

Oscar Eduardo Cumpe¹

¹ Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Escuela de Posgrado. Prof. titular de Economía y Financiación de Empresas (UdeMM)

La aplicación del Capability Maturity Model en empresas de software de Argentina *

Resumen

El objetivo de este trabajo es establecer el marco conceptual para la aplicación del Modelo de Capacidad de Madurez (CMM, por sus siglas en inglés) en empresas de *software* de Argentina, por medio de su interpretación teórica y la evaluación de los resultados alcanzados por su aplicación exitosa en empresas corporativas de otros países. Se plantea como hipótesis que, a través del entendimiento conceptual del modelo y el análisis de los resultados de su aplicación en empresas corporativas, es viable su aplicación en nuestro país con similares resultados. Metodológicamente, se propone la utilización del método inductivo a partir del análisis de casos particulares, de los cuales se extraen conclusiones de carácter general. Se utiliza una muestra que se compone de: a) encuesta y relevamiento sobre 500 empresas

(fuente CESSI); b) relevamiento de datos existentes en empresas corporativas (fuentes varias). Se completa el trabajo con la descripción completa del aplicativo "CMM'S Tool".

Palabras Claves: Reutilización de *Software*, Calidad de *Software*, Medición en Ingeniería de *Software*, Relaciones Costo-Volumen-Utilidad, Trabajo de Reingeniería.

Abstract

The main objective of this work is to establish the conceptual frame for the application of the Capability Maturity Model (CMM) in software companies of Argentina, through its theoretical interpretation and the assessment of the results by its successful application in corporate companies of other countries. The hypothesis formulates that, from the conceptual understanding of the model and the analysis of the results of its application in corporate companies, its application in our country is highly viable, with similar results. Methodologically, an inductive

method is used: from the analysis of particular cases, general conclusions are drawn. A sample is used for the inference, consisting of: a) a survey on 500 companies (source CESSI); b) a survey on existing data in corporate companies (different sources). The work is completed with a whole description of the "CMM'S Tool" software application.

Keywords: Managing Software Reuse, Software Quality, Software Engineering Measurement, Cost-Volume Profit Relationships, Reengineering Work

Introducción

Se da cuenta en el presente trabajo de un nuevo enfoque integral, sistemático, disciplinado y cuantitativo, que permite gestionar la maduración de los proyectos de *software*, y de una metodología que establece un lenguaje común en la organización y que hace posible construir un conjunto amplio de procesos y prácticas de desarrollo, proveer un marco de trabajo para priorizar acciones y efectuar

* Trabajo escrito por O. Cumpe, sobre la base de su tesis para optar al grado de Magíster en Administración de Negocios, bajo la dirección de P. Tolón Estarellés.

presupuestos confiables y consistentes. Otras contribuciones de importancia son: mejora en la reducción del tiempo de ciclo, aumento de la productividad, calidad y recuperación de la inversión, y a la vez brindar criterios de auto evaluación y mejoramiento continuo, que permiten alcanzar la madurez en el proceso organizacional.

Para lograr los beneficios antes enunciados, es necesario en primer lugar que la dirección de la organización comprenda e internalice que mejores prácticas se transforman en definitiva en mayores beneficios, que se reflejarán en los números y por lo tanto en la rentabilidad sobre la inversión. Logrado esto, es necesario que se adopte un modelo o una estructura de trabajo tendiente a alcanzar un proceso mejorado.

Debido a la explosión de las tecnologías de procesamiento de información y a la globalización que permite el auge de usuarios más informados, es cada vez más demandado un desarrollo de herramientas que den garantías de calidad. También es cada vez más apreciado por los usuarios de sistemas de información el saber reconocer cuando se está frente a un desarrollo que satisfaga requisitos mínimos de calidad y que permita ahorrar recursos en tiempo y costo.

Al respecto, la diversa información reunida en el desarrollo de este trabajo demuestra que en nuestro país la situación está en niveles iniciales. Sin embargo, también comprueba que, afortunadamente, existe preocupación en la materia y que habría una cantidad de empresas que se han ubicado alrededor de la necesidad de tener un proceso de desarrollo de *software* que provea un camino de mejora en el desempeño de la organización.

Todas las personas que se han enfrentado al desarrollo de productos de *software*, o bien los clientes que los solicitan, se han planteado cuánto dinero les costará hacerlo, cuánto dinero están perdiendo por malas estimaciones en los proyectos de desarrollo de *software*, o si se están desarrollando o requiriendo las aplicaciones que realmente se necesitan. Es decir, existe incertidumbre alrededor del proceso de desarrollo; es difícil predecir, difícil imaginar qué se necesita, en recursos, tiempo y dinero, y cual será el producto final obtenido. Hace ya mucho tiempo, y muy especialmente desde el advenimiento de la Ingeniería de *Software*, que este proceso dejó de ser una actividad basada en la intuición, la inspiración y la experiencia de algunos iluminados. Ésta debe ser hoy una actividad que se debe regir por los criterios y la rigurosidad de una disciplina como lo es la ingeniería. Por lo tanto, la aplicación de métodos, procedimientos y herramientas debe ser una constante para obtener los resultados esperados y, desde un punto de vista económico, proveer los adecuados retornos a las inversiones hechas en este campo. No hay por ahora otro camino conocido para asegurar algún grado de éxito en el proceso de desarrollo de *software*. Profesionalizar no significa solamente contar con profesionales con estudios formales en el área, sino que dichos profesionales se comporten como tales y pongan en práctica métodos, procedimientos y herramientas que la ingeniería de software y la tecnología actual ponen a nuestro alcance.

La alta gerencia de las organizaciones debe preocuparse de que los métodos y los procedimientos para el proceso de desarrollo de sistemas de información al menos existan, sean formales y se apliquen en cada uno

de los proyectos que se llevan a cabo, ya sea que se desarrollen internamente o que se le contraten a terceros. No se conoce por ahora otra forma de asegurar, por ejemplo, un cierto nivel de calidad en el *software* producido.

Actualmente, el proceso de desarrollo que incluye un conjunto de actividades, métodos, prácticas y transformaciones para desarrollar y mantener *software* y sus productos asociados, tales como planificación del proyecto, diseño de documentos, código, casos y manuales de usuarios, está centrado sólo en el producto, con lo cual se pierde la visión de cómo hacerlo bien. Cambiar el enfoque hacia el proceso permite predecir sus salidas y las características del producto y proyectar tendencias.

Mejorar en el proceso de *software* asegura mejoras cuantificables del producto de software. En consecuencia, se puede afirmar que un proceso de *software* mejorado implica un proceso y un producto de software mejorado. Una organización madura logrará incrementar la visibilidad del desempeño del proyecto, su calidad, la predicción de sus resultados, su habilidad para manejar complejidades y, finalmente, el ánimo del equipo de desarrollo.

En Argentina existe materia prima que muchos países no poseen: personas con excelentes aptitudes para el desarrollo de alta tecnología, que buscarán mejores horizontes a medida que las diferencias entre el contexto nacional y el exterior se hagan mayores, no sólo económicamente, sino también profesionalmente.

Las condiciones existen, y con creatividad, innovación, énfasis en la calidad y voluntad de cambio es posible avanzar y exportar productos y servicios de *software* al resto del mundo. Argentina, tanto en el ámbito público

como en el privado, debe decidir proactivamente, si quiere jugar un rol relevante dentro de la alta tecnología, o continuar siendo un seguidor de las tendencias mundiales y renegar crónicamente de su destino.

Desarrollo

A principios de la década del 80, una firma dedicada al estudio del mercado de Tecnologías de Información (The Standish Group, 2003) publica un informe sobre el éxito de los proyectos de desarrollo en la industria del *software*. El reporte, basado en encuestas hechas sobre proyectos de *software*, informaba los siguientes resultados estadísticos:

El 30 % de los proyectos se cancelaba.
 El 54 % de los proyectos excedía ampliamente los tiempos y los costos estimados.

El 16 % de los proyectos finalizaba exitosamente dentro del tiempo, el costo y la funcionalidad prevista.

En respuesta a la situación alarmante del momento, el Departamento de Defensa de los EE.UU. funda el SEI (Software Engineering Institute) (SEIR, 2003) en la Carnegie Mellon University, con el propósito de estudiar el problema y encontrar alguna solución.

En 1991, el SEI publica el modelo CMM (Capability Maturity Model) (CMM, 1991). El modelo está orientado a la mejora de los procesos relacionados con el desarrollo de *software*, para lo cual contempla las consideradas mejores prácticas de ingeniería de *software* y de gerenciamiento.

A partir de ese momento, el Departamento de Defensa exige que sus proveedores estén certificados en CMM, lo que impulsa a que el modelo tenga una amplia aceptación y se

convierta en un estándar de facto dentro de la industria del *software*.

El modelo CMM original define cinco niveles de madurez dentro de los cuales se puede encontrar una organización:

Nivel 1 – *Inicial*: el proceso de *software* es impredecible, sin control y reactivo. El éxito de los proyectos depende del talento de los individuos.

Nivel 2 – *Repetible*: existen procesos básicos de gestión de proyectos (costo, calendario, funcionalidad). Los procesos existentes hacen que se puedan repetir éxitos en proyectos de similares características.

Nivel 3 – *Definido*: existe un proceso de *software* documentado y estandarizado dentro de la organización. To-

dos los proyectos utilizan una versión a medida del proceso.

Nivel 4 – *Manejado*: la organización recolecta métricas del proceso *software* y de los productos desarrollados. Tanto el proceso como los productos se entienden y controlan cuantitativamente.

Nivel 5 – *Optimizante*: existe una mejora continua del proceso *software*, basada en la realimentación cuantitativa del proceso y en la puesta en práctica de ideas y tecnologías innovadoras.

De acuerdo al modelo, el desempeño en general de una organización mejora notablemente a medida que incrementa su nivel de madurez. La Figura 1 muestra de manera conceptual

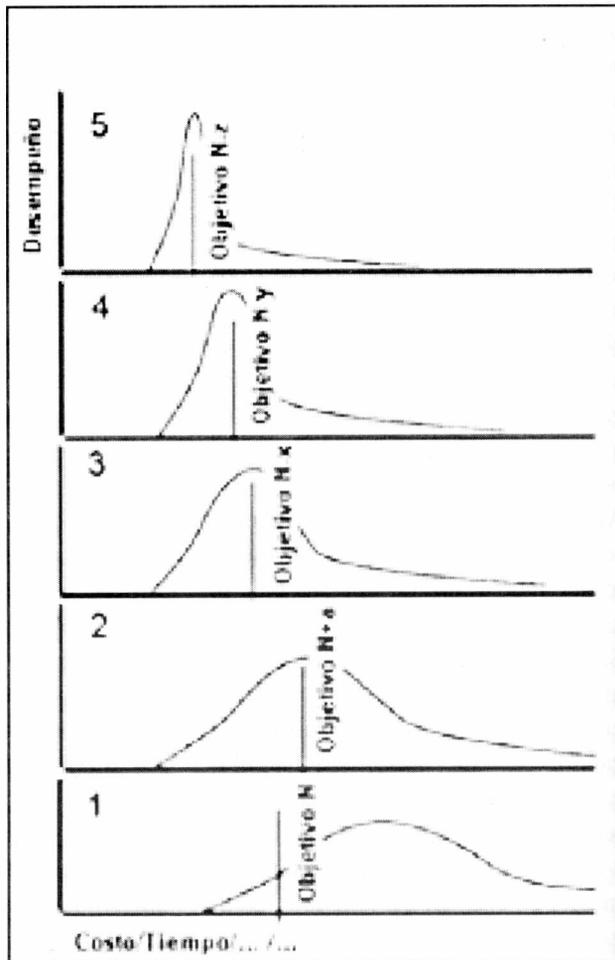


Figura 1: Variación del desempeño con respecto a los objetivos fijados para los factores costo y tiempo, de acuerdo al nivel de CMM de la organización.

las mejoras en el desempeño para cada nivel, contemplando los factores Tiempo y Costo (CMM, 1993). Conclusiones similares se pueden extraer para otros factores, como la Funcionalidad y la Calidad.

En la misma se puede observar lo siguiente:

- En las organizaciones que se encuentran en el nivel 1, los objetivos son generalmente ampliamente excedidos por la realidad.

- En las organizaciones que se encuentran en el nivel 2, se establecen objetivos más acordes a la realidad.

- En las organizaciones que se encuentran los niveles 3, 4, y 5, existe una menor dispersión de la realidad con respecto a los objetivos, y el desempeño mejora con cada nivel.

Algunas de las organizaciones que adoptaron el modelo, entre muchas otras, fueron (SEI, 2003): Accenture, AT&T, Boeing, Ericsson, Fuji Xerox, Hewlett Packard, Hyundai, IBM, Motorola, Nasa, NCR, NEC, PriceWaterhouseCoopers, Samsung, Siemens y United Airlines.

Luego del éxito alcanzado por CMM, el SEI desarrolló modelos similares para otras disciplinas entre las cuales figuraban la ingeniería de sistemas, la adquisición de software, las personas, y el desarrollo integrado de productos (CMMS, 2003).

A mediados de la década del 90, el SEI decide unificar todos los modelos, embarcándose en un esfuerzo que culmina en el año 2002, dando origen a una nueva generación llamada CMMI (Capability Maturity Model Integration) (CMMI, 2002).

El nuevo modelo CMMI brinda un marco con una estructura común para todas las disciplinas (ingeniería de software, ingeniería de sistemas, desarrollo integrado de productos,

adquisición de productos, personas) y agrega una nueva forma de representación, además de la conocida representación por niveles. La nueva forma de representación se llama Continua y está orientada a medir la mejora en los procesos de manera individual, en vez de hacerlo de manera conjunta como en la representación por niveles (CMMI, 2002). Dentro de esta nueva generación de modelos, el sucesor directo del CMM original es el denominado CMMI-SW (CMMI-SW, 2002). Este modelo presenta una mayor cobertura con respecto a las áreas de proceso, y agrega el concepto de representación continua.

En paralelo con el desarrollo de CMMI, el SEI elaboró un método para la evaluación formal del modelo denominado SCAMPI (Standard CMMI Appraisal Method for Process Improvement; SCAMPI, 2001). El método define una serie de reglas para la evaluación del modelo, las cuales deben utilizarse para valorar las distintas partes del mismo durante una evaluación formal. Estas reglas hacen que sea necesario utilizar herramientas, ya que el método deja de ser una simple encuesta para convertirse en una evaluación detallada y casi matemática.

La situación actual con respecto a los modelos es la siguiente:

El SEI planea discontinuar el modelo CMM (SW-CMM) original en diciembre del 2003, para lo cual sugiere a todas las organizaciones certificadas migrar a CMMI-SW.

Las grandes organizaciones certificadas en SW-CMM planean migrar a CMMI-SW.

Muchas organizaciones pequeñas planean certificarse en CMMI-SW, con el fin de poder acceder al mercado de las exportaciones.

La preparación previa a la certificación CMMI-SW es larga y costosa, por lo que las organizaciones utilizan el concepto de "evaluación interna" como paso preparatorio.

Una evaluación interna es algo difícil de llevar a cabo para las organizaciones recién iniciadas en el tema, y no existe un soporte adecuado de herramientas que le faciliten el camino.

En síntesis, el nuevo modelo trae aparejado un problema no trivial para las organizaciones, en lo referente a costos y tiempos necesarios para la preparación previa a su adopción o a una certificación. El problema se ve más acentuado en las organizaciones pequeñas, donde los recursos económicos, humanos y temporales suelen ser menores que en las grandes organizaciones.

En este sentido, sería deseable contar con una herramienta que asista a las organizaciones en la conducción de una evaluación interna, indicando paso a paso los aspectos del modelo CMMI y los cuatro criterios de evaluación del método SCAMPI, de manera de disminuir los costos y tiempos necesarios para la preparación previa a una certificación.

Existen actualmente tres herramientas de evaluación para CMMI:

- *CMM-Quest*: permite efectuar evaluaciones de acuerdo al modelo CMMI-SE/SW en su representación continua. La evaluación se limita a asignar valores a los objetivos; no permite evaluaciones a nivel de prácticas (por debajo del nivel de los objetivos). No brinda soporte para el método SCAMPI (CMM-Quest, 2001).

- *IME Toolkit*: permite efectuar evaluaciones de acuerdo al modelo CMMI-SE/SW. Las evaluaciones consisten en asignar valores numéricos a las prácticas, sobre cuya base la herramienta genera puntajes para las

áreas de proceso. No brinda soporte para el método SCAMPI. No posee guías de asistencia para la evaluación (IME Toolkit, 2003).

- Appraisal Wizard: soporta evaluaciones para gran parte de los modelos CMM y los métodos de evaluación propuestos por el SEI a lo largo de la historia (entre ellos, todos los CMMI

y SCAMPI). Está pensada para cubrir todas las necesidades del método SCAMPI, requiriendo amplios conocimientos del mismo por parte del usuario. Requiere que éste ingrese todos los valores que se asignan en las distintas instancias de evaluación (prácticas, objetivos, áreas de proceso) y no cuenta con la capacidad de

sugerir valores facilitando las tareas de ingreso de datos. Al brindar un soporte tan amplio y detallado, la herramienta no es para nada sencilla de utilizar (Appraisal Wizard, 2003). La Tabla 1 muestra en forma comparativa las características de estas herramientas.

Tabla 1. Características de herramientas

| Particularidad | CMM-Quest | IME Toolkit | Appraisal Wizard |
|--|--------------------------------------|--|---|
| Interfaz de usuario | Fácil, muy amigable | Medianamente amigable | Difícil, poco amigable |
| Tipo de usuario | Novato | Experto | Experto |
| Modelos soportados | CMMI SE/SW (representación continua) | Está basada en CMMI-SE/SW, no lo soporta formalmente | Gran parte de los CMM y todos los CMMI (ambas representaciones) |
| Método SCAMPI | No | No | Sí |
| Nivel de granularidad | Grueso (sólo objetivos) | Fino (hasta prácticas específicas) | Fino (hasta prácticas específicas) |
| Ayudas online | Sí | No | Sí |
| Navegación de la estructura del modelo | No | No | No |
| Generación de valores sugerido | No | Sí | No |
| Selección del nivel de granularidad para la evaluación | No | No | No |

De la tabla anterior se desprende que existe un área no soportada por las herramientas existentes, conformada principalmente por características orientadas a los usuarios novatos, como la navegación de la estructura del modelo, las guías paso a paso, la generación automática de valoraciones, y la selección del nivel de granularidad para la evaluación.

La preparación que debe llevar a cabo una organización como paso previo a la adopción del modelo CMMI-SW o a su certificación es un proceso largo,

costoso, y requiere de recursos humanos altamente capacitados. Normalmente, las organizaciones llevan a cabo una evaluación interna como parte del proceso de preparación.

En la actualidad no existen herramientas que brinden un soporte adecuado para la conducción de evaluaciones internas basadas en el método de evaluación SCAMPI (SCAMPI, 2001). Las herramientas existentes o bien son demasiado informales y no soportan el método (CMM-Quest, 2001; IME Toolkit, 2003), o bien es-

tán orientadas a usuarios profesionales y no resultan amigables a los usuarios novatos (Appraisal Wizard, 2003).

Aplicativo "CMM'S Tool"

El aplicativo permitirá a cualquier organización, empresa o usuario tener los siguientes beneficios directos:

- Autocalificarse en el modelo CMM
- Recorrer el modelo CMM (simplificado) con ayuda de búsqueda específica.

- Efectuar búsqueda de terminología y definiciones específica de procesos y calidad total.
- Interrelacionarse con páginas WEB determinadas por el usuario.
- Disponer de una ayuda permanente en el uso del aplicativo.

Los beneficios indirectos son:

- Establecer, con base en las calificaciones del modelo, planes de mejora a través del análisis de los resultados de la matriz Importancia-Desempeño por área clave.
- Monitorear el compromiso del equipo de Dirección.
- Identificar rápidamente fortalezas y debilidades a través de la configuración del front-end².

En la Figura 2 se muestra un diagrama del aplicativo.

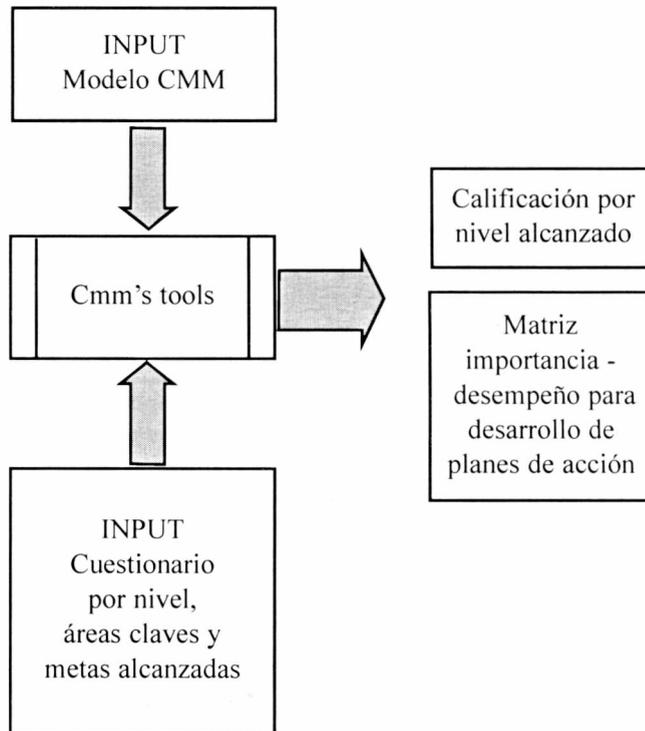
Leguaje de programación: Delphi 5.
Componentes adicionales: Companion Tools.

Base de Datos: Componentes Client DataSet.

Se diseñó un sistema modular cuyo detalle es el siguiente:

- a) Informatización de un resumen del modelo CMM que abarque sus distintos niveles, áreas claves, metas y actividades.
- b) Dependiendo de que nivel del CMM se quiera evaluar se presen-

Figura 2: Diagrama general del aplicativo desarrollado.



tarán las correspondientes prácticas clave y actividades relacionadas.

c) El usuario introducirá la respuesta mediante front-end y el aplicativo otorgará la calificación correspondiente. Dicha calificación será a través de una búsqueda de concordancia entre una matriz conformada por las respuestas del usuario y las tablas referenciales que asignarán puntuaciones.

Por cada nivel KPA se tendrá como datos de salida las fortalezas y debilidades posibilitando la diagramación de los planes de mejora correspondiente. La Tabla 2 brinda una descripción de alcance, funcionalidad y resultados obtenidos.

Los niveles, las áreas claves y las metas abarcadas en la norma simplificada son:

Inicial: Compromiso del Equipo de Dirección con la Calidad.

Repetible: Control Administrativo Básico.

Optimizado: Control de los Procesos.

Definido: Definición de Procesos.

Administrado: Medición de los Procesos.

La calificación del desempeño se indica en la Tabla 3.

Además de la calificación de desempeño es necesario agregar una calificación de importancia, que se establece según:

- 1 – Nada Importante
- 2 – Poco Importante
- 3 – Medianamente Importante
- 4 – Muy Importante
- 5 – Clave para el Desarrollo del Negocio.

¹ Granularidad se refiere a la especificidad con la que se define un nivel de detalle. En jerarquías se empieza por la parte más alta, siendo la granularidad mínima el nivel más bajo.

² El front-end es la parte del software que interactúa con el o los usuarios.

³ En la terminología de sistemas se suele utilizar la palabra "matcheo".

Tabla 3. Calificación de desempeño

| Valor | Enfoque | Utilización |
|--|---|---|
| <p>Pobre (0)</p> | <ul style="list-style-type: none"> • No se reconoce su necesidad en el ámbito de la administración • No existe habilidad por parte de la organización o el equipo • No existe compromiso por parte de la organización o equipo • La práctica no es evidente | <ul style="list-style-type: none"> • Ningún sector de la organización o el equipo utiliza la práctica • Ningún sector de la organización o el equipo demuestra interés |
| <p>Débil (2)</p> | <ul style="list-style-type: none"> • La administración comienza a reconocer la necesidad • Se empiezan a crear procedimientos y procesos para apoyar las metas. | <ul style="list-style-type: none"> • Una pequeña parte de la organización o el equipo implementa la práctica • Monitoreo limitado o verificación en uso |
| <p>Justa (4) (Para metas que requieren un procedimiento documentado: Si no hay un procedimiento documentado, no se da una calificación mayor a 4)</p> | <ul style="list-style-type: none"> • Compromiso incompleto por parte de la administración • Existe un camino definido para implementar la meta | <ul style="list-style-type: none"> • Cierta consistencia en el uso • Se usa en la mayoría de la organización o el equipo • Hay monitoreo o verificación de uso para una parte de la organización o el equipo (por al menos un mes) |
| <p>Marginalmente Calificado (6) (Para metas que requieren un procedimiento documentado: El procedimiento esta documentado y se ha utilizado al menos una vez)</p> | <ul style="list-style-type: none"> • Gran compromiso por parte de la administración • La implementación de la meta está en buen camino • Existen ítems de apoyo | <ul style="list-style-type: none"> • Se utiliza en la mayoría de la organización o el equipo • Monitoreo o verificación del uso de la mayoría de las partes de la organización o el equipo (por al menos 3 meses) |
| <p>Calificado (8) (La organización cumple muy fuertemente lo plasmado en las guías. Dicho cumplimiento es asimilado fuertemente por cada una de las personas que componen la organización)</p> | <ul style="list-style-type: none"> • Compromiso total por parte de la administración • La administración es pro-activa • La práctica representa una parte integral del proceso • Ítems de apoyo alientan y facilitan el logro de la meta | <ul style="list-style-type: none"> • Se ocupa a lo largo de la organización o el equipo • Es consistente a lo largo de la organización • Se monitorea o verifica a lo largo de la organización (al menos por 6 meses) |
| <p>Sobresaliente (9)</p> | <ul style="list-style-type: none"> • Se comparte proactivamente el proceso afuera de la organización • Se reconoce la excelencia por externos a la organización | <ul style="list-style-type: none"> • Proceso de mejora continua mantenido (por al menos un año) |
| <p>Clase Mundial (10)</p> | <ul style="list-style-type: none"> • La administración provee con fervor liderazgo y compromiso. La excelencia de la organización es reconocida por externos | <ul style="list-style-type: none"> • Uso consistente y con penetración a lo largo de toda la organización o equipo • Se monitorea o verifica a lo largo de toda la organización |

Por cada área clave de cada nivel evaluado, se asigna un nivel de desempeño (0 a 10) y una importancia (1 a 5). La organización puede calificar (a través del desempeño) todas las áreas claves, asignándoles una importancia relativa. Pueden existir áreas más importantes con baja calificación (necesidad de un plan de acción inmediato) y otras menos importantes con alta calificación (tal vez no merezcan un plan de acción en el corto plazo).

Conclusiones

Un *software* de buena calidad debe principalmente satisfacer las expectativas de clientes y usuarios. Aun cuando la respuesta parece obvia, es en este punto donde fallan la mayoría de las empresas de nuestro país dedicadas a la elaboración de programas de software.

A pesar de que en los últimos años la industria informática nacional ha recibido muchos elogios, en especial al compararla con la del resto de Latinoamérica, pareciera que todavía existe un bajo desempeño en el ámbito de los desarrolladores. En particular ésta es un área muy inmadura en sus procesos. Existen dos variables que necesariamente deben considerarse al embarcarse en el desarrollo de un *software*: calidad y productividad. La primera depende de múltiples factores, como el personal a cargo del proceso, el tipo de herramientas usadas y la meto-

dología aplicada. El segundo aspecto se refiere a la capacidad de la empresa de reducir costos previniendo las fallas antes de que se produzcan.

Este modelo es esencial en la detección del nivel de madurez alcanzado, y representa un camino evolutivo para pasar de un proceso inmaduro a uno disciplinado y controlado dentro de la compañía.

Esta conciencia de mejorar debería trasladarse también a los usuarios, quienes podrían ser en el futuro un factor importante que impulse la demanda de las organizaciones dedicadas al desarrollo, por aprender las ventajas de éste u otros modelos que les permitan alcanzar un nivel de madurez adecuado.

De esta forma, quizás se podría alcanzar la difícil meta de tener un estándar de desarrollo que satisfaga tanto a los analistas de software como a los consumidores.

CMM no es una teoría, sino un nuevo enfoque y cuando se entiende se reconocen los beneficios, lo que hace que la organización mejore constantemente.

Acorde con toda la información recopilada, se puede concluir que en nuestro país existe preocupación por mejorar los procesos de desarrollo de software, pero todavía se está en el camino inicial para lograrlo.

Referencias

APPRAISAL WIZARD (2003) Formal or informal appraisal tool, Integrated

System Diagnostics Incorporated. Demo disponible en el sitio de la empresa, <http://www.isd-inc.com>; página vigente en septiembre de 2008.

IME Toolkit (2003) Interim Maturity Evaluation Toolkit, Management Information Systems.

CMM (1993) Capability Maturity Model SM for Software, Version 1.1, Technical Report CMU/SEI-93-TR-024, ESC-TR-93-177, February 1993. Mark C. Paulk, Bill Curtis, Mary Beth Chrissis. Charles V. Weber.

CMM-Quest (2001) Self assessment tool, HM&S IT-Consulting GmbH. Demo disponible en el sitio de la empresa, <http://www.cmm-quest.com/>.

CMMI (2002) Capability Maturity Model Integration. Disponible en <http://www.sei.cmu.edu/cmmi/cmmi.html>, página vigente en septiembre de 2008.

SCAMPI (2001) Standard CMMI Appraisal Method for Process Improvement. <http://www.sei.cmu.edu/publications/documents/01.reports/01hb001.html>, página vigente en septiembre de 2008.

SEIR (2003) Compiled list of Organizations who have Publicly Announced their Maturity Levels after having an Appraisal Performed. Software Engineering Information Repository.

THE STANDISH GROUP (2003) Disponible en <http://www.standish-group.com>, página vigente en septiembre de 2008.