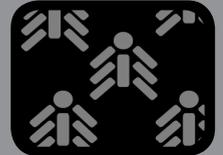


VALIDACIÓN DEL MODELO LIVIANO DE GESTIÓN DE REQUISITOS PARA PEQUEÑAS EMPRESAS DE DESARROLLO DE SOFTWARE

LIGHT MODEL VALIDATION MANAGEMENT
REQUIREMENTS FOR SMALL SOFTWARE
DEVELOPMENT



AUTOR

LUIS MERCHÁN PAREDES
PhD. En Dirección de Proyectos
*Universidad San Buenaventura
Director programa de Ingeniería de Sistemas
lmerchan@usbcali.edu.co
COLOMBIA

INSTITUCIÓN

*UNIVERSIDAD SAN BUENAVENTURA
USB Cali
Avenida 10 de Mayo, La Umbría, Vía a Pance
PBX: 488 22 22 - 318 2200 / FAX: 4882231 / A.A. 7154
y 25162
informacion@usbcali.edu.co
www.usbcali.edu.co
Valle del Cauca, Cali – COLOMBIA

RECEPCIÓN: Abril 14 de 2010

ACEPTACIÓN: Junio 15 de 2010

TEMÁTICA: Marcos de Trabajo y Desarrollo de Requisitos de Sistemas y Software

TIPO DE ARTÍCULO: Artículo de investigación científica y tecnológica

RESUMEN ANALÍTICO

Actualmente las pequeñas empresas de desarrollo de software están abordando uno de sus mayores retos: constituirse en industria; es decir, pasar de desarrollos puntuales y muchas veces artesanales a industrializar toda su producción de software y para ello requieren del conocimiento y aplicación de los modelos y metodologías de calidad que tiene esta industria. Estas empresas, conformadas hasta por nueve personas, no logran aplicar estos modelos pues su gran obstáculo se observa en los altos costos, los prolongados tiempos de implementación y el gran esfuerzo humano.

El Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software (LIDIS), siendo consecuente con esta situación, adelantó una investigación con el objetivo de proponer y validar un modelo liviano de gestión de requisitos apropiado al contexto de las pequeñas empresas que, basado en los estándares internacionales, les permitiera la aplicación de buenas prácticas a sus procesos de desarrollo de software.

El modelo fue validado en nueve proyectos de tres empresas desarrolladoras de software. El artículo presenta inicialmente la caracterización de las empresas pequeñas de software, luego muestra el modelo propuesto y posteriormente el diseño experimental con sus respectivos resultados que confirman la validez del modelo.

PALABRAS CLAVES: Gestión de requisitos, Proyectos de desarrollo de software, Empresas de software

ANALYTICAL SUMMARY

Currently small software development companies are addressing one of its greatest challenges: industry become, that is, moving from specific developments and often craft to industrialize their entire production of software and therefore require knowledge and application of models and quality methodologies this industry has. These companies made up to nine people, fail to implement these models as their major obstacle is observed in the high costs, long implementation times and the great human endeavor.

The Research Laboratory for the Development of Software Engineering (LIDIS), being consistent with this situation, advanced research with the aim to propose and validate a lightweight requirement management model appropriate to the context of small businesses, based on international standards, allowed the application of good practices to their software development processes.

The model was validated in nine projects in three software development companies. The article initially presents the characterization of small business software, then shows the proposed model and then design their own experimental results confirming the validity of the model.

INTRODUCCIÓN

Un factor estratégico para las empresas desarrolladoras de software se relaciona con la aplicación de modelos de mejoramiento de procesos, que puedan disminuir los costos de producción y la inversión de recursos en el soporte y mantenimiento de sus productos.

Modelos como el Capability Maturity Model Integration (CMMI®) [1] del Software Engineering Institute (SEI) o el ISO/IEC-15504 Standard of the International Organization for Standardization (ISO®), conocido como SPICE (Software Process Improvement and Capability dEtermination) [2] aplicados por la industria del software, además de ser extranjeros y ajenos a las condiciones y características de las empresas pequeñas, son extensos y difíciles de entender para éstas y los servicios de capacitación, consultoría y evaluación son costosos.

Estos modelos, pensados principalmente para empresas medianas y grandes [3], difícilmente pueden ser aplicados en empresas pequeñas debido a que su implementación supone una gran inversión en tiempo y recursos, además de la alta complejidad de las recomendaciones y el largo plazo del retorno de la inversión, a la vez que no contemplan las necesidades específicas de las empresas en su contexto [6]. De

hecho, no existe un marco de procesos universal que pueda ser aplicado a diferentes contextos y diferentes tipos de empresas [4].

En Colombia, la Red Colombiana de Calidad de Software (RCCS) está adelantado un proyecto que busca brindar apoyo a las pymes en la implementación de modelos de calidad y para ello toma como referente a CMMI®. Con el proyecto, se pretende que "18 empresas lleven a cabo una valoración (CMMI – Based Appraisal) reconocida por el Software Engineering Institute -SEI-, bajo el método Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM), Tipo A, que define el nivel de calidad de sus procesos, de acuerdo con el Modelo CMMI - DEV V 1.2." [5]

Sin embargo, este esfuerzo no es suficiente para orientar procesos de mejoramiento en las pequeñas empresas de software, las cuales por lo general atienden mercados o nichos de productos que por sus características de tamaño e innovación no son ofrecidos por las empresas medianas y grandes [7, 8, 9]. Lo anterior ha promovido personalizaciones de modelos tales como TOPS (Toward Organized Software Processes in SMEs), SPIQ (Software Process for better Quality), MESOPYME (Modelo de mejora del proceso de Software orientado a la pyme), SPM (Software Process Matrix) y COMPETISOFT, entre otros [10].

El LIDIS, consciente de esta necesidad, definió un modelo de mejoramiento de procesos para el contexto de las empresas pequeñas de software y como resultado de un trabajo de priorización [11], en el que se contemplaron las siete áreas de procesos correspondientes al nivel dos de CMMI, se consideró el área de requisitos como la más crítica por las empresas.

El modelo implementado es más liviano, flexible y adaptable a empresas o áreas internas de desarrollo de software e intenta ser un modelo que se adapte y crezca en el tiempo. Liviano porque se considera más fácil de usar y no enfatiza en la planeación y documentación detallada como si lo hacen los modelos tradicionales. También se distingue por ser flexible en la medida que permite responder a los cambios que se presentan en forma constante en los procesos de desarrollo de software.

El modelo fue estructurado a través de las etapas de elicitación, especificación y gestión.

1. CARACTERÍSTICAS DE LA GESTIÓN DE REQUISITOS EN LAS EMPRESAS PEQUEÑAS

Estas empresas de software, por sus particularidades a la luz de los estándares, han venido ganando terreno en el reconocimiento de sus limitantes. Para ello el Standards for the International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/Sub Committee 7 (ISO/IEC JTC 1/SC7) ha establecido el Work Group 24 (WG24) con el ánimo de propender por modelos de madurez enfocados principalmente a empresas que cuentan con muy poco personal y recursos. Es así como también se acuña el nombre de Very Small Enterprise (VSE) [12].

En la reunión de Thailandia, en mayo de 2006, el WG24 estableció dos perfiles para empresas: de 1 a 9 y de 10 a 25 empleados [13], enviando así un mensaje a la comunidad internacional sobre la necesidad de trabajar estándares, modelos, técnicas y herramientas que soporten a las empresas pequeñas.

En un estudio de caracterización de la industria local (sur occidente colombiano), adelantado por el LIDIS, se estableció que el 87% de las 41 empresas que fueron diagnosticadas, estaban conformadas por menos de nueve empleados y presentaban características similares respecto a la informalidad en los procesos que aplicaban y a la forma en que dirigían y organizaban sus actividades [14].

El mismo estudio, en cuanto a la gestión de requisitos presentó el siguiente diagnóstico:

- El 44,89% de las empresas en estudio ignoran (no hacen o no saben) los criterios para la aceptación de proveedores de requisitos. Igualmente, el 48,97% ignora los criterios para la aceptación de requisitos.
- En los criterios para la priorización de requisitos, el 48,97% los ignoran (no hacen o no saben).
- El 36,73% ignora el historial de cambios en los requisitos y el 46,93% ignora la administración y gestión de cambios en los requisitos.
- El 53,06% ignora la administración de la trazabilidad.
- El 46,93% ignora los mecanismos de verificación y control de requisitos.

En cuanto a la aplicación del modelo CMMI®, las empresas estudiadas manifiestan que es necesario por la estructura y certificación que ofrece, pero requiere muchos recursos y tiempo someterse a un proyecto de mejora bajo sus lineamientos [14].

En lo referente a los procesos en el desarrollo de software, la caracterización [14] logró identificar debilidades en las pequeñas empresas al compararlas con las establecidas por el modelo de CMMI en las áreas clave de nivel dos, y particularmente en el proceso de gestión de requisitos se pudo establecer que no tienen procedimientos formales por lo que en determinados proyectos usan unos y en otros no, generando muchas veces conflictos en la documentación y en el uso de técnicas y herramientas de soporte. Así mismo, se entendió que entre todas las áreas (propuestas por CMMI), la de requisitos era la que más se necesitaba adaptar a las necesidades del contexto [14].

En razón a lo anterior se decidió adelantar una investigación que condujera a un modelo liviano de gestión de requisitos que permitiera a las pequeñas empresas desarrolladoras de software, apropiarlo considerando su particular contexto pero sin perder de vista los referentes que tarde o temprano tendrán que implementar para procesos de certificación que les permita acceder fácilmente a mercados nacionales e internacionales.

2. MODELO PROPUESTO

2.1 ALCANCE

El modelo de gestión de requisitos tiene como objetivo asegurar la definición de los requisitos e identificar y corregir las inconsistencias de los mismos a partir de una serie de etapas que permitan comprender las necesidades de los clientes y brindar una solución sin ambigüedad, que responda a las expectativas de las partes.

2.2 ETAPAS DEL MODELO

El modelo ha sido estructurado en tres etapas principales:

Elicitación: El propósito de esta etapa (Figura 1) es obtener un óptimo entendimiento de los requisitos del software entre los clientes y los desarrolladores e identificar los requisitos funcionales y no funcionales.

Figura 1: Actividades etapa de elicitación

Etapa 1: Elicitación
AR-1: Planificación de elicitación
AR-2: Evaluación de proveedores de requisitos
AR-3: Elección de estrategia de recolección de requisitos
AR-4: Identificación de los procesos del sistema
AR-5: Identificación de requisitos funcionales y no funcionales

Especificación: El propósito de esta etapa (figura 2) es describir cada uno de los requisitos del software, sus políticas y sus restricciones, para asegurar la calidad del producto final mediante las validaciones correspondientes.

Figura 2: Actividades etapa de especificación

Etapa 2: Especificación
AR-6: Descripción detallada de requisitos
AR-7: Validación de requisitos

Gestión: El propósito de esta etapa (figura 3) es administrar efectivamente los cambios a los requisitos, encontrar y mantener las relaciones entre ellos e identificar las inconsistencias que puedan ocurrir en el transcurso del desarrollo del software.

Figura 3: Actividades etapa de gestión

Etapa 3: Gestión
AR-8: Priorización de requisitos
AR-9: Administración trazabilidad de requisitos
AR-10: Asignación de requisitos al grupo de desarrollo

La Tabla 1 muestra una relación de las prácticas y sub prácticas dadas por el referente CMMI con las etapas y actividades que se tomaron para el diseño del modelo propuesto.

Tabla 1: Comparativo de CMMI y el modelo propuesto

Prácticas y sub prácticas del referente	Etapa y actividades del modelo
PE 1.1 Obtener entendimiento de requisitos	Etapa de elicitación
SubP1. Establecer criterios para distinguir los proveedores de requisitos apropiados.	AR-1. Planificación de elicitación. AR-2. Evaluación de proveedores de requisitos.
SubP2. Establecer criterios objetivos para aceptar los requisitos.	AR-3. Elección de estrategia de recolección de requisitos.

<p>SubP3. Analizar los requisitos para asegurar que los criterios establecidos son cumplidos.</p> <p>SubP4. Llegar a tener un entendimiento de los requisitos con los proveedores de manera que los participantes del proyecto pueden comprometerse con ellos.</p> <p>PE 1.2 Obtener aseguramiento de requisitos</p> <p>SubP1. Evaluar el impacto de los requisitos sobre compromisos existentes.</p> <p>SubP2. Negociar y grabar compromisos.</p>	<p>AR-4. Identificación de los procesos del sistema.</p> <p>AR-5. Identificación de requisitos funcionales y no funcionales.</p>	<p>SubP1. Mantener la rastreabilidad de los requisitos para garantizar que la fuente de los requisitos de bajo nivel (derivados) este documentada.</p> <p>SubP2. Mantener la rastreabilidad de un requerimiento hacia sus requisitos derivados como también a su asignación de funciones, objetos, personas, procesos, y productos de trabajo.</p> <p>SubP3. Mantener trazabilidad horizontal de función a función, y a través de interfaces.</p> <p>SubP4. Generar la matriz de rastreabilidad de requisitos.</p>	<p>AR-9. Administración trazabilidad de requisitos.</p>
<p>PE 1.3 Administrar cambios a requisitos</p> <p>Sub P1. Capturar todos los requisitos y todos los cambios a requisitos que son dados o generados por el proyecto.</p> <p>SubP2. Mantener la historia de los cambios en los requisitos con la razón de los cambios.</p> <p>SubP3. Evaluar el impacto de los cambios en los requisitos desde el punto de vista de los stakeholders relevantes.</p> <p>SubP4. Hacer los requisitos y sus cambios, disponibles para el proyecto.</p>	<p>Etapas de especificación</p> <p>AR-6. Descripción detallada de requisitos.</p> <p>AR-7. Validación de requisitos.</p>	<p>PE 1.5 Obtener inconsistencias en requisitos</p> <p>SubP1. Revisar los planes de trabajo, actividades, y productos de trabajo para consistencia con los requisitos y los cambios hechos a ellos.</p> <p>SubP2. Identificar la fuente de las inconsistencias y la razón.</p> <p>SubP3. Identificar cambios que tengan que ser hechos para planes y productos de trabajo resultado de los cambios hechos a la línea base de requisitos.</p>	<p>AR-8. Priorización de requisitos.</p>
<p>PE 1.4 Mantener trazabilidad bidireccional a requisitos</p>	<p>Etapas de gestión</p>	<p>SubP4. Iniciar acciones correctivas.</p>	<p>AR-10. Asignación de requisitos al grupo de desarrollo.</p>

En resumen, en el modelo propuesto se conservan las cinco prácticas específicas (PE) y se pasa de dieciocho subprácticas específicas (SubP) a diez actividades (AR).

3. DISEÑO DEL EXPERIMENTO DE VALIDACIÓN

Una vez definido el modelo con sus entradas, salidas, actividades, controles y plantillas de soporte, se seleccionó el marco experimental sobre el cual se adelantaría en forma ordenada y sistemática la respectiva validación del modelo.

Con el diseño y ejecución del experimento se esperaba confirmar que el modelo de gestión de requisitos propuesto fuera viable y se ajustara a pequeñas empresas de software, lo cual se esperaba lograr a través de los resultados que se obtuvieran de las métricas establecidas.

Para aplicar el experimento se tuvieron en cuenta los siguientes conceptos [15]:

3.1 UNIDAD EXPERIMENTAL

Objeto o espacio al cual se le aplica el experimento y donde se mide y analizan las métricas que se investigan. Se consideró como unidad el proceso de gestión de requisitos.

3.2 SUJETOS EXPERIMENTALES

Son aquellas personas u objetos sobre los que se va a llevar a cabo el experimento. Como sujetos se consideraron las tres empresas que participan en el experimento mediante la implementación del modelo.

3.3 FACTORES

Representan las características de la unidad experimental que se toman para el experimento. Los valores que pueden tomar los factores se conocen como niveles.

El experimento tomó dos factores: el tamaño del proyecto de software y la experiencia de los sujetos experimentales en gestión de requisitos.

El tamaño de un proyecto de software es un factor diferenciador para clasificarlo de acuerdo con el esfuerzo estimado. Este tamaño se clasifica en tres niveles:

- *Nivel 1 (Proyectos pequeños):* Son los proyectos de software que emplean un tiempo menor a tres meses durante el ciclo de vida.
- *Nivel 2 (Proyectos medianos):* Son los proyectos de software que emplean entre tres y nueve meses durante el ciclo de vida de su desarrollo.

- *Nivel 3 (Proyectos grandes):* Son los proyectos de software que emplean un tiempo mayor a nueve meses durante el ciclo de vida de su desarrollo.

La experiencia de los sujetos experimentales se entiende como la práctica prolongada que proporciona el conocimiento o habilidad de quienes intervienen en el experimento y se clasifica en tres niveles.

- *Nivel 1 (Baja experiencia):* Se presenta en las empresas que no tienen institucionalizado el proceso de gestión de requisitos durante el ciclo de vida de desarrollo.
- *Nivel 2 (Mediana experiencia):* Se presenta en las empresas que definen un proceso formal para gestionar los requisitos o que el producto tiene definido un roadmap y se aplica la medición como mecanismo de control.
- *Nivel 3 (Alta experiencia):* Se presenta en las empresas que definen un proceso formal para gestionar los requisitos y se aplica la medición como mecanismo de control; o en las que incluyen prácticas maduras para el control de cambios y utilizan los resultados de la medición para identificar tendencias y tomar decisiones respecto a nuevos proyectos y mejora de las prácticas existentes.

No se tomó como factor la estructura organizacional en razón a que el estudio de caracterización [14] presentó que el 92% de las empresas no tienen una estructura organizacional definida y que el 87% de ellas están compuestas por menos de nueve empleados, de los cuales el 67% son ingenieros de sistemas.

Tampoco la antigüedad y la experiencia comercial fueron tomadas como factor en razón a que el mismo estudio presentaba que el 59,19% de las empresas tenían en promedio tres años de antigüedad y el 82% de los proyectos no superaban en presupuesto los US\$ 50.000.

3.4 MÉTRICAS

Sistema que permite medir las variables de un modelo para darle pertinencia y validez.

De acuerdo con la anterior definición, se estableció que la investigación adelantada requería validar que si una empresa aplicaba apropiadamente el modelo, el número de requisitos defectuosos debería disminuir. De la misma manera, interesaba saber qué factores afectaban la estabilidad de los requisitos y su relación con la aplicación del modelo propuesto y determinar la aplicabilidad total o parcial de sus procesos y procedimientos. Finalmente, si los requisitos son el eje central del desarrollo del software, una mejora en este proceso debería repercutir en la productividad.

En razón a lo anterior se consideraron cuatro métricas: requisitos defectuosos, madurez de requisitos, completitud del modelo y por último, productividad del modelo.

Tabla 2: Métrica para requisitos defectuosos

VARIABLES	DESCRIPCIÓN
Definición	Porcentaje de requisitos defectuosos de la línea base de requisitos definida.
Propósito	Medir la estabilidad de los requisitos.
Fórmula	$\frac{[RD]}{RT} \times 100$ RD = Número de requisitos defectuosos. RT = Número de requisitos totales.
Variables	Número de requisitos defectuosos: Para la obtención de esta métrica es necesario que los requisitos primero estén definidos y luego especificados, y así se validan de acuerdo con los criterios establecidos en la lista de chequeo. Con la lista de chequeo de cada requerimiento se identifica si es defectuoso una vez no cumpla con alguno de los criterios establecidos.
Medición	Número de requisitos totales: Por cada proyecto a partir del listado de especificación de requisitos se toma el total de requisitos definidos. Se debe tomar para cada aplicación del modelo.

Tabla 3: Métrica para madurez de requisitos

VARIABLES	DESCRIPCIÓN
Definición	Índice de madurez en que se encuentran los requisitos.
Propósito	Medir la estabilidad de los requisitos.
Fórmula	$\frac{[RT - (RC + RA + RE)]}{RT}$ RT = Número de requisitos totales. RC = Número de requisitos cambiados. RA = Número de requisitos añadidos. RE = Número de requisitos eliminados.

Variables

Número de requisitos totales: Por cada proyecto a partir del listado de especificación de requisitos se toma el total de requisitos definidos.

Número de requisitos cambiados, añadidos o eliminados: La métrica se obtiene a través de la plantilla de solicitud de cambios, por lo que es necesario que los requisitos se encuentren definidos, especificados y validados. Por cada solicitud de cambio, y según el tipo de éste, se calculan las métricas.

La métrica se toma para cada aplicación del modelo y no para el producto completo, teniendo en cuenta:

- La primera versión del listado de requisitos.
- Versiones posteriores del listado de requisitos para cada fase o para cada versión.
- La plantilla de cambios en los requisitos.

Medición

Tabla 4. Métrica para la completitud del modelo

VARIABLES	DESCRIPCIÓN
Definición	Porcentaje que se ha seguido en la ejecución del modelo a través de sus etapas definidas.
Propósito	Medir el estado de completitud de las etapas y actividades establecidas en el modelo de gestión de requisitos.
Fórmula	$\frac{[TAR]}{[TXR]}$ TAR = Total de actividades realizadas. TXR = Total de actividades a realizar.
Variables	Total de actividades realizadas: La métrica se calcula siguiendo las actividades del modelo y revisando cuáles se llevaron a cabo. Total de actividades a realizar: La métrica comprende todas las actividades del modelo, sin importar si se ejecutaron o no.

Medición	<p>La métrica se toma a medida que se va ejecutando el modelo de gestión de requisitos y se debe considerar:</p> <ul style="list-style-type: none"> • La lista de las actividades del modelo de gestión de requisitos, ordenadas teniendo en cuenta las etapas. • El plan de actividades realizadas actualizado. 	<p>Se obtiene la cantidad de requisitos del listado de requisitos de software, la cantidad de analistas que trabajaron en su desarrollo y las horas y días de trabajo reales utilizados en la etapa de desarrollo.</p>
----------	--	--

Tabla 5: Métrica para productividad del modelo

Variables	Descripción
Definición	Cantidad de requisitos especificados en una unidad de tiempo.
Propósito	<p>Medir la productividad del equipo de trabajo en el proceso de desarrollo de los requisitos.</p> $\frac{RT}{NA \times (HT \times DT)}$
Fórmula	<p>RT = Número de requisitos totales. NA = Número de analistas. HT = Horas de trabajo diario. DT = Días de trabajo.</p> <p>Número de requisitos totales: Por cada proyecto a partir del listado de especificación de requisitos se toma el total de requisitos definidos.</p> <p>Número de analistas: La métrica se obtiene a partir de la información del personal involucrado en el proyecto de software y el rol que desempeñan.</p>
Variables	<p>Horas de trabajo diario: La métrica se obtiene mediante el cronograma de actividades, pero en vez del esfuerzo estimado, se toma el esfuerzo real.</p> <p>Días de trabajo: Los días de trabajo se toman del cronograma de actividades, revisando la fecha para ver los días en que realmente se trabajó.</p>

3.5 APLICACIÓN DEL EXPERIMENTO DE VALIDACIÓN

La validación en las empresas reales tuvo los siguientes condicionamientos:

- Cada empresa debía seleccionar tres proyectos basados en las características que se muestran a continuación.

Tabla 6: Características de los proyectos

Variables	Descripción
Nombre	Nombre del proyecto.
Tipo	Tipo de proyecto de desarrollo
Tamaño	Tamaño estimado del proyecto
Tipo de interacción con el cliente	Tipo de relación con el equipo de trabajo de la organización cliente.
Plataforma tecnológica	Plataforma tecnológica de desarrollo
Empresa cliente	Descripción de la empresa para la cual se desarrolla el proyecto de desarrollo de software
Alcance	Procesos aplicados (del ciclo de vida de los proyectos de software)
Tiempo	Duración estimada del proyecto

- Los proyectos debían ser diferentes pero conservando las características similares (alcance, recursos, tiempos y complejidad).

Se brindó a las empresas un acompañamiento metodológico en la aplicación del modelo.

- Las empresas debían aplicar el modelo en tres fases a saber:

- La primera, correspondía a la aplicación del modelo a un proyecto ya realizado en cada empresa, y se debía estimar los valores de las métricas a partir de su información histórica. Esta primera fase se denominó SM (Sin Modelo).
- La segunda, correspondía a la aplicación del modelo a un nuevo proyecto (diferente al anterior) en cada empresa. A esta segunda fase se le denominó CM1 (Con el Modelo, primera aplicación).
- La tercera, correspondía a la aplicación del modelo a un nuevo proyecto (diferente a los dos anteriores). A esta fase se le denominó en particular CM2 (Con el Modelo, segunda aplicación). Es preciso anotar que en esta fase era decisión de cada empresa aplicar el modelo en su totalidad o parcialmente.

4. RESULTADOS Y ANÁLISIS

4.1 RESULTADOS EN CUANTO A REQUISITOS DEFECTUOSOS

$$\text{Cálculo: } \frac{[RD]}{RT} \times 100$$

NR = Número de requisitos defectuosos

RT = Número de requisitos totales

A continuación se visualiza la tabulación para la obtención de la métrica para requisitos defectuosos y los respectivos resultados para cada uno de los proyectos de software de las empresas (ver Tabla 7).

Tabla 7: Tabulación de la métrica requisitos defectuosos

Métrica	Empresa-1			Empresa-2			Empresa-3		
	SM	CM1	CM2	SM	CM1	CM2	SM	CM1	CM2
NR	10	4	3	3	2	2	10	6	3
RT	68	76	62	15	33	42	22	22	29
Variación	15%	5%	5%	20%	6%	5%	45%	27%	10%

A partir de la implementación del modelo en la segunda y tercera fase (CM1 y CM2), se puede apreciar la reducción en el número de requisitos defectuosos. Igualmente, en cuanto más inmadura esté la empresa en el proceso más impacto en la reducción se verá, como es el caso de la Empresa-3.

Para la Empresa-1 la mejora significó pasar de 10 a 3 requisitos defectuosos.

Para la Empresa-2 la mejora significó pasar de 3 a 2 requisitos defectuosos, pero se debe tener en cuenta que para la tercera fase el número de requisitos eran más del doble (pasó de 15 a 42).

Para la Empresa-3 la mejora significó pasar de 10 requisitos defectuosos a 3, a pesar de que el número de requisitos aumenta (de 22 a 29).

4.2 RESULTADOS EN CUANTO A MADUREZ DE REQUISITOS

$$\text{Cálculo: } \frac{[RT - (RC + RA + RE)]}{RT}$$

RT = Número de requisitos totales

RC = Número de requisitos cambiados

RA = Número de requisitos añadidos

RE = Número de requisitos eliminados

A continuación se muestra la tabulación y resultados para la métrica madurez de requisitos para cada uno de los proyectos de software de las empresas (Ver Tabla 8). A partir de la implementación del modelo en la segunda

Tabla 8: Tabulación de la métrica madurez de requisitos

Métrica	Empresa-1			Empresa-2			Empresa-3		
	SM	CM1	CM2	SM	CM1	CM2	SM	CM1	CM2
RT	68	76	62	15	33	42	22	22	29
RC	10	5	3	9	8	6	10	6	4
RA	12	8	5	2	2	2	6	1	2
RE	4	0	0	1	0	0	3	0	0
Variación	62%	83%	87%	20%	70%	81%	14%	68%	79%

y tercera fase (CM1 y CM2) se puede apreciar un avance en la madurez de requisitos en razón a que las etapas aplicadas por el nuevo modelo aseguran que los requisitos definidos sean en gran medida los desarrollados, dejando poco margen a nuevos requisitos o cambios en los mismos.

Para la Empresa-1 se presenta un incremento del índice de madurez de requisitos al pasar de 26 requisitos con problemas (RC+RA+RE) a sólo 8.

Para la Empresa-2 se presenta un incremento del índice de madurez de requisitos al pasar de 12 requisitos con problemas a sólo 8, pero teniendo presente que se pasó de 15 a 42 requisitos.

Para la Empresa-3 se presenta un representativo incremento del índice de madurez de requisitos ya que de un total de 29 (antes eran 22, de los cuales, 19 presentaban problemas) sólo 6 tienen problemas.

4.3 RESULTADOS EN CUANTO A COMPLETITUD DEL PROCESO

$$\text{Cálculo: } \frac{[RD]}{RT} \times 100$$

TAR = Total de actividades realizadas

TXR = Total de actividades a realizar

A continuación se muestra la tabulación y resultados para la métrica completitud del proceso para cada uno de los proyectos de software de las empresas (Ver Tabla 9).

A partir de la implementación del modelo propuesto en la segunda y tercera fase (CM1 y CM2) se puede apreciar un gran logro en la completitud del proceso en cuanto a que las etapas y actividades definidas en el modelo se convierten en estándares para el proceso y por ende aplicadas correctamente en el desarrollo de requisitos.

Los beneficios obtenidos en la primera fase (CM1) para las tres empresas hicieron que la aplicaran al 100% en la segunda fase (CM2) con resultados igualmente satisfactorios.

4.4 RESULTADOS EN CUANTO A PRODUCTIVIDAD

$$\text{Cálculo: } \frac{RT}{NA \times (HT \times DT)}$$

RT = Número de requisitos totales

NA = Número de analistas

HT = Horas de trabajo diario

DT = Días de trabajo

En la siguiente tabla se encuentra el tabulado y resultados para la métrica productividad en el desarrollo de los requisitos para cada uno de los proyectos de software de las empresas (Ver Tabla 10).

Tabla 9: Tabulación de la métrica completitud del proceso

Métrica	Empresa-1			Empresa-2			Empresa-3		
	SM	CM1	CM2	SM	CM1	CM2	SM	CM1	CM2
TR	5	10	10	6	10	10	4	10	10
TA	10	10	10	10	10	10	10	10	10
Variación	50%	100%	100%	60%	100%	100%	40%	100%	100%

Tabla 10: Tabulación de la métrica productividad

Métrica	Empresa-1			Empresa-2			Empresa-3		
	SM	CM1	CM2	SM	CM1	CM2	SM	CM1	CM2
TR	68	76	62	15	33	42	22	22	29
NA	3	2	2	2	2	2	1	1	1
HT	14	10	10	5	5	5	6	5	5
DT	15	25	18	30	44	39	20	20	17
Variación	11%	15%	17%	5%	7%	11%	18%	22%	34%

Se puede apreciar el aumento en la productividad en razón a una mejor planificación del recurso humano. Es decir, se logra al afinar lo planificado con lo ejecutado.

Para la Empresa-1 la productividad se refleja en un analista menos y unas jornadas de trabajo menores.

Para la Empresa-2 la productividad se refleja en que se aumenta el tamaño del software a desarrollar (número de requisitos) y su complejidad, manteniendo el mismo número de analistas e igual jornada laboral.

Para Empresa-3 la productividad se logra al mantener el mismo equipo de trabajo, reduciendo a su vez la jornada laboral.

5. FUTURAS INVESTIGACIONES

Deja la investigación dos líneas de trabajo que deben ser abordadas: Una primera, aplicar nuevamente el modelo propuesto en un área de impacto diferente (empresas medianas, por ejemplo). Una segunda, adelantar nuevos modelos livianos en otras áreas de proceso claves (estimación y gestión de riesgos) que fueron igualmente priorizadas en el estudio de diagnóstico realizado [14].

6. CONCLUSIONES

Después de investigar diferentes modelos existentes que tratan el tema de mejora de procesos en gestión de requisitos para empresas de desarrollo de software, se puede concluir que no son los más indicados para su implementación en empresas pequeñas de software, aclarando que no son los mejores debido a que en el caso colombiano, y en particular las empresas pequeñas (entre 1 y 9 personas), manejan otro nivel de recursos que para un país en vía de desarrollo son muy costosos y difíciles de obtener.

El modelo generado tiene procesos y prácticas que son comunes con otros modelos enfocados hacia las empresas pequeñas, siendo importante precisar que así se utilicen procesos iguales a nivel mundial, este modelo está orientado a actividades que permiten una mejor administración del desarrollo de software debido a que se remiten al contexto Colombiano y estrictamente a empresas que se encuentran en proceso de maduración. Se genera así en el sur occidente colombiano una cultura de calidad, sin necesidad de contar con grandes recursos tanto humanos como económicos y tecnológicos.

A nivel de venta de servicios, las empresas participantes en el proyecto manifestaron que los clientes percibían que eran "buenas y serias" al valorar propuestas soportadas en procesos formales para acometer las tareas de requisitos (proceso de mayor interacción con el cliente).

El modelo propuesto abre un camino hacia la transferencia de buenas prácticas de industria a las empresas pequeñas, esperando que estos resultados permitan aplicar el modelo en otros escenarios, tanto nacionales como internacionales.

7. RECONOCIMIENTOS

Se reconoce a las empresas la participación en la presente investigación. Igualmente a los estudiantes Jennifer Robles y Haiver Delgado por el trabajo de aplicación de instrumentos del trabajo de campo.

8. REFERENCIAS

- [1] SEI Capability Maturity® Integration (CMMi), Version 1.1., CMMi for Software Engineering (CMMi-SW, V1.1) Continuous Representation, CMU/SEI-2002-TR-028, Software Engineering Institute, Pittsburgh, 2002.
- [2] ISO/IEC 15504:2003-06, Information Technology – Software Process Assessment (parts 1-9), International Organization for Standardization.
- [3] PINO, Francisco. GARCIA, Félix. PLATINI, Mario. Revisión sistemática de procesos de software en micros, pequeñas y medianas empresas. Revista española de innovación, calidad e ingeniería de software. Vol 2 No. 1. 2006. pp. 6-23.
- [4] ACUÑA, Sivia. FERRE, Xavier. LÓPEZ, Marta y MATE, Luis. The Software Process: Modeling, Evaluation and Improvement. Handbook of Software Engineering and Knowledge Engineering. World Scientific Publishing Company, Argentina, 2000.
- [5] RCCS. Red Colombiana de Calidad de Software. (Colombia), (cited 20 may.,2010). http://rccs.cidlisuis.org/index.php?option=com_content&task=view&id=12&Itemid=20
- [6] ANACLETO, Alessandra. GRESSE VON WANGENHEIM, Christiane. SALVIANO, Clenio and SAVI, Rafael. A method for process assessment in small software companies. En: International SPICE Conference on process assessment and improvement. (4th: 2004: Portugal).
- [7] PETERSSON, Frederik. IVARSSON, Martin. GORSCHKE, Tony and OHMAN, Peter. A practitioner's guide to light weight software process assessment and improvement planning. The Journal of Systems and Software. Volume 81 , Issue 6. 2008. pp. 972-995

- [8] GRESSE VON WANGENHEIM, Christiane. WEBER, Sergio. HAUK, Jean and TRENTIN, Gisele. Experiences on establishing software processes in small companies. *Information and software technology*. Volume 48 , Issue 9. 2006. pp. 890-900.
- [9] HABRA, Najj. ALEXANDRE, Simon. DESHARNAIS, Jean-Marc. LAPORTE, Claude. And RENAULT, Alain. Initiating software process improvement in very small enterprises: Experience with a light assessment tool. *Information and Software Technology*, Volume 50, Issues 7-8. 2008. pp 763-771
- [10] MAS, Antonia. y AMENZUEL, Esperanza. Modelo de mejora de procesos aplicados a Pymes. En revista Española de Innovación, Calidad e Ingeniería de Software, Vol 1, No 2, p. 7-29. Madrid: España, 2005.
- [11] MERCHÁN, Luis., y URREA, Alba, REBOLLAR, Rubén. Definición de una metodología ágil de ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur occidente colombiano. En Revista Guillermo de Ockham, Vol. 6, No 1, p. 37-50. Cali: Universidad de San Buenaventura, 2008.
- [12] LAPORTE, Claude. APRIL, Alain y RENAULT, Alain. Applying ISO/IEC software engineering standards in small settings: historical perspectives and initial achievements. En: *SPICE Conference on Process Assessment and Improvement*. (6th:2006: Luxembourg).
- [13] LAPORTE, Claude. The Application of International Software Engineering Standards in Very Small Enterprises. En: *ENCUENTRO DE CALIDAD DE SOFTWARE*. (2006: Cartagena)
- [14] MERCHÁN, Luis y URREA, Alba. Caracterización de las empresas pertenecientes a la industria emergente de software del sur occidente colombiano Caso red de parques PARQUESOFT. En: *CONGRESO COLOMBIANO DE COMPUTACIÓN*. (2do:2007: Bogotá)
- [15] JURISTO, Natalia y MORENO, Ana. *Basics of software Engineering Experimentation*. Boston: Kluwer Academic Publisher, 2001. 395p.