Investigación sobre las prácticas de ingeniería de software en México

RESUMEN

El objetivo de la presente investigación es identificar los principales elementos que conforman la gestión del software y con ello diagnosticar el estado actual de la producción de software en las organizaciones. Para su elaboración se llevó a cabo un proceso de consulta con la comunidad de software en colaboración con la Asociación Mexicana de Calidad para la Ingeniería de Software (AMCIS) y el Laboratorio de Sistemas de Información del CIC-IPN durante el mes de agosto del 2001.

Se identificó el contexto en el cual se encuentra la organización y la administración de los proyectos. La parte de administración de proyectos, se enfocó en las áreas de control tales como: tecnología, los costos, los planes, y el desarrollo del producto, requiere mediciones comprensivas para la toma de decisiones dentro de los proyectos, y finalmente la perspectiva de la calidad de software(modelos de calidad, factores que influyen en la toma de decisiones, atributos que indican calidad de los productos). Finalmente se presentan las conclusiones del análisis de las actividades de ingeniería de software en la ciudad de México.

Presentada por:
María de la Luz Villalobos Hernández. ESCA-IPN
e-mail:luzv@rocketmail.com
Agustín Francisco Gutiérrez Tornés CIC-IPN.

Palabras clave : Calidad, ingeniería de software, control, administración de proyectos.

1.Introducción

La industria del software posee una gran tolerancia por parte de sus consumidores. Mantiene una visión de los esfuerzos actuales y de la forma en que facilitan las bases para trabajar en línea con los usuarios cuyos proyectos son afectados por los cambios en la práctica de la ingeniería de software, proporciona una amplia perspectiva de los esfuerzos por mejorar y ayudar en sus expectativas.

Un importante indicador de la creciente participación de la informática en la economía mexicana, es el Producto Interno Bruto Informático (PIB), el cual creció 27.2% en términos reales en el año 2000 respecto a 1999, esto es 4 veces más que la economía en su conjunto. Con ello el sector informático participa con 3.5% del total de la economía. A su interior el sector servicios profesionales en informática creció 4.9% en términos reales [1]. Las organizaciones dedicadas al servicio de "análisis de sistemas y procesamiento informático", se encuentran en su mayoría en el Distrito Federal en donde se ubican casi 5 de cada 10 empresas del ramo.

El cambio acelerado y de competitividad global que vive el mundo, donde la liberalización de las economías y la libre competencia caracterizan el entorno de convivencia para el sector empresarial, ha originado que las empresas se preocupen por ser más eficientes, brindando productos y servicios de calidad. Desde la década de los años setenta, el tema de la calidad ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, realizando investigaciones con dos objetivos fundamentales: la obtención de software con calidad y la evaluación de la calidad del software.

El área de gestión y control de la calidad en proyectos de software en México es relativamente nueva y las empresas si bien reconocen la importancia de la calidad, no se encuentran suficientemente preparadas para aceptar los nuevos retos que trae consigo y poner en práctica sus principios y técnicas. Con el objeto de diagnosticar y orientar las acciones de las prácticas de ingeniería de software, se realizó una investigación acerca de la perspectiva de la administración de proyectos informáticos, y el impacto del tema de calidad del software como ventaja competitiva. Este reporte, sintetiza los resultados de una consulta con ciudadanos vinculados con la actividad.

2.Aseguramiento de la calidad de software

El aseguramiento de la calidad es una actividad inmersa en todo el proceso de ingeniería de software, cubriendo aspectos tales como: métodos y herramientas de análisis, diseño, codificación y prueba; revisiones técnicas formales; el control de la documentación del software y de los cambios realizados; un procedimiento que asegure un ajuste a los estándares de desarrollo de software(cuando sea posible); y mecanismos de medida y de información[2].

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, para lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, y elevar la productividad.

Para controlar la calidad del software es necesario, definir los parámetros, indicadores o criterios de medición, Tom De Marco[3] plantea: "usted no puede controlar lo que no se puede medir". Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes.

3.Metodología

Al buscar estadísticas acerca de las empresas que desarrollan software en la ciudad de México, los datos encontrados en el SIEM[A1], del censo de 1993 bajo el rubro de unidades económicas por sector según clase de actividad contempla 224 empresas prestadoras de servicios de análisis de sistemas y procesamiento informático. Sin embargo, éstas empresas no revelan el total actual de las empresas que desarrollan software, pues no contempla instituciones gubernamentales, instituciones educativas, de investigación, etc. Al no encontrar estadísticas acerca de la forma como es producido el software, se diseñó un instrumento piloto que se aplicó a 25 líderes de proyecto como muestra. Esta primera fase, reveló que las personas encuestadas desconocían la forma en como la calidad total podría influir en la gestión de los proyectos. Por tanto, el instrumento fue ajustado para aplicarlo no sólo a los líderes de proyecto, sino a consultores, programadores, administradores de bases de datos, etc., para obtener una visión más amplia.

Para su elaboración se llevó a cabo un proceso de consulta con la comunidad. Enfocándose en dos partes esenciales: los antecedentes para ubicar el contexto en el cual se encuentra la organización y la gestión de los proyectos. La segunda parte, se enfocó en las áreas de control de los proyectos tales como: tecnología, los costos, los planes, y el desarrollo del producto, requiere mediciones comprensivas para la toma de decisiones dentro de los proyectos, y finalmente la perspectiva de la calidad de software(modelos de calidad, factores que influyen en la toma de decisiones, atributos que indican calidad de los productos).

El cuestionario de 20 preguntas[A2], se extendió a las organizaciones que producen software en la ciudad de México, invitando a una población de 100 personas en colaboración con la Asociación Mexicana de Calidad para la Ingeniería de Software (AMCIS) y el Laboratorio de Sistemas de Información del CIC-IPN y en tres semanas proporcionó una respuesta favorable por parte de los encuestados, se descartaron 28 cuestionarios por no estar completamente contestados, quedando solo 72, cuyos resultados son presentados a continuación.

4.Presentación de resultados

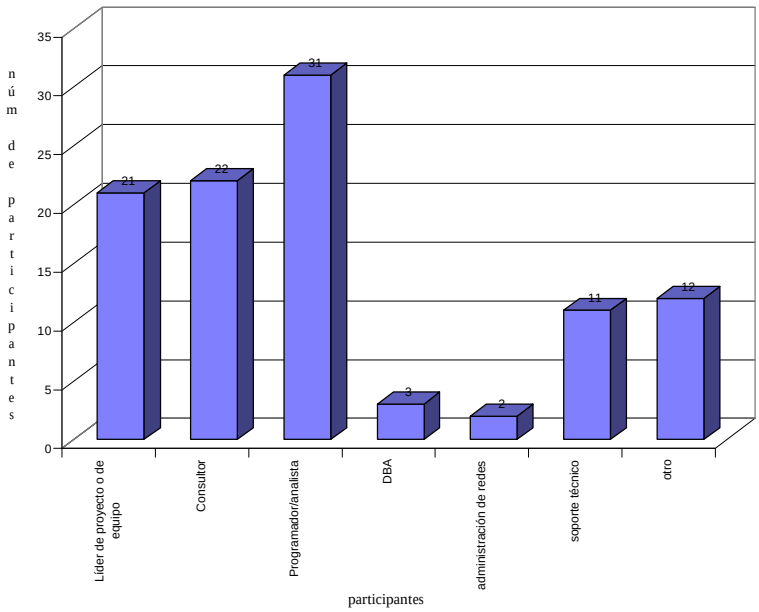
Para comprender las perspectivas de los encuestados, se clasificó a la población según su posición actual en la organización tal y como se muestra en la tabla y gráfica1.

Tabla 1. Posición que ocupa el participante dentro de la organización.

Posición actual	Porcentaje (%)
1. Líder de proyecto	20.58
2. Consultor	21.56
3. Programador/analista	30.39
4. Administrador de bases de datos(DBA)	2.94
5. Administración de redes	1.96
6. Soporte técnico	10.78
7. Otro	11.76

Fuente: Elaboración propia, agosto 2001

Gráfica 1. Posición actual.



Fuente: Elaboración propia, agosto 2001.

Los encuestados podían marcar varias opciones, dependiendo de las funciones que desempeñan dentro de su proyecto actual. El mayor porcentaje lo ocuparon los programadores/analistas, seguidos de los líderes de proyecto y consultores, los encargados de brindar soporte técnico y en solo un 4.9% los administradores de red y de las bases de datos, también participaron con un 11.76% en auditores de sistemas, gerentes, encargados del área de sistemas, investigadores, etc., agrupado en otros.

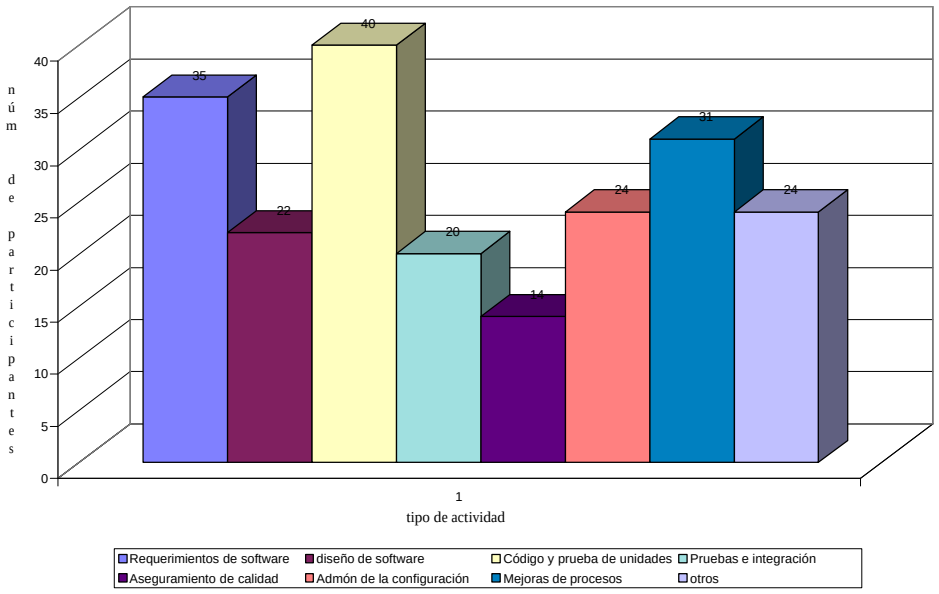
También fue necesario identificar las actividades donde se desempeñaban, considerando que podrían desempeñarse en actividades simultáneas, se propusieron 7 actividades generales y se dio un margen para que los encuestados mencionarán alguna otra actividad no contemplada. Ver tabla y gráfica 2.

Tabla 2. Actividades que desempeña el participante.

Actividades	**Porcentaje (%)**
1.Requerimientos de software	16.66
2.Diseño de software	10.47
3.Código y pruebas de unidades	19.04
4.Pruebas e integración	9.52
5.Aseguramiento de la calidad	6.66
6.Administración de la configuración	11.42
7.Mejoras de los procesos	14.76
8.Otro	11.42

Fuente: Elaboración propia, agosto 2001.

Gráfica 2. Actividades en las que se desempeña



Fuente: Elaboración propia, agosto 2001.

Se dividieron en dos rubros las actividades: básicas y de aseguramiento de la calidad. En las actividades básicas la fase de análisis, diseño, codificación y pruebas integraban un 55.69%, en cambio las actividades de aseguramiento de la calidad, administración de la configuración, mejoras de los procesos integraban solo un 32.86%. El 11.42% restante abarca las funciones de auditorías e investigación.

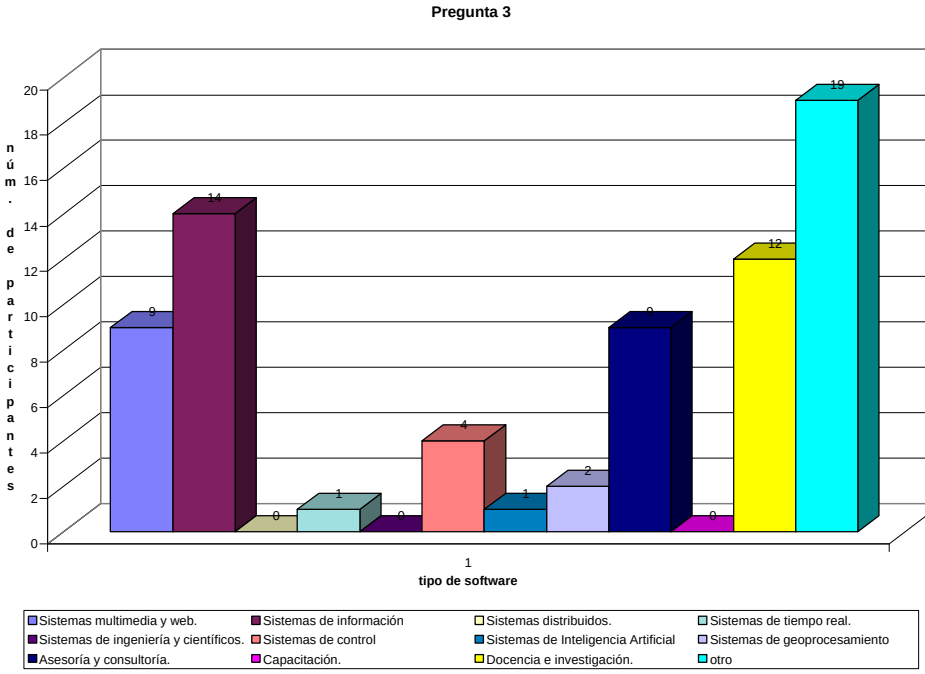
La mayoría de las organizaciones poseen diferentes proyectos o contratos que son tratados de manera simultánea. Para poder hacer correlaciones entre la perspectiva de las respuestas proporcionadas y las expectativas de métodos de control, era necesario identificar el giro de la empresa, es decir, conocer el tipo de software que se desarrolla, por ello se definió 10 giros principales. Ver gráfica y tabla 3.

Tabla 3. Giro de la empresa

Giro	**Porcentaje (%)**
1.Multimedia y web	12.67
2.Sistemas de información	19.71
3.Sistemas distribuidos	0
4.Sistemas de tiempo real	1.40
5.Sistemas de ingeniería y científicos	0
6.Sistemas de control	5.63
7.Sistemas de inteligencia artificial	1.40
8.Sistemas de geoprocesamiento	2.81
9.Asesoría y consultoría	12.67
10.Capacitación	0
11.Docencia e investigación	16.9
12.Otro	26.76

Fuente: Elaboración propia, agosto 2001.

Gráfica 3. Giro de las empresas



Fuente: Elaboración propia, agosto 2001.

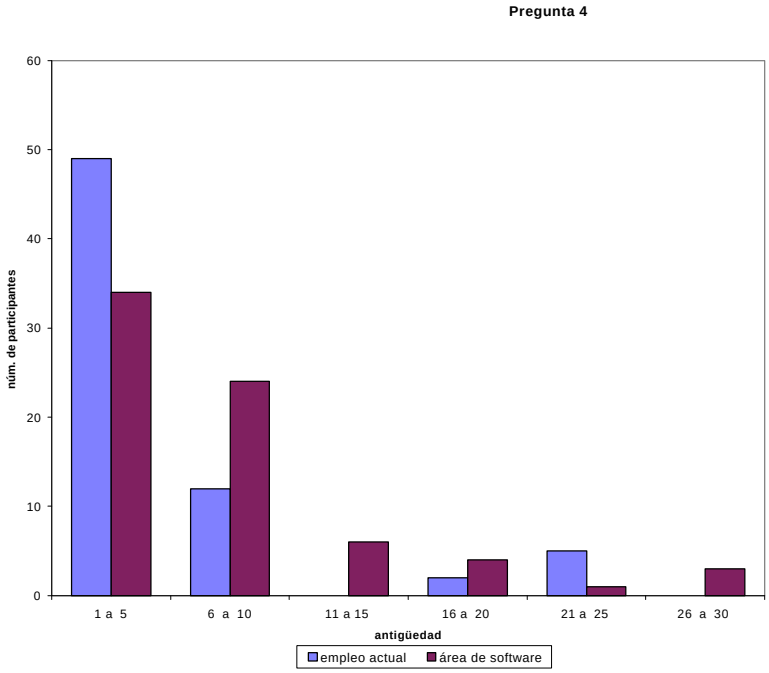
De los líderes de proyecto y las áreas donde se desempeñan el 28.57% es en el área de sistemas de información, y el 23.8% en las áreas de asesoría y otras actividades, en cambio los consultores el 27.7% se dedican al área de docencia y el 18.18% al desarrollo de proyectos multimedia y web.

4.1.Experiencia laboral

Se pidió a las participantes revelaran el número de años que llevan laborando en la presente empresa, así como el número de años de experiencia que poseen en el área de software.

El tiempo promedio de años laborando en la empresa actual es de 5, con un rango de 1 a 24 años, en el caso de los años de experiencia en el área de software fue de 7.84, con un rango de 1 a 30 años. Según datos de la ANUIES[4] en 1999, la población escolar de posgrado por área de estudio en Computación Sistemas manejaba 2857 personas cursando especializaciones, maestrías y doctorados lo cual indica una creciente demanda de profesionales altamente calificados. Varios de los encuestados poseían menos años de experiencia en el área de software a pesar de estar laborando por mas años en la misma empresa. Ver gráfica 4.

Gráfica 4. Experiencia laboral



Fuente: Elaboración propia, agosto 2001.

4.2.Construcción de equipos de trabajo

El método de organización en equipos de producción de software tiene sus orígenes en el concepto de equipo: jefe-programadores,propuesto inicialmente por Harlan Mils y descrito por Baker [5]. El núcleo del equipo está compuesto por un ingeniero senior (programador jefe), el cual planifica, dirige el análisis y las actividades de desarrollo, coordina y revisa todas las actividades técnicas del equipo; un personal técnico (normalmente de dos a cinco personas), y por un ingeniero de apoyo, que ayuda al ingeniero senior en sus actividades y puede reemplazarle con una pérdida mínima de continuidad del proyecto.

El enfoque de la calidad total radica en decrementar el tiempo en el ciclo de vida, incrementar la productividad, crear mayor satisfacción en los clientes y éxito en los negocios. El cambio consistirá en definir lo que realmente significa el enfoque de la calidad para dirigirla[6]. Partiendo de las estadísticas de la SECOFI[7], donde se postula que la mayor parte de las empresas de servicios en México se componen de micro y pequeñas empresas, se trató de demostrar que las empresas dedicadas al área de producción de software en México son micros y pequeñas empresas.

Yourdon [8], propone la siguiente esquematización de las organizaciones de software:

Pequeños. Utilizan 1 programador de 1 a 6 meses. Los programas pequeños no tienen interacción con otros programas; los estándares técnicos de documentación y notaciones, así como las revisiones sistemáticas de los proyectos deben usarse, aunque el grado de formalidad será menor al empleado en proyectos grandes.
Medianos. Emplean de 2 a 5 programadores con duración de menos de un año, exigen la interacción entre programadores y la comunicación con los usuarios; de ahí que se necesite cierta formalidad en la planeación, documentación y revisión del proyecto.

Grandes. En los proyectos grandes son de 5 a 20 programadores, en ellos surgen los problemas de comunicación, además existe la gran posibilidad de alta rotación de personal asignado al proyecto durante su desarrollo; lo anterior requiere del entrenamiento y la adoctrinación del personal nuevo o de la distribución de responsabilidad entre los integrantes del grupo. Además, son esenciales los procesos sistematizados, documentación estándar y revisiones formales.

Como referencia, también se utilizó el criterio de pequeños proyectos software empleado en CMM. Ver tabla 4.

Tabla 4 Propuesta de grupos de trabajo para desarrollar software en micro y pequeñas empresas.

Variante de pequeño	Número de personas	Tiempo estimado
Pequeño	3-5	6 meses
Muy pequeño	2-3	4 meses
Diminuto	1-2	2 semanas
Indidivual	1	1 semana
Ridículo	1	1 hora

Fuente CMM de SEI version 1.1 [5]

En 1998 en el panel de la conferencia de SEPG en el CMM para pequeños proyectos [9], proyecto pequeño se definió como una duración de 3 a 4 meses con 5 personas o menos. Brodman y Johnson[10] definen una pequeña organización como aquella que posee menos de 50 desarrolladores de software y un pequeño proyecto como aquel menor de 20 desarrolladores.

El equipo desarrollo de software puede estar asesorado por uno o más especialistas (por ejemplo: expertos en telecomunicaciones, diseñadores de bases de datos,etc.), por una plantilla de soporte(por ejemplo, documentadores, técnicos , ejecutivos,etc.,) y por un administrador de base de datos. El tiempo estimado para realizar los proyectos de software se categorizó en 5 rangos. Ver tabla y gráfica 5.

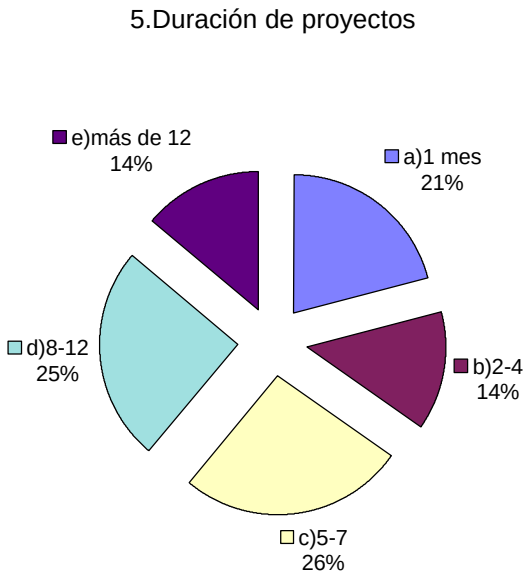
Tabla 5. Tiempo promedio de duración de los proyectos

Tiempo estimado	Porcentaje (%)
1 mes	20.83%
2 a 4 meses	13.88%
5 a 7 meses	26.38%
8 a 12 meses	25%
más de 12 meses	12.5%

Fuente: Elaboración propia, agosto 2001.

Los programadores, analistas y demás miembros del grupo de sistemas de una empresa adoptan las directrices del líder de grupo, o de los miembros del equipo con mayor experiencia que hayan trabajado en buenos equipos y poseen alguna idea de cómo actuar. Sin embargo, en la mayoría de los casos los equipos tienen que atravesar por una variedad de sucesos por ejemplo: las metas, los roles de cada uno de los miembros del equipo, sus responsabilidades, como se tomaran las decisiones, los estándares y procedimientos que ocuparán y en que forma lo harán, los objetivos de calidad, el seguimiento de la calidad y que se hará para alcanzarla, los procesos que se utilizarán para desarrollar el producto, la estrategia de desarrollo y como se debe producir el diseño, la integración y las pruebas de producto, etc.

Gráfica 5. Tiempo estimado para realizar sus proyectos.



Fuente: Elaboración propia, agosto 2001.

El número de personas que componen el área de sistemas dentro de la empresa se categorizó por intervalos. Ver tabla y gráfica 6.

Tabla 6. Número de personas que componen el área de desarrollo de software

Personal empleado para desarrollar software	**Porcentaje (%)**
• 1 a 5	38.88
• 6 a 10	16.66
• 11 a 15	6.94
• 16 a 21	8.33
• más de 21	29.16

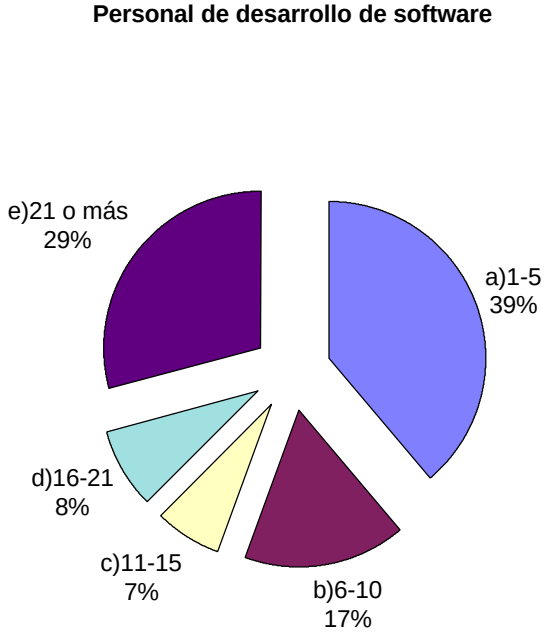
Fuente: Elaboración propia, agosto 2001.

Existen tantas estructuras de organización de personal de sistemas como organizaciones. Esto ayuda a comprobar que la mayoría de las empresas son micro y pequeñas. En un pequeño proyecto de software, una sola persona puede analizar los requerimientos, realizar el diseño, generar el código y llevar a cabo las pruebas. Cuando se manejan aplicaciones para multimedios y herramientas para desarrollos web, el personal requerido oscila entre 1 y 5, en cambio para la construcción o actualización de módulos de sistemas de mayor complejidad se pueden ocupar más de 21 personas, como en los casos de las empresas de consultoría de sistemas de gestión, donde atienden varios clientes a la vez. Ver gráfica 6.

En proyectos grandes, el número de programadores es tan amplio que las diferencias individuales en la productividad tienden a equilibrar el promedio, aunque los módulos desarrollados por un programador débil pueden exhibir pobre calidad y retrasarse en su entrega. Los proyectos pequeños y medianos, de menos de 5 personas, son muy sensibles a la capacidad individual de los programadores. A medida que el tamaño aumenta, debe involucrarse más personal.

Desafortunadamente, el añadir gente a un proyecto tiene a menudo un efecto destructivo en el mismo, haciendo incluso que la agenda se alargue más, pues mientras se enseña no se trabaja, y el proyecto se retrasa. Debido al incremento de la oferta de servicios informáticos, las organizaciones no pueden tardarse más de 3 a 4 meses en elaborar las aplicaciones

Gráfica 6. Personal que compone el área de software



Fuente: Elaboración propia, agosto 2001.

4.3. Administración de los procesos de software

Los procesos de software definen las principales actividades que contribuyen a su desarrollo, la selección o el mantenimiento de un producto de software ajustado para darle un servicio. También define las relaciones y las interacciones que pueden existir entre las actividades, y los documentos que son resultado de dichas actividades.

Para ésta sección, se delimitaron 4 posibles respuestas para que los encuestados emitieran su opinión en base a los siguientes criterios:

1. Si. Cuando la práctica está bien establecida y se ejecuta consistentemente como un procedimiento estándar de operación.
2. No. La práctica no está bien establecida o es realizada inconsistentemente.
3. No aplica. Posee el conocimiento requerido acerca del proyecto u organización pero la pregunta no es aplicable al proyecto actual del encuestado.
4. Desconoce. Posee incertidumbre de cómo responder a la pregunta.

4.3.1.Plan documentado

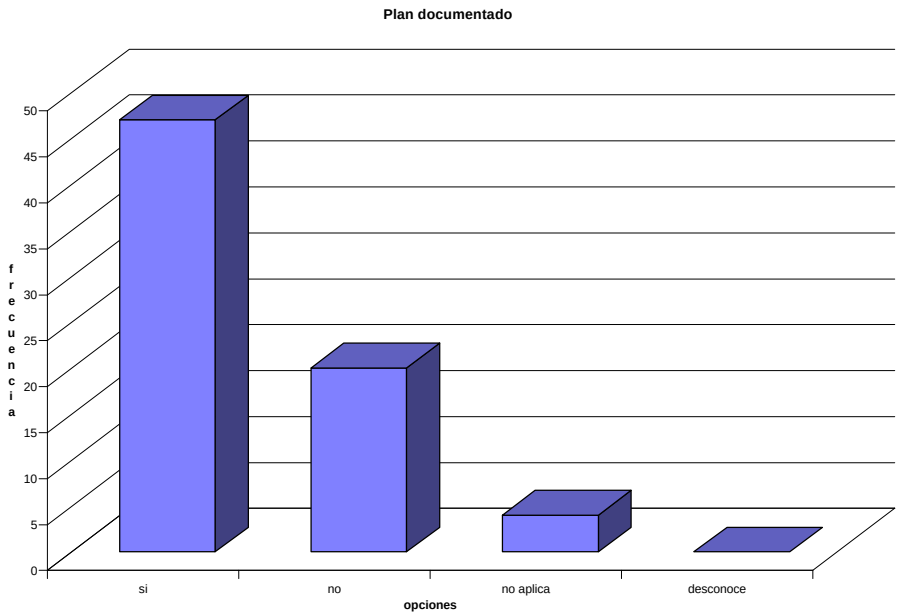
Para tener una buena administración por proyectos se requiere que el analista o el programador y su jefe inmediato elaboren un plan de trabajo en el cual se especifiquen actividades, metas, personal participante y tiempos. Este plan debe ser revisado periódicamente (semanal, mensual, etc.) para evaluar el avance respecto a lo programado. Ver gráfica y tabla 7.

Tabla 7. Seguimiento de algún plan documentado que guíe el desarrollo para elaborar un proyecto.

Opciones	Porcentaje (%)
• **Sí**	66.19
• **No**	28.16
• **No aplica**	5.63
• **Desconoce**	0

Fuente: Elaboración propia, agosto 2001.

Gráfica 7. Aplicación de planes documentados para construir software.



Fuente: Elaboración propia, agosto 2001.

Sólo en un 66.19% se sigue un plan documentado para desarrollar un proyecto. Aunque en las organizaciones cuente con los datos necesarios para realizar los planes de desarrollo, sus programas están basados en amplias suposiciones las cuales a menudo son irreales.

4.3.2.Análisis de requerimientos

El análisis de los requerimientos requiere de gran comunicación y pueden surgir problemas tanto para el analista como para el cliente. Es un proceso de descubrimiento y refinamiento. Tanto el que desarrolla el software como el cliente, tienen un papel activo en la especificación de requerimientos. El cliente intenta reformular su concepto, algo nebuloso, de la función y comportamiento de los programas en detalles concretos, el que desarrolla el software actúa como interrogador, consultor y el

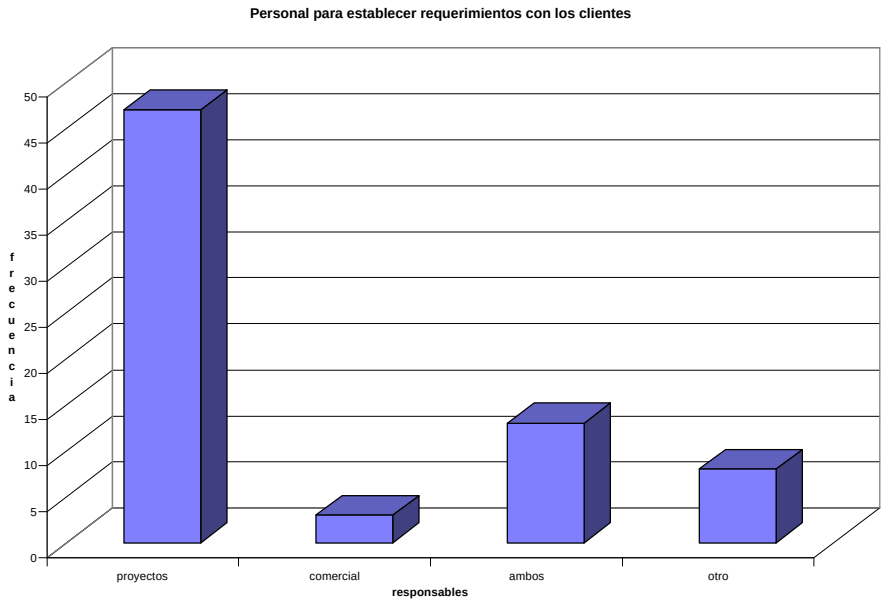
que resuelve los problemas. El analista debe establecer contacto con el equipo técnico y de gestión del usuario/cliente y con la empresa que vaya a desarrollar el software. El gestor del programa puede servir como coordinador para facilitar el establecimiento de los caminos de comunicación. El objetivo del analista es reconocer los elementos básicos del programa tal como lo percibe el usuario/cliente[11]. El software por su naturaleza es muy cambiable. Ver tabla y gráfica 8.

Tabla 8. Personal encargado de definir los requerimientos junto con el cliente

Opciones:	**Porcentaje (%)**
• **Líder de proyecto**	66.19
• **Líder comercial**	4.22
• **Ambos**	18.30
• **otro**	11.26

Fuente:Elaboración propia, agosto 2001.

Gráfica 8. Personal que realiza el establecimiento de los requerimientos con los clientes.



Fuente:Elaboración propia, agosto 2001.

En la gráfica partiendo de la hipótesis de que el análisis de los requerimientos lo realizaban los líderes comerciales y no los líderes de proyecto, al investigar, se demostró que el 65.27% de las entrevistas con los clientes las realizan los líderes de proyecto, mientras el 4.16% era realizado por los líderes comerciales, y solo el 18.05% lo realizaban en conjunto los líderes comercial y de proyecto con el cliente.

4.3.3.Evaluación de tecnologías

Existen varias técnicas para hacer la evaluación de las tecnologías, las cuales pueden ser útiles en los seguimientos de los procesos y así compararlos con procesos similares utilizados por otros individuos, grupos u organizaciones. Primero se busca realizar procesos de medición de tecnologías que sean independientes de los procesos y que refleje su capacidad y robustez. (ver tabla y gráfica 9.)

Tales comparaciones deben ser válidas para los diferentes tipos de procesos en los siguientes términos:

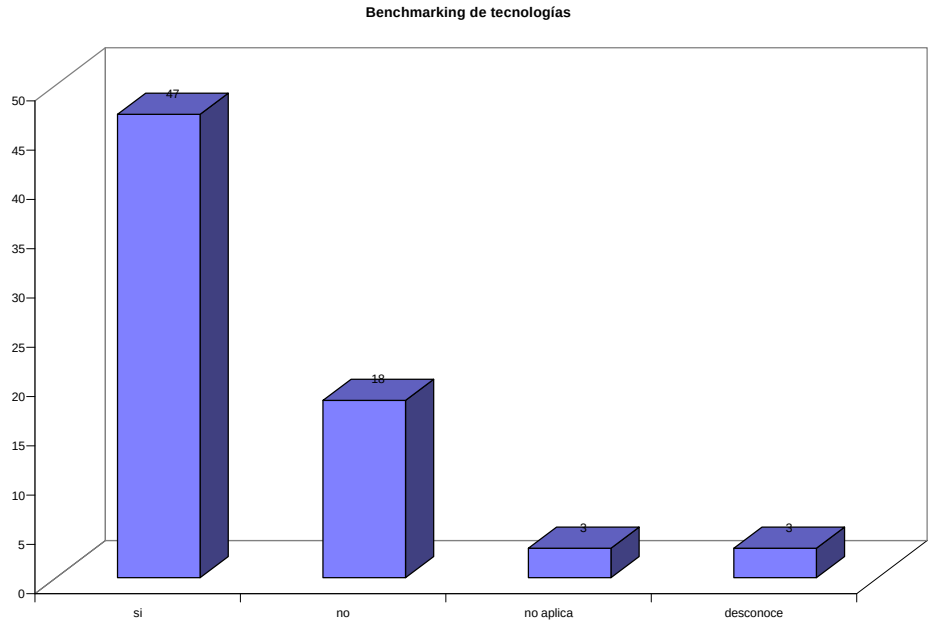
- Medir la habilidad de la tecnología para elaborar productos de alta calidad.
- Organizar la información del mejor al peor.
- Indicar la habilidad de las tecnologías para eliminar problemas.
- Estimar los costos.

Tabla 9. Revisión de las tecnologías existentes en el mercado antes de comenzar un proyecto.

Opciones:	**Porcentaje (%)**
• **Sí**	66.19
• **No**	25.35
• **No aplica**	4.22
• **Desconoce**	4.22

Fuente: Elaboración propia, agosto 2001.

Gráfica 9. Aplicación de una revisión de tecnologías existentes en el mercado para ofrecer una buena solución al cliente



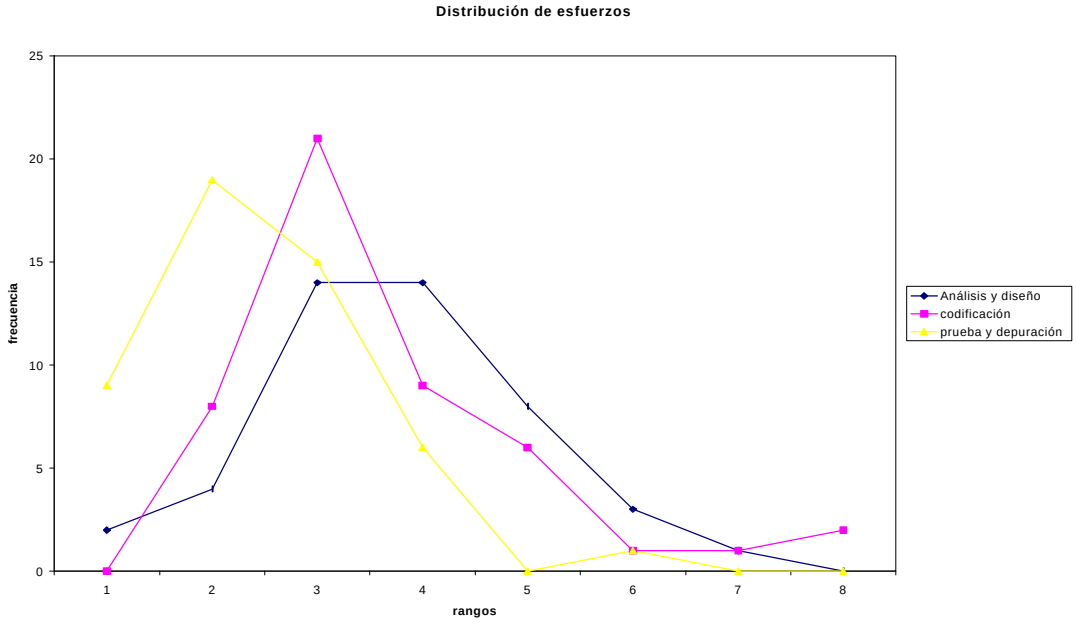
Fuente: Elaboración propia, agosto 2001.

El nivel tecnológico utlizado en un proyecto de programación incluye aspectos como selección del ambiente computacional, prácticas de programación y herramientas de programación disponibles. Al inicio del proyecto se revisan las nuevas tecnologías existentes en el mercado para determinar la más adecuada a las necesidades del cliente donde el 61.9% de los líderes de proyecto si evalúa las tecnologías, mientras que solo el 59% de los consultores lo efectúan.

4.3.4.Prácticas de software

Si nadie observa los métodos que los ingenieros de software utilizan, nadie más sabrá como trabajan. Entonces los ingenieros no tienen que cambiar sus métodos de trabajo si no lo desean. Las prácticas de software son largas por sí mismas e informales. Roger Pressman[2] ilustra una distribución 40-20-40 recomendada de esfuerzos entre las distintas fases de definición y desarrollo. Pone un mayor énfasis en las tareas iniciales de análisis y diseño y en la tarea terminal de prueba.

Gráfica 10. Porcentajes de tiempo invertidos para las fases del ciclo de vida



Fuente: Elaboración propia, agosto 2001.

En términos generales el porcentaje de tiempo invertido en las actividades de análisis y diseño es de 37.18%, para la etapa de codificación un 35.67%, y finalmente para la etapa de pruebas un 23.33% Dependiendo de lo crítico que sea el software, así será la cantidad de prueba que se requiera(ver gráfica 10). Dichos registros muestran el tiempo esperado para desarrollar el software. Los líderes y consultores invierten más tiempo en diseñar, en cambio los programadores invierten mas tiempo en compilar y corregir errores que en cualquier otra actividad.

4.3.5.Capacitación

Uno podría preguntarse porque la comunidad de software ha sido tan lenta para adoptar los principios de calidad. La respuesta aparece en que estos métodos son difíciles de introducir y no obvios por sí mismos. Por ejemplo, pocos ingenieros creen que es más eficiente el encontrar los defectos por las revisiones de código, que por las pruebas y la depuración.

Las organizaciones deben proporcionar la capacitación adecuada a su personal.

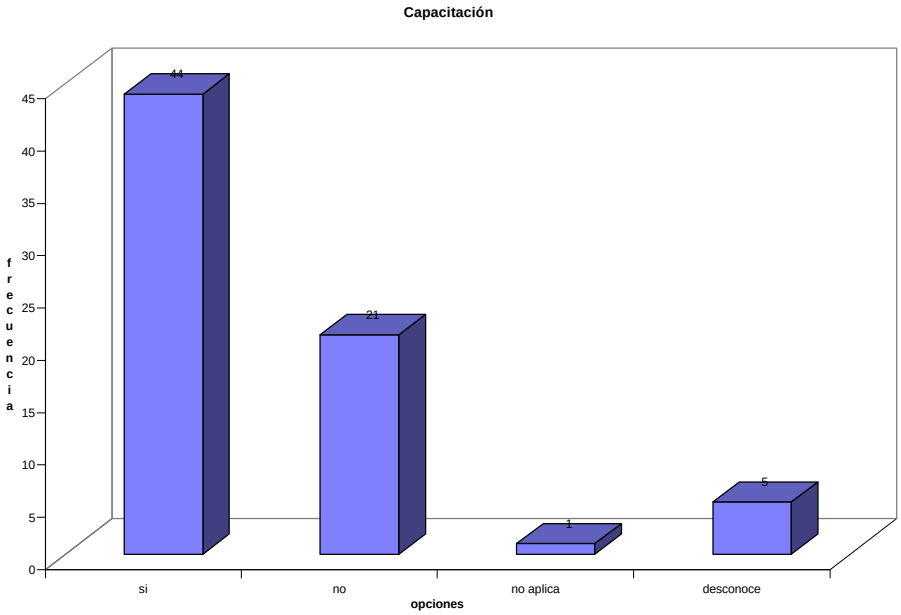
Los líderes de proyecto reciben capacitación solo un 61.9%, en contraste con los investigadores que no reciben apoyos para programas de capacitación en las instituciones en donde laboran[12], en cambio los consultores si reciben capacitación en un 63%. Los administradores de bases de datos y los encargados del soporte técnico reciben capacitación en un 90%. Ver gráfica 11 y tabla 10.

Tabla 10. La capacitación proporcionada es la necesaria para desarrollar las bases y el conocimiento requerido para desempeñar sus actividades

Opciones:	**Porcentaje (%)**
• **Sí**	61.97
• **No**	29.57
• **No aplica**	1.40
• **Desconoce**	7.04

Fuente: Elaboración propia, agosto 2001.

Gráfica 11. Capacitación proporcionada.



Fuente: Elaboración propia, agosto 2001.

4.3.6.Cambios en los proyectos

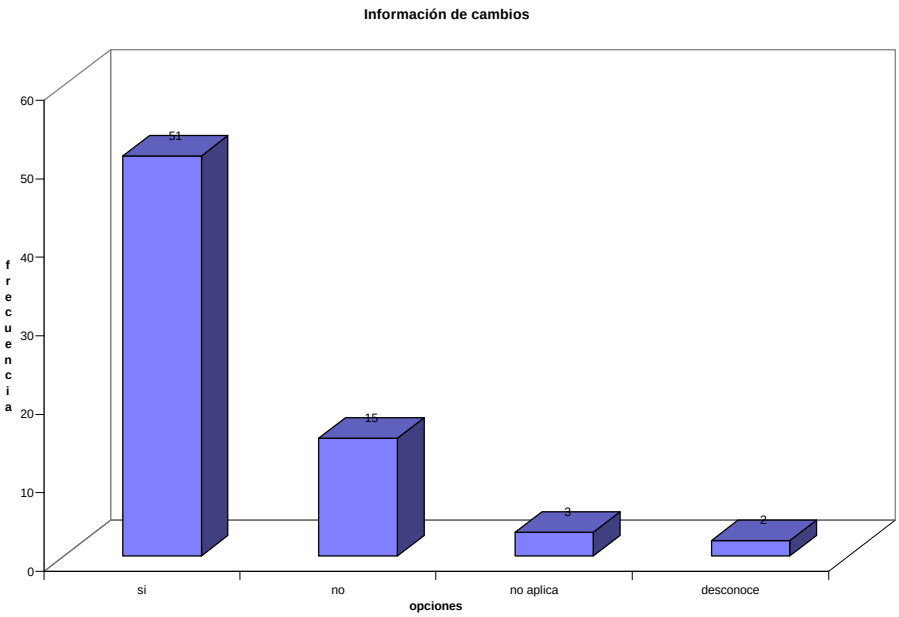
La gestión de calidad es más beneficiosa en la medida en que sea posible incorporar elementos de ella en etapas más tempranas del ciclo de desarrollo de software. Es claro que un error de desarrollo entre más cerca de su fuente de origen se detecte será fácil de corregir y por lo mismo el costo de aquella corrección será mucho menor. Ver tabla 11.

Tabla 11. Comunicación entre los miembros del proyecto

Opciones:	**Porcentaje (%)**
• **Sí**	70.83
• **No**	20.83
• **No aplica**	4.22
• **Desconoce**	2.81

Fuente: Elaboración propia, agosto 2001.

Gráfica 12. Comunicación entre los miembros del proyecto para realizar los cambios y ajustar los planes.



Fuente: Elaboración propia, agosto 2001.

Por tradición, se considera que la tarea de programación es una actividad individual y privada, de modo que muchos programadores tienen poco contacto social y prefieren trabajar en forma aislada[13]. El incremento del tamaño del producto de software, disminuye la productividad al aumento de la complejidad de las interacciones entre los diversos componentes del sistema, y a causa del incremento de comunicación necesario entre programadores, gerentes y clientes.

El desarrollo de software es demasiado complejo si no se emplea un control de calidad adecuado. Debe haber un tiempo prudente para investigar la mejora de los procesos de software. Al medir la notificación de los cambios en los proyectos sólo el 70.83% comunica a los demás miembros que el proyecto va a ser modificado sobre todo en las áreas de soporte técnico, administración de bases de datos y de redes, y el 20.83% confirman la falta de comunicación entre los miembros. Ver gráfica 12.

4.3.7.Control de proyectos

La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar, es decir, el proceso se descompone en un conjunto relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para cada tarea.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas. La estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables(humanas, técnicas, de entorno, políticas, etc.) que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo. Ver tabla 12.

Tabla 12. Factores que compara para obtener estimaciones

Opciones:	**Porcentaje (%)**
1. Tamaño	29.85
2. Costos	27.61
3. Duración	14.92
4. Recursos humanos	13.43
5. Tecnología	14.17

Fuente: Elaboración propia, agosto 2001.

Al analizar las actividades que componen este proceso desde la perspectiva de la calidad organizacional, para ello se comparan las estimaciones iniciales con los resultados en términos de: tamaño del proyecto aparece como el factor determinante con un 29.8%, seguido por el costo total del proyecto con un 27.6%, posteriormente a niveles similares quedan la duración cronológica con un 14.92% y los aspectos tecnológicos con un 14.17%, y finalmente la estimación de los recursos humanos con un 13.43%. Ver gráfica 13.

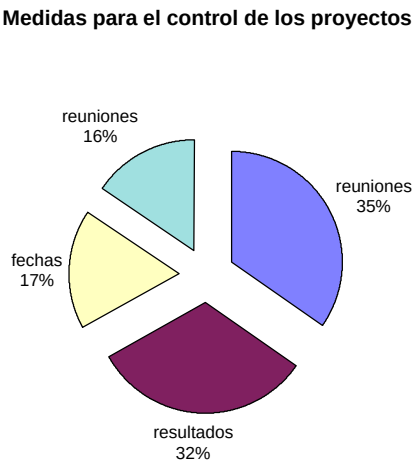
Al pensar en los medios de control de las etapas de desarrollo el 34.78% efectúa reuniones periódicas para conocer el estado del proyecto, situación que es muy aconsejable pues reduce significativamente la tasa de errores acumulados y con ello, los costos.

Gráfica 13 .Estimaciones para el control de proyectos

Fuente: Elaboración propia, agosto 2001

La evaluación final de los resultados de todas las revisiones realizadas solo se presentó en un 32.17%, en cambio el identificar si las tareas se han cumplido en las fechas programadas reflejó un 17.39. Las reuniones informales con los miembros para conocer sus valoraciones subjetivas obtuvo un 15.65%. Tanto los investigadores como el personal dedicado a las áreas de control y sistemas de tiempo real demostraron poner mayor énfasis en estos cuatro factores, aunado claro está a su nivel de experiencia en la coordinación de proyectos.Ver gráfica 14.

Gráfica 14. Actividades para mantener un control del proyecto



Fuente: Elaboración propia, agosto 2001.

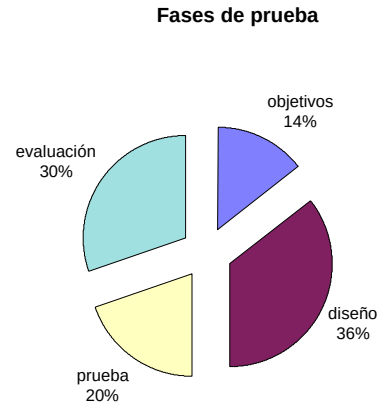
4.3.8.Pruebas

El plan de prueba describe la forma en la cual se mostrará a los clientes que el software funciona correctamente, explicando quién realizó las pruebas, las razones y cómo se condujeron las pruebas[14].

En cuanto a los planes de prueba solo el 15.27% realiza el plan completo de pruebas, en promedio 13.63% de los lideres de proyecto utilizan un plan de pruebas, los demás solo ejecutan pruebas y las evalúan. Ver gráfica 15.

Cuando los componentes poseen gran nivel de defectos en las pruebas, generalmente poseen grandes defectos en sus usos subsecuentes [15]. Las pruebas de los defectos son entonces un buen indicador de la calidad en el trabajo. Se decidió agrupar los tipos de prueba en sólo dos grupos: pruebas de caja blanca y negra.

Gráfica 15. Fases del plan de prueba aplicadas

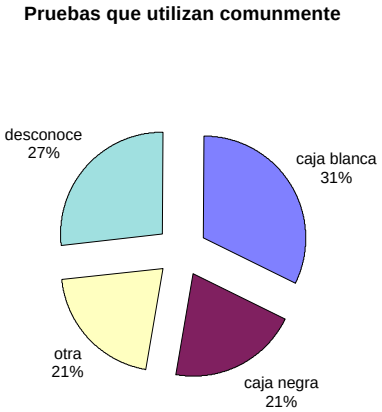


Fuente: Elaboración propia, agosto 2001

La prueba de la caja blanca permite obtener casos de prueba que garanticen que al menos una vez se ejerciten todos los caminos independientes de cada módulo, se ejercitan todas las decisiones lógicas en sus vertientes verdadera y falsa; se ejecutan todos los bucles en sus límites y con sus límites operacionales, ejerciten las estructuras internas de datos para asegurar su validez, el 31% aplica esta prueba. Ver gráfica 16.

La prueba de la caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Intenta encontrar errores en funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. Esta prueba se emplea en un 21%.

Gráfica 16. Tipos de pruebas empleadas comúnmente.



Fuente: Elaboración propia, agosto 2001

En total, sólo el 38.04% utiliza ambas. Tanto en el caso de los líderes de proyecto como en el de los consultores y programadores la mayoría reflejó aplicar las pruebas de la caja negra en mayor medida (8%) a diferencia de las de la caja blanca.

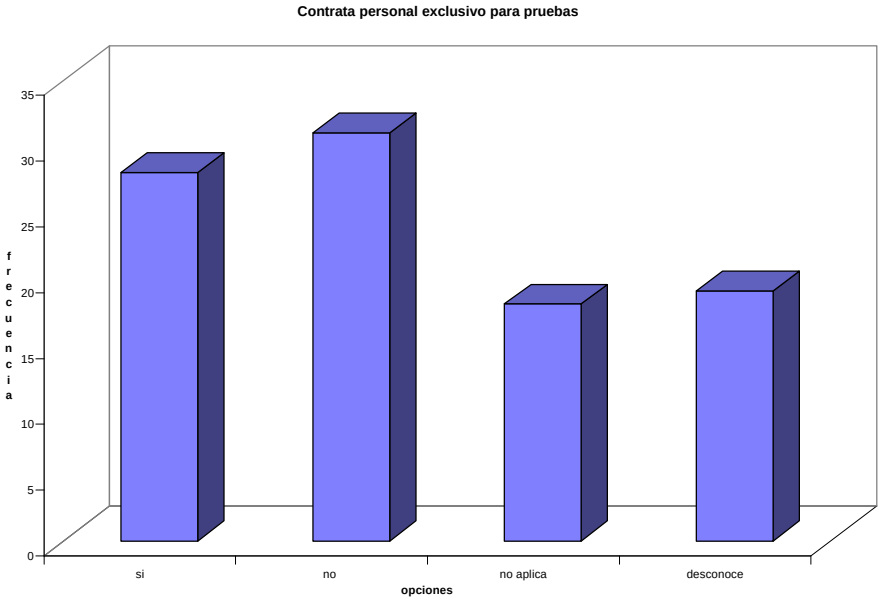
El 21% utilizan otro tipo de pruebas, entre las que se ubicaron las siguientes herramientas de automatización de pruebas:

1. Verificadores de prueba. Miden el alcance interno en relación a la estructura de control e informa del valor de cobertura al experto de garantía de calidad.
2. Generadores de archivos de prueba donde los procesadores generan y rellenan con valores predeterminados, archivos de entrada típicos para posteriores pruebas de programas.
3. Simuladores de entorno donde modelizan el entorno externo del software de tiempo real para simular dinámicamente las condiciones reales de operación.
4. Pruebas de resistencia donde se enfrentan los programas a situaciones anormales.

Si la prueba se lleva a cabo con éxito mostrará hasta que punto las funciones del software se adecuan a las especificaciones y se cubren los requisitos de rendimiento.

El personal de pruebas, evalua los documentos y registros usados en la elaboración del sistema, así como todas las salidas y reportes, la descripción de las actividades de flujo de la información y de procedimientos, los archivos almacenados, su uso y su relación con otros archivos y sistemas, su frecuencia de acceso, su conservación, su seguridad y control, la documentación propuesta, las entradas y salidas del sistema y los documentos fuentes a utilizar. Ver gráfica 17

Gráfica 17. Contratación de personal exclusivo para realizar pruebas.



Fuente: Elaboración propia, agosto 2001

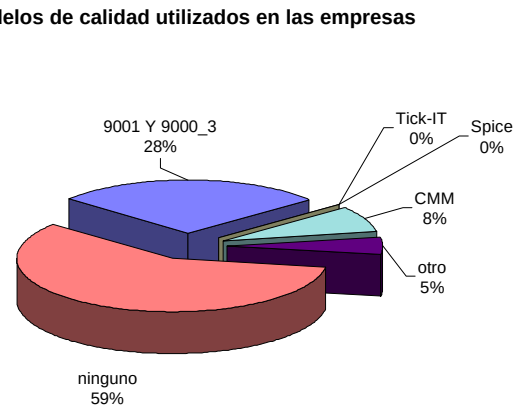
El 6.94% de los encuestados respondió que contrata personal externo exclusivamente para realizar las pruebas.

4.3.9.Modelos de calidad

Uno de los campos en los que más se ha trabajado es en la utilización de modelos de evaluación de calidad de software que tratan de aportar un medio para definir y descomponer el concepto de calidad de software en características más sencillas de evaluar y medir.

Entre los modelos de aseguramiento y gestión de la calidad del software se propusieron cuatro: ISO 9001 e ISO 9000-3, ISO SPICE, ISO Tick-IT, Capability Maturity Model (CMM), contemplando la opción de que los usuarios pudiesen utilizar algún otro modelo, o en su defecto que no aplicasen ninguno. Ver gráfica 18.

Gráfica 18. Aplicación de los modelos más comunes de aseguramiento de la calidad de software



Fuente:Elaboración propia, agosto 2001.

El 28% utilizan el ISO 9001 e ISO 9000-3, el 8% utilizan el CMM, el 5.33% utilizan otros modelos, modelos propios que ellos van ajustando para satisfacer las necesidades de los clientes, el 58.67% contestó que aun no aplican ningún modelo.

4.3.10.Calidad de software

Debido a que son muchos los factores para medir la calidad, se optó por obtener información acerca de tres temas específicos: los factores de calidad de un producto, los problemas a los que se enfrentan al elaborar un software y los factores que inciden en la productividad del software. Nuevamente se delimitaron los rangos de respuestas desde los niveles *bajo, medio, alto y muy alto* . Si se encuentra a niveles alto para un determinado proyecto, la productividad del desarrollo de software será significativamente más alta que si el mismo factor estuviera por debajo de la media (desfavorable).

Los factores de calidad que describen a un software según McCall [16] centrándose en características operativas, su capacidad de soportar cambios y su adaptabilidad a nuevos entornos. El Estándar ISO/IEC 9126[17] proporciona definiciones de características y subcaracterísticas que sugieren algunos aspectos a analizar para buscar objetivos de calidad para un producto de software. El enfoque general debe estudiar el riesgo relacionado con cada uno de éstos aspectos, y en caso de presentarse un elevado nivel de riesgo se debe especificar un conjunto de objetivos de calidad para el producto de manera que

pueda limitarse el riesgo. A continuación mencionaremos los requerimientos para las características de Calidad.

Funcionalidad . "Conjunto de atributos que mantienen la existencia de un conjunto de funciones así como sus propiedades especificas. Las funciones son aquellas que satisfacen un estado o las necesidades implícitas."

Confiabilidad . "Conjunto de atributos que respaldan la capacidad del software para mantener su desempeño bajo ciertas condiciones en un periodo del tiempo".

Eficiencia. "Conjunto de atributos que respaldan la relación entre el nivel de ejecución del software y el monto de los recursos utilizados bajo las condiciones establecidas."

La eficiencia del software está influenciada por su implementación. La arquitectura de software, la estructura de los algoritmos y los lenguajes de programación tienen un gran impacto. Es recomendable que la fase de diseño de la arquitectura del software se tome en cuenta, revisándola y justificándola para alcanzar la eficiencia de los objetivos.

Usabilidad. "Conjunto de atributos que respaldan el esfuerzo requerido para su uso, así como en el aseguramiento individual a través de un estado o un conjunto implicado de un conjunto de usuarios".

Mantenimiento. "Conjunto de atributos que respaldan el esfuerzo requerido para especificar las modificaciones". El mantenimiento es esencialmente importante cuando el objetivo es adquirir el producto completo incluyendo la trazabilidad de los procesos de desarrollo del software.

Portabilidad. "Un conjunto de atributos que respaldan la habilidad de software para ser transferido de un ambiente a otro".

Reusabilidad. “Conjunto de atributos que respaldan la posibilidad de que el producto de software se use en otro dominio”[18].

La eficiencia demostró ser el factor más importante con un 57.14% junto con la confiabilidad con un 54.93%, seguidas por la funcionalidad con un 50.7%, la mantenibilidad con un 47.89%, la usabilidad con un 42.85%, la portabilidad con un 39.44% y de menor importancia fue la reusabilidad con un 38.02%. Los primeros cuatro factores los ubicaron en niveles más altos, la usabilidad, la portabilidad y la reusabilidad se ubicaron en nivel medio. Podría inferirse que la productividad es significativamente alta.

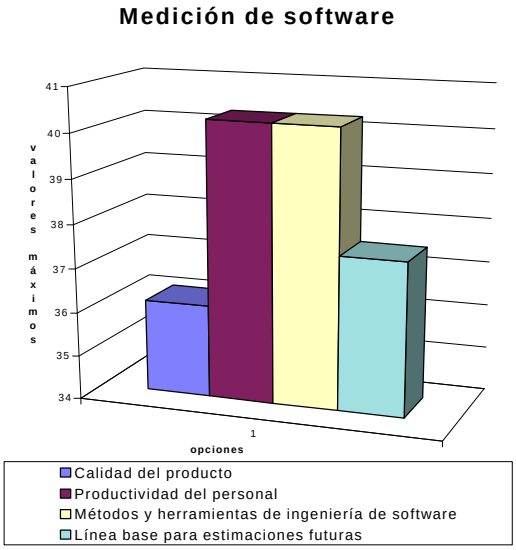
A pesar de contar con herramientas de cuarta generación y de técnicas de orientación a objetos, se observó que se sacrifica la reusabilidad de sus componentes, por ello esta característica fue la de menor importancia. Tracz observó que los programadores de software primero deben encontrar la utilidad de la reusabilidad[19]. La utilidad dependerá del marco de referencia de la aplicación realizada así como de sus características internas. Además, los líderes de proyecto no observan que los sistemas de software frecuentemente siguen patrones similares. Sería posible capturar sus patrones como elementos de software reusable que podrían ser aplicados a muchos desarrollos diferentes.

Garvin define la calidad percibida por el usuario como la combinación de los atributos del producto que proporcionan la mayor satisfacción a un usuario especificado[20]. Los productos pueden tener calidad en relación al propósito para el que fueron diseñadas, podría decirse que es auto-evidente.

4.3.11. Medición de software

La razón fundamental para medir el software y los procesos de software es para obtener datos que nos ayuden a manejar el control para la planificación, los costos y la calidad de los productos de software. Es importante el ser consistente en cuanto a las medidas tales como el tamaño, los defectos, los esfuerzos y los tiempos planeados.

Gráfica 19. Motivos por los cuales de debe medir el software



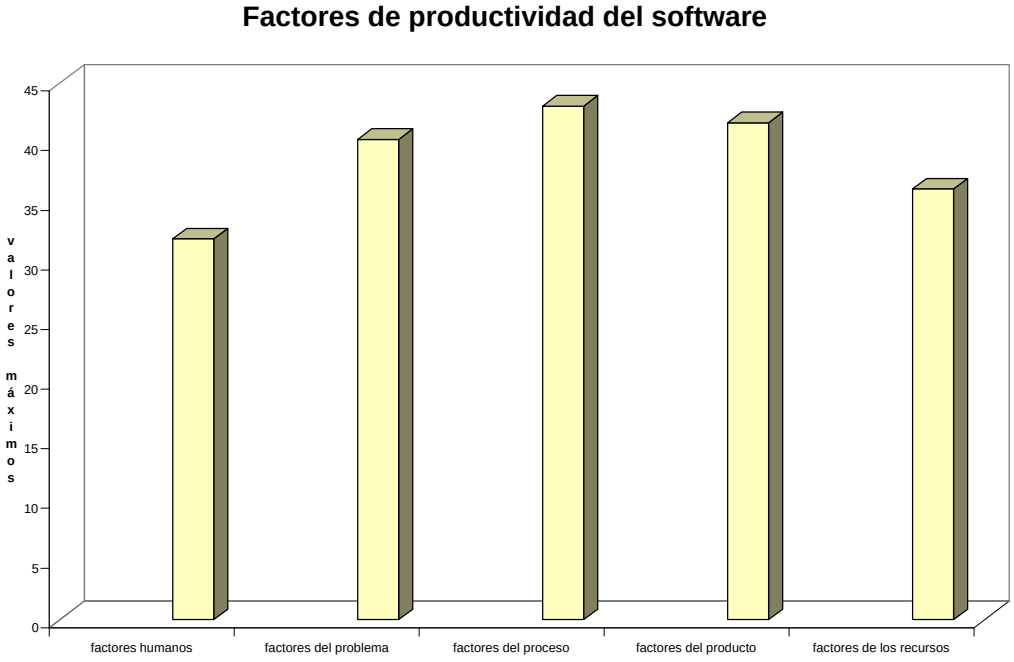
Fuente: Elaboración propia, agosto 2001

Los motivos que se señalaron como de mayor interés fueron: con un 68% el establecimiento de una línea base para estimaciones futuras, con un 59%, el uso de los nuevos métodos y herramientas de ingeniería de software para su posterior incorporación al proceso de producción, en tercer lugar con un 54.5% el indicar el grado de calidad de los productos. Ahora bien, debido a que la perspectiva de este estudio se enfoco a personal del área de sistemas, el factor de medición de productividad no tuvo la importancia esperada a diferencia de lo que hubiera sucedido si la encuesta hubiese sido contestada por personal del área de recursos humanos. Se reconoce como un problema importante la insuficiencia de estímulos. Ver gráfica 19.

4.3.12. Factores de productividad

Debido a las características propias del análisis y la programación, es muy frecuente que la implantación de los sistemas se retrase y se llega a suceder que una persona lleve trabajando varios años dentro de un sistema o bien que se presenten irregularidades en las que los programadores se ponen a realizar actividades ajenas a la dirección de informática. Para la encuesta se solicitó su calificación sobre algunos de los problemas considerados como principales para la productividad del software. Ver gráfica 20.

Gráfica 20.Factores que inciden en la productividad



Fuente: Elaboración propia, agosto 2001

Dentro de esta pregunta influyó los años de experiencia en el área de software. En la introducción de las estrategias, los involucrados en el área de sistemas poseen dificultades para adoptar nuevos métodos. Pues han aprendido a desarrollar software durante su educación formal siguiendo sus prácticas con pocos ajustes y refinamientos. El factor de los recursos utilizados refiriéndose a las herramientas CASE y a los recursos de hardware y software mostró el mayor interés por parte de los encuestados con un 59%. Para los investigadores, encargados del área y gerentes con más de 10 años de experiencia, el factor humano con 54.5% fue el determinante en la productividad del software argumentando que si no se logra transmitir a los demás miembros del equipo las especificaciones del sistema aunado a una motivación continua, los miembros del proyecto no realizarán su trabajo con mayor interés. Los factores del problema tales como la complejidad del proyecto y los cambios en las restricciones o requisitos de diseño solo mostraron un 45% de importancia media alta, siendo que reflejaban los intereses de las personas de sistemas multimedia y web. Sin embargo, tanto los factores del proceso y del producto demostraron tener una importancia media de un 40.5%.

5.Conclusiones

La calidad del producto y de los procesos van de la mano. Cuando la calidad no se incluye en un sistema, el proceso de desarrollo se encarga solo de encontrar y componer los defectos. Como resultado, los proyectos se enfocan a reparar los defectos conforme a su grado de importancia para los usuarios. Las personas que no tiene una cultura de calidad, causan retrasos en las entregas de los proyectos, porque si los incluyen, el costo del proyecto se elevará, seguramente estos individuos salen más baratos, pero los errores cometidos, lo son mucho más.

Para establecer un ambiente de medición del software, la organización del software debe definir una colección de procesos de datos y registro de medios. Los problemas de software se reportan típicamente como los vehículos usados para coleccionar datos acerca de los problemas y defectos. Los datos son ensamblados como parte de un problema de análisis y procesos de corrección son precisamente los datos que caracterizan o proporcionan valores a los atributos de los problemas y los defectos a medir. Sin embargo, estas facilidades del proceso en los aspectos de colección de datos de problemas de software y medición de defectos, la variedad de las actividades encontradas y reportes de problemas relatados hace difícil para comunicar claramente cuando se define o se especifica los problemas y medición de defectos.

En realidad, no deben buscarse super-programadores brillantes o genios para ser parte de un equipo, tan solo, aquellos dispuestos a aprender y a compartir su conocimiento. Por ejemplo, si el programador ha practicado o esta dispuesto a programar en pareja, eso dice mucho de su perfil y de su capacidad de aprendizaje.

Al construir un producto de baja calidad y es puesto a prueba, éste revela el costo sobreestimado y que además los procesos no pudieron completarse en el plan estimado. El trabajo de software puede ser estimado, planeado y controlado para ajustarse a los planes. El producir software libre de defectos no es un trabajo sencillo, aunque no exista evidencia alguna que no pueda realizarse.

En el caso de los modelos de gestión y aseguramiento de la calidad, aun es poca la difusión de los mismos en la comunidad dedicada a esta área, y se necesita no sólo conocer las perspectivas internacionales de la calidad, sino también informar de las consecuencias que podría ocasionar al implementar alguno de los modelos.

Las organizaciones que desarrollan software en México son en su mayoría pequeños y microempresarios que no se interesan por utilizar métodos de control de sus proyectos, su perspectiva radica en cumplir sólo con los requerimientos en las fechas estipuladas, sin ofrecer a los clientes servicios de valor agregado. Los líderes de proyecto deberían probar controles efectivos suficientemente flexibles para ajustarse a los cambios o tomar ventajas de nuevas oportunidades. La necesidad de crear e implementar un proceso continuo para la administración efectiva incluye a todos los miembros (proveedores, clientes, desarrolladores). Balanceando los costos, programas y las consideraciones técnicas, pensando en futuro identificando incertidumbres, anticipando las entradas potenciales, administrando los recursos de los proyectos y actividades. Por lo mismo es importante centrarse en la calidad del producto como un medio para obtener mejores productos y a partir de ahí, influir en la calidad del proceso.

Como se ha podido observar, el adoptar una certificación bajo alguno de los modelos de calidad, nos da una oportunidad de permanecer en el mercado, pero desde una perspectiva de un líder de proyecto, que desconoce las metodologías de aseguramiento y gestión de la calidad, un modelo de calidad no permite la adopción de un método ágil de desarrollo, debido a que se tarda demasiado tiempo en establecer políticas y en reorganizar procesos, y con ello, los métodos tradicionales encarecen evidentemente los productos de software.

Referencias:

[1] Jimeno Solis, Marco A. Comportamiento del PIB en México. http://www.inegi.gob.mx

[2] Pressman, Roger. Ingeniería de software. Edit. McGraw-Hill, 3ª edición, Madrid España.1996.

[3] De Marco, Tom. Why does software cost so much?. Dorset house, New York, NY, USA, 1995.

[4] ANUIES. Anuario estadístico de la ANUIES 1998. http://www.anuies.mx

[5] Baker, F.T. "Chief Programmer Team Managment of Production Programming", IBM systems Journal, vol. 11 no. 1 1972, pp.56-73.

[6] Paulk, Mark C. y otros, "Capability Maturity Model for Software, versión 1.1", Software Engineering Institute, CMU/SEI-93-TR-24, Febrero de 1993.

[7]SECOFI http://www.siem.org.mx

[8] Yourdon, E. Techniques of Program structure and design; Prentice Hall, Englewood Cliffs, NJ, USA, 1975.

[9]Hadden, Rita. How scalable are CMM key practices?, Crosstalk: the journal of Defense Software Engineering, Vol II, No. 4, april 1998.

[10] Johnson, Donna L., and Broadman, Judith G. Applying the CMM to small organizations and small projects. Proceedings of the 1998 software engineering process group conference, Chicago IL, USA, 9-12, March 1998.

[11]Balzer, R. And Goodman, N. Principles of good software specification, Proc. On specifications of Reliable Software, IEEE, 1979,

[12]Guzmán Arenas, Adolfo. Realidades y perspectivas de la computación en México. Julio 1999 ISBN 970-18-3332-5. Serie Verde, No. 22.

[13] Cougar, D., and R. Zawacki "What motivates DP Professionals?" Datamation, September 1978.

[14]Shari Lawrence, Pfleeger. Software engineering theory and practice. Edit. Prencite Hall, 2nd edition.

[15] Weinberg, Gerald M. Quality software managment vol. 1, systems thinking. Edit. McGrawHill, México, 1991.

[16] McCall J A, Richards PK y Walters GF; Factors in software quality, Vols I,II,III; US Rome Air Development Center Reports NTIS AD/A-049 014, 015, 055, 1977.

[17] ISO, Software product evaluation. Quality characteristics and guidelines for their use, ISO, 1991.

[18] Santa Olalla Salgado, Rene, Santa Olalla Salgado, Javier & Gutiérrez Tornés, Agustín F. Monografía de la reusabilidad del software. Reporte técnico. Serie verde No. 43. Noviembre 2000, México.

[19]Tracz, Will. Software Reuse Maxims, ACM SIGSOFT Software Engineering Notes, Vol. 13, No.4, October 1988.

[20] Bevan, Nigel & Azuma, Motoei. Quality in use: Incorporating human factors into software engineering life cycle, Uk-Japan.

Anexo A1

Estadísticas del SIEM
Empresas, Padrón, Estadísticas por CMAP de Servicios.

951004
Servicios de análisis de sistemas y procesamiento informático 224

Estadísticas del INEGI. Aspectos informáticos
Unidades económicas por sector según clase de actividad (datos referentes a 1993).

Fuente: Censos económicos 1994. INEGI.
*principio de confidencialidad estadística.
Servicios: Comprende las clases censales 831113:Servicios de alquiler de equipo electrónico para el procesamiento informático y 951004. Servicios de análisis de sistemas y procesamiento informático.

Estado	Servicios	
	Clase 831113	Clase 951004
Distrito Federal	14	236
Nacional	71	636

Anexo A2

El presente cuestionario es de carácter académico y anónimo, tiene por objetivo recabar información sobre los factores que inciden en la producción de software, por lo cual se agradece su participación y se le informa que los datos que usted proporcione serán de carácter estrictamente confidencial.

Sección I. Antecedentes
Instrucción: Marque con una X la opción seleccionada.

1.¿Cuál de las siguientes opciones describe su posición actual (puede marcar varias)
( )Líder de proyecto o de equipo
( )Consultor
( )Programador/analista
( )Admón redes
( )Soporte técnico
( )Otro _____

2.¿En cuál de las siguientes actividades se desempeña (puede marcar varias)?
( ) Requerimientos de software
( ) Diseño de software
( ) Código y pruebas de unidades
( ) Pruebas e integración
( ) Aseguramiento de calidad de software
( ) Administración de la configuración
( ) Mejoras del proceso de software
( ) otro_____

3.¿Cuál es el giro de su empresa?
( ) Sistemas multimedia y web.
( ) Sistemas de información (sistemas de gestión, SMBD, SMBDOO, etc.).
( ) Sistemas distribuidos.
( ) Sistemas de tiempo real.
( ) Sistemas de ingeniería(CAD/CAM) y científicos.
( ) Sistemas de control (Sistemas de adquisición de datos).
( ) Sistemas de Inteligencia Artificial(sistemas expertos, etc.).
( ) Sistemas de geoprocesamiento
( ) Asesoría y consultoría.
( ) Capacitación.
( ) Docencia e investigación.
( ) otro_____

4.Experiencia laboral:
¿Cuánto tiempo lleva en la presente empresa? ____ años
¿Cuánto tiempo tiene en el área de software? ____ años

5.¿De cuántas personas se compone el área de desarrollo de software en su empresa? ( )
a)1-5 b)6-10 c)11-15 d)16-21 e)21 o más

6.-¿El tiempo promedio de duración de sus proyectos es de...? ( )
a)1 mes b)2-4 meses c)5-7 meses d)8-12 meses e)más de 12 meses

Sección 2 Gestión de proyectos de Software

Instrucción: Elija la opción de acuerdo a los criterios siguientes:
Si- La práctica está bien establecida y se ejecuta consistentemente como un procedimiento estándar de operación.
No- La práctica no está bien establecida o es realizada inconsistentemente.
No aplica – Posee el conocimiento requerido acerca del proyecto u organización pero usted siente que la pregunta no aplica para el proyecto.
Desconoce – Ud. posee incertidumbre de cómo responder a la pregunta.
Utilice los espacios para comentarios para cualquier aclaración acerca de sus respuestas a las preguntas.

1.¿Para elaborar un proyecto se sigue algún plan documentado que guíe el desarrollo?
( ) Si ( ) No ( ) No aplica ( ) Desconoce

2.¿Dentro de los proyectos, quienes son las personas encargadas de colaborar con el cliente para establecer el sistema de requerimientos?
( )líder de proyecto
( )líder comercial
( )líder de proyecto y comercial
( )otro_____

3.¿Al inicio del proyecto se revisan las nuevas tecnologías existentes en el mercado para determinar la más adecuada a las necesidades del cliente?
( ) Si ( ) No ( ) No aplica ( ) Desconoce
Comentarios:

4.¿En cuanto a la distribución de esfuerzos, que porcentaje le asigna a cada fase (el total debe ser 100%)?
( )Análisis y diseño
( )Codificación
( )Prueba y depuración

5.¿La capacitación proporcionada es la necesaria para desarrollar las bases y el conocimiento requerido para desempeñar sus actividades?
( ) Si ( ) No ( ) No aplica ( ) Desconoce

6.¿Mantiene informados a todos los miembros del equipo acerca de los cambios del proyecto(ej.: aquellos que realizaron el trabajo y aquellos que son los responsables del trabajo)?
( ) Si ( ) No ( ) No aplica ( ) Desconoce

7.¿Los resultados de los proyectos actuales se comparan con las estimaciones iniciales de los planes de software en términos de... ? (puede marcar varias)
( )Tamaño del proyecto
( )Costos
( )Duración cronológica del proyecto
( )Recursos humanos
( )Tecnología

8.¿Cuáles de las siguientes actividades realiza para mantener un control del proyecto? (puede marcar varias)
( )reuniones periódicas sobre el estado del proyecto
( )evaluar los resultados de todas las revisiones realizadas en todo el proceso
( )identificar si las tareas se han alcanzado en las fechas programadas
( )reuniones informales con los miembros para conocer sus valoraciones subjetivas

9.¿Para realizar las pruebas del sistema, cuáles de los siguientes pasos realiza?
( ) Establece objetivos de las pruebas
( ) Diseña los casos de prueba
( ) Prueba los casos
( ) Evalúa los resultados de las pruebas

10.¿Qué tipo de pruebas aplica a un proyecto?
( )pruebas que demuestren la función es completamente operativa
( )pruebas que aseguren que las operaciones internas se ajustan a las especificaciones
( ) otra _____

11.¿La empresa contrata personal exclusivo para las pruebas del sistema?
( ) Si ( ) No ( ) No aplica ( ) Desconoce

12.¿Cuál de los siguientes modelos de gestión y aseguramiento aplica?
( )ISO 9001 e ISO 9000-3
( )ISO SPICE
( )ISO Tick-IT
( )Capability Maturity Model (CMM)
( )otro _____
( )ninguno

Instrucción. En las siguientes tres cuestiones seleccione de la lista de valores anotando el número que crea conveniente para cada opción. **(1-Bajo 2- Medio 3-Alto, 4- Muy Alto)**

13.De acuerdo a su nivel de importancia los siguientes factores de calidad que describen a un software son:

( ) Confiabilidad (Que no falle, que no tenga errores)
( ) Eficiencia (El grado en que se aprovechan los recursos a disposición del producto de software)
( ) Funcionalidad (Se refiere al grado en que el software satisface los requerimientos, hace lo esperado)
( ) Mantenibilidad (¿Puedo actualizarlo?)
( ) Portabilidad (¿Podré usarlo en otro ambiente computacional?)
( ) Reusabilidad (¿Puede una parte o todo utilizarse en otro software?)
( ) Usabilidad (Se refiere a la facilidad de uso.)

14.De las siguientes razones, anote el número que a su criterio le corresponda a cada opción que señale los motivos por los cuales se debe medir el software.

( ) Indicar la calidad del producto
( ) Evaluar la productividad de la gente que desarrolla el producto
( ) Evaluar el uso de nuevos métodos y herramientas de ingeniería de software
( ) Establecer una línea base para futuras estimaciones

15.De acuerdo a su experiencia anote el número que corresponda a los factores que inciden en la productividad del software

( ) Factores humanos. La cantidad y la experiencia del personal de desarrollo.
( ) Factores del problema. La complejidad del problema a resolver y el número de cambios en las restricciones o los requisitos del diseño.
( ) Factores del proceso. Técnicas de análisis y diseño utilizadas.
( ) Factores del producto. Fiabilidad y rendimiento del sistema basado en computadora.
( ) Factores de los recursos. Disponibilidad de herramientas CASE, de recursos de hardware y de software.