el proceso de desarrollo.

Ejemplo: Diagrama de casos de uso para control de parqueaderos.

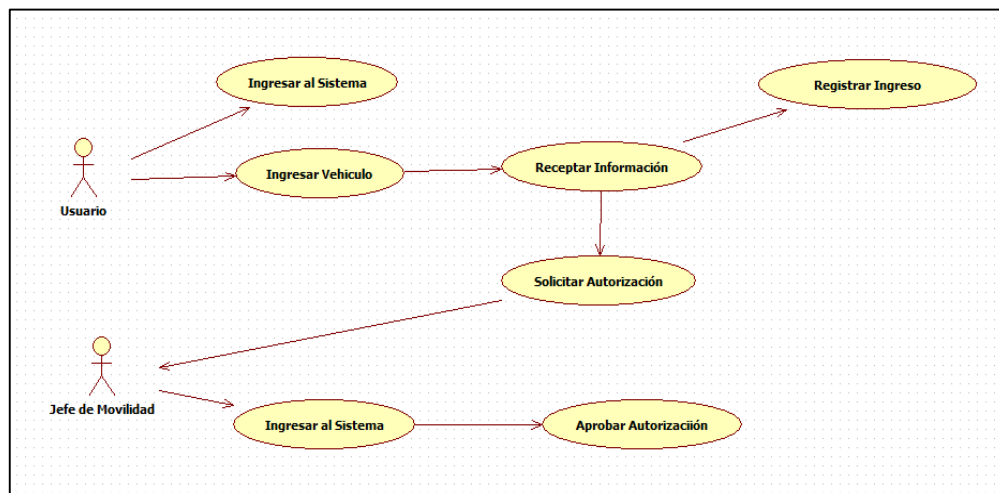


Figura 2. Diagrama de casos de uso.

2.2 PROCESOS DE LA INGENIERÍA REQUISITOS

De acuerdo al estudio de la ingeniería de requisitos de Klaus Pohl en su libro “Requirements Engineering”, propone el siguiente proceso, el cual especifica que la meta del proceso de ingeniería de requisitos es crear y mantener un documento de requisitos del sistema. El proceso general corresponde a cinco subprocesos de alto nivel de la ingeniería de requisitos, estos son: documentación, elicitación, negociación, validación y gestión de requisitos. La **Figura 3** ilustra la relación entre estos subprocesos.

² Hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios.

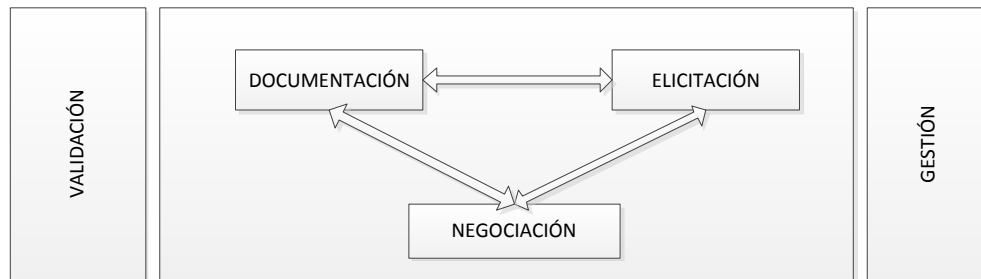


Figura 3. Proceso de ingeniería de requisitos

Fuente: (Phol K. , 2010)

Las actividades que se muestran en la **Figura 3** se refieren al descubrimiento, documentación y verificación de requisitos. Sin embargo, en casi todos los sistemas los requisitos cambian. Las personas involucradas desarrollan una mejor comprensión de lo que quieren que haga el software; la organización que compra el sistema cambia; se hacen modificaciones a los sistemas software, procesos y personal de la organización. El proceso de gestionar estos cambios en los requisitos se denomina gestión de requisitos.

2.2.1 Documentación

Esta actividad se enfoca a documentar la información importante provocada o desarrollada cuando se realiza una actividad principal (elicitación, negociación, documentación) o transversal (validación y gestión) en la ingeniería de requisitos, éstas pueden ser:

- Se obtienen los requisitos y se documentan.
- Se negocian con los participantes y se documenta.
- Se integran con el resto de la documentación.

Pohl además indica que a la par de cada actividad se debe validar y gestionar los requisitos.

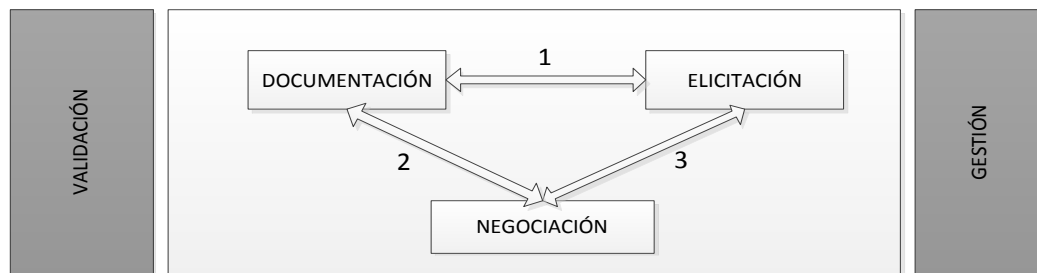


Figura 4. Relación entre las fases del proceso de ingeniería de requisitos

Fuente: (Phol K. , 2010)

2.2.1.1 Características de la Documentación

Las características de la buena documentación son (Kovits, 1998):

- **Persistencia:** Durante un proyecto, una gran cantidad de diferentes tipos de información son elicitados y desarrollados. Sin una apropiada documentación los stakeholders involucrados pueden difícilmente memorizar y reflejar toda la información.
- **Referencias comunes:** Mediante la documentación de la información relevante y poniendo a disposición de todos los stakeholders, una referencia común se prevé que puede ser accedida por todos los participantes del proyecto.
- **Promover la comunicación:** La referencia común promueve discusiones sobre los hechos documentados y por lo tanto soporta comunicación entre las partes interesadas
- **Promover la objetividad:** La diferencia del intercambio verbal de información, el intercambio de información por medio de documentos es generalmente menos susceptibles de alteración no deseada de la información, debido a la interpretación subjetiva.

2.2.1.2 Importancia de la documentación

La importancia de la documentación es (Kovits, 1998):

- **Apoyar el entrenamiento de nuevos empleados:** Si un nuevo stakeholder se une al proyecto la información documentada provee una base excelente para comenzar a familiarizarse con el proyecto. El stakeholder puede tener acceso de forma selectiva a la información relevante necesaria en una situación determinada o para realizar una tarea específica.
- **Preserva el Conocimiento Experto:** Típicamente, no todos los miembros del proyecto conocen todos los aspectos relevantes de un sistema tal como todas las tecnologías relevantes o todos los aspectos relevantes relacionados con el contexto. La documentación de los conocimientos de estos expertos hacen que este conocimiento este a disposición de todos los participantes del proyecto y por lo tanto reduce la dependencia de los expertos individuales.
- **Ayuda a reflexionar sobre los problemas:** Cuando se está documentando la información, el autor es forzado a estructurar la información en una forma apropiada por lo tanto, el autor tiene que reflexionar sobre la información que a menudo conduce a la identificación de brechas e inconsistencias.

2.2.1.3 Información para ser documentada

Cada actividad de ingeniería de requisitos produce diferentes tipos de información. Parte de la información debe ser persistente mediante la documentación de una manera apropiada. En la siguiente parte, se proporcionan algunos ejemplos de información relevante que es recolectada o creada durante la ingeniería de requisitos. (Phol K. , 2010)

- Soluciones alternativas para conflictos.
- Petición de cambio.
- Decisiones acerca de petición de cambio.
- Diferentes puntos de vista de stakeholders.
- Identificación de contradicciones.
- Identificación de errores.
- Identificación de brechas en documentos de requisitos.
- Identificación de requisitos.
- Verificación de resultados.

- Requisitos nuevos e innovadores.
- Priorización de requisitos.
- Procesos de información.
- Personas responsables de actividades.
- Resultados del uso de prototipos.
- Revisión de resultados.
- Riesgos.
- Argumentos de los stakeholders.
- Deseos y necesidades de los stakeholders.

2.2.2 Elicitación

En esta fase, los ingenieros de requisitos trabajan con los clientes y los usuarios finales del sistema para determinar el dominio de la aplicación, qué servicios deben proporcionar el sistema, el rendimiento requerido del sistema, las restricciones hardware, etc.

Existen varias ideas, intenciones o necesidades en la mente de los stakeholders, dichas necesidades se pueden especificar en lenguaje natural o usando modelos para especificación de requisitos.

“La obtención y análisis de requisitos pueden afectar a varias personas de la organización. El término stakeholder se utiliza para referirse a cualquier persona o grupo que se verá afectado por el sistema, directa o indirectamente. Entre los stakeholders se encuentran los usuarios finales que interactúan con el sistema y todos aquellos en la organización que se pueden ver afectados por su instalación. Otros stakeholders del sistema pueden ser los ingenieros que desarrollan o dan mantenimiento a otros sistemas relacionados, los gerentes del negocio y los expertos en el dominio del sistema.” (Phol K. , 2010)

La elicitación es una de las principales actividades en el proceso de la ingeniería de requisitos. Existen 3 objetivos principales para esta actividad.

1. Identificar las fuentes relevantes de los requisitos.

2. Elicitar los requisitos existentes desde las fuentes identificadas.
3. Desarrollar nuevos e innovadores requisitos.

2.2.2.1 Identificación de las fuentes relevantes de los requisitos

En cada proceso de la ingeniería de requisitos existen requisitos ya identificados por los stakeholders. Sin embargo, muchas fuentes de requisitos son inicialmente desconocidas y por lo tanto deben ser elicitados.

Si una fuente no es identificada, ésta obviamente no puede ser considerada durante el proceso de elicitación de los requisitos. Al no considerar las fuentes relevantes de requisitos durante la elicitación se va a producir una especificación de requisitos incompleta.

Para la identificación de las fuentes desconocidas pero relevantes, existen 2 procesos como se puede ver en la **Figura 5**. Primero, los ingenieros en requisitos identifican las fuentes potenciales. Segundo, los ingenieros en requisitos evalúan la relevancia de las fuentes identificadas.

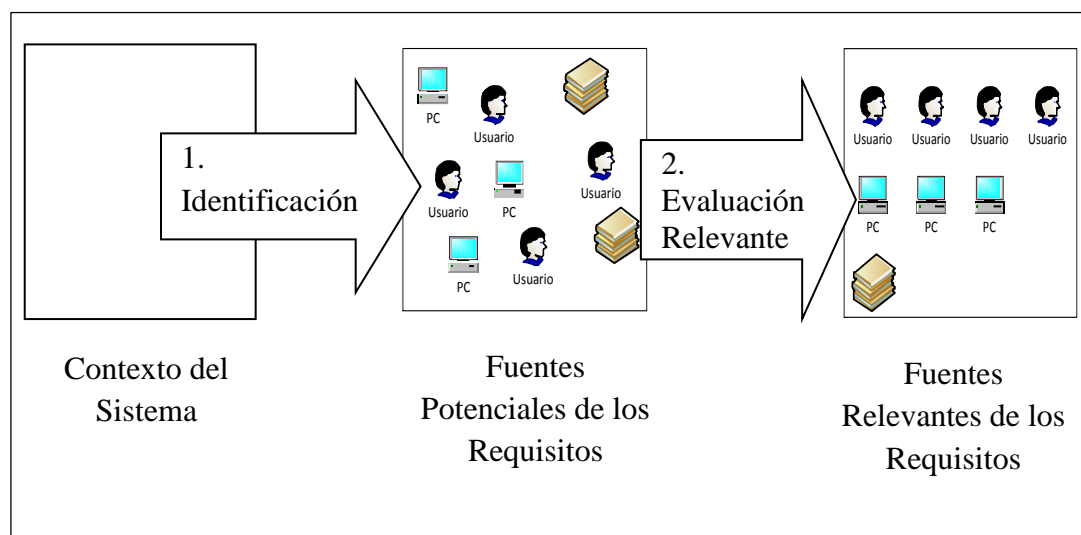


Figura 5. Identificación de fuentes relevantes

Fuente: (Phol K. , 2010)

2.2.2.2 Elicitación de los requisitos existentes

En la siguiente parte se describe como un requisito existente puede ser elicitado de una fuente relevante de requisito, como se puede apreciar en la **Figura 6**.

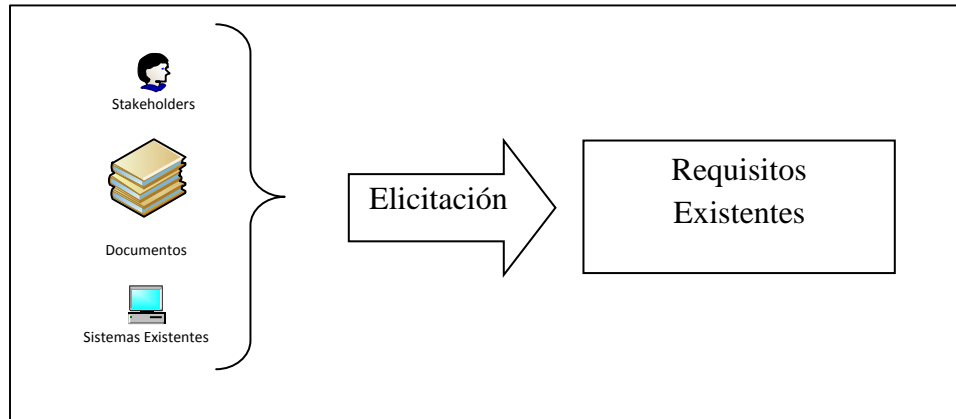


Figura 6. Fuentes de requisitos existentes

Fuente: (Phol K. , 2010)

2.2.2.2.1 Elicitación de requisitos existentes de los Stakeholders

Los requisitos existentes pueden ser elicitados de los stakeholders a través de conversaciones, cuestionario u observaciones.

En una conversación, un stakeholder cuenta su ingeniería en requisitos o sus requisitos. Las conversaciones pueden tomar lugar en una entrevista o en algún taller.

La elicitación por medio de cuestionarios, los stakeholders escriben sus propios requisitos.

Para los requisitos elicitados a través de la observación, se realiza una observación relevante a los stakeholders, los requisitos que los stakeholders no expresan por ser parte de la rutina diaria

2.2.2.2.2 Elicitación de requisitos existentes de los documentos

Los ingenieros en requisitos tienen que leer y analizar todos los documentos relevantes. Por ejemplo si se está haciendo un sistema para un vehículo, éste debe cumplir con las normas de seguridad establecidas por los diferentes estándares.

2.2.2.2.3 Elicitación de requisitos de los sistemas existentes

Este tipo de requisitos son elicitados directamente desde el sistema existente, desde los stakeholders que son familiarizados con los procesos del sistema o desde la documentación. Se obtiene requisitos de los errores que sucedían con dicho sistema.

2.2.2.2.4 Desarrollar nuevos e innovadores requisitos:

Los requisitos nuevos e innovadores no pueden ser elicitados de la misma forma que los requisitos existentes. Los nuevos requisitos tienen que ser desarrollados en un proceso creativo, la elicitación de un nuevo requisito puede ser apoyada de algunas de las técnicas creativas existentes. Ejemplo elicitación de requisitos mediante el método KJ.

2.2.2.3 Técnicas de elicitación

Existen varias técnicas para realizar el proceso de elicitación de requisitos, entre las técnicas más conocidas están las 6 que se describen a continuación:

2.2.2.3.1 Entrevistas

Fundamentalmente se darán a conocer los siguientes 3 tipos de entrevistas: entrevistas estándar, entrevistas exploratorias, entrevista no estructurada. (Oppenheim, 2000)

Durante una **entrevista estándar**, el entrevistador realiza al entrevistado una serie de preguntas previamente preparadas concernientes a un tema de interés. Independientemente de las respuestas dadas, el entrevistador no se desvía de las preguntas preparadas. Las entrevistas estándar son apropiadas cuando las opiniones de varios stakeholders coinciden en un mismo tema y deberán ser tomadas en cuenta. Los resultados de las entrevistas estándar son fáciles de comparar debido a las preguntas estandarizadas.

Una **entrevista exploratoria** es una conversación por medio de la cual el entrevistador elicitación información acerca de la opinión o punto de vista del entrevistado con respecto algún tema. La entrevista se basa en una lista de preguntas preparadas. Durante una entrevista exploratoria, el entrevistador puede desviarse de la pregunta

preparada, por ejemplo, para una investigación adicional con respecto a alguna respuesta dada por el entrevistado. El resultado de la entrevistas son cualitativas, por lo tanto, el resultado de diferentes entrevistas exploratorias concernientes al mismo tema son difíciles de comparar la una con la otra.

Las **entrevistas no estructuradas**, normalmente no hacen uso de un catálogo de preguntas preparadas. El entrevistador hace preguntas generales libremente y permite al entrevistado dirigir la conversación en la dirección que crea conveniente según su propio criterio. El resultado de diferentes entrevistas no estructuradas son muy difíciles, sino imposible, de comparar. Por lo tanto recomienda, entrevistas estándar o exploratorias para elicitación de requisitos.

Además, se puede diferenciar entre entrevistas individuales o grupales:

- **Entrevistas individuales:** Durante una entrevista individual, un stakeholder solo responde las preguntas. El resultado de una entrevista individual refleja la opinión de un solo stakeholder.
- **Entrevistas grupales:** Durante la entrevista a un grupo, un grupo de stakeholders responden las preguntas. Las respuestas de diferentes stakeholders a una pregunta influyen a los demás, ellos desarrollan una conversación participativa.

2.2.2.3.2 Taller

En contraste a las entrevistas, los requisitos no son elicitados de manera individual, son resultados de un grupo de trabajo. Si el taller es bien preparado y conducido, es una técnica muy exitosa para la elicitación de requisitos.

Según (Leffingwell & Widring, 2000) las principales tareas a realizarse en un taller son:

- **Definición del objetivo:** Se definen los objetivos del taller y el plan de trabajo. Una correcta definición del objetivo del taller logra efectos positivos en los participantes del taller.
- **Definición de procedimientos del taller:** En el contexto de un taller, normalmente diferentes técnicas son usadas, tales como: Tormenta de ideas, el

método KJ, objetivos interactivos y definición de escenarios, debates, trabajos en grupo y presentación de resultados.

Es necesario elaborar una agenda que sirva como orientación a través del taller, es aconsejable planear descansos de 5 a 10 minutos cada 45 minutos durante un taller.

- **Buscar, invitar a los participantes y pactar los objetivos.**

Un stakeholder quien pueda contribuir a lograr los objetivos del taller y por lo tanto los resultados deseados, debe ser invitado al taller. Si el número de potenciales participantes de taller es muy amplio, la técnica para evaluar los recursos relevantes para requisitos puede ser utilizada para determinar cuáles participantes son los de mayor importancia para lograr los objetivos del taller. Se recomienda invitar entre 5 a 15 participantes para un taller.

El invitar a los participantes se debe hacer con un debido tiempo de anticipación de 4-6 semanas antes del taller, asegurando así que la mayoría de invitados asistan.

Junto con la invitación al taller, el stakeholder podría ser informado de los objetivos del mismo, también puede ser informado al inicio del taller.

- **Buscar un lugar para el taller.**

Se requiere una sala lo suficientemente amplia para alojar a todos los participantes del taller, si éste incluye trabajos en sub grupos se debe tener más salas para evitar disturbios (interrupción del trabajo en grupo) entre los mismos, por ejemplo llamadas a teléfonos celulares pueden afectar disminuyendo la concentración, motivación en los participantes y por lo tanto resultados negativos.

Se debe tener equipo tecnológico adecuado para la sala en donde se realiza el taller, es recomendable una pizarra, proyectores, conexión a internet, computadores y/o impresora si es necesario.

- **Nombrar un moderador.**

El moderador tiene la tarea de guiar el taller, en particular el moderador debería detectar conflictos entre los puntos de vista de los participantes del

taller y ayudar a resolver los conflictos. Para poder guiar el taller el moderador debe conocer muy bien los objetivos del mismo. Durante el taller el moderador debe asegurarse de que las actividades del grupo contribuyen con los objetivos del mismo, de no ser así el moderador debe detener las actividades y guiarlas hacia los objetivos, sin embargo esto se debe hacer con mucho cuidado ya que se puede inhibir la creatividad de los participantes. Se recomienda un moderador externo el cual será neutral al momento de resolver conflictos entre los participantes.

- **Nombrar una persona que tome el tiempo.**

Es recomendable invitar a una persona adicional, de preferencia alguien neutral y con experiencia en llevar los tiempos y documentación del taller, sino se cuenta con un experto se debe escoger una persona entre los participantes del taller para realizar la toma de tiempo y documentación. Si no se tiene una persona encargada de esta tarea los participantes pueden tomar más tiempo del necesario para cada actividad.

2.2.2.3.3 Grupos de enfoque

El concepto clave de grupo de enfoque es que los stakeholders se centran en un tema elegido para identificar los requisitos respecto a este tema.

El tema de enfoque puede ser un tópico de interés tal como un procedimiento de trabajo en la compañía, o un ítem tangible como un sistema dado. Si el tema de enfoque es un sistema existente, los stakeholders pueden, por ejemplo, enfocarse en la identificación de problemas con el sistema actual y su uso. (Kuniavsky, 2003)

Se puede establecer tres tipos de grupos de enfoque:

- **Exploratorio:** Durante una sesión de grupo de enfoque exploratorio, el principal objetivo es la obtención de nuevos requisitos para el sistema.
- **Comparativo:** Tiene como objetivo elicitación un conjunto inicial de requisitos basados en productos de la competencia o una versión previa del sistema. El sistema existente y cualquier documentación disponible puede ser utilizada durante la sesión.

- **Priorización:** Tiene como objetivo dar prioridad a los requisitos que ya estén listos, e identificar requisitos faltantes.

2.2.2.3.4 Observación

Se elicitan los requisitos mediante la observación de stakeholders o de sistemas existentes. Elicitar requisitos por medio de la observación tiene una importante ventaja comparada con los talleres y entrevistas. Los stakeholders son capaces de proveer una mejor descripción de sus actividades mientras ellos realizan estas actividades de forma retroactiva, es decir, después de haber realizado las actividades. Existen dos tipos de observación las cuales son: Observación directa y la observación etnográfica.

Durante la **observación directa**, se vigila a los stakeholders mientras ellos realizan una tarea en particular, se analizan esas actividades, y realizan preguntas. La observación directa de un sistema existente implica que el observador vigile detenidamente a varios stakeholders usando el sistema o como el sistema opera de manera autónoma (si es capaz de hacerlo). (Beyer & Holtzblatt, 1998)

Durante la **observación etnográfica**, el observador invierte un largo periodo de tiempo con los stakeholders para aprender las actividades y entender su forma de trabajar y los procedimientos en su trabajo. Por lo tanto, el observador en lo posible realiza todo el flujo de trabajo de los stakeholders. (Kotonya & Sommerville, 1997)

2.2.2.3.5 Cuestionario

Los stakeholders anotan los requisitos del sistema por sí mismos. A diferencia de las entrevistas, el stakeholder puede elegir libremente el momento en que escribe los requisitos dentro de un plazo determinado. De la psicología de aprendizaje se sabe que al anotar las propias ideas en forma estructurada apoya la reflexión sobre las mismas.

2.2.2.3.6 Perspectiva basada en lectura

Es una técnica probada para asegurar la calidad de documentos. (Oppenheim, 2000) Durante la perspectiva basada en lectura, un ingeniero de requisitos o algún otro stakeholder (en lo posterior llamado el “lector”) lee un documento desde una perspectiva definida previamente, por ejemplo desde la perspectiva de un usuario o desde la perspectiva de un tester. Cuando se lee un documento en esta forma, el lector puede

ignorar los detalles irrelevantes para la selección de la perspectiva, por ejemplo, cuando se lee un documento desde una perspectiva de usuario, el lector debe ignorar detalles técnicos concernientes a la realización del sistema, es decir, la información técnica indirectamente relacionada para el uso del sistema. Enfocarse en una perspectiva específica conduce a mejorar los resultados en el aseguramiento de la calidad de las actividades. Desde la perspectiva basada en la lectura para orientar el análisis de los resultados de documentos existentes, es recomendable aplicar estas técnicas para la elicitación de requisitos a partir de documentos existentes. (Zowghi & Coulin, 2005)

2.2.2.4 Técnicas de asistencia para elicitación

Las técnicas de asistencia difieren con respecto al esfuerzo que se requiere para apoyar las 3 sub-actividades (desarrollar nuevos e innovadores requisitos, elicitar requisitos existentes, identificar fuentes de requisitos) para la obtención de requisitos. La siguiente **Figura 7** muestra la idoneidad de las técnicas de asistencia para la obtención de las 3 sub-actividades.

Idoneidad de las técnicas para las sub-actividades	<i>Desarrollar nuevos e innovadores requisitos</i>			
	<i>Elicitar requisitos existentes</i>			
	<i>Identificar fuentes de requisitos</i>			
Técnicas	Esfuerzo			
Lluvia de ideas	Muy bajo	✓		✓
Prototipos	Depende de la tecnología		✓	✓
Método KJ	Muy bajo	✓	✓	✓
Mapas Mentales	Muy bajo	✓	✓	✓
Elicitación de las listas de verificación	Muy bajo	✓	✓	✓

Figura 7. Idoneidad de las técnicas para las sub-actividades

Fuente: (Phol K. , 2010)

2.2.2.4.1 Lluvia de ideas

La lluvia de ideas es comúnmente considerada como una técnica creativa. El objetivo de la lluvia de ideas es generar un gran número de nuevas ideas. Esto se realiza con un grupo de stakeholders, por ejemplo durante un taller.

2.2.2.4.2 Prototipos

“Un prototipo es una versión inicial de un sistema software que se utiliza para demostrar conceptos, probar opciones de diseño y, en general, informarse más del problema y sus posibles soluciones” (Sommerville, Ingeniería de Software, 2005)

La principal ventaja de usar prototipos en la ingeniería de requisitos es que los prototipos permiten a los stakeholders experimentar directamente sus propias expectativas y los efectos de sus requisitos.

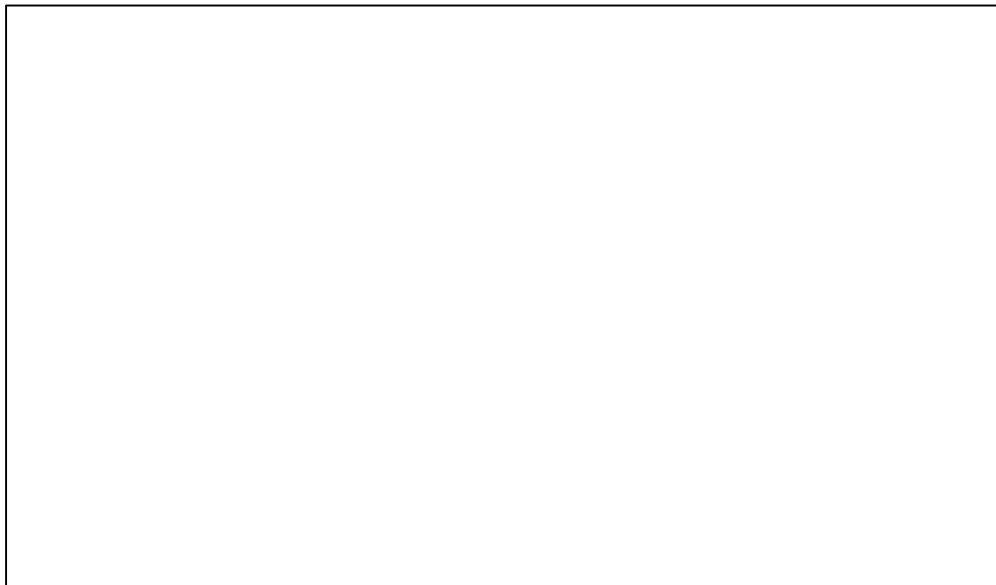
2.2.2.4.3 Método KJ

El método KJ fue originalmente desarrollado por *Kawakita* para ayudar a los grupos de stakeholders en el desarrollo de nuevas ideas. En la ingeniería de requisitos, el método es utilizado para elicitación de requisitos y fuentes de requisitos desde cada participante. Cada participante bosqueja sus ideas en un conjunto de fichas. Cada ficha debe contener palabras claves que caracterizan a un requisito.

Cuando se aplica el método KJ, cada participante tiene la posibilidad de ser escritor de sus ideas de forma independiente.

2.2.2.4.4 Mapas Mentales

En cada mapa mental, hay un tema central, los otros términos que están relacionados con el tema son organizados alrededor del tema central como las ramas de un árbol. En la **Figura 8** se puede ver un mapa mental.



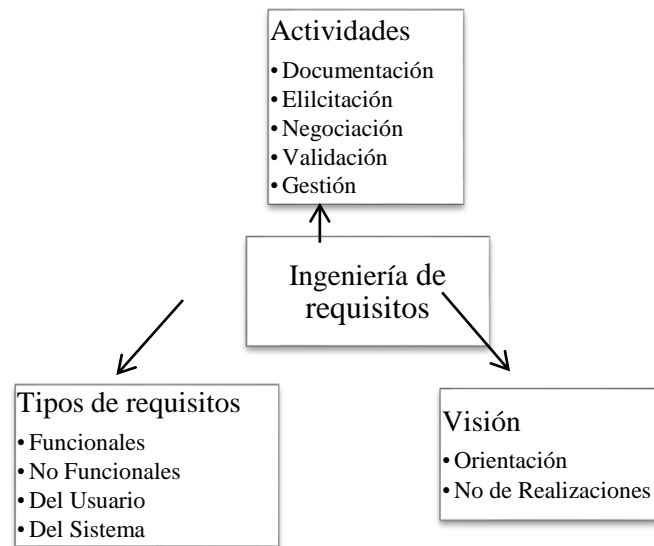


Figura 8. Mapa mental ingeniería de requisitos.

Fuente: (Phol K. , 2010)

2.3.1.1.1 Listas de verificación

Una lista de verificación contiene un número de artículos documentados en forma de preguntas o afirmaciones, que son relacionadas igual que un requisito. Las listas de verificaciones son muy útiles cuando muchos aspectos de un tema complejo tienen que ser considerados y ninguno de estos tiene que ser olvidado.

Durante la ingeniería de requisitos, las listas de verificaciones son utilizadas para otras actividades además de la elicitación de requisitos como por ejemplo la validación de los requisitos.

2.2.3 Negociación

2.2.3.1 Fundamentos de la Negociación

Objetivo de la Negociación de Requisitos

El sistema a ser desarrollado debe considerar y realizar todos los deseos y necesidades de los stakeholders. Por lo general existen conflictos cuando se realizan los requisitos. El objetivo de la negociación es lograr avances en los requisitos mediante la

identificación de los conflictos, análisis de los conflictos y la resolución de los conflictos identificados.

2.2.3.2 Gestión de Conflictos

La administración de conflictos cuenta con 4 sub actividades que son las siguientes:

- Identificar los conflictos
- Analizar los conflictos
- Resolver los conflictos
- Documentar la resolución de conflictos

2.2.3.2.1 Identificación de conflictos

Durante la ingeniería de requisitos siempre pueden surgir inconvenientes en las actividades. Se debe reconocer los conflictos, debido a que no siempre son obvios. Por ejemplo un stakeholder puede no expresar su desaprobación de un requisito de forma inmediata. Y las personas encargadas de realizar los requisitos deben prestar atención especial durante todas las actividades para detectar y documentar los posibles conflictos.

Los ingenieros en requisitos necesitan buscar activamente los conflictos que se han mantenido sin detectar durante las otras actividades de la ingeniería de requisitos.

2.2.3.2.2 Análisis de conflictos

Para determinar el tipo de conflictos, se debe realizar la siguiente clasificación de los mismos, de acuerdo a (Moore, 2003).

- **Conflictos de Datos:** Un conflicto de datos es causado por falta de información, desinformación o por diferentes interpretaciones.
- **Conflictos de Intereses:** Un conflicto de intereses es causado por intereses subjetivos u objetivos o por objetivos de los stakeholders.
- **Conflictos de Valores:** Un conflicto de valores es causado por diferentes criterios, los cuales los stakeholders aplican cuando evalúan un tema.
- **Conflictos relacionales:** Un conflicto relacional es causada por el desequilibrio de poder entre los diferentes stakeholders.

2.2.3.2.3 Resolución de conflictos

Para resolver un conflicto se puede aplicar una de las tres estrategias básicas.

- **Negociación:** Tiene como objetivo resolver el conflicto mediante el intercambio de información, argumentos y opiniones. El conflicto es resuelto cuando una de las partes del conflicto es aceptado por las otras partes para encontrar una solución. La ventaja de una resolución a través de la negociación es que todas las partes en conflicto son consideradas.
- **Solución Creativa:** En la solución creativa las partes en conflicto abandonan sus puntos de vista para desarrollar nuevos puntos de vista que sean aceptados por todas las partes. La ventaja de esta resolución es que todas las partes en conflicto encuentran una solución, la desventaja es que el desarrollo de esta solución puede tomar mucho tiempo e influir en otros requisitos.
- **Decisión:** En esta estrategia los conflictos son resueltos mediante una autoridad como puede ser el líder del proyecto o un representante del cliente. Basándose en la circunstancias la autoridad decidirá cual punto de vista será tomado en cuenta.

2.2.3.2.4 Documentar la resolución de conflictos

La documentación de la resolución de conflictos es recomendada con el fin de hacerlo trazable para las futuras actividades de ingeniería de requisitos. Cuando no se documenta la resolución, la solución alcanzada se puede olvidar y pueden volver a surgir nuevos conflictos. Por ejemplo, algún stakeholder puede preguntar por un requisito que es un conflicto con otro requisito y que será descartado durante la resolución de conflictos.

2.2.3.3 Técnicas de Negociación

Existen diferentes técnicas de negociación entre las principales están: Win-Win y Matriz de Interacción.

2.2.3.3.1 Win-Win

El objetivo de esta técnica es hacer que todos los stakeholders lleguen a ganar cuando existe un conflicto en la negociación de requisitos. La resolución de un conflicto mediante la creación de una situación de Win-Win proporciona la confianza entre las partes interesadas y también aumenta su voluntad de llegar a un compromiso. (Boehm & Ross, 1989)

2.2.3.3.2 Matriz de Interacción

Una matriz de interacción puede ser usada para visualizar y documentar los requisitos que tienen conflicto o redundancia.

Se realiza una matriz con todos los requisitos, debido a que se van a cruzar todos los requisitos entre sí.

- Cuando existe un problema de redundancia entre diferentes requisito se coloca una R.
- Cuando existe algún tipo de conflicto o contradicción entre los requisitos se coloca una C.

	Requisito 1	Requisito 2	Requisito 3	Requisito n
Requisito 1				
Requisito 2			C	
Requisito 3	R			
Requisito n				

Figura 9. Matriz de Interacción

Fuente: (Sommerville, Ingeniera de Software, 2005)

Cuando existe un problema de redundancia como se puede ver entre el requisito 1 y 3, la solución más óptima es quitar esos dos requisitos y hacer uno nuevo.

Cuando existe un problema de conflicto como se puede ver entre el requisito 2 y 3, la solución es mucho más compleja, debido a que puede ser un problema de especificación o problema de concepción del usuario final, la persona encargada de los requisitos debe acudir donde el usuario y buscar juntos una solución.

2.2.4 Validación de Requisitos

La calidad del resultado del procedimiento para la obtención de requisitos (elicitación, documentación y negociación) necesita ser validada, para comprobar la calidad de los requisitos, estos deben comunicarse de nuevo a los stakeholders con el propósito de detectar desviaciones entre el documento de requisitos y las necesidades, deseos verdaderos de los stakeholders.

2.2.4.1 Riesgos de una validación insuficiente.

Partiendo de que la obtención de requisitos sirven como referencia para el diseño, implementación, y testeo, los defectos de los requisitos afectan en las actividades del desarrollo del software. Si un error en los requisitos no es detectado hasta la etapa de liberación del sistema, todas las demás etapas del proceso de desarrollo de software (requisitos, diseño, código fuente, testeo y manuales) necesitan ser revisados.

Defectos en los requisitos podrían tener consecuencias legales. Los requisitos son a menudo parte del contrato entre el cliente y el contratista. Si estos son insuficientemente validados y sus defectos son pasados por alto, esto con el tiempo podría conducir a disputas y conflictos relativos a las obligaciones contractuales reales como la funcionalidad y la calidad garantizada.

Además, defectos como la ambigüedad, incorrección o requisitos incompletos relativos a las leyes, regulaciones, estándares que aplican al sistema y éste desarrollo puede tener graves implicaciones. Por ejemplo, si un requisito que exige el cumplimiento de un estándar específico es olvidado desde la especificación y es además pasado por alto, significa que se podría necesitar rehacer el trabajo para la aceptación del sistema.

La validación de requisitos, requiere un esfuerzo adicional durante la ingeniería de requisitos. Sin embargo, esto también reduce el coste y los riesgos causados por defectos en los requisitos de las fases de desarrollo posteriores. La **Figura 10** muestra el progreso cualitativo del esfuerzo para reparar los errores con o sin una adecuada validación de los requisitos, durante la ingeniería de requisitos. Detectando y arreglando errores en

artefactos de requisitos tempranamente evita que los defectos tengan que ser arreglados más tarde en las fases de desarrollo a un costo significativamente más alto.

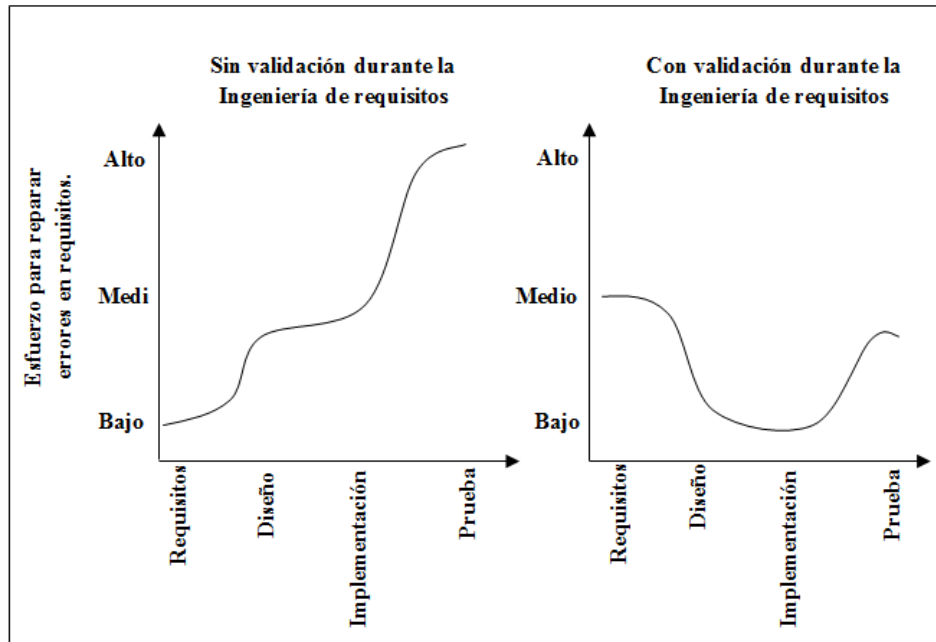


Figura 10. Progreso del esfuerzo necesario para reparar requisitos defectuosos

Fuente: (Phol K. , 2010)

2.2.4.2 Técnicas de Validación

2.2.4.2.1 Inspecciones

El objetivo de la Inspección es detectar los problemas en el documento de requisitos. Una característica importante en una inspección es el cumplimiento estricto de un proceso definido.

“Inspección: Un proceso de mejora de la calidad para el material escrito. Consta de dos componentes dominantes; Producto de mejora (documento en sí) y la mejora de los procesos (tanto de la producción y la inspección de documentos)” (Gilb & Graham, 1993)

2.2.4.2.2 Control Documental

Cuando se realizan controles documentales, el autor del requisito distribuye el requisito a un conjunto de stakeholders quienes harán un control documental individual.

Si existe algún defecto se comenta al autor del requisito el error o se crea una sesión de grupo para tratar todos los inconvenientes encontrados.

2.2.4.2.3 Tutoriales

Durante los tutoriales, el autor del requisito lo presenta a un grupo de stakeholders. Explica el requisito y señala los puntos críticos en los que le gustaría que los stakeholders comenten y así conseguir una retroalimentación y corregir cualquier inconveniente. (Phol K. , 2010)

2.2.4.2.4 Prototipos

Este tipo de validación se realiza cuando los requisitos son muy difíciles de detectar con las técnicas antes mencionadas, por esta razón los defectos de los requisitos no son detectados hasta que el usuario final interactúe con el sistema.

De acuerdo a (T., 2007), la validación de los requisitos usando prototipos es el método más efectivo para detectar defectos en los requisitos.

El principal inconveniente con este tipo de validación es el tiempo y el costo que va a tomar corregir algún tipo de problema que se presente, debido a que se tiene que volver a realizar el análisis correspondiente.

2.2.5 Gestión de los Requisitos

2.2.5.1 Fundamentos de la gestión de requisitos

Existen diferentes interpretaciones para el termino gestión de requisitos, autores como (Sommerville, Ingeniera de Software, 2005) y (Kotonya & Sommerville, 1997) consideran la gestión de cambios y sus actividades asociadas como, gestión de requisitos. Los objetivos principales son:

- Gestión del documento de requisitos.
- Observación del contexto del sistema.
- Gestión de las demás actividades de ingeniería de requisitos.

2.2.5.1.1 Gestión del documento de requisitos

Durante la ingeniería de requisitos existe una gran cantidad de requisitos que son elicitados, documentados, negociados, validados y asignados a componentes del sistema. Cada requisito ha sido especificado y utilizado para diseñar y testear las funcionalidades del sistema. El objetivo de esta actividad es mantener continuamente la pista de todos los requisitos, sus atributos relevantes, relaciones y además su evolución. (Phol K. , 2010)

2.2.5.1.2 Observación del contexto del sistema

El objetivo de esta actividad es identificar los cambios en el contexto y estimar el impacto en estos cambios. Los cambios deben ser direccionados durante la ingeniería de requisitos. Los siguientes son los cambios de contexto frecuentes:

- Una nueva tecnología o un nuevo producto surge.
- Una ley o cambio de estándar.
- Evolución de los objetivos de los stakeholders.
- Participación de stakeholders adicionales.
- Cambios en el personal de la organización.

2.2.5.1.3 Gestión de las demás actividades de ingeniera de requisitos

Tiene como objetivo supervisar, controlar y ajustar el trabajo previsto en las actividades de elicitación, documentación, negociación y validación.

Existen dos enfoques básicos para gestionar las actividades de la ingeniería de requisitos. La fase orientada al enfoque y el enfoque situacional.

Fase orientada al enfoque

En esta fase, las actividades de ingeniería de requisitos están embebidas en el proceso del modelo en cascada, con ciclos de retroalimentación y en un orden secuencial.

Enfoque situacional

En contraste con el enfoque anterior, en el enfoque situacional la actividad ejecutada depende del status del requisito y la situación actual del proceso.

El objetivo de esta actividad es determinar la secuencia de las actividades de la ingeniería de requisitos, prevenir la realización de las actividades que produzcan poco o ningún progreso. Por ejemplo, se debe evitar la secuencia de una actividad en la que los stakeholders obtengan solo la comprensión de los requisitos y llegar a pequeños acuerdos.

2.2.5.2 Trazabilidad de los Requisitos

“La trazabilidad de los requisitos se refiere a la habilidad para describir y seguir la vida de un requisito, hacia adelante y hacia atrás o en ambas direcciones (es decir desde sus orígenes, a través de su desarrollo y la especificación, hasta su posterior despliegue y uso, y a través de todos los períodos de un continuo refinamiento y la iteración en cualquiera de estas fases)” (Gotel & Finkelstein, 1994).

La calidad de la información de la trazabilidad influye en la calidad del sistema desarrollado. En la **Tabla 2** se muestran los ejemplos del uso de la información de trazabilidad durante el desarrollo de sistema.

Tabla 2

Ejemplos del uso de la información de la trazabilidad durante el desarrollo del sistema

Aspecto	Uso de la Trazabilidad
Gestión del Cambio	La trazabilidad permite analizar, en el caso de un cambio, que otros artefactos se ven afectados por dicho cambio. Además, la información de trazabilidad apoya la predicción de los esfuerzos necesarios para integrar el cambio.
Calidad de Mantenimiento y Reparación de Garantía	La Trazabilidad facilita la identificación de las causas y el impacto de los errores, la identificación de las partes del sistema afectado por un error, y el pronóstico de los esfuerzos necesarios para corregir el error.
Re-ingeniería	La Trazabilidad apoya la re-ingeniería de sistemas heredados, relacionando las funciones del sistema heredado de los requisitos para

el nuevo sistema y la documentación de los componentes del nuevo sistema cuenta con estos requisitos.

Gestión del Riesgo

La trazabilidad entre los requisitos y otros documentos es compatible con la gestión de riesgos, facilitando la identificación de los procesos de desarrollo que están potencialmente afectados por un riesgo o amenaza.

Responsabilidad

La Información de la trazabilidad se puede utilizar para asignar el esfuerzo del desarrollo a las necesidades individuales. Esto permite, determinar el esfuerzo de un grupo de stakeholders para un determinado requisito.

Mejora de Procesos

La Trazabilidad apoya la mejora de procesos. La información de la trazabilidad se puede utilizar para rastrear problemas en el proceso de desarrollo. La planificación y ejecución de acciones de mejora puede ser dirigida hacia la eliminación de las causas reales de los problemas.

Fuente: (Phol K. , 2010)

2.2.5.2.1 Tipos de relación de trazabilidad

Existen diferentes tipos de relación de trazabilidad, los 5 principales son los siguientes: (Phol K. , 1996)

- Condición.
- Contenido.
- Abstracción.
- Evolución.

Condición: Es usado para documentar que requisito define una restricción para otro requisito. Por ejemplo un objetivo puede definir una restricción para un escenario.

Contenido: Es usado para documentar las dependencias entre los contenidos de los requisitos y las otras fases del proceso de desarrollo.

Abstracción: Representa la abstracción de las dependencias entre los artefactos de los requisitos, esta relación puede ser usada para documentar un objetivo que clasifique a una serie de soluciones orientadas a los requisitos o un escenario abstracto.

Evolución: Documenta un tipo de relación temporal entre los requisitos y otros documentos desarrollados, aquí se puede documentar por ejemplo que una solución

orientada al requisito sea basada sobre un escenario, o que de un componente arquitectónico se den una serie de requisitos.

2.2.5.2.2 Documentación de la relación de trazabilidad

La trazabilidad puede ser documentada de diferentes maneras, así como de la forma textual, puede ser documentada usando hipervínculos o usando un modelo de información el cual define la estructura. (Phol K. , 2010)

Referencia textual: Esta es la forma simple de documentar una relación de trazabilidad, se incluye el identificador, autor y se incluye la fuente del artefacto.

Hipervínculos: Las relaciones también pueden ser documentadas usando hipervínculos, el hipervínculo debe ser desde la fuente del artefacto al destino del artefacto donde fue creado. Durante el análisis de la trazabilidad, los hipervínculos ayudan a ser visualizados.

Modelos: Se utiliza un modelo de información para definir y estructurar la información de trazabilidad y los tipos de relaciones. Se define un modelo específico para un proyecto y un tipo de relación de trazabilidad y es usado en el proyecto.

2.2.5.2.3 Presentación de la información de la trazabilidad

Para la presentación de la trazabilidad se puede realizar de la siguiente forma.

Matriz de trazabilidad

Las filas de una matriz de trazabilidad representan las fuentes consideradas de los requisitos. Las columnas de la matriz representan los requisitos de destino. La **Figura 11** ilustra una matriz de trazabilidad.

	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
Escenario 1	Satisface			
Escenario 2	basado en	conflicto		Satisface
Escenario 3			satisface	
Escenario 4	Conflicto			Basado en

Figura 11. Matriz de trazabilidad para varias relaciones

Fuente: (Phol K. , 2010)

2.2.5.3 Prioridad de los requisitos

Debido a la restricción y limitación de los recursos como tiempo, esfuerzo, personal no todos los requisitos pueden ser considerados durante el desarrollo del sistema con la misma intensidad y con el mismo grado de importancia.

Prioridad de un requisito

“La prioridad de un documento de requisito es la importancia del requisito con respecto a uno o varios criterios de priorización. La prioridad de un requisito puede ser determinada para cada requisito de forma aislada o por comparación de los requisitos”
(Phol K. , 2010)

2.2.5.3.1 Preparación de las actividades para la priorización.

La priorización de un conjunto de requisitos debe tener siempre un objetivo claro e involucrar todos los stakeholders relevantes (clientes, jefes de proyecto, usuarios, etc.). Existen 3 actividades las cuales deben ser realizadas antes de definir las prioridades.

- Determinación de los Stakeholders para ser involucrados en la priorización de los requisitos.
- Definir los criterios de priorización para ser usados.
- Seleccionar una técnica apropiada de priorización.

2.2.5.3.2 Técnicas para la priorización de requisitos

Cada técnica de priorización está basada sobre la evaluación de los stakeholders.

Técnica de clasificación y Top-Ten

Las técnicas simples para la priorización de los requisitos son las siguientes.
(Lauesen, 2002)

- *Clasificación Ad Hoc:* cuando la clasificación ad hoc es aplicada, los requisitos son clasificados por un stakeholder individual o por un grupo de stakeholders con respecto a escoger un criterio de riesgo.
- *Técnica Top-Ten:* cuando esta técnica es aplicada un número fijo de requisitos son aplicados (típicamente 10). La selección es determinada por un solo

criterio de selección. Por ejemplo la importancia de los requisitos para el éxito del sistema.

Un criterio de clasificación

La priorización de requisitos está basada en un solo criterio, tiene un criterio común para una clasificación y ese criterio es el grado de necesidad de un requisito. Existen las siguientes clases para priorizar la necesidad de un requisito. (IEEE Std 830, 1998)

- *Esencial*: implica que el sistema no será aceptable a menos que los requisitos sean cumplidos correctamente.
- *Condicional*: Implica que los requisitos aumentarán el sistema.
- *Opcional*: Implica que los requisitos pueden o no ser necesarios.

Clasificación Kano

Esta clasificación representa un enfoque para clasificar las características del sistema o requisitos del cliente con respecto al efecto de la satisfacción del cliente (Kano, 1984).

El punto de partida de la clasificación Kano es la clasificación de los requisitos en tres clases:

- Requisitos de calidad básica
- Requisitos de calidad mejorable
- Requisitos de sobre calidad.

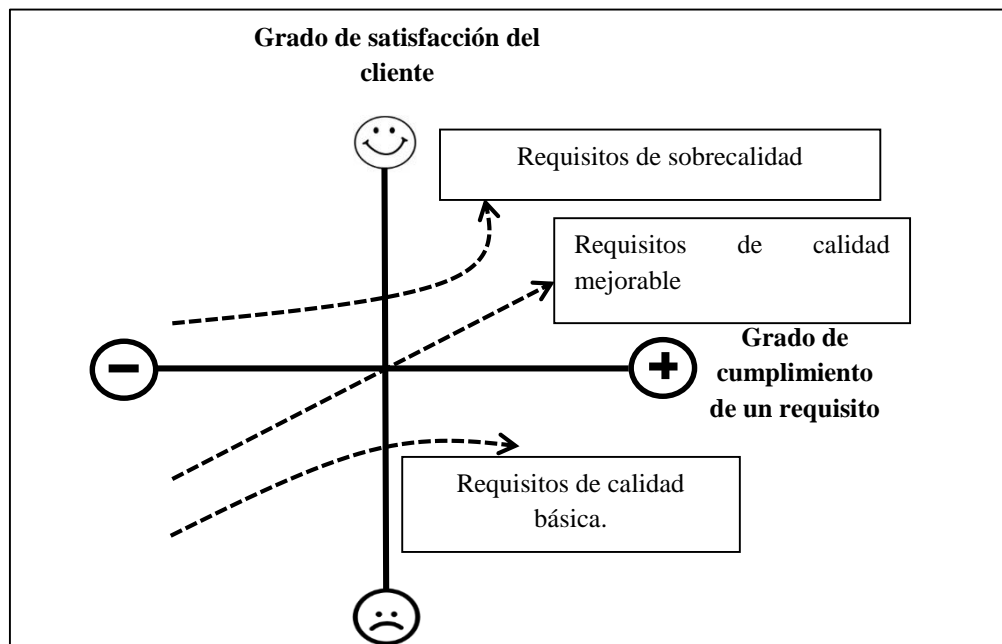


Figura 12. Clasificación Kano

Fuente: (Phol K. , 2010)

2.3 MODELOS, ESTANDARES Y NORMAS DE CALIDAD

2.3.1 Modelo Integrado de Capacidad de Madurez – CMMI

2.3.1.1 Introducción

CMMI® para Desarrollo (CMMI-DEV) consta de buenas prácticas que tratan las actividades de desarrollo aplicadas a productos y servicios. Aborda las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento.

El énfasis está en el trabajo necesario para construir y mantener el producto completo.

CMMI-DEV contiene 22 áreas de proceso. De esas áreas de proceso, 16 son áreas de proceso base, 1 es un área de proceso compartida y 5 son áreas de proceso específicas de desarrollo.

Todas las prácticas del modelo CMMI-DEV se centran en las actividades de la organización desarrolladora. Cinco áreas de proceso se centran en las prácticas específicas del desarrollo: tratando *desarrollo de requisitos*, solución técnica, integración del producto, verificación y validación.

2.3.1.2 Evolución del CMMI

El proyecto CMMI Integration se creó para resolver el problema de usar múltiples CMMs. La combinación de los modelos seleccionados en un marco de mejora único pretendía que fuera usado por organizaciones en su búsqueda de la mejora de procesos para toda la empresa.

El desarrollo de un conjunto de modelos integrados implicó más que una simple combinación de los materiales de los modelos existentes. Al usar procesos que fomentan el consenso, el Equipo del Producto CMMI creó un marco que da cabida a múltiples

constelaciones.

El primer modelo a desarrollar fue el CMMI para Desarrollo (entonces denominado simplemente “CMMI”). La **Figura 13** ilustra los modelos que condujeron a la versión 1.3 de CMMI.

Inicialmente, CMMI era un modelo que combinaba tres modelos fuente: el Capability Maturity Model for Software (SW-CMM) v2.0 draft C, el Systems Engineering Capability Model (SECM) [EIA 2002a], y el Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98.

Estos tres modelos fueron seleccionados debido al éxito en su adopción o por su prometedor enfoque para mejorar los procesos en una organización.

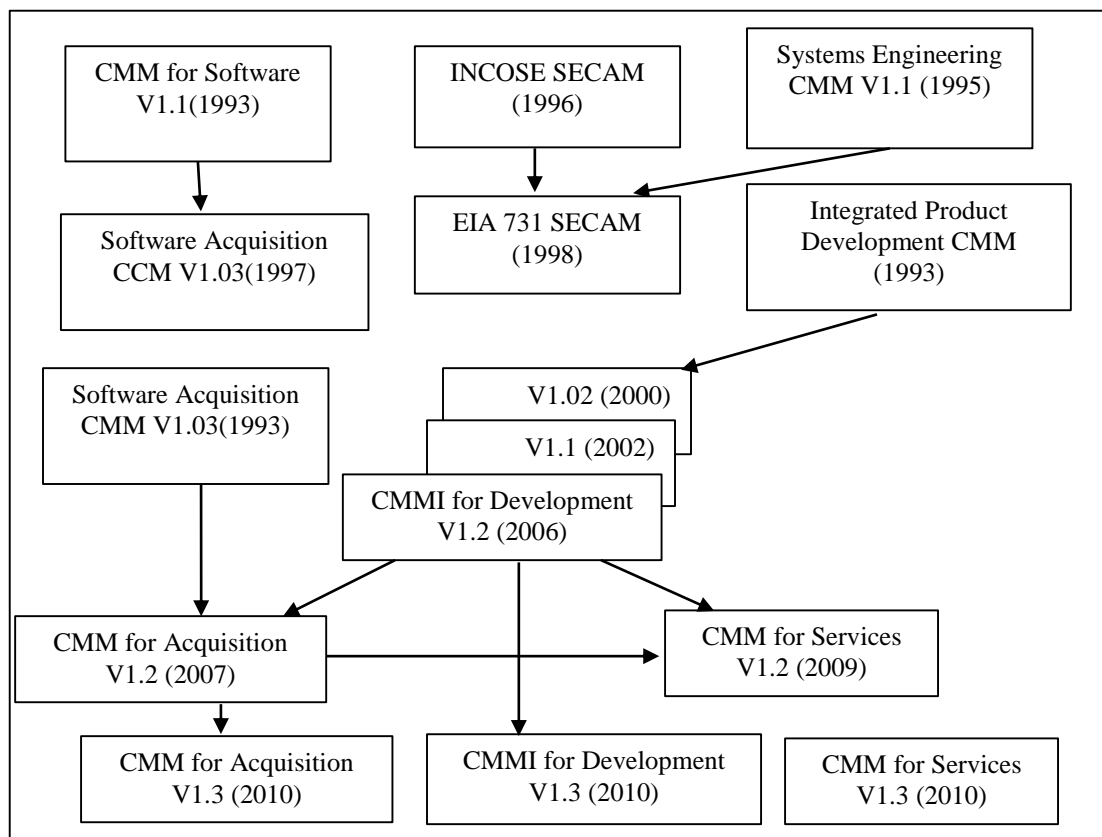


Figura 13. La historia de los CMMs

Fuente: (CMMI® para Desarrollo, 2010)

El primer modelo CMMI (V1.02) fue diseñado para usarse por organizaciones de desarrollo en su búsqueda de la mejora de procesos para toda la empresa. Fue publicado en 2000. Dos años más tarde se publicó la versión 1.1, y cuatro años después se publicó la versión 1.2.

A la vez que se publicó la versión 1.2, otros dos modelos CMMI estaban siendo planificados. Debido a estos nuevos modelos planificados, el nombre del primer modelo CMMI tuvo que cambiar y pasar a ser CMMI para Desarrollo y se creó el concepto de constelaciones.

El modelo CMMI para Adquisición se publicó en 2007. Como fue elaborado a partir de la versión 1.2 del modelo CMMI para Desarrollo, también se denominó versión 1.2. Dos años más tarde se publicó el modelo CMMI para Servicios. Como fue construido sobre los otros dos modelos, también fue denominado versión 1.2.

En 2008 se realizaron planes para comenzar a desarrollar la versión 1.3, que aseguraría la consistencia entre los tres modelos y mejoraría el material de alta madurez en todos los modelos. La versión 1.3 de CMMI para Adquisición [Gallagher 2011, SEI 2010b], CMMI para Desarrollo [Chrissis 2011] y CMMI para Servicios [Forrester 2011, SEI 2010a] se publicaron en noviembre de 2010.

2.3.1.3 CMMI para Desarrollo - CMMI-DEV

CMMI para Desarrollo es un modelo de referencia que cubre las actividades para desarrollar tanto productos como servicios. Las organizaciones de numerosos sectores, incluyendo aeroespacial, banca, hardware, software, defensa, automoción y telecomunicaciones, utilizan el CMMI para Desarrollo. CMMI para Desarrollo contiene prácticas que cubren la gestión de proyectos, la gestión de procesos, la ingeniería de

sistemas, la ingeniería de hardware, la *ingeniería de software* y otros procesos de soporte utilizados en el desarrollo y mantenimiento.

Las 22 áreas de proceso se presentan a continuación por orden alfabético de sus acrónimos en inglés:

- Análisis Causal y Resolución (CAR).
- Gestión de Configuración (CM).
- Análisis de Decisiones y Resolución (DAR).
- Gestión Integrada del Proyecto (IPM).
- Medición y Análisis (MA).
- Definición de Procesos de la Organización (OPD).
- Enfoque en Procesos de la Organización (OPF).
- Gestión del Rendimiento de la Organización (OPM).
- Rendimiento de Procesos de la Organización (OPP).
- Formación en la Organización (OT).
- Integración del Producto (PI).
- Monitorización y Control del Proyecto (PMC).
- Planificación del Proyecto (PP).
- Aseguramiento de la Calidad del Proceso y del Producto (PPQA).
- Gestión Cuantitativa del Proyecto (QPM).
- Desarrollo de Requisitos (RD).
- Gestión de Requisitos (REQM).
- Gestión de Riesgos (RSKM).
- Gestión de Acuerdos con Proveedores (SAM).
- Solución Técnica (TS).
- Validación (VAL).
- Verificación (VER).

2.3.1.4 Gestión de requisitos en CMMI-DEV v1.3

El área de proceso de “Requirements Management” (REQM) corresponde al nivel 2 de CMM, en la representación por etapas y está ubicada dentro de la categoría de

proceso de gestión de proyectos. Su propósito es gestionar los requisitos de los productos y de los componentes del producto del proyecto, e identificar inconsistencias entre esos requisitos y los planes y productos de trabajo del proyecto.

Las prácticas de REQM son la base para la definición y ejecución de las actividades del proyecto y constituyen la entrada para la planificación del proyecto. Básicamente aquí se garantiza que los requisitos son válidos, tanto por el contenido como por la fuente de donde se originan, que están comprometidos y que si son modificados se realiza de manera controlada. Para poder determinar el impacto de los cambios es imprescindible contar con un plano de los requisitos hacia las diferentes entidades del producto y a la inversa con el objetivo de obtener una rastreabilidad de los requisitos.

Cuando se presentan cambios a requisitos, pueden presentarse inconsistencias particularmente por problemas de comunicación dentro del equipo que desconoce los cambios que han sido aprobados. REQM debe identificar esas situaciones y tomar acciones correctivas para garantizar su solución.

2.3.1.4.1 Actividades según CMMI-Dev para gestionar los requisitos

SG1 Los requisitos son gestionados y las inconsistencias con los planes y productos de trabajo del proyecto son identificadas.

- SP1.1 Desarrollar una comprensión del significado de los requisitos con los proveedores de los requisitos.
- SP1.2 Obtener el compromiso de los participantes de proyecto sobre los requisitos.
- SP1.3 Gestionar los cambios a los requisitos a medida que evolucionan durante el proyecto
- SP1.4 Mantener la trazabilidad bidireccional entre requisitos y productos de trabajo
- SP1.5 Asegurar que planes del proyecto y productos de trabajo permanecen alineados con requisitos.

2.3.1.2 Desarrollo de requisitos en CMMI Dev v1.3

El área de proceso de Requirements Development (RD) corresponde al nivel 3 en la representación por etapas y está ubicada dentro de la categoría de proceso de ingeniería. Su objetivo es producir y analizar los requisitos del cliente, del producto y de los componentes del producto.

Las prácticas definidas en esta área de proceso permiten determinar todos los requisitos del proyecto, ya sea para el desarrollo o mantenimiento. Parte de los requisitos del cliente que son derivados en requisitos del producto hasta refinarlos al nivel de requisitos de los componentes del producto, todo esto durante el ciclo de vida del producto. El proceso de obtención y refinamiento de los requisitos va estrechamente vinculado con el diseño del producto por lo que estas prácticas están relacionadas, de manera recursiva, con las establecidas en el área de proceso de Technical Solution (TS).

En la **Figura 14** se muestra como los cambios que se requieran a los requisitos establecidos en RD son gestionados por REQM, por lo que RD y REQM están estrechamente relacionados y operan de manera concurrente. La razón por la que REQM está en un nivel de madurez inferior a RD puede estar relacionado con el principio de CMMI que establece que primero hay que establecer un control y orden en las actividades que se realizan, esto se logra con REQM, para después especializar la forma en que se realizan las actividades, en este caso las prácticas de RD.

RD establece en las primeras metas específicas la definición de los requisitos del cliente y los requisitos del producto y componentes. Con la meta tres (definición de requisitos de componentes) complementa las dos anteriores (Requisitos de cliente y de producto) para garantizar que todos los requisitos obtenidos son adecuadamente analizados y validados durante las diferentes etapas del desarrollo del producto.



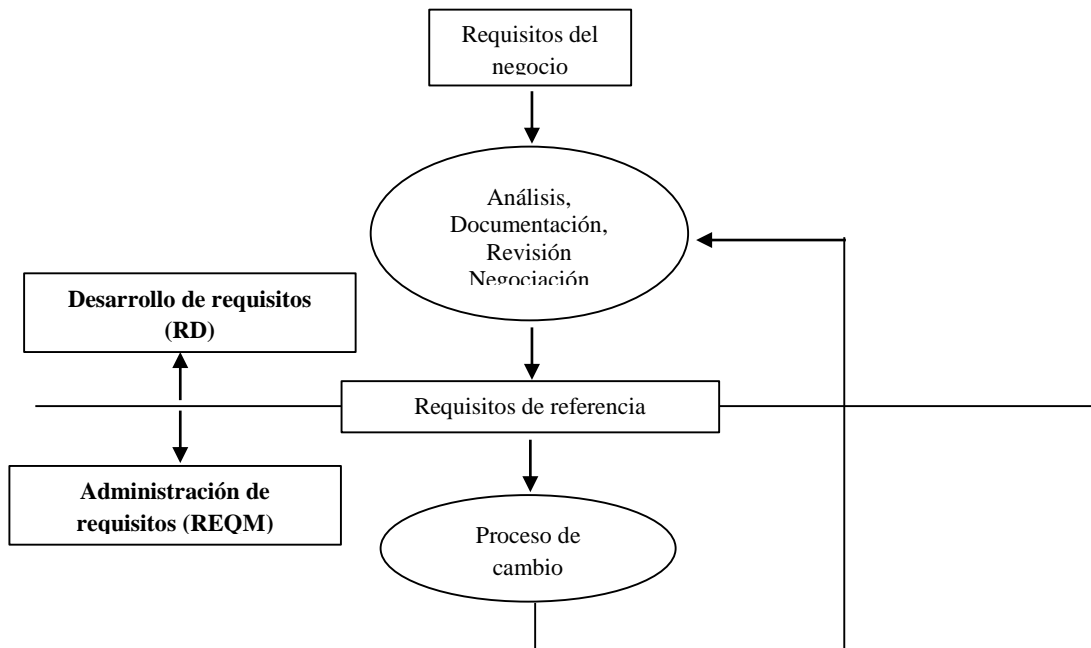


Figura 14. Proceso de cambio en requisitos gestionados por requirements management

Fuente: (CMMI® para Desarrollo, 2010)

2.3.1.2.1 Actividades según CMMI-Dev para Desarrollo de requisitos

Definir los requisitos del cliente

SG1 Las necesidades, expectativas, restricciones e interfaces de los interesados son recogidas y traducidas a requisitos del cliente.

- SP1.1 Obtener las necesidades, las expectativas, las restricciones, y las interfaces de los interesados para todas las fases del ciclo de vida del producto.
- SP1.2 Transformar las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos del cliente.

Derivar los requisitos del producto y componentes del producto

SG2 Los requisitos del cliente son refinados y elaborados para desarrollar los requisitos del producto y de componentes del producto.

- SP2.1 Establecer y mantener los requisitos del producto y de componentes del producto, los cuales están basados en los requisitos del cliente.

- SP2.2 Asignar los requisitos para cada componente del producto.
- SP2.3 Identificar los requisitos de la interfaz.

Analizar y validar los requisitos definidos

SG3 Los requisitos son analizados y validados, y una definición de la funcionalidad requerida es desarrollada.

- SP3.1 Establecer y mantener los conceptos operativos y los escenarios asociados.
- SP3.2 Establecer y mantener una definición de la funcionalidad requerida y los atributos de calidad.
- SP3.3 Analizar los requisitos para asegurarse de que son necesarios y suficientes.
- SP3.4 Analizar los requisitos para equilibrar las necesidades y las restricciones de los interesados.
- SP3.5 Validar los requisitos para asegurar que el producto resultante se ejecutará según lo previsto en el entorno del usuario.

2.3.1.3 Ventajas y Desventajas

2.3.1.3.1 Ventajas

El Modelo Integrado de Madurez de la Capacidad para el desarrollo (CMMI-DEV) entre sus principales ventajas cuenta con las siguientes:

- **Mejora la planificación:** Permite que se establezcan planes más realistas y de acuerdo a lo que la empresa es capaz de hacer.
- **Reduce el Re-trabajo:** Reduce el re-trabajo al mejorar la planificación y seguimiento, la comunicación, las responsabilidades, y la detección temprana de errores.
- **Mejora la calidad del producto:** Con una apropiada obtención de requisitos, la detección temprana de errores, uso de inspecciones y pruebas, la rastreabilidad de los requisitos, la planificación y seguimiento.

2.3.1.3.2 Desventajas

Algunas de sus debilidades son:

- El CMMI puede llegar a ser excesivamente detallado para algunas

organizaciones especialmente las medianas y pequeñas empresas.

- Puede ser considerado como un mecanismo de control y no como una herramienta útil para simplificar el proceso de obtención de requisitos.
- El alto costo de implantación que exige tanto económico como en esfuerzo puede ser considerado simplemente demasiado grande para pequeñas organizaciones.

2.3.2 ISO/IEC 25030:2007 Software Engineering

La serie de estándares del SQuaRE tiene varias divisiones bajo el título general Software product Quality Requirements and Evaluation (Requisitos y Evaluación de la calidad de los productos de software): (Standardization, 2007)

Este estándar internacional es una revisión de la ISO/IEC 9126-1: 2001, e incorpora las mismas características de la calidad del software con algunas mejoras.

Este estándar internacional se conforma con los procesos técnicos definidos en la ISO/IEC 15288:2002 relacionado con la definición y el análisis de los requisitos de calidad. La Figura 15 a continuación ilustra la organización de las series de SQuaRE, representado en divisiones. Las divisiones dentro del modelo SQuaRE, son:

Quality Requiriments Division 2503n	Quality Model Division 2501n	Quality Evaluation Division 2504n
	Quality Management Division 2500n	

	Quality Measurement Division 2502n	
--	--	--

Figura 15. Organización de la serie de estándares de SQuaRE

Fuente: (Standardization, 2007)

2.3.2.1 ISO/IEC 2503 – Requisitos de calidad

De acuerdo con el criterio de SQuaRE adoptada sobre los requisitos de calidad, el sistema de software es generalmente parte de un sistema mayor y más complejo, los requisitos de software y requisitos del sistema están estrechamente relacionados y los requisitos de software no pueden considerarse de forma aislada. En consecuencia, es importante tener en cuenta los requisitos de calidad de software a principios de ciclo de vida, como una parte importante de la especificación de los requisitos del software.

El modelo de calidad está organizado jerárquicamente en características y sub-características, hasta que los atributos sean los elementos medibles. Los atributos especifican los requisitos de calidad del software en términos de medidas y valores.

Los requisitos de calidad de software tienen tres puntos de vista diferentes: la calidad de los requisitos internos, requisitos externos y calidad del uso (ver **Figura 16 y 17**). (Standardization, 2007).

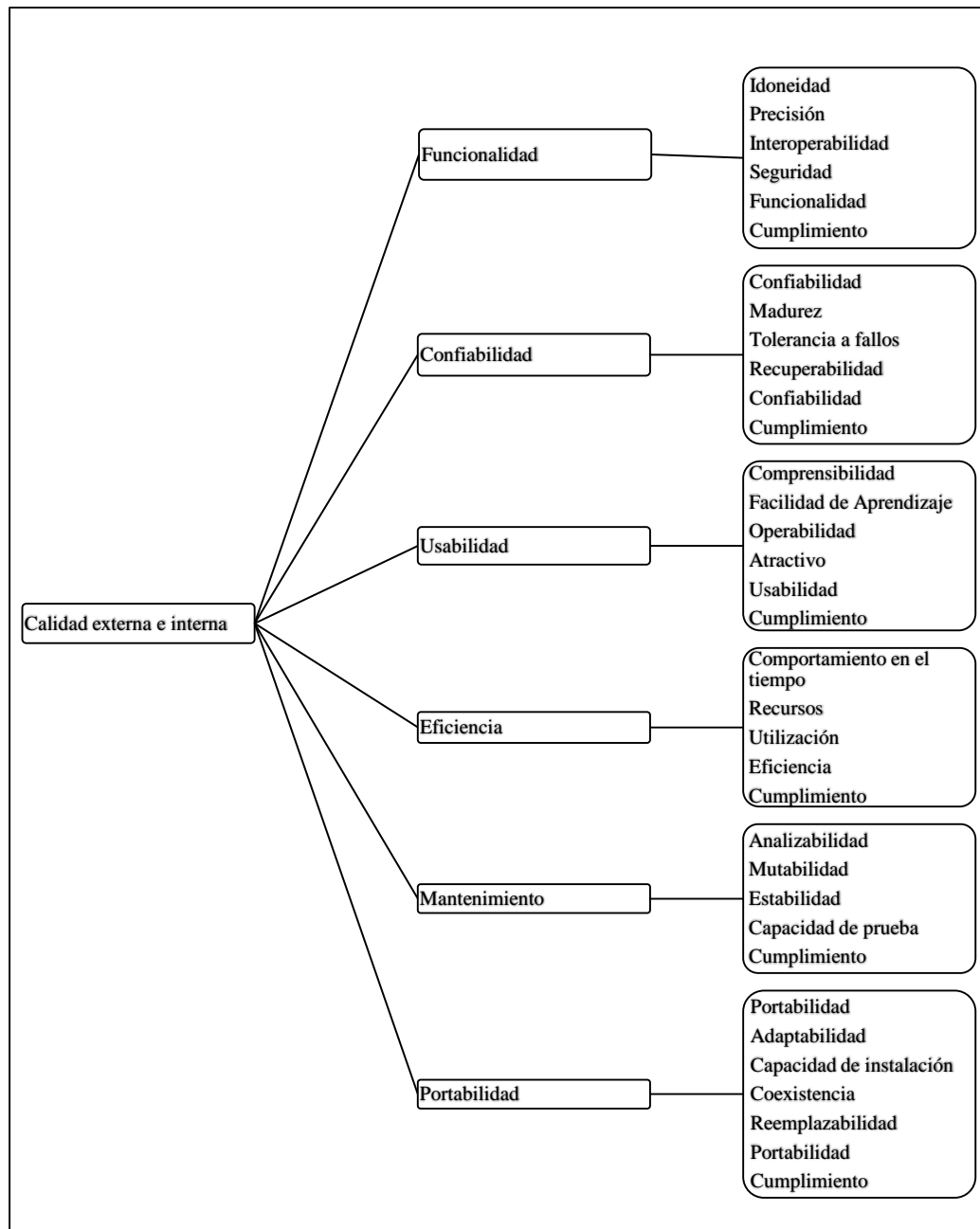


Figura 16. Modelo de calidad interno y externo

Fuente: (International Organization for Standardization, 2001)

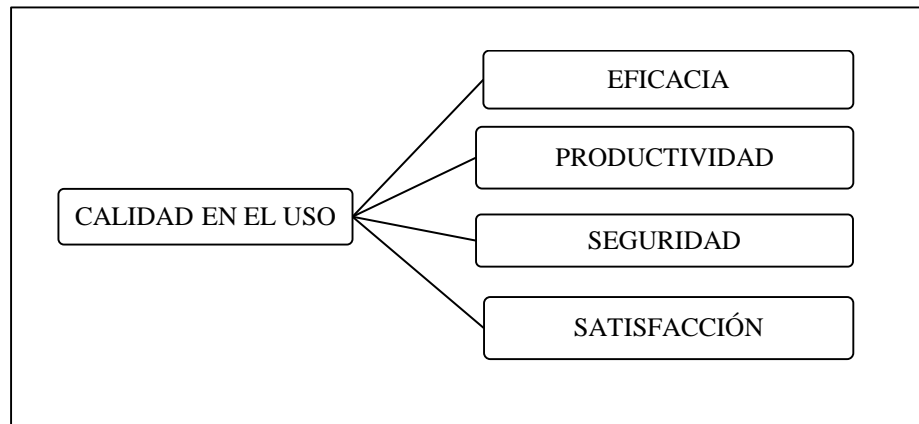


Figura 17. Modelo de calidad en el uso.

Fuente: (International Organization for Standardization, 2001)

La norma proporciona recomendaciones y orientaciones para especificar estos requisitos, esto debe contribuir a garantizar que se ajustan a las necesidades de las partes interesadas, declaraciones claras, precisas, correctas, completas, coherentes, verificables y medibles. La norma ISO/IEC 25030 - Requisitos de calidad ayuda a especificar los requisitos de calidad, ya sea durante la obtención y análisis de requisitos de calidad del producto software, o como insumo para el proceso de evaluación, y es un complemento útil a la norma ISO / IEC 9126-1.

En particular, proporciona una guía para identificar los requisitos de calidad de software, para validar la integridad de la especificación de requisitos y determinar la garantía de calidad y criterios de aceptación para un producto software.

Los requisitos del sistema son una formulación precisa de las necesidades de los stakeholders, estos requisitos son considerados como el punto de vista técnico de los requisitos, los requisitos del sistema son verificables y tienen las características que el sistema debe poseer para cumplir con los requisitos de los stakeholders. Los requisitos del sistema pueden incluir requisitos de software, hardware, datos, sistema mecánico, la organización empresarial humana, etc. Además puede provenir de una variedad de partes interesadas, incluidos los usuarios finales, las organizaciones y los organismos oficiales.

La ISO/IEC 25030 se centra principalmente en los requisitos de software (ver **Figura 18**). Los requisitos de los productos de software incluyen los requisitos

funcionales y de calidad (requisitos de propiedades inherentes) y los requisitos de gestión (requisitos de propiedad asignados).

Requisitos de sistema	Requisitos de software	Requisitos de productos de software	Requisitos de propiedad inherente	Requisitos Funcionales	
				Requisitos de calidad del software	Requisitos de calidad en uso
					Requisitos de calidad interna
			Requisitos de calidad externa		
			Requisitos de propiedad asignada	Requisitos de alto nivel impuestos por la alta gerencia, leyes vigentes, tal como impuestos ej. Impuesto I.V.A.	
		Requisitos de desarrollo de software	Requisitos del proceso de desarrollo		
	Requisitos de la organización de desarrollo.				
	Otros requisitos del sistema	Incluye por ejemplo requisitos para el hardware del computador, datos, partes mecánicas y procesos de negocio.			

Figura 18. Categorización de los requisitos

Fuente: (Standardization, 2007)

Los Requisitos funcionales incluyen los requisitos específicos del dominio de aplicación así como también un tipo de requisitos de calidad. Los requisitos de calidad también pueden implicar:

- Requisitos arquitectónicos y estructurales.
- Requisitos de propiedad asignados. Representan los requisitos que se pueden cambiar sin cambiar el software, incluyendo, por ejemplo requisitos de precio, fecha de entrega, el futuro producto y proveedor de productos, por lo que estos por lo tanto no se consideran una característica de la calidad del software.

De acuerdo con la norma ISO /IEC25030:

- Los requisitos funcionales determinan lo que el software es capaz de hacer.
- Los requisitos de calidad determinar qué tan bien funciona el software.

En otras palabras, los requisitos de calidad muestran el grado en que el software es capaz de proporcionar y mantener sus servicios especificados.

2.3.3 Estándar IEEE 830 -Especificación de Requisitos Software

2.3.3.1 Introducción

El estándar 830-1998 fue generado por un equipo de trabajo del IEEE, su finalidad es la integración de los requisitos del sistema desde la perspectiva del usuario, cliente y desarrollador.

El contenido y la caracterización de una buena especificación de requisitos de software (SRS) se describe y representa en varios esquemas SRS. El objetivo es especificar los requisitos de software a desarrollar, pero también se puede aplicar para ayudar en la selección de los productos internos y comerciales del mismo. (IEEE Std 830, 1998)

2.3.3.2 Características importantes

Las características deseables para una buena especificación de requisitos software que se indican en el IEEE son las siguientes:

- Correcta.
- No ambigua.
- Completa.
- Verificable.
- Consistente.
- Clasificada.
- Modificable.
- Explorable.
- Utilizable durante las tareas de mantenimiento y uso.

2.3.3.2.1 Corrección

La ERS es correcta si y sólo si todo requisito que figura en ella refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.

2.3.3.2.2 Ambigüedad

Un documento es no ambiguo si y solo si cada requisito descrito tiene una única interpretación. Cada característica del producto final debe ser descrita utilizando un término único y, en caso de que se utilicen términos similares en distintos contextos, se deben indicar claramente las diferencias entre ellos. Incluso se puede incluir un glosario en el que indicar cada significado específicamente.

2.3.3.2.3 Completitud

Una ERS es completa si:

- Incluye todos los requisitos significativos del software (relacionados con la funcionalidad, ejecución, diseño, atributos de calidad o interfaces externas).
- Existe una definición de respuestas a todas las posibles entradas, tanto válidas como inválidas, en todas las posibles situaciones.
- Cumple con el estándar utilizado. Si hay alguna parte del estándar que no se utiliza, se debe razonar suficientemente de por qué no se ha utilizado dicho apartado.
- Aparecen etiquetadas todas las figuras, tablas, diagramas, etc. Así como definidos todos los términos y unidades de medida empleados

La ERS debe ser siempre completa, aunque en ocasiones esto no será posible. Por ejemplo si todavía no se han determinado los formatos de los informes finales o por cualquier razón se está esperando la publicación de un Real Decreto o un reglamento sobre impuestos.

2.3.3.2.4 Verificabilidad

Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina puedan chequear que el software satisface dicho requerimiento.

2.3.3.2.5 Consistencia

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

2.3.3.2.6 Clasificación

No todos los requisitos son igual de importantes. Lo ideal es el establecimiento de prioridades, de modo que la implementación de un requisito de menor prioridad no emplee excesivos recursos

2.3.3.2.7 Modificabilidad

Una ERS es modificable si cualquier cambio puede realizarse de manera fácil, completa y consistente. Para ello, es deseable tener una organización coherente y fácil de usar en la que aparezca el índice o una tabla de contenidos fácilmente accesible.

2.3.3.2.8 Explorabilidad

Una ERS es explorable si el origen de cada requerimiento es claro tanto hacia atrás (origen que puede ser un documento, una persona etc.) como hacia delante (componentes del sistema que realizan dicho requisito).

Utilizable durante las tareas de mantenimiento y uso

En la ERS también se deben tener en cuenta las necesidades de mantenimiento. El personal que no ha intervenido directamente en el desarrollo debe ser capaz de encargarse de su mantenimiento. Así, dicha ERS actúa a modo de plano de la aplicación, permitiendo incluso modificaciones que no requieran un cambio en el diseño.

2.3.3.3 Plantilla (IEEE Std 830, 1998)

Este estándar consta con una plantilla que contiene lo siguiente:

a. Introducción

- a.1. **Propósito:** Definición de por qué y para qué se realizará el análisis y desarrollo del software.
- a.2. **Ámbito del sistema:** Descripción del entorno actual del sistema.
- a.3. **Definiciones, acrónimos y abreviaturas:** Palabras claves referentes al desarrollo del mismo que le puedan ayudar al lector del documento a tener una idea más clara de lo que se pretende.
- a.4. **Referencias:** Bibliografía que se va a utilizar en la elaboración del documento.
- a.5. **Visión general del documento:** Muestra una pequeña narración donde se cuenta todo lo que se hará y encontrará dentro del mismo.

b. Descripción general

- b.1. **Perspectiva del producto:** Es lo que se espera obtener al final del proceso según los requisitos del cliente.
- b.2. **Funciones del producto:** Presenta lo que cada aplicación o interfaz del producto hará o estará definido.
- b.3. **Características de los usuarios:** Características de los usuarios que van a manejar o a interactuar directa o indirectamente con dicha aplicación.
- b.4. **Restricciones:** Algunas prohibiciones de la aplicación que se harán a algunos usuarios.
- b.5. **Suposiciones y dependencias:** Algunas características que pueden depender más adelante de alguna.
- b.6. **Requisitos futuros:** Son algunas pautas que el cliente puede llegar a pedir en un momento dado.

c. Requisitos específicos

En esta sección se describe la funcionalidad del sistema desde dos perspectivas.

Requisitos no funcionales

Es la descripción de la forma cómo operan las entradas y las salidas del sistema. Se complementa, mas no se repite lo señalado anteriormente.

Requisitos funcionales

Describen las acciones fundamentales, sobre procesamiento para las entradas y la generación de salidas. Es una lista sobre las cosas que se deben hacer.

- c.1. **Interfaces externas:** Se capturan los requisitos que describen cómo debe ser la comunicación del sistema con el usuario y el mundo exterior.
- c.2. **3.2 Funciones:**
 - c.2.1. Roles de los usuarios en el sistema
 - c.2.2. Requisitos funcionales del sistema
- c.3. **Requisitos de rendimiento:** Son requisitos asociados con tipos de respuesta, almacenamiento y demás.
- c.4. **Restricciones de diseño:** Conocidos también como requisitos no funcionales, es decir, no afectan directamente el sistema.
- c.5. **Atributos del sistema:** Requisitos en los que el cliente desea que se desarrolle dicha aplicación, por ejemplo: plataforma.
- c.6. **Otros requisitos:** El usuario más adelante puede pedir algunos otros requisitos que podrán ser redefinidos para el desarrollo del Software

d. Apéndices

- d.1. Anexos

2.3.3.4 Objetivo del Estándar IEEE 830 en la especificación de requisitos

El cliente debe participar activamente en la especificación de requisitos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo. Así el cliente se siente partícipe del propio desarrollo.

Ayuda a los desarrolladores a entender qué quiere exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que quiere. La ERS (Especificación de Requisitos Software) permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costos de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.

Se pueden desarrollar sus propios estándares para definir las necesidades que más le sean convenientes.

Una buena especificación de requisitos software ofrece una serie de ventajas entre las que destacan el contrato entre cliente y desarrolladores, la reducción del esfuerzo en el desarrollo debido a que el cliente especifica lo que tiene que realizar, una buena base para la estimación de costos y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados.

Esta especificación debe ser parte de la documentación asociada al software que se está desarrollando, por tanto debe definir correctamente todos los requisitos, pero no más de los necesarios. Esta documentación no debería describir ningún detalle de diseño, modo de implementación o gestión del proyecto, ya que los requisitos se deben describir de forma que el usuario pueda entenderlos. Al mismo tiempo, se da una mayor flexibilidad a los desarrolladores para la implementación.

Recomendaciones

Se debe utilizar un lenguaje entendible en cada uno de los puntos del estándar, para evitar problemas de ambigüedad con los términos utilizados. Se deben describir los requisitos necesarios, no deben ser muy pocos y tampoco muy exagerados, debido a que pueden surgir problemas al momento de ponerlos en práctica.

Se debe diferenciar entre los requisitos funcionales y los no funcionales en el momento que se realice la parte de los requisitos específicos. Se debe realizar una especificación más detallada en los requisitos funcionales, como va a operar el sistema, por su parte los requisitos no funcionales solo describen la forma de cómo va a funcionar el sistema (fiabilidad, tiempo de respuesta, etc.) aquí no se debe repetir lo que se señaló en los requisitos funcionales.

2.3.3.5 Ventajas y Desventajas

2.3.3.5.1 Ventajas

Ayuda a tener un los requerimiento bien especificados en un contrato, lo que se va a desarrollar y los puntos concretos que se van a realizar.

Reducción en el esfuerzo del desarrollador en tratar de descifrar lo que el cliente solicita y forma parte de la documentación asociada al software.

Tiene una buena planificación, lo que hace que sea un proceso ordenado y a su vez contribuye con la reducción de tiempo y costos.

Se tiene una base para seguir mejorando, si en algún momento se lo necesita.

2.3.3.5.2 Desventajas

El estándar es antiguo y debe adecuarse a las nuevas metodologías, tecnologías, etc.

No especifica claramente cómo se debe realizar un estudio para la obtención de los requisitos, solamente ayuda a la especificación de los mismos los cuales fueron analizados en un paso anterior.

CAPÍTULO 3: PLAN DE INVESTIGACIÓN DE CAMPO

3.1 Objetivo General

Investigar las prácticas de la ingeniería de requisitos en las empresas de desarrollo de software de la ciudad de Quito y generar una propuesta metodológica para la mejora de dichas prácticas.

3.2 Objetivos Específicos

- Evaluar la situación actual de las prácticas en ingeniería de requisitos en las empresas de desarrollo de software de la ciudad de Quito.
- Conocer las técnicas utilizadas por las empresas o departamentos de Software de la ciudad de Quito, en cada una de las fases de la ingeniería de requisitos

3.3 Justificación

La investigación tendrá como objetivo generar una propuesta metodológica que contribuya a mejorar las prácticas de ingeniería de requisitos en la industria de Software en el Ecuador, considerando una muestra significativa de organizaciones y empresas dedicadas al desarrollo de software. Para el efecto, se realizará una investigación aplicada, exploratoria y cuantitativa, que permita conocer la situación real, entorno a las prácticas de la ingeniería de requisitos en la ciudad de Quito.

3.4 Metodología de la investigación de Campo

Se utilizará la Metodología de la Investigación Científica, la cual es definida como “la ciencia que aporta un conjunto de métodos, categorías, leyes y procedimientos que garantizan la solución de los problemas científicos con un máximo de eficiencia.” (Infomed, 2009)

3.4.1 Tipos de investigación

Según (Creswell, 2003), la investigación puede ser de los siguientes tipos:

- A. Por la forma en que la investigación es usada: *Básica o Aplicada*.
- B. Por el propósito del estudio: *Exploratoria, Descriptiva o Explicativa*.
- C. Por la técnica de recolección de datos: *Cualitativa, Cuantitativa o Mixta*.

D. Por el tiempo en evaluación de la investigación: *Transversal, Longitudinal o Estudio de Caso.*

La investigación aplicada.- es definida como “Aquella que incluye cualquier esfuerzo sistemático y socializado por resolver problemas o intervenir situaciones, aunque no sea programático, es decir, aunque no pertenezca a una trayectoria de investigaciones descriptivas y teóricas. En ese sentido se concibe como investigación aplicada tanto la innovación técnica, artesanal e industrial como la propiamente científica.” (Padrón, 1999) El objetivo del trabajo de fin de carrera es el generar una propuesta de mejora en las prácticas de ingeniería de requisitos dirigida a la industria de software de la ciudad de Quito.

La investigación exploratoria.-“Los estudios exploratorios permiten aproximarse a fenómenos desconocidos, con el fin de aumentar el grado de familiaridad y contribuyen con ideas respecto a la forma correcta de abordar una investigación en particular.” (Grajales, 2001) Tomando en cuenta que no existen trabajos de investigación sobre ingeniería de requisitos en la industria de software en la ciudad de Quito, el presente trabajo de fin de carrera se aproximará a conocer las prácticas actuales de ingeniería de requisitos en la industria de software y contribuirá con una nueva propuesta acerca de cómo se debería abordar investigaciones posteriores.

La investigación cuantitativa.-“El objetivo de una investigación cuantitativa es el de poder adquirir conocimientos fundamentales y la elección del modelo más adecuado que nos permita conocer la realidad de una manera más imparcial, ya que se recogen y analizan los datos a través de los conceptos y variables” (Juan, 2000) , De tal forma permitirá examinar los datos en forma numérica. Nivel de usabilidad – porcentaje - de uso de mejores prácticas y estándares internacionales.

La investigación transversal.-Investigaciones transversales son: “Investigaciones en las cuales se obtiene información de objetivo de estudio (población o muestra) una única vez en un momento dado”, ya que se obtendrá información de una muestra significativa de la población total de industrias y organizaciones dedicadas al desarrollo de software en un momento dado para cada una.

3.4.2 Hipótesis

El correcto uso de todas las fases de la ingeniería de requisitos, ayuda a disminuir las complicaciones en el desarrollo de un proyecto de software.

3.4.3 Unidad de análisis

La unidad de análisis para el presente trabajo de fin de carrera serán los procedimientos que cada empresa u organización aplican en la ingeniería de requisitos.

3.4.4 Población

Para el presente trabajo de fin de carrera la población serán las empresas y organizaciones dedicadas al desarrollo de software en la ciudad de Quito.

3.4.5 Muestra

Para determinar la muestra se utilizó la siguiente fórmula.

N= número de la población = 80

E= margen de error =5%

Z= nivel de confianza =1.65 (90%)

P= probabilidad de éxito =15%

Q= probabilidad de fracaso =85%

n= Igual al tamaño de la muestra

$$n = \frac{z^2 * P * Q * N}{(N - 1) * E^2 + Z^2 * P * Q}$$

Justificación de la asignación de valores:

Z: Para este tipo de población se considera el 90% de nivel de confianza.

P: se considera que la probabilidad de obtener resultados positivos está representada por un 15% este valor representa el tamaño óptimo de la muestra.

Q: Probabilidad de Fracaso de la investigación pueda representar en la recopilación un 85% valor que representa el tamaño óptimo de la muestra.

E: Se considera que el total de cuestionarios que serán administrados a los sujetos de análisis, el 5% de ellas pueda tener deficiencias en la información.

$$n = \frac{1.65^2 * 0.85 * 0.15 * 80}{(80 - 1) * 0.05^2 + 1.65^2 * 0.85 * 0.15}$$
$$n = \frac{28.35}{0.55}$$
$$n = 51.3$$

3.5 Elaboración del Instrumento de investigación

Para la recopilación de la información necesaria en la presente propuesta, se estructuró un cuestionario con 8 preguntas; entre abiertas, cerradas y de opción múltiple, el cual se administrará a los sujetos de análisis. El formato se encuentra en el **Anexo A**.

CAPÍTULO 4: ANÁLISIS DE RESULTADOS

Contexto.- Con el fin de lograr los objetivos planteados al inicio del trabajo de fin de carrera, se realizó la investigación de las prácticas de ingeniería de requisitos en empresas de desarrollo de software de la ciudad de Quito. El estudio está basado en una muestra de 51 empresas consideradas grandes, medianas y pequeñas, se empleó una encuesta guiada para la obtención de información, la cual se encuentra en el **Anexo A**. La información fue procesada en un software estadístico, para su análisis e interpretación se generaron tablas y gráficos para mejorar la comprensión de los resultados.

Se presentan los resultados en orden, partiendo de la clasificación de las empresas de desarrollo de software y posteriormente, se presentan los resultados relacionados con el conocimiento y práctica de la ingeniería de requisitos en las empresas de desarrollo de software en la ciudad de Quito.

4.1 ANÁLISIS DE RESULTADOS POR PREGUNTA

4.1.1 Pregunta N° 1

Tiempo de vida de la empresa o departamento de desarrollo de software. (Años)

Objetivo: Conocer el tiempo de vida de la empresa o departamento de desarrollo de software, como uno de los criterios para contrastar con la madurez de las prácticas de ingeniería de requisitos.

Tabla 3.

Tiempo de vida la empresa o departamento de desarrollo de software

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	1-4 años	10	19.6	19.6	19.6
	5-15 años	28	54.9	54.9	74.5
	más de 15 años	13	25.5	25.5	100.0
Total		51	100.0	100.0	

Análisis: Según lo respondido por las empresas encuestadas se obtuvo que el 19.6% tiene un tiempo de vida de 1-4 años, 54.9% tiene de 5 -15 años y el 25.5% tiene más de 15 años.

4.1.2 Pregunta N° 2

Cantidad de personas en el área de desarrollo de Software.

Objetivo: Conocer el número de personas que desempeñan funciones técnicas en el departamento de desarrollo de Software.

Tabla 4

Cantidad de personal técnico en el área de desarrollo de software

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	1-5 personas	17	33.3	33.3	33.3
	6-20 personas	21	41.2	41.2	74.5
	más de 20 personas	13	25.5	25.5	100.0
	Total	51	100.0	100.0	

Análisis: Según lo respondido por las empresas encuestadas se obtuvo que:

- El 33.3% son empresas que cuentan de 1 a 5 personas desempeñando funciones técnicas el área de desarrollo de software por lo que éstas empresas serán consideradas pequeñas.
- El 41.2% son empresas que cuentan de 6 a 20 personas desempeñando funciones técnicas el área de desarrollo de software por lo que éstas empresas serán consideradas medianas.
- El 25.5% son empresas que cuentan con más de 20 personas desempeñando funciones técnicas el área de desarrollo de software por lo que éstas empresas serán consideradas grandes.

4.1.3 Pregunta N° 3

Número de proyectos de software desarrollados en los 2 últimos años, de acuerdo a su tamaño.

Objetivo: Conocer el total de proyectos realizados y clasificarlos según su tiempo de desarrollo.

Tabla 5.

Proyectos realizados según el tiempo de desarrollo

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	menor a 1 mes de desarrollo	398	68.98	68.98	68.98
	1 - 6 meses de desarrollo	83	14.38	14.38	83.36
	más 6 meses de desarrollo	96	16.64	16.64	100.00
Total		577	100	100	

Análisis: Según lo respondido por las empresas encuestadas se obtuvo que el 68.98% han desarrollado proyectos cuya duración fue menor a un mes, según lo que expresaron estos proyectos se refieren a mantenimiento de software, el 14.38% a desarrollado proyectos cuya duración fue de 1 - 6 meses y el 16.64% desarrolló proyectos cuya duración fue mayor a 6 meses.

4.1.4 Pregunta N° 4

¿La empresa o departamento cuenta con personal especializado en ingeniería de requisitos?

Objetivo: Conocer si las empresas o departamentos cuentan con personal especializado en ingeniería de requisitos, y el número de personal especializado.

Tabla 6

Número de personal especializado en ingeniería de requisitos.

Tamaño de la empresa o departamento, según número de empleados.		Número de personal especializado en ingeniería de requisitos.							Total
		0	1	2	3	4	5	8	0
más 20 empleados	Especializado Ing. Requisitos	NO	9						9
		SI				1	1	2	4
6 - 20 empleados	Especializado Ing. Requisitos	NO	13						13
		SI		2	2	2	1	1	8
1 - 5 empleados	Especializado Ing. Requisitos	NO	14						14
		SI		3					3
Total	Especializado Ing. Requisitos	NO	36						36 70.6%
		SI		5	2	2	2	2	15 29.4%
TOTAL			36	5	2	2	2	2	51

Análisis: Según lo respondido por las empresas encuestadas se obtuvo que el 70.6% no tiene personal especializado en ingenierías de requisitos lo cual indica que ésta es una falencia en el proceso de ingeniería de requisitos debido a su alto porcentaje, el 29.4% cuenta con personal especializado.

4.1.5 Pregunta N° 5

¿La empresa o departamento cuenta con personal especializado en ingeniería de software?

Objetivo: Conocer si las empresas o departamentos cuentan con personal especializado en ingeniería de software y el número de personas especializadas.

Tabla 7

Número de personal especializado en ingeniería de software.

Tamaño de la empresa o departamento, según número de empleados			Número de personal especializado en ingeniería de software													Total		
			0	1	2	3	4	5	6	8	10	14	15	23	26		30	45
más 20 empleados	Esp. Ing. Software	N O	1															1
		SI			1	1		1	1	2	1		1	1	1	1	1	12
6 - 20 empleados	Esp. Ing. Software	N O	3															3
		SI		1	6	2	1	3	1	1	1	1	1					18
1 - 5 empleados	Esp. Ing. Software	N O	4															4
		SI		8	2		1	2										13
Total	Esp. Ing. Software	N O	8															8 15.7%
		SI		9	9	3	2	6	2	3	2	1	2	1	1	1	1	43 84.3%
TOTAL			9	9	9	3	2	6	2	3	2	1	2	1	1	1	1	51

Análisis: Según lo respondido por las empresas encuestadas se obtuvo que el 84.3% cuenta con personal especializado en ingeniería de software, y el 15.7% no cuenta con personal especializado.

4.1.6 Pregunta N° 6

¿Tiene algún tipo de certificación del proceso de desarrollo de software?

Objetivo: Conocer si las empresas o departamentos cuentan con algún proceso de certificación al momento de realizar el desarrollo de un software.

Tabla 8**Número de empresas con certificación CMMI**

Tamaño de la empresa o departamento, según número de empleados	CMMI		Total
	NO	SI	
más de 20 empleados	12	1	13
6 - 20 empleados	21	0	21
1 - 5 empleados	17	0	17
Total	50 98.03%	1 1.97%	51 100%

Análisis CMMI: Según lo respondido por las empresas encuestadas se obtuvo que el 98.03% no cuenta con esta certificación, el 1.97% tiene certificación CMMI, al analizar las características de la empresa se encontró que esta es una transnacional catalogada como grande ya que cuenta con más de 20 empleados.

Tabla 9.**Número de empresas con certificación ISO**

Tamaño de la empresa o departamento, según número de empleados	ISO		Total
	NO	SI	
más de 20 empleados	12	1	13
6 - 20 empleados	17	4	21
1 - 5 empleados	17	0	17
Total	46 90.2%	5 9.80%	51 100%

Análisis ISO: Según lo respondido por las empresas encuestadas se obtuvo que el 90.2% no cuenta con esta certificación, el 9.8% tiene certificación ISO, este porcentaje corresponde a una empresa, la cual cuenta con más de 20 empleados (grandes) y 4 empresas cuentan con 6 - 20 empleados (medianas) lo que indica que las empresas consideradas medianas optaron por conseguir una certificación genérica y no una específica como lo es CMMI.

4.1.7 Pregunta N° 7

Describe el proceso de ingeniería de requisitos que sigue su empresa o departamento.

Objetivo: Conocer cada fase del proceso de ingeniería de requisitos que utilizan en la empresa o departamento de desarrollo de software.

De las fases mencionadas en cada proceso de ingeniería de requisitos se obtuvo los siguientes resultados.

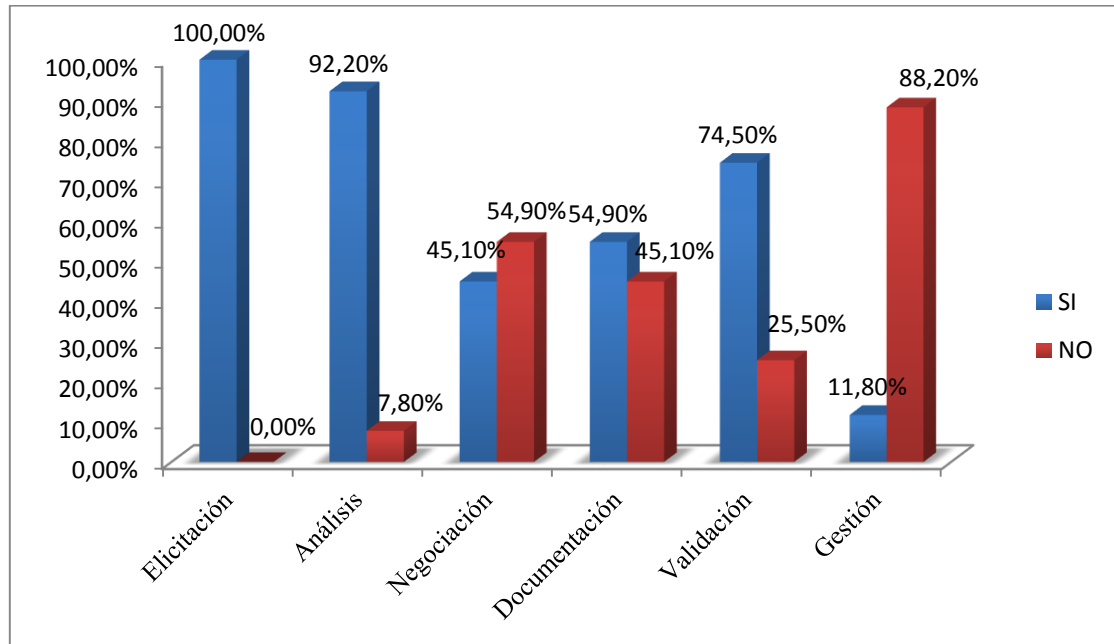


Figura 19. Porcentaje de utilización de cada fase del proceso de ingeniería de requisitos, descritas por las empresas.

Análisis: Como se puede observar en la **Figura 19**, de las empresas encuestadas el 100% realiza la fase de elicitación. El 92.2% realiza el análisis y el 7.8% no lo realiza. El 45.10% realiza la fase de negociación y el 54.90% no realiza la misma. El 54.9% realiza la fase de documentación y el 45.1% no realiza esta fase. El 74.5% realiza la fase de validación y el 25.5% no realiza esta fase. El 11.8% realiza la fase de gestión y el 88.2% no la realiza. Del análisis de estos resultados se puede concluir que pocas empresas cumplen con todas las fases del proceso de ingeniería de requisitos y la

mayoría ejecutan alguna fase, lo cual no permite garantizar una calidad en los requisitos obtenidos.

Llama la atención el número de pocas empresas que no realiza la fase de gestión de los requisitos, debido a que esto afecta a los futuros cambios y el impacto que estos pueden tener.

4.1.8 Pregunta N° 8

Señale las técnicas y/o estándares que aplica en cada fase del proceso de ingeniería de requisitos.

Objetivo: Conocer que técnicas utilizan en cada fase del proceso de ingeniería de requisitos

En relación a las fases de ingeniería de requisitos, se preguntó si realizan o no las mismas; las respuestas obtenidas fueron.

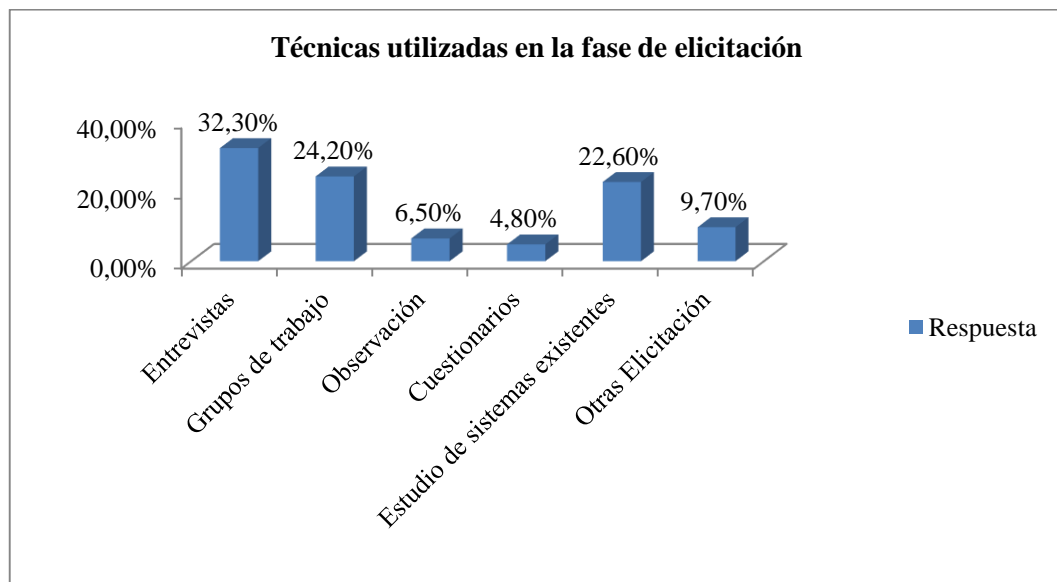


Figura 20. Porcentaje de técnicas utilizadas para elicitación de requisitos

Análisis: Según lo respondido por las empresas encuestadas el 100% realiza la fase de elicitación, la técnica más utilizada es la entrevista con el 32.3%, seguida por los grupos de trabajo con el 24.2%, el estudio de sistemas existentes con el 22.6%, otras técnicas de elicitación 9.7%, la observación con el 6.5% y por último los cuestionarios con el 4.8%. Se puede apreciar que las tres técnicas más utilizadas por las empresas para

la elicitation de requisitos son las entrevistas, seguidas por grupos de trabajo y estudio de sistemas existentes.

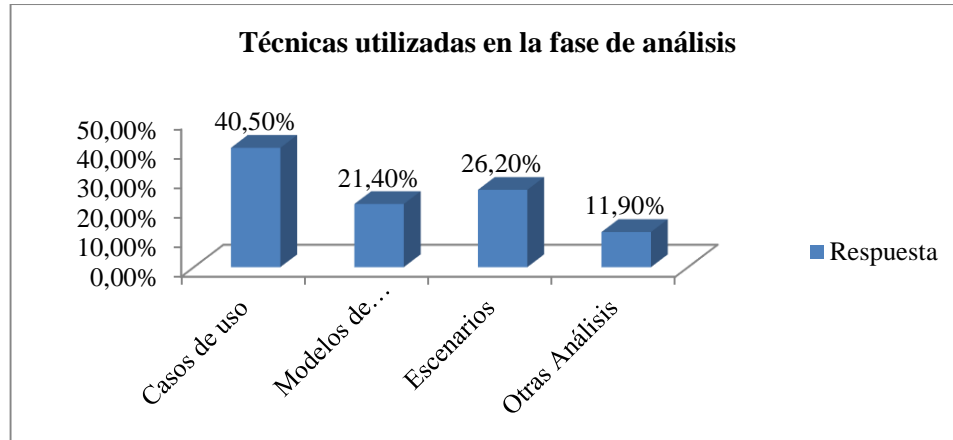


Figura 21. Porcentaje de técnicas utilizadas para el análisis de requisitos.

Análisis: Según lo respondido por las empresas encuestadas el 92.16% realiza la fase de análisis, la técnica más utilizada es casos de uso con el 40.5%, seguida por escenarios con el 26.2%, el modelo de clases, responsabilidades y colaboraciones con el 21.4%, y por último otras técnicas con el 11.9%. Donde se puede apreciar que las técnicas más utilizadas son los casos de uso, escenarios y modelos de clases.

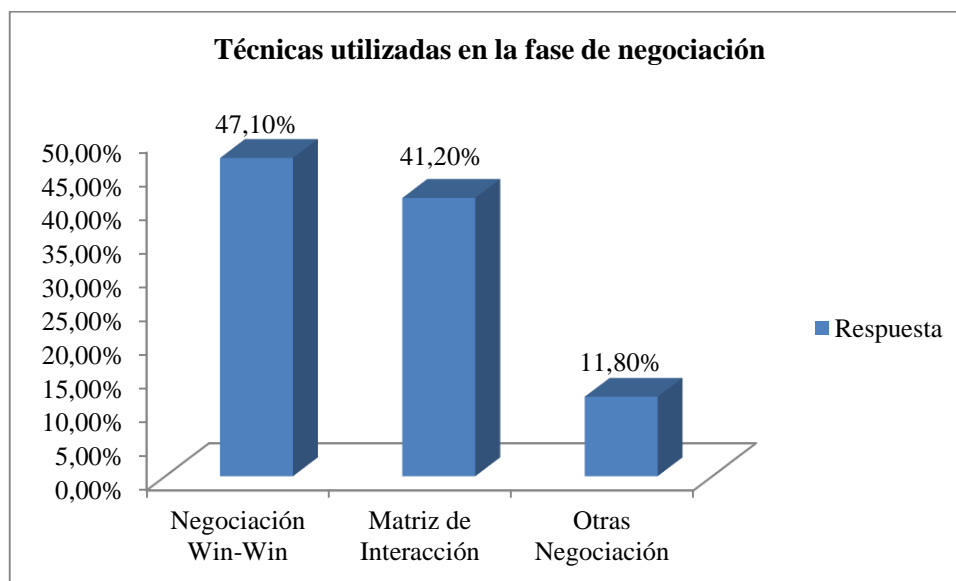


Figura 22. Porcentaje de técnicas utilizadas para la negociación de requisitos.

Análisis: Según lo respondido por las empresas encuestadas el 68.63% realiza la fase de negociación, la técnica más utilizada es la negociación Win-Win con el 47.1%, seguida por la matriz de interacción con el 41.2%, y por último otras técnicas de negociación con el 11.8%. Se puede apreciar que las técnicas más y utilizadas son la negociación win-win y la matriz de interacción.

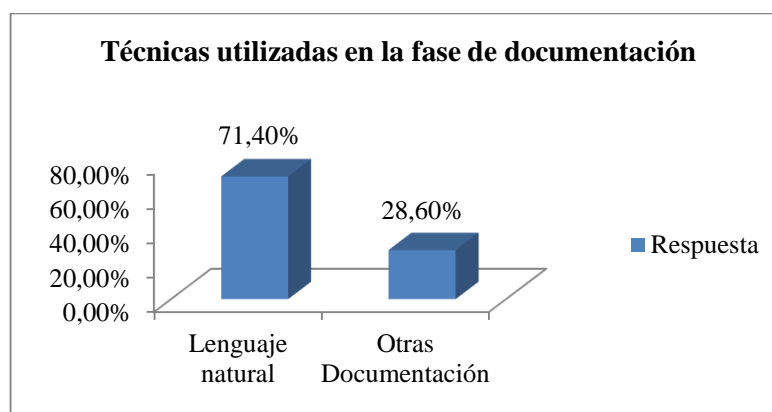


Figura 23. Porcentaje de técnicas utilizadas para documentar requisitos.

Análisis: Según lo respondido por las empresas encuestadas el 88.2% realiza la fase de negociación, la técnica más utilizada es el lenguaje natural con el 71.4%, y otras técnicas de documentación con el 28.6%. Se puede apreciar que la técnica más utilizada por las empresas es el lenguaje natural.

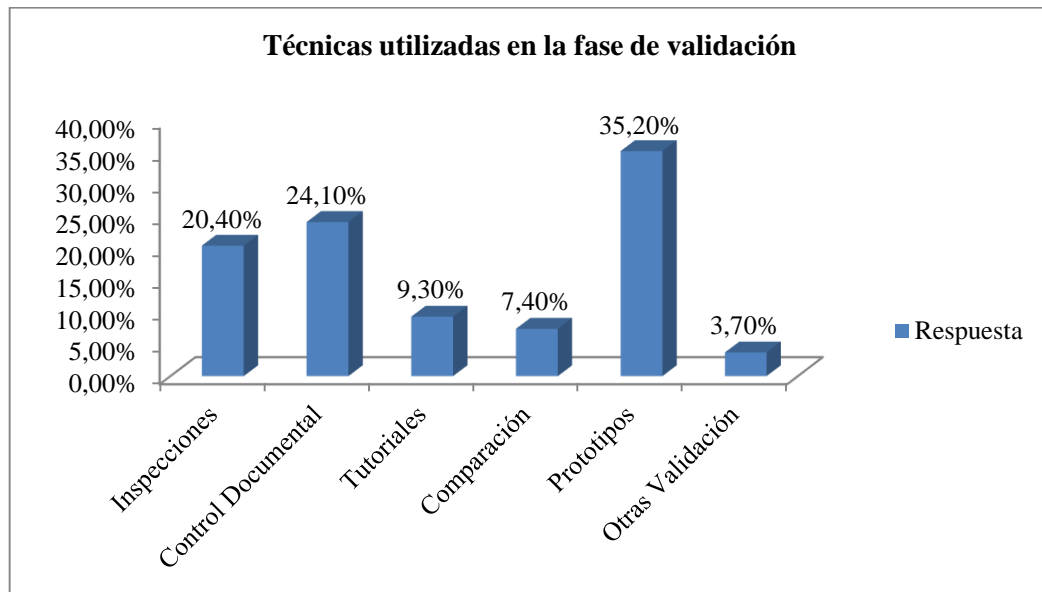


Figura 24. Porcentaje de técnicas utilizadas para validar requisitos en las empresas

Análisis: Según lo respondido por las empresas encuestadas el 98,04% realiza la fase de validación, la técnica más utilizada son los prototipos con el 35,2%, seguida por control documental con el 24,1%, las inspecciones con el 20,4%, tutoriales con el 9,3% y por último otras técnicas de validación con el 3,7%. Se puede apreciar que las tres técnicas más utilizadas por las empresas son prototipos, seguido por el control documental

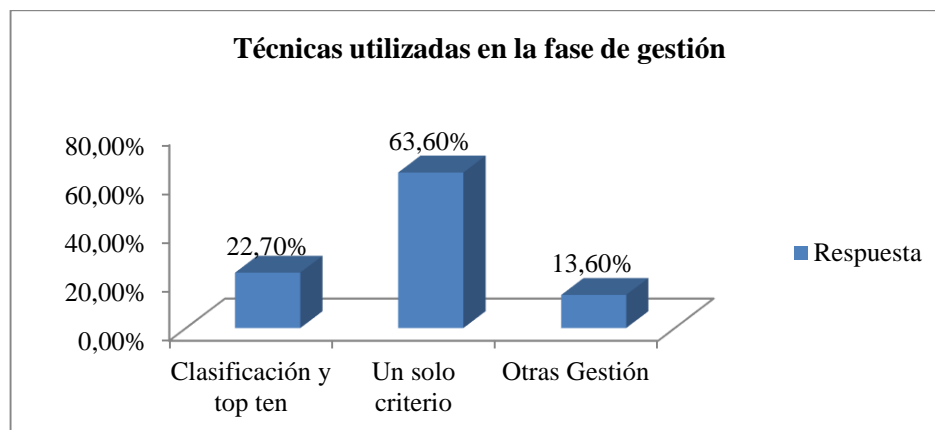


Figura 25. Porcentaje de técnicas utilizadas para gestionar requisitos.

Análisis: Según lo respondido por las empresas encuestadas el 72.55% realiza la fase de gestión, la técnica más utilizada es un solo criterio con el 63.6%, seguida por la clasificación top ten con el 22.7%, y por último otras técnicas de gestión con el 13.6%. Se puede apreciar que la técnica más utilizada para realizar la fase de gestión de requisitos es un solo criterio.

Análisis entre las preguntas 7 y 8

Mediante los siguientes gráficos se pueden observar las distintas diferencias entre lo respondido en la pregunta 7 que era una pregunta abierta y la pregunta 8 que era una pregunta cerrada.

Pregunta 7

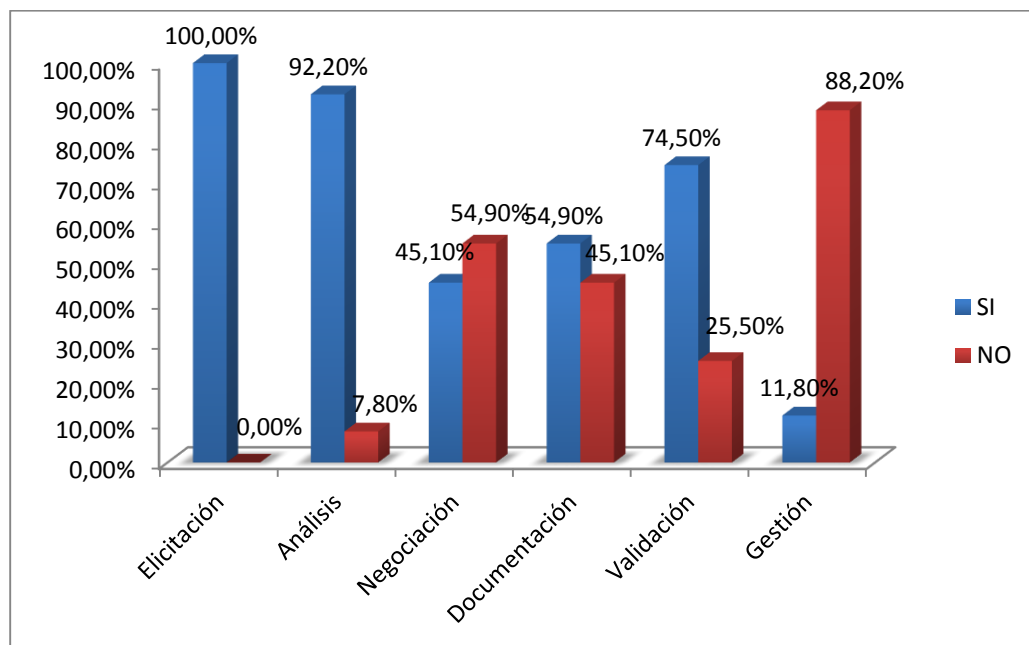


Figura 26. Porcentaje de utilización de cada fase del proceso de ingeniería de requisitos, descritas por las empresas en la pregunta 7

Pregunta 8

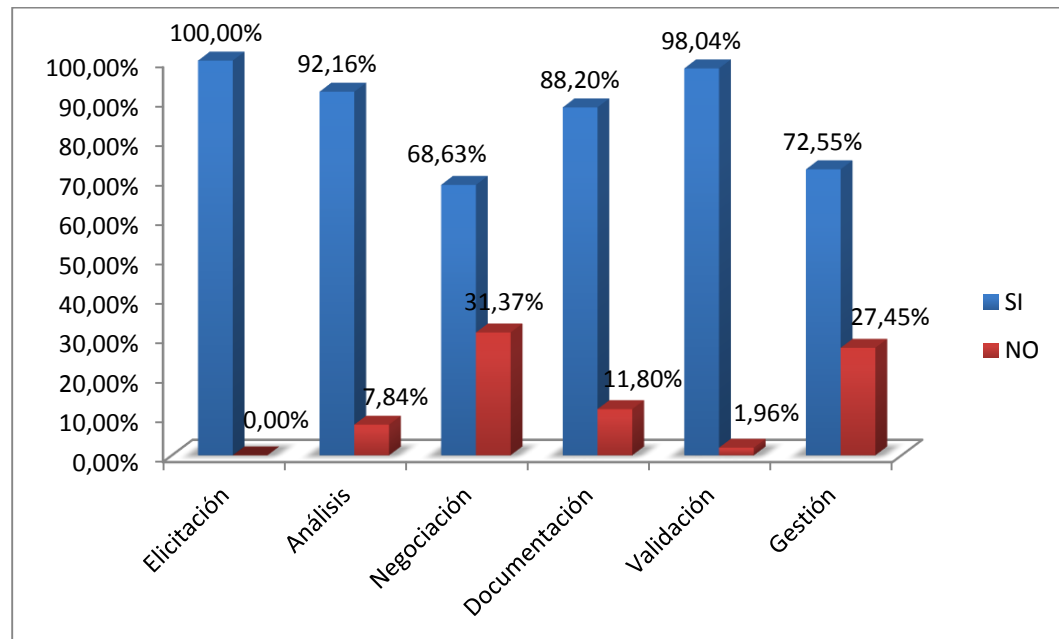


Figura 27. Porcentaje de utilización de cada fase del proceso de ingeniería de requisitos, descritas por las empresas en la pregunta 8.

La pregunta 7 se realizó de manera abierta para que el encuestado manifieste de una manera natural como realiza el proceso de ingeniería de requisitos, mientras que la pregunta 8 se hizo mediante selección múltiple.

Las fases donde existe una gran diferencia son documentación, validación y gestión, lo cual indica que estas son las fases no están claras o no se aplican correctamente al momento de realizar la ingeniería de requisitos.

4.2 ACOTACIONES A LA APLICACIÓN DE LA ENCUESTA GUIADA:

En la encuesta guiada se obtuvieron opiniones, en base a la experiencia en ingeniería de requisitos de las personas representantes de cada una de las empresas, como resultado a continuación se muestra las razones principales por la que las prácticas de ingeniería de requisitos no se realizan correctamente:

- Poca comprensión de los requisitos del sistema (alcance, requisitos funcionales, requisitos no funcionales) por parte del cliente.
- Evolución constante de las necesidades por parte de los usuarios.

- Acceso insuficiente al conocimiento y experiencia del dominio en el que el sistema actuará.
- Se tiene una percepción de que la documentación en muchos casos es extensa y no es lo suficientemente específica para ayudar en el proceso de desarrollo por lo que no genera valor y es dejada de lado.

Para el presente trabajo de fin de carrera se optó por encuestas guiadas para la obtención de información, pero se podrían haber elegido otros instrumentos de investigación para un estudio adicional. A pesar de que es efectivo y se pudo obtener información importante de la experiencia del personal encuestado, el método basado en encuestas guiadas no refleja todos los procesos de ingeniería de requisitos que las empresas utilizan o dejan de hacerlo. Por lo tanto, para otros estudios de seguimiento se pueden elegir otros métodos de investigación como el taller, para explorar a fondo los puntos de vista reales de los sujetos o un estudio patrocinado por el gobierno, asociaciones vinculadas directamente con el desarrollo de software en el que se exija una auditoría a los procesos de ingeniería de requisitos en las empresas.

Aunque este estudio estuvo limitado por su alcance y su escala, realizar encuestas a las empresas de desarrollo de software fue una tarea complicada, debido a la confidencialidad de la información y la poca apertura de muchas de las empresas para la realización de la misma, sin embargo se espera que los resultados sean útiles para los profesionales de la industria y para los investigadores académicos que deseen mejorar las prácticas actuales.

Existen herramientas para realizar un adecuado proceso de ingeniería de requisitos como por ejemplo Rational Requisite Pro, pero debido al desconocimiento o los altos costos por la licencia de este tipo de software, las empresas no utilizan estas herramientas.

CAPÍTULO 5: PROPUESTA METODOLÓGICA PARA EL PROCESO DE INGENIERÍA DE REQUISITOS

5.1 OBJETIVO DE LA METODOLOGÍA

El objetivo de esta metodología es la definición de las fases a realizar, actividades y técnicas a emplear en el proceso de ingeniería de requisitos para el desarrollo de software. Se basa en los resultados de la investigación realizada en el capítulo IV, analizados para obtener una propuesta metodológica que se adapte a la realidad de la industria de desarrollo de software en la ciudad de Quito, empleando las técnicas más utilizadas en cada fase del proceso.

En esta metodología se plantea una adaptación del modelo propuesto por Dr. Klaus Pohl profesor de la universidad de Duisburg-Essen Alemania, dicho modelo fue propuesto en el año 2010, el mismo que es estructurado para una fácil comprensión de las técnicas de ingeniería de requisitos, basado en el conocimiento y experiencia del autor en proyectos reales en la industria de software, Klaus Pohl ha realizado investigaciones sobre ingeniería de requisitos desde la década de 1990, por tales motivos se escogió como el modelo a ser adaptado a la industria ecuatoriana. La metodología se adapta al mercado ecuatoriano donde su principal interés es la rapidez con la que se pueda obtener los requisitos, por tal motivo en la presente existen tareas iterativas que ayudan a un mejor análisis, gestión y rapidez en el proceso de ingeniería de requisitos.

5.2 FASES ACTIVIDADES Y TÉCNICAS RECOMENDADAS

Las fases con sus respectivas actividades recomendadas para obtener los requisitos del software son:

Fase: Elicitación

Actividad 1: Identificar los proveedores de requisitos.

Actividad 2: Planificar y efectuar las actividades de elicitación.

Fase: Análisis

Actividad 1: Identificar los procesos del sistema.

Actividad 2: Preparación de requisitos finales.

Fase: Especificación

Actividad 1: Descripción de requisitos.

Actividad 2: Almacenar los requisitos en un repositorio central.

Fase: Gestión

Actividad 1: Priorizar requisitos.

Actividad 2: Administrar los cambios en los requisitos.

El orden recomendado de realización para estas fases es: fase 1 a fase 4, es recomendable que sean de varias iteraciones, todas las fases y actividades son importantes y se deben realizar, no así las técnicas ya que las mismas dependerán de la magnitud de los proyectos a realizar.

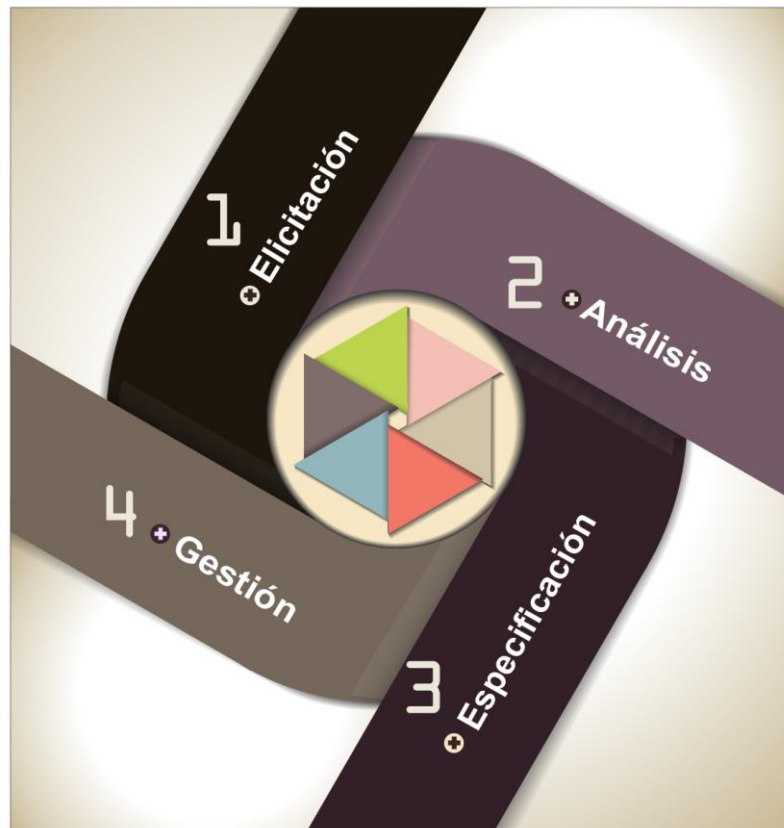


Figura 28. Diagrama del proceso de ingeniería de requisitos propuesta

A continuación se describirán las actividades y las técnicas recomendadas para cada fase del proceso.

5.2.1 Elicitación

5.2.1.1 Actividad 1: Identificar los proveedores de requisitos

Definición de proveedores de requisitos.

Se entiende como proveedores de requisitos a las personas o las fuentes de información que cuenta con las siguientes características:

- Personas que describan el proceso/sistema actual y su problemática.
- Personas con acceso a diagramas de los procesos.
- Estudios de mercado previo que incluyan (Proyectos nuevos):
 - Análisis del consumidor.
 - Análisis de la competencia.
 - Estrategia.

Objetivo de la actividad.

El objetivo es identificar las personas o las fuentes de información que cumplan con las características de proveedores de requisitos anteriormente mencionadas.

Como indican los métodos tradicionales es preferible iniciar con los líderes funcionales de la organización dueña del software a desarrollarse, ya que ellos tienen el conocimiento global de los procesos y comprenden el negocio, después se puede continuar con los futuros usuarios del sistema (usuarios finales) ya que ellos pueden aportar información más detallada sobre los procesos de la organización y la problemática del software actual en caso de existir. En caso que el software a desarrollarse sea nuevo, se puede comenzar por entrevistar a personas que tengan experiencia en el negocio, conocer los posibles consumidores, revisar documentación especializada en el tema, estudiar otras aplicaciones/software de la competencia. Tomar en cuenta las condiciones del área para competir, condiciones del entorno global y el sector al que se dirigirá la nueva aplicación/software.

Sin embargo en los métodos tradicionales de ingeniería de requisitos se corre el riesgo de no tomar en cuenta otras potenciales fuentes de requisitos y por consecuencia afectar a la calidad de los requisitos. Para evitar esto se recomienda la identificación de fuentes potenciales de requisitos.

5.2.1.2 Actividad 2: Planificar y efectuarlas actividades de elicitación

Definición de actividades de elicitación

Se refiere a reunir el grupo de proveedores de requisitos principalmente con el/los analistas de requisitos, donde se trataran temas sobre el levantamiento de requisitos de stakeholder y de sistemas existentes, los más importantes (prioritarios) para el grupo de proveedores, otra actividad que se debe realizar es el levantamiento de requisitos que se puedan obtener de documentos propios del negocio que se está estudiando.

Objetivos de la actividad

La planificación de actividades de elicitación de requisitos tiene como objetivo obtener las necesidades prioritarias para el grupo de proveedores de requisitos y así obtener los objetivos/funciones prioritarios que van a ser analizados en la obtención de los requisitos, reuniones en las que se obtendrán notas, actas, formularios, grabaciones de audio/video o cualquier otra documentación que se considere necesaria y generen valor para el correcto análisis y estimación de los requisitos. Además en cada reunión se puede resolver posibles conflictos que se hayan detectado en iteraciones previas al proceso, las reuniones deben ser planificadas con anticipación tomando en cuenta las siguientes recomendaciones:

- Notificarles a los participantes con suficiente tiempo para que las reuniones lleguen a ser exitosas y productivas.
- Enviar cada participante una agenda de la reunión y trazar un objetivo a lograr, para que el participante pueda preparar toda la información necesaria y en la misma solo deba tomar decisiones invirtiendo menor cantidad de tiempo en la reunión.
- La frecuencia con que se realizarán las reuniones con los proveedores de requisitos puede ser de entre 2 a 4 semanas.

De esta forma se demuestra al cliente la organización con la que se llevara a cabo el proceso de elicitación y la importancia de su participación en todo el proyecto.

Para la elicitación de requisitos se recomienda utilizar las siguientes técnicas que son las más utilizadas en la industria:

- Entrevistas.
- Grupos de trabajo.
- Estudio de sistemas existentes.
- Perspectiva basada en lectura.

5.2.2 Análisis

5.2.2.1 Actividad 1: Identificar los procesos del sistema

Objetivos de la actividad

Una vez que la información fue recolectada a través de diferentes técnicas de elicitación se deben identificar los procesos de la organización, estos deben ser claros y puntuales. Para realizar esta fase del proceso es fundamental conocer las partes del proceso que necesitan ser mejoradas. Con los procesos identificados se realiza un análisis preciso y se evita que existan muchos cambios en los requisitos.

Para identificar los procesos relevantes se puede utilizar criterios de calidad definidos por las necesidades de cliente.

5.2.2.2 Actividad 2: Preparación de requisitos finales

Objetivos de la actividad

El objetivo es tener visualizaciones, requisitos funcionales, no funcionales etc. El equipo de requisitos decide en base a los objetivos lo que necesita ser especificado y construido a través del estudio de los procesos mediante técnicas que ayuden a obtener requisitos claros y puntuales de las necesidades del cliente.

Las técnicas que se pueden utilizar en esta fase del proceso son:

- Casos de Uso.
- Diagrama de Clases.

- Escenarios.
- Historias de Usuario.

5.2.3 Especificación

5.2.3.1 Actividad 1: Descripción de requisitos

Objetivos de la actividad

Realizar una especificación de todos los requisitos obtenidos y analizados, con el fin de tener un documento donde se especifique en lenguaje natural el objetivo, funciones, restricciones, alcance, etc. que va a tener el requisito y ser un documento de respaldo tanto para el cliente como para el equipo de desarrollo.

Se puede utilizar la norma IEEE 830 que sirve para especificar los requisitos del software, es una norma muy sencilla y fácil de aplicar.

5.2.3.2 Almacenar los requisitos en un repositorio central

Objetivos de la actividad

Los requisitos deben estar al alcance de todo el personal de desarrollo y no solo en manos de la persona de realizó la especificación. Por motivos de disponibilidad para que cualquier persona que desee conocer la información del requisito lo pueda buscar y no tenga la necesidad de acudir a otra persona, se deben colocar todos los documentos en un repositorio central.

Por motivos de seguridad se debe dar acceso de lectura a estos documentos para que no cambien la información, solo las personas que tengan autorización podrán cambiar, eliminar o agregar nuevos componentes.

5.2.4 Gestión

5.2.4.1 Actividad 2: Priorizar requisitos

Objetivos de la actividad

Priorizar los requisitos según lo que el cliente considere importante. De esta forma se realizan los trabajos más importantes desde el inicio del desarrollo y en caso de existir cambios se pueden corregir de manera oportuna.

Las técnicas de priorizar más utilizadas en las empresas son:

- Priorización mediante criterios.
- Priorización mediante una valorización (Clasificación Top -Ten).

5.2.4.2 Actividad 1: Administrar los cambios en los requisitos

Objetivos de la actividad

Los objetivos principales de esta actividad son:

- Verificar que el cambio solicitado haya sido autorizado.
- Evaluar el impacto del cambio del requisito con el grupo de proveedores de requisitos.
- Mantener un historial de cambios en los requisitos.
- Identificar la relación de requisitos con otros elementos y procesos del sistema
- Realizar un seguimiento al estado de los requisitos.

Esta práctica es importante al momento de realizar el control de cambios en alguna parte del sistema, de esta manera se puede generar respuestas rápidas a los cambios y por lo tanto ahorrar recursos (tiempo y costos).

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Del análisis realizado en el presente trabajo se pudo determinar que solo una empresa de desarrollo de software en la ciudad de Quito cuenta con una certificación CMMI, la cual obtuvo antes de ingresar al país. Lo cual indica que hasta el momento ninguna empresa nacional ha obtenido una certificación de este tipo en sus procesos de desarrollo de software. Con respecto a las certificaciones ISO, existe alrededor de un 10% de empresas que cuentan con esta certificación las cuáles son consideradas medianas. Sin embargo este porcentaje de empresas que optaron por conseguir certificaciones en sus procesos es muy bajo, como para obtener software de calidad en el medio local y así aportar en el cambio de la matriz productiva que impulsa el gobierno ecuatoriano.
- Al realizar el análisis de lo descrito por las empresas sobre las fases utilizadas en el proceso de ingeniería de requisitos (Capítulo 4, pregunta 7), se encontró, que existe mayor porcentaje en la fase de validación y menor porcentaje en la fase de documentación, lo cual conlleva a deducir, que no se pone énfasis en la documentación escrita que es un proceso importante que permite preservar una base sobre un proyecto, que puede ayudar a los miembros de la empresa para la toma de decisiones, sobre el mismo o para futuros proyectos, esta es una falencia en los procesos que se presenta el medio local.
- La fase de gestión de requisitos presenta menor porcentaje de uso en el proceso de ingeniería de requisitos, las pocas empresas que realizan esta fase utilizan la técnica denominada “**Un solo criterio**”. Esta técnica permite priorizar de acuerdo a un grado de necesidad del requisito, lo cual puede conllevar a un desbalance entre los requisitos priorizados ya que no se toman en cuenta todas las características, que pueden cambiar el estatus del nivel de prioridad para su desarrollo.
- Según lo manifestado por las empresas, aproximadamente el 70% de proyectos de software desarrollados tienen una duración menor a un mes, indicando además que son tareas de mantenimiento (corrección de errores), con lo descrito

en el presente trabajo de fin de carrera, se puede concluir que este alto porcentaje, es debido a que se realiza una deficiente especificación de requisitos lo que provoca que se haga un reproceso de los desarrollos y que se invierta mayor tiempo en la corrección de errores.

- Según los datos proporcionados por las empresas solamente el 29.4% cuentan con personal especializado en ingeniería de requisitos, al existir poco personal capacitado en el área indica que las empresas dedicadas al desarrollo de software en la ciudad de Quito, no cuentan con una visión clara del proceso de ingeniería de requisitos y tampoco le dan la importancia necesaria.
- Según las personas encuestadas uno de los principales problemas que presentan es que no se cumplen con los cronogramas de desarrollo de software, debido a que en todos los proyectos existen cambios en los requisitos, esto se da porque los clientes con frecuencia no identifican lo que necesitan o solamente tienen una leve idea de sus necesidades y a medida que el software evoluciona también lo hacen los requisitos. Según lo manifestado esto se debe a una deficiente especificación de los requisitos y a la poca colaboración o desconocimiento de las personas encargadas de brindar información por parte del cliente.

6.2 RECOMENDACIONES

- Debido al alto costo de una certificación CMMI el gobierno local debería impulsar a las empresas que se dedican al desarrollo de software para que puedan obtener estas certificaciones para su proceso de desarrollo y de esta forma lograr un software de calidad y contribuir a la implementación de la matriz productiva. Otras de las alternativas en cuánto a certificaciones se podría optar por una certificación ISO lo cual estaría al alcance de las empresas consideradas pequeñas y medianas.
- Todas las empresas deben realizar una adecuada documentación de los requisitos que se van a desarrollar, los cuales deben estar especificados correctamente de tal forma que se pueda llegar a entender el dominio y tomar decisiones más apegadas a la realidad y por lo tanto realizar una validación idónea. Además esto

permitirá contar con un documento de respaldo el cual sirva de apoyo para la trazabilidad en el caso que existan futuros cambios en los requisitos.

- Se debe poner mayor énfasis en la fase de gestión de requisitos, debido a la importancia que esta fase tiene, cuando sucedan cambios en los requisitos, la manera en la que fueron priorizados será primordial. Existen técnicas básicas que utilizan las empresas de desarrollo de software, pero se recomienda utilizar la técnica de priorización Kano, que a pesar de utilizar más tiempo, los requisitos quedaran priorizados según el grado de satisfacción del cliente y esto va a ser de gran ayuda para la implementación de una solución de acuerdo a la demanda del cliente.
- Es importante realizar una adecuada ingeniería de requisitos para que no exista inconvenientes a futuro y el mantenimiento sea mínimo, esto ayudará a reducir costos, obtener una buena relación con los clientes y expandirse en el mercado de software local e internacional, que según el estudio está en crecimiento en los últimos años.
- Las empresas desarrolladoras de software deben invertir más tiempo en el proceso de ingeniería de requisitos y personal especializado, estas personas deben ser proactivas y con muy buena capacidad de análisis para predecir cambios y necesidades futuras, cuya única responsabilidad sea realizar el proceso de ingeniería de requisitos de forma correcta.
- Si bien los requisitos tienden a cambiar o evolucionar, la metodología propuesta en el presente trabajo de fin de carrera tiene como objetivo la iteración de sus fases, de manera que se puede detectar cambios o evolución de requisitos en etapas tempranas del proceso desarrollo, además se incluye la interacción con el cliente, se debe poner principal énfasis en ayudarlo a conocer que requisitos para el producto final son su responsabilidad, esto facilita la comunicación con el cliente, los ingenieros de requisitos e ingenieros de software, de esta forma se pueden controlar los cambios de una manera más efectiva.

BIBLIOGRAFÍA

- Beyer, H., & Holtzblatt, K. (1998). *Contextual design, Defining Customer, Centered System*. San Francisco: Morgan Kaufmann.
- Boehm. (2007). *Software Engineering*. New Jersey: Wiley-IEEE Computer Society Pr.
- Boehm, B., & Ross, R. (1989). *Theory-W Software Project Management-Principles and Examples*. New Jersey: IEEE Transactions on Software Engineering.
- CIU, S. I. (2012, 08 01). Clasificación Internacional de Industria Uniforme. Quito, Pichincha, Ecuador.
- CMMI® para Desarrollo, S. E. (2010). *CMMI para Desarrollo*. Massachusetts: Universitaria Ramon Areces.
- Creswell. (2003). *Research Design: qualitative, quantitative, and mixed methods approaches*. Londres: Sage.
- Gilb, T., & Graham, D. (1993). *Software Inspection*. Addison - Wesley.
- Gotel, O., & Finkelstein, A. (1994). *An Analysis of the Requirements Traceability Problem*. Los Alamitos: IEEE Society Press.
- Grajales, T. (2001, enero 1). <http://tgrajales.net/>. Retrieved from <http://tgrajales.net/investipos.pdf>
- IEEE Std 610. (1990). *Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- IEEE Std 830. (1998). *Software requirements specification*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- Infomed, P. d. (1 de febrero de 2009). *Infomed*. Obtenido de <http://files.sld.cu/iss/2009/02/curso-metodologia.pdf>
- International Organization for Standardization. (2001). *ISO/IEC 9126*.
- Juan, m. (2000). <http://juanherrera.files.wordpress.com>. Retrieved from <http://juanherrera.files.wordpress.com/2008/11/investigacion-cuantitativa.pdf>
- Kano, N. (1984). *Attractive Quality and Must-Be Quality*. Journal of the Japanese Society for Quality Control.

- Kotonya, G., & Sommerville, I. (1997). *Requeriments Engineering - Processes and Techniques*. Chichester: Wiley.
- Kovits, B. (1998). *Practical Software Requirements - A Manual of Content and Style*. Greenwich: Manning.
- Kuniavsky, M. (2003). *Observing the User Experience*. San Francisco: Morgan Kauffmann.
- Laguna, M. A. (2012). *Departamento de Informática Universidad de Valladolid*. Retrieved from <http://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>
- Lauesen, S. (2002). *Software Requeriments - Styles and Techniques*. London: Addison-Wesley.
- Leffingwell, D., & Widring, D. (2000). *Managing Software Requirements - A Unified Approach*. Addison-Wesley.
- Loucopoulos, P., & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.
- MIPRO, M. d. (31 de 10 de 2013). <http://www.mipro.gob.ec/>. Recuperado el 06 de 06 de 2012, de <http://www.mipro.gob.ec/>: <http://www.youtube.com/watch?v=gh8TuyAm1k0>
- Moore, C. (2003). *The Mediation Process - Practical Strategies for Resolving Conflicts*. San Francisco: Jossey-Bass.
- Oppenheim, A. (2000). *Questionnaire Design, Interviewing and Attitude Measurement*. Leicester University Press.
- Ortas. (2001). *Aproximacion a la Ingenieria de Requerimientos*. Uruguay: Universidad ORT.
- Padrón, J. (1999). <http://padron.entretemas.com/>. Retrieved from <http://padron.entretemas.com/InvAplicada/index.htm>
- Phol, K. (1996). *Process - Centred Requeriments Engineering*. Somerset: Wiley.
- Phol, K. (2010). *Requirements Engineering Fundamentals, Principles, and Techniques*. Berlin: Springer.
- Pressman, R. (2006). *Ingeniería del Software: Un enfoque práctico*. Mexico DF: McGraw Hil.

Sommerville. (2005). *Ingeniera de Software*.

Sommerville. (2005). *Ingeniera de Software*. Madrid: Pearson Educación.

Standardization, I. O. (2007). *ISO/IEC 25030*. Ginebra.

Superintendencia de Compañías. (2013, 09 01). <http://www.supercias.gob.ec/>. Retrieved from [http://www.supercias.gob.ec/:
http://www.supercias.gob.ec/portalinformacion/index.php?archive=portaldeinformacion/consultadirectorioparametro.zul](http://www.supercias.gob.ec/:http://www.supercias.gob.ec/portalinformacion/index.php?archive=portaldeinformacion/consultadirectorioparametro.zul)

T., J. (2007). *Estimating Software Costs*. New York: McGraw-Hill.

Zowghi, D., & Coulin, C. (2005). *Requirements elicitation - A Survey of Techniques*. New York: Springer.

GLOSARIO

C

CMMI

CMMI.(Modelo Integrado de Capacidad de Madurez).- consta de buenas prácticas que tratan las actividades de desarrollo aplicadas a productos y servicios. · 8, 10, 11, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 89, 90, 110, 112, 116

E

Elicitar

Elicitar.- es el acto de extraer información de una persona o un grupo de personas. · 32, 40, 42

ERS

ERS(Especificación de Requisitos Software).- permite al cliente definir todos los requisitos que desea y a los desarrolladores tener una base fija en la cual trabajar. · 74, 75, 76, 78

M

método KJ

Método KJ.- Método utilizado para elicitar requisitos y fuentes de requisitos. · 35, 37, 43

R

Rational Requisite Pro

Rational Requisite Pro.- es una familia de software de IBM para el despliegue, diseño, construcción, pruebas y administración de proyectos en el proceso de desarrollo de software. · 100

REQM

REQM(Requirements Management).- gestionar los requisitos de los productos y de los componentes e identificar inconsistencias entre requisito, planes y productos del proyecto. · 63, 64, 65

S

SQuaRE

SQuaRE(Software product Quality Requirements and Evaluation) .- tiene el objetivo de crear un marco para la evaluación de la calidad del producto de software. · 10, 68, 69

SSL

SSL(secure sockets layer).- protocolos criptográficos que proporcionan comunicaciones seguras por una red. · 22

Stakeholders

Stakeholders.- Quienes pueden afectar o son afectados por las actividades de una empresa. · 29, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 46, 47, 48, 51, 52, 53, 54, 57, 58, 71

T

Tester

Tester.- quien investiga un producto de software con el objetivo de obtener información acerca de su calidad y del valor que representa para quienes lo utilizan. · 41

W

Win-Win

Win-Win.- significa que las partes negociadoras han llegado a un acuerdo después de tomar

plenamente en cuenta los intereses de los
demás, de manera que el acuerdo no puede

ser mejorado aún más por cualquier otro
acuerdo. · 47, 94

HOJA DE LEGALIZACIÓN DE FIRMAS**ELABORADO POR**

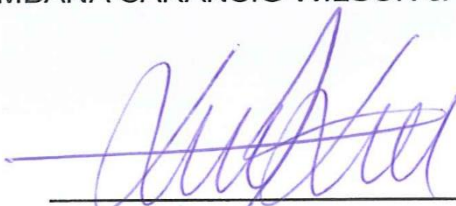
SIMBAÑA QUINSASAMIN JUAN GABRIEL



Sr. Simbaña Quinsasamin Juan Gabriel

ELABORADO POR

SIMBAÑA SARANSIG WILSON JAVIER



Sr. Simbaña Saransig Wilson Javier

DIRECTOR DE LA CARRERA

ING. MAURICIO CAMPAÑA



Sr. Ing. Mauricio Campaña



Sangolquí, mayo del 2015: