

# Definición de una metodología ágil de ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur-occidente colombiano\*

Definition of a responsive methodology of engineering of requirements for emergent software development companies in the southwestern part of Colombia.

**Luis Merchán**  
**Alba Urrea**  
**Rubén Rebollar**

## Resumen

Como industria, el software requiere de productos y servicios de alta calidad, lo cual se logra mediante la aplicación de modelos y metodologías de calidad reconocidos internacionalmente. Las empresas emergentes

no logran aplicar estas metodologías pues su gran obstáculo se observa en los altos costos de implementación, el recurso humano requerido y los estándares exigidos que restringen la creatividad, parte importante de su capital. El Laboratorio de Investigación para el Desarrollo de la Ingeniería de

\* Artículo producto del proyecto de investigación denominado "Implementación de los modelos livianos de gestión de requerimientos y gestión de la configuración y diseño de un modelo liviano de apoyo al área de aseguramiento de calidad del proceso de desarrollo de software para pequeñas empresas del Valle del Cauca". Grupo de investigación Lidis.

• Fecha de recepción del artículo: octubre de 2007 • Fecha de aceptación: marzo de 2008.

**LUIS MERCHÁN.** Ingeniero de Sistemas. Especialista en Finanzas. Magíster en Administración. Doctorado en Dirección de proyectos. email: lmerchan@usbcali.edu.co

**ALBA URREA.** Ingeniera de Sistemas. Magíster en Ingeniería de Sistemas y Computación. e-mail: aurrea@usbcali.edu.co

**RUBÉN REBOLLAR.** Ingeniero Industrial. Doctor en Ingeniería Industrial. e-mail: rrebollar@unizar.es

Software (LIDIS), siendo consecuente con esta situación, adelantó una investigación que propone un modelo liviano de mejoramiento de los procesos de desarrollo de software partiendo de la caracterización de las empresas emergentes.

**Palabras clave:** Ingeniería de requerimientos, desarrollo de software, empresas emergentes.

---

## Summary

As an industry, software requires high quality products and services, which are obtained by means of the application of high quality models and methodologies recognized worldwide. Emergent companies do not manage to apply these methodologies, due to the high costs of implementation, the recruitment of a qualified human resource and the demanded standards that restrict creativity, which are an important part of their capital. The Research Laboratory for the Development of Software Engineering (LIDIS), as a response to this situation, carried out an investigation that proposes a light model of improvement of the software development processes, starting off from the characterization of emergent companies.

**Key words:** Engineering of requirements, software development, emergent companies.

---

## Introducción

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

Si bien la caracterización de las empresas emergentes del sur occidente colombiano

muestra la necesidad de utilizar modelos de mejoramiento de procesos propios, fáciles de entender y aplicar, algunos países como México ya tiene participación en la normalización internacional, mediante el Sistema Normativo Mexicano (NOM-NMX), empleado en MoProSoft (Modelo de Procesos de Software), el cual ha sido fruto de investigaciones similares basadas en modelos como SW-CMM, el estándar ISO 9000 y el reporte técnico ISO/IEC TR 15504; igualmente Brasil cuenta con MARES (Método de Validación de Proceso de Software en Micro y Pequeñas Empresas) (Karim Pickler, et al, 2007). Se debe destacar que con la adopción de un modelo liviano de mejoramiento de procesos se pretende habilitar, incluso en algunos casos, la fase inicial de preparación para la obtención de certificados ISO 9000 y reducir la brecha para la obtención de una evaluación CMM o CMMI nivel 2.

Este artículo inicia con una revisión de literatura pertinente seguida de un diagnóstico para las empresas emergentes, basándose en la muestra de población y el instrumento de recolección de datos, en seguida se presenta un análisis de los datos para determinar los factores problema en el proceso de ingeniería de requerimientos. Luego, se muestra la relación de fases y actividades junto con las descripciones, las entradas y salidas de cada fase y las buenas prácticas que acompañan la metodología ágil. Se concluye con una discusión sobre los hallazgos y cómo estos son suplidos en la metodología.

---

## Revisión de literatura

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. Como disciplina, establece el proceso de definición de requerimientos en una sucesión de actividades mediante las cuales lo que debe hacerse se “elicit”, se modela y analiza (Choque, 2001).

La obtención de requerimientos trata sobre la comunicación entre los desarrolladores, clientes y usuarios para definir el nuevo sistema que brinde una solución al problema; a esta definición del sistema se le denomina especificación de requerimientos, que a su vez sirve como contrato entre el cliente y los desarrolladores. Tanto la obtención de requerimientos y su análisis se enfoca en la visión que tiene el usuario del sistema.

Según las investigaciones realizadas (Somerville, 1997), en el proceso de ingeniería de requerimientos son esenciales diversas actividades. En esta propuesta de metodología ágil dichas actividades serán presentadas secuencialmente, sin embargo, en un proceso efectivo de ingeniería de requerimientos estas son aplicadas de manera continua y en orden variado.

La metodología ágil de ingeniería de requerimientos que se propone en esta

investigación está basada en el modelo de madurez de capacidades integrado (CMMI, 2002), el estándar ISO/IEC 12207 (estándar para los procesos de ciclo de vida del software de la organización ISO) (ISO, 2006) y el referente teórico sobre procesos y procedimientos de ingeniería de requerimientos (Beck, 200), (Bennington, 1956), (Boehm, 1998), (Choque, 2001), (Fowler, 2001), (Jacobson et al, 1992), (Jeffries et al, 2001), (Johnson, 1992), (Moprosoft, 2006), (Newkirk, 2004), (Nielsen, 1994), (Marck, 1994), (Paech, 1998). De estos referentes se seleccionaron las mejores prácticas para los procesos de ingeniería de software con el fin de desarrollar la metodología orientando los esfuerzos a que dichos procesos siempre sean desarrollados de manera óptima y de acuerdo con el contexto de las empresas emergentes (Ver Tabla 1).

Tabla 1  
Análisis comparativo mejores prácticas en ingeniería de requerimientos

INGENIERÍA DE REQUERIMIENTOS	MODELOS ESTÁNDARES		METODOLOGÍA ÁGIL
	Procesos y Procedimientos	CMMI	ISO 12207
Elicitación	Establecer criterios para distinguir proveedores de requerimientos apropiados. Alcanzar un entendimiento de los requerimientos para que los integrantes del proyecto se puedan comprometer con ellos.	Establecer y documentar los requerimientos del software.	Mejores prácticas para implementar Establecer criterios para distinguir proveedores de requerimientos.
Especificación	Establecer criterios objetivos para la aceptación de requerimientos. Analizar los requerimientos para garantizar que los criterios establecidos se cumplan.	Evaluar los requerimientos del software. Llevar a cabo revisiones conjuntas.	Mejores prácticas para implementar Evaluar los requerimientos del software a partir de criterios establecidos.
Gestión	Obtener un compromiso con los requerimientos. Administrar los cambios de los requerimientos. Mantener la trazabilidad de los requerimientos.		Mejores prácticas para implementar Priorizar los requerimientos. Administrar cambios. Realizar la trazabilidad.

## Diagnóstico para las empresas emergentes de software

### Características de la población

Como marco de desarrollo de la caracterización, se tomó al sur occidente colombiano por tratarse del *cluster* de desarrollo de software más grande y maduro del país. Del total de 306 empresas, se trabajó sobre una muestra de 49 empresas (16%) (Ver Tabla 2). En Merchán, et al (2007), se detalla el trabajo de campo y de experimentación seguido para recolectar y obtener los resultados de caracterización de la industria y su diagnóstico.

### Instrumento de recolección de datos

Se realizaron entrevistas a los ingenieros involucrados en las actividades de desarrollo de software de cada una de las empresas de la muestra a través de un cuestionario conformado por 3 secciones:

1. Información general.
2. Información de los modelos de procesos que la empresa utiliza en las iniciativas de mejoramiento.
3. Información de los procesos para el desarrollo de software que realiza la empresa en cuanto a:
  - Ingeniería de requerimientos.
  - Planeación de proyectos.
  - Administración de configuración.
  - Aseguramiento de calidad en el software.

Tabla 2  
Población

REFERENTE	NÚMERO TOTAL DE EMPRESAS
ParqueSoft Cali	183
ParqueSoft Palmira	8
ParqueSoft Tulúa	19
ParqueSoft Popayán	34
ParqueSoft Pereira	21
ParqueSoft Manizales	27
ParqueSoft Pasto	14

### Análisis de datos

El diagnóstico se fundamentó en la aplicación de métodos, técnicas y herramientas en las fases del proceso de ingeniería de requerimientos (de elicitación, de especificación y de gestión de requerimientos) por parte de las empresas emergentes.

En ingeniería de requerimientos se examinaron aspectos como:

- La ejecución de procesos y procedimientos de administración de requerimientos.
- El uso de herramientas que soportan el proceso de administración de requerimientos

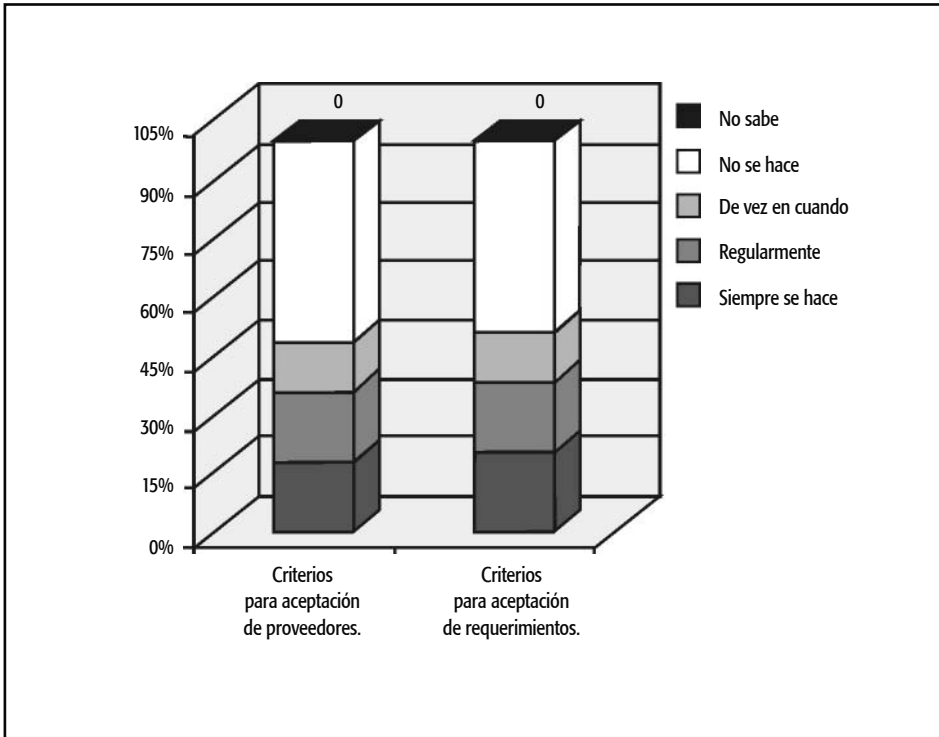
De lo anterior, se logran evidenciar los factores problema presentes en las tres fases de ingeniería de requerimientos (Ver gráficas 1 y 2).

En los resultados obtenidos, se evidencia que las empresas pertenecientes a la industria emergente de software necesitan de metodologías que soporten las actividades de sus procesos de desarrollo de software. A pesar que la mayoría de ellas asegure su utilización, las actividades no se realizan de la forma correcta o en otros casos no se aplican correctamente.

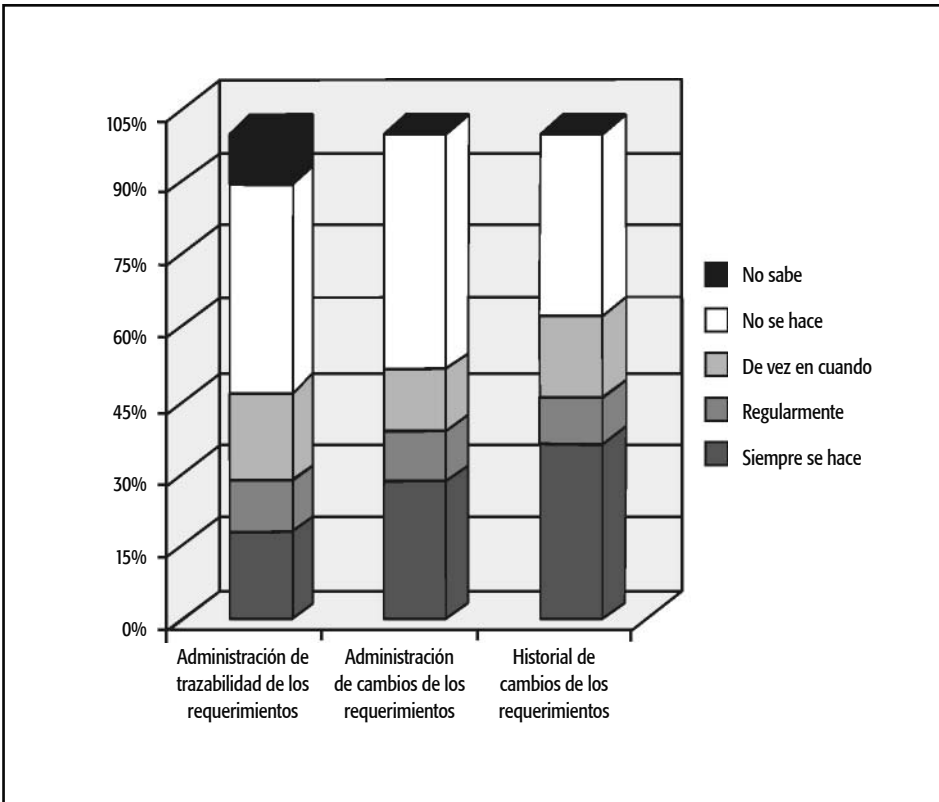
En el proceso de ingeniería de requerimientos que efectúan estas empresas, se encontraron notables problemas en cada una de sus fases:

- Gran parte de las empresas encuestadas (58.3%) no establecen criterios para la aceptación de proveedores de requerimientos, lo que puede traer falencias para el desarrollo del proyecto por cuanto las fuentes de los requerimientos no tienen dominio de la problemática del sistema ni de sus procesos, así como obtener requerimientos mal interpretados e inconsistentes e incluso omitir requerimientos que pueden ser necesarios para la solución.
- Un alto porcentaje de ellas (48.71%) no establece criterios para la aceptación de requerimientos, factor que puede ocasionar la presencia de errores en el desarrollo de los requerimientos que redunden en

**Gráfica 1**  
Fases elicitación y especificación de requerimientos



**Gráfica 2**  
Fase gestión de requerimientos



retrasos en los cronogramas de trabajo, trabajo repetitivo o hasta rechazo del cliente.

- En la fase de gestión de requerimientos, se observa que las empresas (43.6%) no realizan la administración de la trazabilidad de los requerimientos (factor esencial en la fase de gestión de requerimientos), y por ello las dificultades que pueden presentar en la ubicación de fallas en los requerimientos desde la documentación hasta el producto final. Tanto el historial como la administración de cambios son también factores ignorados. Esta situación conduce al desconocimiento del impacto que los cambios puedan generar, las razones de ellos, cómo se hicieron, cuándo fueron hechos y quién los autorizó, entre otros. Esta información histórica de los cambios es útil para la toma futura de decisiones.

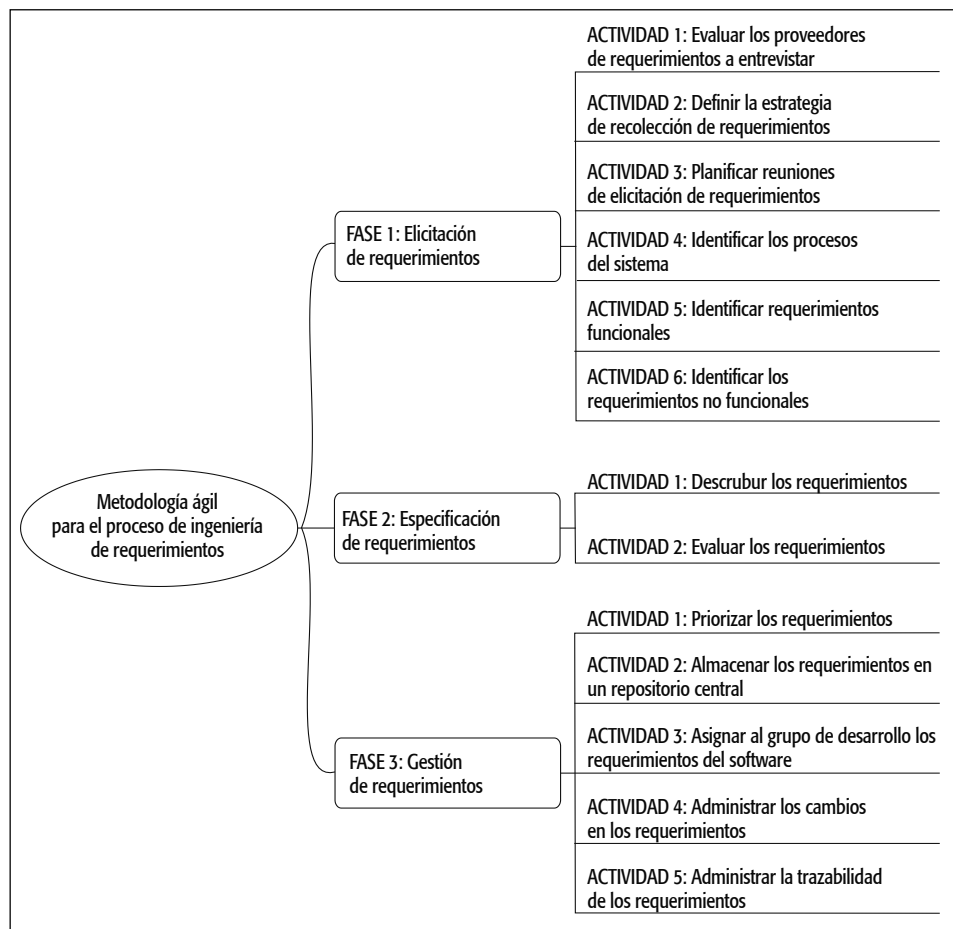
A partir de este diagnóstico se creó la necesidad de disponer de una metodología ágil del proceso de ingeniería de requerimientos, ya que por el concepto mismo de “metodología ágil” les representará, a las empresas emergentes, una facilidad de implementación, uso de pocos recursos y un apoyo en la ejecución de sus procesos de ingeniería de software.

### Metodología ágil para el proceso de ingeniería de requerimientos

Basándose en el diagnóstico de la industria de software del sur occidente colombiano, la metodología ágil va dirigida a dos tipos de proyectos: desarrollo a la medida e ideas de negocio. La metodología se ha estructurado en tres fases (Ver gráfica 3).

**Elicitación de requerimientos:** su propósito es obtener un óptimo entendimiento

Gráfica 3  
Estructura de la metodología ágil para el proceso de ingeniería de requerimientos



de los requerimientos del software en los participantes –los usuarios finales y los desarrolladores–, e identificar los requerimientos funcionales y no funcionales.

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

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

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

- Entradas.
- Actividades.
- Salidas.

## Elicitación

Esta fase tiene tres entradas (Ver Tabla 3) y seis actividades planteadas las cuales se detallan a continuación:

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

El objetivo es verificar que las personas a entrevistar o las fuentes de información

cumplen con los criterios establecidos en la Tabla 3.

En los proyectos de desarrollo a la medida, normalmente es preferible iniciar con los líderes funcionales, ya que ellos tienen una visión global de los procesos y comprenden el dominio del problema. Luego se continúa con los futuros usuarios, es decir, usuarios finales, porque ellos pueden aportar información más detallada sobre el entorno operacional de la organización. En la Tabla 4 se pueden observar los criterios para distinguir los proveedores de requerimientos.

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

Esta actividad tiene como objetivo verificar que la estrategia de recolección de requerimientos sea la recomendada, según el tipo de proyecto a desarrollar. La estrategia ayuda tanto a desarrolladores como a usuarios a conocer la problemática del entorno a mejorar e identificar las necesidades de la organización cliente, mientras que para los emprendedores, les permite conocer el entorno donde aspiran consolidar su negocio.

Para los proyectos de desarrollo a la medida pueden aplicar estrategias tales como las entrevistas y el desarrollo de aplicaciones de agregación (JAD) (Ver Tabla 5).

Tabla 3  
Entradas por tipo de proyecto

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

Tabla 4  
Criterios para la distinción de proveedores de requerimientos

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

Tabla 5  
Estrategias para la recolección de requerimientos

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

### 3. Actividad 3: Planificar reuniones de elicitación de requerimientos

Esta actividad tiene como objetivo:

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

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

En los proyectos de desarrollo de software a la medida, esta actividad es fundamental ya que en estas reuniones los desarrolladores conocen las necesidades de la organización cliente e identifican quiénes serán los usuarios del nuevo sistema.

### 4. Actividad 4: Identificar los procesos del sistema

Esta actividad tiene por objetivos:

- Identificar entradas y salidas de los procesos.



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

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

### 5. Actividad 5: Identificar requerimientos funcionales

Esta actividad tiene por objetivos:

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

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

### 6. Actividad 6: Identificar los requerimientos no funcionales

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

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

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

Ver en la Tabla 6 la salida que tiene la fase de elicitación de requerimientos.

Tabla 6  
Salida de la fase 1 para proyectos de desarrollo de software a la medida e idea de negocio

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

## Especificación

Esta fase, que tiene tres entradas (Ver Tabla 7), presenta las siguientes actividades:

### 1. Actividad 1: Describir los requerimientos

Esta actividad tiene como objetivo:

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

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

### 2. Actividad 2: Evaluar los requerimientos

Esta actividad tiene por objetivo verificar que los casos de uso, historias de usuarios o

prototipos cumplan con los criterios establecidos. La práctica consiste en examinar la especificación para asegurarse que todos los requerimientos de software han sido escritos sin ambigüedades, sin inconsistencias, sin omisiones y errores, han sido detectados y corregidos; que los productos de trabajo sean conformes a los estándares establecidos para el proceso, para el proyecto y para el producto. Esta práctica de evaluación de requerimientos es indispensable en la fase de especificación de los mismos, ya que una vez el requerimiento cumpla o no con los criterios establecidos de evaluación, es aceptado o no; finalmente esto conduce a corregir inconsistencias y a tomar decisiones inmediatas sin tener que esperar a la fase final del proyecto. Generalmente no se realiza para todos los requerimientos sino para una muestra de ellos.

Ver en la Tabla 8 la salida que provee la fase de especificación de requerimientos.

Tabla 7  
Entradas para la fase de especificación de requerimientos

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

Tabla 8  
Salida fase de especificación de requerimientos

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

## Gestión

Esta fase, que tiene tres entradas indicadas en la Tabla 9, plantea las siguientes actividades:

### 1. Actividad 1: Priorizar los requerimientos

Esta actividad tiene como objetivo:

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

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

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

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

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

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

Esta actividad tiene como objetivo:

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

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

### 4. Actividad 4: Administrar los cambios en los requerimientos

Esta actividad tiene como objetivo:

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

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

Tabla 9  
Entradas fase de gestión de requerimientos

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

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

### 5. Actividad 5: Administrar la trazabilidad de los requerimientos

Esta fase tiene por objetivo:

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

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

Esta fase tiene la salida mostrada en la Tabla 10.

### Control en la metodología de ingeniería de requerimientos del software

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

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

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

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

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

### Conclusiones

Se revisó el proceso de ingeniería de requerimientos en las empresas emergentes de desarrollo de software del sur occidente colombiano dándose a conocer la necesidad que ellas tienen de implementar metodolo-

Tabla 10  
Salida fase de gestión de requerimientos

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

gías que soporten estas actividades. La gran mayoría de empresas asegura que emplean metodologías de desarrollo pero realmente no realizan estas actividades de la forma correcta, o en otros casos no las realizan, lo que ocasiona problemáticas en cuanto a criterios para la aceptación de proveedores de requerimientos, criterios para la aceptación de requerimientos y ausencia de administración de la trazabilidad y de los cambios de los requerimientos. Para ofrecer una alternativa viable de mejoramiento de procesos se definió y documentó una metodología ágil de ingeniería de requerimientos para las empresas emergentes la cual comprende tres fases: elicitación, especificación y gestión de requerimientos. Cada fase define las actividades necesarias y los objetivos para realizar estas actividades teniendo en cuenta dos tipos de proyectos: desarrollo a la medida e ideas de negocio.

## Reconocimientos

Se reconoce a las empresas encuestadas la participación en la presente investigación. Igualmente a los estudiantes Karina Hernández y Ángela Lerma y la profesora Patricia Hoyos por el trabajo de consolidación y análisis de los resultados del trabajo de campo.

## Limitaciones y futuras investigaciones

La metodología ágil de ingeniería de requerimientos está en la etapa de aplicación en empresas piloto —empresas pertenecientes al contexto de la industria emergente— buscando que dicha metodología sea retroalimentada en el ambiente real con la adición de más actividades u objetivos en cada fase y evaluando el impacto de su implementación en estas empresas.

Es conveniente el desarrollo de una herramienta informática dirigida a las empresas pertenecientes a la industria emergente que soporte en su totalidad la etapa de ingeniería de requerimientos del software, de tal forma que se puedan agregar más funcionalidades a las herramientas ya existentes para hacer más óptimo este proceso.

## Bibliografía

- BECK, K. (2000). *Extreme Programming Explained. Embrace Change*, Pearson Education, 1999. Traducido al español como: *Una explicación de la programación extrema. Aceptar el cambio*. Addison Wesley.
- BENNINGTON. (1956). *Agile requirements method*. [en línea]. [Consulta: Mayo 2006]. Disponible en: <<http://www.therationaledge.com/content/agileRequirements>>
- BOEHM, Barry. (1998). *Spiral model of software development and enhancement*, IEEE computer, Vol. 21.
- CMMI. (2002). Capability Maturity Model® Integration (CMMISM), Version 1.1 MMI/SE/SW/IPPD/SE, V1.1; [en línea]. [Consulta: Agosto 2006]. Disponible en: <<http://www.sei.cmu.edu/cmmi/>>
- COCKBURN, Alistair. (1999). *Metodologías ágiles y tradicionales en el desarrollo de software*. Patricio Letelier y M<sup>a</sup> Carmen Penadés, Universidad Politécnica de Valencia.
- CHOQUE ASPIAZU, Guillermo. (2001). *Ingeniería de requerimientos*. [en línea]. [Consulta: Julio 2006]. Disponible en: <[http://www.umsanet.edu.bo/docentes/gchoque/Art\\_IngRequerim.pdf](http://www.umsanet.edu.bo/docentes/gchoque/Art_IngRequerim.pdf)>
- FOWLER, Martin. (2001). *The new Methodology*.
- CARROL, J.M. (1995). *Scenario-Based Design: Envisioning Work and Technology in System Development*. Wiley. New York.
- GRUDIN, J. (1990). *Obstacles to user involvement in interface design in large product development organization*, Proceeding IFIP INTERACT'90 Third International Conference on Human-Computer Interaction, Cambridge, Inglaterra.
- ISO 9000, 9002, 9003. [en línea]. [Consulta: Agosto 2006]. Disponible en: <<http://www.iso.org/iso/en/iso9000-14000/index.html>>
- JACOBSON, I.; CHRISTERSON, M.; JONSSON, P. y OVERGAARD, G. (1992). *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison – Wesley, Reading, MA.
- JEFFRIES, R., ANDERSON, A., HENDRICKSON, C. (2001). *Extreme Programming Installed*. Addison-Wesley.
- HIGHSMITH, Jim. (2000). *Metodologías ágiles y tradicionales en el desarrollo de software*. Patricio Letelier y M<sup>a</sup> Carmen Penadés. Uni-

- versidad Politécnica de Valencia. [en línea]. [Consulta: septiembre 2006]. Disponible en: <<http://www.willydev.net>>
- JOHNSON, P. (1992). *Human Computer Interaction: Psychology. Task Analysis and Software Engineering*. Londres: McGraw Hill Int.
  - MACAULAY, M. (1996). *Requirements engineering*. Springer Verlag, Londres.
  - Mc CONNELL, Agile. (1996). *Requirements Methods*. [en línea]. [Consulta: julio 2006]. Disponible en: <<http://www.therationaledge.com/content/agileRequirements>>
  - MERCHAN, L., URREA, A. (2007). *Caracterización de las Empresas Pertenecientes a la Industria Emergente de Software del Sur Occidente Colombiano Caso Red de Parques PARQUESOFT*. Congreso Colombiano de Computación - CCC 2007. Pontificia Universidad Javeriana, Bogotá D.C.
  - KARIM PICKLER, Kênia. GRESSE VON WANGENHEIM, Christiane. SALVIANO, Clenio F. *Projeto 4.05 Propondo um método de avaliação de processo de software em micro empresas incubadas*. [en línea]. [Consulta: febrero de 2007]. Disponible en: <[www.mct.gov.br/upd\\_blob/5011.pdf](http://www.mct.gov.br/upd_blob/5011.pdf)>
  - MOPROSOFT. *Modelo de Procesos para la Industria de Software. MoProSoft Versión 1.1 Mayo 2003*. [en línea]. [Consulta: julio de 2006]. Disponible en: <<http://www.lania.mx/biblioteca/manuales/moprosoft>>
  - NEWKIRK, J.; MARTIN, R.C. (2004). *Extreme Programming in Practice*. Addison-Wesley.
  - NIELSEN J. y R. L. MACK (eds.), *Usability Inspection Methods*. Wiley. New York, 1994.
  - MARKC C.; PAULK. A. (1994). *Comparison of ISO 9001 and the Capability Maturity Model for Software*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Technical Report CMU/SEI 94-TR-12.
  - PAECH, B. (1998). *The four levels of use case description*, 4th Int. Workshop on Requirements Engineering: Foundations for Software Quality, Pisa.
  - PIATTINI, M., – DARYANANI, Rama. (1995). *Elementos y Herramientas en el Desarrollo de los Sistemas de Información*.
  - PRESSMAN, R. (2005) *Software engineering*. 6a edición. McGraw hill.
  - RUBIN, J. (1994). *Handbook of usability testing*. Wiley. New York.
  - RATIONAL UNIFIED PROCESS. RUP. (1998). [en línea]. [Consulta: julio de 2006]. Disponible en: <[http://www.therationaledge.com/content/jan\\_01/t\\_rup\\_ge.html](http://www.therationaledge.com/content/jan_01/t_rup_ge.html)>
  - WIRFS-BROCK, R. & WILKERSON, B. & WIENER, L. (1990). *Designing Object-Oriented Software*, Prentice Hall.
  - SAIEDIAN H. DALE, R. (1999). *Requirements Engineering: Making the connection between the software developer and customer*. Department of Computer Science – University of Nebraska.
  - SCHWABER . *Agile requirements methods*. [en línea]. [Consulta: febrero de 2006]. Disponible en: <<http://www.therationaledge.com/content/agileRequirements>>
  - SOMMERVILLE, I. *Integrated Requirements Engineering: A Tutorial*. [en línea]. [Consulta: marzo de 2006]. Disponible en: <<http://www.computer.org/software>>
  - SWEBOK. *Guide to the Software Engineering Body of Knowledge*. Version 2004. [en línea]. [Consulta: febrero de 2006]. Disponible en: <[http://www.swebok.org/ironman/pdf/SWEBOK\\_Guide\\_2004.pdf](http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf)>
  - WAKE, W.C. (2002). *Extreme Programming Explored*. Addison-Wesley.