

Ingeniería de Requerimientos

Guillermo Choque Aspiazu

gchoque@umsanet.edu.bo

Resumen

En este trabajo se presenta una visión de lo que constituye la ingeniería de requerimientos, se ofrece las definiciones iniciales de requerimiento y se introduce el proceso asociado con los pasos de: identificación, análisis y negociación, especificación, modelado, validación y gestión del requerimiento. Luego se presentan las técnicas principales de elicitación de requerimientos, desde las entrevistas, pasando por el desarrollo conjunto de aplicaciones, la tormenta de ideas y los casos de uso.

Palabras clave: *requerimiento, funcional, especificación, aplicación, tormenta de ideas, casos de uso.*

1. INTRODUCCION

A medida que pasan los años, la ingeniería del software ha introducido y popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso de desarrollo en sí. Se han publicado muchos libros y artículos relacionados con este tema, con el modelado de procesos del negocio y la reingeniería. Al mismo tiempo, un número creciente de herramientas automatizadas surgieron para ayudar a definir y aplicar un proceso de desarrollo de software efectivo. Hoy en día la economía globalizada depende más de sistemas automatizados que en épocas pasadas; esto ha llevado a los equipos de desarrollo a enfrentarse con una nueva década de procesos y estándares de calidad.

Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado y por ende no descrito totalmente. Uno de los resultados más importantes de la aplicación del proceso de ingeniería de sistemas es la especificación de un sistema basado en computadora que se describe de manera genérica en los siguientes niveles: vista global de todo el sistema, vista del dominio, vista del elemento y vista detallada. Como se observa en la Fig. 1, esta jerarquía está organizada de manera deductiva, de lo general a lo particular.

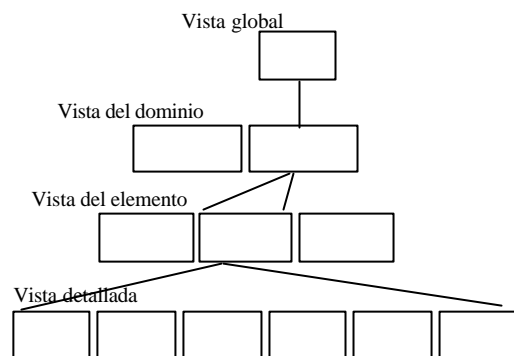


Fig. 1. Jerarquía de sistemas computacionales

Fuente: [Pressman, 2002]

En este contexto se presenta un desafío a los ingenieros del software ¿Cómo se puede asegurar que se ha especificado un sistema que recoge las necesidades del cliente y satisface sus expectativas?. Según Pressman (2002) no hay una respuesta segura a esta difícil pregunta, pero un sólido proceso de ingeniería de requerimientos¹ es la mejor solución de la que se dispone en este momento.

La ingeniería de requerimientos facilita el mecanismo apropiado para comprender lo que requiere el cliente, analizando necesidades, confirmando su factibilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requerimientos para que se transformen en un sistema operacional [Thayer y Dorfman, 1997].

¹ También denominada ingeniería de requisitos en la terminología utilizada por Pressman (2002).

La ingeniería de requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo del producto software.

2. PROCESO DE LA INGENIERIA DE REQUERIMIENTOS

Antes de presentar el proceso de la ingeniería de requerimientos, es necesario comentar la definición abstracta del termino requerimiento, en este entendido, de las muchas definiciones que existen, se presenta la definición que aparece en el glosario de la IEEE²: *“(1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. (3) Una representación documentada de una condición o capacidad como en (1) o (2).*

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

Ahora, ingresando en la temática relacionada con la ingeniería del software, es posible mencionar que de acuerdo con las investigaciones realizadas por Somerville y Sawyer (1997), el proceso de la ingeniería de requerimientos puede ser descrito en seis pasos que se describen en los siguientes párrafos.

² IEEE Standar 610.12-1990.

2.1. Identificación

Este primer paso del proceso aparenta ser bastante simple, por el hecho de preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos del producto, cómo los sistemas o producto se ajustan a las necesidades del negocio, y finalmente como el sistema o producto será utilizado en el trabajo diario. Esto que aparenta ser bastante simple, suele tornarse en una tarea bastante complicada.

Se sugiere un conjunto de acciones para la obtención de requerimientos, que están descritos en las siguientes tareas:

- a) Calorar el impacto en el negocio y la viabilidad técnica del sistema propuesto;
- b) Identificar las personas que ayudaran a especificar los requerimientos y contrastar su papel en la organización;
- c) Definir el entorno técnico³ en el sistema o producto a desarrollar e integrar;
- d) Definir una o más técnicas⁴ de obtención de requerimientos.
- e) Solicitar la participación de muchas personas para que los requerimientos se definan desde diferentes puntos de vista, asegurarse de identificar lo fundamental de cada requerimiento registrado.
- f) Identificar requerimientos ambiguos como candidatos para el prototipado, y
- g) Crear escenarios de uso para ayudar a los clientes o usuarios a identificar mejor los requerimientos fundamentales [Somerville y Sawyer, 1997].

Es menester indicar que cada uno de los productos resultantes de este primer paso, deben ser revisados por las personas que hayan participado en la obtención de sus requerimientos

2.2. Análisis y negociación

Una vez que se identifican los requerimientos, estos configuran la base del análisis de requerimientos. Los requerimientos se agrupan por categorías y se organizan en subconjuntos, se estudia cada requerimiento en relación con el

³ Hace referencia a: arquitectura de computadora, sistema operativo, necesidad de telecomunicación.

⁴ Algunas técnicas son: entrevistas, grupos de trabajo, equipos de discusión. Son tratadas mas adelante.

resto, se examinan los requisitos en su consistencia, completitud y ambigüedad, clasificando posteriormente sobre la base de las necesidades de los clientes o usuarios.

Al inicio de la actividad de análisis de requerimientos es menester resolver las siguientes cuestiones:

- a) ¿Cada requerimiento es consistente con los objetivos generales del sistema?
- b) ¿Tienen todos los requerimientos especificados el nivel adecuado de abstracción?
- c) ¿El requerimiento es necesario o representa una característica añadida que puede no ser esencial a la finalidad del sistema?
- d) ¿Cada requerimiento está delimitado y sin ambigüedad?
- e) ¿Existe un origen conocido para cada requerimiento?
- f) ¿Existen requerimientos incompatibles con otros requerimientos?
- g) ¿Es posible lograr cada requerimiento en el entorno técnico donde se integrará el sistema?
- h) ¿Se puede probar el requerimiento una vez implementado?

Es común en clientes y usuarios solicitar más de lo que puede realizarse, consumiendo recursos de negociación limitados. También es común proponer requerimientos contradictorios argumentando que su versión es “esencial por necesidades especiales”, obligando en algunos casos al ingeniero del sistema a la inclusión forzosa del requerimiento. El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Los clientes, usuarios y el resto de intervinientes deberán clasificar sus requerimientos y discutir los posibles conflictos según su prioridad [Pressman, 2002].

2.3. Especificación

Según Pressman (2002), en el contexto de un sistema basado en computadoras, el término especificación significa distintas cosas para diferentes personas. Una especificación puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de escenarios de uso, un prototipo o una combinación de lo anteriormente citado.

Somerville y Sawyer (1997) sugieren que debe desarrollarse una “plantilla estándar” y utilizar la misma en la especificación del sistema, argumentando que así se conseguirían requerimientos presentados de manera consistente y comprensible. No obstante, es necesario en muchas ocasiones buscar la flexibilidad cuando una especificación va a ser desarrollada. Para grandes sistemas, un documento escrito, combinado con descripciones en lenguaje natural y modelos gráficos puede ser la mejor alternativa.

La especificación del sistema es el producto final que se obtiene del análisis de los requerimientos del sistema obtenidos por el ingeniero. Sirve como fundamento para la ingeniería del hardware, la ingeniería del software, la ingeniería de las bases de datos y la ingeniería humana. Describe la función y características de un sistema de computación y las restricciones que gobiernan su desarrollo. La especificación delimita cada elemento del sistema. La especificación del sistema describe la información⁵ que entra y sale del sistema [Pressman, 2002].

2.4. Modelado del sistema

Para especificar completamente el producto a desarrollar es necesario contar con un modelo del detallado del producto con toda la información circundante al mismo, es decir, un anteproyecto o representación del producto en tres dimensiones que muestre la posición de los elementos constituyentes del mismo.

Con el modelo es relativamente sencillo asegurar la eficiencia del trabajo y la estética visual de los componentes del producto. Es importante evaluar los componentes del sistema y sus interrelaciones, determinar como están reflejados los requerimientos y valorar como se ha concebido la estética en el sistema [Pressman, 2002].

2.5. Validación

La validación de requerimientos examina las especificaciones para asegurar que todos los requerimientos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido

⁵ El autor se refiere tanto a datos como a control.

corregidos y que el resultado del trabajo se ajuste a los estándares establecidos para el proceso, el proyecto y el producto.

El primer mecanismo, y por supuesto uno de los mecanismos más importantes, es la revisión técnica formal. El equipo de revisión incluye ingenieros del sistema, clientes, usuarios y otros intervinientes que examinan la especificación del sistema⁶ buscando errores en el contenido o en la interpretación, áreas donde se necesitan aclaraciones, información incompleta, inconsistencias, requerimientos contradictorios, requerimientos imposibles o inalcanzables [Pressman, 2002].

2.6. Gestión

La gestión de requerimientos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requerimientos y sus cambios asociados en cualquier momento. De manera similar a la Gestión de Configuración del Software (GCS), la gestión de requerimientos comienza con la actividad de identificación. A cada requerimiento se le asigna un identificador único que puede tomar la siguiente forma:

<tipo de requerimiento><numero de requerimiento>

El tipo de requerimiento puede tomar alguno de los siguientes valores: F=funcional, D=datos, C=comportamiento, I=interfaz, S=salida

Según Pressman (2002), una vez que los requerimientos han sido identificados se desarrollan un conjunto de matrices para su seguimiento. Cada matriz de seguimiento identifica los requerimientos relacionados con uno o más aspectos del sistema o su entorno.

Entre las posibles matrices de seguimiento están:

- a) De características. Muestra los requerimientos identificados con relación a las características definidas por el cliente del sistema.
- b) De orígenes. Identifica el origen de cada requerimiento.
- c) De dependencias. Indica como se relacionan los requerimientos entre sí.

⁶ Se recomienda examinar pequeñas partes de la especificación, de forma que se pueda centrar la atención en un aspecto específico de los requerimientos.

d) De subsistemas. Vincula los requerimientos a los subsistemas que los manejan.

e) De interfaces. Muestra como los requerimientos están vinculados a las interfaces externas o internas del sistema.

3. TÉCNICAS DE ELICITACION

A continuación, se describen algunas de las técnicas para obtener los productos de las tareas que se han descrito. Las técnicas más habituales en la elicitación de requerimientos son las entrevistas, el desarrollo conjunto de aplicaciones⁷, el brainstorming o tormenta de ideas y la utilización de escenarios [Weidenhaput et al., 1998] más conocidos como casos de uso [Booch et al., 1999].

A estas técnicas se las suele apoyar con otras técnicas complementarias como la observación in situ, el estudio de documentación, los cuestionarios, la inmersión en el negocio del cliente [Goguen y Linde, 1993] o haciendo que los ingenieros de requerimientos sean aprendices del cliente [Beyer y Holtzblatt, 1995].

3.1. Entrevistas

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más naturales entre personas. En las entrevistas se pueden identificar tres fases: preparación, realización y análisis [Piattini et al., 1996].

3.1.1. Preparación de entrevistas

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

- a) Estudiar el dominio del problema. Conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas [Goguen y Linde, 1993], y para generar en los clientes y usuarios la confianza de que el ingeniero de requerimientos entiende sus problemas. Para conocer el dominio del

⁷ También conocida como Joint Application Development (JAD)

problema se puede recurrir a técnicas de estudio de documentación, a bibliografía sobre el tema, documentación de proyectos similares realizados anteriormente, la inmersión dentro de la organización para la que se va a desarrollar o a periodos de aprendizaje por partes de los ingenieros de requerimientos [Beyer y Holtzblatt, 1995].

- b) Seleccionar a las personas a las que se va a entrevistar. Se debe minimizar el número de entrevistas a realizar, por lo que es fundamental seleccionar a las personas a entrevistar. Normalmente se comienza por los directivos, que pueden ofrecer una visión global, y se continúa con los futuros usuarios, que pueden aportar información más detallada, y con el personal técnico, que aporta detalles sobre el entorno operacional de la organización. Tal como se recomienda en [Piattini et al., 1996], conviene también estudiar el perfil de los entrevistados, buscando puntos en común con el entrevistador que ayuden a romper el hielo.
- c) Determinar el objetivo y contenido de las entrevistas. Para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido. Previamente a su realización, se pueden enviar cuestionarios que los futuros entrevistados deben rellenar y devolver, y un pequeño documento de introducción al proyecto de desarrollo, de forma que el entrevistado conozca los temas que se van a tratar y el entrevistador recoja información para preparar la entrevista. Es importante que los cuestionarios, si se usan, se preparen cuidadosamente teniendo en cuenta quién los va a responder y no incluir conceptos que se asuman conocidos cuando puedan no serlo.
- d) Planificar las entrevistas. La fecha, hora, lugar y duración de la entrevista deben fijarse teniendo en cuenta siempre la agenda del entrevistado. En general, se deben buscar sitios agradables donde no se produzcan interrupciones y que resulten naturales a los entrevistados, tal como se describe en [Goguen y Linde, 1993].

3.1.2. Realización de entrevistas

Según [Piattini et al., 1996], dentro de la realización de las entrevistas se distinguen tres etapas:

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

B. Desarrollo. La entrevista en si no debería durar más de dos horas, distribuyendo el tiempo en un 20% para el entrevistador y un 80% para el entrevistado. Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo [Robertson y Robertson, 1999]. Durante esta fase se pueden emplear distintas técnicas:

- a) Preguntas abiertas. También denominadas de libre contexto [Gause y Weinberg, 1989], estas preguntas no pueden responderse con un "sí" o un "no", permiten una mayor comunicación y evitan la sensación de interrogatorio. Por ejemplo, "¿Qué se hace para registrar un pedido?", "Dígame qué se debe hacer cuando un cliente pide una factura". Estas preguntas se suelen utilizar al comienzo de la entrevista, pasando posteriormente a preguntas más concretas. En general, se debe evitar la tendencia a anticipar una respuesta a las preguntas que se formulan [Raghavan et al., 1994]. En [Gause y Weinberg, 1989] se exponen interesantes ejemplos de este tipo de preguntas y consejos para su utilización. Una posibilidad es utilizar plantillas como mecanismos tanto de obtención de información, ya que su estructura indica la información a buscar, como de registro de las respuestas a este tipo de preguntas.
- b) Utilizar palabras apropiadas. Se deben evitar tecnicismos que no conozca el entrevistado y palabras o frases que puedan perturbar

emocionalmente la comunicación [Goleman, 1996].

- c) Mostrar interés en todo momento. Es fundamental cuidar la comunicación no verbal durante la entrevista: tono de voz, movimiento, expresión facial, etc. Por ejemplo, para animar a alguien a hablar puede asentirse con la cabeza, decir "ya entiendo", "sí", repetir algunas respuestas dadas, hacer pausas, poner una postura de atención, etc. Debe evitarse bostezar, reclinarse en el sillón, mirar hacia otro lado, etc. [Davis, 1985].

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

3.1.3 Análisis de las entrevistas

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc. Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

3.2. Desarrollo conjunto de aplicaciones

La técnica denominada desarrollo conjunto de aplicaciones (JAD) fue desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrollan a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

Esta técnica se base en cuatro principios [Raghavan et al., 1994]: (a) dinámica de grupo, (b) el uso de ayudas visuales para mejorar la comunicación⁸, (c) mantener un proceso

organizado y racional y (d) una filosofía de documentación WYSIWYG⁹, por la que durante las reuniones se trabaja directamente sobre los documentos a generar.

El JAD tiene dos grandes pasos, el JAD/Plan cuyo objetivo es elicitación y especificación de requerimientos, y el JAD/Design, en el que se aborda el diseño del software. Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia, aunque cuando se aplica suele tener buenos resultados, especialmente para elicitación de requerimientos en el campo de los sistemas de información [Raghavan et al., 1994].

En comparación con las entrevistas individuales, presenta las siguientes ventajas:

- a) Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- b) Todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada, no sólo los ingenieros de requerimientos.
- c) Implica más a los clientes y usuarios en el desarrollo.

3.2.1 Participantes del JAD

Tal como se expone en [Raghavan et al., 1994], se pueden distinguir seis clases de participantes o roles en el JAD:

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

⁸ Es decir diagramas, transparencias, multimedia, herramientas CASE, etc.

⁹ What You See Is What You Get, lo que se ve es lo que se obtiene.

de que se utilicen herramientas software durante las sesiones, debe ser capaz de manejarlas eficientemente.

- c) Patrocinador ejecutivo. Es el que tiene la decisión final de que se lleve a cabo el desarrollo. Debe proporcionar a los demás participantes información sobre la necesidad del nuevo sistema y los beneficios que se espera obtener de él.
- d) Representantes de los usuarios. Durante el JAD/Plan, suelen ser directivos con una visión global del sistema. Durante el JAD/Design suelen incorporarse futuros usuarios finales.
- e) Representantes de sistemas de información. Son personas expertas en sistemas de información que deben ayudar a los usuarios a comprender qué es o no factible con la tecnología actual y el esfuerzo que implica.
- f) Especialistas. Son personas que pueden proporcionar información detallada sobre aspectos muy concretos, tanto del punto de vista de los usuarios porque conocen muy bien el funcionamiento de una parte de la organización, como desde el punto de vista de los desarrolladores porque conocen perfectamente ciertos aspectos técnicos de la instalación hardware de la organización.

3.2.2 Fases del JAD

Dentro de la técnica del JAD se distinguen tres fases [Raghavan et al., 1994]:

A. Adaptación. Es responsabilidad del jefe del JAD, ayudado por uno o dos analistas, adaptar la técnica del JAD para cada proyecto. La adaptación debe comenzar por definir el proyecto a alto nivel, para lo cual pueden ser necesarias entrevistas previas con algunos clientes y usuarios. También suele ser necesario recabar información sobre la organización para familiarizarse con el dominio del problema, por ejemplo utilizando técnicas complementarias como el estudio de documentación o la observación in situ. Una vez obtenida una primera idea de los objetivos del proyecto, es necesario seleccionar a los participantes, citarles a las reuniones y proporcionarles una lista con los temas que se van a tratar en las reuniones para que las puedan preparar. El jefe del JAD debe decidir la duración y el número de sesiones a celebrar, definir el formato de la documentación

sobre la que se trabajará y preparar transparencias introductorias y todo el material audiovisual que considere oportuno.

B. Celebración de las sesiones JAD. durante las sesiones, los participantes exponen sus ideas y se discuten, analizan y refinan hasta alcanzar un acuerdo. Los pasos que se recomienda seguir para este proceso son los siguientes:

- a) Presentación. Se presenta y se da la bienvenida a todos los participantes por parte del patrocinador ejecutivo y del jefe del JAD. El patrocinador ejecutivo expone brevemente las necesidades que han llevado al desarrollo y los beneficios que se esperan obtener. El jefe del JAD explica la mecánica de las sesiones y la planificación prevista.
- b) Definir objetivos y requerimientos. El jefe del JAD promueve la discusión para elicitación los objetivos o requerimientos de alto nivel mediante preguntas como: "¿Por qué se construye el sistema?", "¿Qué beneficios se esperan del nuevo sistema?", "¿Cómo puede beneficiar a la organización en el futuro?", "¿Qué restricciones de recursos disponibles, normas o leyes afectan al proyecto?", "¿Es importante la seguridad de los datos?". A medida que se van elicitando requerimientos, el analista los escribe en transparencias o en algún otro medio que permita que permanezcan visibles durante la discusión.
- c) Delimitar el ámbito del sistema. Una vez obtenido un número importante de requerimientos, es necesario organizarlos y llegar a un acuerdo respecto al ámbito del nuevo sistema. En el caso de los sistemas de información, es útil identificar a los usuarios potenciales (actores) y determinar qué tareas les ayudará a realizar (casos de uso).
- d) Documentar temas abiertos. Aquellas cuestiones que hayan surgido durante la sesión que no se han podido resolver, deben documentarse para las siguientes sesiones y ser asignadas a una persona responsable de su solución para una fecha determinada, para lo cual puede utilizarse una plantilla de solución de conflictos.
- e) Concluir la sesión. El jefe del JAD concluye la sesión revisando con los demás participantes la información elicitada y las decisiones tomadas. Se da la oportunidad a todos los participantes de expresar cualquier consideración adicional, fomentando por

parte del jefe del JAD el sentimiento de propiedad y compromiso de todos los participantes sobre los requerimientos elicitados.

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

- a) Completar la documentación. Los analistas recopilan la documentación generada durante las sesiones en documentos conformes a las normas o estándares vigentes en la organización para la que se desarrolla el proyecto.
- b) Revisar la documentación. La documentación generada se envía a todos los participantes para que la comenten. Si los comentarios son lo suficientemente importantes, se convoca otra reunión para discutirlos.
- c) Validar la documentación: una vez revisados todos los comentarios, el jefe del JAD envía el documento al patrocinador ejecutivo para su aprobación. Una vez aprobado el documento se envían copias definitivas a cada uno de los participantes.

3.3. Tormenta de ideas

La tormenta de ideas¹⁰ es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Gause y Weinberg, 1989]. Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla. Como técnica de elicitación de requerimientos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requerimientos son todavía muy difusos. Frente al JAD, el brainstorming tiene la ventaja de que es muy fácil de aprender y requiere poca organización, de hecho, hay propuestas de realización de brainstorming por vídeo-conferencia a través de Internet [Raghavan et al., 1994]. Por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

3.3.1. Fases

En el brainstorming se distinguen las siguientes fases [Raghavan et al., 1994]:

A. Preparación. La preparación para una sesión de brainstorming requiere que se seleccione a los participantes y al jefe de la sesión, citarlos y preparar la sala donde se llevará a cabo la sesión. Los participantes en una sesión de brainstorming para elicitación de requerimientos son normalmente clientes, usuarios, ingenieros de requerimientos, desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.

B. Generación. El jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de semilla para que se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema semilla, bien por un orden establecido por el jefe de la sesión, bien aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea suficiente para pasar a la siguiente fase. Durante esta fase se deben observar las siguientes reglas:

- a) Se prohíbe la crítica de ideas, de forma que los participantes se sientan libres de formular cualquier idea.
- b) Se fomentan las ideas más avanzadas, que aunque no sean factibles, estimulan a los demás participantes a explorar nuevas soluciones más creativas.
- c) Se debe generar un gran número de ideas, ya que cuantas más ideas se presenten más probable será que se generen mejores ideas.
- d) Se debe alentar a los participantes a combinar o completar las ideas de otros participantes. Para ello, es necesario, al igual que en la técnica del JAD, que todas las ideas generadas estén visibles para todos los participantes en todo momento. Una posibilidad es utilizar como semilla objetivos del sistema e ir identificando requerimientos.

C. Consolidación. En esta fase se deben organizar y evaluar las ideas generadas durante la fase anterior. Se suelen seguir tres pasos:

¹⁰ Conocida también como brainstorming.

- a) Revisar ideas. Se revisan las ideas generadas para clarificarlas. Es habitual identificar ideas similares, en cuyo caso se unifican en un solo enunciado.
- b) Descartar ideas. Aquellas ideas que los participantes consideren excesivamente avanzadas se descartan.
- c) Priorizar ideas. Se priorizan las ideas restantes, identificando las absolutamente esenciales, las que estarían bien pero que no son esenciales y las que podrían ser apropiadas para una próxima versión del sistema a desarrollar.

D. Documentación. Después de la sesión, el jefe produce la documentación oportuna conteniendo las ideas priorizadas y comentarios generados durante la consolidación.

3.4. Casos de uso

Los casos de uso son una técnica para la especificación de requerimientos funcionales propuesta inicialmente en [Jacobson et al., 1993] y que actualmente forma parte de la propuesta de UML [Booch et al., 1999]. Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables. Los actores son personas u otros sistemas que interactúan con el sistema cuyos requerimientos se están describiendo [Schneider y Winters, 1998]. Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requerimientos funcionales [Firesmith, 1997], ya que facilitan la elicitación de requerimientos y son fácilmente comprensibles por los clientes y usuarios.

Los casos de uso también pueden servir de base a las pruebas del sistema y a la documentación para los usuarios [Weidenhaput et al., 1998]. A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su utilización concreta [Cockburn, 1997]. En esta metodología se propone la utilización de los casos de uso como técnica tanto de elicitación como de especificación de los requerimientos funcionales del sistema. Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran y se describen usando

lenguaje natural en forma de patrones lingüísticos.

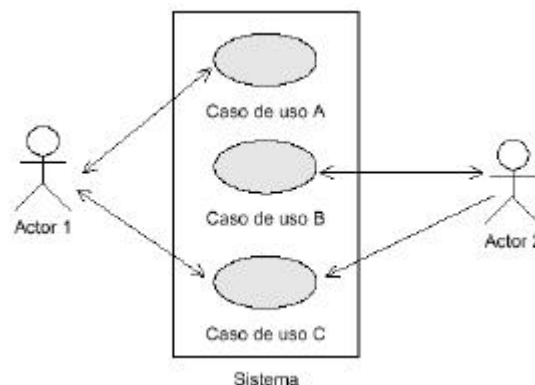


Fig. 2. Diagrama de casos de uso
Fuente: [Booch et al, 1999]

3.4.1 Diagramas de casos de uso

Los casos de uso tienen una representación gráfica en los denominados diagramas de casos de uso [Booch et al., 1999]. En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Un ejemplo de este tipo de diagramas puede verse en la Fig. 2. Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

3.4.2 Relaciones entre casos de uso

A veces conviene establecer relaciones entre distintos casos de uso para simplificar su descripción. Las dos relaciones posibles y sus semánticas según [Booch et al., 1999] son las siguientes, cuya representación gráfica puede verse en el ejemplo de la Fig. 3.

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

uso B se realiza siempre dentro del caso de uso A. Además, siempre que ocurre A ocurre también B, por lo que se dice que B es un caso de uso abstracto [Jacobson et al., 1997]. Un caso de uso es abstracto si no puede ser realizado por sí mismo, por lo que sólo tiene significado cuando se utiliza para describir alguna funcionalidad que es común a otros casos de uso. Por otra parte, un caso de uso será concreto si puede ser iniciado por un actor y realizado por sí mismo. Se suele utilizar esta relación cuando se detectan subsecuencias de interacciones comunes a varios casos de uso. Dichas subsecuencias comunes se sacan "factor común" de los casos de uso que las contienen y se les da forma de casos de uso que son incluidos por los casos de uso de los que se han "extraído". De esta forma se evita repetir las mismas subsecuencias de interacciones una y otra vez en varios casos de uso.

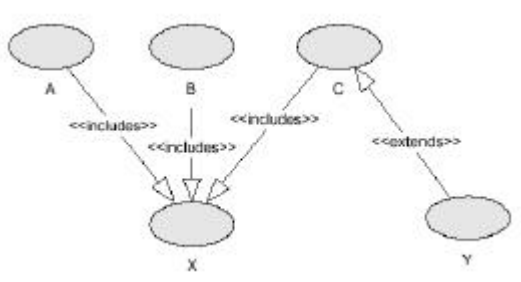


Fig. 3. Representación de includes y extends
Fuente: [Booch et al., 1999]

- b) **extends:** un caso de uso A extiende a otro caso de uso B cuando A es una subsecuencia de interacciones de B que ocurre en una determinada circunstancia. En cierta forma, A completa la funcionalidad de B. El caso de uso A puede realizarse o no cuando se realiza el caso de uso B, según se den las circunstancias. Por otro lado, el caso de uso A puede ser un caso de uso abstracto o concreto, en cuyo caso puede ocurrir sin necesidad de que ocurra el caso de uso B.

3.4.3 Organización de casos de uso

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

casos de uso es recurrir a los paquetes descritos en la propuesta de UML [Booch et al., 1999]. De esta forma, los casos de uso pueden organizarse en niveles, facilitando así su comprensión. Cada paquete contiene a otros paquetes o a varios casos de uso. En el caso de que los casos de uso se agrupen por criterios funcionales, los paquetes que los agrupan pueden estereotiparse como subsistemas [Schneider y Winters, 1998], tal como puede verse en el ejemplo de la Fig. 4.

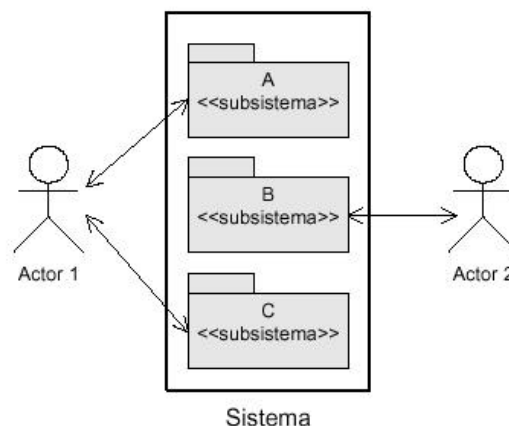


Fig. 4. Representación de los paquetes de casos de uso
Fuente: [Schneider y Winters, 1998]

4. CONCLUSIONES

El resultado más importante de la aplicación del proceso de ingeniería de requerimientos es la especificación adecuada para la construcción de un producto software, esta especificación está asociada de manera lógica al seguimiento y aplicación de un proceso que comienza con la identificación del requerimiento a través de sucesivas conversiones con el cliente, luego viene el análisis del requerimiento y la negociación del mismo, la especificación del requerimiento se encuentra a continuación como primer elemento resultante de la aplicación de los pasos anteriores, posteriormente se ejecuta el modelado del requerimiento, la validación de la correctitud y finalmente la gestión del requerimiento. También se presenta las técnicas principales de elicitación de requerimientos, estas consideran la entrevista como el elemento útil para la adquisición del requerimiento, luego se considera el desarrollo conjunto de aplicaciones, la tormenta de ideas y los casos de uso.

BIBLIOGRAFÍA

- Beyer, H. R. y K. Holtzblatt. Apprenticing with the Customer. *Communications of the ACM*, 38(5), Mayo 1995.
- Booch, G., J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- Cockburn, A. Structuring Use Cases with Goals. *Journal of Object-Oriented Programming*, Sept. y Nov./Dic. 1997.
Disponible en: <http://members.aol.com/acockburn/papers/usecas.es.htm>
- Davis, F. *La comunicación no verbal*, volumen 616 del Libro de Bolsillo. Alianza Editorial, 1985.
- Firesmith, D. G. *Uses Cases: the Pros and Cons*, 1997.
Disponble en <http://www.ksscary.com/usecjrnl.html>
- Gause, D. C. y G. M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.
- Goguen, J. A. y C. Linde. Techniques for Requirements Elicitation. *En Proceedings of the First International Symposium on Requirements Engineering*, 1993.
Disponble en <http://www.cse.ucsd.edu/goguen>
- Goleman, D. *La Inteligencia Emocional*. Kairós, 1996.
- Jacobson, I., M. Christerson, P. Jonsson, y G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 4ta. edición, 1993.
- Piattini, M. G., J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Ra-Ma, 1996.
- Pressman, R.S.; *Ingeniería del Software: un enfoque práctico*. McGraw Hill, 2002.
- Raghavan, S., G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. *Educational Materials CMU/SEI-94-EM-10*, Software Engineering Institute, Carnegie Mellon University, 1994.
Disponble en <http://www.sei.cmu.edu>
- Robertson, S. y J. Robertson. *Mastering the Requirement Process*. Addison-Wesley, 1999.
- Schneider, G. y J. P. Winters. *Applying Use Cases: a Practical Guide*. Addison-Wesley, 1998.
- Somerville, I. y P. Sawyer; *Requeriments Engineering*. Wiley 1997.
- Thayer, R.h. y M. Dorfman; *Software Requirements Engineering*. Wiley. 1997.
- Weidenhaput, K., K. Pohl, M. Jarke, y P. Haumer. Scenarios in System Development: Current Practice. *IEEE Software*, 15(2):34-45, Marzo/Abril 1998.
Este artículo aparece en ICRE'98: <http://sunsite.informatik.rwthachen.de/CREWS/reports97.htm>