

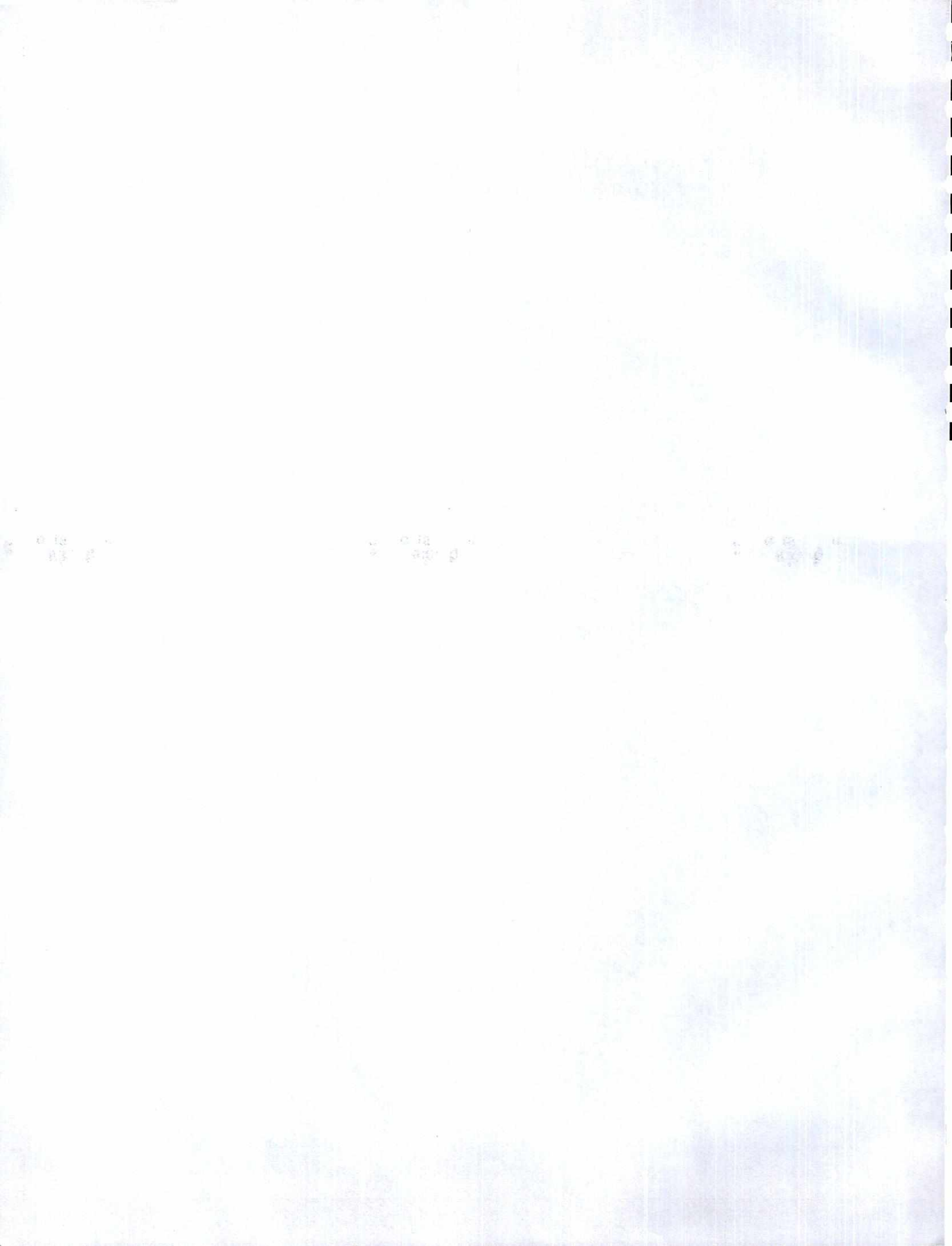
UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

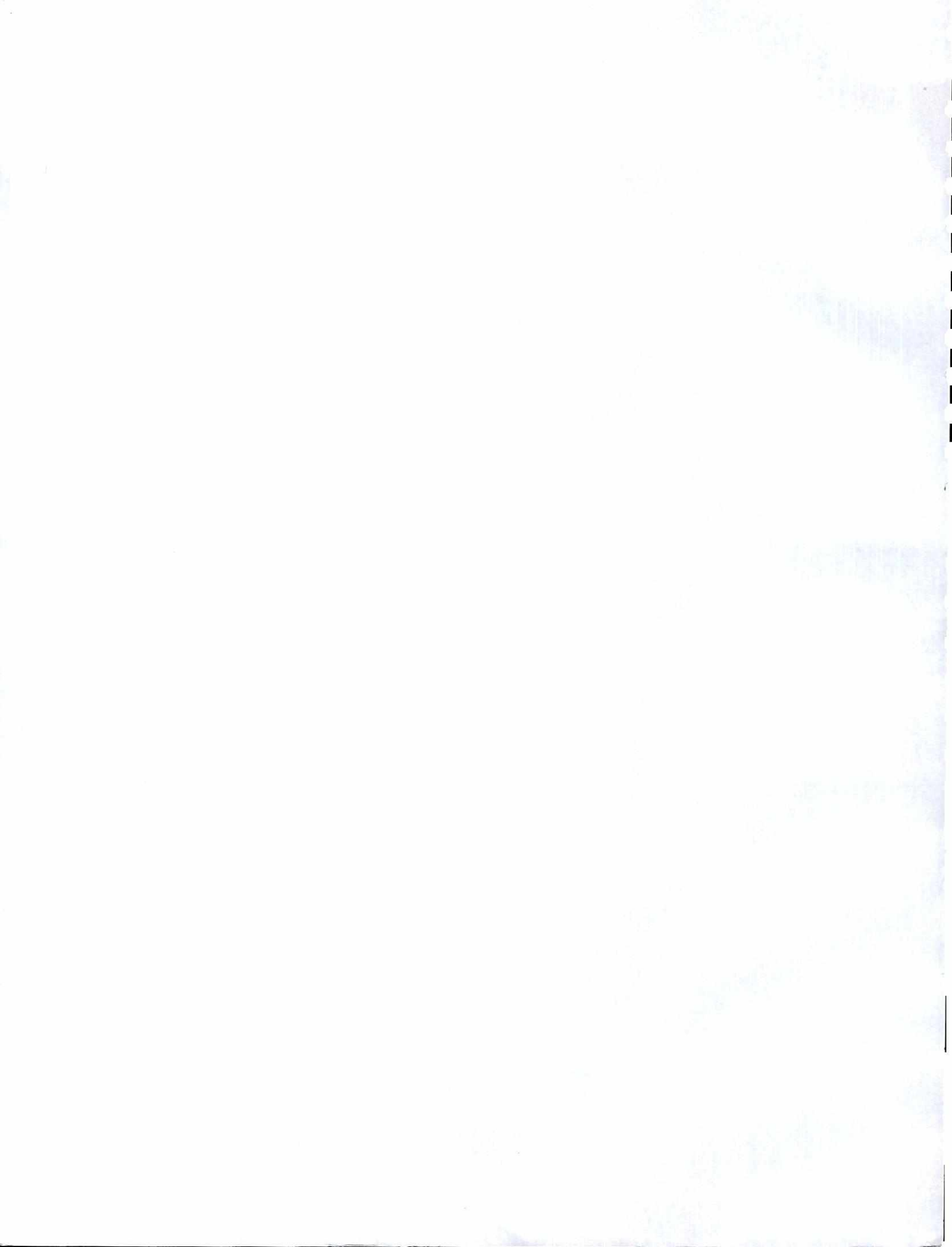
Propuesta de mecanismos eficientes para el
manejo de requerimientos con clientes
internos para el desarrollo y mantenimiento de
sistemas de información



Modelo de trabajo profesional presentado por Roberto Antonio Gómez
Díaz para optar al grado de Master en Tecnología y Ciencias de la
Computación

Guatemala de la Asunción
2003





Propuesta de mecanismos eficientes para el
manejo de requerimientos con clientes
internos para el desarrollo y mantenimiento de
sistemas de información

一 姓 名 姓 名

二 姓 名 姓 名

三 姓 名 姓 名

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

Propuesta de mecanismos eficientes para el
manejo de requerimientos con clientes
internos para el desarrollo y mantenimiento de
sistemas de información

Modelo de trabajo profesional presentado por Roberto Antonio Gómez
Díaz para optar al grado de Master en Tecnología y Ciencias de la
Computación

Guatemala de la Asunción
2003

1 "17 6"

1 "17 6"

1 "17 6"

Vo.Bo.

Arturo

Ing. Arturo Rivera Perezgil, MSE

Tribunal examinador:

Rolando Rodríguez Lima

Ing. Rolando Rodríguez Lima, MSc

Arturo

Ing. Arturo Rivera Perezgil, MSE

Patricia Castillo

Ing. Patricia Castillo Avila, MSc

Guatemala, 26 de mayo de 2003



DEDICATORIA

Para mi esposa María Eugenia
y mis hijos:
María Isabel,
Mario Roberto,
Rafael Antonio y
Fernando Antonio.

100

100

100

PREFACIO

A lo largo de mi carrera profesional, he observado el surgimiento de varias metodologías que han revolucionado la industria. Sin embargo, la aplicación de ciertas técnicas ha traído algunos sinsabores; otras no se aplican tan claramente y otras experiencias, en cambio, fueron muy exitosas. Al reflexionar sobre el éxito y fracaso de estas prácticas surgen varias hipótesis respecto a lo que determina el resultado en cada caso. Entre éstas se puede mencionar educación, entrenamiento, aplicación, novedad, etc. Este trabajo es el resultado de la observación, discusión, reflexión acerca de los fenómenos que en muchas ocasiones se han manifestado.

Quiero expresar mi agradecimiento en primer lugar a mi familia por su apoyo durante la etapa oscura del trabajo diario. Agradezco también la colaboración de clientes internos que me han hecho percatarme de lo que es importante y lo que no lo es. A mis colegas, con los que muchas veces discutimos y trabajamos la solución de estos problemas, y a mis profesores que me compartieron sus valiosos conocimientos. En particular, deseo expresar mi gratitud al asesor de este trabajo, Arturo Rivera, MSE, con quien discutí ampliamente sobre la aplicabilidad de algunas técnicas y siempre tuvo la cortesía de oírme y motivarme para que buscara una solución. También agradezco la colaboración e interés que Patricia Castillo; MS, mostró al apoyar el desarrollo de este trabajo. A todos muchas gracias.

18 82 3

19 82 3

20 82 3

RESUMEN

La adecuada selección de modelos de mejoramiento no es una cuestión trivial para organizaciones pequeñas. Este trabajo presenta el problema orientado al adecuado manejo de requerimientos con clientes internos para el desarrollo y mantenimiento de sistemas de información en pequeñas y medianas organizaciones guatemaltecas. Se concluye que las técnicas deben considerar no sólo el tamaño de la organización y del equipo de desarrollo, sino que también, es imprescindible tomar en cuenta aspectos culturales, experiencia de dominio y factores de control de calidad. Se hace énfasis en herramientas que fomenten la comunicación y negociación, de baja tecnología, bajo costo y perdurables en todo el proceso de mejoramiento organizacional.

100

100

100

INDICE

	Página
PREFACIO _____	V
RESUMEN _____	VII
LISTA DE CUADROS _____	XI
LISTA DE ILUSTRACIONES _____	XIII
I. INTRODUCCIÓN _____	1
II. DEFINICIÓN DEL PROBLEMA _____	3
III. MARCO TEÓRICO _____	9
IV. PROPUESTA DE SOLUCIÓN _____	17
V. CONCLUSIONES Y RECOMENDACIONES _____	33
VI. GLOSARIO _____	35
VII. BIBLIOGRAFÍA _____	37

1000

1000

1000

LISTA DE CUADROS

	Página
<i>Cuadro 1 Niveles de calidad en el proceso de mejoramiento de la empresa</i> _____	10
<i>Cuadro 2 Niveles de capacidad del CMMI</i> _____	11
<i>Cuadro 3 Grados de ignorancia de Armour</i> _____	20
<i>Cuadro 4 Meta modelo de jerarquías de requerimientos</i> _____	23
<i>Cuadro 5 Prioridades de requerimientos</i> _____	23
<i>Cuadro 6 Ejemplo de cuadro para registro de requerimiento</i> _____	26
<i>Cuadro 7 Riesgos comunes en la captura y manejo de requerimientos</i> _____	27
<i>Cuadro 8 Resumen de técnicas, área de aplicación y uso recomendado</i> _____	33

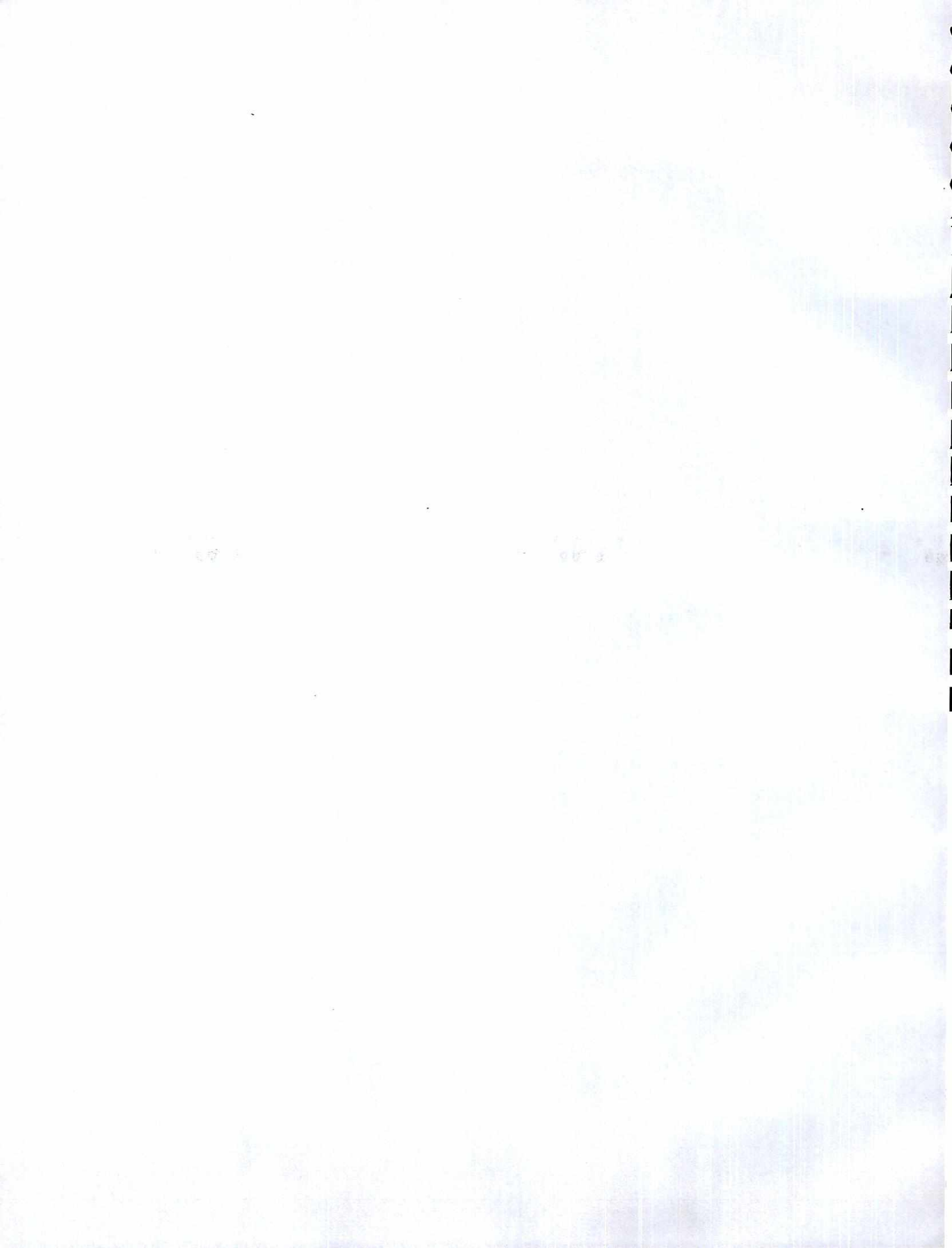
Fig 4

Fig 3

Fig 2

LISTA DE ILUSTRACIONES

	Página
<i>Ilustración 1</i> <i>Proceso de producción convencional</i> _____	7
<i>Ilustración 2</i> <i>Proceso de producción con sistemas de información</i> _____	7
<i>Ilustración 3</i> <i>Diagrama del CMMI</i> _____	13



I. INTRODUCCIÓN

El presente trabajo es producto de las lecciones aprendidas en el manejo de requerimientos para clientes internos en el desarrollo y mantenimiento de sistemas de información de pequeñas y medianas organizaciones guatemaltecas.

En el Capítulo II se presenta el tema y se hace énfasis en la importancia que los requerimientos tienen en el desempeño adecuado de los sistemas de información para los clientes internos de una organización. Se limita el estudio a organizaciones guatemaltecas típicas que se clasifican como pequeñas y medianas.

El trabajo se fundamenta en dos corrientes importantes: calidad y requerimientos. Este tema se discute en el Capítulo III. Se inicia con la presentación del concepto de calidad total, su evolución y orientación a la ingeniería de software. Luego se presenta a la ingeniería de requerimientos, haciendo énfasis en su importancia estratégica para el desarrollo e implementación exitosa de artefactos de software.

El Capítulo IV propone una solución al manejo de requerimientos basada en la interpretación del modelo del CMMI, donde se persigue mejorar la comunicación y negociación. Para que la negociación sea exitosa se presenta un patrón negociador latinoamericano en la Sección A. La Sección B propone la utilización del experto de dominio y sugiere una técnica para determinar la persona adecuada. La Sección C propone un esquema práctico para clasificar y establecer prioridades de requerimientos. La Sección D justifica el uso de QFD como la principal herramienta para manejo de requerimientos. Los riesgos comunes que se presentan en el proceso de requerimientos con clientes internos se resumen en la Sección E. Por último, se recomienda la técnica de inspecciones formales para completar el ciclo de calidad en el manejo de requerimientos.

Las conclusiones y recomendaciones del Capítulo V resumen las cualidades que debe tener cualquier herramienta de mejoramiento cuando ésta se desea usar para mejorar el manejo de requerimientos de sistemas de información para clientes internos de pequeñas y medianas organizaciones guatemaltecas.

La presentación de los temas no es rigurosa, ya que el propósito de este trabajo es que sirva como una guía para líderes que contratan servicios internos para el desarrollo y mantenimiento de sistemas de información para pequeñas y medianas empresas guatemaltecas. Contiene, sin embargo, suficiente información técnica que puede ser de utilidad para los nuevos miembros del equipo de desarrollo que se orientan hacia servicio al cliente interno.

La bibliografía sugerida incluye trabajos que desarrollan a profundidad los temas que se presentan en este trabajo.

2 d

e a

2

II. DEFINICIÓN DEL PROBLEMA

En Guatemala, la mayoría de empresas se caracterizan por ser familiares, con liderazgos fuertes y centralizados. Los equipos de desarrollo de software de estas organizaciones lo componen cinco personas máximo. Esta característica no debe considerarse como desventaja, pero sí se debe tomar en cuenta que deja poco espacio para formalidades técnicas, especialmente cuando se trata de utilizar técnicas iterativas para recolección de requerimientos.

Según las observaciones del autor, el desarrollo y mantenimiento de sistemas de información para clientes internos tiene complicaciones adicionales a los problemas propios del desarrollo de software. Estas complicaciones generalmente se relacionan a metodologías extranjeras, peculiaridades culturales de los clientes internos, y las diferencias intrínsecas entre el desarrollo de software interno o externo. A continuación se enumeran algunas de estas dificultades.

A. Problemas recurrentes

Las metodologías extranjeras no se adaptan a los proyectos guatemaltecos comunes por las siguientes razones:

- Muchos casos documentados no tienen relación con la mayoría de proyectos de software que son comunes en Guatemala.
- La escala de las organizaciones y los costos en los proyectos de software extranjeros tienen muy poca relación con la escala de las organizaciones guatemaltecas comunes
- Se asumen patrones culturales muy distintos.
- La ingeniería de software implica muchas tareas de apoyo y control que requieren de recursos adicionales, los cuales no siempre están disponibles en los grupos de desarrollo de software guatemaltecos.

Peculiaridades culturales de los clientes internos:

- Lo que interesa es el resultado final: programas que hagan algo, las formalidades técnicas no representan el resultado final buscado por lo que no se valoran “yo pago por programas no por esto ...”.
- Actitudes negativas tales como:
 - o “Nadie mejor que yo conoce el valor que mi trabajo tiene para la organización, por lo que mis requerimientos son los más importantes”
 - o “usted no va a decirme como debo hacer mi trabajo”
 - o “usted qué sabe si acaba de entrar”
 - o “si usted no sabe nada que es lo que quiere programar”
 - o “si usted no sabe entonces no opine, mejor cállese”
 - o “usted a jugar viene”
 - o “y para qué quiere saber eso”
 - o “en lugar de preguntar tanto mejor venga a trabajar”

- o “esto no me sirve para nada”
- o “mejor probemos ahora hacerlo así en lugar de lo que le dije antes”
- o “este dato no sirve para nada, nadie lo usa”
- o “usted no sabe de lo que está hablando ... como que no entiende”
- o “para dentro de un mes no me sirve, lo quiero HOY”
- o “ustedes sí se que complican todo”
- o “y cuantas veces quiere que se lo repita”
- o “yo no firmo nada”
- o “sí lo firmé, pero de todos modos hay que cambiarlo”
- o “yo nunca dije eso”
- o cualquier problema se convierte en escándalo, carencia del concepto de negociación
- Falta de visión del resultado esperado, lo que provoca
 - o Incertidumbre sobre la utilidad del trabajo que se realiza.
 - o Información suministrada con gotero
 - o Información desordenada en espera de que algo funcione
 - o Excesiva información detallada con el fin de asustar
- Desconocimiento de la cadena de producción de la organización, se desprecia o desconoce la importancia del trabajo de los demás, se desconoce la relación que la propia función tiene con los demás.
- Poca noción del costo que para la organización tiene la inversión que se realiza, o, inconformidad porque ese recurso financiero no se invierte en otros aspectos tales como mejorar sueldos.
- Malinchismo, este fenómeno es muy común, desprecia la capacidad interna, prefiriendo lo extranjero. Decepción porque la organización cuenta con un equipo de desarrollo de software sin experiencia.
- Abuso de autoridad, es extremadamente difícil para un técnico novato distinguir quién tiene más autoridad que otro, y por su falta de experiencia, se ve forzado a tomar decisiones bajo presión por la influencia del que abusa o ejerce su autoridad. Cuando la decisión es equivocada se le critica asperamente, tildándolo de complaciente o de falta de carácter. Si el equipo de desarrollo supiera exactamente la naturaleza del trabajo no tendría necesidad de recolectar requerimientos.
- Poca supervisión por parte de la alta gerencia, no se establecen ni revisan prioridades de los requerimientos, ni mucho menos verifican límites ni precedencia de autoridades –“... hay que dejarlo solo un rato para que aprenda ...”.
- Última hora: por naturaleza, los guatemaltecos dejamos para última hora las cosas importantes. Cuando se planifica una ronda de recolección de requerimientos los clientes internos no planifican con antelación ni discuten entre sí sus necesidades. Los requerimientos y prioridades se “fabrican” a última hora.
- Los requerimientos crecen sin control, “... aprovechemos que vinieron ...”
- Muchos equipos de desarrollo novatos construyen castillos en el aire, elevando peligrosamente las expectativas de los clientes internos hacia el producto que recibirán, lo que provoca grandes decepciones en la entrega de sistemas.

B. Diferencia entre el software desarrollado internamente y el software desarrollado por terceros

Las ventajas más comunes que tiene el desarrollo y mantenimiento interno son:

- Control de la evolución del software. Permite evaluar inmediatamente el impacto de cada avance.
- Exclusividad. Cuando el software apoya el valor agregado que provee la organización, especialmente cuando este valor agregado es un factor diferenciador de mercado.
- Responsabilidad directa y puntual en los resultados. Los clientes internos y el equipo de desarrollo se involucran directamente en la búsqueda de soluciones.
- Control de costos asumiendo avance continuo, de lo contrario el costo de desarrollo puede elevarse considerablemente sin que sea perceptible. Evita distracciones de cálculos de costo/beneficio especialmente si los beneficios serán a largo plazo.
- Disponibilidad.
- Comunicación directa y más fluida, confianza en el trato.
- Aprovecha y desarrolla el conocimiento interno para la innovación tecnológica de la organización.

Cuando no se cuenta con la guía apropiada, las desventajas más conocidas son:

- Empirismo.
- Confusión entre el concepto de evolución del software y evolución mal planificada y desordenada, provocada con los requerimientos por goteo o incremento desmedido y desordenado de los requerimientos.
- Falta de técnicas de negociación para evitar o reducir conflictos de requerimientos y prioridades, especialmente cuando existen islas de poder.
- Excesiva comunicación informal.
- Falta de apoyo o compromiso de la gerencia con los procesos manejados formalmente, confundiéndolos con burocracia (especialmente al ver la cantidad de documentación que puede generarse).
- Sobre/sub contratación de personal.

La contratación de un proveedor externo para el desarrollo y mantenimiento de sistemas de información tiene las siguientes ventajas:

- Especialización en la resolución de problemas similares (si es que se selecciona al proveedor adecuado).
- Provee un punto de vista distinto que puede ser beneficioso para la solución del problema.
- Se incorporan nuevos conocimientos y prácticas a la organización.
- Se aprovechan las mejores prácticas que el proveedor ya haya adquirido.
- En el caso de productos terminados: disponibilidad inmediata de soluciones exitosas y rápida implementación, puede hacerse una correspondencia entre los requerimientos de la organización y las características de la solución.

- La falta de confianza en el trato hace que la comunicación sea más cuidadosa.

Las principales desventajas son:

- La solución deja de ser exclusiva, a menos que se elaboren cuidadosos acuerdos de confidencialidad que no siempre son honrados.
- El costo del producto o del desarrollo es elevado, podría afectar el flujo de efectivo en la organización.
- El desarrollo de componentes y opciones adicionales dependen de la rentabilidad que para el proveedor externo representen, y siempre está en libertad de decidir no realizarlas.
- Puede provocar innecesaria o indeseable dependencia de terceros.
- Puede pagarse por opciones innecesarias en el caso de productos terminados.
- La disponibilidad no siempre es permanente.
- Mutua evasión de responsabilidades, tanto de parte del cliente como del proveedor.

Como puede observarse, la contratación interna o externa dependerá de la situación particular de cada caso. Sin embargo, en ambas circunstancias, los sistemas de información poseen un fuerte vínculo con el servicio al cliente. La calidad del servicio al cliente interno tiene una relación directa con el servicio al cliente externo debido a que afecta el proceso de producción. Cuando se orienta el trabajo hacia el servicio al cliente, en particular al cliente interno, se adquiere un compromiso muy serio: entregar productos de calidad, asumiendo plena responsabilidad por el diseño, producción y funcionamiento de herramientas que apoyen efectivamente los productos y servicios que la organización ofrece a sus clientes finales ¹. Cuando se contratan proveedores externos para un proyecto de desarrollo de software y no es exitoso puede perder el cliente o el proveedor; por el contrario, en los proyectos internos que fracasan siempre pierde la organización.

C. Relación entre requerimientos y software

La construcción exitosa de un sistema de información inicia con la captura de requerimientos de una forma metódica y ordenada. Sin embargo, con los sistemas de información para clientes internos, el proceso de recolección de requerimientos generalmente es informal. Consecuentemente, provoca retrasos en la implementación, no se satisfacen las necesidades del cliente o se produce una escalada de requerimientos.

Cliente es toda aquella persona u organización en la que repercute todo o parte del proceso. El proceso puede ser un producto o servicio recibido de forma interna o

¹ Berry, L., 1995. *On Great Service: A framework for Action*. New York, USA. The Free Press. pp 292
Disney Institute, 1990. *Service, Disney Style*. USA. Disney Institute.

Tschohol, J., Franzmeier, S., 1997. *Servicio al Cliente: El arma secreta de la empresa que alcanza la excelencia*. México. Editorial Pax México. pp 351

Wiersema, F., 1996. *Customer Intimacy*. California, USA. Knowledge Exchange

externa². Cliente interno es, entonces, toda persona de una organización que recibe el resultado de un proceso anterior, llevado a cabo en la misma organización, a la que podemos concebir como integrada por una red interna de proveedores y clientes. Este mismo cliente interno se convierte en proveedor cuando el resultado de su trabajo es el producto para el siguiente cliente interno; en otras palabras, el cliente interno participa con su trabajo del proceso de producción. Es así como se forman las cadenas cliente-proveedor en las cuales una misma persona puede tener doble rol, siendo mediador entre ambos, los **requerimientos** que cada uno de ellos tienen, al desempeñar su papel de cliente o proveedor, en un momento específico².

De lo anterior, se deduce que los sistemas de información para clientes internos automatizan la mediación de requerimientos entre las funciones cliente/proveedor y afectan directamente la calidad del proceso de producción. Conforme los sistemas de información participan más del proceso, mayor es el riesgo de que este proceso se complique o alargue más.

Ilustración 1
Proceso de producción convencional

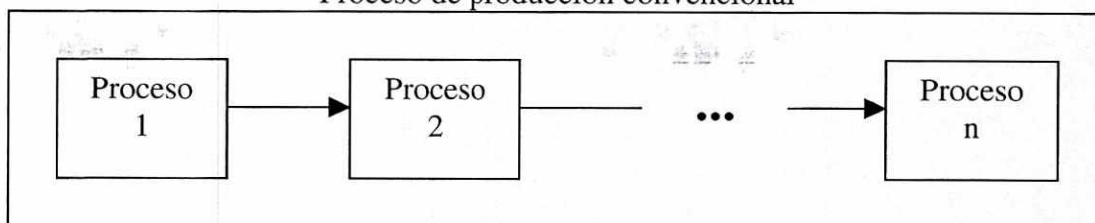
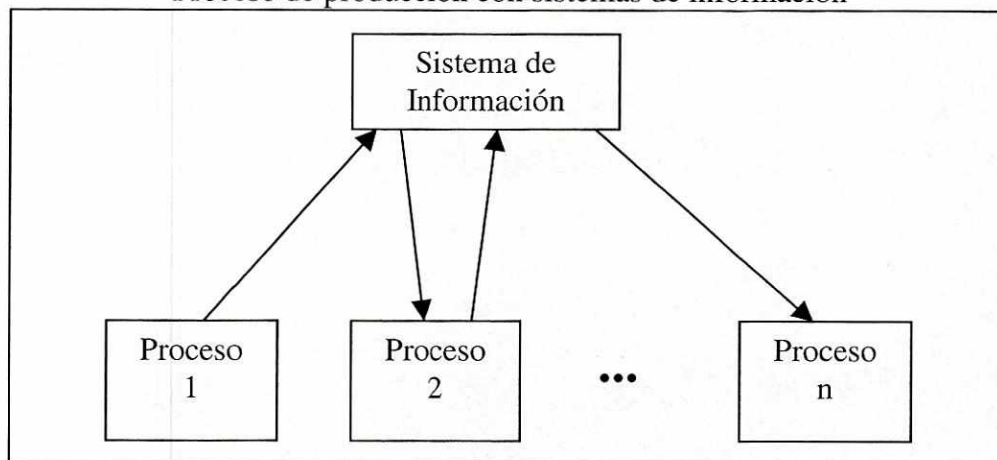


Ilustración 2
Proceso de producción con sistemas de información



² Altamirano, J., 2003 <Calidad Total y su Aseguramiento Mediante la Norma NMX-CC/ISO 9000> Facultad de Química, Universidad Autónoma de México.
<http://tenoch.pquim.unam.mx/academico/economia/calidad/manual.htm>

Es evidente entonces que lo que se busca en los sistemas de información desarrollados internamente es que agilicen este proceso y no que lo compliquen más. También resulta evidente que los requerimientos se concentran en un solo punto. Estas son dos de las razones principales por las que equivocadamente se trata de obviar algunas formalidades que son necesarias en manejo de requerimientos. Sin embargo, saber qué pasos tomar y cuáles eliminar, así como tener la capacidad de agilizar los procesos requiere, no solamente de conocimientos técnicos, sino también, conocimiento de la empresa y del problema que se quiere resolver. Una persona con estas cualidades se conoce como *experto de dominio* o experto en el campo de aplicación.

Queda demostrada la importancia suprema de los requerimientos en el desarrollo de sistemas de información, especialmente cuando éste es para clientes internos. En las secciones siguientes se proponen diversas técnicas para el manejo adecuado de requerimientos. Cuando la organización ya sabe cómo manejará los requerimientos es cuando podrá elegir las técnicas de recolección de información que sean más adecuadas a cada problema. Esto se considera tema para otro trabajo y no será cubierto en el presente documento.

III. MARCO TEÓRICO

Este trabajo se construyó sobre la base de dos corrientes importantes: Mejoramiento organizacional e Ingeniería de requerimientos.

A. Mejoramiento organizacional

El mejoramiento organizacional persigue la introducción de técnicas para robustecer los procesos de las organizaciones. Esto se logra a través de mediciones orientadas a identificar e institucionalizar (repetir) prácticas acertadas. El primer proceso clave de mejoramiento lo constituye la administración de requerimientos debido a su relación directa con la satisfacción del cliente.

1. Proceso de mejoramiento de los procesos. A mediados de la década de 1980 surge el concepto de Calidad Total como una técnica y forma de vida que motiva a las organizaciones a mejorar. Los beneficios son mayores utilidades e incremento en la capacidad competitiva de la organización a través del mejoramiento de los procesos de la empresa que son el apoyo competitivo fundamental³.

El concepto de Calidad Total se desarrolla a partir del trabajo de los doctores Feigenbaum y Shewhart, quienes hacen énfasis en la utilización de los métodos estadísticos para el control de los procesos. Mas tarde, Deming y Juran desarrollan los procesos integrales de control de calidad³.

Para lograr el mejoramiento de los procesos, Harrington propone que el primer paso debe ser la certificación y calificación del proceso. La certificación se refiere a una actividad o pieza de equipo que satisface los requerimientos del cliente inmediato. La calificación implica la evaluación de un proceso completo compuesto por componentes individualmente certificados que brindan al cliente repetida satisfacción de sus expectativas. Los niveles de calidad se presentan en el Cuadro 1.

Al inicio del proceso de mejoramiento, todos los procesos se califican en 6 y conforme se analizan se les asigna el nivel que corresponda. Luego de implementada la modernización, los procesos deberán evolucionar en calificación con el objetivo de obtener un mejoramiento continuo. Esto no significa que el objetivo final sea llegar al nivel 1, pues esto puede ser costoso e innecesario. Se aspira a clase mundial para los procesos críticos de la empresa³.

Para analizar el nivel de calificación que corresponde a un proceso, el autor sugiere tomar en cuenta los siguientes criterios para cada nivel de calidad:

³ Harrington, H. J., 1993. Mejoramiento de los procesos en la empresa. Bogotá, Colombia. McGraw-Hill. pp 309

- a. Mediciones relacionadas con el cliente final
- b. Mediciones y/o desempeño del proceso
- c. Alianzas con proveedores
- d. Documentación
- e. Entrenamiento
- f. Mediciones relacionadas con la competencia
- g. Adaptabilidad del proceso
- h. Mejoramiento continuo

Cuadro 1
Niveles de calidad en el proceso de mejoramiento de la empresa

Nivel	Estado	Descripción
6	Desconocido	No se ha determinado el estado del proceso
5	Comprendido	Se comprende el proceso y funciona según la documentación. Se inicia el proceso de mejoramiento de la empresa
4	Efectivo	El proceso se mide sistemáticamente, ha comenzado la modernización y se satisfacen las expectativas del cliente final
3	Eficiente	El proceso se moderniza y es más eficiente
2	Sin errores	El proceso es altamente efectivo y eficiente, se satisfacen también las necesidades de clientes internos. Los índices de stress son bajos
1	Calidad mundial	El proceso está en el 10% de los mejores del mundo y continúa mejorando

2. **CMMI** (*Capability Maturity Model Integration*) El concepto de Calidad Total evoluciona con el trabajo de los doctores Crosby, Deming, Juran y Humphrey, a lo que hoy se conoce como CMMI (*Capability Maturity Model Integration*), un modelo que permite logros en el mejoramiento de procesos y desarrollo de productos. El principal objetivo de este modelo es el de construir herramientas de soporte para el mejoramiento de los procesos utilizados para el desarrollo y mantenimiento de sistemas y productos, suministrando un enfoque integrado en toda la organización para mejorar los procesos, reduciendo la redundancia, complejidad y el costo que resulta al usar diversos modelos. Los dos factores más valiosos del CMMI son:

- Una guía de QUE es lo que hay que hacer.
- Un modelo que se ha desarrollado intensamente especialmente para la ingeniería de software y puede interpretarse como calidad total aplicada a software ⁴.

De similar forma que los niveles propuestos por Harrington, el modelo de CMMI representa el mejoramiento con niveles de capacidad. Estos niveles de capacidad consisten de prácticas genéricas y específicas para el mejoramiento de áreas de procesos.

⁴ CMMI Product Team, 2002. Capability Maturity Model Integration (CMMI) Version 1.1 CMU/SEI-2002-TR-011. Software Engineering Institute. Pennsylvania, USA. Carnegie Mellon University

A medida que se satisfacen las metas genéricas y específicas para los procesos de un área dentro del nivel de capacidad se alcanzan beneficios en el proceso de mejoramiento.

Los niveles de capacidad se enfocan en la habilidad que adquieren las organizaciones en mejorar su rendimiento y control de áreas de procesos. Estos niveles también permiten llevar un registro, evaluar y demostrar el progreso que las organizaciones adquieren en estas áreas de procesos. Los niveles de capacidad se basan en las destrezas adquiridas en el nivel anterior.

Los seis niveles de capacidad son:

Cuadro 2
Niveles de capacidad del CMMI

Nivel	Capacidad	Descripción
0	Incompleto	Proceso incompleto o que no se realiza. No se alcanzan los objetivos
1	Efectuado	Proceso efectuado, se alcanzan las metas. El trabajo realizado produce resultados definidos con entradas definidas
2	Administrado	Es un proceso efectuado, planificado y ejecutado de acuerdo a una política, emplea personal calificado con los recursos adecuados para producir productos controlados, involucra a clientes claves; es monitoreado, controlado, revisado; además, se evalúa su adherencia a la descripción del proceso. Se ocupa también de la institucionalización del proceso y en alcanzar otros objetivos del proceso: costo, calendario, calidad.
3	Definido	Es un proceso administrado que se ajusta a los procesos estandarizados de la organización, en cuanto a guías, contribución de productos, mediciones y otro tipo de información de mejoramiento de procesos en la organización
4	Administración Cuantitativa	Es un proceso definido que se controla con técnicas cuantitativas y estadísticas. Se establecen los objetivos cuantitativos de calidad y mejoras del rendimiento del proceso y se usan como guías para la administración del proceso. El rendimiento y calidad de ejecución se definen en términos estadísticos y se administran a lo largo de la vida del proceso.
5	Optimizando	Es un proceso administrado cuantitativamente que se cambia y adapta para alcanzar objetivos de negocios presentes y futuros.

Para aplicar el modelo del CMMI, primero se debe seleccionar entre cuatro modelos:

- a. Ingeniería de sistemas
- b. Ingeniería de software
- c. Desarrollo integrado de productos y procesos
- d. Cadena del negocio (clientes y proveedores)

El modelo de ingeniería de software está compuesto por las siguientes categorías:

- 1) Administración del proceso
- 2) Administración del proyecto
- 3) Soporte y apoyo
- 4) Ingeniería

La categoría de ingeniería la componen seis procesos clave (KPA: *key process area*):

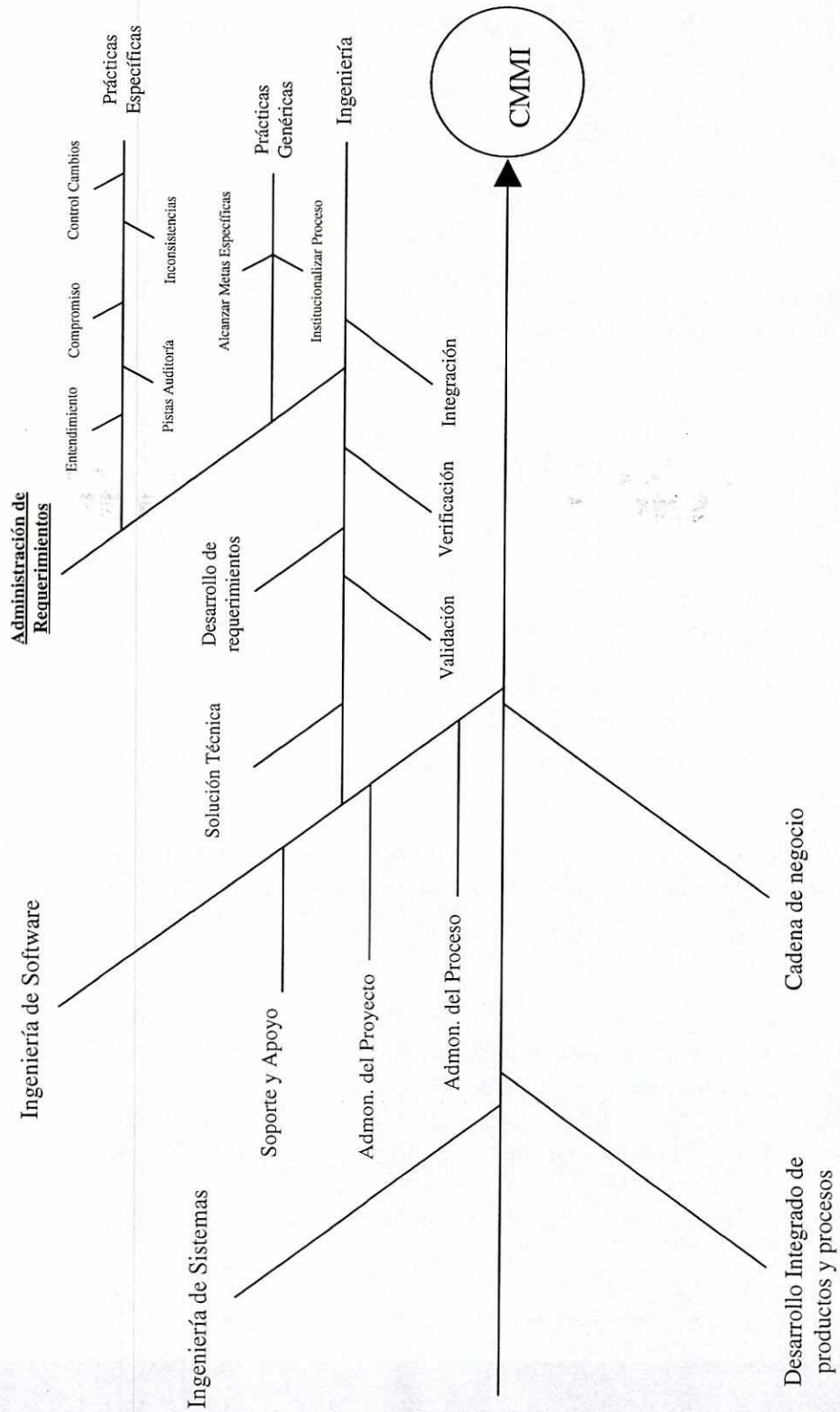
- a) Administración de requerimientos. Administra los requerimientos de productos y componentes del proyecto, identifica inconsistencias entre los requerimientos y la planificación de productos del proyecto.
- b) Desarrollo de requerimientos. Produce y analiza los requerimientos del cliente, del producto y de los componentes.
- c) Solución técnica. Diseña, desarrolla e implementa soluciones a los requerimientos. Abarca también los ciclos de vida correspondientes.
- d) Integración de producto. Ensambla el producto con todos sus componentes, asegura su correcto funcionamiento y entrega el producto.
- e) Verificación. Asegura que el producto seleccionado cumpla con los requerimientos especificados.
- f) Validación. Demuestra que el producto o componente cumple su propósito cuando es instalado en el ambiente para el cual fue diseñado.

Para una adecuada administración de requerimientos se recomienda alcanzar la meta específica del nivel y lograr las seis metas genéricas. A continuación se detallan las metas y sus prácticas correspondientes:

Meta específica y prácticas recomendadas.

- Administración de requerimientos: entendimiento, compromiso, control de cambios, pistas de auditoría bi-direccionales entre requerimiento y producto, identificar inconsistencias.

Ilustración 3
Diagrama del CMMI



Metas genéricas y prácticas recomendadas.

- Alcanzar metas específicas: establecer prácticas básicas
- Institucionalizar el proceso administrado: establecer política organizacional, planificar el proceso, proveer recursos, asignar responsabilidad, entrenar personal, administrar configuración, identificar e involucrar a clientes clave, monitorear y controlar el proceso, adherirse a evaluaciones objetivas, revisar estado de avance con niveles administrativos superiores
- Institucionalizar el proceso definido
- Institucionalizar el proceso cuantitativo
- Institucionalizar el proceso optimizando

Este trabajo se concentra en la KPA Administración de requerimientos orientado a grupos pequeños de desarrollo y mantenimiento de sistemas de información para clientes internos de pequeñas y medianas empresas en Guatemala.

B. Ingeniería de requerimientos

La ingeniería de requerimientos busca generar especificaciones completas y correctas de las necesidades de los clientes internos y la forma en que se pueden cumplir; genera la base de información necesaria para apoyar la implementación del producto, su aceptación en el campo y su futura evolución.

El proceso de recolección de requerimientos se conoce en inglés como *requirement elicitation* que literalmente en español significa *sonsacar requerimientos*: procurar con maña y astucia que uno diga o descubra lo que sabe y reserva; los requerimientos no sólo están allí para ser capturados: hay que buscarlos, descubrirlos y definirlos.

Gause y Weinberg definen desarrollo como el proceso de transformar los deseos en productos que satisfacen estos deseos. La parte que se encarga de intentar descubrir qué es lo que se desea es el proceso de requerimientos. Para entender este proceso, hay que enfocarse en cinco palabras clave: deseo, producto, personas, intentar y descubrir⁵.

Según las definiciones de Loucopoulos y Karakostas, la ingeniería de requerimientos se enfoca en los problemas asociados con errores en la especificación funcional y que resultan en mantenimientos costosos o fallas en el software. El proceso de requerimientos puede dividirse en tres subprocesos: recolección, especificación y validación. Estos interactúan con dos entidades externas: los clientes y el dominio del problema. Indican también que no existe un proceso único y útil a todas las organizaciones, ya que cada una tiene que desarrollar sus propios procesos que se

⁵ Gause, D., Weinberg, G., 1989. Exploring Requirements: Quality before Design. New York, USA. Dorset House Publishing. pp 299

adecúen a los sistemas que desarrolla, la cultura organizacional, el nivel de experiencia y la habilidad personal de los involucrados en la ingeniería de requerimientos ⁶.

La ingeniería de requerimientos se ocupa de la identificación de metas que deben alcanzarse con la visión que se tiene del sistema, la conversión de tales metas en servicios y restricciones y la asignación de responsabilidades a los agentes (humanos, software, dispositivos). La ingeniería de requerimientos involucra los procesos de análisis del dominio, captura, especificación, evaluación, negociación, documentación y evolución. La obtención de requerimientos de alta calidad es una actividad compleja y crítica. Se considera a la ingeniería de requerimientos como el área de mayor importancia en la investigación y práctica de la ingeniería de software ⁷.

La ingeniería de requerimientos no se trata simplemente de preguntarle a las personas qué es lo que quieren. Se requiere de un cuidadoso análisis de la organización, el dominio de aplicación, las expectativas de uso del sistema, y conocimiento específico del problema ⁸.

⁶ Dawson, L., Swatman, P., 2000. <<The use of object-oriented models in requirements engineering: A field study>>. Proceedings of the 22nd international conference on information systems. Limerick, Ireland. Association for Computing Machinery.

⁷ van Lamsweede, A., 2000. <<Requirements engineering the Year 00: A research perspective>>. Proceedings of the 22nd international conference on software engineering. Limerick, Ireland. Association for Computing Machinery.

⁸ Maliyakkal, S., 1998. <<Requirements elicitation>>. Seminario Methoden und Werkzeuge zur Softwareproduktion. Berlin, Alemania. <http://www.citeseer.com>.

19

20

21

IV. PROPUESTA DE SOLUCIÓN

Toda la bibliografía consultada resalta que la experiencia ha demostrado que sin un proceso y requerimientos de calidad no se pueden construir y mantener sistemas de información útiles. Este fenómeno también se ha observado en innumerables proyectos de software en organizaciones guatemaltecas, especialmente con clientes internos de pequeñas y medianas empresas, donde se considera que las formalidades técnicas no son apropiadas. Generalmente, estos proyectos son desarrollados por técnicos con poca experiencia para determinar cuáles son los pasos más adecuados para dirigir este proceso.

En este sentido, Garcia ⁹ reconoce que la aplicación de procesos de mejoramiento no son triviales en organizaciones pequeñas, ya que son percibidas como técnicas apropiadas a grandes organizaciones con muchos recursos; es más, el uso de modelos como el CMMI no está profusamente documentado para pequeños grupos, pero sí lo está para grandes proyectos aeroespaciales y similares. Woodruff ¹⁰ también señala que la pequeña organización es equivalente a pocos recursos, en donde el esfuerzo se concentra en diseñar, desarrollar y probar sistemas y no en administrar requerimientos. Sin embargo, propone las siguientes técnicas:

- Jerarquía compacta de documentación.
- Refinamiento de requerimientos.
- Registro y mantenimiento de requerimientos.

Es oportuno entonces proponer técnicas prácticas para la solución del problema, haciendo énfasis en:

- Modelos de mejoramiento organizacional orientados a la calidad.
- Perdurables en todo el proceso de mejoramiento organizacional.
- Bajo costo administrativo y tecnológico.
- Capacidad para mejorar la comunicación y negociación.
- Conocedoras de las peculiaridades culturales.

Con base a lo anterior, se propone la siguiente solución:

- CMMI (*Capability Maturity Model Integration*), como modelo de mejoramiento organizacional basado en la interpretación del modelo de administración de requerimientos del CMMI.
- Patrón negociador latinoamericano, como modelo para negociación interna.
- Grados de ignorancia, como técnica para reclutar expertos de dominio.

⁹ Garcia, S., 1999. <<Successful use of the Software Capability Maturity Model in a Small Organization>>. Technical Program Submission ASQ, Software Division's, 9th International Conference on Software Quality. Pennsylvania, USA. <http://seir.sei.cmu.edu>.

¹⁰ Woodruff, W., 1997. <<Requirements management for small organizations: A field guide to effective requirements management under SEI's CMM>>. Software Engineering Institute Repository. Pennsylvania, USA. <http://seir.sei.cmu.edu>.

- Valoración de requerimientos, como técnica para recolectar y validar requerimientos.
- Versión modificada del HOQ-QFD (*House of Quality, Quality Function Deployment*), como técnica para documentación y validación de requerimientos.
- Análisis de riesgos: como apoyo a la administración de proyectos de software.
- Inspecciones formales: como técnica para control de calidad.

A. Patrón negociador latinoamericano

El proceso de requerimientos de Gause ⁵ hace un fuerte énfasis en la capacidad negociadora y facilitadora del recolector de requerimientos, ya que durante este proceso surgen gran cantidad de conflictos, especialmente cuando se trata de resolver ambigüedades, prioridades, valoración y validación de requerimientos. Es importante notar, sin embargo, que el patrón negociador anglosajón es muy distinto al patrón negociador latinoamericano, particularmente el guatemalteco. Además de los patrones identificados por Ogliastri ¹¹, los guatemaltecos somos especialmente regateadores, críticos ásperos, y ante los conflictos rápidamente pasamos a posiciones de fuerza y violencia. Pocas veces buscamos reducir el conflicto a un proceso negociador.

Según Ogliastri, el patrón negociador latinoamericano es:

- Regateador, amistoso, informal e impreciso, prefiere negociar entre amigos
- Distributivo: suma total = 0, ganan unos y pierden otros
- No sacrifica el corto plazo para obtener ventajas a la larga, prefiere satisfacciones inmediatas en cada trato
- Flexible en el manejo del tiempo, su compromiso de tiempo es intención de cumplir, pero impaciente con la demora de otros
- Varios pueden participar, pero las decisiones se centralizan al más alto nivel
- Negociador de paquetes no de detalles, intuición, poca preparación para negociar, no hace prenegociaciones – prefiere negociación informal fuera de la mesa de negociaciones
- Argumentador genérico, vago en términos pero sólido en principios generales
- Acuerdos rápidos, la idea súbita genial, imprecisos, de palabra, o, paradójicamente, de minuciosa legalidad
- Pueden rápidamente convertirse a posiciones de fuerza y violencia

Ogliastri ¹² también sugiere las siguientes técnicas genéricas, que pueden ser efectivas para negociar con este patrón.

- Con sus colegas, no negocie como si fueran adversarios ni tampoco lo vuelva amistad; mantenga su relación en un cordial y respetuoso colegaje.

¹¹ Ogliastri, E., 1998. <<El estilo negociador de los latinoamericanos. Una investigación cualitativa>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.

¹² Ogliastri, E., 1997. <<La nueva manera de negociar: un decálogo>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.

- Su propósito central es resolver un problema conjunto; no es ganarle al otro ni llegar a un acuerdo a toda costa.
- Piense a largo plazo en los resultados que tendrá esta negociación tanto para la relación con el otro como para futuras negociaciones.
- Busque la solución favorable que daría un juez o árbitro externo; apele a los números, a las razones... deje de lado las amenazas y las “mentiras”.
- Piense en el otro tanto como en usted mismo; haga ofertas que sean buenas para usted pero al menos aceptables para el otro.
- Redefina el problema de la manera más amplia posible para que incluya los intereses reales más amplios del otro y los suyos propios; utilice la negociación como un medio para redefinir el problema haciendo preguntas y abriéndose a nuevas maneras de concebir la solución.
- Haga preguntas y ofrezca gradualmente intercambios de información. Determine en cuáles elementos hay una total incompatibilidad con el otro lado, en cuales hay exactamente el mismo interes, y en cuales se tiene diferente valoración para cada uno.
- Amplíe el “paquete”, busque enriquecer para todos el resultado de la negociación. Proceda a regatear sanamente lo que sea regateo; utilice estándares o criterios objetivos externos a los negociadores. Al mismo tiempo, busque e identifique el “bloque” total de puntos que debe negociarse.
- Haga cálculos antes de sentarse a negociar. ¿Cuál es su alternativa externa (lo que usted haría si no llega a un acuerdo)? ¿Cuáles son sus prioridades: cuál es el valor relativo de los diferentes puntos que va a negociar?
- Determine las diferencias de valor que puede haber entre usted y la contraparte, y trate de intercambiar los puntos de más valor para usted por los de más valor para el otro. De esta manera se creará valor: habrá más para ambos.

Estas técnicas genéricas son más útiles cuando ambas partes conocen el proceso negociador. Si solamente una de las partes está conciente de ellas, la negociación es un poco más complicada. Si ninguno conoce el concepto de negociación entonces ambas partes están a las puertas de un conflicto. La diferencia de estas técnicas genéricas con la negociación rutinaria guatemalteca con clientes internos es el regateo intenso, esto es cuando hay negociación, de lo contrario, no hay negociación, simplemente imposición. Este es el caso cuando los equipos de desarrollo son novatos, tanto en el aspecto técnico como en el dominio del negocio y los procesos negociadores. Simplemente no existen opciones para negociar, los requerimientos son específicos (según la contraparte). En este caso, los clientes internos pasan del regateo a tomar posiciones de fuerza y poder. Debido a estas razones, es recomendable evitar el conflicto nombrando un negociador o facilitador y entrenar ambos equipos en las técnicas de negociación.

A diferencia de la negociación con clientes internos, la negociación con clientes externos es más natural, y generalmente más protocolaria debido principalmente a la limitada confianza y disponibilidad entre las partes. La negociación se realiza en términos financieros, por lo que las concepciones son más evidentes. Las posiciones de poder se limitan porque las partes están en libertad de recindir un contrato. Ante la

amenaza de un conflicto, los caminos son claros: arbitraje externo o demanda legal, ambas instancias conllevan costos adicionales.

Lo más importante en cualquier proceso negociador es llegar preparado y conservar la relación con el cliente, ya sea externo o interno.

B. Experiencia de dominio

El experto de dominio es aquella persona que puede guiar el proceso de recolección de requerimientos. No necesariamente tiene que ser el recolector. En cualquier actividad de desarrollo y mantenimiento de sistemas de información existen al menos dos dominios: el dominio del problema y el dominio técnico. Se entiende como dominio, el campo de acción o territorio donde se desarrolla el trabajo. Experto de dominio, es aquella persona que tiene conocimiento perfecto (o conocimiento profundo y detallado) del área de aplicación. El experto en el dominio del problema asiste en:

- Precisar la visión.
- Aclarar terminología.
- Descubrir asunciones ocultas.
- Sugerir las formalidades más apropiadas de acuerdo a la cultura organizacional.
- Identificar procesos rutinarios.
- Aclarar prioridades.
- Descubrir el flujo organizacional y valor agregado de cada par cliente-proveedor.

Cuadro 3
Grados de ignorancia de Armour

Grado	Descripción genérica	Descripción aplicada al software
0	Falta de ignorancia: se puede demostrar el conocimiento de algo	Se tiene la respuesta correcta (experto de dominio ideal)
1	Falta de conocimiento: se sabe que es lo que no se conoce, pero sabe cómo resolver la falta de conocimiento	Se tiene la pregunta correcta (candidato a experto de dominio)
2	No se tiene conciencia de que se ignora algo	No se tiene la pregunta (inicio de proyecto: peligro)
3	Falta de Proceso, no se sabe cómo resolver la ignorancia de algo que no se conoce	No se sabe cómo resolver la falta de conocimiento (peligro)
4	Desconocimiento de los grados de ignorancia	No se conocen los grados de ignorancia

Si se concibe el software no como un producto, sino como un almacenamiento y codificación de conocimiento, puede verse al experto de dominio como el proveedor del conocimiento que debe implementarse en el software. El opuesto al conocimiento es la

ignorancia. Armour¹³ propone la una técnica para descubrir el grado de ignorancia, la cual se considera útil para determinar quién puede ser experto de dominio (Cuadro 3). Si no se tiene quién dé respuestas exactas y correctas, pero se cuenta con alguien que sabe hacer las preguntas correctas, entonces seguramente se llegará a la respuesta correcta (niveles 0 y 1). Se consideran los niveles 2 y 3 como peligrosos. En el nivel 2, al principio del proyecto de desarrollo de software, lo más con lo que se cuenta es con la necesidad de resolver un problema y muchas veces no está claro, se sabe que habrá mucho aprendizaje, pero no se sabe qué se aprenderá. El nivel 3 es peligroso porque las metodologías no son la respuesta a los problemas, son más bien técnicas para construir preguntas y disciplinas para identificar las áreas donde pueda tenerse grado de ignorancia 2, las respuestas que se buscan tienen que venir de los clientes internos.

Podría suponerse que encontrar un experto de dominio dentro de los clientes internos es una tarea trivial. Por observaciones del autor, éste no siempre es el caso debido principalmente a los siguientes problemas:

- Desconocimiento de la visión y alcances del proyecto.
- Desconocimiento y desprecio por cambios a futuro.
- Desconocimiento de las funciones de otros clientes internos, muchos solamente se preocupan de su propia labor sin importarles como este afecta a los demás.
- Reacción al cambio.
- Poco interés en asumir responsabilidades, nadie está interesado en firmar sus requerimientos, muy pocas (o ninguna) personas se interesan en opinar sobre la conveniencia o factibilidad de un cambio de procedimientos.
- Pocos clientes internos asumen responsabilidad por documentos firmados.
- Todos los requerimientos tienen el mismo valor de prioridad, importancia, simultaneidad, urgencia.
- Si una función o dato del sistema no les es de utilidad directa, inmediatamente se tacha de innecesaria.
- Pocos clientes internos quieren identificar con claridad cuándo están completos todos sus requerimientos, porque siempre queda algo pendiente.
- Algunos ven con simpleza.
- Esperan que el equipo de desarrollo sea quien proponga requerimientos correctos, ya que supuestamente deben conocer su trabajo
- Nadie parece tener tiempo para dedicarse al planteamiento de requerimientos

El método del MSF¹⁴ se propone que al equipo de desarrollo se incluya a un Gerente de Producto. Las funciones del gerente de producto son similares a las de un experto de dominio. La importancia radica en la ***corresponsabilidad*** que esta persona comparte con el equipo de desarrollo en que el producto de software a construir sea exitoso. El gerente de producto se asegura de que:

¹³ Armour, P., 2000. <<The five orders of ignorance>>. Communications of the ACM. New York, USA. Association for Computing Machinery.

¹⁴ Microsoft Solution Framework: Principles of Application Development. 1999. Course Number 1516A. Microsoft Corporation, USA

- el equipo de desarrollo comprenda las expectativas de la organización,
- se definan los requerimientos y expectativas de los clientes,
- se controlen expectativas,
- ante el equipo de desarrollo, actúe como abogado de los clientes y patrocinadores,
- ante los clientes, actúe como abogado del equipo de desarrollo,
- administre el lanzamiento del producto.

C. Clasificación y establecimiento de prioridades de requerimientos

Cuando los requerimientos están debidamente identificados, analizados, validados, clasificados y con la prioridad debida, permiten una mejor planificación del proyecto de desarrollo. En esta sección se presentan sugerencias de cómo clasificar y establecer prioridades en los requerimientos, una vez más, haciendo énfasis en facilitar la comunicación entre los clientes internos y técnicos.

Los subsistemas que componen los sistemas de información para clientes internos no son productos aislados. Por el contrario, están íntimamente relacionados entre sí. Como se menciona en la Definición del Problema, forman parte esencial de un proceso de producción de la organización y constantemente evolucionan para crear productos altamente relacionados y complejos.

Kuusela ¹⁵ y Savolainen ¹⁶ presentan la ingeniería de requerimientos para familias de productos como alternativa al estudio de requerimientos en productos relacionados. Los autores enfocan más su estudio hacia productos para diferentes sectores de mercado con clientes externos y lo usan para estimar probabilidades de que algunos requerimientos cambien. El cálculo de probabilidades de que un requerimiento cambie está fuera del contexto de este trabajo. La utilidad estriba en la forma en que clasifican los requerimientos ya que estos ayudan a determinar criterios arquitectónicos del sistema.

Muchas técnicas clasifican los requerimientos en funcionales y no funcionales. Sin embargo los autores proponen una variante:

- Objetivos de diseño: esencia de los requerimientos, definen la funcionalidad que se espera del sistema.
- Compatibilidad y decisiones de diseño: muchos autores indican que no deben tomarse decisiones de implementación en la fase de requerimientos, pero la práctica demuestra que cada requerimiento que se agrega debe ser compatible con la arquitectura del

¹⁵ Kuusela, J., Savolainen, J., 2000. <<Requirements engineering for product families>>.. Proceedings of the 22nd international conference on software engineering. Limerick, Ireland. Association for Computing Machinery.

¹⁶ Savolainen, J., Kuusela, J., 2001. <<Volatility analysis framework for product lines>>. SSR. Ontario, Canada. Association for Computing Machinery.

sistema, por lo que se deben registrar las decisiones que se hacen en cada etapa para permitir que los nuevos requerimientos se mantengan factibles.

- Jerarquía de requerimientos: luego de clasificar los requerimientos en objetivos y decisiones de diseño, construyen una jerarquía para establecer prioridades y dependencias.

Para el propósito de este trabajo, lo relevante es el meta-modelo de jerarquías y los niveles de prioridad de requerimientos de Kuusela y Savolainen.

Como puede observarse, estas clasificaciones son prácticas ya que se utiliza un vocabulario que no es ajeno a los clientes internos.

Cuadro 4
Meta modelo de jerarquías de requerimientos

Metas	Qué es lo que el sistema debe hacer
Características	Servicios que el sistema debe proveer al cliente
Requerimientos funcionales	Lo que el cliente no ve; requerimientos técnicos
Requerimientos no funcionales	Propiedades del sistema, metas detalladas que pueden medirse con restricciones
Restricciones	Refinan requerimientos, parámetros de satisfacción, propiedades del sistema

Cuadro 5
Prioridades de requerimientos

Valor	Clasificación	Descripción
0	No permitido	No debe implementarse
1	Indiferente	No importa si lo tiene o no
2	Podría ser útil	Puede usarse en algunos casos
3	Deseable	Agrega valor, puede omitirse sin afectar el sistema
4	Útil	Requerimiento útil en la práctica, provee algún beneficio tangible al cliente
5	Algo importante	Es útil, tiene sentido y provee beneficios tangibles al cliente
6	Importante	El cliente lo identifica claramente como útil
7	Muy importante	Beneficioso al cliente, argumento de convencimiento, provee beneficios tangibles
8	Esencial	No incluirlo puede afectar la utilidad del sistema
9	Obligatorio	Si no cumple con esto, el sistema no es útil

D. QFD (*Quality Function Deployment*)

El proceso de requerimientos se fundamenta en tres actividades principales: recolección de información, análisis y validación. Las dos primeras actividades enfocan la importancia de comprender el qué, cómo, y por qué de los requerimientos. La

validación de requerimientos se concentra en clasificar, establecer prioridades, identificar de riesgos, volatilidad, planeación del proyecto, proceso de decisión, negociación, comunicación, control de calidad, pista de auditoría entre requerimientos y componentes, etc.¹⁷. Para el CMMI⁴ la definición de validación es el aseguramiento de que el requerimiento sea el correcto para construir el producto correcto.

Entre las técnicas más recomendadas para validar requerimientos se encuentran los prototipos y QFD.

Los prototipos tienen la ventaja de ser muy útiles, tanto para el usuario como para el desarrollador ya que ambos logran enfocarse a un mismo objetivo. Su principal desventaja es que no se trata de la solución final. Los clientes creen que están aceptando un componente terminado cuando en realidad lo que están logrando es un acuerdo en la funcionalidad de los casos que se prueban. Para resolver este último problema, la técnica sugiere desechar el prototipo e iniciar la construcción del producto final, sin embargo el costo del desarrollo del prototipo impide que sea desechable.

QFD (*quality function deployment*: despliegue de función y calidad) es un sistema de calidad que fue desarrollado por los doctores Shigeru Mizuno y Yoji Akao durante la década de 1960, cuando exploraban formas de aplicar algoritmos al diseño correcto de productos desde la primera vez. Esto fue motivado por la preocupación causada por el distanciamiento cada vez mayor entre el productor y consumidor. QFD persigue enlazar las necesidades del cliente con los procesos de diseño, desarrollo, ingeniería, fabricación de productos y servicios. Resulta muy útil para validar requerimientos, ya que permite capturar los deseos del cliente y posteriormente validar que los requerimientos propuestos sean reflejados adecuadamente en el producto. Ayuda en establecer prioridades, identificar conflictos y requerimientos del cliente que no fueron tomados en cuenta por el desarrollador, identifica también aquellas características propuestas por el desarrollador que no están basados en requerimientos reales¹⁷.

QFD es una iniciativa que ha evolucionado a través de los años, lo que ha generado una gran cantidad de diagramas y métodos. Uno de los diagramas que se aplican a este trabajo es una versión modificada de la matriz de referencias cruzadas (*HOQ: House of Quality*). Aunque algunos autores indican que este diagrama por sí mismo no es QFD, el uso es adecuado a este trabajo porque se usa como un componente dentro de un esquema de mejoramiento.

En proyectos de desarrollo y mantenimiento de sistemas de información para clientes internos se ha observado, con mucha frecuencia, la cantidad de conflictos que se genera cuando, luego de concluir un ciclo de recolección de información, se intenta obtener el compromiso de los clientes internos con los requerimientos planteados por ellos mismos. La forma más común es al preparar un documento que resuma los requerimientos con

¹⁷ Christel, M., Kang, K., 1992. <<Issues in requirements elicitation>>. Technical Report CMU/SEI-92-TR-12. Pennsylvania, USA. <http://www.sei.cmu.edu>.

tecnicismos o diagramas de modelos y esperar que los clientes internos los firmen en señal de aceptación y compromiso. Si los documentos de requerimientos se preparan cuidadosamente, resultan muy extensos. Es importante distinguir dos tipos de documentación en esta etapa: de requerimientos y técnica. La forma resumida y directa en que se representan los requerimientos en QFD facilitan la presentación de los requerimientos ante el cliente. Él podrá autorizar el desarrollo sin tener que leer una gran cantidad de documentación, con lo que se evitan conflictos innecesarios.

Para validar requerimientos con diagramas y modelos es necesario entrenar a los clientes internos en la nomenclatura del modelo para lograr una mejor comunicación. Se ha observado que el interés en aprender decae directamente con el tiempo que demora el entrenamiento. Más aún, si se logra mantener el interés, no siempre se logra el objetivo de validar exitosamente el modelo. Lo que los clientes internos quieren y entienden son sus requerimientos, no los detalles de implementación. Debido a que una de las estructuras básicas de QFD es una matriz de referencias cruzadas, su funcionamiento es intuitivo y permite que el equipo de validación (desarrolladores y clientes internos) se enfoque directamente en los requerimientos, las prioridades, criterios de aceptación, conflictos posibles, etc. En resumen, lo que para el cliente es deseable en el sistema.

Como se menciona anteriormente también, lo importante de los productos de software no es el código en sí, sino el conocimiento que está codificado en ellos, y este conocimiento es volátil. Este conocimiento no puede generarse en un solo evento, es un proceso evolutivo. Mientras más conocimiento se adquiere también aumentan los requerimientos, las organizaciones y los sistemas evolucionan, en otras palabras, los satisfactores generan nuevas necesidades. Los requerimientos volátiles resultan excesivamente costosos de implementar repetidas veces con prototipos. QFD logra capturar esa volatilidad de una forma más práctica, evitando el desarrollo de código prematuramente.

Cuando una herramienta puede enseñarse y utilizarse en toda la organización se logra institucionalizar mejoras de una forma más efectiva. Los bajos requerimientos tecnológicos de una herramienta permite que pueda utilizarse en organizaciones con presupuestos limitados. QFD tiene la ventaja de que puede iniciarse con papel y lápiz, puede utilizarse con hojas electrónicas y puede utilizarse una herramienta específica. Esto demuestra que QFD es una herramienta escalable. La escalabilidad de la herramienta permite el uso durante todo el ciclo de mejoramiento de la organización, lo que reduce los costos asociados.

La matriz de referencias cruzadas, uno de los tantos diagramas de QFD, es culturalmente bien aceptada. Esto fue probado informalmente por el autor con algunos resultados positivos. A pesar que la técnica no se aplicó profundamente, si se construyeron listas de requerimientos y criterios de aceptación junto a los clientes internos. Cuando se entregó el producto con el respaldo de requerimientos y uno a uno se verificó el cumplimiento de ellos, los clientes internos quedaron más conformes con lo que pidieron. En ocasiones, cuando fue necesario hacer cambios por requerimientos mal

E. Análisis de riesgos

El análisis de riesgos permite identificar posibles problemas y prevenirlos antes que se manifiesten. Estos riesgos siempre son menospreciados cuando se inicia un proyecto, especialmente si éste se conduce por novatos. En los capítulos anteriores se hace énfasis en la volatilidad de los requerimientos debido fundamentalmente a que el desarrollo de software consiste en la codificación de conocimiento: mientras más se sabe, más requerimientos surgen. No debe tomarse como sorpresa que los requerimientos cambien, pues es un hecho que cambiarán. Aceptar esta realidad en el análisis de riesgos es fundamental para anticiparse a lo que inevitablemente pasará. Adicionalmente a los riesgos comunes, están las reacciones al cambio ¹⁹.

Los riesgos que a continuación se presentan se tomaron de Dawson ²⁰ y Maliyakkal ²¹. Se compararon a los riesgos usuales que se presentan en equipos de desarrollo y mantenimiento de sistemas de información para clientes internos y se encontró coincidencia en muchos de ellos.

Lo importante de los riesgos es identificarlos a tiempo y vigilarlos continuamente.

Cuadro 7
Riesgos comunes en la captura y manejo de requerimientos

Riesgo	Causa	Estrategia de manejo
Especificaciones erróneas	Mal recolectados, mal comunicados	Apoyarse con expertos de dominio, entrenarse en el uso de la herramienta de recolección
Asumciones erróneas	Clientes internos que asumen mal, desarrolladores que asumen mal	Apoyarse con expertos de dominio, incluir a más personas en el proceso de captura de requerimientos, no precipitarse en tomar decisiones
Requerimientos incompletos debido a clientes internos indecisos, inseguros y novatos	No contar con personal clave, ellos lo consideran una labor tediosa	Apelar al liderazgo y conocimiento del personal clave, plantear la necesidad de contar con su tutoría, demostrar los requerimientos recolectados hasta el momento, verificar que la técnica de recolección sea la adecuada

¹⁹ Block, P., 1989. <<Como entender la resistencia>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.

Kanter, R. M., 1985. <<El manejo del aspecto humano del cambio>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE

²⁰ Dawson, R., 2000. <<Twenty dirty tricks to train software engineers>>. Limerick, Ireland. Association for Computing Machinery.

²¹ Maliyakkal, S., 1998. <<Requirements elicitation>>. Seminario Methoden und Werkzeuge zur Softwareproduktion. Berlin, Alemania. <http://www.citeseer.com>.

Cambios de requerimientos y prioridades	Clientes internos que cambian de parecer debido a errores humanos, cambios legales o en condiciones del mercado	Documentar debidamente la recolección de requerimientos, informarse de plan estratégico de la organización, revisar prioridades
Ideas, presiones y requerimientos conflictivos	Contradicciones	Documentarlas en QFD, negociar entre las personas que presentan conflicto, si la situación es extrema solicitar el apoyo de un negociador
Clientes internos con diferentes personalidades	Clientes internos mal intencionados, mentirosos, indecisos	Documentar los errores, buscar otros participantes
Eliminación o reducción de recursos	Limitaciones de recursos, personal, horas, presupuesto	Plantear el problema pero no descuidar la calidad del producto
Actividades adicionales que eliminan planificación	Emergencias, labores de mantenimiento, atención inesperada de clientes internos debido a errores imprevistos o solicitudes fuera de lugar	Documentar el incidente, asignar parte del tiempo de cada desarrollador a labores de mantenimiento
Cambio en fechas de entrega	Cambios legales, cambios en condiciones de mercado, presiones políticas	Documentar el incidente, informarse de condiciones de mercado y estrategia de la organización, revisar prioridades
Heredar sistemas antiguos para actualización y mantenimiento	Cambios o actualizaciones a sistemas mal documentados, frágiles, críticos y sin mayor entrenamiento previo	Documentar el incidente, asignar equipos quirúrgicos
Clientes internos que constantemente presionan por sus productos	Interés en producto, presiones de trabajo, falta de visibilidad	Brindar la mayor visibilidad posible demostrando las actividades realizadas y las que faltan por realizar, revisar prioridades de requerimientos
Clientes indiferentes al proceso, solo quieren el producto final	Puede presentarse de dos formas: no quieren interferir o no quieren involucrarse para evitar responsabilidades	Brindar la mayor visibilidad posible demostrando las actividades realizadas y las que faltan por realizar cuando estas sean solicitadas
Clientes que creen que el proceso solo se aplica a la industria química o automotriz pero inaplicable a software interno	Clientes internos que esperan que el equipo de desarrollo produzca software por generación espontánea	Brindar la mayor visibilidad posible demostrando las actividades realizadas y las que faltan por realizar, presentarles esta guía

F. Inspecciones formales

Las inspecciones formales fueron desarrolladas en 1970 por Michael Fagan en IBM para mejorar la calidad del software, aumentar la productividad de los programadores y detectar los errores en una etapa temprana. La ventaja principal del proceso de inspección es su aplicabilidad a revisión de requerimientos, y también en forma análoga, a otros productos como diseños, códigos, etc. Las inspecciones formales involucran la interacción de cinco elementos:

- Proceso de inspección definido.
- Roles de participantes definidos.
- Recolección formal de datos del proceso y producto.
- Producto que se inspecciona.
- Infraestructura de soporte.

Existen muchas técnicas de control de calidad aplicadas al software, que en general buscan aislar el proceso de pruebas del proceso de desarrollo. En equipos pequeños tal forma de trabajo no es práctica, no sólo porque no hay suficiente recurso humano, sino también porque los clientes internos, ante la presencia de una falla, no pueden esperar a completar un largo proceso de pruebas y control de calidad para continuar su trabajo. Otra desventaja de las otras técnicas de control de calidad tradicionales es que pocas fomentan la propagación del conocimiento en el equipo de desarrollo y la responsabilidad de liberar un buen producto reside en quien hace las pruebas y no en quien desarrolla el producto.

La forma usual en que operan los equipos pequeños es asignando una parte del sistema a cada persona del equipo de desarrollo. En otras palabras, cada miembro del equipo es responsable de uno o más subsistemas y ejecuta todas las funciones relacionadas al mismo. La especialización de funciones solamente se da cuando se entrena a un nuevo miembro del equipo o cuando esa función es crítica. Esta forma de organización tiene sus ventajas y desventajas, independientemente de que esta sea la forma correcta o no de organizar el equipo de desarrollo. Su ventaja principal es que eventualmente se cuenta con expertos de dominio internamente. La desventaja principal es que tiende a crear islas de conocimiento. La ventaja de contar con un experto de dominio ya se discutieron en anteriormente.

Cuando dentro del equipo de desarrollo se forman islas de conocimiento se generan riesgos para el individuo, el equipo de desarrollo y la organización, y si no se detectan a tiempo, su manifestación resulta bastante peligrosa y afecta directamente la continuidad de la organización. Para el individuo, el problema es que toda la responsabilidad recae sobre él, y todas las decisiones (personales y de trabajo) se tornan críticas. Esto eleva la posibilidad de cometer errores y resulta dañino para la salud del individuo (aumenta los niveles de ansiedad, estrés, soledad y sensación de falta de apoyo). Para el equipo de

desarrollo, la falta de un integrante (por vacaciones, enfermedad, renuncia, despido, fallecimiento) es crítica, no sólo porque repentinamente alguien tiene que hacerse cargo de una responsabilidad que no sabe como sacar adelante, sino que tampoco puede dejar a un lado sus actividades normales. Para la organización es sumamente delicado ya que existe el riesgo de que una buena parte de la organización se paralice ante una problema porque su riesgo está concentrado en un solo punto de falla.

Una forma de manejar este problema es con la documentación adecuada, pero es sabido que una documentación adecuada no siempre está presente especialmente en organizaciones pequeñas y con clientes internos. En la sección anterior, se propuso el uso de QFD, no solo como una técnica de validación, sino además, como una solución al problema de documentación de requerimientos.

En el modelo de Extreme Programming ²² el problema de islas de conocimiento se resuelve con la rotación continua de parejas de trabajo. Este sistema se fundamenta en búsqueda de calidad y, bien manejadas, pueden ser muy prácticas y fáciles de adoptar, ya que resultan muy naturales. Se ha observado informalmente que la práctica de parejas de trabajo es utilizada naturalmente para resolver problemas y para entrenamiento. Las desventajas de estas formas de trabajo es que no forman expertos de dominio y las parejas de trabajo no siempre son prácticas o posibles en equipos pequeños.

Para el propósito de este trabajo, la técnica de inspecciones formales ²³ (también conocidas como técnicas de lectura ²⁴) tiene las siguientes ventajas:

- Proactiva, ya que puede pronosticar problemas futuros que son útiles para el análisis de riesgos, aprovecha la experiencia del equipo para darle forma e interpretación correcta al requerimiento, identifica la relación que el requerimiento pueda tener con otros requerimientos, puede ilustrar futuras estrategias de implementación.
- Perdurable en todo el proceso de mejoramiento empresarial.
- Bajo en requerimientos de tecnología.
- Identifica errores.
- Propaga el conocimiento en el equipo aumentando el número de expertos de dominio.
- Mantiene al equipo comunicado.

La inspección formal puede llevarse a cabo con tres personas como mínimo. El protocolo que se presenta a continuación puede ajustarse a cada caso, pero sí es importante que se observen las recomendaciones.

²² Extreme Programming. 2002. <http://www.extremeprogramming.org>.

²³ Software formal inspections guidebook. 1993. NASA-GB-A302. Washington, DC, USA. Office of safety and mission assurance, National Aeronautics and space administration. pp 68

²⁴ Travassos, G., Shull, F., Fredericks, M., Basili, V., 1999. <<Detecting defects in object oriented designs: Using reading techniques to increase software quality>>. Proceedings of the 1999 ACM SIGPLAN conference on object-oriented programming, systems, languages and applications. New York, USA. Association for Computing Machinery.

En forma resumida, las inspecciones formales consisten en que el desarrollador presenta su trabajo ante todo el equipo, esta presentación se basa en el siguiente protocolo:

1. Proceso de planificación.

- o Seleccionar el producto.
- o Seleccionar participantes (moderador, registrador, lector, autor, inspectores).
- o Informar a participantes los detalles del producto, estándares, requerimientos y cualquier otra información relevante a la inspección.
- o Informar lugar, fecha y hora del evento, se busca que el proceso de inspección no pase de dos horas.

2. Proceso de preparación.

- o Cada participante estudia el producto antes de la reunión de inspección.
- o Es imperativo llegar preparado.
- o Puede hacerse presentaciones previas para informar acerca del proceso, el producto y cualquier otro detalle, especialmente si los inspectores son nuevos, si el producto es nuevo o la técnica es novedosa.

3. Proceso de inspección.

- o El lector presenta e interpreta ordenada y lógicamente uno a uno los detalles del componente (documentos, modelos, líneas de código, etc.).
- o Los inspectores pueden interrumpir en cualquier momento cuando se detecte cualquier error.
- o Si es necesario, se puede interrumpir la lectura y puede discutirse la naturaleza del error, el autor aclara cualquier punto según sea necesario.
- o Si efectivamente es un error, el grupo clasifica el error y el grabador registra el error identificado, no es necesario brindar una solución en este momento. Los errores pueden clasificarse como graves (por ejemplo, no se cumplen los requerimientos) o menores (errores tipográficos, errores de estándar, etc.).
- o La lectura prosigue después de registrar el error.
- o El moderador mantendrá la discusión bajo control, cuidando el enfoque, ritmo, tono y tiempo de la inspección y discusión. Si la discusión se prolonga, declarará el punto no resuelto, se registra la decisión y la inspección continúa.
- o Al completar la inspección, se decidirá si la reunión se prolonga por una hora más para tratar los puntos pendientes, si se agotan las tres horas, se programará una segunda reunión de inspección.
- o Al final de la reunión, el registrador resume los errores y los informa.
- o El moderador y el autor se reunirán brevemente para establecer prioridades y estimar el tiempo para reparar los errores.
- o Si hay tiempo durante la tercera hora, pueden discutirse alternativas para reparar los errores encontrados.

4. Proceso de corrección de errores.

- o El autor será responsable de corregir todos los errores graves encontrados, puede apoyarse con equipos quirúrgicos.
- o Si hay tiempo y es meritorio se corregirán los errores menores.
- o El autor se reunirá con el moderador para verificar la corrección de errores.
- o Si es necesario, puede programarse otra reunión de reinspección cuando los errores son muchos o el tema es complejo.
- o Si todos los errores graves se resuelven se aprueba el producto y puede pasar a la fase de pruebas.

Consideraciones y recomendaciones importantes de la inspección formal:

- o Es importante cuidar el tono de la plática para que sea una búsqueda de errores y no una crítica al producto, autor, etc.
- o El tono apropiado de la inspección fomenta la participación y detección de errores
- o Para evitar sesgo en la inspección se prefiere que los jefes no estén presentes. Si el jefe participa deberá separar sus funciones y enfocarse únicamente en el aspecto técnico de la inspección
- o Para agilizar la inspección deberá llegarse preparado
- o Es importante la selección de inspectores adecuados, se ha observado que si se seleccionan indiscriminadamente se generan muchos conflictos, y puede incluso desecharse un producto por la ignorancia de algunos inspectores
- o Para que la reunión de inspección sea exitosa deberá concentrarse únicamente en eso. No es permitido tocar otros temas, ni mezclar reuniones.

V. CONCLUSIONES Y RECOMENDACIONES

Las pequeñas y medianas organizaciones guatemaltecas pueden beneficiarse con las prácticas de mejoramiento organizacional. Este trabajo es una guía que conduce el uso del CMMI aplicado a la fase de manejo de requerimientos especialmente orientado al uso dentro de equipos pequeños con poca experiencia. Se apoya la selección de técnicas con observaciones informales del autor al estar expuesto al desarrollo de sistemas de información para clientes internos guatemaltecos.

El manejo apropiado de requerimientos es fundamentalmente un problema humano. Por esta razón, una estrategia para solucionar el mismo tiene necesariamente que considerar los factores humanos involucrados. Este documento persigue identificar y resaltar los aspectos culturales que afectan el manejo apropiado de requerimientos en organizaciones guatemaltecas.

Cuadro 8
Resumen de técnicas, área de aplicación y uso recomendado

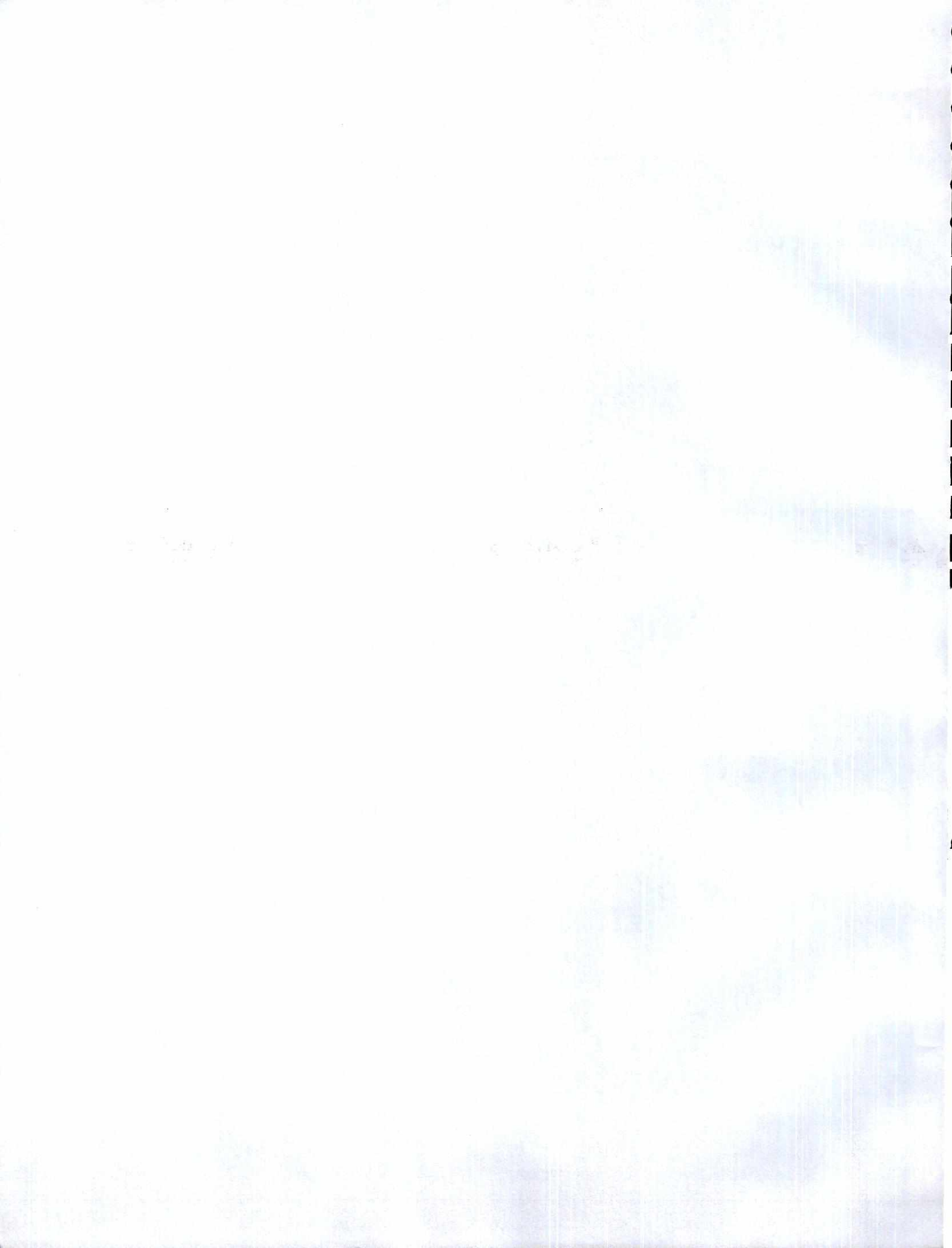
Técnica o Método	Área de aplicación	Cómo aplicarlo en pequeñas y medianas organizaciones guatemaltecas
CMMI: Ingeniería de Software	Mejoramiento organizacional aplicado a software	Usarlo como guía para identificar prácticas específicas y genéricas que son importantes en el proceso de desarrollo de software
Patrón negociador latinoamericano	Negociación interna	<ul style="list-style-type: none"> • Para convertir conflictos en procesos negociadores • Entrenando al equipo de software y a los clientes internos antes de iniciar cualquier proyecto de software
Grados de Ignorancia	Expertaje de dominio	Identificando al(los) experto(s) antes de: <ul style="list-style-type: none"> • Iniciar proyecto de software • Incursionar en cualquier área
Valoración de requerimientos	Recolección y validación de requerimientos	<ul style="list-style-type: none"> • Usando selección múltiple • Orientando discusiones y negociaciones • Identificando palabras claves
HOQ QFD	Validación y documentación de requerimientos	Agrupando requerimientos afines con su criterio de aceptación y valor
Análisis de Riesgos	Administración de proyectos de software	Analizarlos continuamente durante todo el proceso de producción de software
Inspecciones Formales	Control de Calidad	Aplicarlas continuamente durante todo el proceso de producción de software

Las organizaciones pequeñas y medianas no pueden permitirse un largo proceso de prueba y error en la construcción de sistemas de información que apoyen el trabajo interno. Por tal motivo, este trabajo resalta la importancia técnica de los requerimientos, la negociación, comunicación, análisis de riesgos y control de calidad, con el propósito que sea de utilidad tanto para equipos de desarrollo de software como para gerentes y propietarios de organizaciones guatemaltecas.

Según el análisis presentado anteriormente, para hacer eficiente el manejo de requerimientos con clientes internos para el desarrollo y mantenimiento de sistemas de información, se presenta a continuación en la Cuadro 8 un resumen de técnicas, el área de aplicación y uso recomendado.

VI. GLOSARIO

- CMMI:** *Capability Maturity Model Integration*; modelo desarrollado por el Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon, U.S:A.
- HOQ:** *House of Quality*; cuadro de referencias cruzadas del modelo de QFD.
- QFD:** *Quality Function Deployment*; metodología para validación de requerimientos.
- Organización:** Estructura administrativa de una o más personas, que como grupo, administran uno o más proyectos, comparten el mismo líder y actúan con las mismas políticas.
- Software:** Producto que está destinado a usarse en un computador



VII. BIBLIOGRAFÍA

- Altamirano, J., 2003 <Calidad Total y su Aseguramiento Mediante la Norma NMX-CC/ISO 9000> Facultad de Química, Universidad Autónoma de México. <http://tenoch.pquim.unam.mx/academico/economia/calidad/manual.htm>
- Armour, P., 2000. <<The five orders of ignorance>>. Communications of the ACM. New York, USA. Association for Computing Machinery.
- Berry, L., 1995. On Great Service: A framework for Action. New York, USA. The Free Press. pp 292
- Block, P., 1989. <<Como entender la resistencia>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.
- ____; 1989. <<Como tratar con la resistencia>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.
- Christel, M., Kang, K., 1992. <<Issues in requirements elicitation>>. Technical Report CMU/SEI-92-TR-12. Pennsylvania, USA. <http://www.sei.cmu.edu>.
- Clarke, S. Et al., 1999. <<Subject-Oriented design: Towards improved alignment of requirements, design and code>>. OOPSLA. New York, USA. Association for Computing Machinery.
- CMMI Product Team, 2002. Capability Maturity Model Integration (CMMI) Version CMU/SEI-2002-TR-011. Software Engineering Institute. Pennsylvania, USA. Carnegie Mellon University.
- Dawson, L., Swatman, P., 2000. <<The use of object-oriented models in requirements engineering: A field study>>. Proceedings of the 22nd international conference on information systems. Limerick, Ireland. Association for Computing Machinery.
- Dawson, R., 2000. <<Twenty dirty tricks to train software engineers>>. Limerick, Ireland. Association for Computing Machinery.
- Diccionario de la lengua española. 1970. Decimo novena edición. Madrid, España. Real Academia Española de la Lengua.
- Simon and Schuster's International Dictionary. 1973. New York, USA. Simon and Schuster's, Inc.

- Disney Institute, 1990. Service, Disney Style. USA. Disney Institute.
- Garcia, S., 1999. <<Successful use of the Software Capability Maturity Model in a Small Organization>>. Technical Program Submission ASQ, Software Division's, 9th International Conference on Software Quality. Pennsylvania, USA. <http://seir.sei.cmu.edu>.
- Gause, D., Weinberg, G., 1989. Exploring Requirements: Quality before Design. New York, USA. Dorset House Publishing. pp 299
- Harrington, H. J., 1993. Mejoramiento de los procesos en la empresa. Bogotá, Colombia. McGraw-Hill. pp 309
- Kanter, R. M., 1985. <<El manejo del aspecto humano del cambio>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.
- Kelliher, T. Et al., 2001. Six Sigma meets Extreme Programming: Changing the way we work. GE Corporate R&D. New York, USA. <http://seir.sei.cmu.edu>.
- Kuusela, J., Savolainen, J., 2000. <<Requirements engineering for product families>>.. Proceedings of the 22nd international conference on software engineering. Limerick, Ireland. Association for Computing Machinery.
- Maliyakkal, S., 1998. <<Requirements elicitation>>. Seminario Methoden und Werkzeuge zur Softwareproduktion. Berlin, Alemania. <http://www.citeseer.com>.
- Microsoft Solution Framework: Principles of Application Development. 1999. Course Number 1516A. Microsoft Corporation, USA
- Mazur, G., 1994. <<QFD for small business: A shortcut through the "Maze of Matrices". The Sixth Symposium on Quality Function Deployment. Michigan, USA. <http://www.mazur.net>.
- _____; 1997. <<9 House of quality checks>>. Japan Business Consultants, Ltd. Michigan, USA. <http://www.mazur.net>.
- _____; 1998. <<Strategy deployment for small and medium enterprises>>. University of Michigan. Michigan, USA. <http://www.mazur.net>.
- Nuseibeh, B., Easterbrook, S., 2000. <<Requirements engineering: A roadmap>>. Proceedings of the conference on the future of software engineering. Limerick, Ireland. Association for Computing Machinery.
- Ogliastri, E., 1997. <<La nueva manera de negociar: un decálogo>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.

- ____; 1998. <<El estilo negociador de los latinoamericanos. Una investigación cualitativa>>. Programa de Alta Gerencia. Alajuela, Costa Rica. INCAE.
- Over, J., Burton, D., 1998. <<The Personal software process (PSP): A full day tutorial>>. Software Engineering Institute Repository. Pennsylvania, USA. <http://seir.sei.cmu.edu>.
- Quality Function Deployment. 2002. Quality Function Deployment Institute. <http://www.qfdi.org>.
- Savolainen, J., Kuusela, J., 2001. <<Volatility analysis framework for product lines>>. SSR. Ontario, Canada. Association for Computing Machinery.
- Theelin, T., 2000. <<Teams in Software Development>>. Department of Communication Systems. Sweden. Lund University.
- Travassos, G., Shull, F., Fredericks, M., Basili, V., 1999. <<Detecting defects in object oriented designs: Using reading techniques to increase software quality>>. Proceedings of the 1999 ACM SIGPLAN conference on object-oriented programming, systems, languages and applications. New York, USA. Association for Computing Machinery.
- Tschohol, J., Franzmeier, S., 1997. Servicio al Cliente: El arma secreta de la empresa que alcanza la excelencia. México. Editorial Pax México. pp 351
- van Lamsweede, A., 2000. <<Requirements engineering in the Year 00: A research perspective>>. Proceedings of the 22nd international conference on software engineering. Limerick, Ireland. Association for Computing Machinery.
- Wiersema, F., 1996. Customer Intimacy. California, USA. Knowledge Exchange.
- Woodruff, W., 1997. <<Requirements management for small organizations: A field guide to effective requirements management under SEI's CMM>>. Software Engineering Institute Repository. Pennsylvania, USA. <http://seir.sei.cmu.edu>.
- Software formal inspections guidebook. 1993. NASA-GB-A302. Washington, DC, USA. Office of safety and mission assurance, National Aeronautics and space administration. pp 68
- Extreme Programming. 2002. <http://www.extremeprogramming.org>