

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

PROGRAMA DE GRADUADOS EN ELECTRONICA,
COMPUTACION, INFORMACION Y COMUNICACIONES



"MEJORA DEL PROCESO DE DESARROLLO
DE SOFTWARE EN MEXICO, UN ESTUDIO
EXPLORATORIO"

TESIS

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRO EN ADMINISTRACION DE
TECNOLOGÍAS DE INFORMACION

POR

JOSE ANTONIO ALVARADO MARQUEZ

MONTERREY, NUEVO LEON

JULIO DE 2004

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**

CAMPUS MONTERREY

**PROGRAMA DE GRADUADOS EN ELECTRÓNICA,
COMPUTACIÓN, INFORMACIÓN Y COMUNICACIONES**



**“MEJORA DEL PROCESO DE DESARROLLO DE SOFTWARE EN MÉXICO, UN
ESTUDIO EXPLORATORIO”**

TESIS

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO
ACADEMICO DE MAESTRO EN ADMINISTRACIÓN DE TECNOLOGÍAS DE
INFORMACIÓN**

POR:

JOSÉ ANTONIO ALVARADO MÁRQUEZ

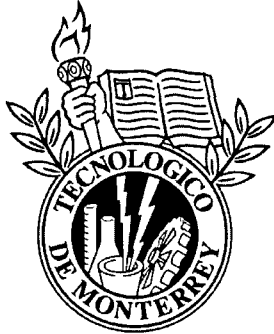
MONTERREY, N.L.

JULIO 2004

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

PROGRAMA DE GRADUADOS EN ELECTRÓNICA,
COMPUTACIÓN, INFORMACIÓN Y COMUNICACIONES



**“MEJORA DEL PROCESO DE DESARROLLO DE SOFTWARE EN MÉXICO, UN
ESTUDIO EXPLORATORIO”**

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO
ACADEMICO DE MAESTRO EN ADMINISTRACIÓN DE TECNOLOGÍAS DE
INFORMACIÓN

POR:

JOSÉ ANTONIO ALVARADO MÁRQUEZ

MONTERREY, N.L.

JULIO 2004

Dedico esta tesis a Dios, a mi país, a mi madre, a mi padre, a mi hermana, a toda mi familia, así como a mis amigos y compañeros.

A todos los anteriores, gracias por creer en mí, gracias por su soporte en los momentos difíciles.

A los miembros del comité de tesis, mi asesor, sinodales y expertos, gracias por brindarme su experiencia, consejos e ideas.

A los que participaron en la encuesta, mil gracias por abrirme sus puertas, sin su apoyo esta tesis no hubiera sido posible, espero que encuentren útil los resultados de esta investigación.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	III
LISTA DE FIGURAS	V
LISTA DE TABLAS	VI
DESCRIPCIÓN GENERAL.....	1
OBJETIVO DE LA TESIS.....	5
PREMISAS.....	5
RESTRICCIONES.....	5
METODOLOGÍA UTILIZADA.....	6
PRODUCTO FINAL Y CONTRIBUCIÓN ESPERADA.....	6
ESTRUCTURA DE LA TESIS.....	6
CAPITULO 2 – REVISIÓN BIBLIOGRÁFICA	8
INTRODUCCIÓN.....	8
COMPLEJIDAD DEL SOFTWARE Y CALIDAD.....	8
PROCESO DE SOFTWARE Y MODELOS DE MADUREZ.....	11
CERTIFICACIÓN INTERNACIONAL EN INGENIERÍA DE SOFTWARE.....	20
<i>Certificación Internacional</i>	20
<i>Certificación internacional en Ingeniería de Software</i>	21
<i>SEI - Capability Maturity Model for Software</i>	21
<i>ISO/IEC 15504 - Software Process Improvement and Capability dEtermination</i>	33
<i>SEI - Capability Maturity Model for Software integrated</i>	35
FACTORES DE ÉXITO PARA EL MEJORAMIENTO DE LA CALIDAD EN LA INDUSTRIA DE SOFTWARE.....	39
APLICACIÓN DEL CMM EN EMPRESAS PEQUEÑAS.....	45
INTERPRETANDO AL CMM.....	46
ACEPTACIÓN DEL CMM EN EL MUNDO.....	49
<i>Tipos de organizaciones valoradas en el mundo en CMM</i>	49
<i>Tamaño de las organizaciones valoradas en el mundo en CMM y CMMi</i>	49
<i>Países donde las organizaciones han sido valoradas en CMM y CMMi</i>	50
<i>Cantidad de valoraciones CMM reportadas al SEI por año</i>	51
<i>Cambios en el nivel de Madurez</i>	52
CMM EN MÉXICO.....	52
CONCLUSIONES.....	54
CAPITULO 3 - METODOLOGÍA DE INVESTIGACIÓN	55
INTRODUCCIÓN.....	55
DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN.....	55
OBJETIVO DE LA INVESTIGACIÓN.....	57
METODOLOGÍA DE LA INVESTIGACIÓN DE CAMPO.....	59
RECOLECCIÓN DE DATOS.....	59
DISEÑO DE LA ENCUESTA.....	59
<i>Sección de Generalidades de la Encuesta</i>	59
<i>Sección intentos de mejora, de factores de éxito y de fracaso en estos intentos</i>	60
PRUEBAS POR PANEL DE EXPERTOS.....	63
VALIDACIÓN DEL CONTENIDO DE LA ENCUESTA.....	64
ACERCA DE LA ENCUESTA FINAL.....	64
LÍMITES DE LA INVESTIGACIÓN.....	65
SELECCIÓN DE LA POBLACIÓN Y MUESTRA.....	65
APLICACIÓN DE LA ENCUESTA.....	66
CONCLUSIONES.....	66
CAPITULO 4 – ANÁLISIS DE RESULTADOS	67
INTRODUCCIÓN.....	67

DEMOGRAFÍA	67
<i>Por PA</i>	76
<i>PA-Giro</i>	76
<i>PA-Tamaño</i>	77
<i>Por Giro</i>	78
<i>Por Tamaño (número total de empleados)</i>	79
<i>Por Tamaño (número de empleados que desarrollan Software)</i>	80
RAZONES PARA QUE LOS ESFUERZOS SE HAYAN REALIZADO DE FORMA AISLADA	85
FACTORES DE FRACASO	88
FACTORES DE ÉXITO	93
COMENTARIOS ADICIONALES	98
CAPITULO 5 – CONCLUSIONES Y FUTURAS INVESTIGACIONES.....	100
CONCLUSIONES	100
HALLAZGOS	100
RECOMENDACIONES.....	106
<i>Listado de Recomendaciones</i>	107
FUTURAS INVESTIGACIONES.....	112
ANEXOS.....	113
ANEXO A. CUESTIONARIO	113
ANEXO B. DATOS DE CONTACTO	122
REFERENCIAS.....	123

LISTA DE FIGURAS

Fig. 2 - 1 Diagrama de categorías de procesos del MoProSoft	19
Fig. 2 - 2 Diagrama de relación entre procesos del MoProSoft	19
Fig. 2 - 3 Componentes del CMM.....	23
Fig. 2 - 4 Niveles de madurez con sus respectivas áreas de proceso claves.....	25
Fig. 2 - 5 Partes que conforman al ISO/IEC 15504.....	33
Fig. 2 - 6 Proceso de valoración del ISO/IEC 15504	34
Fig. 2 - 7 Componentes del CMMi.....	38
Fig. 2 - 8 Tipos de Organización a nivel mundial con valoraciones en CMM.....	49
Fig. 2 - 9 Tamaño de las organizaciones a nivel mundial con valoraciones en CMM.....	49
Fig. 2 - 10 Tamaño de las organizaciones a nivel mundial con valoraciones en CMMi.....	50
Fig. 2 - 11 Países donde se han Realizado valoraciones de CMM.....	50
Fig. 2 - 12 Países donde se han Realizado valoraciones de CMMi.....	51
Fig. 2 - 13 Número de valoraciones reportadas por año al SEI.....	51
Fig. 2 - 14 Cambios de nivel de Madurez en las revaloraciones reportada al SEI a nivel mundial	52
Fig. 2 - 15 Factores de estudio de la investigación.....	58

LISTA DE TABLAS

Tabla 2 - 1 Metas y prácticas claves más importantes para el nivel de madurez 2 31

Tabla 2 - 2 Las áreas de proceso claves asignadas a categorías de procesos 32

Tabla 2 - 3 Factores que afectan al cambio organizacional en los esfuerzos de mejora del proceso de software 40

Tabla 2 - 4 Factores que han llevado a tener y mantener una posición competitiva en el mercado internacional en el desarrollo de Software..... 44

CAPITULO 1 – Introducción

La intención principal de esta investigación es dar a conocer los intentos de mejora o falta de estos en el proceso de desarrollo de software en nuestro país, tomando como base el nivel 2 de certificación del modelo de certificación internacional escalonado CMMi en ingeniería de software, el cuál es el nivel inicial de madurez para empresas que llevan a cabo este proceso. Cabe señalar que la exploración de dichos intentos no se limita a las empresas que se dedican exclusivamente al desarrollo de software, sino a cualquier empresa perteneciente a cualquier industria, siempre y cuando lleve a cabo estas prácticas aunque sea para ellas mismas. Teniendo los factores antes mencionados se pretende elaborar un listado de recomendaciones. Se espera que esta contribución sea de utilidad para todas estas empresas en la mejora de este proceso.

Descripción general

Según Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber (1993) en muchas organizaciones los proyectos de software son excesivamente tardíos y doblan el presupuesto que tenían programado, ya que la organización frecuentemente no provee la infraestructura y soporte necesario para ayudar a que los proyectos anulen este tipo de problemas. Las organizaciones de software inmaduras son reaccionarias y los administradores están enfocados comúnmente en resolver las crisis inmediatas, mejor conocidos como “apaga-fuegos”. Las calendarizaciones y presupuestos son comúnmente excedidos porque no están basados en estimados reales. Cuando se imponen fechas de entrega difíciles, la calidad y la funcionalidad de los sistemas son comprometidas para cumplir con el calendario.

Por otro lado, en una organización madura, los administradores monitorean la calidad de los productos de software y la satisfacción de los clientes. Estas empresas tienen como objetivo la calidad y manejan bases cuantitativas para evaluarla, analizan los problemas con los productos y los procesos; organizan los calendarios y los presupuestos basándose en el rendimiento histórico. Los resultados esperados sobre el costo, calendario, funcionalidad y calidad del producto son comúnmente alcanzados.

Para lograr resultados duraderos de los esfuerzos de mejora es necesario diseñar una ruta evolutiva que incremente los niveles de maduración de software en la compañía. Sin embargo, la computación y sus aplicaciones, sobre todo en desarrollo de software también dependen de la creatividad, afirma Anónimo (1998), y su calidad como producto es resultado de la calidad del proceso que lo origina. "La idea es mezclar el espíritu creativo con la calidad para exportar software", menciona Jesús Fabela (1998).

Dicho mejoramiento continuo puede ocurrir solo a través de un esfuerzo sustentable hacia la construcción de una infraestructura de procesos de Ingeniería de Software efectiva y prácticas administrativas, según Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber (1993). Así mismo, Oktava (1998) nos dice que establecer parámetros guías para los grandes compradores, la definición de la metodología de inducción de ingeniería del

software, la introducción a la metodología en las empresas y la certificación al termino del proceso son puntos importantes del proyecto para generar una cultura de ingeniería de software. Aunque, según Anónimo (1998), también es importante recordar que "La certificación no es el fin último del proyecto. Lo importante es la preocupación por el proceso de desarrollo". "Es importante la estandarización y disciplina en el proceso, porque ya no se puede trabajar solo, el desarrollo de software actualmente es inherente al trabajo en grupo".

Sabiendo lo anterior, ¿Qué podría ganar México mejorando su Industria del Software? ¿Cómo podría superar sus debilidades actuales?

La oportunidad actual, según Bancomext (2001), es que EUA tiene un Déficit de 600,000 expertos en informática EUA, creciendo al 10% anual; la UE, estima números similares. Generándose con ello una demanda infinita para México. Hace 15 años la India facturaba US \$ 20 M, actualmente facturan US \$ 6,500 M, y para el 2008 proyectan exportar US \$ 50,000 M.

La Secretaría de Economía (2001) pone también como oportunidad que México cuenta con una posición favorable para convertirse en un competidor de talla mundial en este ramo, gracias a su ubicación geográfica, perfil demográfico y estado de desarrollo tecnológico. No obstante que el potencial de desarrollo es evidente, la industria de software es apenas incipiente en nuestro país: participa con tan sólo el 0.1% del PIB (cifras de 2000).

México tiene como ventaja sobre otros países una ubicación privilegiada, Costos competitivos, y una cultura empresarial similar a la de EUA. Algunas otras ventajas importantes que tiene el desarrollo de la industria de Software en nuestro país según Bancomext (2001) son: atrae inversionistas en TI y existe un fuerte efecto multiplicador que mejora la economía de su ubicación; es el puente tecnológico entre EUA y Latinoamérica, pudiendo llegar a cambiar esta situación su perfil exportador; promueve el empleo masivo de calidad y evita la fuga de cerebros.

Para mejorar la industria, según la UNCTAD (2001) de la ONU, el desarrollo de esta industria en los países emergentes se puede llevar a cabo a través de dos alternativas de políticas estratégicas. La primera estrategia es la que atención de demanda interna, se basa en el uso de software libre como una herramienta para el fortalecimiento de la infraestructura económica a fin de soportar el desarrollo de otros sectores de la economía y, la segunda estrategia generación de un proveedor internacional, se orienta a considerar a la industria del software como sector prioritario por la generación de nuevos empleos y el crecimiento industrial basado en la creación de oferta de software para exportación. Una opción alternativa para el desarrollo de la industria del software consiste en integrar ambas estrategias bajo un orden secuencial; el uso de software libre como herramienta para el fortalecimiento de la infraestructura económica sería reemplazada gradualmente por la estrategia de desarrollo de software para exportación.

Bancomext (2001) hace las siguientes recomendaciones para fomentar el desarrollo de la industria del software: se requiere de prácticas y de programas de ayuda para facilitar su crecimiento sin lastres tributarios, en especial en dos rubros: los impuestos por ingreso y

facilidades para la compra e importación de tecnología; además se requiere crear una cámara nacional que agrupe a las empresas del ramo o que alguna de las existentes tome ese papel y, darle a esta industria la representatividad que requiere en la economía nacional, además de ser un foro para compartir y promover las prácticas de calidad mundial que se deben implantar en la industria y promover el intercambio de la calidad que les de la capacidad de competir en los mercados internacionales. Hay que tener en cuenta que un paso importante para incursionar en el mercado internacional es el de contar con la certificación de calidad en CMM (Capability Maturity Model), expedido por el SEI (Software Engineering Institute). Otras de las necesidades, según Bancomext (2001), es la de establecer una industria nacional competitiva en el desarrollo de software para competir en el mercado internacional y así introducir las mejores soluciones al Sistema e-México.

Otros puntos importantes a tomar en cuenta, según la Secretaría de Economía (2001) son: Los recursos humanos representan el factor crucial en la industria de software, particularmente tanto en cantidad como en calidad, sin embargo, la vinculación de las empresas de TI con las universidades es mínima; la mayor parte de las empresas no tienen los niveles de desempeño requeridos para incorporarse al mercado internacional; en general, se carece de modelos, normas y de organismos evaluadores de la capacidad de procesos de la producción de software; las evaluaciones internacionales de capacidad de procesos son costosas; los gobiernos estatales deberían acordar con las universidades públicas y privadas, las instituciones de educación técnica de nivel medio y las empresas, el diseño y aplicación de planes y programas de estudio que respondan a las necesidades de la industria, incluyendo maestrías y doctorados, así como intercambios académicos con universidades líderes nacionales e internacionales.

AMITI (2001) también nos dice que la industria norteamericana de software, tomada como tendencia del mercado mundial de software, hace patente que esta industria representa uno de los segmentos más vibrantes de la economía cuya contribución supera a cualquier otra rama de la industria manufacturera, de ese país. En el caso de México, dicha contribución es totalmente insignificante, lo cuál indica que la industria Mexicana de software no ha respondido y esta dejando de aprovechar esta oportunidad económica que actualmente representa el mercado mundial de software. Sin embargo, otros países han detectado oportunamente este 'nicho' de mercado y han respondido con la eficacia y agresividad que demanda el reto de la globalización.

En México, la Industria de Software debería ser considerada como estratégica y prioritaria para todo tipo de apoyos gubernamentales. Un factor muy importante a considerar es que la importancia social y económica de esta industria, radica en su potencial como fuente de empleos, como generadora de divisas y en su impacto en la sustitución de importaciones, además de ser una industria ecológicamente 'limpia'. Esta industria requiere consolidarse en el mercado interior y crecer con calidad en el mercado internacional. Si se dan las condiciones y el apoyo gubernamental, esta industria podría alcanzar una capacidad instalada de 5,000 millones de dólares anuales entre exportaciones y ventas internas para el 2006 (equivalente al 1.5% de la producción anual de software mundial) además de crear trabajos directos y bien remunerados a 100,000 profesionistas y técnicos. De acuerdo con lo anterior, existe un mercado mundial de SW: US \$300B/año reporta (Bancomext, 2001).

AMITI (2001) sugiere que una empresa de software que desee exportar, debe primeramente alcanzar un tamaño mínimo de 250 empleados y crecer arriba de 1,000 para ser realmente competitiva a nivel internacional.

Como caso ejemplar, AMITI (2001) cita a Narayana Murthy (Presidente y Director General de Infosys Technology Ltd. India) y Pankaj Jalote (Jefe del Instituto de Ciencias de la Computación del Instituto Tecnológico de la India), ellos mencionan que el éxito de la industria del software en la India ha sido el tránsito por la calidad, comenzando por productos, siguiendo los proyectos, la norma ISO y actualmente el uso del modelo CMM. En la industria Irlandesa el estándar de calidad a sido ISO, que en palabras de John Corrigan, “les ha dado credibilidad”. También se requiere analizar a fondo los Tratados de Libre Comercio que México tiene firmados principalmente con Norte América y Europa. Esta revisión deberá de hacerse desde la perspectiva de la Industria del Software, para detectar oportunidades y beneficios para México.

Dada la oportunidad y como respuesta a la necesidad de mejorar, surge el Plan Nacional de Desarrollo 2001–2006 (PND) de la Secretaría de Economía (2001), el cual plantea el objetivo de elevar y extender la competitividad del país, mediante la estrategia de promover el uso y aprovechamiento de la tecnología y de la información. La producción de software es una actividad económica que se caracteriza por generar un alto valor agregado y aportar a la economía productos y servicios esenciales para su modernización. Esta industria se basa en el conocimiento, desarrolla habilidades más allá de la manufactura, propicia la innovación tecnológica y genera empleos bien remunerados, no contamina y requiere de relativamente poco capital para iniciar. Bancomext (2001) resalta que la visión del país para el 2010 es tener 10 empresas desarrolladoras de más de 5,000 empleados, varias más mayores de 500, cientos de compañías desarrolladoras especializadas, 75,000 profesionistas empleados, exportaciones de US \$ 3,500 millones, México reconocido como un país de Software.

Bancomext, apoyando dicho plan, desarrolló un esquema financiero que permite a empresas productoras de software (Fábricas de Software) contar con el capital de trabajo para el desarrollo de proyectos de exportación, los beneficios financieros serán para las empresas mexicanas desarrolladoras de software (Fábricas de Software), exportadoras directas o indirectas que cuenten o que se encuentren en el proceso para obtener la certificación CMM al menos nivel 2 o ISO 9001 que incluya las prácticas para obtener la certificación CMM, o que cuenten con contratos de exportación recurrentes de un mismo cliente. (Rafael Salinas Sánchez, 2001). El objetivo es apoyar las necesidades de capital de trabajo de las empresas desarrolladoras de software, tales como capacitación, sueldos, viajes, renta de instalaciones, mobiliario, equipo de cómputo, etc., pago a supervisores y obtención de certificaciones, con base en contratos de servicios específicos. Las líneas de crédito en cuenta corriente con vigencia de hasta 3 años y revisión anual, hasta por 5 millones de dólares en función del análisis de crédito realizado y hasta por el 50% del valor del contrato de exportación (Rafael Salinas Sánchez, 2001).

Algunas de las realidades actuales de nuestro país es que en los años recientes, el gasto en tecnologías de información ha trasladado su énfasis del hardware al software, provocando que la relación entre el segundo y el primero suba de 32.5% en 1996 a 40% en 1999. El

mercado mundial de productos de software rebasa los 153,000 millones de dólares anuales. Estados Unidos es el principal consumidor, con un gasto superior a los 75,000 millones de dólares anuales y una participación de 48.8% en el total mundial. (Secretaría de Economía, 2001), Robert Mullins y Sharon Simonson (2002) señalan que se requiere mucho trabajo para cambiar la percepción de los empresarios estadounidenses sobre la industria mexicana de software y poder competir con países como Singapur, China, India o Pakistán. México actualmente se percibe en el plano internacional como un país sin capacidad de desarrollar tecnología, en particular software, debido a que sólo pocas empresas cuentan con las evaluaciones de capacidad de procesos que demanda el mercado internacional, sólo un reducido grupo de empresas tiene la experiencia de competir en el exterior, la escasez de programadores e ingenieros certificados en las últimas tecnologías y la carencia de infraestructura adecuada, todo esto dificulta la realización de alianzas estratégicas y de la inversión extranjera, además hace falta promoción adecuada para atraer empresas.

Objetivo de la tesis

El objetivo de la investigación es realizar una exploración a nivel nacional para conocer los intentos, actividades y resultados o falta de ellos para lograr una mejora de calidad en Ingeniería de Software, así como en las empresas que no lo hayan intentado, saber por qué motivos no lo han hecho.

- Obtener los factores más importantes que llevaron al éxito o que llevaron al fracaso de los intentos de mejora del proceso de software en nuestro país.
- Proponer un listado de recomendaciones a las empresas para incrementar las posibilidades de éxito de los proyectos de mejora al proceso de software utilizando como base los factores obtenidos en el punto anterior.

Premisas

Al comenzar la investigación se tienen las siguientes premisas:

- Los problemas de desarrollo de Software a los que se enfrentan las empresas mexicanas son los más fundamentales, esto es, no pueden avanzar sin haber resuelto estos problemas primero.
- Los esfuerzos organizacionales para mejorar el proceso de mejora son muy pocos.
- No hay un entendimiento de por qué o para qué llevar a cabo un proceso de mejora.
- No hay alianzas entre las empresas de la industria del Software en México, se cuentan con muchas empresas pero con muy pocos empleados.

Restricciones

- La investigación de campo se realizará en las empresas ubicadas en el territorio nacional, determinadas en la muestra

- La investigación será recabada exclusivamente mediante encuestas
- La cantidad y calidad de la información dependerá en gran medida de la disponibilidad que se tenga por parte de las empresas para contestar las encuestas. Aunque la calidad se considerará aceptable si la mayoría de las personas entrevistadas cuentan con un puesto gerencial o que involucre el desempeño de un profesionalista.

Metodología utilizada

Basándose en los objetivos de esta investigación y tomando en cuenta, como respaldo, el libro de Metodología de la Investigación de Hernández, Fernández y Baptista (2002) que describen cuatro tipos de investigación, este estudio requiere una investigación de tipo exploratoria, ya que este tipo de estudio se realiza, normalmente, cuando el tema o el objetivo a examinar ha sido poco estudiado anteriormente.

Producto final y contribución esperada

Se pretende que al término de esta investigación se haya recabado la información necesaria de los esfuerzos de mejora en el proceso de Software en las empresas mexicanas, para conocer mejor la situación actual del proceso de desarrollo de Software en nuestro país. En base a lo observado proponer una lista de recomendaciones que aumente las probabilidades de éxito en el proceso de mejora de dicho proceso, pretendiendo así mejorar la competitividad de dichas empresas.

Estructura de la tesis

A lo largo de la presente investigación se presenta la información más relevante sobre el tema, la cual está estructurada de la siguiente manera:

En el capítulo 1, correspondiente a la introducción, se hace una descripción de la situación actual y de la importancia de esta investigación en nuestro país, la estructura de la tesis y la contribución esperada.

En el capítulo 2, la revisión bibliográfica, se proporcionan conceptos fundamentales relacionados con esta investigación y la importancia de mejorar el proceso, también se muestra una descripción de algunas de las investigaciones más importantes sobre el tema efectuadas hasta ahora.

En el capítulo 3, denominado metodología de investigación, se describe el problema, los factores de estudio, la investigación de campo, y las herramientas estadísticas que se usaron.

En el capítulo 4, resultados de la investigación, utilizando técnicas estadísticas, se analiza la información obtenida por las encuestas, mostrando los resultados de la investigación de campo, también se presenta un análisis de los resultados que generó el producto final de la tesis.

Para el capítulo 5, se dan a conocer las conclusiones de esta investigación, el listado de recomendaciones y las futuras investigaciones que surgen de esta tesis.

CAPITULO 2 – Revisión bibliográfica

Introducción

El proceso de desarrollo de software es un proceso complejo, involucra muchos elementos y hay muchas formas de llevarlo a cabo, es un proceso que en las empresas mexicanas se hace de forma muy diferente de una a otra, la calidad con que se lleva a cabo este proceso en la mayoría de los casos no es evidente para los clientes, por eso surgen mecanismos de certificación avalados por organizaciones mundialmente conocidas, que buscan avalar esta calidad. Estos mecanismos pueden ser tomados como referencia para conocer la madurez de la empresa en el desarrollo de dicho proceso. Dentro de los modelos de valoración mas conocidos encontramos al CMMi y al ISO/IEC 15504 (también conocido como SPICE), dentro de este capítulo se da una breve descripción de estos y otros modelos de mejora.

Complejidad del software y calidad

La construcción de software puede involucrar elementos de gran complejidad, que en muchos casos no sean tan evidentes como los que se pueden ver en otras ingenierías. Un puente, un edificio, una mina, una red de ferrocarriles son ejemplos de sistemas complejos de otras ingenierías, pero el ingeniero de software construye sistemas cuya complejidad puede parecer que permanece oculta. El usuario supone que en informática todo es muy fácil (Juan Manuel Cuella, 2001).

Para Lionel Briand y Sandro Morasca (1996), la complejidad de un sistema debe ser igual o mayor que la suma de las complejidades de cualquiera de sus módulos aun cuando no hay relaciones entre estos módulos pero estos comparten componentes. La complejidad de un sistema nunca disminuirá cuando las relaciones entre sus componentes aumenten.

Watts Humphrey (2002) nos subraya que el software se ha vuelto un factor crítico en la sociedad moderna. Todos parecen necesitar más y mejor software, más rápido y más barato. Muchos proyectos en desarrollo son ahora tan largos y complejos que unos pocos especialistas brillantes ya no pueden hacerse cargo de ellos. Desafortunadamente no hay señales de alguna nueva tecnología mágica para resolver estos problemas.

Desarrollar Software es un proceso complejo. En este proceso se involucran muchos elementos, se vinculan diversas variables y se ve afectado por muchas situaciones. Es complejo porque lo que se intenta es traducir procesos realizados por personas para ahora realizarlos por un medio computacional. Desarrollar Software es una disciplina, que incluye métodos e involucra conceptos que han ido evolucionando a lo largo de los años. (Asley Cristales, 1996)

Alfredo Weitzenfeld (2003) nos dice que cuanto más grandes son los sistemas de software mayor es su complejidad. Para él se puede hablar de dos aspectos que causan esta complejidad, uno estático y otro dinámico:

- El aspecto *estático* del software tiene que ver con la funcionalidad que el software ofrece. Cuanto mayor es su funcionalidad mayor es el número de requisitos que debe satisfacer un sistema. Esto significa que los sistemas se vuelven más grandes y más difíciles de comprender por la cantidad de información y funciones que manejan. El nivel de complejidad radica en estos aspectos intrínsecos a la aplicación. Para reducir tal complejidad habría que simplificar la funcionalidad que el sistema ofrece.
- El aspecto *dinámico* del software tiene que ver con los cambios que pudieran hacerse en un sistema en el futuro. Según una “ley” de desarrollo de software (Lehman, 1985), “todo programa que se use se modificará”; y cuando un programa se modifica, su complejidad aumenta, siempre y cuando uno no trabaje activamente contra esto. Esto es similar al problema de la entropía, una medida de termodinámica sobre desorden. Según la segunda ley de termodinámica, la entropía de un sistema cerrado no puede ser reducida, solo puede aumentar o posiblemente mantenerse sin cambios. Una alternativa es aplicar reingeniería para reducir esta entropía, y así poder continuar con el mantenimiento del sistema. Por otro lado, cuando se llega a tal desorden, no es económicamente justificable continuar con el sistema, ya que es demasiado caro modificarlo. Lamentablemente, como se vio antes, la historia nos muestra que los sistemas raramente se desarrollan a tiempo, dentro del presupuesto y según las especificaciones originales.

Para Juan Manuel Cuella (sin fecha), existen cinco atributos de un sistema complejo:

- *Jerarquía*. Un sistema complejo se compone de subsistemas relacionados que tienen a su vez sus propios subsistemas y así sucesivamente hasta que se alcanza un nivel “ínfimo” de componentes elementales
- *Componentes*. La elección de qué componentes de un sistema son primitivos depende de la decisión del analista
- *Enlaces entre componentes*. Los enlaces internos de los componentes son más fuertes que los enlaces externos entre los componentes.
- *Patrones*. Los sistemas complejos tienen patrones comunes, que permiten la reutilización de componentes.
- *Evolución*. Los sistemas complejos son la evolución de sistemas simples que funcionaron previamente como formas intermedias estables.

Para Grady Booch (1994), la complejidad inherente al software se deriva de los siguientes cuatro elementos:

- *La complejidad del dominio del problema*
 - Gran cantidad de requisitos que compiten entre sí, incluso contradiciéndose. La forma habitual de especificar los requisitos son grandes cantidades de texto con unos pocos dibujos

- Desacoplamiento de impedancias entre usuarios del sistema y desarrolladores. Los usuarios suelen tener ideas vagas de lo que desean; dificultades de comunicación; distintas perspectivas de la naturaleza del problema
- Modificación de los requisitos con el paso del tiempo, pues los usuarios y los desarrolladores comienzan a compenetrarse mejor. Mantenimiento del software; evolución del software; conservación del software
- *La dificultad de gestionar el proceso de desarrollo*
 - Dirigir un equipo de desarrolladores. Mantener la unidad de acción; conservar la integridad del diseño
 - Manejar gran cantidad de código. Uso de herramientas; uso de lenguajes de muy alto nivel; reutilización de código; uso de componentes software; uso de patrones de diseño
- *El detalle que se puede alcanzar a través del software*
 - Evitar el desarrollo de todo desde el nivel mas inferior
 - Utilización de estándares
- *El problema de caracterizar comportamientos de sistemas discretos*
 - La ejecución de un programa es una transición entre estados de un sistema discreto
 - Es imposible hacer una prueba exhaustiva

La complejidad es una fuente común de errores en el software. Esto se manifiesta de forma abstracta y concreta, según José Manuel Milanés (sin fecha). Abstracta porque la complejidad más allá de ser una medida, es una resultante de la capacidad humana de realizar operaciones exactas y concretas porque numerosos estudios y experiencia de la industria han demostrado que el grado de complejidad tiene una correlación con los errores en el software.

Según Juan Carlos Granja (2000), cada vez más persiste la problemática en el desarrollo de software, ya que el nivel de complejidad va aumentando y no se alcanzan los niveles de calidad deseados, esto es un factor importante para la *crisis del software*.

Para Marcela Ridaio (2002), la mejor forma de dominar la complejidad del software y lograr confiabilidad en los productos de esta industria es llevar un proceso de desarrollo de software sistemático seguido por una implementación rigurosa y disciplinada

Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber nos señalan que conforme los proyectos de software crecen en tamaño y complejidad, la atención cambia de factores técnicos a factores organizacionales y administrativos. Es necesario entonces llevar una metodología a nivel organizacional que maneje la complejidad y la calidad del proceso de software adecuadamente.

Asley Cristales (1996) citando a (Shen, 1987) indica que las metas para toda investigación en ingeniería de Software son mejoras en la productividad y la calidad, y que las propuestas para nuevas herramientas, métodos y ambientes incluyen afirmaciones acerca de estos dos aspectos.

Dos razones importantes para enfatizar sobre calidad de Software según Asley Cristales (1996) citando a Littlewood: La naturaleza crítica de algunas tareas realizadas por la computadora y el creciente costo del desarrollo de Software

La calidad se define como la excelencia de un producto y/o servicio que inicia con el hecho de detectar los verdaderos requerimientos del cliente, y culmina con el ofrecimiento de un producto que satisfaga todos esos requerimientos (Ishikawa, 1986).

La ISO indica que calidad es la totalidad de las cualidades y características de un producto, proceso o servicio que trae consigo su capacidad de satisfacer las necesidades establecidas e implícitas (Stockman, 1990).

El objetivo básico de un proyecto de desarrollo de Software es entregar un sistema que satisfaga los requerimientos funcionales del sistema y sea un producto de calidad (Lowe, 1991)

Proceso de Software y modelos de madurez

El proceso de Software es un conjunto de herramientas, métodos y prácticas que se usan para producir y mantener el software. Los objetivos de la administración del proceso de software es hacer productos de acuerdo a un plan, mientras se mejora la capacidad de la organización para hacer mejores productos. (Watts S. Humphrey, 2002)

Los procesos son como hábitos: difíciles de establecer pero aún mas difíciles de romper. Los ingenieros de software deben estar convencidos de la necesidad del cambio.

Si se quiere un sistema de software de alta calidad es necesario asegurarse de que cada una de las partes sea de alta calidad.

La productividad del software generalmente declina con el incremento del tamaño del producto (Watts S. Humphrey, 2002 citando a Boehm, 81).

El costo de la calidad para Watts S. Humphrey (2002) tiene tres componentes:

- Costos de fallas
 - Son los costos de diagnosticar una falla, hacer reparaciones necesarias y volver a la operación
- Costos de valoración
 - Son los costos de evaluación del producto para determinar su nivel de calidad
- Costos de prevención
 - Son los costos asociados con la identificación de las causas de los defectos y las acciones tomadas para prevenirlos en el futuro

El primer paso para atacar los problemas de Software es tratar todas las tareas de Software como un proceso que puede ser controlado, medido y mejorado. De esta forma podemos definir a un proceso como el conjunto de tareas que cuando se desempeñan correctamente producen resultados deseados. Un proceso de Software efectivo debe considerar las relaciones de todas las tareas requeridas, las herramientas y métodos a usar y la habilidad, capacitación y motivación de las personas involucradas. (Watts S. Humphrey, 1990)

Humphrey (1990) nos dice que para mejorar sus capacidades de Software, las organizaciones deben tomar estos seis pasos:

- Entender el estado actual de su proceso o procesos de desarrollo
- Desarrollar una visión del proceso deseado
- Establecer una lista de acciones para los procesos que se requieren mejorar de mayor a menor importancia
- Producir un plan para cumplir las acciones requeridas
- Comprometer los recursos para ejecutar el plan
- Volver al primer paso

Sami Zahran (1998) nos hace la observación de que los procesos de disciplina de las organizaciones a varios niveles son como las etapas de las personas a diferentes etapas de maduración.

Kim Caputo (1999) nos dice que hay mucha gente talentosa en la industria del software, quienes son muy buenas en lo que hacen, de forma individual. Pero, aunque son muy brillantes, si se enfocan exclusivamente en sus propias responsabilidades, ignorando sus responsabilidades compartidas, no contribuyen al mejoramiento de la empresa del proceso de software.

Los modelos de madurez especifican metas de mejoramiento que constituyen la base para elaborar un plan de acción futuro. Permiten un ordenamiento del proceso, reduciendo los riesgos al enfrentar la complejidad de la producción del software. (Elizabeth Almeraz, Mariana Pérez-Vargas, 2002)

El modelo para el mejoramiento del proceso de Gestión de Configuración de Software se aplica a un conjunto de empresas, se proporcionan los resultados y recomendaciones y se aplica un enfoque para demostrar su aplicabilidad en el desarrollo de un plan de acción futuro. (Rodolfo Villarroel, 2002)

Mark C. Paul, Charles V. Weber (1995) nos dicen que un Nivel de Maduración: Es un periodo evolutivo bien definido para obtener un proceso de software maduro.

Alfredo Weitzenfeld (2003) nos hace una breve descripción de los modelos de madurez mas aceptados por la industria del software:

CMM (Capability Maturity Model)

El modelo "clásico" en el tratamiento de la capacidad de los procesos de desarrollo y su madurez es CMM (Capability Maturity Model) del SEI (Software Engineering Institute). CMM tiene como objetivo evaluar los procesos en sus distintos niveles de madurez, identificar los niveles a través de los cuales una organización debe formarse para establecer una cultura de excelencia en la ingeniería de software. El modelo de madurez de procesos fue generado a través de la experiencia colectiva de los proyectos más exitosos de software, generando así un conjunto de prácticas importantes que deben ser implantadas por cualquier entidad que desarrolla o mantiene software.

El CMM es un marco de referencia representando una ruta de recomendaciones de mejoras para organizaciones que quieren incrementar su capacidad del proceso de Software. Es un modelo descriptivo en el sentido de que describe atributos esenciales que se esperan encontrar en una organización que se encuentra en un nivel en particular. Es un modelo normativo en el sentido de que las prácticas detalladas caracterizan los tipos normales de comportamiento que se esperan en una organización desarrollando proyectos de gran escala. (Mark C. Paul, Charles V. Weber, 1995)

El CMM ha demostrado ser un diferenciador importante de nivel comercial, pues además de permitir mejorar internamente los procesos de las organizaciones, representa una manera estándar e internacional de comparar (hacer benchmarking) objetivamente a diferentes proveedores. De hecho, el Departamento de Defensa de EE.UU. requiere que todos sus proveedores de software sean nivel 3 o superior. Este requerimiento se ha extendido en la mayoría de las grandes empresas de EE.UU. que subcontratan empresas para el desarrollo de sus sistemas de software y por lo mismo la industria de desarrollo de software en la India ha tenido un gran auge. (Elizabeth Almeraz,; Mariana Pérez-Vargas, 2002)

En una entrevista al Dr. Pankaj Jalote, Ex-Vicepresidente de Calidad de Infosys, comentó textualmente, que la razón por la cual en la India se había adoptado el SEI-CMM., era muy sencilla y lógica ya que la industria de software de la India ha estado y está orientada a la exportación de software a diversos países, principalmente a EE.UU. y Europa, países que han determinado el desarrollo de software bajo estándares internacionales. Debido a que sus principales compradores están esparcidos por todo el mundo, la industria de desarrollo de software de la India necesitaba seguir un marco de referencia globalmente aceptado y alineado a los estándares internacionalmente reconocidos. Esto generó la necesidad de adoptar modelos como ISO 9000 y SEI-CMM.. El Dr. Jalote se remontó a la historia de la industria de la India y nos comentó que al inicio, sus principales clientes eran empresas europeas y por lo tanto el primer estándar que adoptaron fue ISO 9000, ya que era el requerimiento del mercado para poder adquirir el software desarrollado por la India. De esta manera, las empresas exportadoras de software de la India, entre 1993 y 1996, se certificaron en ISO 9000. Más adelante tuvieron que

evolucionar a un nuevo estándar ya que el mercado y sus principales compradores, en este caso EE.UU., cambiaron el requerimiento hacia SEI -CMM., y por lo tanto las empresas de la India para seguir en el negocio de desarrollo y exportación de software, dejaron atrás la certificación ISO 9000 y cambiaron al modelo de SEI -CMM. (Elizabeth Almeraz,; Mariana Pérez-Vargas, 2002)

Áreas de proceso claves (KPA): En el CMM, excepto por el nivel 1, cada nivel de maduración se decompone en varias áreas de proceso claves que identifican los factores que deben ser examinados para lograr un nivel de maduración. (Mark C. Paul, Charles V. Weber, 1995). Áreas de proceso claves para el nivel 2: Se centran en los aspectos de los proyectos de software concernientes al establecimiento de controles administrativos básicos.

ISO 9000-3

ESSI-SCOPE (Sin fecha), nos dice que ISO 9000-3 es la parte del ISO 9001 enfocada al desarrollo, provisión y mantenimiento de software. Y que se divide en tres grupos, cada uno con sus propias actividades:

Requerimientos generales de la compañía y administración, que involucra las actividades: Responsabilidad de la gerencia; sistema de calidad; auditoria de sistemas de calidad interno y acción correctiva

Requerimientos de actividades de soporte, que involucra: Administración de la configuración; control de documento; registros de calidad; métricas; reglas, prácticas y convenciones; herramientas y técnicas; compras; producto de software incluido y capacitación.

Requerimientos de proyectos y fase de mantenimiento involucra las actividades denominadas como: General; revisión de contrato; especificación de requerimientos de compra; planeación del desarrollo; planeación de la calidad; diseño e implementación; validación y pruebas; aceptación; replicación, entrega e instalación; así como mantenimiento.

Dado que el ISO 9000-3 es un conjunto de guías para la aplicación del ISO 9001, el método de valoración es el mismo procedimiento de auditoria del ISO 9001 y consiste básicamente en los pasos: Introducción del proyecto, auditoria preliminar de los elementos más importantes de QA, verificación de los elementos y auditoria para la certificación (ESSI-SCOPE, Sin fecha).

La certificación es equivalente al nivel 3 de la escala de SEI.

PSP/TSP

El PSP (Personal Software Process) es una tecnología que tiene como justificación la premisa de que la calidad de software depende del trabajo de cada uno de los

ingenieros de software y de aquí que el proceso diseñado debe ayudar a controlar, manejar y mejorar el trabajo de los ingenieros [Humphrey, 1998].

El objetivo de PSP es lograr una mejor planeación del trabajo, conocer con precisión el desempeño, medir la calidad de productos y mejorar las técnicas para su desarrollo. La instrumentación de esta tecnología consiste en lo que se denomina “evolución del PSP”. Se siguen ciertos pasos comenzando con las líneas base PSP0 y PSP0.1, el proceso personal de planeación PSP1 y PSP1.1, el manejo personal de calidad PSP2 y PSP2.1 y por último el proceso personal cíclico PSP3.

ISO/IEC 15504

El ISO/IEC 15504 conocido también como SPICE (Software Process Improvement and Capability dEtermination) por el nombre del grupo que lo desarrolló [Dorling, 1995] es un modelo de madurez de procesos internacional.

Fomenta productos de calidad, promueve la optimización de procesos y facilita la evaluación del producto a través de los procesos de desarrollo. El ISO/IEC 15504 tiene diversos alcances, se aplica tanto a nivel directivo como a nivel de usuarios para asegurar que el proceso se encuentra alineado con las necesidades del negocio, apoya en que los proveedores de software tengan que someterse a una sola evaluación para aspirar a nuevos negocios y busca que las organizaciones de software dispongan de una herramienta universalmente reconocida para dar soporte a su programa de mejoramiento continuo.

El ISO/IEC 15504 tiene tres características principales: el marco de valor que contempla una dimensión funcional del proceso, la evidencia para la evaluación y la recurrencia dada por la selección de instancias de proyectos o productos.

Está conformado por 9 documentos que permiten instrumentar paso a paso el modelo con su correspondiente evaluación.

El modelo establece un común denominador para una evaluación uniforme de los procesos de software, aunque la evaluación no pretende ser una nueva instancia de certificación, sino que a través de los resultados se pretende demostrar lo adecuado del mismo.

Al igual que CMMi (Capability Maturity Model integrated), el ISO/IEC 15504 integra una serie de niveles por la que sus procesos deberán pasar para obtener cómo resultado final la madurez. Los niveles son: Nivel 0 Incompleto, Nivel 1 Fabricado informalmente, Nivel 2 Planeado, Nivel 3 Bien definido, Nivel 4 Controlado cuantitativamente, y Nivel 5 Mejora continua.

Adicionalmente hay una definición de procesos generales que abarcan a toda la organización y a través de los cuáles se identifica el cómo lograrlos: Cliente – Proveedor CUS, Ingeniería ENG, Administración MAN, Apoyo o soporte SUP y Organización ORG. Los procesos generales son soportados por prácticas específicas

que deberán cumplirse para lograr un paso de niveles, además de la estrecha relación entre los mismos.

ISO/IEC 15504 hace hincapié en la calidad y actualización, así como en la vigencia del producto. Ya que la tecnología es cambiante, las fases que marca el modelo del ISO/IEC 15504 son sin duda uno de los pilares en que se tendrá que trabajar con la mayor dedicación para obtener calidad en el producto y que el servicio del mismo sea excelente, además de generar la confianza necesaria hacia la dirección y hacia el usuario de donde se obtiene la información.

Sus estándares se centran en puntos de proceso de Software pero también está involucrado con personas, tecnología, prácticas administrativas, soporte a clientes y calidad. (Mark C. Paul, Charles V. Weber, 1995)

PEMM

PEMM (Performance Engineering Maturity Model) (Scholz, 1999) presenta un modelo para evaluar los niveles de integración, aplicación, ejecución y diseño, llamado ingeniería de la ejecución del modelo de madurez. Al igual que el ISO/IEC 15504 se apoya en el modelo de madurez de capacidades CMM. El objetivo de PEMM es poder evaluar la Ejecución de la Ingeniería (EI) así como la integración del proceso. El modelo sirve tanto para evaluar una organización como los propios desarrollos de procesos tecnológicos específicos. Sirve también para definir el criterio al escoger un proveedor de software para los productos críticos o semi-críticos de la compañía.

Al igual que el CMM, PEMM cuenta con 5 niveles, los cuales determinan la mejora del comportamiento de ejecución y el decremento del riesgo de ejecución a través de estos niveles.

TickIt

Tick It (Tick It, 1992), desarrollado por el Departamento de Comercio e Industria del Reino Unido, surge por la poca adopción de las normas internacionales de calidad ISO 9000 para el área de desarrollo de software. TickIt es primordialmente una guía que presenta las estrategias para lograr la certificación en la producción de software a través de la interpretación de los estándares ISO.

Los objetivos principales de TickIt son, además de desarrollar un sistema de certificación aceptable en el mercado, estimular a los desarrolladores de software a implementar sistemas de calidad, dando la dirección y guías necesarias para tal efecto. El objetivo de certificación es demostrar que las prácticas necesarias para asegurar la calidad durante el desarrollo de software existen y son verificables (Blackman, 1995). En general el modelo permite certificar cualquier tipo de proyecto a través de una estructura más flexible.

La guía de auditoría provee la liga necesaria para que la conformación (o no) del sistema auditado respecto al modelo TickIt, pueda ser expresada en función de los criterios de la ISO 9001, logrando así la aplicación de esta última al desarrollo de software. Finalmente, los requerimientos de experiencia y conocimientos que se piden a los auditores hacen posible la aplicación del modelo, suponiendo que la experiencia y conocimientos de los auditores no se vean obsoletos frente a prácticas y técnicas nuevas dentro del medio de desarrollo de software.

Esta guía se compone de (i) un capítulo de conceptos de calidad, (ii) la norma ISO 9000-3, (iii) una serie de guías para proveedores y compradores, (iv) una guía para la auditoría del sistema de calidad, (v) el proceso de certificación y (vi) guías complementarias.

Bootstrap

La metodología de este modelo se origina de un proyecto llamado ESPRIT, en cooperación con la industria europea. Su propósito era analizar la conciencia de las buenas prácticas ingeniería de software. La metodología está basada en el SEI – CMM y tiene influencias de otros modelos como el ISO – 9000 y ESA PSS-05-0. (Anna Hoff y Anna Wikström, 2001)

Los pasos de valoración del Bootstrap son:

- Preparación
- Fase de ejecución
- Fase de planeación del mejoramiento

Este modelo cuenta con 6 niveles de madurez en una escala muy similar a los del SEI – CMM y cada nivel lo divide en cuarteles (Anna Hoff y Anna Wikström, 2001)

MoProSoft

Es el Modelo de Procesos para la Industria de Software, desarrollado a solicitud de la Secretaría de Economía para servir de base a la Norma Mexicana para la Industria de Desarrollo y Mantenimiento de Software bajo el convenio con la Facultad de Ciencias, Universidad Nacional Autónoma de México (Oktaba, 2003).

Durante los últimos dos años, en el marco del PROSOFT, el programa en Pro de la Industria del Software mexicana, la mesa 6 (Alcanzar niveles internacionales en capacidad de procesos), encabezada por Hanna Oktaba a solicitud de la SE; desarrolló un Modelo de Procesos para la Industria de Software. La liberación del llamado MoProSoft.

Según Software.net.mx (2003), este modelo representa una de las mejores opciones para que las empresas de desarrollo de software en el país lleguen a la estandarización de su operación, lo que las llevará a estar en la posibilidad de

ofrecer servicios de mayor calidad, volviéndose más competitivos en el mercado local e internacional.

El modelo puede ser utilizado por las pequeñas y medianas empresas de desarrollo de software en México como una herramienta para establecer sus procesos de acuerdo a sus necesidades o bien, para identificar elementos que falten por cubrirse.

Debido a su estructura y diseño, el MoProSoft, resulta de fácil comprensión y aplicación, además de que funciona como base para alcanzar evaluaciones efectivas de otros modelos con reconocimiento internacional, como CMM (Software.net.mx, 2003).

El MoProSoft, contempla tres categorías de procesos:

- Operación. Administración de proyectos de manera específica así como Desarrollo y Mantenimiento de Software.
- Gestión. Gestión de Procesos, Proyectos, y Recursos (humanos, materiales, y de servicios)
- Alta Dirección. Gestión de Negocios, punto que según Software.net.mx (2003) no contemplan otros modelos.

Cada uno de los procesos definidos en el modelo, cuenta con indicadores y metas cuantitativas que permitirán a las empresas evaluarse de una manera objetiva. También se plantean una serie de entregables que arroja el proceso, y que pueden ser adaptadas a las necesidades de cada empresa, e incluso de cada proyecto (Software.net.mx, 2003).

Para Software.net.mx (2003), MoProSoft esta bien fundamentado en modelos como el CMM y normas internacionales como ISO 15504 FDIS, por tanto es candidato a convertirse en una norma mexicana. El establecimiento de este modelo como norma, provocará que tanto el gobierno como la Iniciativa Privada que utiliza los servicios y productos de las empresas desarrolladoras de software tengan la seguridad de que sus proveedores tienen un alto nivel de calidad.

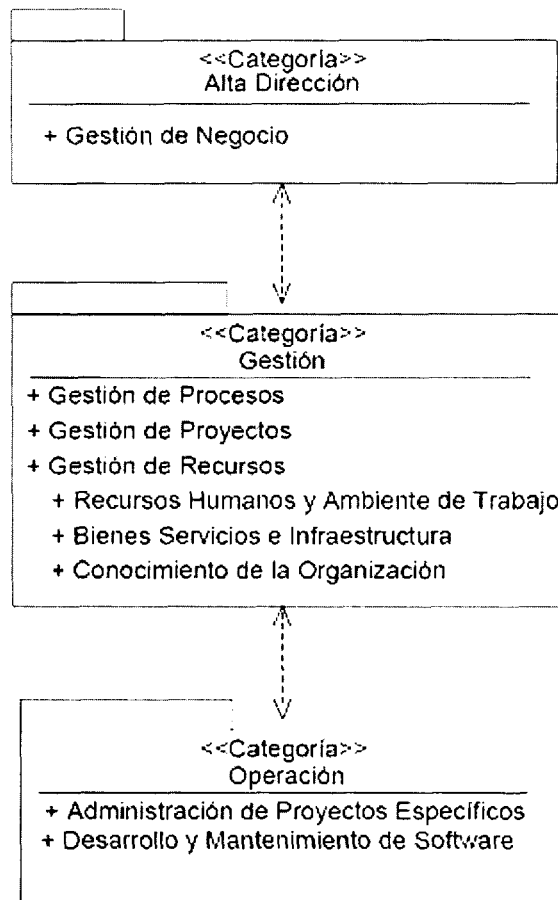


Fig. 2 - 1 Diagrama de categorías de procesos del MoProSoft

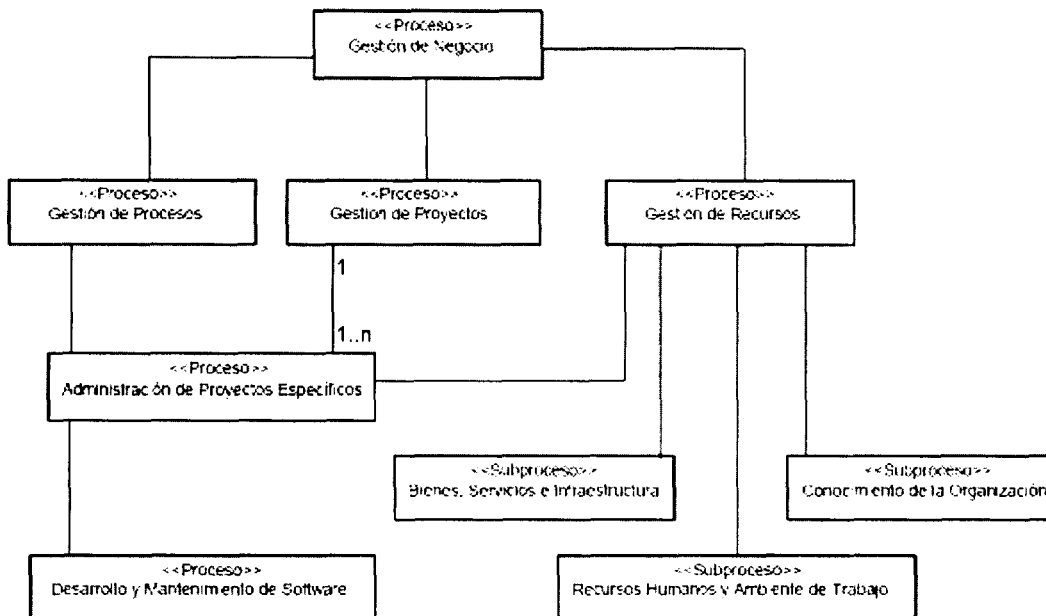


Fig. 2 - 2 Diagrama de relación entre procesos del MoProSoft

Al momento de que se está editando el reporte de esta investigación se lanzó una convocatoria para participar en las pruebas controladas de este modelo.

El alcance de este proyecto comprende el llevar a cabo pruebas controladas de MoProSoft y EvalProSoft en cuatro empresas, así como certificar los procesos implantados en ISO 9000. El proyecto también pretende obtener información sobre el esfuerzo, costo y tiempo necesarios para alcanzar un nivel de capacidades específico en CMMI (Software.net.mx, 2004).

Este proyecto involucra el compromiso de las empresas a cubrir los gastos de certificación en ISO 9000 y asignar los recursos humanos y de infraestructura necesarias para realizar el proyecto.

Los criterios que se definen para que las empresas interesadas puedan participar son:

- Cuenta con proyectos de desarrollo y/o mantenimiento de software a la medida.
- Tiene un mínimo de dos años de existencia.
- Su grupo de desarrollo y mantenimiento de software está formado por un máximo de 20 personas.
- Cuenta con el compromiso de la alta dirección para llevar a cabo la implantación y financiar la certificación ISO 9000 al final del proyecto.
- Disponibilidad de recursos para participar en el proyecto, el cual tendrá una duración aproximada de seis meses.
- Un responsable del proyecto.
- 20 % del tiempo del grupo de desarrollo para capacitación e implantación.
- Se encuentra ubicada en la Ciudad de México y su área metropolitana.

Certificación internacional en Ingeniería de Software

Certificación Internacional

ISO (Sin fecha) nos dice que es necesaria una certificación internacional, ya que permite:

- Progreso mundial en liberación del mercado
- Interpenetración de sectores
- Sistema de comunicaciones mundial
- Estándares globales para tecnologías emergentes
- Desarrollo de países
- El objetivo principal es facilitar la comercialización, intercambio y transferencia de tecnología mediante:
 - Calidad de producto extendida y confiabilidad a un precio razonable

- Mejora en salud, seguridad y protección ambiental y reducción de desperdicio
- Mayor compatibilidad e interoperabilidad de bienes y servicios
- Simplificación para un mejor uso
- Reducción del número de modelos y por lo tanto, reducción de costos
- Incremento en la eficiencia de la distribución

Para Publigraf Internacional, S.A. (Sin fecha), los beneficios pueden incluir:

- Un incremento de la confianza por parte del cliente
- La capacidad de cumplir con el creciente número de requisitos contractuales que descalifican a aquellos licitadores no certificados
- Que se reduzca la probabilidad de auditorias múltiples
- La utilización de un logotipo de registro reconocido
- La inclusión en los directorios de las empresas certificadoras

SGS Uruguay Limitada (Sin fecha) nos dice que la certificación internacional asegura:

- El cumplimiento de regulaciones locales e internacionales.
- Consistencia y uniformidad de producto.
- Incorporación de integridad y confianza al sistema.
- Acceso de marcas en productos.
- Aceleración del reconocimiento y confianza en un producto. Los clientes se inclinarán por los productos certificados

Certificación internacional en Ingeniería de Software

Mark C. Paul, Charles V. Weber (1995) nos resalta los beneficios de adquirir un estándar internacional en el proceso de software son:

- Ayuda a los compradores a determinar la capacidad de los proveedores de Software y asumir los riesgos que involucran la selección de un proveedor sobre otro
- Los proveedores de Software solo se someterán a un esquema de evaluación de proceso y no hacia numerosos esquemas
- Las organizaciones tendrán una herramienta de reconocimiento internacional para soportar sus programas de mejora del proceso
- Ayudará a los administradores a asegurarse de que el proceso está alineado con las necesidades de negocio de la organización

SEI - Capability Maturity Model for Software

Los componentes del CMM incluyen:

- Niveles de Maduración

Es un nivel evolutivo bien definido en la obtención de un proceso de software maduro. Los cinco niveles de maduración proveen el máximo nivel de la estructura del CMM

- Capacidades de proceso

Describen el alcance de los resultados esperados que pueden ser obtenidos llevando a cabo el proceso de software. Con este alcance se pueden predecir los resultados esperados del próximo proyecto de software que la organización puede soportar

- Áreas de proceso claves

Cada nivel de maduración está compuesto de áreas de proceso claves, cada una de ellas identifica un cluster de actividades relacionadas que, cuando se llevan a cabo de forma colectiva, se alcanzan un conjunto de metas consideradas importantes para establecer capacidades de proceso para ese nivel de madurez.

- Metas

Resumen las prácticas claves de las áreas de proceso claves y pueden usarse para determinar si una organización o proyecto y pueden usarse para determinar si las áreas de proceso claves están implantadas efectivamente. Estas metas significan los alcances, límites y propósito de cada área de proceso clave.

- Características comunes

Las prácticas clave están divididas a través de cinco secciones de características comunes:

- *Responsabilidad de actuar.* Describe las acciones que la organización debe tomar para asegurarse que el proceso se establezca y perdure. Comúnmente involucre el establecimiento de políticas empresariales y patrocinio de los altos administradores.
- *Habilidad de actuar.* Describe los prerequisites que deben existir en el proyecto u organización para implantar competentemente el proceso de software. Típicamente involucra recursos, estructuras organizacionales y capacitación.
- *Actividades desarrolladas.* Describe los roles y procedimientos necesarios para implementar un KPA. Comúnmente tiene que ver con el establecimiento de planes y procedimientos, ejecución y rastreo del trabajo y cuando es necesario la toma de acciones correctivas.

- *Mediciones y análisis.* Describe la necesidad de medir el proceso y analizar las mediciones. Típicamente incluye ejemplos de las mediciones que podrían tomarse para determinar el status y efectividad de las actividades desarrolladas.
- *Verificación de la implantación.* Describe los pasos para asegurar que las actividades se están realizando en conformidad con el proceso que se ha establecido. Comúnmente engloba a las revisiones y auditorias mediante la administración y el aseguramiento de la calidad de software (SQA).

Las características comunes indican si la implantación e institucionalización de las áreas de proceso claves es efectiva, repetible y duradera

- Practicas claves

Cada practica clave está descrita en términos de prácticas clave que, cuando se implementan, ayudan a satisfacer las metas. Describen la infraestructura y actividades que mas contribuyen a la implantación e institucionalización efectiva de las áreas claves de proceso

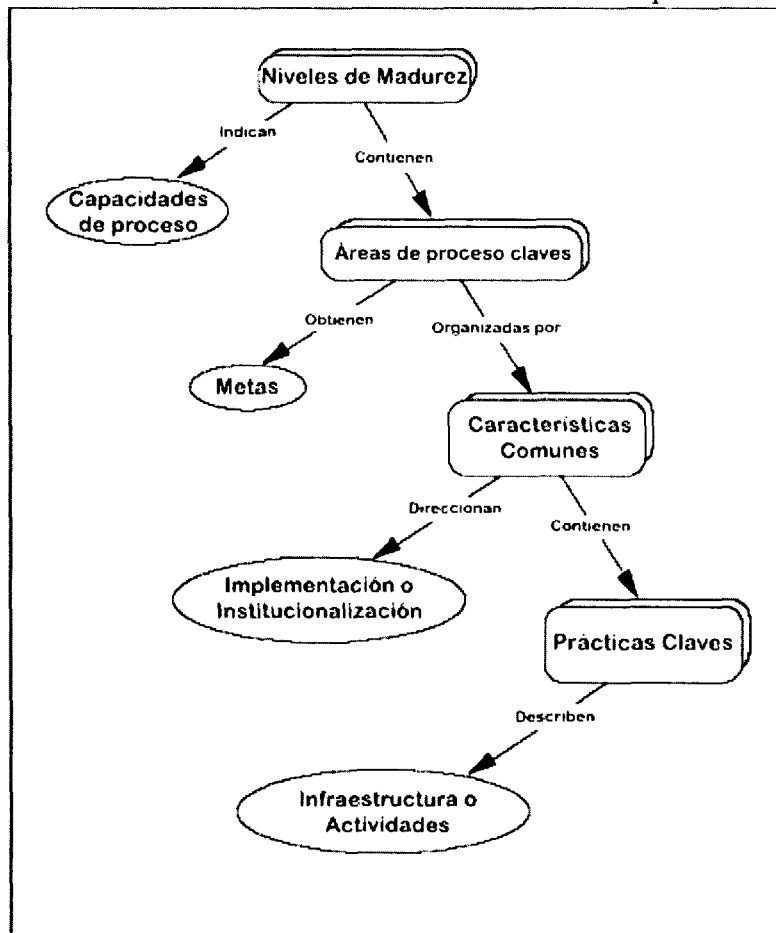


Fig. 2 - 3 Componentes del CMM

A continuación se hace una breve descripción de los niveles de madurez SEI-CMM con sus respectivas KPAs:

Nivel 1 – Inicial

El proceso de software en este nivel se caracteriza por ser caótico. Pocos procesos están definidos y el éxito depende de esfuerzos individuales.

Nivel 2 – Repetible

Procesos básicos de administración de proyectos son establecidos para rastrear costos, calendarización y funcionalidad. Existe una disciplina en el proceso para repetir éxitos anteriores con aplicaciones similares. Las KPAs de este nivel son:

- Administración de requerimientos (RM)
- Planeación de proyectos de Software (PP)
- Rastreo y supervisión de proyectos de Software (PT)
- Administración de subcontratos de Software (SM)
- Aseguramiento de la calidad del Software (QA)
- Administración de configuración de Software (CM)

Nivel 3 – Definible

El proceso de software tanto para las actividades administrativas como ingenieriles es documentado, estandarizado e integrado en un proceso de software estándar para la organización. Todos los proyectos usan una versión estándar aprobada y ajustada del proceso de software de la organización para desarrollo y mantenimiento del mismo. Las KPAs de este nivel son:

- Concentración en los procesos de la organización (PF)
- Programa de capacitación (TP)
- Administración Integral de Software (IM)
- Ingeniería de producto de Software (PE)
- Coordinación de intergrupos (IC)
- Revisiones en par (PR)

Nivel 4 – Administrable

Se recolectan mediciones detalladas del proceso de software y calidad de los productos. Ambos son cuantitativamente entendidos y controlados. Las KPAs de este nivel son:

- Administración cuantitativa de procesos (QP)
- Administración de calidad de Software (QM)

Nivel 5 – Optimizable

Proceso de mejora continua es habilitado por una retroalimentación cuantitativa del proceso y piloteo de ideas y tecnologías innovadoras Las KPAs de este nivel son:

- Prevención de defectos (DP)
- Administración de cambios de Tecnología (TM)
- Administración de cambios de procesos (PC)

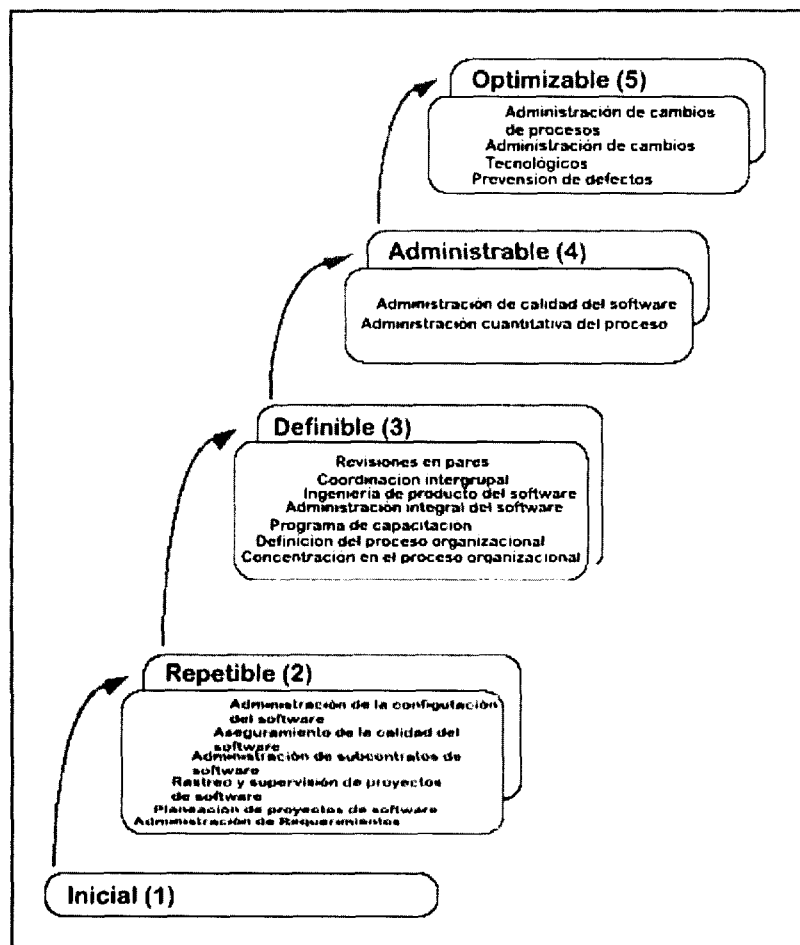


Fig. 2 - 4 Niveles de madurez con sus respectivas áreas de proceso claves

En el CMM se incluye el Nivel 1 (etapa inicial) para establecer una base de comparación de los mejoramientos de los procesos con los niveles más altos de maduración (Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber, 1993) y por lo tanto no tiene KPAs específicamente definidos.

En la siguiente tabla se explican las metas y prácticas claves más importantes para el nivel de madurez 2

KPA (Nivel 2)	Metas	Prácticas clave importantes
Administración de requerimientos (RM)	<ol style="list-style-type: none"> 1. Los requerimientos del sistema asignados al software están controlados para establecer una base para la ingeniería de software y uso administrativo. 2. Planes de software, productos, y actividades se mantienen consistentes con los requerimientos del sistema asignados al software. 	<ol style="list-style-type: none"> 1. El grupo de ingeniería de software revisa los requerimientos asignados antes de incorporarlos al proyecto de software 2. El grupo de ingeniería de software utiliza los requerimientos asignados como la base para la planeación del software, productos de trabajo, y actividades 3. Los cambios a los requerimientos asignados son revisados e incorporados dentro del proyecto de software
Planeación de proyectos de Software (PP)	<ol style="list-style-type: none"> 1. Los estimados del software se encuentran documentados para ser usados en la planeación y rastreo de proyectos de software 2. Las actividades de proyectos de software y compromisos se encuentran planeados y documentados. 3. Los grupos y las personas involucradas están de acuerdo con sus responsabilidades asignadas hacia el proyecto de software. 	<ol style="list-style-type: none"> 1. El grupo de ingeniería de software participa en el equipo de propuesta del proyecto 2. La planeación del proyecto de software es inicializada en las primeras etapas de y en paralelo con el proceso de planeación global. 3. El grupo de ingeniería de software participa con otros grupos relacionados en la planeación global del proyecto durante la vida de dicho proyecto. 4. Los compromisos del proyecto de software hechos a grupos y personas externas a la organización son revisados con la alta administración de acuerdo a un procedimiento documentado. 5. Un ciclo de vida de software con etapas predefinidas de tamaño manejable es identificado o definido. 6. El desarrollo del plan del proyecto de software es llevado a cabo de acuerdo a un procedimiento documentado. 7. El plan para el proyecto de software está documentado. 8. Los productos de trabajo de software que son necesarios para establecer y mantener control del proyecto de software son identificados.

		<ol style="list-style-type: none"> 9. Los estimados para el tamaño (o cambio de tamaño) de los productos de software son derivados de acuerdo a un procedimiento documentado. 10. Los estimados para el esfuerzo y los costos del proyecto de software son derivados de acuerdo un procedimiento documentado. 11. Los estimados para los recursos de cómputo críticos del proyecto son derivados de acuerdo a un procedimiento documentado. 12. La calendarización del proyecto de software es derivada de acuerdo a un procedimiento documentado. 13. Los riesgos de software asociados con el costo, recursos, calendarización, y aspectos técnicos del proyecto son identificados, evaluados y documentados. 14. Los planes para las herramientas de soporte y servicios básicos están preparados. 15. Los datos de la planeación de software son grabados.
<p>Rastreo y supervisión de proyectos de Software (PT)</p>	<ol style="list-style-type: none"> 1. Los resultados actuales y las acciones son rastreadas de acuerdo a los planes de software. 2. Las acciones correctivas son tomadas y manejadas para cerrar cuando los resultados actuales y acciones se desvían significativamente de los planes de software. 3. Los cambios de las responsabilidades son acordados por los grupos y las personas. 	<ol style="list-style-type: none"> 1. Un plan de desarrollo de software documentado es usado para rastrear las actividades de software y el estado de la comunicación. 2. El plan de desarrollo del proyecto de software es revisado de acuerdo a un procedimiento documentado. 3. Los compromisos y los cambios a los compromisos del proyecto de software hechos a grupos y personas externas a la organización son revisados con la alta administración de acuerdo a un procedimiento documentado. 4. Los cambios aprobados a los compromisos que afectan al proyecto de software son comunicados a los miembros del grupo de ingeniería de software y a otros grupos de software relacionados. 5. El tamaño de los productos de trabajo de software son rastreados y acciones correctivas son tomadas en caso de ser necesario. 6. Los esfuerzos y costos del proyecto de software son rastreados y acciones correctivas son tomadas en caso de ser necesario. 7. Los recursos de cómputo críticos del proyecto son rastreados y

		<p>acciones correctivas son tomadas en caso de ser necesario.</p> <ol style="list-style-type: none"> 8. La calendarización del proyecto de software es rastreada y acciones correctivas son tomadas en caso de ser necesario. 9. Las actividades de ingeniería de software son rastreadas y acciones correctivas son tomadas en caso de ser necesario. 10. Los riesgos de software asociados con el costo, recursos, calendarización y aspectos técnicos del proyecto son rastreados. 11. Los datos de mediciones y los datos de replaneación actuales para el proyecto del software son grabados. 12. El grupo de ingeniería de software realiza revisiones internas periódicas para rastrear el progreso técnico, planes, actividades y problemas contra el plan de desarrollo de software. 13. Revisiones formales para direccionar los logros y resultados del proyecto de software son realizadas en impulsos seleccionados del proyecto de acuerdo a un procedimiento documentado.
<p>Administración de subcontratos de Software (SM)</p>	<ol style="list-style-type: none"> 1. El contratista principal selecciona subcontratados de software calificados. 2. El contratista principal y el subcontratado están de acuerdo con sus responsabilidades el uno con el otro. 3. El contratista principal y el subcontratado mantienen una comunicación continua. 4. El contratista principal rastrea los resultados y el rendimiento actuales del subcontratado de software comparándolos con los compromisos establecidos. 	<ol style="list-style-type: none"> 1. El trabajo a ser subcontratado está definido y planeado de acuerdo a un procedimiento documentado. 2. El subcontratado de software es seleccionado, basándose en una evaluación de las habilidades de desarrollar el trabajo por parte de los subcontratados, de acuerdo a un procedimiento documentado. 3. El acuerdo contractual entre el contratista principal y el subcontratado de software es utilizado como las bases para manejar el subcontrato. 4. Un plan de desarrollo del subcontratado de software documentado es revisado y aprobado por el contratista principal. 5. Un plan de desarrollo del subcontratado de software documentado y aprobado es utilizado para rastrear las actividades de software y el estado de la comunicación. 6. Los cambios a las cláusulas del trabajo del subcontratado de software y otros compromisos son resueltos de acuerdo a un procedimiento documentado.

		<ol style="list-style-type: none"> 7. La administración del contratista principal realiza revisiones periódicas de estado/coordiación con la administración del subcontratado de software. 8. Revisiones técnicas periódicas e intercambios son sostenidas con el subcontratado de software. 9. Revisiones formales para acceder los logros y resultados de la ingeniería del subcontratista de software son realizadas en impulsos seleccionados de acuerdo a un procedimiento documentado. 10. El grupo de aseguramiento de la calidad del contratista principal supervisa las actividades de aseguramiento de calidad del subcontratado de software de acuerdo a un procedimiento documentado. 11. El grupo de administración de configuración de software del contratista principal supervisa las actividades del subcontratado en cuanto a la administración de la configuración de acuerdo a un procedimiento documentado. 12. El contratista principal realiza pruebas de aceptación como parte de la entrega de los productos del subcontratado de software de acuerdo a un procedimiento documentado. 13. El rendimiento del subcontratista de software es evaluado en una base periódica, y la evaluación es revisada con dicho subcontratado.
<p>Aseguramiento de la calidad del Software (QA)</p>	<ol style="list-style-type: none"> 1. Las actividades del Aseguramiento de la calidad del Software se encuentran planeadas 2. La adición de productos y actividades de software a los estándares, procedimientos y requerimientos aplicables está verificada de manera objetiva. 	<ol style="list-style-type: none"> 1. Un plan de SQA es preparado para el proyecto de software de acuerdo a un procedimiento documentado. 2. Las actividades del grupo de SQA son llevadas a cabo de acuerdo con el plan de SQA. 3. El grupo de SQA participa en la preparación y revisión del plan, estándares y procedimientos del desarrollo del proyecto de software. 4. El grupo de SQA revisa las actividades de ingeniería de software para verificar el cumplimiento.

	<ol style="list-style-type: none"> 3. Los grupos y personas relacionados se encuentran informados de las actividades y los resultados del aseguramiento de la calidad del software. 4. Los casos de incumplimiento que no se pueden resolver dentro del proceso de software son atendidos por la alta administración. 	<ol style="list-style-type: none"> 5. El grupo de SQA audita productos de trabajo de software determinados para verificar el cumplimiento. 6. El grupo de SQA periódicamente reporta los resultados de sus actividades al grupo de ingeniería de software. 7. Desviaciones identificadas en las actividades de software y productos de trabajo de software son documentadas y manejadas de acuerdo a un procedimiento documentado. 8. El grupo de SQA efectúa revisiones periódicas de sus actividades y descubrimientos con el personal de SQA del cliente, según sea apropiado.
<p>Administración de configuración de Software (CM)</p>	<ol style="list-style-type: none"> 1. Las actividades de administración de configuración de software son planeadas. 2. Están identificados, controlados y disponibles los productos de trabajo de software seleccionados. 3. Los cambios a los productos de trabajo de software seleccionados son controlados. 4. Los grupos y personas relacionadas son informados del estado y contenido actual de las bases del software. 	<ol style="list-style-type: none"> 1. Un plan SCM es preparado para proyecto de software de acuerdo a un procedimiento documentado. 2. Un plan SCM documentado y aprobado es utilizado como la base para desarrollar las actividades SCM. 3. Un sistema de biblioteca de administración de configuración es establecido como el repositorio para las bases del software. 4. Los productos de trabajo de software a ponerse bajo administración de configuración están identificados. 5. Las solicitudes de cambios y los reportes de problemas para todos los elementos/unidades están inicializadas, registradas, revisadas, aprobadas y rastreadas de acuerdo a un procedimiento documentado. 6. Los cambios de las bases están controlados de acuerdo a un procedimiento documentado. 7. Los productos de la biblioteca base de software son creados y su lanzamiento está controlado de acuerdo a un procedimiento documentado. 8. El estado de la configuración de elementos/unidades está registrado de acuerdo a un procedimiento documentado. 9. Los reportes estándar documentando las actividades SCM y el contenido de la base de software están desarrollados y puestos

		<p>disponibles a los grupos y personas involucradas.</p> <p>10. Las auditorias de la base del software es realizada de acuerdo a un procedimiento documentado.</p>
--	--	--

Tabla 2 - 1 Metas y prácticas claves más importantes para el nivel de madurez 2

Categoría	Administración	Organización	Ingeniería
Niveles			
5 Optimizable		<ul style="list-style-type: none"> - Administración del cambio tecnológico - Administración de cambio de procesos 	<ul style="list-style-type: none"> - Prevención de defectos
4 Administrable	<ul style="list-style-type: none"> - Administración cuantitativa del proceso 		<ul style="list-style-type: none"> - Administración de la calidad del software
3 Definible	<ul style="list-style-type: none"> - Administración integral del software - Coordinación intergrupala 	<ul style="list-style-type: none"> - Concentración en el proceso organizacional - Definición del proceso organizacional - Programa de capacitación 	<ul style="list-style-type: none"> - Ingeniería del producto de software - Revisiones en par
2 Repetible	<ul style="list-style-type: none"> - Administración de requerimientos - Planeación de proyectos de Software - Rastreo y supervisión de proyectos de Software - Administración de subcontratos de Software - Aseguramiento de la calidad del Software - Aseguramiento de la calidad del Software - Administración de configuración de Software 		
1 Inicial	<ul style="list-style-type: none"> - Procesos rudimentarios 		

Tabla 2 - 2 Las áreas de proceso claves asignadas a categorías de procesos

ISO/IEC 15504 - Software Process Improvement and Capability dEtermination

El ISO/IEC 15504 (También conocido como SPICE - Software Process Improvement and Capability dEtermination) es una iniciativa internacional para desarrollar un estándar de evaluación de procesos de software. El proyecto es llevado a cabo bajo el auspicio del comité internacional de estándares de ingeniería de Software ISO/IEC JTC 1/SC 7, a través de su grupo de trabajo de evaluación de procesos de Software (WG10). Desde 1993 forma parte de el Organismo Internacional de Estándares (ISO). (European Software Institute, 1997).

Solórzano Corral (sin fecha) nos dice que el ISO/IEC 15504 tiene como objetivo producir un conjunto de estándares y documentos de guía para ayudar a los que adquieren, producen, desarrollan, suministran y mantienen software, en la mejora de la calidad y mantenibilidad dentro de unos márgenes de tiempo y costes razonables.

Sakari Pihlava (1996), de la universidad Helsinki University of Technology, en su tesis de maestría nos dice que consta de 7 documentos: una guía introductoria que describe como todos los documentos encajan juntos y seis guías técnicas relacionadas con practicas fundamentales, evaluación de procesos, evaluación de instrumentos, asesoría calificada y capacitación, mejoramiento de procesos y determinación de capacidades de procesos.

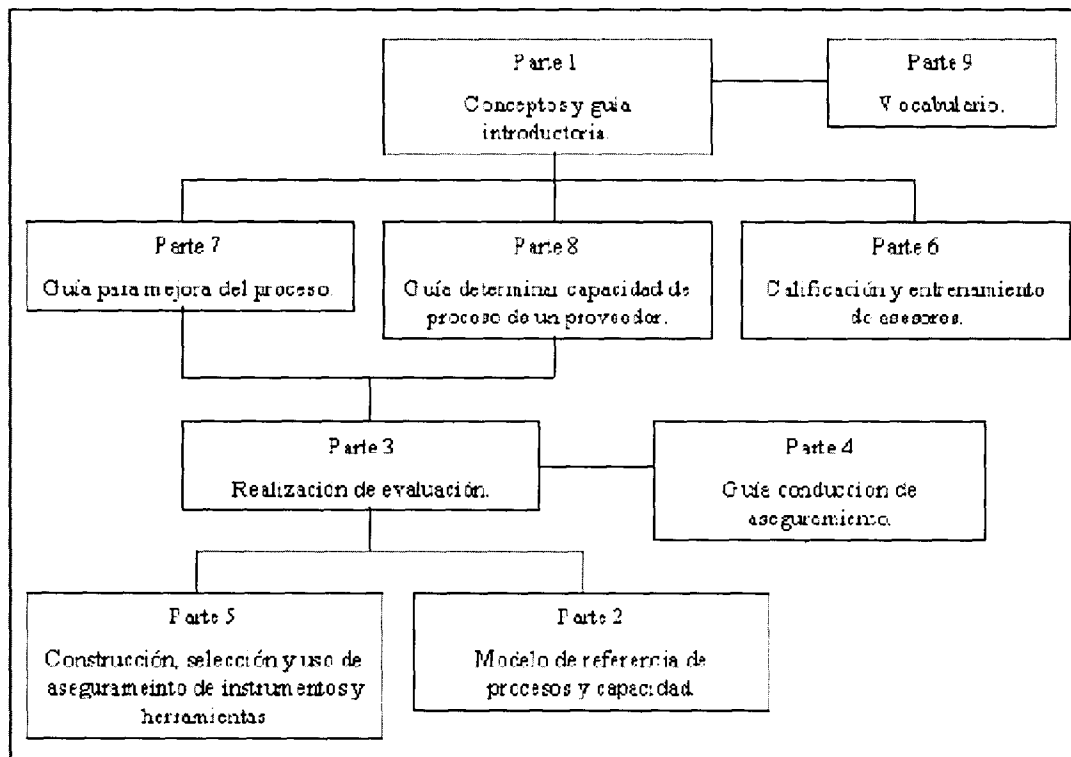


Fig. 2 - 5 Partes que conforman al ISO/IEC 15504

Describe un conjunto de áreas de proceso. Cada práctica básica puede ser vista en función de prácticas genéricas que describen la maduración de los procesos.

El estándar está diseñado para proporcionar resultados de evaluación repetibles y comparables dentro de contextos similares. La repetibilidad está basada en la guía para conducir la evaluación dentro de un marco de valor explícito y completo, y en el fundamento de los juicios en evidencia tipificada y recurrente. El marco incluye una arquitectura para calificar prácticas además de la definición de la experiencia y requisitos para los evaluadores.

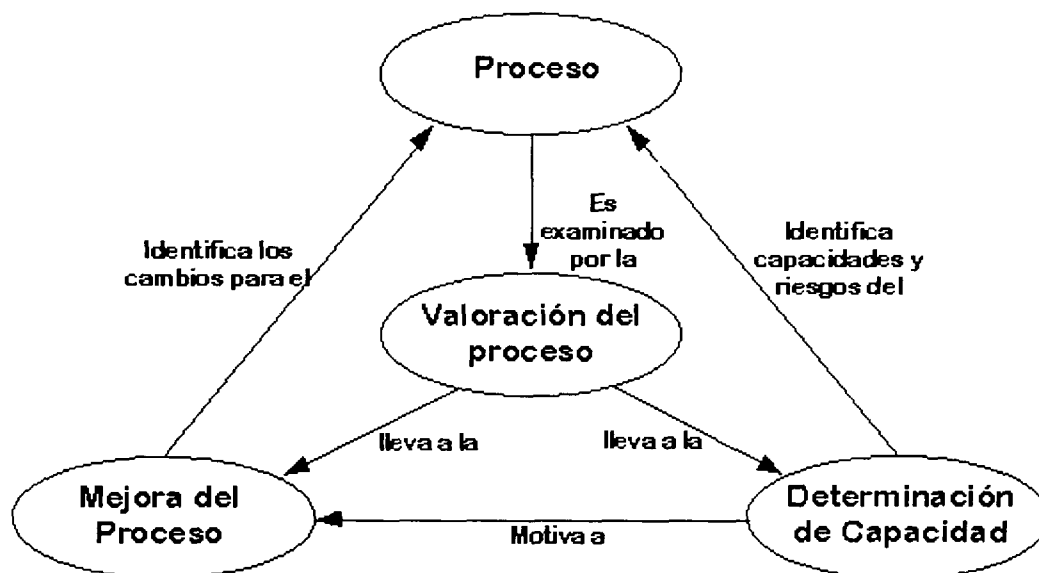


Fig. 2 - 6 Proceso de valoración del ISO/IEC 15504

El modelo es tridimensional: la primera dimensión es funcional (procesos), la segunda de capacidades (niveles), y la tercera de adecuación o efectividad (calificaciones). (Arnoldo Díaz, 1996)

Las practicas genéricas del ISO/IEC 15504, según Sakari Pihlava (1996), soportan la implementación e institucionalización de procesos y están agrupadas en niveles de maduración:

- Cliente – Proveedor, procesos que impactan directamente al cliente
- Ingeniería, procesos que directamente especifican, implementan o mantienen un sistema y producto de Software
- Proyecto, procesos que establecen el proyecto, coordinación y administración de sus recursos
- Soporte, procesos que habilitan y soportan el rendimiento de los otros procesos del proyecto
- Organización, procesos que establecen las metas de la organización y procesos de desarrollo, productos y obtención de recursos, lo que ayudará a la organización a lograr sus objetivos de negocio

Mario Côrtes nos dice que las categorías de los procesos del ISO/IEC 15504 son:

- Procesos primarios
- Procesos de soporte
- Procesos organizacionales

Las cinco categorías de procesos:

- CUS: Cliente – proveedor
- ENG: Ingeniería de Software
- SUP: Soporte
- MAN: Administración
- ORG: Organización

Los niveles de madurez del ISO/IEC 15504 son:

- Nivel 0 – Proceso incompleto
- Nivel 1 – Proceso desempeñado
- Nivel 2 – Proceso administrado
- Nivel 3 – Proceso establecido
- Nivel 4 – Proceso predecible
- Nivel 5 – Proceso optimizable

SEI - Capability Maturity Model for Software integrated

Los componentes del CMMi incluyen:

- Niveles de Maduración

Es un nivel evolutivo bien definido en la obtención mejor de procesos. Los cinco niveles de maduración proveen el máximo nivel de la estructura del CMMi escalonado.

Son una forma de predecir el rendimiento futuro de una organización dentro de una disciplina dada o conjunto de disciplinas.

- Áreas de proceso

Cada nivel de maduración está compuesto de áreas de proceso, cada una de ellas identifica un cluster de actividades relacionadas que, cuando se llevan a cabo de forma colectiva, se alcanzan un conjunto de metas consideradas importantes para establecer capacidades de proceso para ese nivel de madurez.

- Metas específicas

Resumen las prácticas claves de las áreas de proceso y pueden usarse para determinar si una organización o proyecto y pueden usarse para determinar si las áreas de proceso están implantadas efectivamente. Estas metas significan los alcances, límites y propósito de cada área de proceso.

- Metas genéricas

Se llaman genéricas porque la misma definición de meta aparece en múltiples áreas de proceso.

- Características comunes

Las prácticas clave están divididas a través de cinco secciones de características comunes:

- *Responsabilidad de actuar.* Describe las acciones que la organización debe tomar para asegurarse que el proceso se establezca y perdure. Comúnmente involucre el establecimiento de políticas empresariales y patrocinio de los altos administradores.
- *Habilidad de actuar.* Describe los prerrequisitos que deben existir en el proyecto u organización para implantar competentemente el proceso de software. Típicamente involucra recursos, estructuras organizacionales y capacitación.
- *Actividades desarrolladas.* Describe los roles y procedimientos necesarios para implementar una PA. Comúnmente tiene que ver con el establecimiento de planes y procedimientos, ejecución y rastreo del trabajo y cuando es necesario la toma de acciones correctivas.
- *Mediciones y análisis.* Describe la necesidad de medir el proceso y analizar las mediciones. Típicamente incluye ejemplos de las mediciones que podrían tomarse para determinar el status y efectividad de las actividades desarrolladas.
- *Verificación de la implantación.* Describe los pasos para asegurar que las actividades se están realizando en conformidad con el proceso que se ha establecido. Comúnmente engloba a las revisiones y auditorías mediante la administración y el aseguramiento de la calidad de software (SQA).

Las características comunes indican si la implantación e institucionalización de las áreas de proceso es efectiva, repetible y duradera.

- Productos de trabajo típicos

Estos componentes son informativos que proveen ejemplos de resultados de las prácticas específicas o genéricas.

- Subprácticas

Son descripciones detalladas que sirven de guía para interpretar las prácticas específicas o genéricas

- Amplificaciones de disciplina

Componentes informativos que contienen información relevante de acuerdo a una disciplina en particular y están asociadas con prácticas específicas.

- Elaboraciones de prácticas genéricas

Componentes informativos que aparecen en cada área de proceso para servir como guía en como las prácticas genéricas deberían aplicarse de manera única a esa área de proceso.

- Referencias

Son componentes informativos que conducen al usuario a información adicional o más detallada.

- Practicas específicas

Cada practica clave está descrita en términos de prácticas específicas que, cuando se implementan, ayudan a satisfacer las metas. Describen la infraestructura y actividades que mas contribuyen a la implantación e institucionalización efectiva de las áreas claves de proceso

- Practicas genéricas

Proveen institucionalización para asegurar que los procesos asociados con el área de proceso serán efectivos, repetibles y duraderos

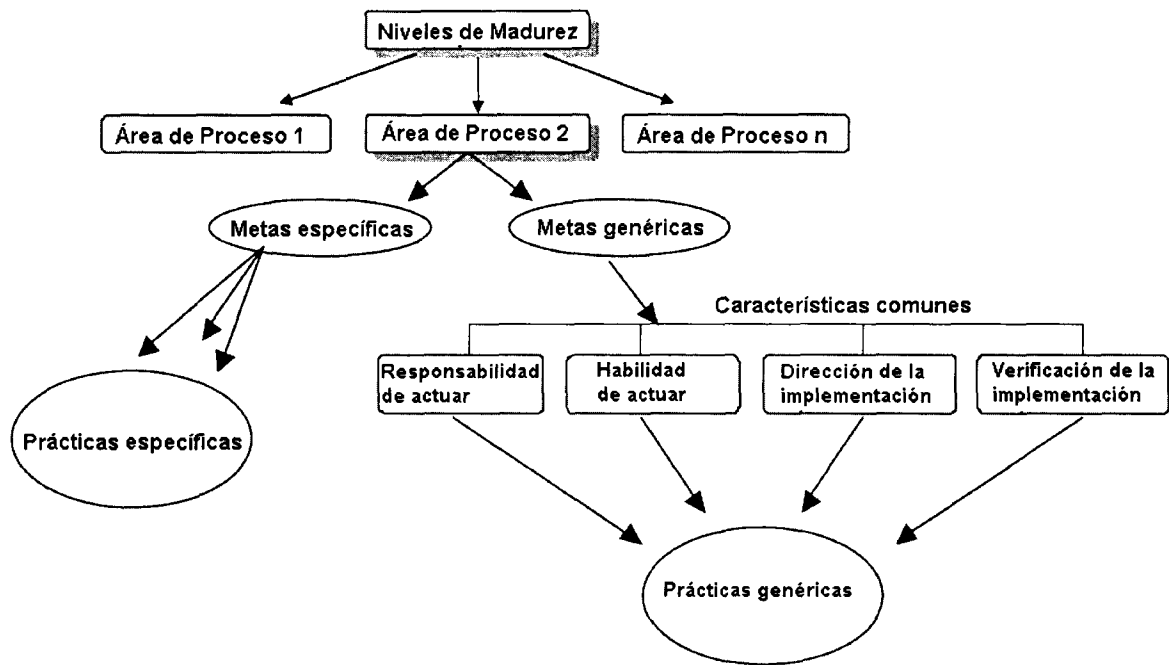


Fig. 2 - 7 Componentes del CMMi

A continuación se hace una breve descripción de los niveles de maduración SEI-CMM con sus respectivas áreas de proceso:

Nivel 1 – Inicial

El proceso de software en este nivel se caracteriza por ser caótico. Pocos procesos están definidos y el éxito depende de esfuerzos individuales. La empresa normalmente no es capaz de proveer un entorno estable.

Nivel 2 – Administrable

Ayudan a asegurar que las prácticas existentes permanecen en momentos de estrés. Cuando estas prácticas son puestas, los proyectos son llevados a cabo y administrados de acuerdo a sus planes documentados. Las áreas de proceso de este nivel son:

- Administración de requerimientos
- Planeación de proyectos de Software
- Supervisión y Control de proyectos de Software
- Administración de contratación de proveedores
- Análisis y mediciones
- Aseguramiento de la calidad de procesos y productos
- Administración de la configuración

Nivel 3 – Definible

El proceso de software tanto para las actividades administrativas como ingenieriles es documentado, estandarizado e integrado en un proceso de software estándar para la organización. Todos los proyectos usan una versión estándar aprobada y ajustada del proceso de software de la organización para desarrollo y mantenimiento del mismo. Las áreas de proceso de este nivel son:

- Requerimientos de desarrollo
- Solución Técnica
- Integración del producto
- Verificación
- Validación
- Concentración en el proceso organizacional
- Definición del proceso organizacional
- Administración de integración del proyecto
- Administración de riesgos
- Análisis y resolución de decisiones

Nivel 4 – Administrable Cuantitativamente

Se recolectan mediciones detalladas del proceso de software y calidad de los productos. Ambos son cuantitativamente entendidos y controlados. Las áreas de proceso de este nivel son:

- Desempeño del proceso organizacional
- Administración cuantitativa de procesos

Nivel 5 – Optimizable

Proceso de mejora continua es habilitado por una retroalimentación cuantitativa del proceso y piloteo de ideas y tecnologías innovadoras. Las áreas de proceso de este nivel son:

- Innovación y distribución organizacional
- Análisis y resolución causal

En el CMMi se incluye el Nivel 1 (etapa inicial) para establecer una base de comparación de los mejoramientos de los procesos con los niveles más altos de maduración y por lo tanto no tiene áreas de proceso específicamente definidas.

Factores de éxito para el mejoramiento de la calidad en la industria de software

Dirk Stelzer y Werner Mellis (1999) nos dicen que la administración del cambio organizacional es un elemento esencial del éxito en el SPI. Para ellos, los factores que

afectan al cambio organizacional en los esfuerzos de mejora del proceso de software se muestran en la siguiente tabla:

Factor de éxito	Explicación
Agentes de cambio y líderes de opinión	Los agentes de cambio comienzan y soportan los proyectos de mejora a nivel corporativo, los líderes de opinión son a nivel local
Comunicación y colaboración entusiasta	El grado en que los esfuerzos de comunicación preceden y acompañan al programa de mejora (comunicación) y al grado en que los miembros del staff de diferentes equipos y departamentos cooperan (colaboración)
Administración de compromisos y soporte	El grado en que la administración a todos los niveles de la organización patrocina el cambio
Dirección del proyecto de mejora	Grado en que una iniciativa de proceso de mejora está planeado y controlado de manera efectiva
Proveer entendimiento extendido	Grado en que el conocimiento del proceso de software actual y las actividades interrelacionadas del negocio es adquirido y transferido a través de la organización
Establecimiento de objetivos relevantes y realistas	El grado en que los esfuerzos de mejora tratan de contribuir al éxito de la organización (relevancia) y el grado en que los objetivos pueden ser alcanzados en un futuro inmediato (realidad)
Estabilización de procesos de cambio	Grado en que el proceso de software está continuamente soportado, mantenido y mejorado a nivel local
Participación del staff	El grado en que los miembros del staff participan en las actividades de mejora
Iniciativas de mejora personalizadas	Grado en que los esfuerzos de mejora son adaptados a las fuerzas y debilidades específicas de los diferentes equipos y departamentos
“Descongelamiento” de la organización	Grado en que la “resistencia interna” de un sistema organizacional por cambiar está reducido

Tabla 2 - 3 Factores que afectan al cambio organizacional en los esfuerzos de mejora del proceso de software

Para Nikhil Kalyanpur (2001) los 15 factores de éxito para el mejoramiento del proceso del software son:

- Poner al SPI en una perspectiva adecuada
- Obtener el compromiso de los altos ejecutivos
- Establecer expectativas realistas
- Centrarse en beneficios a largo término, balanceados por metas a corto plazo
- Conducir todo el negocio a través de administración de la calidad
- Obtener una alineación entre la alta gerencia y la administración media
- Incorporar este proceso a la estructura organizacional
- Establecer al SPI en el contexto organizacional
- Proveer procesos escalables
- Involucrar a los profesionistas recién egresados
- Obtener la participación para la evaluación
- Reducir servicios “pico”
- Inducir y educar a la gente en conceptos de SPI e Ingeniería de Software (SE)
- Proveer de un ambiente organizacional adecuado
- Motivar a las personas para adoptar el SPI

Por su parte, Stelios Frangos (1998) refuerza diciendo que los factores críticos de éxito del SPI son:

- Administración de compromisos
- Participación del personal
- Establecimiento de metas claras
- Entrenamiento

Kim Caputo (1999), en su libro “CMM Implementation Guide” nos da una lista de factores que siguen un patrón simple, en caso de que se busque mejorar el proceso de software mediante un modelo de certificación:

- Entender el modelo para la mejora del proceso (en caso del CMM, todo el proceso y sus prácticas clave contenidas en cada área de proceso)
- Entender la diferencia entre las prácticas actuales de la organización y las prácticas deseadas del modelo
- Definir los procesos necesarios para alcanzar las metas de mejora
- Asegurarse que los procesos sean seguidos y mejorados continuamente

Por otro lado, también identifica dos ejemplos de problemas al tratar de mejorar el proceso de software mediante grupos aislados y no institucionalmente:

- La planeación y actividades de mejora en “nuestro tiempo libre”
- Esfuerzos de mejora sin planeación ni seguimiento

Algunos de los consejos, que según Caputo ayudan en el proceso:

- Proveer soporte para hacer más fácil la adopción del proceso. Minimizar el arranque del esfuerzo inicial tanto como sea posible. Si hace sus trabajos mas fácil, la gente será mas cooperativa en el futuro
- Facilitar el soporte administrativo y en par. Hacer visible el soporte a través de juntas de discusión o retroalimentación escrita.
- Proveer visibilidad del progreso. Ayudar a las personas a centrarse en su progreso en vías de adoptar las prácticas deseadas y ayudarles a darse cuenta que esas prácticas son las que están haciendo la diferencia
- Utilizar un ciclo de mejora del proceso. Hacer correcciones a las cosas que no funcionan, y mantener informadas a las personas involucradas de que algunas cosas pueden mejorarse ahora, y algunas otras cosas podrán ser mejoradas mas tarde. El mejoramiento es continuo, no un evento de “todo o nada”

Algunas de las típicas respuestas de las organizaciones que ofrecen resistencia en la adopción de un modelo de mejora del proceso de software:

- La disciplina siempre interfiere con la creatividad
- Los calendarios están justos, los recursos están fijados y no tenemos ninguna elección. El nivel de calidad también está fijado. El nivel de calidad es lo que sea que esté en la fecha de entrega
- La administración de requerimientos es una perdida de tiempo ya que los requerimientos cambian frecuentemente
- Lo más pronto que se pueda estar en el departamento de “testing”, lo más pronto que vamos a poder estar en producción. Podemos probar la calidad dentro del producto
- Si definimos un solo proceso común y logramos que todos lo sigan, garantizará la calidad del producto
- No tenemos tiempo ni recursos suficientes para seguir un proceso definido
- Nuestro proyecto es diferente. El proceso no aplica a nosotros. La sobrecarga matará al proyecto
- La gente buena puede hacerlo sin un proceso
- Nosotros siempre contratamos personas calificadas, así que no necesitamos capacitación

India, ha maquilado software para Estados Unidos durante 15 años de política gubernamental ininterrumpida de apoyo. El gobierno indio ha invertido más de 25 mil millones de dólares en apoyo a la capacitación, certificación, promoción e infraestructura. Entre los factores que contribuyeron a que India empezara a producir software fue la relación de ingenieros indios residentes en Estados Unidos con las grandes empresas compradoras. Utilizó un capital humano bien capacitado y con baja remuneración para desarrollar proyectos en su territorio. Esto la llevó a alcanzar exportaciones anuales de software de aproximadamente 5,000 millones de dólares en 2000. (Bancomext, 2001)

Irlanda se especializó en el desarrollo de aplicaciones y basó el desarrollo de su industria de software en la atracción de empresas extranjeras por medio de programas de apoyo sumamente agresivos. Logró un rápido crecimiento, teniendo una tasa de crecimiento anual promedio de 19.3% (1993-2001) pero enfrenta la limitante de una mano de obra ya escasa. (Bancomext, 2001)

La industria de software de Canadá está altamente orientada a la exportación. La mayoría de las empresas perciben a Estados Unidos como su mercado doméstico. Canadá basa su producción en las necesidades del mercado externo, con productos enfocados a nichos específicos como: animación, gráficos, administración de documentos, extracción inteligente de datos y administración de escuelas. (Bancomext, 2001)

La industria de software australiana es considerada como sublíder mundial en proveeduría de servicios de aplicación tales como: contenidos de Internet, servicios de administración; servicios en línea que se ofrecen desde Australia a otros países (auditoría, facturación, publicidad). Una ventaja que crea oportunidades a Australia y sus proveedores de servicios es que los trámites y servicios gubernamentales han sido digitalizados. (Bancomext, 2001)

Para Bancomext (2001) los factores que han llevado a tener y mantener una posición competitiva en el mercado internacional en el desarrollo de Software son los que se muestran a continuación.

Concepto / Pais	India	Irlanda	Estados Unidos y Canadá	Israel	Singapur
Aseguramiento de la calidad	X				
Costo bajo de recursos humanos y operación	X			X	X
Diferenciamiento de recursos humanos	X				
Maquila para empresas de otros países (<i>Offshoring</i>)	X				
<i>Outsourcing</i>	X				
Política gubernamental	X				
Recursos humanos	X				
Alianzas internacionales		X	X		
Competencia entre empresas		X			
Competencias administrativas de globalización de recursos humanos y del control de proyectos. Certificación administrativa y de estrategias de negocios. Entrepreneurship		X	X		
Desarrollo de industrias relacionadas (infraestructura de apoyo financiero, físico ambiental, fiscal y logístico ofrecidos por el gobierno y la IP)		X	X		
Diferenciación de recurso humano (niveles profesionales o grado de especialización)		X			
Estabilidad política y económica		X	X		
Conocimiento del idioma Inglés		X	X		
Infraestructura en telecomunicaciones		X	X		X
Niveles educativos elevados		X			
Desarrollo de la industria doméstica			X		
Experiencia acumulada			X		
Innovación tecnológica			X		
Marco normativo (garantía de aplicación de los derechos de propiedad intelectual e industrial)			X		
Nichos especializados de mercado			X		
Competencia ingenieril y tecnológica				X	
Inversión extranjera					X
Aprovechamiento de diferencias de horario respecto de Estados Unidos para un trabajo ligado y continuo					X

Tabla 2 - 4 Factores que han llevado a tener y mantener una posición competitiva en el mercado internacional en el desarrollo de Software

Aplicación del CMM en empresas pequeñas

Mark Paulk (1999) nos comenta algunos de los problemas comunes en la interpretación del CMM para las pequeñas organizaciones/proyectos incluyen:

- ¿Qué significa pequeño? ¿En términos de gente? ¿Tiempo? ¿Tamaño del proyecto? ¿Lo crítico del producto?
- ¿Cuáles son los “requerimientos” del CMM? ¿Son los KPAs o las metas las que no deberían aplicarse a las pequeñas organizaciones/proyectos? ¿No existen variaciones de los buenos procesos?
- ¿Cuáles son los conductores y motivadores que causan el abuso del CMM?

Los factores asociados con la interpretación del CMM para pequeñas organizaciones o proyectos pueden variar, pero en su forma no son diferentes de cualquier otra organización interesada en la mejora del proceso de software.

Utilizar el CMM de manera correcta y efectiva requiere de un juicio y entendimiento profesional de cómo el CMM está estructurado para ser usado para diferentes propósitos. Una pregunta que debería estar presente en todas las discusiones de mejora del proceso de software es: “¿Por qué la organización está interesada en utilizar el CMM?” Si el deseo es mejorar el proceso, con un lazo directo hacia los objetivos del negocio e interés de invertir en dicho mejoramiento, entonces el CMM es un arma útil y poderosa (Mark Paulk, 1999).

Ya se ha hecho el esfuerzo de desarrollar un CMM personalizado para pequeñas organizaciones y proyectos (Mark Paulk, 1999 citando a Ginsberg, 1995), pero la conclusión del taller de personalización del CMM 1995 fue que ni siquiera se pudieron poner de acuerdo en lo que “pequeña” significa en realidad, el resultado fue un reporte sobre como personalizarse para el CMM, en lugar de un CMM personalizado para las organizaciones pequeñas.

Broadman y Johnson a una organización pequeña a aquella que tiene alrededor de 50 desarrolladores de software y a un proyecto pequeño a uno en el que tiene 20 desarrolladores (Johnson, 98 citado por Mark Paulk, 1999).

Uno de los primeros retos de las pequeñas organizaciones al utilizar el CMM es que el objetivo primario de su negocio es sobrevivir! Incluso después de decidir el estado actual es insatisfactorio y el proceso de mejora ayudará, encontrar los recursos y asignar responsabilidades para el proceso de mejora y todavía los siguientes procesos por definir y distribuir es una decisión difícil del negocio.

Para Mark Paulk (1999), las empresas pequeñas tienden a creer que:

- Todos somos competentes – La gente fue contratada para hacer su trabajo y no podemos proporcionar capacitación en términos de tiempo ni de dinero

- Todos nos comunicamos el uno con el otro – “Osmosis” funciona porque estamos muy cerca
- Todos somos héroes – Hacemos lo que es necesario hacer, las reglas no aplican a nosotros (solo aplican en la forma de tener el trabajo hecho), vivimos con cortos ciclos de tiempo y un alto stress

Las empresas pequeñas como las grandes tendrán problemas con requerimientos no documentados, errores de los administradores sin experiencia, asignación de recursos, capacitación, revisiones y documentación del producto. A pesar de esos retos, las organizaciones pequeñas pueden ser extraordinariamente innovadoras y productivas. Aunque hay problemas masivos que pueden requerir de un gran número de personas par resolverlos, en general los equipos pequeños son más productivos que los grandes.

Interpretando al CMM.

¿Dónde aplica el CMM para Software? El CMM fue escrito para proveer buenas prácticas de ingeniería de software y administración para **cualquier** proyecto en **cualquier** entorno (Mark Paulk, 1999).

En la práctica, nos dice Mark Paulk (1999), los ambientes donde se ha necesitado interpretación y personalización del CMM incluyen:

- Programas muy grandes
- Proyectos u organizaciones virtuales
- Proyectos distribuidos geográficamente
- Proyectos de rápido prototipado
- Organizaciones de investigación y desarrollo
- Organizaciones de servicio de software
- Pequeñas organizaciones y proyectos

Las guías de interpretación para pequeños proyectos y pequeñas organizaciones también aplican a grandes proyectos y grandes organizaciones.

El sentido común y la inteligencia son requeridos para utilizar al CMM correcta y efectivamente (Paulk, 1996 citado por el mismo en 1999).

Los componentes “normativos” del CMM son niveles de maduración, áreas de proceso claves y metas. Todas las prácticas en el CMM son informativas. Desde que las prácticas detalladas principalmente ayudan a empresas grandes de software, estas no son necesariamente apropiadas, tal como están escritas, para uso directo de las organizaciones pequeñas. Pero proveen de una perspectiva de cómo alcanzar las metas e implementar un proceso de software repetible, definido, medible y continuamente mejorado.

Una de las recomendaciones más frecuentes en la interpretación es desarrollar un mapa entre la terminología CMM y el lenguaje utilizado por la organización. También los roles deben de especificarse, las personas pueden cubrir diferentes roles; por ejemplo: una persona puede ser el administrador del proyecto, administrador de proyectos de software, administrador de SCM, etc. Estableciendo esto hace la interpretación del CMM mucho más simple y más consistente.

En general, podemos asumir que los KPAs y las metas siempre son relevantes a cualquier entorno, con la excepción de “Administración de subcontrato de Software” (Software Subcontract Management), la cual podría no ser aplicable si no hay subcontrataciones (Mark Paulk, 1999).

En general, según Mark Paulk (1999), nunca se ha visto un entorno en que lo siguiente no fuera necesario:

- Requerimientos del cliente (sistema) documentados
- Comunicación con el cliente (y con los usuarios finales)
- Acuerdo de compromisos
- Planeación
- Procesos documentados
- Estructura de trabajo

Un proyecto pequeño normalmente no necesita un grupo de SCM, pero la administración de configuración y control de cambios son siempre necesarios. Es posible que un grupo independiente de SQA no sea deseable, pero la verificación objetiva de que los requerimientos estén satisfechos siempre lo es. Un grupo independiente de pruebas puede no estar establecido, pero las pruebas siempre son necesarias.

Si vemos la definición de CMM sobre “grupo”, establece que un grupo puede variar de un solo individuo que dedica un tiempo parcial, hasta varios individuos asignados en tiempo parcial de diferentes departamentos, hasta varios individuos de diferentes departamentos en tiempo completo.

Obtener el patrocinio de los altos niveles ejecutivos es un factor crucial en la construcción de las capacidades de la empresa. Como individuos, podemos ejercer profesionalismo y disciplina dentro de nuestro ámbito de control, pero si una organización como un todo está interesada en cambiar su rendimiento, entonces su administración ejecutiva más activa debe apoyar el cambio. Crecimiento en las mejoras, sin apoyo y coordinación conduce a islas de excelencia en vez de una capacidad organizacional mejorada predicha (Mark Paulk, 1999).

En la mayoría de las organizaciones, un grupo de procesos de ingeniería de software (SEPG) o alguno equivalente debería formarse para coordinar la definición, mejora y repartición de actividades del proceso. Las organizaciones pequeñas pudieran no tener un SEPG de tiempo completo, pero la responsabilidad para el mejoramiento debería estar explícitamente asignada y supervisada.

Es necesario documentar los procesos. Las razones para documentar un proceso (o producto) son:

- Comunicar (a otros por ahora y quizás a uno mismo mas adelante)
- Entender (si no se puede poner escrito en realidad no se ha entendido)
- Fortalecer la consistencia (tomar ventaja de la repetibilidad)

Los procesos documentados apoyan al aprendizaje organizacional y previene la reinención de la rueda para problemas comunes. Los documentos no necesitan ser largos ni complejos para ser útiles.

El grado de formalidad requerida para los procesos es un reto frecuente tanto para las empresas pequeñas como para las grandes (Mark Paulk, 1999 citando a Comer, 1998).

La administración de proyectos de software en realidad es una administración de riesgos. En un sentido, el CMM trata de la administración de riesgos.

El factor # 1 en la definición y mejoramiento del proceso es “planeación” (Curtis, 1996 en Mark Paulk, 1999). La planeación es requerida para todo proceso de software importante, pero dentro de los límites de un juicio razonable, las organizaciones determinan qué es “importante” y como deberá desarrollarse el plan.

Abbott (Abbott, 1997 en Mark Paulk, 1999) identifica seis claves para el mejoramiento del proceso de software en pequeñas organizaciones:

- Soporte de los altos ejecutivos
- “staffing” adecuado
- Aplicación de los principios de administración de proyectos a mejoramiento de procesos
- Integración con ISO 9001
- Asistencia de consultores de mejoramiento de procesos
- Un enfoque orientado a dar valor a los proyectos y al negocio

Brodman and Johnson (Johnson, 1998 en Mark Paulk, 1999) identifican siete retos para los proyectos de la pequeña empresa:

- Manejo de requerimientos
- Generación de documentación
- Administración de proyectos
- Asignación de recursos
- Medición del progreso
- Revisiones dirigidas
- Proveer capacitación

Aceptación del CMM en el mundo

Tipos de organizaciones valoradas en el mundo en CMM

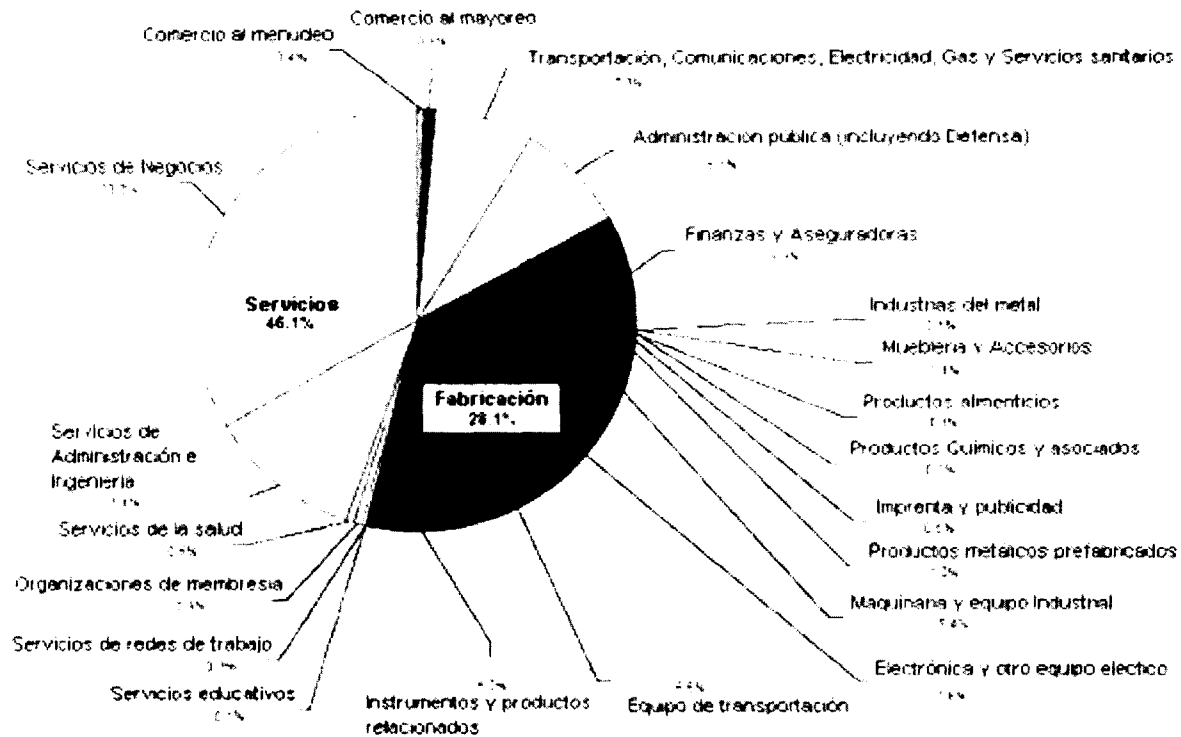


Fig. 2 - 8 Tipos de Organización a nivel mundial con valoraciones en CMM

Como se puede apreciar, el CMM ha sido adoptado por una gran variedad de giros, en donde destacan los de Servicios y de Fabricación. (SEI, 2003)

Tamaño de las organizaciones valoradas en el mundo en CMM y CMMi

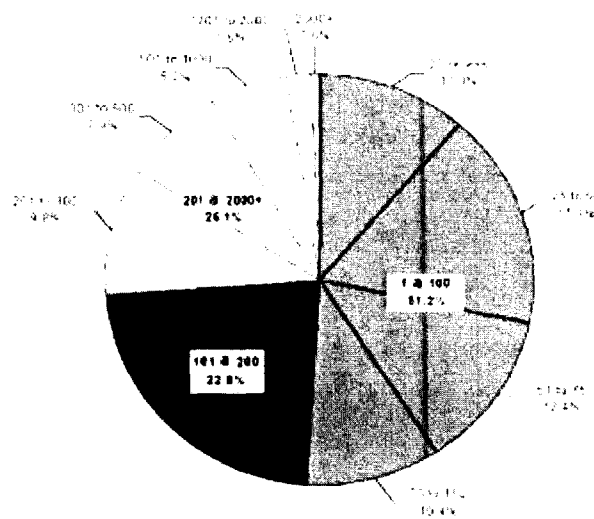


Fig. 2 - 9 Tamaño de las organizaciones a nivel mundial con valoraciones en CMM

Se puede apreciar por la gráfica anterior (SEI, 2003), que la mitad de las empresas valoradas en CMM tienen menos de 100 empleados, lo que desmitifica el hecho de que el CMM es nada más para empresas grandes.

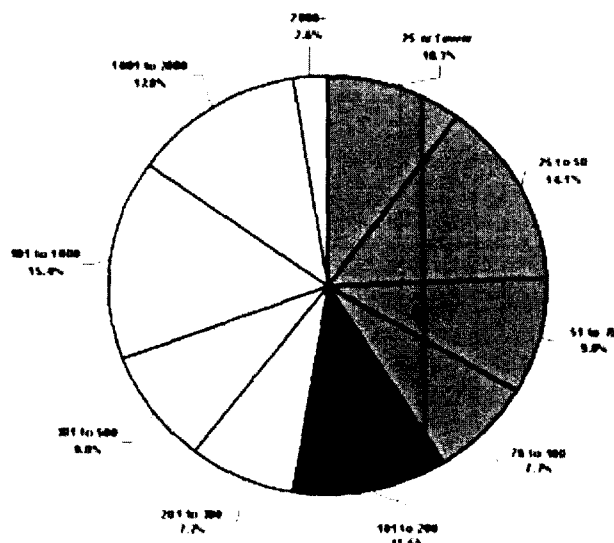


Fig. 2 - 10 Tamaño de las organizaciones a nivel mundial con valoraciones en CMMi

En la gráfica anterior (Phillips, 2004), aunque no llegan a la mitad, las empresas con menos de 100 empleados ocupan un gran porcentaje en las valoraciones del CMMi en el mundo.

Países donde las organizaciones han sido valoradas en CMM y CMMi



Argentina	Australia	Austria	Barbados	Belgium	Brazil	Canada	Chile
China	Colombia	Cyprus	Denmark	Egypt	Finland	France	Germany
Greece	Hong Kong	Hungary	India	Ireland	Israel	Italy	Japan
Korea, Republic of	Malaysia	Mexico	Netherlands	New Zealand	Norway	Pakistan	Peru
Philippines	Poland	Portugal	Puerto Rico	Russia	Saudi Arabia	Singapore	South Africa
Spain	Sweden	Switzerland	Taiwan	Thailand	Turkey	United Kingdom	United States
Uruguay	Venezuela	Vietnam					

Fig. 2 - 11 Países donde se han Realizado valoraciones de CMM

■ Países con empresas valoradas en CMM

La gráfica anterior (SEI, 2003), nuevamente nos da una idea de la gran aceptación mundial que tiene el CMM.



Fig. 2 - 12 Países donde se han Realizado valoraciones de CMMi

■ Países con empresas valoradas en CMMi

En la gráfica anterior se refleja que el CMMi (Phillips, 2004), aún no cuenta con aceptación en muchos países, tal es el caso de toda América Latina.

Cantidad de valoraciones CMM reportadas al SEI por año

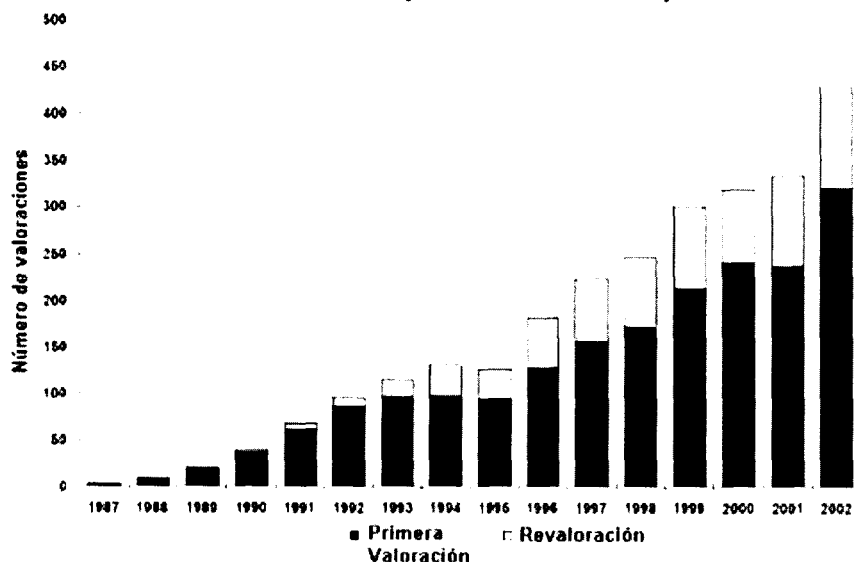


Fig. 2 - 13 Número de valoraciones reportadas por año al SEI

Como se puede apreciar en la gráfica anterior (SEI, 2003), hay una creciente demanda en la adopción del CMM.

Cambios en el nivel de Madurez

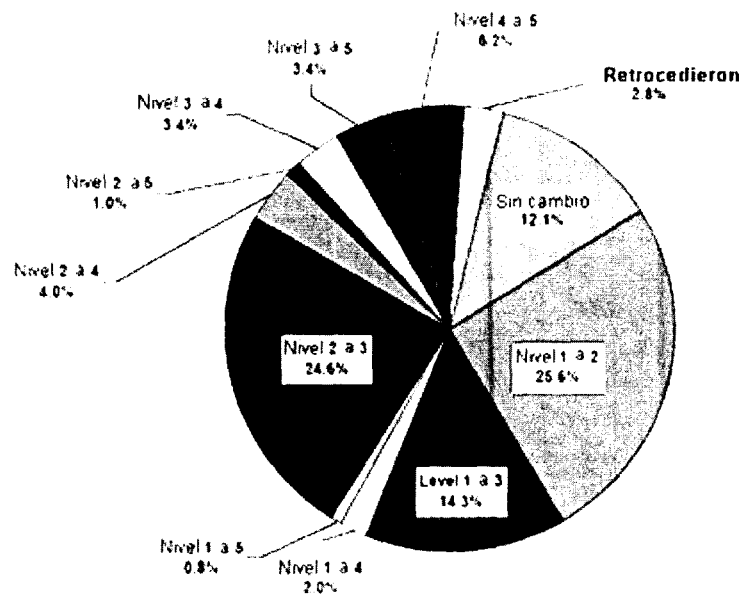


Fig. 2 - 14 Cambios de nivel de Madurez en las revaloraciones reportada al SEI a nivel mundial

Cómo es de esperarse, la gráfica anterior (SEI, 2003), nos demuestra que en la mayoría, los cambios se dan del nivel 1 al 2 o al 3, de ahí los que pasan del nivel 2 al 3. Son muy pocos los casos en donde saltan del nivel 1 al 4 o aún del nivel 1 al 5, se puede asumir que en estos últimos casos, son empresas que ya llevaban acabo un buen proceso de desarrollo de Software, con las prácticas de tales niveles, pero no se habían decidido a reportar su capacidad con el CMM.

CMM en México

Claudia García (2001) desarrolla una tesis relacionada con el modelo de capacidad de madurez y su aplicación en empresas mexicana de software.

Claudia García nos hace la reflexión de que si cada país otorga una escala que midiera el tamaño de sus organizaciones, lo mas seguro es que el rango definido como “grande” en un país de América Latina fuera identificado como “pequeño” en Estados Unidos. Así, hay que tener en cuenta que algunos modelos de madurez, como el CMM, fueron pensados para cubrir las necesidades de las organizaciones de gran tamaño, lo cual significa que esta norma no se aplicará al 100% en una organización latinoamericana grande.

El tamaño de una organización normalmente va de la mano con el número de personas que trabajan en ella. Si nuestra empresa es muy pequeña, lo mas seguro es que queramos ocupar al mayor número de personas para desarrollar software, por lo tanto también es probable que no quede lugar para formar ni un grupo pequeño dedicado exclusivamente para mejorar la calidad del proceso de software.

Para obtener mayor información, Claudia García elaboró un cuestionario que comprendía los siguientes puntos:

- ¿Qué organizaciones en México están en proceso de implantar o ya tienen implantado el CMM como modelo de mejoramiento de procesos de Software?
- ¿Cuánto tarda una empresa mexicana en poder alcanzar un mayor grado de madurez?
- ¿Qué ventajas que se obtienen tras utilizar CMM como modelo de mejoramiento de procesos?
- ¿Es más sencillo obtener primero una certificación ISO y después implantar CMM o primero trabajar con el CMM y después con otra norma de calidad?
- ¿Se notifica al SEI de los resultados obtenidos por las empresas que ya alcanzaron un nivel de madurez mayor al nivel 1?
- ¿Se permite desplegar los resultados obtenidos por las empresas de manera pública?

Menciona de algunos de los aspectos en los que se enfocaron las empresas que ya han logrado alguna certificación. Sin embargo, nos dice que uno de los mayores obstáculos para obtener información específica sobre CMM en México es la falta de conocimiento de cuales son las organizaciones interesadas, así como que las empresas son muy cuidadosas de otorgar información acerca de su estado actual ya que podría ayudar a sus competidores. Por esto, solo se publica el nivel de madurez omitiendo las áreas de mejora encontrada o fortalezas halladas.

Es interesante ver en esta tesis, también el tiempo que normalmente tardan las empresas en el mundo de cambiar de un nivel de madurez a otro:

Cambio de Nivel	Mundial (meses)
del Nivel 1 a Nivel 2	25 meses
del Nivel 2 a Nivel 3	23 meses
del Nivel 3 a Nivel 4	30 meses
del Nivel 4 a Nivel 5	22.5 meses
del Nivel 1 a Nivel 2	25 meses

Tabla 2 - 5 Tiempo que tardan las empresas en cambiar de un nivel de madurez a otro

Aunque en México no existe un organismo que concentre la información de los resultados obtenidos por empresas mexicanas tras haber realizado una valoración (Claudia García, 2001), se sabe que las empresas mexicanas valoradas en CMM al momento de realizar esta investigación son:

- | | |
|---|--|
| <ul style="list-style-type: none"> • Nivel 2 <ul style="list-style-type: none"> ○ Prodigia • Nivel 3 <ul style="list-style-type: none"> ○ Sigma Tao ○ Neoris | <ul style="list-style-type: none"> • Nivel 4 <ul style="list-style-type: none"> ○ <i>ninguna</i> • Nivel 5 <ul style="list-style-type: none"> ○ EDS ○ Softtek ○ AMS (Antes Tecnosys) |
|---|--|

Se puede ver que son muy pocas las empresas valoradas en CMM en nuestro país. También cabe mencionar que a la fecha no hay empresas en México valoradas en CMMi.

Conclusiones

El SPI debería hacerse para ayudar al negocio (no nada mas porque si). Esto es cierto para empresas grandes y pequeñas. La construcción de software es una actividad creativa de diseño intenso. Mientras que la disciplina del proceso es un habilitador crucial de éxito, el objetivo es resolver un problema, y esto requiere de creatividad. El proceso de Software debería ser repetible, aún cuando no sea repetitivo. El balance entre disciplina y creatividad puede ser un reto, según Glass (1995), citado por Mark Paulk (1999).

El CMM representa una ingeniería de “sentido común” que lleva al SPI. Sus niveles de maduración, KPAs, metas y prácticas claves han sido discutidas de manera exhaustiva dentro de la comunidad del software. Mientras que el CMM no es ni perfecto ni exhaustivo, representa un amplio consenso de la comunidad de software y es una herramienta útil para guiar los esfuerzos de mejora, y puede ser usado a ayudar a organizaciones pequeñas a mejorar sus procesos. (Abbott, 1997, Hadden, 1998, Hoffman, 1998, Pitterman, 1998, Sanders, 1998, citados por Mark Paulk (1999).

Las pequeñas organizaciones deberían considerar seriamente al PSP y al TSP (Ferguson, 1997 y Hayes, 1997, citados por (Mark Paulk, 1999).

En esta investigación usamos las 7 áreas de proceso del nivel 2 del CMMi (1er nivel para lograr la certificación) como base para la recopilación de intentos de mejora debido a que actualmente es la evolución del modelo mas reconocido en la industria del software en el ámbito internacional.

CAPITULO 3 - Metodología de Investigación

Introducción

En este capítulo, se describe el problema por el que surge esta investigación, el método utilizado en el desarrollo del proyecto de campo, con el que pretende encontrar el estado actual de las empresas mexicanas en esfuerzos de mejora del proceso de software.

En principio se hace una descripción de las características y objetivos principales de la investigación, así como de la metodología que se utilizó para la recolección de datos.

Posteriormente se presentan las preguntas y los factores a evaluar del cuestionario, así como una explicación del diseño del mismo.

Al final se describe la población, el cálculo del tamaño de la muestra y los métodos estadísticos usados para el análisis y evaluación de los resultados obtenidos en este estudio.

Descripción del problema de investigación

AMITI (2001) nos dice que la empresa mexicana que quiera competir en el mercado internacional debe ofrecer niveles de calidad bajo estándares internacionales, de los cuales se destacan el CMM y el ISO 15504 (SPICE). Actualmente existen 206 empresas desarrolladoras de Software en México, interesadas en exportar, de las cuales solo 5 se han evaluado en CMM.

Una valoración del proceso de software es requerida para identificar las áreas con más alta prioridad para mejoramiento en los procesos actuales y proveer recomendaciones de cómo hacer esos mejoramientos. El objetivo de la valoración debe ser la detección de los problemas, no proveer de soluciones inmediatas. (Humphrey, 1990).

La valoración puede ser llevada a cabo siguiendo un modelo de evaluación del proceso de Software como el CMM o SPICE. Estos modelos contienen preguntas detalladas para cada área de proceso de Software para evaluar y medir su nivel de maduración (Sakari Pihlava, 1996).

Sin embargo, Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber, (1993) nos hace ver que la capacidad de los procesos de software de las organizaciones que están en nivel 1 es impredecible porque los procesos de software están cambiando constantemente conforme se lleva a cabo el trabajo. Calendarios, presupuestos, funcionalidad y calidad del producto generalmente son impredecibles. El rendimiento depende de las capacidades de los individuos y varía dependiendo de sus habilidades natas, conocimiento y motivaciones. Hay pocos procesos de software estables en evidencia y el rendimiento solo puede ser predecible normalmente por individuos en lugar de la capacidad de la organización.

El éxito de las organizaciones que se encuentran en nivel 1 de maduración depende de los esfuerzos heroicos en la organización.

Nos dice Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber (1993) que es importante considerar que el CMM no es prescriptivo; no le dice a la organización como mejorar.

Los procesos permiten a la gente trabajar mas eficientemente incorporando las lecciones aprendidas en documentación, construyendo las habilidades que se necesitan para desempeñar esos procesos (comúnmente con capacitación) y mejorando continuamente por aprender de la gente que desempeña el trabajo. (Mark C. Paulk, Bill Curtis, Mary Beth Crisis y Charles V. Weber, 1993)

Como apuesta por un modelo a seguir por los desarrolladores nacionales se propuso en la reunión del círculo, el CMM (Anónimo, 1998 citando a Circulo de Calidad del Software, 1998)

Hoy debido a la falta de empresas certificadas, según Bancomext (2001), tenemos poca oferta, procesos de calidad e ingeniería casi inexistentes y la industria pulverizada en varias empresas en el país. Por lo que es necesario promover el desarrollo de compañías de excelencia que usen procesos reconocidos, que puedan crecer para lograr las economías de escala para competir, que desarrollen el mejor personal, que generen experiencia para la industria.

Según la Secretaría de economía (2001), la productividad de las empresas desarrolladoras de software es en general baja, debido a la falta de uso de procesos avanzados como procesos de calidad. Esto les impone una fuerte desventaja para competir frente a otros países.

Tres preguntas fundamentales que deben ser contestadas cuando se mejoran los procesos de software (Humphrey, 1990):

- ¿Qué tan bueno es mi proceso de Software actualmente?
- ¿Qué puedo hacer para mejorarlo?
- ¿Por donde comienzo?

Según Humphrey (1990), para el administrador de software quien llevará a cabo el programa de mejoramiento del proceso las dos preguntas más importantes son:

- ¿Cómo administro el programa de mejoramiento?
- ¿Cómo motivo a la organización para afrontar el reto?

Los puntos anteriores, así como la situación problemática hacen evidente que las empresas en México deben preocuparse por la obtención de al menos una de las certificaciones mencionadas para mejorar la calidad de sus procesos de desarrollo de Software y

aprovechar las facilidades financieras y de capacitación que el gobierno actual ofrece para cumplir uno de sus objetivos, desarrollar la industria del Software.

Objetivo de la investigación

La razón de ser de esta investigación, como se mencionó en el capítulo 1, es dar a conocer los factores que llevaron al éxito o que llevaron al fracaso de los intentos de mejora del proceso de software en nuestro país, así como, si no han habido intentos de mejora saber porque no. En base a lo anterior, proponer una lista de recomendaciones.

En resumen:

- Obtener los factores más importantes que llevaron al éxito o que llevaron al fracaso de los intentos de mejora del proceso de software en nuestro país.
- Proponer listado de recomendaciones a las empresas para incrementar las posibilidades de éxito de los proyectos de mejora al proceso de software utilizando como base los factores obtenidos en el punto anterior.

Factores de estudio de la investigación

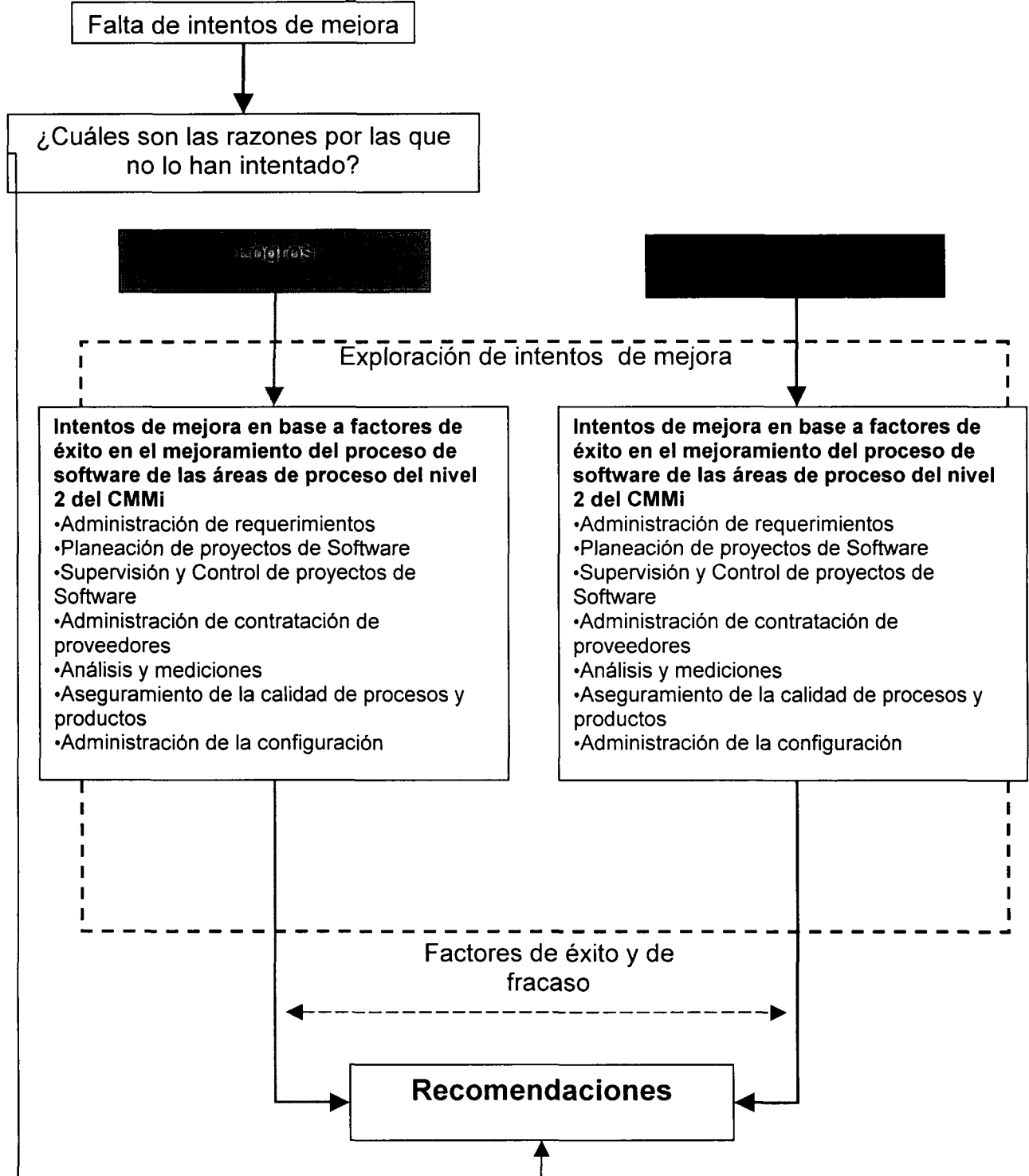


Fig. 2 - 15 Factores de estudio de la investigación

En el diagrama anterior podemos observar al inicio tres grandes grupos, en el primer grupo se identifican los factores mas relevantes por los que no se ha intentado mejorar el proceso de desarrollo de software en las empresas mexicanas, el segundo bloque identifica los factores mas relevantes que han llevado a obtener resultados deseables en dichas empresas y por ultimo se identifican también los factores que han obstaculizado la obtención de resultados favorables en este proceso.

Con la determinación de los factores de éxito se propondrá un listado de recomendaciones para aumentar las probabilidades de éxito en los esfuerzos por mejorar el proceso de desarrollo de Software. Así mismo se darán a conocer los factores de fracaso y las razones que han mencionado las empresas las han impedido intentar mejorar el proceso.

Metodología de la investigación de campo

En base a los objetivos de esta investigación y tomando en cuenta, como respaldo el libro de Metodología de la Investigación de Hernández, Fernández y Baptista (2002) que describen cuatro tipos de investigación, este estudio requiere una investigación de tipo exploratoria, utilizando enfoques tanto cuantitativos como cualitativos, ya que este tipo de estudio se realiza, normalmente, cuando el tema o el objetivo a examinar ha sido poco estudiado. Este tipo de estudio nos llevará a familiarizarnos más con los puntos desconocidos en esta investigación.

Recolección de datos

Se hará uso de una encuesta para medir cada una de las variables antes mencionadas, la cual se llevará acabo mediante un cuestionario que apoye el análisis para determinar en cada caso, ¿En dónde están? ¿A dónde quisieran estar? En intentos de mejora, ¿Qué funcionó? ¿Qué no funcionó? ¿Qué áreas de la empresa están directa o indirectamente involucradas?

Diseño de la encuesta

El diseño de la encuesta contiene, en primer lugar, datos generales de las empresas y, en segundo lugar, pero mas importante, la parte central de la encuesta, que son los factores críticos de éxito y de fracaso hallados en la literatura y en otras investigaciones para cada área de proceso del nivel dos del CMMi y al final un apéndice con una breve descripción del propósito de cada área de proceso de dicho modelo de madurez.

Sección de Generalidades de la Encuesta

En esta sección se incluyen aspectos generales como:

- El puesto actual del entrevistado
- El giro de la empresa

- El número empleados de la empresa
- El número de empleados que se dedican específicamente al desarrollo de software
- La cantidad de años que tiene la empresa desarrollando software
- Si la empresa ha recibido algún tipo de capacitación relacionada con el mejoramiento del proceso de desarrollo de software o no
- Si la empresa ha participado en algún tipo de valoración de su proceso de desarrollo de software, evaluación de capacidades u otro tipo de examinación de dicho proceso de desarrollo de software (por ejemplo, CMM, ISO, etc.) o no

Sección intentos de mejora, de factores de éxito y de fracaso en estos intentos

El objetivo de esta sección es que el encuestado de a conocer en que áreas de proceso ha intentado mejorar y en cuales no, en las que ha intentado mejorar, que de a conocer también si ha sido de forma aislada o a nivel organizacional, marcando el espacio correspondiente para cada una de estas áreas.

A continuación se muestra la parte de la sección correspondiente a la descrita:

Área de proceso	Aún no	Sí, pero en esfuerzos aislados	Sí, a nivel organizacional
Administración de requerimientos			
Planeación de proyectos de Software			
Supervisión y Control de proyectos de Software			
Administración de contratación de proveedores			
Análisis y mediciones			
Aseguramiento de la calidad de procesos y productos			
Administración de la configuración			

Después, en esta misma sección se hace una búsqueda de las razones por qué no se ha intentado realizar una mejora, en caso de haber algún área que no se haya hecho el intento, mostrando una lista de factores posibles y pidiendo una ponderación del 1 al 5 para cada factor, siendo 1 un total desacuerdo y 5 un total acuerdo.

Se puede ver el listado de los factores de la parte de la encuesta correspondiente a lo antes descrito tal como apareció en la misma:

- No existe un entendimiento claro por parte de los altos ejecutivos de la empresa
- Los beneficios reales no se ven
- No se percibe como una necesidad
- Se percibe como muy costoso
- Se considera que no aplica a mi organización

- Tenemos mucho trabajo
- No es parte de la estrategia de la empresa
- Nuestros usuarios son demasiado exigentes
- Nadie ha tomado ese esfuerzo como suyo
- Eso es demasiado técnico para las actividades de nuestra empresa
- Nuestros entregables están reducidos en tiempo de entrega
- Otro:

Adelante, en esta misma sección se hace una búsqueda de las razones por qué no se ha intentado realizar una mejora a nivel organizacional, en caso de haber algún área donde los intentos se hayan efectuado de forma aislada, mostrando también una lista de factores posibles y pidiendo de igual forma una ponderación del 1 al 5 para cada factor, siendo 1 un total desacuerdo y 5 un total acuerdo.

Se puede ver el listado de los factores de la parte de la encuesta correspondiente a lo antes descrito tal como apareció en la misma (se puede apreciar que es el mismo listado que el anterior):

- No existe un entendimiento claro por parte de los altos ejecutivos de la empresa
- Los beneficios reales no se ven
- No se percibe como una necesidad
- Se percibe como muy costoso
- Se considera que no aplica a mi organización
- Tenemos mucho trabajo
- No es parte de la estrategia de la empresa
- Nuestros usuarios son demasiado exigentes
- Nadie ha tomado ese esfuerzo como suyo
- Eso es demasiado técnico para las actividades de nuestra empresa
- Nuestros entregables están reducidos en tiempo de entrega
- Otro:

Continuando en la misma sección se pregunta si los esfuerzos realizados de forma organizacional han sido motivados por la búsqueda de una certificación internacional o no.

Después, dentro de la misma sección, se abren dos incisos (A y B) para las áreas de proceso que hayan intentado mejorarse tanto de forma aislada como a nivel organizacional, apareciendo una lista de factores posibles y pidiendo la ponderación del 1 al 5 para cada factor, siendo 1 un total desacuerdo y 5 un total acuerdo.

El inciso A pregunta básicamente, en caso de que no se hayan logrado los objetivos al intentar mejorar en las áreas seleccionadas, por qué no se lograron.

A continuación se puede ver el listado de los factores de la parte de la encuesta correspondiente al inciso A del párrafo antes descrito tal como apareció en la misma:

- No se ha puesto al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa
- No se obtuvo el compromiso ni el soporte de los altos ejecutivos de la dirección
- No se percibió un cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)
- No hubo disposición de tiempo, recursos, falta de involucramiento, comunicación y entusiasmo por parte del equipo encargado
- No hubo una asignación clara de tareas, responsabilidades ni toma de decisiones
- No se estimó correctamente el nivel actual de la empresa
- No se consideraron las habilidades de las personas involucradas
- No hubo mejora percibida en la calidad de los intentos realizados
- No se definieron metas realistas o claras, ni se establecieron los objetivos relevantes
- La organización no se centró en los beneficios a largo plazo, balanceados por metas a corto plazo
- No se obtuvo una alineación entre la alta gerencia y la administración media o no se incorporó este proceso a la estructura organizacional
- No hubo capacitación a empleados ni se involucraron a los profesionistas recién egresados
- No hubo actividades complementarias tales como gestión de recursos financieros o gestión de personal.
- Inexistencia de mecanismos para captar información y experiencias útiles para el proceso de mejora del desarrollo de software
- Desconocimiento de los cambios en los gustos y necesidades de los clientes
- No se establecieron relaciones de colaboración con los competidores, proveedores u otros agentes (como instituciones públicas)
- Mal manejo de tiempos y actividades; de asignación de personal y recursos; del presupuesto
- Mala selección del personal
- Falta de desarrollo de reportes y explotación de información
- Estamos esperando a que la industria decida un modelo
- Otro:

El inciso B pregunta, en caso de que sí se obtuvieron logros significativos al intentar mejorar en las áreas seleccionadas, cuales son las actividades se consideran que los han llevado al éxito.

A continuación se puede ver el listado de los factores de la parte de la encuesta correspondiente al inciso B del párrafo antes descrito tal como apareció en la misma:

- Se puso al proceso de mejora del desarrollo de software en una perspectiva adecuada
- Se obtuvo el compromiso y soporte de los altos ejecutivos de la dirección
- Se demostró una disposición de tiempo, recursos, involucramiento, comunicación y entusiasmo por parte del equipo encargado
- Cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)

- Hubo una asignación clara de tareas, responsabilidades y toma de decisiones
- Se estimó correctamente el nivel actual de la empresa
- Se consideraron adecuadamente las habilidades de las personas involucradas
- Se reforzó el entendimiento de los procesos para la mejora
- Se definieron metas realistas o claras, y se establecieron los objetivos relevantes
- Hubo estabilización en los procesos de cambio
- La organización se centró en los beneficios a largo plazo, balanceados por metas a corto plazo
- Se obtuvo una alineación entre la alta gerencia y la administración media y/o se incorporó este proceso a la estructura organizacional
- Se motivó a las personas para adoptar el proceso de mejora del desarrollo de software
- Hubo capacitación a empleados y se involucraron a los profesionistas recién egresados
- Hubieron actividades complementarias tales como gestión de recursos financieros o gestión de personal
- Existencia de mecanismos para captar información y experiencias útiles para el proceso de mejora del desarrollo de software
- Conocimiento de los cambios en los gustos y necesidades de los clientes
- Se establecieron relaciones de colaboración con los competidores, proveedores u otros agentes (como instituciones públicas)
- Buen manejo de tiempos y actividades; asignación de personal y recursos; del presupuesto
- Buena selección del personal
- Desarrollo de reportes y explotación de información
- El cumplimiento de los objetivos del proyecto de mejora se vinculó a la evaluación del desempeño del personal involucrado
- Otro:

Para finalizar se pregunta si existen comentarios adicionales que pudieran aportar sobre sus intentos que hayan funcionado o fracasado en el mejoramiento del proceso de desarrollo de software.

Pruebas por panel de expertos

Una vez terminada la primera versión de la encuesta, se lleva a cabo una revisión por un panel de expertos, relacionados con el modelo de calidad CMM y la Ingeniería de Software, los cuales dieron a conocer sus comentarios y aportaciones muy valiosas.

Del diseño y la validación de la encuesta surge la encuesta final (Apéndice A) que se utilizó como instrumento en la investigación de campo.

Validación del contenido de la encuesta

Es necesario tomar en cuenta lo que Noelle-Neumann, citado por Hinojosa (2002), señala, que hay que hacer énfasis en el diseño de los cuestionarios, resaltando los siguientes aspectos que pudieran presentarse:

- Existen dificultades lingüísticas de entendimiento, los conceptos utilizados o el tema puede no decir nada a los encuestados, son entendidos equivocadamente e incluso interpretados de modo diverso.
- En ocasiones los temas a investigar son demasiado abstractos de modo que la inteligencia de muchos entrevistados no alcanza a entenderlo.
- Obtenemos respuestas más abundantes y certeras cuando las preguntas son concretas.
- La capacidad lingüística de los entrevistados no es suficiente, o es dificultada por la reserva o timidez para explicar inmediata y acertadamente la opinión, experiencia y comportamiento acerca de lo que se le pregunta.
- La realidad de la pregunta no es percibida o lo es de modo impreciso.
- La memoria es débil para dar información fiel.
- La fatiga puede eliminar la aptitud o la disposición de las afirmaciones.
- La desconfianza de tipo político, social o religioso pudiera impedir que los entrevistados hablasen.
- El afán de prestigio pudiera falsear respuestas.
- Las preguntas pudieran referirse a cuestiones personales o privadas sobre las que no se desea comunicar nada a una persona ajena.
- Racionalizaciones e ilusiones de todo tipo desfiguran la respuesta.
- Se puede investigar realidades, modos de comportamiento o actitudes inconscientes de los entrevistados, así pues, aun con la mejor voluntad, al ser consultados directamente no pueden dar información.

Acerca de la encuesta final

Esta, como ya se mencionó, se divide en dos secciones que cubren los aspectos de la información deseada. La primera sección busca recabar datos demográficos generales de las empresas, como son: el giro de la empresa, número de empleados, número de empleados que se dedican específicamente al desarrollo de software, etc. Esto con el fin de presentar un perfil básico de las empresas participantes en esta encuesta.

La segunda sección analiza los logros o falta de ellos en base a diversos factores críticos de éxito y de fracaso de la Ingeniería de Software mencionados en el capítulo 2 para cada área de proceso del nivel dos del CMMi.

Límites de la investigación

- Se realizó una investigación de campo en empresas ubicadas en el territorio nacional, determinadas en la muestra.
- La encuesta (en su formato electrónico para el llenado en Internet y en su formato impreso para su llenado presencial) será el único medio de recolección de información, a través de cuestionarios impresos o electrónicos a través de Internet, esta encuesta se localiza en el Anexo 1.
- La información dependerá en gran medida de la disponibilidad, veracidad y conocimiento que se tenga por parte de las empresas para contestar las encuestas.

Selección de la población y muestra

Se recaudó la información para esta investigación de empresas mexicanas que desarrollan software, sea o no su giro principal. Estas empresas fueron seleccionadas del directorio de empresas de la página Web de Ex-Alumnos del ITESM campus Monterrey, del directorio de participantes de la AMITI, de la AETI y algunas de la sección amarilla. Para proteger la confidencialidad de las empresas que forman parte de esta muestra, la encuesta no recopiló los nombres de las mismas.

Dado que se trata de una muestra no probabilística, se considera un tamaño finito para el universo y el tamaño de dicha muestra mediante la formula de universos finitos (Marcos, 2001).

Tomando en cuenta los siguientes datos:

$$\begin{aligned} Z &= \text{nivel de confiabilidad} = 0.99 \\ P &= \text{probabilidad de éxito} = 0.947 \\ e &= \text{error esperado} = 0.0275 \end{aligned}$$

se aplica la formula:

$$n = Z^2 * P(1 - P) / e^2$$

Obteniendo como resultado, $n = 65$

Por lo tanto, el número mínimo de encuestas a recolectar para la validez de esta investigación bajo los criterios que se definen en la formula anterior es aproximadamente de 65. Mediante la encuesta se obtuvo información relevante de 65 empresas de diversos sectores de la República Mexicana. En formato electrónico y en formato presencial.

Aplicación de la encuesta

El proceso de aplicación de la encuesta se llevó a cabo de Agosto del 2003 a Mayo del 2004 en todo el territorio nacional (a través de Internet en las empresas fuera de Nuevo León).

El procedimiento para recabar la información inició vía telefónica o vía correo electrónico, con el cual se solicitó comunicarse con alguna persona enterada del proceso de mejora de desarrollo de Software dentro de la organización, inmediatamente el entrevistador se identificó como alumno del ITESM explicando el objetivo de la investigación, acto seguido se acordaba el llenado de la encuesta, de forma presencial o de forma virtual a través de Internet.

Conclusiones

Se entrevistó en total a 65 empresas en todo el país, de las cuales, el 48% son de Software y Consultoría, el 9% de Manufactura, un 6% de Acero y Papel, otro 6% de Educación y el 31% de diversos giros mas, tales como: Alimentos y bebidas, Almacenaje y Distribución, Bancos y finanzas, Minería, entre otros. Las personas entrevistadas son 29% líderes de proyectos, 28% gerentes generales, 15% directores de sistemas, y el 28% restante otros puestos. Con este número de empresas se cumple con el mínimo descrito en las secciones anteriores. Como se puede ver, el 72% de los entrevistados ocupan puestos importantes dentro de las empresas, por lo que las respuestas aportadas son confiables.

Doce de las 65 entrevistas fueron presenciales, realizadas a empresas ubicadas en Nuevo León, el resto de las entrevistas fueron recabadas vía Internet.

CAPITULO 4 – Análisis de Resultados

Introducción

A continuación se dan a conocer los resultados de la investigación, mismos que se obtienen de haber utilizando técnicas y herramientas estadísticas, se analiza la información obtenida por las encuestas, mostrando los resultados de la investigación de campo, también se presenta un análisis de los resultados que generó el producto final de la tesis.

Demografía

Giro de la empresa

En la siguiente tabla se dan a conocer los giros de las empresas que formaron parte de la encuesta.

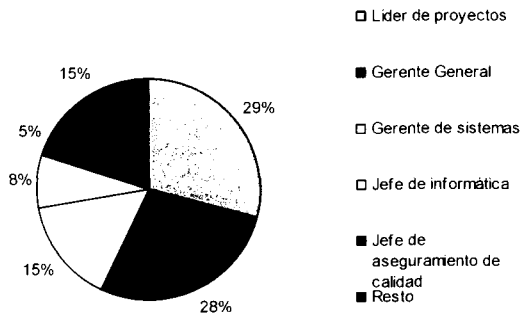
Giro	Número de empresas	%
Software y consultoría	31	47.7 %
Manufactura	6	9.2 %
Acero y papel	4	6.2 %
Educación	4	6.2 %
Alimentos y bebidas	2	3.1 %
Seguros y bienes raíces	2	3.1 %
Telecomunicaciones	2	3.1 %
Gobierno	2	3.1 %
Almacenaje y distribución	1	1.5 %
Bancos, finanzas	1	1.5 %
Minería, petróleo y gas	1	1.5 %
Construcción	1	1.5 %
Otros	8	12.3 %

Los otros giros: Administración de proyectos; Aeronáutica; Estudios de Mercado; Infraestructura computacional a nivel de almacenamiento; Medio masivo de comunicación.

Cómo se puede apreciar, la mayoría de las empresas forman parte de la Industria del Software.

Puesto de los entrevistados

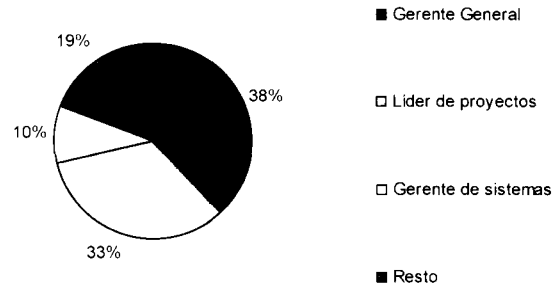
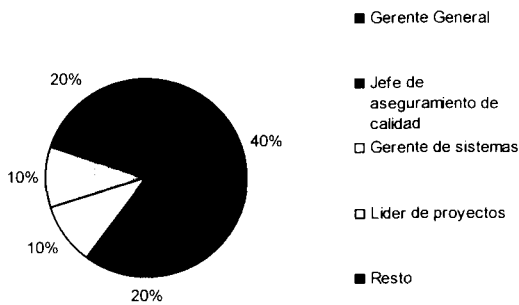
En las siguientes gráficas se muestra la proporción de las personas que contestaron los cuestionarios, el primer grupo nos muestra la proporción global de las empresas entrevistadas, en el siguiente únicamente de las empresas cuyo giro principal es el Software (separando las grandes y las chicas) y el siguiente todas las empresas excluyendo a las que se dedican al Software.



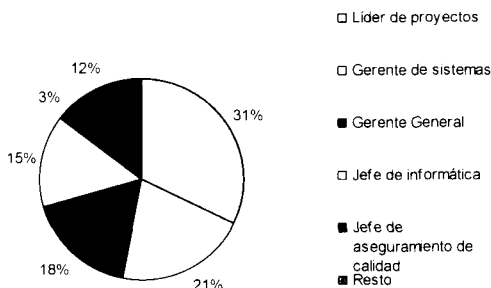
El resto de los puestos:

- Catedrático
- Consultor
- Gerente Comercial
- Gerente de investigación de redes y telecomunicaciones
- Gerente de operaciones
- Operaciones e Ingeniería de Software.

Todas las empresas (n=65)



Empresas de Software grandes (n=10)



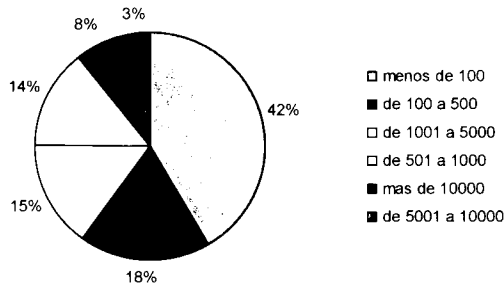
Empresas de Software chicas (n=21)

Como se puede apreciar, en la mayoría de las empresas de la industria del Software, los entrevistados son Gerentes Generales; mientras que en el resto de los giros, los que contestaron son Líderes de Proyecto. Esto nos da una idea del juicio de dichas personas al contestar el cuestionario. Eso sustenta un juicio de expertos por el puesto que ocupan.

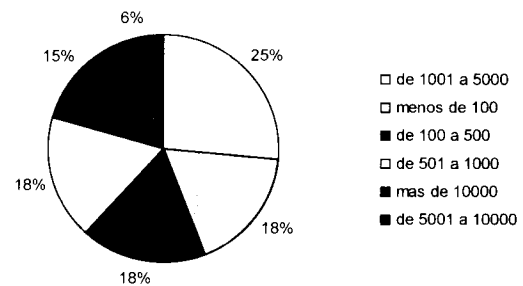
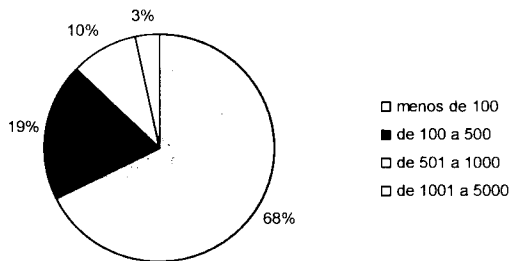
Otros giros (n=34)

Número total de empleados en la empresa

Las siguientes gráficas nos dan una idea del tamaño de las empresas por el número de empleados.



Todas las empresas (n=65)



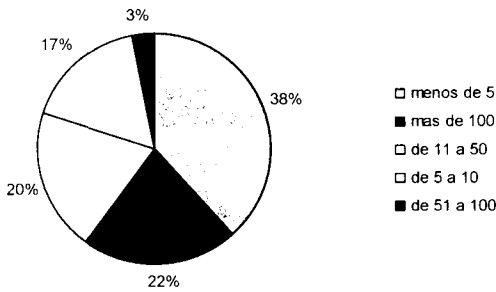
Empresas de Software (n=31)

Otros giros (n=34)

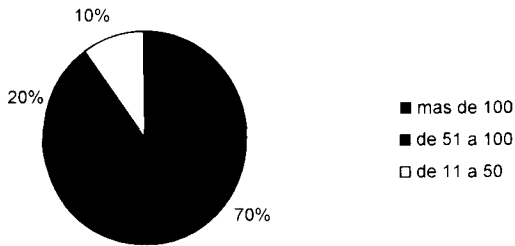
Se nota una marcada diferencia entre las empresas de Software y Consultoría y los demás giros. En el primer caso, la gran mayoría tienen menos de 100 empleados, mientras que en el segundo caso, la mayoría está representada por las empresas de entre 1001 y 5000 empleados. Esto nos denota una industria del Software pulverizada en México. Las empresas que conforman la industria del Software en nuestro país son pequeñas (el 87% de estas tienen menos de 500 empleados). En las siguientes gráficas también se hace una separación entre empresas de Software grandes (con 100 empleados o más) y empresas de Software chicas (menos de 100 empleados).

Número de empleados que se dedican específicamente al desarrollo de Software

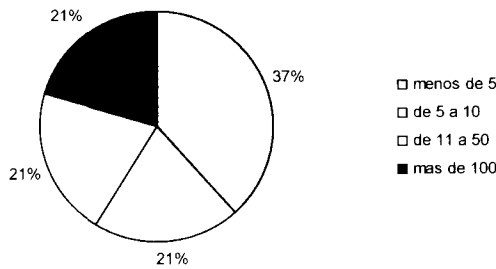
A continuación se muestran los porcentajes del tamaño de las empresas, en cuanto al número de empleados, que aportaron su información.



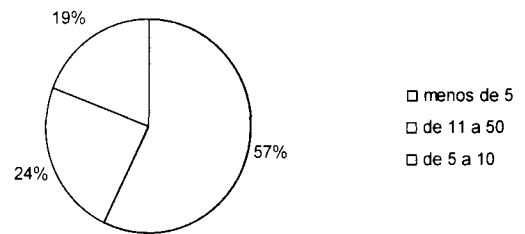
Todas las empresas (n=65)



Empresas de Software grandes (n=10)



Otros giros (n=34)

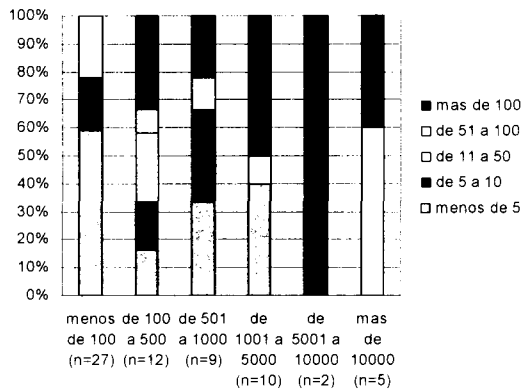


Empresas de Software chicas (n=21)

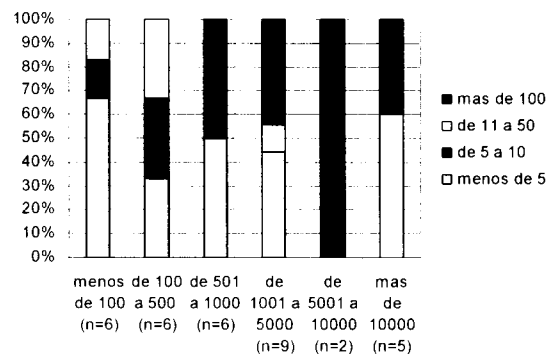
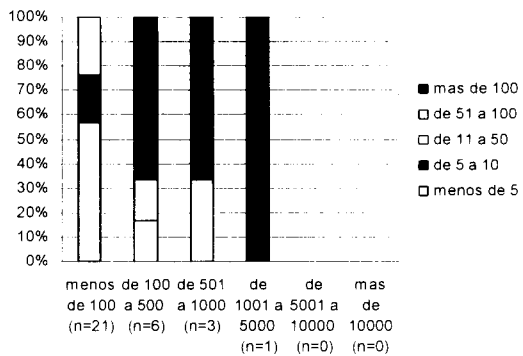
Podemos notar una marcada y esperada diferencia entre las empresas de Software grandes y chicas, las gráficas nos muestran la correlación de que entre mas grande es la empresa de Software, tienen mas empleados que se dedican a desarrollar Software.

Proporción Número total de empleados/Número de empleados que se dedican específicamente al desarrollo de Software

En base a las respuestas recibidas respecto al tamaño de las empresas, por Número total de empleados y Número de empleados que se dedican a desarrollar software se obtuvieron las siguientes proporciones.



Todas las empresas (n=65)



Empresas de Software grandes (n=31)

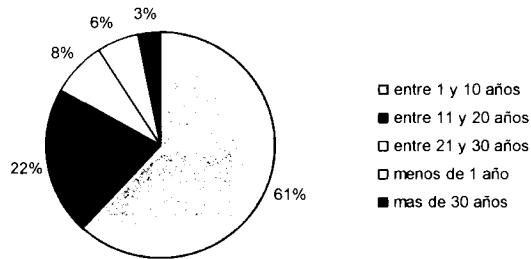
Otros giros (n=34)

En la mayoría de las empresas grandes, que no pertenecen a la industria del Software), tienen muy poca gente desarrollando Software, esto es, optan por el "Outsourcing".

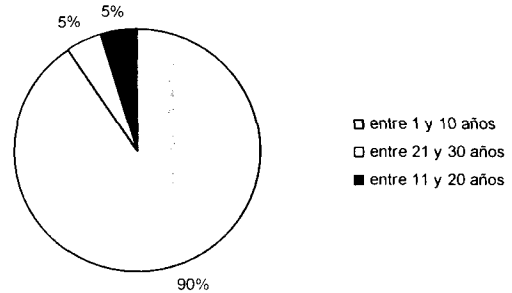
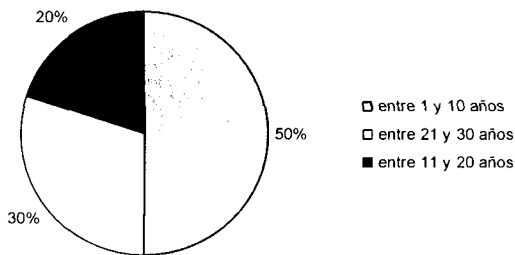
En las empresas que forman parte de la industria del Software, dado que sus desarrolladores de Software son menos de 50, se puede concluir que la mayoría de ellas deben ser de 50 empleados o menos. La gran mayoría de los empleados de estas empresas, se dedican a desarrollar Software. En estas empresas hay pocas situaciones que pudieran contradecir lo anterior, como lo muestra la gráfica para estas empresas, donde tienen entre 501 y 1000 empleados, en estos casos, las empresas sí pertenecen a la industria pero no desarrollan Software, mas bien se dedican a la implementación, consultoría, venta y a proporcionar servicios varios.

Experiencia en el Desarrollo de Software en las empresas participantes

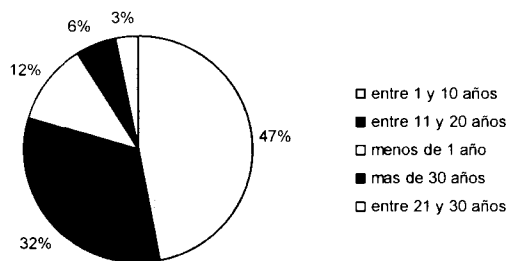
A continuación se ve la experiencia en cuanto al número de años que tienen las empresas desarrollando Software.



Todas las empresas (n=65)



Empresas de Software grandes (n=10)



Empresas de Software chicas (n=21)

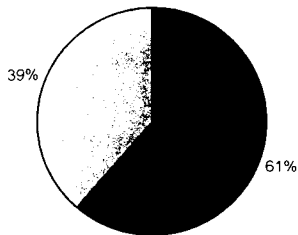
Como se puede apreciar, del total de las empresas, la gran mayoría tienen entre 1 y 10 años de experiencia. Sólo una pequeña porción tienen más de 30 años de experiencia.

Se puede apreciar que la industria del Software es muy joven (la gran mayoría tienen entre 1 y 10 años), esto es mucho más notorio en las empresas chicas de la misma industria.

Otros giros (n=34)

Capacitación relacionada con el mejoramiento del Proceso de Desarrollo de Software

Se les preguntó también si habían recibido algún tipo de capacitación relacionada con el del proceso de mejora de Desarrollo de Software.

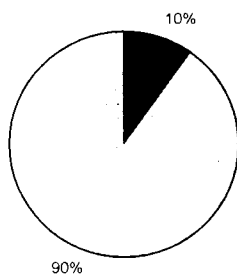


Todas las empresas (n=65)

Capacitaciones recibidas*:

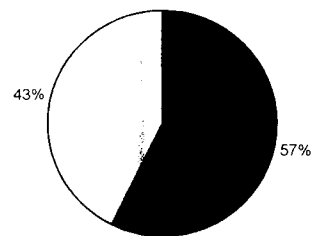
- CMM
- CMMi
- Extreme programming
- Integración RAD-IDS
- ISO 12207
- MSF
- MSM
- PSP/TSP
- Rational
- RUP
- SCT
- Six sigma
- Software Management
- SW Quality Assurance
- Test Process

* Algunas no son metodologías de mejora, pero decidieron dejarse porque son metodologías de desarrollo de software



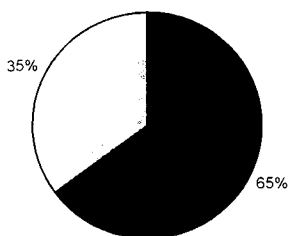
Empresas de Software grandes (n=10)

■ No
□ Si



Empresas de Software chicas (n=21)

El porcentaje de las empresas que no han recibido capacitación en el proceso de mejora es muy grande. Sólo las empresas de Software grandes presentan un marcado interés en la esta categoría. Es importante recalcar que si no hay capacitación no puede haber mejora.



Otros giros (n=34)

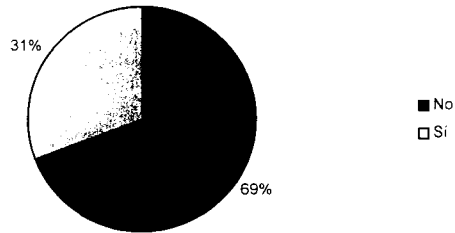
■ No
□ Si

Valoraciones en el Proceso de Desarrollo de Software

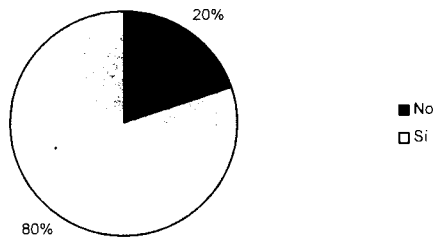
Se cuestionó si la empresa ha participado en algún tipo de valoración de su Proceso de Desarrollo de Software, evaluación de capacidades u otro tipo de examinación.

Valoraciones efectuadas:

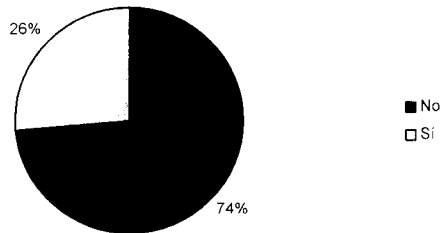
- CMM
- ISO 9000
- MoProSoft



Todas las empresas (n=65)



Empresas de Software grandes (n=10)



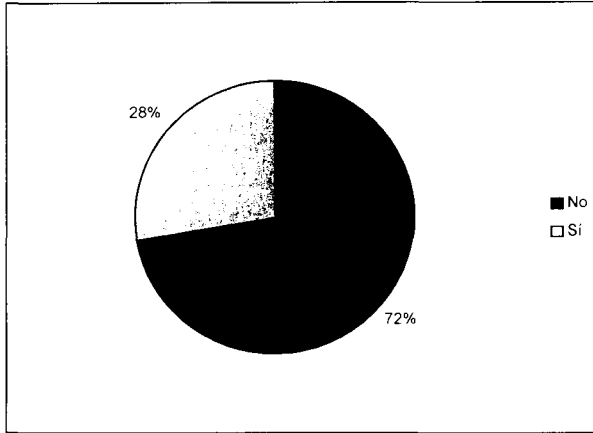
Empresas de Software chicas (n=21)

Se observa que una gran mayoría de las empresas No ha sido valorada en su proceso de Desarrollo de Software. Notablemente, esto es cierto también para las empresas pequeñas de la industria del Software, lo cual resulta preocupante. Sólo las empresas de Software grandes son las que denotan interés en valorar su proceso.

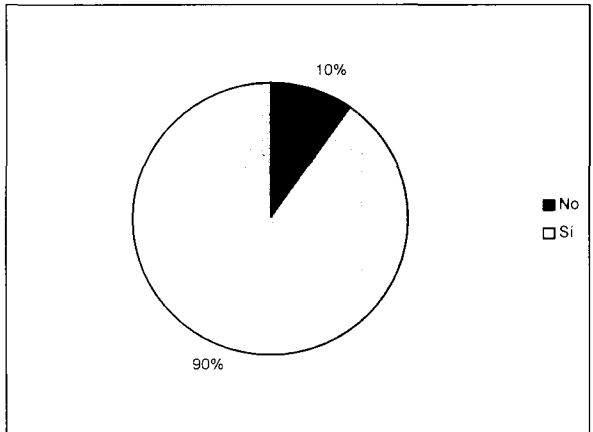
Otros giros (n=34)

Búsqueda de certificación a través del mejoramiento del proceso de Desarrollo de Software.

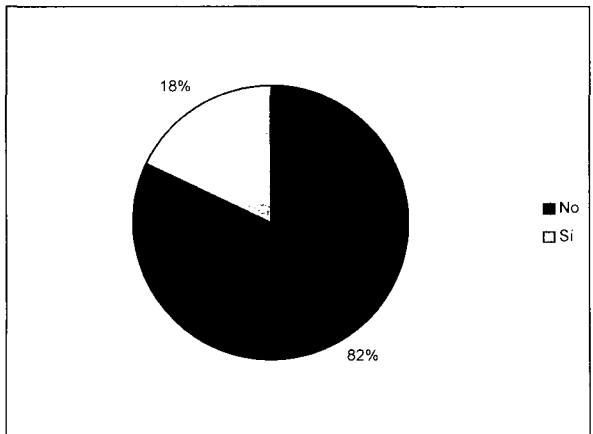
Se recaudó información acerca de si los esfuerzos realizados a nivel organizacional, han sido buscando una certificación en Ingeniería de Software.



Todas las empresas (n=65)



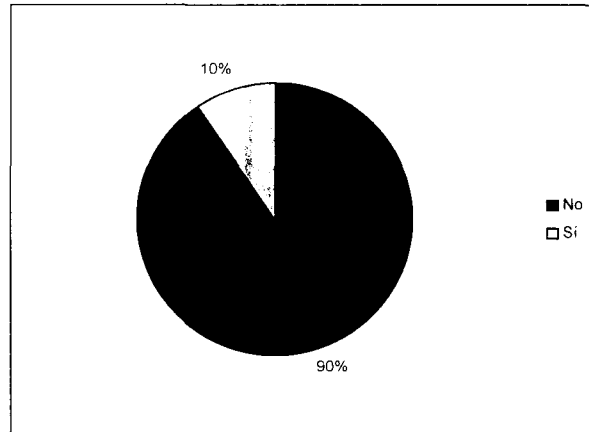
Empresas de Software grandes (n=10)



Otros giros (n=34)

Certificaciones buscadas:

- CMM
- MoProSoft
- Six sigma

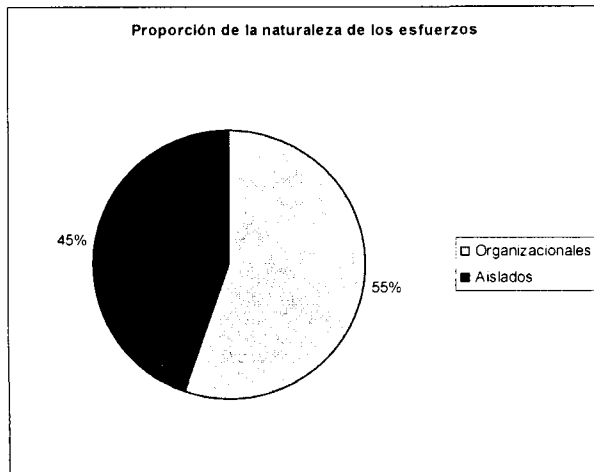


Empresas de Software chicas (n=21)

La gran mayoría de las empresas no están interesadas en una certificación, con excepción de las empresas de Software grandes que sí demuestran un notable interés en certificarse. Las demás empresas de la industria, deberían preocuparse mucho más por encaminar sus esfuerzos hacia una certificación si están interesadas en competir de forma global.

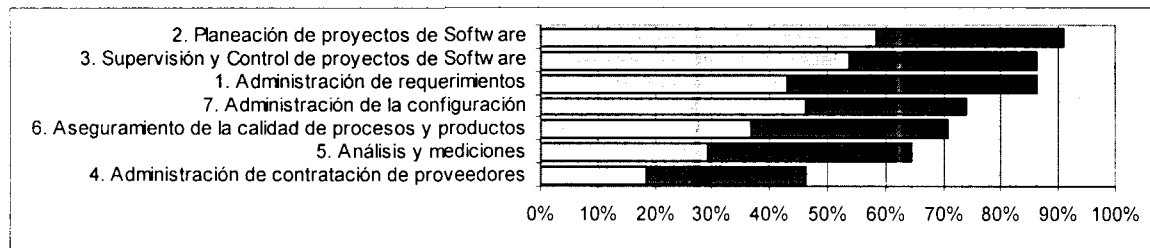
Se puede apreciar una fuerte correlación entre la participación de las empresas en valoraciones del proceso de Desarrollo de Software y su intención por lograr una certificación.

Intentos de Mejora o falta de ellos



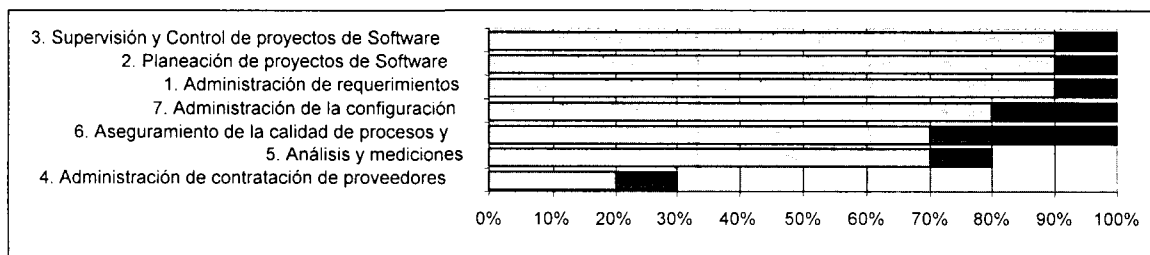
En estas gráficas se puede apreciar los intentos de mejora en base a las 7 áreas de proceso (PA) del nivel 2 del CMMi, mismas que se tomaron como base en esta investigación. La primer área de cada barra representan los intentos a nivel organizacional, la siguiente los intentos aislados y lo que falta para completar el 100% representa las faltas de intentos de mejora.

Por PA

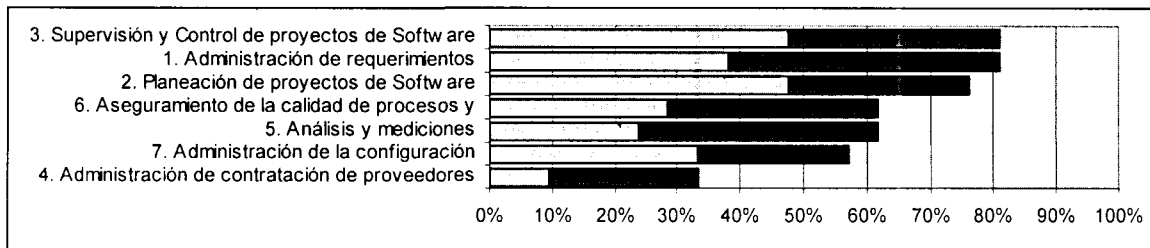


Todas las empresas (n=65)

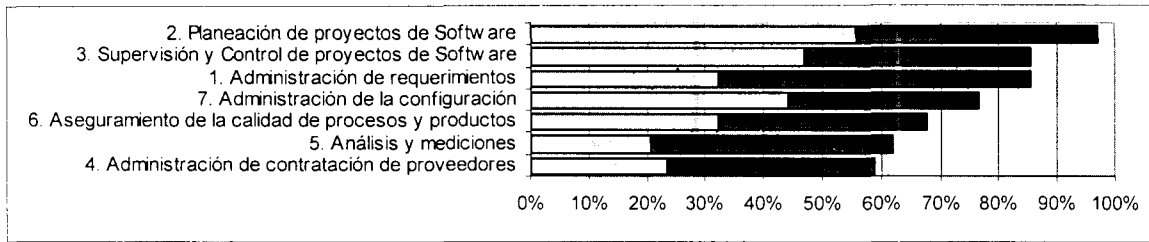
PA-Giro



Empresas de Software grandes (n=10)



Empresas de Software chicas (n=21)

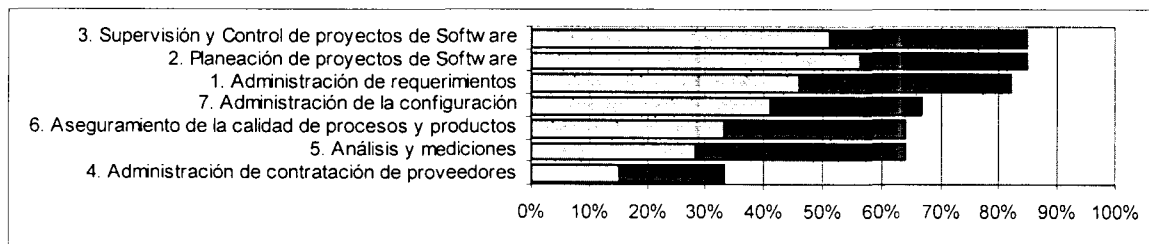


Otros giros (n=34)

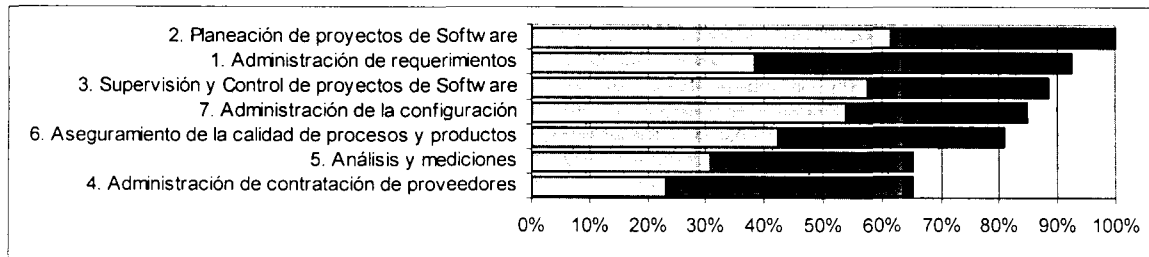
Se puede apreciar que las empresas que no se dedican a desarrollar software muestran mayor cantidad de intentos de mejora que las empresas pequeñas de la industria del Software. Las primeras tres áreas de proceso con más intentos, incluyendo a nivel organizacional son comunes para las empresas pequeñas de Software y las de otras industrias. En estos dos casos, existe una fuerte preocupación por mejorar el control de requerimientos, seguimiento y planeación del proceso. Una razón para que las empresas que no se dedican específicamente a desarrollar Software hayan intentado mejorar mas que las empresas pequeñas de la industria del Software es el uso que le dan al Software que desarrollan, unos para su operación diaria, y otras para venderlo, lo que denota en las empresas pequeñas de Software que no hay una cultura de desarrollo sustentable, es decir, viven al día. Mientras que las empresas grandes de Software se preocupan más por mejorar áreas de configuración, calidad y supervisión de proyectos; la gran mayoría de los esfuerzos en este grupo son organizacionales.

Un punto que es muy importante en las empresas de la industria del Software es que no hay preocupación por la Administración de Proveedores, lo cual refleja el hecho de que no hay alianzas entre las empresas de esta industria. No hay cooperación.

PA-Tamaño



Empresas con 500 empleados o menos (n=39)

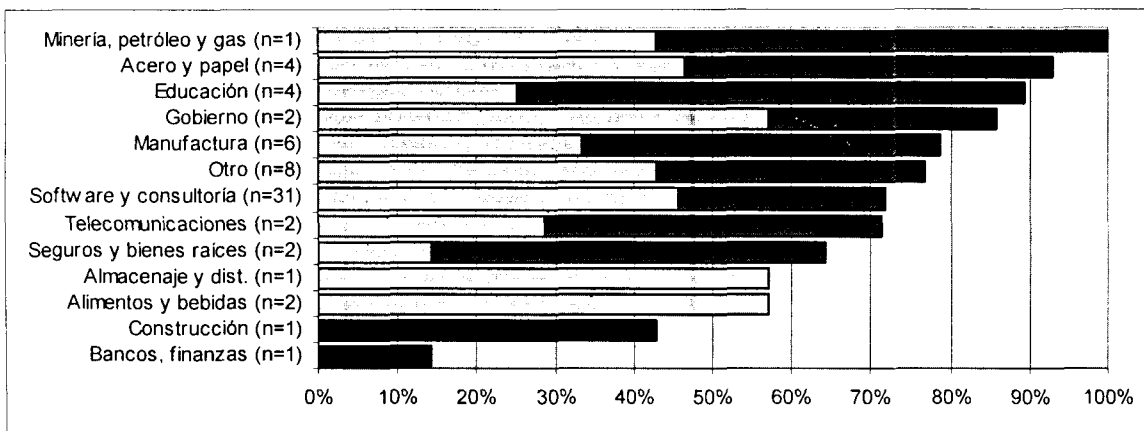


Empresas con más de 500 empleados (n=26)

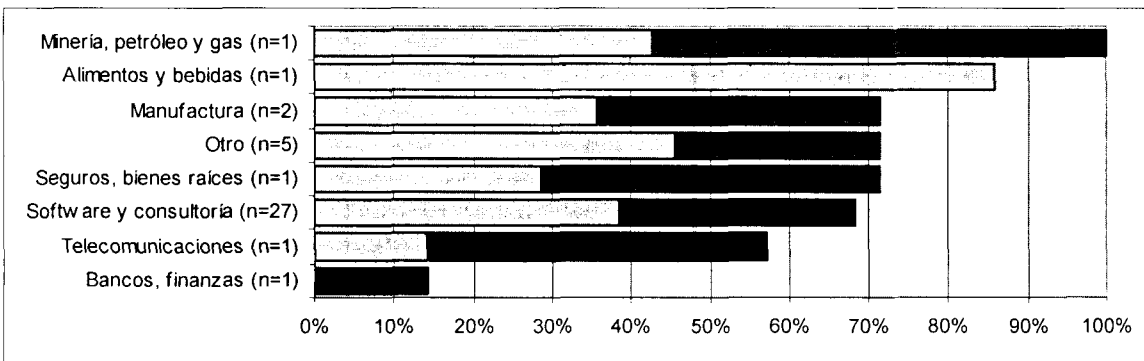
Nuevamente coinciden en estos dos diferentes bloques las áreas con más intentos de mejora. Sin embargo, destaca que las empresas con más de 500 empleados, aunque la mayoría si cuentan con intentos en el área de “Administración de Requerimientos”, la mayoría de estos son de forma aislada. Una razón para que las empresas grandes tengan más intentos, es porque es más difícil integrar a más gente y se necesita de mejoras basadas en modelos sólidos como CMM.

Por Giro

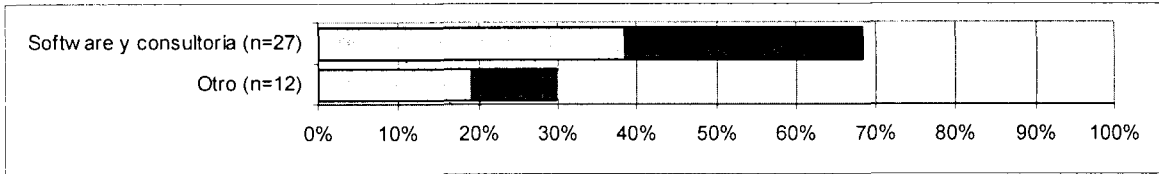
En las siguientes gráficas podemos apreciar los intentos de mejora comparando los diferentes giros que participaron en la encuesta. La primer área de cada barra representan los intentos a nivel organizacional, la siguiente los intentos aislados y lo que falta para completar el 100% representa las faltas de intentos de mejora. Los porcentajes están dados de acuerdo al número de empresas participantes de cada giro y promediados entre las 7 áreas clave de proceso por las que se preguntó. Cabe señalar que se duplicó la gráfica donde se ven las empresas pequeñas, dado que la gran mayoría son de Software, las empresas que no son de dicha industria se agruparon en la segunda gráfica bajo el concepto de Otro.



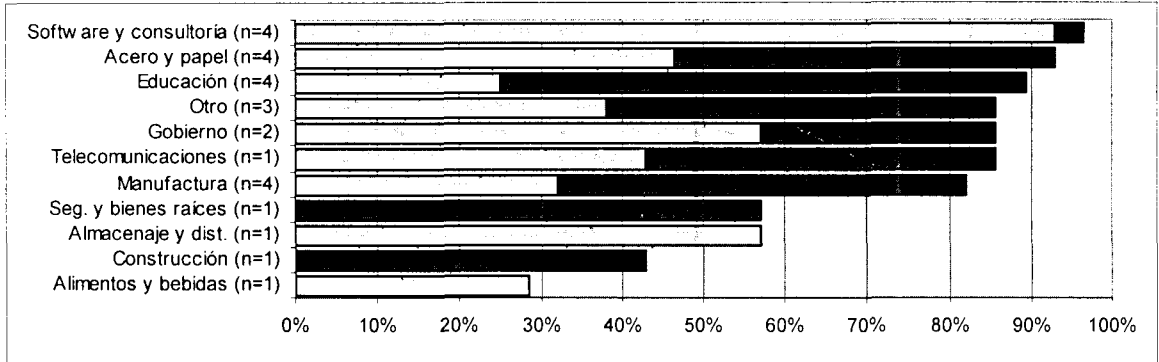
Todas las empresas (n=65)



Empresas con 500 empleados o menos (n=39)



Empresas con 500 empleados o menos (Industria del Software contra el resto) (n=65)

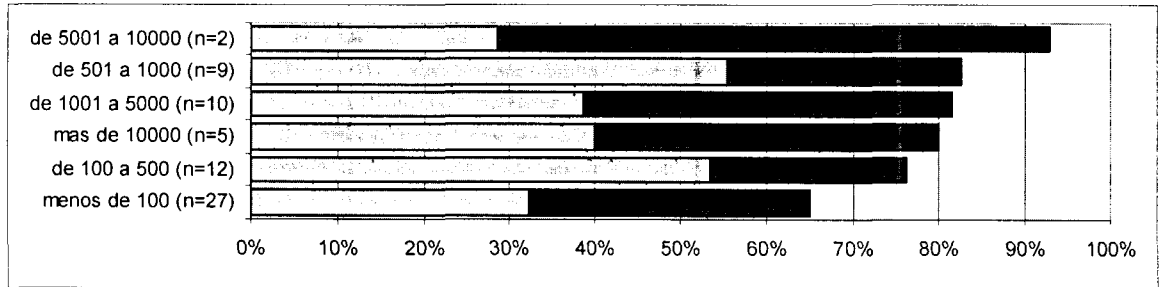


Empresas con más de 500 empleados (n=26)

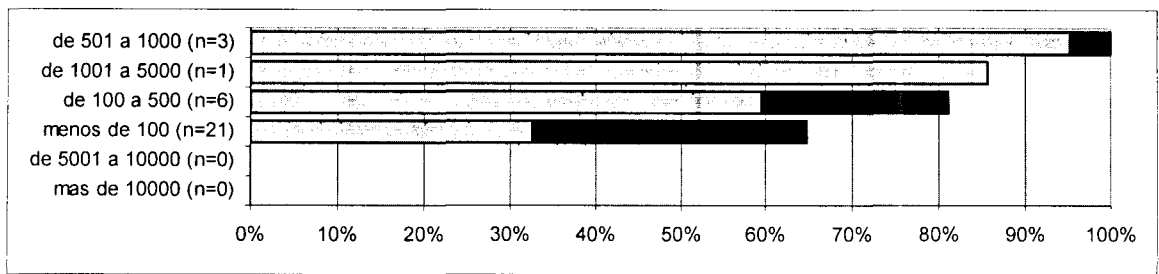
Se puede apreciar una gran falta de esfuerzos a nivel organizacional en las empresas que no pertenecen a la industria del Software.

Por Tamaño (número total de empleados)

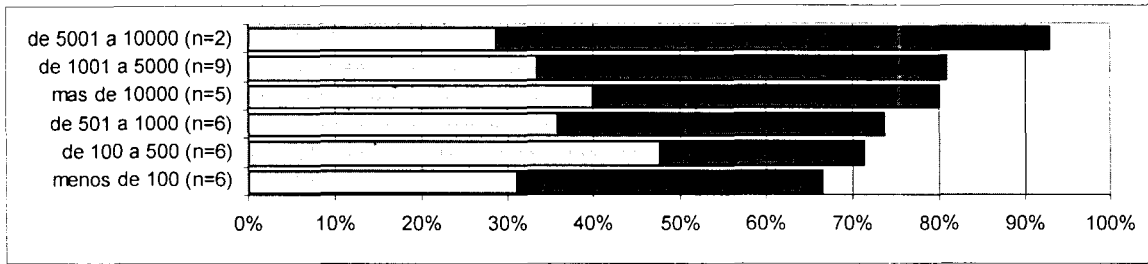
En las siguientes gráficas podemos apreciar los intentos de mejora comparando los diferentes rangos de tamaño que participaron en la encuesta. La primer área de cada barra representan los intentos a nivel organizacional, la siguiente los intentos aislados y lo que falta para completar el 100% representa las faltas de intentos de mejora.



Todas las empresas (n=65)



Empresas de Software (n=31)

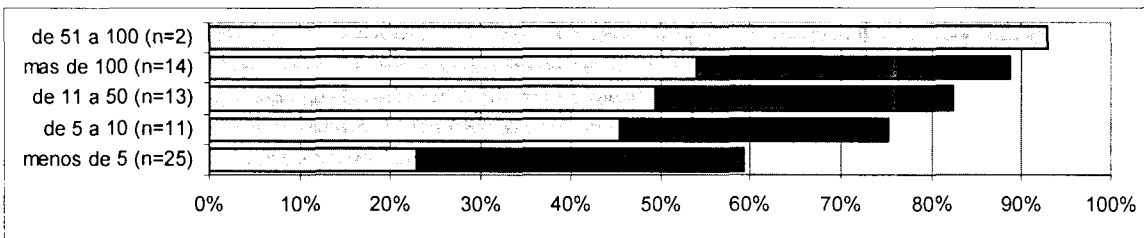


Otros giros (n=34)

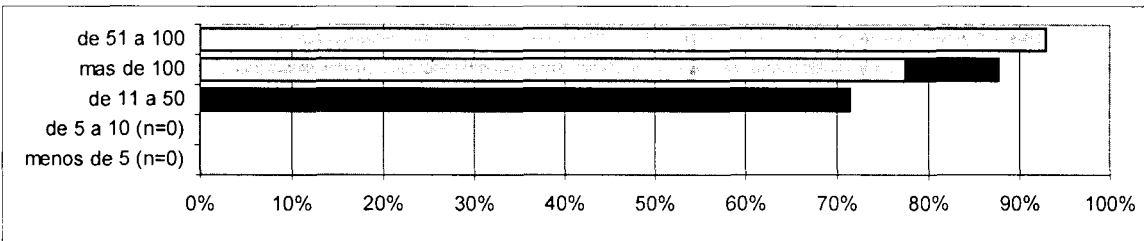
Aquí se destaca que las empresas de otros giros, aunque tienen un buen número de esfuerzos, la gran mayoría son aislados.

Por Tamaño (número de empleados que desarrollan Software)

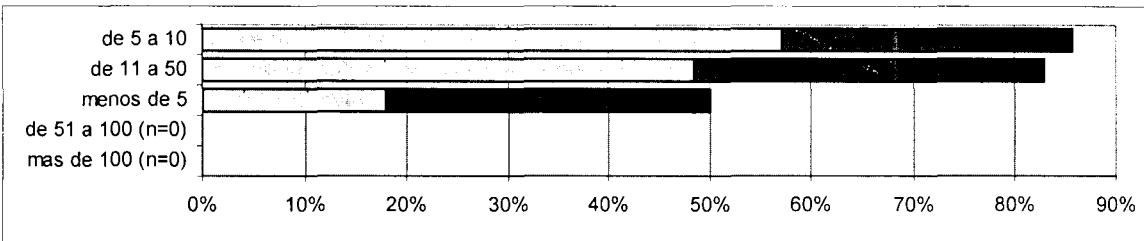
En las siguientes gráficas podemos apreciar los intentos de mejora comparando los diferentes rangos de tamaño que participaron en la encuesta. La primer área de cada barra representan los intentos a nivel organizacional, la siguiente los intentos aislados y lo que falta para completar el 100% representa las faltas de intentos de mejora.



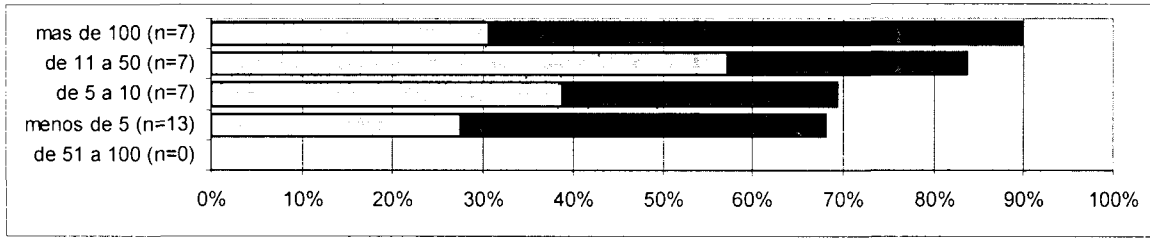
Todas las empresas (n=65)



Empresas de Software grandes (n=10)



Empresas de Software chicas (n=21)

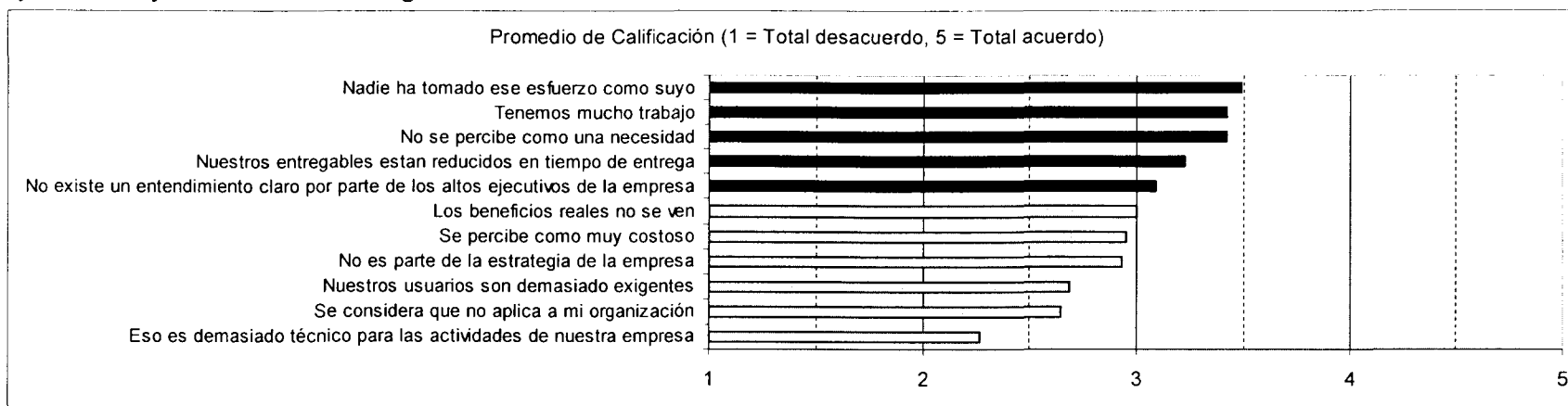


Otros giros (n=34)

Se observa de las gráficas anteriores, que todos los intentos de mejora de las empresas que se dedican al desarrollo de Software y tienen entre 51 y 100 empleados son organizacionales. Así mismo, dentro de las que no se dedican a desarrollar Software, las que mayor proporción de esfuerzos organizacionales tienen son las que cuentan entre 11 y 50 empleados.

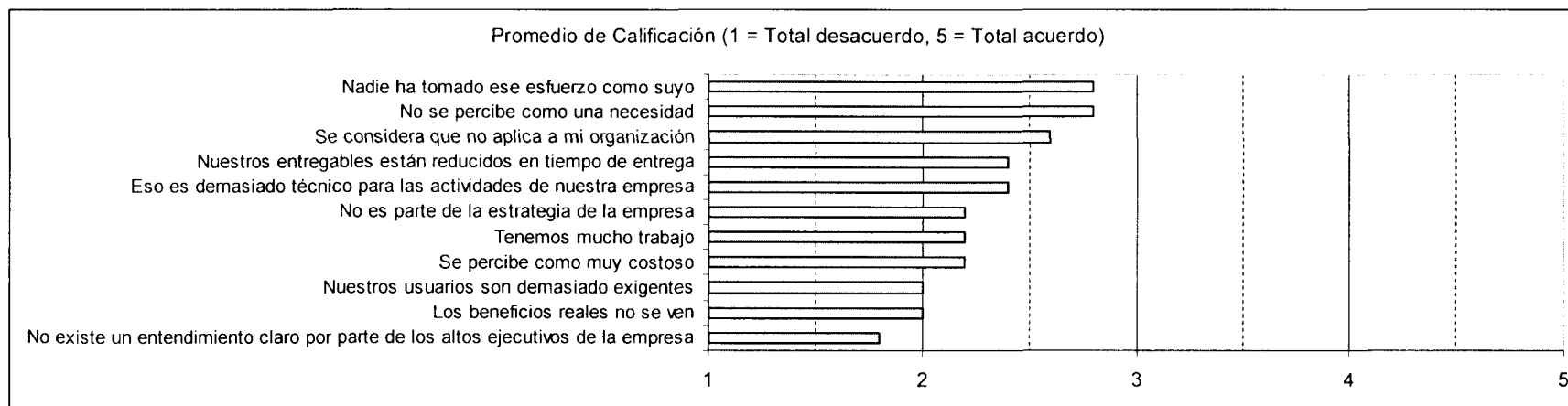
Razones para que No hayan Intentado mejorar

Se le preguntó a las empresas que no han intentado mejorar en algún área, cuáles son los principales obstáculos o razones para que no lo hayan hecho, y se obtuvieron los siguientes factores.

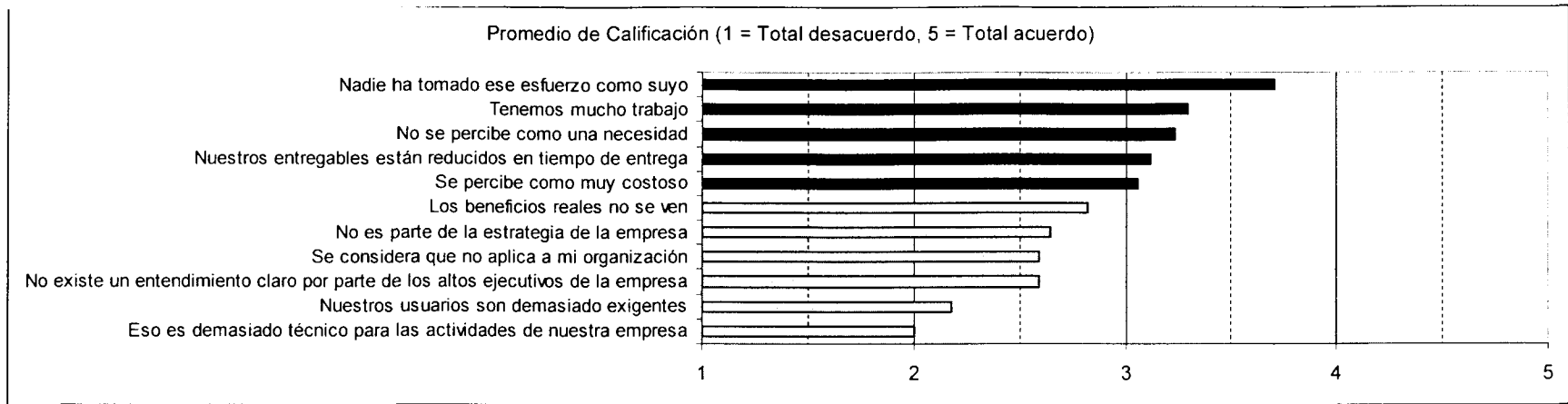


Todas las empresas (n=45)

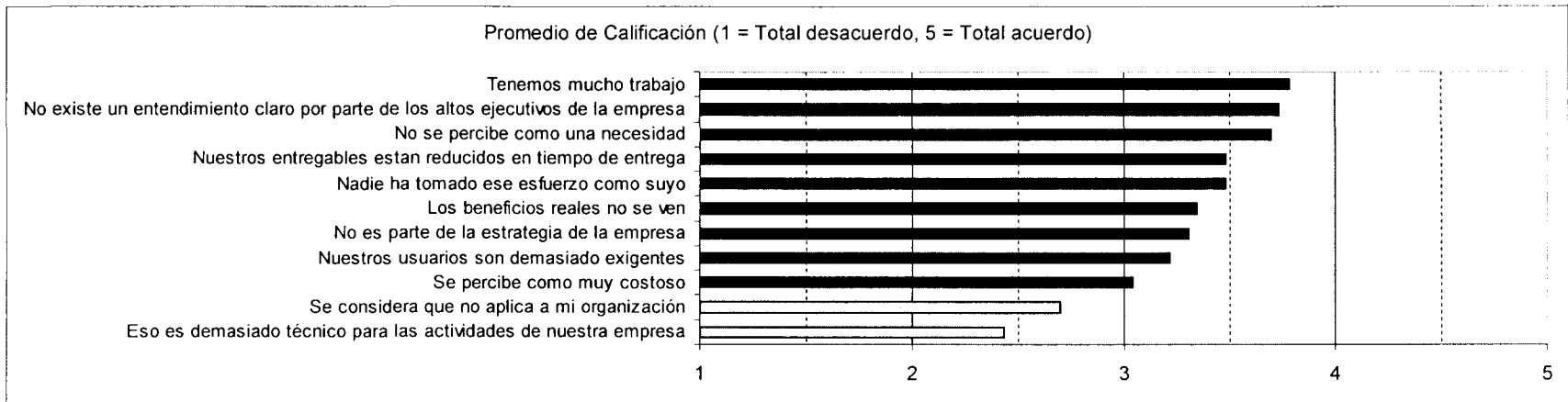
Otras razones para no haber intentado mejorar que nos dieron a conocer son: Porque hay pocos proveedores; La alta gerencia toma decisiones de tiempo sin conocer las implicaciones de lo que quieren o a lo que se comprometen. No se toman en cuenta a los expertos, sino al nivel organizacional más alto, aun y cuando no sepan de lo que están tratando. Poder de negociación mínimo.



Empresas de Software grandes (n=5)

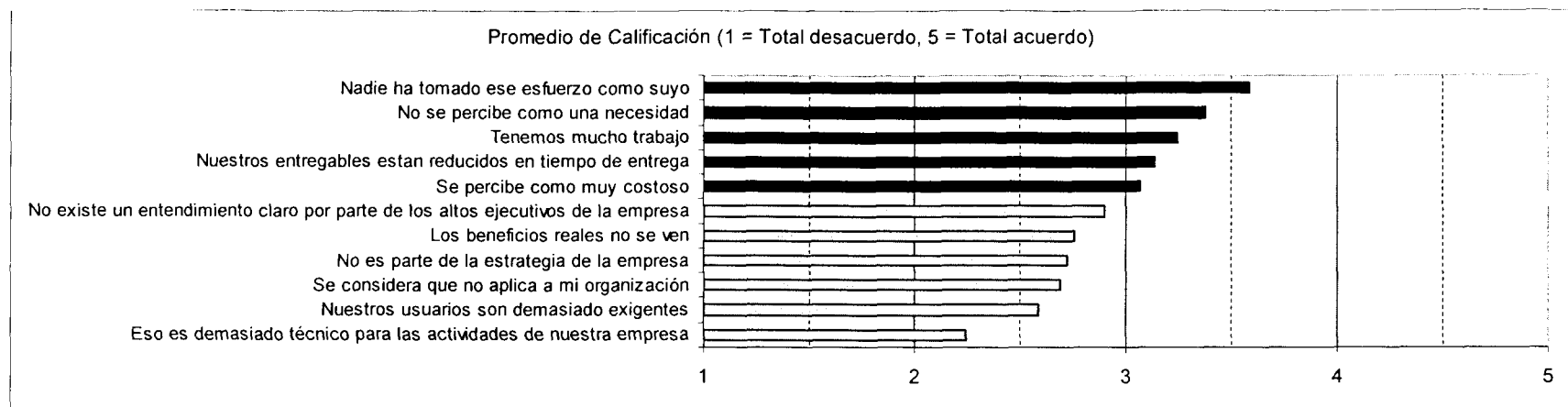


Empresas de Software chicas (n=17)

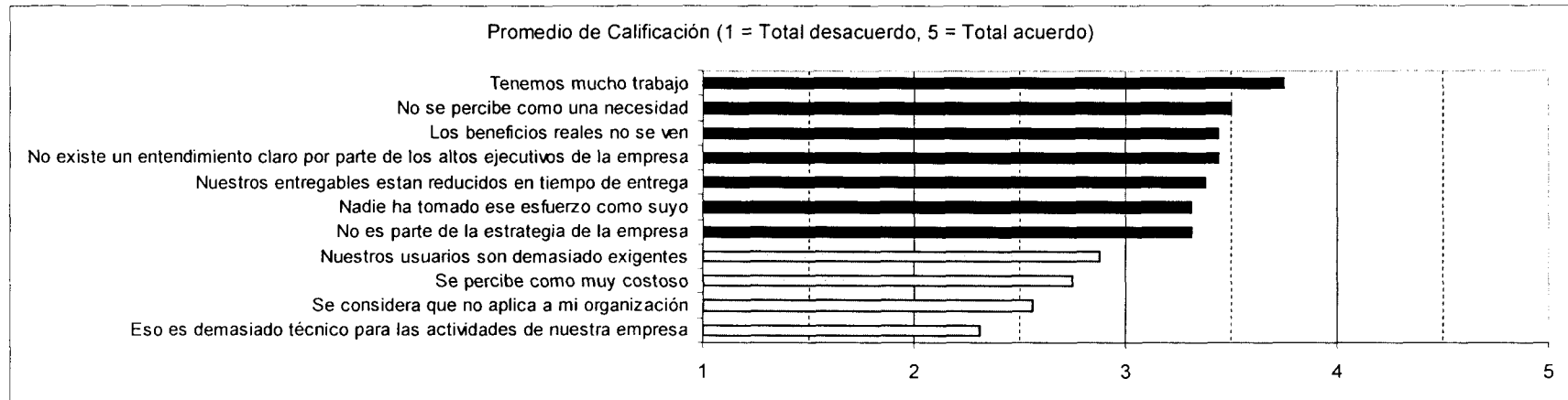


Otros giros (n=23)

No se encontraron factores significativos para no intentar mejorar el proceso en las empresas de Software grandes. Mientras que las razones comunes para no haber intentado alguna de las áreas de proceso en los demás bloques son “No se percibe como una necesidad” y “Tenemos mucho trabajo”. Lo que mas distingue en estas razones a las empresas de Software chicas es la de “Nadie ha tomado ese esfuerzo como suyo”, mientras que lo que destaca en las que no se dedican a desarrollar Software es “No existe un entendimiento claro por parte de los altos ejecutivos de la empresa”.



Empresas con 500 empleados o menos (n=29)

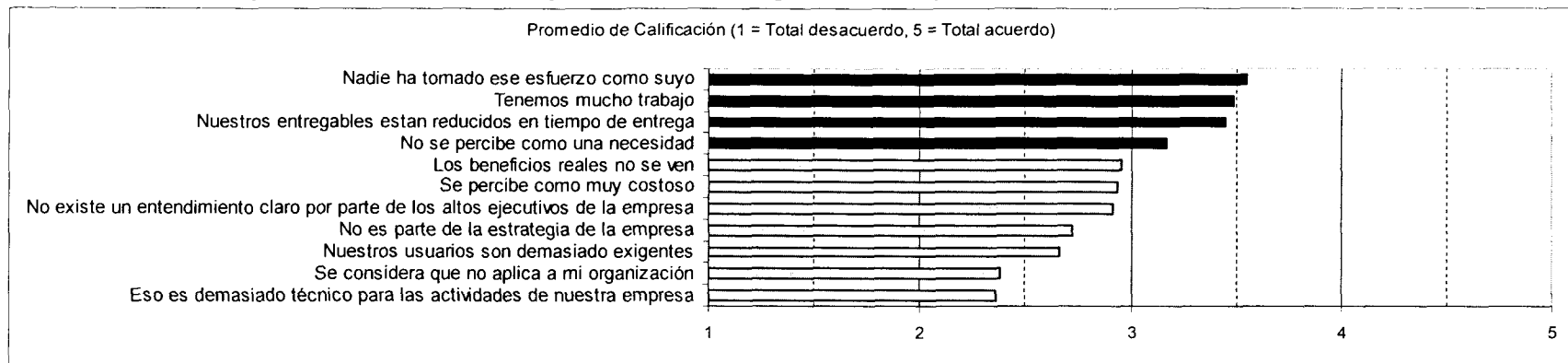


Empresas con más de 500 empleados (n=16)

En estos bloques vuelven a aparecer los mismos factores en común: “No se percibe como una necesidad” y “Tenemos mucho trabajo”. Para las empresas que tienen menos de 500 empleados, lo que más distingue en estas razones es la de “Nadie ha tomado ese esfuerzo como suyo”, mientras que lo que destaca en las empresas grandes es: “Los beneficios reales no se ven”.

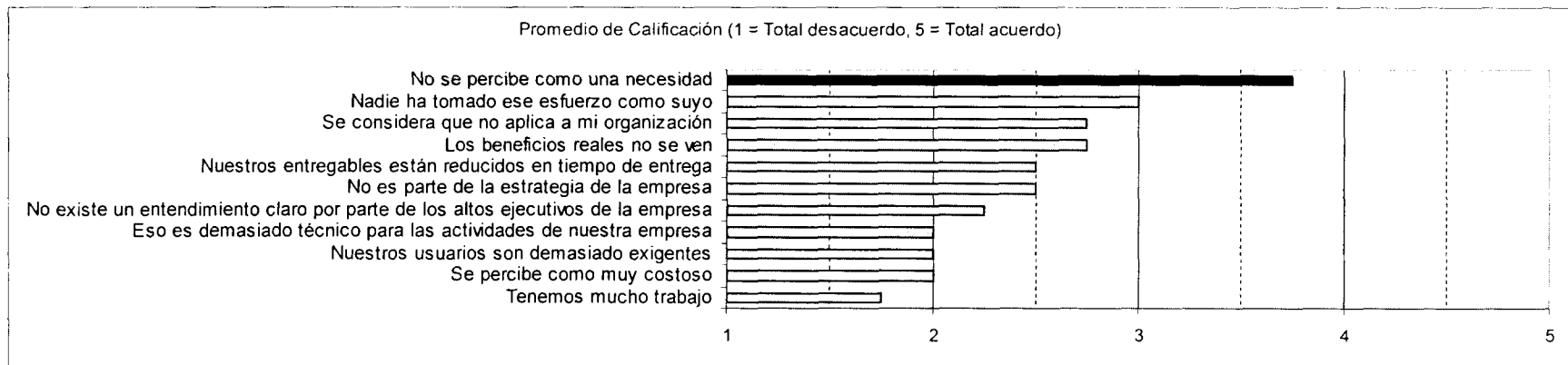
Razones para que los esfuerzos se hayan realizado de forma Aislada

Se le pregunto a las empresas que hayan intentado mejorar algún área pero de forma aislada, cuáles son los principales obstáculos o razones para que lo hayan hecho de esa manera y no de carácter organizacional, y se obtuvieron los siguientes factores.

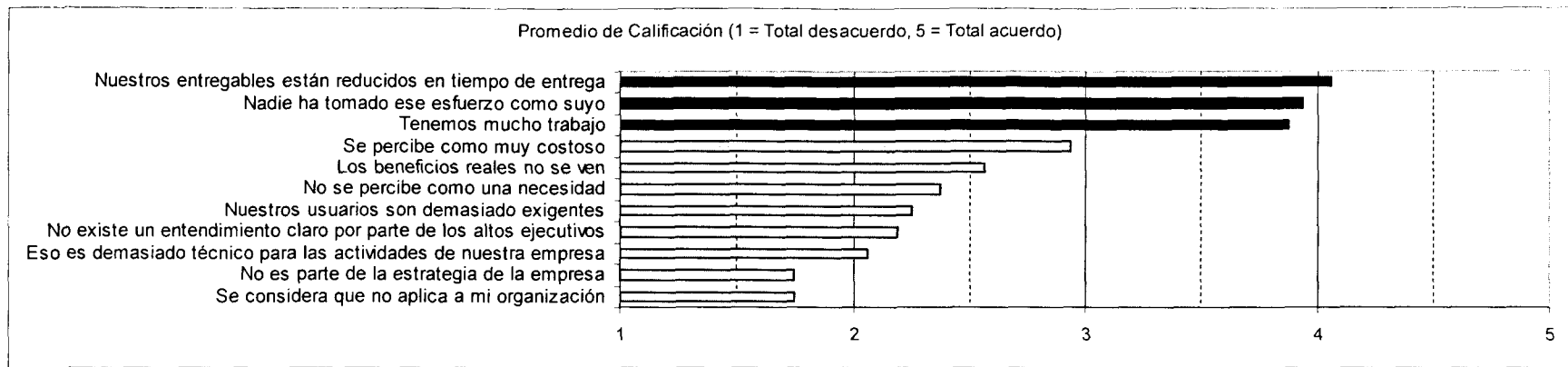


Todas las empresas (n=47)

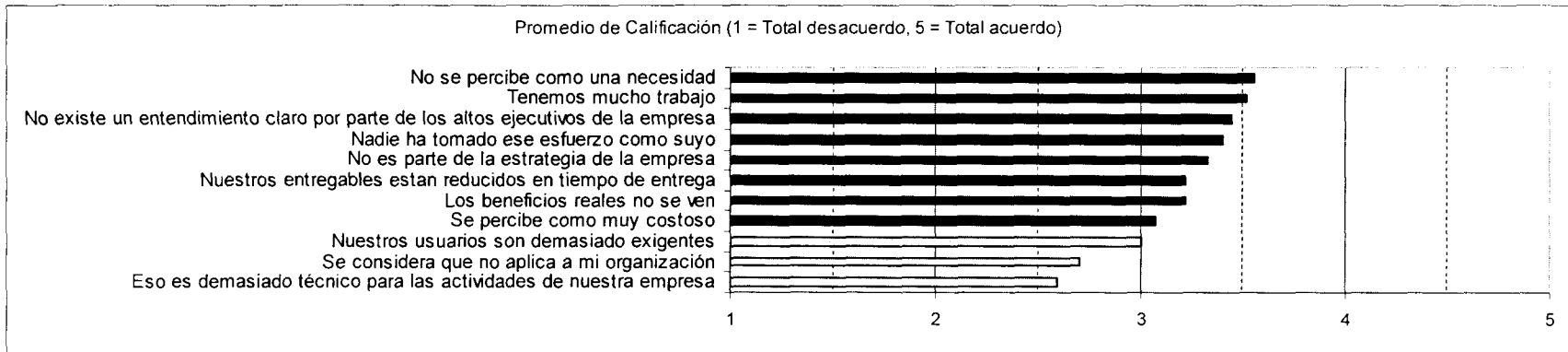
Otras razones mencionadas, para que los intentos de mejora realizados hayan sido de forma aislada: Se va avanzando en cada rubro poco a poco, pero no se ha formalizado como metodología; Faltan líderes, empezando por los dueños que trabajan dentro de la empresa; Objetivos particulares en conflicto de las empresas del corporativo que no son posibles de conjuntar; Existe competencia interna que bloquea el esfuerzo conjunto por tener un bienestar general, cada dirección “jala” por su lado.



Empresas de Software grandes (n=4)

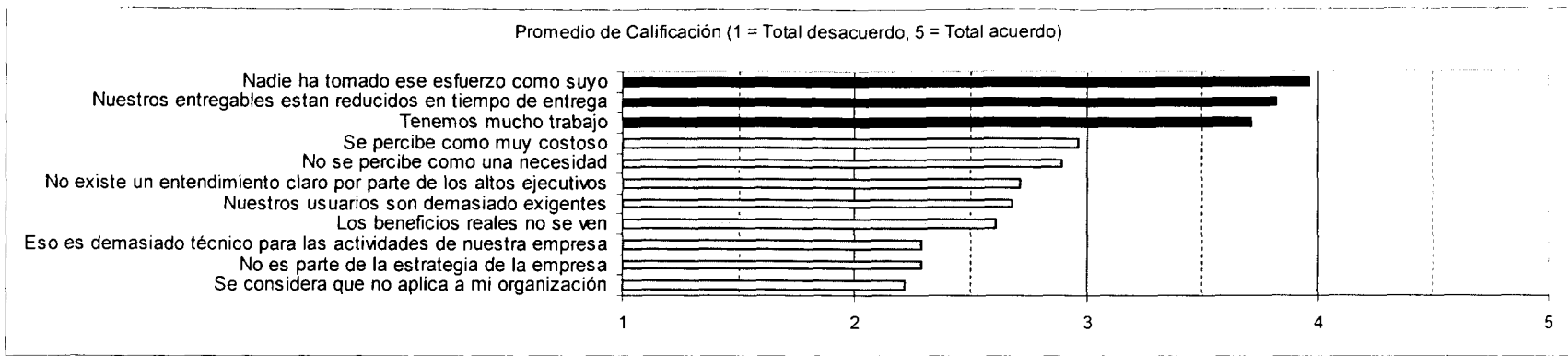


Empresas de Software chicas (n=16)

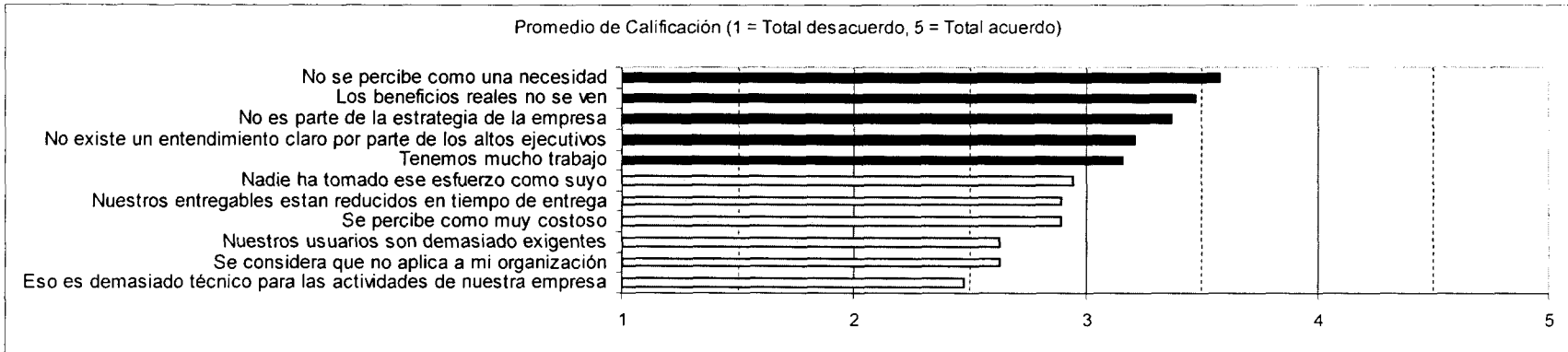


Otros giros (n=27)

El único factor que resalta en las empresas de Software grandes es el de “No se percibe como una necesidad”, este factor se refiere al área de proceso “Administración de Proveedores”. Los bloques restantes están de acuerdo en que un factor importante es el de “Tenemos mucho trabajo”. Los participantes de empresas de Software pequeñas resaltan que “Nuestros entregables están reducidos en tiempo de entrega” y que “Nadie ha tomado ese esfuerzo como suyo”, mientras que el sentir de las demás industrias es que “No se percibe como una necesidad” y que “No existe un entendimiento claro por parte de los altos ejecutivos de la empresa”.



Empresas con 500 empleados o menos (n=28)

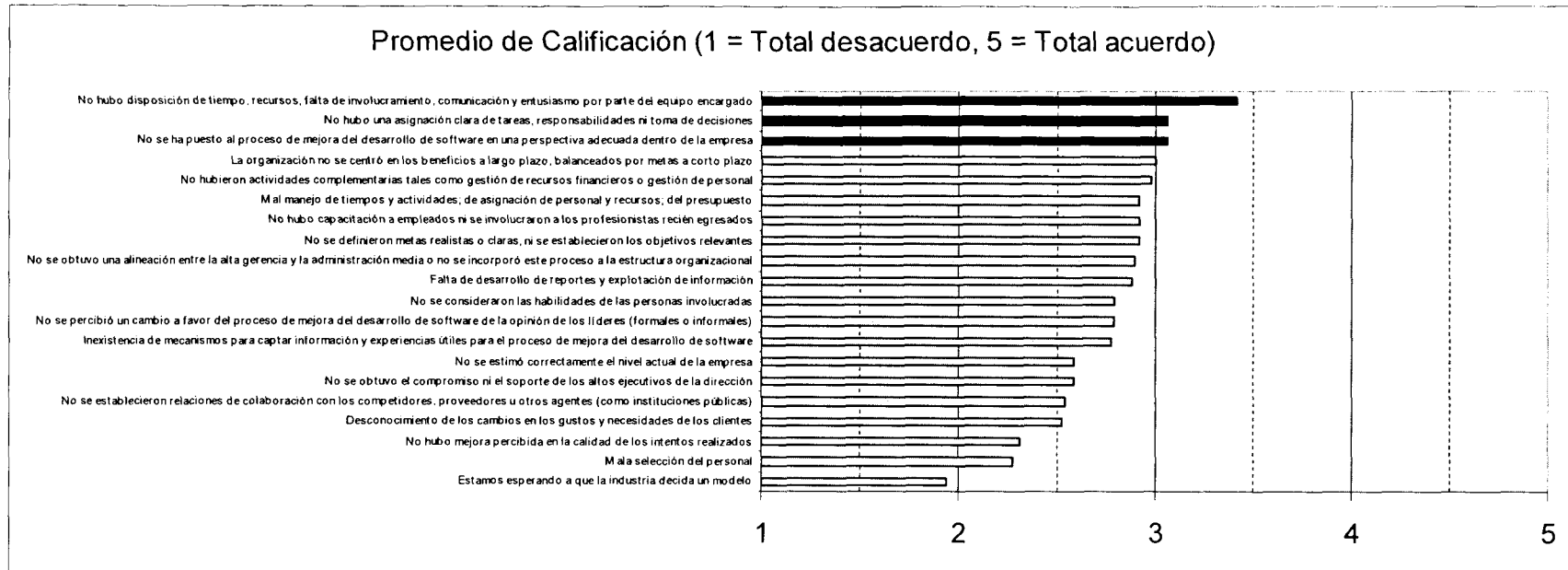


Empresas con más de 500 empleados (n=19)

Los principales factores de los bloques anteriores mucho discrepan entre sí. El bloque perteneciente a las empresas pequeñas manifiesta los siguientes factores como los más importantes: “Nadie ha tomado ese esfuerzo como suyo”, “Nuestros entregables están reducidos en tiempo de entrega” y “Tenemos mucho trabajo”. Mientras que las empresas grandes resaltan de entre los factores que “No se percibe como una necesidad”, “Los beneficios reales no se ven” y que “No es parte de la estrategia de la empresa”. Es interesante ver como el factor “No es parte de la estrategia de la empresa” es muy importante para las empresas las grandes y no lo es para las chicas.

Factores de Fracaso

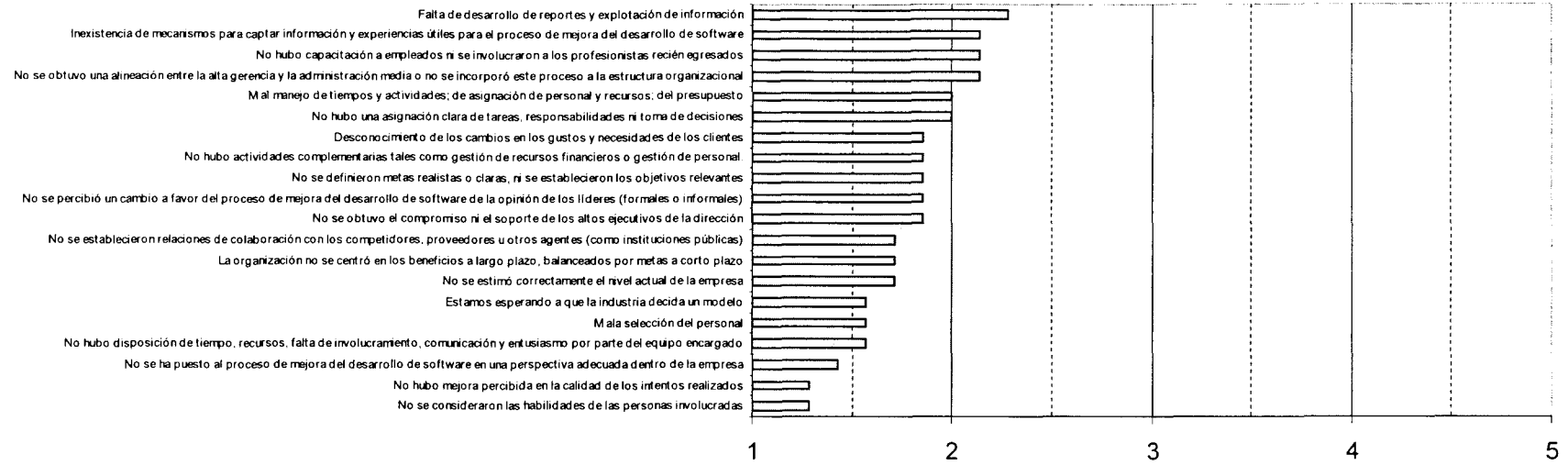
Se le pregunto a las empresas que han intentado mejorar alguna de las áreas, y dichos intentos han fracasado, cuales son las razones por las que consideran que han fracasado, y se obtuvieron los siguientes factores.



Todas las empresas (n=48)

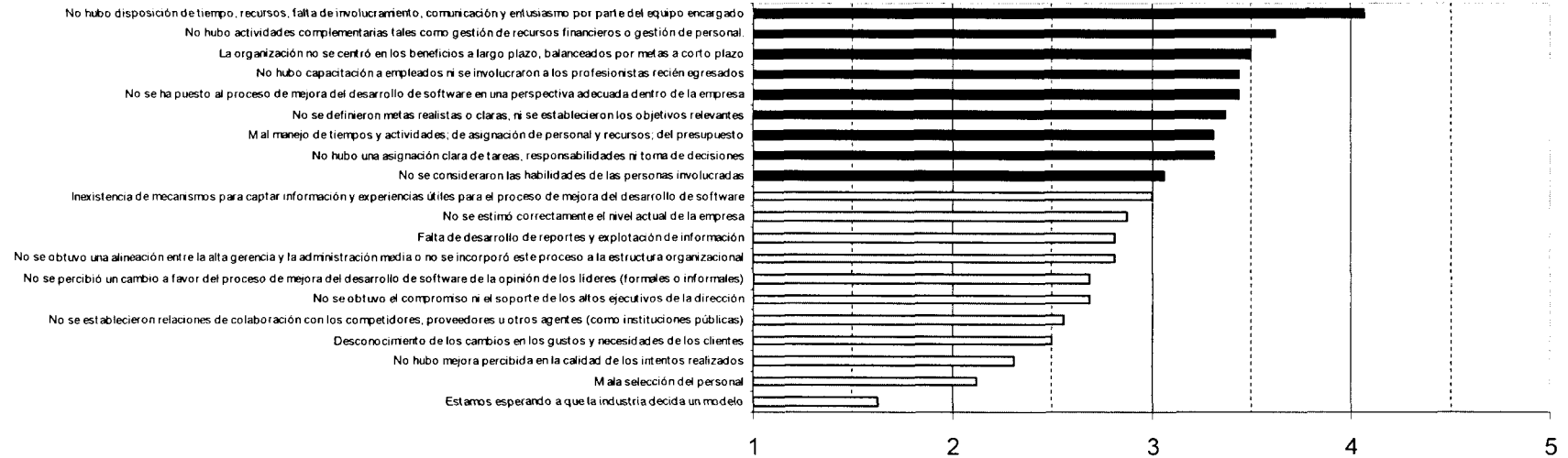
Otros factores de fracaso que nos revelaron: Falta de personal en el área de procesos, el SEPG fue formado 6 meses después de lo planeado; Los desarrolladores no le encuentran practicidad; No se ha tenido contrato con los clientes, que a su vez cambiaban los requerimientos durante el proceso.

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)

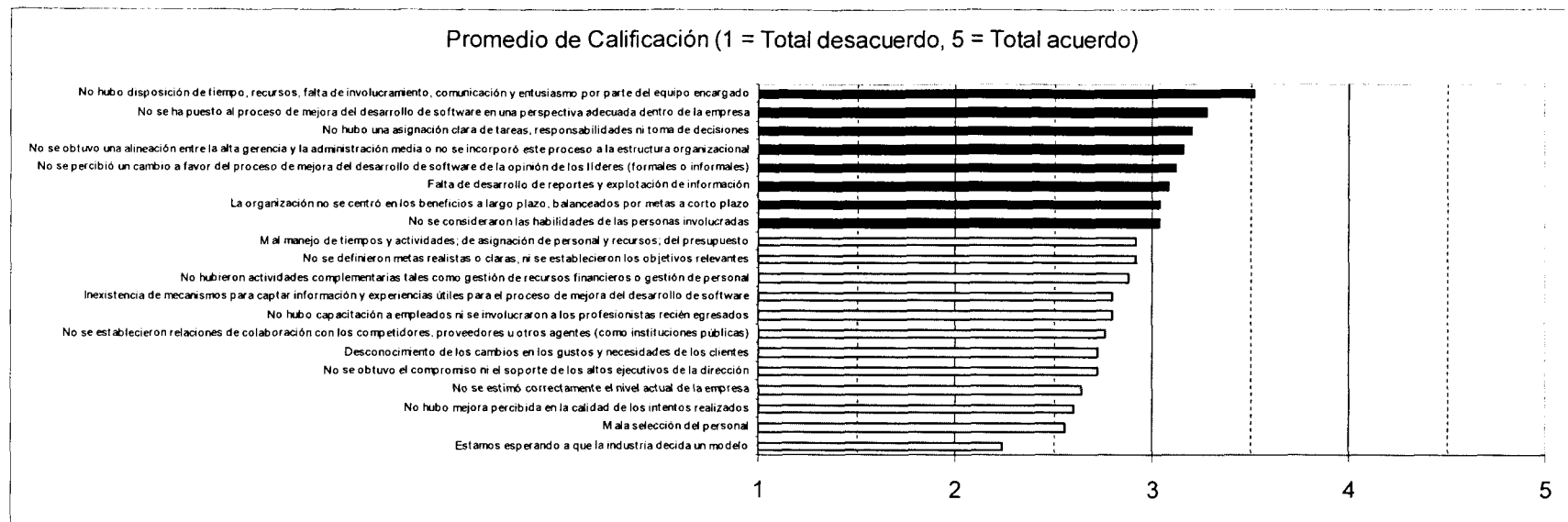


Empresas de Software grandes (n=7)

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



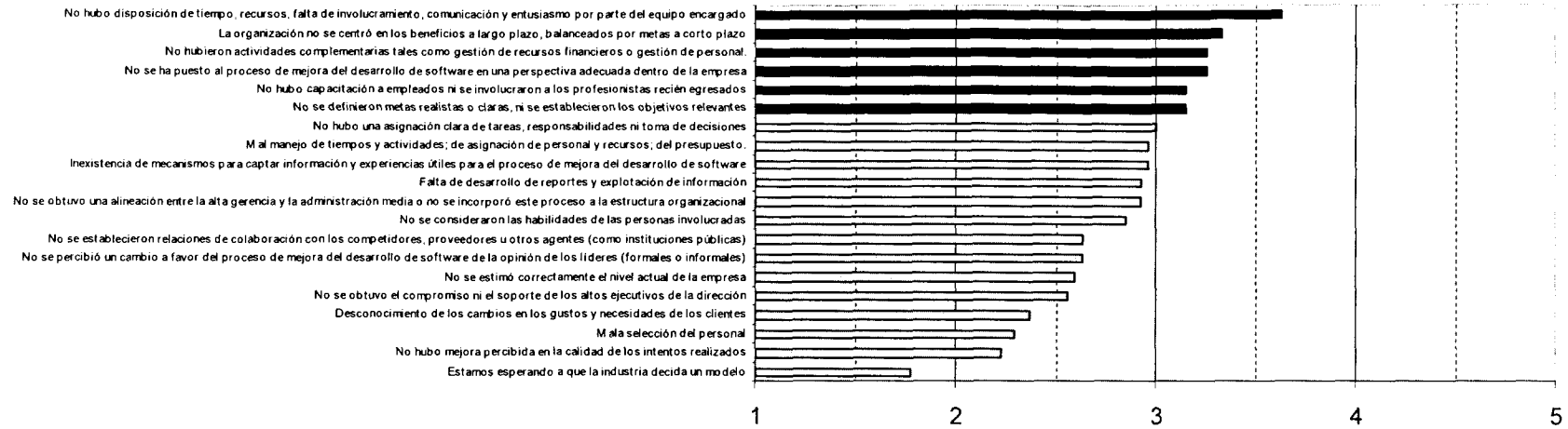
Empresas de Software chicas (n=16)



Otros giros (n=25)

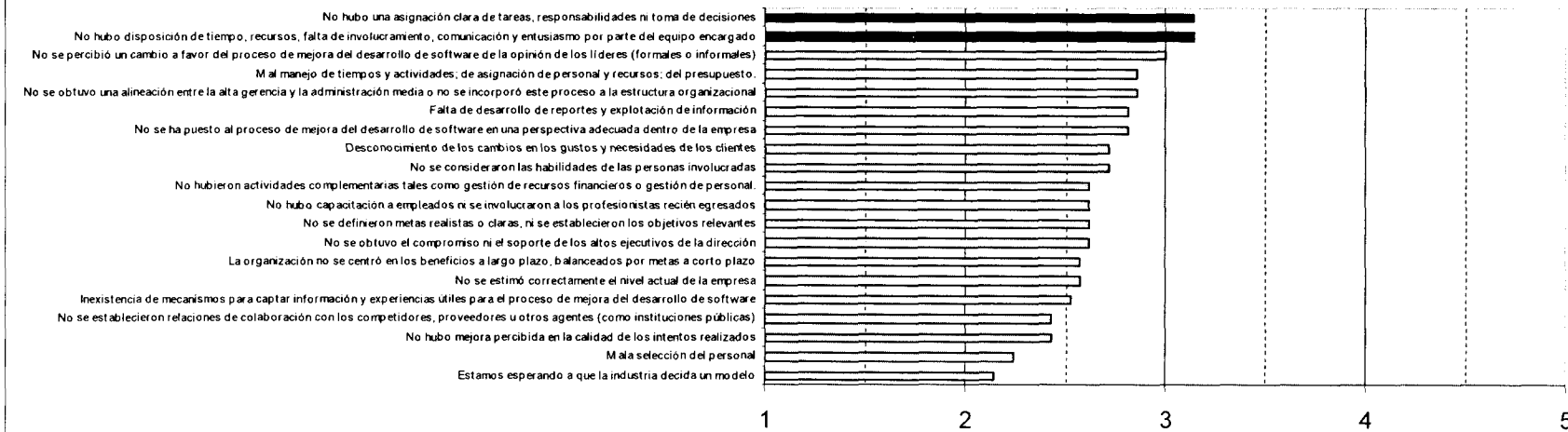
Las empresas de Software grandes no dan a conocer factores significativos de fracaso en el proceso de mejora. Para las demás empresas, el factor principal donde coinciden los bloques anteriores es el de “No hubo disposición de tiempo, recursos, falta de involucramiento, comunicación y entusiasmo por parte del equipo encargado”. Lo que distingue a las empresas de Software chicas son los factores “No hubieron actividades complementarias tales como gestión de recursos financieros o gestión de personal”, “No hubo capacitación a empleados ni se involucraron a los profesionistas recién egresados”, “La organización no se centró en los beneficios a largo plazo, balanceados por metas a corto plazo” y “Mal manejo de tiempos y actividades; de asignación de personal y recursos; del presupuesto”. Lo que distingue a las demás industrias es que “No se ha puesto al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa”, “No hubo una asignación clara de tareas, responsabilidades ni toma de decisiones”, “No se obtuvo una alineación entre la alta gerencia y la administración media o no se incorporó este proceso a la estructura organizacional” y “No se percibió un cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)”.

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



Empresas con 500 empleados o menos (n=27)

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



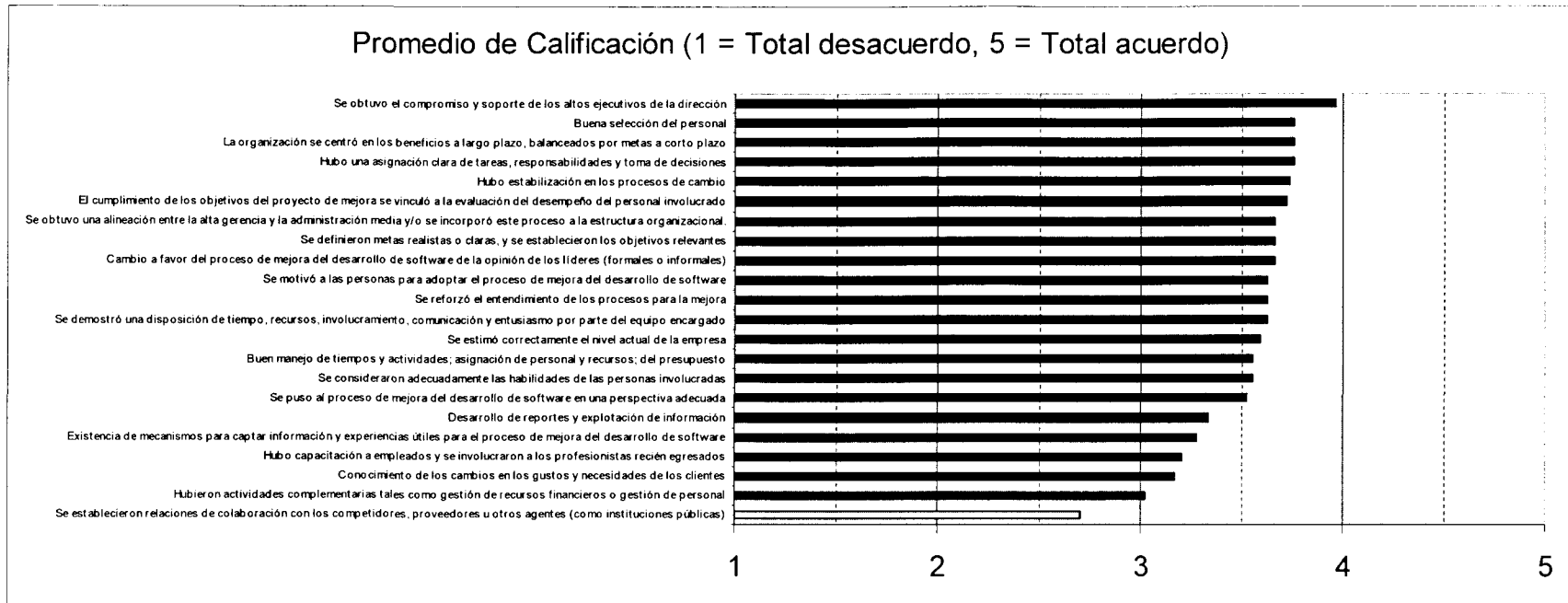
Empresas con más de 500 empleados (n=21)

Los bloques anteriores también tienen en común el factor “No hubo disposición de tiempo, recursos, falta de involucramiento, comunicación y entusiasmo por parte del equipo encargado” dentro de los principales que les han obstaculizado lograr sus objetivos de mejoramiento. Lo distintivo de las empresas chicas es que “La organización no se centró en los beneficios a largo plazo, balanceados por metas a corto plazo”, “No hubieron actividades complementarias tales como gestión de recursos financieros o gestión de personal”, “No se ha puesto al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa” y “No hubo

capacitación a empleados ni se involucraron a los profesionistas recién egresados”. Mientras que a las empresas mas grandes distingue que “No hubo una asignación clara de tareas, responsabilidades ni toma de decisiones”, “No se percibió un cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)”, “Mal manejo de tiempos y actividades; de asignación de personal y recursos; del presupuesto” y “No se obtuvo una alineación entre la alta gerencia y la administración media o no se incorporó este proceso a la estructura organizacional”.

Factores de Éxito

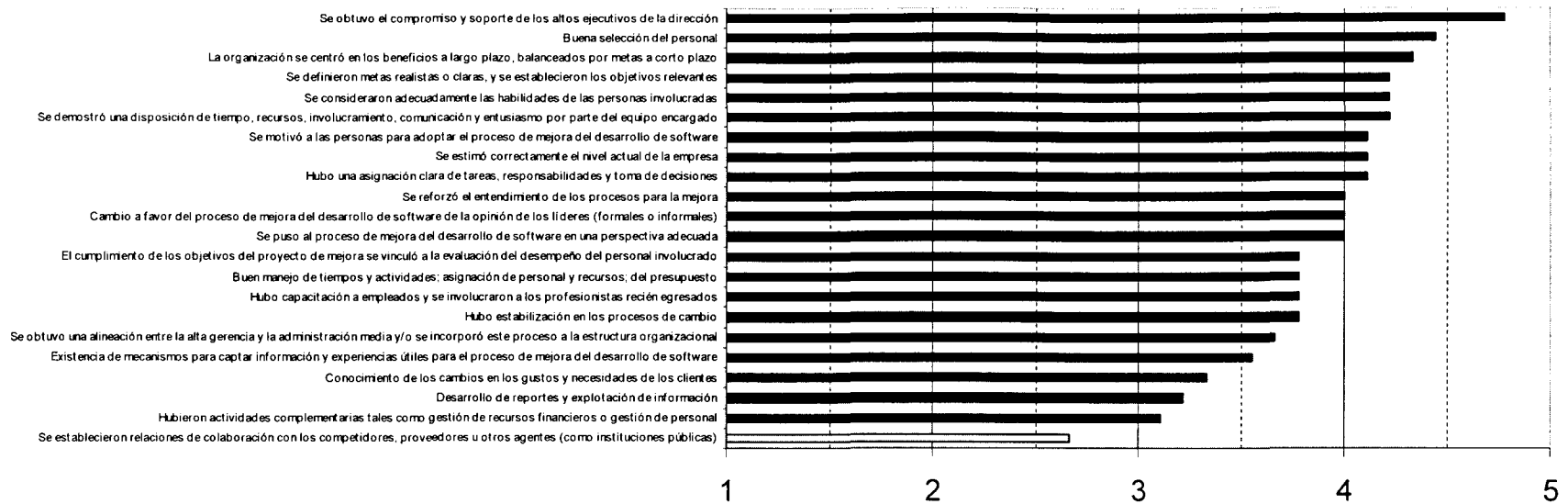
Se le pregunto a las empresas que han intentado mejorar alguna de las áreas, y dichos intentos han tenido éxito, cuales son las razones por las que consideran que han tenido éxito, y se obtuvieron los siguientes factores.



Todas las empresas (n=54)

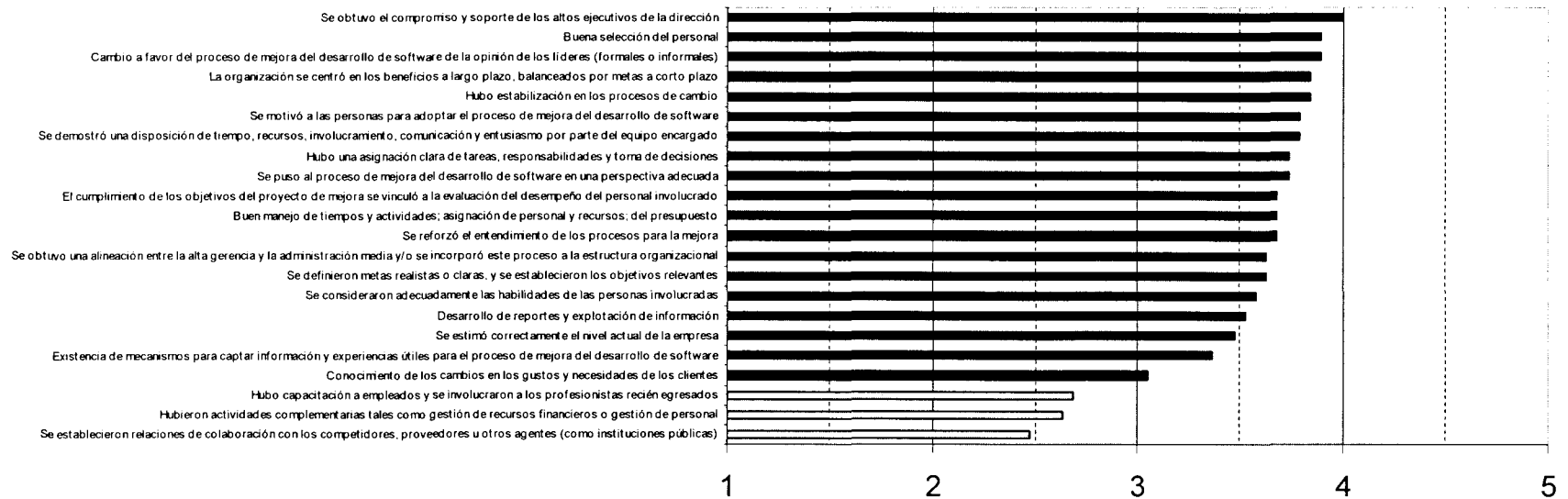
Otros factores de éxito encontrados: El éxito se debió mas a la necesidad que tales puntos fueran implementados, para mejorar la competencia; Entendimiento claro de las necesidades del cliente, no de lo que ellos quieren; Los participantes han visto práctico el seguir una metodología; Porque las personas saben que la mejora es necesaria.

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



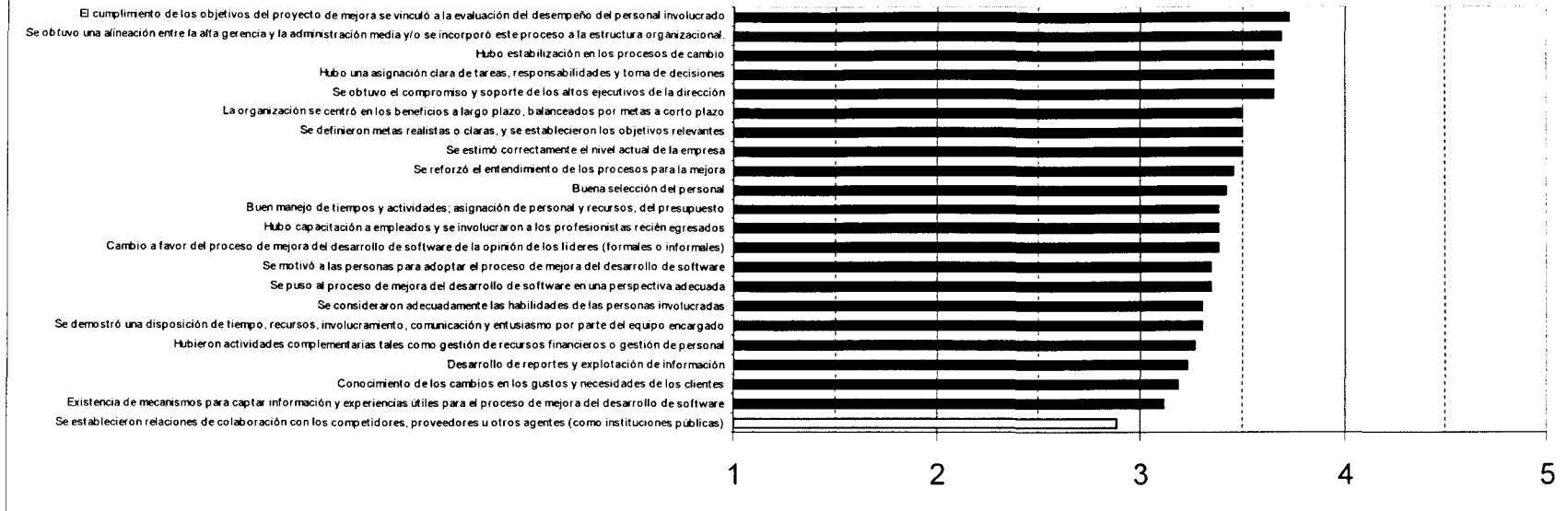
Empresas de Software grandes (n=9)

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



Empresas de Software chicas (n=19)

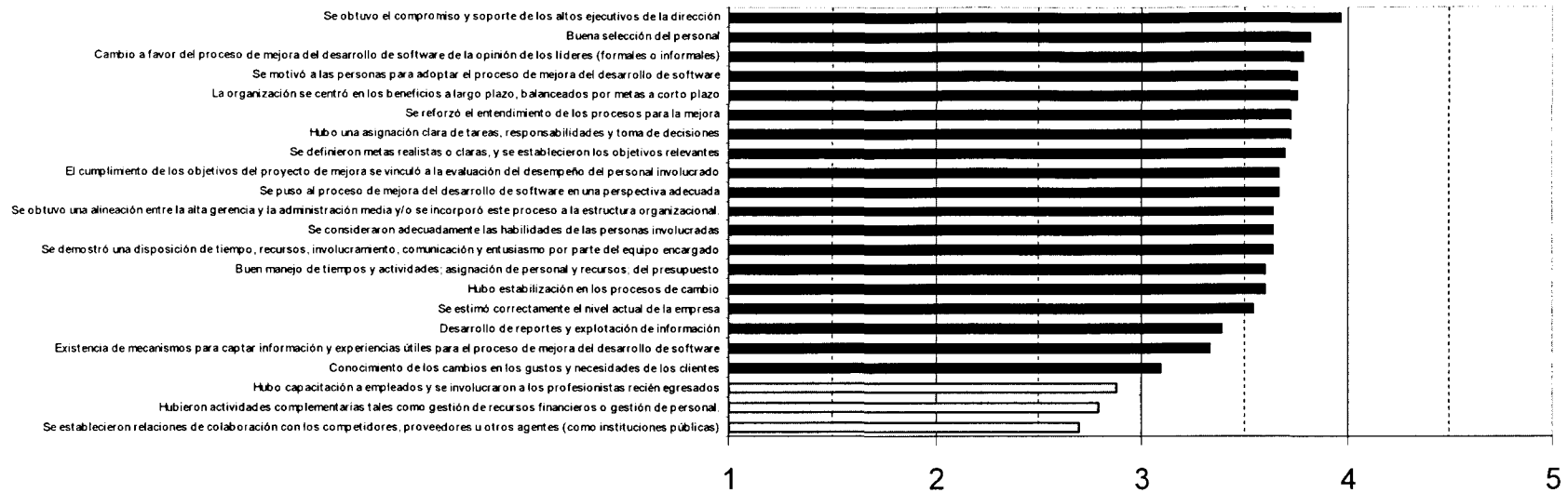
Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



Otros giros (n=26)

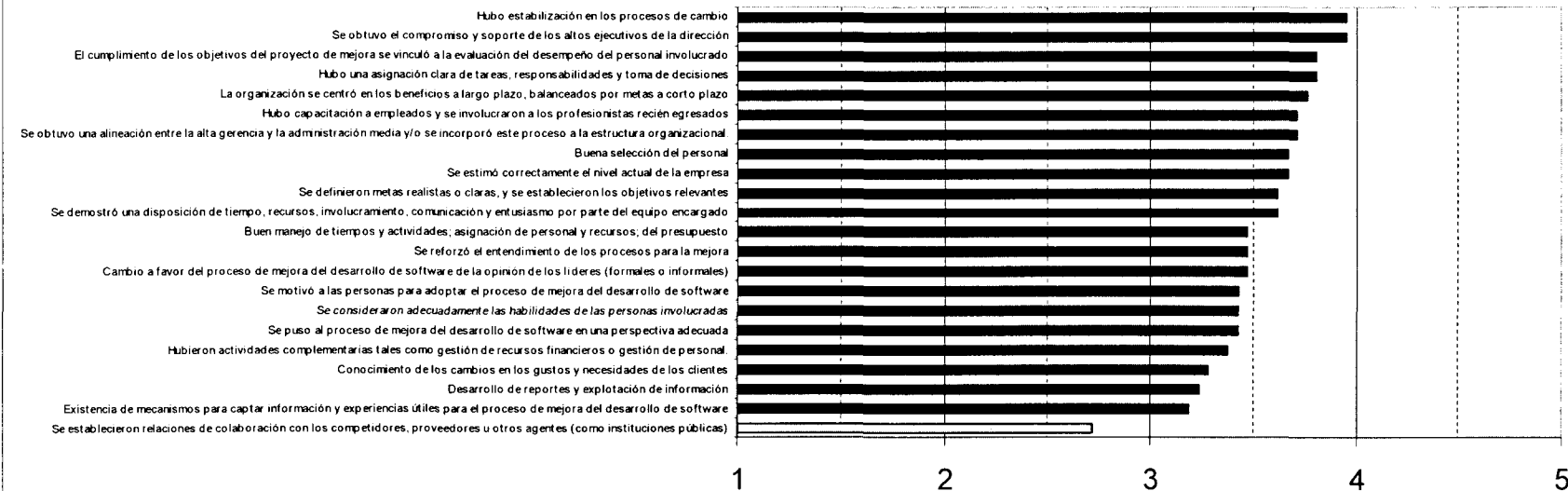
Estos bloques tienen como factor común al “Se obtuvo el compromiso y soporte de los altos ejecutivos de la dirección” en los primeros lugares, mientras que lo que distingue a la industria del Software en cuanto a los factores que consideran los han llevado al éxito en los esfuerzos de mejora son “Buena selección del personal”, “La organización se centró en los beneficios a largo plazo, balanceados por metas a corto plazo”, “Cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)” y “Se demostró una disposición de tiempo, recursos, involucramiento, comunicación y entusiasmo por parte del equipo encargado”. Mientras que el resto de las industrias están de acuerdo con los siguientes factores: “El cumplimiento de los objetivos del proyecto de mejora se vinculó a la evaluación del desempeño del personal involucrado”, “Se obtuvo una alineación entre la alta gerencia y la administración media y/o se incorporó este proceso a la estructura organizacional”, “Hubo estabilización en los procesos de cambio” y “Hubo una asignación clara de tareas, responsabilidades y toma de decisiones”.

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



Empresas con 500 empleados o menos (n=33)

Promedio de Calificación (1 = Total desacuerdo, 5 = Total acuerdo)



Empresas con más de 500 empleados (n=21)

En los bloques anteriores se pueden observar dos factores en común dentro de los primeros lugares, “Se obtuvo el compromiso y soporte de los altos ejecutivos de la dirección” y “La organización se centró en los beneficios a largo plazo, balanceados por metas a corto plazo”. Los factores que consideran más importantes particularmente las empresas pequeñas son “Buena selección del personal”, “Cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)” y “Se motivó a las personas para adoptar el proceso de mejora del desarrollo de software”. Para las empresas grandes los puntos mas importantes son “Hubo estabilización en los procesos de cambio”, “El cumplimiento de los objetivos del proyecto de mejora se vinculó a la evaluación del desempeño del personal involucrado” y “Hubo una asignación clara de tareas, responsabilidades y toma de decisiones”.

Comentarios adicionales

En la encuesta se le pidió al entrevistado que redactara un comentario final respecto al mejoramiento del proceso de Desarrollo de Software. Se encontró el resultado de esta sección de mucho valor. A continuación se dan a conocer los comentarios recibidos clasificados por tema.

Clima Organizacional

- Para nosotros el proceso de mejora a nivel organizacional es posible si lo realizamos con Pasión y Participación de todos los niveles de la empresa, uno de los lemas que utilizamos en nuestras campañas de comunicación es "Pasión + Participación".
- Debe existir un compromiso real en todos los niveles, para que líderes de otros proyectos que no estuvieron involucrados inicialmente no lo apliquen cuando se les complica cumplirlos debido a su propia administración de proyectos.
- Cuando las cosas funcionan es porque lo dejan en manos de un equipo pequeño de personas y les dan tal poder que pueden imponer condiciones. El resto de los esfuerzos aislados se debe a la competencia interna. Mal clima organizacional.
- Otros factores de éxito importantes son la administración del cambio, la integración con personas con mucha experiencia y el trabajo en paralelo con el "cómo" en vez del "qué".
- Se fracasa cuando no hay disciplina, lo contrario también es cierto, se triunfa cuando hay disciplina.
- Funcionado en la medida que la alineación de la estrategia Informática con la estrategia organizacional es más que el color de las ventanas de las aplicaciones o el estilo de cada usuario para involucrarse en las mejoras, si el usuario no está de acuerdo en mejorar entonces un mejor software no representa nada para él, es solo una moda que le quieren imponer "los de sistemas" si se acepta que una mejora vendrá empacada en un mejor software entonces se tienen altas probabilidades de éxito.
- En todo esfuerzo de mejora es sumamente importante que el equipo de mejora siga las mismas reglas que están definiendo, En otras palabras, si deseamos certificar a una organización en CMM el equipo de mejora debe de comportarse como un proyecto CMM.
- La empresa sólo le da el seguimiento a la administración de proyectos, que estos cumplan en tiempo y forma con los requisitos comprometidos. Hay políticas de Software Management pero estas sólo existe en papel, no se aplican por área alguna.

Instrumentación

- Un detalle que lo hace complicado es cuando se quiere estandarizar sobre la marcha, requiere de actividades extras, asignación de tiempos según actividades y alta definición de roles en los procesos, y sobre todo, apoyo de pequeños sistemas que apoyen para eficientizar el tiempo que le dedica personal que no tiene tanto tiempo.
- Es necesario contar con un modelo base que puede ser adquirido o desarrollado, que unifique a las distintas áreas de desarrollo de SW. Esto es, se debe de partir de una

base, aunque esta tenga adecuaciones, pero que sirva de "Jump Start" para todo el personal. En resumen, no reinventar el hilo negro.

- Se batalla mas cuando no se cuenta con una herramienta para registrar las tareas, tales como desarrollos, correcciones, documentar y dar seguimiento de las tareas y pendientes.

Recursos

- Una dificultad es la falta de material "local" humano de apoyo en estos esfuerzos, el reconocimiento por parte de la industria y gobierno (clientes) por el beneficio de los esfuerzos que realizamos las organizaciones por establecer estos modelos de calidad en las empresas. Aunque esta situación cada vez mejora, el costo que involucra pagar un Assessment oficial de CMM en una empresa es casi privativo para las PYMES. El impulso del MoProSoft pudiese ser una alternativa que en México tengamos disponible, debemos alinearnos y apoyar más a los organismos que lo trabajan para el beneficio de todos.
- El mejoramiento del proceso de desarrollo de software es un proceso difícil y no se ha conseguido apoyo económico por parte de la secretaría de economía.

Outsourcing

- Creo que un comentario en General, relacionado al desarrollo de software en esta compañía, es que realmente no es estratégico para la Organización y por lo tanto no se ha incluido como un factor de desarrollo clave dentro de la Organización. Se ha optado por "outsourcar" dichos servicios con empresas calificadas que cubran los estándares de desarrollo normales que marcan este tipo de esfuerzos.
- Se considera importante contar con una certificación como el CMMi, pero no forma parte de la cultura organizacional, pues no forma parte del "core" de la empresa.

Otros...

- Esto es un esfuerzo continuo y que toma muchos años, nosotros llevamos 7 años en esto de la mejora de procesos de software y seguimos encontrando oportunidades de mejora.
- Los usuarios simplemente les interesa una fecha de entrega y que cumpla con las necesidades que ellos tienen. Pero en general, no saben lo que desean y no dan tiempo para poder revisar y definir sus necesidades. Los proyectos son de sistemas y ellos no se integran al mismo a pesar de ser ellos los que usaran el sistema desarrollado.
- Como se menciona, existen esfuerzos aislados traducidos en planeaciones separadas por parte de las áreas involucradas donde cada una define su propio proceso, lo cual, resulta muy difícil de integrar entre dichas áreas, esto se refleja en mejoras individuales pero acrecenta los defectos del proceso global (informal) definido para el manejo de proyectos en el área.
- Es importante además de lo anterior, que escuchemos al cliente, las necesidades se encuentran escuchando y preguntando, generalmente la gente expresa los efectos del problema que quieren resolver, buscamos la mejor forma de resolverlo y la ejecutamos, no nos dejamos llevar por lo que cada quien quiere ver.

CAPITULO 5 – Conclusiones y futuras investigaciones

Conclusiones

Dado el análisis de la información del capítulo anterior, el autor tiene la opinión de que las premisas citadas en el capítulo 1 se comprueban, a continuación se da una breve explicación para la conclusión de cada una de las premisas.

- Podemos ver que los problemas a los que más atacan actualmente las empresas mexicanas son de planeación, seguimiento y requerimientos, los cuales representan las áreas fundamentales del proceso.
- Si sumamos los esfuerzos organizacionales, solo representan el 41%, mientras que los esfuerzos aislados representan el 33% y la falta de intentos el 26%.
- Sobre todo en las empresas que no se dedican al desarrollo de Software, los esfuerzos aislados se dan porque no hay entendimiento claro por parte de los altos ejecutivos de las empresas y por lo tanto no hay soporte gerencial al proceso.
- Como se ve, en la industria del Software la gran mayoría de las empresas son pequeñas en cuanto al número de empleados, también resulta del análisis que a pesar de ser pequeñas no hay colaboración entre ellas, lo que representa un obstáculo para el crecimiento de esta industria en el país.

Las conclusiones obtenidas por esta investigación, mediante el análisis del capítulo anterior, se reflejan en el siguiente resumen de hallazgos, clasificando los puntos importantes por industria y por tamaño.

Hallazgos

Los siguientes hallazgos presentan los factores más importantes reflejados del análisis del capítulo anterior. Estos hallazgos están divididos en 6 grupos, el primero de ellos representa los factores encontrados en todas las empresas, el segundo grupo representa los factores de las empresas de Software grandes, el tercer grupo representa los factores encontrados de las empresas de Software chicas, el cuarto grupo los factores para las empresas de industrias diferentes a las del Software, el quinto para las empresas chicas y el sexto para las empresas grandes.

Para todas las empresas

En este grupo, el único factor universal de éxito que se obtuvo es el de “Obtener el compromiso y soporte de los altos ejecutivos de la dirección”, este aplica a todos los bloques analizados.

Factor de éxito	Factor de Fracaso
<ul style="list-style-type: none">• Obtener el compromiso y soporte de los altos ejecutivos de la dirección	<ul style="list-style-type: none">• No hay factores comunes para todas las categorías
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none">• No hay factores comunes para todas las categorías	<ul style="list-style-type: none">• No hay factores comunes para todas las categorías

Como se puede apreciar, es fundamental obtener el compromiso y soporte de los altos ejecutivos de la dirección, sin este apoyo es difícil que tenga éxito el proceso de mejora.

Para empresas de Software grandes

Como se puede apreciar, los aspectos de capital humano y organizacionales son los más importantes para las empresas de Software grandes, es importante señalar que no se perciben factores significativos para que no hayan intentado mejorar algún área ni factores de fracaso, ya que todos estos presentaron muy bajos promedios.

Factores de éxito	Factores de Fracaso
<ul style="list-style-type: none">• Hacer una buena selección del personal• Centrarse en los beneficios a largo plazo, balanceados por metas a corto plazo• Definir metas realistas y objetivos relevantes• Considerar adecuadamente las habilidades de las personas involucradas	<ul style="list-style-type: none">• No se presentaron factores relevantes para esta categoría
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none">• No se presentaron factores relevantes para esta categoría	<ul style="list-style-type: none">• No lo perciben como una necesidad• Nadie dentro de la empresa, ha tomado ese esfuerzo como suyo

Para empresas de Software chicas

Como se puede apreciar, los aspectos de capital humano y organizacionales también son los más importantes para las empresas de Software chicas. Aquí si se pueden apreciar mas factores dentro del resto de las categorías.

Factores de éxito	Factores de Fracaso
<ul style="list-style-type: none"> • Hacer una buena selección del personal • Promover un cambio a favor del proceso de mejora del desarrollo de software en la opinión de los líderes (formales o informales) • Centrarse en los beneficios a largo plazo, balanceados por metas a corto plazo • Lograr estabilización en los procesos de cambio 	<ul style="list-style-type: none"> • La falta de actividades complementarias tales como gestión de recursos financieros o gestión de personal • El no centrarse en los beneficios a largo plazo, balanceados por metas a corto plazo • Falta de capacitación a empleados e involucramiento de profesionistas recién egresados • No poner al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none"> • Nadie dentro de la empresa, ha tomado ese esfuerzo como suyo • Tienen mucho trabajo • No lo perciben como una necesidad 	<ul style="list-style-type: none"> • Sus entregables están reducidos en tiempo • Nadie dentro de la empresa, ha tomado ese esfuerzo como suyo • Tienen mucho trabajo

Para empresas de otras Industrias

Estas empresas muestran hallazgos un tanto diferentes a las anteriores, aquí lo más importante son las cuestiones de arquitectura organizacional. Lo mismo es cierto para los factores de fracaso. Aquí se resalta la falta de un claro entendimiento del proceso de mejora.

Factores de éxito	Factores de Fracaso
<ul style="list-style-type: none"> • Vincular el cumplimiento de los objetivos del proyecto de mejora a la evaluación del desempeño del personal involucrado • Lograr una alineación entre la alta gerencia y la administración media, así como incorporar este proceso a la estructura organizacional de la empresa • Lograr estabilización en los procesos de cambio • Hacer una asignación clara de tareas, responsabilidades y de toma de decisiones 	<ul style="list-style-type: none"> • No poner al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa • No hacer una asignación clara de tareas, responsabilidades ni de toma de decisiones • No lograr una alineación entre la alta gerencia y la administración media, ni incorporar este proceso a la estructura organizacional de la empresa • No Promover un cambio a favor del proceso de mejora del desarrollo de software en la opinión de los líderes (formales o informales)
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none"> • No hay un entendimiento claro por parte de los altos ejecutivos de la empresa 	<ul style="list-style-type: none"> • Tienen mucho trabajo • No lo perciben como una necesidad • No hay un entendimiento claro por parte de los altos ejecutivos de la empresa

Para empresas chicas (con 500 empleados o menos)

El capital humano nuevamente aparece como uno de los puntos más importante, así como el balanceo de metas a largo y corto plazo. Escasea la capacitación, hay falta involucramiento y de actividades complementarias al proceso.

Factores de éxito	Factores de Fracaso
<ul style="list-style-type: none"> • Centrarse en los beneficios a largo plazo, balanceados por metas a corto plazo • Hacer una buena selección del personal • Promover un cambio a favor del proceso de mejora del desarrollo de software en la opinión de los líderes (formales o informales) • Motivar a las personas a adoptar el proceso de mejora de desarrollo de Software 	<ul style="list-style-type: none"> • El no centrarse en los beneficios a largo plazo, balanceados por metas a corto plazo • La falta de actividades complementarias tales como gestión de recursos financieros o gestión de personal • No poner al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa • Falta de capacitación a empleados e involucramiento de profesionistas recién egresados
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none"> • Nadie dentro de la empresa, ha tomado ese esfuerzo como suyo 	<ul style="list-style-type: none"> • Nadie dentro de la empresa, ha tomado ese esfuerzo como suyo • Sus entregables están reducidos en tiempo • Tienen mucho trabajo

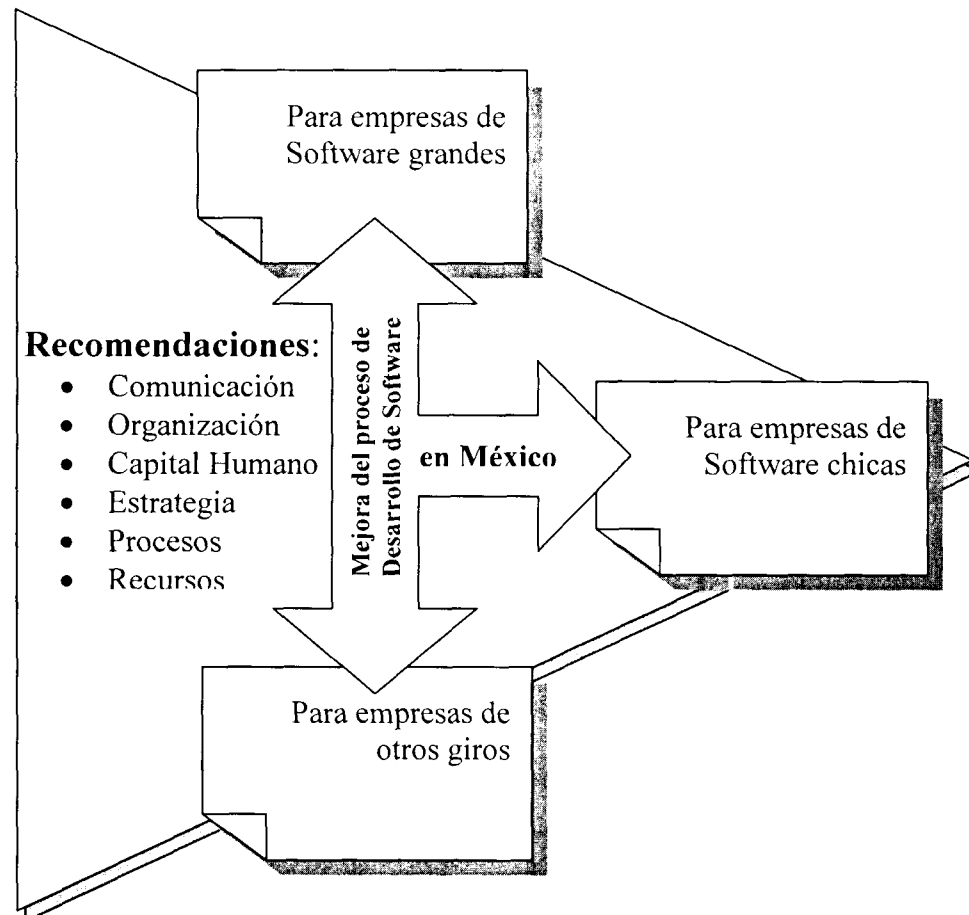
Para empresas grandes (con más de 500 empleados)

El éxito en este grupo, está enfocado a la asignación y cumplimiento de las actividades y el balanceo de metas. El fracaso se logra cuando no está claro lo anterior, también se da por un mal manejo de recursos, tanto financieros como de personas.

Factores de éxito	Factores de Fracaso
<ul style="list-style-type: none"> • Centrarse en los beneficios a largo plazo, pero balanceados por metas a corto plazo • Lograr estabilización en los procesos de cambio • Vincular el cumplimiento de los objetivos del proyecto de mejora a la evaluación del desempeño del personal involucrado • Hacer una asignación clara de tareas, responsabilidades y de toma de decisiones 	<ul style="list-style-type: none"> • No hacer una asignación clara de tareas, responsabilidades ni de toma de decisiones • No Promover un cambio a favor del proceso de mejora del desarrollo de software en la opinión de los líderes (formales o informales) • El mal manejo de tiempos y actividades; de asignación de personal y recursos, incluyendo al presupuesto • No lograr una alineación entre la alta gerencia y la administración media, ni incorporar este proceso a la estructura organizacional de la empresa
Factores por los que no intentan mejorar algún área de proceso	Factores para no institucionalizar los esfuerzos
<ul style="list-style-type: none"> • No ven los beneficios reales de este proceso de mejora 	<ul style="list-style-type: none"> • No lo perciben como una necesidad • No ven los beneficios reales de este proceso de mejora • No lo consideran parte de la estrategia de la empresa

Recomendaciones

A continuación se presentan una serie de recomendaciones propuestas por el autor, estas recomendaciones, como se puede ver en el siguiente modelo, a una de seis categorías - Comunicación, Organización, Capital Humano, Estrategia, Procesos y Recursos- y cada una puede aplicar a las empresas de Software chicas y grandes o a las empresas de otras industrias.



Estas recomendaciones surgen de los hallazgos antes mencionados y del análisis realizado en el capítulo 4, en donde se ve la situación actual de estas empresas en cuanto al proceso.

Del modelo anterior, surge el siguiente listado con las recomendaciones antes mencionadas. Al final del listado se presenta una breve descripción de cada recomendación, para cualquier duda o comentario de esta lista siéntase libre de contactarme, en el Anexo B están los puntos de contacto.

Listado de Recomendaciones

En el siguiente listado, aparece una para cada caso donde aplique. La empresa debe sentirse libre de seleccionar cada recomendación que aplique.

Recomendación \ Tipo de empresa	Empresas de Software chicas	Empresas de Software grandes	Otras empresas
---------------------------------	-----------------------------	------------------------------	----------------

Comunicación

Asegurar el compromiso y apoyo de los altos mandos de la empresa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Definir riesgos y oportunidades de contar o no con un proceso de mejora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Establecer una comunicación constante con los altos ejecutivos, respecto a los avances obtenidos	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Mantener informados a los participantes de los cambios que se presenten durante los proyectos	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Capital Humano

Crear un equipo que se dedique exclusivamente al proceso de mejora	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Seleccionar adecuadamente el personal del proceso de mejora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Recibir más capacitación acerca del proceso	<input checked="" type="checkbox"/>		
Incorporar al equipo de mejora tanto expertos como profesionistas recién egresados	<input checked="" type="checkbox"/>		
Considerar adecuadamente las competencias de cada persona involucrada en el proceso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Estrategia

Centrarse en los beneficios a largo plazo balanceadas con metas a corto plazo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Buscar una certificación del proceso	<input checked="" type="checkbox"/>		
Buscar alianzas entre las empresas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ubicar un área encargada del proceso de mejora dentro de la estructura organizacional	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Recomendación \ Tipo de empresa	Empresas de Software chicas	Empresas de Software grandes	Otras empresas
---------------------------------	-----------------------------	------------------------------	----------------

Procesos

Fomentar el proceso de mejora con los líderes	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Realizar auditorias internas y externas del proceso	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Institucionalizar los esfuerzos de mejora	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Recursos

Reforzar actividades complementarias tales como la administración del personal	<input checked="" type="checkbox"/>		
Mejorar la administración de recursos relacionados al proceso	<input checked="" type="checkbox"/>		

Organización

Vincular el cumplimiento de los proyectos a la evaluación del desempeño del personal			<input checked="" type="checkbox"/>
Hacer una asignación clara de tareas y responsabilidades			<input checked="" type="checkbox"/>

Nota del autor: En los puntos donde hay espacios en blanco, no necesariamente significa que esté mal llevar a cabo esta práctica, sino que los hallazgos de esta investigación no la reflejan como una necesidad inmediata o bien, el bloque de empresas ya sigue dicha recomendación.

Descripción general de las recomendaciones

Comunicación

Asegurar el compromiso y apoyo de los altos mandos de la empresa: Antes de empezar un proyecto de este tipo, asegurarse de contar con el compromiso y apoyo de los altos mandos dentro de la empresa, involucrarlos desde el principio y mantenerlos al tanto de los avances y resultados del proceso de mejora para que mantengan el interés.

Definir riesgos y oportunidades de contar o no con un proceso de mejora: Identificar y comunicar los riesgos y oportunidades asociados de contar o no con un proceso de mejora.

Establecer una comunicación constante con los altos ejecutivos, respecto a los avances obtenidos: Establecer una comunicación estrecha con los altos ejecutivos de la empresa, acerca de los avances que se tienen en los proyectos actuales gracias al proceso de mejora. Si los ejecutivos no tienen un claro entendimiento del proceso, las probabilidades de tener éxito en los objetivos planteados disminuyen.

Mantener informados a los participantes de los cambios que se presenten durante los proyectos: Mantener informados a los participantes en tiempos difíciles para tratar de lograr estabilización en los procesos de cambio. Mejorar la Administración de la configuración.

Capital Humano

Crear un equipo que se dedique exclusivamente al proceso de mejora: Designar a un equipo que se dedique exclusivamente al proceso de mejora, integrarlo de acuerdo a las competencias de los miembros, mantenerlo motivado y capacitado. Llevar a cabo revisiones periódicas con dicho equipo. Administrar adecuadamente los recursos financieros y humanos asociados con el equipo.

Seleccionar adecuadamente al personal del proceso de mejora: Hacer una mejor selección de las personas que colaborarán en el proceso de mejora de desarrollo de Software.

Recibir más capacitación acerca del proceso: Buscar mucha más capacitación en mejoramiento del proceso de desarrollo de Software. Si no hay capacitación, no hay mejoramiento.

Incorporar al equipo de mejora tanto expertos como profesionistas recién egresados: Es necesario involucrar tanto expertos como a personas recién egresadas.

Considerar adecuadamente las competencias de cada persona involucrada en el proceso: Identificar bien los perfiles competitivos de cada uno de los miembros que formarán parte del proceso de mejora y en base a esto asignar las responsabilidades

Estrategia

Centrarse en los beneficios a largo plazo balanceadas con metas a corto plazo: Buscar beneficios a largo plazo usando metas a corto plazo, buscar un crecimiento sustentable.

Buscar una certificación del proceso: Buscar una certificación para tener mas oportunidad de vender, tanto a nivel nacional como internacional. Se necesita la búsqueda de una certificación que avale la madurez de su proceso.

Buscar alianzas entre las empresas: Buscar establecer relaciones de colaboración con los propios competidores para lograr alcanzar proyectos más grandes. Esta visión de colaboración nos permite un crecimiento también a nivel país.

Ubicar un área encargada del proceso de mejora dentro de la estructura organizacional: Es necesario involucrar un área que se dedique específicamente a llevar el proceso de mejora dentro de la empresa.

Procesos

Fomentar el proceso de mejora con los líderes: Promover el proceso de mejora con los líderes (formales o informales), asegurando una amplia comprensión de los beneficios del mismo, si los líderes están motivados, el resto de las personas lo adoptarán más fácilmente.

Realizar auditorias internas y externas del proceso: La valoración del proceso de desarrollo de Software es fundamental. Se necesita considerar mas seriamente hacer valoraciones internas y externas.

Institucionalizar los esfuerzos de mejora: Los esfuerzos de mejora tienen darse a nivel organizacional para garantizar la permanencia del conocimiento del proceso y la repetibilidad del éxito.

Recursos

Reforzar actividades complementarias tales como la administración del personal: Actividades complementarias tales como la administración de recursos financieros y humanos. La administración de proyectos juega un papel fundamental en esta parte.

Mejorar la administración de recursos relacionados al proceso: Mejorar la Administración de proyectos para contemplar el manejo de actividades, personal y recursos financieros.

Organización

Vincular el cumplimiento de los proyectos a la evaluación del desempeño del personal: Vincular directamente el cumplimiento de los objetivos de los proyectos de mejora a la evaluación del desempeño del personal involucrado, esto hará que el proceso y sus integrantes se sientan mas motivados.

Hacer una asignación clara de tareas y responsabilidades: Hacer una asignación clara de tareas y responsabilidades, que cada quien sepa lo que tiene que hacer. Llevar también un registro adecuado de dichas asignaciones.

Futuras investigaciones

A continuación y dado lo observado en esta investigación se proponen tópicos a desarrollar relacionados con esta tesis.

- Una exploración en la pequeña y mediana empresa del país para determinar el uso que se está haciendo de metodologías ágiles para contribuir en la mejora del proceso de desarrollo de software
- Un estudio para determinar los factores que han llevado a tener éxito en el proceso de mejora de desarrollo de software en las empresas certificadas como CMM nivel 5 en México.
- Un estudio exploratorio para encontrar los resultados alcanzados en las empresas mexicanas que hayan utilizado otros modelos de mejora como ISO 15504, TSP/PSP y particularmente el modelo mexicano MoProSoft.
- Sería interesante también un estudio para corroborar la participación del gobierno para impulsar la industria de software, resolviendo preguntas tales como: ¿El gobierno cumplió con lo que había ofrecido a las empresas participantes? ¿Realmente los incentivos se consideran los adecuados para impulsar esta industria? ¿Qué pueden hacer las empresas para mejorar la participación del gobierno?
- Un estudio que muestre una correlación costo-beneficio al mejorar el proceso de desarrollo de software en las empresas mexicanas, que refleje las ventajas económicas a corto y largo plazo del proceso de mejora.
- La transmisión del conocimiento para mejorar el proceso de desarrollo de Software en las empresas pequeñas, evidenciando la existencia o falta de mecanismos para dicha transmisión.
- El rol de la capacitación en la industria del Software como búsqueda de mejoramiento del proceso de desarrollo de Software. ¿Cuánta capacitación es necesaria? ¿Qué temas a abordar son importantes para el mejoramiento? ¿Cómo se puede obtener apoyo financiero para esta capacitación?
- Capitalización por parte de la organización del conocimiento generado en esfuerzos individuales, formas de apalancar y reutilizar el conocimiento generado por el Capital Humano.
- La necesidad real de la industria de Software de mejorar sus procesos y como medirlos ¿Por qué es necesario mejorar este proceso? ¿Qué formas hay de medir el proceso de Software en la industria mexicana?
- Como se descompone la administración de proyectos en la industria de Software mexicana. ¿Qué se está llevando a cabo para llevar una buena Administración de proyectos de Software en México? ¿Cuáles son las mejores prácticas en Administración de proyectos aplicables a nuestro país?

Anexos

Anexo A. Cuestionario

Encuesta para obtener los factores de éxito y/o fracaso en los intentos de mejora del proceso de desarrollo de software enfocados a las áreas de proceso del nivel 2 de certificación del CMMi

Sección I

A continuación se le pide que conteste cuestiones generales con respecto a su empresa.

Su puesto actual:

- Líder de proyectos
- Jefe de informática
- Gerente de sistemas
- Jefe de aseguramiento de calidad
- Gerente General
- Otro:

El giro de la empresa:

- Acero y papel
- Agricultura, pesca
- Alimentos y bebidas
- Almacenaje y distribución
- Bancos, finanzas
- Comercio
- Minería, petróleo y gas
- Manufactura
- Química
- Seguros y bienes raíces
- Servicios y salud
- Software y consultoría
- Construcción
- Telecomunicaciones
- Educación

Transportación
Electricidad y agua
Gobierno
Otro:

¿Cuál es el número de empleados en la empresa?

menos de 100
de 100 a 500
de 501 a 1000
de 1001 a 5000
de 5001 a 10000
mas de 10000

¿Cuántas personas en la empresa se dedican específicamente al desarrollo de software?

menos de 5
de 5 a 10
de 11 a 50
de 51 a 100
mas de 100

¿Cuántos años tiene de experiencia la empresa desarrollando software?

menos de 1 año
entre 1 y 10 años
entre 11 y 20 años
entre 21 y 30 años
mas de 30 años

¿Ha recibido la empresa algún tipo de capacitación relacionada con el mejoramiento del proceso de desarrollo de software?

no
sí, especifique:

¿Ha participado la empresa en algún tipo de valoración de su proceso de desarrollo de software, evaluación de capacidades u otro tipo de examinación de dicho proceso de desarrollo de software? (por ejemplo, CMM, ISO, etc)

no

sí, especifique:

Sección II

Ahora se le pide que conteste de acuerdo a los intentos de mejora en el proceso de desarrollo de software, en áreas de proceso especificadas por el Modelo Integrado de la Capacidad de la Madurez (CMMi) en su nivel 2 de certificación.

Según el CMMi, existen ciertas áreas de proceso que se deben trabajar primero para su mejoramiento en el proceso de desarrollo de software en las empresas, cada una de ellas identifican un conjunto de actividades relacionadas que, cuando se llevan a cabo de forma colectiva, se alcanzan ciertas metas consideradas importantes para establecer capacidades de proceso para el nivel 2 en dicho modelo de madurez.

En esta investigación, se busca determinar los factores de éxito y de fracaso en los esfuerzos o falta de ellos en estas áreas de proceso.

¿Se han hecho esfuerzos por mejorar alguna de las siguientes áreas en el proceso de desarrollo de software en la empresa? (Para una explicación mas detallada de estas áreas, consulte el apéndice A)

Marque en todos los renglones una opción:

Tabla 1			
Área de proceso	Aún no	Sí, pero en esfuerzos aislados o parciales	Sí, a nivel organizacional
Administración de requerimientos			
Planeación de proyectos de Software			
Supervisión y Control de proyectos de Software			
Administración de contratación de proveedores			
Análisis y mediciones			
Aseguramiento de la calidad de procesos y productos			
Administración de la configuración			

- Si en la tabla anterior, marcó en la columna "Aún no" alguna de las áreas de proceso, ¿Qué los ha limitado o impedido para buscar una mejora en esas áreas?

	Totalmente en desacuerdo			Totalmente de acuerdo	
	1	2	3	4	5
No existe un entendimiento claro por parte de los altos ejecutivos de la empresa					
Los beneficios reales no se ven					
No se percibe como una necesidad					
Se percibe como muy costoso					
Se considera que no aplica a mi organización					
Tenemos mucho trabajo					
No es parte de la estrategia de la empresa					
Nuestros usuarios son demasiado exigentes					
Nadie ha tomado ese esfuerzo como suyo					
Eso es demasiado técnico para las actividades de nuestra empresa					
Nuestros entregables están reducidos en tiempo de entrega					
Otro:					

- Si en la tabla 1, eligió en alguna de las áreas de proceso, "Sí, pero en esfuerzos aislados o parciales" - ¿Por qué se han realizado de forma aislada y no como un esfuerzo a nivel organizacional?

	Totalmente en desacuerdo			Totalmente de acuerdo	
	1	2	3	4	5
No existe un entendimiento claro por parte de los altos ejecutivos de la empresa					
Los beneficios reales no se ven					
No se percibe como una necesidad					
Se percibe como muy costoso					
Se considera que no aplica a mi organización					
Tenemos mucho trabajo					
No es parte de la estrategia de la empresa					
Nuestros usuarios son demasiado exigentes					
Nadie ha tomado ese esfuerzo como suyo					

Eso es demasiado técnico para las actividades de nuestra empresa				
Nuestros entregables están reducidos en tiempo de entrega				
Otro:				

- Si en la tabla 1, eligió para alguna de las áreas de proceso, "Si, a nivel organizacional" ¿Ha sido en busca de alguna certificación en Ingeniería de Software?

no
 sí, ¿cuál o cuáles certificaciones?

- Si en la tabla 1, para alguna de las áreas de proceso eligió "Si, pero en esfuerzos aislados" o "Si, a nivel organizacional"

a) Si alguno de los esfuerzos no cumplió con los objetivos, ¿Por qué considera que esto ocurrió?

	Totalmente en desacuerdo			Totalmente de acuerdo	
	1	2	3	4	5
No se ha puesto al proceso de mejora del desarrollo de software en una perspectiva adecuada dentro de la empresa					
No se obtuvo el compromiso ni el soporte de los altos ejecutivos de la dirección					
No se percibió un cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)					
No hubo disposición de tiempo, recursos, falta de involucramiento, comunicación y entusiasmo por parte del equipo encargado					
No hubo una asignación clara de tareas, responsabilidades ni toma de decisiones					
No se estimó correctamente el nivel actual de la empresa					
No se consideraron las habilidades de las personas involucradas					
No hubo mejora percibida en la calidad de los intentos realizados					
No se definieron metas realistas o claras, ni se establecieron los objetivos relevantes					
La organización no se centró en los beneficios a largo plazo.					

balanceados por metas a corto plazo					
No se obtuvo una alineación entre la alta gerencia y la administración media o no se incorporó este proceso a la estructura organizacional					
No hubo capacitación a empleados ni se involucraron a los profesionistas recién egresados					
No hubieron actividades complementarias tales como gestión de recursos financieros o gestión de personal.					
Inexistencia de mecanismos para captar información y experiencias útiles para el proceso de mejora del desarrollo de software					
Desconocimiento de los cambios en los gustos y necesidades de los clientes					
No se establecieron relaciones de colaboración con los competidores, proveedores u otros agentes (como instituciones públicas)					
Mal manejo de tiempos y actividades; de asignación de personal y recursos; del presupuesto.					
Mala selección del personal					
Falta de desarrollo de reportes y explotación de información					
Estamos esperando a que la industria decida un modelo					
Otro:					

b) Si alguno de los esfuerzos cumplió con los objetivos, ¿Qué actividades considera que los han llevado al éxito?

	Totalmente en desacuerdo			Totalmente de acuerdo	
	1	2	3	4	5
Se puso al proceso de mejora del desarrollo de software en una perspectiva adecuada					
Se obtuvo el compromiso y soporte de los altos ejecutivos de la dirección					
Se demostró una disposición de tiempo, recursos, involucramiento, comunicación y entusiasmo por parte del equipo encargado					
Cambio a favor del proceso de mejora del desarrollo de software de la opinión de los líderes (formales o informales)					
Hubo una asignación clara de tareas, responsabilidades y toma de decisiones					

Se estimó correctamente el nivel actual de la empresa				
Se consideraron adecuadamente las habilidades de las personas involucradas				
Se reforzó el entendimiento de los procesos para la mejora				
Se definieron metas realistas o claras, y se establecieron los objetivos relevantes				
Hubo estabilización en los procesos de cambio				
La organización se centró en los beneficios a largo plazo, balanceados por metas a corto plazo				
Se obtuvo una alineación entre la alta gerencia y la administración media y/o se incorporó este proceso a la estructura organizacional.				
Se motivó a las personas para adoptar el proceso de mejora del desarrollo de software				
Hubo capacitación a empleados y se involucraron a los profesionistas recién egresados				
Hubieron actividades complementarias tales como gestión de recursos financieros o gestión de personal.				
Existencia de mecanismos para captar información y experiencias útiles para el proceso de mejora del desarrollo de software				
Conocimiento de los cambios en los gustos y necesidades de los clientes				
Se establecieron relaciones de colaboración con los competidores, proveedores u otros agentes (como instituciones públicas)				
Buen manejo de tiempos y actividades; asignación de personal y recursos; del presupuesto				
Buena selección del personal				
Desarrollo de reportes y explotación de información				
El cumplimiento de los objetivos del proyecto de mejora se vinculó a la evaluación del desempeño del personal involucrado				
Otro:				

- Si en la *tabla 1*, eligió ya sea "Sí, pero en esfuerzos aislados" o "Sí, a nivel organizacional" y existen comentarios adicionales que desee aportar sobre intentos que hayan funcionado o fracasado en el mejoramiento del proceso de desarrollo de software, por favor hágalo en el siguiente espacio:

Apéndice A

A continuación se da una breve descripción del propósito de cada área de proceso.

Administración de requerimientos

El propósito de esta área de proceso en el desarrollo de software es administrar los requerimientos de los productos de los proyectos de desarrollo de software, así como los componentes de dichos productos. También identificar las inconsistencias entre esos requerimientos y la planificación y productos de trabajo de los proyectos de desarrollo de software.

Planeación de proyectos de Software

Aquí la intención es establecer y mantener una planificación que define las actividades de los proyectos de desarrollo de software.

Supervisión y Control de proyectos de Software

El objeto de esta área de proceso es proveer de un entendimiento del progreso del proyecto para que las acciones correctivas apropiadas puedan llevarse a cabo cuando el rendimiento de este se desvíe significativamente de lo planeado.

Administración de contratación de proveedores

El objetivo es administrar la adquisición de productos de proveedores para los cuales existe un contrato o acuerdo formal.

Análisis y mediciones

Esta área de proceso tiene como finalidad desarrollar y preservar una capacidad de medición que sea usada para ayudar en la administración de las necesidades de información.

Aseguramiento de la calidad de procesos y productos

El propósito de esta área de proceso es proveer de soporte y administración con un enfoque objetivo en los procesos y los productos de trabajo asociados a estos.

Administración de la configuración

Establecer y mantener la integridad de los productos de trabajo utilizando configuración en la identificación, control, estatus de contabilidad y auditorías.

Anexo B. Datos de contacto

Para cualquier aportación, duda o comentario, con gusto pueden contactarme a cualquiera de las siguientes direcciones electrónicas:

jalvarado@jaamssoftware.com,
jaamssoftware@hotmail.com

O por medio del web site:

<http://spi.jaamssoftware.com>

Curriculum actualizado:

<http://curriculum.jaamssoftware.com>

Referencias

Almeraz, Elizabeth; Pérez-Vargas, Mariana Comparación práctica de los modelos de madurez SW-CMM vs. CMMi Septiembre del 2002, AVANTARE, <http://www.avantare.com/>

AMITI Esquema de apoyo gubernamental a la industria de software Abril del 2001, AMITI, <http://www.amiti.org.mx/biblioteca.asp#documentos>

Anónimo Certificación Internacional de Productos Orgánicos, Sin fecha, SGS Uruguay Limitada, <http://www.sgs.com.uy/news04.html>

Anónimo interfase-Jalisco, 2 de Febrero de 1998, Reforma

Anónimo SGS-PANAMA: 15 años certificando a las empresas panameñas, Sin fecha, Publigraf Internacional, S.A., <http://publigraf.com/sgs/>

Anónimo Why is international standardization needed? Sin fecha, ISO, <http://www.iso.org/iso/en/aboutiso/introduction/whynEEDED.html>

Anónimo. interfase-Jalisco, 2 de Febrero de 1998, Reforma

Bancomext Desarrollo de software: una oportunidad para México, 2001

Bancomext Industria del software análisis y recomendaciones, Abril del 2001, <http://www.amiti.org.mx/biblioteca.asp#documentos>

Bancomext Recomendaciones para acceder al mercado, 2001, <http://www.bancomext.com/Bancomext/Template/Nacional/default.jsp?seccion=2193>

Bancomext. Desarrollo de software: una oportunidad para México, 2001

Bancomext. Industria del software análisis y recomendaciones, Abril del 2001, <http://www.amiti.org.mx/biblioteca.asp#documentos>

Bancomext. Recomendaciones para acceder al mercado, 2001, <http://www.bancomext.com/Bancomext/Template/Nacional/default.jsp?seccion=2193>

Booch, Grady Object-Oriented Analysis and Design with Applications 1994, Addison Wesley

Briand, Lionel; Morasca, Sandro Property-based Software Engineering Measurement 1996, IEEE, <http://www2.umassd.edu/SWPI/ESEG/cs-tr-3368.pdf>

Corral, Solórzano Estrategias para garantizar la calidad del software en telecomunicaciones, Sin fecha, Telefónica investigación y desarrollo,

<http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol51/estrategias/estrateg.html>

Côrtes, Mario L. Modelos de Qualidade de SW 1998, UNICAMP

Cristales Pavón, Asley Alberto Estudio exploratorio de la aplicación de actividades de SQA en el proceso de desarrollo de Software para lograr el éxito de los sistemas, en la cd. de Monterrey 1996, ITESM

Cuella Lovelle, Juan Manuel La complejidad del desarrollo de software 2001, Universidad de Oviedo, <http://di002.edv.uniovi.es/~cueva/assignaturas/doctorado/2001/complejidad.pdf>

Díaz Olavarrieta, Arnoldo Un Estándar Internacional para la Evaluación del Proceso de Desarrollo de Software ISO/SPICE Marzo de 1996, Universidad EAFIT, <http://dis.eafit.edu.co/~ogarcia/st123/isospic2.htm>

ESSI-SCOPE The International Guideline : ISO 9000-3 Sin fecha, <http://www.cse.dcu.ie/essiscope/sm5/approach/iso9-3-1.html>

Fernández y Fernández, Carlos A. Ingeniería de Software Septiembre del 2000, Instituto de Electrónica y Computación - Universidad Tecnológica Mixteca, <http://nuyoo.utm.mx/~caff/doc/IntroIngdeSoftwareITAO.pdf>

Frangos, Stelios Security Through SPI Methodologies, 9 de Marzo de 1998, ABS Integrated Services Hellas, <http://epic.onion.it/workshops/w05/slides13/>

García Romero, Claudia Ivette El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicana de Software, 2001, Universidad de las Américas-Puebla, http://mailweb.udlap.mx/~tesis/lis/garcia_r_ci/

Granja Álvarez, Juan Carlos Un horizonte en el ámbito de la Calidad del Software Mayo del 2000, NOVATICA, <http://www.ati.es/novatica/2000/145/juagra-145.pdf>

Hinojosa Calderón, Jesús Eduardo Características de las empresas que utilizan outsourcing de proyectos y de personal de tecnologías de información 2002, ITESM

Hoff, Anna; Wikström, Anna BOOTSTRAP Software Process Assessment and Improvement 9 de Diciembre del 2001, Blekinge Institute of technology.

Humphrey, Watts S. A discipline for Software Engineering Octubre del 2002, SEI Series in software engineering, Addison Wesley

Humphrey, Watts S. Managing the Software process, 1990, Addison Wesley

Kalyanpur, Nikhil 15 Success Factors for Software Process Improvement 2001, qaiindia, http://www.qaiindia.com/SEPG_minisite/Past_papers/pdf_files/Nikhil_Kalyanpur-15_Success_Factors_for_Software_Process_Improvement.pdf

Marcos, María del Socorro, Manual para la elaboración de Tesis I MATI-PGIT, Trillas 2ª edición, México, 2001

Milanés, José Manuel Midiendo la Complejidad del Software Sin fecha, SETI, <http://www.gruposeti.com/Recursos/PDFs/COMPLEJIDAD.pdf>

Mullins, Robert; Simonson, Sharon. Mexico shows it wants to play tech by wooing valley firms, 25 de Octubre del 2002, Silicon Valley / San José Business Journal, <http://sanjose.bizjournals.com/sanjose/stories/2002/10/28/story8.html>

Oktaba, Hanna. Modelo de Procesos para la Industria de Software, Mayo del 2003

Paulk, Mark C. Software Process Proverbs, 1997, Software Engineering Institute Carnegie Mellon University

Paulk, Mark C. Using the Software CMM in Small Organizations, 1999, Software Engineering Institute Carnegie Mellon University

Paulk, Mark C.; Curtis, Bill; Crisis, Mary Beth; Weber, Charles V. Capability Maturity Model for Software, Version 1.1, Technical Report, Febrero de 1993, Software Engineering Institute, <http://www.sei.cmu.edu/cmm/>

Paulk, Mark C.; Weber, Charles V. The Capability Maturity Model: Guidelines for improving the software process 1995, Addison Wesley

Paulk, Mark C.; Weber, Charles V.; Crisis, Mary Beth; Garcia, Suzanne M.; Bush, Marilyn Key Practices of the Capability Maturity Model, Version 1.1, Febrero de 1993, Software Engineering Institute Carnegie Mellon University

Phillips, Mike CMMI V1.1 and Appraisal Tutorial 2004, Software Engineering Institute Carnegie Mellon University <http://www.sei.cmu.edu/cmmi/presentations/euro-sepg-tutorial/sld001.htm>

Pihlava, Sakari. A Process Improvement Experience in Small PC Software Companies, 1996, Helsinki university of technology, <http://www.soberit.hut.fi/sw-workmate/potku/thesis.html>

Ridao, Marcela Modelado de Anomalías mediante Escenarios 2002, INTIA, Fac. de Ciencias Exactas - Univ. Nacional del Centro de la Provincia de Buenos Aires, [http://www.inf.puc-rio.br/wer02/zip/Modelado%20anomalias\(10\).pdf](http://www.inf.puc-rio.br/wer02/zip/Modelado%20anomalias(10).pdf)

Salinas Sánchez, Rafael. Esquema financiero para el apoyo a la industria del software, Bancomext, <http://www.bancomext.com/Bancomext/Template/Nacional/default.jsp?seccion=962>

Secretaría de economía. Programa para el Desarrollo de la Industria de Software, Abril del 2001, Secretaría de economía

Software.net.mx. Convocatoria a Pruebas Controladas Moprosoft,
http://www.software.net.mx/NR/exeres/00000061obzvzznrwwirvdj/noticia.asp?NRMODE=Published&NRORIGINALURL=%2fportal%2fesp%2fdesarrollador%2fnoticias%2fconvoca_moprosoft%2ehtm&NRNODEGUID=%7b61741ECA-702A-424F-9CDC-7D35516538A6%7d&NRQUERYTERMINATOR=1&cookie%5Ftest=1, 2004,
Software.net.mx

Software.net.mx. MoProSoft - Modelo de procesos para la industria de software,
<http://www.software.net.mx/portal/esp/desarrollador/moprosoft.htm>, 2003,
Software.net.mx

Stelzer, Dirk; Mellis, Werner Success Factors of Organizational Change in Software Process Improvement Mayo de 1999, Universitaet zu Koeln, <http://www.informatik.uni-koeln.de/wininfo/prof.mellis>

Villaruel Acevedo, Rodolfo Mejoramiento del proceso de gestión de configuración de software Mayo del 2002, Red Chilena para el Mejoramiento del Proceso de Software - SPIN-CHILE, http://www.spin-chile.cl/invitaciones/mail/2002/SPIN_Tutorial3_2002.htm

Weitzenfeld, Alfredo Ingeniería de Software Orientada a Objetos con UML, Java e Internet Abril del 2003, ITAM,
<http://cannes.rhon.itam.mx/Alfredo/Espaniol/Publicaciones/MINT/Intro.pdf>

Zahran, Sami Software Process Improvement: Practical guidelines for business success 1998, Addison Wesley

