



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**“PROPUESTA DE MÉTRICAS PARA LOS REQUISITOS SOFTWARE
EN LA ETAPA DE ELICITACIÓN”**

TESIS

**PARA OBTENER EL GRADO DE
MAESTRO EN TECNOLOGÍAS DE CÓMPUTO APLICADO**

PRESENTA

ING. MYRIAM KARENINA REYES SÁNCHEZ

DIRECTOR DE TESIS

DRA. CARLA LENINCA PACHECO AGÜERO

HUAJUAPAN DE LEÓN, OAX.; ENERO DEL 2015

**Tesis presentada el 9 de enero del 2015,
ante los siguientes sinodales:**

Dr. Carlos Alberto Fernández y Fernández.

Dr. Ivan Antonio García Pacheco.

Dr. José Figueroa Martínez.

Dr. Moisés Homero Sánchez López.

Directora de Tesis:

Dra. Carla Leninca Pacheco Agüero.

Dedicatoria

A la persona más hermosa y maravillosa: Annita, por ser mi más grande motivación.

A mi compañero de vida: Enrique, por tu amor, apoyo y comprensión.

A mis padres y hermanos, por que sin ellos nada de esto hubiera sido posible.

Agradecimientos

A mi directora de tesis, Dra. Carla Leninca Pacheco Agüero, por su paciencia, sus conocimientos, el tiempo dedicado a este proyecto y las facilidades otorgadas para la elaboración de este trabajo de tesis.

A CONACYT por el apoyo económico, así como las facilidades otorgadas para cursar esta maestría y realizar este trabajo de tesis.

A mi esposo Hugo Enrique Martínez Cortés, por haberme apoyado en todo lo que te fue posible, comprenderme, entenderme y amarme aún en mis peores momentos.

A mi madre, que siempre está al pendiente de que logre mis metas, por ser mi gran ejemplo y por todo el apoyo brindado a lo largo de toda mi vida.

A mi padre, quien me motivó a estudiar en esta Universidad, por ser también mi ejemplo y apoyarme para poder lograr todos mis objetivos.

A mis hermanos, por todo el apoyo incondicional que me han brindado y que con su buen humor han alegrado mi vida.

A mi compañero y amigo Carlos Alejandro Pérez Cruz, que junto con la Dra. Lluvia Carolina Morales Reynaga me ayudaron a conseguir el material bibliográfico necesario para la elaboración de este trabajo.

Índice

Dedicatoria.....	v
Agradecimientos.....	vii
Índice.....	ix
Lista de tablas.....	xiii
Lista de figuras.....	xv
Resumen.....	xvii
1. Introducción.....	1
1.1. Contexto del problema.....	1
1.2. Importancia del problema.....	4
1.3. Necesidad de resolución.....	7
1.4. Hipótesis.....	8
1.5. Objetivo general.....	8
1.6. Objetivos específicos.....	8
1.7. Delimitaciones y limitaciones del trabajo.....	8
1.8. Aproximación a la solución.....	9
1.9. Estructura de la tesis.....	10
2. Estado del Arte.....	11
2.1. Marco teórico.....	11
2.1.1. Procesos involucrados en la elicitación de requisitos.....	11
2.1.1.1. Propuesta del SEI.....	11
2.1.1.2. Modelo del SWEBOK.....	12
2.1.1.3. Modelo de Kotonya.....	13
2.1.2. Problemas en la elicitación de requisitos.....	14
2.1.2.1. Según el SEI.....	14
2.1.2.2. Según el SWEBOK.....	15
2.1.2.3. Clasificación de Raghavan.....	16
2.1.2.4. Clasificación de Sommerville.....	18
2.1.3. Técnicas utilizadas en la elicitación de requisitos.....	18
2.1.3.1. Categorización del SEI.....	19

2.1.3.1.1. Entrevistas.....	19
2.1.3.1.2. Lluvia de ideas.....	20
2.1.3.1.3. Diseño de aplicaciones conjuntas (JAD, por sus siglas en inglés).....	21
2.1.3.1.4. Desarrollo de funciones de calidad (QFD, por sus siglas en inglés).....	21
2.1.3.1.5. Escenarios.....	21
2.1.3.1.6. Prototipos.....	22
2.1.3.2. Framework de Nuseibeh-Easterbrook.....	22
2.1.3.2.1. Técnicas tradicionales.....	22
2.1.3.2.2. Técnicas grupales de elicitación.....	22
2.1.3.2.3. Prototipos.....	22
2.1.3.2.4. Técnicas de modelado.....	22
2.1.3.2.5. Técnicas cognitivas.....	23
2.1.3.2.6. Técnicas contextuales.....	23
2.1.3.3. Clasificación del SWEBOK.....	23
2.1.3.3.1. Entrevistas.....	23
2.1.3.3.2. Escenarios.....	23
2.1.3.3.3. Reuniones.....	23
2.1.3.3.4. Observación o etnografía.....	24
2.1.3.4. Clasificación de Hickey-Davis.....	24
2.1.3.4.1. Sesiones Colaborativas.....	24
2.1.3.4.2. Entrevistas.....	24
2.1.3.4.3. Etnografía.....	24
2.1.3.4.4. Lista de tópicos o temas.....	24
2.1.3.4.5. Modelos.....	25
2.1.3.4.6. Cuestionarios.....	25
2.1.3.4.7. Obtención de datos a partir de software existente.....	25
2.1.3.4.8. Categorización de requisitos.....	25
2.1.3.4.9. Conocimiento del conflicto y de la resolución.....	25
2.1.3.4.10. Prototipos.....	26
2.1.3.4.11. Juegos de roles.....	26
2.1.3.4.12. Métodos formales.....	26
2.1.3.4.13. Programación extrema.....	26
2.1.3.5. Otras técnicas.....	26
2.1.3.5.1. Enfoque colaborativo.....	26
2.2. Trabajo relacionado.....	27
3. Solución propuesta.....	29
3.1. Revisión sistemática de literatura.....	29
3.1.1. Identificación de la necesidad de la revisión literaria.....	30
3.1.2. Formulación de preguntas de investigación.....	31

3.1.3. Términos de búsqueda	32
3.1.4. Estrategia de búsqueda	32
3.1.5. Selección de estudios.....	32
3.1.5.1. Establecimiento de los criterios de selección	32
3.1.5.2. Definición del procedimiento de selección	33
3.1.6. Evaluación de la calidad de los estudios seleccionados.	33
3.1.7. Definición de la estrategia de extracción de datos	34
3.1.8. Síntesis de los datos extraídos	34
3.1.8.1. Recuperación de documentos	35
3.2. Resultados.....	36
3.2.1. Visión general de los estudios	36
3.2.1.1. Método de investigación.....	36
3.2.1.2. Año de publicación.....	37
3.2.2. Métricas para los requisitos software en la etapa de elicitación.....	38
3.2.2.1. PI1- técnicas o métodos maduros utilizadas actualmente en la etapa de elicitación de requisitos.....	38
3.2.2.1.1. Técnicas tradicionales.....	38
3.2.2.1.2. Técnicas grupales-colaborativas.....	40
3.2.2.1.3. Prototipos.....	41
3.2.2.1.4. Técnicas de modelado	41
3.2.2.1.5. Técnicas cognitivas.....	42
3.2.2.1.6. Técnicas contextuales.....	44
3.2.2.1.7. Metodologías ágiles.....	44
3.2.2.2. PI2- técnicas que han demostrado mayor efectividad	45
3.2.2.3. PI3- aspectos de las técnicas que influyen en su efectividad	49
3.2.2.4. PI4- Métricas existentes en la etapa de elicitación de requisitos.....	57
3.3. Métricas para la elicitación de requisitos: propuesta.....	66
3.3.1. Paradigma GQM.....	66
3.3.2. Proceso de medición.....	68
3.3.2.1. Definición de procedimientos.....	85
3.3.2.1.1. Definición de medidas	85
3.3.2.1.2. Definir métodos de conteo o recogida-registro y almacenamiento	88
3.3.2.1.3. Diseño de formularios para el registro de medidas	91
3.3.2.1.4. Definir mecanismos de almacenamiento	92
3.3.2.1.5. Definir métodos de análisis	92
3.3.2.1.6. Definir mecanismos de divulgación y realimentación	98
4. Validación y resultados	101
4.1. Contexto del estudio	104
4.2. Método.....	104

4.3. Resultados empíricos	109
5. Conclusiones	111
6. Anexo A.- Acrónimos.....	115
7. Anexo B.- Guía de técnicas de elicitación de requisitos software.....	117
8. Anexo C.- Plantillas.....	129
9. Anexo D.- Requisitos y reporte de resultados	145
10. Bibliografía fundamental	177

Lista de tablas

Tabla 1. Correspondencia de las técnicas de elicitación a un conjunto de propiedades (Christel & Kang, 1992).	20
Tabla 2. Evaluación de calidad.	34
Tabla 3. Puntuación de calidad de los estudios aceptados para las primeras dos preguntas de investigación.	34
Tabla 4. Estudios validados y revisados.	35
Tabla 5. Estudios que muestran los primeros años en los que se aplicaron las técnicas de elicitación.	37
Tabla 6. Lista de técnicas de elicitación tradicionales.	39
Tabla 7. Lista de técnicas de elicitación grupales-colaborativas.	40
Tabla 8. Número de estudios para prototipos.	41
Tabla 9. Lista de técnicas de elicitación de modelado.	42
Tabla 10. Lista de técnicas de elicitación cognitivas.	43
Tabla 11. Número de estudios para técnicas contextuales.	44
Tabla 12. Número de estudios para métodos ágiles.	44
Tabla 13. Efectividad de las técnicas tomando en cuenta los atributos de calidad de los requisitos obtenidos.	48
Tabla 14. Aspectos de las técnicas que las hacer ser más efectivas.	55
Tabla 15. Métricas utilizadas en la etapa de elicitación.	61
Tabla 16. Plantilla para la definición de objetivos de medición (Basili et al., 2007).	67
Tabla 17. Aspectos que cubren las métricas encontradas en la respuesta a la PI4.	69
Tabla 18. Jerarquía de medición (McGarry et al., 2002).	70
Tabla 19. Medidas identificadas del CMMI (Park, 1996).	71
Tabla 20. Identificación de los <i>stakeholders</i> .	72
Tabla 21. Analizar la efectividad de las técnicas de elicitación.	72
Tabla 22. Mejorar el proceso de comunicación y negociación.	72
Tabla 23. Analizar la integración y el refinamiento de la información.	73
Tabla 24. Evaluar la efectividad del proceso de elicitación.	73
Tabla 25. Métricas que han sido propuestas por otros autores para los requisitos software en la etapa de elicitación.	74

Tabla 26. Sub objetivo 1 de nivel 2 para identificar los <i>stakeholders</i>	75
Tabla 27. Sub objetivo 1 de nivel 3 para identificar los <i>stakeholders</i>	76
Tabla 28. Sub objetivo 2 de nivel 3 para identificar los <i>stakeholders</i>	76
Tabla 29. Sub objetivo 1 de nivel 2 para analizar la integración y el refinamiento de la información.....	76
Tabla 30. Sub objetivo 2 de nivel 2 para analizar la integración y el refinamiento de la información.....	76
Tabla 31. Métricas propuestas para el objetivo de identificación de <i>stakeholders</i>	77
Tabla 32. Métricas propuestas para el objetivo de identificación de <i>stakeholders</i> necesarios y relevantes.....	78
Tabla 33. Constructor de medición (McGarry et al., 2002).....	78
Tabla 34. Constructor de medición para identificación de los <i>stakeholders</i>	80
Tabla 35. Constructor de medición para analizar la integración y el refinamiento de la información.....	83
Tabla 36. Medidas primarias.....	86
Tabla 37. Medidas del entorno.....	88
Tabla 38. Medidas de ejecución del proyecto.....	88
Tabla 39. Desarrollo del proceso de elicitación.....	89
Tabla 40. Identificación de <i>stakeholders</i>	90
Tabla 41. Obtención de la lista de deseos de cada <i>stakeholder</i>	90
Tabla 42. Integración, refinamiento y organización de la información que se ha obtenido.....	91
Tabla 43. <i>Postmortem</i>	91
Tabla 44. Resultados del caso de estudio.....	105
Tabla 45. Atributos de calidad de los requisitos.....	109
Tabla 46. Costos y beneficios de aplicar el programa de medición propuesto.....	110

Lista de figuras

Figura 1.1. Tiempo empleado en cada actividad de la Ingeniería de Requisitos (Christel & Kang, 1992).....	4
Figura 1.2. Clasificación de los proyectos para los años 2000-2010 (Standish, 2011).....	5
Figura 3.1. Relación entre las cuatro preguntas de investigación.	31
Figura 3.2. Métodos de investigación en los estudios aceptados.	36
Figura 3.3. Número de estudios por año que muestran los estudios que dan respuesta a las cuatro PI.....	38
Figura 3.4. Número de estudios por técnica de elicitación tradicional y año en el que fueron publicados.....	40
Figura 3.5. Número de estudios por técnica de elicitación grupal-colaborativa y año en el que fueron publicados.	41
Figura 3.6. Número de estudios para la técnica de prototipos y año en el que fueron publicados..	42
Figura 3.7. Número de estudios por técnica de elicitación de modelado y año en el que fueron publicados.....	43
Figura 3.8. Número de estudios por técnica de elicitación cognitiva y año en el que fueron publicados.....	43
Figura 3.9. Número de estudios para la técnica de elicitación etnografía y año en el que fueron publicados.....	44
Figura 3.10. Número de estudios que utilizan métodos ágiles para la elicitación y año en el que fueron publicados.	45
Figura 3.11. Objetivo para identificar los <i>stakeholders</i>	75
Figura 3.12. Objetivo para analizar la integración y el refinamiento de la información.....	75
Figura 3.13. Gráfica que muestra el número de versiones por cada requisito y el número de categorías de stakeholders que poseen este tipo de conocimiento.	94
Figura 3.14. Gráfica que muestra el número de requisitos obtenido de cada <i>stakeholder</i>	95
Figura 3.15. Gráfica que muestra el número de requisitos que fueron propuestos y aceptados por cada <i>stakeholder</i>	96
Figura 3.16. Gráfica que muestra el número versiones por requisito.....	97
Figura 3.17. Gráfica que muestra el porcentaje de requisitos en conflicto.	97
Figura 3.18. Gráfica que muestra las modificaciones hechas por cada versión de lista de deseos y necesidades.	98

Resumen

Debido a que día a día la calidad del software demandada por los usuarios crece, también la complejidad del software se hace cada vez mayor. Es por esto que uno de los objetivos de la Ingeniería de Software es producir software de calidad. Ahora bien, dentro del proceso software un área de vital importancia es la Ingeniería de Requisitos ya que es en esta etapa donde se decide que es lo que se va a desarrollar. Una de las etapas críticas y que más tiempo emplea dentro de la Ingeniería de Requisitos es la elicitación, ya que es aquí en donde se obtienen y descubren las necesidades de los *stakeholders*. Debido a que en esta etapa es fundamental el factor humano se presentan diversos problemas que pueden afectar la calidad de los requisitos obtenidos. A pesar de que existen diversas técnicas para obtener requisitos se siguen teniendo problemas relacionados con la calidad de los mismos como consecuencia de que solo existen algunas métricas que permitan evaluar si se obtuvieron requisitos de calidad, es decir que cumplan con las siguientes características: completos, consistentes, relevantes, viables, trazables, verificables, modificables, y no ambiguos. Es por esto que en este trabajo de tesis se analizarán las técnicas para elicitar requisitos, así como su efectividad, con el objetivo de proponer un conjunto de métricas que ayuden a entender, controlar y mejorar el proceso de elicitación.

1. Introducción

1.1. Contexto del problema

En la década de los 60's se acuñó el término “crisis del software” para etiquetar diversos problemas que se habían identificado en el desarrollo de software: entrega tardía, excesos en el presupuesto, requisitos¹ no satisfechos, software difícil de usar, mantener y mejorar, entre otros (Brooks, 1987; Gibbs, 1994). Con el propósito de darle solución a estos problemas surge la Ingeniería de Software (IS), término utilizado por primera vez en la Conferencia de la Organización del Tratado del Atlántico Norte (NATO, por sus siglas en inglés) celebrada en Alemania en 1968 (Abran, Moore, Bourque & Dupuis, 2004).

A partir de la década de los 70's, empezaron a surgir muchas definiciones para la IS, como las siguientes:

- *“Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software”* (IEEE, 1990).
- *“Es una disciplina que comprende todos los aspectos de la producción del software desde etapas iniciales como la especificación del mismo, hasta el mantenimiento de éste, después de que se ha utilizado”* (Sommerville, 2010).
- *“Una tecnología estratificada donde los estratos son el proceso, los métodos y las herramientas, siendo el enfoque en la calidad² la base que soporta a la IS”* (Pressman, 2009).
- *“Resume las cuestiones que tienen que ver con el desarrollo y mantenimiento del software para que éste se comporte de forma fiable y eficiente, sea asequible de desarrollar y mantener, y satisfaga todos los requisitos que los clientes han definido”* (ACM, 2013).

¹ Un requisito describe los servicios que debe proveer el software, además define de manera detallada las funcionalidades o restricciones de éste (Sommerville & Sawyer, 2006).

² La calidad del software es un concepto difícil de definir ya que existen diferentes perspectivas sobre ésta, de acuerdo al Glosario de Terminología de IS del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por su siglas en inglés) la calidad se define como *“el grado en que un software, componente o proceso cumple con las expectativas del cliente o usuario”* (IEEE, 1990). Además existen algunos estudios hechos en la década de los 70's, por ejemplo el de McCall, Richards y Walters (McCall, Richards & Walters, 1977) o el de Boehm, Brown, Kaspar, Lipow, MacLeod y Merrit (Boehm et al., 1978) en los que se identificaron factores del software que se refieren a la noción de calidad del mismo: corrección, fiabilidad, eficiencia, integridad, facilidad de uso, facilidad de mantenimiento, facilidad de prueba, flexibilidad, portabilidad, reusabilidad y facilidad de interoperación. De acuerdo al estándar ISO 9126 para decir que el software es de calidad, éste debe cumplir ciertos atributos: funcionalidad, confiabilidad, eficiencia, facilidad de uso, facilidad de mantenimiento y portabilidad (ISO, 1991).

- “*Es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlo*” (Boehm, Brown & Lipow 1976).

Tomando como base estas definiciones, se puede decir que la IS es una disciplina que describe un conjunto de actividades para el proceso software de inicio a fin, y cuyo producto final es el software en sí (Sommerville, 2010). De acuerdo con Boehm, Brown y Lipow (Boehm, Brown & Lipow, 1976), Pressman (Pressman, 2009), y Lawrence, Pfleeger y Atlee (Lawrence, Pfleeger & Atlee, 2010) la IS está constituida fundamentalmente por las siguientes actividades:

1. *Análisis y definición de requisitos.* En esta etapa se debe entender cuál es el problema a resolver, obteniendo la información necesaria de los *stakeholders*³ o involucrados para así, definir y especificar tanto la funcionalidad como las restricciones del software.
2. *Diseño.* Determina la forma en cómo se producirá el software para que cumpla con la especificación de los requisitos. En esta etapa se identifican y describen las estructuras fundamentales y sus relaciones, estableciendo una arquitectura completa del software.
3. *Implementación.* Se codifica lo que fue definido en la etapa de diseño, obteniendo un conjunto o unidades de programas y se revisa que cada uno cumpla con su especificación.
4. *Validación y Verificación.* Las unidades se integran y se prueban como un todo para descubrir las faltas, fallos y errores que puedan existir para garantizar que se satisfagan los requisitos iniciales.
5. *Mantenimiento.* Durante esta etapa se corrigen errores no descubiertos en etapas anteriores y se mejora la implementación del software.

Algunos autores como Pressman (Pressman, 2009) sugieren que la IS se encuentra en un estado de enfermedad crónica, debido a que el software evoluciona rápidamente a través del tiempo -sin importar su dominio de aplicación, tamaño o complejidad- por lo que continuamente surgen nuevas necesidades que la IS debe cubrir y desafortunadamente no lo hace.

Durante mucho tiempo, la literatura y la investigación referente a la IS han centrado sus esfuerzos en el desarrollo, procurando crear lenguajes y técnicas de representación, por lo que hasta hace poco relativamente, se prestaba muy poca atención a la tarea de descubrir realmente lo que el software tenía que hacer, aspecto realmente importante ya que el punto de partida del proceso software es el momento en que se define qué es lo que se quiere hacer. El proceso software podrá iniciar, siempre y cuando se haya establecido correctamente cuál será el producto final a entregar.

Ahora bien, cuando se trata de software complejo, la definición de lo que se quiere obtener no es una tarea trivial por lo que es común que muchos productos software no se comporten como es deseable. Es por esto que la primera etapa del desarrollo de software se vuelve crucial, ya que no se puede desarrollar un software si no se comprende el problema a resolver o no se sabe qué es lo que el cliente necesita (Lawrence et al., 2010).

De acuerdo con Brooks y Sommerville, la etapa de requisitos es la parte más difícil del proceso software puesto que es aquí donde debe decidirse y definir lo que se va a desarrollar (Brooks, 1987; Sommerville, 2010). Davis et al. (1993) establecieron que: “*La incapacidad para producir requisitos completos, correctos y no ambiguos se sigue considerando la causa principal de*

³ Los *stakeholders* son las personas que se verán afectadas por el software y quienes tienen un influencia directa o indirecta en los requisitos del mismo (Sommerville & Sawyer, 2000).

la falla del software". De acuerdo con Hofmann y Lehner, muchos de los fracasos de los proyectos software pueden tener su origen en unos requisitos deficientes e incompletos, en la volatilidad de los mismos, y en la falta de participación de los *stakeholders* (Hofmann & Lehner, 2001).

En este ámbito tan importante, se sitúa la Ingeniería de Requisitos (IR) como un área de investigación que procura resolver un punto fundamental en el proceso software: *la definición de lo que se quiere obtener como producto final* (Zowghi & Paryani, 2003).

La IR nace como una rama de la IS, y surge de la necesidad de resolver la difícil tarea de recopilar, analizar y verificar los requisitos (Hull, Jackson & Dick, 2011). La IR ayuda a entender mejor un problema del mundo real haciendo uso de las especificaciones precisas y proporciona un mecanismo para poder trabajar en su solución (Nuseibeh & Easterbrook, 2000; Kotonya & Sommerville, 2000).

De acuerdo con (Christel & Kang, 1992; Kotonya & Sommerville, 2000; Berenbach, Paulish, Kazmeier & Rudorfer 2009) el proceso de la IR es descrito frecuentemente con las siguientes actividades:

- *Elicitación*. Se obtienen los requisitos software. En esta actividad el personal técnico de desarrollo trabaja con los *stakeholders* para determinar el dominio de aplicación, los servicios que el software debe proporcionar, así como la lista de deseos y necesidades del software.
- *Análisis*. Tomando como base la información extraída en la etapa anterior, se hace un análisis para descubrir posibles problemas con los requisitos obtenidos hasta el momento.
- *Especificación*. En esta fase se definen de manera formal y detallada los requisitos en un documento estándar, prototipo o alguna combinación de éstos. La especificación es el producto final que genera la IR.
- *Validación y verificación*. El objetivo de esta etapa es verificar que todos los requisitos especificados sean consistentes, completos, trazables, no ambiguos y validar que describan adecuadamente lo que el software debe realizar. A diferencia de la etapa de análisis en donde se trabaja con requisitos incompletos, aquí se trabaja con la Especificación de Requisitos Software (SRS, por sus siglas en inglés).
- *Gestión*. Los requisitos en el software siempre cambian o evolucionan, por lo que en esta etapa se mantiene el control de los cambios, lo que permite identificar y rastrear los requisitos volátiles del software, aspecto que redundará en una mejor administración de los requisitos.

La mayoría de los modelos existentes en la IR muestran en orden secuencial las actividades que componen este proceso. Pero en realidad, estas actividades no son ejecutadas secuencialmente sino que pueden realizarse en paralelo (véase Figura 1.1). Como puede verse, de todas las actividades que comprende la IR, se puede decir que la etapa crítica es la **elicitación de requisitos** puesto que es fundamental para cualquier proyecto software y al mejorarla se podría perfeccionar la eficiencia del proceso de IR y por ende de la IS.

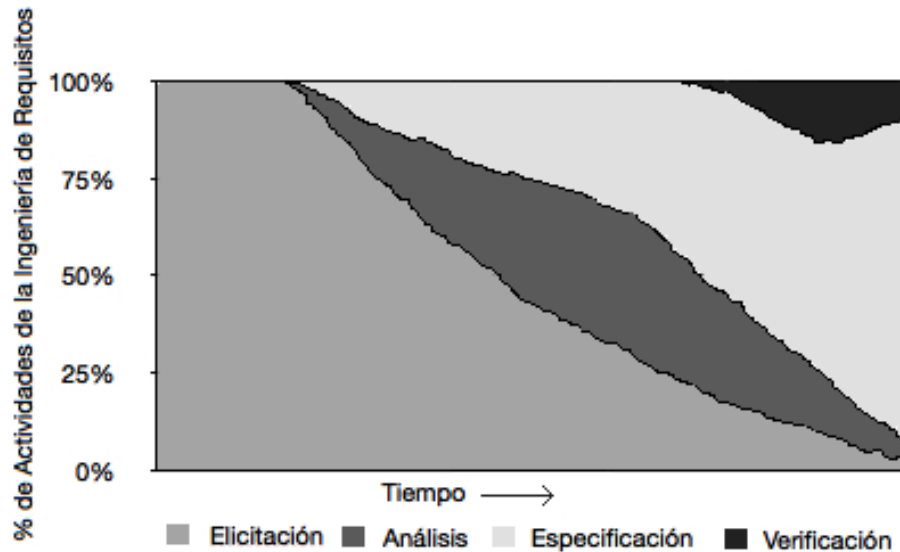


Figura 1.1. Tiempo empleado en cada actividad de la Ingeniería de Requisitos (Christel & Kang, 1992).

Es en la etapa de elicitación donde se consume el mayor tiempo de la IR debido a que surgen muchos problemas al interactuar con los *stakeholders* para obtener requisitos de calidad, puesto que se enfrenta uno de los principales problemas cuando incide el factor humano: la comunicación (Christel & Kang, 1992; Walia & Carver, 2009). Sin embargo existen diversas técnicas que ayudan a descubrir y obtener los requisitos software, pero hay que saber identificar cuáles se adecuan más a un proyecto en particular (Hickey & Davis, 2002). Debido a la naturaleza de esta etapa, es difícil saber si los requisitos obtenidos son de calidad, es decir si son completos, consistentes, trazables, no ambiguos y si satisfacen las necesidades de los *stakeholders* ya que no hay forma de medir y controlar este proceso como consecuencia de que solamente existen algunas métricas⁴ que permiten evaluar ciertos aspectos del producto final.

1.2. Importancia del problema

A pesar de que la IS surgió hace más de 50 años, se siguen registrando las mismas causas de fracaso en los proyectos software que dieron origen a esta disciplina (Gibbs, 1994; Charette, 2005). Por ejemplo, los reportes *Chaos* del *Standish Group*⁵ de los años 2002, 2004 y 2006 mostraban una mejoría en el éxito de los proyectos software (véase Figura 1.2). En el año 2006, Jim Johnson -presidente de *Standish Group*- argumentó que esto se debía principalmente a tres razones: una mejor gestión de proyectos, el desarrollo iterativo y la emergente infraestructura web. Sin embargo, como se puede ver, la publicación del reporte del 2008 revela que hubo un retroceso en los avances que se habían tenido en los últimos 7 años en la industria del software.

⁴ Una métrica es una medida que nos indica el grado en que un software, componente o proceso posee un atributo determinado (IEEE, 1990). Las métricas no son absolutas pero proporcionan una manera sistemática de evaluar la calidad a partir de un conjunto de reglas definidas (Sommerville, 2010).

⁵ El *Standish Group* es una empresa consultora que elabora uno de los estudios más citados en el contexto de proyectos tecnológicos, el *Chaos Report*, cuyo objetivo es documentar y examinar los éxitos y fracasos en el desarrollo y gestión de proyectos de Tecnologías de Información principalmente en Europa y Norteamérica.

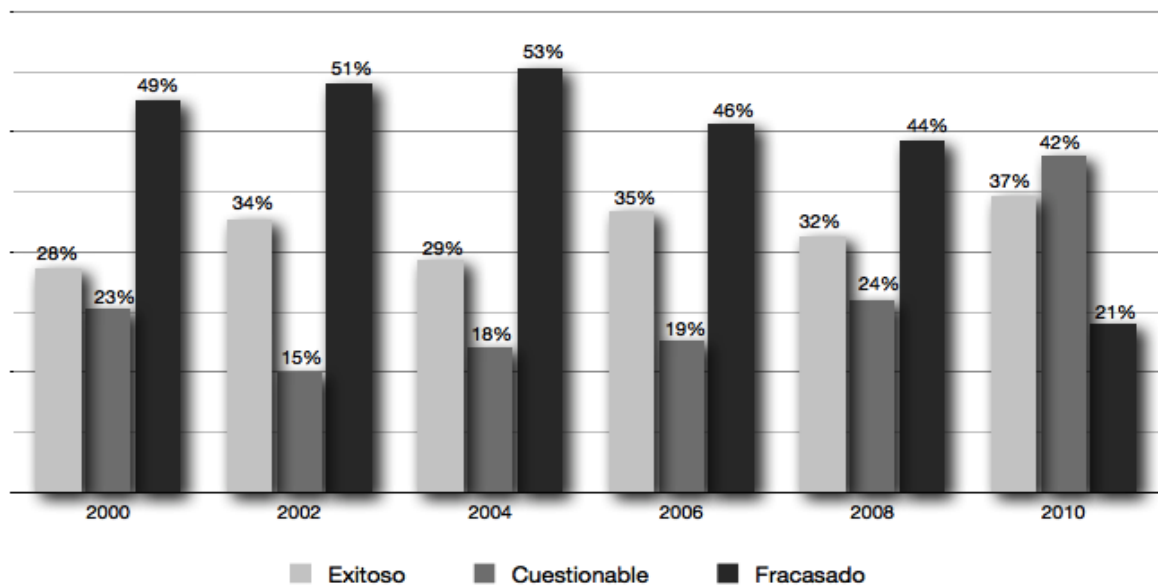


Figura 1.2. Clasificación de los proyectos para los años 2000-2010 (Standish, 2011).

Debe tomarse en cuenta que la primera medida de éxito de un software es el grado en el cuál éste cumple con el propósito para el que fue creado, es decir que satisface los requisitos. Sin embargo, a pesar de que se conocen muchas de las características que deben tener los requisitos correctamente obtenidos, todavía en la actualidad -principalmente en la industria- se incluyen requisitos de pobre calidad⁶ (Firesmith & Team, 2003).

En el 2010, el *Standish Group* identificó las “10 leyes del Caos”, colocando en el número uno la “*Ley de las Dos Caras*” (los *stakeholders* son para el software tanto sus mejores amigos como sus peores enemigos) argumentando que es necesario hacer una correcta identificación de los *stakeholders* que proveerán los requisitos correctos, tener una buena comunicación con ellos, hacer revisiones de los requisitos, priorizarlos, etc. En otras palabras, que el proceso de IR debe ser eficaz (Standish, 2010).

El reporte del 2010 muestra que hasta ese año, solamente el 37% de los proyectos software fueron “exitosos”, es decir que se terminaron en el tiempo que se planificó, que no se rebasó el presupuesto contemplado y que se satisficieron los requisitos de los *stakeholders*. El 42% fueron “cuestionables”, lo que significa que tuvieron atrasos en su entrega, que gastaron más de lo presupuestado, o que no cumplieron con todas las funcionalidades requeridas por el usuario. El restante 21% fueron “fracasos”, es decir que fueron cancelados o que no fueron utilizados (Standish, 2011).

⁶ De acuerdo con (Lauesen, 2002) los atributos de calidad de los requisitos establecen que estos deben ser: a) Correctos. Cada requisito refleja una necesidad, b) Completos. Toda la información necesaria es incluida, c) No deben ser ambiguos, esto significa que todos los *stakeholders* deben entender lo mismo, d) Consistentes. No debe haber requisitos en conflicto, e) Modificables. Los requisitos deben ser fáciles de modificar y mantener, f) Verificables. Debe haber una forma de revisar que el requisito se cumpla, y g) Trazables. Esto significa que se debe conocer la fuente de los requisitos, con qué otros se relaciona y en qué otras etapas del proceso software son utilizados.

Como se puede apreciar en el último reporte del *Standish Group*, el número de proyectos cuestionables representa la cifra más alta en toda la década. Los datos mostrados indican que aún se tienen problemas en la entrega de los productos, principalmente por la comunicación e integración con el cliente, que no se cubren correctamente ni completamente los requisitos, que existen procesos inmaduros así como un manejo inadecuado de los cambios, fallos en las estimaciones y un aumento en la complejidad de la tecnología actual (Standish, 2011).

En un estudio comparativo elaborado por Nizam y Sahibuddin en el año 2011 (Nizam & Sahibuddin, 2011), se determinó cuáles eran los factores que causan que los proyectos software no sean exitosos. Los resultados de este análisis mostraron que los requisitos son uno de los cinco factores críticos para el éxito de un proyecto software. De manera similar, el estudio elaborado por Kaur y Sengupta en el año 2011 (Kaur & Sengupta, 2011) analizó las fallas del software con el propósito de identificar los factores más importantes que las originan. La falla principal detectada en este estudio fue la elicitación de los requisitos (haciendo hincapié en su importancia al ser la primera etapa del proceso software y en donde se debe dejar claro lo que tiene que hacer el producto final).

En resumen, la evidencia presentada en los párrafos anteriores muestra que a pesar de los grandes avances tecnológicos y la evolución de las herramientas para desarrollar software aún se siguen construyendo productos que no cumplen con las expectativas de los *stakeholders*. Lo que indica que uno de los principales problemas puede rastrearse a las etapas tempranas del proceso software: *cuando es necesario definir lo que el producto debe hacer*.

Dentro de la IR, la elicitación de requisitos es considerada como una etapa crítica donde se invierte el mayor tiempo y esfuerzo, ya que uno de sus retos más importantes es garantizar que los requisitos software sean consistentes con las necesidades de la organización y con las futuras necesidades de los *stakeholders* (Reubenstein & Waters, 1991; Christel & Kang, 1992; Berenbach et al., 2009; Nonyelum, 2012).

El proceso de elicitación de requisitos implica un conjunto de actividades cuya responsabilidad es la comunicación, el establecimiento de prioridades, la negociación y la colaboración con todos los *stakeholders*. El éxito de esta etapa depende de una variedad de factores adicionales como el tiempo, el costo, la disponibilidad de recursos, la seguridad del software y todas las restricciones legales o reglamentarias (Zowghi & Coulin, 2005).

Un buen proceso de elicitación puede ayudar a diferenciar entre qué es lo que los *stakeholders* quieren y qué es lo que realmente necesitan (Raghavan, Zelesnik & Ford, 1994). Esta tarea puede parecer trivial sin embargo es, de hecho, fuente de mucha confusión y el origen de numerosos problemas (Scott, Clayton & Gibson, 1991; Abran et al., 2004). De acuerdo con (Kotonya & Sommerville, 2000) algunos de los problemas que se pueden presentar son los siguientes:

- Los *stakeholders* no conocen qué es lo que realmente quieren obtener del software.
- Cada *stakeholder* expresa los requisitos en sus propios términos.
- Los *stakeholders* especifican detalles innecesarios.
- No se comprende el dominio del problema, y
- Los requisitos son volátiles.

Es importante considerar que un proceso maduro de elicitación de requisitos puede ayudar a los *stakeholders* a entender qué es lo que realmente quieren, qué es lo que necesitan, cuáles son las restricciones y alternativas del software que se desarrollará (Sommerville, 2010). Es por esto que en este trabajo de tesis, se propondrán métricas que permitan medir y controlar este proceso antes de avanzar a la especificación de los requisitos. Como se ha mencionado anteriormente, en relación con

los factores de fracaso de un proyecto, deben definirse algunos indicadores de éxito para tener alguna medida de transparencia del proyecto, ya que es importante tener la capacidad de responder a las preguntas ¿se está teniendo un progreso real? y ¿cuál es la calidad de los productos de trabajo? Por ejemplo, en el contexto de esta tesis es necesario preguntarse ¿cómo se puede determinar si los requisitos obtenidos en la etapa de elicitación son de alta calidad?.

1.3. Necesidad de resolución

Para el Instituto de Ingeniería de Software (SEI, por sus siglas en inglés) de la Universidad de Carnegie Mellon, la experiencia de más de 40 años ha demostrado que unos requisitos incorrectos, incompletos o no entendidos son la causa más común de una pobre calidad, un costo excesivo y de una entrega tardía del software. Por lo que la habilidad de emplear un proceso sistemático de elicitación debe ser un aspecto fundamental para un ingeniero de software (Raghavan et al., 1994).

En consecuencia, para mejorar la calidad del software será necesario mejorar la calidad de los requisitos obtenidos, aspecto que se ve influenciado por las técnicas empleadas por el elicitor durante la etapa de elicitación de los requisitos debido a que en este proceso se determinan las necesidades de los usuarios y éstas se comunican a los desarrolladores (Hickey & Davis, 2002).

Pero es importante entender que el mejorar la elicitación de los requisitos no es una acción trivial, ya que es fuente de mucha confusión y el origen del fracaso de numerosos proyectos (Hickey & Davis, 2002; Standish, 2010). Por lo que se plantea que existe una alta probabilidad de que al mejorar el desempeño de esta etapa, a través del uso de métricas que permitan controlar y medir los requisitos obtenidos, se reflejen efectos positivos en el registro de éxitos en la industria del software.

De acuerdo con Pressman, *“La única forma racional de mejorar cualquier proceso es medir los atributos del proceso, desarrollar un conjunto de métricas significativas según estos atributos y entonces utilizar las métricas para proporcionar indicadores que conducirán a una estrategia de mejora”* (Pressman, 2009). Como en otras áreas de la ingeniería, la medición es algo común, puesto que permite llevar un control, sin embargo en la IR, sobre todo en la elicitación de requisitos, esto no ocurre. Ahora bien, *“algo que no se mide, no puede ser controlado y por ende no se puede saber si en realidad se está mejorando”* (DeMarco, 2009).

Las características de los requisitos o indicadores de calidad son extremadamente importantes para determinar la calidad del producto a desarrollar. Ya que éstas pueden ser medidas por inspección (a través de métricas), y las métricas reportadas pueden utilizarse para determinar la calidad de los requisitos de forma individual, así como de la especificación de los requisitos. Además, los datos históricos originados por el uso de métricas durante cierto tiempo pueden utilizarse para identificar tempranamente problemas potenciales, lo que permite realizar acciones correctivas y proporcionar orientación en cuanto a qué tipo de éstas se deben tomar. Por ejemplo, un alto nivel de ambigüedad en un conjunto de requisitos podría indicar que el miembro del equipo de desarrollo que lo obtuvo puede necesitar una formación complementaria sobre la escritura de los requisitos (Berenbach et al., 2009).

En la historia de la IS, se han establecido métricas para el producto y el proceso de software, sin embargo en lo que se refiere al área de IR, el uso de métricas ha sido muy poco desarrollado sobre todo en lo que respecta al área de elicitación que es la que concierne en este trabajo. Por lo que después de analizar las técnicas para elicitar requisitos y su efectividad será posible proponer una serie de métricas que ayuden a entender, controlar y mejorar el proceso de elicitación de requisitos.

1.4. Hipótesis

Teniendo en cuenta la problemática definida anteriormente, la hipótesis de la que se parte para la realización de este trabajo de tesis es:

“Es posible definir métricas que contribuyan a que los ingenieros de software entiendan, controlen y mejoren el proceso de elicitación de requisitos”.

1.5. Objetivo general

A partir de esta hipótesis se establece el siguiente objetivo general:

“Formular métricas para el proceso de elicitación de requisitos, que permitan controlar y medir la calidad de los mismos”.

1.6. Objetivos específicos

Para alcanzar este objetivo será necesario cumplir con los siguientes objetivos particulares:

- Analizar la importancia del proceso de elicitación de requisitos en el desarrollo software.
- Realizar una investigación sobre propuestas similares a la de este trabajo de tesis, y si las hay, analizarlas y establecer una base comparativa.
- Realizar una revisión literaria sobre la efectividad de las diferentes técnicas y/o métodos para la elicitación de requisitos.
- Identificar los aspectos de cada una de las técnicas y/o métodos que influyen de manera directa en la efectividad de las mismas.
- Proporcionar una guía sobre las técnicas de elicitación efectivas.
- Proponer una serie de métricas que midan la calidad de los requisitos en la etapa de elicitación.
- Validar las métricas propuestas en un caso de estudio para determinar si mejoran el proceso de elicitación.

1.7. Delimitaciones y limitaciones del trabajo

Para el desarrollo de esta tesis se tomarán en cuenta las siguientes delimitaciones y limitaciones:

- Este trabajo se enfocará únicamente a la etapa de elicitación de requisitos y probablemente a la de análisis, en lo referente a los atributos de calidad que deben poseer los requisitos.
- Se limitará el análisis de las técnicas de elicitación, ya que existen otras cuestiones referentes a la elicitación de requisitos, como la identificación de *stakeholders*.
- Debido a la gran variedad de técnicas de elicitación, solamente se analizarán las más maduras (en términos de antigüedad y efectividad) descritas en el marco teórico.
- El proyecto de software en que se basará este trabajo, está determinado por el tamaño del equipo (1-6 participantes (Cockburn, 2001)) debido a que la mayoría de empresas desarrolladoras de software en México son Micro, Pequeñas y Medianas Empresas

(MiPyMES), es decir que cuentan con 5 a 50 empleados (SE, 2004). Por lo cual se aplicará en un pequeño proyecto, que de acuerdo al SEI tiene 50000-100000 LOC, con una duración de 6-12 meses, y en el que participan menos de 10 programadores.

- Debido a la limitación que supone el validar estas métricas en diferentes empresas desarrolladoras de software, se propone la validación de éstas en una empresa a través de un caso de estudio durante la etapa de elicitación.

1.8. Aproximación a la solución

Como resultado del creciente número de estudios empíricos y teóricos sobre la IR se hace necesario aplicar un enfoque sistemático para evaluar y uniformar los resultados de la investigación, por esto se propone la aplicación de una revisión sistemática de literatura (SLR, por sus siglas en inglés), específicamente en el área de la elicitación de requisitos para obtener evidencia científica de los estudios sobre las técnicas utilizadas en esta etapa, así como su efectividad. De esta manera, basándose en la evidencia científica obtenida, se hará la propuesta de las métricas que los ingenieros de software puedan utilizar para entender, controlar y mejorar la elicitación de requisitos con el fin de producir software de alta calidad (Mitchell, Agle & Wood, 1997).

De acuerdo con Kitchenham una SLR es *un medio para evaluar e interpretar toda la información relevante disponible sobre una pregunta de investigación, tema de algún área particular o fenómeno de interés* (Kitchenham, 2004).

Una SLR examina e interpreta toda la literatura relevante en cierta área y tiene como objetivo presentar una evaluación de esta investigación usando una síntesis de una metodología rigurosa (Beechman, Badoo, Hall, Robinson & Sharp, 2007).

En este sentido, se seguirán las directrices establecidas por Kitchenham (Kitchenham, 2004; Kitchenham, Dyba & Jorgensen, 2004) para mostrar cuáles son las técnicas de elicitación más utilizadas y efectivas en la elicitación de requisitos.

De acuerdo con Kitchenham y Kitchenham et al., se deben seguir los siguientes pasos (Kitchenham, 2004; Kitchenham et al., 2004):

- Identificación de la necesidad de la revisión literaria.
- Formulación de preguntas de investigación.
- Establecimiento de términos de búsqueda.
- Definición de la estrategia de búsqueda.
- Selección de estudios.
 - Criterios de selección.
 - Procedimiento de selección.
- Estudio de evaluación de la calidad.
- Estrategia de extracción de datos.
- Síntesis de los datos extraídos.

En el Capítulo 3 de esta tesis se explica con más detalle la metodología utilizada así como la descripción de cada uno de los pasos mencionados anteriormente.

1.9. Estructura de la tesis

La estructura del documento de tesis se detalla a continuación.

En el Capítulo 2 se muestra el estado del arte en la etapa de elicitación de requisitos, se describen los procesos que involucra el mismo, los problemas que surgen mientras se lleva a cabo y las técnicas de elicitación que existen, por último se menciona el trabajo relacionado.

En el Capítulo 3 se describe la propuesta de solución, la metodología que se aplicará para lograrla así como el desarrollo de la SLR. También los hallazgos que permitan proponer métricas para la elicitación de requisitos y los procedimientos relacionados para su aplicación.

El Capítulo 4 muestra la validación de las métricas propuestas a través de su aplicación en un caso de estudio, los resultados y hallazgos encontrados.

Por último en el Capítulo 5 se presentan las conclusiones obtenidas durante el desarrollo de este trabajo.

En la sección de Anexos se encontrará la información necesaria para complementar el trabajo, glosario de acrónimos, guía de técnicas de elicitación, las plantillas para la recolección de los datos obtenidos al aplicar las métricas y el reporte de resultados.

2. Estado del Arte

2.1. Marco teórico

La elicitación de requisitos es un proceso delicado en donde el ingeniero de software debe capturar las necesidades y deseos de todas las partes involucradas. Esta etapa del desarrollo de software se define como *el proceso de identificar necesidades y de unificar disparidades entre las comunidades participantes (stakeholders), con el propósito de definir y restringir los requisitos dependiendo de las limitaciones propuestas por dichas comunidades* (Christel & Kang, 1992; Walia & Carver, 2009).

La elicitación de requisitos comienza inherentemente con un conocimiento informal y típicamente involucra a gente que no se encuentra fácilmente relacionada con el desarrollo de software, por lo que pueden surgir problemas. El producto resultante de esta fase es un subconjunto de objetivos de los *stakeholders* que describen una serie de posibles soluciones.

La calidad de los requisitos obtenidos se ve influenciada por las técnicas empleadas por el ingeniero de requisitos durante la elicitación debido a que en este proceso se determinan las necesidades de los usuarios y éstas se comunican a los desarrolladores del software (Hickey & Davis, 2003).

2.1.1. Procesos involucrados en la elicitación de requisitos

Así como el proceso de la IR se lleva a cabo en diferentes etapas, la elicitación de requisitos también puede descomponerse en un conjunto de actividades. Existen diferentes propuestas como las que se analizan a continuación:

2.1.1.1. Propuesta del SEI

Para el SEI, la elicitación de requisitos puede descomponerse principalmente en tres actividades: búsqueda, obtención de información e integración. Así mismo sirve como un marco para el desarrollo de software ya que envuelve aspectos sociales, comunicativos y técnicos, por lo que puede descomponerse en cinco procesos (Christel & Kang, 1992):

1. *Identificación de los stakeholders*, que son vistos como fuentes de requisitos. Éstos pueden ser el usuario final, un software, o factores ambientales, entre otros.
2. *Recolectar la "lista de deseos" de cada stakeholder*. Originalmente esta lista de deseos probablemente contenga ambigüedades, inconsistencias, requisitos no factibles, incompletos o no verificables.

3. *Documentar y refinar la "lista de deseos" de cada stakeholder.* En la lista de deseos se incluyen todas las actividades y datos importantes, y durante esta etapa esta lista es analizada repetidamente hasta que sea consistente. La lista es usualmente de alto nivel, específica el dominio del problema y es elaborada en términos del usuario.
4. *Integrar las listas de deseos a través de diversos stakeholders,* para resolver los conflictos que puedan surgir entre los puntos de vista. La información debe ser coherente y los objetivos viables.
5. *Determinar los requisitos no funcionales,* tales como problemas de rendimiento y fiabilidad, etc., y el estado de éstos en el documento de requisitos.

A pesar de proponer esta diferenciación de actividades, no son claros los medios que se necesitan para llevarlas a cabo.

2.1.1.2. Modelo⁷ del SWEBOK

En el modelo propuesto por el Cuerpo de Conocimiento de Ingeniería de Software (SWEBOK, por sus siglas en inglés) de la IEEE (Abran et al., 2004) la elicitación es considerada como la etapa donde se comienzan a construir el entendimiento y la comprensión del problema a resolver. Además es en ella donde se identifican y establecen las relaciones entre los *stakeholders*; ya que uno de los principios fundamentales de la IR es que exista una buena comunicación entre los usuarios y los ingenieros de software (problema crucial en el desarrollo). Antes de que comience la elicitación, los especialistas en requisitos deben establecer una pauta para la comunicación- ya que son ellos los que deben mediar entre el dominio de los usuarios de software (y otros *stakeholders*)- y las palabras técnicas utilizadas por el ingeniero de software. La elicitación se ocupa de dos aspectos fundamentales: primero, de dónde se obtendrán los requisitos y segundo, cómo se van a recolectar.

La elicitación de requisitos abarca dos áreas:

- Las fuentes de requisitos. Los requisitos software pueden provenir de muchas fuentes y es esencial que todas éstas sean identificadas y que se evalúe su impacto. Esta área esta diseñada para promover el conocimiento de varias fuentes de requisitos y del entorno de trabajo, y así poder gestionar la información obtenida. Los principales puntos a cubrir son:
 - *Metas.* Este término se refiere a los objetivos de alto nivel del software, a los objetivos de negocio o al factor crítico de éxito.
 - *Dominio del conocimiento.* El ingeniero de software necesita tener conocimiento acerca del dominio de aplicación; se puede adquirir o inferir tácitamente a partir de los *stakeholders* cuando éste no esté articulado.
 - *Stakeholders.* El ingeniero de software necesita identificar, representar y gestionar todos los puntos de vista de los diferentes tipos de *stakeholders*.
 - *Ambiente operacional.* Los requisitos son derivados del medio en el cuál se va a ejecutar el software, esta información incluye cierto tipo de restricciones por ejemplo si el software necesita ejecutarse en tiempo real o bien, cómo será la operatividad entre oficinas.

⁷ De acuerdo con la Real Academia de la Lengua un modelo es un esquema teórico de una realidad compleja, que se elabora para facilitar su comprensión y el estudio de su comportamiento (RAE, 2013).

- *Ambiente organizativo*. En muchas ocasiones el software es requerido para apoyar el proceso de negocio, por lo que está condicionado por la estructura, la cultura y las políticas internas de la organización. El ingeniero de software necesita comprender esta situación ya que el nuevo software no deberá afectar ni cambiar los planes de negocio.
- Las técnicas de elicitación. Una vez que las fuentes de los requisitos han sido identificadas, el ingeniero de software puede empezar a elicitar los requisitos. Esta área se enfoca en las técnicas para obtener los requisitos a partir de los *stakeholders*, lo que la hace una actividad difícil ya que se debe elicitar la información correcta.

A pesar de que este modelo provee cuál es la motivación para desarrollar el software, algunos aspectos son vagamente definidos.

2.1.1.3. Modelo de Kotonya

Para Kotonya y Sommerville (Kotonya & Sommerville, 2000), la elicitación de requisitos está formada por cuatro actividades:

- *Entendimiento del dominio de la aplicación*, tener conocimiento general sobre el área en que el software será aplicado.
- *Entendimiento del problema*, en donde deberán comprenderse por completo los detalles específicos del problema de los *stakeholders* y el entorno en el cual será aplicado el software.
- *Entendimiento del negocio*, es decir, se deberá entender cómo es que el software contribuirá e interactuará con los objetivos de negocio.
- *Entendimiento de las necesidades y restricciones de los stakeholders*, se deben entender con detalle las necesidades específicas de la gente que requiere el software.

Una vez que se ha definido el proceso de elicitación, los autores dividen éste en cuatro etapas:

1. *Establecimiento de objetivos*. Los objetivos organizacionales deben ser establecidos incluyendo las metas generales del negocio, una descripción del problema a resolver, restricciones del mismo y el por qué es necesaria la construcción del software.
2. *Adquisición del conocimiento general*, este conocimiento debe incluir información acerca de la organización donde será instalado el software, así como su dominio de aplicación y la información sobre el software ya existente.
3. *Organización del conocimiento obtenido*, el conocimiento obtenido en la etapa anterior debe ser organizado de acuerdo a los siguientes lineamientos:
 - Conocimiento particionado, en donde se deberá considerar la información de acuerdo a las relaciones de agregación que se establezcan.
 - Conocimiento abstracto, en donde están las generalidades.
 - Conocimiento de proyección, que consiste en organizar el conocimiento de acuerdo con la perspectiva del software.
4. *Recolección de los requisitos de los stakeholders*, los *stakeholders* del proyecto deberán ser consultados para descubrir los requisitos.

Cómo se puede observar en lo expuesto anteriormente, existen actividades que son comunes en las propuestas analizadas donde se define el proceso para la elicitación de requisitos, por lo que es posible llegar a un consenso respecto a que la elicitación no es un proceso que se lleve a cabo en una sola actividad, y que debe comprender las siguientes actividades principales:

1. **Identificación de los *stakeholders***, quienes proporcionan la información necesaria para conocer cuál es el problema a resolver, así como sus necesidades y restricciones.
2. **Obtención de la lista de deseos de cada *stakeholder*** a través de la aplicación de las técnicas de elicitación que permitan la extracción de dicha información o conocimiento para su análisis posterior.
3. **Integración, refinamiento y organización de la información obtenida** para poder determinar cuáles son las funcionalidades y limitaciones del software a desarrollar.

Puesto que en esta etapa se requiere de mucha información, pueden surgir algunos problemas ocasionados, ya sea, porque el tiempo para la elicitación no fue suficiente, porque los ingenieros de software carecen de una preparación adecuada, porque los *stakeholders* no se encuentran convencidos de la necesidad de un nuevo software, o bien, porque no existió una adecuada comunicación entre los participantes de esta etapa (Kotonya & Sommerville, 2000). A continuación se definen algunos problemas que es posible encontrar en esta etapa.

2.1.2. Problemas en la elicitación de requisitos

Puesto que la elicitación es un proceso social, los problemas que se presentan en esta etapa no pueden ser resueltos de una forma solamente técnica, ya que debe tomarse en consideración el contexto social en el que se desarrolla este proceso (Goguen & Linde, 1993). Además de que es un proceso impreciso y difícil, por lo que se pueden encontrar algunas dificultades en la realización de esta actividad (Raghavan et al., 1994).

El producto resultante de esta etapa debe ser un subconjunto de metas procedentes de las partes directamente involucradas (*stakeholders*) que describan un número de soluciones posibles, mismas que posteriormente deberán ser validadas para evitar problemas de ambigüedad, exactitud, corrección, completitud, inconsistencia, verificación y trazabilidad. Sin embargo, los problemas que se presentan en esta etapa pueden inhibir la definición y calidad de la especificación de requisitos (Christel & Kang, 1992), por lo que los ingenieros de software deben poner atención al evaluar la prioridad y costo de llevar a cabo ciertos requisitos (Loucopulos & Karakostas, 1995).

Algunos de estos problemas, agrupados de acuerdo al SEI, SWEBOK, Raghavan et al. y Sommerville se describen a continuación.

2.1.2.1. Según el SEI

De acuerdo con el SEI (Christel & Kang, 1992) los problemas de elicitación de requisitos se pueden agrupar en tres categorías:

- *Los problemas de alcance*. Los requisitos pueden abarcar mucha o poca información, por lo que deben especificarse correctamente los límites del software. La elicitación debe enfocarse a la obtención de requisitos y no al diseño de actividades, ya que esto puede resultar en la obtención de requisitos pobres. Dentro de esta categoría se mencionan los siguientes problemas:
 - El alcance del software está mal definido.
 - Información de diseño innecesaria.

- *Problemas de entendimiento* ocasionados dentro de los grupos de *stakeholders* por una pobre comunicación. Estos problemas pueden conducir a obtener requisitos ambiguos, incompletos, inconsistentes y por lo regular incorrectos debido a que no representan las verdaderas necesidades de los participantes. Algunos problemas de entendimiento pueden ser los siguientes:
 - Los *stakeholders* tienen una comprensión incompleta de sus necesidades.
 - Los *stakeholders* tienen una pobre comprensión de las limitaciones tecnológicas.
 - Los ingenieros de software tienen poco conocimiento del dominio del problema.
 - Los *stakeholders* hablan diferentes idiomas.
 - Los *stakeholders* omiten información “obvia”.
 - Los *stakeholders* tienen puntos de vista contradictorios.
 - Los requisitos son a menudo vagos y no comprobables, por ejemplo, fácil de usar y robusto.
- *Problemas de volatilidad*, es decir, los requisitos pueden cambiar a lo largo del tiempo que tome desarrollar un proyecto, ya que las necesidades pueden ir madurando. De la misma forma otras necesidades pueden surgir y si los nuevos requisitos o los cambios no son acomodados adecuadamente los requisitos originales serán incompletos o inconsistentes con la nueva situación, y potencialmente inusables por que representarán información obsoleta. El problema que se puede presentar en esta categoría es:
 - Requisitos que evolucionan con el tiempo.

2.1.2.2. Según el SWEBOK

En el SWEBOK se describe la etapa de elicitación como una actividad fundamentalmente humana dado que es aquí donde se identifica a las personas que van a proveer los requisitos (*stakeholders*), además se establecen las relaciones entre el equipo de desarrollo y estos *stakeholders* (Abran et al., 2004). Por este motivo la elicitación de requisitos es catalogada como una actividad difícil de llevar a cabo y se pueden presentar los siguientes problemas:

- No se identifica correctamente a los *stakeholders*.
- No se establece una buena relación entre los *stakeholders* y el equipo de desarrollo.
- No existe una buena comunicación entre los *stakeholders* y el equipo de desarrollo.
- El equipo de desarrollo se desenvuelve en un mundo técnico.
- Los *stakeholders* se desenvuelven en su propio dominio.
- Los *stakeholders* y el equipo de desarrollo no hablan en los mismos términos.
- Los *stakeholders* tienen dificultades para describir sus tareas.
- Los *stakeholders* proporcionan información inestable, inconsistente o ambigua.
- Los *stakeholders* no quieren cooperar voluntariamente.
- Los especialistas en requisitos no eligen las técnicas de elicitación adecuadas para extraer correctamente los requisitos.
- Los especialistas en requisitos no extraen la información correcta.

- Los especialistas en requisitos no conducen bien el proceso de elicitación.

Debido a que la elicitación de requisitos no es una actividad pasiva y a que se deben establecer las relaciones entre el equipo de desarrollo y los *stakeholders*, pueden surgir algunos de los problemas mencionados anteriormente. Sin embargo existen técnicas de elicitación de requisitos que ayudan a lidiar con algunos de estos problemas para obtener la información correcta de los *stakeholders*.

2.1.2.3. Clasificación de Raghavan

Raghavan, Zelesnik y Ford han identificado un conjunto de problemas que han clasificado en cinco categorías (Raghavan et al., 1994):

- *Problemas de articulación.* En esta categoría se incluyen los problemas relacionados con la articulación de las necesidades de los *stakeholders*. Éstos incluyen problemas relacionados con la expresión de las necesidades de los *stakeholders* y el conocimiento de los desarrolladores.
 - Los *stakeholders* pueden estar conscientes de sus necesidades, pero son incapaces de articularlas adecuadamente.
 - Los *stakeholders* pueden no estar conscientes de sus necesidades.
 - El *stakeholder* puede estar al tanto de una necesidad, pero tiene miedo de expresarla.
 - Los *stakeholders* y desarrolladores malinterpretan conceptos o relaciones, ya que tienen diferentes significados para los términos comunes.
 - Los *stakeholders* no pueden decidirse en algunas cuestiones porque no entienden las consecuencias de la decisión o no entienden las alternativas.
 - Ninguna persona tiene una visión completa del problema.
 - Los desarrolladores no escuchan a los *stakeholders*.
 - Los desarrolladores no comprenden, no aprecian y no se relacionan con los *stakeholders*.
 - Los desarrolladores tienden a anular o dominar a los *stakeholders*.
- *Barreras de comunicación.* Las barreras construidas en la comunicación entre los *stakeholders* y desarrolladores pueden ocasionar muchas dificultades en la elicitación de los requisitos, como las siguientes:
 - Los *stakeholders* y desarrolladores manejan diferentes vocabularios profesionales.
 - Los *stakeholders* tienen preocupaciones diferentes a las de los desarrolladores, estas se relacionan por lo general con los atributos de alto nivel como la facilidad de uso y fiabilidad. Por el contrario, las preocupaciones de los desarrolladores tienen que ver con cuestiones técnicas de bajo nivel, tales como la utilización de recursos, los algoritmos y el hardware/software.
 - Existen problemas con cada formulario o medio de comunicación utilizado.
 - La elicitación de requisitos, por su propia naturaleza es de interacción social, y todos los *stakeholders* son diferentes.

- Existen diferentes tipos de personalidades así como valores entre las personas. Esto puede conducir a dificultades inesperadas en la comunicación.
- *El conocimiento y las limitaciones cognitivas.* Los *stakeholders* y desarrolladores son seres humanos, por lo que cada uno tiene limitaciones cognitivas.
 - El elicitor debe tener un conocimiento adecuado del dominio.
 - Ninguna persona tiene una memoria perfecta.
 - A menudo se utiliza información cuantitativa y estadística para expresar las necesidades y deseos. Sin embargo, las estadísticas informales o intuitivas se interpretan con frecuencia de manera diferente por diferentes personas a causa de las propias experiencias y prejuicios.
 - Las personas a veces tienen dificultades con la escala y complejidad del problema a resolver. Es decir, algunas personas tratan de simplificar el problema, pero no siempre de una forma válida y, otras simplemente ignoran partes del mismo porque no pueden hacerle frente. Por todo esto, la perspectiva del problema puede distorsionarse.
 - Se tiene la tendencia a plantear el problema en términos de la solución preferida.
 - Algunas personas centran rápidamente toda su atención en algunos aspectos limitados del problema, por lo general aquellos aspectos que creen que entienden mejor, o que les afectan directamente.
 - En proyectos grandes, por lo general hay que explorar una variedad de nuevas formulaciones del problema antes de llegar a un consenso sobre la naturaleza del mismo. Algunas personas se sienten incómodas o impacientes con este tipo de exploración.
- *Problemas de comportamiento humano.* La elicitación de requisitos es un proceso social, por lo que se involucran los problemas de conducta.
 - A veces existen conflictos y ambigüedades en las funciones que los *stakeholders* y desarrolladores juegan en el proceso de elicitación requisitos.
 - Algunos *stakeholders* no proporcionan información al elicitor o desarrolladores y en casos extremos sabotean el esfuerzo de desarrollo.
- *Cuestiones técnicas.* Existe otro tipo de dificultades que se podrían caracterizar como técnicas, las más importantes se resumen a continuación.
 - Los problemas a resolver por el software son cada vez más complejos.
 - Los requisitos cambian con el tiempo.
 - Las tecnologías de software y hardware están cambiando rápidamente.
 - Existen muchas fuentes de requisitos.
 - La naturaleza o la novedad del software a menudo impone limitaciones en el proceso de elicitación. Un nuevo software que es similar a otros podría beneficiarse de requisitos anteriores ya que un software completamente novedoso requiere mucho más esfuerzo.

2.1.2.4. Clasificación de Sommerville

Para Sommerville (Sommerville, 2010) la elicitación de requisitos es un proceso difícil por varias razones:

- Los *stakeholders* usualmente no conocen realmente lo que desean obtener de un software, excepto en términos muy generales; a veces encuentran difícil articular lo que quieren del software y otras veces hacen demandas irreales debido a que no conocen los costos de sus peticiones.
- Los *stakeholders* expresan los requisitos con sus propios términos utilizando el lenguaje natural y con el conocimiento implícito de su trabajo. Los ingenieros de requisitos deben comprender estos requisitos a pesar de no tener experiencia en el dominio del cliente.
- Diferentes *stakeholders* tienen requisitos distintos y los expresan de varias formas. Los ingenieros de requisitos deben identificar a todos los *stakeholders* y obtener sus requisitos.
- Los factores políticos influyen en los requisitos software.
- El entorno económico y de negocios en el que se lleva a cabo la elicitación de requisitos es dinámico, por lo que los requisitos cambian.

En resumen, la mayoría de los problemas descritos son ocasionados porque la comunicación que se tiene que llevar a cabo para extraer la información necesaria no fue la correcta. Puesto que la elicitación es una interacción social y la gente involucrada en esta etapa es diferente, pueden surgir dificultades sobre todo de compatibilidad. Esto puede ocasionar una interrupción en la comunicación, sin embargo es labor del elicitador buscar un canal de comunicación apropiado. Por todos los problemas que se pueden presentar en la elicitación de requisitos, se han desarrollado técnicas para resolverlos, como las que se describen en la siguiente sección.

2.1.3. Técnicas utilizadas en la elicitación de requisitos

Para obtener todo el conocimiento necesario en esta etapa, es imprescindible aplicar técnicas específicas que permitan recolectarlo. En los modelos mencionados anteriormente, donde también describen la metodología para llevar a cabo la elicitación, se menciona el uso de técnicas para poder extraer los requisitos.

Una técnica de elicitación puede definirse como un método para mediar la comunicación entre los *stakeholders* de un proyecto (Coughlan & Macredie, 2002). Las técnicas de elicitación deben ser empleadas para obtener la información necesaria para desarrollar un software de calidad. Por lo tanto la calidad de los requisitos obtenidos se ve influenciada por las técnicas empleadas por el elicitador durante este proceso, debido a que en éste se conocen las necesidades de los usuarios y éstas se comunican a los desarrolladores del software (Hickey & Davis, 2003).

Existen muchas técnicas para realizar la elicitación, sin embargo el propósito de todas ellas es el de asistir al ingeniero de software en el entendimiento de las necesidades de los *stakeholders* por lo que el problema que se presenta es escoger una o varias de ellas. Algunos ingenieros piensan que solo una técnica puede ser aplicable a todo tipo de situaciones, pero no es así, ya que existen condiciones que una sola técnica no puede satisfacer (Hickey & Davis, 2003).

Puesto que la elicitación es un proceso iterativo, en cualquier momento las condiciones del proyecto pueden obligar al elicitador a utilizar una técnica específica. El uso de esa técnica puede cambiar las condiciones anteriores y entonces el elicitador tendrá que usar otra técnica diferente

(Hickey & Davis, 2003). Por lo que el elicitador debe conocer y seleccionar aquellas técnicas de elicitación que mejor se adapten a las características y al dominio del proyecto.

Es particularmente importante entender que la elicitación no es una actividad pasiva, y que, aún cuando los *stakeholders* estén disponibles y cooperen en el proceso, el ingeniero de software debe realizar la difícil tarea de obtener la información correcta.

A continuación se describen los *frameworks* propuestos por el SEI, Sommerville y por Nuseibeh y Easterbrook; así como la clasificación propuesta por el SWEBOK y Hickey y Davis sobre la gran variedad de técnicas existentes para la elicitación de los requisitos.

2.1.3.1. Categorización del SEI

Para el SEI las técnicas utilizadas durante la elicitación de los requisitos pueden agruparse en tres categorías según la actividad en la que sea utilizada:

- Obtención de información,
- Análisis y expresión de los requisitos, y
- Validación

La Tabla 2.1 muestra el nivel de aplicación de las técnicas de elicitación dependiendo de la actividad a desarrollar. Esta tabla contiene dos columnas que muestran las propiedades de dichas técnicas. La columna con la etiqueta “*Madurez de la técnica*” indica el tiempo de aplicación de las técnicas. Algunas técnicas son muy maduras puesto que han sido utilizadas por décadas pero existen otras que no son muy utilizadas. La columna etiquetada “*Técnica Definida y Prescriptiva*” muestra la cantidad de indicaciones que conoce el elicitador sobre la técnica (Christel & Kang, 1992).

De acuerdo al SEI, las principales técnicas de elicitación son las siguientes:

2.1.3.1.1. Entrevistas

Las entrevistas son quizás la principal técnica utilizada para la obtención de información, y pueden ser usadas para obtener diversos tipos de conocimiento, dependiendo de a quién se le pregunte y qué tipo de preguntas se elaboren (Lauesen, 2002). Sin embargo, la información recolectada por medio de esta técnica puede guiar hacia factores organizacionales y contextuales que deben tomarse en cuenta durante la elicitación. Si la gente entrevistada es la correcta, la información obtenida representará opiniones de múltiples *stakeholders* de un número de comunidades diferentes afectadas por el desarrollo del software propuesto; lo que representa una enorme responsabilidad para el analista de requisitos el tener que integrar esas diferentes apreciaciones, metas, objetivos y estilos de comunicación utilizando una terminología común. Este problema puede resolverse en gran parte utilizando formatos estructurados o modelos de dominios, ya que éstos permiten mejorar el entendimiento y evitar la ambigüedad entre equipos multidisciplinarios por que proveen una estructura para la adquisición y expresión de los requisitos (Christel & Kang, 1992).

La entrevista estructurada es utilizada, por lo general, cuando ya se tiene algún conocimiento previo del problema, contexto o dominio de aplicación. Las no estructuradas son más abiertas ya que permiten indagar más y se recomiendan cuando no existe mucho conocimiento del problema a resolver. También existen las entrevistas individuales y entrevistas en grupo (Lauesen, 2002), en las segundas un grupo de personas de la misma área de trabajo proporciona información acerca de sus actividades, problemas y temas críticos, de esta forma se obtiene información más completa que la obtenida en las entrevistas individuales.

Tabla 1. Correspondencia de las técnicas de elicitación a un conjunto de propiedades (Christel & Kang, 1992).

Clave ✓✓ - (1) técnica que reconoce fuertemente tal cuestión y provee un significado para tratarla; (2) posee la calidad que se necesita. ✓ - (1) técnica que tiene esta propiedad, aunque quizá indirectamente o no tan fuerte como otra técnica; (2) la técnica posee calidad pero de manera limitada. Sin marca - técnica que no posee del todo esta propiedad o provee muy poco soporte para esta cuestión.	Análisis organizacional/contextual	Nulidad de un diseño prematuro	Varios colaboradores	Varias disciplinas colaboradoras	Problemas de gran tamaño	Múltiples expresiones	Volatilidad	Madurez de la técnica	Técnica definida y prescriptiva
Recolección de Información									
Entrevistas	✓		✓	✓				✓✓	
Entrevistas estructuradas	✓		✓✓	✓			✓	✓	✓✓
Enfoque de equipo	✓✓		✓	✓✓				✓✓	✓
Uso de modelos de dominio y arquitectónicos	✓✓				✓	✓	✓		
Expresión y Análisis de requisitos									
Expresión Controlada de Requisitos (CORE, por sus siglas en inglés)	✓		✓	✓		✓		✓✓	✓✓
Múltiples vistas de Delugach	✓		✓✓	✓✓		✓✓			
Metodología de Sistemas de Software (SSM, por sus siglas en inglés)	✓		✓	✓		✓	✓		✓
Metodología de Planeación y Diseño (PDM, por sus siglas en inglés)									
Validación									
Desarrollo de Funciones de Calidad (QFD, por sus siglas en inglés)		✓			✓		✓		✓✓
Concepto de prototipado					✓	✓	✓		
Proceso iterativo usado de técnicas de elicitación					✓		✓✓		

2.1.3.1.2. Lluvia de ideas

En una sesión de lluvia de ideas se reúne a un grupo de personas que son dirigidas por un facilitador o moderador. Éste debe crear un ambiente estimulante para permitir que la gente que tiene ideas, las expresen sin correr el riesgo de ser evidenciadas o puestas en ridículo (Lauesen, 2002). Generalmente el facilitador anota todas las ideas en una pizarra y así cada idea genera nuevas ideas. Una regla importante de esta técnica es que no se debe criticar ninguna idea. Sin embargo a causa de esto se obtienen ideas poco realistas que necesitan ser priorizadas en conjunto con los participantes.

Durante la lluvia de ideas, la atención se centra en los objetivos y los requisitos para el nuevo software. Esta técnica es utilizada regularmente para plantear algunas cuestiones que se han identificado en las entrevistas.

2.1.3.1.3. Diseño de aplicaciones conjuntas (JAD, por sus siglas en inglés)

Esta técnica, es similar a la lluvia de ideas, la diferencia radica en que ésta última es más organizada. El objetivo del JAD (por sus siglas en inglés), es involucrar a todos los *stakeholders* a través de reuniones altamente estructuradas y focalizadas con el objetivo de promover la cooperación, entendimiento y trabajo en equipo. El propósito de esta técnica es que los *stakeholders* tengan una visión compartida del software a desarrollar (Raghavan et al., 1994). Los participantes típicos en una sesión incluyen a un elicitor, a usuarios finales del producto, a los desarrolladores principales, y a los observadores.

El JAD tiene dos grandes pasos: la planeación y el diseño (Raghavan et al., 1994). El primer paso se enfoca en la elicitación y especificación de requisitos y el segundo, a la etapa de diseño del software. Cada paso se desarrolla en tres fases: *preparación, sesión y conclusión*. La primera fase consiste en la preparación de tareas para la sesión, lo que incluye organizar el equipo y preparar los materiales. La fase de sesión consiste en una o más reuniones estructuradas donde participan desarrolladores y *stakeholders*, en esta fase es donde se recopilan y documentan los requisitos o en dado caso el diseño. En la última fase se vacía la información obtenida en un formato final como la SRS.

2.1.3.1.4. Desarrollo de funciones de calidad (QFD, por sus siglas en inglés)

Es otra técnica útil para la elicitación y validación de requisitos. QFD permite que la voz del cliente sea capturada, y después de que los requisitos del sistema sean validados en función de si estos reflejan o no estas necesidades expresadas por el cliente (Christel & Kang, 1992). QFD ayuda a identificar necesidades de los usuarios que no han sido abordados por el analista, así como características de la propuesta que no admiten ningún requisito. Además de destacar estas omisiones, QFD también documenta los requisitos que son altamente valorados por el usuario y reciben poca atención por parte de los integrantes del equipo.

2.1.3.1.5. Escenarios

Se trata de una sesión de interacción para describir una secuencia de acciones y eventos para el caso específico de una tarea genérica que el software debe llevar a cabo (Kotonya & Sommerville, 2000). Esta técnica es valiosa para proveer un contexto a la elicitación de requisitos, ya que proporciona un *framework*⁸ para formular preguntas acerca de las tareas del usuario, tales como ¿qué pasaría sí? ó ¿cómo se lleva a cabo esta actividad? (Abran et al., 2004).

Esta técnica sirve para aclarar los requisitos relacionados con los procedimientos y flujos de datos. En este sentido, los escenarios son utilizados para proveer mas detalles acerca de la funcionalidad de un software (Sommerville, 2010). El tipo más común de escenario son los casos de uso.

En los casos de uso se muestra el flujo normal de operación, sin embargo también se consideran los flujos alternativos para cubrir todas las posibilidades de eventos que puedan ocurrir al utilizar el software. Se mencionan también las precondiciones y post-condiciones necesarias para alcanzar una funcionalidad, así como los actores que participan o interactúan con el software.

⁸ De acuerdo a la definición del diccionario Word Net un *framework* define un marco de trabajo y en términos generales es una estructura que da soporte a una descripción hipotética de una entidad compleja o proceso (Princeton, 2013).

2.1.3.1.6. Prototipos

Un prototipo es una versión simplificada de una parte del software final. Los desarrolladores experimentan con el prototipo para tener una idea de cómo debe funcionar el software en la vida real. Con esta técnica se pueden obtener dos tipos de requisitos: a nivel del producto y a nivel de diseño (Lauesen, 2002). Para el primer caso, el prototipo demuestra si la funcionalidad requerida es realista (viable) y útil. Los requisitos pueden ser expresados como características o como una descripción de la tarea. En el caso de los requisitos a nivel de diseño, se obtienen los requisitos de la interfaz del producto real ya que ésta debe ser exactamente igual que el prototipo.

Así mismo, los prototipos sirven para clarificar los requisitos, ya que proporcionan un contexto para un mejor entendimiento de la información que los *stakeholders* necesitan proveer.

Esta técnica abarca a su vez, una amplia gama de técnicas que van desde pruebas, dibujos en papel, hasta versiones beta de productos de software. Existen también dos tipos de prototipos los evolutivos y los prototipos desechables (Davis, 1992). Los prototipos evolutivos se utilizan a lo largo del proceso de desarrollo software, mientras que los desechables solo se utilizan en esta etapa de elicitación para la obtención de requisitos.

2.1.3.2. *Framework de Nuseibeh-Easterbrook*

Nuseibeh y Easterbrook proponen la agrupación de las técnicas de elicitación de requisitos en las siguientes clases (Nuseibeh & Easterbrook, 2000):

2.1.3.2.1. Técnicas tradicionales

Este grupo contiene una amplia gama de técnicas utilizadas para la obtención de conocimiento. Éstas incluyen el uso de cuestionarios, encuestas, entrevistas, análisis de la documentación existente (organigramas, modelos de procesos, normas o manuales del software existente).

2.1.3.2.2. Técnicas grupales de elicitación

Esta clase de técnicas tiene por objetivo fomentar acuerdos entre los *stakeholders*, explotando las dinámicas de equipo para obtener una comprensión más profunda de las necesidades. Dentro de este grupo se incluyen la lluvia de ideas, grupos de trabajo, así como talleres de RAD⁹ / JAD.

2.1.3.2.3. Prototipos

Los prototipos se deben utilizar en el caso de que exista una gran incertidumbre acerca de los requisitos, o dónde es necesaria la realimentación temprana entre los *stakeholders* (Davis, 1992). Los prototipos se pueden combinar fácilmente con otras técnicas, por ejemplo para generar una discusión utilizando una técnica de elicitación grupal, o como base para la elaboración de un cuestionario o en un protocolo en el que se dice en voz alta lo que se piensa (*think aloud protocol*).

2.1.3.2.4. Técnicas de modelado

Ofrecen un modelo específico del tipo de información a reunir, y utilizan este modelo para impulsar el proceso de elicitación. Usualmente se utilizan modelos como diagramas de flujos de

⁹ El Desarrollo Rápido de Aplicaciones (RAD, por sus siglas en inglés) es una metodología de desarrollo de software que emplea el Diseño de Aplicaciones Conjuntas (JAD, por sus siglas en inglés) para obtener los requisitos de los *stakeholders*, prototipos, tecnología de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés), generadores de aplicaciones y herramientas similares para acelerar el proceso de diseño (Davis & Yen, 1999).

datos, diagramas de estado, o diagramas UML o métodos basados en metas como KAOS e I*, y métodos basados en escenarios como CREWS.

2.1.3.2.5. Técnicas cognitivas

Incluyen una serie de técnicas desarrolladas originalmente para la adquisición de conocimientos útiles en el desarrollo de software basado en el conocimiento. Estas técnicas incluyen el *análisis de protocolo* (en el que un experto piensa en voz alta mientras realiza una tarea, para proporcionar al observador conocimientos sobre los procesos cognitivos utilizados para realizar la tarea), la *jerarquización (laddering* en inglés) que emplea sondeos para obtener la estructura y el contenido del conocimiento de los *stakeholders*, la *clasificación de tarjetas* en la cuál se pide a los *stakeholders* clasificar tarjetas en grupos dependiendo del nombre de la entidad de dominio que contiene y *rejillas de repertorio* en dónde se construye una matriz de atributos para las entidades, preguntando a los *stakeholders* los atributos aplicables a las entidades y los valores de las celdas para cada entidad.

2.1.3.2.6. Técnicas contextuales

Surgen en la década de los 90's como una alternativa a las técnicas tradicionales y cognitivas. Estas incluyen el uso de técnicas etnográficas como la observación. También se incluyen la etnometodología y el análisis de conversación para el análisis de grano fino en la identificación de patrones en una conversación o interacción.

Este *framework* muestra un gran desacuerdo entre los que proponen las técnicas contextuales y los que siguen utilizando las técnicas tradicionales. Las técnicas contextuales están basadas en la premisa de que el contexto es vital para el entendimiento del comportamiento social y organizacional y que el observador debe estar inmerso en este contexto para experimentar cómo es que los participantes crean sus propias estructuras sociales (Raghavan et al., 1994). Las técnicas contextuales tienen más probabilidad de descubrir el conocimiento tácito que las técnicas tradicionales (Coughlan & Macredie, 2002). Las técnicas tradicionales y cognitivas se basan en el uso de modelos abstractos que son independientes del contexto, mientras que las contextuales insisten en que el contexto es más importante que cualquier otra cosa y complementan o refuerzan la construcción de modelos más generales sobre el fenómeno observado, sin embargo las ventajas de estos dos tipos de técnicas es que pueden ser complementarias.

2.1.3.3. Clasificación del SWEBOK

2.1.3.3.1. Entrevistas

Es un enfoque tradicional para elicitación de requisitos, sin embargo hay que considerar y entender las ventajas y limitaciones de éstas y cómo deben ser llevadas a cabo.

2.1.3.3.2. Escenarios

Es considerado como un medio valioso para proporcionar contexto a la elicitación de requisitos. Así mismo, proporcionan al ingeniero de requisitos un marco para las preguntas sobre las tareas del usuario por ejemplo: ¿qué pasaría si? y ¿cómo se hace?. El tipo más común de escenario es el caso de uso.

2.1.3.3.3. Reuniones

El propósito de esta técnica es tratar de llevar a cabo un efecto unificador en un grupo de *stakeholders*, lo que puede incentivar el surgimiento de más ideas sobre los requisitos software que

si se hiciera de forma individual. Puede utilizarse la lluvia de ideas para que éstas surjan y posteriormente sean refinadas, lo que no puede hacerse usando entrevistas. Permite también, reconocer tempranamente los requisitos conflictivos y dónde es originado el conflicto. Cuando esta técnica trabaja bien, puede generar en un conjunto de requisitos muy consistentes, sin embargo debe ser manejada cuidadosamente para prevenir una situación donde un grupo de *stakeholders* actúe en detrimento de otros.

2.1.3.3.4. Observación o etnografía

La importancia del contexto del software dentro del ambiente organizacional ha permitido la adaptación de técnicas observacionales para elicitar requisitos. Los ingenieros de software aprenden acerca de las tareas del usuario por medio de la inmersión en el ambiente y observando cómo los *stakeholders* interactúan con el software y con cada uno de ellos. Estas técnicas son relativamente caras, pero son instructivas porque ilustran muchas tareas de los usuarios y procesos de negocios que son delicados y complejos para que sus actores puedan describirlos fácilmente.

2.1.3.4. Clasificación de Hickey-Davis

En el estudio publicado por Hickey y Davis (2003) se mencionan las siguientes técnicas de elicitación:

2.1.3.4.1. Sesiones Colaborativas

Las sesiones colaborativas son necesarias para elicitar requisitos de un proyecto software donde no importa la localización geográfica de los *stakeholders* (ya que ahora existen herramientas que facilitan esta actividad) y útiles para crear sistemas innovadores. Estas técnicas (JAD, lluvia de ideas, sesiones de grupo, entre otras) son consideradas como un enfoque estándar para elicitar requisitos.

2.1.3.4.2. Entrevistas

Esta técnica es ampliamente utilizada, principalmente para reunir información inicial cuando se trabaja en nuevos proyectos o nuevos dominios, y para evitar conflictos dentro del grupo de *stakeholders*. Las entrevistas con expertos en la materia son esenciales cuando los usuarios y clientes son inaccesibles.

2.1.3.4.3. Etnografía

Son consideradas por los ingenieros de requisitos como una técnica extremadamente eficaz. Esta técnica se debe utilizar cuando los *stakeholders* están disponibles y hay un sistema existente. Pero es también considerada para reunir los requisitos cuando los usuarios están demasiado ocupados para participar en entrevistas, sesiones de grupo, o cuestionarios. Con esta técnica se pueden analizar las relaciones políticas y de poder cuando se trabaja en nuevas organizaciones.

2.1.3.4.4. Lista de tópicos o temas

Se crea una lista con los tópicos más importantes con el fin de que éstos no sean olvidados y que puedan estar disponibles para su uso posterior. Independientemente de la técnica utilizada, nuevos temas pueden ser añadidos en la lista conforme van apareciendo. Esta técnica permite que el equipo siempre esté enfocado en todos los tópicos y que no solamente siga uno en particular.

2.1.3.4.5. Modelos

Ésta técnica desempeña un papel fundamental dentro de la elicitación de requisitos, ya que ayuda a entender mejor a los *stakeholders*. Es vista por los ingenieros de requisitos como un medio para facilitar la comunicación, descubrir y organizar la información recopilada con otras técnicas de elicitación, y descubrir inconsistencias.

2.1.3.4.6. Cuestionarios

Los cuestionarios se pueden aplicar a un *stakeholder* o a un grupo de estos. Un cuestionario es básicamente una lista de preguntas. Una de las limitaciones de esta técnica es que las preguntas deben ser muy concretas. Algunos estudios de mercado han demostrado que son muy útiles para entender las necesidades de los clientes externos a la organización.

2.1.3.4.7. Obtención de datos a partir de software existente

Esta técnica es utilizada cuando los requisitos son obtenidos para un software que reemplazará a uno ya existente. El software se analiza para obtener las funcionalidades con las que cuenta, sin embargo hay que tener en cuenta que no se debe analizar demasiado ya que el software nuevo podría tener las mismas limitaciones que el anterior. Es posible hacer una analogía de esta técnica con una excavación arqueológica en la que se descubren y analizan los objetos de interés, para de esta manera saber cómo han sido formados. También se pueden organizar reuniones con los usuarios finales del software actual para hacer un análisis.

2.1.3.4.8. Categorización de requisitos

Algunos de los expertos en el área usan la plantilla de Volere¹⁰ propuesta por Robertson y Robertson (Robertson & Robertson, 2006) para asegurarse que todas las categorías de requisitos sean obtenidas. Otros manifiestan opiniones sobre otras dimensiones para la categorización, por ejemplo, clasifican los requisitos como esenciales, esperados y asombrosos (*gee-whiz*). Otros expertos ven a la gestión como una amenaza para el éxito y por lo tanto prefieren etiquetar requisitos a los que llaman "fantasías de gestión." Otros más sienten que es esencial para filtrar las declaraciones recibidas de todos los *stakeholders*, lo que resulta en una lista que contiene únicamente las necesidades. Y a otros les gusta hacer explícita la frontera del software desde principios de la elicitación para que todos los *stakeholders* conozcan lo que está dentro de alcance y lo que no.

2.1.3.4.9. Conocimiento del conflicto y de la resolución

Cuando se tiene un gran número de *stakeholders* involucrados en un proyecto, hay más posibilidad de que se puedan generar conflictos. Si los conflictos surgen en una clase determinada de *stakeholders*, se recomienda que un integrante de esta clase se haga cargo de solucionarlo. Se sugiere también que antes de hacer la elicitación se entienda la jerarquía de la organización, su estructura de poder y lo referente a la política y su campo.

¹⁰ La plantilla de Volere abarca el descubrimiento, la comunicación y la gestión de los requisitos de cualquier tipo de software, hardware, productos, servicios, organización, o cualquier otra cosa que se esté desarrollando. La plantilla de Volere trabaja con cualquier ciclo de vida: secuencial, ágil, outsourcing, etc. y con la mayoría de las herramientas de los requisitos y técnicas de modelado. Las técnicas de requisitos que abarca Volere se basan en la experiencia de análisis de negocios en proyectos a nivel mundial (Robertson & Robertson, 2006).

2.1.3.4.10. Prototipos

En el estudio elaborado por Hickey y Davis (2003) la mayoría de ingenieros de requisitos no consideraron los prototipos como una técnica de elicitación. Los ingenieros de requisitos que sí la mencionaron sugieren utilizarla solamente cuando el desarrollo de los prototipos sea rápido o cuando existe confianza mutua entre los *stakeholders* y el equipo de desarrollo.

2.1.3.4.11. Juegos de roles

Cuando los *stakeholders* son inaccesibles, los integrantes del equipo de desarrollo u otras personas que no son *stakeholders* juegan el rol de los *stakeholders* faltantes o inaccesibles.

2.1.3.4.12. Métodos formales

Los métodos formales son técnicas basadas en las matemáticas. Con estas técnicas es posible especificar, diseñar, implementar y verificar tanto hardware como software. A menudo son apoyados por herramientas que pueden ofrecer una manera rigurosa y eficaz de modelar, diseñar y analizar sistemas (Abramson & Pike, 2011). El rigor matemático de los métodos formales permite a los usuarios analizar y verificar los modelos en cualquier parte del proceso software como la IR.

Los métodos formales pueden ser útiles en la obtención, la articulación, y la representación de los requisitos y sus herramientas pueden proporcionar el soporte necesario para cerciorarse de la integridad, trazabilidad, verificabilidad, reutilización, y evolución de los requisitos (Woodcock, Larsen, Bicarregui & Fitzgerald, 2009). Sin embargo los métodos formales no son muy utilizados en la etapa de elicitación de requisitos, incluso para software donde la seguridad es crítica, ya que los expertos opinan que distancian a los *stakeholders* del proceso.

2.1.3.4.13. Programación extrema

La programación extrema no exige mucho trabajo en lo que se refiere a la elicitación, ni tampoco asegura que se haga por adelantado, ya que en esta técnica los demás *stakeholders* se reemplazan por un cliente omnipresente colocado en el equipo de desarrollo, el cuál responderá las preguntas que surjan durante el desarrollo. Se recomienda que esta técnica sea usada cuando el dominio es sometido a grandes y constantes flujos de trabajo.

2.1.3.5. Otras técnicas

2.1.3.5.1. Enfoque colaborativo

Esta técnica se basa en la utilización de una herramienta como las *wikis*¹¹ o redes sociales para la elicitación de requisitos.

Resumiendo lo anterior, la elicitación de requisitos es una fase en la cuál se provee al *stakeholder* una oportunidad de expresar sus deseos y de explicar sus metas y procesos de negocios, existen muchas técnicas para llevar a cabo la extracción del conocimiento pero cada una de ellas tiene sus debilidades y fortalezas dependiendo del dominio en el cuál serán aplicadas. Por lo que el desempeño de estas técnicas depende en gran medida de la selección apropiada que lleve a cabo el ingeniero de requisitos y de las características propias del problema a resolver.

¹¹ Las *wikis* son herramientas que proporcionan una plataforma flexible para la colaboración asincrónica con el objetivo de crear contenido y son especialmente útiles en los proyectos distribuidos (Louridas, 2006).

Además, los desarrolladores necesitan entender el contexto de sus proyectos de software para asegurar que los métodos que utilizan sean los correctos, lo que por lo regular conduce a que se involucren en la adaptación de actividades para mejorar el desarrollo software (Young, McDonald, Edwards & Thompson, 2001).

2.2. Trabajo relacionado

Revisando la literatura pertinente, se encontró el trabajo realizado por Andriano (Andriano, 2006) en la Universidad Nacional de la Plata, Argentina en Agosto del 2006 titulado “*Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación*”. El objetivo de este trabajo fue analizar comparativamente el proceso de elicitación y las técnicas utilizadas en la obtención de requisitos de dos tipos de software que se desarrollan en Argentina: a medida y empaquetado. Los resultados obtenidos a través de encuestas que fueron aplicadas a desarrolladores y usuarios de software argentinos fueron utilizados para elaborar una guía para el proceso de elicitación de los requisitos de los dos tipos de software analizados, y proponer un conjunto de métricas que pueden ser utilizadas en este proceso.

Las métricas que se proponen en este trabajo se clasifican como métricas directas e indirectas, y cada categoría a su vez se clasifica de acuerdo al tipo de software al que se va a aplicar. Las métricas directas miden el objeto que se desea cuantificar o algún atributo de éste y las métricas indirectas son aquellas que miden otros atributos u otros objetos que no estén relacionados directamente con el que se desee cuantificar.

Andriano (2006) observó que el proceso de elicitación esta condicionado al tipo de software que se va a desarrollar (software a medida o empaquetado) y que el uso de métricas resulta beneficioso para el éxito de esta etapa, sin embargo la validación de las métricas propuestas se realizó de manera subjetiva a través de entrevistas a los desarrolladores.

En esta propuesta se cubren solamente algunos aspectos de la elicitación de requisitos, ya que la autora realizó una comparación de las técnicas de elicitación sin analizar su efectividad (a pesar de que este aspecto es crucial en la obtención de requisitos de calidad). Además, no se definieron las variables a analizar en los datos obtenidos a partir de las encuestas, por lo que no es posible replicar el estudio bajo otras condiciones.

A diferencia del trabajo relacionado analizado, la propuesta que se presenta en este documento de tesis se centra en tres aspectos principales:

1. Determinar y proporcionar evidencia científica sobre las prácticas efectivas recomendadas para usar las técnicas de elicitación.
2. Una vez que esas prácticas efectivas sean obtenidas, es necesario determinar qué aspectos influyen en la realización correcta de la elicitación de requisitos de calidad; y
3. Con los datos obtenidos de los aspectos anteriores, se deben proponer métricas que permitan mejorar esta etapa de la IR.

Como se ha visto a lo largo de este Capítulo 2, la elicitación involucra actividades tales como la identificación de *stakeholders*, la obtención de la lista de deseos y necesidades de cada uno de ellos y por último la integración de toda la información obtenida; sin embargo al llevar a cabo cada una de estas actividades se pueden presentar diversos problemas que fueron mencionados en la

sección 2.1.2. Para tratar estos problemas o aminorarlos se han expuesto también las diversas técnicas de elicitación (sección 2.1.3) que han identificado diferentes autores.

En consecuencia, la metodología utilizada para desarrollar este trabajo de tesis será la aplicación de una revisión sistemática de literatura con el fin de reunir y evaluar la evidencia disponible para detectar los aspectos que influyen en la efectividad en la elicitación de requisitos y cómo pueden ser medidos y controlados por los ingenieros de software a través del establecimiento de métricas.

3. Solución propuesta

3.1. Revisión sistemática de literatura

Debido al impacto que la IR tiene en la calidad del software (particularmente en la elicitación de requisitos), se hace necesario conducir una revisión sistemática de literatura para descubrir qué técnicas aplicadas en esta etapa son efectivas, qué aspectos influyen en su efectividad (para tomarlos en cuenta) y así, proponer métricas que permitan medir su efectividad.

Como consecuencia del creciente número de estudios empíricos y teóricos sobre la IR, es pertinente aplicar un enfoque sistemático para evaluar y uniformar los resultados de la investigación con el fin de proporcionar un resumen equilibrado y objetivo de la evidencia obtenida. Por lo tanto se propone la aplicación de la SLR, específicamente en el área de elicitación de requisitos para permitir, basándose en la evidencia científica obtenida, el proponer métricas para que los ingenieros de software entiendan, controlen y mejoren esta etapa con el fin de producir software de calidad (Mitchell et al., 1997).

Los trabajos de investigación (sintetizados) en una SLR son referenciados como estudios primarios, mientras que la revisión en sí misma es un estudio secundario. La acumulación de evidencia a través de estudios secundarios puede ser valiosa para ofrecer nuevos descubrimientos o para identificar aspectos que pueden ser clarificados por estudios primarios adicionales.

Una SLR tiene como objetivo presentar una evaluación de la literatura relativa al tema de investigación usando la síntesis de una metodología rigurosa (Beechman et al., 2007).

En este Capítulo de la tesis se seguirán las directrices derivadas de las utilizadas en la investigación médica, adaptadas y aplicadas por Kitchenham a la IS (Kitchenham, 2004; Kitchenham et al., 2004) para mostrar cuáles son las técnicas más utilizadas y efectivas en la elicitación de requisitos. De acuerdo con Kitchenham (2004) y Kitchenham et al. (2004), se deben seguir los siguientes pasos:

- Identificación de la necesidad de la revisión literaria.
- Formulación de preguntas de investigación.
- Establecimiento de términos de búsqueda.
- Definición de la estrategia de búsqueda.
- Selección de estudios.
 - Establecimiento de los criterios de selección.
 - Definición del procedimiento de selección.

- Evaluación de la calidad de los estudios seleccionados.
- Definición de la estrategia de extracción de datos.
- Síntesis de los datos extraídos.

Cada uno de estos pasos es descrito en las siguientes subsecciones de este Capítulo.

3.1.1. Identificación de la necesidad de la revisión literaria

Como se ha mencionado en los capítulos anteriores, la importancia de la IR en el proceso software radica en que los atributos de calidad de los requisitos (i.e. que sean correctos, completos, consistentes, no ambiguos, verificables, validables, etc.) son cruciales para obtener un producto software de calidad. Dentro de este proceso, la elicitación de requisitos es una etapa crucial, ya que es aquí donde se conceptualiza el software en función de lo que realmente necesitan los *stakeholders* y no lo que desean. Para esto se requiere también del conocimiento del dominio de la aplicación, del contexto y de los problemas que el software solucionará (Christel & Kang, 1992). Sin embargo, para extraer este conocimiento de los *stakeholders* es necesario aplicar ciertas técnicas de elicitación.

Las técnicas de elicitación son empleadas para obtener la información necesaria para desarrollar un software de calidad, ya que en este proceso se determinan las necesidades de los *stakeholders* y posteriormente éstas se comunican a los desarrolladores. Aunque existen muchas técnicas para llevar a cabo este proceso, no significa que se deban aplicar todas ó que se tenga que seleccionar una sola de este conjunto, ya que no es así de fácil su selección. Si no existe un proceso maduro para la selección de las técnicas de elicitación a utilizar, resulta difícil seleccionar aquella que se adecue más al proyecto. Como consecuencia de esto, el proceso de elicitación muchas veces se hace de forma incorrecta, ya que aunque se identifiquen correctamente a los *stakeholders*, no se puede garantizar que la información extraída en esta etapa sea útil, correcta y completa.

En estudios empíricos como (Hickey & Davis, 2003; Zowghi & Coulin , 2005; Zheyang, 2007) se sugiere que la selección de estas técnicas se debe hacer tomando en cuenta el tipo de software que se va a desarrollar, el estado del proyecto, el dominio de aplicación, la consideración de algunas anomalías que se puedan presentar, el análisis de las ventajas de cada técnica y el tipo de información que éstas pueden proporcionar. Hickey y Davis (Hickey & Davis, 2003) afirman que la selección de la técnica de elicitación debe hacerse en base a la experiencia del ingeniero de requisitos, además sugieren la utilización de modelos para mejorar la comprensión de la situación y la comunicación entre los *stakeholders*. Sin embargo, la literatura referente indica que se deben elegir aquellas técnicas que se ajusten mejor al propósito del proyecto y que se debe tomar en cuenta que cada técnica sirve para obtener diferentes tipos de información por lo que es recomendable escoger varias (Zheyang, 2007).

Cada técnica de elicitación puede proveer cierto tipo de conocimiento y cada una de ellas posee características que las hacen efectivas en ciertos casos (Zheyang, 2007; Zowghi & Coulin, 2005). Por lo tanto, es necesario analizar la efectividad de cada una de ellas para identificar que aspectos influyen en éstas. Empero, a pesar de que actualmente existe una gran variedad de técnicas de elicitación (y siguen surgiendo más), solamente existen algunas métricas que permiten medir su efectividad en base a la calidad de los requisitos obtenidos. Es por esta razón que se hace necesario identificar las técnicas de elicitación maduras (en términos de antigüedad) que hayan demostrado ser efectivas para así obtener información sobre qué aspectos influyen en su capacidad para obtener requisitos de calidad.

Por lo que se hace necesario aplicar una SLR para obtener evidencia científica de los estudios sobre las técnicas utilizadas en la etapa de elicitación de requisitos, así como sobre su efectividad. Además puede ayudar a encontrar información concerniente a cuestiones o factores que puedan mejorar el proceso de elicitación de requisitos y por ende, mejorar el proceso de IR.

3.1.2. Formulación de preguntas de investigación

El enfoque sistemático utilizado para analizar los estudios publicados permitirá identificar la literatura pertinente en donde se presentan diferentes técnicas desarrolladas para la elicitación de requisitos.

Esta evidencia se analizará para conocer qué aspectos o factores influyen en la efectividad de las técnicas de elicitación y así, identificar las métricas que se pueden utilizar para medir la calidad de los requisitos en esta etapa. En este sentido, se hará una búsqueda sistemática de la literatura con el propósito de dar respuesta a las siguientes Preguntas de Investigación (PI) que se han definido en base a los objetivos específicos de este trabajo de tesis.

- **Pregunta de investigación 1 (PI1):** ¿Qué técnicas o métodos maduros son utilizados actualmente en la etapa de elicitación de requisitos del software?
- **Pregunta de investigación 2 (PI2):** ¿Qué técnicas o métodos de elicitación han demostrado mayor efectividad?
- **Pregunta de investigación 3 (PI3):** ¿Qué aspectos de cada técnica de elicitación influyen en su efectividad?
- **Pregunta de investigación 4 (PI4):** ¿Qué métricas se utilizan para medir los requisitos software en la etapa de elicitación?

La Figura 3.1 muestra una visión de la relación entre las cuatro preguntas de investigación con el objetivo de dar una vista más general y comprensiva de la investigación.

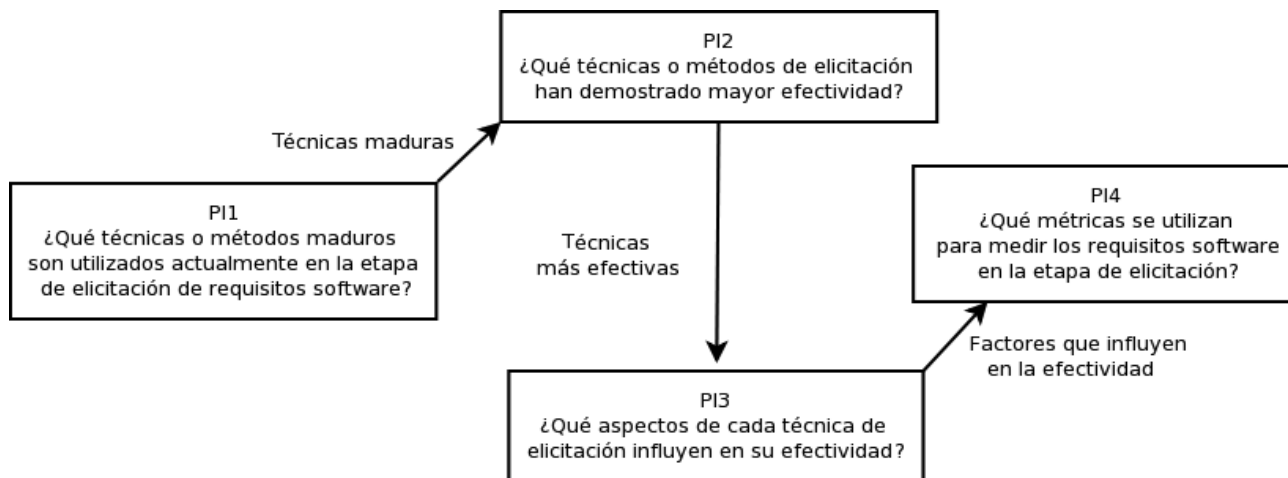


Figura 3.1. Relación entre las cuatro preguntas de investigación.

3.1.3. Términos de búsqueda

De las preguntas que han sido formuladas en la sección anterior se derivan las palabras claves: “*requirements elicitation*”, “*techniques*” y “*effectiveness*”. Las cadenas de búsqueda son construidas usando términos relevantes basados en las preguntas de investigación. Así mismo, se obtiene una lista de sinónimos correspondientes a cada una de las palabras anteriores. Un ejemplo de búsqueda para la PI2 quedaría de la siguiente forma:

Keywords ((elicitacion OR obtaining* OR gaining* OR extracting* OR acquisition* OR discovery* OR capture* OR gathering*) AND requirements AND (techniques* OR methods*) AND (effectiveness* OR effectively* OR effectualness* OR effectuality* OR efficient* OR efficacious* OR efficacy* OR efficiency*))*

Los términos de búsqueda fueron expandidos usando Word Net versión 3.0 (Princeton University, 2013) y el diccionario Soule de sinónimos en inglés. La lista de términos ha sido adaptada para ajustarse a cada una de las cuatro preguntas de búsqueda.

3.1.4. Estrategia de búsqueda

El proceso de SLR recomienda la búsqueda en diversas fuentes electrónicas (Kitchenham, 2004; Kitchenham & Charters, 2007), por lo tanto para realizar la búsqueda se utilizaron las siguientes bases de datos:

- ACM Digital Library
- IEEE Xplore
- Springer Verlag
- Google Scholar
- ScienceDirect
- Metapress
- Wiley InterScience

Con el fin determinar si se han realizado trabajos similares y localizar estudios potencialmente relevantes, la estrategia de búsqueda de esta revisión se dirige principalmente hacia el descubrimiento de artículos publicados (revistas, reportes técnicos o libros) en algunas de las siete bases de datos mencionadas anteriormente, y su corroboración por otras referencias relevantes. En este sentido se realizan búsquedas de prueba usando un determinado número de cadenas de búsqueda formadas por la combinación de las palabras clave y sinónimos antes mencionados.

3.1.5. Selección de estudios

La selección de material para la SLR se basa en los siguientes criterios y procedimiento, respectivamente.

3.1.5.1. Establecimiento de los criterios de selección

El principal criterio para incluir un artículo como un estudio primario es la presentación de datos empíricos o prácticos sobre la efectividad de las técnicas utilizadas en la etapa de elicitación de requisitos. Todo el material que se usará en la SLR será seleccionado en base a los siguientes criterios:

- Que responda directamente a una o más de las preguntas de investigación o sus sinónimos (véase sección 3.1.2).

- Que haya sido publicado entre los años : 1993¹² – 2013.
- Que esté relacionado con alguna técnica de elicitación de requisitos.
- Que contenga datos de la capacidad de la técnica de elicitación para lograr el efecto que se espera sobre los requisitos (calidad).

Debido a que el objetivo de esta revisión son las técnicas de elicitación y su efectividad, se excluyen los siguientes textos:

- Presentaciones en diapositivas.
- Talleres (*workshops*).
- Opiniones, puntos de vista, o anécdotas.
- Herramientas que no provean evaluaciones empíricas de sus aplicaciones.
- Que no pertenezcan al ámbito de la IS.

Estos criterios serán aplicados a estudios tanto del ámbito industrial como del académico.

3.1.5.2. Definición del procedimiento de selección

La selección preliminar de los estudios primarios se basa inicialmente en la revisión del título, resumen y palabras clave, aunque esta búsqueda probablemente se amplíe para incluir la sección de conclusiones en los casos donde el título, palabras clave y resumen no provean suficiente información. Después, todas las fuentes seleccionadas serán analizadas contra el conjunto detallado de criterios de inclusión para obtener los estudios primarios.

Con el fin de evitar alguna duplicación de estudios se examinan todos para descubrir publicaciones repetidas, por ejemplo, si un estudio similar fuera divulgado en diferentes publicaciones, incluso con diferentes autores, solamente el estudio más reciente y amplio es incluido en la revisión (si dos estudios tuviesen las mismas fechas de publicación y extensión en cuanto a la cobertura del tema, solamente se incluye un estudio).

3.1.6. Evaluación de la calidad de los estudios seleccionados.

La evaluación de la calidad de los estudios que han sido seleccionados puede ser usada para guiar la interpretación de la síntesis de los hallazgos encontrados y determinar la fuerza de las inferencias elaboradas (Kitchenham & Charters, 2007). La calidad de cada estudio aceptado fue evaluada de acuerdo al criterio mostrado en la Tabla 2.

El primer criterio permite asegurar que los autores del estudio alcanzan con claridad los fines y objetivos de la investigación realizada. Esta pregunta puede ser respondida positivamente para todas las publicaciones revisadas. El segundo criterio permite cuestionar si el estudio provee suficiente información (ya sea directamente o haciendo referencia a otra literatura relevante) para dar a esta investigación el contexto y las bases apropiadas. Para casi todas las publicaciones (95%) ésta fue respondida positivamente. La última pregunta permite asegurar si el resultado de la investigación fue suficiente para cubrir el propósito de esta investigación. Esta pregunta fue respondida positivamente en casi todas las publicaciones (97%). Las medidas de evaluación fueron establecidas por dos investigadores experimentados en IS de la Universidad Tecnológica de la Mixteca y un investigador independiente. Sin embargo, la puntuación es sólo una heurística para ser

¹² Se eligió el año de 1993 porque fue el año en el que se llevó a cabo el primer simposio de IR.

utilizada como una guía y ningún estudio fue rechazado tomando como base esta puntuación de calidad. Se normalizaron los datos de 158 estudios, combinando el porcentaje obtenido por el criterio de calidad (véase Tabla 3).

Tabla 2. Evaluación de calidad.

Criterio de evaluación	Puntuación Opciones de respuesta para la puntuación (campo en Endnote)
¿Se explica adecuadamente el objetivo de la investigación?	Si = 1/moderadamente = 0.5/no = 0.
¿Se explica claramente el enfoque presentado?	Si = 1/moderadamente = 0.5/no = 0.
¿Para un estudio, cuál es el porcentaje de calidad aceptado basándose en los hallazgos?	No hallazgos = 0/Arriba del 80% = 1 /Abajo del 20% = 0/ Entre = 0.5.
%	Introducir el campo de porcentaje de evaluación de calidad en Endnote.

Tabla 3. Puntuación de calidad de los estudios aceptados para las primeras dos preguntas de investigación.

	Calidad (puntuación)					Total
	Pobre (<26%)	Razonable (26-45%)	Bueno (46-65%)	Muy bueno (66-85%)	Excelente (>86%)	
Número de estudios	19	20	36	51	32	158
Porcentaje de estudios	~12.02%	~12.65%	~22.78%	~32.27%	~20.25%	100%

3.1.7. Definición de la estrategia de extracción de datos

El proceso de extracción de datos se llevó a cabo utilizando Endnote versión X6 para documentar las referencias para cada estudio. De acuerdo con Beechman et al. (2007), cada estudio usado para responder las preguntas de investigación se debe registrar en un formulario de resultados por separado, con el propósito de identificar los tópicos que emanan de los hallazgos reportados en cada estudio aceptado. Para este caso, estos tópicos identificados dan las categorías reportadas en la sección de resultados.

3.1.8. Síntesis de los datos extraídos

Las directrices de Kitchenham no son completamente claras acerca de la naturaleza del proceso de extracción de datos – cuánta categorización es hecha durante la extracción de datos y cuánta es hecha durante el paso de síntesis de datos. Por lo que se ha optado por una extracción de datos trivial, dando como resultado una lista de referencias que fueron mínimamente parafraseadas. Esta categorización se hizo en las partes tempranas de la etapa de síntesis.

En la sección 3.2 se presentan las frecuencias del número de veces que cada tópico es identificado en diferentes fuentes; cada ocurrencia recibe el mismo peso. Las frecuencias simplemente reflejan cuántas veces se identifica el tópico en diferentes estudios, no lo importante que puede ser.

3.1.8.1. Recuperación de documentos

Las búsquedas permitieron obtener más de 243 referencias. Analizando el título y el resumen, se pudieron rechazar aproximadamente 60 de éstas (véase sección 3.1.5.2). Por lo que se puede decir que el número de falsos positivos en el conjunto inicial (estudios que pudieron haber sido relevantes pero en una detallada investigación resultaron no serlo) no fue tan alto. Entonces, inspeccionando 183 estudios en total, se estableció una lista final de 158 estudios. Todos los pasos involucrados en el proceso de selección son mostrados en la Tabla 4.

Los ejercicios de validación incluyen:

- *Acuerdo entre los evaluadores:* De acuerdo al método de Fleiss' Kappa (Fleiss, 1971) el acuerdo interno denota de este modo la consistencia de extracción de datos entre los estudios investigados cuando dos o más investigadores evalúan cada estudio. Así, se realizaron pruebas de inter-evaluación de acuerdo en los 178 estudios encontrados en la búsqueda preliminar. El grupo investigador primario, conformado por expertos de la Universidad Tecnológica de la Mixteca analizó meticulosamente cada uno de estos estudios. Estos investigadores aceptaron 158 estudios, mientras que el investigador independiente analizó aleatoriamente 36 estudios elegidos entre los rechazados y aceptados (aproximadamente uno de cada cuatro estudios de una lista alfabética de 178) de estos un 91% de conformidad fue registrado con las evaluaciones originales. Este porcentaje de coincidencia brinda certeza en las decisiones de aceptación y rechazo.
- *Evaluación independiente:* Se condujo este ejercicio de validación en los 158 estudios aceptados. En este ejercicio se tuvo un alto porcentaje de acuerdo entre los investigadores primarios y los expertos independientes (94.3%). Hubo desacuerdo en 7 de los estudios aceptados, así que se solicitó la opinión de un segundo investigador independiente, quien estuvo de acuerdo en rechazar los 7 estudios después de tomar en cuenta cómo respondieron las preguntas de investigación. Como resultado de este ejercicio 158 estudios fueron aceptados para su inclusión.

Tabla 4. Estudios validados y revisados.

Proceso de selección	# de estudios	Estudios utilizados en la validación
Estudios extraídos de bases electrónicas.	243	n/a
Examinados basándose en el título y resumen.	183	n/a
Estudios- versiones completas disponibles.	243	-
Estudios aceptados (por investigadores primarios).	178	-
Estudios rechazados por el investigador independiente.	158	20 estudios rechazados de los 36 seleccionados aleatoriamente.
Estudios agregados por el investigador independiente.	165	7 estudios agregados de los 36 seleccionados aleatoriamente.
Estudios rechazados en validación 2.	158	7 rechazados.

3.2. Resultados

Un total de 158 estudios versan sobre las técnicas de elicitación maduras que son utilizadas actualmente en la etapa de elicitación de requisitos, algunos indican cuáles son más efectivas que otras en ciertas situaciones, otros destacan los aspectos que las hacen poseer estas ventajas y un número menor indica que medidas o métricas se pueden realizar durante la etapa de elicitación. Las referencias para los 158 estudios incluidos en esta sección se encuentran ordenadas de forma alfabética en la bibliografía de lecturas adicionales. Es muy importante mencionar que debido a las fuentes no estructuradas y al alto nivel de heterogeneidad de los estudios analizados, las técnicas de meta-análisis tales como el agregado de datos no pudieron ser aplicados (Kitchenham, 2004; Kitchenham & Charters, 2007).

También es muy importante mencionar que aproximadamente el 35% de los estudios analizados no fueron validados, por lo que no se pudo analizar su impacto en la elicitación de requisitos.

A continuación, antes de presentar los resultados y análisis para cada pregunta de investigación se da una breve visión de las características generales de los estudios.

3.2.1. Visión general de los estudios

3.2.1.1. Método de investigación

Las publicaciones inspeccionadas fueron clasificadas directamente de acuerdo al método de búsqueda aplicado. La estrategia inicial de categorización es simple: mostrar el método de búsqueda sin interpretar el contenido del estudio. Partiendo de este punto, se definieron las siguientes categorías para clasificar los estudios:

- Empírica, los hallazgos están basados en evidencia directa o experimentos.
- Estudios conceptuales o teóricos basados en el entendimiento del campo de experiencia y referencia a otro trabajo relacionado.
- Otros.

La Figura 3.2 muestra que de esos 158 estudios, el 89% son empíricos, 7% son teóricos y un pequeño número de estudios (4%) son validados de otra forma, donde el trabajo empírico es nuevamente examinado.

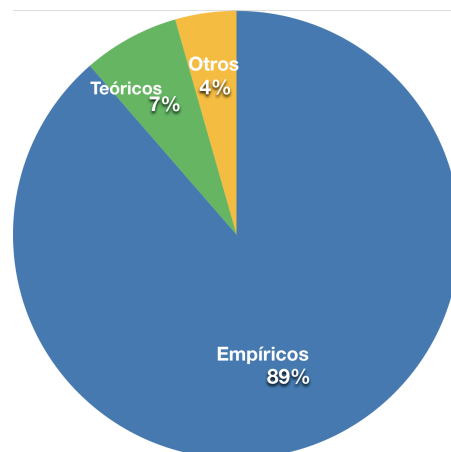


Figura 3.2. Métodos de investigación en los estudios aceptados.

3.2.1.2. Año de publicación

Primero se hizo una revisión de los artículos publicados entre 1993 y 2013 para determinar las técnicas de elicitación que de acuerdo a Christel y Kang (1992) son maduras¹³. La Tabla 5 muestra el año de utilización, dentro del contexto de la elicitación de requisitos, para de cada una de las técnicas analizadas.

Tabla 5. Estudios que muestran los primeros años en los que se aplicaron las técnicas de elicitación.

Técnica	Año de aplicación	Estudios referenciados
Entrevistas.	1980	(Christel & Kang, 1992)
<i>Workshops.</i>	1980	(Christel & Kang, 1992)
<i>Focus group.</i>	1980	(Christel & Kang, 1992)
JAD.	1970	(Christel & Kang, 1992)
QFD.	1992	(Christel & Kang, 1992)
Etnografía.	1980	(Christel & Kang, 1992)
Escenarios.	1990	(Holbrook,1990)
Prototipos.	1980	(Christel & Kang, 1992)
Análisis de protocolo.	2000	(Nuseibeh, & Easterbrook, 2000)
Clasificación de tarjetas.	2000	(Nuseibeh, & Easterbrook, 2000)
Ontología.	1996	(Lin, Fox, & Bilgic 1996)
Modelado.	2000	(Nuseibeh, & Easterbrook, 2000)
I*.	2000	(Nuseibeh, & Easterbrook, 2000)
Goal.	2000	(Nuseibeh, & Easterbrook, 2000)
Casos de uso	1998	(Nuseibeh, & Easterbrook, 2000)
Redes sociales	2007	(Sofia Pereira & Soares, 2007)

En base a estos datos se determinaron las técnicas maduras. Ahora bien, para determinar qué técnicas se utilizan actualmente se determinó un periodo de 6 años, quedando el rango del 2007-2013 como el periodo de tiempo donde se haría la búsqueda sistemática de la literatura. El resultado de la búsqueda se muestra en la Figura 3.3 donde se observa el número de estudios que respondieron a las cuatro preguntas de investigación y el año en que fueron publicados. Como se aprecia en la Figura 3.3, en los primeros 10 años existen pocos estudios sin embargo a partir de 2009 y durante los próximos 4 años se obtuvo mayor evidencia de la aplicabilidad de las técnicas de elicitación así como de propuestas de métricas para la etapa de elicitación.

¹³ Los criterios para decir que una técnica es madura son: que al menos ésta haya sido probada, y que se haya utilizado en la elicitación de requisitos por una década, esto quiere decir, que se debe tener un registro histórico donde se muestre que ha sido aplicada para elicitar requisitos (Christel & Kang, 1992).

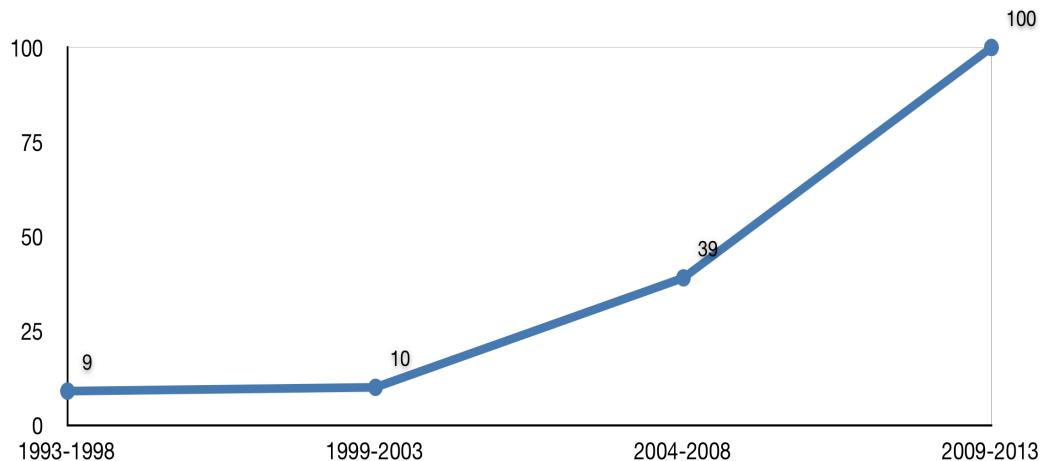


Figura 3.3. Número de estudios por año que muestran los estudios que dan respuesta a las cuatro PI.

3.2.2. Métricas para los requisitos software en la etapa de elicitación

Al contestar las cuatro preguntas de investigación, se espera obtener una imagen amplia de lo que la literatura reporta sobre las métricas que existen para los requisitos software en la etapa de elicitación. Se recolectó información acerca de que métodos o técnicas maduras son actualmente usadas para elicitar requisitos (PI1); cuáles de éstas técnicas han demostrado mayor efectividad (PI2); que aspectos de cada técnica influyen en su efectividad (PI3); y que métricas existen para los requisitos software en la etapa de elicitación (PI4).

3.2.2.1. PI1- técnicas o métodos maduros utilizadas actualmente en la etapa de elicitación de requisitos

89 estudios fueron identificados para dar respuesta a la *Pregunta de Investigación 1 (PI1)*: *¿Qué técnicas o métodos maduros son utilizados actualmente en la etapa de elicitación de requisitos funcionales del software?*

Todos los estudios analizados permiten decir que la elicitación de requisitos no puede realizarse usando una sola técnica de elicitación y que dichas técnicas son seleccionadas dependiendo de las características del proyecto.

De acuerdo al *framework* de Nuseibeh y Easterbrook (Nuseibeh, & Easterbrook, 2000) mostrado en la sección 2.1.3.2 y a la clasificación de Hickey-Davis (Hickey & Davis, 2003) descrita en la sección 2.1.3.4, se hace la agrupación de los estudios obtenidos de las técnicas de elicitación de requisitos en las siguientes clases :

3.2.2.1.1. Técnicas tradicionales

Dentro de este grupo se encuentran las técnicas tales como entrevistas, cuestionarios, encuestas, escenarios y casos de uso. En base a la información obtenida que se puede apreciar en la Tabla 6, se obtuvo un total de 39 estudios para este grupo, siendo la técnica de entrevistas la más utilizada con 22 estudios que hacen referencia a su aplicabilidad en la elicitación, mientras que la técnica menos utilizada es la de cuestionarios con sólo 1 estudio que hacen referencia a ésta. Otro punto a resaltar es que el número de estudios donde se utilizaron técnicas tradicionales para la elicitación de requisitos, ha ido disminuyendo.

Para visualizar la aplicabilidad de las técnicas por año se presenta la Figura 3.4. Respecto a la técnica de entrevistas es posible afirmar que es la técnica más utilizada para la obtención de requisitos software, sin embargo alcanzó su máximo en el 2008 disminuyendo para el año 2009 y aumentando nuevamente para el 2010; sin embargo, después de este último año se presenta una disminución de más del 50%. Esto se debe principalmente a que en el uso de las técnicas de elicitación, en los últimos años, se suele combinar unas con otras, y también al surgimiento de otras técnicas de elicitación como el uso de redes sociales, metodologías ágiles y ontologías¹⁴.

Tabla 6. Lista de técnicas de elicitación tradicionales.

Técnica	Estudios referenciados	Frecuencia (# de estudios)
Entrevistas.	Al Balushi, Sampaio, Dabhi y Loucopoulos (2007), Al-Salem y Abu Samaha (2007), Bee, Danilo y June (2010), Castro y Oliveira (2008), Dogan, Henshaw y Urwin (2009), Estrella et al.(2007), Kim , Park, Lee y Chung (2007), Martin et al. (2012), McFarlane y Cuthbert (2012), Mishra, Mishra y Yazici (2008), Pouyioutas, Dionysiou y Gjermundrod (2013), Reimer, Brimhall, Cao y O'Reilly (2009), Rusu, Russell y Cocco (2011), Salvador, de Oliveira Neto y Kawamoto (2008), Schneider (2007), Sommerville, Lock y Storer (2012), Taa, Abdullah y Norwawi (2010), Van Velsen, van der Geest, ter Hedde y Derks(2008, 2009), Wang et al. (2010), Yamanaka, Noguchi, Yato y Komiya (2010), Zhao, Kan y Theng (2008).	22
Escenarios.	Han, Zhou, Ma, Fang(2012), Hussein, Yu, Han y Colman (2012), Laporti, Borges y Braganholo (2009), Li-ping y Long-jun (2010), Seyff, Graf, Maiden y Grünbacher (2008), Seyff et al. (2009a), Seyff et al. (2009b), Widya et al. (2009).	8
Encuestas.	Ali et al. (2011), Bellotti et al. (2012), Guzzi, Hattori, Lanza, Pinzger y van Deursen (2011), Lázaro, Fides, Navarro y Guillén (2010), Maguire (2013).	5
Casos de uso.	Rosado, Fernández-Medina, López y Piattini (2010), Sindre (2007), Tu y Thomborson (2009).	3
Cuestionarios.	Reimer et al. (2009).	1

Analizando la técnica de cuestionarios se puede ver que el único estudio encontrado donde se utiliza como única para elicitar requisitos fue del año 2009. Otra técnica relacionada a ésta, por la lista de preguntas que se tienen que formular, es la de encuestas, que es más utilizada que los cuestionarios dado que por el avance de la tecnología se pueden elaborar encuestas en línea y de hecho la mayoría de los estudios referenciados mencionan que se utiliza este tipo de comunicación y están presentes en los últimos años que cubre la búsqueda.

En cuanto a las técnicas de escenarios y casos de uso se puede apreciar que la primera es menos utilizada que la segunda, ya que esta se utiliza principalmente para la etapa de especificación de requisitos y para la etapa de diseño de software. De acuerdo a Sommerville (2010) los casos de uso son un tipo especial de escenarios, por lo que para obtener resultados más específicos se hizo la

¹⁴ Una ontología es una descripción formal de objetos y sus propiedades, relaciones, limitaciones y comportamientos (Lin, Fox, & Bilgic 1996).

búsqueda separada de estas técnicas. De esta forma se obtuvieron ocho estudios para la técnica de escenarios en la etapa de elicitación alcanzando su máximo en el 2009 pero disminuyendo en el 2012 con sólo dos estudios, esto quizá se deba a que los escenarios han sido sustituidos por otras técnicas o es preferible usarlos para etapas posteriores a la elicitación de requisitos.

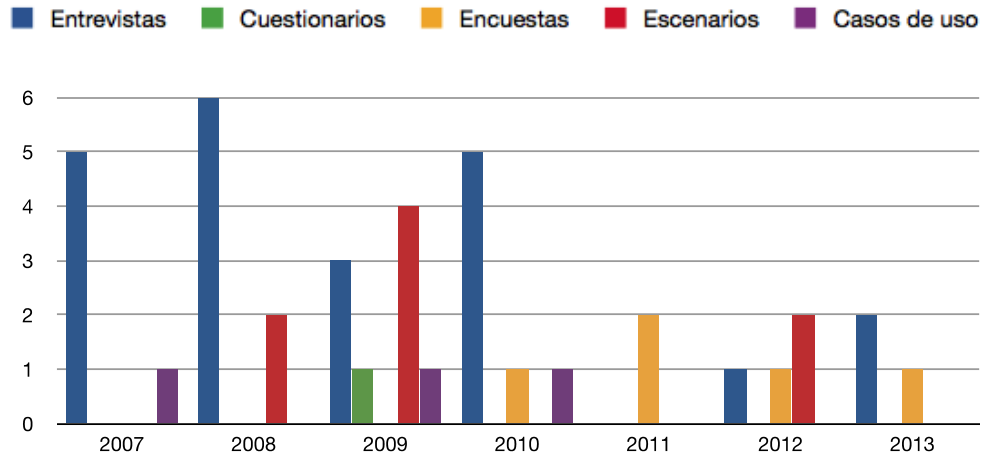


Figura 3.4. Número de estudios por técnica de elicitación tradicional y año en el que fueron publicados.

3.2.2.1.2. Técnicas grupales-colaborativas

Esta clasificación incluye técnicas grupales y colaborativas, dentro de estas últimas se incluyen aquellas que utilizan herramientas para poder facilitar la colaboración entre los *stakeholders* como el uso de *wikis*, redes sociales u otras herramientas más específicas creadas para un proyecto específico. Para esta clase se obtuvo un total de 16 estudios (véase Tabla 7).

Tabla 7. Lista de técnicas de elicitación grupales-colaborativas.

Técnica	Estudios referenciados	Frecuencia (# de estudios)
Redes sociales.	Kukreja (2012,2013), Lim, Damian y Finkelstein (2011), Lim y Finkelstein (2012), Lim, Damian, Ishikawa y Finkelstein (2013).	5
Combinación de técnicas.	Azadegan, Cheng, Niederman y Yin (2013), Laurent y Cleland-Huang (2009), Zapata, Torres, Collazos, Giraldo y Sevilla(2013).	3
Definición de requisitos orientada por puntos de vista (VORD, por sus siglas en inglés).	Aoyama, Ugai, Yamada y Obata (2007), Bendjenna, Zarour y Charrel (2008), Cheng y Liu (2008), Rui-feng, Chao, Rui-fang y Jie (2010).	4
Otra herramienta.	Fernandes et al. (2012), Hainey, Connolly, Stansfield y Boyle(2011).	2
Wikis.	Decker, Ras, Rech, Jaubert y Rieth(2007)	1
Grupos de trabajo.	Azadegan, Papamichail y Sampaio (2013).	1

Como se aprecia en la Figura 3.5, para este grupo de técnicas el número de estudios encontrados indican que su uso es menor que las técnicas tradicionales. Las técnicas que se han

usado a lo largo del periodo de tiempo del 2007 al 2013 son aquellas que utilizan una herramienta para establecer la comunicación entre los *stakeholders*. Como se puede apreciar en la gráfica la técnica de redes sociales es la más utilizada durante estos últimos 3 años, la segunda técnica es VORD, sin embargo esta última sólo es utilizada hasta el 2010. Para el 2007 se utilizaron las wikis y después se hace énfasis en el uso de redes sociales y otro tipo de herramientas. Para las técnicas de grupos de trabajo aunque se han empleado en el último año, su uso no ha sido de manera constante ya que sólo un estudio hace referencia a este tipo de técnica.

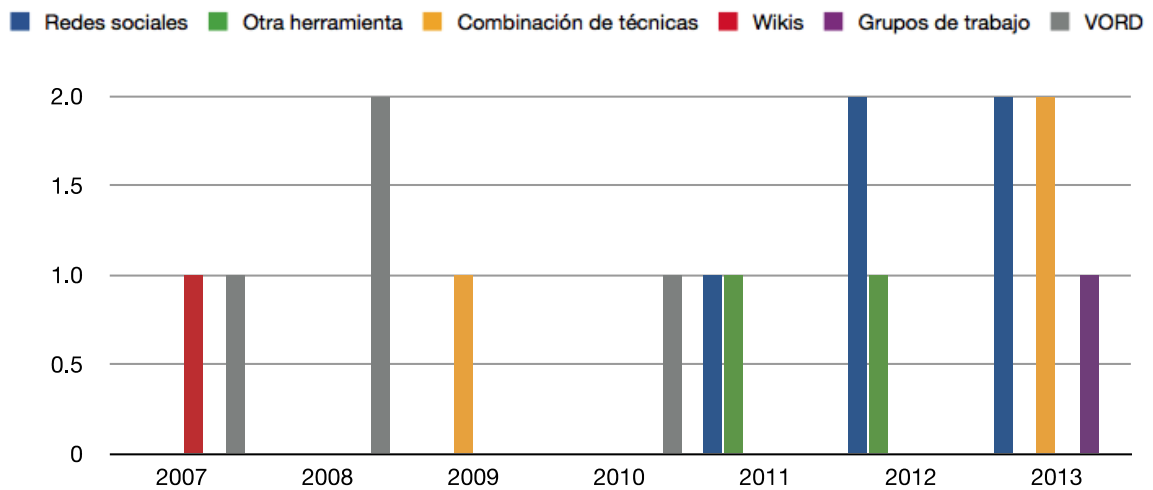


Figura 3.5. Número de estudios por técnica de elicitación grupal-colaborativa y año en el que fueron publicados.

3.2.2.1.3. Prototipos

Para la técnica de prototipos se hallaron 2 estudios referenciados (véase Tabla 8).

Tabla 8. Número de estudios para prototipos.

Técnica	Estudios referenciados	Frecuencia (# de estudios)
Prototipos.	Kamalrudin y Grundy (2011), Ramdoyal, Cleve y Hainaut (2010).	2

El uso de esta técnica se dió principalmente en los años 2010 y 2011 (véase Figura 3.6). Sin embargo, la técnica de prototipos no es muy utilizada en la elicitación de requisitos ya que muchas veces abarca etapas posteriores al desarrollo de software.

3.2.2.1.4. Técnicas de modelado

Dentro de las técnicas de modelado surgieron estudios que utilizan modelos como diagramas de flujos de datos, diagramas de estado, diagramas UML o métodos basados en metas u objetivos como KAOS e I*, y otros que utilizan por ejemplo diagramas de modelado de negocio. Para este grupo se encontraron doce estudios (véase Tabla 9) dentro de los cuáles siete se enfocan al modelado basado en metas (incluyendo cuatro para I*, para KAOS ninguno, y otros métodos tres) y cinco para los que utilizan otros tipos de modelado como los diagramas de estado y el diagrama de procesos de negocio.

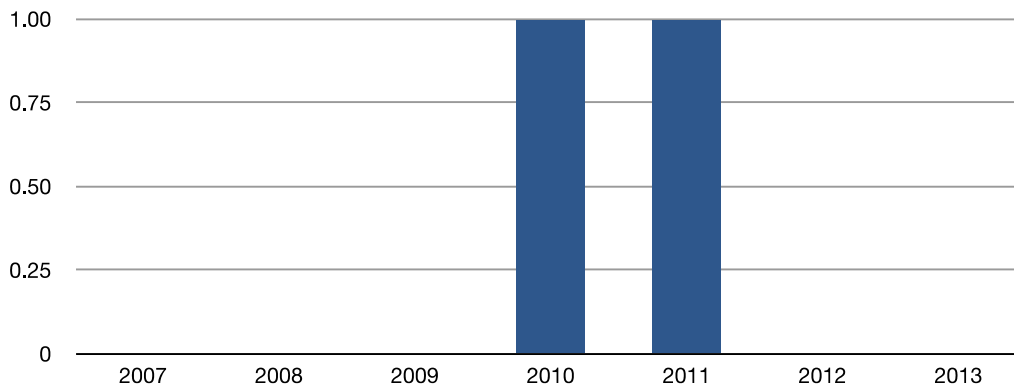


Figura 3.6. Número de estudios para la técnica de prototipos y año en el que fueron publicados.

En general, son las técnicas preferidas para elicitar requisitos hasta el 2011, debido principalmente a que facilitan el análisis de éstos y a que el uso de modelos apoya el entendimiento de los requisitos. Las dos técnicas que se basan en el modelado de metas son más utilizadas con siete estudios y esto se debe principalmente a que se definen los objetivos y en base a éstos se empiezan a elicitar los requisitos software.

Tabla 9. Lista de técnicas de elicitación de modelado.

Técnica	Estudios referenciados	Frecuencia (# de estudios)
Modelado.	Monsalve, April y Abran (2012), De la Vara y Sánchez (2009), Driss, Jamoussi, Jézéquel y Ghézala (2011), Sen y Jain (2007), Urbieta et al. (2012).	5
I*.	An et al. (2009), Maté, Trujillo y Franch (2011), Liu y Jin (2007), Trujillo, Soler, Fernández-Medina y Piattini (2009).	4
Basado en metas.	Chand, Reddy, Rao y Kumar (2010), Clements y Bass (2010), Widya et al. (2009a).	3

La tendencia del uso de estas técnicas ha ido disminuyendo (véase Figura 3.7) ya que se han aplicado más en otras etapas del desarrollo de software como el análisis de requisitos y la etapa de validación. Se utilizan principalmente para entender los objetivos o el contexto del software a desarrollar y no específicamente para obtener la lista de deseos y necesidades de los *stakeholders*.

3.2.2.1.5. Técnicas cognitivas

Las técnicas cognitivas incluyen el análisis de protocolo, la jerarquización (*laddering* en inglés), clasificación de tarjetas (*card sorting*) y rejillas de repertorio. Otra técnica cognitiva que en la actualidad se ha utilizado para elicitar requisitos es la ontología. Para este grupo se obtuvo un total de nueve estudios (véase Tabla 10), de los cuáles en seis estudios se utilizan ontologías para elicitar requisitos.

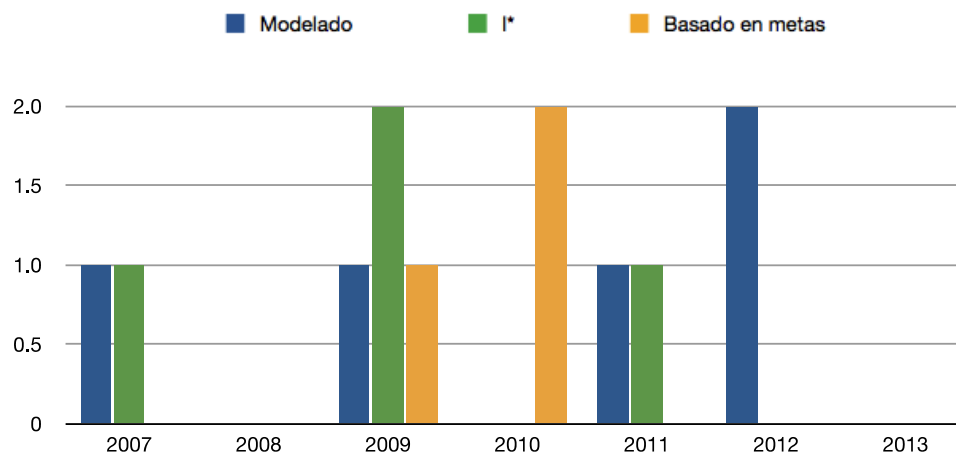


Figura 3.7. Número de estudios por técnica de elicitación de modelado y año en el que fueron publicados.

Tabla 10. Lista de técnicas de elicitación cognitivas.

Técnica	Estudios referenciados	Frecuencia (# de estudios)
Ontologías.	Hilaire, Cossentino, Gechter, Rodriguez y Koukam (2012), Jiang y Yang (2009), Kaiya, Shimizu, Yasui, Kaijiri y Saeki (2010), Li, Jin, Xu y Lu (2011), Vitharana, Jain y Zahed. (2012), Zhang y Lee (2010).	6
Análisis de tareas.	Habibizad Navin, Kiyani, Kheyri y Tahmasebi Rad (2009).	1
Clasificación de tarjetas.	Maiden (2009).	1
Rejillas de repertorio.	Niu y Easterbrook (2007).	1

Como se puede observar en la Figura 3.8 y en la Tabla 10, la técnica más utilizada de acuerdo al número de estudios encontrados en los últimos años es la de ontologías.

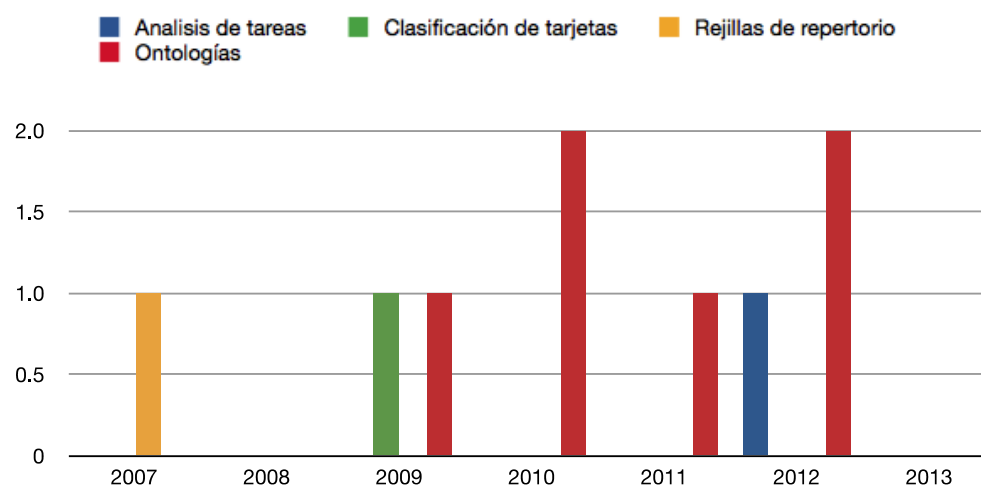


Figura 3.8. Número de estudios por técnica de elicitación cognitiva y año en el que fueron publicados.

3.2.2.1.6. Técnicas contextuales

Dentro de las técnicas contextuales se encuentran la etnografía, *contextual inquiry*, observación, la etnometodología y el análisis de conversación. Se encontraron un total de 5 estudios (véase Tabla 11 y Figura 3.9) que hacen referencia a este tipo de técnicas. La más utilizada es la etnografía con un total de 4 estudios, y uno para etnometodología. Existen técnicas de las que no se encontró evidencia de que hayan sido aplicadas actualmente como la de *contextual inquiry* y análisis de conversación.

Tabla 11. Número de estudios para técnicas contextuales.

Técnica	Estudios referenciados	Frecuencia(# de estudios)
Etnografía.	Brill y Knauss (2011), Doherty, McKay y Black (2007), McKnight y Luz (2010), Mendizabal, Spier y Saad (2012).	4
Etnometodología.	De La Flor ,Jirotko, Luff, Pybus y Kirkham (2010).	1

Las técnicas contextuales en general son más utilizadas dentro del campo de la usabilidad para analizar el comportamiento de los usuarios. A pesar de que no se utilizan mucho en la elicitación de requisitos se siguen utilizando poco otras áreas en los últimos años.

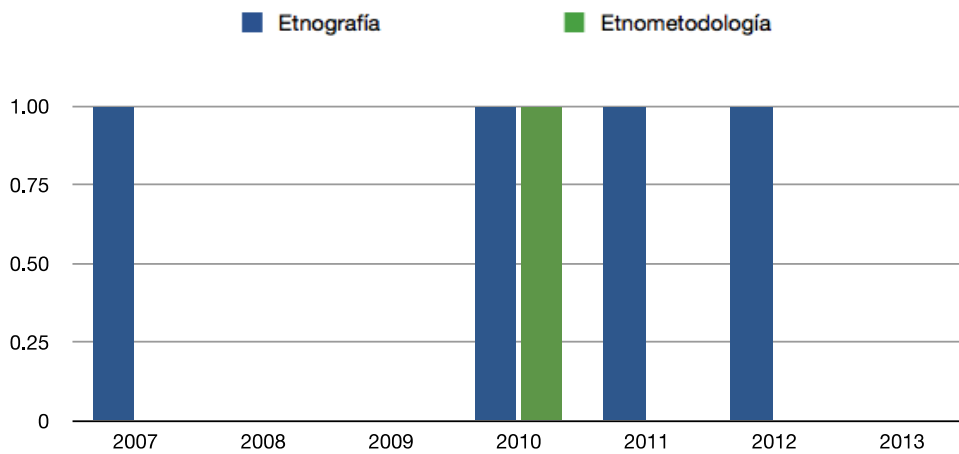


Figura 3.9. Número de estudios para la técnica de elicitación etnografía y año en el que fueron publicados.

3.2.2.1.7. Metodologías ágiles

Esta nueva categoría no ha sido definida en el *framework* de Nuseibeh y Easterbrook, ni en la clasificación de Hickey-Davis, sin embargo en ésta última se incluye a la programación extrema la cuál es parte de las metodologías ágiles. Para este grupo se localizaron seis estudios (véase Tabla 12 y Figura 3.10) que se presentaron con mayor frecuencia en estos últimos años.

Tabla 12. Número de estudios para métodos ágiles.

Técnica	Estudios referenciados	Frecuencia(# de estudios)
Métodos ágiles.	Chowdhury y Huda (2011), Derrick, Read, Nguyen, Callens y De Vreede (2013), Erfurth y Kirchner (2010), Mahmud y Veneziano (2011), Read y Briggs (2012), Sulfaro, Marchesi y Pinna (2007).	6

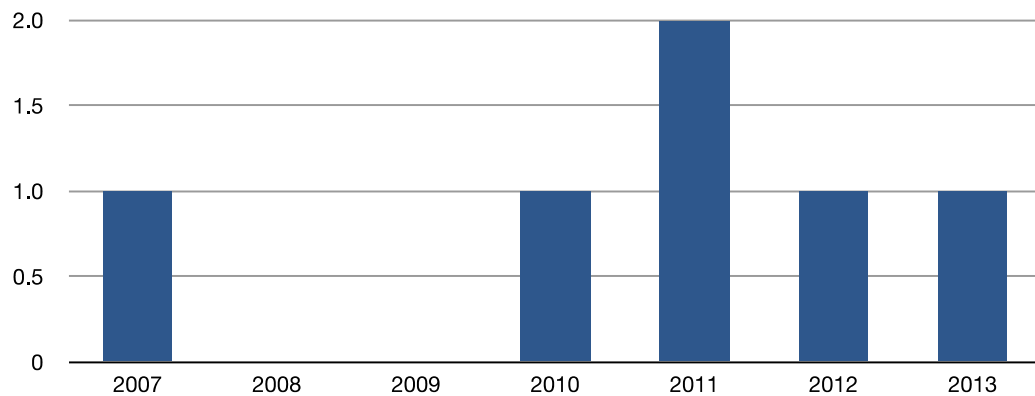


Figura 3.10. Número de estudios que utilizan métodos ágiles para la elicitación y año en el que fueron publicados.

Sulfaron et al. (2007) analizaron las prácticas ágiles en grandes organizaciones en donde identificaron la utilización de *user stories* para la elicitación de requisitos. Erfurth y Kirchner (2010) hicieron una adaptación de las tarjetas CUTA (Análisis Colaborativo de Tareas de Usuario, por sus siglas en inglés) para tener un mejor involucramiento de los *stakeholders* además de una clara definición de requisitos. Mahmud y Veneziano (2011) utilizan la técnica ágil llamada *mind-mapping* dentro de la metodología de Scrum. De la misma forma, Read y Briggs (2012) utilizaron el enfoque de Scrum pero adaptaron la aplicación de la técnica de *user stories* para el desarrollo de un software complejo. Chowdhury y Huda (2011) identifican la utilización de sesiones JAD dentro de la metodología ágil de Desarrollo Adaptable de Software (ASD, por sus siglas en inglés). El estudio de Derrick et al. (2013) se hace uso de la técnica de *group storytelling*, en la cual los *stakeholders* cuentan historias en grupo.

Un aspecto que es importante mencionar, es que 24 de los artículos (~27%) hacen mención a más de una técnica utilizada para la elicitación de requisitos; por ejemplo Bendjenna, Zarour y Charrel (2008) hacen referencia al uso de escenarios, casos de uso, I* y VORD, esto quiere decir que por lo general en un desarrollo software se aplica más de una técnica de elicitación para obtener los requisitos. Otro aspecto que hay que mencionar es que algunas técnicas han dejado de utilizarse por el surgimiento de otras que se han generado gracias a nuevos canales de comunicación ó al uso de nuevas metodologías de desarrollo.

3.2.2.2. PI2- técnicas que han demostrado mayor efectividad

Después de revisar las bases de datos electrónicas y aplicar los criterios de selección, 37 estudios fueron identificados para dar respuesta a la *Pregunta de Investigación 2 (PI2): ¿Qué técnicas o métodos de elicitación han demostrado mayor efectividad?*

Los estudios analizados proporcionan evidencia sobre las técnicas de elicitación más efectivas, tomando en cuenta tanto el contexto en el que son aplicadas como los parámetros que se consideran para medir dicha efectividad.

El uso o la elección de las técnicas de elicitación por parte del ingeniero de requisitos depende del análisis del contexto, tamaño y dominio de aplicación del software a desarrollar, el tipo de información a obtener, las características de los *stakeholders*, entre otros aspectos; por lo que medir la efectividad implica considerar ciertas condiciones y establecer bajo qué situaciones se puede considerar efectiva una técnica. Por ejemplo, Zapata, Torres, E., Sevilla, G., Aballay, L., & Reus (2012) mencionan que para medir la efectividad de una técnica de elicitación, primero se debe tomar

en cuenta el contexto en el que el software se va a desarrollar: ya sea en entornos distribuidos o en entornos tradicionales; y posteriormente se deben considerar otros factores como el tamaño de la empresa, el tamaño del software, el dominio de aplicación, y las herramientas software de comunicación disponibles.

Una vez que se han definido las condiciones en las cuáles se analizará la efectividad de la técnicas de elicitación, se deben determinar los criterios o parámetros que se van a tomar en cuenta para medir dicha efectividad. En los estudios analizados, algunos autores miden la efectividad a través de la calidad de los requisitos obtenidos (Ocker, Fjermestad, Hiltz & Johnson, 1998; Lloyd, Rosson & Arthur, 2002; Duggan & Thachenkary, 2004; Sabahat, Iqbal, Azam & Javed, 2010; Pitula & Radhakrishnan, 2011; Zapata et al., 2012), otros a través de la cantidad de información o conocimiento elicitado, es decir que los requisitos estén completos (e. g. Moody, Blanton & Cheney, 1998; Jones, Lynch, Maiden y Lindstaedt, 2008; Opdahl & Sindre, 2009; Sakhnini, Mich & Berry, 2012), otros autores como Sabahat et al. (2010) y Mahmud y Veneziano (2011) consideran que la satisfacción del usuario es lo que indica si el software desarrollado es el correcto (es decir, que los requisitos obtenidos son correctos, viables y verificables); y otros como Damian, Lanubile y Mallardo (2006), Yamanaka, Noguchi, Yato y Komiya (2010) y Zapata et al. (2012) miden el número de defectos en los requisitos, el grado de volatilidad, el nivel de acuerdo entre los *stakeholders* (lo que permite evaluar si los requisitos son consistentes y precisos).

De acuerdo con (Ocker et al., 1998; Isabirye & Flowerday, 2008; Mishra, Mishra y Yazici, 2008; Thew et al., 2009; Girase, 2012; Zapata et al., 2012) es muy recomendable que el ingeniero de requisitos utilice la combinación de diferentes técnicas de elicitación debido a que bajo ciertas condiciones cada una de ellas posee ventajas y desventajas, por lo que integrarlas ayuda a complementarlas y así obtener información más completa y consistente de las necesidades de los *stakeholders*. Existen enfoques iterativos como los propuestos por Isabirye y Flowerday (2008), Mishra et al. (2008), y Sabahat et al. (2010), que recomiendan la aplicación de técnicas más efectivas en la primera iteración (dándole el nombre de técnicas primarias) y el uso de otras menos efectivas en las siguientes iteraciones (dándole el nombre de técnicas secundarias). Por ejemplo, si no se conoce el contexto del problema se deben aplicar ciertas técnicas que proporcionen información más detallada del dominio de la aplicación en la primera iteración (e. g.. entrevistas) y en una segunda, se aplicarían otras técnicas (e. g. prototipos) que proporcionen información más específica. Isabirye y Flowerday (2008) recomiendan las entrevistas, JAD, *stories* y *focus group* como técnicas primarias y a los prototipos y casos de uso como técnicas que ayudan a complementar la información ya obtenida.

También, al analizar la efectividad de las técnicas de elicitación, algunos de los estudios analizados mencionan los tipos de problemas y desafíos que se presentan en esta etapa así como las técnicas que ayudan a aminorarlos (Mavin, Novak, Wilkinson, Maiden y Lynch, 2008; Sabahat et al., 2010). Dentro de los problemas que son mencionados destacan los siguientes: de comunicación, de lenguaje, tiempo, costos, identificación del dominio, identificación de *stakeholders* y los ya mencionados en la sección 2.1.2.

El 37% de los estudios analizados mencionan dos aspectos importantes a tomar en cuenta a la hora de seleccionar la técnica de elicitación de requisitos: la comunicación y la negociación, ya que debe haber una buena comunicación para obtener los requisitos correctos y además se debe negociar entre los *stakeholders* para llegar a un acuerdo y determinar los requisitos del software a desarrollar.

Como ya se mencionó anteriormente, la mayoría de los estudios fueron realizados tanto para contextos o entornos distribuidos como para tradicionales, colaborativos y metodologías ágiles, y es bajo estas condiciones como se analizó su efectividad. Dentro de los hallazgos se observa que en

entornos de Desarrollo de Software Global o distribuido (DGS, por sus siglas en inglés) es muy importante tener una buena comunicación y llevar a cabo una exitosa negociación entre los *stakeholders* (Calefato, Damian & Lanubile, 2012) ya que éstos se encuentran geográficamente distribuidos, por lo tanto hay que tener en cuenta qué medio de comunicación utilizar y el tipo de herramientas que faciliten esta tarea. Laurent y Cleland-Huang (2009) muestran las prácticas que permiten un proceso de elicitación más efectivo dentro de un entorno colaborativo. En otro estudio (Ocker et al., 1998) se compara la efectividad de cuatro formas de comunicación (*face-to-face*, conferencia por computadora asíncrona, conferencia por computadora síncrona y conferencia por computadora asíncrona combinada con *face-to-face*) dando como resultado que el enfoque combinado es más efectivo en entornos colaborativos.

Los hallazgos encontrados muestran que la técnica de entrevistas es considerada efectiva dentro de entornos DGS, poco menos que los prototipos y escenarios y que es la más utilizada y es considerada como la técnica más efectiva en entornos tradicionales, especialmente si se aplican entrevistas estructuradas. Así mismo, se identificó que al combinar esta técnica con otras técnicas aumenta la calidad de los requisitos elicitados. Además se obtuvo que las entrevistas deben ser utilizadas como una técnica primaria de elicitación de requisitos.

La técnica de prototipos es considerada como la más efectiva dentro de los entornos DGS ya que con ella se puede hacer una representación temprana del producto final y así los clientes se sienten más satisfechos (Sabahat et al., 2010). Sin embargo en otros estudios es considerada como una técnica menos efectiva porque no proporciona mucha información o simplemente ayuda a complementar la información obtenida con otras técnicas (Lloyd et al., 2002; Davis, Dieste, Hickey, Juristo y Moreno, 2006; Isabirye & Flowerday, 2008; Mishra et al., 2008).

Los resultados de la investigación realizada por Sabahat et al. (2010) muestran que la técnica de escenarios es considerada cómo una técnica efectiva dentro de los entornos de DGS ya que pueden sustituir a los prototipos si estos últimos resultan muy costosos o los requisitos son muy difíciles de extraer. Además ayudan a analizar las tareas que debe realizar el software.

Para los casos de uso, se encontró que en combinación con los *misuse cases*¹⁵ mejoran la eficiencia de la elicitación de requisitos (Opdahl & Sindre, 2010). Los *misuse cases* resultan ser eficientes al elicitar requisitos de seguridad, aunque existen otras técnicas más especializadas.

Los cuestionarios se consideran menos efectivos ya que las preguntas pueden ser abiertas o cerradas y en consecuencia no representan efectivamente las necesidades del cliente por lo cual afectan en la consistencia y precisión de los requisitos (Moore & Shipman, 2000; Sabahat et al., 2010).

Dentro la clasificación técnicas grupales-colaborativas, los resultados indican que los grupos de trabajo ayudan a negociar y colaborar efectivamente en la etapa de elicitación. Duggan y Thachenkary (2004), muestran que JAD es una técnica efectiva en la determinación de requisitos de calidad. Isabirye y Flowerday (2008) muestran que las técnicas grupales son efectivas para elicitar requisitos en una primera iteración, para obtener información del contexto y además permiten a negociar y llegar a un acuerdo de manera más rápida y eficiente. Otra técnica dentro de este grupo que ha sido evaluada es la lluvia de ideas, la cuál no ha resultado muy útil ni eficiente (Sakhnini, Mich & Berry, 2012).

Por último respecto de las metodologías ágiles, se encontraron dos proyectos en los que se analizó la efectividad de las técnicas ágiles: un proyecto desarrollado bajo la metodología Scrum y

¹⁵ Es una variante de los casos de uso, y son usados para representar comportamientos no deseados en el software a desarrollar con el objetivo de elicitar requisitos de seguridad (Sindre & Opdahl, 2005).

otro con XP. En estos estudios específicamente se utilizó la técnica de *stories* o narrativas. Isabirye y Flowerday (2008) proponen utilizar ésta técnica como técnica primaria ya que ayuda a establecer una buena comunicación con los *stakeholders*. Otro estudio (Pitula & Radhakrishnan, 2011) demuestra claramente que la técnica de *stories* es una manera eficaz y aceptable para que los *stakeholders* expresen sus preocupaciones y necesidades a través de historias, independientemente de su nivel de alfabetización, idioma o clase social. Esta técnica tiene la ventaja de aprovechar el modo habitual de comunicación que tienen los *stakeholders*, para obtener así una comprensión más completa del problema y del contexto. Isabirye y Flowerday (2008) mencionan que una de las barreras que se pueden encontrar al aplicar las entrevistas es el lenguaje y el contexto cultural, por lo que en este caso sería más conveniente aplicar *stories*.

En la Tabla 13 se pueden ver los resultados encontrados en la búsqueda, de acuerdo a los atributos de calidad de los requisitos software.

Tabla 13. Efectividad de las técnicas tomando en cuenta los atributos de calidad de los requisitos obtenidos.

Atributos de calidad de los requisitos	Técnica que demostró mayor efectividad.	Estudios referenciados (Evidencia a favor)	Estudios referenciados (Evidencia en contra)	Comentarios
Completos, correctos y viables.	Las entrevistas obtienen más información que otras técnicas de elicitación.	Breivik y Supphellen (2003), Browne y Rogich (2001), Carrizo, Dieste y López (2011), Corbridge, Rugg, Major, Shadbolt y Burton (1994), Davis et al. (2006), Janette, Will y Blanton (1996), Lloyd et al. (2002), Moody, Blanton y Cheney (1998), Thew et al. (2009), van Velsen et al. (2008), Yamanaka et al. (2010).	Sabahat et al. (2010).	En algunos estudios se muestra la comparación entre las entrevistas estructuradas y no estructuradas siendo más efectivas las estructuradas.
Completos, correctos, consistentes, precisos y viables.	La técnica de entrevistas en combinación con otras técnicas (cuestionarios y lluvia de ideas) son más efectivas que sólo utilizar entrevistas.	Isabirye y Flowerday (2008), Zapata et al. (2012).		Tanto en entornos de desarrollo de software tradicionales como distribuidos.
Completos, correctos, consistentes.	Las entrevistas son utilizadas como técnicas primarias.	Isabirye y Flowerday (2008), Sabahat et al. (2010).		
Correctos, completos y verificables.	La técnica de prototipos es la más efectiva (es utilizada como técnica primaria).	Moore y Shipman (2000), Sabahat et al. (2010).	Davis et al. (2006), Isabirye y Flowerday (2008), Lloyd et al. (2002), Mishra, et al. (2008).	Es considerada efectiva en entornos distribuidos.

Completos correctos. y	La técnica de escenarios es una técnica efectiva.	Sabahat et al. (2010), Fowlkes et al. (2000), Isabirye y Flowerday (2008), Laporti, Borges y Braganholo (2009), Mavin et al. (2008), Thew et al. (2009), Zmud, Anthony y Stair (1993).		La técnica de escenarios es utilizada como una técnica primaria.
Completos correctos. y	Los casos de uso son una técnica efectiva (se incluyen también los <i>misuse cases</i>).	Laporti, Borges y Braganholo (2009), Opdahl y Sindre (2009), Lloyd et al. (2002), Siau y Wang (2007).		Utilizada como técnica secundaria.
Completos.	Los cuestionarios son una técnica menos efectiva.	Moore y Shipman (2000), Sabahat et al. (2010).		Ayuda a descubrir nuevos requisitos.
Completos, correctos, consistentes, precisos y viables.	Los grupos de trabajo son más efectivos que las entrevistas y etnografía.	Dai y Wang (2009), Girase (2012), Jones et al. (2008).		Son efectivos siempre que sean creativos, además es una técnica muy efectiva para resolver conflictos.
Consistentes y precisos.	Las técnicas de modelado son útiles para analizar y capturar requisitos.	Mahmud y Veneziano (2011).		
Completos, correctos, viables y modificables.	Los métodos ágiles son más efectivos para todo tipos de proyectos.	Isabirye y Flowerday (2008), Li-ping y Long-jun (2010), Thew et al. (2009).		La técnica utilizada en estos entornos de desarrollo son las <i>stories</i> o narrativas.

Por todo lo anteriormente expuesto, se puede concluir que para medir la efectividad de las técnicas de elicitación se debe tomar en cuenta el contexto donde se va a desarrollar y aplicar el software, ya que de esto dependerá la aplicación de las mismas y que una sola técnica por sí misma obtendrá los mismos requisitos que si se aplican varias. Cada una de las técnicas de elicitación posee características propias que la hacen ser efectiva en ciertos contextos y adquirir cierto tipo de conocimiento, además cada una de ellas tiene ventajas y desventajas por lo que es necesario que el ingeniero de requisitos tenga el conocimiento necesario de cada una de ellas y pueda aplicarlas correctamente.

3.2.2.3. PI3- aspectos de las técnicas que influyen en su efectividad

Tomando en cuenta los criterios de selección se identificaron 19 estudios para dar respuesta a la *Pregunta de Investigación 3 (PI3): ¿Qué aspectos de cada técnica de elicitación influyen en su efectividad?*

Como se puede observar en la respuesta a la PI2, para decir que una técnica o método de elicitación es efectiva se tienen que analizar ciertas condiciones del nuevo software a desarrollar como son: el análisis del contexto, tamaño y dominio de aplicación, tipo de información a obtener, características de los *stakeholders*, etc. Ahora bien, una vez que se han identificado estas

condiciones, es necesario analizar e identificar qué características poseen las técnicas o métodos de elicitación que trabajan mejor bajo las condiciones establecidas (i.e. que son efectivas). Un aspecto importante a considerar en esta SLR es que actualmente existe una gran variedad de técnicas de elicitación, más de 40 (Young, 2002; Jiang, Eberlein, Far & Mousavi, 2008), sin embargo unas son variantes de otras, por ejemplo, la técnica *laddering* es un tipo particular de las entrevistas estructuradas (Tiwari, Rathore & Gupta, 2012) y otras tantas son las mismas pero con nombres diferentes (Jiang et al., 2008), por ejemplo *think aloud* y *protocol analysis* (Serna, 2012).

Los estudios que dieron respuesta a esta PI, fueron en su mayoría aquellos cuyo tema central es la propuesta de prácticas para mejorar el proceso de elicitación de requisitos y prevenir defectos en esta etapa. El 81% de los estudios siguen básicamente el siguiente patrón: los autores identifican en qué condiciones son efectivas las técnicas de elicitación analizadas y qué características de las técnicas las hacen ser efectivas. Las condiciones en las que una técnica es efectiva se refiere a la condición general del software (i.e. contexto, tamaño y dominio de aplicación, características de *stakeholders*, tipo de información a obtener) en las que es apropiado aplicar ciertas técnicas de elicitación. Así, cada circunstancia demanda la aplicación de una técnica de elicitación apropiada debido a que cada técnica posee ciertas características que las vuelven apropiadas para una situación, condición o circunstancia específica.

Hickey y Davis (2007) definen una ontología para este esquema, especificando las principales características de las técnicas de elicitación así como del entorno. Para representar las propiedades de las técnicas utilizan un vector multidimensional en el que consideran 10 dimensiones: ubicación física, ubicación temporal, quién almacena la información, rol del analista, convergencia/divergencia, anonimidad, número de *stakeholders*, si se basa en el uso de una herramienta, tipo de enfoque (de producto o humano) y el tipo de elicitación (directa o indirecta). Las características del entorno son agrupadas en 5 categorías: características del dominio del problema, características de la solución del dominio, características de los *stakeholders*, características de los desarrolladores y características del ingeniero de requisitos. Dentro de estas cinco categorías se encuentran 50 características del entorno que pueden influenciar en la decisión de seleccionar una o más técnicas de elicitación. Este estudio permite entender cómo es que se puede determinar la efectividad de algunas técnicas de elicitación.

La investigación realizada por Maiden y Rugg (1996) identificó seis aspectos a tomar en cuenta en la selección de una técnica de elicitación: propósito de los requisitos, tipo de conocimiento, filtrado interno de conocimiento, fenómeno observable, contexto de adquisición e interdependencias entre los métodos. Davis (1999) menciona en qué condiciones son efectivas las técnicas de elicitación, como las características de los stakeholders, características del ingeniero de requisitos y el tipo de información a obtener. Tsumaki y Tamai (2006) clasifica a las técnicas de elicitación en dos dimensiones: el tipo de operación y el tipo de objetivo, esto se refiere a cómo se conduce el proceso de elicitación y cuál es el objetivo a ser analizado; además considera cinco factores del entorno (dominio de aplicación, experiencia del ingeniero de requisitos, recursos de información, involucramiento del usuario, y propiedades de los requisitos) que caracterizan a un proyecto para elegir la técnica de elicitación.

Otros estudios consideran otros factores o parámetros del entorno para hacer la selección de técnicas, por ejemplo Kausar, Tariq, Riaz y Khanum (2010) consideran el tipo de proyecto, objetivo de los *stakeholders*, número de *stakeholders*, grado de involucramiento de los *stakeholders*, estado del proyecto, limitaciones de presupuesto, limitaciones de los recursos, limitaciones de tiempo, nivel de requisitos funcionales correctos, y el nivel de calidad en función de los requisitos no funcionales. Tiwari, Rathore y Gupta (2012) también identifican siete características de las circunstancias en la

que se desarrolla el software: tipo de *stakeholders*, entorno social, dominio del sistema, tipo de usuarios finales, alcance del sistema, habilidad de los analistas y la metodología a seguir, ya sea para el proceso de IR o para todo el proceso de desarrollo del software (Sajid, Nayyar & Mohsin, 2010; Serna, 2012). Serna (2012) considera cuatro factores de selección para las técnicas de elicitación: nivel de abstracción de los requisitos, fuentes de los requisitos, obstáculos en los requisitos y el nivel de certidumbre.

Una vez que se han expuesto las características del contexto en el que se realiza el proceso de elicitación, se mencionan las propiedades de las técnicas de elicitación que las hacen ser efectivas bajo ciertas condiciones. Sutcliffe (1995) señala que un buen enfoque es combinar los prototipos y cuestionarios al elicitar los requisitos, ya que los prototipos son utilizados para complementar los requisitos mientras que los cuestionarios ayudan a obtener conocimiento del dominio.

Maiden y Rugg (1996) presentan doce técnicas de elicitación (observación, entrevistas estructuradas y no estructuradas, *protocol analysis*, *card sorting*, *laddering*, *repertory grids*, lluvia de ideas, prototipos, escenarios, *workshops* y, etnografía) describiendo cada una de ellas e indicando cuáles son las precondiciones de uso (contexto de aplicación), sus ventajas (aspectos que las hacen ser efectivas) y desventajas. Davis (1999) describe cuándo son efectivas nueve técnicas de elicitación (entrevistas, cuestionarios, etnometodología, lluvia de ideas, *storyboard*, prototipos y otras como leer, investigar y el desarrollo evolutivo); las entrevistas son útiles cuando los *stakeholders* son expertos en la materia, tienen tiempo disponible, son accesibles y además se cuenta con un entrevistador experto; los cuestionarios sirven cuando hay muchos *stakeholders* involucrados ya que se pueden determinar las tendencias de opinión acerca de un requisito específico; la etnometodología es efectiva cuando se trata de automatizar alguna función y cuando no se tiene conocimiento del problema o su solución; la lluvia de ideas es eficiente cuando existen muchos *stakeholders* porque cada uno de ellos tiene diferentes conocimientos, así ellos aprovechan las ideas de otros para generar una imagen más completa de los requisitos; el *storyboard* ayuda a representar diferentes estados secuenciales del entorno del problema; los prototipos reducen el riesgo asociado con el desarrollo de un software incorrecto ya que ayudan a aclarar y entender requisitos confusos; las lecturas se aplican a documentos que proporcionarán conocimiento para un mejor entendimiento del dominio; la investigación se aplica cuando los *stakeholders* son creativos sin embargo todavía no existen ideas del software a desarrollar; el desarrollo evolutivo se aplica cuando hay requisitos bien entendidos ya que los *stakeholders* al ver cierta funcionalidad pueden pensar en otras más.

Hofmann (2000) hace una extensión del *framework* propuesto por Maiden y Rugg (1996) incluyendo 3 métodos adicionales: análisis de documentos, cuestionarios y *focus group*; estos autores recalcan la importancia de utilizar diferentes técnicas para elicitar diferentes tipos de requisitos pero también agregan al *framework* la elicitación de requisitos no funcionales clasificando los tipos de conocimiento (comportamiento, funciones y datos) para cada uno de los tipos de requisitos, además muestran la forma de especificación de las técnicas de elicitación estudiadas.

Lausen y Vinter (2001) mencionan que los escenarios y prototipos son técnicas que ayudan a prevenir defectos en los requisitos porque ayudan a evitar la obtención de requisitos incorrectos, incompletos e inconsistentes porque describen gráficamente cuáles serán las tareas de los usuarios, sin embargo poseen una desventaja que tiene que ver con el esfuerzo y el dinero necesarios para elaborarlos.

Young (2002) menciona 8 técnicas de elicitación (entrevistas, análisis de documentos, lluvia de ideas, *workshops*, prototipos, casos de uso, *storyboards*, análisis de interfaces y técnicas de modelado) y las características que las hacen ser efectivas; e.g. las entrevistas tienen la ventaja de poder aplicar preguntas libres de contexto, lo que no obliga a dar alguna respuesta en particular; el

analizar documentos mejora y complementa los requisitos obtenidos porque se consultan otras fuentes de requisitos; la lluvia de ideas ayuda a aumentar la creatividad; los *workshops* son útiles para llegar a un consenso, son menos costosos que las entrevistas múltiples, ayudan a estructurar, priorizar y gestionar los requisitos, son dinámicos, interactivos y cooperativos; los prototipos permiten entender las interacciones con el sistema; los casos de uso proveen un contexto para los requisitos por el hecho de expresar secuencias de eventos y mejora la comunicación dentro del equipo de desarrollo, los *storyboards* no son costosos; el análisis de interfaces clarifica el alcance del producto, ayuda a la evaluación del riesgo, reduce costos de desarrollo y mejora la satisfacción del cliente; y las técnicas de modelado facilitan el entendimiento y ayudan a eliminar ambigüedades e inconsistencias.

Zowghi y Coulin (2005) describen 20 técnicas de elicitación pero hacen una comparación de ocho técnicas (entrevistas, técnicas de dominio, técnicas colaborativas, etnografía, prototipos, técnicas orientadas a objetivos, escenarios y puntos de vista) destacando cinco aspectos (entendimiento del dominio, identificación de fuentes de requisitos, análisis de *stakeholders*, selección de técnicas y enfoques y obtención de requisitos) e indicando cuáles poseen cada una de las técnicas; para el primer aspecto, la única técnica que no ayuda a tener un entendimiento del dominio son los prototipos; para la identificación de fuentes de requisitos los prototipos y la etnografía no son útiles; para la selección de técnicas y enfoques sólo son apropiadas las entrevistas, técnicas de dominio y las colaborativas; para los otros dos aspectos se consideran todas las técnicas de elicitación.

Tsumaki y Tamai (2006) mapean 16 técnicas de elicitación en el diagrama de dos dimensiones antes mencionado, mostrando que cuatro técnicas de elicitación (lluvia de ideas, métodos orientados a objetivos, escenarios, y métodos de descomposición de dominios) difieren entre ellas por lo que cada una se encuentra en un cuadrante; la lluvia de ideas es una técnica no metodológica que es útil aplicar cuando no se tiene experiencia dentro del contexto o dominio de aplicación; los escenarios no son metodológicos pero se aplican cuando se tiene más conocimiento del dominio; las técnicas de descomposición del dominio son metodológicas y se aplican cuando el contexto es estático, y por último los métodos orientados a objetivos que son metodológicos se aplican cuando es necesario analizar a fondo el dominio de aplicación.

Jiang et al. (2008) identifican 46 técnicas dentro de la IR pero sólo consideran 22 técnicas para la elicitación; de igual forma identifican 31 atributos para cada una de las técnicas, pero no todos son aplicables únicamente a la fase de elicitación, es decir se aplican a todas las fases del proceso de IR: habilidad para facilitar la comunicación, habilidad para ayudar a entender cuestiones sociales, habilidad para obtener conocimiento del dominio, habilidad para conseguir conocimiento explícito, habilidad para identificar *stakeholders*, habilidad para identificar requisitos no funcionales, habilidad para identificar puntos de vista, habilidad para modelar y entender requisitos, habilidad para analizar y modelar requisitos con notaciones entendibles, habilidad para analizar requisitos no funcionales, habilidad para facilitar la negociación con los clientes, habilidad para priorizar requisitos, habilidad para identificar la accesibilidad del sistema, habilidad para modelar los requisitos de la interfaz, habilidad para identificar requisitos reusables y apoyar el reuso de requisitos, habilidad para representar requisitos, habilidad para apoyar la verificación de requisitos, integridad de la semántica de notación, habilidad para escribir requisitos precisos, habilidad para escribir requisitos completos, habilidad para apoyar la gestión de requisitos, habilidad para apoyar el diseño de sistemas, implementación de la notación usada, habilidad para identificar requisitos ambiguos, habilidad para identificar inconsistencias o conflictos, habilidad para identificar requisitos incompletos, habilidad para apoyar el proceso de IR basado en COTS, madurez de la herramienta de apoyo, curva de aprendizaje (costo de introducción), costo de aplicación y

complejidad de la técnica. De acuerdo a estos 31 atributos, Jiang et al. (2008) evaluaron seis técnicas, de las cuáles tres son aplicables a la adquisición de requisitos: JAD, prototipos exploratorios y programación extrema; respecto a esta evaluación se observa que JAD facilita la comunicación identificando los puntos de vista de los *stakeholders*, ayuda a entender cuestiones sociales, a obtener conocimiento del dominio e identificar requisitos no funcionales; los prototipos facilitan la comunicación, negociación y la identificación de la accesibilidad del sistema, así como a modelar los requisitos de la interfaz pudiendo ser apoyados por el uso de una herramienta; y respecto a la programación extrema, ésta técnica también facilita la comunicación, ayuda a analizar, modelar y entender los requisitos, a modelar los requisitos de la interfaz y a identificar requisitos ambiguos. Además estos mismos autores identifican 21 atributos de los proyectos software para que de acuerdo a éstos se elijan las técnicas que más se adecuen al proyecto a desarrollar: tamaño del proyecto, volatilidad de los requisitos, categoría del proyecto, grado de seguridad, tipo de producto, tamaño del equipo, complejidad del proyecto, límites de tiempo, relación entre el cliente y la organización, conocimiento del dominio, criterios de calidad del producto, limitaciones de costo, conocimiento del proceso de IR, grado de conocimiento de los requisitos, disponibilidad de un facilitador, heterogeneidad de los *stakeholders*, grado de innovación del proyecto, disponibilidad del cliente, grado de importancia de la reusabilidad, grado de importancia de elicitar el conocimiento implícito, y grado de *outsourcing*.

Pitts y Browne (2007) mencionan que la ventaja de utilizar entrevistas estructuradas radica en saber elaborar el cuestionario, lo que significa que es necesario formular las preguntas de tal forma que se extraiga toda la información necesaria.

Seyff et al. (2009b) encontraron que la etnografía en combinación con los prototipos ayuda a obtener requisitos que no han sido descubiertos por otras técnicas (*workshops*), pero que los prototipos permiten descubrir una mayor cantidad de requisitos ya que proporcionan información adicional. Daud & Bakar (2010) concluyen que la etnografía es una técnica que ayuda a entender los procesos y tareas de los usuarios.

Kausar et al. (2010) destacan los siguientes los beneficios y limitaciones de 8 técnicas de elicitación de requisitos: las entrevistas y cuestionarios son económicos y fáciles de implementar; la entrevista es la manera más simple de obtener requisitos desde múltiples *stakeholders* y la más efectiva para entender problemas de software existentes porque ayuda a tener una visión general de los requisitos; los cuestionarios son una buena forma de recolectar requisitos de muchos *stakeholders* a un bajo costo; las técnicas contextuales son una buena opción cuando no existe limitaciones de tiempo y cuando el usuario no puede explicar sus necesidades; los prototipos son la manera más efectiva para representar funcional y gráficamente el software, y en el caso de ser evolutivos, pueden reducir el tiempo de desarrollo y mejorar la satisfacción del cliente; con los escenarios se pueden extraer requisitos mientras se analiza la interacción entre el software y el usuario; la lluvia de ideas no limita las ideas por lo que genera una gran cantidad de ellas para poder seleccionar las mejores; y por último menciona que los *workshops*, la lluvia de ideas, y las *storyboard* proveen una manera formal y objetiva de elicitar requisitos.

Sajid, Nayyar y Mohsin (2010) describen cuatro técnicas de elicitación (entrevistas, JAD, etnografía, y escenarios) y mencionan que las entrevistas son eficaces para automatizar procesos e innovarlos (i.e. cuando ya existe un software anterior), JAD es útil cuando existen varios *stakeholders* y se desarrollan sistemas críticos y distribuidos, la etnografía es eficaz cuando un sistema debe ser mejorado y los escenarios cuando se cuenta con tiempo y recursos limitados.

González y LeRoy (2011) argumentan a través de un caso de estudio que la lluvia de ideas no es muy efectiva en la elicitación de requisitos porque se obtuvo un número mayor de requisitos con

la aplicación de la técnica *appreciative inquiry*, ya que ésta al ser una técnica colaborativa promueve la participación de los *stakeholders*, mejora la comunicación entre ellos y evita la duplicidad de los requisitos obtenidos.

Serna (2012) expone las características de los métodos conversacionales, observacionales, analíticos y sintéticos para elicitar requisitos: los métodos conversacionales aprovechan la expresión verbal para elicitar el conocimiento explícito; las técnicas observacionales son un medio que ayuda a comprender el dominio de aplicación y ayudan a obtener los requisitos implícitos, además son útiles cuando los *stakeholders* tienen dificultades para expresarse ya que es más fácil demostrar sus tareas; las técnicas analíticas ayudan a recolectar, analizar y estructurar la información existente del dominio de aplicación; y los métodos sintéticos combinan diferentes canales de comunicación, y proporcionan modelos para mostrar las características del sistema y las interacciones con los usuarios.

Tiwari, Rathore y Gupta (2012) describen 27 técnicas de elicitación destacando las ventajas de algunas de ellas; por ejemplo, los cuestionarios son útiles para obtener requisitos precisos y consistentes, en tanto las encuestas sirven para elicitar requisitos de un gran número de *stakeholders*; las técnicas colaborativas permiten llegar a un consenso y a un entendimiento común, así mismo acceden a las expectativas de los usuarios; las técnicas cognitivas permiten obtener y analizar información del dominio de aplicación; las técnicas observacionales permiten entender las tareas que realizan los usuarios y el contexto en el que se desarrollan; por último describen los métodos ágiles, destacando que estos son exitosos para requisitos volátiles y utilizan un medio eficaz de comunicación: *face-to-face*.

Al Mrayat, Norwawi, y Basir (2013) utilizan la misma clasificación que propone Serna (2012) para los métodos de elicitación, y además hacen una comparativa entre los cuatro grupos mostrando sus principales ventajas.

Cómo se puede apreciar, en sólo un 26% de los estudios analizados (Kausar et al., 2010; Sajid, Nayyar & Mohsin, 2010; Serna, 2012; Tiwari, Rathore & Gupta, 2012; Al Mrayat et al., 2013) agrupan las técnicas de elicitación mediante la identificación de alguna característica en particular. Por ejemplo, Kausar et al. (2010) en un inicio las agrupa en técnicas: tradicionales, basadas en grupo, basadas en escenarios y contextuales, pero después propone agruparlas solamente en técnicas básicas y técnicas avanzadas siendo estas últimas las que involucran más a los *stakeholders*. Sajid et al. (2010), Serna (2012) y Al Mrayat et al. (2013) agrupan los métodos considerando la forma en que son aplicados: conversacionales, observacionales analíticos y sintéticos. Por último, Tiwari et al. (2012) las clasifican de acuerdo a la naturaleza de comunicación en: tradicionales, contextuales, colaborativas y cognitivas.

La mayoría de los estudios mencionan que es importante la experiencia del ingeniero de requisitos en la selección de la técnica de elicitación, ya que un buen elicitor emplea múltiples técnicas y usa las correctas para cierto tipo de problemas y situaciones (Davis, 1999). Además recalcan la importancia que tiene la comunicación en el proceso de elicitación así como lo importante que es la combinación de las técnicas de elicitación y el utilizar enfoques iterativos.

En la Tabla 14 se pueden ver resumidos los resultados encontrados en la búsqueda, mostrando las técnicas de elicitación y sus características por las que éstas son más efectivas.

Después de contestar esta pregunta de investigación, se puede concluir que seleccionar un sólo método para elicitar requisitos no es suficiente para poder obtener requisitos de calidad, i.e. completos, correctos, consistentes, relevantes, viables, modificables, verificables, trazables y sin ambigüedades, por lo que es recomendable utilizar dos o más técnicas que ayuden a conseguir una imagen clara de éstos (Seyff et al., 2009b; Sajid, Nayyar & Mohsin, 2010).

Tabla 14. Aspectos de las técnicas que las hacer ser más efectivas.

Técnica de elicitación	Aspectos que influyen en la efectividad de la técnica	Estudios referenciados
Entrevistas.	<p>Son económicas y fáciles de implementar.</p> <p>Son efectivas para entender los problemas en los sistemas ya existentes.</p> <p>Descubren aspectos básicos del contexto y/o dominio de aplicación.</p> <p>Proporcionan una mejor comprensión del problema.</p> <p>Útiles para la mejora de procesos.</p> <p>Mejoran la elicitación si se desarrollan las preguntas correctas.</p> <p>Utilizan preguntas libres de contexto.</p> <p>Existen diferentes variantes.</p> <p>Son sistemáticas.</p> <p>Pueden elicitar conocimiento estructurado.</p>	González y Leroy, (2011), Kausar et al. (2010), Maiden y Rugg (1996), Serna (2012), Sutcliffe (1995), Young (2002).
Cuestionarios y encuestas.	<p>Económicos y fáciles de implementar.</p> <p>Recolectan datos en masa.</p> <p>Los cuestionarios bien diseñados elicitan requisitos consistentes y precisos.</p> <p>Averiguan tendencias de opinión.</p>	Kausar et al. (2010), Tiwari, Rathore y Gupta (2012).
Observacionales (Etnografía, etnometodología, <i>contextual inquiry</i>).	<p>Descubren requisitos adicionales cuando los <i>stakeholders</i> no pueden explicarlos.</p> <p>Recolectan requisitos referentes a la usabilidad y eficiencia.</p> <p>Obtienen una amplia comprensión del dominio y una comprensión más exacta del problema.</p> <p>Son eficientes para adquirir requisitos implícitos.</p> <p>Facilitan la comprensión del contexto.</p> <p>Útiles si el equipo de desarrollo no tiene experiencia en este tipo de proyecto.</p> <p>Simples de realizar y no requieren mucho entrenamiento.</p>	Daud y Bakar (2010), Davis (1999), Kausar et al. (2010), Maiden y Rugg (1996), Pitts y Browne (2007), Serna (2012).
Análisis de protocolo.	<p>Provee un buen entendimiento de los procesos o tareas que realizan los <i>stakeholders</i>.</p> <p>Proporciona información de tareas específicas.</p>	Maiden y Rugg (1996), Tiwari, Rathore y Gupta (2012).
Metodologías ágiles.	<p>Son adaptables a requisitos cambiantes.</p> <p>Mejora la colaboración entre los <i>stakeholders</i> y desarrolladores.</p> <p>Utilizan la conversación <i>face to face</i>, lo cual genera una mejor comunicación.</p>	Tiwari, Rathore y Gupta (2012).
Prototipos.	<p>Representan gráfica y funcionalmente el sistema.</p> <p>Capturan muchos detalles de la interfaz.</p> <p>Pueden reducir el tiempo de desarrollo si son</p>	Davis (1999), Kausar et al. (2010), Maiden y Rugg (1996), Serna (2012), Sutcliffe (1995), Young (2002).

	<p>evolutivos.</p> <p>Logran una comunicación intensiva y colaboración entre los <i>stakeholders</i>.</p> <p>Útiles para aclarar requisitos confusos.</p>	
Modelado.	<p>Ayudan a los <i>stakeholders</i> a analizar sus procesos o tareas.</p> <p>Útiles para resolver conflictos entre <i>stakeholders</i>.</p> <p>Ayudan a descubrir y eliminar inconsistencias y ambigüedades en los requisitos.</p>	Tiwari, Rathore & Gupta (2012), Young (2002).
Lluvia de ideas.	<p>Genera una gran variedad de ideas.</p> <p>Recolecta la opinión de todos los <i>stakeholders</i>.</p> <p>Es creativa lo cuál ayuda a generar varios tipos de requisitos.</p> <p>Existen muchas herramientas disponibles para apoyar la ejecución de ésta técnica.</p>	Davis (1999), Kausar et al. (2010), Maiden y Rugg (1996), Tiwari, Rathore y Gupta (2012), Tsumaki y Tamai (2006), Young (2002).
<i>Workshops</i> .	<p>Proveen un conjunto completo de varios tipos de requisitos.</p> <p>Son muy útiles para elicitar requisitos de sistemas grandes y complejos.</p> <p>Útiles para resolver conflictos entre <i>stakeholders</i>.</p> <p>No son costosos.</p> <p>Ayudan a dar estructura a los requisitos.</p> <p>Son dinámicos interactivos y cooperativos.</p> <p>Ayudan a priorizar las necesidades.</p> <p>Útiles para gestionar las expectativas de los <i>stakeholders</i>.</p>	Maiden y Rugg (1996), Tiwari, Rathore y Gupta (2012), Young (2002).
JAD.	<p>Útil para resolver conflictos entre <i>stakeholders</i>.</p> <p>Provee comunicación directa con los <i>stakeholders</i>.</p> <p>Recomendado para el desarrollo de sistemas críticos y distribuidos.</p> <p>Mejora la calidad y velocidad del diseño del sistema.</p> <p>No se enfoca demasiado en resolver problemas que se relacionan con los detalles técnicos del software.</p>	Maiden y Rugg (1996), Sajid, Nayyar y Mohsin (2010), Tiwari, Rathore y Gupta (2012).
<i>Story boarding</i> .	<p>Permite que todos los <i>stakeholders</i> tengan un entendimiento común de los requisitos.</p> <p>Facilita la contribución de grandes grupos de <i>stakeholders</i></p> <p>No da la falsa impresión de un sistema ya desarrollado.</p> <p>Útiles cuando se tiene un tiempo limitado y se cuenta con bajo presupuesto.</p> <p>Posee un bajo costo de implementación y a diferencia de los prototipos esta técnica no es riesgosa.</p>	Sajid, Nayyar y Mohsin (2010), Tiwari, Rathore y Gupta (2012), Young (2002).

Casos de uso.	Ayudan a representar los requisitos funcionales del sistema. Facilitan la comunicación dentro del equipo de desarrollo.	Tiwari, Rathore & Gupta (2012), Young (2002).
Escenarios.	Tienen una manera realista de elicitar requisitos. Provee un mejor entendimiento del flujo de datos de una tarea. Útiles cuando se tiene un tiempo limitado y se cuenta con bajo presupuesto. Ayudan a prevenir defectos. Útiles para refinar requisitos. Se puede integrar con métodos estructurados para elicitar requisitos efectivamente, por ejemplo las entrevistas estructuradas.	Kausar et al. (2010), Lausen y Vinter (2001), Maiden y Rugg (1996), Sajid, Nayyar y Mohsin (2010), Seyff et al. (2009b), Sutcliffe (1995).
<i>Card sorting.</i>	Genera información respecto a la asociación y grupos de datos. Ayuda a obtener detalles del conocimiento de dominio de aplicación. Posee un formato estándar para representar el conocimiento. Se puede automatizar con la ayuda de una herramienta software.	Maiden y Rugg (1996), Tiwari, Rathore y Gupta (2012).
<i>Repertory grid.</i>	Provee una distinción entre diferentes dominios de información.	Tiwari, Rathore y Gupta (2012).

3.2.2.4. *PI4- Métricas existentes en la etapa de elicitación de requisitos*

Considerando los criterios y términos de búsqueda, 13 estudios fueron identificados para dar respuesta a la *Pregunta de Investigación 4 (PI4): ¿Qué métricas se utilizan para medir los requisitos software en la etapa de elicitación?*

Al analizar los 13 artículos que dan respuesta a esta pregunta se puede ver que la mayoría de las métricas propuestas se enfocan únicamente a dos actividades del proceso de elicitación: la comunicación y negociación.

Estos 13 estudios, se catalogaron de la siguiente forma: tres que se enfocan en métricas para los requisitos en la etapa de elicitación, seis que se enfocan en actividades del proceso de elicitación, tres de ellos consideran alguna cuestión específica dentro del proceso de elicitación, y solamente uno se enfoca al documento de requisitos como resultado del proceso de elicitación.

En el estudio de Zapata et al. (2012) se propone medir la calidad del documento de requisitos - visto como el resultado de la elicitación- tomando en cuenta la completitud y la corrección de los requisitos; estas métricas fueron adaptadas del estudio de Lloyd et al. (2002) en el que se aplicaron a una SRS. Nurmuliani, Zowghi y Powell, 2004 así como Nakatani et al. (2011) proponen medir la volatilidad de los requisitos durante la etapa de elicitación, uno de los problemas principales que se presentan en este proceso. Otro de los estudios aceptados, el de Palmer y Evans (1994), propone que por medio del uso de métricas se identifiquen requisitos que se encuentren en riesgo, es decir aquellos que no estén especificados adecuadamente. En los estudios de Dutoit y Bruegge (1998), In y Roy (2001), Barragáns et al. (2008), y Zin y Pa (2009) se proponen métricas para medir ciertos

aspectos que involucran los procesos de comunicación y negociación con los *stakeholders* y la forma en cómo se llevan a cabo estos dos procesos durante las sesiones de elicitación (Hanakawa y Obana, 2012). Otra de las tareas que involucra el proceso de elicitación es la adquisición del conocimiento del dominio de aplicación, por lo que Kaiya y Saeki (2006) proponen métricas para detectar si los requisitos son incompletos o inconsistentes tomando en cuenta la definición ontológica del dominio de aplicación. Por otro lado se encontraron dos enfoques más específicos; el primero, propuesto por Berenbach y Borotto (2006) quienes plantean métricas basadas en modelos y el de Kassou y Kjiri (2012) que utilizan métricas de seguridad. Por último también se incluye un estudio (Svensson, Host & Regnell, 2010) que desarrolla una SLR enfocada a requisitos no funcionales. Todos estos estudios son descritos a detalle en esta sección.

Como se ha expuesto antes, el proceso de elicitación de requisitos consiste en descubrir y obtener los deseos y necesidades de los *stakeholders* con respecto al software a desarrollar (Christel & Kang, 1992). Por lo que este proceso involucra la comunicación de todos los *stakeholders*, aspecto que puede generar problemas de transmisión de información, por esta razón Dutoit y Bruegge (1998) proponen métricas para obtener el grado de comunicación entre ellos. Las métricas propuestas miden aspectos de artefactos generados por software colaborativo, tales como correo electrónico, memorándum u otro tipo de mensajes. En los casos de estudio presentados en los artículos de los autores mencionados, se mide el número de mensajes entre los *stakeholders* por día durante proceso de IR, y también se centra en el uso de la técnica de casos de uso para su análisis.

Durante las reuniones entre los *stakeholders* se da un proceso de negociación para tener una visión compartida de los requisitos software por lo que In y Roy (2001) miden el grado de consenso entre dos *stakeholders* para saber cuál es el nivel de acuerdo entre ellos y lo nombran factor de consenso. El factor de consenso es calculado por una fórmula que está basada en la desviación estándar; el valor obtenido es clasificado en alguno de estos tres intervalos: completamente de acuerdo, parcialmente de acuerdo y ningún acuerdo (In & Roy, 2001). Debido a que en estas reuniones usualmente participa una gran cantidad de *stakeholders* es inevitable enfrentarse a múltiples puntos de vista, sobre todo al tratar de integrar todas estas perspectivas tratando de que la SRS sea completa y consistente (Barragáns et al., 2008). Estos autores proponen una medida para obtener el nivel de inconsistencia y también medir el grado de conflicto y la incertidumbre de los puntos de vista mientras proceso de especificación evoluciona.

Otro estudio que tiene que ver con la medición del proceso de comunicación entre los *stakeholders* es el de Zin y Pa (2009), quienes proponen la medición de cinco variables (entrada, habilidades de comunicación, personalidad, medio y procedimientos) para así obtener un indicador de los desacuerdos de comunicación entre los *stakeholders*. Cada una de estas variables posee 3 atributos medibles, usando una escala Likert del 0 al 3, donde el 0 se refiere a “fuertemente de acuerdo” y el 3 es “fuertemente desacuerdo”. El tener conocimiento de las diferencias que existen entre los *stakeholders* ayuda a tomar acciones en etapas tempranas y así asegurar que los requisitos de software producidos durante la etapa de elicitación reflejan las necesidades de los *stakeholders*.

Svensson, Host y Regnell (2010) condujeron una SLR para poder identificar, evaluar y analizar los estudios referentes a los requisitos no funcionales, también conocidos como Requisitos de Calidad (QR, por sus siglas en inglés). Los autores identifican cinco áreas importantes dentro de la gestión de QR: elicitación, métricas, dependencias, estimación de costo y priorización. La elicitación es incluida por los autores ya que es la primera etapa donde se identifican QR y por esta razón es que es importante cuantificarlos y establecer métricas. Entre estos hallazgos, destacan los siguientes:

- Diversos estudios indican que los requisitos no funcionales deben ser medibles o cuantificables; y que la medición es más difícil y esencial para los QR que para los requisitos funcionales.
- Se utiliza una mezcla de escalas para medirlos: absoluta (no hay intervalos), mínimo-máximo (establece dos límites creando un intervalo), valor máximo y valor mínimo.
- Se identificaron dos enfoques para cuantificar QR y esto se centran básicamente en que la información y propiedades que describen sean cuantificables utilizando intervalos.

Dado que en algunas técnicas de elicitación los requisitos software se discuten en reuniones entre los *stakeholders*, es importante tomar en cuenta la calidad de una reunión ya que ésta influye en la calidad de los requisitos y esto a su vez, en la baja calidad del software (Hanakawa & Obana, 2012).

Hanakawa y Obana (2012) proponen nuevas métricas para medir la calidad de las reuniones de los *stakeholders*, clasificándolas en métricas básicas y métricas avanzadas. Las métricas básicas se refieren al tiempo de inicio y fin de una intervención, quiénes participan y el tiempo de inicio y fin de un tema. Las métricas avanzadas son nueve: el promedio del tiempo de participación de un ingeniero de sistemas, promedio del tiempo de participación de un *stakeholder*, el número de veces de participación del ingeniero de sistemas, el número de veces de participación de los *stakeholders*, la razón del tiempo de participación de los *stakeholders* respecto a todo el tiempo de la discusión, la razón del tiempo de participación del ingeniero de sistemas respecto a todo el tiempo de la discusión, la razón del número de participaciones de los *stakeholders* respecto al número total de participaciones, la razón del número de participaciones del ingeniero de sistemas respecto al número total de participaciones y el máximo del número de veces de las participaciones de los *stakeholders* entre la participación de un ingeniero de sistemas. Cabe destacar que los autores hacen una separación entre los *stakeholders* y el ingeniero de sistemas; indicando que éste último juega un rol muy importante en la elicitación porque es el encargado de extraer las demandas y objetivos de los *stakeholders*, y por lo tanto debe poseer las habilidades necesarias para esta tarea. Por último, ellos muestran la utilización de estas métricas en dos proyectos de software educativo a gran escala. El primero es un software educativo al que se le agregaron dos funcionalidades, un calendario y una función de gestión. El segundo es para la creación de un software para salas de conferencias de gran escala, donde más de 200 personas asisten utilizando NINTENDO DS®. Por último los autores discuten su eficiencia concluyendo que las métricas utilizadas ayudan a predecir faltas en la especificación.

Palmer y Evans (1994) definen métricas de riesgo basadas en requisitos (i.e. identificar requisitos que se encuentren en riesgo). Los requisitos en riesgo se relacionan con los problemas que estos puedan tener. Las categorías de riesgos en los que puede situarse un requisito son: el que se utilice terminología ambigua, que existan términos redundantes, una construcción inconsistente, requisitos en conflicto, no trazables y, requisitos incompletos. El *framework* propuesto por los autores utiliza tres clases de métricas: el conteo, la razón de riesgo y el tamaño relativo. El conteo incluyen el número de requisitos en ciertas categorías de riesgo, el número de requisitos en una categoría determinada y el número de requisitos en una categoría que está en una categoría de riesgo diferente. La razón de riesgo, es la razón de los requisitos en una categoría de riesgo entre el total de requisitos. Por último, el tamaño relativo es una medida basada en texto de los requisitos pertenecientes a una categoría de riesgo; se obtiene midiendo parámetros tales como el número de objetos de datos y las funcionalidades asociadas en una categoría. El tamaño relativo provee una medida cuantitativa del esfuerzo de desarrollo relativo necesario para la implementación de los requisitos de cierta categoría.

Nurmuliani et al. (2004) se centran en identificar las causas de la volatilidad de los requisitos. Exponen que debido a que el desarrollo de software es un proceso dinámico causa que los requisitos cambien continuamente. La medida de volatilidad de los requisitos se define como la razón entre el número de requisitos cambiados (i.e., agregados, modificados o eliminados) y el número de requisitos elicitados en cierto periodo de tiempo. Esta medida ayuda a visualizar que tan estables son los requisitos en el sistema.

Kaiya y Saeki (2006) proponen una serie de métricas para medir el grado de conocimiento del dominio de aplicación del software. En primer lugar proponen hacer uso de una ontología para representar los términos y la estructura semántica del dominio del problema a resolver. En base a estas definiciones proponen usar medidas para poder estimar la calidad de los requisitos y así detectar la no completitud, la inconsistencia, las ambigüedades, o errores en las características estructurales de la ontología y los requisitos software. A continuación se describen cada una de estas definiciones:

- *Correctos*. Los requisitos que son mapeados a elementos ontológicos pueden ser considerados como requisitos apropiados para el dominio del problema.
- *Completos*. Los elementos ontológicos que no han sido mapeados a los requisitos pueden ser candidatos a ser requisitos faltantes
- *Consistencia*. Si los requisitos incluyen conceptos ontológicos que son conectados a través de relaciones contradictorias en la ontología, entonces son inconsistentes.
- *Precisión*. Si los requisitos son mapeados a algunos conceptos que no tienen relaciones semánticas, esto podría tener un múltiple significado.

Berenbach y Borotto (2006) describen un enfoque formal basado en lo que establece el Modelo de Madurez y de Capacidad Integrado (CMMI, por sus siglas en inglés) para medir, durante un proceso de desarrollo los requisitos basados en modelos. Estos autores recomiendan el uso de herramientas que apoyen el proceso de medición como RequisitePro, DOORS y Caliber RM, ya que el no utilizar alguna podría hacer que el proceso de medición consuma mucho tiempo. CMMI considera la calidad de requisitos como un pre-requisito para completar las tareas del área de proceso referente a desarrollo de requisitos del nivel 3. Debido a que éste es un enfoque basado en modelos, utiliza métricas para verificar que los modelos sean completados, estas medidas son: casos de uso no definidos, artefactos ocultos, clases no instanciadas, dependencia circular, casos de uso completos, asociaciones ilegales y relaciones entre los casos de uso. A diferencia de la captura de requisitos basada en texto, en este estudio se demostró que el enfoque basado en modelos provee una realimentación inmediata a los analistas de requisitos de la calidad de su trabajo.

Barragáns et al. (2008) proponen una métrica para conocer el nivel de inconsistencia entre los puntos de vista de varios *stakeholders*. Estos autores formalizan la definición de la inconsistencia y establece que existen diferentes niveles para ésta. Dentro los niveles de inconsistencia ellos destacan que ésta se da en cada uno de los puntos de vista y por otro lado se genera cuando se mezclan todos estos. La métrica que los autores establecen para medir el grado de inconsistencia ayuda a priorizar las cuestiones que deben resolverse de manera más urgente por lo que este indicador ayuda a descubrir los conflictos más relevantes.

Nakatani et al. (2011) destacan que la volatilidad de los requisitos indica el grado de madurez de éstos (ya que los requisitos del software no cambian una vez que estos alcanzan el 100% de madurez) y definen un método para estimar la Eficiencia de la Madurez de un Requisito (RME, por sus siglas en inglés) partiendo de la estabilidad y accesibilidad de los requisitos. La estabilidad de los requisitos se obtiene del Índice de Madurez de los Requisitos (RMI, por sus siglas en inglés.) El

RMI es obtenido dividiendo el número de requisitos que no han cambiado sobre el número total de requisitos elicitados. La Accesibilidad del Origen de los Requisitos (RSA, por sus siglas en inglés) se define como el inverso de la distancia máxima de comunicación entre el analista de requisitos y el origen de los requisitos; la distancia no es geográfica sino más bien implica las dificultades o problemas de comunicación. La RME es obtenida sumando $-0.31 + 0.27 * RMI + 0.75 * RSA$, dónde el valor máximo que se puede obtener es de 0.71, lo que significa que la fuente de requisitos es cercana y los requisitos son estables; y el valor mínimo de -0.31 quiere decir que la fuente de requisitos es lejana y los requisitos son inestables. Por último los autores exponen que con ayuda de esta métrica es posible estimar hasta qué etapas del proyecto se dejará de elicitar requisitos.

Kassou y Kjiri (2012) proponen un enfoque de elicitación de requisitos de seguridad utilizando el uso de métricas de seguridad. Esta propuesta se basa en la utilización de métricas existentes en el estándar 27002 creado por la Organización Internacional de Normalización (ISO, por sus siglas en inglés) que cubre la mayoría de aspectos de la seguridad de la información y el Modelo de Madurez y Capacidad de Ingeniería de Seguridad y Sistemas (SSE-CMM, por sus siglas en inglés) que plantea los procesos de seguridad que intervienen en un proyecto software. Estas métricas son provistas por modelos de madurez los cuáles deducen diferentes niveles de madurez basándose en varios criterios agrupados en dimensiones. En este enfoque no se describen las métricas, pero si se describe el proceso de elicitación a través de la medición.

Un estudio dentro del contexto de desarrollo distribuido es el presentado por Zapata et al. (2012) donde la efectividad de una técnica de elicitación es medida a través del uso de las métricas propuestas por Lloyd et al. (2002), pero ajustándolas a la etapa de elicitación. La métrica utilizada, denominada CDRS (Calidad del Documento de Requisitos Software) indica la calidad del documento de requisitos a través de la fórmula (1):

$$CDRS = 0.05 NAD + 0.3 RE + 0.25 RSD + 0.4 RS \quad (1),$$

donde:

- NAD (Nivel de Adecuación general del Documento), mide los aspectos referentes a la organización del documento.
- RE (porcentaje de Requisitos Evolucionados), mide el porcentaje de requisitos que se identifican como una evolución de un requisito de software básico.
- La RSD (Porcentaje de Requisitos sin Defecto), mide el porcentaje de requisitos que no poseen defectos de precisión, ambigüedad, vaguedad, etc. Estos defectos son atribuibles a deficiencias en el proceso de elicitación.
- El porcentaje de requisitos soportados (RS) indica la completitud del documento producido.

Los sub-indicadores RE y RS son los que reflejan mejor la efectividad de un proceso de elicitación por lo que tienen mayor ponderación.

En la Tabla 15 se resumen las métricas encontradas en esta SLR, así como las medidas asociadas a éstas.

Tabla 15. Métricas utilizadas en la etapa de elicitación.

Aspecto a medir	Métrica	Definición	Medida asociada	Estudios referenciados
Comunicación.	Grado de	Métrica definida para artefactos de	Frecuencia de mensajes.	Dutoit y Bruegge

	comunicación.	comunicación generados por software colaborativo.	Otro tipo de información obtenida de artefactos generados por software colaborativo.	(1998)
Negociación.	Grado de consenso.	Mide el grado de consenso entre 2 <i>stakeholders</i> .	Factor de consenso $Cf_j(m, n) = \sqrt{\sum_{1,k} (V_{mi} - V_{ni})^2 / k}$ donde, m y n son los <i>stakeholders</i> , D _j es el ítem de decisión, k es el criterio asociado para evaluar D _j , y V son los votos.	In y Roy (2001)
Comunicación.	Nivel de comunicación.	Mide la brecha de comunicación existente entre los <i>stakeholders</i> .	Se miden 5 variables Variable: entrada Indicadores: Medición de la información requerida. Características de la información. Magnitud de cambios. Variable: habilidades de comunicación Indicadores: Interpersonal. Verbal. Escrita. Variable: personalidad Indicadores: Conocimiento de dominio. Conocimiento técnico. Calidad. Variable: medio Indicadores: Información. Marco de tiempo. Retroalimentación. Variable: procedimientos Indicadores: Gestión y políticas. Formato del reporte. Estándar.	Zin y Pa (2009)
Comunicación y negociación.	Calidad de una reunión.	Mide el nivel de calidad en el que es realizada una reunión entre los <i>stakeholders</i> .	Métricas básicas: Tiempo de inicio y fin de una intervención, quiénes participan y el tiempo de inicio y fin de un tema.	Hanakawa y Obana (2012)

			<p>Métricas avanzadas:</p> <p>El promedio del tiempo de participación de un ingeniero de sistemas, promedio del tiempo de participación de un <i>stakeholder</i>, el número de veces de participación del ingeniero de sistemas, el número de veces de participación de los <i>stakeholders</i>, la razón del tiempo de participación de los <i>stakeholders</i> respecto a todo el tiempo de la discusión, la razón del tiempo de participación del ingeniero de sistemas respecto a todo el tiempo de la discusión, la razón del número de participaciones de los <i>stakeholders</i> respecto al número total de participaciones, la razón del número de participaciones del ingeniero de sistemas respecto al número total de participaciones y el máximo del número de veces de las participaciones de los <i>stakeholders</i> entre la participación de un ingeniero de sistemas.</p>	
Riesgo.	Nivel de riesgo.	Ayuda a identificar los problemas que puedan tener los requisitos.	El conteo, la razón de riesgo y el tamaño relativo de desarrollo.	Palmer y Evans (1994)
Atributo de calidad.	Volatilidad.	Indica los cambios (requisitos agregados, eliminados y modificados) y las razones de cambio de los requisitos.	Contar el número de requisitos modificados (aumentos, cambios y eliminaciones).	Ali (2006), Bokhari y Siddiqui (2011), Costello y Liu (1995), Nurmaliani et al. (2004)
	Trazabilidad.	Indica el grado en el cual el equipo de desarrollo mantiene el registro del estado de los requisitos en la matriz de trazabilidad.	Número de requisitos trazados consistentemente tanto en un nivel posterior como en el anterior, número de requisitos trazados consistentemente en el nivel más alto y en el nivel más bajo, estadísticas de conexión entre los requisitos, métricas de trazabilidad de inconsistencia y métricas de trazabilidad indefinida.	Ali (2006), Bokhari y Siddiqui (2011), Costello y Liu (1995),
	Complejidad.	Indica la completitud de todas las secciones de la SRS, que todos los requisitos estén especificados en el nivel correcto, así como su grado de descomposición.	Descomposición de requisitos, desarrollo de la especificación de requisitos y métricas de los requisitos que deben ser completados.	Ali (2006), Bokhari y Siddiqui (2011), Costello y Liu (1995)
		Indica la completitud	Número de casos de uso no definidos,	Berenbach y

		de los modelos utilizados en la etapa de elicitación de requisitos.	artefactos ocultos, clases no instanciadas, dependencia circular, casos de uso completos, asociaciones ilegales y relaciones entre los casos de uso.	Borotto (2006)
	Consistencia de la interfaz.	Indica la consistencia y completitud de la información de la interfaz en cada nivel de la especificación.	No definida.	Costello y Liu (1995)
Calidad.	Densidad de defectos.	Indica el número de defectos detectados en los requisitos.	Número de defectos detectados en los requisitos y son clasificados por tipo, criticidad y fuente.	Costello y Liu (1995)
	Densidad de faltas.	Indica las faltas que son detectadas durante y después de la ejecución de pruebas.	Número de faltas que son detectadas durante y después de la ejecución de pruebas y son clasificados por tipo, criticidad y fuente.	Costello y Liu (1995)
Proceso.	Progreso de los requisitos.	Indica todo sobre el progreso de los requisitos.	Volatilidad, trazabilidad, completitud, densidad de defectos, densidad de faltas, consistencia de la interfaz, reporte de problemas y de acciones.	Costello y Liu (1995)
Conocimiento del dominio.	Grado de conocimiento del dominio de aplicación.	Mide el nivel de conocimiento que se tiene acerca del dominio de aplicación.	En base a la utilización de ontologías se mide que los requisitos sean: correctos, completos, consistentes y precisos.	Kaiya y Saeki (2006)
Tamaño de los requisitos.	Tamaño.	Mide el tamaño de los requisitos apoyándose de la técnica de casos de uso.	Número de acciones atómicas en el flujo principal, número de acciones atómicas en el flujo alternativo y el flujo más largo entre la primera y última acción atómica.	Ali (2006), Bokhari y Siddiqui (2011)
Madurez.	Eficiencia de la madurez de un requisito.	Mide el grado de madurez de los requisitos.	$RME = -0.31 + 0.27 * RMI = 0.75 * RSA$ donde, RME es la eficiencia de madurez de los requisitos RMI es el índice de madurez de los requisitos RSA es la accesibilidad a la fuente de requisitos.	Nakatani et al. (2011)
Seguridad.	Métricas de seguridad.	Métricas que ayudan a elicitar todos los requisitos referentes a cuestiones de seguridad.	% aplicaciones clasificadas. Calidad del proceso de clasificación de aplicaciones. % aplicaciones críticas. % aplicaciones con procedimientos con planes de protección. Calidad del proceso de aplicación del plan de protección. Procedimiento de control de acceso para intercambio de información en políticas de seguridad.	Kassou y Kjiri (2012)

			<p>Uso de técnicas criptográficas estipuladas en políticas de seguridad.</p> <p>Políticas de seguridad alineadas con requisitos legales.</p> <p>Procedimientos de autorización de servicios en políticas de seguridad.</p> <p>Nb de mensajes de incidentes del control de acceso.</p> <p>% de servicios con técnicas de autenticación débiles.</p> <p>% de servicios cuyos derechos de acceso están alineados con políticas de seguridad.</p> <p>Uso de una puerta de enlace de red para restringir la conexión de servicios.</p> <p>% de mensajes restringidos por la puerta de enlace de red de acuerdo a políticas de control de acceso.</p> <p>Uso de control de enrutamiento para redes.</p> <p>% de control de enrutamiento que está alineado con políticas de seguridad.</p> <p>Uso de menú para controlar el acceso a las funciones de servicio.</p> <p>% de funciones de servicio modificadas sin control de acceso.</p> <p>Procedimientos de gestión de cambio.</p> <p>% de servicios gestionados de acuerdo a los procedimientos de gestión de cambio.</p> <p>Uso de registros de auditoría.</p> <p>Detalles de registro de auditoría.</p> <p>Sincronizar el reloj.</p>	
--	--	--	--	--

Tomando en cuenta todo lo anteriormente expuesto, se puede decir que debido a la gran variedad de problemas que surgen durante la elicitación de requisitos es necesario que existan métricas que midan las características de calidad de los requisitos obtenidos y que permitan que éstos puedan ser monitoreados y cuantificados objetivamente. Es importante mencionar que la efectividad de las técnicas de elicitación se ve reflejada en la calidad de los requisitos obtenidos, por lo que el usar ciertas técnicas de elicitación en situaciones específicas (e.g., los casos de uso se utilizan para elicitar requisitos cuando se quiere describir el flujo de interacción entre el sistema y los *stakeholders*) es posible medir algunas características de dichos requisitos.

Las métricas son utilizadas para evaluar objetivamente aspectos durante el proceso de desarrollo de software, e.g., para determinar si se conocen los requisitos, calendario, y presupuesto, para apoyar a la gestión de riesgo y tomar una acción preventiva y correctiva, etc. Todo esto debe ser tomado en cuenta ya que la calidad del software debe mantenerse desde la primera fase del desarrollo para evitar problemas posteriores que impliquen altos porcentajes de re-trabajo y por ende de retraso en el calendario o exceso en el presupuesto.

3.3. Métricas para la elicitación de requisitos: propuesta

Para poder mejorar el proceso de elicitación de requisitos es necesario hacer una evaluación del proceso en sí, para de esta forma proveer la retroalimentación a todos los *stakeholders* participantes (desarrolladores, líder de proyecto, clientes, entre muchos más) con el objetivo de tener un mejor control del proceso o bien para mejorarlo. De acuerdo con Lord Kelvin¹⁶ “*Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre*”. Por lo que el mecanismo ideal para mejorar un proceso, es la medición de productos y procesos a través de objetivos que sean medibles e impulsada por los modelos apropiados (Basili, 1993).

Ahora bien, una vez que se ha terminado la SLR, para identificar qué métricas existen en la IR -incluyendo aquellas que miden ciertos aspectos dentro del proceso de elicitación (e.g., calidad de los requisitos obtenidos, proceso de comunicación, proceso de negociación, conocimiento obtenido, calidad y efectividad del proceso de elicitación)- es necesario utilizar el paradigma *Goal-Question-Metric* (GQM), que alinea los objetivos organizacionales y de medición con las métricas disponibles.

3.3.1. Paradigma GQM

El paradigma GQM (Basili & Selby, 1984; Basili & Weiss, 1984) representa un enfoque sistemático para definir, ajustar y evaluar un conjunto de objetivos basados en las necesidades de un proyecto u organización utilizando para ello la medición. El enfoque se definió originalmente para la evaluación de defectos de un conjunto de proyectos en el entorno de la Agencia Nacional Espacial y Aeronáutica de EUA (NASA, por sus siglas en inglés) específicamente en el *Goddard Space Flight Center*. El resultado de la aplicación del paradigma es la especificación de un sistema de medición dirigida a un conjunto particular de problemas y un conjunto de reglas para la interpretación de los datos medidos.

El método GQM contiene cuatro fases: la fase de planeación, la fase de definición, la fase de recogida de datos y la fase de interpretación (Van Solingen & Berghout, 1999). En la primera fase que es la de planeación, se selecciona, define y caracteriza un proyecto para la aplicación de la medición generando como resultado un plan de proyecto. En la segunda (de definición), se llevan a cabo todas las actividades para definir formalmente el programa de medición. Dentro de esta fase se genera el plan que especifica los objetivos de medición, el conjunto de preguntas y el conjunto de métricas. El modelo resultante contiene 3 niveles:

- El nivel conceptual (*Goal*) contiene los *objetivos de medición*; los cuáles son definidos para un objeto (relacionado con productos, procesos, o recursos), especificando los objetivos de las mediciones (Caldiera & Rombach, 1994). Los *productos* son artefactos, entregables y documentos que son producidos durante el ciclo de vida del software (e.g. especificaciones, diseño, programas, pruebas, etc.). Los *procesos* se refieren a las actividades relacionadas con el desarrollo de software (e.g. especificación, análisis, codificación, etc.). Los *recursos* son ítems usados por los procesos con el objetivo de producir sus salidas (e.g. personal, hardware, software, etc.).
 - Los *objetivos* pueden ser definidos para algún objeto, por una variedad de razones, con respecto a varios modelos de calidad, desde varios puntos de vista y relativos a

¹⁶ Lord Kelvin (William Thomson), fue un físico y matemático británico que realizó importantes trabajos en el campo de la termodinámica. En 1848 Lord Kelvin estableció la escala absoluta de temperatura que lleva su nombre (EcuRed, 2014).

un entorno en particular. El objetivo es definido a través de un conjunto de parámetros en una plantilla (Basili et al., 2007). Los *parámetros* de la plantilla (véase la Tabla 16) incluyen el propósito (¿Qué objeto? y ¿Por qué), la perspectiva (¿Qué aspectos? y ¿Quién?) y las características del entorno (¿Dónde?).

Tabla 16. Plantilla para la definición de objetivos de medición (Basili et al., 2007).

	Descripción	Ejemplo
Objeto	Procesos, productos, otros modelos de experiencia.	<i>Analiza</i> el proceso de pruebas del sistema.
Propósito	Caracterización, evaluación, predicción, motivación, mejora.	<i>Con el propósito</i> de evaluación.
Enfoque	Costo, corregir, remover defectos, cambios, confiabilidad, facilidad de uso, etc.	<i>Con respecto</i> a la generación de defectos.
Perspectiva	Usuario, cliente, líder de proyecto, desarrollador, organización, etc.	<i>Desde el punto de vista</i> de la corporación.
Entorno	Factores problemáticos, factores de personas, factores de recursos, factores de procesos, etc.	<i>En el contexto</i> de la organización.

- El *propósito* incluye la definición del objeto u objetos de estudio, qué se hará y por qué. Puede incluir varios objetos pero se deben evitar objetivos complejos, dividiendo estos en objetivos más simples.
 - La *perspectiva* significa definir un ángulo o ángulos particulares de evaluación. Se puede elegir más de un modelo (por ejemplo, defectos y cambios) y más de un punto de vista (e.g. de la organización o del líder de proyecto). Se define básicamente el conjunto de personas que quieren conocer la información así como todos los aspectos de evaluación desde cierto punto de vista.
 - Por último, en el *entorno* se define el contexto de estudio y todos los aspectos del proyecto, así, esto puede ser categorizado correctamente para tener datos para futuras comparaciones en proyectos similares.
- El nivel operacional (*Question*) genera un conjunto de preguntas usadas para especificar cómo deben ser alcanzados los objetivos (Caldiera & Rombach, 1994). Las preguntas tratan de caracterizar al objeto de la medición respecto a las cuestiones de calidad seleccionadas para determinar su calidad desde el punto de vista seleccionado.
 - El nivel cuantitativo (*Metric*) contiene métricas que son recolectadas con el fin de responder la pregunta establecida en el nivel anterior (Caldiera & Rombach, 1994). Es decir, se trata de un conjunto de datos asociados con cada pregunta con el objetivo de responderlas de manera cuantitativa. Estos datos o métricas pueden ser objetivas o subjetivas. Las métricas objetivas dependen solamente del objeto que es medido y no del punto de vista del que son recolectados, por ejemplo, el número de versiones del documento, horas invertidas en una tarea, tamaño del programa. Las métricas subjetivas dependen del objeto que es medido y además del punto de vista desde el que son recolectadas, por ejemplo, legibilidad de un texto o el nivel de la satisfacción del usuario.

En la tercera fase, que es la recogida de datos se recolectan los datos de la medición. Por último en la fase de interpretación se procesan y analizan los datos recopilados para dar respuesta a las preguntas formuladas para así satisfacer los objetivos que han sido planteados (Van Solingen y Berghout, 1999).

El paradigma GQM provee un *framework* para establecer objetivos de medición y refinarlos en preguntas con el objetivo de proveer una especificación de los datos necesarios para ayudar a cumplir los objetivos. Además, tiene una estructura jerárquica y en esta estructura, los objetivos, preguntas y métricas pueden tener una relación de muchos a muchos. Este paradigma enfatiza que la medición es realizada para cumplir un propósito y las medidas que son útiles para una organización. La aplicación de GQM involucra también el desarrollo de mecanismos para la obtención de datos y para la recolección, validación y análisis de éstos. La medición crea una base de conocimiento que ayuda a mejorar la planeación, a conocer mejor el proceso y los productos generados.

3.3.2. Proceso de medición

Para asegurar que un proceso o sus productos resultantes son de calidad se deben tener indicadores que describan el proceso o producto; y para ello, se necesita llevar a cabo un proceso de medición. En el caso particular de este trabajo de tesis se sigue el proceso definido por el *Practical Software Measurement* (PSM), que provee una guía para implementar un proceso de medición de software. El PSM dirige la estructura de la información de medición usando el **Modelo de Información de Medición** y describe las actividades y tareas usando el **Modelo de Proceso de Medición**. El Modelo de Información de Medición ayuda a relacionar las necesidades de información al proceso o productos de un proyecto de software, es decir, provee una estructura para relacionar las necesidades de información, medidas y productos de información. Mientras que el Modelo de Proceso de Medición trabaja junto con el Modelo de Información de Medición para definir un *framework* que permita implementar el programa de medición apropiado en un proyecto en particular (McGarry, 2004).

En este trabajo de tesis se utiliza el paradigma GQM para definir las métricas y ligarlas a las necesidades de información, lo cual ayudará a conocer y analizar el proceso de elicitación de requisitos, i.e. evaluar la efectividad de las técnicas de elicitación, la calidad de los requisitos obtenidos y por ende la calidad de las actividades que están involucradas en este proceso. Una vez que se hayan definido las métricas se seguirá el Modelo de Proceso de Medición del PSM para su aplicación en un caso de estudio.

Sin embargo, antes de aplicar el paradigma GQM es necesario describir de las actividades (qué y cómo se debe hacer) relacionadas con el proceso de elicitación, para así poder definir las métricas correspondientes. En la sección 2.1.1 de este trabajo de tesis se identificaron tres actividades importantes dentro de la elicitación (Abran et al., 2004; Christel & Kang, 1992; Kotonya & Sommerville, 2000):

- **Identificación de los *stakeholders***, quienes son las fuentes de requisitos y proporcionan la información necesaria para conocer el problema a resolver, a entender el dominio de aplicación, y a conocer los objetivos del negocio así como sus propias necesidades y restricciones.
- **Obtención de la lista de deseos y necesidades de cada *stakeholder*** a través de la aplicación de una o más técnicas de elicitación que permitan la extracción de dicha información o conocimiento para su análisis posterior. El proceso de obtención no es una simple transferencia de conocimiento ya que en realidad involucra otras dos actividades: la comunicación y la negociación (Hofmann, 2000). Por lo que es necesario establecer una

buena comunicación entre todos los *stakeholders* para así obtener requisitos correctos, i.e. que contengan el conocimiento necesario para el desarrollo del software adecuado. Para lograr esto es imperante saber elegir el canal de comunicación adecuado y establecer las condiciones óptimas para una comunicación efectiva, ya que mientras se lleva a cabo dicha comunicación, surge el proceso de negociación (necesario para conciliar intereses y obtener los requisitos más relevantes para los objetivos de los *stakeholders*). Sin embargo, para llevar a cabo estas dos actividades es necesario considerar otra cuestión no menos importante: qué técnica(s) se debe elegir para elicitar de forma efectiva requisitos de calidad.

- **Integración, refinamiento y organización de la información obtenida** para poder determinar cuáles son las funcionalidades y limitaciones del software a desarrollar. Una vez que el proceso de elicitación termina, el producto final es una lista de deseos y necesidades de los *stakeholders*, lo que se transcribe en la primera versión de la SRS (ésta proveerá los lineamientos necesarios para desarrollar el software).

Ahora bien, haciendo un análisis de los hallazgos que dan respuesta a la PI4, se puede observar que desafortunadamente las métricas existentes no cubren todas las actividades del proceso de elicitación ni del producto final de esta etapa. En este sentido, la Tabla 17 resume las métricas encontradas y el atributo del proceso o producto que miden.

Tabla 17. Aspectos que cubren las métricas encontradas en la respuesta a la PI4.

Actividad del proceso de elicitación	Aspecto a medir	Métrica
Proceso de obtención de información.	Comunicación.	Grado de comunicación.
Proceso de obtención de información.	Negociación.	Grado de consenso.
Proceso de obtención de información.	Comunicación.	Comunicación.
Proceso de obtención de información.	Comunicación y negociación.	Calidad de una reunión.
Lista de deseos y necesidades	Riesgo de problemas.	Nivel de riesgo.
Lista de deseos y necesidades	Atributo de calidad de un requisito.	Volatilidad.
Lista de deseos y necesidades	Atributo de calidad de un requisito.	Trazabilidad.
Lista de deseos y necesidades	Atributo de calidad de un requisito.	Compleitud.
Refinamiento de la información obtenida.	Atributo de calidad.	Consistencia de la interfaz.
Lista de deseos y necesidades	Proceso.	Progreso de los requisitos.
Proceso de obtención de información.	Conocimiento del dominio.	Grado de conocimiento del dominio de aplicación.
Lista de deseos y necesidades	Tamaño de los requisitos.	Tamaño.
Lista de deseos y necesidades	Madurez de un requisito.	Eficiencia de la madurez de un requisito.
Lista de deseos y necesidades	Requisitos de seguridad.	Métricas de Seguridad.

Después de realizar este análisis, se encontró que las actividades que no son cubiertas son:

- Identificación de los *stakeholders*, e
- Integración, refinamiento y organización de la información obtenida.

Para la primera actividad, **identificación de los stakeholders**, hasta el momento, no se han encontrado métricas que ayuden a analizarla y evaluarla; por lo que para proponer dichas métricas es necesario cubrir los aspectos que se encuentran involucrados en ella.

Para la actividad de obtención de información de las fuentes de requisitos, se encontraron algunas métricas, sin embargo son limitadas; por ejemplo, las métricas propuestas para la comunicación se enfocan únicamente al canal de comunicación utilizado (e.g., correos electrónicos, etc.) y a la técnica de elicitación utilizada, centrándose en aquellas que se llevan a cabo por reuniones o consideran solamente algunos factores dentro de la comunicación; ahora bien, para el proceso de negociación sólo se considera el nivel de participación de los *stakeholders* y el grado de consenso entre ellos, restringiendo ésta última medida para aplicarla a grupos integrados por dos *stakeholders*.

Para la actividad de **integración, refinamiento y organización de la información obtenida**, no se encontraron métricas puesto que se supone que el interés se enfoca al producto resultante: la SRS. Para la SRS se proponen métricas que miden algunos de los atributos de calidad de los requisitos obtenidos: *correctos, completos, precisos, consistentes, no ambiguos, modificables, verificables, trazables, entendibles, relevantes, no redundantes y reusables*.

Dentro del entorno de la medición del proceso software, existen dos guías para la selección y aplicación de métricas dependiendo de las necesidades de información que se tengan. La primera guía es el PSM, que propone un proceso efectivo para la medición de software basada en los objetivos de información y que puede ser aplicado a todo tipo de proyectos dentro del contexto de la adquisición, desarrollo y mantenimiento de proyectos software y sistemas.

Al igual que el paradigma GQM, uno de los principios del PSM es que tanto el proceso de medición como la selección de medidas deben dirigirse por las necesidades de información del proyecto. De acuerdo con PSM, las necesidades de información pueden ser agrupadas en áreas llamadas categorías de información, las cuáles representan las principales cuestiones que deben ser gestionadas y ayudan a seleccionar las medidas apropiadas. Una vez identificadas las necesidades de información, se sigue la jerarquía establecida por GQM para seleccionar las métricas: se identifican las categorías de información, después los conceptos medibles que se pueden relacionar con preguntas, y por último las medidas.

Para el caso de la elicitación de requisitos, al ser un proceso en sí, se identifican las categorías que son aplicables a éste así como al producto resultante (la lista de deseos y necesidades) y se considera también la satisfacción de los *stakeholders*. En la Tabla 18 se muestran las medidas identificadas que cubren algunas de las necesidades de información del proceso de elicitación, sin embargo no se cubre del todo.

Tabla 18. Jerarquía de medición (McGarry et al., 2002).

Categoría de información	Conceptos medibles	Pregunta	Medidas
Calendario y progreso.	Completitud de hitos ¹⁷ .	¿Se conocen los hitos del proyecto?	Fechas de hitos.
	Camino crítico.		Tiempo de holgura.
Recursos y costo.	Esfuerzo personal.	¿El personal tiene las habilidades necesarias?	Nivel del personal.
	Desempeño financiero.	¿El proyecto esta gastando	Nivel de experiencia.
			Presupuesto.

¹⁷ Son los puntos finales de alguna actividad del proceso software, los cuales son definidos en la planeación del proyecto. Los hitos representan el fin de una etapa lógica del proyecto y pueden tener una salida formal (Sommerville, 2010).

		el presupuesto acordado y cumpliendo con el calendario?	Costo.
	Entorno y recursos de apoyo.	¿Son suficientes las instalaciones, equipo y materiales disponibles?	Tiempo disponible. Tiempo usado.
Calidad del producto (lista de deseos y necesidades).	Correcto.	¿Se identificaron problemas que pueden ser resueltos?	Defectos.
Realización del proceso.	Eficiencia del proceso.	¿El proceso es lo suficientemente eficiente para conocer los compromisos y objetivos planeados?	Productividad. Tiempo del ciclo.
	Efectividad del proceso.	¿Cuánto esfuerzo adicional es gastado debido al re-trabajo?	Esfuerzo de re-trabajo.
Satisfacción del cliente.	Comentarios de los clientes.	¿El proyecto conoce las expectativas del usuario?	Índice de satisfacción.

La segunda guía para la aplicación de medidas es el mapa de medición elaborado por el SEI del modelo CMMI versión 1.1. El CMMI, en las prácticas específicas respecto a proceso de medición, establece que primero se deben identificar las necesidades de información para así establecer los objetivos de la medición y por último, especificar las medidas en base a estos objetivos. En este sentido, Park (1996) identifica y tabula todas las medidas y actividades de medición del CMMI en su versión 1.1, el autor presenta una tabla que resume lo establecido en el modelo referente a la medición. La Tabla 19 muestra las métricas identificadas para determinar la funcionalidad de un software y otras métricas de los requisitos que se podrían aplicar a la etapa de elicitación de requisitos.

Tabla 19. Medidas identificadas del CMMI (Park, 1996).

Medida	Áreas de proceso
Número de requisitos almacenados, por categoría: Número de requisitos de seguridad. Número de requisitos de configuración del sistema. Número de requisitos de funcionalidad. Número de requisitos de fiabilidad.	Gestión de requisitos (Medición y análisis 1) – Nivel 2.
Número de defectos encontrados en los requisitos.	Definición organizacional del proceso (Actividad 5.1) – Nivel 3. Gestión cuantitativa del proceso (Actividad 4.2) – Nivel 4.
Dificultad de los defectos encontrados en los requisitos.	Definición organizacional del proceso (Actividad 5.1) – Nivel 3. Gestión cuantitativa del proceso (Actividad 4.2) – Nivel 4.
Estabilidad de los requisitos.	Gestión integrada de software (Actividad 7.2) – Nivel 3.
Número de requisitos reusados.	Gestión integrada de software (Actividad 6.3) – Nivel 3.
Tamaño estimado y actual de los cambios incorporados.	Ingeniería del producto software (Medición y análisis 2) – Nivel 3.

Número total de requisitos.	Gestión cuantitativa del proceso (Actividad 5.6) – Nivel 4.
-----------------------------	---

Después de que se han expuesto las actividades que se llevan a cabo en el proceso de elicitación de requisitos y las métricas existentes, se definen los objetivos de medición a través del paradigma GQM para entender, controlar, evaluar y mejorar este proceso.

Para la primera actividad dentro del proceso de elicitación, **identificación de los stakeholders**; se establece como necesidad principal de información el determinar si se están seleccionando adecuadamente las fuentes de requisitos necesarias. La plantilla que define este objetivo de medición se muestra en la Tabla 20.

Tabla 20. Identificación de los *stakeholders*.

	Descripción
Objeto	Analizar el proceso de identificación de <i>stakeholders</i> .
Propósito	Con el objetivo de mejorarlo e identificar a las fuentes de requisitos necesarias y relevantes.
Enfoque	Con respecto a la plantilla de Volere propuestas por Robertson y Robertson (Robertson & Robertson, 2006).
Perspectiva	Desde el punto de vista del ingeniero de requisitos.
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

En la actividad de **obtención de la lista de deseos y necesidades de cada stakeholder** se identifican dos necesidades de información: la primera hace referencia a la efectividad de las técnicas de elicitación para determinar si se han elegido las más adecuadas para el proyecto software, y la segunda, correspondiente a analizar los procesos de comunicación y negociación para mejorarlos. La Tabla 21 y 22 describen los objetivos de medición que cubren estas dos necesidades de información.

Tabla 21. Analizar la efectividad de las técnicas de elicitación.

	Descripción
Objeto	Analizar el proceso de obtención de requisitos.
Propósito	Con el propósito de medir la efectividad de las técnicas de elicitación.
Enfoque	Enfocándose a la elicitación efectiva de información.
Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Tabla 22. Mejorar el proceso de comunicación y negociación.

	Descripción
Objeto	Analizar la comunicación y negociación durante el proceso de obtención de los requisitos.
Propósito	Con el propósito de mejorar la comunicación y negociación entre los <i>stakeholders</i> .
Enfoque	Enfocándose a establecer las condiciones óptimas para su realización.

Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Por último, en la actividad de **integración, refinamiento y organización de la información que se ha obtenido**, es necesario analizar precisamente el proceso de integración y refinamiento, verificando que no se pierda información importante o que se consideren datos que son irrelevantes (véase Tabla 23).

Tabla 23. Analizar la integración y el refinamiento de la información.

	Descripción
Objeto	Analizar el proceso de integración y refinamiento de información.
Propósito	Con el propósito de verificar que la información esté completa en la lista de deseos y necesidades.
Enfoque	Con respecto a la identificación adecuada de las metas del negocio, problema a resolver, dominio de aplicación y necesidades
Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Hasta el momento se han cubierto todas las actividades que forman parte del proceso de elicitación y también del producto resultante (la lista de deseos y necesidades de los *stakeholders*). Sin embargo, falta evaluar la efectividad del proceso completo, por lo tanto se hace necesario establecer un objetivo más para cubrir esta actividad (véase Tabla 24).

Tabla 24. Evaluar la efectividad del proceso de elicitación.

	Descripción
Objeto	Analizar el proceso completo de elicitación.
Propósito	Con el propósito de evaluar su efectividad.
Enfoque	Enfocándose a analizar los defectos en los requisitos, el documento completo de la lista de deseos y necesidades, así como el esfuerzo y el tiempo invertido en esta fase de revisión.
Perspectiva	Desde el punto de vista del líder del proyecto.
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Una vez que se han definido los objetivos de medición para el proceso de elicitación, y considerando que ya existen algunas métricas propuestas para el mismo, se procede a definir las preguntas que ayudarán a alcanzar los objetivos mencionados anteriormente. En la primera columna de la Tabla 25 se listan los cuatro objetivos que son cubiertos por las métricas que han sido encontradas en los estudios que dan respuesta a la PI4, las métricas en sí se muestran en la tercera columna y son las que dan respuesta a las preguntas formuladas en la segunda columna.

Tabla 25. Métricas que han sido propuestas por otros autores para los requisitos software en la etapa de elicitación.

Objetivo (Goal)	Pregunta (Question)	Métrica (Metric)
Analizar la efectividad de las técnicas de elicitación.	¿Cómo son los requisitos, respecto a los atributos de calidad, que se han obtenido con las técnicas de elicitación aplicadas?	Calidad de la lista de deseos y necesidades.
Mejorar los procesos de comunicación y negociación.	¿Cómo se efectúa el proceso de comunicación entre los <i>stakeholders</i> ?	Grado de comunicación.
		Nivel de comunicación.
		Calidad de una reunión.
	¿Cuál es el nivel de acuerdo entre los <i>stakeholders</i> y en que condiciones se da la negociación?	Grado de consenso.
		Calidad de una reunión.
Evaluar la efectividad del proceso de elicitación.	¿Cuál es el nivel de calidad de los requisitos obtenidos?	Densidad de defectos.
		Compleitud.
	¿Qué tan estables son los requisitos?	Volatilidad.
		Eficiencia de madurez.
	¿Qué problemas han surgido al elicitar los requisitos?	Nivel de riesgo.
	¿Cómo son los requisitos obtenidos respecto al dominio de aplicación?	Métricas de calidad de un requisito (Correcto, completo, consistente y no ambiguo).
	¿Cuál es el nivel de satisfacción de los <i>stakeholders</i> ?	Consistencia de la interfaz.
		Reporte de problemas y de acciones.
	¿Qué tan completa es la lista de deseos y necesidades?	Progreso de los requisitos.
		Medición del número de páginas del documento de la lista de deseos y necesidades.
¿Cómo es la productividad del proceso de elicitación?	Tiempo estimado y real dedicado a la etapa de elicitación.	
	Esfuerzo estimado y real dedicado a la etapa de elicitación.	

Como se puede ver en la Tabla 25, las métricas existentes sólo cubren 3 de los 5 objetivos del proceso de elicitación:

- Analizar la efectividad de las técnicas de elicitación,
- Mejorar el proceso de comunicación y negociación,
- Evaluar la efectividad del proceso de elicitación.

Para las dos actividades faltantes (la identificación de *stakeholders* e integración, refinamiento y organización de la información obtenida) se generaron dos objetivos de medición (véase Tabla 20 y Tabla 23), los cuáles a su vez se desglosaron en dos y tres niveles o sub objetivos, respectivamente (véase Figuras 3.11 y 3.12).

Para el primer objetivo, *identificación de los stakeholders*, se describen los sub objetivos 1.1, 1.1.1 y 1.1.2 (véase Tablas 26, 27 y 28). Para el segundo objetivo, *analizar la integración y refinamiento de la información*, se describen los sub objetivos 2.1 y 2.2 (véase Tablas 29 y 30).

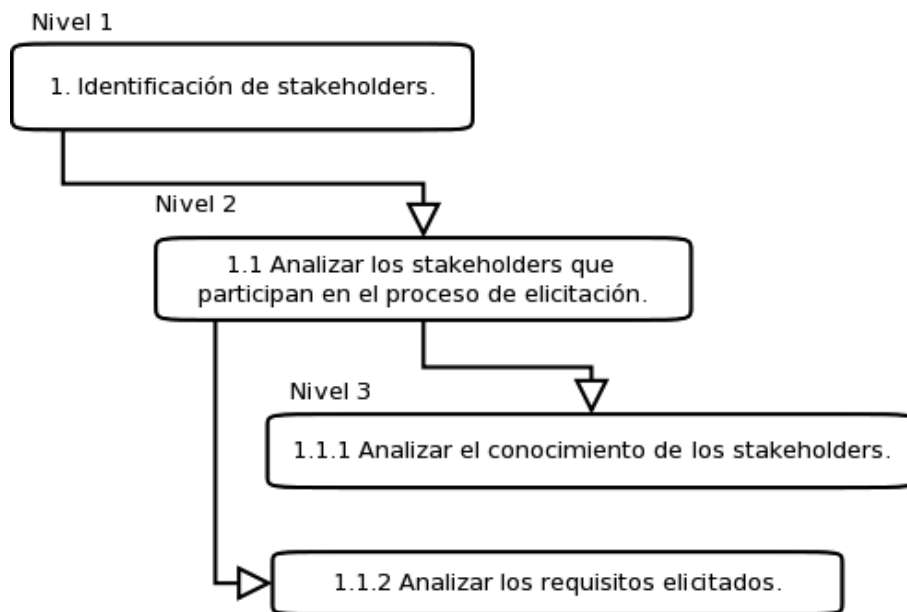


Figura 3.11. Objetivo para identificar los *stakeholders*.

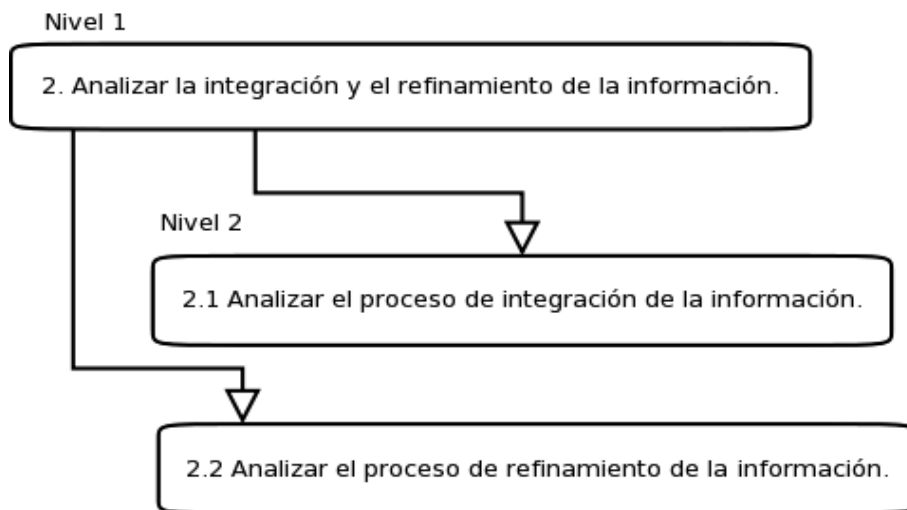


Figura 3.12. Objetivo para analizar la integración y el refinamiento de la información.

Tabla 26. Sub objetivo 1 de nivel 2 para identificar los *stakeholders*.

	Descripción
Objeto	Analizar qué <i>stakeholders</i> participan en el proceso de elicitación.
Propósito	Con el objetivo de seleccionar las fuentes de requisitos necesarias y relevantes.
Enfoque	Con respecto a la plantilla de Volere (Robertson & Robertson, 2006).
Perspectiva	Desde el punto de vista del ingeniero de requisitos.

Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.
----------------	--

Tabla 27. Sub objetivo 1 de nivel 3 para identificar los *stakeholders*.

	Descripción
Objeto	Analizar el conocimiento de los <i>stakeholders</i> .
Propósito	Con el objetivo de evaluar si son relevantes y necesarios para el proyecto a desarrollar.
Enfoque	Con respecto a la plantilla de Volere (Robertson & Robertson, 2006).
Perspectiva	Desde el punto de vista del ingeniero de requisitos.
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Tabla 28. Sub objetivo 2 de nivel 3 para identificar los *stakeholders*.

	Descripción
Objeto	Analizar los requisitos elicitados.
Propósito	Con el objetivo de verificar que las fuentes de requisitos han sido elegidas adecuadamente.
Enfoque	Con respecto a la completitud de información.
Perspectiva	Desde el punto de vista del ingeniero de requisitos.
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Tabla 29. Sub objetivo 1 de nivel 2 para analizar la integración y el refinamiento de la información.

	Descripción
Objeto	Analizar el proceso de integración de información.
Propósito	Con el propósito de verificar que se tomen en cuenta todos los deseos y necesidades de los <i>stakeholders</i> que estén alineados con el objetivo del proyecto.
Enfoque	Con respecto a la resolución adecuada de conflictos.
Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

Tabla 30. Sub objetivo 2 de nivel 2 para analizar la integración y el refinamiento de la información.

	Descripción
Objeto	Analizar el proceso de refinamiento de información.
Propósito	Con el propósito de verificar que la información en la lista de deseos y necesidades esté completa.
Enfoque	Con respecto a la identificación adecuada de las metas del negocio, problema a resolver, dominio de aplicación así como deseos, necesidades y restricciones de los <i>stakeholders</i> .
Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo esté integrado por menos de 10 personas.

En este sentido, las Tablas 31 y 32 muestran las preguntas que han sido formuladas para cada uno de los objetivos faltantes.

Tabla 31. Métricas propuestas para el objetivo de identificación de *stakeholders*.

Objetivo (<i>Goal</i>)	Pregunta (<i>Question</i>)	Métrica (<i>Metric</i>)
Nivel 2 – Analizar los <i>stakeholders</i> que participan en el proceso de elicitación.	¿Cuántos <i>stakeholders</i> son considerados para la elicitación de requisitos?	Número de <i>stakeholders</i> que han participado en la elicitación (por categoría y en total).
	¿Cuántos <i>stakeholders</i> propusieron requisitos relevantes y prioritarios para el proyecto?	Número de <i>stakeholders</i> que propusieron requisitos acorde con el objetivo del proyecto.
		Número de requisitos propuestos por cada <i>stakeholder</i> .
		Número de requisitos definitivos ¹⁸ propuestos por cada <i>stakeholder</i> .
	¿Cuántos <i>stakeholders</i> propusieron requisitos necesarios o urgentes para el proyecto?	Número de <i>stakeholders</i> que propusieron requisitos urgentes para el proyecto.
		Número de requisitos propuestos por cada <i>stakeholder</i> .
Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .		
Nivel 3 – Analizar el conocimiento de los <i>stakeholders</i> .	¿Qué tan relevantes son los <i>stakeholders</i> ?	Número de <i>stakeholders</i> que conocen el dominio de aplicación del proyecto a desarrollar.
		Número de <i>stakeholders</i> que tienen experiencia sobre el problema a resolver.
		Número de <i>stakeholders</i> que conocen los objetivos del negocio.
Nivel 3 – Analizar los requisitos elicitados.	¿Se están elicitando todas las perspectivas del sistema?	Número total de categorías de <i>stakeholders</i> que han participado en la elicitación.
		Número de versiones de cada requisito elicitado.
		Número de requisitos elicitados por cada <i>stakeholder</i> .
	¿Se han elicitado todos los deseos y necesidades de todos los <i>stakeholders</i> participantes?	Número de requisitos que se especificaron completamente.
		Número de requisitos revisados para estar acordes con los objetivos (proyecto y negocio).
		Número de cuestiones sin resolver o

¹⁸ De acuerdo con Hull et al. (2011), un requisito es considerado como definitivo cuando ha sido acordado y aceptado por todos los *stakeholders*.

	¿Se han elicitado los requisitos adecuados?	faltantes en el documento.
		Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades ya analizada.
		Número total de requisitos elicitados.
		Número de técnicas de elicitación que se utilizaron.
		Número de <i>stakeholders</i> que participaron en cada técnica de elicitación.
		Número de requisitos que se elicitaron por cada técnica usada.

Tabla 32. Métricas propuestas para el objetivo de identificación de *stakeholders* necesarios y relevantes.

Objetivo (<i>Goal</i>)	Pregunta (<i>Question</i>)	Métrica (<i>Metric</i>)
Nivel 2 - Analizar el proceso de integración de la información.	¿Cuántos puntos de vista por cada <i>stakeholder</i> serán integrados a la lista de deseos y necesidades?	Número de diferentes versiones de cada requisito (incluyendo su origen).
	¿En que requisitos no se pudo llegar a un acuerdo?	Número de requisitos en conflicto.
Nivel 2 - Analizar el proceso de refinamiento de la información.	¿Cuántas veces se ha modificado la lista de deseos y necesidades?	Número de versiones de la lista de deseos y necesidades.
		Número de requisitos agregados.
		Número de requisitos modificados.
		Número de requisitos eliminados.
		Número total de requisitos elicitados.

Para cada uno de los objetivos descritos anteriormente se definen las medidas siguiendo la plantilla del *constructor de medición*¹⁹ definida por el PSM (véase Tabla 33).

Tabla 33. Constructor de medición (McGarry et al., 2002).

Constructor de medición	
Necesidad de información	<Son las necesidades de información que se relacionan directamente con los objetivos de medición. >
Categoría de información	<PSM define 7 categorías de información: calendario y progreso, recursos y costo, tamaño y estabilidad del producto, calidad del producto, realización del proceso, efectividad de la tecnología y satisfacción del cliente. >
Concepto medible	<PSM provee una lista de conceptos medibles que son agrupados en cada una de las categorías de información. >

¹⁹Un *constructor de medición* es una estructura detallada que relaciona lo que debe ser medido respecto a una necesidad específica de información; también describe cómo se cuantifican los atributos software y cómo se transforman en indicadores para el apoyo a la toma de decisiones. El constructor de medición debe implementarse en la práctica a través de un proyecto software en particular y debe satisfacer las necesidades de información (McGarry et al., 2002).

Entidades relevantes	<Son las entidades del software que serán medidas, incluye procesos, productos, proyectos y recursos. >
Atributos	<Es una propiedad que caracteriza una entidad software, ésta puede medirse de forma cualitativa o cuantitativa. >
Medidas base	<Es una medida de un sólo atributo definido por un método específico de medición. Es funcionalmente independiente de otras medidas y captura información. >
Método de medición	<Es una secuencia lógica de operaciones que se describen genéricamente y son usadas para cuantificar un atributo respecto a una escala especificada. >
Tipo de método	<El tipo de método de medición depende de la naturaleza de las operaciones usadas para cuantificar un atributo en particular. Existen dos tipos de método: <ul style="list-style-type: none"> ▪ Subjetivo. La cuantificación involucra el juicio y la valoración humana. ▪ Objetivo. Cuantificación basada en reglas tales como el conteo, y que pueden ser implementadas por una persona o ser automatizadas.>
Escala	<Una escala es un conjunto ordenado de valores continuos o discretos, o un conjunto de categorías a las cuales se mapea un atributo. La escala define el rango de posibles valores que pueden ser producidos al ejecutar un método de medición. >
Tipo de escala	<El tipo de escala depende de la relación entre los valores y la escala. Existen cuatro tipos de escalas: <ul style="list-style-type: none"> ▪ Proporción. Nivel de medida en el que existe un cero absoluto y en el que distancias iguales corresponden a iguales cantidades del atributo. ▪ Intervalo. Son datos numéricos para los cuáles distancias iguales corresponden a iguales cantidades del atributo sin el uso del 0. ▪ Ordinal. Los valores se clasifican en categorías entre las cuáles se da una relación de orden. ▪ Nominal. Es el nivel más básico, ya que sólo se utilizan los números para clasificar, no para cuantificar u ordenar las categorías.>
Unidad de medida	<La unidad de medida es una cantidad particular que ha sido definida y adoptada por convención. >
Medida derivada	<Una medida derivada es una medida o cantidad que es definida como una función de dos o más medidas base y/o derivadas. >
Función de Medición	<Es un algoritmo o cálculo realizado para combinar dos o más valores de medidas base y/o derivadas. >
Indicador	<Un indicador es una medida que provee una estimación o evaluación de atributos específicos derivados de un modelo de análisis relacionado con las necesidades de información definidas.>
Modelo de análisis	<Es un algoritmo o cálculo que involucra dos o más medidas base y/o derivadas con un criterio de decisión asociado. El modelo de análisis está basado en el entendimiento, la suposición y la relación esperada entre lo medido y su comportamiento sobre el tiempo.>
Criterio de decisión	<Son los umbrales numéricos, objetivos y límites usados para determinar alguna acción para ciertos resultados. El criterio de decisión ayuda a interpretar los resultados de medición y puede estar basado en un entendimiento conceptual del comportamiento esperado o calculado. >

Los constructores de medición para cada uno de los objetivos se muestran en la Tabla 34 y 35 respectivamente. Para cada una de las medidas e indicadores se han creado plantillas de recolección de la información necesaria, las cuales se describen detalladamente en el Anexo C de esta tesis.

Tabla 34. Constructor de medición para identificación de los *stakeholders*.

Constructor de medición	
Necesidad de información	Analizar el proceso de identificación de los <i>stakeholders</i> con el propósito de mejorarlo e identificar las fuentes de requisitos necesarias y relevantes.
Categoría de información	Realización del proceso.
Concepto medible	Efectividad del proceso.
Entidades relevantes	<ol style="list-style-type: none"> 1. <i>Stakeholders</i>. 2. Deseos y necesidades (requisitos).
Atributos	<ol style="list-style-type: none"> 1. Cantidad, conocimiento de dominio de aplicación, conocimiento del contexto del problema y conocimiento de los objetivos y metas del negocio. 2. Completitud y cambios.
Medidas base	<ol style="list-style-type: none"> 1. Número de <i>stakeholders</i> participantes (por categoría). 2. Número de requisitos acordados con el objetivo del proyecto, por cada <i>stakeholder</i>. 3. Número de <i>stakeholders</i> que propusieron requisitos urgentes para el proyecto. 4. Número de requisitos necesarios propuestos por cada <i>stakeholder</i>. 5. Número de requisitos definitivos propuestos por cada <i>stakeholder</i>. 6. Número de <i>stakeholders</i> que conocen el dominio de aplicación. 7. Número de <i>stakeholders</i> que conocen el problema a resolver. 8. Número de <i>stakeholders</i> que conocen los objetivos de negocio. 9. Número total de categorías de <i>stakeholders</i>. 10. Número de versiones diferentes de cada uno de los requisitos. 11. Número de requisitos elicitados por cada <i>stakeholder</i>. 12. Número de requisitos que se especificaron completamente. 13. Número de requisitos revisados para estar acordados con los objetivos (proyecto y negocio). 14. Número de cuestiones sin resolver o faltantes en el documento. 15. Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades ya analizada. 16. Número total de requisitos elicitados. 17. Número de técnicas de elicitación utilizadas. 18. Número de <i>stakeholders</i> que participaron en cada técnica de elicitación. 19. Número de requisitos que se elicitaron por cada técnica usada.
Método de medición (por medida)	<ol style="list-style-type: none"> 1. Contabilizar el número <i>stakeholders</i>, por categorías. 2. Contabilizar el número de requisitos acordados con el objetivo del proyecto, propuestos por cada <i>stakeholder</i>. 3. Contabilizar el número de <i>stakeholders</i> que propusieron requisitos urgentes para el proyecto. 4. Contabilizar el número de requisitos necesarios propuestos por cada <i>stakeholder</i>. 5. Contabilizar el número de requisitos definitivos propuestos por cada <i>stakeholder</i>. 6. Contabilizar el número de <i>stakeholders</i> que conocen el dominio de aplicación. 7. Contabilizar el número de <i>stakeholders</i> que conocen el problema a resolver. 8. Contabilizar el número de <i>stakeholders</i> que conocen los objetivos de negocio. 9. Contabilizar el número total de categorías de <i>stakeholders</i>. 10. Contabilizar el número de versiones diferentes de cada uno de los requisitos.

	<ol style="list-style-type: none"> 11. Contabilizar el número de requisitos elicitados por cada <i>stakeholder</i>. 12. Contabilizar el número de requisitos que se especificaron completamente. 13. Contabilizar el número de requisitos revisados para estar acordes con los objetivos (proyecto y negocio). 14. Contabilizar el número de cuestiones sin resolver o faltantes en el documento. 15. Contabilizar el número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades ya analizada. 16. Contabilizar el total de requisitos elicitados. 17. Contabilizar el número de técnicas de elicitación utilizadas. 18. Contabilizar el número de <i>stakeholders</i> que participaron en cada técnica de elicitación. 19. Contabilizar el número de requisitos que se elicitaron por cada técnica usada.
Tipo de método (por medida)	<ol style="list-style-type: none"> 1. Objetivo. 2. Objetivo. 3. Objetivo. 4. Objetivo. 5. Objetivo. 6. Objetivo. 7. Objetivo. 8. Objetivo. 9. Objetivo. 10. Objetivo. 11. Objetivo. 12. Objetivo. 13. Objetivo. 14. Objetivo. 15. Objetivo. 16. Objetivo. 17. Objetivo. 18. Objetivo. 19. Objetivo.
Escala (por medida)	<ol style="list-style-type: none"> 1. Enteros, de 0 a infinito²⁰. 2. Enteros, de 0 a infinito. 3. Enteros, de 0 a infinito. 4. Enteros, de 0 a infinito. 5. Enteros, de 0 a infinito. 6. Enteros, de 0 a infinito. 7. Enteros, de 0 a infinito. 8. Enteros, de 0 a infinito. 9. Enteros, de 0 a infinito. 10. Enteros, de 0 a infinito. 11. Enteros, de 0 a infinito.

²⁰ De acuerdo a los argumentos de McGarry et al. (2002), es necesario establecer un intervalo de valores dentro de los cuáles un atributo puede ser mapeado, para el caso de los números enteros se considera de 0 a infinito.

	<ul style="list-style-type: none"> 12. Enteros, de 0 a infinito. 13. Enteros, de 0 a infinito. 14. Enteros, de 0 a infinito. 15. Enteros, de 0 a infinito. 16. Enteros, de 0 a infinito. 17. Enteros, de 0 a infinito. 18. Enteros, de 0 a infinito. 19. Enteros, de 0 a infinito.
Tipo de escala (por medida)	<ul style="list-style-type: none"> 1. Proporción. 2. Proporción. 3. Proporción. 4. Proporción. 5. Proporción. 6. Proporción. 7. Proporción. 8. Proporción. 9. Proporción. 10. Proporción. 11. Proporción. 12. Proporción. 13. Proporción. 14. Proporción. 15. Proporción. 16. Proporción. 17. Proporción. 18. Proporción. 19. Proporción.
Unidad de medida (por medida)	<ul style="list-style-type: none"> 1. Número de <i>stakeholders</i> participantes (por categoría). 2. Número de requisitos acordados con el objetivo del proyecto, por cada <i>stakeholder</i>. 3. Número de <i>stakeholders</i> que propusieron requisitos urgentes para el proyecto. 4. Número de requisitos necesarios propuestos por cada <i>stakeholder</i>. 5. Número de requisitos definitivos propuestos por cada <i>stakeholder</i>. 6. Número de <i>stakeholders</i> que conocen el dominio de aplicación. 7. Número de <i>stakeholders</i> que conocen el problema a resolver. 8. Número de <i>stakeholders</i> que conocen los objetivos de negocio. 9. Número total de categorías de <i>stakeholders</i>. 10. Número de versiones diferentes de cada requisitos. 11. Número de requisitos elicitados por cada <i>stakeholder</i>. 12. Número de requisitos que se especificaron completamente. 13. Número de requisitos revisados para estar acordados con los objetivos (proyecto y negocio). 14. Número de faltas por requisito. 15. Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades. 16. Número total de requisitos elicitados. 17. Número de técnicas de elicitación utilizadas.

	18. Número de <i>stakeholders</i> que participaron en cada técnica de elicitación. 19. Número de requisitos que se elicitaron por cada técnica usada.
Medida derivada	1. Total de <i>stakeholders</i> que participaron en la elicitación. 2. Frecuencia con la que se elicitaron requisitos por cada <i>stakeholder</i> .
Función de Medición	1. Sumar el número de <i>stakeholders</i> por cada categoría. $T_s = \sum_{i=1}^n s_i \text{ donde,}$ <p>T_s es el total de <i>stakeholders</i>, s_i es el número de <i>stakeholders</i> por cada categoría i, y n es el número total de categorías.</p> 2. Dividir el número de <i>stakeholders</i> que han participado en la elicitación entre el número de días o semanas que ha durado este proceso. $f_e = \frac{k}{t} \text{ donde,}$ <p>f_e es la frecuencia con la que los <i>stakeholders</i> han participado en la elicitación, k es el número de <i>stakeholders</i> que participaron en la elicitación, y t es el tiempo que ha durado la elicitación.</p>
Indicador	1. Porcentaje de <i>stakeholders</i> que participan en la elicitación, es decir cuántos requisitos definitivos fueron propuestos por cada <i>stakeholder</i> y cuántos se obtuvieron al final. 2. Versiones de los requisitos elicitados. 3. Completitud de los requisitos elicitados.
Modelo de análisis	Se espera que se eliciten los requisitos de los <i>stakeholders</i> más relevantes para el sistema, sin que se desechen otros puntos de vista que son importantes y siempre verificando que los requisitos estén completos.
Criterio de decisión	1. Analizar si el número de requisitos elicitados es menor al número de <i>stakeholders</i> participantes, si no fuera el caso se debe aplicar otra técnica de elicitación para elicitar nuevamente más requisitos. 2. Analizar el tiempo de participación de cada <i>stakeholder</i> en las sesiones de elicitación respecto al tiempo total de este proceso con el objetivo de verificar que cada <i>stakeholder</i> tuvo asignado un tiempo apropiado para expresar sus deseos y necesidades. Considerar la importancia que tienen los <i>stakeholders</i> para el proyecto para así determinar si ésta es proporcional al nivel de participación de cada uno en el proceso de elicitación. 3. Las funcionalidades del sistema indicarán el nivel de completitud de los requisitos elicitados. Es decir, que los requisitos no deben tener información que todavía no haya sido determinada, o datos que no se hayan especificado completamente o bien, que hagan referencia a objetos que no estén descritos. Si es así, esto indicará que se deben volver a elicitar para complementar la información de los requisitos.

Tabla 35. Constructor de medición para analizar la integración y el refinamiento de la información.

Constructor de medición	
Necesidad de información	Analizar el proceso de integración y refinamiento de información con el propósito de verificar que la información esté completa.
Categoría de información	Calidad del producto.
Concepto medible	Corrección, exactitud (<i>correctness</i>).

Entidades relevantes	<ol style="list-style-type: none"> 1. Lista de deseos y necesidades. 2. Requisitos.
Atributos	<ol style="list-style-type: none"> 1. Tamaño y cambios. 2. Cambios e inconsistencias.
Medidas base	<ol style="list-style-type: none"> 1. Número de las versiones de los requisitos que han sido elicitados. 2. Número de requisitos en conflicto. 3. Número de versiones de la lista de deseos y necesidades analizada. 4. Número de requisitos agregados. 5. Número de requisitos modificados. 6. Número de requisitos eliminados.
Método de medición (por medida)	<ol style="list-style-type: none"> 1. Contabilizar el número de las versiones de los requisitos que han sido elicitados. 2. Contabilizar el número de requisitos en los que no se haya llegado a un acuerdo. 3. Contabilizar el número de versiones de la lista de deseos y necesidades analizada. 4. Contabilizar el número de requisitos agregados. 5. Contabilizar el número de requisitos modificados. 6. Contabilizar el número de requisitos eliminados.
Tipo de método (por medida)	<ol style="list-style-type: none"> 1. Objetivo. 2. Objetivo. 3. Objetivo. 4. Objetivo. 5. Objetivo. 6. Objetivo.
Escala (por medida)	<ol style="list-style-type: none"> 1. Enteros, de 0 a infinito. 2. Enteros, de 0 a infinito. 3. Enteros, de 0 a infinito. 4. Enteros, de 0 a infinito. 5. Enteros, de 0 a infinito. 6. Enteros, de 0 a infinito.
Tipo de escala (por medida)	<ol style="list-style-type: none"> 1. Proporción. 2. Proporción. 3. Proporción. 4. Proporción. 5. Proporción 6. Proporción
Unidad de medida (por medida)	<ol style="list-style-type: none"> 1. Número de versiones diferentes de cada uno de los requisitos. 2. Número de requisitos en conflicto. 3. Número de versiones de la lista de deseos y necesidades. 4. Número de requisitos modificados. 5. Número de requisitos eliminados. 6. Número de requisitos eliminados.
Medida derivada	Número total de requisitos elicitados.
Función de Medición	Cuando se tengan contabilizados los requisitos que ya estén especificados se sumarán los que se han agregado y se restarán los requisitos eliminados.

	$R_T = R_B + R_A - R_E$, donde R_B son los requisitos que ya están especificados, R_A los requisitos agregados y R_E los requisitos eliminados.
Indicador	<ol style="list-style-type: none"> Número de requisitos en conflicto al integrar la información. Número de veces que se ha refinado cada requisito y por ende la lista de los mismos.
Modelo de análisis	Se deben considerar todos los puntos de vista de los <i>stakeholders</i> (versiones de los requisitos) al integrar la información obtenida de ellos, por lo cuál también se debe verificar los desacuerdos existentes entre ellos. Así mismo, se debe contabilizar el número de versiones del documento de deseos y necesidades para saber cuantas veces se ha refinado la información y el número total de requisitos aceptados, modificados o eliminados.
Criterio de decisión	<ol style="list-style-type: none"> Si no se consideran todas las perspectivas de los <i>stakeholders</i> los requisitos elicitados estarán incompletos; por esto es muy importante que los requisitos sean revisados y aprobados por todos los <i>stakeholders</i> para que exista certidumbre de que la información esté completa y en un futuro no se produzcan cambios en la lista de deseos y necesidades que afecten el desarrollo software. El hecho de que no existan muchas versiones de la lista de deseos y necesidades puede indicar que no se está considerando información irrelevante o perdiendo detalles importantes de los objetivos del proyecto, por lo que será necesario refinar más la misma.

3.3.2.1. Definición de procedimientos

Una vez que se han determinado las métricas que darán respuesta a las preguntas formuladas, y que a su vez satisfacen los objetivos de medición para el proceso de elicitación; se deben definir y especificar cada una de las medidas que se han formulado con el propósito de poder implementarlas en un proyecto software.

Primero se hará la definición de cada una de las medidas, para especificar qué es lo que se desea obtener del proceso de medición; después se describirán los procedimientos de recolección para saber cómo hacer las mediciones, es decir, los pasos a seguir para recoger los datos. Ahora bien, cuando ya se realizó la medición, los datos obtenidos se deben registrar en una plantilla o formulario y también se debe definir dónde se almacenará esta información. Cuando se han recolectado los datos de la medición se procede a analizar esta información que ha sido obtenida, por lo cual es necesario definir los métodos de análisis que se aplicarán; además se debe especificar cómo se comunicarán los resultados a los interesados del proceso de medición y cómo se realimentara este último.

3.3.2.1.1. Definición de medidas

Las medidas que se han identificado en la sección anterior son definidas en la Tablas 36 y 38. La Tabla 36 describe las medidas base, su unidad de medida y el rango en el que es deseable que se encuentre el valor nominal de cada una de ellas. La Tabla 38 describe las medidas derivadas, su unidad de medida y el rango en el que es deseable que se ubique el valor nominal de éstas. Por último la Tabla 37 describe las medidas del entorno, que ayudan a interpretar y a caracterizar las medidas anteriores dentro del entorno en el que se está realizando la medición, estas medidas no fueron definidas para satisfacer los objetivos de medición pero ayudarán a contextualizar las medidas base y derivadas.

Tabla 36. Medidas primarias.

Medida	Definición	Unidad de medida	Rango
Diversidad de los <i>stakeholders</i> participantes.	Es el número de <i>stakeholders</i> que se han identificado por cada categoría utilizando la plantilla de Volere (Robertson & Robertson, 2006).	Número de <i>stakeholders</i> participantes (por categoría).	1-3
Nivel de importancia de los requisitos.	El número de requisitos acordes con el objetivo que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos acordes con el objetivo del proyecto, por cada <i>stakeholder</i> .	1-50% del total.
Nivel de demanda de los requisitos propuestos por cada <i>stakeholder</i>	Es el número de <i>stakeholders</i> que se han identificado por cada categoría (utilizando la plantilla de Volere (Robertson & Robertson, 2006) y que han propuesto requisitos que deben ser atendidos inmediatamente (su demanda es importante y el tiempo de atención es crítico).	Número de <i>stakeholders</i> que propusieron requisitos urgentes o necesarios para el proyecto.	1-2 por categoría.
Nivel de relevancia de los requisitos.	El número de requisitos necesarios para el proyecto que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos necesarios propuestos por cada <i>stakeholder</i> .	1-50% del total.
Nivel de acuerdo entre los <i>stakeholders</i> .	El número de requisitos que han sido aceptados por todos los participantes.	Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .	80%-100% del total
Conocimiento del dominio de aplicación por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento referente al dominio de aplicación.	Número de <i>stakeholders</i> que conocen el dominio de aplicación.	1-Número total de <i>stakeholders</i> .
Conocimiento del problema por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento del problema a resolver.	Número de <i>stakeholders</i> que conocen el problema a resolver.	1-Número total de <i>stakeholders</i> .
Conocimiento de los objetivos del negocio por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento de los objetivos del negocio.	Número de <i>stakeholders</i> que conocen los objetivos de negocio.	1-Número total de <i>stakeholders</i> .
Categorías de los <i>stakeholders</i> .	Es el número de categorías en la que los <i>stakeholders</i> participantes han sido caracterizados.	Número total de categorías de <i>stakeholders</i> .	1-14
Nivel de conflicto entre los <i>stakeholders</i> .	Es el número de versiones existentes para cada requisito por cada una de las versiones de las listas de deseos y necesidades.	Número de versiones diferentes de cada uno de los requisitos.	1-Número total de <i>stakeholders</i> .
Productividad de cada <i>stakeholder</i> .	El número total de requisitos definitivos que se han elicitado por cada uno de los <i>stakeholders</i> .	Número de requisitos elicitados por cada <i>stakeholder</i> .	1-100%
Completitud de los requisitos.	El número de requisitos que están completos, es decir bien especificados (no tienen detalles que estén por ser determinados).	Número de requisitos que se especificaron completamente.	80%-100%
Requisitos correctos.	Es el número de requisitos que fueron revisados para que estuvieran acordes con los objetivos	Número de requisitos revisados para estar	80%-100%

	del proyecto y negocio.	acordes con los objetivos (proyecto y negocio).	
Densidad de omisiones en la lista de deseos y necesidades.	Es el número de cuestiones omisas en el documento que contiene la lista de deseos y necesidades: <ul style="list-style-type: none"> ▪ Datos sin especificar. ▪ Información que este pendiente o por determinar. ▪ Entidades u objetos que son referenciados pero no son descritos. 	Número de faltas por requisito.	Ninguna
Nivel de aprobación por parte de los <i>stakeholders</i> de la lista de deseos y necesidades.	Es el número de <i>stakeholders</i> que ha aprobado y firmado alguna versión de la lista de deseos y necesidades.	Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades.	Número total de <i>stakeholders</i>
Tamaño de la lista de deseos y necesidades.	Se incluirá el número total de requisitos que fueron elicitados: <ul style="list-style-type: none"> ▪ Requisitos modificados. ▪ Requisitos agregados. ▪ Requisitos eliminados. 	Número total de requisitos elicitados.	Número total de <i>stakeholders</i> -infinito.
Eficacia de la aplicación de técnicas de elicitación.	Es el número de técnicas aplicadas a lo largo del proceso de elicitación.	Número de técnicas de elicitación utilizadas.	1-Total de técnicas existentes.
	Es la cantidad de <i>stakeholders</i> que participaron en cada técnica de elicitación.	Número de <i>stakeholders</i> que participaron en cada técnica de elicitación.	1-Número total de <i>stakeholders</i> .
	El número de requisitos que fueron elicitados por cada técnica de elicitación utilizada.	Número de requisitos que se elicitaron por cada técnica usada.	1-99% del total.
Nivel de desacuerdo entre los <i>stakeholders</i>	Es el número de los requisitos que presenten alguno de los siguientes problemas: <ul style="list-style-type: none"> ▪ Diferentes versiones por cada <i>stakeholder</i>. ▪ El requisito no haya sido especificado completamente por algún problema entre los <i>stakeholders</i>. 	Número de requisitos en conflicto.	0-20% del total
Nivel de refinamiento de la lista de deseos y necesidades.	Número de las versiones existentes de la lista de deseos y necesidades.	Número de versiones de la lista de deseos y necesidades.	1-Número total de técnicas de elicitación.
Volatilidad de los requisitos.	Número de requisitos que fueron agregados a la versión de la lista de deseos y necesidades que se este analizando.	Número de requisitos agregados.	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.
	Número de requisitos que fueron modificados en la versión de la lista de deseos y necesidades que se este analizando.	Número de requisitos modificados.	1-Número de <i>stakeholders</i> que participaron en la

			elicitación por esta ocasión.
	Número de requisitos que fueron eliminados de la versión de la lista de deseos y necesidades que se esté analizando.	Número de requisitos eliminados.	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.

Tabla 37. Medidas del entorno.

Medida	Definición	Unidad de medida	Rango
Nivel de experiencia del elicitor.	Estimación subjetiva del nivel de experiencia del elicitor.	Muy alta, alta, regular, baja, muy baja.	Regular-muy alta.
Nivel de colaboración de los <i>stakeholders</i> .	Estimación subjetiva del nivel de colaboración de los <i>stakeholders</i> .	Muy buena, buena, regular, mala, muy mala.	Regular-muy alta.
Participación negativa de los <i>stakeholders</i> .	Es el porcentaje de <i>stakeholders</i> que ha presentado poca colaboración y accesibilidad durante el proceso de elicitación.	Porcentaje de <i>stakeholders</i> .	1-99% del total

Tabla 38. Medidas de ejecución del proyecto.

Medida	Definición	Unidad de medida	Rango
Participación de los <i>stakeholders</i> .	Número de todos los <i>stakeholders</i> participantes durante todo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron activamente durante la elicitación	1-Total de <i>stakeholders</i> identificados al inicio.
Frecuencia de elicitación de requisitos	Indica el tiempo en el que se ha llevado a cabo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron en la elicitación por día.	15%-25% del total de <i>stakeholders</i> identificados.
Tamaño de la lista de deseos y necesidades.	El número de requisitos que se han elicitado en la última versión de la lista de deseos y necesidades.	Número de requisitos elicitados.	Número total de <i>stakeholders</i>

3.3.2.1.2. Definir métodos de conteo o recogida-registro y almacenamiento

Se han definido las métricas a implementar dentro del proceso de elicitación, y ahora se definirán los procedimientos para recolectarlas, indicando para cada una de ellas: la fuente de los datos, responsabilidades, frecuencia o periodicidad de recolección y herramienta requerida, para así hacer que estos procedimientos sean claros y repetibles.

Como ya se ha mencionado, el proceso de elicitación involucra 3 actividades: **identificación de *stakeholders*, obtención de la lista de deseos y necesidades de cada *stakeholder* y, por último la integración, refinamiento y organización de la información obtenida.** Para cada una de estas actividades se han identificado otras sub-actividades, dentro de las cuales se hace la incorporación de las plantillas de registro de medidas. Se ha agregado a demás una etapa de verificación y análisis (*postmortem*) para la realimentación.

La Tabla 39 describe el proceso completo de elicitación, la Tabla 40, 41 y 42 describen las 3 actividades de la elicitación, y por último la Tabla 43 describe el *postmortem* de la elicitación.

Tabla 39. Desarrollo del proceso de elicitación.

Propósito.	Guiar en el desarrollo de las actividades de identificación de los <i>stakeholders</i> y en la integración y refinamiento de la lista de deseos y necesidades obtenidas de ellos.
Criterios de entrada.	<ul style="list-style-type: none"> ▪ Descripción del proyecto. ▪ Plantilla propuesta para la identificación de los <i>stakeholders</i> (se encuentra en el Anexo C). ▪ Plantilla propuesta para el control de las versiones de los requisitos (se encuentra en el Anexo C). ▪ Plantillas de Volere (Robertson & Robertson, 2006), la primera para el análisis de <i>stakeholders</i> y la segunda para la especificación atómica de requisitos (tarjeta o cubierta de los requisitos)²¹. ▪ Cronómetro (opcional, ya que se utilizará para medir el tiempo que dura una reunión con los <i>stakeholders</i>).
Actividades	
Actividad 1: Identificación de <i>stakeholders</i>.	<ul style="list-style-type: none"> ▪ Identificar y analizar a todos los <i>stakeholders</i> involucrados en el proyecto utilizando la plantilla de Volere (Robertson & Robertson, 2006). ▪ Analizar el tipo de conocimiento (de dominio de aplicación, contexto del problema y objetivos y metas del negocio) que poseen los <i>stakeholders</i> para determinar los que son necesarios y relevantes.
Actividad 2: Obtención de la lista de deseos de cada <i>stakeholder</i>. ²²	<ul style="list-style-type: none"> ▪ Elegir las técnicas de elicitación que más se adecuen al proyecto. ▪ Establecer las condiciones óptimas para aplicar cada una de las técnicas de elicitación y promover una buena comunicación y negociación entre los <i>stakeholders</i>. ▪ Aplicar cada una de las técnicas de elicitación. ▪ Almacenar los deseos y necesidades (requisitos) de cada <i>stakeholder</i>.
Actividad 3: Integración, refinamiento y organización de la información que se ha obtenido.	<ul style="list-style-type: none"> ▪ Integrar los deseos y necesidades (requisitos) de cada <i>stakeholder</i>. ▪ Resolver los conflictos, inconsistencias, ambigüedades y otros problemas que puedan presentarse en la información obtenida. ▪ Almacenar los requisitos en el documento de la lista de deseos y necesidades, llevando un adecuado registro de su origen y relación con otros requisitos.
Actividad 4: Postmortem.	<ul style="list-style-type: none"> ▪ Analizar la información obtenida de los <i>stakeholders</i>. ▪ Analizar los requisitos obtenidos (lista de deseos y necesidades). ▪ Completar la plantilla propuesta para el control de requisitos.
Criterios de salida.	<ul style="list-style-type: none"> ▪ Especificación de requisitos software.

²¹ La plantilla para el análisis de los *stakeholders* se encuentra disponible en <http://www.volere.co.uk/stakeholder.xls> y la de la tarjeta de requisitos se muestra en la página 4 de la plantilla de especificación de requisitos (el documento está disponible en http://www.volere.co.uk/pdf%20files/template_es.pdf) Estas dos plantillas se encuentran en el Anexo C traducidas al español.

²² Para esta actividad existen métricas que se han propuesto por parte de los autores de los estudios que dieron respuesta a la PI4, sin embargo, el objetivo de desarrollarla es la recolección de datos necesarios en las dos plantillas propuestas.

	<ul style="list-style-type: none"> ▪ Lista de <i>stakeholders</i> que participaron en la elicitación. ▪ Plantilla Volere (Robertson & Robertson, 2006) completada. ▪ Plantillas de identificación de <i>stakeholders</i> completada. ▪ Plantilla de control de las versiones de los requisitos terminada.
--	---

Tabla 40. Identificación de *stakeholders*.

Propósito	Guiar en el proceso de identificación de <i>stakeholders</i> .
Criterios de entrada	<ul style="list-style-type: none"> ▪ Descripción del proyecto. ▪ Plantilla Volere (Robertson & Robertson, 2006) para el análisis de <i>stakeholders</i>. ▪ Plantilla de identificación de <i>stakeholders</i>.
1. Analizar los <i>stakeholders</i> involucrados en el proyecto.	<ul style="list-style-type: none"> ▪ Utilizar la plantilla de Volere (Robertson & Robertson, 2006) para identificar a todos los <i>stakeholders</i> involucrados en el proyecto.
2. Analizar el conocimiento de los <i>stakeholders</i>.	<ul style="list-style-type: none"> ▪ Analizar el conocimiento de los <i>stakeholders</i> identificados con ayuda de la plantilla de Volere (Robertson & Robertson, 2006) y rellenar la plantilla para la identificación de <i>stakeholders</i> con los datos de los cada uno de ellos y el tipo de conocimiento que poseen. ▪ Determinar qué <i>stakeholders</i> participarán en las sesiones de elicitación.
Criterios de salida	<ul style="list-style-type: none"> ▪ Lista de <i>stakeholders</i> que participan en la elicitación. ▪ Plantilla de Volere (Robertson & Robertson, 2006) completada. ▪ Plantilla de identificación de <i>stakeholders</i> completada en los campos de análisis de los <i>stakeholders</i>.

Tabla 41. Obtención de la lista de deseos de cada *stakeholder*.

Propósito	Debido a que esta actividad ya está cubierta por métricas que han sido propuestas en otros estudios, el desarrollo de la actividad es únicamente para guiar en la recolección de los datos de la plantilla de identificación de <i>stakeholders</i> y la plantilla de control de requisitos dentro del proceso de obtención de la lista de deseos de cada <i>stakeholder</i> .
Criterios de entrada	<ul style="list-style-type: none"> ▪ Lista de <i>stakeholders</i> que participan en la elicitación. ▪ Plantilla de identificación de <i>stakeholders</i>. ▪ Plantilla Volere (Robertson & Robertson, 2006) para la especificación atómica de requisitos. ▪ Plantilla de control de requisitos
1. Elección de técnicas de elicitación.	<ul style="list-style-type: none"> ▪ Analizar las técnicas de elicitación existentes y elegir las que más se adecuen al proyecto.
2. Obtención de requisitos.	<ul style="list-style-type: none"> ▪ Establecer las condiciones óptimas para la aplicación de cada una de las técnicas de elicitación. ▪ Aplicar las técnicas de elicitación y obtener la lista de deseos y necesidades de los <i>stakeholders</i>.
3. Almacenar los requisitos.	<ul style="list-style-type: none"> ▪ Almacenar o documentar la lista de deseos y necesidades teniendo un adecuado registro de su origen y su relación con otros requisitos utilizando la tarjeta para la especificación atómica de requisitos.

Criterios de salida	<ul style="list-style-type: none"> ▪ Lista de deseos y necesidades de cada <i>stakeholders</i> utilizando las tarjetas para la especificación atómica de los requisitos. ▪ Plantilla de control de requisitos con la información de los requisitos obtenidos, su origen, y número elicitado por cada <i>stakeholder</i>.
----------------------------	--

Tabla 42. Integración, refinamiento y organización de la información que se ha obtenido.

Propósito	Guiar en el proceso de integración, refinamiento de la información que se ha obtenido.
Criterios de entrada	<ul style="list-style-type: none"> ▪ Lista de los <i>stakeholders</i> que participaron en la elicitación. ▪ Lista de deseos y necesidades (requisitos). ▪ Plantilla de control de requisitos.
1. Integración de la información.	<ul style="list-style-type: none"> ▪ Integrar la lista de deseos y necesidades de todos los <i>stakeholders</i>. ▪ Registrar y resolver los conflictos generados en cada requisito.
2. Refinamiento de información.	<ul style="list-style-type: none"> ▪ Analizar la información recolectada en cada requisito para descartar aquella que no sea útil. ▪ Analizar los requisitos para determinar si ya están completos.
Criterios de salida	<ul style="list-style-type: none"> ▪ Lista de deseos y necesidades. ▪ Plantilla de control de requisitos, ya completada.

Tabla 43. *Postmortem*.

Propósito	Guiar en el <i>postmortem</i> del proceso de elicitación.
Criterios de entrada	<ul style="list-style-type: none"> ▪ Lista de los <i>stakeholders</i> que participaron en la elicitación. ▪ Lista de deseos y necesidades (requisitos). ▪ Plantilla de identificación de <i>stakeholders</i> y plantilla de control de requisitos terminadas.
1. Stakeholders	<ul style="list-style-type: none"> ▪ Completar la plantilla para la identificación de <i>stakeholders</i> con la información de los requisitos obtenidos.
2. Lista de deseos y necesidades.	<ul style="list-style-type: none"> ▪ Verificar que se haya recolectado toda la información necesaria para completar la lista de deseos y necesidades.
3. Requisitos elicitados.	<ul style="list-style-type: none"> ▪ Verificar que la información de cada uno de los requisitos este completa.
Criterios de salida	<ul style="list-style-type: none"> ▪ Lista de deseos y necesidades revisada. ▪ Plantilla de identificación de <i>stakeholders</i> completada.

3.3.2.1.3. Diseño de formularios para el registro de medidas

Los procedimientos definidos anteriormente hacen uso de plantillas para el registro de medidas, la primera es para almacenar los datos de los *stakeholders* y la segunda para los datos de los requisitos. Estas dos plantillas así como las instrucciones para su llenado se encuentran en el Anexo C.

Para la plantilla de identificación de los *stakeholders* es necesario almacenar información referente a dos entidades: los *stakeholders* y los requisitos. Al iniciar el proceso de elicitación lo primero es analizar los *stakeholders*, y esto se hace por medio del análisis del conocimiento que

poseen, en la plantilla se indicará el tipo de conocimiento que tienen indicando además el nombre, la clase y el rol al que pertenecen. Ahora bien, cuando se estén elicitando los requisitos se rellenará la tabla de la sección referente al tiempo, en la que se indicará el tiempo que duró cada sesión de elicitación indicando también el número de veces que se ha efectuado y además se indicará la técnica de elicitación utilizada y el número de requisito obtenidos. Una vez que se hayan elicitado los requisitos se indicará por cada *stakeholder* el número de requisitos obtenidos por cada uno de ellos: tanto los propuestos como los definitivos, los que son necesarios para el proyecto y los que están acordes con los objetivos del proyecto. Posteriormente se contabilizarán los *stakeholders* que poseen conocimiento del dominio de aplicación, los objetivos del negocio y proyecto, experiencia en el problema a resolver; y se calculará el total de cada una de las columnas de requisitos, así como el tiempo que duró todo el proceso de elicitación y así conocer la frecuencia de elicitación que es el número de *stakeholders* elicitados por día.

En la plantilla del control de requisitos se almacenarán los cambios hechos a todas las versiones de la lista de deseos y necesidades. Para cada versión de la lista de deseos y necesidades se indicarán el número de requisitos que fueron agregados, modificados y eliminados, así como los requisitos que presenten algún conflicto. Para cada requisito se almacenará la fuente u origen (*stakeholder* que lo propuso) y el número de versiones existentes por cada lista de deseos y necesidades.

Los datos recolectados en las plantillas son medidas básicas o medidas primarias. En el caso de las medidas derivadas y medidas de ejecución del proyecto definidas en las tablas anteriores, es necesario procesar datos que se han recolectado en las dos plantillas anteriores. Estas operaciones se automatizaron en las plantillas de almacenamiento que fueron creadas en EXCEL (que se encuentra en el CD adjunto) y se explican en la siguiente sección

3.3.2.1.4. Definir mecanismos de almacenamiento

Para la definición de la base de datos, donde se puedan almacenar y procesar los datos de la medición y así obtener datos históricos, se utilizará una hoja de cálculo EXCEL (que se encuentra en el CD adjunto). Todos los datos recolectados en las plantillas definidas en la sección anterior se deben almacenar en dos hojas de cálculo (con el mismo formato que se usó para la recolección de la información contenida en cada una de ellas) pero con otros campos adicionales en los que se almacenarán los totales de las columnas necesarias.

En la platilla de análisis de *stakeholders* se almacenará el total de *stakeholders* que posean cierto tipo de conocimiento el número total de requisitos que fueron propuestos, definitivos, acordes con los objetivos y necesarios, el número total de sesiones por cada *stakeholders* y el tiempo total dedicado a ellos. Además se obtendrá la frecuencia de elicitación dividiendo el número de *stakeholders* que participaron en la elicitación entre la suma del tiempo que se dedico a cada uno de ellos.

Para la segunda plantilla, que es la de control de requisitos, la hoja de cálculo ayudará a generar el total de requisitos que se tengan por cada versión de lista de deseos y necesidades.

3.3.2.1.5. Definir métodos de análisis

Para cada uno de los objetivos se definirán las técnicas de análisis de datos y se mostrarán cómo es que las métricas pueden dar respuesta a estos objetivos. Primero se hace una breve descripción de los aspectos que son necesarios analizar para responder a las preguntas de medición, y después se identifican las relaciones existentes entre las medidas recolectadas, indicando además las señales de alerta que ayudan a identificar los problemas existentes para que se puedan tomar las

acciones correctivas correspondientes (i.e. si las medidas obtenidas son significativamente diferentes de las que se tienen planeadas o están fuera de los límites establecidos).

En el caso del primer objetivo de medición (**identificación de los *stakeholders***) se debe analizar qué *stakeholders* participarán en el proceso de elicitación, para esto es necesario definir el tipo de conocimiento (de dominio de aplicación, contexto del problema, y objetivos y metas del proyecto y del negocio) que debe poseer cada uno de ellos.

Una vez analizadas las fuentes de los requisitos se procederá a revisar si los requisitos obtenidos de cada uno de los *stakeholders* están acordes con los objetivos del negocio y si son necesarios para el proyecto (esto les permitirá pasar de propuestos a definitivos). El número de requisitos relevantes e importantes en relación al número total de requisitos que propuso cada *stakeholder* indicará si la elección de éstos fue adecuada respecto a los objetivos del proyecto.

Posteriormente es necesario verificar que la lista de deseos y necesidades esté completa, que se hayan elicitado todas las perspectivas de los *stakeholders* respecto al software a desarrollar, que los requisitos sean los correctos, etc.

Para hacer una visualización más clara del procedimiento de análisis del primer objetivo, se han identificado las siguientes consideraciones en las que se describen las relaciones entre las medidas recolectadas y las señales de alerta a tomar en cuenta.

- El número de categorías en las que son clasificados los *stakeholders* se debe comparar con el número de versiones que se tenga de cada uno de los requisitos (véase Figura 3.13). Cuando se hayan elicitado los requisitos de cada uno de los *stakeholders*, se tendrá una lista de deseos y necesidades por cada uno de ellos, sin embargo al integrar esta información para obtener una sola lista puede que se tengan diferentes versiones de un mismo requisito, este número es el que debe ser comparado con el número de categorías de los *stakeholders* participantes. Entre más grande sea la diferencia entre estos dos indicadores, más clara será señal de alerta, ya que o bien se han considerado *stakeholders* que no conocen la información descrita por el requisito o faltan *stakeholders* que no han participado en la elicitación.

Por ejemplo, si el número de versiones para el requisito con identificador llamado I1 (véase Figura 3.13) es 3 y las categorías de *stakeholders* que se identificaron son 9, se tienen 3 versiones de un requisito en el que hay como mínimo 9 puntos de vista diferentes y la diferencia entre ellos es de 6, el doble del número de las versiones del requisito. Sin embargo se debe considerar el número de requisitos proporcionado por cada *stakeholder*, ya que esto puede confirmar el hecho de que se eligieron a los *stakeholders* incorrectos (i.e. que no poseen el tipo de conocimiento necesario y que por lo tanto hace falta considerar a otros) o bien, proporcionar otro tipo de información como el que sí se hayan considerado a los *stakeholders* adecuados pero que éstos no hayan proporcionado toda la información necesaria (por lo que el requisito podría estar incompleto).

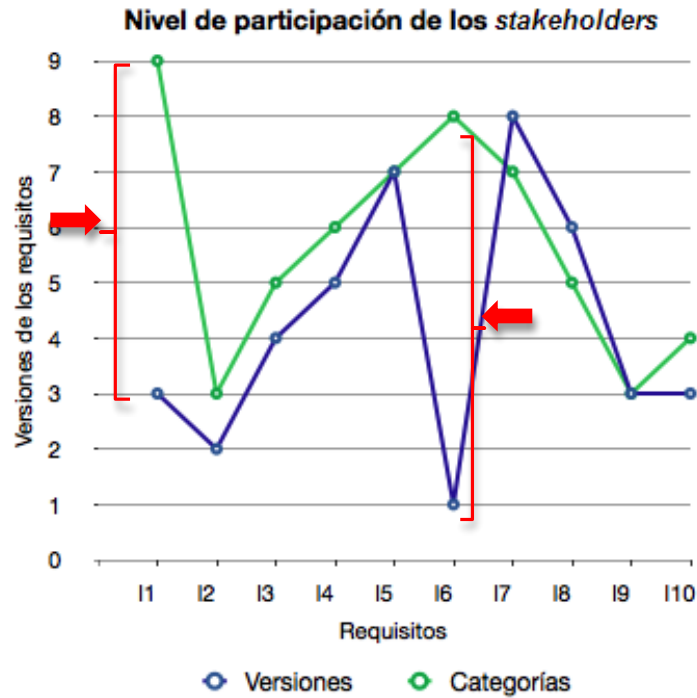


Figura 3.13. Gráfica que muestra el número de versiones por cada requisito y el número de categorías de stakeholders que poseen este tipo de conocimiento.

En la Figura 3.13 se indican con flechas las señales de alerta que hay que analizar para identificar si un requisito está incompleto, e.g., al no considerar todos los puntos de vista de los *stakeholders* o si éstos no proporcionaron la información necesaria para evitar futuros conflictos entre los requisitos.

- Para tener una mejor visualización de la relación que existen entre las medidas relacionadas con los requisitos y los *stakeholders*, se graficará el número de requisitos propuestos por cada uno de ellos (número de requisitos *versus* número de *stakeholders*) así como el número de los requisitos acordes con los objetivos del proyecto y negocio y los que al final fueron necesarios para el proyecto (requisitos definitivos). Por ejemplo, en la Figura 3.14 se puede apreciar el número de requisitos que se han propuesto y cuántos de ellos están acordes con los objetivos, de esta forma se puede verificar más fácilmente si el *stakeholder* que posee cierto tipo o tipos de conocimiento propuso un mayor o menor número de requisitos.

Esta gráfica se analiza contra la plantilla de Volere (Robertson & Robertson, 2006) para verificar el tipo de conocimiento que poseen los *stakeholders* y el número de requisitos propuestos por ellos y los que están acordes con los objetivos del proyecto y negocio. Además se analiza también el número de requisitos definitivos después de analizar su prioridad, importancia y relevancia.

- Otra pregunta que debe ser respondida es ¿qué métricas pueden indicar si los requisitos están completos? La respuesta no es del todo suficiente porque se deben considerar cuestiones subjetivas tales como el hecho de saber el nivel de información y detalle que debe contener cada requisito. Sin embargo, hay indicadores que sí pueden ser medidos objetivamente, e.g. que no existan requisitos pendientes de ser especificados (y en caso de existir, analizar la

densidad de faltas por cada requisito) o bien, que se haga referencia a objetos que no han sido descritos. Así mismo, debe analizarse también el porcentaje de los requisitos que están acordes con los objetivos y metas del proyecto y negocio.

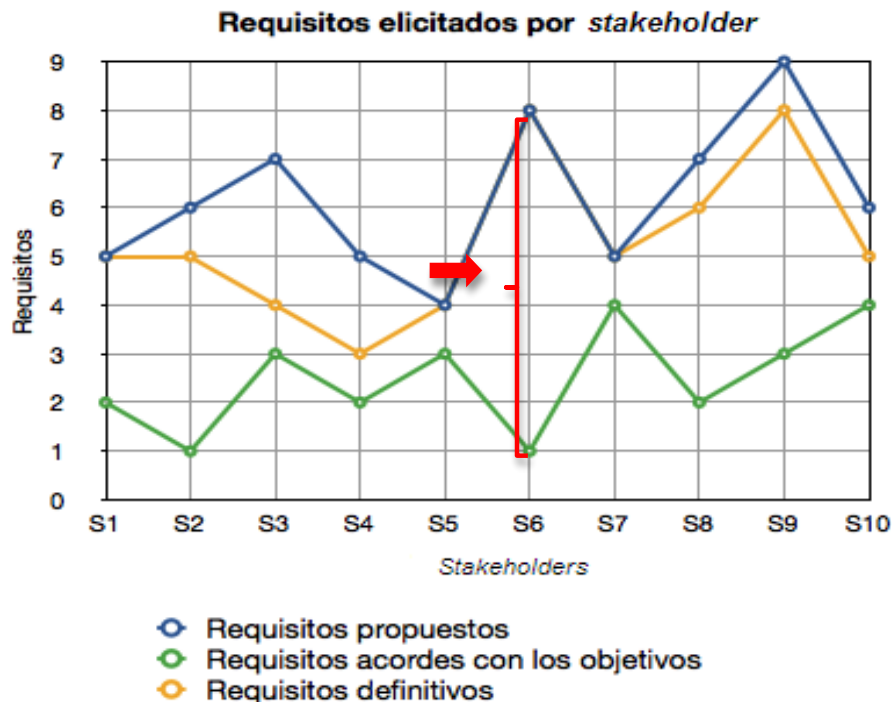


Figura 3.14. Gráfica que muestra el número de requisitos obtenido de cada *stakeholder*.

En la Figura 3.14 se puede ver un ejemplo donde hay una gran brecha entre el número de requisitos propuestos y los que finalmente estuvieron acorde con los objetivos del negocio, este aspecto puede indicar que se está considerando demasiada información irrelevante para el proyecto. En este sentido, la hoja de cálculo (que se encuentra en el CD adjunto) ayudará a obtener el porcentaje de estos requisitos con el objetivo de que no solo se tenga la información visualmente si no conozca la cantidad exacta.

- Por último se debe conocer si los requisitos son los adecuados para el software a desarrollar, es decir, si son los correctos. Para esto, se debe obtener el porcentaje de los requisitos que se han elicitado más de una vez (obtenido de la hoja de cálculo propuesta), que técnicas se aplicaron para elicitarlos, qué *stakeholders* participaron en la elicitación y cuántos requisitos se han elicitado de cada uno de ellos.

Además, el análisis de los requisitos obtenidos (número de requisitos definitivos *versus* requisitos propuestos, véase Figura 3.15) permitirá confirmar la relevancia e importancia que tiene los *stakeholders* participantes.

La relación que se da entre la frecuencia de utilización de las técnicas y la eficacia de las mismas, puede indicar qué tan efectivo fue el proceso de elicitación y si éste puede ser la causa de la obtención de requisitos de pobre calidad (i.e. si en la elicitación participaron muchos *stakeholders* pero se aplicaron técnicas con las que se obtuvieron pocos requisitos no se puede garantizar que éstas hayan sido las correctas).

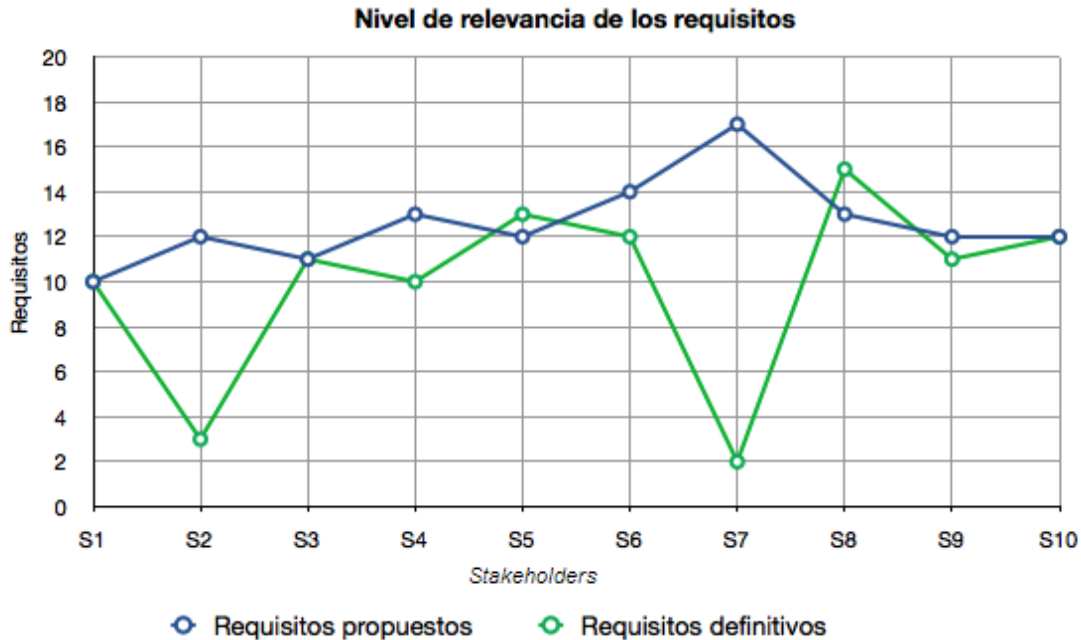


Figura 3.15. Gráfica que muestra el número de requisitos que fueron propuestos y aceptados por cada *stakeholder*.

Las señales de alerta a considerar para el cumplimiento del primer objetivo son:

- Aplicación de pocas técnicas de elicitación.
- Requisitos incompletos.
- Bajo nivel de acuerdo entre los *stakeholders*, lo que quiere decir que existen conflictos en los requisitos.
- Número menor al rango permitido de *stakeholders* que poseen cierto tipo de conocimiento: de contexto, del proyecto o del dominio del problema con respecto al total.
- Número pequeño de *stakeholders* que hayan propuesto requisitos necesarios o relacionados con los objetivos.

Para el segundo objetivo: integración y refinamiento de la lista de deseos y necesidades, es necesario analizar si al generar los requisitos definitivos se consideraron todas las perspectivas, necesidades e intereses de los *stakeholders*. Con respecto a la integración, se deben considerar los diferentes puntos de vista que necesitan ser integrados en relación al número de requisitos que presenten algún conflicto. Para el refinamiento de éstos se evalúa el número de versiones de la lista de deseos y necesidades tomando en cuenta la medida de completitud de los requisitos para así asegurar que no se tiene información extra ni alguna que no se ha especificado. Los puntos a considerar son los siguientes:

- Antes de integrar la lista de deseos y necesidades se debe contabilizar el número de versiones que se tengan de cada uno de los requisitos (véase Figura 3.16), con el objetivo de conocer cuánta información debió ser completada para poder lograr un consenso entre los *stakeholders* sobre ese requisito definitivo. También se debe analizar el número de requisitos que presentaron algún conflicto y que por supuesto debería ser menor al número de requisitos integrados (véase Figura 3.17).

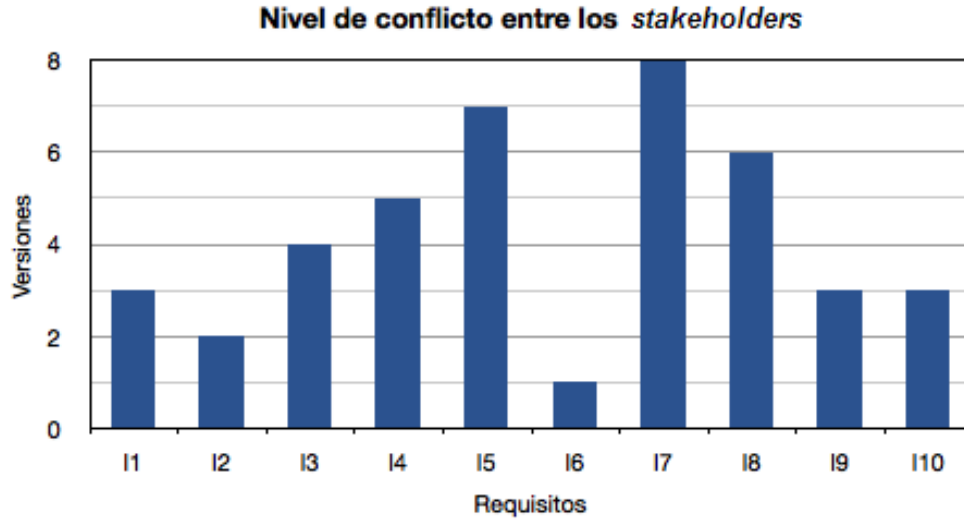


Figura 3.16. Gráfica que muestra el número versiones por requisito.

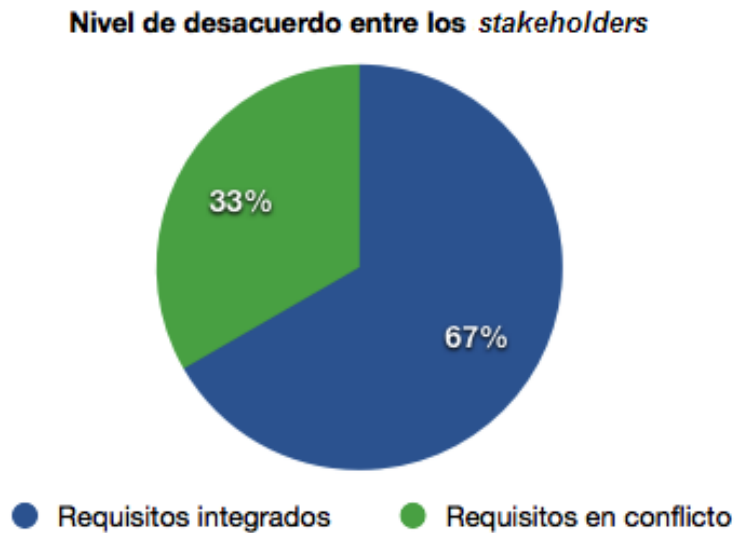


Figura 3.17. Gráfica que muestra el porcentaje de requisitos en conflicto.

- Una vez que se ha integrado toda la información, estos requisitos deben ser refinados para asegurar que no se esté considerando información de más o de menos. Para lograr esto es necesario analizar el número de versiones que se tiene de la lista de deseos y necesidades así como la volatilidad de los requisitos (agregaciones, modificaciones, eliminaciones) por documento y en total (véase Figura 3.18). Por lo tanto si en alguna versión de la lista de deseos y necesidades se tienen demasiados cambios es necesario ver la causa de estos.

Las señales de alerta que se deben tomar en cuenta y que repercuten en el logro del segundo objetivo de medición son:

- Que exista un número pequeño de versiones de la lista de deseos y necesidades, lo que puede significar que la información obtenida no ha sido refinada.
- El número de conflictos en los requisitos sea mayor al permitido y exista un número alto de versiones por cada requisito. Esto quiere decir que hace falta negociación y/o conciliación de intereses entre los *stakeholders*.
- Una alta densidad de omisiones por cada requisitos y en general de toda la lista de deseos y necesidades puede indicar que los requisitos están incompletos.

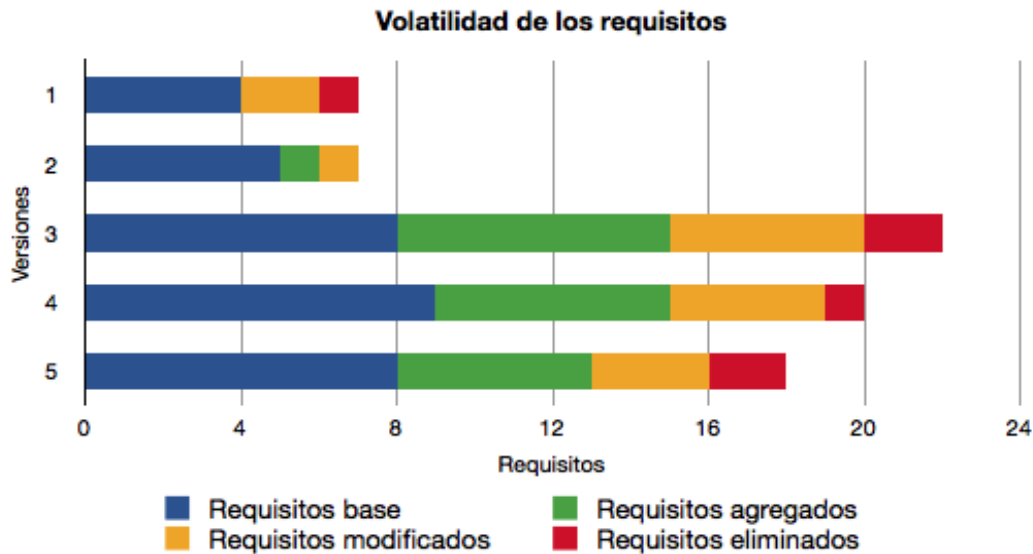


Figura 3.18. Gráfica que muestra las modificaciones hechas por cada versión de lista de deseos y necesidades.

3.3.2.1.6. Definir mecanismos de divulgación y realimentación

Una vez que el proceso de medición ha terminado se procede a dar a conocer los resultados. En esta actividad se debe decidir cómo se van a reportar las métricas, i.e. definir el formato del reporte de resultados, indicar la periodicidad y los mecanismos de reporte, así como la distribución y disponibilidad de los mismos.

Para esto, en primer lugar se debe elegir el formato de reporte de los datos, éste debe indicar la forma de visualizar las métricas (periodicidad, qué métricas están relacionadas entre sí y qué tipo de gráficas muestran mejor los resultados para así comparar los valores de cada una de ellas), en qué parte del documento se debe hacer el análisis y si los objetivos u otros valores deben estar incluidos en el reporte.

La periodicidad de los datos extraídos define cada cuándo se debe generar el reporte así como su distribución, por ejemplo si se debe esperar a que una fase del proceso software termine para así poder extraer los datos de alguna métrica y en consecuencia reportarla, o si deben ser reportados diariamente. Ahora bien, los mecanismos de reporte indican el formato en que la información referente a la métrica es entregada, por ejemplo, si se entrega por escrito o por correo electrónico, etc.

La distribución de los resultados servirá para determinar quiénes deben recibir los reportes de resultados y quiénes tendrán acceso a la información recolectada por cada métrica. En cuanto a la disponibilidad de las métricas, se deben definir algunas restricciones para tener acceso a ellas ya sea para leerlas, usarlas o modificarlas.

Para definir los mecanismos de divulgación se ha definido una plantilla de reporte de resultados que se encuentra en el Anexo C (en la que se dan a conocer las medidas que deben presentarse y su análisis).

Como se puede ver, en este Capítulo 3 se desarrolló la SLR, mostrando los estudios que dieron respuesta a las cuatro PI y los hallazgos que la evidencia científica proporcionó sobre las métricas utilizadas en la etapa de elicitación de los requisitos software. Ahora bien, una vez que se analizaron las respuestas a estas PI, específicamente a la cuarta, en la que se describen las métricas propuestas para el proceso de elicitación, se seleccionó el paradigma GQM para hacer una propuesta de las métricas faltantes que deben cubrir los objetivos del proceso de obtención de requisitos. Bajo este paradigma se identificaron los objetivos de medición, las preguntas generadas a partir de éstos, y las métricas que dan respuesta a éstas. Una vez identificadas las medidas, se definieron los procedimientos necesarios para el desarrollo de un programa de medición, cuya validación será a través de un caso de estudio descrito a continuación.

4. Validación y resultados

Para la validación de las métricas propuestas en el Capítulo 3 se llevó a cabo un caso de estudio en el entorno industrial. Pfleeger (2005) recomienda el uso de la experimentación rigurosa para la evaluación de tecnologías, métodos o herramientas y sus efectos en las organizaciones, procesos y productos. En la IS el uso de estudios empíricos contribuye a incrementar el conocimiento (Runeson & Höst, 2009), ayuda a los ingenieros de software en la toma de decisiones, y mejora la calidad del proceso-producto del software (Pfleeger, 2005). También, la experimentación transforma el desarrollo de software en una actividad científicamente predecible, gracias al conocimiento de la relación entre los procesos de producción y los productos obtenidos (Hannay, Sjoberg & Dyba, 2007) y por lo tanto, se pueden hacer evaluaciones de manera objetiva y científica.

De acuerdo con Wohlin, Höst y Henningston (2003) dentro de la IS existen principalmente tres tipos de métodos de investigación empírica: estudios (*surveys*), casos de estudio y experimentos formales. Los casos de estudio se usan para investigar un solo fenómeno o entidad dentro de un tiempo determinado, i.e. los investigadores recolectan información detallada durante, por ejemplo, un sólo proyecto por un período de tiempo determinado (Creswell, 2008; Runeson & Höst, 2009). Los casos de estudio ayudan a la industria a evaluar los beneficios de métodos y herramientas, y proveen una manera efectiva de asegurar que los cambios en los procesos generan los resultados deseados, además de que ayudan a evitar problemas a gran escala (Wholin, Höst & Henningston, 2003).

Por definición, los casos de estudio se llevan a cabo en entornos del mundo real, por lo que independientemente del nivel de control aplicado tienen un alto grado de realismo. Los datos recolectados en un caso de estudio pueden ser cuantitativos o cualitativos y el objetivo del estudio puede ser de naturaleza exploratoria, descriptiva explicativa o de mejora.

En este caso, se eligió la aplicación de un caso de estudio como la estrategia empírica experimental para evaluar los efectos del uso de métricas en la etapa de elicitación de requisitos software. De acuerdo con Wohlin, Höst y Henningston (2003) existen tres diferentes estrategias para desarrollar un caso de estudio:

- Comparar los resultados obtenidos de la propuesta aplicada a un proyecto y una línea base.
- Desarrollar dos proyectos en paralelo (proyectos gemelos) escogiendo uno de ellos como base.
- Aplicar la nueva propuesta en algunos componentes seleccionados y comparar los resultados con los componentes no seleccionados.

En este sentido, se utilizó la primer estrategia definida por Wohlin Höst y Henningston (2003), ya que de un proyecto previo se tenía la SRS que se utilizará como línea base. Por lo tanto, partiendo de la descripción del sistema, se procedió a aplicar las métricas que se han definido en el Capítulo 3 al mismo proyecto, que a partir de ahora se denominará A, para generar una lista de deseos y necesidades que permitieran comparar ambos productos (tanto la SRS del proyecto previo, como la lista de deseos y necesidades del proyecto nuevo).

El caso de estudio se enfocó en la aplicación de las métricas en el proceso de elicitación en una organización clasificada como MiPyME, es decir con menos de siete empleados. Los datos se recolectaron de dos grupos de trabajo que realizaron la misma tarea (elicitar requisitos para el proyecto A): un equipo de trabajo que no utilizó métricas en el proceso de elicitación y el otro equipo de trabajo que aplicó el programa de medición. La SRS base fue producida por un equipo de trabajo integrado por cuatro personas (que se denominará como grupo de control): un líder de proyecto, dos ingenieros de requisitos, y un gestor; para este grupo solo fue identificado un *stakeholder* más (cliente con el rol de jefe ejecutivo). Siguiendo los objetivos establecidos en este trabajo de tesis, se llevó a cabo el proceso de elicitación para producir la lista de deseos y necesidades por parte de un líder de proyecto, dos ingenieros de requisitos y un gestor (equipo denominado grupo experimental); este grupo aplicó las métricas propuestas y de acuerdo con ellas se utilizó la plantilla de Volere (propuesta por Robertson y Robertson (2006)) para la identificación de los *stakeholders*, estos últimos se describen más adelante. En los dos grupos de trabajo participaron el mismo número de personas en los equipos de trabajo, sin embargo, la diferencia radica en el número de *stakeholders* identificados por los ingenieros de requisitos para el proyecto. La selección de los miembros de ambos grupos fue hecha basándose en su experiencia y conocimiento en el proceso de la elicitación de requisitos, y en el conocimiento del dominio del proyecto. Es importante mencionar que todos los participantes tenían los mismos años de experiencia en la empresa (3 años).

Ahora bien, partiendo de la hipótesis establecida en el Capítulo 1 de esta tesis:

H₁: *Es posible definir métricas que contribuyan a que los ingenieros de software entiendan, controlen y mejoren el proceso de elicitación de requisitos.*

Se definió una hipótesis nula:

H₁₀: *No es posible definir métricas que contribuyan a que los ingenieros de software entiendan, controlen y mejoren el proceso de elicitación de requisitos.*

Para poder capturar diversos aspectos que conllevan la aplicación de las métricas, fue necesario definir un número de variables respuesta. Estas variables respuesta fueron derivadas del objetivo general descrito en el Capítulo 1 (los atributos de calidad de los requisitos software) y de las medidas propuestas en el Capítulo 3 (medidas para llevar a cabo la identificación de *stakeholders* y la integración y refinamiento de los requisitos obtenidos).

Las variables de respuesta resultantes fueron:

- Diversidad de los *stakeholders* participantes. ¿Cuántos *stakeholders* por categoría participan?
- Nivel de importancia de los requisitos. ¿Cuántos requisitos están acordes con los objetivos?
- Nivel de demanda de los requisitos propuestos por cada *stakeholders*. ¿Cuál es el nivel de urgencia de los requisitos?
- Nivel de relevancia de los requisitos. ¿Cuál es el número de requisitos necesarios para el proyecto?

- Nivel de acuerdo entre los *stakeholders*. ¿Cuál es el nivel de acuerdo entre los requisitos?
- Conocimiento del dominio de aplicación por parte de los *stakeholders*. ¿Cuántos *stakeholders* poseen conocimiento del dominio de aplicación?
- Conocimiento del problema por parte de los *stakeholders*. ¿Cuántos *stakeholders* poseen conocimiento del problema a resolver?
- Conocimiento de los objetivos del negocio por parte de los *stakeholders*. ¿Cuántos *stakeholders* poseen conocimiento de los objetivos del negocio?
- Categorías de los *stakeholders*. ¿Cuántas categorías de los *stakeholders* participaron?
- Nivel de conflicto entre los *stakeholders*. ¿Cuántas versiones diferentes existen de los requisitos?
- Productividad de cada *stakeholder*. ¿Cuántos requisitos han sido propuestos por cada *stakeholder*?
- Completitud. ¿Qué porcentaje de los requisitos están completos?
- Corrección. ¿Qué porcentaje de los requisitos están correctos?
- Densidad de omisiones en la lista de deseos y necesidades. ¿Cuántas faltas existen en la lista de deseos y necesidades?
- Nivel de aprobación por parte de los *stakeholders* de la lista de deseos y necesidades. ¿Cuántos *stakeholders* han aprobado la lista de deseos y necesidades?
- Eficacia de la aplicación de técnicas de elicitación. ¿Cuántos requisitos fueron obtenidos por cada técnica aplicada y por cada *stakeholder*?
- Nivel de desacuerdo entre los *stakeholders* (Consistencia). ¿Cuántos requisitos están en conflicto o son consistentes?
- Nivel de refinamiento de la lista de deseos y necesidades. ¿Cuál es el nivel de refinamiento de la lista de deseos y necesidades?
- Volatilidad de los requisitos. ¿Cuántos requisitos han sido agregados, modificados y eliminados?
- Participación de los *stakeholders*. ¿Cuántos *stakeholders* participaron activamente?
- Frecuencia de elicitación de requisitos. ¿Con qué frecuencia se han llevado a cabo las sesiones de elicitación?
- Tamaño de la lista de deseos y necesidades. ¿Cuántos requisitos contiene la lista de deseos y necesidades?
- No ambigüedad. ¿Qué porcentaje de los requisitos son no ambiguos?
- Trazabilidad. ¿Qué porcentaje de los requisitos son trazables?

Para responder estas preguntas se examinó la documentación del proyecto, se observó a los participantes durante su trabajo y se evaluaron los requisitos obtenidos en la última versión de la lista de deseos y necesidades de acuerdo a las métricas ya definidas para cada uno de los atributos.

4.1. Contexto del estudio

Como se mencionó anteriormente, el estudio fue realizado con la participación de RAGASOFT S. A. de C. V., una microempresa software. Cabe mencionar que RAGASOFT utiliza el estándar 830-1988 de la IEEE (IEEE, 1998) para la especificación de requisitos y el refinamiento de los casos de uso. Ahora bien, el proyecto desarrollado en este caso de estudio fue el denominado “Control de Información – Telecomunicaciones y Seguridad Electrónica” para la creación del producto llamado “TeleControl”. TeleControl es un sistema que sería utilizado como un portal de información para controlar la información interna de la empresa; mediante sus interfaces respectivas, y servicios relacionados al negocio del cliente. Esta versión de TeleControl incluye los siguientes módulos: módulo Web, gestión de usuarios, control de almacén, administración de usuarios, punto de venta, y control de equipo en servicio.

Para el desarrollo del caso de estudio se siguieron las cuatro fases del proceso GQM (planificación, definición, implementación e interpretación, descritas en la sección 3.3.1 del Capítulo 3).

Para el caso del grupo experimental se utilizaron las plantillas propuestas por Robertson y Robertson (Robertson & Robertson, 2006): tanto para la actividad de la identificación de los *stakeholders* como para el análisis de los *stakeholders* se utilizó la plantilla de Volere como tal, y para el almacenamiento de los requisitos se utilizó la tarjeta o cubierta para los requisitos también de Volere. Por otro lado, el grupo de control no realizó la actividad de identificación de *stakeholders* ya que dio por hecho que el cliente sería el único involucrado al cual se debía elicitar; para el caso del registro de los requisitos, se utilizó una minuta de reuniones en las que quedaron especificados los requisitos elicitados por cada sesión, esta minuta fue firmada por el cliente y líder de proyecto para indicar que estaban de acuerdo con lo establecido en los requisitos.

4.2. Método

Los datos recolectados se obtuvieron de tres formas: revisión de documentos históricos, observación de participantes y por medio de las plantillas para la recolección de métricas. Durante la revisión de documentos históricos, un investigador experto²³ revisó si la lista de deseos y necesidades estaba elaborada correctamente así como la calidad de los requisitos obtenidos. En cuanto a la observación de los participantes, fue necesario participar en tres sesiones de elicitación de requisitos (dos entrevistas y un grupo de enfoque) de requisitos para obtener de primera mano información referente a la extracción de requisitos. En la sección 3.3.2.1.2 del Capítulo 3 se definieron detalladamente los procedimientos para la recolección de los datos que dan soporte a las métricas mientras que en el Anexo C se encuentran las plantillas usadas para la obtención de medidas. La definición de las medidas y procedimientos incluyen:

- Definición de medidas.
- Todos los resultados posibles (rango de valores) de las medidas.
- La persona (rol) que recolecta cada medida.
- Periodo y momento en el que deben ser recolectadas las medidas.

²³ El investigador experto no fue parte del grupo de investigación y no estuvo consiente de que su trabajo formó parte de un experimento (este caso de estudio). El experto hizo una evaluación a ciegas sobre ambos conjuntos de requisitos sin saber como fueron obtenidos para evaluar su calidad (obtenidos del grupo experimental y de control).

- Los medios (en este caso plantillas) que deben ser usados para recolectar las medidas.
- Las herramientas automáticas que deben ser utilizadas.

El proceso de elicitación que aplica el programa de medición definido en el capítulo anterior, fue desarrollado de la siguiente forma:

- **Identificación de los *stakeholders*.** En esta actividad se identificaron a las posibles fuentes de requisitos (*stakeholders*) que pudieran proporcionar la información necesaria utilizando la plantilla de Volere (Robertson & Robertson, 2006) para el análisis de *stakeholders*. Se identificó a cuatro *stakeholders* (persona de mantenimiento software, usuario operativo técnico dentro de la clase de usuario, un modelador de datos del negocio como consultor interno y un especialista en seguridad como consultor externo; además de los miembros del equipo de desarrollo formado por un líder de proyecto, dos ingenieros de requisitos, y un gestor). Cada uno de ellos poseía conocimiento ya sea en el problema a resolver, en el dominio de aplicación, en los objetivos del negocio, así como sus propias necesidades y restricciones. Una vez identificados los *stakeholders* se rellenó la plantilla de “Identificación de *stakeholders*” en la sección de *stakeholders*, indicando el tipo de conocimiento que poseían.
- **Obtención de la lista de deseos y necesidades de cada *stakeholder*.** Durante esta actividad los ingenieros de requisitos decidieron aplicar dos técnicas de elicitación: entrevistas y grupos de enfoque, las entrevistas las aplicaron únicamente al cliente y al usuario, mientras que los grupos de enfoque se aplicaron (después de las entrevistas) a todos los *stakeholders* incluyendo a los miembros del equipo de desarrollo (con el fin de conciliar intereses y generar una lluvia de ideas). En la primera sesión de elicitación se extrajeron 48 requisitos funcionales (42 del cliente, y 6 del usuario), y en la segunda se elicitaron 5 requisitos funcionales, se modificaron 30 requisitos y se eliminaron 3. El tiempo que duró cada entrevista y la reunión del grupo de enfoque se almacenó en la plantilla de “Identificación de *stakeholders*”.
- **Integración, refinamiento y organización de la información obtenida.** Para esta actividad se integraron los requisitos elicitados de cada *stakeholder* y se generó una lista de requisitos (ambos documentos se muestran en la sección de Resultados de este capítulo), los cuáles fueron almacenados utilizando la plantilla de Volere (Robertson & Robertson, 2006) para la especificación atómica de requisitos, en ella se almacenó el origen de los requisitos y sus modificaciones (datos que también fueron almacenados en la plantilla de “control de requisitos”). Una vez que el proceso de elicitación terminó, se aplicaron las métricas definidas en la sección 3.3.2.1.1 del Capítulo 3.

Las medidas recolectadas a lo largo del proceso de elicitación se muestran en la Tabla 44. En la columna “valor” se indican los valores de cada una de las métricas y para una mejor interpretación se muestra su definición y el rango de valores permitidos para cada una.

Tabla 44. Resultados del caso de estudio.

Medida	Definición	Unidad de medida	Valor	Rango
Diversidad de los <i>stakeholders</i> participantes.	Es el número de <i>stakeholders</i> que se han identificado por cada categoría utilizando la plantilla de Volere.	Número de <i>stakeholders</i> participantes (por categoría).	1 mantenimiento 1 usuario 1 cliente 1 consultor interno	1-3

			1 consultor externo	
Nivel de importancia de los requisitos.	El número de requisitos acordes con el objetivo que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos acordes con el objetivo del proyecto, por cada <i>stakeholder</i> .	Mantenimiento- 1.9% Usuario – 11.7% Cliente – 80.35% Consultor interno – 0% Consultor externo – 3.9%	1-50% del total.
Nivel de demanda de los requisitos propuestos por cada <i>stakeholder</i> .	Es el número de <i>stakeholders</i> que se han identificado por cada categoría (utilizando la plantilla de Volere) y que han propuesto requisitos que deben ser atendidos inmediatamente (su demanda es importante y el tiempo de atención es crítico).	Número de <i>stakeholders</i> que propusieron requisitos urgentes o necesarios para el proyecto.	3 (cliente, usuario y consultor interno)	1-2 por categoría.
Nivel de relevancia de los requisitos.	El número de requisitos necesarios para el proyecto que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos necesarios propuestos por cada <i>stakeholder</i> .	84.31%	1-50% del total.
Nivel de acuerdo entre los <i>stakeholders</i> .	El número de requisitos que han sido aceptados por todos los participantes o <i>stakeholders</i> .	Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .	100%	80%-100% del total
Conocimiento del dominio de aplicación por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento referente al dominio de aplicación.	Número de <i>stakeholders</i> que conocen el dominio de aplicación.	3	1-Número total de <i>stakeholders</i> .
Conocimiento del problema por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento del problema a resolver.	Número de <i>stakeholders</i> que conocen el problema a resolver.	2	1-Número total de <i>stakeholders</i> .
Conocimiento de los objetivos del negocio por parte de los <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que poseen conocimiento de los objetivos del negocio.	Número de <i>stakeholders</i> que conocen los objetivos de negocio.	2	1-Número total de <i>stakeholders</i> .
Categorías de los <i>stakeholders</i> .	Es el número de categorías en la que los <i>stakeholders</i> participantes han sido caracterizados.	Número total de categorías de <i>stakeholders</i> .	5	1-14
Nivel de conflicto entre los <i>stakeholders</i> .	Es el número de versiones existentes para cada requisito por cada una de las versiones de las listas de deseos y necesidades.	Número de versiones diferentes de cada uno de los requisitos.	Un promedio de 2 versiones	1-Número total de <i>stakeholders</i> .
Productividad de cada	El número total de requisitos definitivos que se han elicitado	Número de requisitos elicitados	Mantenimiento- 1 Usuario – 6	1-100%

<i>stakeholder.</i>	por cada uno de los <i>stakeholders.</i>	por cada <i>stakeholder.</i>	Cliente – 42 Consultor interno - 0 Consultor externo - 2	
Compleitud de los requisitos.	El número de requisitos que están completos, es decir bien especificados (no tienen detalles que estén por ser determinados).	Número de requisitos que se especificaron completamente.	51	80%-100%
Requisitos correctos.	Es el número de requisitos que fueron revisados para que estuvieran acordes con los objetivos del proyecto y negocio.	Número de requisitos revisados para estar acordes con los objetivos (del proyecto y negocio).	51	80%-100%
Densidad de omisiones en la lista de deseos y necesidades.	Es el número de cuestiones omisas en el documento que contiene la lista de deseos y necesidades: <ul style="list-style-type: none"> ▪ Datos sin especificar. ▪ Información que este pendiente o por determinar. ▪ Entidades u objetos que son referenciados pero no son descritos. 	Número de faltas por requisito.	Dos faltas fueron identificadas en uno de los requisitos.	Ninguna.
Nivel de aprobación por parte de los <i>stakeholders</i> de la lista de deseos y necesidades.	Es el número de <i>stakeholders</i> que ha aprobado y firmado alguna versión de la lista de deseos y necesidades.	Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades.	5	Número total de <i>stakeholders.</i>
Tamaño de la lista de deseos y necesidades.	Se incluirá el número total de requisitos que fueron elicitados: <ul style="list-style-type: none"> ▪ Requisitos modificados. ▪ Requisitos agregados. ▪ Requisitos eliminados. 	Número total de requisitos elicitados.	51	Número total de <i>stakeholders</i> -infinito.
Eficacia de la aplicación de técnicas de elicitación.	Es el número de técnicas aplicadas a lo largo del proceso de elicitación.	Número de técnicas de elicitación utilizadas.	2	1-Total de técnicas existentes.
	Es la cantidad de <i>stakeholders</i> que participaron en cada técnica de elicitación.	Número de <i>stakeholders</i> que participaron en cada técnica de elicitación.	2 en entrevistas y los 5 en el grupo de enfoque	1-Número total de <i>stakeholders.</i>
Nivel de desacuerdo entre los <i>stakeholders.</i>	Es el número de los requisitos que presenten alguno de los siguientes problemas: <ul style="list-style-type: none"> ▪ Diferentes versiones por cada <i>stakeholder.</i> 	Número de requisitos en conflicto.	5	0-20% del total.

	<ul style="list-style-type: none"> El requisito no haya sido especificado completamente por algún problema entre los <i>stakeholders</i>. 			
Nivel de refinamiento de la lista de deseos y necesidades.	Número de las versiones existentes de la lista de deseos y necesidades.	Número de versiones de la lista de deseos y necesidades.	2	1-Número total de técnicas de elicitación.
Volatilidad de los requisitos.	Número de requisitos que fueron agregados a la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos agregados.	5	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.
	Número de requisitos que fueron modificados en la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos modificados.	30	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.
	Número de requisitos que fueron eliminados de la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos eliminados.	3	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.
Participación de los <i>stakeholders</i> .	Número de todos los <i>stakeholders</i> participantes durante todo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron activamente durante la elicitación	5	1-Total de <i>stakeholders</i> identificados al inicio.
Frecuencia de elicitación de requisitos.	Indica el tiempo en el que se ha llevado a cabo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron en la elicitación por día.	2 el primer día y los 5 en la segunda sesión	15%-25% del total de <i>stakeholders</i> identificados.
Tamaño de la lista de deseos y necesidades.	El número de requisitos que se han elicitado en la última versión de la lista de deseos y necesidades.	Número de requisitos elicitados.	51	Número total de <i>stakeholders</i> .

Una vez desarrollado el proceso de elicitación se obtuvo la lista de deseos y necesidades B (véase Anexo D). Como resultado de la aplicación de estas métricas se obtuvo que la diversidad de los *stakeholders* está dentro del rango permitido, más del 50% de los requisitos están acordes con el objetivo del proyecto con un 92.2 % de relevancia, un 85.7% de acuerdo y un nivel de conflicto de 11.4%. No hubo densidad de faltas identificadas en los requisitos en la última versión.

Es importante mencionar que los datos recolectados en este caso de estudio, son del tipo cuantitativo, y que después de su recolección, fueron validados por el investigador experto para

proceder a su análisis. La validación de los datos recolectados consistió en revisar su corrección, completitud y consistencia.

Una vez realizada la validación de los datos, todas las medidas y datos recolectados fueron almacenados en las bases de datos ya definidas, para posteriormente procesarlos y analizarlos, y así generar los reportes.

4.3. Resultados empíricos

Los resultados del caso de estudio, descrito anteriormente, se detallan en la Tabla 45. Por cada grupo de trabajo (experimental y de control) se muestra una columna de “Atributos de calidad” en la que se especifican la lista de los atributos (correctos, completos, consistentes, no ambiguos y trazables) que fueron evaluados tanto en la listas de deseos y necesidades obtenidas por el grupo experimental como en la SRS obtenida por el grupo de control.

Tabla 45. Atributos de calidad de los requisitos .

Grupo de control		Grupo experimental	
Atributos de calidad	Valor	Atributos de calidad	Valor
Correctos	67%	Correctos	98%
Completos	60%	Completos	98%
Consistentes	62%	Consistentes	94%
No ambiguos	58%	No ambiguos	96 %
Trazables	25%	Trazables	90%

Como se puede observar, la diferencia entre los requisitos obtenidos por el grupo de control y los obtenidos por el grupo experimental se ve reflejada en la calidad de los requisitos obtenidos. Sin embargo, es importante mencionar que cómo siempre sucede cuando se implanta un programa de medición, el proceso de elicitación de requisitos- empleando las métricas propuestas- requirió un poco más de tiempo para su aplicación (aproximadamente un 37%) que el que tardó el grupo de control (véase Tabla 45).

Ahora bien, la diferencia entre los requisitos obtenidos por el grupo de control y el grupo experimental se puede observar claramente, aunque hay que recalcar que el número de requisitos elicitados difiere solo por 6 requisitos funcionales más que fueron elicitados por el grupo experimental pero con una mejor calidad que el grupo de control. En este sentido, se puede afirmar que las métricas propuestas ayudaron a tener un mayor control en el proceso de elicitación y por ende, mejoraron la forma de llevarlo a cabo, lo que se ve reflejado en la calidad de los requisitos obtenidos. A pesar de que el hecho de implementar un programa de medición tiene sus desventajas (tiempo empleado en planear, desarrollar, procesar y analizar los datos, uso de medidas inapropiadas, interferencia en los procesos, riesgos de una mala interpretación), las medidas definidas no afectaron de forma negativa en el proceso ya que fueron fácilmente recolectadas. Los resultados obtenidos nos permiten aceptar la hipótesis (H_1 : *Es posible definir métricas que contribuyan a que los ingenieros de software entiendan, controlen y mejoren el proceso de elicitación de requisitos*).

Por último, los costos y beneficios en términos de tiempo (véase Tabla 46) de aplicar el programa de medición fueron documentados (el reporte esta disponible en el Anexo C) y almacenados en la base de datos de la empresa como datos históricos para una futura aplicación o mejora del programa de medición.

Tabla 46. Costos y beneficios de aplicar el programa de medición propuesto.

Costos por actividad	Beneficios
<ul style="list-style-type: none"> Identificación de <i>stakeholders</i>. <p>El tiempo empleado para desarrollar esta actividad fue, para el grupo de control de 2 días con 13 horas (aún cuando no se identificaron correctamente a todos los involucrados) mientras que el grupo experimental se tardó 1 día con 6 horas identificando a cuatro <i>stakeholders</i> además del equipo de desarrollo.</p>	<p>El tiempo invertido en la actividad de identificación de los <i>stakeholders</i> se vió reducido en un 51% con las métricas propuesta, es decir 1 día con 7 horas menos. Lo que resultó en un ahorro de \$381.94 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>
<ul style="list-style-type: none"> Obtención de la lista de deseos y necesidades de los <i>stakeholders</i>. <p>Los ingenieros de requisitos del grupo de control utilizaron 15 días con 10 horas para aplicar las sesiones de elicitación (no se cuenta con los datos de qué tipo de técnicas fueron aplicadas con los <i>stakeholders</i> debido a que se tuvieron problemas de comunicación y entendimiento), mientras que el grupo experimental lo hizo en 3 días con 8 horas con las dos técnicas seleccionadas de acuerdo con las características de los <i>stakeholders</i> identificados (entrevistas y grupos de enfoque).</p>	<p>El tiempo empleado por el grupo experimental para la obtención de la lista de deseos y necesidades a través de las sesiones de elicitación fue de un 78%, es decir 12 días con 2 horas menos. Por lo tanto se pudo ahorrar \$3573 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>
<ul style="list-style-type: none"> Integración, refinamiento e integración de la información. <p>La actividad de refinamiento e integración de información para el grupo experimental fue de 1 día, distribuyéndose de la siguiente forma: 23 horas para la modificación de algunos de los requisitos durante las mismas sesiones de elicitación (por las características propias de las técnicas, que permiten hacerlo) y las otras 5 para eliminar la ambigüedad e inconsistencia de los requisitos. Mientras que el grupo de control realizó esta actividad en varias sesiones, donde se emplearon 8 días.</p>	<p>En esta actividad, el tiempo empleado por el grupo de control fue reducido en un 88%, es decir 6 días con 20 horas menos para el grupo experimental, lo que conlleva una reducción de \$2020.50 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>

Como puede verse en los resultados mostrados, éstos permiten confirmar la hipótesis establecida de que: *Es posible definir métricas que contribuyan a que los ingenieros de software entiendan, controlen y mejoren el proceso de elicitación de requisitos*, ya que la propuesta central de esta tesis, permitió que los ingenieros de requisitos tuvieron un mayor entendimiento y control del proceso de elicitación, y en consecuencia se mejorará la ejecución del mismo, obteniendo así un mayor número de requisitos de calidad en un tiempo y a un costo menor.

5. Conclusiones

El éxito del desarrollo de un proyecto software es sinónimo de que éste sea entregado dentro del tiempo y presupuesto planeado, y lo más importante, que cumpla con las necesidades de los *stakeholders* (Brooks, 1987; Gibbs, 1994; Standish, 2010, 2011). De acuerdo a los estudios *CHAOS* realizados por el *Standish Group* (Standish, 2010, 2011) una de las principales causas de fracasos en los proyectos software pueden rastrearse a la etapa de requisitos.

Ahora bien, para que un software cumpla con los requisitos especificados, es indispensable dedicar los recursos necesarios al proceso de ingeniería de requisitos, específicamente en la primera etapa de este proceso: la elicitación de requisitos software. La elicitación de requisitos es una etapa crucial del proceso software en la que se decide qué es lo que se va a desarrollar, por lo tanto es importante saber cuáles son las necesidades de los *stakeholders* a satisfacer y no sólo considerar lo que éstos quieren. Es también en esta etapa donde se identifican tanto el contexto y dominio de la aplicación, como los objetivos del proyecto y negocio, los *stakeholders* que van a participar, y los deseos y necesidades de éstos. Para la identificación de los requisitos se deben aplicar técnicas de elicitación que ayuden al ingeniero de requisitos a conciliar y negociar intereses entre los participantes, además de tomar en cuenta que deben proveer un entorno que facilite la comunicación entre ellos. Una vez obtenidos los requisitos, éstos deben ser revisados para ser aceptados por todos los *stakeholders* (validación) y su calidad debe ser verificada para garantizar que sean completos, correctos, consistentes, no ambiguos, verificables, trazables y modificables (verificación). Entre más rápido se conozcan las características y restricciones que debe poseer el software, más pronto se evitarán problemas y errores a lo largo de cada una de las etapas del proceso software, ya que como consecuencia de que estos sean arrastrados a las siguientes etapas, los costos de arreglarlos se incrementan y en muchos de los casos el proyecto fracasa.

Ahora bien, para poder entender qué es lo que está sucediendo en el proceso de elicitación es necesario definir medidas que permitan interpretar, controlar y mejorar la elicitación de requisitos. Citando nuevamente la frase de DeMarco (2009), “*algo que no se mide, no puede ser controlado y por ende no se puede saber si en realidad se está mejorando*” es evidente que se hace necesario establecer indicadores que ayuden a los ingenieros de requisitos a controlar el proceso de elicitación para que, por ende, se mejore la calidad de los requisitos obtenidos.

De acuerdo con los objetivos planteados en el Capítulo 1 y con los resultados obtenidos de la aplicación de las métricas en el caso de estudio descrito en el Capítulo 4, se puede concluir lo siguiente:

- La utilización de las plantillas de Volere propuestas por Robertson y Robertson (2006) ayudaron a mejorar las actividades de identificación de los *stakeholders* y la captura de los requisitos, respectivamente.

- Las técnicas de elicitación utilizadas por el grupo experimental al ser elegidas de acuerdo a las características del proyecto y de los *stakeholders*, mostraron una mayor efectividad que la utilizada por el grupo de control al elicitar un mayor número de requisitos.
- Se desarrolló la actividad de identificación de *stakeholders* incorporando las métricas propuestas ya que no se encontraron estudios que propusieran métricas para ésta. En esta actividad se utilizó la plantilla de Volere para el análisis de los *stakeholders*, la cuál permitió identificar un mayor número de participantes así como el conocimiento que debían tener, a diferencia del proyecto considerado como línea base. Una vez que fueron identificados los *stakeholders* necesarios para el desarrollo del proyecto, se aplicaron las métricas correspondientes y se recogieron los resultados, los cuáles permitieron corroborar que se habían identificado correctamente tanto los *stakeholders* como el conocimiento necesario que éstos debían tener. La elección de las técnicas de elicitación aplicadas en este proyecto se hizo utilizando la guía sobre las técnicas de elicitación que se ubica en el Anexo B de esta tesis. De igual forma, para conocer la eficacia de las técnicas utilizadas se aplicaron las métricas y se recolectaron los indicadores respectivos, sin embargo como el proyecto desarrollado en este caso de estudio es pequeño no hubo mucha información que analizar ya que sólo fueron dos sesiones de elicitación en las cuáles se aplicaron sólo dos técnicas. Durante la aplicación de las métricas se pudo conocer el estado real del proceso de elicitación, lo que permitió que ante cualquier señal de alerta identificada se pudiese corregir rápidamente la actividad. Otra cuestión importante a hacer notar es que ciertas métricas necesitaron ayuda del conocimiento del contexto para ser interpretadas correctamente; e. g. ciertos *stakeholders* proporcionaron pocos requisitos, pero fue porque éstos solo colaboraban mínimamente en el proyecto ya que su conocimiento fue requerido para casos muy puntuales.
- Para la actividad de integración y refinamiento de información se propusieron y aplicaron métricas que garantizaran la generación de la última versión de la lista de deseos y necesidades. Las métricas aplicadas en esta etapa ayudaron a conciliar intereses entre los *stakeholders* y a negociar ciertos aspectos que originaban algún conflicto en los requisitos, sobre todo donde se involucró a más de un *stakeholder*. En cuanto al refinamiento de la información, ésta se analizó solo de manera cuantitativa, así mismo, se contabilizaron las versiones que se obtuvieron tanto de los requisitos como de la lista de deseos y necesidades, en este caso en particular debido al tamaño del proyecto se analizaron dos versiones de la lista de deseos y necesidades. Sin embargo con la información analizada fue suficiente para emitir un diagnóstico de los requisitos obtenidos hasta el momento.
- Las métricas propuestas ayudaron a contribuir en el desarrollo sistemático y efectivo del proceso de elicitación ya que se proporcionó un esquema adecuado para la identificación de los *stakeholders*, así como para la integración, refinamiento y almacenamiento de los requisitos, facilitando la gestión adecuada del proceso de elicitación.

Por todo lo anterior, se puede concluir que la hipótesis propuesta en el inicio de esta tesis es aceptada ya que con la evidencia recolectada en el caso de estudio se puede afirmar que es posible aplicar métricas que ayuden a los ingenieros de requisitos a entender, controlar y mejorar el proceso de elicitación con el objetivo de mejorar la calidad de los requisitos obtenidos.

Sin embargo, es necesario aplicar otros ciclos de medición para poder mejorar el programa propuesto a partir de los costos y beneficios analizados. También se debe reconocer que el caso de estudio fue realizado en una MiPyME y en un sólo proyecto software, por lo que no es posible

generalizar los resultados obtenidos ya que este programa de medición tendría que ser aplicado a otros proyectos para analizar y mejorar la interpretación de cada una de las métricas. Es posible suponer también que es posible formular métricas un poco más específicas, es decir que se adapten a cada una de las técnicas de elicitación utilizadas y a las tareas relacionadas con la actividad de obtención de la lista de deseos y necesidades de cada *stakeholder* (en esta actividad no se propusieron métricas debido a que otros autores han realizado algunas propuestas que cubren con el objetivo de medición identificado).

Por último, se sugiere trabajar en las siguientes líneas de investigación si se desea continuar abordando el tema trabajado en esta tesis:

- Es posible automatizar el proceso de incorporación de métricas, lo que incluye desarrollar una herramienta para la recolección de datos, procesamiento y almacenamiento de información, así como para el análisis y la interpretación de dichas métricas o indicadores; de esta forma se puede disminuir el tiempo de implantación del programa.
- Se pueden proponer métricas para analizar otras etapas del proceso de IR: análisis, especificación, validación-verificación, y gestión de requisitos software.
- Es necesario realizar otros casos de estudio en los que se apliquen las métricas propuestas para que con los resultados obtenidos se pueda ir configurando una base de datos históricos que permitan deducir resultados en proyectos similares.
- Evaluar la eficacia y escalabilidad de las métricas propuestas, en proyectos software más grandes que el del caso de estudio.
- Es posible proponer otras métricas que ayuden a complementar las métricas propuestas con el fin de proporcionar un análisis más completo.

6. Anexo A.- Acrónimos

ACM	Asociación para Sistemas de Computadora (<i>Association for Computer Machinery</i>).
ASD	Desarrollo Adaptable de Software(<i>Adaptive Software Development</i>)
CD	Disco Compacto.
CDRS	Calidad del Documento de Requisitos Software.
CMMI v1.1	Modelo de Madurez y de Capacidad Integrado (<i>Capability Maturity Model Integration v1.1</i>).
COTS	Componente tomado fuera del estante (<i>Commercial Off-The-Shelf</i>).
CORE	Expresión Controlada de Requisitos (<i>COntrolled Requirements Expression</i>).
CREWS	Ingeniería de requisitos cooperativa con escenarios (<i>Co-Operative Requirements Engineering with scenarios</i>).
CUTA	Análisis Colaborativo de Tareas de Usuario (<i>Collaborative User's Task Analysis</i>).
DGS	Desarrollo Global de Software .
EUA	Estados Unidos de América.
GQM	Paradigma Objetivo- Pregunta- Métrica (<i>Goal- Question- Metric</i>).
IEEE	Instituto de ingenieros eléctricos y electrónicos (<i>Institute of Electrical and Electronics Engineers</i>).
I*	Lenguaje de modelado de requisitos orientado a agentes.
IR	Ingeniería de Requisitos
ISO	Organización Internacional de Normalización (<i>International Organization for Standardization</i>).
IS	Ingeniería de Software
JAD	Diseño de Aplicaciones Conjuntas (<i>Joint Application Design</i>).
KAOS	Lenguaje de modelado de requisitos orientado a objetivos.

LOC	Líneas de Código (<i>Lines of code</i>)
MiPyME	Micro, Pequeña y Mediana Empresa.
NAD	Nivel de Adecuación del Documento.
NASA	<i>Nacional Aeronautics and Space Agency.</i>
NATO	Organización del Tratado del Atlántico Norte (<i>North Atlantic Treaty Organization</i>)
PDM	Metodología de Planeación y Diseño (<i>Planning and Design Methodology</i>).
PSM	<i>Practical Software Measurement.</i>
RAD	Desarrollo Rápido de Aplicaciones (<i>Rapid application development</i>).
RE	Requisitos Evolucionados.
RMI	Índice de Madurez de los Requisitos.
QFD	Desarrollo de Funciones de Calidad (<i>Quality Function Deployment</i>).
QR	Requisitos de Calidad (<i>Requirements Quality</i>).
RS	Requisitos Soportados.
RSA	Accesibilidad del Origen de los Requisitos.
RSD	Requisitos sin Defecto.
SE	Secretaría de Economía del Gobierno Mexicano.
SEI	Instituto de Ingeniería de Software (<i>Software Engineering Institute</i>).
SLR	Revisión Sistemática de Literatura (<i>Systematic Literature Review</i>).
SRS	Especificación de Requisitos Software (Software Requirements Specification)
SSE-CMM	Modelo de Madurez y Capacidad de Ingeniería de Seguridad y Sistemas (<i>Systems Security Engineering Capability Maturity Model</i>).
SSM	Metodología de Sistemas de Software (<i>Soft Systems Methodology</i>).
SWEBOK	Cuerpo de conocimiento de la Ingeniería del Software (<i>Software Engineering Body of Knowledge</i>)
UML	Lenguaje Unificado de Modelado (<i>Unified Modeling Language</i>)
VORD	Definición de Requisitos Orientada a puntos de Vista (<i>Viewpoint-Oriented Requirements Definition</i>)
XP	Programación Extrema (<i>Extreme Programming</i>).

7. Anexo B.- Guía de técnicas de elicitación de requisitos software

El proceso de elicitación de requisitos es una parte crucial dentro de la IR y dentro de éste es muy importante seleccionar la técnica adecuada de elicitación para extraer requisitos ya que ésta puede influir en la calidad de los requisitos obtenidos. Por todo esto, el objetivo de esta sección es proveer una guía que ayude al ingeniero de requisitos a seleccionar la técnica de elicitación más efectiva en cierto contexto.

El proceso de elicitación involucra diferentes *stakeholders*, y cada uno de ellos tiene diferentes características, objetivos individuales y/o organizacionales, así como diferentes formas de expresar y entender su conocimiento. Para poder elicitar este conocimiento existe una gran variedad de técnicas de elicitación, pero es complicado decidir cuál o cuáles se ajustan más a los *stakeholders* y características del proyecto a desarrollar. En el estudio realizado por Hickey & Davis (2003) los ingenieros de requisitos con poca experiencia escogen la única técnica que ellos conocen, o bien la técnica que aplicaron por última vez y esto es debido a la falta de conciencia de los métodos disponibles y al poco conocimiento y entendimiento que se tiene de las técnicas de elicitación. Sin embargo las consecuencias de una mala elección en la técnica o combinación de técnicas puede resultar en requisitos de baja calidad y en consecuencia, en el desarrollo del software inapropiado. Por estas razones es importante y útil conocer las ventajas y desventajas de cada una de ellas, ya que el problema no radica en la cantidad de métodos existentes sino en saber elegirlos. Además, hay que considerar que se necesita más de un método de elicitación para capturar efectivamente requisitos de calidad y que su elección depende de las características del proyecto, *stakeholders* y el producto software a desarrollar.

7.1. Estructura de la guía

En las siguientes subsecciones se explica cada una de las técnicas de elicitación, proporcionando un total de catorce técnicas. Para cada subsección se sigue la siguiente estructura:

1. Primero se describe brevemente técnicas de elicitación,
2. Después se describe de forma general las situaciones en las que es recomendable aplicar las técnicas de elicitación, lo que incluye: las condiciones del proyecto así como las características de los *stakeholders* en los que es recomendable usarlas.

7.1.1. Métodos Ágiles

Los métodos ágiles han sido desarrollados con el objetivo de permitir la entrega de software de manera más rápida y asegurar que el software satisface las necesidades de los *stakeholders* (Paetsch, Eberlein, & Maurer, 2003). Todos estos enfoques tratan de mejorar la satisfacción del

cliente, adaptándose a los cambios en los requisitos, colaborando cercanamente con los *stakeholders* y entregando frecuentemente un prototipo del software.

De manera general, los métodos ágiles se aplican cuando:

- El tamaño del proyecto es pequeño o mediano.
- El tamaño del equipo de desarrollo es pequeño.
- Se tiene una alta limitación en el tiempo de entrega y limitaciones en los costos de desarrollo.
- Se trate de el desarrollo de nuevas aplicaciones o en ciertos casos del mejoramiento de un software existente y su complejidad sea baja o media.
- La volatilidad de los requisitos sea alta.
- Los stakeholders involucrados no sean muchos pero posean conocimiento del dominio, problema y objetivos del negocio. Estos deben de tener la disponibilidad de colaborar con el equipo de desarrollo y poseer buenas habilidades de comunicación.
- No importa que se tenga una alta heterogeneidad de los *stakeholders* ni el grado de experiencia del ingeniero de requisitos.

En la Tabla 47 se muestran las características de los *stakeholders* que podrían participar en la elicitación de los requisitos.

Tabla 47. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	Operacional
Tipo	Primarios
Habilidades interpersonales	Comunicación y colaboración

7.1.2. Prototipos

El prototipo de un software es la versión inicial de un sistema y ésta versión está disponible en la etapa de elicitación para ayudar a extraer y validar requisitos. Se consideran dos clases de prototipos (Davis, 1992): los desechables y los evolutivos. Los prototipos evolutivos proveen de un software funcional a los *stakeholders* y llegan a formar parte del software final, en cambio los desechables solamente ayudan a entender requisitos.

La técnica de prototipos se aplica cuando:

- El tamaño del proyecto y del equipo de desarrollo es pequeño.
- Se tiene limitación en el tiempo de entrega y en los costos de desarrollo.
- La complejidad del proyecto no es muy alta pero si la volatilidad de los requisitos.
- El proyecto considera un bajo grado de seguridad.
- El tipo de producto a desarrollar es nuevo o para mejorar alguna funcionalidad.

- Los *stakeholders* deben tener conocimiento del problema y habilidades de comunicación y colaboración.

En la Tabla 48 las características que deben poseer los *stakeholders* para aplicarla.

Tabla 48. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	Operacional
Tipo	Primarios
Habilidades interpersonales	Comunicación y colaboración

7.1.3. Análisis de documentos

En esta técnica se hace el análisis de documentos tales como planes estratégicos, estudios de mercado, regulaciones, estándares, manuales, entre otros. Estos documentos contiene requisitos estáticos y los ingenieros de requisitos los analizan manualmente.

Es recomendable utilizar esta técnica cuando:

- El tamaño del proyecto y del equipo de desarrollo es pequeño o mediano.
- La complejidad del proyecto es alta y se tienen limitaciones de costo y tiempo.
- La volatilidad de los requisitos no es muy alta pero el grado de seguridad del proyecto es alta.
- El tipo de producto a desarrollar es un nuevo producto o aquél que este basado en COTS, y que también considere un alto grado de seguridad.
- Exista una gran variedad de documentación de procesos que se desean automatizar.
- Sea necesario analizar regulaciones, estándares o leyes que el software debe cumplir.
- Se pretendan reutilizar requisitos de sistemas ya existentes.

Hay que tomar en cuenta que algunas veces los documentos no están bien organizados o no contienen la descripción correcta de algún proceso. La Tabla 49 muestra las características de los *stakeholders*.

Tabla 49. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Documentos
Conocimiento	Dominio, problema, negocio o técnico
Experiencia	No aplica
Rol	No aplica
Tipo	No aplica
Habilidades interpersonales	No aplica

7.1.4. Entrevistas

Es la técnica de elicitación más utilizada, y básicamente consiste en que una persona llamada entrevistador realiza una serie de preguntas a otra que es el entrevistado (Diaper, 1989). Las entrevistas se pueden clasificar en entrevistas estructuradas, no estructuradas ó una combinación de estas conocida como semiestructuradas.

En las entrevistas estructuradas el ingeniero de requisitos pregunta a los *stakeholders* una lista de preguntas que han sido formuladas con anticipación. Para este tipo de entrevistas, la condición es que hay que identificar cuestiones y preguntas relevantes.

En las entrevistas no estructuradas, el ingeniero de requisitos pregunta cuestiones a los *stakeholders* sin una lista preparada de preguntas.

Se recomienda usar las entrevistas cuando:

- El tamaño del proyecto es grande o mediano y no se tienen muchas limitaciones en el tiempo y costo.
- El tipo de producto a desarrollar es nuevo o se haga una actualización del mismo.
- La volatilidad de los requisitos no es alta y en su mayoría son aplicadas cuando se desarrolla nuevo software.
- La complejidad del proyecto puede ser baja, mediana o alta.
- El grado de seguridad que se requiera en el proyecto sea mediano o alto.
- Se desea obtener un panorama inicial de nuevos proyectos software o cuando se trabaja en un nuevo dominio de aplicación.
- Se necesita conocer si las cuestiones políticas u organizacionales pueden influir al aplicar una técnica colaborativa y así reprimir a cierto tipo de stakeholders.
- Exista algún conflicto que sea difícil resolver entre los stakeholders, para poder aislarlo y tratarlo por separado.
- Existan expertos en el dominio y los stakeholders sean inaccesibles.
- Se desea identificar cuestiones que son importantes para los stakeholders.
- No se cuente con mucho tiempo para planearlas ya que son simples de realizar.

Las características de los *stakeholders* se describen en la Tabla 50.

Tabla 50. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	Operacional, Estratégico o Táctico
Tipo	Primarios
Habilidades interpersonales	Comunicación

Sin embargo hay que considerar que:

- Su efectividad depende del entrevistador, de la forma de expresión y disponibilidad de los *stakeholders*.
- Son técnicas estáticas, esto quiere decir que son menos eficientes al investigar todos los posibles escenarios y limitaciones del software.

7.1.5. Cuestionarios

Los cuestionarios son una alternativa a las entrevistas, éstos permiten elicitar requisitos sistemáticamente de diferentes *stakeholders*

Es recomendable utilizarlos cuando:

- Ya se identificaron ciertas cuestiones específicas y concretas a las que se desea abordar.
- Se necesite obtener información estadística.
- Se necesite conocer las necesidades de un grupo de *stakeholders* en particular.

Aunque hay que tomar en cuenta que:

- Se limita a cuestionar solamente las preguntas formuladas.
- No se pueden aclarar dudas que se tengan respecto a las respuestas.
- Puede haber malas interpretaciones de las preguntas de parte de los *stakeholders*.
- Su efectividad depende del formato y preparación de las preguntas.

7.1.6. Rejillas de repertorio

Para llevar a cabo esta técnica los *stakeholders* son consultados para conocer cuáles son los atributos aplicables a ciertas entidades del dominio. Se tiene que elaborar una matriz de atributos que se ajuste a las entidades identificadas y con una semántica acorde al dominio de aplicación.

Esta técnica es buena aplicarla cuando

- El conocimiento necesita ser representado en un formato estándar.
- Se requiere hacer un análisis estadístico.

Pero se debe considerar que al utilizarla:

- El conocimiento elicitado es limitado a identificar el objeto y sus atributos.
- No se elicitan procedimientos ni estructuras.
- Se pueden presentar problemas al tener que establecer un rango de valores para los atributos.

7.1.7. Observación

El ingeniero de requisitos observa las prácticas o las tareas de los *stakeholders* en su lugar de trabajo. Algunas veces el ingeniero de requisitos necesita entrenamiento para el uso de métodos etnográficos y también hay que considerar que se debe conseguir el acceso al lugar de trabajo de los *stakeholders*.

Se recomienda utilizarla en el caso de que:

- El tamaño del proyecto, su complejidad y el equipo de desarrollo pueden ser pequeño o mediano.
- Se tengan altas limitaciones de tiempo y costo.

- La volatilidad de los requisitos puede ser baja, mediana o grande.
- Ya exista un sistema y se cuente con la disponibilidad de los stakeholders para ser observados.
- La volatilidad de los requisitos y el grado de seguridad del proyecto no importa ya que esta puede ser baja, mediana o alta.
- Los *stakeholders* no cuenten con mucho tiempo disponible.
- Se quiera conocer cuestiones que los stakeholders dan por hecho que se conocen.
- A los *stakeholders* les sea difícil expresarse.
- Se necesite entender procesos complejos.
- Se desee conocer más a detalle el dominio de aplicación.

Las características de los *stakeholders* se muestran en la Tabla 51.

Tabla 51. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	Con sistemas similares o en el trabajo realizado.
Rol	Operacional, Estratégico o Táctico
Tipo	Primarios
Habilidades interpersonales	No importa

Sin embargo, hay que considerar que:

- Se puede capturar gran cantidad de datos irrelevantes.
- Requiere de mucho tiempo por parte del ingeniero de requisitos.
- Puede no detectar eventos que suceden con poca frecuencia.
- Pueden surgir problemas al reportar cierto tipo de información confidencial de la organización donde laboran los *stakeholders*.

7.1.8. Análisis de protocolo

En esta técnica los *stakeholders* hablan en voz alta mientras realizan sus tareas. Para poder llevarla a cabo se deben identificar con anticipación, cuáles son las tareas de los *stakeholders* a analizar.

Se recomienda utilizar cuando:

- Se necesite entender procesos complejos.
- Se desee identificar requisitos implícitos.
- Se quiera obtener información acerca de tareas específicas.

Pero es importante considerar que esta técnica consume mucho tiempo y que hay procesos o tareas que son complicadas de entender y analizar. Las características de los *stakeholders* se muestran en la Tabla 52.

Tabla 52. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	Con sistemas similares o en el trabajo realizado.
Rol	Operacional, Estratégico o Táctico
Tipo	Primarios
Habilidades interpersonales	No importa

7.1.9. Técnicas de ordenamiento

Dentro de estas técnicas se considera el *card sorting*, en la cuál los *stakeholders* ordenan en grupos un conjunto de tarjetas, en las cuáles describen entidades del dominio de aplicación y también se indica el criterio de ordenamiento. Se deben identificar las entidades del dominio de aplicación.

Se recomienda utilizar en el caso de que:

- Se desee elicitar conocimiento acerca del dominio de aplicación.
- Se pretenda utilizar un formato estándar para almacenar el conocimiento.
- Se pretenda priorizar ciertas funcionalidades o entidades del dominio.

Sin embargo esta técnica se limita a solo identificar el objeto y sus atributos y no se obtiene conocimiento de las estructuras o procedimientos.

7.1.10. Laddering

El ingeniero de requisitos al igual que en las entrevistas estructuradas utiliza un conjunto de preguntas para obtener conocimiento de los *stakeholders*. Trata de manera jerarquizada el conocimiento.

Se recomienda utilizarla en las mismas condiciones que una entrevista pero además se debe considerar que es útil aplicarla si se pretende jerarquizar el conocimiento obtenido.

En general se recomienda utilizarla cuando:

- El tamaño del proyecto y del equipo de desarrollo sea grande.
- Las limitaciones de costos y tiempo no sean muy altas.
- La volatilidad de los requisitos sea alta.
- La complejidad del proyecto sea mediana y el grado de seguridad a considerar sea mediano.
- El tipo de producto a desarrollar sea nuevo.

En la Tabla 53 se muestran las características de los *stakeholders*.

Tabla 53. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas

Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	Operacional, Estratégico o Táctico
Tipo	Primarios
Habilidades interpersonales	Comunicación

7.1.11. Colaborativas

La ventaja del uso de estas técnicas es que se puede tener un grupo de múltiples (grupos grandes y de diversos tipos) *stakeholders* para llevar a cabo una sesión colaborativa; incluso si ellos se encuentran en diferentes lugares, la reunión se puede realizar con el apoyo de tecnologías existentes. Cuando se necesita visualizar sistemas innovadores el uso de esta técnica ayuda a obtener más requisitos de este tipo en comparación con otras técnicas. En general, para los ingenieros de requisitos estas son consideradas un enfoque estándar o por defecto para la obtención de requisitos.

7.1.11.1. Workshops

Los *workshops* son un tipo de talleres o reuniones en los cuáles se trata de algún tema en particular. En la Tabla 54 se muestran las características de los *stakeholders*. De manera general es recomendable aplicar esta técnica cuando:

- El tamaño el proyecto no importa, este puede ser pequeño, mediano o grande, así como el tamaño del equipo.
- La volatilidad de los requisitos puede ser baja o mediana.
- El grado de seguridad requerido para el proyecto no importa.
- El tipo de producto software a desarrollar puede ser nuevo.
- Se tienen limitaciones en el tiempo de entrega y en los costos de desarrollo.
- La complejidad del proyecto no sea muy alta y el tipo de producto a desarrollar es nuevo.
- Los *stakeholders* involucrados sean de muchos tipos. Estos deben de tener la disponibilidad de colaborar con el equipo de desarrollo y poseer buenas habilidades de comunicación.

Tabla 54. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	No importa
Tipo	No importa
Habilidades interpersonales	Comunicación y colaboración

7.1.11.2. Grupos de enfoque

Es una técnica colaborativa en la cuál se discute un tema en particular, es considerada también como un tipo de entrevista grupal (Goguen & Linde, 1993). En la Tabla 55 se describen las características de los *stakeholders*.

Una vez analizados los casos particulares, se puede concluir que es recomendable utilizar la técnica de grupos de enfoque cuando:

- El tamaño del proyecto y del equipo de desarrollo es pequeño o mediano.
- Se tienen limitaciones en el tiempo de entrega y en los costos de desarrollo.
- La complejidad del proyecto no sea muy alta.
- El grado de seguridad requerido en el proyecto puede ser bajo, mediano o alto
- La volatilidad de los requisitos no importa ya que puede ser alta, mediana o baja.
- El tipo de producto a desarrollar es nuevo o en todo caso se puede ampliar cuando se este desarrollando un mejoramiento del producto.
- Los *stakeholders* involucrados sean de muchos tipos. Estos deben de tener la disponibilidad de colaborar con el equipo de desarrollo y poseer buenas habilidades de comunicación.

Tabla 55. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	No importa
Tipo	No importa
Habilidades interpersonales	Comunicación y colaboración

7.1.11.3. JAD

Es un tipo de *workshop* en el cuál 8-20 *stakeholders* toman decisiones a través del consenso, sin embargo el moderador o facilitador debe ser entrenado. Requiere que exista una planeación y también existen actividades que pueden surgir después de las reuniones. Se recomienda utilizar cuando:

- El tamaño el proyecto puede ser pequeño, mediano o grande; así como el equipo de desarrollo no importa el tamaño.
- Se pueden tener o no limitaciones en tiempo y costo.
- El grado de seguridad requerido para el proyecto es importante ya que puede ser alto o en todo caso mediano.
- La complejidad del proyecto es también puede ser alta, mediana o baja.
- El tipo de producto a desarrollar puede ser un nuevo software, un producto serial o na actualización del producto.
- Se pretenda mejorar la calidad y diseño del software.

- Se desea llevar a cabo una reunión mas sistematizada.

En la Tabla 56 describe las características de los *stakeholders*. Hay que tomar en cuenta que esta técnica requiere mucho tiempo y disponibilidad de los *stakeholders*.

Tabla 56. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	No importa
Tipo	No importa
Habilidades interpersonales	Comunicación y colaboración

7.1.12. Lluvia de ideas

La lluvia de ideas se lleva a cabo cuando un ingeniero de requisitos reúne a los *stakeholders* para que generen una gran cantidad de ideas, centrándose en la cantidad de ideas generadas y no en la evaluación de éstas.

Se recomienda utilizar esta técnica cuando:

- El tamaño y la complejidad del proyecto no son muy altas
- Se tienen algunas limitaciones en tiempo y costo.
- Se trate de un nuevo producto ó de una nueva versión.
- Existe una gran cantidad y variedad de *stakeholders*.
- La complejidad del proyecto puede ser baja o mediana.
- El grado de seguridad requerido es mediano.
- Se pretenden elicitar cuestiones novedosas o creativas para el software a desarrollar.

Pero hay que tomar en cuenta que:

- Es susceptible a centrarse en procesos de un grupo de *stakeholders*.
- En muchas veces se elicitán cuestiones irrelevantes.
- Requiere que la mayoría de los *stakeholders* estén disponibles.
- Es necesario que el facilitador pueda organizar y conducir la reunión.

En la Tabla 57 se muestran las características que deben poseer los *stakeholders*.

Tabla 57. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	Personas
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	No importa

Tipo	No importa
Habilidades interpersonales	Comunicación y creatividad

7.1.13. VORD

Es un método establecido por Kotonya y Sommerville (1996), el cuál esta orientado a los puntos de vista de los *stakeholders*. En la fase de elicitación se enfoca a identificar los punto de vista y de estructurarlos

Se recomienda utilizar esta técnica cuando:

- El tamaño del proyecto y del equipo de desarrollo no es muy grande.
- Se tienen algunas limitaciones en el costo y tiempo del proyecto.
- La volatilidad de los requisitos puede ser baja, mediana o alta.
- El grado de seguridad requerido incluye baja, mediana o alta.
- El tipo de producto a desarrollar es nuevo o esta basado en COTS.
- Existe una gran cantidad y variedad de stakeholders.
- Se pretenden utilizar el enfoque VORD para todo el proceso de IR.

En la Tabla 58 describe las características de los *stakeholders*.

Tabla 58. Características de las fuentes de requisitos.

Características de los <i>stakeholders</i>	
Fuente de requisitos	No importa
Conocimiento	Dominio, problema, negocio
Experiencia	No importa
Rol	No importa
Tipo	No importa
Habilidades interpersonales	Comunicación

8. Anexo C.- Plantillas

En este anexo están descritas las plantillas que se utilizan durante todo el proceso de medición:

- Para la identificación de *stakeholders*.
- Para el control de requisitos.
- Para el reporte de resultados.

Las dos primeras son para la recolección de datos, y la tercera plantilla se utilizará como una guía para hacer la presentación del análisis de datos y de los resultados obtenidos en el proceso de medición.

En el reporte de resultados se indica la periodicidad de los resultados, la cuál debe coincidir con el de obtención de medidas (al terminar la fase de elicitación cuándo se genera la última versión de la lista de deseos y necesidades). En cuanto a la forma de entregar el reporte ya dependerá de los interesados si es de forma electrónica o impresa. Por último se debe indicar quiénes serán los destinatarios de esta información así como el tipo de acceso que tendrán algunos interesados en las métricas obtenidas.

Además de estas plantillas se han incorporado la plantilla de análisis de *stakeholders* y la tarjeta de especificación de requisitos que se utilizan dentro de los procedimientos definidos para el programa de medición en el proceso de elicitación.

STAKEHOLDERS

Stakeholder			Tipo de conocimiento			Requisitos			
Clase	Rol	Nombre	Dominio de aplicación	Experiencia en el problema a resolver	Objetivos y metas del negocio y proyecto	Acordes con el objetivo	Necesarios	Propuestos	Definitivos
Total clase									
Total clase									
...									
...									

Esta plantilla fue elaborada a partir de la plantilla de Volere para el análisis de *stakeholders* (la plantilla esta disponible en www.volere.co.uk/stakeholder.xls y fue creada por la colaboración entre Suzanne Robertson, Chris Rupp y Ian Alexander). Esta plantilla (en el idioma español) se adjunta en el contenido de este anexo.

En los campos de clase y rol de *stakeholders* se utilizarán las propuestas en la plantilla de análisis de *stakeholders*.

INSTRUCCIONES DE LLENADO

Propósito	Identificar a los <i>stakeholders</i> necesarios y relevantes para el proyecto.
Cabecera	<p>Introducir:</p> <ul style="list-style-type: none"> ▪ Nombre del proyecto. ▪ Nombre del responsable del programa de medición. ▪ Versión de la plantilla. ▪ Fecha de elaboración. ▪ Autor. ▪ Revisor. ▪ Indicar si la plantilla es aprobada.
Stakeholders	<ul style="list-style-type: none"> ▪ Introducir las clases y roles de <i>stakeholders</i> que se identificaron para este proyecto considerando todas las clases que indica la plantilla de Volere. ▪ Introducir los nombres de cada <i>stakeholders</i> de acuerdo a cada rol y clase a la que pertenezca. ▪ Indicar en los campos de tipos de conocimiento con un símbolo (✓) si el stakeholder posee este tipo de conocimiento. ▪ Contabilizar los <i>stakeholders</i> por cada categoría (o clase), el total de todos los <i>stakeholders</i> se obtendrá con la ayuda de la hoja de cálculo utilizada para almacenar y procesar los datos. <p>Para cada <i>stakeholders</i> que participe en la elicitación introducir:</p> <ul style="list-style-type: none"> ▪ Los requisitos acorde con el objetivo del proyecto que proporcionó. ▪ Los requisitos necesarios para el proyecto que proporcionó. ▪ El número de requisitos propuestos. ▪ El número de requisitos definitivos. <p>Para generar el total de las 4 columnas de requisitos y obtener el total de <i>stakeholders</i> participantes se almacenará la información en la hoja de cálculo de EXCEL</p>
Tiempo	<p>Para cada <i>stakeholder</i> que participe en la elicitación introducir:</p> <ul style="list-style-type: none"> ▪ Nombre. ▪ Técnica de elicitación que se le aplicó para elicitar os requisitos. ▪ Requisitos elicitados. ▪ Hora inicio y hora fin de la reunión.

Control de requisitos

Nombre del proyecto: _____

Responsable del programa de medición: _____

Versión	Fecha	Autor	Revisó	Aprobado

OBJETIVO

Esta plantilla permite recolectar los indicadores y métricas para analizar las actividades de integración y refinamiento de información.

LISTA DE DESEOS Y NECESIDADES

Métrica	Valor		
	Versión <i>l</i> de la lista	...	Versión <i>n</i> de la lista
Requisitos en conflicto			
Requisitos base			
Requisitos agregados			
Requisitos modificados			
Requisitos eliminados			
TOTAL			

REQUISITOS

Versión de la lista de deseos y necesidades :			
Identificador del requisito	Origen (fuente del requisito)	Modificaciones	Número de versión

INSTRUCCIONES DE LLENADO

Propósito	Analizar la información obtenida, el refinamiento y organización de los requisitos.
Cabecera	<p>Introducir:</p> <ul style="list-style-type: none"> ▪ Nombre del proyecto. ▪ Nombre del responsable del programa de medición. ▪ Versión de la plantilla. ▪ Fecha de elaboración. ▪ Autor. ▪ Revisor. ▪ Indicar si la plantilla es aprobada.
Lista de deseos y necesidades	<ul style="list-style-type: none"> ▪ Indicar el número de requisitos que presentan algún conflicto. ▪ Introducir los requisitos que forma parte de la versión que se indique de la lista de deseos y necesidades. ▪ Registrar el número de requisitos agregados. ▪ Registrar el número de requisitos modificados. ▪ Registrar el número de requisitos eliminados. ▪ Obtener el total de requisitos (se obtendrá con la hoja de cálculo creada en EXCEL).
Requisitos	<p>Por cada requisito introducir:</p> <ul style="list-style-type: none"> ▪ Identificador para saber de que requisito se trata. ▪ El nombre del <i>stakeholder</i> que lo propuso para conocer su origen (pueden ser más de un <i>stakeholder</i>). ▪ En las campo de modificaciones, se introducirán los nombres de los <i>stakeholders</i> que han modificado el requisito. ▪ Las versiones que se tengan del requisito.

Reporte de resultados

[Nombre del proyecto]

[Versión]

[Fecha]

INSTRUCCIONES

Este documento es una plantilla que indica el formato a utilizar para la presentación de las métricas aplicadas en el proceso de medición que han sido propuestas en este trabajo de tesis.

Se debe considerar que

- *Los textos en color azul son indicaciones que deben eliminarse para sustituirse por los contenidos descritos en cada sección y subsección.*
- *Los textos entre del tipo “[texto]” permiten la sustitución directa de texto con el estilo adecuado a la sección.*
- *Las tablas deben tener el mismo formato que se indica al presentar la información contenida en ella.*
- *Las gráficas presentadas son un ejemplo del tipo que se puede presentar para una mejor visualización pero pueden ser sustituidas por otras.*

Contenido

Se generará una tabla de contenidos para visualizar el contenido del documento.

[Inserte aquí el texto]

Historial de Versiones

Versión	Fecha	Autores	Comentarios
[No.]	[Fecha]	[Lista de nombres]	[Lista de observaciones]

Aprobación del Documento

El presente documento ha sido aceptado y aprobado por:

Nombre	Cargo	Fecha	Firma
[Lista de nombres]	[Cargo]	[Fecha]	

Introducción

La introducción debe proporcionar una visión del desarrollo del programa de medición. Debe incluir el propósito del documento así como los objetivos del programa de medición (utilizando el paradigma Goal-Question-Metric (GQM)) y un breve resumen de los resultados y conclusiones obtenidas en el programa de medición.

[Inserte aquí el texto]

Propósito

En esta subsección se debe definir cuál es el propósito general del documento

[Inserte aquí el texto]

Paradigma Goal-Question-Metric

En esta subsección se especificarán los objetivos de medición que se han identificado, las preguntas formuladas para cada uno de ellos así como las medidas establecidas para dar respuesta a estas preguntas. Para la definición de cada uno de los objetivos de medición se utilizará la Tabla 1.

[Inserte aquí el texto]

Tabla 1. Plantilla para la definición de objetivos de medición, preguntas y métricas.

Objetivo	
Objeto	
Propósito	
Enfoque	
Perspectiva	
Entorno	
Preguntas	
Pregunta 1	
...	
Pregunta n	
Métricas	
Pregunta 1	<i>Métrica 1</i>
	...
	<i>Métrica n</i>
...	
Pregunta n	

Definición de medidas

Definición formal y textual de todas las medidas utilizadas para dar respuesta a los objetivos anteriores. Estas definiciones se pueden utilizar como referencia a considerar para la consulta, el análisis y la generación de las conclusiones de este documento.

Además de definir las medidas se debe indicar como fueron recolectadas incluyendo quiénes la recolectaron, con que frecuencia y que medio se utilizó para la recolección.

[Inserte aquí el texto]

Tabla 2. Plantilla para la definición de medidas.

Medida	Definición	Unidad de medida	Rango	Personal o rol encargado de su recolección	Frecuencia de recolección	Medio usado para recolectar

Visión general del documento

En esta subsección se explica brevemente cuál es la organización del así como su contenido y los resultados obtenidos.

[Inserte aquí el texto]

Análisis de datos

En este apartado se muestran todas las gráficas obtenidas de los datos recolectados, así como los métodos de análisis aplicados y las señales de alerta que se han identificado.

[Inserte aquí el texto]

Análisis de datos

En esta subsección se describen los datos analizados para cada uno de los objetivos de medición. Se utilizarán los métodos de análisis definidos en el Capítulo 3 y se incorporarán las gráficas analizadas en esta sección. Se incluirán todas las observaciones, interpretaciones que se hayan formulado.

*Para el primer objetivo de medición (**identificación de stakeholders**):*

Se debe graficar el número de stakeholders identificados por cada categoría (véase Figura 1). Para más detalles se pueden especificar los valores de la gráfica en una tabla (Tabla 3).

Tabla 3. Ejemplo de número de stakeholders por categoría.

Categoría	Número de stakeholders
<i>Interfaz tecnológica</i>	<i>13</i>
<i>Usuario</i>	<i>4</i>
<i>Cliente</i>	<i>5</i>
<i>Patrocinador</i>	<i>7</i>
<i>Miembros del equipo</i>	<i>0</i>

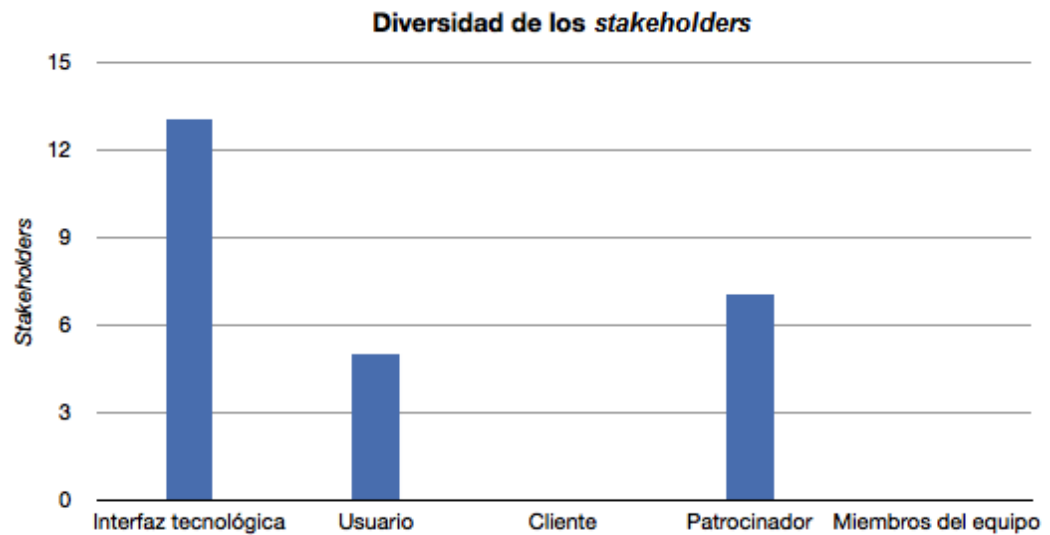


Figura 1. Ejemplo de una gráfica mostrando el número de *stakeholders* por cada categoría identificada.

En esta gráfica se puede visualizar en que categorías se han identificado un mayor número de stakeholders.

Ahora bien, para complementar la información referente a los stakeholders se graficará por cada requisito el número de categorías de stakeholders así como el número de versiones proporcionadas por cada stakeholder (véase Figura 2).

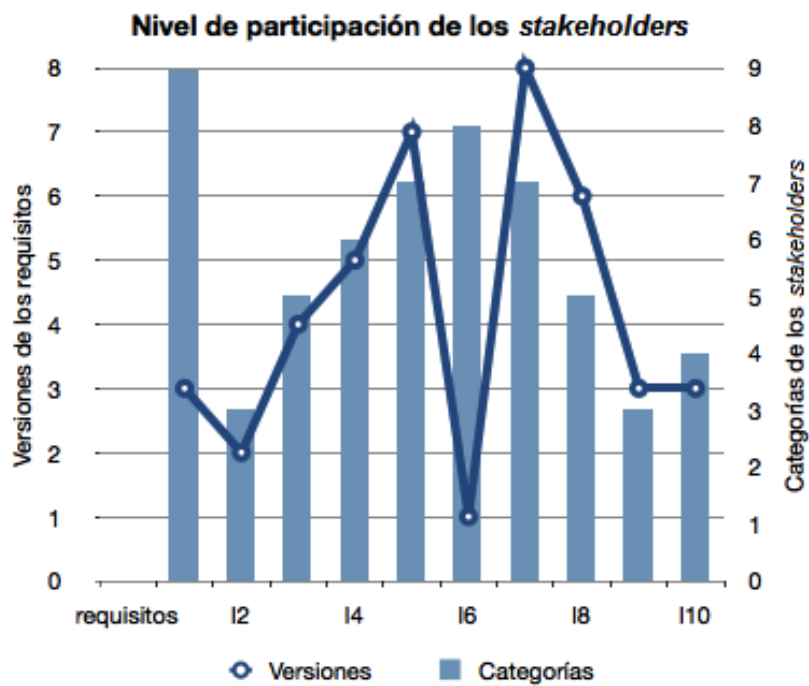


Figura 2. Ejemplo de una gráfica mostrando el número de versiones por requisitos y el número de categorías de *stakeholders*.

Para terminar la información presentada en este objetivo de medición se indicará el porcentaje o número de requisitos que están incompletos y que no son del todo correctos o que no están acordes con los objetivos del proyecto (este número se mostrará por cada stakeholder). La Figura 3 muestra el número de requisitos con algunos de los problemas mencionados y el número de requisitos definitivos que fueron proporcionados por los stakeholders.

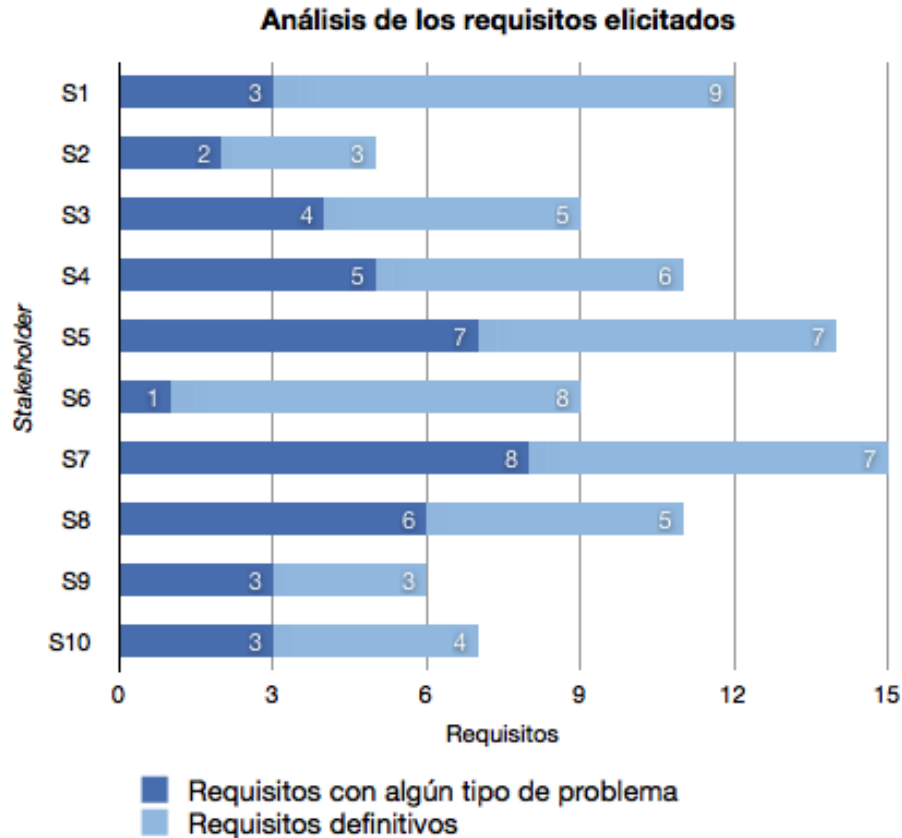


Figura 3. Requisitos que son necesarios para el proyecto de todos los que han sido propuestos por los *stakeholders*.

Con ayuda de estas 3 gráficas se podrá analizar de manera general la información proporcionada por cada *stakeholder*. Debajo de cada gráfica se describirán las observaciones e interpretaciones que se hayan realizado.

Para el segundo objetivo (*integración y refinamiento de la lista de deseos y necesidades*):

Se graficarán todas las versiones de cada requisito, proporcionadas por cada *stakeholder* (véase Figura 4).

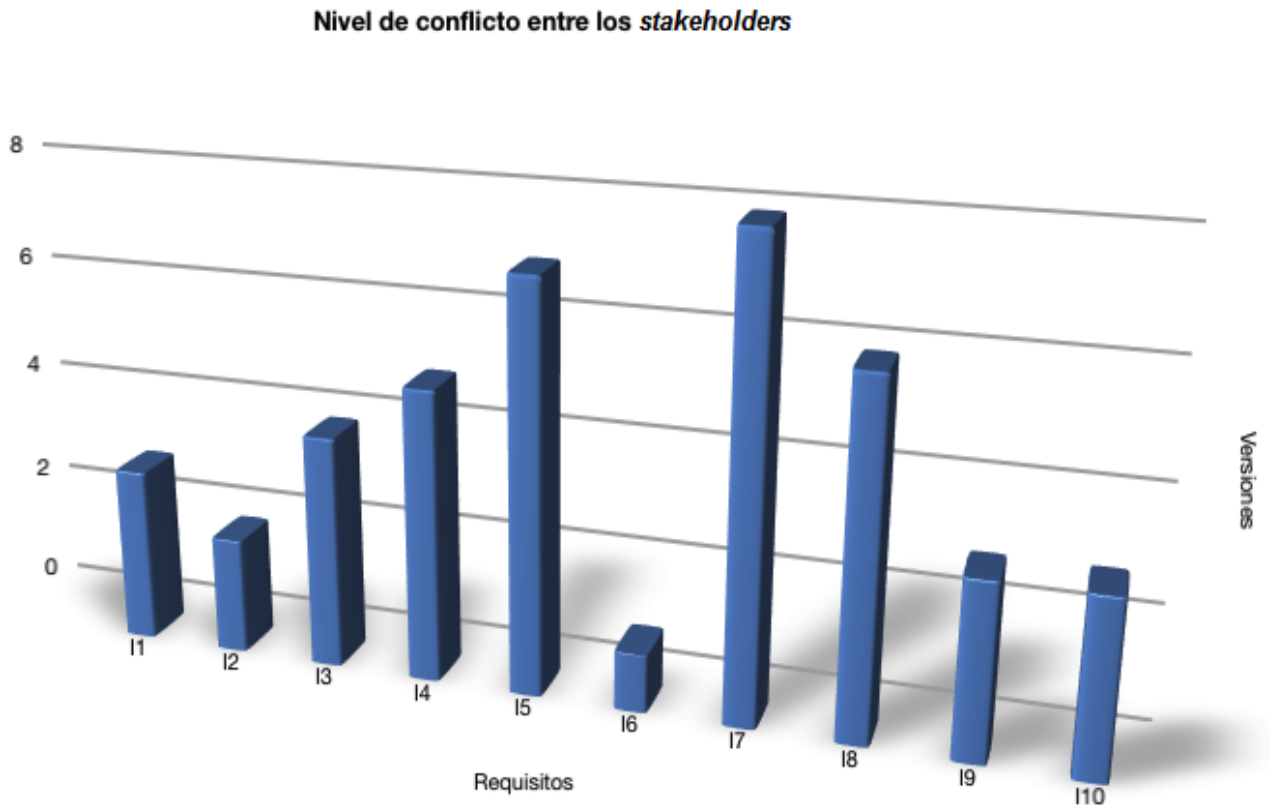


Figura 4. Versiones de cada requisito, obtenidas de diferentes *stakeholders*.

Para verificar como han evolucionado los requisitos obtenidos se graficará el número de los requisitos propuestos, el número de los requisitos acordes con los objetivos del proyecto y el número de los requisitos definitivos. Con esto se podrá visualizar el porcentaje que ha quedado en la versión final de la lista de deseos y necesidades contra el conjunto inicial de requisitos propuestos.

También se debe analizar el porcentaje de requisitos (véase Figura 6) que no fueron integrados y que presentan algún conflicto ocasionado por la falta de conciliación de los intereses de los stakeholders.

Por último, otro dato que se tiene que presentar es el número de versiones de la lista de deseos y necesidades, ya que esta indicará el nivel de refinamiento de los requisitos obtenidos.

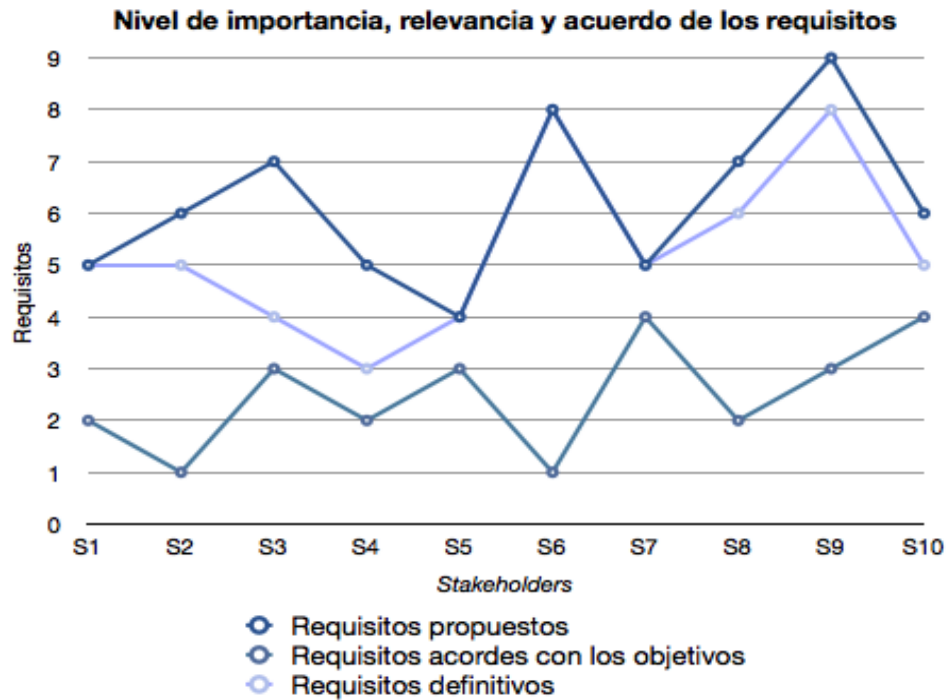


Figura 5. Número de requisitos por *stakeholders* clasificados por estado.



Figura 6. Porcentaje de requisitos integrados y en conflicto.

Señales de alerta

En este apartado se hará una lista con todas las señales de alerta por cada uno de los objetivos de medición identificadas en las dos secciones anteriores y las acciones o decisiones que se deben tomar (véase Tabla 3).

[Inserte aquí el texto]

Tabla 4. Ejemplo de señales de alerta identificadas.

Señales de alerta	Acciones a implementar
<i>Número pequeño de requisitos definitivos por cada stakeholder.</i>	<i>Análisis de los requisitos proporcionados para verificar que sean completos y correctos a partir de las técnicas utilizadas para elicitar.</i>

Conclusiones

En esta última sección se describirán las conclusiones obtenidas a partir de los métodos de análisis aplicados a los datos.

Una vez que los resultados del proceso de medición son analizados, se procederá a revisar el proceso de medición indicando si los objetivos de medición fueron cubiertos. Además se deben indicar tanto los beneficios como los costos económicos del programa de medición que se ha implementado (véase Tabla 4).

[Inserte aquí el texto]

Tabla 5. Ejemplo de análisis de costo- beneficio del programa de medición.

Costos	Beneficios
<i>Tiempo empleado para planificar y llevar a cabo el programa de medición.</i>	<i>Mejora en la calidad de los requisitos obtenidos.</i>
...	...

Tarjeta o cubierta de los requisitos

Requisito #: Identificador único	Tipo de requisito: El tipo de requisito	Número de caso de uso o evento: Lista de eventos/caso de uso que necesitan este requisito
Descripción: Una declaración de describa la intención del requisito.		
Razón: Una justificación del requisito.		
Autor: La persona que originó el requisito.		
Criterio de ajuste: Una medida del requisito tal que sea posible evaluar que la solución empata con el requisito original.		
Satisfacción del cliente: Grado de la felicidad de un stakeholder si el requisito es exitosamente implementado. Escala del 1="desinteresado" al 5="extremadamente agradecido".	Insatisfacción del cliente: Medida de la infelicidad de un stakeholder si el requisito no es parte del producto final. Escala del 1="duramente indiferente" al 5="extremadamente disgustado".	
Prioridad: Un valor (e.g. alta, media y baja) asignado por el cliente	Conflictos: Otros requisitos que no pueden ser implementados si este lo es.	
Materiales de apoyo: Indica los documentos que explican este requisito.		
Historial: Creación, cambios.		

Volere
Copyright ©Atlantic Systems Guild

Esta tarjeta es una guía para escribir cada requisito de forma atómica y también es llamada "tarjeta nieve".

9. Anexo D.- Requisitos y reporte de resultados

En las siguientes Tablas se describen los 51 requisitos que fueron elicitados del proyecto por el grupo experimental. También se anexa el reporte de resultados de la aplicación del programa de medición a este proceso de elicitación.

Tabla 59. Requisito funcional 1.

Requisito #: RF-001	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl almacenará y controlará la información de los proveedores con los que cuenta la empresa. La información que el sistema deberá de controlar es el RFC, nombre o razón social, dirección (calle, colonia, ciudad código postal, estado), teléfonos, lista de contactos, lista de productos, fecha de alta y datos de compra (crédito, monto, días, tipo, productos adquiridos, tiempo de entrega y comentarios).		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de los proveedores.		
Autor: Cliente		
Criterio de ajuste: La lista de proveedores puede ser modificada, ya sea agregando, modificando, y eliminando alguno de estos, además puede mostrar los datos de cada uno de ellos.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014 Usuario - Modificado el 22 de octubre del 2014 Consultor interno - Modificado el 22 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 60. Requisito funcional 2.

Requisito #: RF-001-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl permitirá bloquear a los proveedores.		
Razón: Bloquear a proveedores para no realizarles compras		
Autor: Cliente		
Criterio de ajuste: Los proveedores bloqueados serán los mismos que estén en la lista de proveedores a los que no se deben realizar compras		
Satisfacción del cliente: 4	Insatisfacción del cliente: 3	
Prioridad: Media	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 61. Requisito funcional 3.

Requisito #: RF-002	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar la información de los productos que maneja la empresa. La información que el sistema deberá de controlar es la clave del producto, línea (tipo de producto), imagen, marca, proveedor, proveedor sustituto, descripción del producto, imprimible, inventario, unidad, precio (1 al 5), moneda, impuesto, comisión, comisión especial, descuentos, mínimo, máximo, local donde se almacena, clave adicional, numero de piezas, numero de serie, kit, lote.		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de los productos.		
Autor: Cliente		
Criterio de ajuste: El sistema permitirá agregar, modificar, eliminar y consultar la información de productos, la cual debe corresponder con el inventario físico.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 20 de octubre del 2014		
Consultor interno - Modificado el 20 de octubre del 2014		

Tabla 62. Requisito funcional 4.

Requisito #: RF-003	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar la información de las líneas de productos que maneja la empresa. La información que el sistema deberá de controlar es la clave de la línea, el nombre de la línea y una descripción.		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de las líneas de productos.		
Autor: Usuario		
Criterio de ajuste: El sistema permitirá agregar, modificar, eliminar y consultar la información de las líneas de productos, la cual debe corresponder con los tipos de productos manejados en la organización.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 63. Requisito funcional 5.

Requisito #: RF-004	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar la información de las marcas de productos que maneja la empresa. La información que el sistema deberá de controlar es la clave de la marca y el nombre de la marca.		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de las marcas.		
Autor: Usuario		
Criterio de ajuste: El sistema permitirá agregar, modificar, eliminar y consultar la información de las líneas de productos, la cual debe corresponder con los tipos de productos manejados en la organización.		
Satisfacción del cliente: 4	Insatisfacción del cliente: 4	
Prioridad: Media	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 64. Requisito funcional 6.

Requisito #: RF-005	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá permitir el registro de traslado de productos entre locales. Para realizar los traslados el sistema deberá de almacenar el numero de traslado, fecha de traslado, lugar de origen, persona que entrega, persona que recibe, lugar de destino, persona que entrega en destino, persona que recibe en destino, producto, cantidad, descripción, existencia, serie – lote y observaciones.		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de los traslados.		
Autor: Cliente		
Criterio de ajuste: El sistema permitirá agregar, modificar, eliminar y consultar la información de los traslados, la cual debe corresponder con los traslados realizados y registrados en la bitácora de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario- Modificado el 21 de octubre del 2014		

Tabla 65. Requisito funcional 7.

Requisito #: RF-006	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar las compras que realiza la empresa. Para cada compra el sistema deberá de registrar el numero de compra, proveedor , local, numero de factura, remisión, fecha de compra, fecha de entrega, descuento, descuento neto, producto, tipo de cambio, cargos únicos, cantidad, descripción, precio, crédito, abono, pagado, numero de piezas, importe, numero de serie o lote, subtotal, impuestos, total y observaciones.		
Razón: El sistema permitirá crear, consultar, modificar y eliminar los datos de las compras.		
Autor: Cliente		
Criterio de ajuste: El sistema permitirá agregar, modificar, eliminar y consultar la información de las compras, la cual debe corresponder con las compras realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 20 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario- Modificado el 21 de octubre del 2014		
Consultor interno- Modificado el 21 de octubre del 2014		

Tabla 66. Requisito funcional 8.

Requisito #: RF-006-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl podrá cancelar una orden de compra y deberá confirmar la cancelación con el proveedor.		
Razón: Permitir la cancelación de las compras.		
Autor: Usuario		
Criterio de ajuste: Las ordenes de compra registradas por el sistema corresponderán con las realizadas.		
Satisfacción del cliente: 2	Insatisfacción del cliente: 2	
Prioridad: Baja	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 67. Requisito funcional 9.

Requisito #: RF-007	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar las ventas a clientes de la empresa. Cuando el usuario registre una venta, el sistema deberá de registrar el numero de venta, cliente, fecha de venta, local, vendedor, producto, tipo de cambio, cantidad, descripción, precio, importe, descuento, descuento neto, subtotal, impuestos, total y observaciones.		
Razón: Controlar información de las ventas.		
Autor: Cliente		
Criterio de ajuste: Las ventas registradas corresponderán con las ventas realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario – Modificado el 21 de octubre del 2014		

Tabla 68. Requisito funcional 10.

Requisito #: RF-007-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá controlar las notas de crédito de la empresa. Para cada nota de crédito el sistema deberá de registrar el numero de la nota de crédito, cliente, local, fecha de devolución, vendedor, numero de factura emitida en la venta original, producto, cargos únicos, cantidad, descripción, precio, piezas, importe, numero de serie o lote, descuento, descuento neto, subtotal, impuestos, total y observaciones.		
Razón: Controlar información de las notas de crédito.		
Autor: Cliente		
Criterio de ajuste: Las notas de crédito corresponderán con las ventas realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 69. Requisito funcional 11.

Requisito #: RF-007-2	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las devoluciones realizadas por los clientes de la empresa. Cuando el usuario registre una venta, el sistema deberá de registrar el numero de venta, cliente, fecha de venta, local, vendedor y la razón de la devolución.		
Razón: Controlar información de las devoluciones.		
Autor: Usuario		
Criterio de ajuste: Las devoluciones corresponderán con las ventas y devoluciones realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 70. Requisito funcional 12.

Requisito #: RF-008	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de poder generar una nota de remisión para el cliente que lo solicite conteniendo la fecha de remisión, numero de remisión, fecha de vencimiento, crédito, abono, pagado. Esta nota de remisión podrá ser creada en un formato que pueda ser impreso al estar conectado con una impresora de punto de venta.		
Razón: Generar notas de remisión .		
Autor: Cliente		
Criterio de ajuste: Las notas corresponderán con las ventas realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario – Modificado el 21 de octubre del 2014		

Tabla 71. Requisito funcional 13.

Requisito #: RF-009	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de poder generar una factura para el cliente que lo solicite conteniendo le numero de factura, fecha de la factura, fecha de vencimiento, crédito, abono, pagado y observaciones.		
Razón: Generar facturas..		
Autor: Cliente		
Criterio de ajuste: Las facturas corresponderán con las ventas realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 72. Requisito funcional 14.

Requisito #: RF-010	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de poder generar un backorder para el cliente en caso de ser requerido, conteniendo el tipo de backorder, descuento por artículo, anticipo y notas generales.		
Razón: Generar <i>backorder</i> .		
Autor: Cliente		
Criterio de ajuste: Las <i>backorders</i> generadas corresponderán con los pedidos realizados anticipadamente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 73. Requisito funcional 15.

Requisito #: RF-011	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las ventas al público en general de la empresa. Cuando el usuario registre una venta el sistema deberá de registrar el numero de venta, cliente, fecha de venta, local, vendedor, producto, tipo de cambio, cantidad, descripción, precio, importe, descuento, descuento neto, subtotal, impuestos, total.		
Razón: Controlar la información de las ventas de mostrador.		
Autor: Cliente		
Criterio de ajuste: Las ventas registradas corresponderán con las realizadas.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		

Historial: Creado el 21 de octubre del 2014 Usuario – Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild
---	---

Tabla 74. Requisito funcional 16.

Requisito #: RF-011-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las ventas al público en general de la empresa omitiendo el registro del cliente y registrando unicamente el numero de venta, fecha de venta, productos, cantidad, descuento, importe, forma de pago, remisión.		
Razón: Controlar la información de las ventas de mostrador.		
Autor: Usuario		
Criterio de ajuste: Las ventas registradas corresponderán con las realizadas.		
Satisfacción del cliente: 4	Insatisfacción del cliente: 4	
Prioridad: Media	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Cliente – Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 75. Requisito funcional 17.

Requisito #: RF-011-2	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las ventas a futuro. Cuando el usuario registre una venta a futuro, TeleControl deberá de registrar el numero de venta, fecha de venta, productos, cantidad, importe, descuento, descuento neto, forma de pago, remisión.		
Razón: Controlar la información de las ventas de mostrador.		
Autor: Usuario		
Criterio de ajuste: Las ventas registradas corresponderán con las ventas realizadas a futuro.		
Satisfacción del cliente: 3	Insatisfacción del cliente: 3	
Prioridad: Media	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Cliente – Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 76. Requisito funcional 18.

Requisito #: RF-012	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de otras salidas de producto del almacén diferentes a las ventas. Cuando el usuario registre una salida, TeleControl deberá de registrar el numero de salida, concepto, local, numero de factura, fecha de salida, producto, cantidad, descripción, precio, piezas, numero de serie o lote, importe, subtotal, impuestos, total, fecha de pago, abono, forma de pago, observaciones.		
Razón: Controlar la salida de productos que sea por un concepto diferente al de venta.		
Autor: Cliente		
Criterio de ajuste: Las salidas registradas corresponderán con el inventario de productos fisico		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Usuario – Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 77. Requisito funcional 19.

Requisito #: RF-013	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los clientes de la empresa. Para llevar el control de los clientes, TeleControl deberá de registrar la clave del cliente, nombre, calle, colonia, código postal, estado, teléfono, RFC, nombre de contacto 1, nombre de contacto 2, fecha de alta, nombre de vendedor, monto de crédito (aplicar porcentaje automático), plazo de pago, fecha de revisión de factura, día y hora para cobros, precios, tipo de cliente, descuento, correo, efectivo.		
Razón: Controlar la información de los clientes.		
Autor: Cliente		
Criterio de ajuste: Los datos controlados corresponderán con el catálogo de clientes de la empresa		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Usuario – Modificado el 21 de octubre del 2014 Consultor interno- Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 78. Requisito funcional 20.

Requisito #: RF-013-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: En caso de problemas con el cliente, TeleControl permitirá al usuario bloquear a dicho cliente.		
Razón: Bloquear ventas a los clientes.		
Autor: Cliente		
Criterio de ajuste: Los datos controlados corresponderán con el catálogo de clientes morosos o problemáticos de la empresa.		
Satisfacción del cliente: 3	Insatisfacción del cliente: 3	
Prioridad: Media	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 79. Requisito funcional 21.

Requisito #: RF-014	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los vendedores con los que cuenta la empresa. Para llevar el control, el sistema permitirá registrar la clave del vendedor, nombre, dirección (calle, colonia, ciudad), teléfono, correo, puesto, ingreso, CURP, RFC y comisiones. El sistema permitirá crear, consultar, modificar y eliminar los datos registrados.		
Razón: Controlar la información de los vendedores.		
Autor: Cliente		
Criterio de ajuste: Los datos controlados corresponderán con los datos registrados de su información personal		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 80. Requisito funcional 22.

Requisito #: RF-015	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de permitir a los usuarios levantar pedido de material. Para levantar un pedido de material, TeleControl deberá de registrar el numero de pedido, cliente, fecha de venta, local, vendedor, producto, tipo de cambio, cantidad, descripción, precio, importe, descuento, descuento neto, subtotal, impuestos, total.		
Razón: Controlar la información de los pedidos.		

Autor: Cliente	
Criterio de ajuste: Los pedidos registrados corresponderán con los pedidos realizados	
Satisfacción del cliente: 5	Insatisfacción del cliente: 5
Prioridad: Alta	Conflictos:
Materiales de apoyo:	
Historial: Creado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild
Usuario – Modificado el 21 de octubre del 2014	

Tabla 81. Requisito funcional 23.

Requisito #: RF-016	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de lo equipos que se encuentran en servicio. Para llevar el control, el sistema permitirá registrar el número de servicio, descripción, marca, modelo, serie, defecto, facturar, garantía, diagnostico, estado, persona que recibe, fecha de ingreso, nombre del cliente, teléfono, persona que entrega, fecha de entrega, reparación, fecha de reparación, informe de reparación, observaciones técnicas, coso del servicio, refacciones o material utilizados y su costo.		
Razón: Controlar la información de los equipos que se encuentran en servicio.		
Autor: Cliente		
Criterio de ajuste: Los equipos registrados corresponderán con los equipos que se encuentran físicamente en servicio.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	
Usuario – Modificado el 21 de octubre del 2014		

Tabla 82. Requisito funcional 24.

Requisito #: RF-017	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las actividades realizadas por los usuarios del sistema. En cada operación realizada por el usuario, el sistema deberá de registrar fecha, hora, usuario, sucursal y actividad u operación realizada.		
Razón: Registrar las actividades realizadas en el sistema por el usuario.		
Autor: Cliente		
Criterio de ajuste: Las actividades registradas en el sistema corresponderán con las que se hayan realizado.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 83. Requisito funcional 25.

Requisito #: RF-018	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los tipos de clientes con los que cuenta la empresa. Para llevar el control, TeleControl permitirá registrar la clave para el tipo de cliente, el nombre del tipo y la descripción del tipo de cliente.		
Razón: Controlar la información de los clientes.		
Autor: Cliente		
Criterio de ajuste: Las actividades registradas en el sistema corresponderán con las realizadas..		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	
Usuario - Modificado el 21 de octubre del 2014		

Tabla 84. Requisito funcional 26.

Requisito #: RF-019	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los cargos a clientes con los que cuenta la empresa. Para llevar el control, el sistema permitirá registrar el número del folio del cargo, clave del cliente, nombre del cliente, saldo global del cliente, fecha en que se aplica el cargo, monto del cargo, notas sobre el cargo, referencia, fecha limite de liquidación.		
Razón: Controlar la información de los cargos a clientes.		
Autor: Cliente		
Criterio de ajuste: Las cargos realizados a los clientes corresponderán con las ventas y pagos realizados.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 85. Requisito funcional 27.

Requisito #: RF-020	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los abonos a clientes con los que cuenta la empresa. Para llevar el control, TeleControl permitirá registrar el número del folio del cargo, clave del cliente, nombre del cliente, saldo global del cliente, fecha del abono, forma de pago, monto del abono, notas sobre el abono, referencia, fecha de vencimiento, imprimir recibo, concepto.		
Razón: Controlar la información de los abonos a clientes.		
Autor: Cliente		
Criterio de ajuste: Las actividades registradas en el sistema corresponderán con las ventas y abonos realizados.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 86. Requisito funcional 28.

Requisito #: RF-021	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los cargos a proveedores con los que cuenta la empresa. Para llevar el control, TeleControl permitirá registrar el número de registro, clave del proveedor, nombre del proveedor, saldo global del proveedor, monto del cargo, forma de pago, fecha de registro, notas, referencia.		
Razón: Controlar la información de los cargos a proveedores.		
Autor: Cliente		
Criterio de ajuste: Las actividades registradas en el sistema corresponderán con las compras y cargos realizados.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 87. Requisito funcional 29.

Requisito #: RF-022	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los abonos a proveedores con los que cuenta la empresa. Para llevar el control, el sistema permitirá registrar el número de abono, clave del proveedor, nombre del proveedor, saldo global del proveedor, monto del abono, fecha de abono, notas, referencia.		
Razón: Controlar la información de los abonos a proveedores.		
Autor: Cliente		

Criterio de ajuste: Las actividades registradas en el sistema corresponderán con las compras y abonos realizados.	
Satisfacción del cliente: 5	Insatisfacción del cliente: 5
Prioridad: Alta	Conflictos:
Materiales de apoyo:	
Historial: Creado el 21 de octubre del 2014 Usuario - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild

Tabla 88. Requisito funcional 30.

Requisito #: RF-023	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las cuentas de bancos con los que cuenta la empresa. Para llevar el control, el sistema permitirá registrar el número de cuenta, nombre del banco, tipo de moneda, número de cuenta contable asociada, número de próximo cheque, forma, estado de la cuenta.		
Razón: Controlar la información de las cuentas de bancos.		
Autor: Cliente		
Criterio de ajuste: Las datos de las cuentas de banco corresponderán con las cuentas de bancos de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 89. Requisito funcional 31.

Requisito #: RF-024	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control del catálogo de tipos de movimientos. Para llevar el control, el sistema permitirá registrar la clave del movimiento y el nombre o descripción del movimiento.		
Razón: Controlar la información de los tipos de movimientos.		
Autor: Cliente		
Criterio de ajuste: Los tipos de movimientos son los manejados en la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 90. Requisito funcional 32.

Requisito #: RF-025	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los movimientos bancarios. Para llevar el control de los movimientos bancarios, el sistema permitirá registrar el número de movimiento, fecha de registro del movimiento, tipo de movimiento, número de cuenta bancaria, nombre del banco, número de cuenta contable, saldo a favor, monto a girar o depositar, clasificación, concepto, número de cheque, fecha en que se hará valido el movimiento.		
Razón: Controlar la información de los movimientos bancarios realizados.		
Autor: Cliente		
Criterio de ajuste: El registro de movimientos deberán empatar con los realizados por la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 91. Requisito funcional 33.

Requisito #: RF-026	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de generar los estados de cuenta de las diferentes cuentas bancarias registradas. Para generar el estado de cuenta, TeleControl deberá de recibir el número de la cuenta, periodo de tiempo para generar el estado y permitir que solo se genere el estado con movimientos bancarios o en forma general.		
Razón: Generar estados de cuenta.		
Autor: Cliente		
Criterio de ajuste: Los estados de movimientos deberán empatar con los generados por los bancos.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 92. Requisito funcional 34.

Requisito #: RF-027	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de generar análisis de los gastos registrados por parte de la empresa. Para generar el análisis de gastos, el sistema deberá de recibir el número de la cuenta, periodo de tiempo para generar el estado y permitir que solo se genere el análisis de una sola cuenta o en forma general.		
Razón: Generar análisis de gastos.		
Autor: Cliente		
Criterio de ajuste: el análisis de gastos corresponderá con el que genera el contador.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 93. Requisito funcional 35.

Requisito #: RF-028	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las cuentas contables manejadas por la empresa. Para llevar el control de las cuentas contables, TeleControl permitirá registrar el número de la cuenta, nombre de la cuenta, subcuenta de, nombre de la cuenta inmediata superior, tipo de cuenta, clasificación, fecha de registro, detalle, grupos.		
Razón: Controlar las cuentas contables.		
Autor: Cliente		
Criterio de ajuste: El control de cuentas contables corresponderá con las registradas en los libros contables de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 94. Requisito funcional 36.

Requisito #: RF-029	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los periodos contables de la empresa. Para llevar el control de los periodos contables, el sistema permitirá registrar el nombre del periodo, fecha de inicio, fecha de término y el estado del periodo.		
Razón: Registrar los periodos contables.		
Autor: Cliente		

Criterio de ajuste: Los periodos contables deben estar acordes con los registrados en los libros contables de la empresa.	
Satisfacción del cliente: 5	Insatisfacción del cliente: 5
Prioridad: Alta	Conflictos:
Materiales de apoyo:	
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild

Tabla 95. Requisito funcional 37.

Requisito #: RF-030	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los grupos contables de la empresa. Para llevar el control de los grupos contables, el sistema permitirá registrar la clave del grupo y el nombre del grupo.		
Razón: Registrar los grupos contables.		
Autor: Cliente		
Criterio de ajuste: Los grupos contables deben estar acordes con los que existen en la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 96. Requisito funcional 38.

Requisito #: RF-031	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de la elaboración de pólizas de la empresa. Para llevar el control de las pólizas, TeleControl permitirá registrar la fecha de elaboración, número de póliza, tipo de póliza, concepto de póliza, elaboró, revisó y autorizó.		
Razón: Controlar las pólizas de la empresa.		
Autor: Cliente		
Criterio de ajuste: Las pólizas registradas serán las mismas que se emitan físicamente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 97. Requisito funcional 39.

Requisito #: RF-031-1	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de la elaboración de pólizas de banco de la empresa. Para llevar el control de las pólizas de banco, el sistema permitirá registrar la fecha de elaboración, número de póliza, tipo de póliza, concepto de póliza, elaboró, revisó y autorizó.		
Razón: Elaborar pólizas de banco.		
Autor: Cliente		
Criterio de ajuste: Las pólizas registradas serán las mismas que se emitan físicamente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 98. Requisito funcional 40.

Requisito #: RF-031-2	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de la elaboración de pólizas comerciales de la empresa. Para llevar el control de las pólizas comerciales, TeleControl permitirá registrar la fecha de elaboración, número de póliza, tipo de póliza, concepto de póliza, elaboró, revisó y autorizó.		
Razón: Elaborar pólizas comerciales.		
Autor: Cliente		
Criterio de ajuste: Las pólizas registradas serán las mismas que se emitan físicamente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 99. Requisito funcional 41.

Requisito #: RF-031-3	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de la elaboración de pólizas de cierre de la empresa. Para llevar el control de las pólizas de cierre, el sistema deberá de recibir el periodo de tiempo que se desea cerrar.		
Razón: Elaborar pólizas de cierre.		
Autor: Cliente		
Criterio de ajuste: Las pólizas registradas serán las mismas que se emitan físicamente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 100. Requisito funcional 42.

Requisito #: RF-032	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de generar análisis comerciales de la empresa. Para generar el análisis comercial, TeleControl deberá de recibir el periodo de tiempo para generar el análisis comercial.		
Razón: Elaborar análisis comercial.		
Autor: Cliente		
Criterio de ajuste: El análisis comercial que fue generado corresponde con el que es emitido por el contador.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Consultor interno - Modificado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Tabla 101. Requisito funcional 43.

Requisito #: RF-033	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los ensambles de kits de productos en la empresa. Para llevar el control de los kits de productos, el sistema deberá de registrar la clave del producto kit, número de local de partes, número de local de kits, número de existencias, cantidad de productos a ensamblar. El sistema permitirá consultar, modificar y eliminar los datos registrados.		
Razón: Control de ensambles.		
Autor: Cliente		
Criterio de ajuste: Los kits ensamblados corresponderán con el inventario físico de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	

Prioridad: Alta	Conflictos:
Materiales de apoyo:	
Historial: Creado el 21 de octubre del 2014 Usuario - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild

Tabla 102. Requisito funcional 44.

Requisito #: RF-034	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de los desensambles de kits de productos en la empresa. Para llevar el control de los kits de productos desensamblados, el sistema deberá de registrar la clave del producto kit, número de local de partes, número de local de kits, número de existencias, cantidad de productos a desensamblar.		
Razón: Control de desensambles.		
Autor: Cliente		
Criterio de ajuste: Los kits desensamblados corresponderán con el inventario físico de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Usuario - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 103. Requisito funcional 45.

Requisito #: RF-035	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de las configuraciones de kits de productos en la empresa. Para llevar el control de las configuraciones de los kits de productos, el sistema deberá de registrar la clave del producto kit, clave del producto que lo compone, descripción, cantidad.		
Razón: Configurar kits de productos.		
Autor: Cliente		
Criterio de ajuste: Los kits configurados corresponderán con el inventario físico de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Usuario - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 104. Requisito funcional 46.

Requisito #: RF-036	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de permitir realizar cotizaciones de los productos existentes. Para realizar las cotizaciones el sistema deberá de pedir al usuario el nombre del cliente, fecha de la cotización, vendedor, tipo de moneda, precio, producto, tipo de cambio, cantidad, descripción, precio, descuento, cargos únicos, importe, subtotal, impuestos, total.		
Razón: Realizar cotizaciones.		
Autor: Cliente		
Criterio de ajuste: Los precios de las cotizaciones corresponderán con los precios de los productos registrados.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014 Usuario - Modificado el 21 de octubre del 2014	Volere Copyright ©Atlantic Systems Guild	

Tabla 105. Requisito funcional 47.

Requisito #: RF-037	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de llevar el control de otras entradas a la empresa. Para cada entrada se registrará el producto, tipo, precio, razón de entrada y observaciones.		
Razón: Registrar otras entradas diferentes a compras.		
Autor: Cliente		
Criterio de ajuste: Las entradas registradas deben corresponder con el inventario físico de la empresa.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 106. Requisito funcional 48.

Requisito #: RF-038	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de registrar la información de los usuarios que usen el sistema. Por cada usuario se almacenará el nombre de usuario, password, tipo, y en caso de que sea un vendedor se ingresará su clave.		
Razón: Registrar la información básica de los usuarios.		
Autor: Cliente		
Criterio de ajuste: Los datos serán corroborados con los que hayan proporcionado los usuarios del sistema.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 107. Requisito funcional 49.

Requisito #: RF-039	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de registrar la información de los tipos de usuarios que usen el sistema. Por cada tipo de usuario se almacenará la clave, el nombre del tipo de usuario y descripción.		
Razón: Registrar la información básica de los tipos de usuarios.		
Autor: Cliente		
Criterio de ajuste: Los datos y permisos deberán corresponder con ls permisos otorgados por el cliente.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild
Usuario - Modificado el 21 de octubre del 2014		

Tabla 108. Requisito funcional 50.

Requisito #: RF-040	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de controlar todos los módulos del sistema dependiendo del acceso que se tengan de acuerdo al tipo de usuario.		
Razón: Control de permisos al sistema.		
Autor: Cliente		
Criterio de ajuste: Los permisos corresponderán a los que se les haya asignado dependiendo del tipo de usuario.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014		Volere Copyright ©Atlantic Systems Guild

Usuario - Modificado el 21 de octubre del 2014	
--	--

Tabla 109. Requisito funcional 51.

Requisito #: RF-041	Tipo de requisito: Funcional	Número de caso de uso o evento:
Descripción: TeleControl deberá de respaldar los datos del sistema una vez al mes y cuando el usuario con permisos se lo indique.		
Razón: Crear respaldos.		
Autor: Cliente		
Criterio de ajuste: Los respaldos deberán contener la información almacena en el sistema hasta la fecha indicada.		
Satisfacción del cliente: 5	Insatisfacción del cliente: 5	
Prioridad: Alta	Conflictos:	
Materiales de apoyo:		
Historial: Creado el 21 de octubre del 2014	<p style="text-align: right;">Volere Copyright ©Atlantic Systems Guild</p>	
Usuario - Modificado el 21 de octubre del 2014		

Reporte de resultados

Control de Información-Telecomunicaciones y Seguridad Electrónica

Versión 1.0

13 de octubre del 2014

Historial de Versiones

Versión	Fecha	Autores	Comentarios
1.0	13/10/2014	Myriam Karenina Reyes Sánchez	El documento fue creado considerando la versión 1.1 de la SRS del proyecto en cuestión.

Aprobación del Documento

El presente documento ha sido aceptado y aprobado por:

Nombre	Cargo	Fecha	Firma
Ing. Alberto Martínez Ramírez	Jefe de Proyecto	13/10/2014	
Dra. Carla Lenín Pacheco Agüero	Director del programa de medición	13/10/2014	

Introducción

El programa de medición fue implementado en el proyecto Control de Información-Telecomunicaciones y Seguridad Electrónica en la creación del producto TeleControl, para el cuál se consideraron 2 versiones de la lista de deseos y necesidades obtenidas en la etapa de elicitación.

Propósito

El propósito de ese documento es presentar los valores obtenidos de las métricas y el análisis realizado a partir de estos valores. En este documento se indica el formato que se debe seguir para presentar los datos recolectados, así como las gráficas generadas para una mejor visualización y análisis. Las conclusiones obtenidas del análisis serán dadas a conocer al jefe de proyecto así como las señales de alerta que puedan tener un impacto negativo en el proyecto.

Paradigma *Goal-Question-Metric*

El objetivo de aplicar el proceso de medición a un proyecto software de una empresa es la validación de las métricas que se han propuesto en este trabajo de tesis. Por lo tanto, los dos objetivos de medición que se han identificado y no se han cubierto en la etapa de elicitación se describen en la Tabla 1. Las preguntas formuladas y relacionadas a cada objetivo se describen en la misma tabla junto con las medidas establecidas que darán respuesta a estas preguntas. Estos objetivos ya fueron descritos en la sección 3.3.2 del Capítulo 3.

Tabla 1. Plantilla para la definición del primer objetivo de medición.

Objetivo	
Objeto	Analizar el proceso de identificación de <i>stakeholders</i> .
Propósito	Con el objetivo de mejorarlo e identificar a las fuentes de requisitos necesarias y relevantes.
Enfoque	Con respecto a la plantilla de Volere.
Perspectiva	Desde el punto de vista del ingeniero de requisitos.
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y dónde el equipo de desarrollo este integrado por menos de 10 personas.
Preguntas	
Pregunta 1	¿Cuántos <i>stakeholders</i> son considerados para la elicitación de requisitos?
Pregunta 2	¿Cuántos <i>stakeholders</i> propusieron requisitos relevantes y prioritarios para el proyecto?
Pregunta 3	¿Cuántos <i>stakeholders</i> propusieron requisitos necesarios o urgentes para el proyecto?
Pregunta 4	¿Qué tan relevantes son los <i>stakeholders</i> ?
Pregunta 5	¿Se están elicitando todas las perspectivas del sistema?
Pregunta 6	¿Se han elicitado todos los deseos y necesidades de todos los <i>stakeholders</i> participantes?
Pregunta 7	¿Se han elicitado los requisitos adecuados?
Métricas	
Pregunta 1	Número de <i>stakeholders</i> que han participado en la elicitación (por categoría y en total).
Pregunta 2	Número de <i>stakeholders</i> que propusieron requisitos acorde con el objetivo del proyecto.
	Número de requisitos propuestos por cada <i>stakeholder</i> .

	Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .
Pregunta 3	Número de <i>stakeholders</i> que propusieron requisitos urgentes para el proyecto.
	Número de requisitos propuestos por cada <i>stakeholder</i> .
	Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .
Pregunta 4	Número de <i>stakeholders</i> que conocen el dominio de aplicación del proyecto a desarrollar.
	Número de <i>stakeholders</i> que tienen experiencia sobre el problema a resolver.
	Número de <i>stakeholders</i> que conocen los objetivos del negocio.
Pregunta 5	Número total de categorías de <i>stakeholders</i> que han participado en la elicitación.
	Número de versiones de cada requisito elicitado.
	Número de requisitos elicitados por cada <i>stakeholder</i> .
Pregunta 6	Número de requisitos que se especificaron completamente.
	Número de requisitos revisados para estar acordes con los objetivos (proyecto y negocio).
	Número de cuestiones sin resolver o faltantes en el documento.
	Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades ya analizada.
Pregunta 7	Número total de requisitos elicitados.
	Número de técnicas de elicitación que se utilizaron.
	Número de <i>stakeholders</i> que participaron en cada técnica de elicitación.
	Número de requisitos que se elicitaron por cada técnica usada.

Tabla 2. Plantilla para la definición del segundo objetivo de medición.

Objetivo	
Objeto	Analizar el proceso de integración y refinamiento de información.
Propósito	Con el propósito de verificar que la información esté completa en la SRS.
Enfoque	Con respecto a la identificación adecuada de las metas del negocio, problema a resolver, dominio de aplicación y necesidades
Perspectiva	Desde el punto de vista de los <i>stakeholders</i> .
Entorno	En el contexto de algún proyecto software pequeño (6 a 12 meses y con 50000-100000 LOC) y donde el equipo de desarrollo este integrado por menos de 10 personas.
Preguntas	
Pregunta 1	¿Cuántos puntos de vista por cada <i>stakeholder</i> serán integrados a la lista de deseos y necesidades?
Pregunta 2	¿En que requisitos no se pudo llegar a un acuerdo?
Pregunta 3	¿Cuántas veces se ha modificado la lista de deseos y necesidades?
Métricas	
Pregunta 1	Número de diferentes versiones de cada requisito (incluyendo su origen).
Pregunta 2	Número de requisitos en conflicto.
Pregunta 3	Número de versiones de la lista de deseos y necesidades.
	Número de requisitos agregados.

	Número de requisitos modificados.
	Número de requisitos eliminados.
	Número total de requisitos elicitados.

Definición de medidas

Las medidas que se recolectaron se definen en la Tabla 2, para una mejor referencia en ella se proporciona su definición, unidad de medida, el rango de valores aceptables, cuál fue la frecuencia de su recolección y que medio se utilizó para su recolección. Todos estos datos son proporcionados para que se tenga una referencia para la consulta, el análisis y la generación de las conclusiones.

En cuanto a la frecuencia de recolección, los datos fueron recolectados durante la etapa de elicitación y la persona encargada de registrar los datos fueron los ingenieros de requisitos participantes en el mismo proceso.

Visión general del documento

En la siguiente sección se desarrolla el análisis de los datos recolectados, mostrando las gráficas que fueron generadas y datos relevantes. Después se indican las señales de alerta y las acciones a implementar para .Por último se describen las conclusiones obtenidas a partir de los análisis realizados.

Tabla 3. Plantilla para la definición de medidas.

Medida	Definición	Unidad de medida	Valor	Valor	Rango	Medio usado para recolectar
Diversidad de los <i>stakeholders</i> participantes.	Es el número de <i>stakeholders</i> que se han identificado por cada categoría utilizando la plantilla de Volere.	Número de <i>stakeholders</i> participantes (por categoría).	1 mantenimiento 1 usuario 1 cliente 1 consultor interno 1 consultor externo	1 por cada categoría (usuario, cliente y miembros del equipo)	1-3	Plantillas
Nivel de importancia de los requisitos.	El número de requisitos acordes con el objetivo que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos acordes con el objetivo del proyecto, por cada <i>stakeholder</i> .	Mantenimiento- 1.9% Usuario – 11.7% Cliente – 80.35% Consultor interno – 0% Consultor externo – 3.9%	38 por solo un <i>stakeholder</i> , 84.4 %	1-50% del total.	Plantillas
Nivel de demanda de los requisitos propuestos por cada <i>stakeholders</i> .	Es el número de <i>stakeholders</i> que se han identificado por cada categoría (utilizando la plantilla de Volere) y que han propuesto requisitos que deben ser atendidos inmediatamente (su demanda es importante y el tiempo de atención es crítico).	Número de <i>stakeholders</i> que propusieron requisitos urgentes o necesarios para el proyecto.	3 (cliente, usuario y consultor interno)	1	1-2 por categoría.	Plantillas
Nivel de relevancia de los requisitos.	El número de requisitos necesarios para el proyecto que han sido propuestos por cada uno de los <i>stakeholders</i> .	Número de requisitos necesarios propuestos por cada <i>stakeholder</i> .	84.31%	45 requisitos- 100%	1-50% del total.	Plantillas
Nivel de acuerdo entre los <i>stakeholders</i> .	El número de requisitos que han sido aceptados por todos los participantes.	Número de requisitos definitivos propuestos por cada <i>stakeholder</i> .	100%	45 requisitos – 100%	80%-100% del total	Plantillas
Conocimiento del dominio de aplicación por parte de los	Es el número de <i>stakeholders</i> que poseen conocimiento referente al dominio de aplicación.	Número de <i>stakeholders</i> que conocen el dominio de aplicación.	3	1	1-Número total de <i>stakeholders</i>	Plantillas

<i>stakeholders.</i>						
Conocimiento del problema por parte de los <i>stakeholders.</i>	Es el número de <i>stakeholders</i> que poseen conocimiento del problema a resolver.	Número de <i>stakeholders</i> que conocen el problema a resolver.	2	1	1-Número total de <i>stakeholders</i>	Plantillas
Conocimiento de los objetivos del negocio por parte de los <i>stakeholders.</i>	Es el número de <i>stakeholders</i> que poseen conocimiento de los objetivos del negocio.	Número de <i>stakeholders</i> que conocen los objetivos de negocio.	2	1	1-Número total de <i>stakeholders</i>	Plantillas
Categorías de los <i>stakeholders.</i>	Es el número de categorías en la que los <i>stakeholders</i> participantes han sido caracterizados.	Número total de categorías de <i>stakeholders.</i>	5	1	1-14	Plantillas
Nivel de conflicto entre los <i>stakeholders.</i>	Es el número de versiones existentes para cada requisito por cada una de las versiones de las listas de deseos y necesidades.	Número de versiones diferentes de cada uno de los requisitos.	Un promedio de 2 versiones	1 para todos los requisitos	1-Número total de <i>stakeholders</i>	Plantillas
Productividad de cada <i>stakeholder.</i>	El número total de requisitos definitivos que se han elicitado por cada uno de los <i>stakeholders.</i>	Número de requisitos elicitados por cada <i>stakeholder.</i>	Mantenimiento- 1 Usuario – 6 Cliente – 42 Consultor interno - 0 Consultor externo - 2	45 de un solo <i>stakeholder</i>	1-100%	Plantillas
Completitud de los requisitos.	El número de requisitos que están completos, es decir bien especificados (no tienen detalles que estén por ser determinados).	Número de requisitos que se especificaron completamente.	51	45	80%-100%	Plantillas
Requisitos correctos.	Es el número de requisitos que fueron revisados para que estuvieran acordes con los objetivos del proyecto y negocio.	Número de requisitos revisados para estar acordes con los objetivos (proyecto y negocio).	51	38	80%-100%	Plantillas
Densidad de omisiones en la lista de deseos y necesidades.	Es el número de cuestiones omisas en el documento que contiene la lista de deseos y necesidades: ▪ Datos sin especificar.	Número de faltas por requisito.	Dos faltas en 1 de los requisitos fueron identificadas	Ninguna	Ninguna	Plantillas

	<ul style="list-style-type: none"> ▪ Información que este pendiente o por determinar. ▪ Entidades u objetos que son referenciados pero no son descritos. 					
Nivel de aprobación por parte de los <i>stakeholders</i> de la lista de deseos y necesidades.	Es el número de <i>stakeholders</i> que ha aprobado y firmado alguna versión de la lista de deseos y necesidades.	Número de <i>stakeholders</i> que han aprobado la lista de deseos y necesidades.	5	2	Número total de <i>stakeholders</i>	Plantillas
Tamaño de la lista de deseos y necesidades.	Se incluirá el número total de requisitos que fueron elicitados: <ul style="list-style-type: none"> ▪ Requisitos modificados. ▪ Requisitos agregados. ▪ Requisitos eliminados. 	Número total de requisitos elicitados.	51	45	Número total de <i>stakeholders</i> -infinito.	Plantillas
Eficacia de la aplicación de técnicas de elicitación.	Es el número de técnicas aplicadas a lo largo del proceso de elicitación.	Número de técnicas de elicitación utilizadas.	2	1	1-Total de técnicas existentes.	Plantillas
	Es la cantidad de <i>stakeholders</i> que participaron en cada técnica de elicitación.	Número de <i>stakeholders</i> que participaron en cada técnica de elicitación.	2 en entrevistas y los 5 en el grupo de enfoque	1	1-Número total de <i>stakeholders</i>	
Nivel de desacuerdo entre los <i>stakeholders</i>	Es el número de los requisitos que presenten alguno de los siguientes problemas: <ul style="list-style-type: none"> ▪ Diferentes versiones por cada <i>stakeholder</i>. ▪ El requisito no haya sido especificado completamente por algún problema entre los <i>stakeholders</i>. 	Número de requisitos en conflicto.	5	0	0-20% del total	Plantillas
Nivel de refinamiento	Número de las versiones existentes de la lista de deseos y	Número de versiones de la lista de deseos y	2	1	1-Número total de	Plantillas

de la lista de deseos y necesidades.	necesidades.	necesidades.			técnicas de elicitación.	
Volatilidad de los requisitos.	Número de requisitos que fueron agregados a la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos agregados.	5	31	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.	Plantillas
	Número de requisitos que fueron modificados en la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos modificados.	30	14	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.	Plantillas
	Número de requisitos que fueron eliminados de la versión de la lista de deseos y necesidades que se estén analizando.	Número de requisitos eliminados.	3	0	1-Número de <i>stakeholders</i> que participaron en la elicitación por esta ocasión.	Plantillas
Participación de los <i>stakeholders</i> .	Número de todos los <i>stakeholders</i> participantes durante todo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron activamente durante la elicitación	5	1	1-Total de <i>stakeholders</i> identificados al inicio.	Plantillas
Frecuencia de elicitación de requisitos	Indica el tiempo en el que se ha llevado a cabo el proceso de elicitación.	Número de <i>stakeholders</i> que participaron en la elicitación por día.	2 el primer día y los 5 en la segunda sesión	1 por día	15%-25% del total de <i>stakeholders</i> identificados	Plantillas

Tamaño de la lista de deseos y necesidades.	El número de requisitos que se han elicitado en la última versión de la lista de deseos y necesidades.	Número de requisitos elicitados.	51	45	Número total de <i>stakeholders</i>	Plantillas
---	--	----------------------------------	----	----	-------------------------------------	------------

Análisis de datos

Dentro de esta sección se describen todas las gráficas generadas a partir de los datos que han sido recolectados, se describe cómo fue hecho el análisis, es decir, qué métodos fueron utilizados y cuáles fueron los resultados obtenidos a partir de este análisis, lo que incluye las señales de alerta que fueron identificadas.

Análisis de métricas obtenidas

En esta subsección se describen los datos analizados para cada uno de los objetivos de medición. El análisis comienza en la Tabla 2 que contiene todos los valores de las métricas que han sido definidas, estos valores son necesarios para poder elaborar las gráficas y tener una mejor visualización de ellos para hacer un análisis más completo.

Para el primer objetivo de medición (**identificación de *stakeholders***):

El número de *stakeholders* identificados por cada categoría se puede visualizar en la gráfica (véase Figura 1), pero como se puede ver sólo hay un *stakeholder* por cada categoría (véase Tabla 4).

Tabla 4. Número de *stakeholders* por categoría.

Categoría	Número de <i>stakeholders</i>
Mantenimiento	1
Usuario	1
Cliente	1
Consultor interno	1
Consultor externo	1

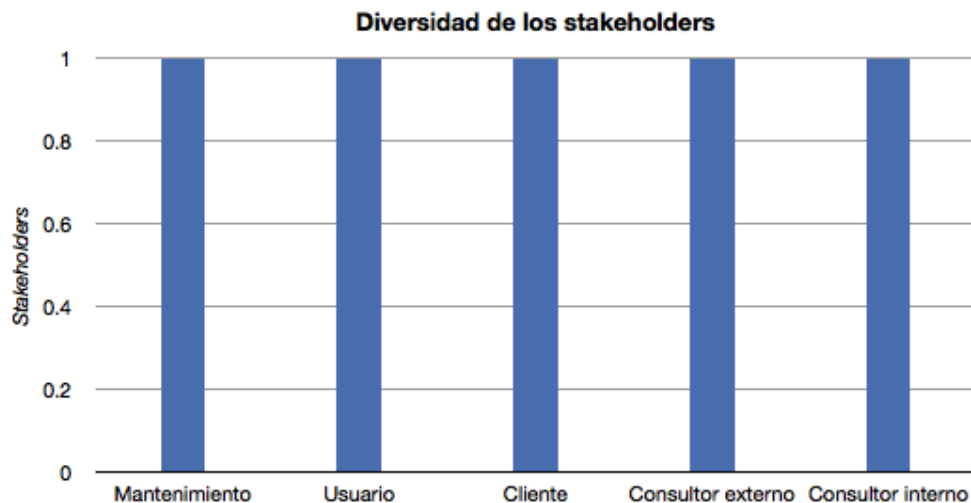


Figura 1. Número de *stakeholders* por cada categoría identificada.

En esta gráfica se puede observar que se han identificado a 5 *stakeholders* para que participen en la etapa de elicitación, sin embargo lo que nos indican la métrica “Participación

de los *stakeholders*” es que solo dos de ellos participaron activamente en el proceso, ya que ellos proporcionaron más del 50% de los requisitos de la última versión de la lista de deseos y necesidades. Aunque puede darse el caso que éstos *stakeholders* no proporcionen toda la información necesaria para el proyecto, este no es el caso, ya que la lista de requisitos por parte del ha sido aprobada por todas los participantes, y el hecho de que los otros tres *stakeholders* no hayan participado tan activamente se debe a que solo fueron requeridos para cuestiones muy puntuales. Otro valor a considerar es que aunque se ha presentado algún conflicto entre los requisitos se ha podido llegar a un acuerdo entre ellos, por lo tanto se ha logrado conciliar los intereses para poder obtener el requisitos que mejor represente sus necesidades.

Como se puede ver en la Figura 2, la gráfica indica que por cada uno de los requisitos (para todos los 52 requisitos elicitados) existe un número de versiones por cada uno de ellos. Solamente para 13 requisitos existen dos versiones, y estas han sido proporcionadas por difentes *stakeholders* lo que nos indica que se están considerando difeentes puntos de vista.

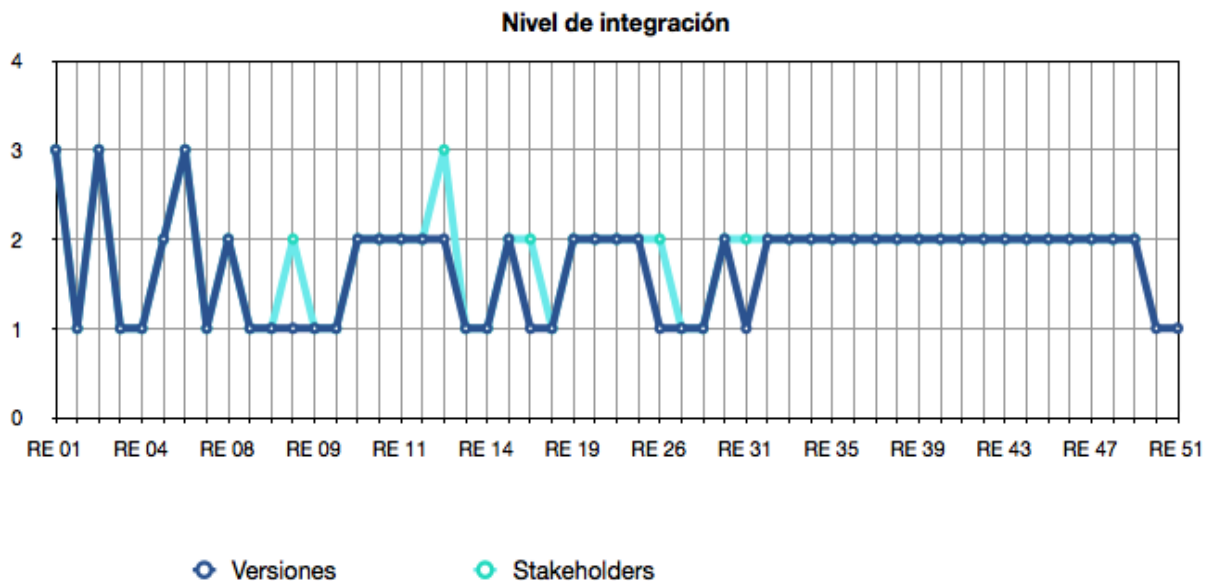


Figura 2. Número de versiones por requisitos y el número de categorías identificadas.

Ahora bien, las medidas obtenidas respecto a que los requisitos sean completos y correctos por si solas indican que estos lo son, y complementando esta información al considerar los factores anteriores (que diferentes *stakeholders* han proporcionado los requisitos y que han sido aprobados por todos los participantes) se puede dar por hecho que en su mayoría estos son completos y correctos. Lo que si se puede afirmar con seguridad es que solo un requisito quedo pendiente por especificar en dos de sus datos y que la mayoría de ellos están acordes con el objetivo del negocio y del proyecto.

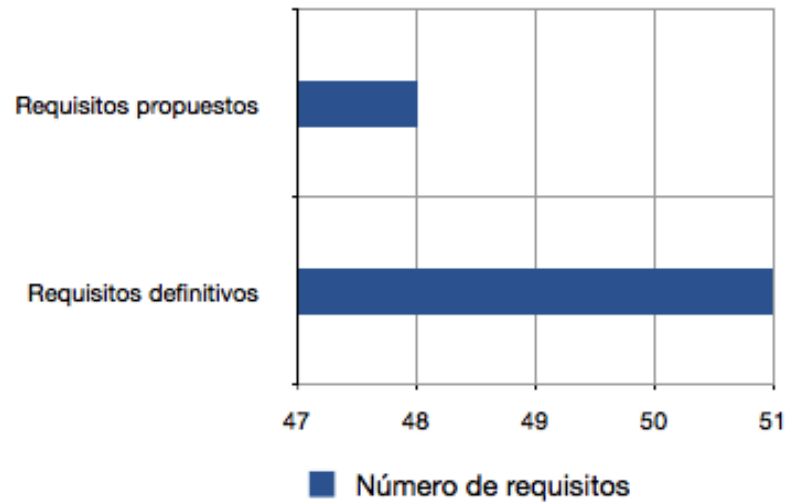


Figura 3. Requisitos que son necesarios para el proyecto de todos los que han sido propuestos por los *stakeholders*.

De todos los requisitos que han sido propuestos se agregaron cinco, se modificaron 30, y se eliminaron tres, por lo tanto los requisitos fueron modificados, lo que indica que los requisitos han sido refinados y su información es más completa.

Para el segundo objetivo (**integración y refinamiento de la lista de deseos y necesidades**):

En este caso al graficar todas las versiones de cada requisito, proporcionadas por cada *stakeholder* (véase Figura 2) es decir los requisitos proporcionados solamente por el cliente se puede ver que para 13 requisitos se tienen dos versiones y que esta segunda versión contiene más información que la primera y una mejor especificación de cada uno de ellos.

En cuanto al análisis de la evolución de los requisitos se considera que todos los requisitos son necesarios (aunque no todos tienen la misma prioridad para el proyecto), y que los requisitos definitivos exceden por un 4% a los que inicialmente se habían propuesto.

La gráfica de la Figura 4 representa el porcentaje de requisitos que fueron integrados, y indicando las diferentes versiones de cada requisito.

En cuanto al nivel de refinamiento solo existen dos versiones de la lista de deseos y necesidades para las cuales no se puede garantizar que la información este completa. La última versión contiene 51 requisitos los cuáles han sido revisados por todos los *stakeholders* y han sido aprobada por todos ellos.

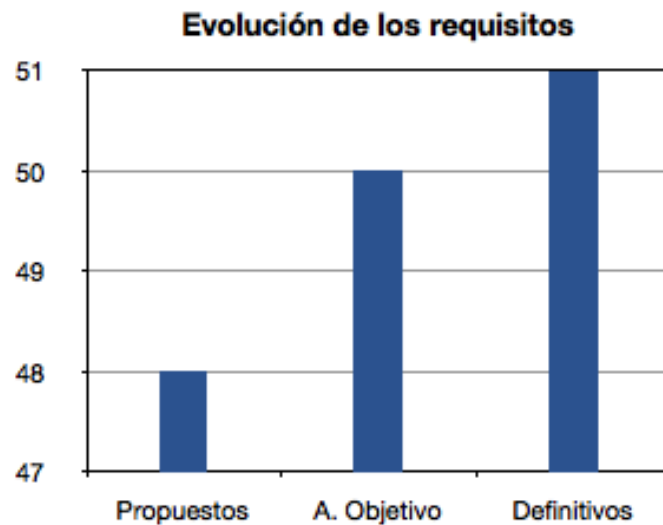


Figura 4. Número de requisitos propuestos, acordes con el objetivo y definitivos.

Señales de alerta

Las señales de alerta nos indican que es lo que está fuera del rango permitido o aceptable y el cual también se ha indicado al definir cada una de las medidas. Estas señales de alerta nos indican que algo puede tener un impacto no deseado que puede alterar los requisitos que se han obtenido y por lo cual se recomienda emprender acciones para generar la lista de deseos y necesidades lo más cercana a las necesidades de los *stakeholders* y a un proceso efectivo de elicitación.

Tabla 5. Señales de alerta que ha sido identificadas.

Señales de alerta	Acciones a implementar
Todos los requisitos han sido proporcionados en su mayoría por un solo <i>stakeholder</i> de los 5 que han sido identificados.	Análisis de los requisitos proporcionados para confirmar que es el más involucrado con el proyecto.
Existen diferentes versiones de los requisitos para ser integrados.	Revisión de la lista de deseos y necesidades con todos los <i>stakeholders</i> que han sido identificados. Aprobación de la lista de deseos y necesidades por parte de todos los <i>stakeholders</i> .
Número pequeño de versiones de la lista de deseos y necesidades.	Refinamiento de la información obtenida. Integración de la información obtenida por cada uno de los <i>stakeholders</i> .
Todos los requisitos tienen en su mayoría la misma prioridad.	Revisión de los requisitos para determinar mejor su prioridad dependiendo de su urgencia para el proyecto así como el nivel de dependencia del proyecto.

Conclusiones

Todos los datos analizados apuntan a que solo ha participado en su mayoría un *stakeholder* de los 5 que se han identificado en un comienzo del proyecto, por lo tanto se debe analizar esta situación ya que los requisitos obtenidos para confirmar que estos tengan todas las propiedades de calidad que deben poseer.

Los datos obtenidos cubren los objetivos de medición hasta cierto punto, ya que el proyecto es pequeño y debido a las características propias de este, en el que solo ha participado un pequeño número de *stakeholders* y dentro de una sola fase de elicitación, no se pudieron obtener más medidas para el análisis de otras métricas, pero los datos recolectados son suficientes para indicar que puede estarse realizando de manera incorrecta y que acciones se pueden implementar. Los costos y beneficios identificados hasta el momento de haber implementado el proceso de medición se pueden ver en la Tabla 4.

Tabla 6. Análisis de costo- beneficio del programa de medición.

Costos por actividad	Beneficios
<ul style="list-style-type: none"> • Identificación de <i>stakeholders</i>. <p>El tiempo empleado para desarrollar esta actividad fue, para el grupo de control de 2 días con 13 horas (aún cuando no se identificaron correctamente a todos los involucrados) mientras que el grupo experimental se tardó 1 día con 6 horas identificando a cuatro <i>stakeholders</i> además del equipo de desarrollo.</p>	<p>El tiempo invertido en la actividad de identificación de los <i>stakeholders</i> se vió reducido en un 51% con las métricas propuesta, es decir 1 día con 7 horas menos. Lo que resultó en un ahorro de \$381.94 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>
<ul style="list-style-type: none"> • Obtención de la lista de deseos y necesidades de los <i>stakeholders</i>. <p>Los ingenieros de requisitos del grupo de control utilizaron 15 días con 10 horas para aplicar las sesiones de elicitación (no se cuenta con los datos de qué tipo de técnicas fueron aplicadas con los <i>stakeholders</i> debido a que se tuvieron problemas de comunicación y entendimiento), mientras que el grupo experimental lo hizo en 3 días con 8 horas con las dos técnicas seleccionadas de acuerdo con las características de los <i>stakeholders</i> identificados (entrevistas y grupos de enfoque).</p>	<p>El tiempo empleado por el grupo experimental para la obtención de la lista de deseos y necesidades a través de las sesiones de elicitación fue de un 78%, es decir 12 días con 2 horas menos. Por lo tanto se pudo ahorrar \$3573 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>
<ul style="list-style-type: none"> • Integración, refinamiento e integración de la información. <p>La actividad de refinamiento e integración de información para el grupo experimental fue de 1 día, distribuyéndose de la siguiente forma: 23 horas para la modificación de algunos de los requisitos durante las mismas sesiones de elicitación (por las características propias de las técnicas, que permiten hacerlo) y las otras 5 para eliminar la ambigüedad e inconsistencia de los requisitos. Mientras que el grupo de control realizó esta actividad en varias sesiones, donde se emplearon 8 días.</p>	<p>En esta actividad, el tiempo empleado por el grupo de control fue reducido en un 88%, es decir 6 días con 20 horas menos para el grupo experimental, lo que conlleva una reducción de \$2020.50 tomando en cuenta un salario promedio para un analista de requisitos en México de \$8871 mensuales (SG, 2013).</p>

10. Bibliografía fundamental

1. ACM (2013). Computing Degrees & Careers, Software Engineering. http://computingcareers.acm.org/?page_id=12. Último acceso: Marzo 2013.
2. Abramson, D. & Pike, L. (2011). When Formal Systems Kill: Computer Ethics and Formal Methods. *APA Newsletter on Philosophy and Computers*, 11(1), 1-17.
3. Abran A., Moore J. W., Bourque P. & Dupuis R. (2004). *SWEBOK: Guide to the Software Engineering Body of Knowledge v3.0*, IEEE Computer Society.
4. Andriano, N. V. & Balzarini, M. (2006). *Comparación del Proceso de Elicitación de Requisitos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación*. Tesis presentada para la obtención del título de Magíster en Ingeniería de Software. Universidad Nacional de La Plata, Facultad de Informática, Argentina.
5. Basili, V. R. (1993). Applying the Goal/Question/Metric paradigm in the experience factory. *Software Quality Assurance and Measurement: A Worldwide Perspective*, 21-44.
6. Basili, V. R., & Selby, R. W. (1984). Data collection and analysis in software research and management. In *Proceedings of the American Statistical Association and Biomeasure Society*, 13-16.
7. Basili, V. R., & Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering* 10(6), 728-738.
8. Basili, V., Heidrich, J., Lindvall, M., Munch, J., Regardie, M., & Trendowicz, A. (2007). GQM⁺ Strategies--Aligning Business Strategies with Software Measurement. In *First International Symposium on Empirical Software Engineering and Measurement*, IEEE Computer Society Press, 488-490
9. Beechman, S., Badoo, N., Hall, T., Robinson, H. & Sharp, H. (2007). Motivation in Software Engineering: a Systematic Review. *Information and Software Technology Journal*, 50(9-10), 860-878.
10. Berenbach, B., Paulish, D. J., Kazmeier, J. & Rudorfer A. (2009). *Software & Systems Requirements Engineering: In Practice*. New York: McGraw-Hill Osborne Media.
11. Boehm, B., Brown, J. & Lipow, M. (1976). Quantitative evaluation of software quality. In *Proceedings of the 2nd international conference on Software engineering Transactions on Computers*, IEEE Computer Society Press, 592-605.

12. Boehm, B., Brown, J., Kaspar, H., Lipow, M., MacLeod, G. & Merrit, M.J. (1978). Characteristics of Software Quality. *TRW Series of Software Technology* (1). North-Holland Publishing Company.
13. Brooks, F. R. (1987). No silver bullet: Essence and Accidents of Software Engineering. *IEEE Computer Magazine*, 20(4), 10-19.
14. Caldiera, V. R. B. G., & Rombach, H. D. (1994). Goal Question Metric Paradigm. *Encyclopedia of Software Engineering*, 1, 528-532.
15. Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 49(9), 42-49.
16. Christel, M. G. & Kang, K.C. (1992). *Issues in Requirements Elicitation* (Technical Report CMU/SEI-92-TR-012 or ESC-TR-92-012) Software Engineering Institute, Carnegie Mellon University.
17. Cockburn A. (2001). *Agile Software Development*. Boston, US: Addison-Wesley Professional.
18. Coughlan, J. & Macredie, R.D. (2002) Effective Communication in Requirements Elicitation: A comparison of Methodologies. *Requirements Engineering*, 5(2), 47-60.
19. Creswell, J. W.(2008). *Educational research: Planning, conducting and evaluating quantitative and qualitative research*. New Jersey, Pearson Prentice Hall.
20. Davis, A. (1992). Operational Prototyping: A New Development Approach. *IEEE Software*, 9(5), 70-78.
21. Davis, A., Overmyer, S., Jordan, K., Dandashi, J. C., Dinh, A., Kincaid, G., Ledebor, G., Reynolds, P., Sitaram, P., Ta, A. y Theofanos M. (1993). Identifying and Measuring Quality in a Software Requirements Specification. In *Proceedings of 1st Software Metrics Symposium*, 141-152.
22. Davis, W. S. & Yen, D. C. (1999). Rapid application development (RAD). *The Information Consultant's System Handbook: System Analysis and Design*. Florida: CRC Press.
23. Delgadillo Correa, L. E. (2009). KIXAA VII, *Eficiencia en la creación de software para PyMES*. Tesis de Licenciatura. Instituto Politécnico Nacional, Mexico.
24. De Marco, T. (2009). Software engineering: An Idea Whose Time Has Come and Gone? *Software, IEEE*, 26(4), 95-96.
25. EcuRed: Enciclopedia cubana. Obtenido de: http://www.ecured.cu/index.php/Lord_Kelvin. Último acceso: Mayo 2014.
26. Fernandes, J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J. M., & Silva, M. M. D. (2012). iThink: A Game-Based Approach Towards Improving Collaboration and Participation in Requirement Elicitation. *Procedia Computer Science*, 15, 66-77.
27. Fleiss, J., 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5), 378-382.
28. Gibbs, W. W. (1994). Software's Chronic Crisis. *Scientific American*, 271(3), 72-81.
29. Goguen, J. A. & Linde, C. (1993). Techniques for Requirements Elicitation. In *Proceedings of IEEE International Symposium Requirements Engineering*, 152-164.

30. Hannay, J. E., Sjoberg, D. I., & Dyba, T. (2007). A systematic review of theory use in software engineering experiments. *IEEE Transactions on Software Engineering*, 33(2), 87-107.
31. Hickey, A. M. & Davis, A. M. (2002). Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS)*, IEEE Computer Society Press.
32. Hickey, A. M. & Davis, A. M. (2003). Elicitation Technique Selection: How Do Experts Do It? In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, 169-178.
33. Hofmann, H. F. & Lehner, F. (2001). Requirements Engineering as a Success Factor in Software Projects. *Software, IEEE*, 18(4), 58-66.
34. Holbrook III, H. (1990). A scenario-based methodology for conducting requirements elicitation. *ACM SIGSOFT Software Engineering Notes*, 15(1), 95-104.
35. Hull, E., Jackson, K. & Dick, J. (2011). *Requirements Engineering*. New York: Springer.
36. IEEE(1990). IEEE Standard Glossary of Software Engineering Terminology.
37. ISO/IEC 9126 (International Organization for Standardization/International Electrotechnical Commission) (1991). Tecnología de la información - Evaluación de Producto de Software - Características de calidad y directrices para su utilización.
38. Kaur, R. & Sengupta, J. (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, 2(2), 2-3.
39. Kitchenham, B.A. (2004). *Procedures for Performing Systematic Reviews* (Joint Technical Report Software Engineering Group). Department of Computer Science Keele University (UK) and Empirical Software Engineering, National ICT Australia, 33.
40. Kitchenham, B. A., Dyba, T. & Jorgensen, M. (2004) Evidence-based software engineering. In *Proceedings of 26th International Conference on Software Engineering (ICSE, 2004)*, 273-281.
41. Kitchenham, B.A. & Charters, S., (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. (EBSE Technical Report, EBSE-2007-01). Software Engineering Group, School of Computers Science and Mathematics, Keele University (UK) and Department of Computer Science, University of Durham (UK).
42. Kotonya, G. & Sommerville, I. (2000). *Requirements Engineering: Processes and techniques*. New York, US: Wiley.
43. Lauesen, S. (2002). *Software requirements, Styles and techniques*. Harlow, UK: Pearson Education Limited.
44. Lawrence, P. S., Pfleeger, S., & Atlee, J. M. (2010). *Software Engineering: theory and Practice*. Boston, MA: Pearson Education.
45. Lin, J., Fox, M. S., & Bilgic, T. (1996). A requirement ontology for engineering design. *Concurrent Engineering*, 4(3), 279-291.

46. Louridas, P. (2006). Using wikis in software development. *Software, IEEE*, 23(2), 88-91.
47. Loucopulos, P & Karakostas, V. (1995). *Systems Requirements Engineering*. New York, US: McGraw Hill.
48. McCall, J. A., Richards, P. K. & Walters, G. F. (1997). *Factors in software quality. I (Concepts and definitions of software quality)*. General Electric, California, US.
49. McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., & Hall, F. (2004). *Practical software measurement: objective information for decision makers*. Addison-Wesley Professional.
50. Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *Academy of management review*, 22(4), 853-886.
51. Nizam, N. M. H. & Sahibuddin S. (2011). Critical success factors for software projects: A comparative study. *Scientific Research and Essays*, 6(10), 2174-2186.
52. Nonyelum, O. F. (2012) Requirement elicitation problems in software development - A case study of a GSM service provider. *Indian J. Innovations Dev.*, 1(8), 599-604.
53. Nuseibeh, B. & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, ACM, 35-46.
54. Park, R. E. (1996). *CMM (sm) Version 1.1 Measurement Map* (No. CMU/SEI-96-SR-003). Carnegie-Mellon University, Software Engineering Institute.
55. Pfleeger, S. L. (2005). Soup or art? The role of evidential force in empirical software engineering. *Software, IEEE*, 22(1), 66-73.
56. Pressman, R. S. (2009). *Software Engineering, A practical approach*. New York, US: Mc Graw Hill.
57. Princeton University (2013). Word Net (version). Obtenido de: <http://wordnetprinceton.edu>. Último acceso: Abril 2013.
58. RAE (2013). Obtenido de: <http://www.rae.es/rae.html>. Último acceso: Mayo 2013.
59. Raghavan, S., Zelesnik, G. & Ford, G. (1994). *Lectures Notes in Requirements Elicitation* (CMU/SEI-94-EM-10). Software Engineering Institute, Carnegie Mellon University.
60. Reubenstein, H. & Waters, R. (1991). The Requirements Apprentice: Automated Assistance for Requirements Acquisition. *IEEE Transactions on Software Engineering*, 17(3), 226-240.
61. Robertson, S. & Robertson, J. (2006). *Mastering the Requirement Process*. New York, US: Addison-Wesley.
62. Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2), 131-164.
63. Scott, A. C., Clayton, J. E. & Gibson, E. L. (1991). *A practical guide to knowledge acquisition*. Boston: Addison-Wesley Longman Publishing Company, Inc.
64. SE (2004). Estudio del nivel de madurez y capacidad de procesos de la industria de tecnologías de información en el área metropolitana de Monterrey, Nuevo León y el Distrito Federal y su área metropolitana. Secretaría de Economía del Gobierno Mexicano.

65. Sindre, G., & Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1), 34-44.
66. SG (2013). Estudios de salarios SG 2013. *Software Gurú*, 42, 18-24.
67. Sofia Pereira, C., & Soares, A. L. (2007). Improving the quality of collaboration requirements for information management through social networks analysis. *International Journal of Information Management*, 27(2), 86-103.
68. Sommerville, I. & Sawyer, P. (2006). *Requirements Engineering*. New York, US: John Wiley & Sons.
69. Sommerville I. (2010). *Software Engineering*. New York, US: Addison-Wesley
70. Standish (2010). CHAOS Summary 2010 Report. www.standishgroup.com. Último acceso: Abril 2013.
71. Standish (2011). CHAOS Report 2011. www.standishgroup.com. Último acceso: Abril 2013.
72. Van Solingen, R., & Berghout, E. (1999). *The Goal/Question/Metric Method: a practical guide for quality improvement of software development* (40). London: McGraw-Hill.
73. Walia, G. S. & Carver, J. C. (2009). A systematic literature review to identify and classify software requirement errors. *Information and Software Technology*, 51(7), 1087-1109.
74. Wohlin, C., Höst, M., & Henningsson, K. (2003). Empirical research methods in software engineering. In *Empirical methods and studies in software engineering* (7-23). Springer Berlin Heidelberg.
75. Woodcock, J., Larsen, P. G., Bicarregui, J., & Fitzgerald, J. (2009). Formal methods: Practice and experience. *ACM Computing Surveys (CSUR)*, 41(4), 19.
76. Young, M., McDonald, S., Edwards, H.M. & Thompson, J. B. (2001). Quality & People in the Development of Situationally Specific Methods. In *Proceedings IEEE. Second Asia-Pacific Conference on Quality Software*, 199-203.
77. Zheyang, Z. (2007). Effective Requirements Development - A Comparison of Requirements Elicitation techniques. In *SQM2007 University of Tampere, Finland, INSPIRE* 225-240.
78. Zowghi, D. & Paryani, S. (2003). Teaching Requirements Engineering Through Role Playing. In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, 8(12), 233-241.
79. Zowghi, D. & Coulin C. (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements*, 19-46. Springer Berlin Heidelberg.

Lecturas adicionales

1. Al Balushi, T. H., Sampaio, P. R. F., Dabhi, D., & Loucopoulos, P. (2007). ElicitO: a quality ontology-guided NFR elicitation tool. In *Requirements Engineering: Foundation for Software Quality* (306-319). Springer Berlin Heidelberg.
2. Al Mrayat, O. I., Norwawi, N. M., & Basir, N. (2013). Requirements Elicitation Techniques: Comparative Study. *Optimization*, 27(32), 6.

3. Ali, N., Wu, W., Antoniol, G., Di Penta, M., Guéhéneuc, Y. G., & Hayes, J. H. (2011). Moms: Multi-objective miniaturization of software. In *Software Maintenance (ICSM), 2011 27th IEEE International Conference on* (153-162). IEEE.
4. Al-Salem, L. S., & Abu Samaha, A. (2007). Eliciting Web application requirements—an industrial case study. *Journal of Systems and Software, 80*(3), 294-313.
5. An, Y., Dalrymple, P. W., Rogers, M., Gerrity, P., Horkoff, J., & Yu, E. (2009). Collaborative social modeling for designing a patient wellness tracking system in a nurse-managed health care center. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (2). ACM.
6. Anwar, F., & Razali, R. (2012). A Practical Guide to Requirements Elicitation Techniques Selection-An Empirical Study. *Middle-East Journal of Scientific Research, 11*(8), 1059-1067.
7. Aoyama, K., Ugai, T., Yamada, S., & Obata, A. (2007). Extraction of viewpoints for eliciting customer's requirements based on analysis of specification change records. In *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific* (33-40). IEEE.
8. Azadegan, A., Cheng, X., Niederman, F., & Yin, G. (2013). Collaborative Requirements Elicitation in Facilitated Collaboration: Report from a Case Study. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on* (569-578). IEEE.
9. Azadegan, A., Papamichail, N., & Sampaio, P. (2013). Applying collaborative process design to user requirements elicitation: A case study. *Computers in Industry, 64*(7), 798-812.
10. Barragáns Martínez, A. B., Pazos Arias, J. J., Fernández Vilas, A., García Duque, J., López Nores, M., Díaz Redondo, R. P., & Blanco Fernández, Y. (2008). On the interplay between inconsistency and incompleteness in multi-perspective requirements specifications. *Information and Software Technology, 50*(4), 296-321.
11. Bee, C. B., Danilo, B., & June, V. (2010). Understanding the use of Elicitation Approaches for Effective Requirements Gathering. In *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on* (325-330). IEEE.
12. Bellotti, F., Berta, R., De Gloria, A., Lavagnino, E., Dagnino, F., Ott, M., & Mayer, I. S. (2012). Designing a Course for Stimulating Entrepreneurship in Higher Education through Serious Games. *Procedia Computer Science, 15*, 174-186.
13. Bendjenna, H., Zarour, N. E., & Charrel, P. J. (2008). MAMIE: A Methodology to Elicit Requirements in Inter-company Co-operative Information Systems. In *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on* (290-295). IEEE.
14. Berenbach, B., & Borotto, G. (2006). Metrics for model driven requirements development. In *Proceedings of the 28th international conference on Software engineering* (445-451). ACM.
15. Bokhari, M. U., & Siddiqui, S. T. (2011). Metrics for Requirement Engineering and Automated Requirement Tools. In *Proceedings of the 5th National Conference INDIACOM-2011*.
16. Breivik, E., & Supphellen, M. (2003). Elicitation of product attributes in an evaluation context: A comparison of three elicitation techniques. *Journal of Economic Psychology, 24*(1), 77-98.

17. Brill, O., & Knauss, E. (2011). Structured and unobtrusive observation of anonymous users and their context for requirements elicitation. In *Requirements Engineering Conference (RE), 2011 19th IEEE International* (175-184). IEEE.
18. Browne, G. J., & Rogich, M. B. (2001). An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. *Journal of Management Information Systems*, 17(4), 223-250.
19. Calefato, F., Damian, D., & Lanubile, F. (2012). Computer-mediated communication to support distributed requirements elicitation and negotiations tasks. *Empirical Software Engineering*, 17(6), 640-674.
20. Carrizo, D., Dieste, O., & López, M. (2011). Identifying moderator variables through requirements elicitation experiments limitations. In *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement* (22-25). ACM.
21. Castro, A., & Oliveira, E. (2008). The rationale behind the development of an airline operations control centre using Gaia-based methodology. *International Journal of Agent-Oriented Software Engineering*, 2(3), 350-377.
22. Chand, M.G., Reddy, K.N., Rao, A.A. & Kumar, J.K.(2010). An approach to requirements elicitation and analysis using goal. In *Software Technology and Engineering (ICSTE), 2010 2nd International Conference on* (218-221). IEEE.
23. Cheng, J., & Liu, Q. (2008). Using Stakeholder Analysis for Improving Statechart Merging in Software Requirement Management. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for* (1217-1222). IEEE.
24. Chowdhury, A. F., & Huda, M. N. (2011). Comparison between adaptive software development and feature driven development. In *Computer Science and Network Technology (ICCSNT), 2011 International Conference on* (363-367). IEEE.
25. Clements, P., & Bass, L. (2010). Using Business Goals to Inform a Software Architecture. In *Requirements Engineering Conference (RE), 2010 18th IEEE International* (69-78). IEEE.
26. Corbridge, C., Rugg, G., Major, N. P., Shadbolt, N. R., & Burton, A. M. (1994). Laddering: technique and tool use in knowledge acquisition. *Knowledge Acquisition*, 6(3), 315-341
27. Costello, R. J., & Liu, D. B. (1995). Metrics for requirements engineering. *Journal of Systems and Software*, 29(1), 39-63.
28. Dai, C. F., & Wang, M. L. (2009). Distributed requirement elicitation and negotiation based on the hall for workshop of meta-synthetic engineering. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (1452-1456). ACM.
29. Damian, D., Lanubile, F., & Mallardo, T. (2006). The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations. In *Proceedings of the 28th international conference on Software engineering* (917-920). ACM.
30. Daud, N. M. N., & Bakar, N. A. A. A. (2010). Experimenting on ethnography in requirement elicitation from beginner perspective. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on* (64-66). IEEE.

31. Davis, A. M. (1999). Achieving Quality in Software Requirements. *Software Quality Professional*, 1(3), 37-44.
32. Davis, A., Dieste, O., Hickey, A., Juristo, N., & Moreno, A. M. (2006). Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In *Requirements Engineering, 14th IEEE International Conference* (179-188). IEEE.
33. Decker, B., Ras, E., Rech, J., Jaubert, P., & Rieth, M. (2007). Wiki-based stakeholder participation in requirements engineering. *Software, IEEE*, 24(2), 28-35.
34. De La Flor, G., Jirotko, M., Luff, P., Pybus, J., & Kirkham, R. (2010). Transforming scholarly practice: Embedding technological interventions to support the collaborative analysis of ancient texts. *Computer Supported Cooperative Work (CSCW)*, 19(3-4), 309-334.
35. De la Vara, J. L., & Sánchez, J. (2009). BPMN-based specification of task descriptions: approach and lessons learnt. In *Requirements Engineering: Foundation for Software Quality* (124-138). Springer Berlin Heidelberg.
36. Derrick, D. C., Read, A., Nguyen, C., Callens, A., & De Vreede, G. J. (2013). Automated Group Facilitation for Gathering Wide Audience End-User Requirements. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on* (195-204). IEEE.
37. Diaper, D. (1989). *Knowledge elicitation: principle, techniques and applications*. Springer-Verlag New York, Inc..
38. Dogan, H., Henshaw, M., & Urwin, E. (2009). A 'Soft' Approach to TLM Requirements Capture to Support Through-Life Management. In *Knowledge Science, Engineering and Management* (458-469). Springer Berlin Heidelberg.
39. Doherty, G., McKnight, J., & Luz, S. (2010). Fieldwork for requirements: Frameworks for mobile healthcare applications. *International journal of human-computer studies*, 68(10), 760-776.
40. Driss, M., Jamoussi, Y., Jézéquel, J. M., & Ghézala, H. H. B. (2011). A multi-perspective approach for web service composition. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services* (106-111). ACM.
41. Duggan, E. W., & Thachenkary, C. S. (2004). Integrating nominal group technique and joint application development for improved systems requirements determination. *Information & Management*, 41(4), 399-411.
42. Dutoit, A. H., & Bruegge, B. (1998). Communication metrics for software development. *Software Engineering, IEEE Transactions on*, 24(8), 615-628.
43. Erfurth, I., & Kirchner, K. (2010). Requirements Elicitation with Adapted CUTA Cards: First Experiences with Business Process Analysis. In *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on* (215-223). IEEE.
44. Estrella, F., Hauer, T., McClatchey, R., Odeh, M., Rogulin, D., & Solomonides, T. (2007). Experiences of engineering Grid-based medical software. *International journal of medical informatics*, 76(8), 621-632.
45. Fowlkes, J. E., Salas, E., Baker, D. P., Cannon-Bowers, J. A., & Stout, R. J. (2000). The utility of event-based knowledge elicitation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(1), 24-35.

46. Girase, S. (2012). Comparison of various Elicitation Techniques and Requirement Prioritisation Techniques. *International Journal of Engineering*, 1(3).
47. Goguen, J. A., & Linde, C. (1993). Techniques for requirements elicitation. *Requirements Engineering*, 93, 152-164.
48. Gonzales, C. K., & Leroy, G. (2011). Eliciting user requirements using Appreciative inquiry. *Empirical Software Engineering*, 16(6), 733-772.
49. Guzzi, A., Hattori, L., Lanza, M., Pinzger, M., & van Deursen, A. (2011). Collective code bookmarks for program comprehension. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on* (101-110). IEEE.
50. Györkös, J. (1994). Measurements in software requirements specification process. *Microprocessing and microprogramming*, 40(10), 893-896.
51. Habibizad Navin, A., Kiyani, F., Kheyri, J., & Tahmasebi Rad, H. (2009). Transforming Business Tasks to Data Flow Models for Decrease Process in Analysis Systems. In *Proceedings of the 2009 International Conference on Advanced Computer Control* (706-710). IEEE Computer Society.
52. Hainey, T., Connolly, T. M., Stansfield, M., & Boyle, E. A. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education*, 56(1), 21-35.
53. Han, D., Zhou, K., Ma, Y., & Fang, C. (2012). A Context-Based Requirement Acquisition and Description Method. In *Internet Computing for Science and Engineering (ICICSE), 2012 Sixth International Conference on* (112-116). IEEE.
54. Hanakawa, N., & Obana, M. (2012). A metrics for meeting quality on a software requirement acquisition phase. In *Product-Focused Software Process Improvement* (260-274). Springer Berlin Heidelberg.
55. Hickey, A. M., & Davis, A. M. (2003). Elicitation technique selection: how do experts do it?. In *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International* (169-178). IEEE.
56. Hickey, A. M., & Davis, A. M. (2007). An Ontological Approach to Requirements Elicitation Technique Selection. In *Ontologies* (403-431). Springer US.
57. Hilaire, V., Cossentino, M., Gechter, F., Rodriguez, S., & Koukam, A. (2012). An approach for the integration of swarm intelligence in MAS: An engineering perspective. *Expert Systems with Applications*, 40(4), 1323-1332.
58. Hofmann, H. F. (2000). Requirements Engineering Activities and Methods. In *Requirements Engineering* (27-88). Deutscher Universitätsverlag.
59. Hussein, M., Yu, J., Han, J., & Colman, A. (2012). Scenario-Driven development of context-aware adaptive web services. In *Web Information Systems Engineering-WISE 2012* (228-242). Springer Berlin Heidelberg.
60. Isabirye, N., & Flowerday, S. (2008). A model for eliciting user requirements specific to South African rural areas. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology* (124-130). ACM.

61. In, H., & Roy, S. (2001). Visualization issues for software requirements negotiation. In *Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International* (10-15). IEEE.
62. Janette, M. W., Will, R. P., & Blanton, J. E. (1996). Enhancing knowledge elicitation using the cognitive interview. *Expert Systems with Applications*, 10(1), 127-133.
63. Jiang, L. (2005). *A framework for the requirements engineering process development*. Tesis Doctoral, University of Calgary, Canadá.
64. Jiang, H., & Yang, X. (2009). Performance requirement elicitation for financial information system based on ontology. In *TENCON 2009-2009 IEEE Region 10 Conference* (1-5). IEEE.
65. Jiang, L., Eberlein, A., Far, B. H., & Mousavi, M. (2008). A methodology for the selection of requirements engineering techniques. *Software & Systems Modeling*, 7(3), 303-328.
66. Jones, S., Lynch, P., Maiden, N., & Lindstaedt, S. (2008). Use and influence of creative ideas and requirements for a work-integrated learning system. In *International Requirements Engineering, 2008. RE'08. 16th IEEE* (289-294). IEEE.
67. Kaiya, H., & Saeki, M. (2006). Using domain ontology as domain knowledge for requirements elicitation. In *Requirements Engineering, 14th IEEE International Conference* (189-198). IEEE.
68. Kaiya, H., Shimizu, Y., Yasui, H., Kaijiri, K., & Saeki, M. (2010). Enhancing domain knowledge for requirements elicitation with web mining. In *Software Engineering Conference (APSEC), 2010 17th Asia Pacific* (3-12). IEEE.
69. Kamalrudin, M., & Grundy, J. (2011). Generating essential user interface prototypes to validate requirements. In *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering* (564-567). IEEE Computer Society.
70. Kassou, M., & Kjiri, L. (2012). A maturity metric based approach for eliciting SOA security requirements. In *Network Security and Systems (JNS2), 2012 National Days of* (7-11). IEEE.
71. Kausar, S., Tariq, S., Riaz, S., & Khanum, A. (2010, October). Guidelines for the selection of Elicitation Techniques. In *Emerging Technologies (ICET), 2010 6th International Conference on* (265-269). IEEE.
72. Kim, B. H., Park, S. B., Lee, G. B., & Chung, S. Y. (2007). Framework of integrated system for the innovation of mold manufacturing through process integration and collaboration. In *Computational Science and Its Applications-ICCSA 2007* (1-10). Springer Berlin Heidelberg.
73. Kukreja, N. (2012). Winbook: a social networking based framework for collaborative requirements elicitation and WinWin negotiations. In *Proceedings of the 2012 International Conference on Software Engineering* (1610-1612). IEEE Press.
74. Kukreja, N. (2013). Decision theoretic requirements prioritization: a two-step approach for sliding towards value realization. In *Proceedings of the 2013 International Conference on Software Engineering* (1465-1467). IEEE Press.
75. Laporti, V., Borges, M. R., & Braganholo, V. (2009). Athena: A collaborative approach to requirements elicitation. *Computers in Industry*, 60(6), 367-380.

76. Lauesen, S., & Vinter, O. (2001). Preventing requirement defects: An experiment in process improvement. *Requirements Engineering*, 6(1), 37-50.
77. Laurent, P., & Cleland-Huang, J. (2009). Lessons learned from open source projects for facilitating online requirements processes. In *Requirements Engineering: Foundation for Software Quality* (240-255). Springer Berlin Heidelberg.
78. Laurent, P., & Cleland-Huang, J. (2009). Requirements-gathering collaborative networks in distributed software projects. In *Requirements: Communication, Understanding and Softskills, 2009 Collaboration and Intercultural Issues on* (26-30). IEEE.
79. Lázaro, J. P., Fides, A., Navarro, A., & Guillén, S. (2010). Ambient Assisted Nutritional Advisor for elderly people living at home. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (198-203). IEEE.
80. Li, G., Jin, Z., Xu, Y., & Lu, Y. (2011). An engineerable ontology based approach for requirements elicitation in process centered problem domain. In *Knowledge Science, Engineering and Management* (208-220). Springer Berlin Heidelberg.
81. Lim, S. L., Damian, D., Ishikawa, F., & Finkelstein, A. (2013). Using Web 2.0 for stakeholder analysis: StakeSource and its application in ten industrial projects. In *Managing requirements knowledge* (221-242). Springer Berlin Heidelberg.
82. Lim, S. L., Damian, D., & Finkelstein, A. (2011). StakeSource2. 0: using social networks of stakeholders to identify and prioritise requirements. In *Software Engineering (ICSE), 2011 33rd International Conference on* (1022-1024). IEEE.
83. Lim, S. L., & Finkelstein, A. (2012). StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation. *Software Engineering, IEEE Transactions on*, 38(3), 707-735.
84. Li-ping, D., & Long-jun, H. (2010). Requirements acquirement based on two-world simulation method. In *Computer Science and Education (ICCSE), 2010 5th International Conference on* (1865-1867). IEEE.
85. Liu, L., & Jin, Z. (2007). Requirements Analyses Integrating Goals and Problem Analysis Techniques. *Tsinghua Science & Technology*, 12(6), 729-740.
86. Lloyd, W. J., Rosson, M. B., & Arthur, J. D. (2002). Effectiveness of elicitation techniques in distributed requirements engineering. In *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on* (311-318). IEEE.
87. Maguire, M. C. (2013). An analysis of specialist and non-specialist user requirements for geographic climate change information. *Applied ergonomics*, 44(6), 874-885.
88. Mahmud, I., & Veneziano, V. (2011). Mind-mapping: An effective technique to facilitate requirements engineering in agile software development. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on* (157-162). IEEE.
89. Maiden, N. A. M., & Rugg, G. (1996). ACRE: selecting methods for requirements acquisition. *Software Engineering Journal*, 11(3), 183-192.
90. Maiden, N. (2009). Card sorts to acquire requirements. *Software, IEEE*, 26(3), 85-86.

91. Martin, J. L., Clark, D. J., Morgan, S. P., Crowe, J. A., & Murphy, E. (2012). A user-centred approach to requirements elicitation in medical device development: A case study from an industry perspective. *Applied ergonomics*, 43(1), 184-190.
92. Maté, A., Trujillo, J., & Franch, X. (2011). A modularization proposal for goal-oriented analysis of data warehouses using I-star. In *Conceptual Modeling-ER 2011* (421-428). Springer Berlin Heidelberg.
93. Mavin, A., Novak, M., Wilkinson, P., Maiden, N., & Lynch, P. (2008). Using Scenarios to Discover Requirements for Engine Control Systems. In *International Requirements Engineering, 2008. RE'08. 16th IEEE* (235-240). IEEE.
94. Mendizabal, O. M., Spier, M., & Saad, R. (2012). Log-based approach for performance requirements elicitation and prioritization. In *Requirements Engineering Conference (RE), 2012 20th IEEE International* (297-302). IEEE.
95. McFarlane, D., & Cuthbert, R. (2012). Modelling information requirements in complex engineering services. *Computers in Industry*, 63(4), 349-360.
96. McKay, K. N., & Black, G. W. (2007). The evolution of a production planning system: A 10-year case study. *Computers in Industry*, 58(8), 756-771.
97. Mishra, D., Mishra, A., & Yazici, A. (2008). Successful requirement elicitation by combining requirement engineering techniques. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the* (258-263). IEEE.
98. Monsalve, C., April, A., & Abran, A. (2012). On the expressiveness of business process modeling notations for software requirements elicitation. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society* (3132-3137). IEEE.
99. Moody, J. W., Blanton, J. E., & Cheney, P. H. (1998). A theoretically grounded approach to assist memory recall during information requirements determination. *Journal of Management Information Systems*, 15(1), 79-98.
100. Moore, J. M., & Shipman, F. M. (2000). A comparison of questionnaire-based and GUI-based requirements gathering. In *Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on* (35-43). IEEE.
101. Nakatani, T., Tsumaki, T., Tsuda, M., Inoki, M., Hori, S., & Katamine, K. (2011). Requirements maturation analysis by accessibility and stability. In *Software Engineering Conference (APSEC), 2011 18th Asia Pacific* (357-364). IEEE.
102. Niu, N., & Easterbrook, S. (2007). So, You Think You Know Others' Goals? A Repertory Grid Study. *Software, IEEE*, 24(2), 53-61.
103. Nurmuliani, N., Zowghi, D., & Powell, S. (2004). Analysis of requirements volatility during software development life cycle. In *Software Engineering Conference, 2004. Proceedings. 2004 Australian* (28-37). IEEE.
104. Ocker, R., Fjermestad, J., Hiltz, S. R., & Johnson, K. (1998). Effects of four modes of group communication on the outcomes of software requirements determination. *Journal of Management Information Systems*, 15(1), 99-118.
105. Opdahl, A. L., & Sindre, G. (2009). Experimental comparison of attack trees and misuse cases for security threat identification. *Information and Software Technology*, 51(5), 916-932.

106. Palmer, J. D., & Evans, R. B. (1994, October). Software risk management: requirements-based risk metrics. In *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on* (1, 836-841). IEEE.
107. Pitts, M. G., & Browne, G. J. (2007). Improving requirements elicitation: an empirical investigation of procedural prompts. *Information Systems Journal*, 17(1), 89-110
108. Pitula, K., & Radhakrishnan, T. (2011). On eliciting requirements from end-users in the ICT4D domain. *Requirements Engineering*, 16(4), 323-351.
109. Pouyioutas, P., Dionysiou, I., & Gjermundrod, H (2013). ReProTool Version 3.0–The Faculty Module for Designing and Enhancing University Programmes to Comply with the ECTS Label. In *Global Engineering Education Conference (EDUCON), 2013 IEEE* (16-22). IEEE.
110. Ramdoyal, R., Cleve, A., & Hainaut, J. L. (2010). Reverse engineering user interfaces for interactive database conceptual analysis. In *Advanced Information Systems Engineering* (332-347). Springer Berlin Heidelberg.
111. Read, A., & Briggs, R. O. (2012). The Many Lives of an Agile Story: Design Processes, Design Products, and Understandings in a Large-Scale Agile Development Project. In *System Science (HICSS), 2012 45th Hawaii International Conference on* (5319-5328). IEEE.
112. Reimer, Y. J., Brimhall, E., Cao, C., & O'Reilly, K. (2009). Empirical user studies inform the design of an e-notetaking and information assimilation system for students in higher education. *Computers & Education*, 52(4), 893-913.
113. Rosado, D. G., Fernández-Medina, E., López, J., & Piattini, M. (2010). Analysis of secure mobile grid systems: a systematic approach. *Information and Software Technology*, 52(5), 517-536.
114. Rui-feng, S., Chao, Y., Rui-fang, Z., & Jie, X. (2010). Acquire multi-viewpoint from domain. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on* (598-602). IEEE.
115. Rusu, A., Russell, R., & Cocco, R. (2011). Simulating the software engineering interview process using a decision-based serious computer game. In *Computer Games (CGAMES), 2011 16th International Conference on* (235-239). IEEE.
116. Sabahat, N., Iqbal, F., Azam, F., & Javed, M. Y. (2010). An iterative approach for global requirements elicitation: A case study analysis. In *Electronics and Information Engineering (ICEIE), 2010 International Conference On* (361-366). IEEE.
117. Sajid, A., Nayyar, A., & Mohsin, A. (2010). Modern trends towards requirement elicitation. In *Proceedings of the 2010 National Software Engineering Conference* (9). ACM.
118. Sakhnini, V., Mich, L., & Berry, D. M. (2012). The effectiveness of an optimized EPMcreate as a creativity enhancement technique for Web site requirements elicitation. *Requirements Engineering*, 17(3), 171-186.
119. Salvador, V. F. M., de Oliveira Neto, J. S., & Kawamoto, A. S. (2008). Requirement engineering contributions to voice user interface. In *Advances in Computer-Human Interaction, 2008 First International Conference on* (309-314). IEEE.
120. Schneider, K. (2007). Generating fast feedback in requirements elicitation. In *Requirements Engineering: Foundation for Software Quality* (160-174). Springer Berlin Heidelberg.

121. Sen, A. M., & Jain, S. K. (2007). An Agile Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements engineering. In *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on* (41-47). IEEE.
122. Serna, M. E. (2012, October). Analysis and selection to requirements elicitation techniques. In *Computing Congress (CCC), 2012 7th Colombian* (1-7). IEEE.
123. Seyff, N., Graf, F., Grünbacher, P., & Maiden, N. (2008). Mobile Discovery of Requirements for Context-Aware Systems. In *Requirements Engineering: Foundation for Software Quality* (183-197). Springer Berlin Heidelberg.
124. Seyff, N., Graf, F., Maiden, N., & Grünbacher, P. (2009a). Scenarios in the wild: Experiences with a contextual requirements discovery method. In *Requirements Engineering: Foundation for Software Quality* (147-161). Springer Berlin Heidelberg.
125. Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009b). Exploring how to use scenarios to discover requirements. *Requirements Engineering*, 14(2), 91-111.
126. Siau, K., & Wang, Y. (2007). Cognitive evaluation of information modeling methods. *Information and Software Technology*, 49(5), 455-474.
127. Sindre, G. (2007). Mal-activity diagrams for capturing attacks on business processes. In *Requirements Engineering: Foundation for Software Quality* (355-366). Springer Berlin Heidelberg.
128. Sommerville, I., Lock, R., & Storer, T. (2012). Information requirements for enterprise systems. In *Large-Scale Complex IT Systems. Development, Operation and Management* (266-282). Springer Berlin Heidelberg.
129. Sulfaro, M., Marchesi, M., & Pinna, S. (2007). Agile practices in a large organization: the experience of poste Italiane. In *Agile Processes in Software Engineering and Extreme Programming* (219-221). Springer Berlin Heidelberg.
130. Sutcliffe, A. (1995). Requirements rationales: integrating approaches to requirement analysis. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques* (33-42). ACM.
131. Svensson, R. B., Host, M., & Regnell, B. (2010) quality requirements: A systematic review. In *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on* (261-268). IEEE.
132. Taa, A., Abdullah, M. S., & Norwawi, N. M. (2010). RAMEPs: a goal-ontology approach to analyse the requirements for data warehouse systems. *WSEAS Transactions on Information Science and Applications*, 7(2), 295-309.
133. Thew, S., Sutcliffe, A., Procter, R., de Bruijn, O., McNaught, J., Venters, C. C., & Buchan, I. (2009). *Requirements engineering for e-science: experiences in epidemiology. Software, IEEE*, 26(1), 80-87.
134. Tiwari, S., Rathore, S. S., & Gupta, A. (2012). Selecting requirement elicitation techniques for software projects. In *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on* (1-10). IEEE.
135. Trujillo, J., Soler, E., Fernández-Medina, E., & Piattini, M. (2009). An engineering process for developing Secure Data Warehouses. *Information and Software Technology*, 51(6), 1033-1051.

136. Tsumaki, T., & Tamai, T. (2006). Framework for matching requirements elicitation techniques to project characteristics. *Software Process: Improvement and Practice*, 11(5), 505-519.
137. Tu, Y. C., & Thomborson, C. (2009). Preliminary security specification for New Zealand's igovt system. In *Proceedings of the Seventh Australasian Conference on Information Security*, 98(79-88). Australian Computer Society, Inc.
138. Urbieto, M., Retschitzegger, W., Rossi, G., Schwinger, W., Gordillo, S., & Luna, E. R. (2012). Modelling adaptations requirements in Web workflows. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services* (72-81). ACM.
139. Van Velsen, L., van der Geest, T., ter Hedde, M., & Derks, W. (2008). Engineering user requirements for e-government services: a dutch case study. In *Electronic Government* (243-254). Springer Berlin Heidelberg.
140. Van Velsen, L., van der Geest, T., ter Hedde, M., & Derks, W. (2009). Requirements engineering for e-Government services: A citizen-centric approach and case study. *Government Information Quarterly*, 26(3), 477-486.
141. Vitharana, P., Jain, H., & Zahedi, F. (2012). A knowledge based component/service repository to enhance analysts' domain knowledge for requirements analysis. *Information & Management*, 49(1), 24-35.
142. Wang, T., Si, Y., Xuan, X., Wang, X., Yang, X., Li, S., & Kavs, A. J. (2010). A QoS ontology cooperated with feature models for non-functional requirements elicitation. In *Proceedings of the Second Asia-Pacific Symposium on Internetware* (17). ACM.
143. Widya, I., Bults, R., van Beijnum, B. J., Sandsjö, L., Schaake, L., Jones, V., & Hermens, H. (2009a). Requirements elicitation in a telemedicine pain-treatment trial. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International* (309-314). IEEE.
144. Widya, I., Van Beijnum, B. J. F., Bults, R., Jones, V., Hermens, H., Sandsjö, L., & Schaake, L. (2009b). Early phase requirements assessment of a teletreatment trial. In *Proceedings of the 2009 ACM symposium on Applied Computing* (395-396). ACM.
145. Yamanaka, T., Noguchi, H., Yato, S., & Komiya, S. (2010). A proposal of a method to navigate interview-driven software requirements elicitation work. *WSEAS Transactions on Information Science and Applications*, 7(6), 784-798.
146. Young, R. R. (2002). Recommended requirements gathering practices. *CrossTalk*, 15(4), 9-12.
147. Zapata, S., Torres, E., Collazos, C. A., Giraldo, F. D., & Sevilla, G. (2013). Distributed elicitation of software requirements: An experimental case from Argentina and Colombia. In *Computing Colombian Conference (8CCC), 2013 8th* (1-7). IEEE.
148. Zapata, S., Torres, E., Sevilla, G., Aballay, L., & Reus, M. (2012). Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios. In *2012 XXXVIII Conferencia Latinoamericana en Informatica (CLEI)* (1-7). IEEE.
149. Zhang, T., & Lee, B. (2010). Complementary Classification Techniques based Personalized Software Requirements Retrieval with Semantic Ontology and User Feedback. In *2010 IEEE 10th International Conference on Computer and Information Technology (CIT)*, (1358-1363). IEEE.

150. Zhao, J., Kan, M. Y., & Theng, Y. L. (2008). Math information retrieval: user requirements and prototype implementation. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries* (187-196). ACM.
151. Zin, A. M., & Pa, N. (2009). Measuring communication gap in software requirements elicitation process. In *Proceedings of the 8th WSEAS International Conference on Software engineering, parallel and distributed systems* (66-71). World Scientific and Engineering Academy and Society (WSEAS).
152. Zmud, R. W., Anthony, W. P., & Stair Jr, R. M. (1993). The use of mental imagery to facilitate information identification in requirements analysis. *Journal of Management Information Systems*, 9(4), 175-191.
153. Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements* (19-46). Springer Berlin Heidelberg.