



“Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación.”

Tesis presentada a la Facultad de Informática de la Universidad Nacional de La Plata como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software

Autor: Ing. Natalia Valeria Andriano

Directora: Dr. Mónica Balzarini

Co-Director: Dr. Gustavo Rossi

Agosto 2006

UNLP
Facultad de Informática
Universidad Nacional de La Plata
La Plata, Argentina

UBP
Universidad Blas Pascal
Córdoba, Argentina



Maestría en Ingeniería de Software
Universidad Blas Pascal – Universidad Nacional de la Plata



Resumen

El presente trabajo de tesis tiene como objetivo la comparación de las técnicas de elicitación de requerimientos aplicables a la ingeniería de software ya sean éstas tanto para el desarrollo de software a medida como empaquetado. El análisis se realizó en base a dos encuestas, una destinada a desarrolladores de software y otra a usuarios considerando una serie de buenas prácticas recomendadas para el proceso de elicitación de requerimientos. Los resultados permitieron comprobar la hipótesis que establece que la alternativa de software a desarrollar (a medida o empaquetado) condiciona el proceso de elicitación. Además se realizaron entrevistas a desarrolladores diseñadas exclusivamente para probar la validez de una serie de métricas propuestas para la evaluación y monitoreo de la elicitación. Las métricas propuestas y seleccionadas son específicas y todas pretenden contribuir a la mejora de uno de los procesos más crítico para el éxito del desarrollo del software, i.e. la elicitación de requerimientos. El trabajo culmina con el desarrollo de una guía para la realización del proceso de elicitación de requerimientos tanto para el desarrollo de software a medida como empaquetado.

Palabras clave: ingeniería de requerimientos, desarrollo de requerimientos, técnica de elicitación, métricas de software, , entrevistas, encuestas, tormenta de ideas, Win Win, RAD, JAD, prototipación, observación.

Abstract

The objective of this thesis work is the comparison of known requirements elicitation techniques, for both the development of sized-software and packaged software. The analysis was made on the basis of two surveys, one destined to software developers and the other destined to users. It was also considered a series of good recommended practices for the requirements elicitation process. The results allowed the verification of the stated hypothesis that establishes that the type of software to be developed (sized or packaged), conditions the elicitation process. In addition interviews were designed dedicated exclusively to developers to prove the validity of a series of proposed metrics for this process. The selected metrics try to contribute to the control and monitoring for the improvement of such a critical process as it is the elicitation of requirements. To finalize the work a guide was developed for the accomplishment of the requirements elicitation process, regardless the type of software to be developed (sized as packaged).

Keywords: *requirements development, elicitation techniques, software metrics, requirements engineering, interviews, surveys, brainstorming, Win Win, RAD, JAD, prototyping, observation.*

Índice de contenido

Resumen	3
Abstract	4
Propósito de la tesis	10
Organización de la tesis	10
Agradecimientos	11
Capítulo 1: Introducción	12
Objetivos	16
Hipótesis	17
Capítulo 2: Marco teórico	18
Alternativas de productos de software	18
Requerimientos e ingeniería de requerimientos	22
Dificultades en el proceso de elicitación de requerimientos	28
Técnicas de elicitación de requerimientos	29
Métricas de software	43
Capítulo 3: Materiales y métodos	47
Técnicas de elicitación	47
Operalización de las variables	49
Métodos de análisis	49
Métricas	50
Capítulo 4: Resultados	51
Parte I: Estadística descriptiva para la totalidad de encuestados (sin considerar contexto definido por la alternativa de software)	51
Parte II: Estudio comparativo del proceso de elicitación de requerimientos entre tipos de software	70
Parte III: Uso de métricas	79
Capítulo 5: Propuestas de métricas para el proceso de elicitación de requerimientos	82
Capítulo 6: Discusión y conclusiones	94
Métricas directas	98
Métricas indirectas	99
Trabajos Futuros	100
Anexos	101
Anexo I Encuesta a desarrolladores	101
Anexo II Encuesta a usuarios	107
Anexo III: Métricas para el proceso de administración de requerimientos	111
Anexo IV: Buenas prácticas utilizadas. Salidas de los análisis Chi-cuadrado para cada practica en función del tipo de software	113
Anexo V: Análisis de cumplimiento	120
Anexo VI Técnicas de elicitación utilizadas según la alternativa de software	125
Anexo VII Entrevista realizada a desarrolladores	131
Anexo VIII Análisis de varianza	134



Referencias	138
Otros Materiales Consultados	144

Índice de Figuras

Figura 1: Costo de reparación de un error en función de la etapa en la que se detecta.....	12
Figura 2: Factores del coste en proyectos de software reales [Standish:1994].....	13
Figura 3: Ciclo de vida del conocimiento	14
Figura 4: Procesos de la ingeniería de requerimientos [Wieggers:1999].....	23
Figura 5: Actividades principales de la administración de requerimientos [Wieggers:1999].....	27
Figura 6: Desarrollo tradicional [Creative Data: 1995_1]	32
Figura 7: Win-Win Pasos a seguir [Klier]	37
Figura 8: WinWin – Situaciones posibles [Boehm]	38
Figura 9: Win Win – Modelo en espiral [Boehm].....	38
Figura 10: Técnicas vs. necesidades de elicitación [Sindre:2005]	42
Figura 11: Tipos de software	51
Figura 12: Tiempo estimado de desarrollo vs. tiempo real de desarrollo.....	52
Figura 13: Nivel de experiencia de las personas que participaron	52
Figura 14: Causas del desvío.....	53
Figura 15: Grado de satisfacción del cliente respecto del software desarrollado	53
Figura 16: Impacto del grado de satisfacción del usuario en la empresa.....	54
Figura 17: Etapas del ciclo de vida del software tenidas en cuenta a la hora del desarrollo	55
Figura 18: Importancia de la etapa de requerimientos según la percepción de los desarrolladores	55
Figura 19: Asignación de personas para la elicitación de requerimientos.....	57
Figura 20: Surgimiento de la idea del desarrollo del software.....	58
Figura 21: Técnicas de elicitación de requerimientos utilizadas.....	58
Figura 22: Definición de las técnicas de elicitación.....	59
Figura 23: Problemas de entendimiento por parte del cliente sobre las actividades de elicitación.....	59
Figura 24: Categorías de usuarios	60
Figura 25: Cantidad de software empaquetado utilizado por los usuarios	61
Figura 26: Cantidad de software a medida utilizado por los usuarios	61
Figura 27: Razón por la preferencia de sw empaquetado.....	62
Figura 28: Razón por la preferencia de sw a Medida.....	63
Figura 29: Via de conocimiento de la funcionalidad ofrecida por el sw empaquetado.....	63
Figura 30: Medida en la que el sw cumple con las expectativas	64
Figura 31: Factores atribuibles a la demora.....	65
Figura 32: Técnicas de elicitación utilizadas para el relevamiento de los requerimientos de cambio en sw empaquetado	65
Figura 33: Nivel de participación en el desarrollo de un software empaquetado.....	66
Figura 34: Nivel de participación en el testing de un software empaquetado.....	66
Figura 35: Porcentaje de utilización de sw a medida, según la percepción del usuario	67
Figura 36: Definición del objetivo y el alcance del producto a medida	67
Figura 37: Técnicas de elicitación utilizadas para el relevamiento de requerimientos en sw a medida	68
Figura 38: Comparación de la utilización de buenas prácticas para la elicitación de requerimientos entre software a medida y empaquetado.....	70
Figura 39: Comparación técnicas de elicitación utilizadas para sw a medida y empaquetado.....	72
Figura 40: Primer plano factorial asociado al análisis de correspondencias múltiples.....	74
Figura 41: Técnicas de elicitación utilizadas según la opinión de los usuarios.....	75
Figura 42: Porcentaje de cumplimiento del sw a medida con respecto a los requerimientos establecidos	76
Figura 43: Porcentaje de cumplimiento del sw empaquetado respecto a los requerimientos establecidos	76

Figura 44: Adaptabilidad del software de acuerdo a la alternativa de software.....	77
Figura 45: Preferencias de los usuarios según características genéricas del software	78
Figura 46: Utilización de métricas para el proceso de requerimientos	79
Figura 47: Utilización de métricas para el proceso de elicitación de requerimientos	80
Figura 48: Utilidad en la toma de métricas	80
Figura 49: Métricas tomadas luego de la elicitación de requerimientos	81
Figura 50: Medición de la satisfacción del usuario	81
Figura 51: Métricas según el orden de mérito.....	92
Figura 52: Encuesta desarrolladores – Página 1	101
Figura 53: Encuesta desarrolladores – Página 2	103
Figura 54: Encuesta desarrolladores – Página 3	105
Figura 55: Encuesta desarrolladores - Página 4.....	106
Figura 56: Encuesta usuarios – Página 1.....	107
Figura 57: Encuesta usuarios – Página 2.....	110
Figura 58: Encuesta usuarios - Página 3	110
Figura 59: Listado de métricas para el proceso de administración de requerimientos [Loconsole:2003].	112

Índice de Tablas

Tabla 1: Propuesta de métricas para el proceso de elicitación de requerimientos	82
Tabla 2: Estadística descriptiva. Valores promedios, mínimos y máximos para cada métrica.....	85
Tabla 3: Estadística descriptiva para cada variable según la alternativa de software, significancia estadística de la diferencia de valores promedios entre software a medida y empaquetado y de la asociación de la variable con la satisfacción del usuario y de los desarrolladores de software.	86
Tabla 4: Métricas propuestas por los desarrolladores de software.....	93
Tabla 5: Propuesta de métricas directas	98
Tabla 6: Propuesta de métricas indirectas	99

Propósito de la tesis

Esta tesis tiene como propósito principal la comparación del proceso de elicitación de requerimientos tanto para software empaquetado como para software a medida. El tema surgió como una necesidad planteada en un congreso argentino de informática durante el año 2004 (33 JAIIO en la Ciudad de Córdoba, Septiembre de 2004) en el que se resaltó la falta de desarrollo de software empaquetado en contraposición con el desarrollo de software a medida existente en la industria nacional. Surgió también la necesidad de expresar la importancia que posee la etapa de requerimientos para el éxito de los proyectos de desarrollo de software y más precisamente la importancia de la etapa de elicitación de los mismos para la cuál no existen métricas específicas a pesar de su impacto sobre el éxito del proyecto de desarrollo. En este trabajo no sólo se plantea la comparación del proceso de elicitación entre las dos alternativas de software (empaquetado y a medida) sino que además se discute sobre la necesidad e importancia de métricas específicas para lograr el desarrollo de un producto de calidad. El trabajo culmina con una propuesta de métricas para el proceso de elicitación de requerimientos, y con la confección de una guía para la realización del proceso de elicitación de requerimientos.

Organización de la tesis

La tesis se encuentra organizada en 6 capítulos los cuales se detallan a continuación: Capítulo 1: Introducción, define los objetivos generales y específicos del trabajo y planea la hipótesis del mismo, Capítulo 2: Marco Teórico, define y describe los objetivos y metas de la ingeniería de requerimientos, explicando los distintos procesos que la componen y define en forma detallada las técnicas de elicitación más citadas en la literatura. Además presenta una descripción de dos alternativas de software: software a medida y software empaquetado. Capítulo 3: Materiales y Métodos, define los materiales (encuestas y entrevistas) y los métodos estadísticos utilizados para validar la hipótesis postulada en el capítulo 1; Capítulo 4: Resultados, plantea los hallazgos obtenidos a través del análisis de las dos encuestas realizadas y el análisis de las entrevistas a desarrolladores respecto al tema métricas; el Capítulo 5: Propuesta de Métricas para el proceso de elicitación de requerimientos, esta sección plantea el objetivo y los beneficios de la realización de mediciones durante el proceso de requerimientos y propone una serie de métricas directas e indirectas para el proceso de elicitación de requerimientos específicamente y el Capítulo 6: Discusión y conclusiones donde se realiza una serie de comentarios finales basados en los resultados obtenidos.

Agradecimientos

Quisiera agradecer enormemente a todas aquellas personas que me ayudaron e incentivaron para la realización y finalización de esta maestría, entre las que puedo nombrar: Ing. Diego Rubio, quien siempre me apoyó en todas mis actividades, AUS Juan Carlos Cuevas y al Ing. Álvaro Ruiz de Mendarozqueta. También quisiera agradecer a la Dra. Mónica Balzarini por su continua guía y enseñanza. y a todas las personas que se tomaron la molestia de responder a las encuestas y entrevistas.

Capítulo 1: Introducción

Actualmente en la Argentina existe una creciente demanda de exportación de productos de Software. Este hecho inusual representa una gran oportunidad y a la vez un desafío debido a que nuestro país posee importantes capacidades para desarrollar software a medida (desarrollado para satisfacer una demanda particular), pero carece de éstas para el desarrollo de productos empaquetado (desarrollado para satisfacer las necesidades de un mercado común). No obstante, la necesidad de desarrollo de software empaquetado fue expresada durante la 33 JAIIO en Córdoba, Septiembre de 2004.

Para conseguir y asegurar mercados internacionales es necesario desarrollar productos de calidad, es decir, desarrollar el producto que el cliente/mercado desea, dentro del cronograma establecido y con los costos de desarrollo definidos. El primer paso en este camino lleva a cumplimentar la etapa de desarrollo de requerimientos. El desarrollo de requerimientos, así como la administración de los mismos, es importante ya que si se realizara de manera errónea se podría estar desarrollando el software incorrecto [Bohem:1981].

En la Fig. 1 se presenta el costo relativo de reparación de un error del producto en función de la etapa de desarrollo en la que se detecta. Se pone de manifiesto que cuanto más tarde en el ciclo de vida se detecta un error, más cuesta repararlo [Bohem:1981].

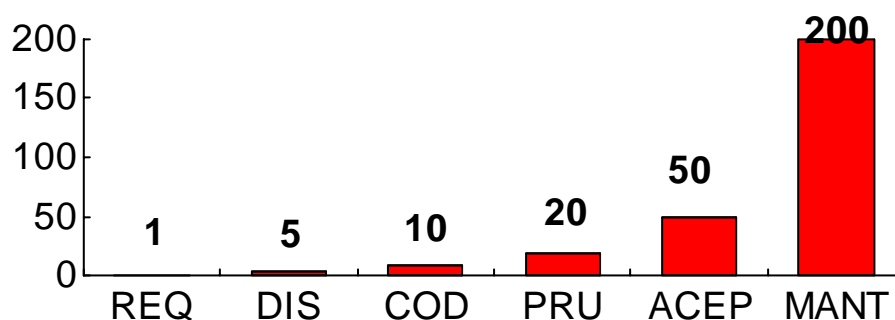


Figura 1: Costo de reparación de un error en función de la etapa en la que se detecta.
Referencias: REQ: requerimiento, DIS: diseño, COD: codificación, PRU: prueba, ACEP: aceptación, MANT: mantenimiento [Bohem:1981]

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

Como se visualiza en la Fig. 2, los problemas con los requerimientos constituyen una de las fuentes principales de inconvenientes (37%) [Standish:1994]

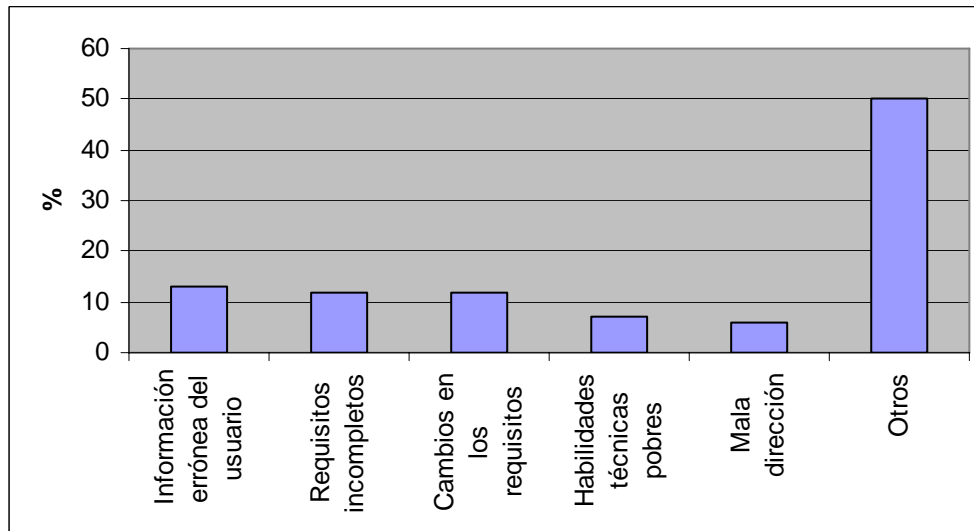


Figura 2: Factores del coste en proyectos de software reales [Standish:1994]

Los errores en la etapa de requerimientos tienen un importante impacto [Standish:1994]:

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

Entre los problemas asociados a la ingeniería de software, los requerimientos son los problemas más frecuentes. Se debe asignar gran esfuerzo en la recolección de requerimientos lo antes posible en el desarrollo del sistema. Puede suceder que: los requerimientos no reflejan las necesidades reales de los clientes y/o que los requerimientos son inconsistentes entre si y/o son incompletos. Estos problemas provienen de un mal entendimiento entre los clientes, las personas que desarrollan los requerimientos y los ingenieros de software que desarrollan o mantienen el sistema [COMP]. El principal problema en la recolección de requerimientos es de comunicación. Por otro lado, es casi imposible probar formalmente que todos los requerimientos reflejan las necesidades reales de los usuarios. La naturaleza difusa de la recolección de requerimientos convierte a esta tarea en un eslabón débil de la cadena de actividades de desarrollo de software. Al mismo tiempo, la recolección de requerimientos es una de las fases más importantes ya que mediante ésta se determina cuál es la naturaleza del producto que se va a desarrollar. Todas las fases subsecuentes en el desarrollo están basadas en la recolección de requerimientos. Si los requerimientos son

incorrectos, el software desarrollado puede representar un buen código pero no satisfará al cliente, y por tanto el trabajo será en vano [COMP].

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

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

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

Estos dos tipos de conocimiento componen lo que se conoce como “ciclo de vida del conocimiento” que se esquematiza en la Fig. 3. El problema de no conectar el conocimiento del usuario con el conocimiento adquirido en el diseño ocurre frecuentemente en la práctica y conduce a productos con más funciones de aquellas que los usuarios necesitarán. La elicitación del conocimiento del usuario proveerá un enlace desde el usuario hacia el desarrollador del producto [Sakol:2001].

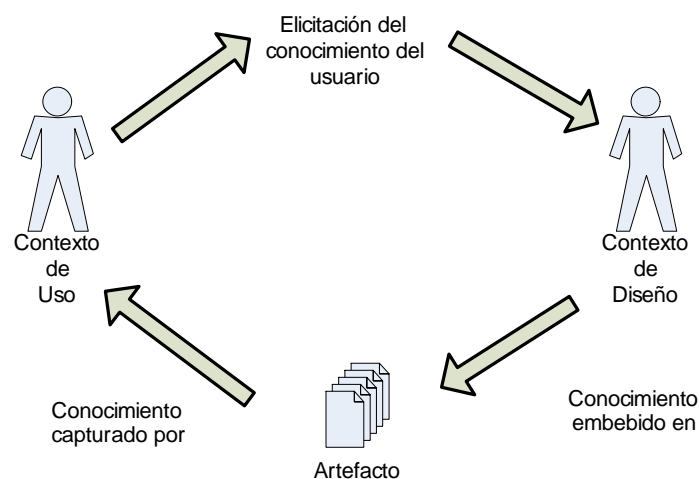


Figura 3: Ciclo de vida del conocimiento

La elicitación de requerimientos es el proceso de descubrir los requerimientos para un sistema a través de la comunicación con los clientes, usuarios del sistema

y otras personas que tengan algún tipo de interés y conocimiento sobre el producto a desarrollar [Madigan].

El principal resultado del proceso de elicitación de requerimientos es la serie de requerimientos que deberán ser utilizados por el equipo de desarrollo de software para crear un producto correcto y de manera apropiada. Además de este resultado principal, el proceso ofrece una serie de salidas intangibles. Cuando existe un buen proceso de elicitación de requerimientos, se ayuda a los usuarios a entender qué es lo que quieren, qué es lo que necesitan, cuáles son las restricciones y alternativas, ventajas y desventajas de cada una. Desde el punto de vista de los ingenieros y desarrolladores, la existencia de un buen proceso de elicitación de requerimientos ayuda a los mismos a resolver el problema correcto, es decir, lo que realmente desea y necesita el cliente, a resolver un problema factible, a ganar la confianza del cliente y su cooperación y a ganar conocimiento sobre el dominio del problema [Tuffley:2005].

Cuando el proceso de elicitación es pobre es posible que se dé respuesta a un problema equivocado, los usuarios podrían expresar su descontento ya que los desarrolladores no los escuchan y se produciría un desarrollo caótico en el sentido que se pierde información, se toman decisiones incorrectas, los costos y el cronograma se salen del plan y se pierde dinero [Tuffley:2005].

La alternativa de producto de software que se desarrolla podría afectar el proceso de elicitación ya que el contacto con el usuario y el conocimiento del mismo surgen desde medios diferentes según se desarrolle software a medida o empaquetado. Los usuarios para una u otra alternativa de software tienen diferentes expectativas y, muy probablemente, diferente percepción del proceso de elicitación de requerimientos.

Existen diferentes elementos a elicitar durante la fase de requerimientos [Nuseibeh]: límites del sistema, *stakeholders*, metas, tareas, factibilidad, riesgos, etc. La literatura sugiere que la elicitación de requerimientos se realiza de distintas maneras, siguiendo una o más de numerosas metodologías disponibles; sin embargo la práctica de implementación de este proceso pareciera tener aún más variantes debido a la dificultad en formalizar indicadores directos o indirectos (métricas) para la evaluación y monitoreo del proceso. Algunos de los beneficios más inmediatos y cuantificables de las métricas respecto a los requerimientos son [TickIT:2002]:

- Validación de la completitud de la definición de los requerimientos
- Identificación y medición de la conformidad de los requerimientos de software
- Identificación y medición del logro de los objetivos de diseño del software
- Validación de la completitud de los objetivos de testing del software

- Identificación del criterio de aceptación del usuario por un producto de software completo
- Provisión de una medida de confianza en los requisitos de calidad específicos del producto, por ejemplo, de confiabilidad y de funcionalidad

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

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

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

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

En el Anexo III se presenta una lista relativamente extensa de métricas para el proceso de administración de los requerimientos [Loconsole;2003].

Lo expuesto anteriormente sobre métricas muestra que existen distintas propuestas para el proceso de ingeniería de requerimientos, más específicamente para los cambios y administración de los mismos. Sin embargo, es importante notar que no sucede lo mismo a nivel del proceso de elicitación de requerimientos propiamente dicho.

Objetivos

Objetivo general

Analizar comparativamente la implementación del proceso de elicitación de requerimientos en el desarrollo de software a medida y software empaquetado en Argentina.

Objetivos específicos

1. Describir técnicas de elicitación que pueden ser utilizadas para el desarrollo de software a medida y/o empaquetado.
2. Analizar comparativamente la implementación de dichas técnicas y buenas prácticas recomendadas para la elicitación según la alternativa de software a desarrollar, tanto desde la perspectiva de los desarrolladores como desde la perspectiva de los usuarios.

3. Clasificar las métricas propuestas para la etapa de elicitación de requerimientos de acuerdo con su potencialidad bajo los distintos contextos definidos según la alternativa de software.

Hipótesis

La alternativa de software (a medida o empaquetado) condiciona el proceso de elicitación de requerimiento.

El uso de métricas apropiadas para la evaluación y monitoreo del proceso permiten generar conocimiento que conduce a obtener mayores beneficios no sólo en la ingeniería de requerimientos sino también a la elicitación de los mismos.

Capítulo 2: Marco teórico

Alternativas de productos de software

El software empaquetado constituye una porción significativa del mercado del software a nivel internacional y presenta particularidades ausentes durante el desarrollo de un software a medida. Las principales diferencias se encuentran en los modelos del ciclo de vida y la fase de arquitectura [Oudshoorn:2004]. En esta tesis se analizarán las similitudes y diferencias en la ingeniería de requerimientos, mas específicamente durante la elicitación de requerimientos.

Software empaquetado

Muchas veces son la solución ideal. En ciertos rubros donde la forma de trabajo es simple y no hay demasiadas particularidades, un software empaquetado dirigido al rubro puede constituir la opción correcta. Existen actividades o negocios bastante estandarizados como por ejemplo remiserías y kioscos, donde no tiene sentido desarrollar un software a medida ya que por menor costo se puede adquirir uno empaquetado que cubre todos los requerimientos. En numerosas ocasiones, se encuentra un software empaquetado que fue diseñado para atender a requerimientos iguales a los que se elicitan en una situación particular. El software empaquetado se caracteriza por ser, en general, más económico y se asocia a un proceso de disponibilidad rápido: “Se compra y se instala” [TechRepublic].

Pero es también posible que luego de su funcionamiento se descubra que para ciertas costumbres de trabajo el sistema no sirve o que el sistema quede retrasado ante un cambio de reglas de negocio. En el caso de software empaquetado generalmente es difícil encontrar a personas que respondan por él para hacer una modificación. Si fue desarrollado por una empresa importante, será más fácil solicitar que alguien lo modifique pero, muy probablemente, el costo sea alto. Esto lleva a que en innumerables ocasiones se tiene que modificar la forma de trabajo para que el sistema sirva (cuando debería ser al revés!) [Brain Desarrollos]. Muchas compañías pierden tiempo y dinero tratando de lograr que el software empaquetado encuadre con sus operaciones [TechRepublic].

El software empaquetado es mejor para funciones que cambian lentamente y que no son estratégicas para la empresa, como por ejemplo la contabilidad. Sin embargo, si se piensa que una función es estratégica o que el modelo de negocio es único las soluciones empaquetadas que han sido desarrolladas para cubrir un espectro amplio de requerimientos podrían ser insuficientes para cubrir requerimientos específicos [TechRepublic].

La selección de un software empaquetado es una tarea difícil que consume tiempo. Existen algunos principios y criterios de evaluación que se debieran considerar para realizar una elección exitosa de esta alternativa de software [TechRepublic]:

- El costo de compra del software empaquetado debe ser evaluado cuidadosamente ya que existen modelos que varían ligeramente en cuanto a dar respuesta a un problema específico pero poseen precios muy distintos. Asignar un costo a un software empaquetado no es una tarea sencilla. Algunos vendedores tasan al software dependiendo de la cantidad de usuarios conectados, otros por el nombre del usuario (aunque no esté conectado). Algunos lo tasan a partir de la plataforma utilizada. El costo también puede variar en el caso de que se incluya un mantenimiento anual, actualizaciones periódicas, etc.
- Algunos software empaquetados están fuertemente integrados en su funcionalidad, mientras que otros pueden ser instalados fácilmente en módulos.
- Algunos software empaquetados poseen una herramienta basada en interfaces para configurar y mantener fácilmente el software. Otros requieren el aprendizaje de un lenguaje o base de datos propietaria que puede involucrar un período muy extenso de entrenamiento de los usuarios. En compañías grandes, la implementación de la primera versión del software empaquetado generalmente lleva meses, y es una tarea difícil, ya que el software es un medio de conocimiento, entonces toda la organización necesita ser adaptada a los procesos de trabajo del software, la implementación se convierte así en una actividad que consume tiempo, que es riesgosa y costosa [Software improvement group:2005].
- Mantener el software empaquetado lo más simple como sea necesario para realizar el trabajo. El software empaquetado puede tener una flexibilidad limitada o bien atender a cada una de las características que podrían presentarse. Antes de la selección, hay que recordar que cada una de esas características o funcionalidades debe ser mantenidas, los usuarios deben ser entrenados, y el hardware debe ser actualizado para que sea apropiado para soportarlas.
- Algunos vendedores de software empaquetados poseen diferentes enfoques con respecto a la arquitectura, se debe asegurar que esa arquitectura sea compatible con la de la empresa.

A continuación se presenta una serie de preguntas que convendría realizarse al momento de comprar un software empaquetado [Informat:2002]:

- Soporta estándares abiertos, sistemas operativos y hardware populares?.
- Posee herramientas de desarrollo para la modificación de las estructuras de datos y las pantallas de entrada?
- Posee APIs para conectarse a otros software empaquetados?
- Soporta herramientas de administración de sistemas populares como por ejemplo: Tivoli?
- Utilidades de back-up, administración de performance y sistemas?

“El software empaquetado es mejor para funciones que cambian lentamente y que no son estratégicas para la empresa.”

Software a medida

En muchas oportunidades el cliente necesita una aplicación determinada que cumpla con una tarea pequeña que en realidad es bastante importante. Estas situaciones pueden ser de lo más variadas y no hay software empaquetado que las cumpla. En estas ocasiones, una aplicación a medida es la solución. Sin embargo, el software a medida tiene fama de demorar demasiado tiempo entre el desarrollo y la depuración de errores y de tener alto costo. La sola mención de encargar un sistema a medida pareciera provocar dolor de cabezas a cualquier emprendedor. Pero la realidad es bastante diferente [Brain Desarrollos].

El sistema a medida se hace codo a codo entre quién lo encarga y quién lo desarrolla, la elicitación de requerimientos surge de la comunicación directa entre desarrolladores y usuarios. Es un sistema que está hecho exclusivamente para quien lo necesita abarcando cualquier situación prevista y adaptándose lo mejor posible a los requerimientos de cada cliente particular. Si hay cualquier cambio en una situación que implique un cambio en el programa, se tendrá previo contacto con la persona que lo diseñó para implementar el cambio. Así, se puede pensar que el sistema a medida es más flexible a las necesidades y nuevos requerimientos que el sistema empaquetado. Un software correctamente diseñado, es una solución invaluable para cualquier empresa. El costo de una forma de trabajo ineficiente, a largo plazo, es enormemente mayor al del desarrollo de un sistema a medida [Brain Desarrollos].

“Un software correctamente diseñado, es una solución invaluable para cualquier empresa.” [Brain Desarrollos]

Algunos autores [Software improvement group:2005] señalan que aunque el software comparte una cantidad de propiedades con “los productos”, no es simplemente un producto. El software debe entenderse como un medio para el almacenamiento de conocimiento. Los usuarios que adquieren software empaquetado, en realidad están adquiriendo conocimiento sobre su proceso de negocio o resolución de problema. Eso está bien en el caso que dicho conocimiento no se encuentre disponible para el usuario, pero sería cuestionable si el conocimiento está disponible. En tal caso, el usuario (compañía) abandona una situación donde el conocimiento es suyo para alcanzar otra donde el

conocimiento específico se pierde a favor de una solución desarrollada en sentido amplio que pone ese conocimiento disponible también para la competencia de su negocio. Más de una vez sucede, que las decisiones referidas a la alternativa de software no son tomadas explícitamente ya que el software no es considerado como un medio para el almacenamiento de conocimiento [Software improvement group:2005].

El desarrollo del software a medida es competitivo ya que el software en general cambia constantemente. Como el usuario (compañía) cambia y se adapta, el software también necesita ser adaptado a los clientes, a la compañía, a la competencia y al mercado. La implementación de un sistema de software a medida puede realizarse en menos tiempo que la del un software empaquetado ya que no es necesario cambiar la forma de trabajo para adaptarse a los procesos del software [SystemsAlliance].

A pesar de lo expuesto anteriormente, el debate respecto a la alternativa de software más apropiado continúa y no se puede concluir que el software a medida sea siempre más apropiado. Algunos autores [SystemsAlliance] citan que las organizaciones seleccionan productos de software empaquetados en vez de los productos de software a medida para poder [SystemsAlliance]:

- Acelerar el proceso de implementación.
- Decrementar los riesgos del desarrollo y costos que se incurren en los proyectos de desarrollo de software a medida.
- Planificar ciclos de mantenimiento predecibles.
- Aumentar la confiabilidad dentro del ciclo de vida del producto a un costo menor.
- Permitir una arquitectura extensible

Requerimientos e ingeniería de requerimientos

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

- son expresiones de las necesidades de *stakeholders* para alcanzar una meta particular. [Nuseibeh]
- una condición o capacidad necesaria dada por un usuario con el objetivo de resolver un problema o alcanzar un objetivo [IEEE610:1990]
- expresan las necesidades y restricciones atribuibles a un producto de software que contribuye a la solución de algún problema del mundo real [Kotonya:2000]

“Requerimientos son expresiones de las necesidades...”[6]

Los requerimientos pueden ser categorizados en varios niveles de abstracción, importancia, alcance, exactitud y detalle [COMP]. Por ejemplo:

- Requerimientos muy generales que expresan con términos amplios qué es lo que el sistema debería hacer.
- Requerimientos funcionales que definen partes de la funcionalidad del sistema.
- Requerimientos no funcionales que agregan restricciones al desarrollo del sistema
- Requerimientos de implementación que declaran cómo el sistema debe ser implementado.
- Requerimientos de performance que especifican una performance mínima aceptable para el sistema
- Requerimientos de usabilidad que especifican el tiempo máximo aceptable para demostrar el uso del sistema

Los requerimientos pueden ser clasificados como funcionales o de calidad. Los requerimientos funcionales describen qué es lo que el sistema debe hacer para dar soporte a las funciones y objetivos del usuario [Sommerville:1997]. Deben responder a las siguientes preguntas [DOE]:

- Cómo las entradas son transformadas en salidas?
- Quién inicia y recibe información específica?
- Qué información debe estar disponible para que cada función sea ejecutada?

Los requerimientos de calidad o no funcionales imponen restricciones de cómo los requerimientos funcionales deben ser implementados [Sommerville:1997]. Son derivados de las necesidades identificadas. El enfoque para la especificación de requerimientos de calidad está basado en la definición de un modelo de calidad. Un modelo de calidad debe ser utilizado para especificar los requerimientos de calidad. La ISO/IEC 9126-1 [ISO9126-1:2001] provee dicho modelo [ISO/IEC 25030:2003].

Definidos así los requerimientos, se intentará responder a las preguntas ¿qué es? y ¿qué abarca? la Ingeniería de Requerimientos:

- La Ingeniería de Requerimientos es un proceso cíclico que involucra: la elicitación, especificación, validación y verificación de requerimientos [Hans van Vliet].
- La Ingeniería de Requerimientos se ocupa de la elicitación, análisis, especificación y validación de requerimientos de software [SWEBOK:2004].
- Es un término relativamente nuevo, que abarca las actividades de descubrir, documentar y mantener una serie de requerimientos para un sistema basado en computadora. El uso del término “ingeniería” implica que se deben usar una serie de técnicas sistemáticas y repetibles para asegurar que los requerimientos sean completos, consistentes y relevantes, [Sommerville:1997].

En otras palabras, la ingeniería de requerimientos involucra el descubrir cuáles son las metas, necesidades y expectativas de los *stakeholders*, ajustar las expectativas de los mismos y comunicarlas a los desarrolladores [Nuseibeh].

La ingeniería de requerimientos está formada por una serie de procesos bien diferenciados (Fig. 4) [Wiegiers:1999].

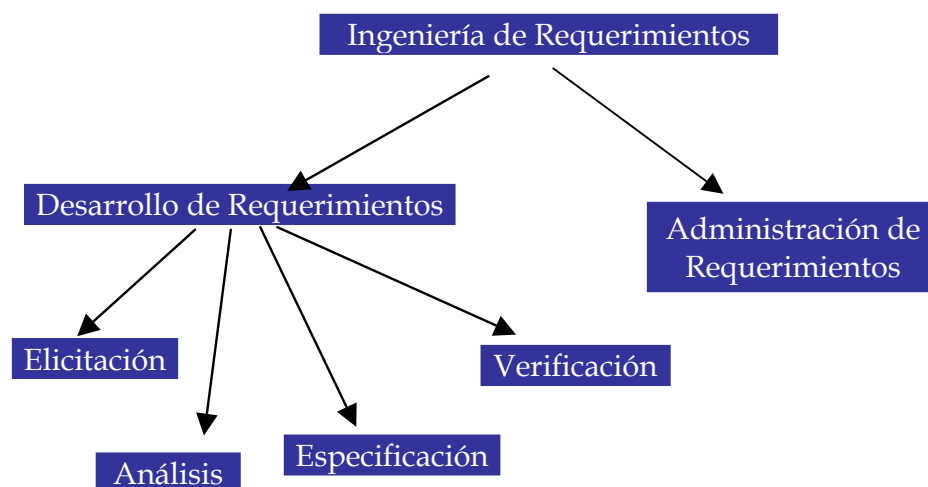


Figura 4: Procesos de la ingeniería de requerimientos [Wiegiers:1999]

Proceso de elicitación de requerimientos:

La elicitación del conocimiento del usuario, fue representada unidireccionalmente como una forma de transferencia del conocimiento adquirido en el contexto del uso de un producto hacia el conocimiento asociado al diseño (Fig.3). La definición y especificación de las necesidades del usuario es una tarea difícil, pero aún más compleja es la tarea de la comunicación de estas de necesidades hacia los desarrolladores. Esta comunicación debe llevar a un entendimiento acabado del producto y cuales son las necesidades reales del mismo.

La elicitación es la habilidad de trabajar en colaboración con los clientes y/o representantes de ellos para descubrir las necesidades actuales del producto y acordar la visión y las metas del proyecto propuesto. Ya que las personas que van a interactuar con el sistema, ya sea este empaquetado o a medida, no demandan exactamente las mismas funciones, parte del proceso de elicitación es la identificación temprana de las diferentes clases de usuarios y sus características [Borland:2005].

El proceso de elicitación de requerimientos trata de identificar la procedencia de los requerimientos y la manera en la que el ingeniero de software los puede recolectar. Es la primera etapa en la construcción de un entendimiento del problema que el software debe resolver. Es fundamentalmente una actividad humana, y es en donde los *stakeholders* se identifican y comienzan a establecer las relaciones entre el equipo desarrollador y el cliente. A este proceso también se lo conoce como “Captura de requerimientos”, “Descubrimiento de requerimientos” y “Adquisición de requerimientos” [13; 14; 15; 16].

Este proceso, que merece una inversión por adelantado, debe ser llevado a cabo por un analista de negocios hábil que debe cumplir un rol crítico para la administración del proyecto forjando un enlace intrínseco entre las necesidades de los *stakeholders*, el sistema y los requerimientos de software [Borland:2005].

Dentro de las actividades principales definidas en la elicitación de requerimientos se encuentran: la identificación de los *stakeholders*, la elicitación de los requerimientos funcionales, la identificación de restricciones, la definición de los escenarios y la identificación de los requerimientos de calidad [ISO/IEC 25030:2003].

“Elicitación es aprender a preguntar consistentemente “Por qué?...” [7].

Proceso de análisis de requerimientos

En el proceso de análisis de requerimientos se trata, precisamente, de analizar la información recibida desde los usuarios, para distinguir las necesidades de tareas, los requerimientos funcionales, atributos de calidad, soluciones sugeridas, de información extraña [Wieggers:1999].

Se define a la etapa de análisis de requerimientos como la actividad de transformar requerimientos informales en requerimientos técnicos mediante el aseguramiento de que los mismos reflejan los atributos de calidad de los requerimientos y que expresan las necesidades de los clientes. El análisis es una actividad iterativa. Los pasos del proceso deberán ser repetidos una cierta cantidad de veces, existe una constante comunicación entre la consulta de los clientes, los usuarios finales y los desarrolladores [Avionics:1998].

Es el proceso de razonamiento acerca de los requerimientos que han sido elicitados; involucra actividades como el examen de requerimientos en busca de conflictos o inconsistencias, la combinación de requerimientos relacionados y la identificación de requerimientos perdidos [Tuffley:2005].

Proceso de especificación de requerimientos

Es el proceso de grabado o el registro de los requerimientos en una o más formas, incluyendo el lenguaje natural y formal, representaciones simbólicas o gráficas [Tuffley:2005]. La especificación de los requerimientos es el paso en donde los resultados de la identificación de los requerimientos se “retratan” [Brackett:1990].

“La ambigüedad es la causa raíz de la mayoría de los problemas del entendimiento y documentación de los requerimientos.” [7]

Como regla general, los requerimientos funcionales que describen qué es lo que el sistema y el software deben hacer deben estar separados de los requerimientos no funcionales que son los requerimientos de atributos de calidad especificados por el cliente, como son confiabilidad, seguridad y escalabilidad [Borland:2005].

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

Dado que los usuarios muchas veces no son capaces de pensar en todas las situaciones posibles en las cuales el software puede ser utilizado, es tarea de los desarrolladores documentar los requerimientos desde un punto de vista que admita el *testing*. En este proceso se darán a conocer posibles situaciones que no se hubiesen tenido en cuenta la primera vez [Borland:2005].

Proceso de validación de requerimientos

Es el proceso de confirmación con el cliente o usuario del software que los requerimientos son válidos, correctos y completos. La validación es crítica para resaltar las disparidades entre las perspectivas de los *stakeholders* y para descubrir suposiciones que pueden quedar enmascaradas en la comunicación oral [Borland:2005].

El proceso de validación de requerimientos se define como, un paso dentro de la ingeniería de requerimientos para establecer los criterios y técnicas para asegurarse que el software cumple con los requerimientos [Brackett:1990]. El cliente y los desarrolladores deben llegar a un acuerdo sobre el criterio de aceptación y las técnicas o métricas a ser utilizadas durante el proceso de validación, como por ejemplo, la ejecución de un plan de *testing* para determinar que el criterio ha sido alcanzado [Collofello:1988]. Dentro del proceso de validación se pueden nombrar las siguientes actividades [Avionics:1998]:

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

Proceso de administración de requerimientos

No es suficiente para las organizaciones recolectar requerimientos desde múltiples *stakeholders* e incorporarlos independientemente dentro de un sistema. Es necesario administrar estos de manera simultánea, además alguien, o algún equipo, debe ser responsable de administrar esos requerimientos a lo largo del ciclo de vida de manera tal de mantener la visibilidad y control del proceso de entrega del software [Borland:2005].

El acuerdo de los requerimientos es el puente que une el desarrollo de los requerimientos y la administración de los mismos. La administración de los requerimientos incluye todas las actividades que mantienen la integridad y exactitud de los requerimientos a medida que el proyecto progresa. Como se muestra en la Fig. 5 la administración de los requerimientos enfatiza [Wiegers:1999]:

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

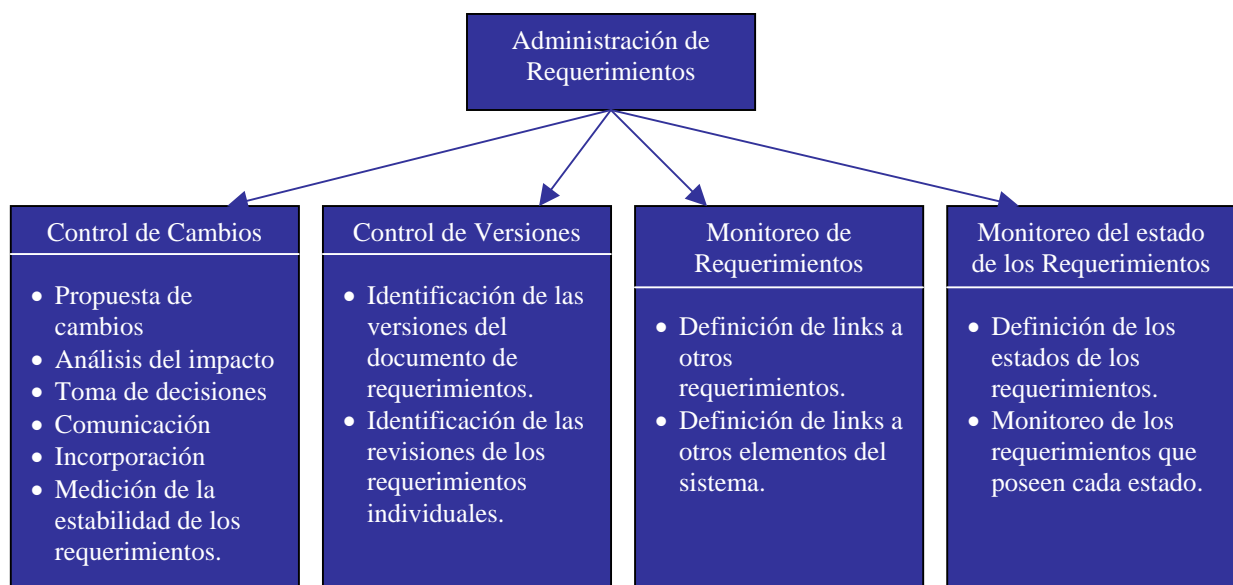


Figura 5: Actividades principales de la administración de requerimientos [Wieggers:1999]

Dificultades en el proceso de elicitación de requerimientos

Como se dijo anteriormente la elicitación de requerimientos es el primer y más crítico paso dentro de la ingeniería de requerimientos. Realizado de manera errónea nos llevará a productos con calidad pobre, fechas de entregas tardías y costos fuera del presupuesto [Bohem:1981].

El propósito de la elicitación es recolectar información de los clientes (problemas, necesidades, etc) y capturar esa información en el formato de posibles requerimientos (tanto técnicos como no técnicos) junto con la justificación que da soporte a cada requerimiento [Avionics:1998].

Durante la elicitación de requerimientos es necesario identificar todas y cada una de las personas afectadas por este proceso. Esta primera actividad es importante ya que es necesaria la elicitación desde múltiples puntos de vista porque sola persona no sabe todo. Durante este proceso de identificación temprana de *stakeholders* es necesario tener en cuenta, al menos, a los líderes de proyectos, a los ingenieros de requerimientos de software, dentro de las áreas de soporte a los especialistas en documentación y a los potenciales usuarios [Tuffley:2005], entre otros (cliente, gobierno, etc).

La complejidad del proceso se asocia a una serie de dificultades en su implementación [Madigan]:

- Problemas de articulación: se pueden presentar varios escenarios. El usuario está conciente de cuáles son sus necesidades pero no es capaz de expresarlas; el usuarios no está conciente de cuáles son sus necesidades; el usuario está conciente de cuáles son sus necesidades pero siente miedo al expresarlas; los usuarios no pueden decidirse sobre lo que desean y necesitan; el desarrollador no saben escuchar a los usuarios; el desarrollador no entiende lo que los usuarios le están expresando; el desarrollador es muy dominante e intimida al usuario a expresar sus necesidades.
- Barreras en la comunicación: medios de comunicación insuficientes; diferentes preocupaciones entre los usuarios; diferentes estilos de interacción entre el desarrollador y el usuario; diferentes tipos de personalidades y valores.
- Limitaciones en el conocimiento: pobre conocimiento del dominio; preconcepción de una solución; dificultades con la escala y la complejidad del problema a resolver
- Problemas de comportamiento humano: ambigüedades; los usuarios asumen que otra persona será el encargado de mencionarle la necesidad al desarrollador; el desarrollador asume que se le proveerá toda la información completa del dominio; los usuarios asumen que el desarrollador hará las preguntas correctas.

- Problemas técnicos: los requerimientos cambian con el tiempo; la tecnología cambia rápidamente; existen demasiadas fuentes de requerimientos; la similitud o novedad del sistema.

Técnicas de elicitación de requerimientos

Para lograr minimizar estas dificultades, a continuación se presenta una breve descripción de alguna de las técnicas de elicitación conocidas:

1. Entrevistas

Las entrevistas elicitación información detallada de un individuo. Para su correcta aplicación requieren habilidades de comunicación social, capacidad para escuchar y conocimiento de tácticas de entrevistas. Se definen cuatro fases para la realización de entrevistas [Tuffley:2005]:

- Identificación de candidatos: se deben seleccionar cuidadosamente los candidatos a ser entrevistados, ya que un candidato mal seleccionado podría resultar en una pérdida de tiempo o en la elicitación de datos incorrectos. Una de las primeras persona que se debe entrevistar es el auspiciante del proyecto, es decir la persona que patrocina la realización del software. Luego se debe revisar el organigrama de la empresa o institución que usará el producto para realizar una selección de entrevistados con conocimiento del puesto de trabajo o rol que cumple cada persona. Una opción para la selección de personas a entrevistar es la de preguntar a quienes ya se entrevistó a quien más se podría entrevistar. Esta resulta una buena práctica en el caso que el organigrama de la empresa no esté actualizado y se encuentren personas relevantes para el proyecto cuya función no se encuentra claramente esTablacida en el organigrama.
- Preparación: es la etapa previa a la entrevista propiamente dicha. El entrevistador deberá hacer arreglos con respecto al cronograma de entrevistas, se deberán planear cada una de las entrevistas teniendo en cuenta el tiempo en el que se llevará a cabo y un tiempo adicional para destinarle al entrevistado en el caso que quiera hacer algún comentario extra y para la realización de un resumen de dicha entrevista. Se deberá preacordar, el horario de entrevista y asegurar que el entrevistado acepte la misma previamente. El entrevistador deberá fijar cuál es el objetivo de la entrevista, deberá enviar cualquier información relevante sobre ésta con suficiente tiempo de antelación, de manera tal que el/los entrevistado/s tengan tiempo de preparar el material. Además se deberán preparar una lista de preguntas teniendo en cuenta los tipos (abiertas y/o cerradas), el conocimiento en general que se posea, la coherencia y sentido lógico en la organización de las mismas, el tiempo que insumirá cada pregunta y su respuesta, y el tiempo final para realizar un control general. La lista de preguntas se podrá generar de acuerdo al rol de las personas a entrevistar,

es decir generar una lista de preguntas específicas para el rol que cumple cada uno de los potenciales usuarios. Una práctica recomendable es preparar un cuaderno de anotaciones, asegurarse de llevar algo con que escribir y en el caso de preguntas abiertas un grabador por si el entrevistado permite ser grabado.

- Desarrollo de la entrevista: en el momento que comience la entrevista el entrevistador deberá introducirse y explicar el objetivo y las metas de la entrevista, las preguntas se deberán realizar teniendo en cuenta el orden especificado y tratando de que la conversación no se salga de curso. A medida que el entrevistado va respondiendo las preguntas se deberá ir explorando las respuestas a través del resumen, y refraseo para obtener la confirmación de las mismas. El entrevistador deberá escuchar activamente cada una de las palabras que el entrevistado diga, no se debe anticipar a las respuestas, dejar que el entrevistado termine la frase antes de seguir con la próxima respuesta. Se deberán explorar problemas desde distintos puntos de vista, controlar los errores ya sean de observación, de interpretación, enfoque, ambigüedad u otros conflictos. Se debe permitir que el entrevistado realice algunas preguntas pero que no se vaya del tema. Se deberá tener especial cuidado con el lenguaje corporal, mantener el proceso visible y no olvidar el tiempo de duración de la entrevista. Es importante finalizar la entrevista explicando cuales son las acciones a seguir y agradeciendo la colaboración.
- Continuación: antes de comenzar con una nueva entrevista se deberá realizar un resumen de la entrevista realizada, enviar una copia al entrevistado y pedir confirmación del resumen realizado.

2. Encuestas

La encuesta constituye una técnica simple que permite obtener información de una gran cantidad de personas de una manera rápida y eficiente [Jarke:1993]. La posibilidad de repetir las observaciones o preguntas sobre un número de sujetos hace que la encuesta sea una de las técnicas más apropiadas para proveer datos que pueden ser explorados estadísticamente para determinar tendencias, promedios y desvíos o alejamientos de los valores centrales.

Las encuestas deben elicitar el punto de vista del cliente, sus metas, procesos, problemas, mejoras sugeridas y estructura lógica del producto, entre otros [Green:2003].

Una de las ventajas que provee esta técnica es que las encuestas pueden ser reutilizadas en nuevos proyectos. Una encuesta podría ser tabulada completamente y ser re-expresado en una serie de diferentes diagramas. Las tablas y los diagramas deben ser validados por el cliente. Como resultado se obtiene una manera rápida y efectiva de obtener una caracterización detallada de la perspectiva del cliente [Green:2003].

Se recomienda que las encuestas sean prototipadas y puestas a prueba (pruebas piloto) antes de ser distribuidas en sentido amplio.

3. Tormenta de Ideas

La tormenta de ideas es una técnica simple para la generación de ideas. Se reúne un conjunto de personas relacionadas al producto y éstas sugieren y exploran sus ideas libremente. Generalmente las reuniones se realizan con 4 a 10 personas; una de ellas deberá actuar como líder para comenzar pero no para restringir la expresión de los restantes participantes. Se genera una amplia variedad de puntos de vistas, se estimula el pensamiento creativo, se logra construir una imagen más completa del problema, se provee un ambiente social mucho más confortTabla y de fácil aprendizaje [Tuffley:2005]. La tormenta de ideas involucra tanto a la idea de generación como a la de reducción. Las ideas más creativas e innovadoras generalmente surgen de la combinación de ideas no relacionadas. La tormenta de ideas presenta algunas reglas [Madigan]:

- No está permitida la crítica y el debate.
- Dejar que la imaginación vuele.
- Generar la mayor cantidad de ideas posibles.
- Cambiar y combinar las ideas.
- Reducción de ideas:
 - Eliminar las ideas que claramente no pertenecen al contexto
 - Agrupar las ideas similares en un súper tópico.
 - Priorizar las idean restantes.

4. RAD (*Rapid Application Design*)

RAD o Diseño Rápido de Aplicaciones es un proceso de desarrollo de software que permite la construcción de sistemas útiles en menos de 90 días [Maner:1997]. Es una metodología para comprimir el análisis, el diseño, la construcción y las fases de prueba en una serie de ciclos de desarrollo cortos e iterativos. La técnica tiene un número de ventajas distintas sobre el modelo tradicional o secuencial de desarrollo [Creative Data: 1995_1] (Fig. 6).

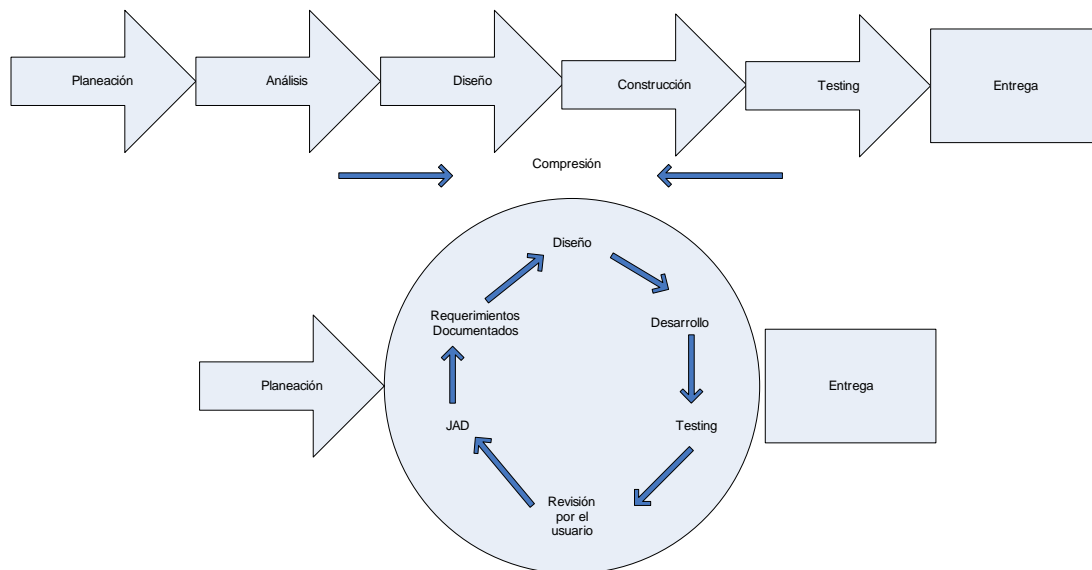


Figura 6: Desarrollo tradicional [Creative Data: 1995_1]

Los proyectos del RAD se forman con pequeños equipos compuestos de desarrolladores, usuarios finales y técnicos. Los equipos pequeños, combinados con los ciclos de desarrollos cortos e iterativos optimizan la velocidad, la visión de unidad y el propósito, tanto como la comunicación informal eficaz y la gerencia del proyecto [Creative Data: 1995_1].

Un principio importante, fundamental del desarrollo iterativo es que cada iteración haga entrega de una versión funcional del sistema final. Esta versión funcional del sistema final es una porción correctamente dirigida, completamente funcional y no es igual que un prototipo. Por ejemplo, la primera iteración pudo entregar 10% del 100%, la segunda iteración 25% y así hasta llegar al 100% [Creative Data: 1995_1].

Dentro de los problemas que RAD soluciona podemos mencionar [Maner:1997]:

- Con los métodos tradicionales de desarrollo, el cliente espera demasiado tiempo para ver algún resultado.
- Con los métodos tradicionales de desarrollo, el desarrollo puede llevar tanto tiempo que el negocio del cliente puede cambiar fundamentalmente para cuando el sistema esté listo para ser utilizado.
- Con los métodos tradicionales de desarrollo, no existe nada hasta que el 100% del proceso está terminado, entonces el 100% del software es entregado.

RAD posee las siguientes características y restricciones [Maner:1997]:

- Usa un equipo híbrido: los equipos deben consistir de 6 personas incluyendo a los desarrolladores y a usuarios del sistema full-time y a cualquier persona con algún interés sobre los requerimientos.

- Los desarrolladores elegidos para formar parte de un equipo RAD deben poseer múltiples talentos, es decir, deben poder ejecutar tareas de analistas, diseñadores y programadores.
- Utiliza herramientas especializadas que dan soporte a: desarrollo visual, creación de prototipos falsos (simulaciones), creación de prototipos funcionales, lenguajes múltiples, cronograma de equipo, trabajo en equipo y colaboración, uso de componentes reusables, uso de APIs estándares, control de versiones, entre otros.
- RAD hace uso de “time boxing”, es decir, que las características secundarias son eliminadas para poder mantenerse de acuerdo al plan.
- Utiliza un modelo iterativo y evolutivo de prototipación
- Los clientes, desarrolladores y la gerencia deben aceptar entregables informales, como por ejemplo:
 - Prototipos en papel en vez de prototipos realizados en computadora
 - Notas de talleres del usuario en vez de documentos formales de los requerimientos.
 - Notas de reuniones de los diseñadores en vez de documentos formales de diseño.
 - PRINCIPIO: crear la menor cantidad de documentación posible, solo la necesaria para facilitar algún futuro desarrollo o mantenimiento.
- El equipo de desarrollo debe poseer la autoridad necesaria para poder tomar ciertas decisiones que en otras circunstancias las hace la gerencia.
- El ciclo de vida del proyecto no debe superar los 6 meses.
- La iteración debe ser utilizada de una manera tal que el proceso del desarrollo converja hacia una solución aceptable.
- La prototipación debe incorporar requerimientos de desarrollo rápidamente, en tiempo real, y obtener consenso temprano.

5. JAD (*Joint Application Design*)

Originalmente JAD fue diseñado para reunir a desarrolladores de sistemas y usuarios y obtener distintas opiniones en un ambiente productivo y creativo. Las reuniones fueron una manera de obtener requerimientos y especificaciones de calidad [Yatco:1999].

Según [Jennerich:1990]; JAD define una serie de actividades *pre-workshop* a llevar a cabo, entre las que podemos mencionar:

- Identificación de los objetivos y limitaciones del proyecto: es vital poseer objetivos claros para el *workshop* y para el proyecto. La definición del alcance identifica las funciones del negocio que están dentro de los límites del proyecto. También se evalúan tanto el diseño del proyecto como la complejidad de la implementación.

- Identificación de los factores críticos de éxito: es importante identificar los factores claves de éxito tanto para el proyecto de desarrollo como para las funciones del negocio que se están estudiando. Cómo sabremos si los cambios planificados han sido efectivos? Cómo se medirá el éxito?
- Definición de los entregables del proyecto: generalmente los resultados de un *workshop* son la documentación y un diseño. Es importante definir el nivel de detalle y el formato que poseerá la documentación. Qué tipo de diagramas se utilizarán? Una buena idea es utilizar las herramientas CASE.
- Definición el cronograma de las actividades del *workshop*: los *workshops* varían en duración entre uno a cinco días. El *workshop* inicial para un proyecto no debería ser menor a tres días. Luego del *workshop* inicial, la construcción del equipo está lista. También se pueden organizar *workshops* de menor duración para las fases siguientes del proyecto, como por ejemplo para verificar un prototipo.
- Selección de los participantes: los participantes de un *workshop* pueden ser, los usuarios del negocio, los desarrolladores y expertos externos.
- Preparación del material del *workshop*: el gerente del proyecto y el moderador ejecutan un análisis y construyen un diseño preliminar en el que el *workshop* se enfocará. El material del *workshop* consiste de documentación, hojas de trabajo, diagramas e incluso apuntes que le permitirán a los participantes entender la función del negocio.
- Organización de las actividades y ejercicios: el moderador deberá diseñar ejercicios y actividades que permitan la construcción de entregables intermedios que contribuyen al resultado final. Es importante tener en cuenta el nivel de detalle utilizado que será apropiado para cada ambiente. Lo más importante de un diagrama es que pueda ser entendido por los usuarios.
- Preparar, informar y educar a los participantes: todos lo participantes deben estar concientes de los objetivos y limitaciones del proyecto y de los entregables esperados del *workshop*.
- Coordinar la logística del *workshop*: los *workshop* deben ser llevados adelante fuera de la empresa para poder evitar interrupciones. Se deben preparar proyectores, pantallas, computadoras, mesas, fibras, entre otros. Esta tarea es responsabilidad del moderador.

También JAD implica una serie de actividades a llevar a cabo post-*workshop*:

- Luego de haber realizado el *workshop*, es importante haber obtenido una respuesta sobre cada problemas expuesto.
- El moderador y el experto en documentación trabajan en conjunto para finalizar la documentación del *workshop*. El gerente de proyecto es el que recibirá esta documentación.

- La documentación circula por toda la organización para continuar alistando la ayuda y las aprobaciones para el desarrollo se proyectan en caso de necesidad.
- El diseño se mueve hacia adelante para la inclusión de un pedido, para la adquisición del software de uso o hacia un prototipo o una fase de la generación del código. Puede contener los detalles tales como disposiciones y menús de pantalla. El modelo de los datos contendrá volúmenes y capacidades. El modelo de proceso especificará volúmenes de la transacción. Si el diseño se convierte en prototipo, deberían organizarse una serie de *workshops* de uno o dos días para evaluar y validar dicho prototipo.

Algunos de los beneficios que provee JAD son [Jennerich:1990]:

- Incentiva al consenso y a la propiedad: el enfoque de *workshop* rápidamente alcanzará consenso y compromiso entre los usuarios.
- Mejora la calidad del diseño: el *workshop* mejora la calidad de la fase de diseño ya que fuerza una definición de ese entregable en forma temprana. Durante el *workshop* los participantes están enfocados en una meta común. Los usuarios del *workshop* poseerán un mejor entendimiento de los problemas del sistema y del volumen de trabajo que deberá ser realizado.
- Los equipos de proyecto se mantienen focalizados: en el *workshop* los participantes desarrollarán una visión y un lenguaje común del proyecto para la discusión de los problemas. Estos elementos permanecerán con el equipo durante la vida del proyecto.
- Reducción en los costos del ciclo de vida: en 1989 Capers Jones, un experto en la productividad de la industria de la computación, estudió 60 proyectos de desarrollo, y encontró que sin JAD el 35% de la funcionalidad a ser desarrollada se perdió y que esa pérdida tuvo un impacto en por lo menos el 50% del código. Con JAD menos del 10% de la funcionalidad se perdió y eso tuvo un mínimo impacto en el código.

6. Prototipación

La prototipación es el proceso de creación de un sistema que ilustra las características relevantes del mismo. Comienza por el estudio preliminar de los requerimientos del usuario y concluye con una serie de requerimientos formulados y un prototipo desechable. Es un proceso iterativo el cual permite la definición del comportamiento del sistema, y en particular la apariencia del mismo. Esta técnica es realmente útil cuando el prototipo puede ser construido rápidamente [Tuffley:2005].

Un prototipo de requerimientos de software es una implementación parcial del sistema, se construye para ayudar a los desarrolladores, usuarios y clientes en la

obtención de un mejor entendimiento del sistema, en especial de los requerimientos que están menos claros [Madigan].

En particular, uno puede sugerir que no se deberían construir prototipos a menos que realmente se piense que la construcción será rápida. Por otro lado, la prototipación debe realizarse solo cuando existe confianza mutua con el usuario [Davis:2003]. La prototipación es una herramienta valiosa en la clarificación de requerimientos. Pueden actuar de manera similar en escenarios, al proveer a los usuarios un contexto en el cual se los puede entender mejor sobre la información que proveen. Existe una gran gama de técnicas de prototipado, desde borradores en papel de diseños de pantallas hasta versiones beta de los productos de software. Existe un fuerte solapamiento de su uso para la elicitación de requerimientos respecto a la validación de requerimientos [SWEBOK:2004].

La prototipación es un medio común para la validación de la interpretación de los ingenieros de software de los requerimientos de software, así como también para la elicitación de los mismos. Dentro del proceso de elicitación de requerimientos, existe una amplia variedad de técnicas de prototipado. La ventaja de los prototipos es que hacen más fácil la interpretación de las suposiciones de los ingenieros de software, y pueden poner en evidencia el porqué de equivocaciones. Por ejemplo, el comportamiento dinámico de una interfaz de usuario puede ser mejor entendida a través de un prototipo animado que a través de una descripción textual o modelos gráficos. También existen desventajas, dentro de las que se pueden nombrar: el peligro de que la atención del usuario se aparte de la funcionalidad principal por problemas cosméticos o problemas de calidad del prototipo. Una desventaja es que los prototipos pueden ser costosos de desarrollar. Sin embargo, si se evita el gasto incurrido al intentar satisfacer requisitos erróneos, su costo puede ser justificado muy fácilmente [13; 10]

La prototipación ha sido muy útil para la elicitación de requerimientos, donde generalmente existe una gran cantidad de incertidumbre, o donde existe la necesidad de obtener una retroalimentación de los *stakeholders*. La prototipación usualmente se combina con otras técnicas, como por ejemplo al utilizar un prototipo para provocar una discusión sobre una técnica de elicitación en grupo, o como base para una encuesta [Nuseibeh].

7. Win Win

Es una técnica de elicitación de requerimientos basada en la recolección a través de una herramienta de soporte. Esta técnica da resultados positivos tanto al desarrollador como al usuario y ayuda en la determinación de prioridades [Tuffley:2005].

EsTablance objetivos y procedimientos para los equipos integrados de productos - qué es lo que deben tratar de hacer los participantes? Cómo deberían proceder? Cómo sabrán cuando hayan terminado?

La condición fundamental de éxito de esta técnica es, que el proyecto será exitoso si y solo si se crean “ganadores” con cada uno de los *stakeholders* [Sommerville:1997]. Las principales características documentadas de la técnica *Win-Win* son:

- Acentúa la necesidad de conseguir un aporte de cada una de las partes implicadas.
- La aceptación del sistema es una resolución muy probable.
- Se facilita la resolución de conflictos a nivel social.
- La evolución de los requerimientos es activa.

Los pasos a seguir para llevar adelante esta técnica son los siguientes [Klier]:

- Identificar los *stakeholders* que sean críticos para el éxito del proyecto.
- Identificar las condiciones de ganancia de los *stakeholders*, las cuales representan los requerimientos específicos de sistema.
- Los ingenieros de requerimientos identifican problemas de conflicto en las condiciones de ganancia, y guían los *stakeholders* involucrados durante la negociación de compromiso. Las condiciones de ganancia no conflictivas y el compromiso de los *stakeholders* son propuestas como una especificación de sistema.
- Negociar los acuerdos de *Win-Win* de alto nivel.
 - Inventar opciones para la ganancia mutua (Fig. 7)

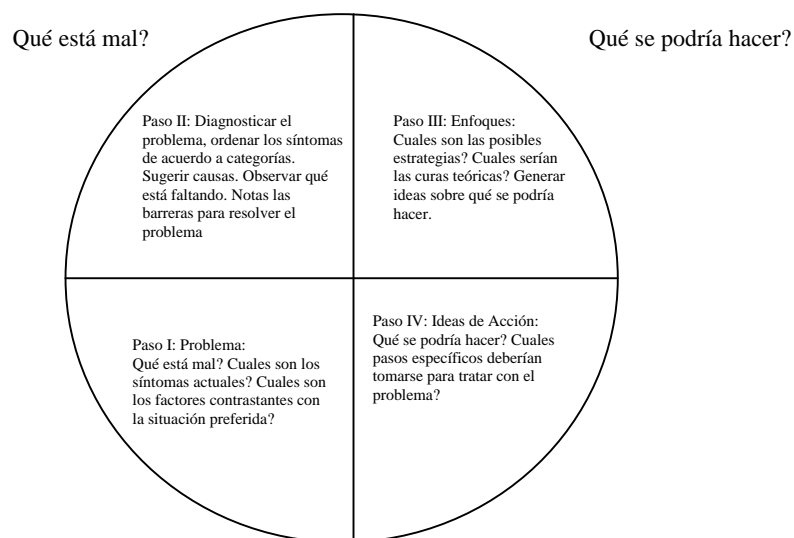


Figura 7: Win-Win Pasos a seguir [Klier]

- Manejar las expectativas.
- Agrupar los acuerdos de *Win-Win*.
- Desarrollar los puntos anteriores hasta que el producto esté desarrollado completamente.

- Confrontar y resolver los riesgos de las situaciones de *win-lose* y *lose-lose*.

Dentro de las negociaciones *Win-Win* existen una serie de situaciones que se pueden presentar dependiendo, de las habilidades del desarrollador para elicitar correctamente los requerimientos y de la capacidad del usuario de expresar correctamente sus necesidades. A continuación se presentan las posibles situaciones que pueden darse en una negociación *Win-Win* [Boehm] (Fig. 8).

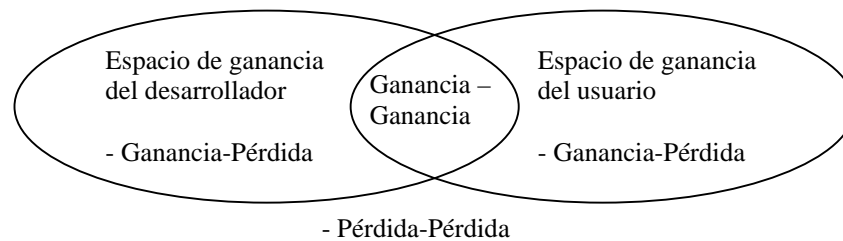


Figura 8: WinWin – Situaciones posibles [Boehm]

Como muestra la Fig. 8 la situación ideal, tanto para los desarrolladores como para el cliente, es la circunferencia del medio en donde se unen las necesidades de ambas partes, es decir, cada una de las partes logra satisfacer sus propias necesidades sin afectar a la otra parte.

A continuación la Fig. 9 muestra el ciclo de vida de un ciclo de negociaciones *Win-Win* definido:

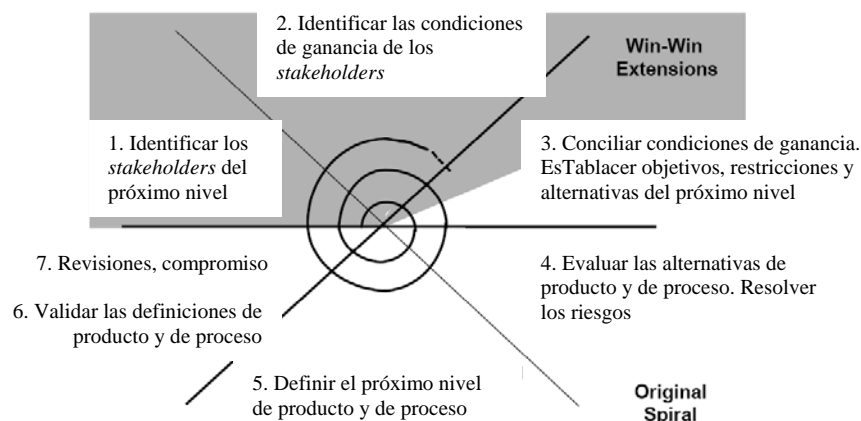


Figura 9: Win Win – Modelo en espiral [Boehm]

Se reconocen así siete actividades importantes: 1) identificar los *stakeholders*, 2) identificar las condiciones de ganancia de los *stakeholders*, 3) conciliar las condiciones de ganancia, esTablacer los objetivos, restricciones y alternativas del próximo nivel, 4) evaluar las alternativas de proceso y producto. Resolver los

riesgos, 5) definir el nuevo nivel de proceso y producto, incluyendo las partes, 6) validar las definiciones de proceso y producto y por último 7) revisar y obtener compromiso. Como lo muestra la figura es un ciclo iterativo, en el cual, cuando se termina de ejecutar la última acción se comienza nuevamente otro ciclo.

Este enfoque *Win-Win* puede mostrarse como una técnica de Ingeniería de Requerimientos desde un enfoque de arquitectura. Los *stakeholders* utilizan un grupo de ayuda del sistema para identificar condiciones de ganancia, y negociar acuerdos *win-win*. Dentro de las actividades de esta técnica podemos mencionar la realización de cronogramas para condiciones de ganancia, conflictos y acuerdos; la definición de categorías de dominio y un glosario; análisis de los conflictos de las condiciones de ganancia, generando opciones de solución y la coordinación del progreso.

Dentro de los beneficios que ofrece *Win-Win* [Boehm] se mencionan los siguientes:

- Mantiene a los *stakeholders* involucrados con el proceso de elicitación de requerimientos.
- Provee una guía colaborativa operacional.
- Provee un criterio para la evaluación del éxito.
- Reduce el ciclo de vida del proyecto - especialmente para las colaboraciones distribuidas.

8. Observación

En un estudio realizado [Davis:2003], cinco de nueve expertos resaltaron a las técnicas etnográficas de observación como extremadamente efectivas para la elicitación de requerimientos. En particular, tres implicaron que la observación de los usuarios debería realizarse siempre y cuando esté disponible y exista un sistema. La observación es una buena forma de recolectar requerimientos cuando los usuarios están demasiado ocupados para involucrarse con las entrevistas, sesiones en grupos o encuestas. Uno utiliza la observación para evaluar las relaciones políticas y de poder mientras se trabaja en organizaciones nuevas. Muchos analistas reconocen que los *stakeholders* deberían ser observados siempre que sea posible [Davis:2003].

La técnica de observación posee un enfoque en el cual la base es que el mundo social está ordenado. Este orden social puede no ser obvio, y difícil de describir por el sentido común. No puede asumirse que el orden social posee una estructura a priori. El orden social es logrado sobre una base de momento a momento a través de acciones colectivas de los participantes (en vez de a través de cualquier estructura preexistente). El orden social será solo observable en el caso que el observador se sumerja en él. La observación debe realizarse en un ambiente natural [Easterbrook:2000-2004].

Es un método en el cual el observador tiende a volverse parte de la comunidad de interés, a través del desarrollo de un rol legítimo dentro de la misma

[Goguen:1993]. La asunción que los métodos sociales de la ciencia se diferencian de aquellos utilizados por las personas estudiadas es desafiada por la metodología de la étnica (*ethnomethodology*), que discute que los científicos sociales empleen las mismas clases de actividades que hacen al sentido como miembros de la cultura estudiada. Esto se contrapone con la objetividad científica, o por lo menos, contra la demanda que los analistas tengan de un acceso único a la objetividad [Goguen:1993].

La importancia del contexto del software dentro del ambiente organizacional ha llevado a la adaptación de técnicas de observación para la elicitación de requerimientos. Los ingenieros de software aprenden sobre las tareas del usuario a través de la inmersión dentro del ambiente laboral y la observación de cómo los usuarios interactúan con el software y entre ellos. Estas técnicas son relativamente costosas, pero son instructivas ya que ilustran que muchas de las tareas y procesos de negocio de los usuarios son demasiado sutiles y complejas como para describirlas fácilmente [SWEBOK:2004].

Esta técnica solo se debería utilizar cuando existe la necesidad de obtener la magnitud total de proceso a ser resuelto o para verificar una hipótesis de un enfoque. Si la tarea a ser observada es rutinaria para un experto, entonces el experto se sentirá librado a dar explicaciones más detalladas de su enfoque y así el observador obtendrá un conocimiento más significativo. Si el problema es difícil, entonces el experto puede tener problemas para alcanzar una solución. Con la ayuda de un experto que colabore, estas situaciones pueden proveer una gran cantidad de información, en donde varias alternativas de solución pueden ser exploradas [Kennisacquisite].

El observador pasa tiempo con el sujeto a observar, hasta poder convertirse en un miembro del grupo. Y por lo tanto poder realizar estudios longitudinales apropiados [Easterbrook:2000-2004].

Muchos de los métodos convencionales asumen la preexistencia de categorías, esto puede guiar mal al observador [Easterbrook:2000-2004]. La técnica de observación tiene la característica de ser discreta, y participativa. Se usa para la demostración de tareas [Sindre:2005]. Este método es contextualizado y revela detalle que otros métodos no logran revelar [Easterbrook:2000-2004].

La observación es una técnica que sirve para la elicitación del conocimiento tácito de una situación actual. Otras de las ventajas que presenta esta técnica es que representa una carga horaria mínima para los usuarios, ya que ellos sólo deben realizar su trabajo de manera normal y natural [Sindre:2005].

La observación permite a la ingeniería de requerimientos comprobar los procesos que actualmente se están ejecutando. Las respuestas de los *stakeholders* ayudan a corroborar la factibilidad de los posibles cambios [Green:2003].

Es un método que consume demasiado tiempo de los desarrolladores. Tiene como resultado una imagen rica que es difícil de analizar. No se puede decir demasiado sobre los resultados de las propuestas de cambio [Easterbrook:2000-

2004]. No existe la objetividad científica, por lo tanto se utiliza la teoría de mediciones propia del observador [Easterbrook:2000-2004].

Pasos a seguir en la observación [Green:2003]

- Se observa al *stakeholders* ejecutar su tarea.
 - Se toma nota cada vez que una acción es ejecuta.
- Se realizan preguntas para clarificar la naturaleza del trabajo y para elicitación un rango completo del trabajo.
- Tomar nota de las sesiones de observación.
- Analizar las notas de la sesión:
 - Procesos actuales
 - Cambios posibles, factibles, y deseables al proceso
 - Implicaciones de los requerimientos para un sistema basado en computadora

En la Fig. 10 se presenta un cuadro comparativo de algunas de las técnicas de elicitación de requerimientos teniendo en cuenta, el objetivo de la elicitación [Sindre:2005].

TDT4175 - Information Systems, Spring 2005

Techniques vs elicitation need (Lauesen, 2002)

What to elicit?	S	I	O	T	D	Q	B	F	W	w	P	p	c	A	N	R	C	G	d
Present work																			
Present problems																			
Goals, key issues																			
Future sys ideas																			
Realistic possib.																			
Consequences, risks																			
Commitment																			
Conflict resolution																			
Requirements																			
Priorities																			
Completeness																			

The darker, the better. For list of techniques, see next slide

10

TOT4175 - Information Systems, Spring 2005

List of techniques for table previous slide

S = Stakeholder analysis	P = Prototyping
I = Interview	p = Pilot experiment
O = Observation	c = similar Companies
T = Task demo	A = Ask suppliers
D = Document analysis	N = negotiation
Q = Questionnaires	R = risk analysis
B = Brainstorm	C = Cost/benefit
F = Focus groups	G = Goal-domain analysis
W = Domain workshop	d = domain-requirement analysis
w = Design workshop	

11

Figura 10: Técnicas vs. necesidades de elicitación [Sindre:2005]

Métricas de software

El proceso de desarrollo de software puede verse beneficiado por el uso de métricas apropiadas. Estas provienen de un proceso de medición que lleva a cuantificar el proceso. La medición es el proceso por el cual se asigna una serie de números o símbolos a uno o más atributos de las entidades o procesos bajo consideración; esta asignación se realiza de acuerdo a una serie de reglas definidas y claramente identificadas. Una vez que la asociación evento-valor ha sido realizada, es posible comparar las entidades de interés a través de la comparación de estos valores [Definición].

La medida es un factor clave en el desarrollo del software ya que: ayuda a entender qué está pasando durante el desarrollo y el mantenimiento, permite controlar el desarrollo del proyecto, y estimula la mejora de procesos y productos.

Las mediciones representan una escala, a través de la cuales la madurez del objeto o del proceso es cuantificada [Fred:1979]. No se podrá llegar a la etapa de predicción sino se pasa por la etapa de cuantificación.

“No se puede controlar lo que no se puede medir y no se puede predecir ni controlar lo que no se puede medir” [Fenton:1998]

De acuerdo a Fenton y Pfleeger [Fenton:1998], la medición a nivel del proceso de desarrollo de software puede realizarse a través de 1) Métricas o mediciones directas, o 2) Métricas indirectas. Las directas emergen cuando se mide exactamente el objeto o atributo de la entidad que se quiere medir. Ejemplos de mediciones directas de software son: las líneas de código (LOC = *lines of code*) producidas, los errores en el código fuente, el tamaño de la memoria, la velocidad de ejecución, etc. Para obtener métricas directas existen ciertas propiedades [Kitchenham:1995] que deben ser satisfechas:

1. El atributo a ser medido debe permitir que las entidades sean distinguibles unas de otras
2. Una medición válida debe obedecer a la condición de representación.
3. Cada unidad de un atributo que contribuya a una medida válida, es equivalente
4. Diferentes entidades pueden poseer el mismo valor de atributo (dentro de los límites de los errores de mediciones).

Por el contrario las métricas indirectas para un atributo de una entidad se conforman involucrando otros atributos o incluso otras entidades y no a partir de los valores observados directamente sobre la entidad.

Los atributos a medir pueden ser internos o externos. Los atributos internos son los que pueden ser medidos a través del examen de la entidad,

independientemente de su comportamiento. Los atributos externos son medidos por medio de otros atributos. De acuerdo con [Runeson:2000], los atributos externos son en mayor medida mediciones indirectas y los atributos internos son mediciones directas.

En el área de ingeniería de software, el proceso de medición en sí mismo sigue bajo investigación. A pesar de unos 30 años de investigación y experiencia existe poca claridad respecto a las métricas más apropiadas para las distintas etapas de la vida del software, en especial sobre aquellas referidas a requerimientos los antecedentes son escasos. La mayoría de las métricas propuestas surgieron de la preocupación de las instituciones por la calidad del software; para ello necesitan tener un control preciso, predecible y repetible del proceso de desarrollo y por tanto necesitan cuantificarlo. De la gran cantidad de métricas propuestas es difícil saber cuál o cuáles representan con mayor exactitud aquellos atributos que pretende cuantificar y cuáles son las que se aceptan más universalmente. La validez (falta de sesgo) de una métrica asegura que el valor esperado de la misma esté representada exactamente la cualidad que se pretendió cuantificar [Horst:1995].

Algunas de las ventajas de la medición en el software son [Loconsole;2003]:

- Incremento en el entendimiento y control del proceso de desarrollo de software
- Incremento en la capacidad de mejora del proceso de desarrollo de software
- Estimaciones más exactas del costo y cronograma del proyecto de software
- Mayor evaluación objetiva de cambios en técnicas, herramientas y métodos
- Estimaciones más exactas del esfuerzo con respecto a cambios en el costo y cronograma del proyecto
- Decremento de los costos de desarrollo debido al incremento en la productividad y eficiencia
- Mejora en la satisfacción y confianza del cliente debido a mayor calidad en la productividad
- Menor costo en las actividades de mediciones. La cantidad de tiempo invertido por un equipo de desarrollo de software en un programa de mediciones es casi el 20% de su tiempo, reducido en un 1% una vez que el equipo adquiere experiencia.

Las mediciones pueden también ocasionar problemas como son aquellos de falta de honestidad en las mediciones, miedo o resistencia a las mediciones y uso ético de las métricas ya que éstas muchas veces se asocian con problemas sociales sobre los que los ingenieros de software están poco acostumbrados a conducir. Vale también notar que muchas veces el énfasis será puesto en aquello que se

está midiendo y lo que no se está midiendo será ignorado gradualmente. Para un programa exitoso de mediciones siempre es necesario prever un feedback de sesiones con quienes recolectan datos y con quienes los analizan y comunican. El análisis del costo-beneficio del proceso de medición es realmente necesario para la evaluación del proceso.

Pocos antecedentes de métricas existen que traten directamente la cuantificación de atributos del proceso de elicitación de requerimientos. La herramienta automática conocida como “Automated Requirement Measurement (ARM)” , desarrollada por el Centro Tecnológico de Aseguramiento de Software (Software Assurance Technology Center (SATC)) de la NASA Goddard Space Flight Center como herramienta temprana del ciclo de vida y especialmente para la administración de requerimientos, ha sido usada para determinar los requisitos que se especifican para un desarrollo en lenguaje natural. El objetivo de la ARM es proveer métricas que puedan ser utilizadas por el líder de proyecto para evaluar la calidad del documento de requerimientos. La herramienta no tiene la intención de evaluar la exactitud de los requerimientos especificados, sino que es una ayuda para “escribir los requerimientos correctamente” (i.e. no para “escribir los requerimientos correctos”). La herramienta ARM busca en cada línea del documento de requerimientos palabras específicas y frases que el SATC ha identificado como indicadores de calidad. Al utilizar estos indicadores, la herramienta crea un archivo que incluye tres reportes. Este archivo contiene un “Reporte Resumen”, el cual incluye la cantidad total de veces que cada indicador de calidad aparece en el documento de requerimientos; un “Reporte Detallado”, el cual especifica la localización dentro del archivo fuente de cada declaración de la especificación identificada por la herramienta y una copia de la declaración de la especificación y un reporte débil de la frase el cual lista la locación y las especificaciones que contengan los indicadores que han sido considerados ser frases que debilitan la especificación [ARM:1998].

La importancia de la documentación correcta de los requerimientos ha causado que la industria del software produzca un número significativo de ayudas para la creación y administración de los documentos de especificación de requerimientos y para la creación de declaraciones individuales de requerimientos. Sin embargo muy pocas de estas ayudas asisten en la evaluación de la calidad del proceso de elicitación de requerimientos o de las mismas declaraciones individuales de estos [Rosenberg:1998]. En particular la aplicación de la herramienta ARM sobre documentos de requerimientos ha permitido desarrollar las siguientes métricas:

1. Líneas de texto-tamaño del documento
2. Imperativos – palabras y frases que ordenan que algo debe ser hecho o ser provisto. El número de imperativos se utiliza como una cuenta de la cantidad de requerimientos.

3. Continuaciones – frases que siguen un imperativo e introduce la especificación de los requerimientos a un nivel más detallado, para una cuenta suplementaria de requerimientos.
4. Directivas – referencias provistas a figuras, tablas y notas
5. Frases débiles – cláusulas que son aptas para causar incertidumbre y dejar lugar a múltiples interpretaciones.
6. Incompletitud – declaraciones dentro del documento que tienen la palabra TBD (To Be Determined – a ser determinado) o TBS (To Be Supplied – a ser provisto)
7. Opciones – palabras que parecen darle a los desarrolladores alternativas para satisfacer las especificaciones para que pueden ser ambiguos.

De esta manera la herramienta al evalúa la declaración de especificación individual y el vocabulario utilizado para declarar requerimientos como así también la estructura de los documentos de requerimientos [Rosenberg:1996].

Capítulo 3: Materiales y métodos

Técnicas de elicitación

Para lograr mayor entendimiento del estado del arte en nuestro medio, con respecto al proceso de elicitación de requerimientos y particularmente de la técnicas más utilizadas para los distintos tipos de software se realizaron dos encuestas (Anexos I y II), una destinada a desarrolladores de software y otra destinada a usuarios de software.

Procedimientos:

- 1) Determinación del objetivo, debido a que las encuestas fueron destinadas a dos poblaciones distintas, fue necesario la definición clara del objetivo de cada tipo de encuesta en particular.
- 2) Definición de los criterios de inclusión de los encuestados, todos los encuestados fueron argentinos; para la encuesta a desarrolladores se consideraron encuestados que alguna vez desarrollaron software o que actualmente lo están haciendo, y para la encuesta a usuarios no se determinaron *a priori* criterios de inclusión pero se registraron las siguientes variables para ser usadas en una posible post-estratificación de respuestas: edad, sexo, cantidad de software aproximado que utilizó durante toda su vida y la posición (estudiante universitario de grado, estudiante de post grado, gerente de empresas, empleado de empresas que utilizan software y profesionales varios).
- 3) Distribución, ambas encuestas fueron enviadas a distintas empresas del medio, profesores y alumnos de cursos de grado en universidades de Córdoba (Universidad Tecnológica Nacional, Universidad Blas Pascal, Universidad Católica de Córdoba, Nacional de Córdoba), y cursos de post grado (Universidad Blas Pascal).
- 4) Diseño de la encuesta, ambas encuestas fueron prediseñadas bajo el marco teórico de la investigación realizado sobre el proceso de ingeniería de requerimiento, proceso de elicitación de requerimientos y técnicas de elicitación.
- 5) Prototipación de las encuestas, las encuestas fueron prototipadas y puestas a pruebas mediante la colaboración de personas idóneas conocedoras del tema. De esta prototipación y control surgieron algunas correcciones o ajuste fundamentales para lograr que las encuestas fuesen contestadas en un lapso de tiempo no mayor a 20 minutos.
- 6) Tiempo de exposición en el medio, las encuestas se enviaron el día 1 de Junio del 2005 y se mantuvo la recepción de las mismas hasta el 30 de Septiembre del 2005, es decir, 4 meses de disponibilidad. Las encuestas fueron subidas a una página de Internet, de manera tal que cualquier persona pudo tener acceso directo a las mismas, la dirección para la encuesta a desarrolladores es: <http://www.hemocor.com.ar/tesisnati/index.asp> y para usuarios: <http://www.hemocor.com.ar/tesisnati/encuestaRequerimientosusuarios/index.asp>.

Estructura de la encuesta para desarrolladores

La encuesta fue estructurada en 6 secciones.

1. La primera sección “Introducción” presenta el objetivo de la encuesta, explicando cuáles son las metas generales y particulares de la misma
2. La segunda sección “Datos Generales” tiene como objetivo introducir al encuestado en la selección de la alternativa de software (a medida o empaquetado) sobre el cual deberá responder la encuesta, haciendo alusión a preguntas específicas de las características particulares del software seleccionado, como ser año de desarrollo, cantidad de personas involucradas, nivel de experiencia, tecnología utilizada, etc.
3. La tercera sección “Con respecto a los requerimientos” hace referencia a prácticas específicas, de la alternativa de software seleccionado, de la forma en que el proceso de elicitación de requerimientos fue llevado a cabo.
4. La cuarta sección “Con respecto a la forma en la que desarrolló software” hace referencia a prácticas específicas de la forma en que se llevó a cabo el proceso de desarrollo del software.
5. La quinta sección “Con respecto a las técnicas” tiene como objetivo indagar sobre las prácticas desarrolladas según la técnica de elicitación seleccionada. Esta sección se divide en nueve sub-secciones, en donde la primera trata sobre prácticas genéricas de todas las técnicas que se evalúan en esta encuesta, y las siguientes sub-secciones hacen referencia a prácticas tenidas en cuenta para cada técnica en particular, entrevistas, tormenta de ideas, prototipación, observación, encuestas, JAD, RAD y Win Win.
6. La última sección de la encuesta “Sugerencias y observaciones” permite a la persona encuestada emitir cualquier opinión relacionada con la encuesta y/o realizar alguna aclaración que considere necesaria.

Estructura de la encuesta para usuarios

La encuesta fue estructurada en 5 secciones.

1. La primera sección “Introducción” presenta el objetivo de la encuesta, explicando cuáles son las metas generales y particulares de la misma.
2. La segunda sección “Datos Generales” tiene como objetivo obtener información específica del perfil del usuario que está contestando la encuesta, indagando sobre sus datos personales, género, cantidad de software a medida y empaquetado que utilizó en su vida aproximadamente y la posición que ocupa dentro de una serie de categoría preseleccionadas.
3. La tercera sección “Software Empaquetado” hace referencia a la preferencia del usuario por el software empaquetado y del servicio que se le proporcionó en el momento de la adquisición del mismo; haciendo mención a las técnicas utilizadas por los proveedores del software a la hora de elicitar los requerimientos.

4. La cuarta sección “Software a Medida” hace referencia a la preferencia del usuario por el software a medida y del servicio que se le proporcionó en el momento de la adquisición del mismo; haciendo mención a las técnicas utilizadas por los proveedores del software a la hora de elicitar los requerimientos.
5. La última sección de la encuesta “Sugerencias y observaciones” permite a la persona encuestada emitir cualquier opinión relacionada con la encuesta y realizar alguna aclaración que considere necesaria.

Operalización de las variables

Los datos obtenidos como respuesta a las preguntas cerradas de la encuesta fueron volcados en una matriz de dimensiones $n \times p$, donde n es el número de encuestados y p el número de variables.

Se trabajó con un total de 147 variables para la encuesta a desarrolladores y 68 variables para la encuesta a usuarios. El total de desarrolladores encuestados fue 55 y el total de usuarios encuestados fue 40.

Las respuestas se codificaron usando variables de tres tipos:

- Cuantitativas: como año de desarrollo, edad, cantidad de software a medida utilizado, etc.
- Cualitativas - Categorizadas ordinales: estas variables surgieron de preguntas con múltiples modalidades o categorías excluyentes y siempre se asignó el valor 1 a las categorías de mayor orden. Por ejemplo el nivel de experiencia: 1=Muy Alto, 2=Alto, 3=Medio, 4=Bajo, 5=Muy Bajo.
- Cualitativas - Binarias: estas variables surgieron de preguntas con múltiples modalidades de respuestas NO excluyentes. Por ejemplo: causas de desvío en el tiempo de desarrollo real y planeado cuyas modalidades eran: no se relevaron requerimientos, los requerimientos relevados no eran claros, los requerimientos relevados eran ambiguos, los requerimientos relevados eran inconsistentes entre si, hubo muchos cambios en el negocio y no se manejaron bien esos cambios, etc. Un encuestado podría seleccionar más de una opción y por ello se creó una variable binaria para cada modalidad (1: modalidad seleccionada, 0: modalidad no seleccionada).
- Las respuestas a preguntas abiertas fueron anexadas al registro de cada encuestado para ser utilizadas en un análisis complementario al análisis estadístico cuando fuese necesario.

Métodos de análisis

En una primera etapa cada tipo de encuesta fue analizada en forma separada y finalmente se realizaron análisis comparativos. Para cada tipo de encuesta se realizó una estadística descriptiva por variable, usando promedios, valores mínimos y máximos para las variables cuantitativas y cálculos de frecuencia,

gráficos de sectores y barras apiladas para las variables cualitativas. Se construyeron tablas de contingencia de dos vías (tablas de clasificación cruzada) y se utilizó el estadístico X^2 de Pearson (Agresti, 2000) para realizar pruebas de hipótesis sobre la información volcada en estas tablas ya sea concluir sobre la presencia/ausencia de asociaciones entre pares de variables como para comparar proporciones entre los tipos de software para las variables de interés. Todas las pruebas fueron realizadas con un nivel de significación del 5%.

Además se construyeron gráficos BIPLOT con las coordenadas repuntos filas y columnas de todas las tablas de contingencia obtenidas mediante la técnica de análisis multivariado de correspondencias múltiples (Balzarini, 2003). El análisis multivariado fue utilizado para evidenciar asociaciones entre modalidades de las distintas variables en estudio y para caracterizar el contexto de elicitación de requerimiento para cada uno de los tipos de software si es que estos muestran diferencias significativas. Todos los análisis fueron realizados con el software estadístico INFOSTAT [ARM:1998].

Métricas

Culminado el proceso de análisis de las encuestas se planificó una entrevista que fue realizada a 40 desarrolladores, 12 de software empaquetado y 28 de software a medida, todos residentes en la Ciudad de Córdoba, Argentina. El instrumento diseñado para realizar la entrevista se presenta en el Anexo VII. Un 65% de los desarrolladores trabajaban para empresas de desarrollo de software y el resto desarrollan software de manera independiente. La entrevista se realizó con el fin de recabar información precisa sobre cantidades asociadas al proceso de elicitación que podrían servir como métricas directas o para derivar métricas indirectas. En la Tabla 1 se lista las métricas propuestas y una breve descripción de su significado. Además de recabar información sobre las métricas propuestas, se solicitó a cada uno de los entrevistados que expresara una cantidad que, a su juicio, podría ser usada como una métrica para el proceso de elicitación.

Se solicitó además que asignaran un orden de mérito para 5 métricas que fueron seleccionadas como principales candidatas dentro de aquellas propuestas en esta tesis. Para cada una de éstas se preguntó además si se consideraba práctica o impracticable de obtener en futuros desarrollos.

La información relevada fue codificada según una escala cuantitativa ordinal para cada variable y volcada en una tabla de datos que fue procesada mediante análisis de varianza [ARM:1998], haciendo uso del conocimiento a priori del grupo de pertenencia de cada entrevistado (desarrollador de software a medida o empaquetado). También se realizaron análisis de varianza para cada una de las variables clasificando las observaciones según los niveles de satisfacción tanto del cliente como del desarrollador respecto al producto entregado; se concibió a la satisfacción del cliente como una medida resumen o síntesis del proceso de desarrollo y por tanto una medida del proceso de elicitación.

Capítulo 4: Resultados

Parte I: Estadística descriptiva para la totalidad de encuestados (sin considerar contexto definido por la alternativa de software)

Parte I – 1: Encuestas a desarrolladores de software

En esta sección se presentan los resultados obtenidos de la encuesta realizada a desarrolladores de software, tanto empaquetado como a medida.

Para esta encuesta se tuvieron en cuenta personas que alguna vez desarrollaron alguna alternativa de software o que actualmente está desarrollando uno, ya sea a medida o empaquetado, no se hizo distinción alguna con respecto a la profesión, puesto en el que trabaja, edad o sexo.

Esta encuesta fue realizada desde el mes de Junio hasta el mes de Septiembre inclusive del año 2005. Un total de 55 desarrolladores pudieron contestarla de los cuales sólo el 16% de los mismos desarrollaron en algún momento un software empaquetado (Fig. 11) mientras que el 86% de los encuestados desarrolló software a medida.

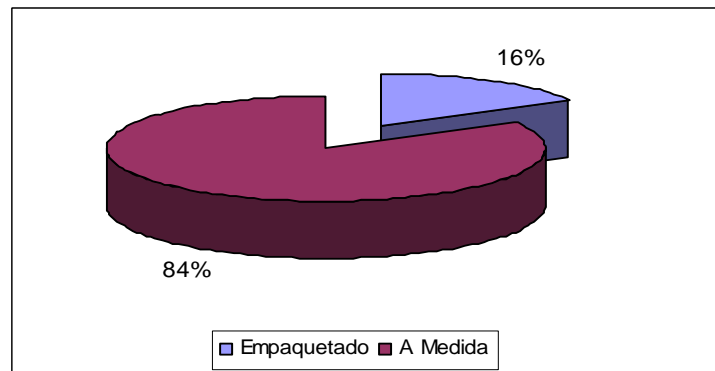


Figura 11: Tipos de software

Sólo el 24% de los encuestados desarrolló un software que no presentó desviaciones en las estimaciones de tiempo y el tiempo real de desarrollo. Lo que sugiere que la mayoría de los proyectos que se desarrollan actualmente poseen frecuentes desviaciones (Fig. 12). La Figura se construyó dividiendo las encuestas según el proyecto implicase un desvío mayor al 100% de lo estimado; entre el 50% y el 100%, entre el 20% y el 50% o un desvío menor a un 20% de lo planificado.

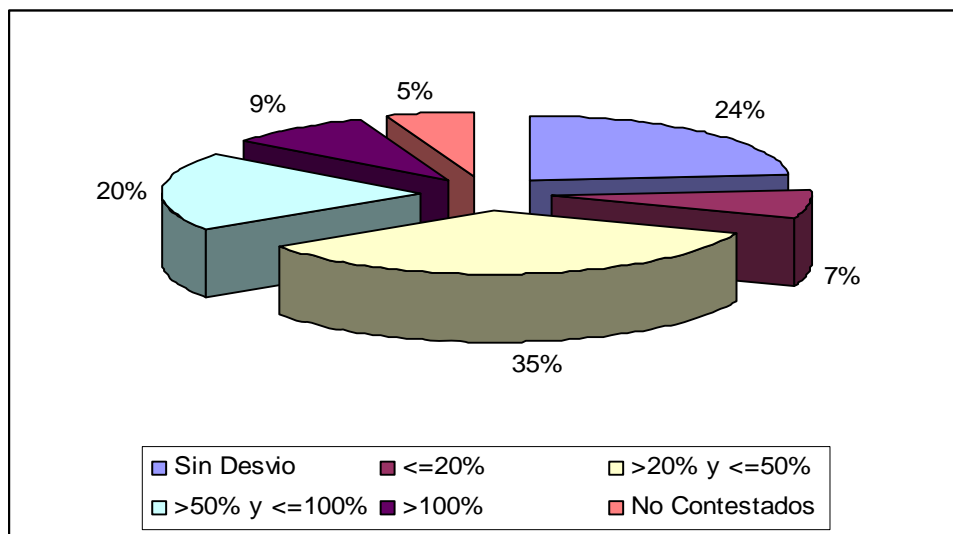


Figura 12: Tiempo estimado de desarrollo vs. tiempo real de desarrollo

Teniendo en cuenta las características propias de los proyectos seleccionados para contestar la encuesta, la Fig. 13 muestra que el 49% los encuestados considera que el nivel de experiencia de las personas asignadas al mismo es medio, el 16% considera que el nivel de experiencia es bajo y el 31% considera que el nivel de experiencia es alto.

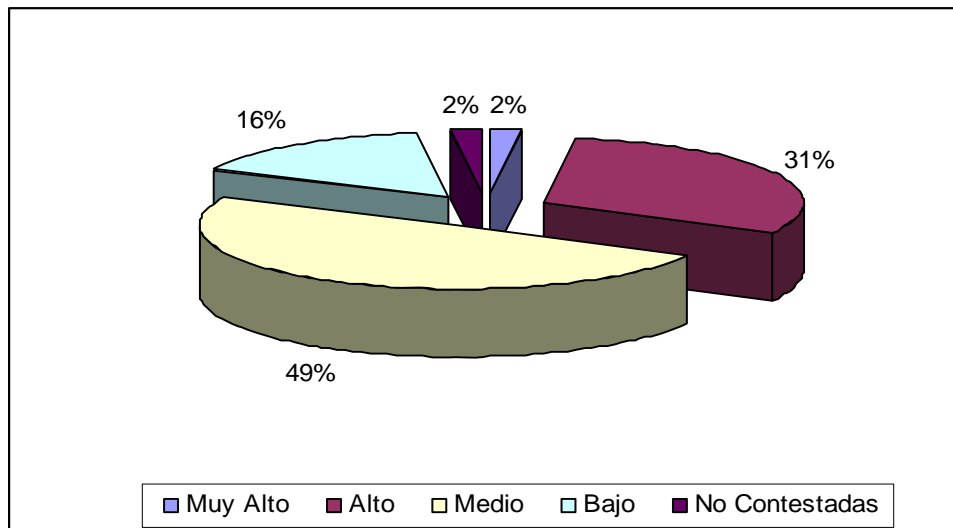


Figura 13: Nivel de experiencia de las personas que participaron

En la Fig. 14 se muestra las distintas causas que fueron identificadas como posible responsables del desvío entre estos tiempos (Tiempo estimado de desarrollo vs tiempo real de desarrollo). Las causas de desvío señaladas con un cuadrado (símbolo ■) son aquellas propias de la etapa de requerimientos. Los resultados sugieren que las causas con mayor frecuencia son las pertenecientes a esta etapa, es decir, 2 - Los requerimientos relevados no eran claros y 9 - Cambios en los requerimientos; además más de una causa referida a requerimientos

pueden presentarse, confirmando la importancia que la literatura revisada cita sobre la elicitación de requerimientos.

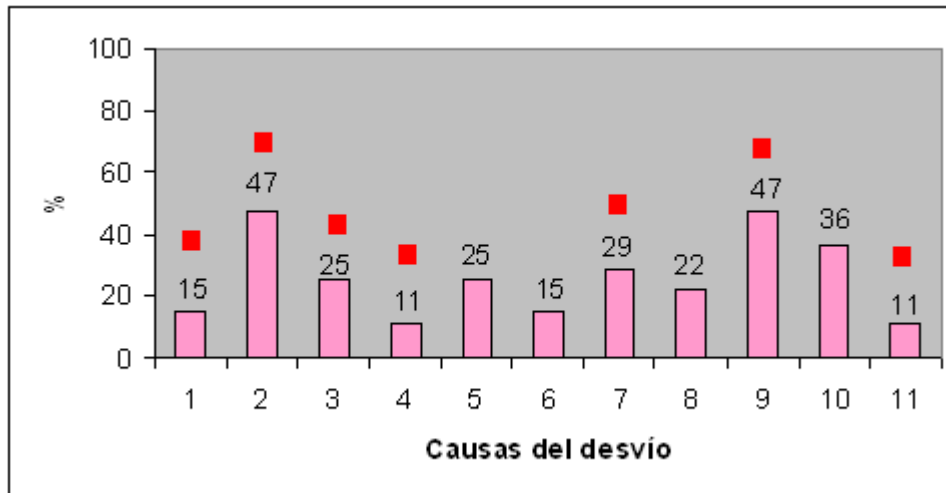


Figura 14: Causas del desvío.

Referencias: 1 - No se relevaron los requerimientos, 2 - Los requerimientos relevados no eran claros, 3 - Los requerimientos relevados eran ambiguos, 4 - Los requerimientos relevados eran inconsistentes entre si, 5 - Hubo muchos cambios en el negocio del usuario y no se manejaron bien esos cambios, 6 - No se realizó diseño, 7 - No se entendieron los requerimientos del usuario, 8 - El equipo desarrollador no tenía experiencia en la tecnología utilizada, 9 - Cambios en los requerimientos, 10 - No se diseñaron casos de test, 11 - Los requerimientos no eran verificables.

La Fig. 15 muestra que el 11% de los desarrolladores consideran que el grado de satisfacción del cliente respecto del software entregado es muy alto, el 56% de los encuestados respondió que es alto, y sólo el 2% de los mismos consideraron que el grado de satisfacción fue muy bajo.

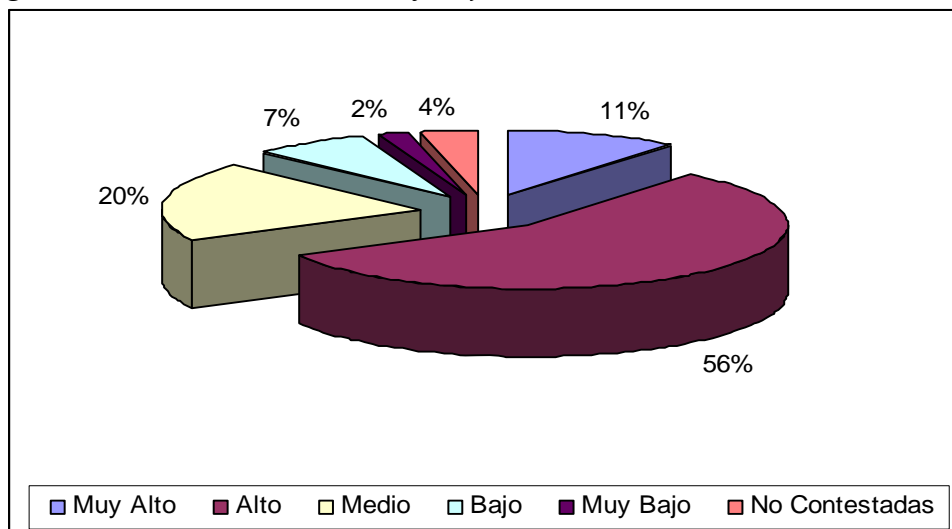


Figura 15: Grado de satisfacción del cliente respecto del software desarrollado

Los resultados indican que la percepción del desarrollador sobre la calidad del producto entregado, es en general buena y por tanto no conciben que la satisfacción del cliente pueda ser gravemente afectada por las demoras ocasionadas ya que éstas se produjeron en el 74% de los proyectos (Fig. 12).

Sin embargo, si se observa la Fig. 16, que muestra que se considera en general que el impacto que la satisfacción del cliente posee en la empresa es alto (33%) o muy alto (43%)

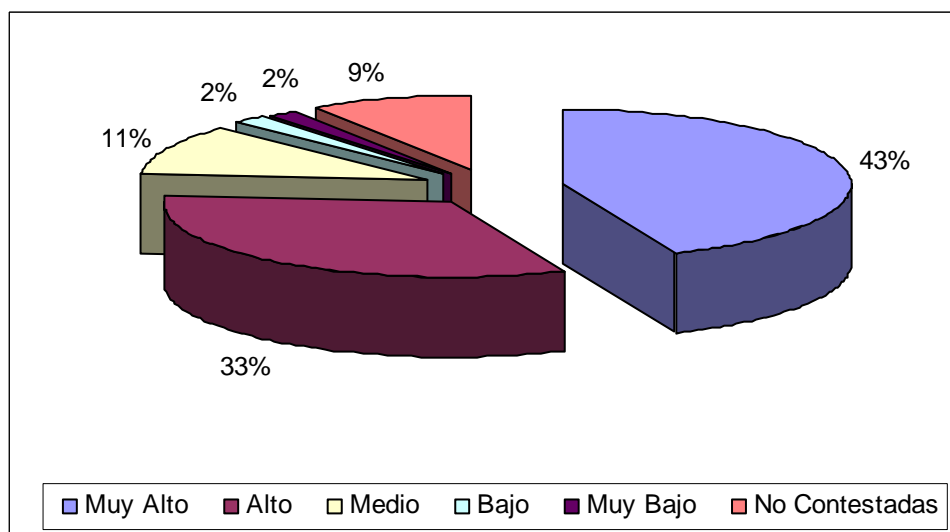


Figura 16: Impacto del grado de satisfacción del usuario en la empresa

Dentro del ciclo de vida tradicional de desarrollo (Requerimientos, Análisis, Diseño, Codificación, Testing) en la Fig. 17 vemos que más del 84% de los encuestados planifica y realiza la etapa de Requerimientos, más del 71% de los mismos realiza la etapa de Análisis, más del 76% la etapa de Diseño, más del 84% la etapa de Codificación y sólo el 60% de los encuestados realizan la etapa de Testing. Cabe aclarar que ninguno de los encuestados respondió afirmativamente a la falta de realización de alguna de estas etapas, es decir, que al menos una de ellas es realizada a la hora del desarrollo del software.

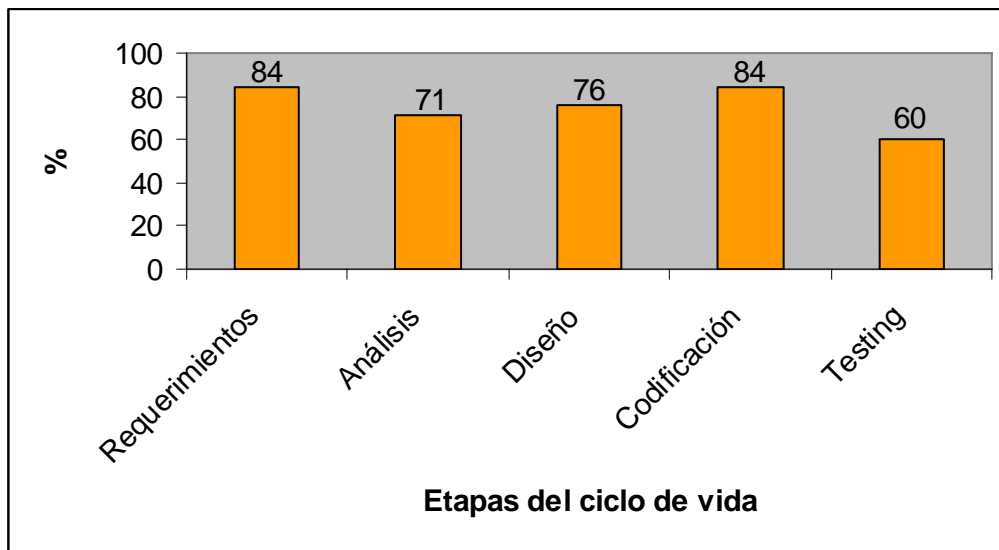


Figura 17: Etapas del ciclo de vida del software tenidas en cuenta a la hora del desarrollo

La Fig. 18 muestra la importancia de la etapa de requerimientos según la percepción de los desarrolladores, el 86% de los encuestados respondió que la misma es Muy Importante; el 5% respondió que era Importante y ninguno de los encuestados respondió que era Poco Importante o que No tiene sentido realizarla pero un 9% no contestó la pregunta.

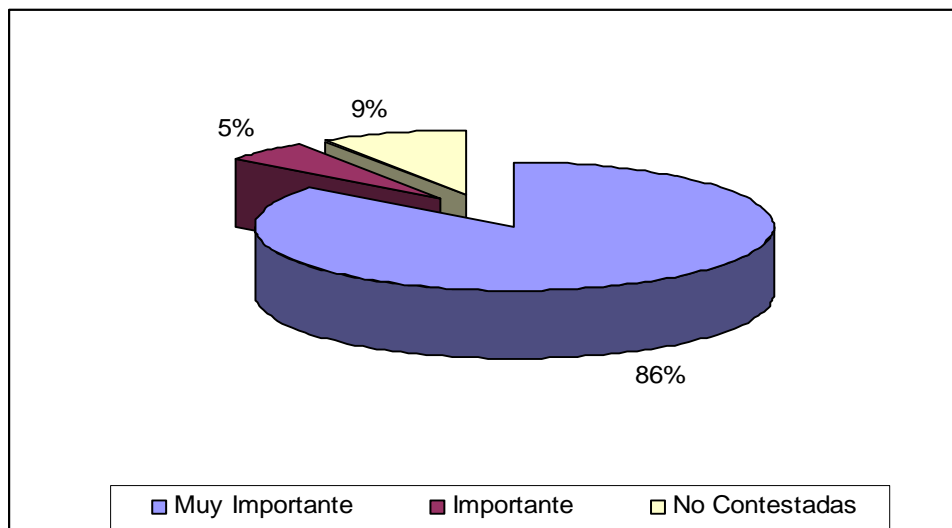


Figura 18: Importancia de la etapa de requerimientos según la percepción de los desarrolladores

Si se consideran las prácticas más utilizadas y recomendadas [Sommerville:1997] se puede concluir que un importante porcentaje de los encuestados respondió de manera afirmativa en la realización de las mismas. Dentro de las prácticas recomendadas se tuvieron en cuenta las siguientes (Anexo IV):

- “Se tienen en cuenta la factibilidad del sistema”, en donde el 62,75% de los encuestados respondió afirmativamente; el 13,73% respondió negativamente y el 23,53% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se tienen en cuenta consideraciones políticas y organizacionales”, en donde el 68,63% respondió afirmativamente; el 13,73% respondió negativamente y el 17,65% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se registra del origen de los requerimientos”, en donde el 64,71% respondió afirmativamente; el 19,61% respondió negativamente y el 15,69% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se tienen en cuenta las restricciones del ambiente”, en donde el 80,39% de los encuestados respondió afirmativamente; el 11,76% respondió negativamente y solo el 7,84 % respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se tienen en cuenta el registro de la razón o importancia de los requerimientos”, en donde el 62% de los encuestados respondió afirmativamente; el 26% respondió negativamente y el 12% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se recolectan requerimientos desde múltiples puntos de vista”, en donde el 30% de los encuestados respondió afirmativamente; el 48% respondió negativamente y el 22% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se reutilizan requerimientos”, esta demostró ser una de las prácticas menos aplicadas y utilizadas ya que solo el 44,9% de los encuestados la realiza; el 38,78% no las utiliza y el 16,33% no tiene conocimientos de la aplicación de la misma.
- “Se identifican las clases de clientes”, en donde el 67,39% de los encuestados respondió afirmativamente; el 21,74% respondió negativamente y el 10,87% respondió que no poseía conocimientos de la aplicación de esta práctica.
- “Se planifican las actividades”, en donde el 84,78% respondió que se afirmativamente y solo el 15,22% respondió negativamente, demostrando ser esta una de las prácticas más utilizadas por los desarrolladores de software.
- “Se realiza un resumen luego de cada actividad”, en donde el 57,78% respondió afirmativamente; el 40% respondió negativamente y el 2,22% respondió que no poseía conocimientos de la aplicación de esta práctica.

Ya que la etapa de requerimientos es considerada una de las más importantes dentro del ciclo de vida del software, y teniendo en cuenta que la asignación de

las personas que establecen los requerimientos es un paso importante dentro de esta etapa, se decidió indagar sobre la manera en que esta asignación era realizada; dentro de las respuestas obtenidas la Fig. 19 muestra que se encontró que casi un 70% de los encuestados respondió que las mismas eran asignadas teniendo en cuenta el rol que poseían dentro de la empresa, un 2% respondió que la asignación se realizaba teniendo en cuenta la antigüedad de la persona en la empresa o al azar; un 36% respondió que estas personas eran asignadas teniendo en cuenta la experiencia, y solo un 5% respondió que las personas eran asignadas para establecer los requerimientos de acuerdo a otro criterio de selección.

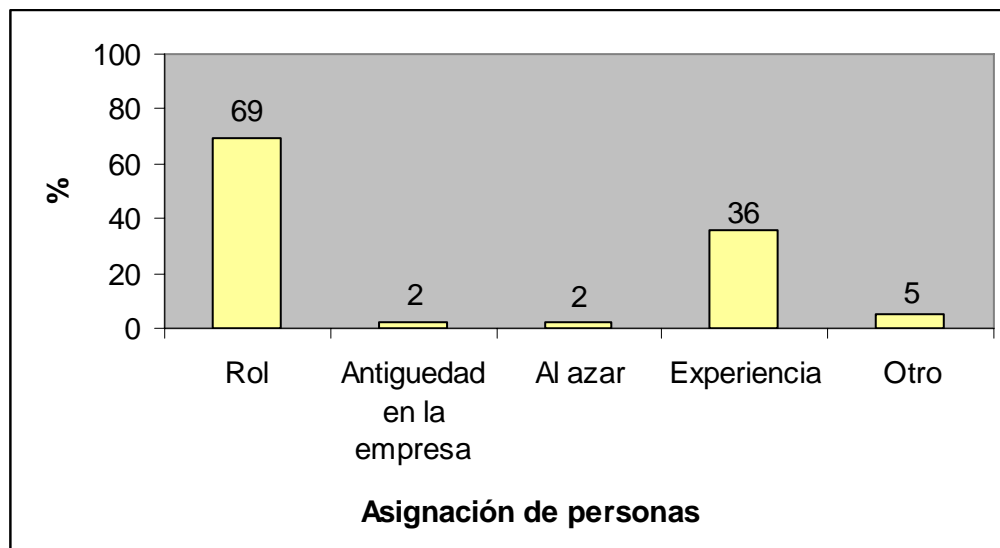


Figura 19: Asignación de personas para la elicitación de requerimientos

La Fig. 20 muestra el origen de la idea de la realización del proyecto el 87% de los encuestados respondió por petición del cliente (categoría 1), el 22% respondió que gracias a un estudio de mercado (categoría 2), y el 57% respondió que gracias a una necesidad interna, ya sea de la Gerencia (42% - categoría 3) o de los propios desarrolladores (16% categoría 4), y sólo el 2% respondió que era gracias a otro factor influyente.

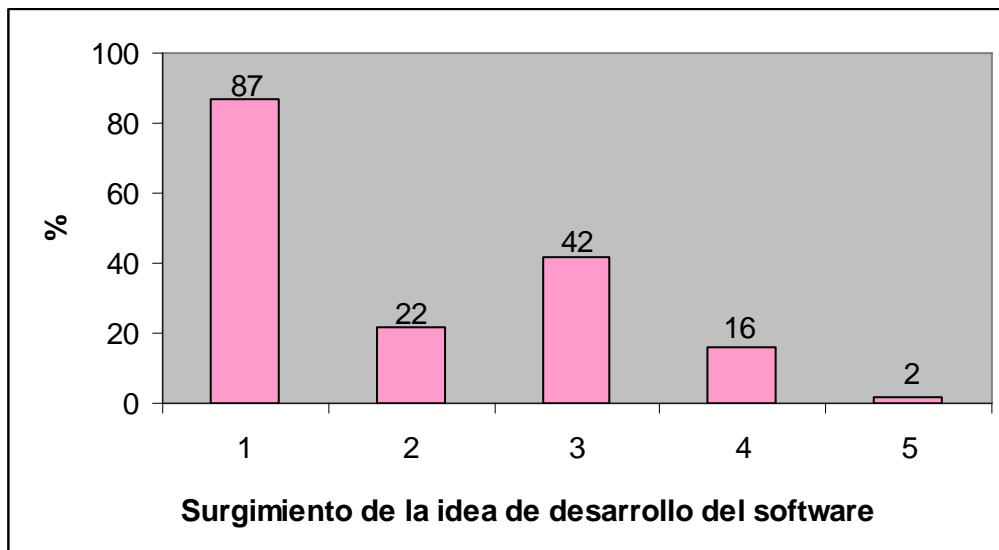


Figura 20: Surgimiento de la idea del desarrollo del software.

Referencias: 1-Petición del cliente, 2-Estudio de mercado, 3-Necesidad interna (Gerencia), 4-Necesidad interna (Desarrolladores), 5-Otra

En la Fig. 21 se observa que las técnicas menos utilizadas por los desarrolladores de nuestro medio son JAD, RAD, y Win Win, y a la misma vez las técnicas más utilizadas y conocidas por los encuestados son: Encuestas, Entrevistas, Prototipación, Tormenta de Ideas y Observación; sin embargo también existe un alto porcentaje de los encuestados que no conocen o no utilizan estas técnicas (Encuestas: 50%, Entrevistas: 62%, Prototipación: 35%, Tormenta de Ideas: 31%, Observación: 34%).

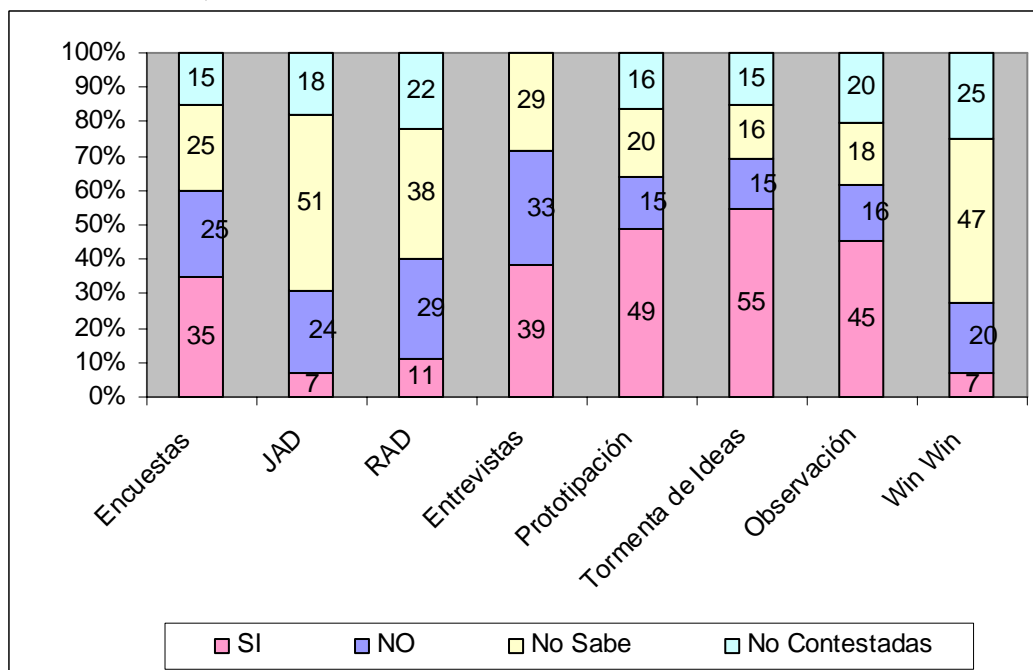


Figura 21: Técnicas de elicitación de requerimientos utilizadas.

Sin embargo teniendo un alto porcentaje de conocimiento y utilización de las técnicas más comunes, la Fig. 22 muestra que solo el 25% de estas técnicas están definidas en forma organizacional, es decir que cada proyecto define su propia técnica de elicitación a la hora de recavar información de los usuarios (62%).

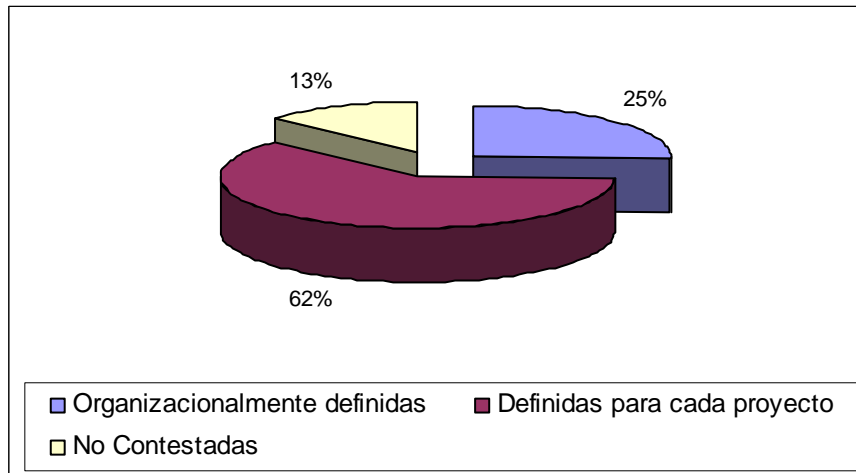


Figura 22: Definición de las técnicas de elicitación

Al momento de llevar a cabo la etapa de requerimientos y más precisamente la etapa de la elicitación de los mismo, en la Fig. 23 se observa que sólo el 20% de los encuestados respondió no haber tenido problemas para que el cliente entienda las actividades que se estaban realizando, es decir, la importancia y objetivo de dicha etapa; y el 47% respondió afirmativamente, es decir que la mayoría de las personas que llevaron a cabo la etapa de elicitación de requerimientos tuvieron o sufrieron algún tipo de problemas con el cliente.

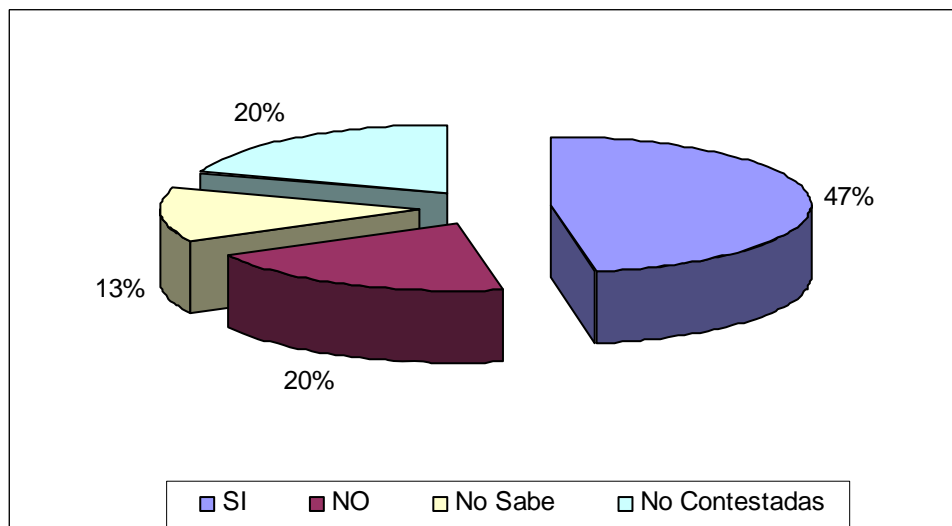


Figura 23: Problemas de entendimiento por parte del cliente sobre las actividades de elicitación

Parte I – 2: Encuestas usuarios de software

En esta sección se presentan los resultados obtenidos de la encuesta realizada a usuarios de software, tanto empaquetado como a medida.

Esta encuesta fue realizada desde el mes de Junio hasta el mes de Septiembre inclusive del año 2005. Un total de 40 usuarios respondieron la misma de los cuales, el 25% eran estudiantes universitario de grado, el 8% gerentes de empresas, el 27% profesionales independientes, el 8% son estudiantes de post-grado y el 27% empleados de empresas que utilizan alguna clase de software. Fig. 24.

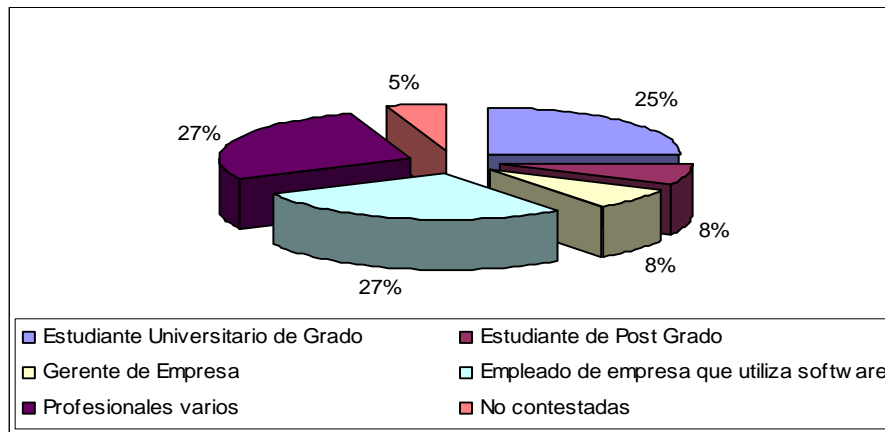


Figura 24: Categorías de usuarios

Del total de los encuestados el 60% resultaron ser de género masculino y el 40% de género femenino; un 30% dijo haber utilizado en su vida menos de 10 software empaquetados aproximadamente, un 10% respondió que utilizó más de 10 pero menos de 20, un 15% respondió que utilizó más de 20 y menos de 30 y un 30% de los encuestados respondió que en su vida utilizó más de 30 software empaquetados. Fig. 25.

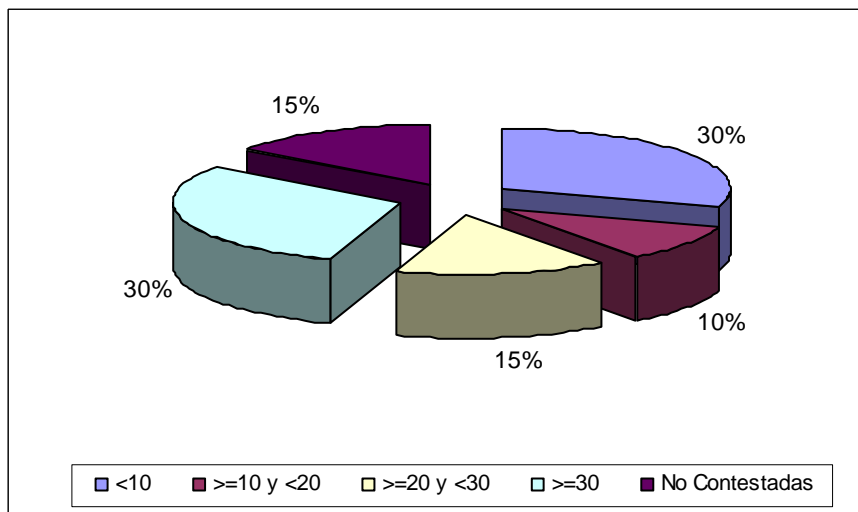


Figura 25: Cantidad de software empaquetado utilizado por los usuarios

Si tenemos en cuenta la cantidad de software a medida utilizado por los usuarios, vemos en la Fig. 26, que el 62% de los mismos utilizó menos de 10 software a medida en su vida, el 5% respondió haber utilizado más de 10 pero menos de 20, el 5% respondió haber utilizado más de 20 pero menos de 30, solo el 3% de los encuestados respondió que utilizó más de 30 software a medida en su vida, estos últimos fueron aquellos ligados a una empresa. Los resultados corroboran que la cantidad de software empaquetado utilizado es mayor que la cantidad de software a medida.

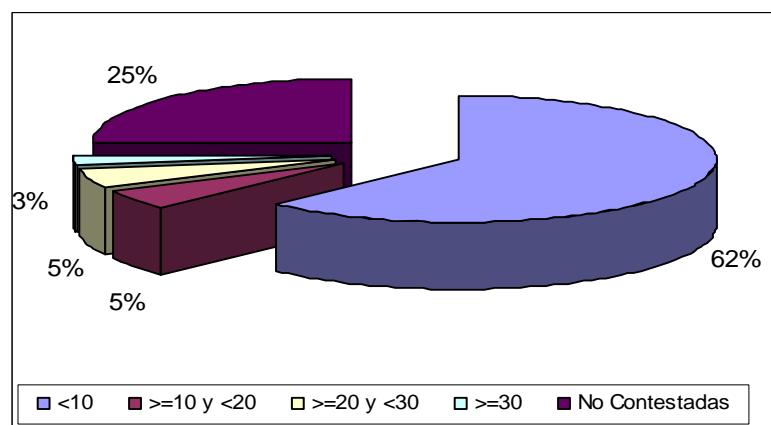


Figura 26: Cantidad de software a medida utilizado por los usuarios

Dentro de las razones expresadas por la preferencia de los usuarios por el software empaquetado, el 65% de los encuestados respondió que prefiere esta alternativa de software por el costo del mismo, es decir que el 65% de los encuestados presuponen que el costo del software empaquetado es significativamente inferior al del costo del software a medida. También existió un porcentaje relativamente alto de respuestas afirmativas para el tipo

funcionalidad (ya definida o específica) que provee el software empaquetado, ya que el 52% de los encuestados prefieren al software empaquetado porque los mismos proveen una funcionalidad ya definida. Es importante notar que el 35% de los encuestados prefieren un software a medida debido a que el mismo provee la posibilidad de definir una funcionalidad específica para la empresa. Fig. 27 y Fig. 28.

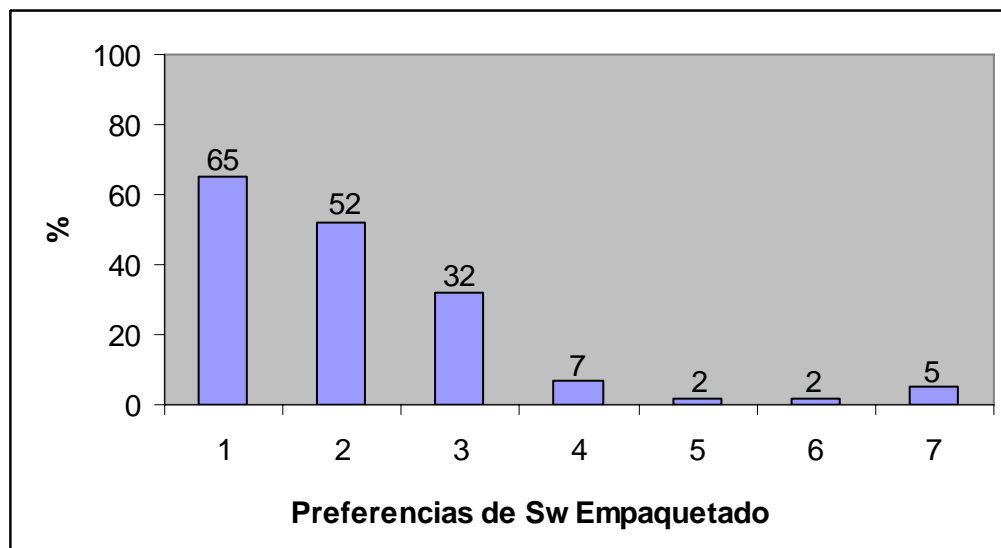


Figura 27: Razón por la preferencia de sw empaquetado.

Referencias : 1) Costos, 2) Funcionalidad ya definida, 3) Falta de tiempo para el desarrollo de un software desde 0, 4) Falta de personal para asignar a la definición del software a medida, 5) No confía en los software desarrollados a medida, y prefiere un software ya probado y testeado, 6) Nunca usó un software empaquetado, 7) Otro

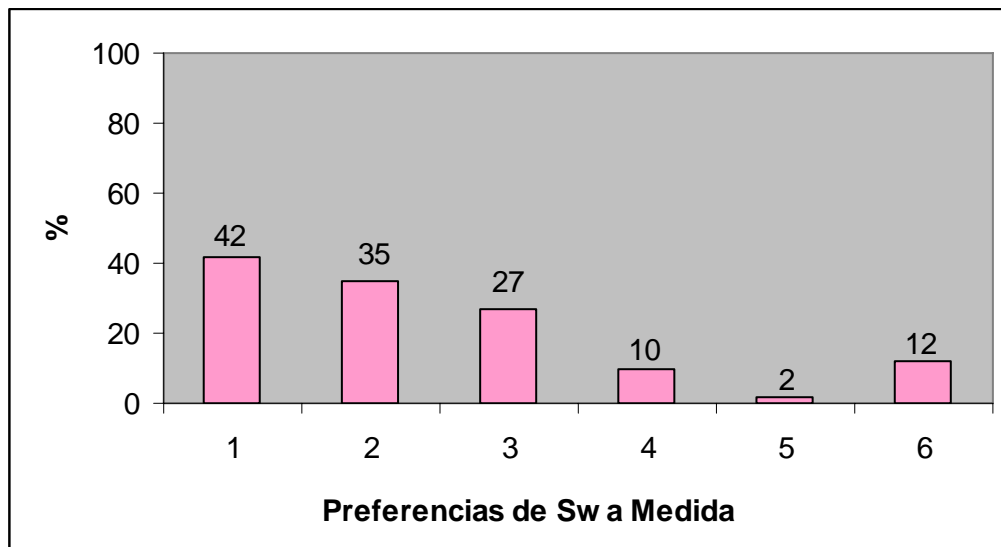


Figura 28: Razón por la preferencia de sw a Medida.

Referencias: 1) Costos, 2) Funcionalidad específica requerida, 3) No se quiere arriesgar a que el software empaquetado no funcione en su empresa, 4) No existe ningún software empaquetado útil para su empresa, 5) No confía en los software empaquetados, y prefiere definir usted la funcionalidad del mismo, 6) Otro

La Fig. 29 muestra el medio más frecuente por el cual los usuarios se enteran de la funcionalidad del software empaquetado es debido a la competencia (47%), el 20% se entera debido a publicidades, el 25% debido a recomendaciones externas y solo el 12% se entera de esta funcionalidad debido a la investigación propia.

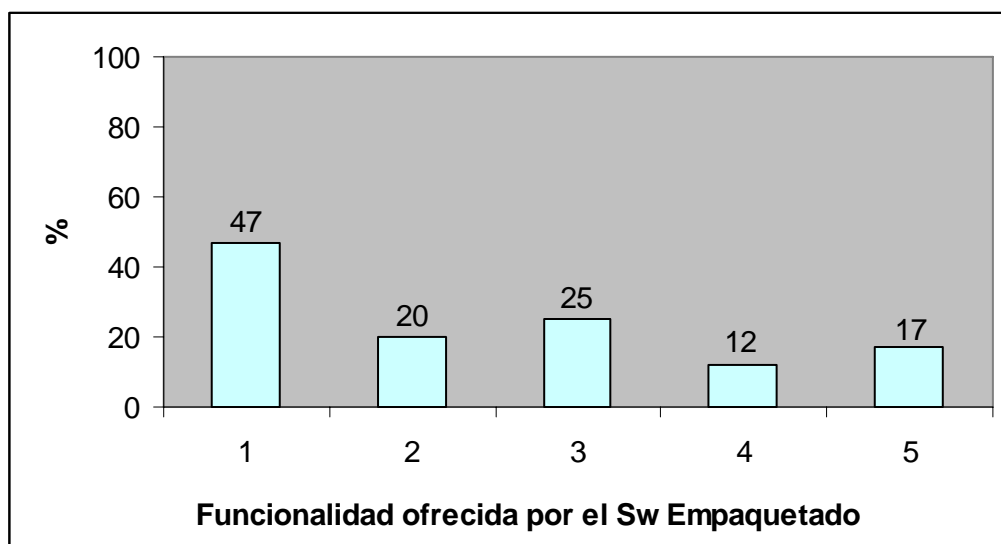


Figura 29: Via de conocimiento de la funcionalidad ofrecida por el sw empaquetado.

Referencias: 1-Competencia, 2-Publicidades, 3-Recomendaciones externas, 4-Investigación propia, 5-Otra

Teniendo en cuenta que el software empaquetado provee una funcionalidad predeterminada, definida de manera genérica para un grupo amplio de usuarios, es importante notar que los usuarios encuestados en este trabajo sienten que esta alternativa de software cumple en gran medida con sus expectativas (64%) y solo el 10% de los mismos no se encuentra satisfecho con la funcionalidad que esta alternativa brinda. Fig. 30.

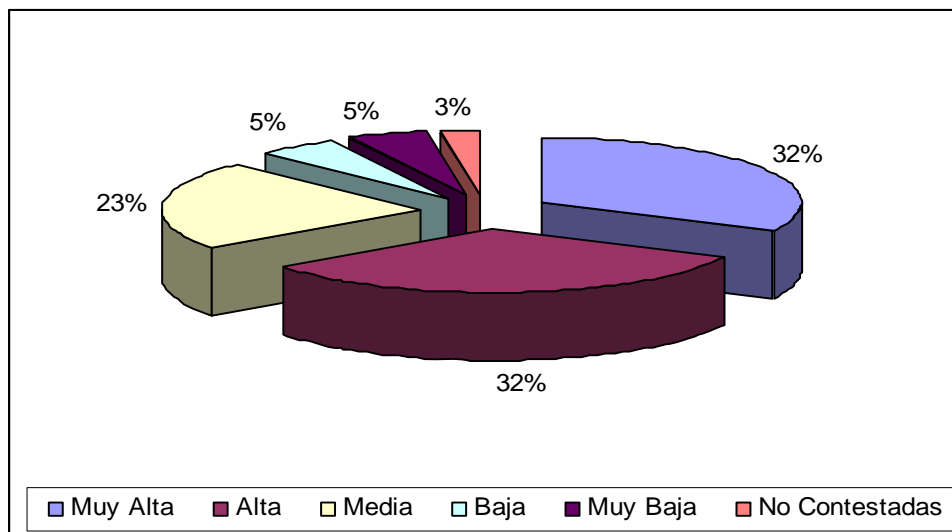


Figura 30: Medida en la que el sw cumple con las expectativas

A pesar de la alta satisfacción de los usuarios con el software empaquetado, se observó que el 52% de los encuestados respondió que el software necesitó ser adaptado a las necesidades de su empresa, y solo el 18% respondió que los cambios propuestos fueron implementados para satisfacer esas necesidades y entregados en tiempo. El 60% de los encuestados respondió que no considera que recibió un buen servicio post-venta.

Dentro de los factores atribuibles a la demora de la entrega, mostrados en la Fig. 31, aquellos referidos a requerimientos se identificaron el 30% de las veces (sumatoria de las categorías 1: Los requerimientos de cambio no fueron correctamente relevados: 10% y 2) Muchos cambios en los requerimientos: 12,5%).

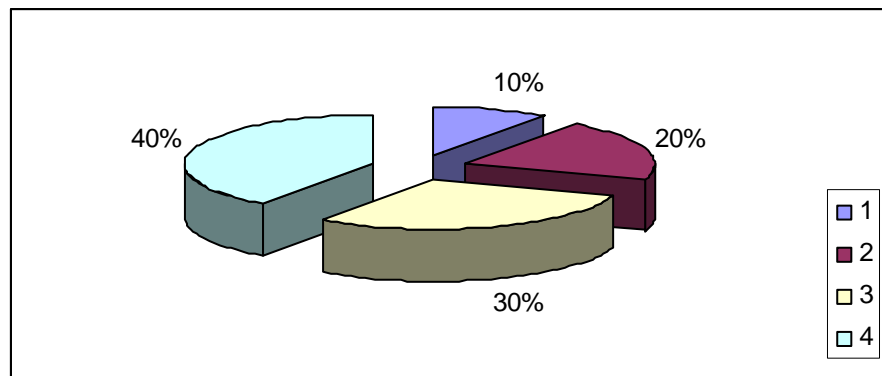


Figura 31: Factores atribuibles a la demora.

Referencias: 1) Los requerimientos de cambios no fueron correctamente relevados, 2) Muchos cambios en los requerimientos, 3) Se produjeron cambios en la tecnología utilizada en la empresa, 4) Otro

Para relevar los requerimientos de cambios, el 27% de los proveedores utilizó como técnicas de elicitación a las Entrevistas, el 27% a JAD, el 17% a las Tormentas de Ideas y el 17% a Win Win (técnicas más utilizadas). Solo el 4% utilizó a RAD, el 0% a las encuestas y el 4 tanto a la Observación como a la Prototipación. Fig. 32.

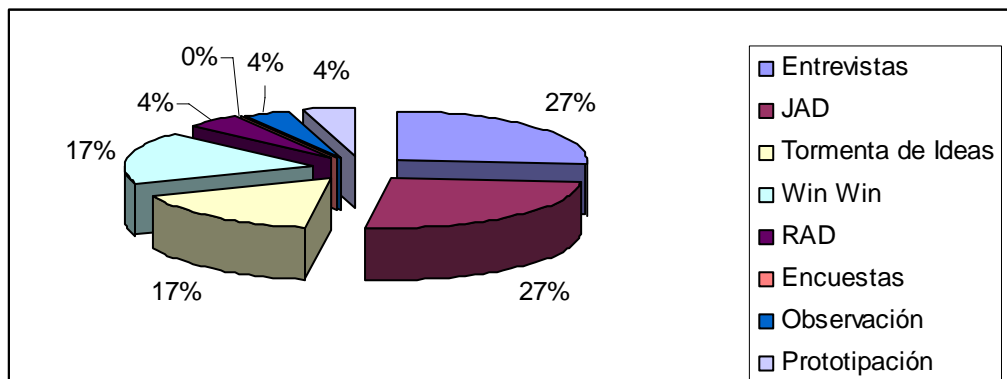


Figura 32: Técnicas de elicitación utilizadas para el relevamiento de los requerimientos de cambio en sw empaquetado

Cuando se preguntó a los usuarios si alguna vez había participado en la definición de la funcionalidad de un software empaquetado se detectó que sólo el 33% de los mismos había participado. Mientras que el 54% de los encuestados había sido parte del testing de un software empaquetado. Fig. 33 y Fig. 34.

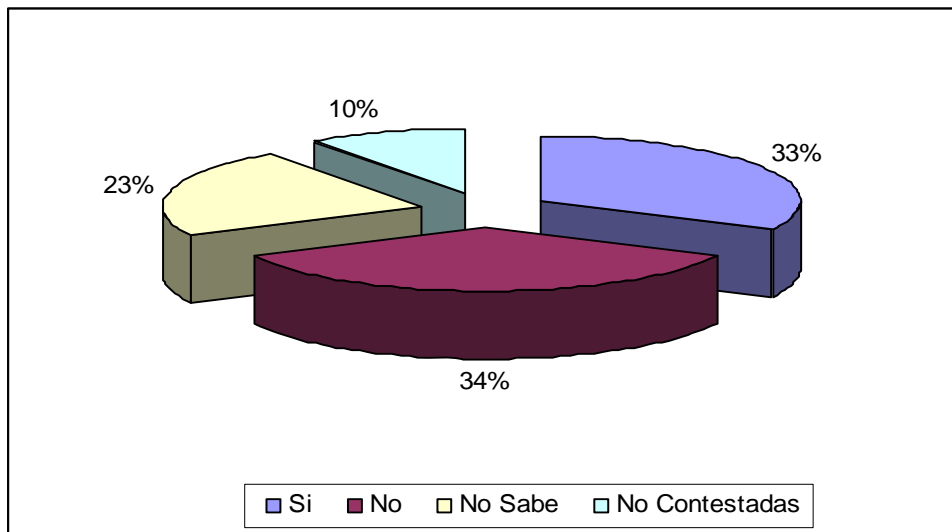


Figura 33: Nivel de participación en el desarrollo de un software empaquetado

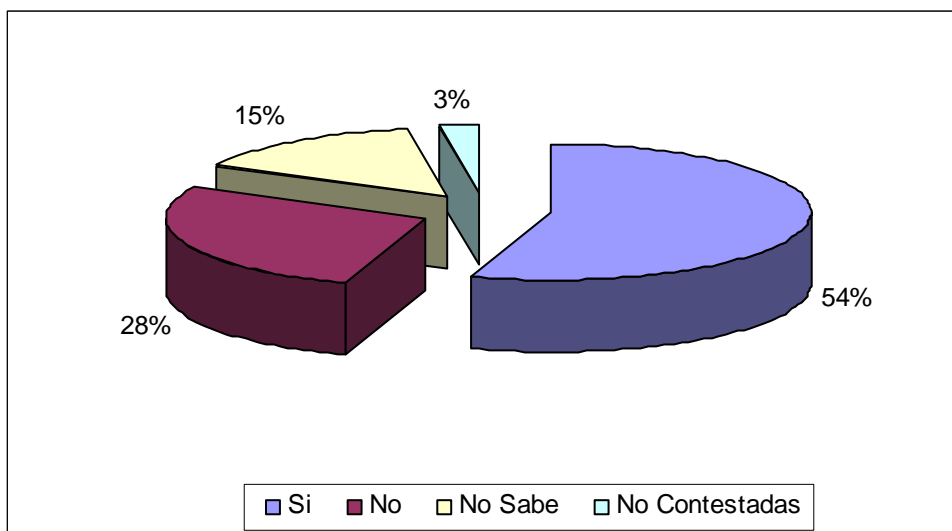


Figura 34: Nivel de participación en el testing de un software empaquetado

Respecto al software a medida, la Fig. 35 muestra la distribución porcentual de los usuarios según el nivel de uso de Software a medida. El 82% de los encuestados coinciden en que el porcentaje de utilización del software a medida que ha sido adquirido por la empresa, es menor al 30%

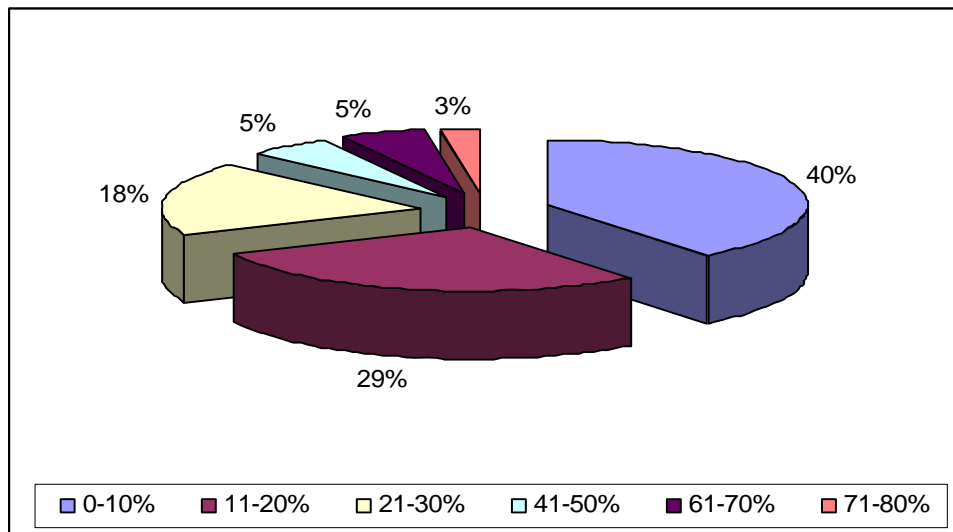


Figura 35: Porcentaje de utilización de sw a medida, según la percepción del usuario

Para el software a medida se quiso saber de donde surge la necesidad, cual es el objetivo que el mismo debe cumplir, sus alcances y límites. Los resultados sugieren que el 35% de los encuestados expresa que la necesidad surgió de la automatización de las tareas de los operarios, para hacerlas más efectivas. El 20% de los encuestados respondió que surgió gracias a asesores calificados que recomendaron el desarrollo de dicho software, y el 35% respondió que surgió gracias a potenciales clientes y el 10% respondió que la necesidad surgió debido a otros factores. Fig. 36.

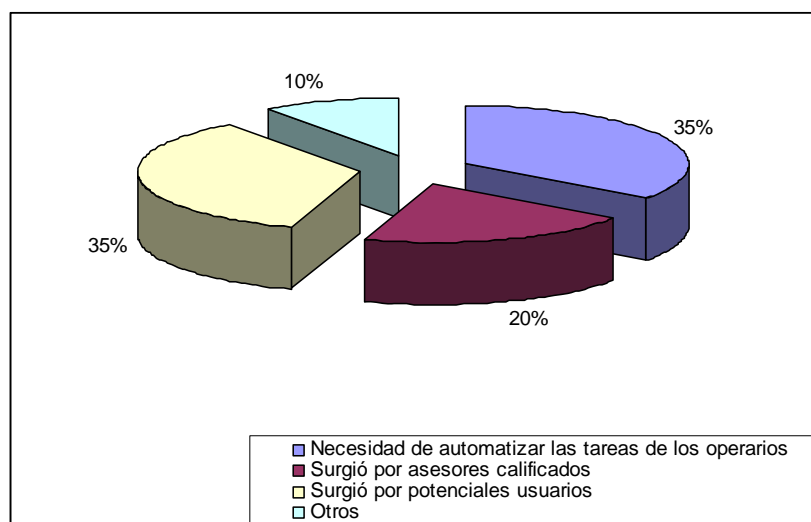


Figura 36: Definición del objetivo y el alcance del producto a medida

Al establecerse el objetivo y los alcances del sistema a desarrollar, cuando se realizó la pregunta "Se relevaron los requerimientos para el software a desarrollar?" el 52% de los encuestados respondió afirmativamente, el 38%

respondió negativamente y el 5% respondió no tener conocimientos sobre esta actividad (5% no respondieron).

Dentro de las técnicas utilizadas por el equipo desarrollador para el software a medida, la Fig. 37 muestra que el 60% de los encuestados respondió que la técnica utilizada para elicitar los requerimientos del sistema fue la Entrevista, el 32% la Tormenta de Ideas, el 20% de los encuestados respondió que la técnica de elicitación utilizada fue JAD y un 15% respondió que la técnica utilizada fue Win Win. Las Encuestas (2%), la Observación (7%), RAD (2%) y la Prototipación (2%) fueron técnicas utilizadas en menor medida para este objetivo.

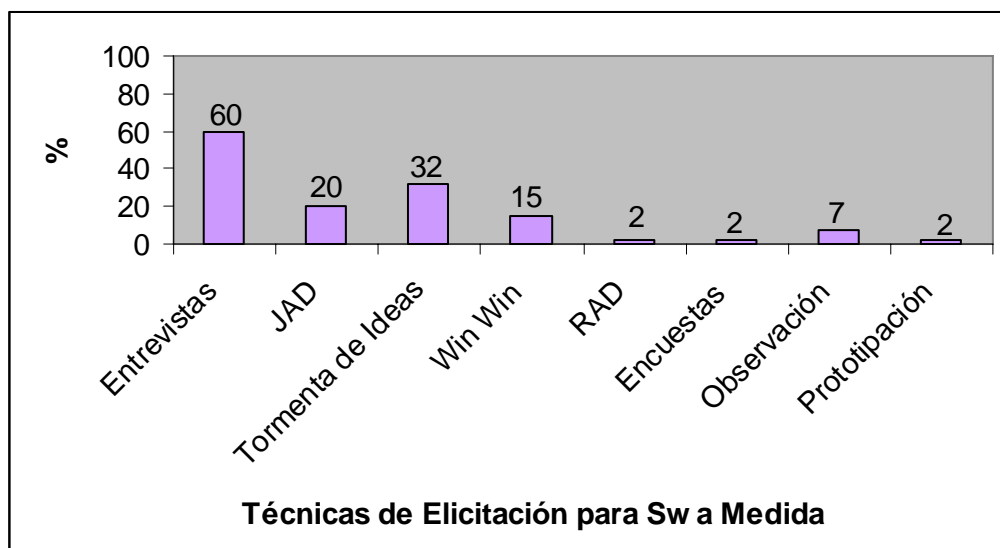


Figura 37: Técnicas de elicitación utilizadas para el relevamiento de requerimientos en sw a medida

Se presenta a continuación un análisis de “cumplimiento” de lo prometido para el software a medida. (Anexo V)

- “El software desarrollado cumple con los requerimientos establecidos?”, el 51,35% respondió afirmativamente, el 24,32% respondió negativamente, el 24,32% respondió no tener conocimiento sobre el tema y el 8% no respondió la pregunta.
- “El software se entregó en tiempo y forma según lo planeado?”, el 42,5% respondió afirmativamente, el 22,5% respondió negativamente, el 27,5% respondió no tener conocimiento sobre el tema y el 7,5% no respondió la pregunta.
- “El costo del software fue lo pactado?”, el 37,5% respondió afirmativamente, el 30% respondió negativamente, el 25% respondió no tener conocimiento sobre el tema y el 7,5% no respondió la pregunta.
- “El software debió ser corregido por fallas (el software se "clavaba")?”, el 52,5% respondió afirmativamente, el 20% respondió negativamente, el

22,5% respondió no tener conocimiento sobre el tema y el 5% no respondió la pregunta.

- “El software debió ser corregido debido a una funcionalidad mal definida?” , el 40% respondió afirmativamente, el 30% respondió negativamente, el 25% respondió no tener conocimiento sobre el tema y el 5% no respondió la pregunta.
- “Se sintió parte del desarrollo del producto?” , el 35% respondió afirmativamente, el 47,5% respondió negativamente, el 12,5% respondió no tener conocimiento sobre el tema y el 5% no respondió la pregunta.
- “Le hubiese gustado participar de forma más activa durante el proceso de desarrollo?, el 42,5% respondió afirmativamente, el 35% respondió negativamente, el 17,5% respondió no tener conocimiento sobre el tema y el 5% no respondió la pregunta.

Los resultados parecieran sugerir que los proveedores de software a medida, no logran cumplir con los planes, ni en tiempo, ni en costo, ni en funcionalidad.

Más del 50% de los usuarios encuestados no formó parte del desarrollo del software, por lo que se deduce que probablemente el proceso de elicitación de requerimientos no fue realizado de la manera correcta, ya que no involucró dentro del proceso a los principales *stakeholders* y consecuentemente un bajo porcentaje no tuvo correcciones de funcionalidad, por fallas o completa satisfacción del nivel de cumplimiento de los requerimientos.

Parte II: Estudio comparativo del proceso de elicitación de requerimientos entre tipos de software

Parte II - 1: Tipos de software y buenas prácticas para el proceso de elicitación de requerimientos

Se evaluó la asociación entre la alternativa de software y las siguientes prácticas recomendadas para el desarrollo de software [Sommerville:1997]. La Fig. 38 muestra a continuación la comparación del desarrollo de software a medida y empaquetado, de acuerdo a la frecuencia de utilización de estas prácticas durante el proceso de elicitación de requerimientos.

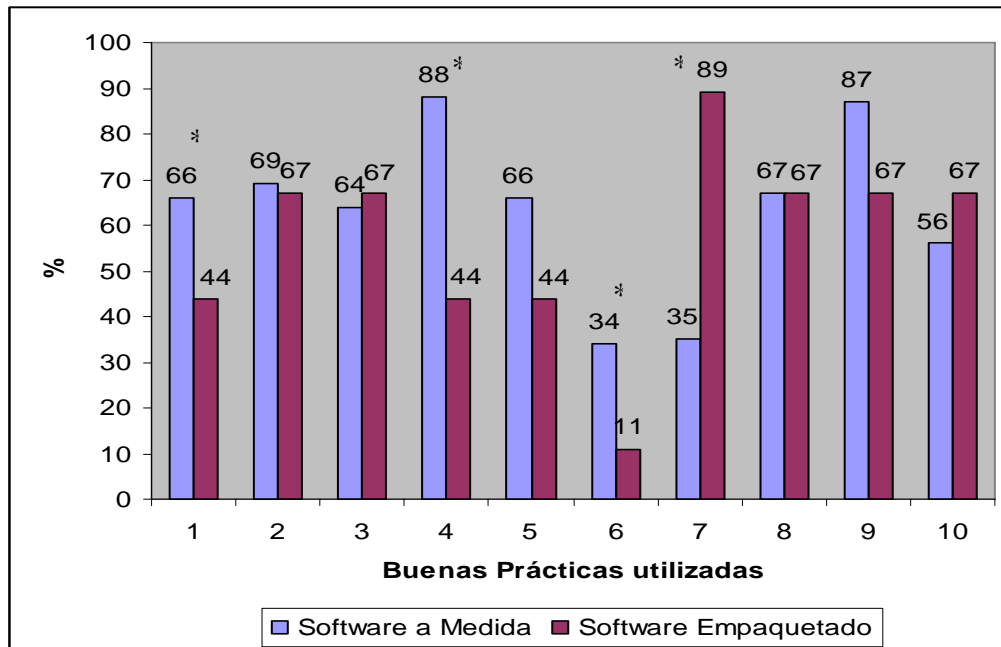


Figura 38: Comparación de la utilización de buenas prácticas para la elicitación de requerimientos entre software a medida y empaquetado.

Referencias: 1) Evaluar la factibilidad del sistema; 2) Tener en cuenta consideraciones políticas y organizacionales; 3) Registrar el origen de los requerimientos; 4) Tener en cuenta restricciones del ambiente; 5) Registrar la razón o importancia de los requerimientos; 6) Recolectar requerimientos desde múltiples puntos de vista; 7) Reutilizar requerimientos; 8) Identificar todas las clases de usuarios; 9) Planificar las actividades; 10) Realizar un resumen de cada una de las actividades realizadas

El asterisco (*) indica que existe una diferencia estadísticamente significativa entre ambos tipos de software en cuanto a la frecuencia de utilización de la práctica asociada ($p < 0,05$). Los resultados sugieren que la evaluación de la factibilidad del sistema, la consideración de restricciones del ambiente; la recolección de requerimientos desde múltiples puntos de vista se realiza y la reutilización de requerimientos; son prácticas que están estadísticamente asociadas a la variable, en este caso la alternativa de software. La presencia de las

diferencias encontradas corrobora la hipótesis de que el tipo de alternativa de software condiciona la utilización o no de las buenas prácticas sugeridas por la ingeniería de requerimientos (Anexo IV). Las diferencias en frecuencias observadas para las restantes actividades no fueron estadísticamente significativas ($P > 0,05$).

Parte II – 2: Tipos de software y técnicas de elicitación utilizadas según la opinión de los desarrolladores

El siguiente es un gráfico que muestra en comparación las técnicas utilizadas para la elicitación de requerimiento en las diferentes alternativas de software.

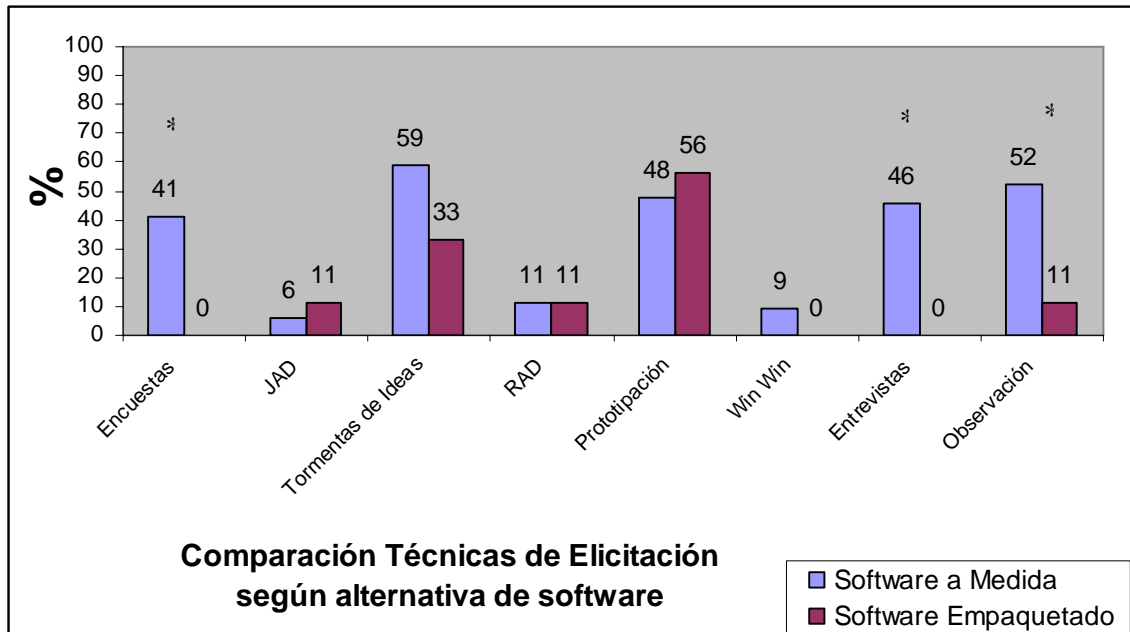


Figura 39: Comparación técnicas de elicitación utilizadas para sw a medida y empaquetado

En la Fig. 39 se observa que se encontraron diferencias estadísticamente significativas ($p < 0,05$) respecto a las técnicas más utilizadas para la elicitación de requerimientos cuando se comparan aquellas más usadas en el software a medida respecto al empaquetado. Se puede concluir que la Tormenta de Ideas y la Prototipación se identificaron como técnicas comunes de elicitación más utilizadas. Sin embargo las Entrevistas, las Encuestas y la Observación son técnicas de elicitación estadísticamente asociadas a la variable (Anexo VI)

Parte II – 3: Asociación entre modalidades de varias variables simultáneamente (Análisis Multivariado). Perfil de los desarrolladores de software a medida y empaquetado según elicitación de requerimientos.

El plano factorial presentado en la Fig. 40 define un perfil bastante marcado de los desarrolladores de software a medida el cual se contrapone al perfil de los desarrolladores del software empaquetado en relación a características o atributos asociados a las buenas prácticas para el proceso de elicitación. Dentro de las características propias de cada uno de estos perfiles, se pueden nombrar las siguientes: los desarrolladores de software a medida de nuestro medio tienen en cuenta toda y cada una de las buenas prácticas recomendadas en la literatura [Sommerville:1997], mientras que los desarrolladores de software empaquetado no lo hacen en la misma medida. Principalmente, los desarrolladores de software a medida a la hora de elicitar los requerimientos del cliente tienen en cuenta la factibilidad técnica del sistema, las restricciones del ambiente sobre el cual dicho sistema deberá correr, las consideraciones políticas y organizacionales y registran el origen de los requerimientos, este punto es particularmente importante debido a que así se mantiene identificados cual es o fue el *stakeholders* que manifestó la necesidad de que el software satisfaga ciertos requerimientos. Tarea que cobra importancia a hora del análisis de los requerimientos en busca de inconsistencias y de incompletitudes y sobre todo a la hora de realizar la trazabilidad de los requerimientos contra el producto final; el reuso de los requerimientos probablemente sea una de las prácticas menos utilizadas por los desarrolladores de software a medida, pero la misma presenta grandes ventajas si se la realiza de manera correcta (decremento el costo y esfuerzo durante el proceso de elicitación, análisis y especificación de los requerimientos). La planificación de las actividades, que también caracteriza a los desarrollos a medida, es una de las prácticas más importantes, no solo para la etapa de requerimientos sino también para el resto del proyecto en su totalidad. Sin la planificación, los líderes de proyectos no poseerían una forma fácil y dinámica de control sobre el proyecto, es decir, no sabrían si el mismo se encuentra dentro del presupuesto o no, o incluso si el proyecto se encuentra demorado o no, entre otras.

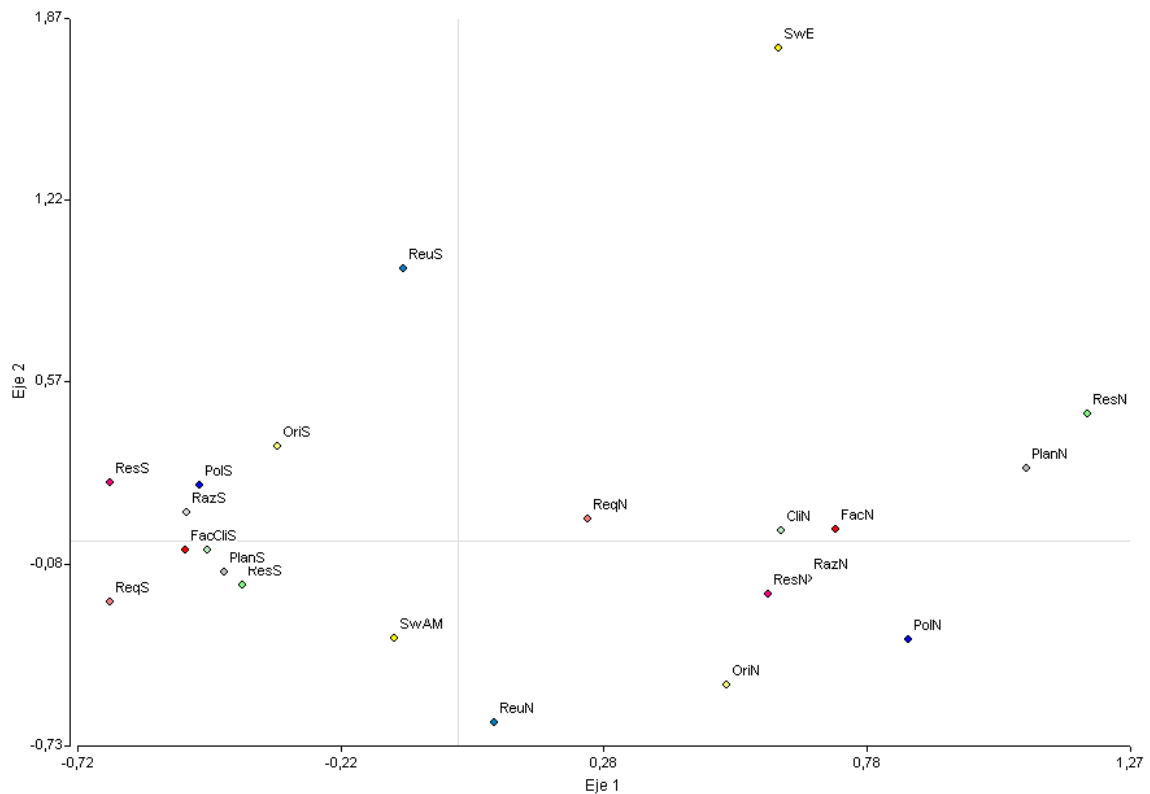


Figura 40: Primer plano factorial asociado al análisis de correspondencias múltiples.

Referencias: Alternativas de software: Software empaquetado (SwE), Software a medida (SwAM)

Se tienen en cuenta la factibilidad técnica del sistema: FacS (SI), FacN (NO); Se tienen en cuenta restricciones del ambiente: ResS (SI), ResN (NO); Se tienen en cuenta consideraciones políticas y organizacionales: PolS (SI), PolN (NO); Se registra el origen de los requerimientos: OriS (SI), OriN (NO); Se registra la razón o importancia de los requerimientos: RazS (SI), RazN (NO); Se registran requerimientos desde múltiples puntos de vista: ReqS (SI), ReqN (NO); Se reutilizan requerimientos: ReuS (SI), ReuN (NO); Se planifican las actividades: PlanS (SI), PlanN (NO); Se identifican todas las clases de clientes: CliS (SI), CliN (NO); Se realiza un resumen luego de cada actividad: ResS (SI), ResN (NO)

Parte II – 4: Tipos de software y técnicas de elicitación utilizadas según la opinión de los usuarios

A continuación la Fig. 41 muestra cual es la opinión de los usuarios de software según su percepción del proceso de elicitación de requerimientos que los desarrolladores de ambos tipos de software realizaron.

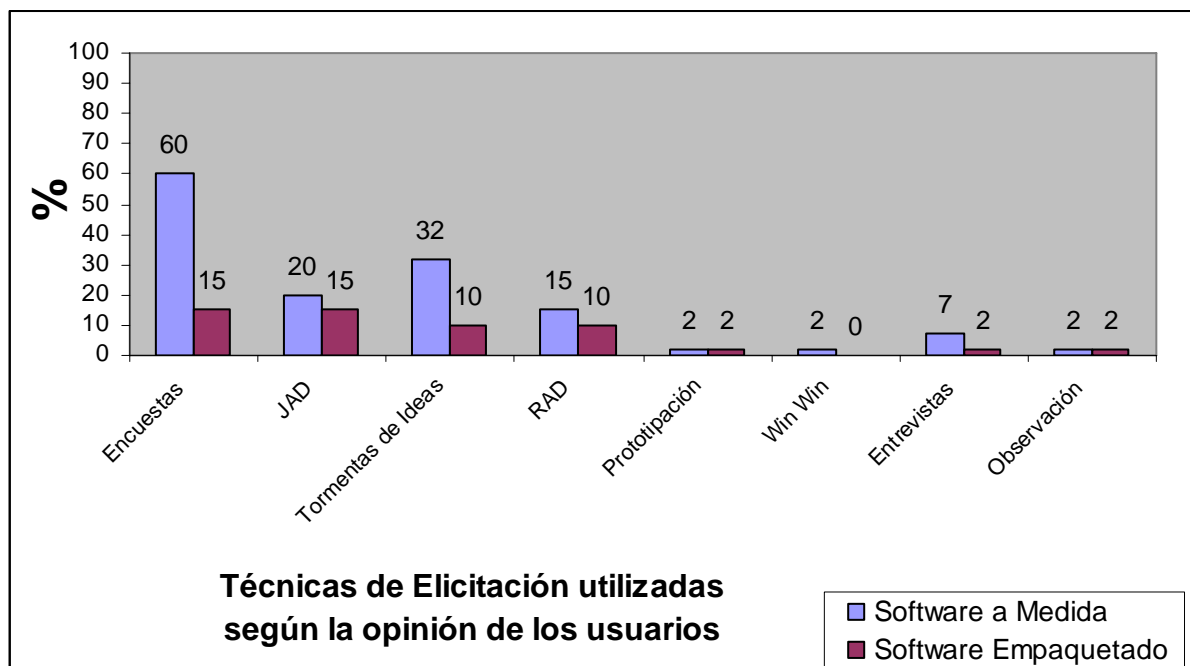


Figura 41: Técnicas de elicitación utilizadas según la opinión de los usuarios

La mayoría de los usuarios de software a medida (60%) reconocen a la Encuesta como la técnica de elicitación más común, en un segundo lugar a la Tormenta de Ideas y en un tercer lugar a JAD. A pesar de las ventajas citadas para la Prototipación, Entrevistas y Observación como técnicas de elicitación participativas, éstas fueron escasamente reconocidas por los usuarios.

Parte II – 5: Tipos de software y porcentaje de cumplimiento según la opinión de los usuarios

Las Fig. 42 y 43 muestran cual es el porcentaje de cumplimiento del software, ya sea a medida o empaquetado, con respecto a los requerimientos establecidos una vez que los mismos fueron elicitados e implementados, según la percepción de los usuarios. Para el análisis comparativo según la alternativa de software desarrollado, se tuvo en cuenta la pregunta “Se implementaron todos los cambios que usted propuso?” para el software empaquetado y la pregunta “El software desarrollado cumple con los requerimientos establecidos?” para el software a medida.

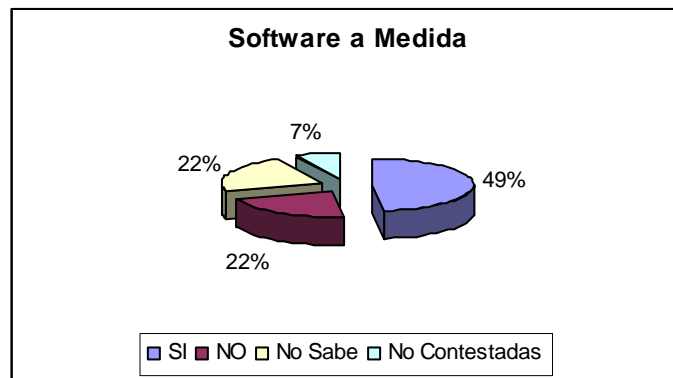


Figura 42: Porcentaje de cumplimiento del sw a medida con respecto a los requerimientos establecidos

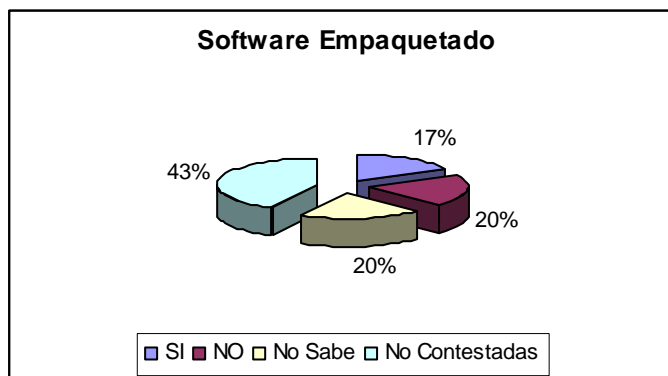


Figura 43: Porcentaje de cumplimiento del sw empaquetado respecto a los requerimientos establecidos

Como se muestra en la Fig. 42 el 49% de los usuarios respondió que el software a medida cumple con sus expectativas, mientras que como se puede observar en la Fig. 43 solo el 17% de los mismos respondió que así lo hace el software empaquetado.

Parte II – 6: Alternativas de software y la necesidad de adaptación según la opinión de los usuarios

En la Fig. 44 se observa la medida en que el software no cumplió con los requerimientos establecidos y/o especificados, se tuvo en cuenta para las encuestas el grado de adaptabilidad que el software entregado presentó.

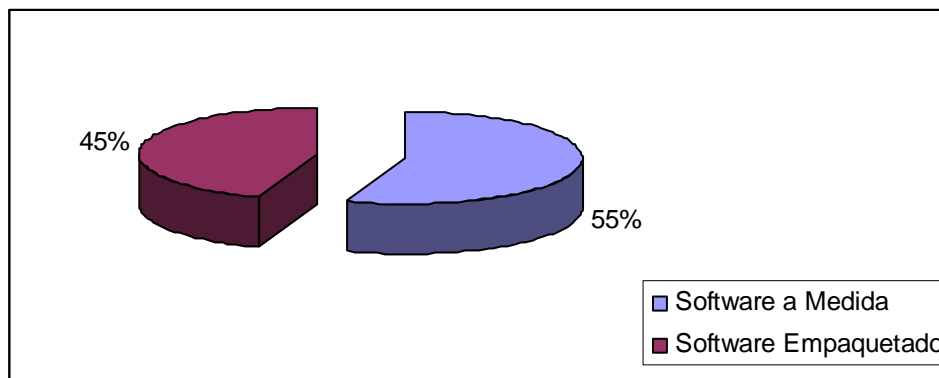


Figura 44: Adaptabilidad del software de acuerdo a la alternativa de software.

Nota: Para la realización del gráfico anterior se tuvo en cuenta la pregunta “El software necesitó ser adaptado a su empresa?” para software empaquetado y para software a medida las preguntas “El software debió ser corregido debido a fallas?” y “El software debió ser corregido debido a funcionalidad mal definida?”

El gráfico de la Fig. 44 muestra que el 55% de los encuestados respondió que el software desarrollado a medida tuvo que ser adaptado a las necesidades de la empresa de acuerdo a fallas encontradas en el mismo, esta necesidad de adaptación se debió a dos causas principales, la primera se debe a que el software “se clavaba”, es decir, el software producía errores graves los cuales no permitían su correcta operación, y la segunda debido a que la funcionalidad que el software ofrecía estaba mal definida, es decir, que los requerimientos del software no fueron correctamente elicitados o correctamente especificados. Mientras que el 45% de los encuestados respondió que el software empaquetado comprado o adquirido necesitó ser adaptado a las necesidades propias de la empresa debido a que el mismo no presentaba las características adecuadas para la empresa.

Parte II – 7: Comparación de preferencias por una alternativa de software

La Fig. 45 muestra las preferencias de los usuarios por un software a medida o por uno empacutado, vemos que el 65% de los encuestados prefiere la adquisición de un software empacutado debido al costo que el mismo posee en comparación con un software a medida. Teniendo en cuenta la funcionalidad que el software ofrece, vemos que el 52% de los encuestados también prefiere la adquisición de un software empacutado en vez de un software a medida. Si se tiene en cuenta la confianza ofrecida por una u otra alternativa de software, los encuestados parecen no tener una diferencia marcada con respecto a este factor de comparación.

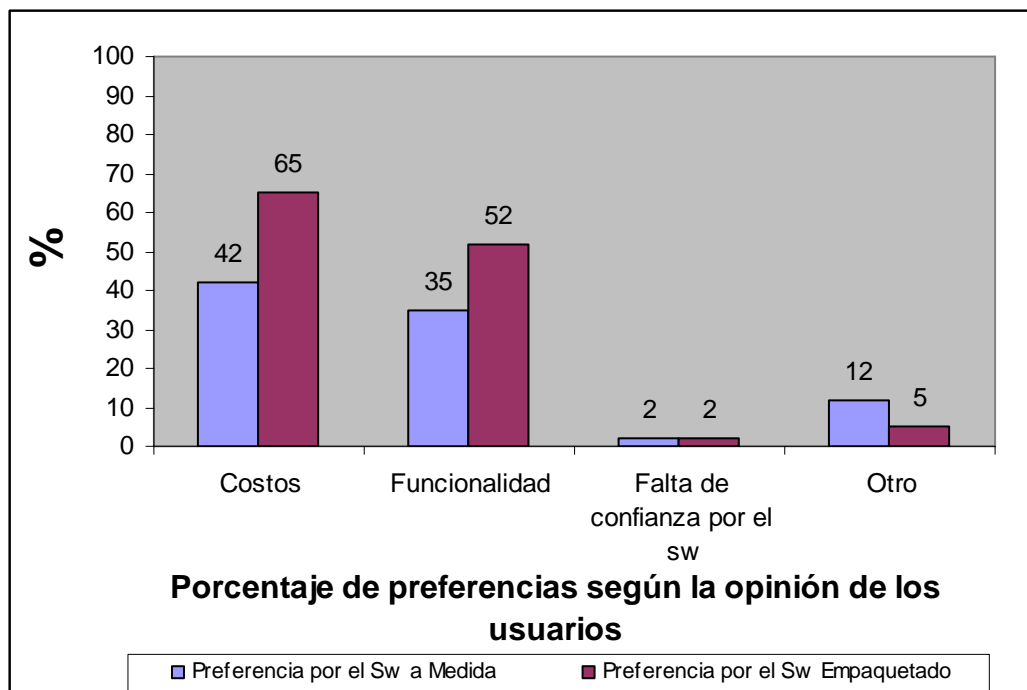


Figura 45: Preferencias de los usuarios según características genéricas del software

Parte III: Uso de métricas

La selección de métricas para el proceso de elicitación fue identificada como un aspecto clave en la gestión exitosa de esta fase del desarrollo y por ello ese objetivo de esta tesis realizar una propuesta al respecto. Las encuestas elaboradas tanto para desarrolladores como para usuarios incluyeron preguntas orientadas a analizar la percepción de estos actores respecto al uso de métricas en el desarrollo de software. La finalidad de estas preguntas fue considerar las prácticas que cada uno de los encuestados posee con respecto a las métricas que se tienen en cuenta para los procesos de requerimientos y más específicamente para el proceso de elicitación de requerimientos.

La Fig. 46 muestra que el 87% de los encuestados no tiene conocimientos sobre la utilización de alguna métrica para el proceso de requerimientos, 9% declara no utilizar ninguna y sólo el 4% de los encuestados habían utilizado alguna métrica para evaluar y monitorear esta importante etapa.

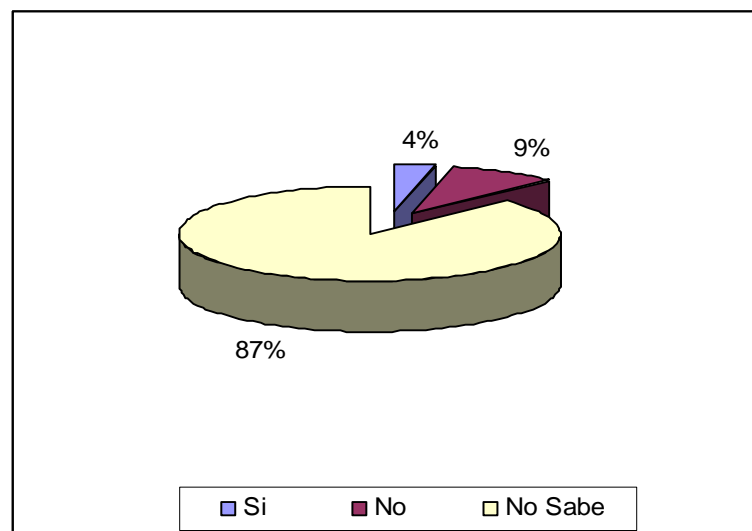


Figura 46: Utilización de métricas para el proceso de requerimientos

Más específicamente para el proceso de elicitación de requerimientos el 42% respondió que no posee conocimientos sobre la aplicación de métricas para la elicitación, el 47% afirmó que no se toman métricas en esta etapa del ciclo de vida del proyecto y el 11% de los encuestados no respondió la encuesta. Es decir ni desarrolladores ni usuarios de nuestro medio perciben que la etapa de elicitación puede ser cuantificada, sugiriendo que el conocimiento de la misma no es maduro y que por tanto será difícil realizar predicciones sobre la adaptabilidad y funcionalidad del software, ni evaluaciones de posibles impactos de esta temprana etapa sobre las posteriores. Fig. 47.

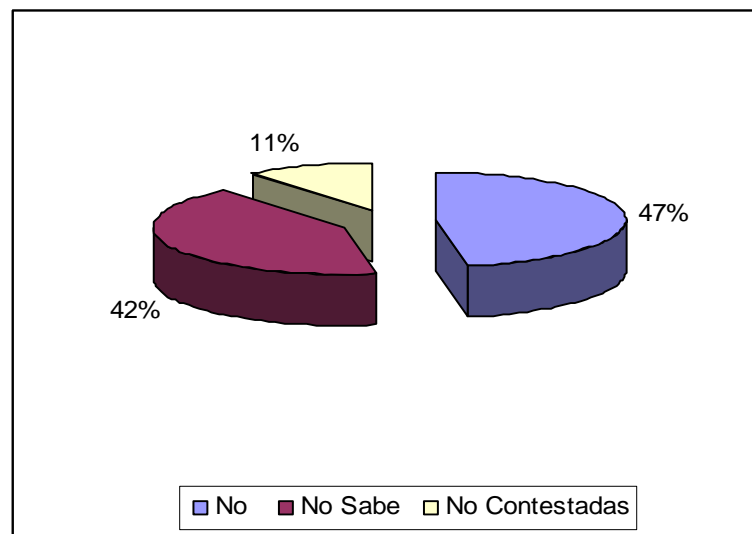


Figura 47: Utilización de métricas para el proceso de elicitación de requerimientos

La Fig. 48 muestra que sólo el 15% de los encuestados respondió que le parece útil tomar algún tipo de métrica para la elicitación de requerimientos, el 9% negó la utilidad y la gran mayoría (65% de los encuestados) respondió que no sabe si es útil o no tomar algún tipo de métrica en esta etapa del ciclo de desarrollo.

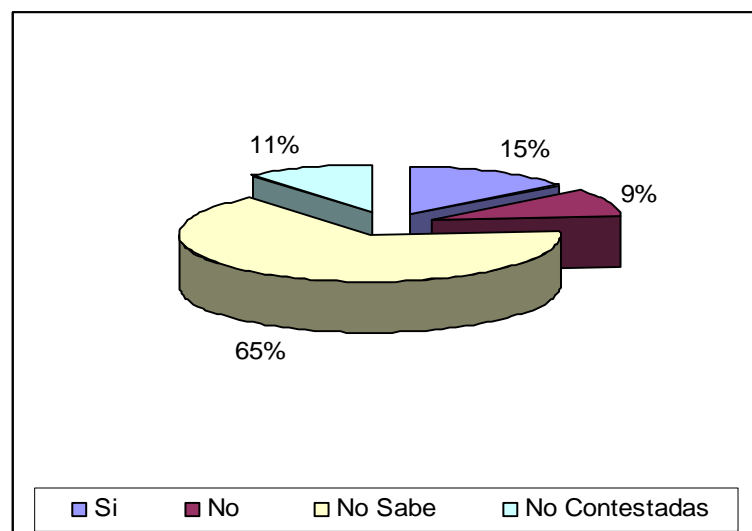


Figura 48: Utilidad en la toma de métricas

La Fig. 49 muestra que según las encuestas sólo el 11% de los desarrolladores toma algún tipo de métricas para cualquier aspecto de los demás procesos del ciclo de vida del software (Análisis, Diseño, Codificación, Testing), el 44% directamente no las toma y el 36% no tiene conocimiento si la empresa en la que trabaja las tiene en cuenta o no.

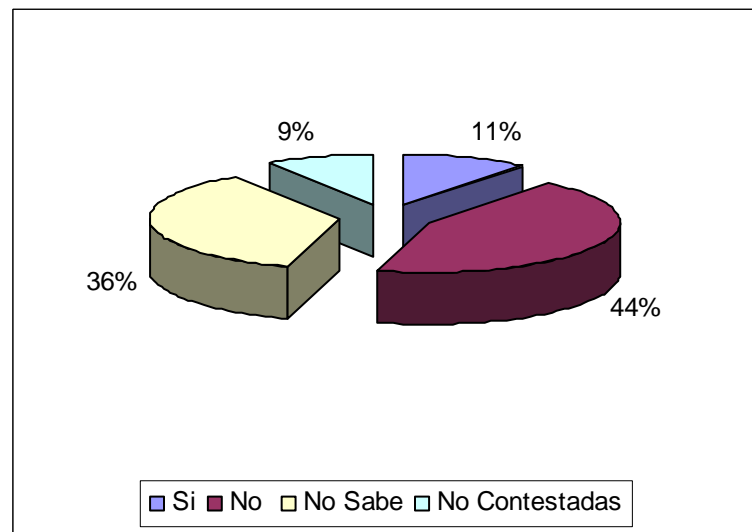


Figura 49: Métricas tomadas luego de la elicitación de requerimientos

Dentro de las métricas que se toman luego del proceso de requerimientos, y más específicamente después del proceso de elicitación de requerimientos, la más utilizada es la medición del grado de satisfacción del usuario. A pesar que sólo el 11% de los encuestados respondió favorablemente respecto al uso de alguna métrica durante el ciclo de vida del software, el 56% de los encuestados respondió que si se considera la satisfacción del usuario, 38% son desarrolladores de software a medida y solo el 18% de desarrolladores de software empaquetado, el 23% respondió que no lo hacen y el 16% respondió que no está seguro de que esa métrica sea tomada en su empresa.

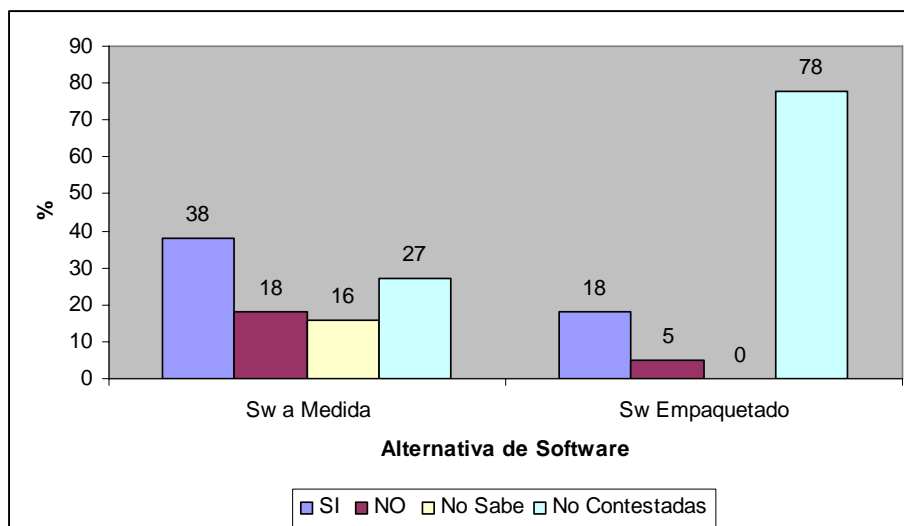


Figura 50: Medición de la satisfacción del usuario

Capítulo 5: Propuestas de métricas para el proceso de elicitación de requerimientos

Teniendo en cuenta la importancia y la falta de métricas definidas para el proceso de elicitación de requerimientos, en la primera encuesta, se analizó la percepción de los desarrolladores respecto a una serie de métricas que podrían ser útiles para dicho proceso. En la Tabla 1 se presenta la lista de métricas propuestas sobre las que se recabó información tanto de desarrolladores de software a medida como empaquetado. Cabe destacar que se toma como requerimiento candidato a los requerimientos identificados que todavía no han sido analizados, es decir, el proceso de análisis de requerimientos todavía no ha sido iniciado.

Tabla 1: Propuesta de métricas para el proceso de elicitación de requerimientos

Métrica Propuesta	Descripción de la Métrica
Cantidad de personas elicidadas	Cantidad de personas que se tuvieron en cuenta para la recolección de requerimientos
Cantidad de <i>stakeholders</i> involucrados con el software	Cantidad de personas afectadas por el software a desarrollar pero que no necesariamente fueron elicitados, por ejemplo: usuarios finales, gerentes, sponsors, etc
Cantidad de veces que se tuvo que relevar requerimientos del mismo <i>stakeholders</i>	Cantidad de veces que un mismo <i>stakeholders</i> tuvo que ser elicitado debido a que los requerimientos candidatos identificados no fueron entendidos
Cantidad de puntos de vistas elicitados	Los puntos de vista se definen de acuerdo al rol/responsabilidades que una persona tiene
Cantidad de técnicas de elicitación utilizadas	Cantidad de técnicas utilizadas durante todo el proceso de elicitación teniendo en cuenta la cantidad total de personas elicidadas
Cantidad de requerimientos candidatos identificados	Cantidad de requerimientos identificados en una primera instancia de elicitación, sin pasar por el proceso de análisis de requerimientos
Porcentaje de requerimientos candidatos funcionales	Cantidad de requerimientos relacionados con la funcionalidad del

Métrica Propuesta	Descripción de la Métrica
	software sobre el total de requerimientos elicitados.
Porcentaje de requerimientos candidatos de calidad	Cantidad de requerimientos de calidad relacionados al software sobre el total de requerimientos elicitados.
Cantidad de inconsistencias (o incompletitudes) encontradas en requerimientos candidatos identificados teniendo en cuenta la misma técnica de elicitación utilizada y diferente el usuario (para un misma alternativa de software)	Cantidad de requerimientos inconsistentes, es decir por ejemplo que dos requerimientos se contradigan, teniendo en cuenta la misma técnica de elicitación utilizada, pero considerando distintas personas elicitadas
Cantidad de inconsistencias (o incompletitudes) encontradas en requerimientos candidatos identificados teniendo en cuenta distintas técnicas de elicitación utilizada y el mismo usuario (para un misma alternativa de software)	Cantidad de requerimientos inconsistentes, es decir por ejemplo que dos requerimientos se contradigan, teniendo en cuenta la misma persona elicitada pero distinta técnica de elicitación aplicada
Cantidad de requerimientos reutilizados	Cantidad de requerimientos reutilizados de proyectos anteriores que aplican al proyecto actual
Cantidad de requerimientos bajo la línea base	Cantidad de requerimientos revisados y aprobados por un comité de evaluación
Cantidad de requerimientos candidatos realmente implementados	Número de requerimientos identificados en una primera instancia que fueron realmente implementados como tales en el producto de software final
Cantidad de requerimientos no identificados durante el proceso de elicitación	Número de requerimientos que surgieron en etapas posteriores a la elicitación de requerimientos, sino en las etapas de análisis, especificación o validación de los mismos
Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos	Número de requerimientos que surgieron en etapas posteriores en el ciclo de vida del software, por ejemplo en análisis, diseño, codificación, testing, etc.

Métrica Propuesta	Descripción de la Métrica
Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto	Cantidad de horas hombre insumidas en el proceso de elicitación de requerimientos sobre la cantidad de horas insumidas totales del proyecto
Porcentaje de esfuerzo en toda la etapa de requerimiento respecto del total del proyecto	Cantidad de horas hombre insumidas en la etapa de requerimientos (tanto elicitación, análisis, especificación, validación) sobre la cantidad de horas insumidas totales del proyecto
Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto	Cantidad de horas insumidas en el proceso de elicitación de requerimientos sobre la cantidad de horas insumidas totales del proyecto
Porcentaje de tiempo en toda la etapa de requerimiento respecto del total del proyecto	Cantidad de días, teniendo en cuenta el tiempo de vida (<i>cycle time</i>) del proyecto insumidas en la etapa de requerimientos (tanto elicitación, análisis, especificación, validación) sobre la cantidad de días totales del proyecto
Cantidad de páginas del documento de requerimientos	Cantidad total de páginas que posee el documento de requerimientos una vez que el mismo ha sido revisado y aprobado por un comité de evaluación.
Nivel de satisfacción del cliente	Grado en el cual el cliente o usuario final está conforme o no con el producto desarrollado
Nivel de satisfacción del desarrollador o equipo de desarrolladores respecto al proceso de elicitación de requerimientos	Grado en el cual el desarrollador o el equipo de desarrolladores está conforme o no con los resultados de la etapa de elicitación
Nivel de satisfacción del desarrollador o equipo de desarrolladores respecto al software desarrollado	Grado en el cual el desarrollador o el equipo de desarrolladores está conforme o no con el producto desarrollado

Las Tablas 2 y 3 muestran la estadística descriptiva de cada variable (métrica candidata) para el total de desarrolladores encuestados y según la alternativa de software desarrollado, respectivamente (Anexo VIII).

Tabla 2: Estadística descriptiva. Valores promedios, mínimos y máximos para cada métrica

Variable	Media	Mínimo	Máximo
Cantidad de personas elicítadas	3	1	6
Cantidad de <i>stakeholders</i> involucrados con el software	5	2	10
Cantidad de veces que se tuvo que relevar requerimientos del mismo <i>stakeholders</i>	8	2	20
Cantidad de puntos de puntos de vistas	2	1	4
Cantidad de técnicas de elicitación utilizadas	2	1	3
Cantidad de requerimientos candidatos identificados	23	7	55
Cantidad de requerimientos reutilizados	5	0	20
Cantidad de requerimientos reutilizados	19	0	60
Cantidad de requerimientos candidatos realmente implementados	28	7	93
Cantidad de requerimientos no identificados durante el proceso de elicitación	8	0	20
Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos	3	0	15
Cantidad de páginas del documento de requerimientos	18	2	30
Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto	15	5	40
Porcentaje de esfuerzo en la etapa de requerimientos respecto del total del proyecto	23	5	45
Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto	14	5	40
Porcentaje de tiempo utilizado en la etapa de requerimientos respecto del total del proyecto	24	5	45
Porcentaje de requerimientos candidatos funcionales	56	10	90
Porcentaje de requerimientos candidatos de calidad	17	0	100

Tabla 3: Estadística descriptiva para cada variable según la alternativa de software, significancia estadística de la diferencia de valores promedios entre software a medida y empaquetado y de la asociación de la variable con la satisfacción del usuario y de los desarrolladores de software.

Cantidad de personas elicitadas						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	3	2	6	*	No significativo	No Significativo
Empaquetado	2	1	3			

Cantidad de <i>stakeholders</i> involucrados con el software						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	6	2	10	*	No Significativo	No Significativo
Empaquetado	4	2	5			

Cantidad de veces que se tuvo que relevar requerimientos del mismo <i>stakeholders</i>						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	6	2	20	*	No Significativo	*
Empaquetado	12	7	20			

Cantidad de puntos de vistas elicitados						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	2	1	4	No significativo	No Significativo	No Significativo
Empaquetado	3	1	4			

Cantidad de técnicas de elicitación utilizadas						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	2	1	3	No significativo	No Significativo	*
Empaquetado	2	1	2			

Cantidad de requerimientos candidatos identificados						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	26	7	55	No significativo	*	No Significativo
Empaquetado	18	10	30			

Cantidad de requerimientos reutilizados						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	6	0	20	*	No Significativo	*
Empaquetado	0	0	0			

Cantidad de requerimientos bajo línea base						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	22	0	60	No significativo	No Significativo	No Significativo
Empaquetado	12	10	13			

Cantidad de requerimientos candidatos realmente implementados						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	23	7	66	No significativo	*	No Significativo
Empaquetado	39	10	93			

Cantidad de requerimientos no identificados durante el proceso de elicitación						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	6	0	15	*	No Significativo	No Significativo
Empaquetado	18	15	20			

Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	2	0	15	*	No Significativo	No Significativo
Empaquetado	8	5	10			

Cantidad de páginas del documento de requerimientos						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	18	10	30	No significativo	No Significativo	No Significativo
Empaquetado	17	2	30			

Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	17	5	40	*	No Significativo	*
Empaquetado	8	5	10			

Porcentaje de esfuerzo utilizado en la etapa de requerimientos respecto del total del proyecto						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	24	5	45	No significativo	No Significativo	*
Empaquetado	20	15	25			

Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	17	5	40	*	No Significativo	*
Empaquetado	8	5	15			

Porcentaje de tiempo utilizado en la etapa de requerimientos respecto del total del proyecto						
Alternativa de software	Med	Mín	Máy	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	25	5	45	No significativo	No Significativo	*
Empaquetado	23	15	35			

Porcentaje de requerimientos candidatos funcionales						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	55	10	90	No significativo	No	No Significativo
Empaquetado	58	10	85		Significativo	

Porcentaje de requerimientos candidatos de calidad						
Alternativa de software	Med	Mín	Máx	Respecto de la alternativa de software?	Respecto de la satisfacción del cliente?	Respecto a la satisfacción de los desarrolladores sobre el producto entregado?
A medida	21	0	100	No significativo	No	*
Empaquetado	7	0	20		Significativo	

Nota: Las comparaciones y asociaciones estadística fueron realizadas con un nivel de significación del 5%.

Los resultados muestran que las siguientes cantidades propuestas como posibles métricas presentan valores promedios con diferencias estadísticamente significativas ($P < 0,05$) entre alternativas de software: Cantidad de personas elicitadas, Cantidad de *stakeholders* involucrados, Cantidad de veces que se tuvo que relevar requerimientos del mismo *stakeholders*, Cantidad de requerimientos reutilizados, Cantidad de requerimientos no identificados durante el proceso de elicitación, Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos, Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto, Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto. Las restantes cantidades no mostraron diferencias entre alternativas de software, es decir que de ser utilizadas como métricas estas no dependerían de la alternativa de software que se desarrolle. Mientras que Cantidad de personas elicitadas, Cantidad de *stakeholders* involucrados con el software, Cantidad de requerimientos reutilizados y Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto son principalmente utilizadas por desarrolladores de software a medida; Cantidad de veces que se tuvo que relevar requerimientos del mismo *stakeholders*, Cantidad de requerimientos no identificados durante el proceso de elicitación y Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos tuvieron valores promedios más altos en el desarrollo de software empaquetado. Los valores de las métricas Cantidad de

puntos de vistas elicitados y Cantidad de técnicas de elicitación utilizadas fueron bajos en ambos tipos de software. Los valores de Cantidad de requerimientos candidatos identificados, Porcentaje de requerimientos candidatos funcionales, Porcentaje de requerimientos candidatos de calidad, Cantidad de requerimientos bajo línea base, Cantidad de requerimientos candidatos realmente implementados y Cantidad de páginas del documento de requerimientos, fueron muy variables probablemente porque son más dependientes de las características del producto en desarrollo. Si bien las métricas Porcentaje de esfuerzo en toda la etapa de requerimientos respecto del total del proyecto, Porcentaje de tiempo en toda la etapa de requerimientos respecto del total del proyecto y Porcentaje de tiempo utilizado durante en la etapa de elicitación respecto del total del proyecto son relativas y por tanto menos dependiente de la complejidad del producto (métricas adimensionales), no mostraron el mismo comportamiento. Mientras Porcentaje de esfuerzo en toda la etapa de requerimientos respecto del total del proyecto y Porcentaje de tiempo en toda la etapa de requerimientos respecto del total del proyecto no difirieron significativamente entre los tipos de software, la Porcentaje de tiempo en toda la etapa de elicitación respecto del total del proyecto fue estadísticamente mayor en el desarrollo de software a medida.

De estas métricas propuestas las que presentaron relación estadísticamente significativa con el nivel de satisfacción del cliente fueron Cantidad de requerimientos candidatos identificados y Cantidad de requerimientos candidatos realmente implementados. Sin embargo las cantidades que se relacionaron estadísticamente con la satisfacción de los desarrolladores respecto al producto fueron más numerosas : Cantidad de veces que se tuvo que relevar requerimientos del mismo *stakeholders*, Cantidad de técnicas de elicitación utilizadas, Porcentaje de requerimientos candidatos de calidad, Cantidad de requerimientos reutilizados, Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto, Porcentaje de esfuerzo en toda la etapa de requerimientos respecto del total del proyecto, Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto y Porcentaje de tiempo utilizado en toda la etapa de requerimientos respecto del total del proyecto.

En la Fig. 51 se muestra la percepción de los desarrolladores según 5 métricas a las que debieron asignar órdenes de mérito (mientras más alto el orden mejor posicionada la métrica desde la perspectiva de los que responden). Los resultados muestran que más allá del nivel de satisfacción del cliente, que no solo representa una medida del resultado del proceso de elicitación sino de todo el proceso, la cantidad de requerimientos candidatos identificados fue la métrica mejor posicionada de aquellas consultadas.

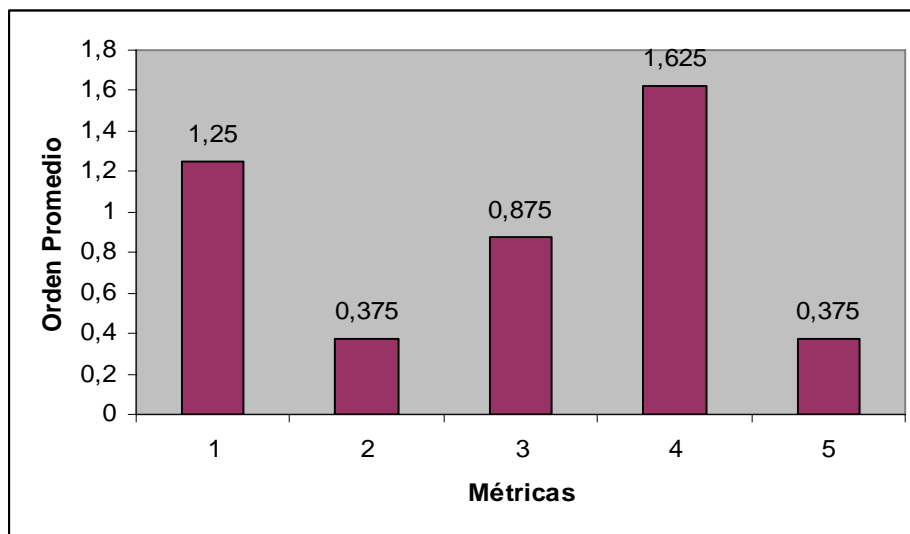


Figura 51: Métricas según el orden de mérito.

Referencias: 1) Cantidad de requerimientos candidatos identificados; 2) Porcentaje de requerimientos candidatos identificados; 3) Porcentaje de tiempo utilizado en la etapa de elicitación de requerimientos respecto del total del proyecto; 4) Nivel de satisfacción del cliente respecto del proceso de Elicitación; 5) Nivel de documentación del proceso de elicitación

Los resultados mostraron que 13 de los 40 desarrolladores entrevistados consideraron que el Nivel de Satisfacción del cliente respecto del proceso de elicitación de requerimientos es la métrica más importante dentro de las propuestas. Así mismo, 19 de los 40 desarrolladores consideraron que el Porcentaje de tiempo utilizado en la etapa de elicitación de requerimientos respecto del total del proyecto es la métrica menos importante .

Para concluir con el análisis de las entrevistas realizadas a los desarrolladores, se recuerda que se le solicitó a los mismos que realicen una proposición sobre alguna métrica de software correspondiente al proceso de elicitación de requerimientos que les resultase importante de obtener. A continuación se muestra en la Tabla 4 los resultados obtenidos luego de la entrevista.

Tabla 4: Métricas propuestas por los desarrolladores de software

Cód.	Proponga una metrica	Frecuencia
1	Cantidad de requerimientos identificados con una tecnica determinada	3
2	Densidad de defectos	3
3	Requerimientos funcionales vs. requerimientos no funcionales	3
4	Volatilidad de los requerimientos	3
5	Porcentaje de requerimientos candidatos implementados/porcentaje de tiempo respecto del total del proyecto utilizado en la elicitación de requerimientos	3
6	Porcentaje de requerimientos modificados sobre los requerimientos implementados	3
7	Satisfacción del cliente	3
8	Requerimientos identificados	3
9	Ratio entre requerimientos elicitados/requerimientos desarrollados	4
10	Cantidad de requerimientos bajo línea base	4
11	Cantidad de requerimientos extraídos a partir de una previa presentación de un prototipo funcional al cliente	4
12	No propone	4

Las métricas propuestas más comunes son: Ratio entre requerimientos elicitados/requerimientos desarrollados, Cantidad de requerimientos bajo línea base y Cantidad de requerimientos extraídos a partir de una previa presentación de un prototipo funcional al cliente.

Capítulo 6: Discusión y conclusiones

La ingeniería de requerimientos es una actividad compleja que requiere un proceso definido que debe ser bien planificado y monitoreado. Se centra principalmente en la extracción de información, desde múltiples y diversas fuentes, y además involucra la organización de la información relacionada al problema que necesita ser resuelto. Consiste de una secuencia de etapas: elicitación, análisis, especificación, validación y administración de requerimientos [Wiegers:1999]. El análisis realizado en este trabajo de tesis está enfocado al proceso de elicitación de requerimientos y ha sido realizado teniendo en cuenta dos ambientes o contextos, el desarrollo de software a medida y el desarrollo de software empaquetado.

La literatura revisada es contundente en exclamar que el proceso de elicitación de requerimientos es un paso crítico para el éxito de un proyecto de desarrollo de software y que no sólo significa el relevamiento o recolección de los requerimientos, sino que implica un esfuerzo participativo, una tarea para llevar a cabo conjuntamente entre desarrolladores y usuarios [Tuffley:2005].

Consecuentemente existen problemas en esta etapa del ciclo de vida del software que tienen que ver con la naturaleza humana: relaciones sociales, comunicación efectiva y administración de la información [COMP]. Entre los problemas asociados a la ingeniería de software, los requerimientos son la fuente de los problemas más frecuentes por lo tanto se debe asignar gran esfuerzo en la recolección de los mismos lo antes posible en el desarrollo del sistema. Puede suceder que los requerimientos no reflejan las necesidades reales de los clientes y/o que los requerimientos son inconsistentes entre si y/o son incompletos. Estos problemas provienen de un mal entendimiento entre los clientes, las personas que desarrollan los requerimientos y los ingenieros de software que desarrollan o mantienen el sistema [COMP]. El principal problema en la recolección de requerimientos es de comunicación.

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

El primer paso en el desarrollo de un producto de calidad lleva a cumplimentar la etapa de desarrollo de requerimientos. El desarrollo de requerimientos, así como la administración de los mismos, es importante ya que si se realizara de manera errónea se podría estar desarrollando el software incorrecto [Bohem:1981]. El primer paso del desarrollo de requerimientos es la elicitación de los mismos.

La elicitación de requerimientos es el proceso de descubrir los requerimientos para un sistema a través de la comunicación con los clientes, usuarios del sistema y otras personas que tengan algún tipo de interés y conocimiento sobre el producto a desarrollar [Madigan]. El principal resultado del proceso de elicitación de requerimientos es la serie de requerimientos que deberán ser utilizados por el equipo de desarrollo de software para crear un producto correcto y de manera apropiada [Tuffley:2005].

Por otro lado, al evaluar la elicitación de requerimientos desde los dos contextos seleccionados se puede decir que el software empaquetado se caracteriza por ser, en general, más económico y se asocia a un proceso de disponibilidad rápido: “Se compra y se instala”. El software empaquetado es mejor para funciones que cambian lentamente y que no son estratégicas para la empresa. Sin embargo, si se piensa que una función es estratégica o que el modelo de negocio es único las soluciones empaquetadas que han sido desarrolladas para cubrir un espectro amplio de requerimientos podrían ser insuficientes para cubrir requerimientos específicos [TechRepublic].

El sistema a medida se hace codo a codo entre quién lo encarga y quién lo desarrolla, la elicitación de requerimientos surge de la comunicación directa entre desarrolladores y usuarios. Es un sistema que está hecho exclusivamente para quien lo necesita abarcando cualquier situación prevista y adaptándose lo mejor posible a los requerimientos de cada cliente particular. Si hay cualquier cambio en una situación que implique un cambio en el programa, se tendrá previo contacto con la persona que lo diseñó para implementar el cambio. Así, se puede pensar que el sistema a medida es más flexible a las necesidades y nuevos requerimientos que el sistema empaquetado. Un software correctamente diseñado, es una solución invaluable para cualquier empresa. El costo de una forma de trabajo ineficiente, a largo plazo, es enormemente mayor al del desarrollo de un sistema a medida [Brain Desarrollos].

La naturaleza adaptativa y diversa de la elicitación de conocimiento del usuario hace que sea difícil formalizar el proceso de elicitación y aún más complicado cuantificarlo a través de métricas apropiadas. El uso de éstas en nuestro medio es casi imperceptible; este trabajo tiene como objetivo principal la proposición de una serie de métricas directas e indirectas que podrían favorecer a dicho proceso.

El análisis realizado permite confirmar que el proceso de requerimientos, y más específicamente el proceso de elicitación de los mismos es un factor clave para el éxito de los proyectos de desarrollo de software, tanto sean estos a medida como empaquetado. Más de 2/3 de los proyectos de desarrollo de software que se realizan en nuestro medio poseen desviaciones entre el tiempo planificado de entrega y el tiempo real, y esas desviaciones tienen como causa principal al proceso de requerimientos (65%). Fig. 14.

Los resultados permitieron corroborar, como se demostró, la hipótesis planteada para este trabajo - el contexto definido por la alternativa de software (a medida o empaquetado) modifica el proceso de elicitación.

Al investigar y analizar comparativamente las técnicas de elicitación seleccionadas, que pueden ser utilizadas para el desarrollo de software a medida y/o empaquetado se encuentran diferencias a nivel de los desarrolladores. Se encontraron asociaciones entre las técnicas elegidas y la alternativa de software a desarrollar. Se puede concluir que la Tormenta de Ideas y la Prototipación se identificaron como técnicas comunes de elicitación más utilizadas. Sin embargo, como se demostró estadísticamente, las Entrevistas, las Encuestas y la Observación son técnicas de elicitación que están fuertemente asociadas a la alternativa de software a desarrollar (a medida o empaquetado) (Fig. 39). Por otro lado, según la percepción de los usuarios, reconocen a las Encuestas como la técnica de elicitación más comúnmente utilizada por los desarrolladores.

Con respecto a las buenas prácticas recomendadas en la literatura para la etapa de elicitación de requerimientos [Sommerville:1997], se observó que éstas son aplicadas en mayor medida por los desarrolladores de software a medida que por los desarrolladores de software empaquetado (Parte II - 3, Fig. 40). Sin embargo, teniendo en cuenta prácticas específicas como: evaluar la factibilidad del sistema, tener en cuenta restricciones del ambiente, recolectar requerimientos desde múltiples puntos de vistas y reutilizar requerimientos; se demostró que dichas prácticas están estadísticamente asociadas con la alternativa de software a desarrollar (Parte II - 1, Fig. 38).

Respecto a los usuarios de nuestro medio se concluye que estos prefieren la adquisición de un software empaquetado debido al costo que el mismo posee en comparación con un software a medida (65%). Pero por otro lado, teniendo en cuenta la funcionalidad que el software ofrece, más de la mitad de los usuarios encuestados pareciera prefiere la adquisición de un software empaquetado en vez de un software a medida, a pesar de que si se considera la confianza ofrecida por una u otra alternativa de software, los usuarios no expresa una diferencia marcada con respecto a este factor de comparación. (Parte II - 7, Fig. 45). Al evaluar el grado de cumplimiento que el software desarrollado/comprado tiene

para los usuarios vemos que tanto el software a medida como el software empaquetado adquiridos, tuvieron que ser adaptados, en mayor o menor medida, a las necesidades propias de la organización. (Parte II – 6, Fig. 44)

El nivel de desarrollo de software empaquetado en nuestro país es incipiente, solo el 16% de los encuestados respondieron que alguna vez tuvieron la oportunidad de desarrollar uno (Fig. 11). Este hallazgo debe ser considerado desde dos puntos de vistas: 1) como un llamado de atención y una oportunidad de mejora a la industria del software nacional, en el sentido de promocionar el desarrollo y la difusión de esta alternativa de software y 2) como información a considerar a la hora de seleccionar técnicas y métricas para el proceso de elicitación de requerimientos en nuestro medio, ya que como sugiere el trabajo éstas no son exactamente las mismas para ambientes donde se desarrolla software a medida respecto a aquellos orientados al desarrollo de software empaquetado.

En función de los resultados obtenidos se elaboró una guía para tener en cuenta durante la elicitación de requerimiento y se propuso una serie, al menos empíricamente fundamentada, de métricas para el proceso de elicitación.

Esta guía como algunas guías existentes [Madigan], tiene en cuenta las buenas prácticas recomendadas por [Wieggers:1999], y además a diferencia de estas, toma en consideración como una parte muy importante a la alternativa de software elegida para el desarrollo del software y a la recolección, análisis y monitoreo de métricas del software. A continuación se presenta dicha guía:

1. Identificar los principales objetivos del software a desarrollar, teniendo en cuenta la alternativa de software elegida.
2. Evaluar la factibilidad del sistema.
3. Tener en cuenta las consideraciones políticas y organizacionales de la empresa.
4. Identificar restricciones del dominio.
5. Identificar *stakeholders* del sistema.
6. Identificar y definir las métricas a recolectar.
7. Identificar la/s personas responsables para la recolección de las métricas.
8. Definir un repositorio para registrar métricas.
9. Identificar personas claves a elicitar, teniendo en cuenta por ejemplo rol/responsabilidades que desarrolla en la empresa.
10. Seleccionar técnicas de elicitación de requerimientos según corresponda (por ejemplo: de acuerdo a la alternativa de software elegida, rol de la persona, etc).
11. Registrar las fuentes de los requerimientos.
12. Registrar la razón de los requerimientos.

13. En el caso que alguno de los requerimientos identificados no fue claramente entendido, utilice otra técnica de elicitación distinta a la ya utilizada, por ejemplo prototipación.
14. Reutilizar requerimientos.
15. Recolectar métricas.
16. Evaluar, monitorear y tomar acciones correctivas continuamente.

La medición es el proceso por el cual se asigna una serie de números o símbolos a uno o más atributos de las entidades o procesos bajo consideración; esta asignación se realiza de acuerdo a una serie de reglas definidas y claramente identificadas. Una vez que la asociación evento-valor ha sido realizada, es posible comparar las entidades de interés a través de la comparación de estos valores [Definición]. La medida es un factor clave en el desarrollo del software ya que: ayuda a entender qué está pasando durante el desarrollo y el mantenimiento, permite controlar el desarrollo del proyecto, y estimula la mejora de procesos y productos. Las métricas propuestas fueron arregladas en la tabla según la alternativa de software (a medida o empaquetado) para la que parecieran ser más propicias. En el caso de no existir división entre las celdas de la tabla, debe entenderse que la métrica puede ser utilizada para ambas alternativas de software

Métricas directas

Tabla 5: Propuesta de métricas directas

Software a Medida	Software Empaquetado
<ul style="list-style-type: none"> • Cantidad de <i>stakeholders</i> involucrados con el software 	<ul style="list-style-type: none"> • Cantidad total de mercados identificados: hace referencia a la cantidad total de mercados identificados en una primera instancia a los cuales el software empaquetado apunta.
<ul style="list-style-type: none"> • Cantidad de técnicas de elicitación utilizadas 	
<ul style="list-style-type: none"> • Cantidad de requerimientos candidatos identificados 	
<ul style="list-style-type: none"> • Cantidad de veces que se tuvo que relevar requerimientos del mismo <i>stakeholder</i> (debido a que los requerimientos candidatos identificados no fueron entendidos) 	
<ul style="list-style-type: none"> • Cantidad de puntos de vistas elicitados 	
	<ul style="list-style-type: none"> • Cantidad de requerimientos reutilizados

Software a Medida	Software Empaquetado
<ul style="list-style-type: none"> Cantidad de requerimientos bajo línea base 	
<ul style="list-style-type: none"> Cantidad de requerimientos candidatos realmente implementados 	
<ul style="list-style-type: none"> Cantidad de requerimientos no identificados durante el proceso de elicitación 	
<ul style="list-style-type: none"> Cantidad de requerimientos no identificados durante el proceso de desarrollo de requerimientos 	
<ul style="list-style-type: none"> Cantidad de páginas del documento de requerimientos 	
<ul style="list-style-type: none"> Nivel de satisfacción del cliente 	
<ul style="list-style-type: none"> Nivel de satisfacción del desarrollador o equipo desarrollador respecto del proceso de elicitación 	
<ul style="list-style-type: none"> Nivel de satisfacción del desarrollador o equipo desarrollador respecto del producto entregado. 	

Métricas indirectas

Tabla 6: Propuesta de métricas indirectas

Software a Medida	Software Empaquetado
<ul style="list-style-type: none"> Porcentaje de requerimientos funcionales candidatos identificados del total de requerimientos candidatos identificados 	
<ul style="list-style-type: none"> Porcentaje de requerimientos de calidad candidatos identificados del total de requerimientos candidatos identificados 	
<ul style="list-style-type: none"> Porcentaje de tiempo utilizado en la etapa de elicitación de requerimientos del total del tiempo de la etapa de requerimientos 	
<ul style="list-style-type: none"> Porcentaje de tiempo utilizado en la etapa de desarrollo de requerimientos del total del tiempo del proyecto 	
<ul style="list-style-type: none"> Porcentaje de esfuerzo utilizado en la etapa de elicitación de requerimientos del total del esfuerzo de la etapa de requerimientos 	
<ul style="list-style-type: none"> Porcentaje de esfuerzo utilizado en la etapa de desarrollo de requerimientos del total del esfuerzo del proyecto 	

Trabajos Futuros

Como se demostró, la etapa de desarrollo de requerimientos, y más precisamente el proceso de elicitación de los mismos, es un paso crítico y muy importante para el desarrollo exitoso de un proyecto de software, tanto de software empaquetado como a medida, sin embargo aún no se encuentra dicho proceso totalmente refinado para lograr un resultado de calidad.

Es por ello, que es necesaria la implementación de la guía de elicitación propuesta a una gama de diversos tipos de proyectos para demostrar su capacidad de adaptación a la alternativa de software elegida, ya que esta guía hace énfasis en las buenas prácticas recomendadas para la elicitación de requerimientos, la selección más cuidadosa de las técnicas de elicitación a utilizar y a la recolección e identificación de métricas adecuadas para lograr una mejora continua en este proceso.

Anexos

Anexo I Encuesta a desarrolladores

INTRODUCCION

Esta encuesta tiene como objetivo estudiar y comparar el proceso de elicitation de requerimientos en dos contextos: el desarrollo de software a medida y el desarrollo de software empaquetado.

La hipótesis de este trabajo es que el contexto definido por el tipo de software a desarrollar (a medida o empaquetado) modifica el proceso de Elicitación.

Esta encuesta ha sido confeccionada como parte del trabajo de tesis realizado para la Maestría en Ingeniería de Software de la Universidad Nacional de la Plata. Los datos obtenidos de la misma serán utilizados anónimamente (se presentarán resultados genéricos sin enlace a su fuente). Igualmente la encuesta posee una sección opcional con datos del encuestado (Nombre, e-mail y Empresa) para tener un punto de contacto con el mismo en caso de ser necesario sólo para fines de aclaraciones relacionadas a esta encuesta. Como parte de estos datos opcionales usted podrá marcar la opción de recibir una copia o link al lugar donde se encuentre publicado el trabajo final de esta tesis.

La encuesta consta de 3 (tres) pagina a ser llenadas. El tiempo promedio para el llenado de la encuesta es de 10 minutos.

Por cualquier duda relacionada con esta encuesta no dude en contactarme via e-mail a natalia.andriano@gmail.com.

Siguiente >>

Proceso de Elicitacion de Requerimientos

Agradecimientos: A Diego Rubio

Autor: Natalia Andriano

Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Figura 52: Encuesta desarrolladores – Página 1

19. Utiliza alguna métrica para el proceso de Elicitación de Requerimientos en particular? Cual/es?
 Sí No No Sabe
20. Tomaría alguna otra métrica? Cual/es? Por qué?
 Sí No No Sabe
21. Utiliza algún método de medición para determinar el grado de satisfacción de los usuarios? Cual/es?
 Sí No No Sabe
22. Qué impacto considera usted, que tiene el grado de satisfacción de los usuarios, en su empresa?
 Muy Alta Alta Media Baja Muy Baja
23. Según su percepción, evalúe la manera en que se lleva a cabo el proceso de Elicitación de Requerimientos en su empresa, siendo 10 la mejor evaluación?
 0 1 2 3 4 5 6 7 8 9 10

Con respecto a la forma en la que se desarrolla software...

24. Al momento de desarrollar un software, cuales de las siguientes etapas se tienen en cuenta?
 Requerimientos Analisis
 Diseño Codificación
 Testing Ninguna
25. Cómo surge la idea del desarrollo de un proyecto en general?
 Petición de un cliente en particular Estudio de Mercado
 Necesidad Interna (Gerencia) Necesidad Interna (Desarrolladores)
 Otra
26. ¿Utiliza alguna de estas técnicas de elicitación de requerimientos?

Encuesta	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
JAD (Joint Application Design)	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Tormenta de Ideas (BrainStorming)	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
RAD (Rapid Application Development)	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Prototipación	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Win Win	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Entrevistas	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Observación	<input type="radio"/> SI	<input type="radio"/> NO	<input type="radio"/> No Conoce
Ninguna	<input type="radio"/> SI		
27. ¿La/s técnica/s de elicitación usada/s esta/h:
 Organizacionalment definida Se define para cada proyecto en particular
28. ¿Qué tan importante considera usted que es la etapa de requerimientos para el desarrollo de software?
 Muy Importante Importante Poco importante No tiene sentido realizarla

Siguiente >>

Figura 53: Encuesta desarrolladores – Página 2

Con respecto a las técnicas...

Para el llenado de la siguiente etapa "Con respecto a las técnicas..." de la encuesta, deberá pensar en la manera en que se realiza la elicitación de requerimientos dentro de su empresa, sin tener en cuenta una técnica de elicitación en particular.

1. Se identifican las clases de clientes?
 Sí No No Sabe
2. Realiza la planificación de las actividades?
 Sí No No Sabe
3. Se realiza un resumen luego de cada actividad realizada?
 Sí No No Sabe
4. ¿En el caso de que se realice dicho resumen, se utiliza algún modelo definido formalmente?
 Sí No No Sabe
5. ¿Tuvo alguna vez problemas para que el cliente entienda la actividad que se estaba realizando?
 Sí No No Sabe
6. ¿Cómo se asignan las personas para la obtención de los requerimientos?
 Rol Antigüedad en la empresa Al azar Experiencia Otro

Para el llenado de las siguientes etapas correspondientes a cada una de las técnicas de elicitación de requerimientos identificadas en esta tesis deber tener en cuenta lo siguiente:

En el caso de que la técnica de elicitación mencionada (Ej: Entrevistas) no sea utilizada por su empresa, o bien, usted no la conozca, por favor, saltee la técnica

ENTREVISTAS

7. ¿Cuáles de los siguientes pasos usted sigue al momento de realizar una entrevista?
 - a. Se planifica el tipo de entrevistas a realizar? Sí No No Sabe
 - b. Qué tipo de entrevistas realizan generalmente? Estructuradas Semiestructuradas No Sabe
 - c. Se planifican los tiempos para cada entrevista? Sí No No Sabe
 - d. Se planifican los tiempos entre entrevistas? Sí No No Sabe

TORMENTAS DE IDEAS

8. ¿Cuáles de los siguientes pasos usted sigue al momento de realizar una Tormenta de Ideas?
- a. Los objetivos de la Tormenta de Ideas se definen previamente? Sí No No Sabe
- b. Quién define los objetivos? Gerencia El equipo desarrollador Usuarios
- c. Existe un moderador que vaya guiando el procedimiento? Sí No No Sabe
- d. Los resultados de la Tormenta de Ideas son formalmente comunicados a todos los afectados? Sí No No Sabe

PROTOTIPACIÓN

9. ¿Cuáles de los siguientes pasos usted sigue al momento de realizar una Prototipación?
- Entrevistas Tormenta de Ideas RAD
- a. ¿Qué método de elicitación de requerimientos utiliza usted generalmente para obtener un entendimiento a priori del producto a desarrollar? JAD Observación Encuestas
- Win Win No utilizo ninguna
- b. Utiliza algún método para decidir qué requerimientos prototipar? Sí No No Sabe
- Cual?
- c. Solicita la participación de un técnico para realizar la prototipación? Sí No No Sabe

OBSERVACION

10. ¿Cuáles de los siguientes pasos usted sigue al momento de realizar una Observación?
- a. Realiza una planificación (tiempos) para la observación de las distintas personas? Sí No No Sabe
- b. ¿Qué tan ciertos fueron, según su percepción, los requerimientos obtenidos? Muy Certos Certos Usuarios
- Entrevistas Tormenta de Ideas RAD
- c. ¿Qué método de elicitación de requerimientos utiliza usted generalmente para obtener un entendimiento a priori del producto a desarrollar? JAD Observación Encuestas
- Win Win No utilizo ninguna

ENCUESTA

11. ¿Cuáles de los siguientes pasos usted sigue al momento de realizar una Encuesta?
- a. Se diseñan las encuestas teniendo en cuenta el rol de la persona? Sí No No Sabe
- b. Tiene en cuenta la cantidad de personas que pueden contestar las encuestas? Sí No No Sabe
- c. Las preguntas dentro de la encuesta son: Abiertas Cerradas Ambas
- d. Las encuestas son prototipadas antes de ser entregadas Sí No No Sabe
- e. Las encuestas son probadas antes de ser entregadas Sí No No Sabe
12. ¿En qué área trabaja?
- Requerimientos Análisis Diseño Codificación Testing

Siguiente >>

Proceso de Elicitación de Requerimientos

Agradecimientos: A Diego Rubio

Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Autor: Natalia Andriano

Figura 54: Encuesta desarrolladores – Página 3

Sugerencias y Observaciones

Observaciones y/o Comentarios:

Escriba a continuación cualquier observación o comentario en relación a la totalidad de la encuesta:

Sugerencias:

Escriba a continuación cualquier sugerencia que posea en relación al trabajo:

Finalizar >>

Proceso de Elicitación de Requerimientos

Agradecimientos: A Diego Rubio

Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Autor: Natalia Andriano

Figura 55: Encuesta desarrolladores - Página 4

Anexo II Encuesta a usuarios

INTRODUCCION

Esta encuesta tiene como objetivo estudiar y comparar el proceso de elicitation de requerimientos en dos contextos: el desarrollo de software a medida y el desarrollo de software empaquetado.

La hipótesis de este trabajo es que el contexto definido por el tipo de software a desarrollar (a medida o empaquetado) modifica el proceso de Elicitacion.

Esta encuesta ha sido confeccionada como parte del trabajo de tesis realizado para la Maestría en Ingeniería de Software de la Universidad Nacional de la Plata. Los datos obtenidos de la misma serán utilizados anónimamente (se presentarán resultados genéricos sin enlace a su fuente). Igualmente la encuesta posee una sección opcional con datos del encuestado (Nombre, e-mail y Empresa) para tener un punto de contacto con el mismo en caso de ser necesario sólo para fines de aclaraciones relacionadas a esta encuesta. Como parte de estos datos opcionales usted podrá marcar la opción de recibir una copia o link al lugar donde se encuentre publicado el trabajo final de esta tesis.

Por cualquier duda relacionada con esta encuesta no dude en contactarme via e-mail a natalia.andriano@gmail.com.

Siguiente >>

Proceso de Elicitacion de Requerimientos

Agradecimientos: A Diego Rubio

Autor: Natalia Andriano

Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Figura 56: Encuesta usuarios – Página 1

Datos Generales:

DATOS OPCIONALES:	
Nombre y Apellido:	<input style="width: 95%;" type="text"/>
Empresa Actual:	<input style="width: 95%;" type="text"/>
E-mail:	<input style="width: 95%;" type="text"/>
Edad:	<input style="width: 30%;" type="text"/>
Genero:	<input type="radio"/> Femenino <input type="radio"/> Masculino
Que cantidad de software empaquetado ha utilizado durante toda su vida	<input style="width: 30%;" type="text"/>
Que cantidad de software a medida ha utilizado durante toda su vida	<input style="width: 30%;" type="text"/>
Posicionese en una de las siguientes categorias	<input type="radio"/> Estudiante Universitario de Grado <input type="radio"/> Estudiante de Post Grado <input type="radio"/> Gerente de empresas <input type="radio"/> Empleado de empresas que utilizan software <input type="radio"/> Profesionales varios
Deseo obtener una copia o link al trabajo final obtenido a partir de esta encuesta: <input type="checkbox"/>	

Para el llenado de esta encuesta, el usuario deberá tener en cuenta para la sección "SOFTWARE A MEDIDA" el desarrollo de un software propio para la empresa, y para la sección "SOFTWARE EMPAQUETADO" la adquisición o compra de un producto enlatado.

SOFTWARE EMPAQUETADO

1. Indique su preferencia por un software empaquetado y no un software a medida
 - Menores Costos
 - Mayor Funcionalidad
 - Falta de tiempo para el desarrollo de un software desde cero
 - Falta de personal para asignar a la definición del software a medida
 - No confía en los software desarrollados a medida, prefiere un software ya probado y testeado
 - Nunca use software empaquetado
 - Otro:

2. Como se entera de la funcionalidad que este tipo de software prestan?
 - Por la competencia A través de publicidades
 - Recomendación Investigación Propia
 - Otro

3. ¿En que medida este tipo de software cumple con sus expectativas?
 - Muy Alta Alta Media Baja Muy Baja

4. Con la compra del producto, se le proporciona algún servicios de mantenimiento?
 - Si No No Sabe

5. Indique el tipo de mantenimiento que brindan
 - Correctivo (cuando el software falla, los errores son corregidos) Perfectivo (en el caso que usted necesita una funcionalidad extra que el software no la proporciona) Adaptativo (en el caso que el software este diseñado para correr sobre un sistema operativo que usted no posee)

6. Le gustaria poseer ese tipo de servicio (mantenimiento) con la compra de un software empaquetado?
 - Si No Solo algunos

7. Utilizo alguna vez algun software empaquetado que necesito ser adaptado a sus necesidades?
 - Si No No Sabe

8. Si la respuesta es SI a la pregunta anterior, de que manera el proveedor relevo los requerimientos de cambio?
 - Entrevistas: se realizaron reuniones formales, en persona, para obtener informacion sobre la funcionalidad a ser desarrollada
 - JAD: El equipo desarrollador trabajo en conjunto con Usted para realizar el diseno del software
 - Tormenta de Ideas: de llevaron a cabo reuniones para presentar ideas de cuales serian los cambios a introducir en el producto
 - Win Win: se mantuvieron negociaciones con el equipo desarrollador, con respecto a la funcionalidad, costo, fecha de entrega, etc. y lograron establecer una serie de condiciones tanto para la empresa como para el equipo desarrollador
 - RAD: una persona de la empresa trabajo en conjunto con el equipo desarrollador durante el desarrollo del producto
 - Encuestas: se entregaron cuestionarios a distintas personas de la empresa, preguntando sobre la nueva funcionalidad requerida
 - Observacion: una persona del equipo desarrollador se dedico a observar el trabajo dentro de la empresa
 - Prototipacion: antes de comenzar con el desarrollo del producto, el equipo desarrollador entrego a la empresa un prototipo para validar los cambios a realizar
 - Otra

9. Se implementaron todos los cambios que usted propuso?
- Si No No Sabe
10. Los cambios al software se entregaron en la fecha pactada?
- Si No No Sabe
11. A que factores atribuye esta demora?
- Los requerimientos de cambio no fueron correctamente relevados
- Muchos cambios en los requerimientos
- Se produjeron cambios en la tecnología utilizada en la empresa
- Otro
12. Considera usted que se existe un buen servicio post-compra de software empaquetado?
- Siempre Nunca Algunas veces
13. Que relacion mantiene con la empresa que le vendio el software?
- No mantengo ninguna relacion La empresa sigue manteniendo mi software
14. Participo alguna vez en la definicion de la funcionalidad de un software empaquetado?
- Si No No Sabe
15. Participo alguna vez en el testing de un software empaquetado?
- Si No No Sabe

SOFTWARE A MEDIDA

16. Indique su preferencia por un software a medida y no un software empaquetado
- Menores Costos
- Mayor Funcionalidad
- No quiso arriesgarse a que el software empaquetado no se adaptara a sus necesidades
- No existe ningun software empaquetado util para sus necesidades
- No confiaba en los software empaquetados, prefiere definir usted mismo la funcionalidad del software
- Otro
17. Como se definieron el/los objetivo/s y los alcances del producto a desarrollar?
- Necesidad de agilizar-automatizar las tareas para hacerlas mas efectivas
- Surgieron de asesores calificados
- Surgieron de potenciales usuarios
- Otro
18. Se relevaron los requerimientos para el software a desarrollar?
- Si No No Sabe
19. En el caso de haber respondido SI en la pregunta anterior, que metodo utilizo el equipo desarrollador para la obtencion de dichos requerimientos?
- Entrevistas: se realizaron reuniones formales, en persona, para obtener informacion sobre la funcionalidad a ser desarrollada
- JAD: El equipo desarrollador trabajo en conjunto con Usted para realizar el diseno del software
- Tormenta de Ideas: de llevaron a cabo reuniones para presentar ideas de cuales serian los cambios a introducir en el producto
- Win Win: se mantuvieron negociaciones con el equipo desarrollador, con respecto a la funcionalidad, costo, fecha de entrega, etc. y lograron establecer una serie de condiciones tanto para la empresa como para el equipo desarrollador
- RAD: una persona de la empresa trabajo en conjunto con el equipo desarrollador durante el desarrollo del producto
- Encuestas: se entregaron cuestionarios a distintas personas de la empresa, preguntando sobre la nueva funcionalidad requerida
- Observacion: una persona del equipo desarrollador se dedico a observar el trabajo dentro de la empresa
- Prototipacion: antes de comenzar con el desarrollo del producto, el equipo desarrollador entrego a la empresa un prototipo para validar los cambios a realizar
- Otra
20. El software desarrollado cumple con los requerimientos establecidos?
- Si No No Sabe
21. El software se entrego en tiempo y forma segun lo planeado?
- Si No No Sabe
22. El costo del software fue segun lo pactado?
- Si No No Sabe
23. El software debio ser corregido por fallas ("el programa se clavaba"), despues de ser instalado?
- Si No No Sabe

24. El software debio ser corregido debido a alguna funcionalidad mal definida (ejemplo: calculo erroneo)?
 Si No No Sabe
25. Segun su percepcion, que porcentaje del sistema definido se esta usando actualmente?
>
 0-10% 11-20% 21-30% 31-40% 41-50% 51-60% 61-70% 71-80% 81-90% 91-100%
26. Se sintio parte del desarrollo del producto?
 Si No No Sabe
27. Le hubiese gustado participar de forma mas activa durante el desarrollo del producto?
 Si No No Sabe

Siguiente >>

Proceso de Elicitacion de Requerimientos Autor: Natalia Andriano
Agradecimientos: A Diego Rubio
Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Figura 57: Encuesta usuarios – Página 2

Sugerencias y Observaciones

Observaciones y/o Comentarios:

Escriba a continuación cualquier observación o comentario en relación a la totalidad de la encuesta:

Sugerencias:

Escriba a continuación cualquier sugerencia que posea en relación al trabajo:

Finalizar >>

Proceso de Elicitacion de Requerimientos Autor: Natalia Andriano
Agradecimientos: A Diego Rubio
Trabajo realizado como parte de los requisitos para la obtención del título de Magíster en Ingeniería de Software de la Facultad de Informática de la Universidad Nacional de La Plata.

Figura 58: Encuesta usuarios - Página 3

Anexo III: Métricas para el proceso de administración de requerimientos

	Direct/ Indirect	Measure	Scale	Entity
1	Indirect	# Affected groups and individuals about NOC	Ratio	Change request management process
2	Direct	# Base lined requirements.	Ratio	Baseline management process
3	Direct	# Changes per requirement.	Ratio	Change request management process
4	Direct	# Changes to requirements approved.	Ratio	Change request management process
5	Direct	# Changes to requirements incorporated into base line.	Ratio	Change request management process
6	Direct	# Changes to requirements open.	Ratio	Change request management process
7	Indirect	# Changes to requirements per unit of time.	Ratio	Change request management process
8	Direct	# Changes to requirements proposed.	Ratio	Change request management process
9	Direct	# Changes to requirements rejected.	Ratio	Change request management process
10	Direct	# Documents affected by a change.	Ratio	Change request management process, Requirements documentation process
11	Direct	# Final requirements.	Ratio	Requirement management process
12	Direct	# Incomplete requirements.	Ratio	Requirement management process
13	Direct	# Inconsistencies.	Ratio	Requirement management process
14	Direct	# Inconsistent requirements.	Ratio	Requirement management process
15	Direct	# Initial requirements.	Ratio	Requirement management process
16	Direct	# Missing requirements.	Ratio	Requirement management process
17	Direct	# Requirements affected by a change.	Ratio	Requirement management process

18	Indirect	# Requirements schedule for each software build or release.	Ratio	Requirement management process
19	Direct	# TBDs in requirements specifications	Ratio	Requirements specification
20	Indirect	# TBDs per unit of time	Ratio	Requirements specification
21	Direct	# Test case per requirement	Ratio	Requirement management process
22	Indirect	Cost of change to requirement	Ratio	Change request management
23	Indirect	Effort expended on requirements management activity	Ratio	Requirement management process
24	Direct	Kind of documentation	Nominal	Requirements specification process
25	Direct	Major source of request for a change to requirements	Nominal	Change request management
26	Direct	Notification of change (NOC) shall be documented and distributed as a key communication document.	Nominal	Change request management, Requirements documentation process
27	Direct	Phase when change requirements are base lined.	Ordinal	Baseline management process
28	Direct	Phase where change was requested.	Ordinal	Change request management
29	Direct	Reason of change to requirement.	Nominal	Requirement management process
30	Direct	Requirement type for each change to requirements	Nominal	Change request management
31	Direct	Size of a change to requirements.	Ratio	Change request management
32	Direct	Size of requirements	Ratio	Requirement management process
33	Direct	Status of each requirement.	Nominal	Requirement management process
34	Direct	Status of software plans, work products, and activities.	Nominal	Requirement management process
35	Indirect	The computer software configuration item(s) (CSCI) affected by a change to requirements	Nominal	Change request management, Requirement management process
36	Indirect	Time spent in upgrading	Ratio	Requirement management process
37	Direct	Total # Requirements	Ratio	Requirement management process
38	Direct	Type of change to requirements	Nominal	Change request management

Figura 59: Listado de métricas para el proceso de administración de requerimientos [Loconsole:2003].

Anexo IV: Buenas prácticas utilizadas. Salidas de los análisis Chi-cuadrado para cada practica en función del tipo de software

1) SE EVALUA LA FACTIBILIDAD TECNICA DEL SISTEMA?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P10	1	2	Total
1	28	4	32
2	7	0	7
3	7	5	12
Total	42	9	51

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P10	1	2	Total
1	87,50	12,50	100,00
2	100,00	0,00	100,00
3	58,33	41,67	100,00
Total	82,35	17,65	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P10	1	2	Total
1	66,67	44,44	62,75
2	16,67	0,00	13,73
3	16,67	55,56	23,53
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P10	1	2	Total
1	54,90	7,84	62,75
2	13,73	0,00	13,73
3	13,73	9,80	23,53
Total	82,35	17,65	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	6,85	2	0,0326
Chi Cuadrado MV-G2	7,12	2	0,0285
Coef.Conting.Cramer	0,26		
Coef.Conting.Pearson	0,34		

2) SE TIENEN EN CUENTA CONSIDERACIONES POLITICAS Y ORGANIZACIONALES?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P11	1	2	Total
1	29	6	35
2	7	0	7
3	6	3	9
Total	42	9	51

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P11	1	2	Total
1	82,86	17,14	100,00
2	100,00	0,00	100,00
3	66,67	33,33	100,00
Total	82,35	17,65	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P11	1	2	Total
1	69,05	66,67	68,63
2	16,67	0,00	13,73
3	14,29	33,33	17,65
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P11	1	2	Total
1	56,86	11,76	68,63
2	13,73	0,00	13,73
3	11,76	5,88	17,65
Total	82,35	17,65	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	3,03	2	0,2198
Chi Cuadrado MV-G2	4,00	2	0,1350
Coef.Conting.Cramer	0,17		
Coef.Conting.Pearson	0,24		

3) SE REGISTRA EL ORIGEN DE LOS REQUERIMIENTOS?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P12	1	2	Total
1	27	6	33
2	9	1	10
3	6	2	8
Total	42	9	51

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P12	1	2	Total
1	81,82	18,18	100,00
2	90,00	10,00	100,00
3	75,00	25,00	100,00
Total	82,35	17,65	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P12	1	2	Total
1	64,29	66,67	64,71
2	21,43	11,11	19,61
3	14,29	22,22	15,69
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P12	1	2	Total
1	52,94	11,76	64,71
2	17,65	1,96	19,61
3	11,76	3,92	15,69
Total	82,35	17,65	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	0,71	2	0,7024
Chi Cuadrado MV-G2	0,74	2	0,6908
Coef.Conting.Cramer	0,08		
Coef.Conting.Pearson	0,12		

4) SE TIENEN EN CUENTA LAS RESTRICCIONES DEL AMBIENTE?
Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P13	1	2	Total
1	37	4	41
2	5	1	6
3	0	4	4
Total	42	9	51

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P13	1	2	Total
1	90,24	9,76	100,00
2	83,33	16,67	100,00
3	0,00	100,00	100,00
Total	82,35	17,65	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P13	1	2	Total
1	88,10	44,44	80,39
2	11,90	11,11	11,76
3	0,00	44,44	7,84
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P13	1	2	Total
1	72,55	7,84	80,39
2	9,80	1,96	11,76
3	0,00	7,84	7,84
Total	82,35	17,65	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	20,43	2	<0,0001
Chi Cuadrado MV-G2	15,91	2	0,0004
Coef.Conting.Cramer	0,45		
Coef.Conting.Pearson	0,53		

5) SE REGISTRA LA RAZON O IMPORTANCIA DE LOS REQUERIMIENTOS?
Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P14	1	2	Total
1	27	4	31
2	11	2	13
3	3	3	6
Total	41	9	50

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P14	1	2	Total
1	87,10	12,90	100,00
2	84,62	15,38	100,00
3	50,00	50,00	100,00
Total	82,00	18,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P14	1	2	Total
1	65,85	44,44	62,00
2	26,83	22,22	26,00
3	7,32	33,33	12,00
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)
En columnas:P1

P14	1	2	Total
1	54,00	8,00	62,00
2	22,00	4,00	26,00
3	6,00	6,00	12,00
Total	82,00	18,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	4,77	2	0,0922
Chi Cuadrado MV-G2	3,82	2	0,1483
Coef.Conting.Cramer	0,22		
Coef.Conting.Pearson	0,30		

6) SE RECOLECTAN REQUERIMIENTOS DESDE MULTIPLES PUNTOS DE VISTA?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P15	1	2	Total
1	14	1	15
2	21	3	24
3	6	5	11
Total	41	9	50

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P15	1	2	Total
1	93,33	6,67	100,00
2	87,50	12,50	100,00
3	54,55	45,45	100,00
Total	82,00	18,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P15	1	2	Total
1	34,15	11,11	30,00
2	51,22	33,33	48,00
3	14,63	55,56	22,00
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P15	1	2	Total
1	28,00	2,00	30,00
2	42,00	6,00	48,00
3	12,00	10,00	22,00
Total	82,00	18,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	7,41	2	0,0245
Chi Cuadrado MV-G2	6,55	2	0,0378
Coef.Conting.Cramer	0,27		
Coef.Conting.Pearson	0,36		

7) SE REUTILIZAN REQUERIMIENTOS

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P16	1	2	Total
1	14	8	22
2	19	0	19
3	7	1	8
Total	40	9	49

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P16	1	2	Total
1	63,64	36,36	100,00
2	100,00	0,00	100,00
3	87,50	12,50	100,00
Total	81,63	18,37	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P16	1	2	Total
1	35,00	88,89	44,90
2	47,50	0,00	38,78
3	17,50	11,11	16,33
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P16	1	2	Total
1	28,57	16,33	44,90
2	38,78	0,00	38,78
3	14,29	2,04	16,33
Total	81,63	18,37	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	9,21	2	0,0100
Chi Cuadrado MV-G2	11,87	2	0,0026
Coef.Conting.Cramer	0,31		
Coef.Conting.Pearson	0,40		

8) SE IDENTIFICAN LAS CLASES DE CLIENTES?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P29	1	2	Total
1	27	4	31
2	9	1	10
3	4	1	5
Total	40	6	46

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P29	1	2	Total
1	87,10	12,90	100,00
2	90,00	10,00	100,00
3	80,00	20,00	100,00
Total	86,96	13,04	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P29	1	2	Total
1	67,50	66,67	67,39
2	22,50	16,67	21,74
3	10,00	16,67	10,87
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P29	1	2	Total
1	58,70	8,70	67,39
2	19,57	2,17	21,74
3	8,70	2,17	10,87
Total	86,96	13,04	100,00

Estadístico	Valor	gl	p
-------------	-------	----	---

Chi Cuadrado Pearson	0,30	2	0,8626
Chi Cuadrado MV-G2	0,28	2	0,8710
Coef.Conting.Cramer	0,06		
Coef.Conting.Pearson	0,08		

9) REALIZA UNA PLANIFICACION DE LAS ACTIVIDADES CORRESPONDIENTES AL USO DE LAS TECNICAS DE ELICITACION?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P30	1	2	Total
1	35	4	39
2	5	2	7
Total	40	6	46

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P30	1	2	Total
1	89,74	10,26	100,00
2	71,43	28,57	100,00
Total	86,96	13,04	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P30	1	2	Total
1	87,50	66,67	84,78
2	12,50	33,33	15,22
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P30	1	2	Total
1	76,09	8,70	84,78
2	10,87	4,35	15,22
Total	86,96	13,04	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	1,76	1	0,1852
Chi Cuadrado MV-G2	1,45	1	0,2278
Irwin-Fisher bilatera..	0,18		0,2214
Coef.Conting.Cramer	0,14		
Coef.Conting.Pearson	0,19		
Coeficiente Phi	0,20		

10) SE REALIZA UN RESUMEN LUEGO DE CADA ACTIVIDAD?

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P31	1	2	Total
1	22	4	26
2	16	2	18
3	1	0	1
Total	39	6	45

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P31	1	2	Total
1	84,62	15,38	100,00
2	88,89	11,11	100,00
3	100,00	0,00	100,00
Total	86,67	13,33	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P31	1	2	Total
1			
2			
3			
Total			

1	56,41	66,67	57,78
2	41,03	33,33	40,00
3	2,56	0,00	2,22
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P31	1	2	Total
1	48,89	8,89	57,78
2	35,56	4,44	40,00
3	2,22	0,00	2,22
Total	86,67	13,33	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	0,33	2	0,8498
Chi Cuadrado MV-G2	0,46	2	0,7953
Coef.Conting.Cramer	0,06		
Coef.Conting.Pearson	0,08		

Anexo V: Análisis de cumplimiento

1. El software desarrollado cumple con los requerimientos es Tablados?

Tablas de contingencia

Frecuencias absolutas

P20	FA	(%)
1,00	19	51,35
2,00	9	24,32
3,00	9	24,32
Total	37	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P20	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P20	FR	Total
1,00	51,35	51,35
2,00	24,32	24,32
3,00	24,32	24,32
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P20	FR	Total
1,00	51,35	51,35
2,00	24,32	24,32
3,00	24,32	24,32
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	5,41	2	0,0670
Chi Cuadrado MV-G2	5,08	2	0,0789
Coef.Conting.Cramer	0,38		
Coef.Conting.Pearson	0,36		

2. El software se entregó en tiempo y forma según lo planeado?

Tablas de contingencia

Frecuencias absolutas

P211	FA	(%)
1,00	17	42,50
2,00	9	22,50
3,00	11	27,50
99,00	3	7,50
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P211	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P211	FR	Total
1,00	42,50	42,50
2,00	22,50	22,50
3,00	27,50	27,50
99,00	7,50	7,50

Total 100,00 100,00

Frecuencias relativas al total (expresadas en porcentajes)

P211	FR	Total
1,00	42,50	42,50
2,00	22,50	22,50
3,00	27,50	27,50
99,00	7,50	7,50
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	10,00	3	0,0186
Chi Cuadrado MV-G2	11,02	3	0,0116
Coef.Conting.Cramer	0,50		
Coef.Conting.Pearson	0,45		

3. El costo del software fue lo pactado?

Tablas de contingencia

Frecuencias absolutas

P221	FA	(%)
1,00	15	37,50
2,00	12	30,00
3,00	10	25,00
99,00	3	7,50
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P221	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P221	FR	Total
1,00	37,50	37,50
2,00	30,00	30,00
3,00	25,00	25,00
99,00	7,50	7,50
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P221	FR	Total
1,00	37,50	37,50
2,00	30,00	30,00
3,00	25,00	25,00
99,00	7,50	7,50
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	7,80	3	0,0503
Chi Cuadrado MV-G2	9,32	3	0,0254
Coef.Conting.Cramer	0,44		
Coef.Conting.Pearson	0,40		

4. El software debió ser corregido debido a fallas?

Tablas de contingencia

Frecuencias absolutas

P231	FA	(%)
1,00	21	52,50
2,00	8	20,00
3,00	9	22,50

99,00	2	5,00
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P231	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P231	FR	Total
1,00	52,50	52,50
2,00	20,00	20,00
3,00	22,50	22,50
99,00	5,00	5,00
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P231	FR	Total
1,00	52,50	52,50
2,00	20,00	20,00
3,00	22,50	22,50
99,00	5,00	5,00
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	19,00	3	0,0003
Chi Cuadrado MV-G2	19,26	3	0,0002
Coef.Conting.Cramer	0,69		
Coef.Conting.Pearson	0,57		

5. El software debió ser corregido debido a funcionalidad mal definida?

Tablas de contingencia

Frecuencias absolutas

P241	FA	(%)
1,00	16	40,00
2,00	12	30,00
3,00	10	25,00
99,00	2	5,00
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P241	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P241	FR	Total
1,00	40,00	40,00
2,00	30,00	30,00
3,00	25,00	25,00
99,00	5,00	5,00
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P241	FR	Total
1,00	40,00	40,00
2,00	30,00	30,00
3,00	25,00	25,00
99,00	5,00	5,00
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	10,40	3	0,0155
Chi Cuadrado MV-G2	12,98	3	0,0047
Coef. Conting. Cramer	0,51		
Coef. Conting. Pearson	0,45		

6. Se sintió parte del desarrollo del producto?

Tablas de contingencia

Frecuencias absolutas

P26	FA	(%)
1,00	14	35,00
2,00	19	47,50
3,00	5	12,50
99,00	2	5,00
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P26	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P26	FR	Total
1,00	35,00	35,00
2,00	47,50	47,50
3,00	12,50	12,50
99,00	5,00	5,00
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P26	FR	Total
1,00	35,00	35,00
2,00	47,50	47,50
3,00	12,50	12,50
99,00	5,00	5,00
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	18,60	3	0,0003
Chi Cuadrado MV-G2	20,44	3	0,0001
Coef. Conting. Cramer	0,68		
Coef. Conting. Pearson	0,56		

7. Le hubiese gustado participar de forma más activa durante el proceso de desarrollo?

Tablas de contingencia

Frecuencias absolutas

P27	FA	(%)
1,00	17	42,50
2,00	14	35,00
3,00	7	17,50
99,00	2	5,00
Total	40	100,00

Frecuencias relativas por filas (expresadas en porcentajes)

P27	FR	Total
1,00	100,00	100,00
2,00	100,00	100,00
3,00	100,00	100,00
99,00	100,00	100,00
Total	100,00	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

P27	FR	Total
1,00	42,50	42,50
2,00	35,00	35,00
3,00	17,50	17,50
99,00	5,00	5,00
Total	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

P27	FR	Total
1,00	42,50	42,50
2,00	35,00	35,00
3,00	17,50	17,50
99,00	5,00	5,00
Total	100,00	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	13,80	3	0,0032
Chi Cuadrado MV-G2	16,03	3	0,0011
Coef.Conting.Cramer	0,59		
Coef.Conting.Pearson	0,51		

Anexo VI Técnicas de elicitación utilizadas según la alternativa de software

ENCUESTAS

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26A	1	2	Total
1	19	0	19
2	10	4	14
3	10	4	14
99	7	1	8
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26A	1	2	Total
1	100,00	0,00	100,00
2	71,43	28,57	100,00
3	71,43	28,57	100,00
99	87,50	12,50	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26A	1	2	Total
1	41,30	0,00	34,55
2	21,74	44,44	25,45
3	21,74	44,44	25,45
99	15,22	11,11	14,55
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26A	1	2	Total
1	34,55	0,00	34,55
2	18,18	7,27	25,45
3	18,18	7,27	25,45
99	12,73	1,82	14,55
Total	83,64	16,36	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	6,85	3	0,0767
Chi Cuadrado MV-G2	9,49	3	0,0234
Coef.Conting.Cramer	0,25		
Coef.Conting.Pearson	0,33		

JAD

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26B	1	2	Total
1	3	1	4
2	11	2	13
3	24	4	28
99	8	2	10
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26B	1	2	Total
------	---	---	-------

1	75,00	25,00	100,00
2	84,62	15,38	100,00
3	85,71	14,29	100,00
99	80,00	20,00	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26B	1	2	Total
1	6,52	11,11	7,27
2	23,91	22,22	23,64
3	52,17	44,44	50,91
99	17,39	22,22	18,18
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26B	1	2	Total
1	5,45	1,82	7,27
2	20,00	3,64	23,64
3	43,64	7,27	50,91
99	14,55	3,64	18,18
Total	83,64	16,36	100,00

	Estadístico	Valor	gl	p
Chi Cuadrado Pearson		0,41	3	0,9377
Chi Cuadrado MV-G2		0,39	3	0,9431
Coef.Conting.Cramer		0,06		
Coef.Conting.Pearson		0,09		

TORMENTA DE IDEAS

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26C	1	2	Total
1	27	3	30
2	7	1	8
3	6	3	9
99	6	2	8
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26C	1	2	Total
1	90,00	10,00	100,00
2	87,50	12,50	100,00
3	66,67	33,33	100,00
99	75,00	25,00	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26C	1	2	Total
1	58,70	33,33	54,55
2	15,22	11,11	14,55
3	13,04	33,33	16,36
99	13,04	22,22	14,55
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26C	1	2	Total
1	49,09	5,45	54,55
2	12,73	1,82	14,55
3	10,91	5,45	16,36

99	10,91	3,64	14,55
Total	83,64	16,36	100,00

	Estadístico	Valor	gl	p
Chi Cuadrado Pearson		3,30	3	0,3470
Chi Cuadrado MV-G2		3,03	3	0,3865
Coef. Conting. Cramer		0,17		
Coef. Conting. Pearson		0,24		

RAD

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26D	1	2	Total
1	5	1	6
2	15	1	16
3	17	4	21
99	9	3	12
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26D	1	2	Total
1	83,33	16,67	100,00
2	93,75	6,25	100,00
3	80,95	19,05	100,00
99	75,00	25,00	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26D	1	2	Total
1	10,87	11,11	10,91
2	32,61	11,11	29,09
3	36,96	44,44	38,18
99	19,57	33,33	21,82
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26D	1	2	Total
1	9,09	1,82	10,91
2	27,27	1,82	29,09
3	30,91	7,27	38,18
99	16,36	5,45	21,82
Total	83,64	16,36	100,00

	Estadístico	Valor	gl	p
Chi Cuadrado Pearson		1,96	3	0,5806
Chi Cuadrado MV-G2		2,19	3	0,5345
Coef. Conting. Cramer		0,13		
Coef. Conting. Pearson		0,19		

PROTOTIPACION

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26E	1	2	Total
1	22	5	27
2	8	0	8
3	8	3	11
99	8	1	9

Total 46 9 55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26E	1	2	Total
1	81,48	18,52	100,00
2	100,00	0,00	100,00
3	72,73	27,27	100,00
99	88,89	11,11	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26E	1	2	Total
1	47,83	55,56	49,09
2	17,39	0,00	14,55
3	17,39	33,33	20,00
99	17,39	11,11	16,36
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26E	1	2	Total
1	40,00	9,09	49,09
2	14,55	0,00	14,55
3	14,55	5,45	20,00
99	14,55	1,82	16,36
Total	83,64	16,36	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	2,79	3	0,4244
Chi Cuadrado MV-G2	3,98	3	0,2640
Coef.Conting.Cramer	0,16		
Coef.Conting.Pearson	0,22		

WIN WIN

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26F	1	2	Total
1	4	0	4
2	10	1	11
3	22	4	26
99	10	4	14
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26F	1	2	Total
1	100,00	0,00	100,00
2	90,91	9,09	100,00
3	84,62	15,38	100,00
99	71,43	28,57	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26F	1	2	Total
1	8,70	0,00	7,27
2	21,74	11,11	20,00
3	47,83	44,44	47,27
99	21,74	44,44	25,45
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26F	1	2	Total
------	---	---	-------

1	7,27	0,00	7,27
2	18,18	1,82	20,00
3	40,00	7,27	47,27
99	18,18	7,27	25,45
Total	83,64	16,36	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	2,75	3	0,4317
Chi Cuadrado MV-G2	3,24	3	0,3556
Coef.Conting.Cramer	0,16		
Coef.Conting.Pearson	0,22		

ENTREVISTAS

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26G	1	2	Total
1	21	0	21
2	15	3	18
3	10	6	16
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26G	1	2	Total
1	100,00	0,00	100,00
2	83,33	16,67	100,00
3	62,50	37,50	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26G	1	2	Total
1	45,65	0,00	38,18
2	32,61	33,33	32,73
3	21,74	66,67	29,09
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26G	1	2	Total
1	38,18	0,00	38,18
2	27,27	5,45	32,73
3	18,18	10,91	29,09
Total	83,64	16,36	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	9,33	2	0,0094
Chi Cuadrado MV-G2	11,63	2	0,0030
Coef.Conting.Cramer	0,29		
Coef.Conting.Pearson	0,38		

OBSERVACION

Tablas de contingencia

Frecuencias absolutas

En columnas:P1

P26H	1	2	Total
1	24	1	25
2	8	1	9
3	6	4	10
99	8	3	11
Total	46	9	55

Frecuencias relativas por filas (expresadas en porcentajes)

En columnas:P1

P26H	1	2	Total
1	96,00	4,00	100,00
2	88,89	11,11	100,00
3	60,00	40,00	100,00
99	72,73	27,27	100,00
Total	83,64	16,36	100,00

Frecuencias relativas por columnas (expresadas en porcentajes)

En columnas:P1

P26H	1	2	Total
1	52,17	11,11	45,45
2	17,39	11,11	16,36
3	13,04	44,44	18,18
99	17,39	33,33	20,00
Total	100,00	100,00	100,00

Frecuencias relativas al total (expresadas en porcentajes)

En columnas:P1

P26H	1	2	Total
1	43,64	1,82	45,45
2	14,55	1,82	16,36
3	10,91	7,27	18,18
99	14,55	5,45	20,00
Total	83,64	16,36	100,00

Estadístico	Valor	gl	p
Chi Cuadrado Pearson	8,01	3	0,0458
Chi Cuadrado MV-G2	7,99	3	0,0461
Coef.Conting.Cramer	0,27		
Coef.Conting.Pearson	0,36		

Anexo VII Entrevista realizada a desarrolladores

Desarrollo de software independiente SI NO

Alternativa de Software A Medida Empaquetad

Todas las preguntas deben responderlas pensando en un software. Pueden dar una cantidad o un rango (por ejemplo entre 5 y 10). Si no se animan a dar el rango pueden decidir entre cantidad baja, media, alta (y vos entra el datos como baja, media, alta y no con números así no se confunden con las otras cantidades)

Cantidad de personas elicadas	
Cantidad de <i>stakeholders</i> involucrados con el software	
Cantidad de veces que se tuvo que relevar requerimientos del mismo <i>stakeholders</i> (debido a que los requerimientos candidatos identificados no fueron entendidos)	
Cantidad de puntos de vistas elicados	
Cantidad de técnicas de elicación utilizadas	
Cantidad de requerimientos candidatos identificados	
Porcentaje de requerimientos candidatos funcionales	
Porcentaje de requerimientos candidatos de calidad	
Cantidad de inconsistencias (o incompletitudes) encontradas en requerimientos candidatos identificados teniendo en cuenta la misma técnica de elicación utilizada y diferente el usuario (para un mismo tipo de software)	
Cantidad de inconsistencias (o incompletitudes) encontradas en requerimientos candidatos identificados teniendo en cuenta distintas técnicas de elicación utilizada y el mismo usuario (para un mismo tipo de software)	

Cantidad de requerimientos reutilizados	
Cantidad de requerimientos bajo la línea base	
Cantidad de requerimientos candidatos realmente implementados	
Cantidad de requerimientos no identificados durante el proceso de elicitación	
Cantidad de requerimientos no identificados durante la etapa de desarrollo de requerimientos	
Porcentaje de esfuerzo utilizado en la etapa de elicitación respecto del total del proyecto	
Porcentaje de esfuerzo en toda la etapa de requerimiento respecto del total del proyecto	
Porcentaje de tiempo utilizado en la etapa de elicitación respecto del total del proyecto	
Porcentaje de tiempo en toda la etapa de requerimiento respecto del total del proyecto	
Cantidad de páginas del documento de requerimientos	
Nivel de satisfacción del cliente	<input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo
Nivel de satisfacción del desarrollador o equipo de desarrolladores respecto al proceso de elicitación de requerimientos	<input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo
Nivel de satisfacción del desarrollador o equipo de desarrolladores respecto al software desarrollado	<input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo

- Proponga una métrica _____

- Orden de mérito (1 – 5 siendo 1 la más importante):

Cantidad de Requerimientos candidatos identificados.	<input type="checkbox"/> Práctica <input type="checkbox"/> Impráctica
Porcentaje de requerimientos candidatos implementados	<input type="checkbox"/> Práctica <input type="checkbox"/> Impráctica
Porcentaje de tiempo utilizado en la etapa de elicitación de requerimientos respecto del total del proyecto	<input type="checkbox"/> Práctica <input type="checkbox"/> Impráctica
Nivel de satisfacción del cliente respecto al proceso de elicitation	<input type="checkbox"/> Práctica <input type="checkbox"/> Impráctica
Nivel de documentación del proceso de elicitación	<input type="checkbox"/> Práctica <input type="checkbox"/> Impráctica

Anexo VIII Análisis de varianza

Análisis de la varianza

Variable	N	R ²	R ² Aj	CV
Cantidad de personas	37	0.18	0.16	44.80

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	12.47	1	12.47	7.71	0.0087
Alternativa de softw	12.47	1	12.47	7.71	0.0087
Error	56.56	35	1.62		
Total	69.03	36			

Variable	N	R ²	R ² Aj	CV
Cantidad de stakehol	37	0.11	0.08	51.13

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	32.43	1	32.43	4.33	0.0448
Alternativa de softw	32.43	1	32.43	4.33	0.0448
Error	262.00	35	7.49		
Total	294.43	36			

Variable	N	R ²	R ² Aj	CV
cantidad de veces qu	28	0.25	0.22	66.01

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	214.74	1	214.74	8.44	0.0074
Alternativa de softw	214.74	1	214.74	8.44	0.0074
Error	661.68	26	25.45		
Total	876.43	27			

Variable	N	R ²	R ² Aj	CV
Cantidad de puntos d	34	4.9E-03	0.00	41.11

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	0.16	1	0.16	0.16	0.6927
Alternativa de softw	0.16	1	0.16	0.16	0.6927
Error	32.22	32	1.01		
Total	32.38	33			

Variable	N	R ²	R ² Aj	CV
Cantidad de tecnicas	34	0.03	0.00	37.97

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
------	----	----	----	---	---------

Modelo	0.42	1	0.42	0.86	0.3612
Alternativa de softw	0.42	1	0.42	0.86	0.3612
Error	15.84	32	0.50		
Total	16.26	33			

Variable	N	R ²	R ² Aj	CV
Cantidad de requerim	31	0.07	0.03	60.27

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	396.48	1	396.48	2.02	0.1661
Alternativa de softw	396.48	1	396.48	2.02	0.1661
Error	5697.45	29	196.46		
Total	6093.94	30			

Variable	N	R ²	R ² Aj	CV
Cantidad de inconsis	22	0.13	0.09	107.51

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	403.13	1	403.13	3.01	0.0984
Alternativa de softw	403.13	1	403.13	3.01	0.0984
Error	2682.74	20	134.14		
Total	3085.86	21			

Variable	N	R ²	R ² Aj	CV
Cantidad de inconsil	22	0.07	0.02	152.08

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	165.82	1	165.82	1.50	0.2346
Alternativa de softw	165.82	1	165.82	1.50	0.2346
Error	2208.00	20	110.40		
Total	2373.82	21			

Variable	N	R ²	R ² Aj	CV
Cantidad de requeril	22	0.07	0.02	154.45

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	79.13	1	79.13	1.46	0.2416
Alternativa de softw	79.13	1	79.13	1.46	0.2416
Error	1086.74	20	54.34		
Total	1165.86	21			

Variable	N	R ²	R ² Aj	CV
Cantidad de requeril2	25	0.07	0.03	87.77

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	472.95	1	472.95	1.66	0.2106
Alternativa de softw	472.95	1	472.95	1.66	0.2106

Error	6559.61	23	285.20
Total	7032.56	24	

Variable	N	R ²	R ² Aj	CV
Cantidad de requeri3	31	0.07	0.04	92.92

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	1478.03	1	1478.03	2.20	0.1485
Alternativa de softw	1478.03	1	1478.03	2.20	0.1485
Error	19449.45	29	670.67		
Total	20927.48	30			

Variable	N	R ²	R ² Aj	CV
Cantidad de requeri4	25	0.54	0.52	57.45

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	642.32	1	642.32	27.06	<0.0001
Alternativa de softw	642.32	1	642.32	27.06	<0.0001
Error	545.92	23	23.74		
Total	1188.24	24			

Variable	N	R ²	R ² Aj	CV
Cantidad de requeri5	28	0.34	0.32	112.73

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	154.64	1	154.64	13.52	0.0011
Alternativa de softw	154.64	1	154.64	13.52	0.0011
Error	297.36	26	11.44		
Total	452.00	27			

Variable	N	R ²	R ² Aj	CV
Cantidad de paginas	30	5.0E-03	0.00	51.14

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	11.25	1	11.25	0.14	0.7106
Alternativa de softw	11.25	1	11.25	0.14	0.7106
Error	2242.25	28	80.08		
Total	2253.50	29			

Variable	N	R ²	R ² Aj	CV
Porcentaje de esfuer	31	0.14	0.11	72.53

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	515.61	1	515.61	4.53	0.0419
Alternativa de softw	515.61	1	515.61	4.53	0.0419
Error	3300.77	29	113.82		
Total	3816.39	30			

Variable	N	R ²	R ² Aj	CV
Porcentaje de esfuel	31	0.03	0.00	50.94

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	106.89	1	106.89	0.79	0.3828
Alternativa de softw	106.89	1	106.89	0.79	0.3828
Error	3947.82	29	136.13		
Total	4054.71	30			

Variable	N	R ²	R ² Aj	CV
Porcentaje de tiempo	31	0.12	0.09	75.80

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	459.82	1	459.82	3.88	0.0584
Alternativa de softw	459.82	1	459.82	3.88	0.0584
Error	3433.27	29	118.39		
Total	3893.10	30			

Variable	N	R ²	R ² Aj	CV
Porcentaje de tiemp1	31	2.7E-03	0.00	50.04

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	11.61	1	11.61	0.08	0.7812
Alternativa de softw	11.61	1	11.61	0.08	0.7812
Error	4284.77	29	147.75		
Total	4296.39	30			

Variable	N	R ²	R ² Aj	CV
porcentaje de requer	31	1.9E-03	0.00	62.16

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	65.28	1	65.28	0.05	0.8183
Alternativa de softw	65.28	1	65.28	0.05	0.8183
Error	35220.59	29	1214.50		
Total	35285.87	30			

Variable	N	R ²	R ² Aj	CV
Porcentaje de requel	31	0.04	0.01	192.57

Cuadro de Análisis de la Varianza (SC tipo III)

F.V.	SC	gl	CM	F	p-valor
Modelo	1371.10	1	1371.10	1.27	0.2691
Alternativa de softw	1371.10	1	1371.10	1.27	0.2691
Error	31314.77	29	1079.82		
Total	32685.87	30			

Referencias

- [Alves] Carina Alves, João Bosco Pinto Filho, Jaelson Castro. Analysing the Tradeoffs Among Requirements, Architectures and COTS Components. Centro de Informática, Universidade Federal de Pernambuco Recife, Pernambuco
- [ARM:1998] Software Assurance Technology Center. Automated Requirement Measurement (ARM) tool. 1998 <http://satc.gsfc.nasa.gov/tools/arm/>
- [Avionics:1998] Avionics Software Engineering. Requirements Management Guidebook. Software Engineering Management Working Group. 1998
- [Boehm] Barry Boehm. The Win Win Spiral Model. University of Southern California and Groupware System Center for Software Engineering
- [Bohem:1981] Bohem, Barry R. 1981. Software engineering economics. Englewood Cliffs, N.J.: Prentice - Hall
- [Borland:2004] 1158 Project Retrospectives Miroslav Novak Systems Engineer Borland. Borland Conference 2004
- [Borland:2005] Borland. MITIGATING RISK WITH EFFECTIVE REQUIREMENTS ENGINEERING How to improve decision-making and opportunity through effective requirements engineering. Part two in a series about understanding and managing risk. April 2005
- [Brackett:1990] John W. Brackett. Boston University. Software Requirements. SEI Curriculum Module SEI-CM-19-1.2. Enero 1990.
- [Brain Desarrollos] Brain Desarrollos. Software a la medida de sus problemas. <http://www.braindesarrollos.com.ar/>
- [Brown:1996] Brown, A. W. and Wallnau, K. C. Engineering of Component-Based Systems,

- Component-Based Software Engineering. Software Engineering Institute, IEEE Computer Society Press, 1996.
- [Collofello:1988] Collofello, J.S. The Software Technical Review Process. Curriculum Module SEI-CM-3-1.5, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., Junio 1988.
- [COMP] Requirements Engineering. COMP 354 | Software Engineering I. Course Notes. Fall 2003
- [CreativeData: 1995_1] Creative Data. Development Methodology - RAD and JAD
<http://www.credata.com/research/methodology.html>. 1995
- [Davis:2003] Alan M. Davis. Elicitation Technique Selection: How Do Experts Do It? Ann M. Hickey. Department of Information Systems, The University of Colorado at Colorado Springs. 2003
- [Definicion] Definicion.org
www.definicion.org/medicion
- [DOE] Software Engineering methodology. Chapter 4.0 Requirements Definition Stage. DOE
- [Easterbrook:2000-2004] Steve Easterbrook 2000-2004. Department of Computer Science. University of Toronto. Lecture 4: Requirements Elicitation. 04-Elicitation-techniques.pdf
- [Fenton:1998] E Norman Fenton, and Pfleeger L Shari, "*Software Metrics- A Rigorous & Practical approach*", 2nd Edition, International Thomson Publishing, Boston, MA, 1998
- [Fred:1979] Roberts Fred S, "*Measurement Theory with Applications to Decision making, Utility, and the Social Sciences*". Addison Wesley Publishing Company, 1979
- [Goguen:1993] Joseph A. Goguen Centre for Requirements Oxford University Computing Lab and Foundations and Charlotte Linde Institute for Research on Learning Palo Alto, California. Techniques for Requirements

- Elicitation. In Proceedings, Requirements Engineering 1993, edited by Stephen Fickas and Anthony Finkelstein, IEEE Computer Society, 1993, pages 152{164.
- [Green:2003] Stewart Green. Eliciting *Stakeholders s'* Knowledge of Goals and Processes to derive IT Support. The University of the West of England. 2003
- [Hans van Vliet] Software Engineering: Principles and Practice. Second Edition. Hans van Vliet. John Wiley & Sons.
- [Horst:1995] Zuse Horst, "History of Software Measurement", 1995
<http://www.literateprogramming.com/m3hist.pdf>
- [IEEE610:1990] IEEE Std. 610.12, 1990 IEEE Computer Dictionary - Compilation of IEEE Standard Computer Glossaries, 610-1990
- [Informit:2002] Informit.com. Software: Packaged or Build. 2002
www.informit.com/articles/article.asp?p=25771&seqnum=4&rl=1
- [ISO/IEC 25030:2003] ISO/IEC 25030. Software Engineering - Software quality requirements and evaluation (SQuARE) - Quality Requirements. 2003
- [ISO9126-1:2001] ISO/IEC 9126-1:2001 Software engineering -- Product quality -- Part 1: Quality model
- [Jarke:1993] Matthias Jarke. Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis. 1993
<http://citeseer.ist.psu.edu/jarke93theorie.html>
- [Jennerich:1990] Bill Jennerich. Joint Application Design Business Requirements Analysis for Successful Re-engineering. 1990
<http://www.bee.net/bluebird/jaddoc.htm>
- [Kennisacquisitie] Kennisacquisitie en Validatie van Kennissystemen. Blok 4.1 AI, najaar 2004 NB ten opzichte van vorig jaar iets meer validatie!. Institute for knowledge and agent technology.

- [Kitchenham:1995] Barbra Kitchenham, Pfleeger L Shari and Fenton E Norman,
“Towards a Framework for Software Measurement Validation”, In IEEE Transactions on software Engineering, 21(12), December, 1995
- [Klier] Wolfgang Klier. Early Phase Requirements Engineering *The WinWin Spiral Model*. TUM - Lehrstuhl Brügge - Allen Dutoit
- [Kotonya:2000] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, 2000.
- [Loconsole;2003] Annabella Loconsole and Jürgen Börstler. Theoretical Validation and Case Study of Requirements Management Measures UMEÅ UNIVERSITY Department of Computing Science. 2003
- [Madigan] Michael Madigan StorageTek Manager, PAL Engineering. Requirements Elicitation. <http://ece-www.colorado.edu/~swengctf/standalone/presentations/293,20,RequirementsElicitationGuidelines1>
- [Maner:1997] Walter Maner
<http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>.
1997
- [Nuseibeh] Bashar Nuseibeh & Steve Easterbrook. “Requirements Engineering: A Roadmap”. Department of Computing, Imperial College of Science, Technology & Medicine. London SW7 2BZ, UK
- [Oberndorf:1997_1] Oberndorf, P. and Brownsword, L. Are You Ready for COTS? Software Institute Engineering. August 1997.
- [Oberndorf:1997_2] Oberndorf, P. Facilitating Component-Based Software Engineering: COTS and Open Systems. Proceedings of the Fifth International Symposium on Assessment of Software Tools - SAST'97 . June 1997.

- [Oudshoorn:2004] IJsbrand Oudshoorn. Vrije Universiteit Amsterdam student number: 1275208 Development of Packaged Software. April 2004.
- [Rosenberg:1996] Wilson, W., Rosenberg, L., Hyatt, L., Automated Quality Analysis of Natural Language Requirement Specifications in Proc. Fourteenth Annual Pacific Northwest Software Quality Conference, Portland OR, 1996
- [Rosenberg:1998] Dr. Linda Rosenberg. Software Metrics and Reliability. Unisys/NASA GSFC. 1998
http://64.233.167.104/search?q=cache:0rQNGE-qpGQJ:satc.gsfc.nasa.gov/support/ISSRE_NOV98/software_metrics_and_reliability.html+software+requirements+metrics&hl=en
- [Runeson:2000] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B. and Wesslen, A., Experimentation in Software Engineering An Introduction, Kluwer Academic Publishers, Boston/Dordrecht/London, 2000
- [Sakol:2001] Teeravarunyou, Sakol. Object-Mediated User Knowledge Elicitation Method. A Methodology in Understanding User Knowledge, Institute of Design - Illinois Institute of Technology, Chicago, USA Sato, Keiichi, Institute of Design - Illinois Institute of Technology, Chicago, USA. The proceeding of the 5th Asian International Design Research Conference, Seoul, Korea, October 2001
- [Sindre:2005] Guttorm Sindre, IDI Requirements elicitation techniques Information Systems, Spring 2005
- [Software improvement group:2005] Software improvement group. Packaged or Custom Software? 2005. <http://www.software-improvers.com/HOME/Topics/packsoft.html>
- [Sommerville:1997] Ian Sommerville and Pete Sawyer. Requirements Engineering: A good practice guide. Lancaster University. 1997

- [Standish:1994] Standish94, <http://www.standishgroup.com/chaos/toc.php>
- [SWEBOK:2004] Guide to the Software Engineering Body of Knowledge 2004 Version SWEBOK IEEE – 2004 Version
- [SystemsAlliance] SystemsAlliance. PACKAGED SOFTWARE DEPLOYMENT
<http://www.systemsalliance.com/Capabilities/packagedsoftwaredeployment.html>
- [TechRepublic] TechRepublic. Evaluate packaged software carefully before you buy. <http://techrepublic.com.com/5100-10878-1058616.html#>
- [TickIT:2002] Getting The Measure Of TickIT. Guidance and information about the emerging ISO measurement standards for improving software processes and how they relate to ISO 9001:2000. 2002
- [Tuffley:2005] Angela Tuffley CIT3190 IT Project Course. Requirements Elicitation & Management. 2005
- [Wiegers:1999] Wiegers K. E. 1999. Software Requirements. Microsoft Press.
- [Yatco:1999] Mei C. Yatco. Joint Application Design/Development System Analysis, Fall 1999
<http://www.umsl.edu/~sauter/analysis/JAD.html>

Otros Materiales Consultados

- [Antonelli] Leandro Antonelli, Alejandro Oliveros. LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires. Fuentes utilizadas por desarrolladores de software en Argentina para elicitar requerimientos
- [Bartholomew:1995] Doug Bartholomew. A Packaged Solution Off-the-shelf software will fuel manufacturers' reengineering efforts. 1995
<http://www.informationweek.com/545/500manuf.htm>
- [Bertolami:2005] Mabel Bertolami Facultad de Ingeniería, Universidad Nacional de la Patagonia S. J. B. Comodoro Rivadavia, Argentina and Alejandro Oliveros Facultad de Ingeniería, Universidad Nacional de Buenos Aires Buenos Aires, Argentina Facultad de Informática, Universidad Nacional de La Plata La Plata, Argentina Estimate of the Functional Size in the Requirements Elicitation. August 2005
- [Boehm:1995] B. Boehm, P. Bose, E. Horowitz, and M. J. Lee, "Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach," Proceedings of the 17th International Conference on Software Engineering (ICSE-17), Seattle, April 1995,
<http://sunset.usc.edu/TechRpts/Papers/icse.conf.95.ps>
- [Boehm:1998] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy, "Using the WinWin Spiral Model: A Case Study", IEEE Computer, July 1998, pp. 33-44,
http://sunset.usc.edu/aegyed/publications/-Using_the_WinWin_Spiral_Model-A_Case_Study.pdf
- [Burge] Janet E. Burge. Knowledge Elicitation Methods. Artificial Intelligence Research Group. Worcester Polytechnic Institute

- [Business Solutions] Select Business Solutions. <http://www.selectbs.com>
- [Carmel] Erran Carmel. American Hegemony in Packaged Software Trade and the "Culture of Software" <http://www.indiana.edu/~tisj/readers/abstracts/13/13-1%20Carmel.html>
- [Chung_1] Lawrence Chung. Requirements Engineering: Introduction. Computer Science Program, The University of Texas, Dallas
- [Chung_10] Lawrence Chung. From SADT to a Formal OO-RML. Computer Science Program, The University of Texas, Dallas
- [Chung_11] Lawrence Chung. Agents Structural Functional Requirements. Computer Science Program, The University of Texas, Dallas.
- [Chung_2] Lawrence Chung. Scenario Analysis. Why, What & How. Computer Science Program, The University of Texas, Dallas
- [Chung_3] Lawrence Chung. Non-Functional Requirements. Computer Science Program, The University of Texas, Dallas
- [Chung_4] Lawrence Chung. What is RE?. Types of Errors. Computer Science Program, The University of Texas, Dallas
- [Chung_5] Lawrence Chung. Behavioral Requirements. Computer Science Program, The University of Texas, Dallas
- [Chung_6] Lawrence Chung. Requirements Elicitation. Computer Science Program, The University of Texas, Dallas
- [Chung_7] Lawrence Chung. Requirements Elicitation: Part 2. Computer Science Program, The University of Texas, Dallas
- [Chung_8] Lawrence Chung. Requirements Analysis, Modeling &

Specification. Computer Science Program, The University of Texas, Dallas

[Chung_9] Lawrence Chung. Modeling Enterprise Requirements. Computer Science Program, The University of Texas, Dallas

[Creative Data:1995_2] Creative Data. Development Methodology - Joint Application Development (JAD). 1995
<http://www.credata.com/research/jad.html>

[Davis:1993] A.M. Davis, Software Requirements: Objects, Functions and States, Prentice Hall, 1993.

[Developer Connection:2005] Developer Connection. Specifying Version Information for Packaged Software. 2005
http://developer.apple.com/documentation/DeveloperTools/Conceptual/SoftwareDistribution/Concepts/sd_pkg_version_info.html

[Dorfman] Merlin Dorfman and Richard H. Thayer. Software Engineering

[Emam:1995] Khaled El Emam and Nazim H. Madhavji. School of Computer Science, McGill University. Centre de Recherche Informatique de Montreal (CRIM). Measuring the success of Requirements Engineering Processes. 1995 IEEE.

[Faulk:1995] Faulk. SOFTWARE REQUIREMENTS: A TUTORIAL. 1995

[Felici:1999] Massimo Felici, Stuart Anderson Requirements Evolution of an Avionics Safety-Critical Industrial Case Study Predictive Changes Risk/Cost Analyses. 1999

[Fowler:1995] Susan Fowler. JAD: joint application design. 1995
<http://www.techwhirl.com/techwhirl/archives/9509/techwhirl-9509-00620.html>

[Goguen:1993] J. Goguen and C. Linde, "Techniques for Requirements Elicitation," presented at International Symposium on Requirements Engineering, 1993.

- [Gonzalez:1993] A.J. Gonzalez & D.D. Dankel (1993), *The Engineering of Knowledge-based Systems: Theory and Practice*, Prentice-Hall Inc., Englewood Cliffs, New Jersey: hfdst 10, 13 t/m 14,
- [Gregoriades:2004] Andreas Gregoriades, Jae-Eun Shin and Alistair Sutcliffe
Centre for HCI Design, School of Informatics
University of Manchester. *Human-Centred Requirements Engineering. Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04)*
- [Hall:1997] Anthony Hall. *What's the Use of Requirements Engineering?* 1997 IEEE
- [Hickey:2003] Ann M. Hickey and Alan M. Davis. Department of Information Systems, The University of Colorado at Colorado Springs. *Elicitation Technique Selection: How Do Experts Do It?. Proceedings of the 11th IEEE International Requirements Engineering Conference. 2003 IEEE.*
- [IEEE:1988_1] 982.1-1988 IEEE Standard Dictionary of Measures to Produce Reliable Software
- [IEEE:1988_2] 982.2-1988 IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software
- [IEEE:1997] Richard Thayer, *Software Requirements Engineering*, IEEE, 1997
- [IEEE:2000] MBASE, "Avoiding the Software Model-Clash Spiderweb," *IEEE Computer*, November, 2000, pp. 120-122.
- [Jiang:2004] Li Jiang. Dept. of Electrical & Computer Engineering, University of Calgary, Canada Computer Engineering Department, American University of Sharjah, UAE. *A Methodology for Requirements Engineering Process Development. Proceedings of the 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'04).*

- [Jiang:2005] Li Jiang, Armin Eberlein and Behrouz H. Far. Dept. of Electrical & Computer Engineering, University of Calgary, Canada. Computer Engineering Department, American University of Sharjah, UAE. Combining Requirements Engineering Techniques -Theory and Case Study. Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)
- [Kelvin:1889] Lord Kelvin, "Popular Lectures and Addresses", 1889
<http://www-history.mcs.st-andrews.ac.uk/history/Mathematicians/Thomson.html>
- [Kern:2002] Harris Kern. Software: Packaged or Build. 2002
<http://www.phptr.com/articles/article.asp?p=25771&rl=1>
- [Knight:2002] John Knight and Marie Jefsoutine. Understanding the user: Research methods to support the digital media designer. Research Issues in Art, Design and Media is edited by Darren Newbury and Nick Stanley, and published by The Research Training Initiative, Birmingham Institute of Art and Design, University of Central England. Issue No.3 Summer 2002
- [Ko:1999] Dong-Gil Ko. Information Requirements Analysis and Multiple Knowledge Elicitation Techniques: Experience with the Pricing Scenario System. Management Information Systems, Katz Graduate School of Business. University of Pittsburgh, Pittsburgh, Pennsylvania 15260. Proceedings of the 32nd Hawaii International Conference on System Sciences - 1999
- [Kotonya:1995] G. Kotonya and I. Sommerville, "Requirements Engineering with Viewpoints", Software Eng. Journal 1996; 11(1): 5-11,
<ftp://ftp.comp.lancs.ac.uk/pub/reports/1995/CSEG.10.95.ps.Z>
- [Lawrence:2001] Brian Lawrence, Karl Wieggers, and Christof Ebert
The Top Risks of Requirements Engineering IEEE

SOFTWARE November/December 2001.

- [Leffingwell:2000] Dean Leffingwell and Don Widrig. *Managing Software Requirements; A Unified Approach,* Addison-Wesley, 2000
- [Loucopulos:1995] P. Loucopulos and V. Karakostas, *Systems Requirements Engineering*, McGraw-Hill, 1995.
- [Macaulay:1996] Linda Macaulay Department of Computation, University of Manchester Institute of Science and Technology, *Requirements for Requirements Engineering Techniques*. 1996 IEEE.
- [Mead:2003] Nancy R. Mead. *Requirements Engineering for Survivable Systems*. *Networked Systems Survivability*. September 2003.
- [Mead:2003] Nancy E. Mead. Software Engineering Institute. *Requirements Engineering in Practice: Making the business case for Requirements Engineering*. Proceedings of the 11th IEEE International Requirements Engineering Conference. 2003 IEEE.
- [Niessink] Frank Niessink. *"Software Requirements"*
- [O'Connor:2004] Debra L. O'Connor. *MEASURING TEAM COGNITION: CONCEPT MAPPING ELICITATION AS A MEANS OF CONSTRUCTING TEAM SHARED MENTAL MODELS IN AN APPLIED SETTING.*, Purdue University Tristan E. Johnson, Learning Systems Institute, Florida State University Mohammed K. Khalil, Learning Systems Institute, Florida State University *Concept Maps: Theory, Methodology, Technology*. Proc. of the First Int. Conference on Concept Mapping A. J. Cañas, J. D. Novak, F. M. González, Eds. Pamplona, Spain 2004
- [O'Connor:2004] Debra L. O'Connor, Purdue University Tristan E. Johnson, Learning Systems Institute, Florida State University Mohammed K. Khalil, Learning Systems Institute, Florida State University *MEASURING TEAM COGNITION: CONCEPT*

MAPPING ELICITATION AS A MEANS OF
CONSTRUCTING TEAM SHARED MENTAL
MODELS IN AN APPLIED SETTING. Pamplona,
Spain 2004

- [Pfleeger:2001] S.L. Pfleeger, "Software Engineering: Theory and Practice," second ed., Prentice Hall, 2001, Chap. 4.
- [RADC-TR-83-175] Software Quality Measurement for Distributed Systems, RADC-TR-83-175
- [Reeker] Larry H. Reeker, U.S. National Institute of Standards and Technology. SWEBOK Knowledge Area Jump-Start Document for Requirements Analysis.
- [Schreiber:1999] A.Th. Schreiber et al. (1999), Knowledge Engineering and Management: The CommonKADS Methodology, The MIT Press, Cambridge Massachussets, London, England: hfst 8
- [SEI:1992_1] Software Measurement for DoD Systems: Recommendations for Initial Core Measures (CMU/SEI-92-TR-19). 1992
- [SEI:1992_2] Software Size Measurement: A Framework for Counting Source Statements (CMU/SEI-92-TR-20). 1992
- [SEI:1992_3] Software Effort and Schedule Measurement: A Framework for Counting Staff-hours and Reporting Schedule Information (CMU/SEI-92-TR-21). 1992
- [SEI:1992_4] Software Quality Measurement: A Framework for Counting Problems and Defects (CMU/SEI-92-TR-22)
Goal-Driven Software Measurement - A Guidebook (CMU/SEI-96-HB-002)
- [SEI:1997] Practical Software Measurement: Measuring for Process Management and Improvement (CMU/SEI-97-HB-003). 1997
- [Seth] Rishi Seth. The Bible of Indian IT industry.
PACKAGED SOFTWARE: Multi-factor Revival

<http://www.dqindia.com/dqtop20/2004/ArtIndseg.asp?artid=59645>

- [Shehata:2004] Mohamed Shehata. Dept. of Electrical & Computer Engineering, University of Calgav, Canada Computer Engineering Department, American University of Sharjah, UAE. A REQUIREMENTS INTERACTION DETECTION PROCESS GUIDE. CCECE 2004 - CCGE12004, Niagara Falls, May/may 2004.
- [STEP:1991] STEP, Operational Requirements for Automated Capabilities. 1991
- [Thayer] Requirements Engineering, Thayer, SMC 10/97, version 2
- [Van der Werf:1992] B.M. van der Werf (1992) Methoden van Kennisacquisitie. In: Simons et al. (eds.) Kennis in Organisaties: Toepassing en theorie van Kennissystemen, Coutinho.
- [Woody:2005] Carol Woody, PhD. Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements. March 2005.