



DIRECCIÓN DE POSGRADO E INVESTIGACIÓN

MAESTRÍA EN SISTEMAS COMPUTACIONALES

**MATERIA: CONTROLES EN EL DESARROLLO DE SISTEMAS
M EN C. MARIO FARÍAS- ELINOS**

TEMA:

Fábrica de Software

Alumnos:

PEDRO ALBERTO LEZAMA RAMOS
JUAN CARLOS AGUILAR FRANCO
ADRIAN GÓMEZ GALLARDO

AGOSTO DEL 2002

INDICE

CONTENIDO	PÁGINA
1. Antecedentes	4
2. Metodologías Propietarias	5
2.1. Metodologías Microsoft Solution Framework	
2.2. Fase 1 – Estrategia y alcance	
2.3. Fase 2 – Planificación y Prueba de Concepto	
2.4. Fase 3 – Estabilización	
2.5. Fase 4 – Despliegue	
3. Metodologías SunReady	8
3.1. Características Principales	
3.2. Fase 1- Diseño de la solución	
3.3. Análisis del Nivel de Servicios SunReady	
3.4. Servicio de Diseño de Ambientes de Producción SunReady	
3.5. Fase 2 – Planeación de la Implementación	
3.6. Fase 3 – Implementación del Prototipo	
3.7. Fase 4 – Implementación del ambiente de Producción	
3.8. Fase 5 – Colocación del ambiente en Producción	
3.9. Servicios de Soporte para el ambiente SunReady	
3.10. Beneficios de la Metodología Sun Ready	
4. CMM / Introducción	13
4.1. Niveles de Madurez y Áreas Clave de CMM	
4.2. Caracterización Nivel 2. Repetible	
4.3. Área claves de procesos para el Nivel 2. Repetible	
4.4. Administración de Requerimientos	
4.5. Planificación de proyectos de Software	
4.6. Supervisión y Seguimiento de Proyectos de Software	
4.7. Puntos de Función	
4.8. Componentes de la Evaluación	
4.9. Complejidad para Cada Función	
4.10. Cálculo de PFs	
4.11. Fórmula para medir un Proyecto de Mejoramiento	
4.12. Caso de Uso	
5. Aplicaciones con Microsoft	25
5.1. Beneficios	
5.2. Nace el Plan Fast-Track para el desarrollo del Software Mexicano	

6. Fábricas de Software en México	29
6.1. Sofftek	
6.2. TECHNOS Consultores	
6.3. Sistemas Verticales S.A. de C.V.	
6.4. Hildebrando	
7. Realidades y Perspectivas de la Computación en México / Adolfo Guzmán Arenas	35
7.1. Es atrevido tratar de Medir “Procesos de Fabricación” en Software	
7.2. La ventaja de Tener al Fabricante de Software a la vuelta d la esquina	
7.3. Los equipos, dispositivos, periféricos su Fabricación y Venta	
7.4. Software a la Medida	
7.5. Que ocurre en México	
8. Programa para Desarrollar la Industria del Software México	42
9. Impulsan Estados Desarrollo del Software	44
10. Meta: Acelerar la Industria del Software	47
11. Fábricas de Software Publicaciones	50
11.1. Información del X Symposium	
12. Conclusiones	53

1.- ANTECEDENTES.

A fines de los años 60's e inicios de los 70's, surge en la industria del software el concepto de fábrica de software. Este nace como una respuesta a la necesidad de aliviar la incertidumbre que se tenía en el desarrollo de proyectos de software en aspectos como:

- Confiabilidad de los productos.
- Mantener en presupuesto y calendario los proyectos de desarrollo de software.
- Falta de una definición y seguimiento adecuado a los procesos de producción, así como un medio efectivo de medir su desempeño y la productividad de las personas que lo ejecutan.
- Falta de estandarización en los métodos y herramientas empleados en los procesos. Esto provocaba reinventar el "hilo negro" cada vez, además del nulo reuso de los productos que los mismos procesos de producción generan.
- Falta de herramientas para hacer rastreables los productos (requerimientos, especificaciones de productos, etc.) que generan los procesos.

El fundamento para establecer fábricas de software, fue basado en tratar de obtener los beneficios que las líneas de producción industrial produjeron en la calidad de los productos así como la productividad lograda. Dichas líneas de producción se basan en tener definido un proceso y el flujo que sigue cada operación del proceso, además de proveer las herramientas necesarias para llevar a cabo la acción deseada. Entre más segmentado es el proceso, mas simple se vuelve cada operación logrando con esto una división del trabajo orientada a la especialización. Esto logra elevar la productividad, sin embargo, trae situaciones que en un momento determinado hay que afrontar como el contar con personas que tienen un dominio restringido de acción.

Vista de esta forma, la cadena de producción es un mecanismo que permite a un trabajador estar habilitado para tomar una acción, altamente eficaz y eficiente, en un momento adecuado del proceso.

A esto se le llama "situación de resolución". En este sentido, aprovechando las tecnologías de información es posible crear cadenas virtuales de producción en donde el proceso y su flujo estén claramente definidos de tal forma que generen situaciones de resolución para cada trabajador. (Para mayor detalle consulte Soluciones Avanzadas, Febrero1996: "La Cadena Virtual: Hacia la fundamentación de la Reingeniería", Bracho, Díaz.)

Fue entre otras por estas causas, que varias empresas en el mundo principalmente Norteamericanas y Japonesas, invirtieron recursos en la definición y puesta en marcha de sus fábricas de software.

Entre las empresas Norteamericanas pioneras en este ámbito, se encuentra System Development Corporation (SDC), que en los años 70's concibió un modelo de fábrica de software para tratar de resolver los problemas antes mencionados. Su modelo giró alrededor de dos partes: El control que

incluía las áreas de control de proyectos y aseguramiento de calidad y el área de implementación que incluía las áreas de diseño, construcción y pruebas de sistema, ambos coordinados a través de procedimientos y estándares.

Este modelo retroalimentándose a través de diversos proyectos, desembocó en lo que sería el manual de desarrollo de software SDC. El resultado que trajo el esfuerzo de contar con dicho manual, fue que los proyectos comenzaron a converger en productos con menos defectos y en el tiempo y presupuesto establecidos.

El modelo de SDC, rápidamente influenció a empresas del Japón que comenzaron a definir su propio modelo de fábrica de software. Una de las fabricas pioneras es Hitachi que es la más grande y antigua, se estima que en 1991 contaba con 7,000 empleados trabajando en el desarrollo de diversos tipos de aplicaciones para instituciones financieras, de seguros, control de inventarios, recursos humanos, etc. Durante los años 70's Hitachi trabajó arduamente en la definición y medición de los procesos de desarrollo de software, incorporando elementos de medición y control de calidad. Sólo que no todo funcionó a la primera, entre los múltiples problemas que tuvieron que resolver fue la incapacidad de incorporar conceptos de fabrica de software como la estandarización del ciclo de desarrollo del proyecto y el reuso de componentes. Un problema adicional fue que trataron de estandarizar un sólo proceso de desarrollo para cualquier tipo de aplicaciones. Esto provocó proyectos fuera de fecha y presupuesto establecido.

Fue a finales de los años 70's que lograron organizar su fábrica de software alrededor de un manual que contenía enfoques de ingeniería y fábrica. En el, se incorporaron diversas técnicas del diseño y codificación estructurados así como tiempos estándares para cada actividad, inspecciones de productos y análisis de defectos del proceso entre otros elementos. Fue después de este esfuerzo que decidieron invertir en el desarrollo de sus propias herramientas de software para soportar sus funciones. A partir de este momento la fábrica logró una mejora impresionante en su desempeño, pasan de atrasos en la entrega de proyectos al departamento de aseguramiento de calidad de 72% a 12%.

Otra fábrica de software pionera en el Japón es Toshiba. Durante los años 70's, se enfocó en la definición de su modelo de fábrica de software alrededor de cuatro puntos: Estandarizar los procesos de desarrollo para reducir variaciones entre proyectos; Reuso exhaustivo de diseños y programas para construir nuevos sistemas a fin de reducir el trabajo redundante y maximizar la productividad; Introducir el uso de herramientas estándar a fin de elevar los niveles de desempeño de las personas; Proveer entrenamiento extensivo en las personas.

Para poder realizar lo anterior, Toshiba se centró fuertemente en el reuso de componentes como una medida estratégica para incrementar la productividad. Desarrolló diversas herramientas estandarizadas para soportar el diseño, identificación y control de componentes reusables, generadores de código, pruebas del software y control de proyectos. Estas herramientas permitieron elevar la productividad de una manera impresionante, pasaron de 1,400 líneas de código fuente ensambladas en 1976 a mas de 3,000 en 1986. Para Toshiba, gran parte de su éxito se debe al reuso de componentes.

Estas experiencias han llevado a varias definiciones o acepciones del término "fabrica de software" y a enfatizar alguno de los elementos que aparecen o explican parte de los éxitos referidos. Tal es el caso de una definición que declara como elemento esencial de una fábrica a los componentes de reuso.

Hace falta una definición, que al aplicarla, lleguemos todos a la misma conclusión y que incluya los elementos que han sido reconocidos y aceptados por la industria como esenciales. Lo ideal sería llegar a algo así de claro como... $\text{Activo} = \text{Pasivo} + \text{Capital}$. Más aún, tal como los estados financieros, debería poder ser dictaminable o certificable por un tercero.

<http://www.certum.com/Publicaciones/FabSoft.pdf>

2.- METODOLOGÍAS PROPIETARIAS

Metodología Microsoft Solution Framework

Fase 1 - Estrategia y alcance

En esta fase deberían tener lugar los siguientes trabajos:

- Elaboración y aprobación del Documento de Alcance y Estrategia definitivo: debe ser un documento de consenso con la participación del mayor número de agentes implicados en el proyecto. En este documento quedarán definitivamente reflejadas las funcionalidades y servicios que, ineludiblemente, debe ofrecer la solución a implantar.
- Formación del Equipo de Trabajo y distribución de competencias y responsabilidades: generalmente se definen como áreas principales la de Diseño de Arquitectura, Pruebas de Laboratorio, Documentación, Logística y Coordinación.
- Elaboración del Plan de Trabajo: deben marcarse fechas y contenidos para esta fase y las siguientes. Los mecanismos y protocolos de intercambio de información y coordinación deben quedar suficientemente bien establecidos y consensuados.
- Elaboración de la matriz de Riesgos y Plan de Contingencia: los principales riesgos detectados deben tener un plan de mitigación y actuación y revisarse con periodicidad.

Fase 2 - Planificación y Prueba de Concepto

Deben alcanzarse los siguientes objetivos e hitos:

- Documento de Planificación y Diseño de Arquitectura: es el documento principal, donde se describen en detalle los aspectos funcionales y operativos de la nueva plataforma. La aprobación de este documento es el hito principal de esta fase, y supone la directriz última de todos los trabajos técnicos, que, a partir de ese momento, deben ser consistentes con esta Guía. Si en el curso de las fases sucesivas fuera necesario revisar estos contenidos, se deberá hacer por acuerdo y conocimiento de todo el equipo de trabajo y se llevará un registro de versiones que permita hacer un seguimiento adecuado de estas revisiones.
- Documento de Plan de Laboratorio - Prueba de Concepto: la descripción del contenido del laboratorio de prueba de concepto, los diversos escenarios a simular, los criterios de validez, el control de incidencias y las métricas de calidad son objetivos a cubrir en este documento. Es un documento dinámico, en el que se recoge la idea y la experiencia práctica al llevarla a cabo en ambiente controlado y aislado. La etapa de prueba de laboratorio concluye cuando la maqueta ofrece todos los servicios y funciones descritos en el Documento de Alcance y Estrategia, y su grado de estabilidad y rendimiento es considerado como "suficiente".

Fase 3 - Estabilización

La solución implantada en la maqueta se pasa a un entorno real de explotación, restringido en número de usuarios y en condiciones tales que se pueda llevar un control efectivo de la situación. Los hitos y objetivos fundamentales de esta fase son:

- **Selección del entorno de prueba piloto:** se acordará la composición y ubicación del conjunto de máquinas y usuarios que entrarán en la prueba. Esta selección se recomienda que se haga atendiendo a la mayor variedad posible de casos, de manera que puedan aflorar el máximo de incidentes potenciales en el menor tiempo posible. La dimensión de la muestra tiene también que calcularse, sin perder de vista que la prueba piloto no es el despliegue propiamente, sino una fase de observación en la que es absolutamente crítico establecer unos cauces efectivos de tratamiento de los errores.
- **Gestión de Incidencias:** aunque esta labor se habrá iniciado en la fase anterior, el éxito de la prueba piloto dependerá de que se forme un sistema de recogida de incidentes (help desk o similar), de atención al usuario (formación, consultas) y de resolución de problemas y documentación de los mismos (versión de la plataforma).
- **Revisión de la documentación final de Arquitectura:** el documento de Planificación y Diseño de Arquitectura se puede ver alterado parcialmente como resultado de esta fase. El documento final, aprobado por consenso, supone el principal documento del Proyecto y la culminación de los trabajos de diseño, al menos en sus líneas principales. Este documento se considerará definitivo cuando la solución puesta en marcha se muestre estable y el número de incidencias graves (de intervención o de resolución) sea nulo y la cantidad de las consideradas leves quede por debajo de un límite establecido en las Métricas de Calidad.
- **Elaboración de la documentación de Formación y Operaciones:** con vistas al soporte post proyecto y los programas de formación a usuarios y administradores, en esta fase deben elaborarse las Guías de Usuario, de Administración, las "paso-a-paso", y otros cuyos contenidos deben acordarse previamente.
- **Elaboración del Plan de Despliegue:** se debe consensuar la fecha de finalización de la fase Piloto, y las condiciones de calidad que debe cumplir la solución final para iniciar el despliegue. En el Plan deben identificarse las fases, estrategia de implantación, fechas, tareas a realizar, procedimientos de validación y método de control de incidencias.
- **Elaboración del Plan de Formación:** con anterioridad al despliegue definitivo, debe haberse aprobado el Plan de Formación orientado a usuarios finales y administradores, y debe hacerse compatible con los ritmos acordados en el Plan de Despliegue.

Fase 4 - Despliegue

Se llevarán a cabo en esta fase los planes diseñados en la anterior, principalmente el de despliegue y el de formación. Los principales trabajos e hitos a conseguir son, en este caso, además de los obvios (implantación de la plataforma, puesta en servicio de todas las funciones, formación a los usuarios y administradores), los siguientes:

- Continuación con las labores de recepción de incidencias, clasificación, tratamiento, resolución y distribución de fixes o intervención on-site.

- Registro de mejoras y sugerencias, funcionalidades no cubiertas y novedades a incorporar en sucesivas versiones de la plataforma, incluyendo mejoras aportadas por los fabricantes de software (nuevas versiones o Service Packs, por ejemplo)
- Revisión de las Guías y manuales de usuario, rectificación de errores y obtención de los documentos de formación definitivos.
- Entrega de los documentos definitivos acordados como "deliverables" en la fase de Visión Scope.
- Revisión (si procede) de la matriz de riesgos, las métricas de calidad y establecimiento de los estándares de calidad y SLA definitivos.
- Finalmente, entrega del Proyecto y cierre del mismo, con o sin apertura de nuevo proyecto en base a la información y experiencia obtenidos.

La duración fase de despliegue, puesto que debe planificarse, no puede establecerse a priori. Depende de numerosos factores externos al propio proyecto (incluyendo factores de oportunidad política o de negocio) que pueden retardar o acelerar la conclusión.

<http://www.microsoft.com/msf>

<http://www.microsoft.com/Spain/Technet/fases/default.asp#1>

<http://www.microsoft.com/business/services/mcsmsf.asp>

3.- Metodología SunReady

3.1. CARACTERÍSTICAS PRINCIPALES

- Metodología bien definida de cinco fases, la cual combina la experiencia de Sun en las áreas de consultoría, servicios educacionales y soporte de sistemas.
- Solución integral para el diseño, especificación, implementación y producción de sistemas de misión crítica
- Menor riesgo en las implementaciones complejas
- Mayor satisfacción de los usuarios, gracias a un nivel más alto de disponibilidad del sistema

3.2. Fase 1: Diseño de la solución

El objetivo de la primera fase es proporcionarle información acerca de sus compromisos actuales de servicio y los pasos que debe tomar para desarrollar un ambiente de producción que pueda lograr los niveles de servicio esperados. Durante esta fase, el equipo de SunReady recopila y analiza información para ofrecer recomendaciones específicas sobre el personal, procesos y productos requeridos. Luego, trabaja estrechamente con su personal para formular el diseño de toda su solución. Esta fase se ejecuta a través de dos servicios exclusivos: el Análisis del Nivel de Servicios SunReady y el Servicio de Diseño de Ambientes de Producción SunReady.

3.3. Análisis del Nivel de Servicios SunReady

Este servicio de corta duración le permite comprender a un alto nivel los compromisos de nivel de servicio con sus usuarios. El análisis examina los niveles de servicio diarios que los usuarios necesitan

para llevar a cabo sus tareas cotidianas. Esta información sirve como base para comprender el ambiente de producción requerido para cumplir estos compromisos.

Análisis del Nivel de Servicios SunReady

- Entrevistas para identificar los requerimientos de los usuarios con respecto a los servicios de la empresa
- Informe del Análisis del Nivel de Servicios SunReady
- Definir el alcance del Servicio de Diseño de Ambientes de Producción SunReady

3.4. Servicio de Diseño de Ambientes de Producción SunReady

Durante este segundo servicio, un equipo de Sun lleva a cabo una evaluación detallada y rigurosa del ambiente de producción de su centro de cómputo. Las entrevistas con el personal del centro de cómputo se enfocan en los aspectos de administración de servicios, administración de cuentas, administración de problemas, administración de implementaciones y programas, control de cambios, administración de recursos, administración del rendimiento, administración de optimizaciones y administración de personal. Se efectúa una comparación entre los requerimientos de su ambiente de producción (personal, procesos y productos) y las capacidades de su centro de cómputo actual.

Servicio de Diseño de Ambientes de Producción SunReady

- Entrevistas con el personal del centro de cómputo para identificar las prácticas que se llevan a cabo en el centro de cómputo
- Efectuar un análisis de las brechas entre las capacidades actuales de su centro de cómputo y los requerimientos del ambiente de producción
- Proponer los cambios necesarios para satisfacer los requerimientos del ambiente de producción
- Crear un plan estratégico para el ambiente de producción
- Recomendar las herramientas del centro de cómputo necesarias para lograr el ambiente de producción deseado
- Describir los objetivos de los niveles de servicio que Sun proporcionará
- Recomendar los servicios para proveer soporte al ambiente de producción

3.5. Fase 2: Planeación de la implementación

Durante la segunda fase, el equipo de SunReady define las especificaciones y planes detallados, basándose en los elementos de diseño de la solución identificados en la primera fase.

- Personal. La segunda fase incluye un servicio de evaluación detallado de las habilidades del personal del cliente, así como la iniciación de programas de entrenamiento para el personal del ambiente de producción.
- Procesos. Su equipo SunReady comienza a captar y especificar los procesos claves que necesitarán implementarse en el ambiente de producción final.
- Productos. La metodología SunReady utiliza las mejores prácticas asociadas a la implementación y prueba de las herramientas de producción y del sistema, concentrándose en los requerimientos de sus otros equipos de producción (tales como PCs/estaciones de trabajo,

red, seguridad y centro de soporte técnico). Se definen y redactan las pruebas de aceptación de acuerdo a los compromisos de los niveles de servicio identificados en la primera fase.

3.5. Planeación de la implementación

- Crear un conjunto de reglas detalladas para la implementación de la arquitectura técnica
- Desarrollar procedimientos de pruebas de aceptación basados en los objetivos de los niveles de servicio y disponibilidad
- Determinar los requerimientos de la interfaz del ambiente de producción
- Realizar una evaluación de las habilidades del personal
- Proporcionar un informe de evaluación de habilidades y documentos con el plan de entrenamiento
- Iniciar el entrenamiento del personal del centro de cómputo
- Formular un plan de soporte para el ambiente de producción
- Especificar los procesos y herramientas para la implementación

3.6. Fase 3: Implementación del prototipo

Durante la tercera fase, el equipo SunReady y su personal desarrollan e integran los componentes para el ambiente de producción propuesto, incluyendo las herramientas de producción, en un ambiente prototipo. Gracias a que las pruebas se ejecutan en un ambiente prototipo seguro, se reducen los riesgos asociados a la integración de sistemas complejos a un ambiente de producción.

La tercera fase permite a Sun y a su personal desarrollar, integrar y comenzar a probar los componentes de producción, a la vez que se optimizan las políticas y los procedimientos operacionales. Las actividades de desarrollo de educación, entrenamiento y habilidades de esta fase se utilizan para implementar con éxito la próxima fase.

Al final de la tercera fase, se desarrollan y aprueban en el ambiente prototipo todos los componentes de producción especificados. Se desarrolla el manual de operación SunReady, el cual refleja los resultados de la implementación de la solución para el ambiente prototipo.

3.6. Implementación del prototipo

- Desarrollar los componentes para el ambiente prototipo
- Completar la integración inicial y las pruebas del ambiente prototipo
- Completar el entrenamiento del personal
- Elaborar y validar el manual de operación
- Obtener aprobaciones del ambiente prototipo

3.7. Fase 4: Implementación del ambiente de producción

En la cuarta fase, los especialistas de Sun y su personal configuran e integran el ambiente de producción y comienzan una rigurosa serie de pruebas para garantizar la estabilidad y la capacidad de recuperación del sistema y de los datos. Se incluyen pruebas de resistencia y falla del ambiente de producción, preparando el camino para colocar la solución en producción.

Tanto Sun como su personal recientemente entrenado, llevan a cabo la aprobación de las pruebas del ambiente de producción. Estas pruebas se basan en el procedimiento de trabajo y los objetivos

establecidos para los niveles de servicio. El equipo SunReady evalúa los resultados de las pruebas de resistencia y falla, efectuando cualquier cambio necesario al manual de operación. Una vez que usted aprueba el ambiente de producción, todo estará listo para que la solución entre en marcha.

Planeación de la implementación

- Implementar el ambiente de producción
- Completar las pruebas de resistencia para validar la capacidad y el rendimiento del ambiente
- Completar las pruebas de falla para verificar tanto las capacidades de recuperación automáticas como por procesos
- Efectuar pruebas de aprobación del ambiente de producción
- Ajustar y validar el manual de ejecución, basándose en los resultados de las pruebas
- Obtener aprobación del ambiente en el servidor de producción
- Obtener aprobación del plan de entrega para proveer soporte al ambiente de producción

3.8. Fase 5: Colocación del ambiente en producción

En la fase final, el ambiente se coloca en producción. Una vez concluida esta fase, el ambiente de producción SunReady estará activo.

Su equipo del centro de cómputo desempeña un papel esencial en este proceso. Tanto el rendimiento como la disponibilidad son monitoreados muy de cerca según el procedimiento de trabajo y los objetivos de los niveles de servicio. Se inicia un período de transición por parte de su equipo SunReady, para garantizar que su personal se sienta confortable con el ambiente de producción desde el punto de vista administrativo y operacional. Se concluye así la implementación de la Metodología SunReady y pueden evaluarse sus requerimientos de soporte para el ambiente de producción.

3.9. Colocación del ambiente en producción

- Colocar el ambiente en producción
- Monitorear el rendimiento y la disponibilidad durante el período de transición
- Implementar los servicios de soporte contratados



Servicios de soporte para el ambiente SunReady

Para atender una amplia variedad de requerimientos posteriores a la implementación, Sun ofrece servicios de soporte para el ambiente SunReady. Los especialistas de Sun proporcionan un proceso para garantizar que cualquier cambio en los requerimientos de los niveles de servicio también se refleje en el ambiente de producción. Los especialistas de Sun prestan los servicios de soporte y se mantienen al tanto de su operación, monitorean y elaboran los informes sobre los niveles de servicio y recomiendan los cambios necesarios para optimizar el rendimiento y satisfacer sus objetivos.

3.10. Beneficios de la Metodología SunReady

- Un ambiente de producción personalizado, diseñado para ayudarlo a cumplir con sus compromisos de niveles de servicio con sus usuarios
- Menor riesgo operacional a través de una metodología de cinco fases que incluye el análisis, integración y pruebas del nivel de producción
- Un proceso de servicio integral que garantiza que la arquitectura técnica esté soportada por procesos sólidos y personal adecuadamente entrenado
- Una infraestructura preparada para el futuro que evoluciona fácilmente al ritmo de las crecientes demandas de su ambiente
- Transferencia del conocimiento operacional de Sun a su personal a través de la evaluación de habilidades, asesoría y programas de entrenamiento personalizados
- Operación autosuficiente y soporte para su ambiente Sun a través de procesos totalmente documentados
- Implementación correcta y económica, basada en la tecnología para centros de cómputo líder del mercado y las capacidades básicas de comercio electrónico

http://ve.sun.com/service/sunps/Methodology_Span.html

4.- CMM/ Introducción

En noviembre de 1986, The Software Engineering Institute (SEI), con la asistencia de MITRE corporation, comenzó a desarrollar un proceso de madurez por niveles, que ayudaría a las organizaciones a mejorar sus procesos de desarrollo de software. En septiembre de 1987, el SEI (Software Engineering Institute) publicó una breve descripción de los niveles de madurez de los procesos de software, el cual fue desarrollado en el libro de Humphrey "Managing the Software Process".

Después de cuatro años de experiencia con la madurez del proceso de software, el SEI evolucionó la madurez y publicó Capability Maturity Model for Software (CMM).

La primera publicación de las CMM fue revisada y usada por la comunidad de software durante 1991 y 1992. La versión 1.1 fue publicada en 1993. Y en 1996 fue liberada la versión 2 del CMM, que evolucionó integrando diferentes métodos en la mejora de los procesos, como los estándares ISO.

EL Modelo de Madurez de Capacidades ("Capability Maturity Model") es un marco de trabajo que describe los elementos claves de un proceso de software eficaz. Describe un camino de mejoramiento evolutivo para pasar desde un proceso inmaduro a un proceso maduro y disciplinado, basado en conocimientos adquiridos de evaluaciones de los procesos de software y extensos feedback con industrias y el gobierno.

Por lo tanto, las disposiciones del CMM son definitivamente aplicables a todo aquello que esté directamente relacionado con el desarrollo y mantenimiento de sistemas informáticos.

4.1 Niveles de Madurez y Áreas Clave de CMM

El CMM proporciona un marco para evolucionar la organización en cinco niveles de madurez (ver esquema en Figura 4-1). El nombre que acompaña a las flechas de la figura indica el tipo de proceso capaz de ser institucionalizado por la organización. Los cinco niveles pueden ser descritos brevemente como:

1._Inicial: El proceso de software se caracteriza como ad hoc, y ocasionalmente caótico. Pocas actividades están definidas y el éxito de los proyectos depende del esfuerzo individual. Carencia de procedimientos formales, estimaciones de costo, planes del proyecto, mecanismo de administración para asegurar que los procedimientos se siguen.

2._Repetible: Son establecidas las actividades básicas para la administración de proyectos de software para el seguimiento de costos, programación y funcionalidad. El éxito está en repetir prácticas que hicieron posible el éxito de proyectos anteriores. Por lo tanto hay fortalezas cuando se desarrollan proyectos similares y gran riesgo cuando se enfrentan nuevos desafíos.

3._Definido: Las actividades del proceso de software para la administración e ingeniería están documentadas, estandarizadas e integradas en un proceso de software estándar para la organización.

4._Administrado: Medidas detalladas de las actividades del proceso y calidad del producto son registradas. El proceso de software y el producto entregado son cuantitativamente entendidos y controlados.

5._Optimizado: Existe una mejora continua de las actividades, la que se logra, a través, de un feedback con estas mismas y también a partir de innovadoras ideas y tecnologías. La recolección de datos es automatizada y usada para identificar elementos más débiles del proceso, se hacen rigurosos análisis de causas y prevención de defectos.

Synapsis S.A. se encuentra ubicada en el nivel 1 de madurez, y con el objetivo de visualizar hacia dónde se enfocarán las propuestas que se harán en este proyecto, se describirá con mayor detalle las características de madurez del nivel 2.

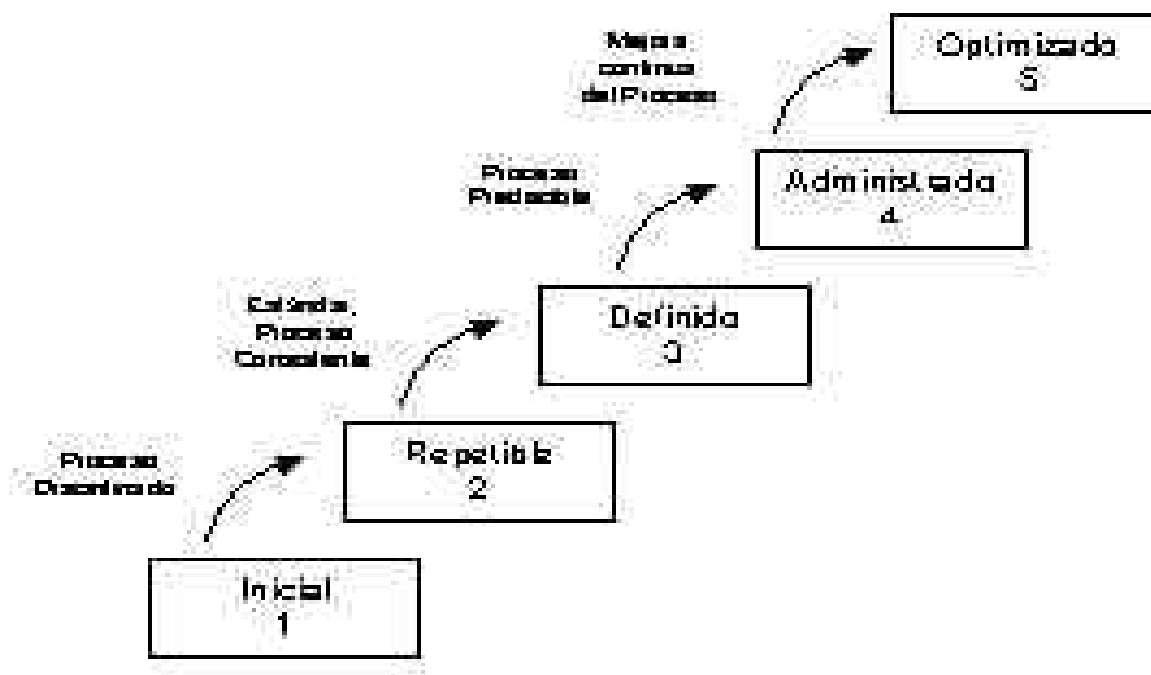


Figura 4-1: Los cinco niveles del proceso de maduración

4.2. Caracterización del Nivel 2. Repetible

En este nivel las políticas para la administración de proyectos de software y procedimientos para implementar dichas políticas están establecidas. La planificación y administración de nuevos proyectos, están basadas en experiencia con similares proyectos y se mejora estableciendo disciplinas básicas.

Un proceso efectivo puede ser caracterizado como un proceso donde se tienen experiencias anteriores de proyectos similares, existe documentación, es especializado, medido y capaz de ser mejorado.

Las organizaciones que se encuentran en el nivel 2, tienen instalado un sistema básico de control para la administración de sus proyectos. Por esto los problemas en el cumplimiento de acuerdos son identificados a medida que ellos surgen.

Estándares y planes para los proyectos de software son definidos, y la organización garantiza que ellos sean seguidos fielmente.

La organización en este nivel puede definirse como disciplinada, porque la planificación y seguimiento de proyectos de software es estable y los anteriores éxitos pueden ser repetidos.

4.3. Áreas claves de proceso para el Nivel 2. Repetible

Cada nivel de madurez (excepto el nivel 1) se descompone en áreas claves de proceso (KPA), prácticas claves e indicadores claves.³

- Áreas claves: Identifican objetivos a ser alcanzados para llegar a un nivel de madurez especial.
- Prácticas claves: Procedimientos y actividades que contribuyen a alcanzar los objetivos.
- Indicadores claves: Ayudan a determinar el cumplimiento de los objetivos, forma la base para el procedimiento de evaluación.

El nivel de madurez se establece como aquél en que se satisfacen todas las áreas claves en forma continua. Estas áreas claves priorizan los esfuerzos para mejorar el proceso de desarrollo de software en la organización.

Las áreas claves correspondientes al nivel 2 son:

- Administración de requerimientos
- Planificación de proyectos de software
- Supervisión y seguimiento de proyectos de software
- Gestión de subcontratos de software
- Aseguramiento de calidad de software
- Administración de la configuración de software

4.4. Administración de requerimientos

El propósito de la administración de requerimientos es establecer un común entendimiento de las necesidades del usuario, las que deben estar documentadas. El entendimiento de los requerimientos es fundamental para la construcción de software que será realizado al cliente. Revisar los requerimientos e interactuar con el cliente es parte de establecer este entendimiento. Los requerimientos del cliente están cambiando frecuentemente, por lo que la documentación y el control de éstos es prerequisite para usarlos como base para la estimación, planificación, desarrollo y seguimiento de las actividades del proyecto de software, a través de su ciclo de vida.

4.5. Planificación de proyectos de software

El propósito de la planificación es establecer un plan razonable para el desarrollo y administración del proyecto de software. Los planes razonables están basados en estimaciones realistas del trabajo que

permitan establecer los compromisos necesarios para desarrollar el proyecto. La planificación incluye pasos para estimar la cantidad de trabajo y los recursos necesarios.

4.6. Supervisión y seguimiento de proyectos de software

El propósito del seguimiento y supervisión de proyectos de software es establecer una visibilidad adecuada del progreso real del proyecto, para que la gestión pueda tomar acciones efectivas cuando el proyecto se desvía significativamente de la planificación realizada.

Gestión de subcontratos de software El propósito de la administración de subcontrato de software es seleccionar subcontratistas calificados y administrarlos efectivamente.

Aseguramiento de calidad de software El propósito del aseguramiento de la calidad de software involucra una revisión y análisis del producto de software y las actividades, para verificar que cumplen con los procedimientos y estándares aplicados.

Administración de la configuración de software El propósito de la administración de la configuración de software es establecer y mantener la integridad de los productos de software, a través de su ciclo de vida. Esto significa controlar los cambios de los productos, registrar y mantener las bibliotecas de programas.

El rediseño se centrará en las primeras tres áreas claves.

En este proyecto las mejoras más significativas se refieren a la Planificación de Proyectos de Software, ya que es debido a esta falencia que se producen costos no considerados en los proyectos y entregas del producto en fechas fuera del plazo comprometidas con el cliente. Para que esta mejora tenga sentido es necesario que los requisitos del usuario se relacionen con la métrica utilizada para realizar la planificación, es por esta razón que se tratará también en este proyecto la Administración de Requerimientos. Y finalmente, a raíz de los cambios que se propondrán al enfocarse en las dos primeras áreas claves, es que se justifica realizar Supervisión y Seguimiento del Proyecto para tomar medidas cuando éste se aleje de la planificación inicial.

En Synapsis S.A. no se subcontratan servicios de software, por esto el área clave de Gestión de subcontratos de software no será tratada en este proyecto. Tampoco el Aseguramiento de calidad de software y la Administración de la configuración.

En el siguiente punto se describe la métrica "Puntos de Función", elegida para desarrollar el área clave Planificación de Proyectos de Software.

La métrica tradicional para estimar el esfuerzo de desarrollo y productividad ha sido LOC (Lines of Code) o SLOC (Source Lines of Code). Sin embargo fueron elegidos los puntos de función porque:

- Es independiente del lenguaje, herramienta o metodología utilizada en la implementación
- Pueden ser estimados a partir de la especificación de requisitos o especificaciones de diseño, haciendo posible de este modo la estimación del esfuerzo de desarrollo en etapas tempranas del mismo.

En resumen, los puntos de función aparecen con ventajas substanciales por sobre las líneas de código, para fines de estimación temprana del tamaño del software, y por ende, del esfuerzo de desarrollo. Además es una medida ampliamente utilizada, y con éxito, en muchas organizaciones que desarrollan software en forma masiva [Varas95].

4.7. Puntos de Función

Todo proyecto de ingeniería de software debe partir con un buen plan, pero lamentablemente, la planificación es una tarea nada de trivial. Uno de los aspectos que dificulta la labor de administradores y jefes de proyecto en torno a la planificación es la difícil tarea de realizar una estimación de costos y plazos realistas.

El manejador principal de los costos de un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad.

Al contar con una estimación temprana del tamaño de lo que se desea desarrollar, se puede realizar una estimación del esfuerzo en etapas tempranas del desarrollo. Esto debido a que el tamaño del software es la variable manejadora de costo principal del desarrollo.

La métrica Puntos de Función (PFs), desarrollada por A. J. Albrecht [Albrecht79], fue la primera métrica orientada a la función y sugiere un acercamiento a la medida de productividad. Los puntos de función se obtienen utilizando una relación empírica basada en medidas cuantitativas del dominio de información de software y valorizaciones subjetivas de la complejidad del software.

Esta técnica aporta una medida estándar del tamaño de los sistemas de información, y sirve de base para la estimación del esfuerzo requerido para el desarrollo de los proyectos.

La medida de los sistemas de información mediante los puntos de función proporciona una estimación del tamaño del software independiente de la tecnología utilizada en su desarrollo y dependiente únicamente de la funcionalidad que el sistema proporciona al usuario. Esto quiere decir que la estimación se refiere a los resultados que se obtienen de un sistema de información y no cómo se producen internamente estos resultados.⁴

4.8. Componentes de la Evaluación

Para calcular los PF se deben realizar dos conteos:

- Conteo de tipos de funciones transaccionales
- Conteo de tipos de funciones de datos

Las Funciones Transaccionales representan la funcionalidad provista al usuario de los procesos de datos de una aplicación. El conteo de tipos de funciones transaccionales determina la cantidad de Entradas Externas, Salidas Externas y Consultas Externas.

Entradas Externas (EI): Cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación. Las entradas deben ser distinguidas de las peticiones, que se contabilizan por separado. En general cada archivo lógico requerirá de tres tipos de entradas: agregar, cambiar y borrar. También se define como datos o información de control que llega desde fuera de los límites de la aplicación o proyecto que está siendo medido.

Salidas Externas (EO): Cada salida que proporciona al usuario información orientada a la aplicación. La salida se refiere a reportes, datos a pantalla, mensajes de error, etc. Los elementos de datos individuales dentro de un informe no se cuentan por separado. También se define como datos o información de control que se envía desde fuera de los límites de la aplicación.

Consultas Externas (EQ): Es una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva. Se cuenta cada petición por separado.

También se define como una combinación de entradas (requerimientos) y salidas (recuperación) de información desde la aplicación.

El conteo de Funciones de Datos determina la cantidad de Archivos Lógicos Internos y de Archivos de Interfaz Externa perteneciente a la aplicación que está siendo contabilizada.

Archivos Lógicos Internos (ILF): Cada maestro lógico (o sea, una agrupación lógica de datos que puede ser una parte de una gran base de datos o un archivo independiente).

También se define como datos relacionados lógicamente o información de control que se encuentra dentro de los límites de la aplicación.

Archivos Interfaz Externos (EIF): Todas las interfaces legibles por la máquina (por ejemplo, archivos de datos en cinta o disco) que son utilizados para transmitir información a otro sistema.

También se define como datos relacionados lógicamente o información de control que se encuentran fuera de los límites de la aplicación.

4.9. Complejidad para cada Función

Para cada función de datos o transaccionales identificada se debe determinar su complejidad. La complejidad para los ILF e EIF es la misma y se obtiene basándose en la cantidad de DETs y RETs contenidos en cada función. En cambio la complejidad de las EI, EO y EQ se obtiene basándose en la cantidad de DETs y FTRs involucrados. A continuación se describen estos elementos.

Tipo Elemental de Dato (DET) es un campo no recursivo perteneciente a un archivo lógico interno o a un archivo de interfaz externa y que es reconocible por el usuario.

Tipo Elemental de Registro (RET) es subgrupo de elementos de datos reconocibles por el usuario y que están dentro de un archivo lógico interno o dentro de un archivo de interfaz externa.

Tipo de Archivo Referenciado (FTR) es un archivo lógico interno que es leído o mantenido por una función o un archivo de interfaz externa que es leído por una función de transacción (EI, EO o EQ).

A continuación se muestran las tablas que determinan la complejidad de las funciones de datos y transaccionales, a partir de la relación que exista entre DET/RET o DET/FTR.

Tabla que permite obtener la complejidad para un ILF e EIF.

	1 a 19 DETs	20 a 50 DETs	51 o más DETs
--	-------------	--------------	---------------

1 RET	Baja	Baja	Media
2 a 3 RETs	Baja	Media	Alta
4 o más RETs	Media	Alta	Alta

Tabla 4-1: Complejidad de ILFs e EIFs [IFPUG]

Tabla que permite obtener la complejidad para un EI.

	1 a 4 DETs	5 a 15 DETs	16 o más DETs
0 a 1 FTR	Baja	Baja	Media
2 FTRs	Baja	Media	Alta
3 o más FTR s	Media	Alta	Alta

Tabla 4-2: Complejidad de EI [IFPUG]

Tabla que permite obtener la complejidad para un EO.

	1 a 5 DETs	6 a 19 DETs	20 o más DETs
0 a 1 FTR	Baja	Baja	Media
2 a 3 FTRs	Baja	Media	Alta
4 o más FTR s	Media	Alta	Alta

Tabla 4-3: Complejidad de EO [IFPUG]

Tabla que permite obtener la complejidad para un EQ en el lado Input.

	1 a 4 DETs	5 a 15 DETs	16 o más DETs
0 a 1 FTR	Baja	Baja	Media
2 FTRs	Baja	Media	Alta
3 o más FTR s	Media	Alta	Alta

Tabla 4-4: Complejidad de EQ - lado Input [IFPUG]

Tabla que permite obtener la complejidad para un EQ en el lado Output.

	1 a 5 DETs	6 a 19 DETs	20 o más DETs
0 a 1 FTR	Baja	Baja	Media
2 a 3 FTRs	Baja	Media	Alta
4 o más FTR s	Media	Alta	Alta

Tabla 4-5: Complejidad de EQ - lado Output [IFPUG] La complejidad total de la EQ se obtiene comparando las complejidades determinadas para el lado Input y para el lado Output, prevaleciendo la complejidad mayor.

4.10. Cálculo de PFs

Fórmula para medir una Aplicación

Para cada tipo de función y de acuerdo a su complejidad, se determinó un factor de peso, el cual se utiliza para obtener el cálculo de puntos de función bruto o no ajustados. Este cálculo consiste en sumar, para cada tipo de función, la cantidad de elementos de cada complejidad multiplicada por el factor de peso correspondiente.

Tipo de Función	Simple	Medio	Complejo	Total
N° Entradas Externas	___ * 3	___ * 4	___ * 6	
N° Salidas Externas	___ * 4	___ * 5	___ * 7	
N° Consultas Externas	___ * 3	___ * 4	___ * 6	
N° Archivos Lógicos Internos	___ * 7	___ * 10	___ * 15	
N° Archivos de Interfaz Externa	___ * 5	___ * 7	___ * 10	
Puntos de Función Bruto (PFB):				

Tabla 4-6: Cálculo Punto de Función Bruto

Para obtener el punto de función, el punto de función bruto se ajusta multiplicándolo por un factor, como lo indica la siguiente relación:

PF: Punto de Función

PFB: Punto de Función Bruto

Fa: Factor de ajuste

Este factor se denomina factor de ajuste y representa los valores de ajuste de la complejidad considerando características técnicas y operacionales del sistema. Este factor se basa en la respuesta a un cuestionario de 14 ítems. Se evalúa cada ítem en una escala de 0 a 5 según el grado de influencia que posean en el sistema.

El factor de ajuste está dado por:

Gi: Grado de influencia, valor que va de 0 a 5.

Significado de la escala:

0 = La característica no está presente o no influye si está presente

1 = La característica tiene una influencia Incidental

2 = La característica tiene una influencia moderada

3 = La característica tiene una influencia promedio

4 = La característica tiene una influencia significativa

5 = La característica tiene una influencia fuerte, es esencial

Cuestionario de 14 ítems:

1. Comunicación de datos
2. Proceso de datos distribuido
3. Rendimiento (Performance)
4. Alta utilización del entorno operativo
5. Volumen de transacciones
6. Entrada de datos interactiva
7. Eficiencia de uso por parte del usuario
8. Actualización interactiva
9. Procesamiento complejo
10. Reusabilidad del código
11. Facilidad de conversión e instalación

- 12. Operatividad en cuanto al manejo de seguridad y recuperación de errores
 - 13. Facilidad de soportar múltiples instalaciones en diferentes organizaciones
 - 14. Facilidad para permitir cambios o modificaciones
- Significado de cada ítem:

1. Comunicación de datos

Los datos e información de control usados en la aplicación son enviados o recibidos por las facilidades de comunicación. Los terminales conectados localmente a la unidad de control son considerados para usar estas facilidades de comunicación. El protocolo es un conjunto de convenciones que permiten la transferencia o intercambio de información entre dos sistemas o dispositivos. Todas las conexiones de datos requieren algún tipo de protocolo.

2. Proceso de datos distribuido

El proceso distribuido de datos o funciones es una característica de la aplicación, dentro de los límites de la misma. El procesamiento distribuido se refiere a la utilización de llamadas a procedimientos remotos para operar con datos o funciones que se encuentran en máquinas distintas de donde se encuentra instalada la aplicación.

3. Rendimiento (Performance)

Se mide los objetivos de rendimiento (performance) iniciales o probados por el usuario, en relación a la influencia de los tiempos de respuesta (throughput) en el desarrollo, instalación y mantención de la aplicación.

4. Alta utilización del entorno operativo

La configuración para un entorno operativo fuertemente utilizado requiere consideraciones especiales de diseño, lo que hace de ésta una característica propia de la aplicación. Se refiere a si la aplicación requerirá de un procesador especial, si hay consideraciones de seguridad o si se aplican restricciones en los distintos componentes de un sistema.

5. Volumen de transacciones

Esta característica tiene una alta influencia en el diseño, desarrollo, instalación y mantención de una aplicación. Se refiere a la periodicidad con que se requiere generar gran cantidad de transacciones. Por ejemplo, un proceso de facturación genera una gran cantidad de transacciones al término de cada mes.

6. Entrada de datos interactiva

Se determina la cantidad de entrada de datos en línea y funciones de control que son provistas en la aplicación.

7. Eficiencia de uso por parte del usuario

Son las facilidades incluidas en el diseño de una aplicación para ayudar a obtener una mayor eficiencia de uso por parte del usuario. Por ejemplo, ayudas de navegación, menús, ayuda y documentos en línea, impresión remota, etc.

8. Actualización interactiva

Se verifica si la aplicación permite hacer una actualización en línea de los archivos lógicos internos.

9. Procesamiento complejo

Se determina qué componentes están en el procesamiento que hace la aplicación:

- Control sensitivo y/o procesamiento de seguridad específico de la aplicación
- Procesamiento lógico extensivo
- Procesamiento de excepciones que resultan de transacciones incompletas que deben ser aprobadas otra vez.

10. Reusabilidad del código

Se verifica si el código de la aplicación ha sido específicamente diseñado, desarrollado y mantenido para ser usado en otras aplicaciones.

11. Facilidad de conversión e instalación

Una fácil conversión e instalación son características de una aplicación. Se determina si existe un plan de conversión e instalación y/o herramientas de conversión que han sido provistas y probadas durante la fase de prueba del sistema.

12. Operatibilidad en cuanto al manejo de seguridad y recuperación de errores

Se determina si la aplicación provee una partida, respaldo y recuperación de errores efectiva, lo que es estudiado en la fase de prueba del sistema. Se verifica si la aplicación minimiza la necesidad de actividades manuales, tales como, montar cintas, manipulación de papel, intervención directa de un operador, etc.

13. Facilidad de soportar múltiples instalaciones en diferentes organizaciones

Se verifica si la aplicación ha sido diseñada y desarrollada para ser instalada en múltiples sitios y para múltiples instalaciones de la misma.

14. Facilidad para permitir cambios o modificaciones

Se refiere a si la aplicación ha sido específicamente diseñada y desarrollada para facilitar el cambio. Características como las siguientes pueden aplicarse:

- Se proveen consultas e informes flexibles para manipular una recuperación de datos simple, de complejidad media o alta

- Datos de control de gestión almacenados en tablas que son mantenidas por el usuario con procesos interactivos en línea, pero cuyos cambios sólo tienen efecto el día siguiente
- Datos de control de gestión almacenados en tablas que son mantenidas por el usuario con procesos interactivos en línea, pero cuyos cambios sólo tienen efecto inmediatamente

4.11. Fórmula para Medir un Proyecto de Mejoramiento

$$EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB)$$

Donde:

EFP : Cuenta de puntos de función para proyecto de mejoramiento

ADD : Cuenta no ajustada de PF para funciones que fueron agregadas con el mejoramiento

CHGA : Cuenta no ajustada de PF para funciones que fueron modificadas con el mejoramiento

CFP : Cuenta no ajustada de PF sumados por la conversión de funcionalidad

VAFA : Valor del factor de ajuste después del mejoramiento

DEL : Cuenta no ajustada de PF para funciones que fueron eliminadas con el mejoramiento

VAFB : Valor del factor de ajuste antes del mejoramiento

En el punto 4.3. se da el marco teórico de la metodología utilizada para desarrollar el área clave Administración de Requerimientos. Esta metodología para la captura y definición de los requerimientos, denominada Casos de Uso, fue elegida porque permite relacionar fácilmente los requerimientos solicitados por el cliente con las funciones transaccionales y de datos, necesarias para el cálculo de los puntos de función y la consecuente planificación del proyecto.

4.12. Caso de Uso

Los casos de uso fueron introducidos por Ivar Jacobson en 1994 y permiten realizar la especificación de un sistema centrada en el usuario. En esencia un caso de uso representa una interacción típica entre un usuario y un sistema o aplicación computacional. Para ilustrar lo anterior, supongamos un usuario utilizando un procesador de texto. Dos casos de uso típicos podrían ser "asignar negrita a una palabra" o "crear un índice". Con los ejemplos anteriores se pueden identificar algunas de las propiedades de los casos de uso:

- Capturan alguna funcionalidad visible para el usuario
- Pueden ser pequeños o grandes
- Satisfacen un objetivo del usuario

En su forma más simple, los casos de uso son identificados mediante conversaciones con el usuario y discutiendo las distintas operaciones que el sistema debe proveer. Cada funcionalidad u objetivo discreto debe ser documentado asignándole un nombre y una breve descripción.

La funcionalidad que describe un caso de uso determinado puede ser interpretada como una Interacción de Sistema u Objetivo de Usuario. La interacción de sistema permite describir las operaciones que el usuario realiza para satisfacer un objetivo. El objetivo del usuario, como su nombre lo indica, refleja un objetivo que el usuario desea satisfacer. Por ejemplo, siguiendo con el ejemplo del procesador de texto, algunas interacciones de sistema podrían ser "asignar formato a un párrafo", "definir un nuevo estilo" o "modificar un estilo". Sin embargo, éstas no reflejan el objetivo del usuario que podría ser "mantener un formato consistente para todos los documentos". Es necesario tomar en cuenta esta diferencia al momento de modelar los casos de uso.

Los casos de uso incluyen, además, Actores. Un actor representa una entidad externa que se relaciona directamente con el sistema. Los actores representan humanos, máquinas u otros sistemas. En definitiva, un actor corresponde al rol que juega alguna de las entidades anteriores frente al sistema. Puede haber muchos usuarios con un mismo rol. Un usuario también puede poseer distintos roles. Por esta razón, al considerar los actores del sistema se debe centrar el análisis en los roles y no en las personas.

Los actores son los que realizan los casos de uso. Un actor puede realizar varios casos de uso; alternativamente, un caso de uso puede ser realizado por varios actores.

Los actores apoyan la identificación de los casos de uso. Enfrentado a un sistema muy grande, intentar identificar la lista de casos de uso puede ser una tarea muy compleja. La alternativa adecuada es identificar la lista de actores del sistema y luego intentar identificar los casos de uso de cada actor.

Como se mencionó anteriormente, los actores no necesariamente deben ser humanos. Un actor también puede ser un sistema externo que requiere información del sistema actual. La interacción con sistemas externos produce confusión en la definición de los actores que representan a los sistemas externos. Es recomendable que se incluyan actores para reflejar sistemas externos sólo cuando el actor (externo) requiere del caso de uso modelado. De esta forma, las entidades que están indirectamente relacionadas con el sistema no son consideradas actores debido a que sus necesidades deben ser comunicadas al sistema mediante un actor. Cada actor utiliza el sistema de distintas maneras; de lo contrario los actores no serían distintos.

Cada caso de uso describe la secuencia posible de interacciones entre el sistema y uno o más actores como consecuencia de un estímulo inicial de alguno de los actores. Para definir los casos de uso se deben agrupar todas aquellas transacciones "similares"; aquellas que el usuario vería como variaciones de algún tema. Por ejemplo, un caso de uso para un banco podría ser "realizar una transacción en la caja". Los casos de uso que se pueden desprender del anterior podrían ser realizar un depósito, efectuar un retiro, transferencias, etc.

Las excepciones corresponden a situaciones anormales e incluye las medidas que es necesario realizar para poder completar el caso de uso.

Las precondiciones corresponden a las condiciones que se deben cumplir para que se pueda realizar el caso de uso.

Las postcondiciones corresponden a condiciones que se deben cumplir una vez que el caso de uso ha terminado exitosamente.

Fuente: <http://www.sei.cmu.edu>

<http://www.avantare.com/antiores/ElModeloCMM.pdf>

<http://www.ucc.edu.co/columbus/salas/sala3523.htm>

<http://www.cutter.com/research/2000/crb000613.html>

http://www.cse.fau.edu/~maria/COURSES/CEN4010-SE/TOPIC3/cmm_contents.htm

5.- Aplicaciones con Microsoft

- Microsoft SQL Server
- Microsoft Visual Studio .Net
- Active Server Pages
- Microsoft Access
- Microsoft Visual Basic

5.1. Beneficios

Listamos abajo algunos beneficios, que muestran como aparecería con una fábrica de software y como se puede mejorar el desempeño de un Centro de Proceso de Datos:

- Estimación de plazos y costos con bastante precisión.
- Alta productividad, permitiendo rápida reducción de backups
- Reducción de costos como factores de improductividad, actualización tecnológica de equipos internos e infraestructura del área de desarrollo.
- Concentración de esfuerzos de profesionales de IT y de los usuarios mismos
- Mejora de procesos de productividad, control total de inventario y mejora de controles de calidad de software
- Sincronización de documentos, código y aplicaciones particulares
- Posibilidad de contratación de etapas específicas de desarrollo.
- Mayor facilidad para entrenar usuarios, técnicos en software y equipo

Líderes empresariales de la informática promueven el desarrollo de un complejo industrial del Software en México, con una inversión de unos 18 millones de dólares sólo en capacitación.

Roberto Solís, líder programa “Fomento al Desarrollo de la Industria del Software”, dijo que éste es auspiciado desde hace un año por empresas desarrolladas nacionales y el corporativo mexicano de Microsoft.

“Se trata de crear una plataforma sólida de técnicos especialistas que aliente a los inversionistas a realizar sus desarrollos en cómputo dentro del el país”, explicó Eva María García, ejecutiva de Microsoft.

Para Rafael Bernal, miembro de la asociación Mexicana de la Industria de las Tecnologías de la Información (AMITI), las fábricas de software tienen alta rentabilidad y su principal soporte es el capital humano.

Al respecto Solís expuso que en Estados Unidos un curso de capacitación igual al que Microsoft ofrece en el marco del programa tiene un costo para los estudiantes de 60 mil dólares. Para la AMITI, la promoción de fábricas de software en México está en sus prioridades ya que es una de las actividades de la Industria que mayor impacto económico tiene en los países que lo albergan.

“Es un segmento con una enorme capacidad de generación de empleos altamente remunerados y con un excelente factor multiplicador, lo que hace crecer la economía de los lugares que promueven este negocio” dijo Bernal.

Estas fábricas, agregó, tienen además la ventaja de ser “industria verde”, pues no generan desechos industriales y dan a pequeñas empresas la posibilidad de participar con productos terminados de gran valor agregado.

Reconocido como el maquilador por excelencia de software, ha sabido explotar este negocio y está exportando hasta cuatro mil millones de dólares al año por este concepto. Pero la enorme demanda para la maquila de productos de cómputo y el déficit mundial de profesional técnicos está agotando su capacidad de servicios y elevando los costos del trabajo de sus programadores.

El sueldo promedio de un programador en la India es de 500 dólares mensuales, estimaron los especialistas.

Según Bernal, “en México, el terreno de los desarrolladores de software está poco explotado y existen muy pocas industrias abocadas a esta tarea”. “Los más conocidos son Softtek y Serinf, también existen fábricas de software que “son privadas”, esto es, que la tiene una empresa para sus propios desarrollos, como Ddémesis, adquirida por GE Capital, y Tecnosys, de IBM” agregó.

“Entendemos que hay intentos de ir al exterior de parte de otras compañías de cierto tamaño y hay algunos casos de pequeñas compañías que están haciendo la posible a través de contactos personales” apuntó.

La consultora Select, filial de International Data Corporation, considera que la actividad de las empresas desarrolladas de software en México económicamente es aún incipiente. El software a la medida carece de demanda importante entre las empresas mexicanas ya que sus costos son elevados, dijo Alejandro Florean Gerente de software e Internet Working de Select. “Pese a ello, durante 1999 generaron importantes utilidades estas se debieron a las necesidades que tenían las empresas de adecuar sus sistemas al año 2000”. Según las proyecciones de Bernal, una fábrica de software con unos tres mil empleados podría ser rentable en dos o tres años con una inversión de los 10 millones de dólares. El programa de Fomento de Desarrollo de la Industria de Software está dirigido a la capacitación de estudiantes de escuelas públicas de educación tecnológica del central estado mexicano de Guanajuato.

Contempla en su fase inicial la capacitación de 300 estudiantes como desarrolladores de informática y pretende promover la inversión directa de capitales en el estado con su consecuente generación de empleos.

A un año de haberse instrumentado, el programa cuenta con cinco células de desarrollo de software financiadas por industriales, donde los estudiantes se desenvuelven sobre proyectos reales.

5.2. Nace el plan fast-track para el desarrollo del Software Mexicano

Hoy día ha terminado la actual administración federal y luego de casi cinco años y medio de trabajo en SECOFI, se ha propuesto la elaboración de un plan fast-track para estimular el desarrollo del software en México.

Fuentes de esta rama industrial indicaron que podría haber la intención del subsecretario Décimo de María Serrano de proponer ese programa al gobierno entrante.

Los informantes comentaron hoy, que podrían reunirse algunas empresas desarrolladoras de programas de computación con el funcionario, quien es uno de los integrantes del gobierno mexicano "que más interés han puesto en los temas de las tecnologías de la información".

Ello ocurre luego que durante años se ha insistido en el gran potencial que significa esa rama industrial, al tener como vecino al mercado más importante del mundo. Se estima que el mercado de programas de computación de EE.UU. es una oportunidad de negocio de 70 mil millones de dólares y que, sin embargo, son las fábricas de software de India, las que han aprovechado mejor esa oportunidad al participar ya con alrededor de tres millones de dólares de ese mercado.

Las oportunidades de negocio con EE.UU. sobresalen aún más cuando se tiene en cuenta que en aquella nación se ha registrado un creciente déficit de desarrolladores que podría superar los 350 mil, problema que también existe en Europa con más de 120 mil personas que se requieren.

Entre las empresas mexicanas que se dedican al desarrollo de software destacan Softec, Microsoft, Praxis y Denesis, entre otras.

Por otro lado, en diversas ocasiones se han denunciado la falta de apoyo de las autoridades mexicanas, por estar retiradas del fomento de cualquier rama industrial.

Asimismo muchos de los intentos de acercamiento entre otras compañías desarrolladoras de programas de computación para concretar planes y esfuerzos conjuntos han fracasado por tratarse de una lucha común entre competidores. Pero, según los observadores, México cuenta con importantes ventajas de lograr un plan de fomento al desarrollo de los programas de computación en México, como es el hecho de que posee recursos humanos de alto nivel en la materia (algo de lo que carece EE.UU. aún cuando es una potencia mundial muy importante), la situación geográfica junto al mercado más importante, aunque con desventajas, como los estímulos fiscales de que gozan los desarrolladores en otras partes del mundo, como en el caso de India que el pago de impuesto no es tan alto como en América latina.

Las mayores incógnitas de muchos mexicanos son:

¿Hacia qué destino se dirige la visión computacional en nuestro país?

Primero se aplicará a problemas donde ya ha tenido éxito en otros países y, posteriormente, surgirán aplicaciones originales aquí. Por la gran diversidad de requerimientos de las áreas de servicios y salud (y por supuesto también administrativa), esta tecnología evolucionará de manera distinta en México. Es más probable que aquí se desarrollen aplicaciones para detectar ciertas configuraciones de células para el diagnóstico de un determinado padecimiento, que sea poco común o raro en otros países.

¿Cuáles son sus pronósticos para el desarrollo generalizado de esta tecnología?

Se reconocerá que hay muchas tareas repetitivas de inspección visual que son mejor realizadas mecánicamente. También existe la tendencia general de dotar de mayores capacidades preceptuales a los robots y computadoras, en especial de visión, por lo que esta tecnología seguirá desarrollando y ampliando su campo de acción.

Vaya, vaya, para muchos de nosotros, esto mas bien parece un sueño “guajiro” (como se dice aquí en México) pero recuerden que nada es difícil si nos aplicamos y hacemos nuestro mejor esfuerzo. Y si otros países han logrado una tecnología digna de una potencia mundial como Japón o EE.UU., no veo por que México no pueda también ser bueno para el desarrollo de software, ya que, los mexicanos contamos con gente muy capaz.

http://www.comsto.org/stuff/notmex_04.htm

<http://www.mexico.com/frames.php>

6.- Fábricas de Software en México

6.1. Sofftek

La compañía de soluciones de negocio y primera fábrica de software en México, Sofftek, implementó la modalidad de teletrabajo desde 1997, con un equipo de desarrolladores que desde Monterrey realizan sistemas para clientes que se encuentran en Estados Unidos y Europa.

Un equipo de 500 personas entre desarrolladores, implementadores en sitio y gente de soporte conforman el grupo que ha puesto en marcha el trabajo a distancia, aunque de acuerdo con Roberto Montelongo, director de la fábrica de software de Sofftek, el número podría aumentar, ya que se tiene contemplado que también sus clientes en México y el resto de Latinoamérica sean atendidos conforme a este esquema.

"Va a ser el modelo que vamos a estar siguiendo en todas partes, tener centros especializados para mejorar cada vez más los procesos, tener economías de escala y darle el beneficio al cliente, que no necesita tener a la gente y gastar en espacio físico", explicó Montelongo, "para lograrlo es vital el desarrollo que la industria de las telecomunicaciones ha tenido en México".

Blanca Treviño, directora ejecutiva de Sofftek, tuvo las agallas de invertir millones en la creación de una fábrica de software en Monterrey (México) para competir con India por el mercado corporativo de EE.UU. La compañía pasó por una severa crisis financiera, pero la apuesta dio fruto. Y ahora Treviño quiere exportar su modelo de fábrica a Europa.- El presidente Ernesto Zedillo, inauguró la empresa Sofftek, en este municipio. San Pedro Garza García, N.L., el día 20 de febrero de 1998.

Referencia:

<http://www.mural.com/tecnologia/articulo/131331/>

<http://zedillo.presidencia.gob.mx/pages/disc/feb98/20feb98-1.html>

<http://148.233.5.66/qp/datos/secciones/picudos/2001/1056/picudos.html>

“QUEREMOS POSICIONAR A MEXICO COMO EXPORTADOR DE TALENTO”

“La primera fábrica de software en América Latina, ubicada en Monterrey, ha sido para Softtek un elemento fundamental para penetrar a los mercados norteamericanos, pues le da la ventaja a México de generar divisas netas”, expresa Robespierre

Agrega que uno de los objetivos de la empresa es posicionar a México como proveedor y exportador de talento e inteligencia, y generador de empleos mejor pagados y con mayor posibilidad de desarrollo profesional, pues las fábricas de software son una industria que no tiene límites en cuanto a crecimiento.

“El logro de nuestras metas se sustenta en estrategias radicales que ya se están implantando. Para ello ha sido necesario generar una serie de compromisos, los cuales confirman el camino de la compañía”.

—¿Y cuáles son esos compromisos?

—Softtek es una compañía basada en el conocimiento. Incrementamos la capacidad de respuesta enfatizando la flexibilidad y rapidez con que la organización atiende y se vuelca al mercado. Seguimos esforzándonos por ser la opción más atractiva para los integrantes, alianzas y socios. Adaptamos la empresa a las nuevas tendencias tecnológicas, empacando hoy nuestra oferta como un e-software.

“TRABAJAMOS SIEMPRE CON UN PASO ADELANTE DE LA TECNOLOGIA”

Softtek es una de las 500 empresas más importantes de México, y su constante dedicación para satisfacer las necesidades de sus clientes ha trascendido más allá de las fronteras, dándole una importante presencia internacional. La apertura de oficinas en nueve países representa el interés de Softtek hacia la expansión de servicios y oportunidades de negocio, y le permite ser una importante fuente de generación de empleos, así como de valor económico y humano para la comunidad.

“Nuestra presencia internacional es una gran ventaja para nuestros clientes, al contar con una fuerte exposición a un alto número de tecnologías de vanguardia que surgen diariamente en diferentes países del mundo. Ello nos da la posibilidad de especializarnos en los diferentes procesos de adopción de tecnología de cada mercado, acordes en distintas prácticas de clase mundial”. Además su modelo de servicio, conjunta capacidades como la tecnología, el conocimiento, los procesos y las personas, para ofrecer soluciones integrales a la medida de cada proyecto; con una amplia gama de opciones de entrega para las diferentes soluciones de TI, de manera conveniente, oportuna y efectiva, es decir, en modalidades de administración, mantenimiento, o por el concepto fábrica de software, según convenga.

“El incremento en ventas y la constante reinversión de nuestras utilidades, han hecho posible el desarrollo de proyectos estratégicos, convirtiendo nuestros planes de expansión en hechos tangibles que reflejan nuestra firme creencia en el espíritu emprendedor como motor que permitirá el desarrollo

de todo el país.
“Hoy, nuestro personal ha crecido a un ritmo acelerado y constante y ponemos a disposición del mercado una organización de más de mil 500 especialistas en tecnología de información, comprometidos y dedicados a lo que mejor sabemos hacer: proporcionar soluciones a la medida, ya que implantamos el software para tu evolución..

DOS DECADAS DE TRABAJO INCANSABLE

Con casi 19 años de existencia, Softtek se ha posicionado en México, Estados Unidos, Europa y Latinoamérica, debido a que ofrece soluciones inteligentes, basadas en tecnologías de vanguardia mundial. Además, cuenta con más de 250 socios, quienes son poseedores de la propiedad de la empresa, y que continúan laborando en ella.

—¿Cuáles han sido los logros de Softtek durante este año?

—Como la gran mayoría de las empresas, iniciamos bastante bien, pese a que en el cierre del año anterior se esperaban turbulencias por el cambio de sexenio.

“Durante el primer trimestre se cerraron algunas ofertas, aunque fue muy complicado. Es algo raro, aparentemente estamos conscientes de lo que necesitamos, pero no sabemos en realidad qué es lo que requiere la compañía para la que trabajamos.

“En enero y febrero se cerraron los proyectos que teníamos del año anterior. El segundo semestre fue el más difícil, ya que tuvimos un récord de propuestas, que tardamos dos meses en conocer lo que el cliente necesitaba, hacerle una auditoría y darle un diagnóstico.

“Y vaya, continuamos trabajando, y la verdad nos ha ido bien”.

6.2. Fabricas de Software / México

TECHNOS Consultores, es una empresa 100% Mexicana, nace en el año de 1999, dedicada a proveer soluciones integrales de Tecnologías de Información en el área de Knowledge Management, Workflow y Administración electrónica de documentos.- Ofrece soluciones integrales de automatización de procesos, utilizando las metodologías, estándares y productos de software.

Fue en ese mismo año 1999 que empezamos la distribución en México del software de Workflow para Web Ultimus Workflow Suite, de Ultimus, Inc, al abrir nuestra fábrica de software, enfocada a ofrecer a los clientes soluciones de software integrales considerando todo un ciclo de desarrollo de software tales como Portales Empresariales, Intranets, extranets y aplicaciones Cliente/Servidor

En el año 2001 expandimos nuestra cobertura al ofrecer servicio y soporte técnico en ciudades como México, Guadalajara y León, logrando con ello un mejor servicio con nuestros clientes foráneos.

TECHNOS Consultores, S.A. de C.V. Cuenta con su Oficina matriz en la ciudad de Monterrey, N.L., En donde se encuentran concentradas las áreas administrativas, la fábrica de software etc.

Referencia:

<http://www.technosconsultores.com/main1.html>

6.3. Sistemas Vertical S.A. de C.V

Sistemas Vertical S.A. de C.V., sociedad anónima de capital variable, fundada en 1994 en México, es una fábrica de software que provee soluciones para mercados verticales de finanzas basadas en plataforma Microsoft.

En 1996 se consolidó como una empresa del grupo Sinapsis (<http://www.sinapsis.com>), y desde ese momento opera como la fábrica de software de Sinapsis.

Su sede es la ciudad de México. En 1998 inició exportaciones hacia los mercados latinoamericanos a través de distribuidores.

Las soluciones que ofrecemos en nuestra Fábrica de Software incluyen:

Vertical Cambios Básico. 5.0

Es un sistema integral que automatiza las funciones, control y monitoreo de operaciones de cambio de divisas y metales para una mesa de cambio en una entidad financiera.

Vertical Arrenda Básico. 3.0

Es un sistema integral que automatiza las funciones y se encarga del control administrativo de todas las áreas de una empresa de leasing.

Vertical Factoraje Básico. 6.0

Es un sistema integral que se encarga de la automatización y control administrativo de todas las áreas de una empresa de factoring

Referencia:

<http://www.microsoft.com/colombia/soluciones/empresas/sistemasvertical.asp>

6.4. HILDEBRANDO/ INFRAESTRUCTURA

Somos una firma de consultoría de Sistemas Computacionales con más de catorce años de experiencia, reconocida por la revista "Expansión" como una de las 500 empresas más importantes de México.

Más de 550 consultores prestan servicio a bancos, instituciones financieras, empresas de telecomunicaciones e industrias de servicio, en múltiples plataformas: Windows NT, Redes, Internet/Web, Sistemas Abiertos, Cliente-Servidor, UNIX, Mainframe/Minis, y Cómputo Móvil.

Los proyectos informáticos presentan retos de diversa índole. Nuestra metodología de gestión de proyectos HOMS® asegura el cumplimiento de los objetivos y compromisos, brindando a nuestros clientes una "administración sin sorpresas".

Contamos con oficinas en las ciudades de México, Guadalajara y Monterrey, así como en Miami, USA y Madrid, España.

Nuestra fábrica de software, ubicada en la Ciudad de México, tiene la capacidad e infraestructura para soportar 450 desarrolladores en una multitud de proyectos.

El equipo de Hildebrando incluye algunos de los desarrolladores más destacados en la industria de la información por su experiencia y conocimiento, especializándose en el desarrollo de sistemas a la medida desde 1986.

NUESTRA MISIÓN

"Ser líderes en el mercado de consultoría y desarrollo de sistemas"

Asumiendo el momento histórico que vivimos siendo promotores del cambio y buscando la excelencia.

Ofreciendo a nuestros clientes un servicio de excelencia soportado en el uso de tecnología de punta.

Buscando el desarrollo de nuestro personal por medio de una constante capacitación profesional y personal.

REFERENCIA:

<http://www.hildebrando.com.mx/infraestructura.htm>

<http://www.hildebrando.com.mx>

7.- Realidades y Perspectivas de la Computación en México / Adolfo Guzmán Arenas

7.1 Es atrevido tratar de medir "procesos de fabricación " en software

El grado de madurez de un proceso o fábrica de software, ¿cómo se mide actualmente?

Si la empresa que produce software sigue ciertas reglas y procedimientos, tiene nivel 5; con algunas fallas, alcanza nivel 4; más fallas y le dan nivel 3. Esto es equivalente a medir el "grado de madurez en el proceso de fabricación" de las bisagras, viendo si se llevan en la fábrica ciertas bitácoras y juntas. De seguro las fábricas organizadas producen bisagras con más dureza que otras menos organizadas, pero (en Física) es más fácil medir directamente la calidad del producto (la bisagra), midiendo su dureza, o su resistencia a los esfuerzos cortantes

En particular resultan prematuros los esfuerzos para medir la "madurez de una empresa para producir software de calidad", porque ni siquiera sabemos cómo medir la calidad (dada por la ergonomía, transportabilidad, rehusó,...) del software. Se parece a quien desee medir la "madurez del proceso de fabricación de mujeres bellas", cuando ni siquiera la "belleza femenina" se sabe medir bien. ¿Mediría la altura de la cama? ¿La temperatura de la alcoba?

Una desventaja del estado actual de la Computación es que cambia mucho. Los avances se suceden rápidamente debido a la corta edad de la disciplina. Estimo que el tiempo de vida medio [□]de un concepto en Computación es de cinco años. Es decir, cada cinco años, la mitad de lo que sabemos se vuelve inútil (aunque no lo olvidemos). Esto requiere : a) que el practicante de la Computación tenga que estar constantemente estudiando, leyendo, aprendiendo, suscrito a revistas, yendo a cursos y congresos; b) que a los estudiantes de licenciatura de Computación se les debe dotar de una fuerte base teórica y matemática, porque deberán construir sobre ella (re-aprender)

El número de personas que saben programas y de aquéllos que usan las computadoras, también ha aumentado drásticamente Hay 110,000 estudiantes de licenciatura en Computación en México; es la ingeniería más popular. Por estas razones quizá es aceptable seguirle llamando "Ciencia de la Computación".

El retraso de México en el área

No debemos preocuparnos mucho si estamos atrasados o no, en la Ciencia de la Computación. Como se vio, todo el mundo es ignorante: la disciplina apenas nació. Debemos prestar más atención al atraso relativo, con respecto a otros países similares al nuestro.

También aunque "está bien" procurar hacer Ciencia de la Computación en México, "es mejor" o "es más conveniente" desarrollar *el arte de la Computación*, aplicaciones, la ingeniería (fabricar cosas útiles con ella), la industria. Desarrollo de paquetes, para autoconsumo, para venderlos en el país para exportar. Pensar en innovaciones (recordar que casi todo no se ha descubierto), mejoras, nuevas aplicaciones (estamos en un mercado no saturado) (más de esto en la sección "*Los segmentos de mercado no saturados*").

Para esto, es necesario entender y dominar esta tecnología (o arte) conocerla a profundidad. Ser líderes. [□]En cambio, ser reactivos (esperar a que otros inventen, para nosotros aprender), traducir os importados (en vez de escribir nuestros), aprender a manejar (en vez de *hacer*) paquetes y

programas de aplicación, no conduce al dominio ni liderazgo (aunque se entiende que no todos van a innovar o a escribir libros, muchos serán usuarios o lectores).

Probablemente, la Ciencia de la Computación se desarrollara a partir de avances artesanales o ingenieriles, preferentemente nuestros, y no de cavilar sobre experiencias de otros y unificarlas o generalizarlas": ¡no hay muchas!

7.2 La ventaja de tener al fabricante (de software) a vuelta de la esquina

Trate usted de pedirle a un fabricante de Utah que haga un cambio sencillo en un programa masivo. Convenza al diseñador que venga a discutir con usted la posibilidad de un cambio a ciertas funcionalidades del producto. Pídale que escriba manuales en español. Si el producto es sofisticado, pídale que sus vendedores entiendan lo que venden, que le ayuden efectivamente cuando usted tiene un problema con el paquete.

Existen ciertas ventajas en tener al fabricante de software a la vuelta de la esquina, en la misma ciudad. Además, un fabricante pequeño responde mejor a las solicitudes del cliente. También, en general, los grupos pequeños de programación son muchos más eficientes que los grupos grandes

¿Está al día México en Computación?

Si, porque podemos adquirir máquinas modernas, baratas, fáciles de usar. A poco de introducirse en el comercio máquinas nuevas, software recién hecho, aparece en México.

No, porque casi todo lo que se vende en México se importa. Nuestro valor agregado es magro.

7.3 Los equipos, dispositivos y periféricos. Su fabricación y venta

Existe gran cantidad de equipo confiable y barato de venta en nuestro país. Mucho de este equipo se importa. Se hace poco en México, aunque se hace algo: IBM fabrica en Guadalajara, AS 400s y Lap Tops. Se ensamblan en México Pcs de algunas marcas (Acer,...).

Paquetes y programas de aplicación. Su fabricación y venta

Existe una gran cantidad de paquetes de venta en nuestro país. Casi todos los paquetes se importan. Hotware de México fabrica un prototipeador rápido, llamado Hotware.

Se exporta muy poco software en paquetes. Esto, en comparación con otros países que se parecen al nuestro en cuanto a su estado de desarrollo: La India exporta anualmente mil millones de dólares en software. Israel, 300. España, Chile, Costa Rica, exportan software. México podría seguir este ejemplo y fabricar software de buena calidad, primero para consumo interno y luego para exportar.

7.4 Software a la medida

Cuando una empresa no encuentra la solución a su problema en un paquete de software, recurre a fabricarlo a la medida, desarrollándolo dentro de la empresa (software de autoconsumo) o mandándolo a fabricar (software a la medida). Por esto, hay en México. una industria de cierta importancia de software a la medida, de software de autoconsumo, de consumo de un sólo cliente. Conviene apoyarse en este mercado para tratar de volver a utilizar este software, generalizándolo a fin de convertirlo en paquete. "*Desarrollar software para las industrias que son fuertes en México*", "*¿Qué hacer con el software hecho a la medida?*", "*¿Qué hacer para sustituir productos importados?*" y "*Buscar nichos de aplicación*")

Integración

El diseñador moderno de software utiliza muchos pedazos grandes, ya hechos por alguien más. Se dice que *integra* esos paquetes en una solución. La industria de programas a la medida (Software a la medida) hace a menudo integración, añadiendo valor agregado (código de enlace, modificaciones, adaptaciones) y produciendo una solución original. Es mucho más barato integrar pedazos grandes de código ya hecho, que hacer todo desde cero.

Outsourcing

Por otra parte cuando una empresa hace **outsourcing** hacia un centro de cómputo en el extranjero, se pierden fuentes importantes de trabajo en computación.

A Profesionales de la computación / *En nivel técnico medio superior*

- a) Es menester fabricar libros y software para enseñanza a este nivel.
- b) Los cursos deben estar de acuerdo con la realidad. por ejemplo, evitar dos materias ya obsoletas (Pascal, cobol). Para esto, conviene pagarles a los profesores competitivamente, para así poder obtener educadores actualizados.
- c) En general, me da la impresión de que en esta área reaccionamos, no somos innovadores.
- d) Hace falta construir programas (de cómputo) educativos (courseware) para este nivel.
- e) Los niveles de salarios de los profesores de cómputo son bajos (puesto que existe u mercado externo que fija sus salarios), y es difícil conseguir profesores con experiencia.

En nivel superior o licenciatura

Es muy importante dar a nivel de licenciatura las bases teóricas (matemáticas, física, electrónica, probabilidades, análisis, numérico, teoría de autómatas, y lenguajes formales) y las bases de la profesión (programación -un par de lenguajes de batalla, y un lenguaje esotérico, mínimo-, programación de sistemas, sistemas operativos, arquitectura de computadoras, bases de datos, sistemas de información, análisis y diseño de sistemas de información; redes, computación distribuida, telecomunicaciones) para que el estudiante pueda seguir aprendiendo una vez que salió de la escuela. Estas bases constituyen el temple de un machete que requerirá unas ocho afiladas en los 40 años de

vida profesional que se espera de un egresado. Es importante repartir las materias de computación durante los ocho o nueve semestres que dura la carrera, y no amontonarlas en los dos últimos años (bajo la presión de un "tronco común" con otras ingenierías): También hay que darles algunas cuantas materias "de moda" o actuales, para que el egresado salga inmediatamente a producción. Es el filo del machete. Termino con los siguientes comentarios:

f) Es importante fabricar libros de computación a nivel de licenciatura. Cada profesor de licenciatura debería estar escribiendo sus apuntes, convirtiéndolos en libros, que tengan en la contraportada un disquete con ejemplos y programas.

g) Hace falta fabricar courseware, tutoriales por computadora, videos interactivos, laboratorios virtuales, buscadores de información..Esto ayudaría a mitigar la falta de profesores con experiencia en hacer software.

h) Los niveles de salarios de los profesores de cómputo son bajos (puesto que existe un mercado externo que fija sus salarios), y es difícil de conseguir profesores con experiencia.

En Posgrado

Se requieren en este nivel libros de texto y monografías. Cada profesor debería escribir su libro.

i) Los niveles de salarios de los profesores de cómputo son bajos puesto que existe un mercado externo que fija sus salarios, y es difícil conseguir profesores con experiencia).

Enseñanza de computación en otras disciplinas; usos particulares

La computación puede utilizarse para ser mejor ingeniero civil, por consiguiente, en Ingeniería Civil conviene enseñar algo de Computación. La computación como auxiliar de otras disciplinas se centra en la enseñanza de paquetes y métodos que son de utilidad en esas otras disciplinas.

La Computación como auxiliar de la Computación; Tecnología de Software

También la Computación puede utilizarse para ser mejor profesional de la Computación. Por ejemplo, utilizando un editor de textos que conozca la sintaxis del lenguaje de programación que estoy escribiendo. La enseñanza en esta Área se lleva a cabo dentro de la educación formal en Computación, y comprende la Programación de Sistemas; herramientas para el Programador o Analista Profesional, compiladores, generadores de interfaces gráficas, intérpretes y evaluadores.

Programación de Sistemas

Aquí entran todos los programas que ayudan a mejor utilizar el equipo de cómputo, excepto los sistemas operativos (que por tradición se ven o se enseñan aparte): ensambladores, cargadores, ligadores, expansores de macros intérpretes, compiladores (también se ven en una materia aparte en licenciatura), y otros programas que mejor administran el equipo.

Mi recomendación es que el profesional de la Computación dedique cierto tiempo a desarrollar sus propias herramientas. Por ejemplo, yo uso un archivo de texto que contiene parámetros de inicialización, de manera que el usuario pueda especificar (con la ayuda de un editor de texto, con cierta dificultad) el color del fondo de sus pantallas, el tipo de letra, etc. Y también le ofrezco un programa en un ambiente gráfico que, al interactuar con el usuario, llena tal archivo (y ahora sí, es fácil de usar).

Herramientas CASE

Estas herramientas ayudan en el proceso de fabricación de software. Por ejemplo, Hotware es un paquete (hecho en México) para diseñar prototipos rápidos. Otro ejemplo es el generador de reportes RepCob (Ver "*Buscar nichos de aplicación*"). Hay documentadores automáticos, machotes de aplicaciones (llamados Wizards en Access), editores de especificaciones, convertidores de diagramas entidad-relación-atributo a definiciones de tablas y vistas, etc.

Muchas herramientas **CASE** proveen aún poco valor agregado, por lo que una herramienta bien diseñada, en conjunto con usuarios, tiene nicho. Existe una gran oportunidad aquí, la sugerencia es trabajar en problemas reales, junto con usuarios (programadores, analistas) que proporcionen opiniones independientes sobre la utilidad de lo desarrollado.

¿Cómo se mide?

La investigación es un trabajo creador que produce cosas originales: nuevos métodos, nuevas tarjetas electrónicas, nuevos conceptos, patentes, teoremas, ecuaciones de estado. No busca que las cosas inventadas sean útiles. Busca que sean *nuevas*, originales. Típicamente, se miden (o se notan) en tesis de grado, publicaciones en revistas de investigación, artículos científicos, etc.

La Computación es una ciencia que acaba de nacer, y se manifiesta como un arte. Por ende, hay poco de investigación, pocas teorías, pocos conceptos nuevos.

Si tiene un grupo que dice hacer investigación, exíjales que publiquen en revistas con arbitraje internacional (casi todo lo nuevo se hace fuera de México), y no tan sólo en congresos o conferencias. Proteja sus inventos e innovaciones con patentes, marcas industriales, etc.

7.5 ¿Qué ocurre en México?

Poca investigación se hace en México en Computación. Hay unos 130 especialistas doctores en el Área. Sin embargo, eso no implica que no podamos hacer trabajos originales de calidad. Quizá haya que inspirarse en los problemas nacionales, pero esa no es una receta necesariamente buena. Hay que evitar el *turismo científico* (es más fácil asistir a congresos, a viajes, invitar y ser invitado, organizar reuniones, crear alguna sociedad nacional de informáticos, que sentarse a hacer ciencia). La mayor parte de la investigación se realiza en universidades del sector público. En algunas, la burocracia es excesiva, los apoyos están fragmentados (un organismo otorga dinero para viajes, el otro paga sobresueldos, el otro subsidia publicaciones; las compras hay que efectuarlas centralmente, bajo concursos y criterios rígidos, etc.) y ocurren un año después de lo solicitado. Esto origina que lo poco que el país invierte en investigación, se desperdicie o malogre, y que los investigadores se

desesperen. La tentación es muy grande para "pasarse al otro lado de la barda" y dedicarse a producir software y aplicaciones y no a investigar.

Un esfuerzo saludable reciente es la creación de la revista *Computación y Sistemas*, de cobertura iberoamericana, que publica artículos originales en inglés, español y portugués. El CIC está apoyando su creación y consolidación.

Investigación Aplicada en Transferencia de Tecnología

La investigación, busca crear cosas nuevas. Se mide en conceptos, teoremas, objetos no antes vistos, nuevos materiales, que a menudo aparecen en publicaciones científicas, congresos, reuniones especializadas, tesis de doctorado. *No es investigación algo que ya se hizo, aún cuando no se haya hecho en México.*

La investigación aplicada busca crear cosas útiles. Generalmente, tiene un cliente o usuario, y generalmente, tiene un valor (se mide en pesos, se paga). *No es investigación aplicada aquel trabajo que no tiene cliente.*

La transferencia de tecnología busca incorporar lo que se acaba de inventar (el "estado del arte", normalmente encontrado en revistas científicas y congresos) al "estado de la práctica", normalmente encontrado en paquetes y aplicaciones comerciales.

El desarrollo profesional consiste en que alguien dedique cierto tiempo a aprender o a hacer ejercicios para dominar algo que él no sabe (la transformada **Z** digamos). No es investigación, ni investigación aplicada, ni transferencia de tecnología.

Investigación aplicada

Cuando inventamos algo (apartado "*Investigación en Computación*"), a menudo queremos aplicarlo. Conversamente, cuando un organismo tiene un problema, busca cómo solucionarlo. El desarrollo de las aplicaciones de una ciencia viene dictado por dos fuerzas: el *empuje* lo de la generación de nuevos conocimientos o métodos y el *tirón* lo de la necesidad de resolver problemas reales.

El criterio principal para desarrollar aplicaciones es que sean útiles, es decir, que haya un usuario que les atribuya cierto valor.

No es investigación aplicada "el hacer una aplicación para ver quien la va a usar después". Esto es hacer "trabajos para ponerlos en la estantería". Tampoco debemos hacer desarrollos que "de seguro alguien nos lo va a comprar", a menos que estemos *muy* seguro de eso.

¿Todos los desarrollos útiles son investigación aplicada?. No todos son desarrollos aplicados, o sea, aplicaciones. Que sean *investigación* aplicada requiere además que sean aplicaciones de cosas relativamente resientes, de descubrimientos o innovaciones nuevas ; en cierto sentido, que las aplicaciones (desarrollos útiles) sean ellas mismas novedosas.

Transferencia de tecnología

Podemos tomar la opción de utilizar los avances científicos que ya se llevaron a cabo en otros lados, y transferirlos (transferencia de tecnología) a productos o aplicaciones que nosotros desarrollemos. No es trivial. Hay que saber qué, de los muchos conocimientos o métodos nuevos, va a ser útil para resolver determinada necesidad o nicho de mercado.

Por lo general, el "cliente" mexicano se presta más a colaboraciones con la academia (en computación). Por lo general, los científicos mexicanos (en computación) son más puros y no se quieren ensuciar las manos, que sus colegas de EE.UU, digamos.

Relación entre la Universidad y la Industria

Esta relación debe estrecharse. Hay cierta desconfianza: el sector productivo no cree mucho en la capacidad de las universidades en resolver problemas prácticos, reales. El sector académico no se acerca a la industria (o comercio o sector gobierno), no le gusta "trabajar en problemas mundanos", no es investigación, no agregan puntos para el Sistema Nacional de Investigadores o para las becas de COFAA. Algunas ideas que pueden estrechar la relación son las siguientes:

a) *Incubadoras de microempresas.* Una manera funcional de apoyar el desarrollo de empresas y fuentes de trabajo, es la creación de incubadores, que apoyan con ciertos subsidios a pequeñas industrias, durante un plazo limitado. El Instituto Politécnico Nacional ya creó una.

b) *Participación del sector productivo en el Consejo de Administración de una universidad.* Esta práctica la llevan a cabo, con éxito, las Universidades Tecnológicas (la de Nezahualcóyotl, por ejemplo).

c) *Estancias en la industria de estudiantes, servicio social, prácticas profesionales.* Bien conducida, puede ser provechosa para la empresa y desde luego, para los estudiantes o practicantes.

d) *Estancias en las universidades y centros de investigación de personal de informática de la empresa o sector público.* Para hacer una especialización, Para desarrollar (él solo, o con colaboración de la institución académica) algún proyecto de interés para su patrón.

c) Que el SNI, COFAA, y otros, reconozcan la investigación aplicada (y la transferencia de tecnología) como una labor útil en el desarrollo de una ciencia. O mejor aún, que se permita a los directores de escuelas y centros de investigación, decidir ellos en que van a trabajar sus investigadores, centralizando en tales directivos los fondos para hacer investigación y desarrollo tecnológico que actualmente se encuentran desperdigados en comités, consejos, sistemas, comisiones, colegios, etc.

Referencia:

<http://www.to2.com/desar/homepage.nsf/to2main?readform&url=/desar/cancomin.nsf/vwall/A100EA0B02A9ADB086256BDC004CC677>

8.- Programa para desarrollar la industria del software

Notimex Junio 23 de 2002.-Ciudad de México.- El gobierno mexicano presentará en fecha próxima el Programa Nacional para el Desarrollo de la Industria del Software, que será parte del sistema e-México, y buscará el desarrollo de industria del software, informaron fuentes del sector.

Para el vicerrector de la Universidad de las Américas del estado central de Puebla, Ricardo Villafana, tener un negocio verdaderamente exitoso y de fábricas de software requiere de un enfoque en la mercadotecnia y no en las posibles ventas que se tengan.

"Si se quiere crear una verdadera industria del software (programas de cómputo), se debe averiguar si en realidad existe un mercado, entenderlo y conocer muy bien que características tiene y en qué producto se interesa", consideró Villasana.

Agregó que desafortunadamente la intención del gobierno mexicano por impulsar el desarrollo de fábricas de software podría perderse en el intento, si no se manejan claramente los puntos principales que se requieren para tener un negocio exitoso.

Estos puntos, aclaró, son la participación gubernamental federal y estatal, el apoyo de la banca, la participación de las empresas proveedoras, compañías que van a ser parte del mercado, y la participación directa de las instituciones financieras.

Para Luis Daniel Soto, director de plataforma .NET de Microsoft México, coincidió en que la construcción de programas de cómputo no es tan sencilla como aparenta y las empresas no nacen de la nada, todo debe ser un conjunto de talentos y tecnología, aseguró.

Soto añadió que por las dificultades que representa desarrollar fábricas de software, el Plan Nacional del gobierno mexicano debe tener un buen planteamiento.

"El Plan Nacional para el Desarrollo de la Industria del Software debe contemplar a todos los jugadores de la industria, para no provocar que se produzca un desequilibrio en toda la economía Nacional", comentó Soto.

Según información de Bermúdez, la inversión que requiere el desarrollo de los parques tecnológicos, es muy difícil de estimar porque depende de los niveles, de costos de gastos de cada país y de de cada región.

"En Brasil, por ejemplo, únicamente la planeación de un parque con todas las funciones de mobiliario, cuesta más o menos un millón de dólares, incluye planear el lugar para establecerse, la inversión, los fondos, pero no incluye la construcción, la cual también varía pues hay algunas muy cara", explicó Bermúdez.

En opinión del vicerrector Villafana nunca se debe perder de vista que el negocio no es fabricar software, sino usarlo de forma inteligente, y recomendó estudiar la demanda existente y cual será el mercado a explotar para maximizar resultados.

"Ahora se tiene que iniciar una educación de la utilización del software, principalmente en las instituciones de educación superior, ya que ahí se encuentra el posible mercado del futuro", dijo Villafana.

Soto manifestó que también se debe aprovechar la cercanía geográfica y la coincidencia de horarios con Estados Unidos, ya que son elementos indispensables para estar en buenas posibilidades de desarrollar este proyecto.

Y agregó que estos motivos son los principales motores que impulsan el interés del gigante tecnológico para que la industria de TI continúe avanzando de manera fluida en el país.

"Sentimos este plan como una oportunidad que se puede hacer real y que hacia dentro de Microsoft, seamos vistos como una de las regiones más exitosas con una mejor práctica en desarrollo", finalizó Soto

Referencia

<http://www.reforma.com/tecnologia/articulo/205655/>

9.- 24 DE JUNIO 2002 / IMPULSAN ESTADOS DESARROLLO DE SOFTWARE

Muestran en la segunda edición del Encuentro Nacional de la Industria del Software proyectos y resultados obtenidos hasta el momento

Puebla, México.- Campeche, Guanajuato y Puebla impulsan ya el desarrollo de fábricas de software para generar una industria basada en el conocimiento, que les permita elevar el desarrollo económico y la generación de empleos en sus entidades.

El principal objetivo de estas iniciativas es formar recursos humanos especializados y competitivos en tecnología de punta, para que se integren a las cadenas productivas y así crear las bases del desarrollo de la industria de software.

Tiene Guanajuato resultados

La primera iniciativa fue la de Guanajuato, que implementó un Programa de Fomento a la Industria de Software en 1999, ejemplo que ahora es seguido por Puebla y Campeche.

Durante el segundo Encuentro Nacional de la Industria del Software, Arturo Lara, director general del Consejo de Ciencia y Tecnología del Estado de Guanajuato (Concyteg), explicó que la misión del programa es impulsar una política de Estado que permita aprovechar las oportunidades de la industria de tecnologías de información y las del desarrollo de software, así como establecer las bases para ser una plataforma de empresas y un desarrollador de la industria de software.

Para ello, la estrategia fue negociar con las empresas locales para establecer células de desarrollo de software, con sede en las compañías o en las instituciones educativas, para generar proyectos reales.

"Las células tienen poca rentabilidad, sin embargo es una inversión que se tiene que hacer para tener la capacitación de los problemas reales que enfrentan las empresas", señaló Lara.

Entre los logros de Guanajuato se encuentran tener 11 instituciones académicas involucradas en el programa, la formación de más de 40 profesores y 110 alumnos capacitados.

"A los alumnos les ofrece una oportunidad para tener mejores empleos y obtener una mejor calidad de vida, ya no tienen los autos 'solo en la cabeza', sino que ya llegan en ellos", enfatizó.

Otros resultados son el contar con una curricula adecuada y adaptada a las necesidades de la industria, haber creado ocho células de desarrollo de software y la generación de una cultura del software en el estado.

Roberto Solís, director general de Gmatrix, destacó que la visión en Guanajuato a cinco años es formar recursos humanos, madurar la generación de talento y la promoción de postgrados, incubar empresas y crear parques tecnológicos.

No obstante, dijo, la generación de talento es una condición indispensable, pero no suficiente para detonar la industria del desarrollo del software, sino que se requiere de una política de Estado para

posicionar a la región en el mapa tecnológico mundial.

Busca Puebla Generar Industria

En el caso de Puebla, el estado busca generar una gran industria de software en el 2005 con productos de calidad mundial, aseguró Eugenio Miranda, subsecretario de la Secretaría de Educación Puebla.

Esta iniciativa surgió en el 2001 al impulsar el proyecto "Fomento a la industria de software en el Estado de Puebla" (Fisep) en cada uno de los institutos tecnológicos superiores descentralizados, federales y universidades tecnológicas, la Benemérita Universidad Autónoma de Puebla y en el Instituto Nacional de Astrofísica, Óptica y Electrónica.

Miranda señaló que como base del proyecto se requiere de jóvenes egresados con dominio de tecnologías y alta demanda comercial.

Sin embargo, también se requiere atraer inversión directa nacional e internacional para establecer fábricas de desarrollo de software y la incubación de nuevas empresas, para elevar el desarrollo económico y la generación de empleos.

"La planta productiva del estado no genera el mismo número de empleos que el de los egresados. Buscamos que estos egresados generen sus propias empresas", señaló.

El funcionario indicó que 10 por ciento de los desarrolladores se encuentra en las ciudades como Puebla, San Martín Texmelucan y Tehuacán, sin embargo el 90 por ciento está en la Sierra.

"Es en la montaña donde creemos que están los mejores desarrolladores y no en las grandes ciudades, ahí no hay mayor diversión, lo que los hace demasiado creativos cuando tienen un aula con computadoras conectadas a Internet", enfatizó.

Miranda indicó que han invertido en el programa 120 millones de pesos, de los cuales 25 millones provienen de donaciones de software por Microsoft, un millón en el pago de licenciamiento en instituciones pequeñas y 3 millones de pesos en el licenciamiento de la Benemérita Universidad Autónoma.

"Hemos invertido alrededor de 50 millones de pesos en nuevos centros de cómputo y les hemos hecho llegar el Internet vía satelital o telefónica", señaló.

Equiparan Tecnología y Turismo

Campeche determinó al software un sector estratégico para su desarrollo a largo plazo en su plan 2002, al igual que el sector turístico.

Fabio Aranda, director de proyecto en Campeche, explicó que el objetivo de crear un programa de fomento a la industria del software en el estado es formar, para marzo del 2003, más de 400 alumnos que puedan iniciar su vinculación con la industria de desarrollo de software regional y nacional.

Aranda destacó que la problemática de su estado es que se le ubica como el "gigante dormido", porque tienen atraso y distanciamiento con las tecnologías, esto a diferencia de estados circunvecinos.

"Registramos una fuga muy fuerte de estudiantes por no contar con capacitación en tecnología, ahora las cuatro instituciones de educación superior del estado contemplan las carreras en nuevas tecnologías", señaló.

Agregó que la meta es que las fábricas de software generen una industria que engrandezca al estado.

10.- 16 Junio 2002 / Su meta: acelerar industria del software

Un paquete completo de estímulos fiscales incluirá el programa de apoyo a esta industria que será lanzado en unas semanas

Puebla, México.- Impulsar la creación de una industria del software con valor de 5 mil millones de dólares anuales para el 2010 y situar al México como líder del sector en Latinoamérica en el 2006, son las metas que se plantea el Gobierno Federal con el Programa Nacional de Apoyo para el Desarrollo de la Industria del Software.

Luis Ernesto Derbez, secretario de economía, señaló que con el programa, que será anunciado en unas semanas, el Gobierno busca generar un paquete completo de estímulos fiscales para apoyar a la industria del software, que permita a las empresas incrementar su producción y mejorar su nivel de calidad.

Durante el Segundo Encuentro Nacional de la Industria del Software, realizado en la capital de Puebla, el secretario indicó que con estas acciones se crearán las condiciones que permitan contar con el desarrollo de una industria del software competitiva internacionalmente y asegurar su crecimiento a mediano y largo plazo, aprovechando la posición geográfica, el perfil demográfico y el estado de desarrollo tecnológico del país.

"La industria del software en México es incipiente, participa con sólo el 0.14 por ciento del producto interno bruto... Se estima que hay 390 empresas de desarrollo de software", manifestó el funcionario.

Aún no hay desarrollo

Rocío Ruiz, subsecretaria de comercio de la Secretaría de Economía, explicó que entre los objetivos del nuevo programa están el lograr una producción de software por 5 mil millones de dólares anuales para el 2010, el alcanzar el promedio mundial de gasto en tecnologías de información (TI), y el convertir al país en el centro latinoamericano de soporte y desarrollo de servicios basados en TI.

México, agregó, no ha desarrollado una industria del software importante, porque se le dedica muy pocos recursos, además las empresas que existen son muy pequeñas y no existe ninguna empresa con alto nivel de la calidad suficiente para competir internacionalmente.

Tan sólo el gasto en tecnologías de la información y comunicaciones en el país representa el 3.2 por ciento del PIB, lo cual sitúa al país en el número 50 a nivel mundial, por lo que se tiene la meta de llegar al promedio mundial de 7.5 por ciento del PIB. En el ámbito de software, se tiene la meta de llegar al 0.61 por ciento del PIB, dijo la funcionaria.

"Tenemos buena oportunidad y podemos desarrollarnos como lo hizo India e Irlanda, a través de la capacitación de la gente. Es una industria que usa cerebro de obra y es importante que tengamos programas de educación acordes de las necesidades de la industria", expresó Ruiz.

Por ello, dijo, trabajan en un programa nacional donde no sólo interviene la Secretaría de Economía, sino Bancomext, INEGI, Nafin, secretaria de educación y con asociaciones privadas como Amiti y Amece, para que sea una acción que apoye los requerimientos de la industria.

¿En qué consiste el programa?

El programa contemplará siete líneas de acción a seguir. La primera es promover las exportaciones y la atracción de inversiones; la segunda es contar con educación y formación de personal competente en el desarrollo del software, en cantidad y calidad convenientes.

Como tercer lineamiento está con el contar con un marco regulatorio promotor de la industria. Para ello, dijo Ruiz, buscan adecuar la legislación en materia de propiedad industrial y piratería de software.

"Hay que avanzar la legislación para que no sea un tope en la digitalización de procesos. Los códigos de comercios locales y federales tienen que tener las características para que procesos electrónicos tengan la misma validez en un juicio, que si presentamos el papelito", enfatizó la subsecretaria de comercio interior.

En el cuarto punto se busca desarrollar el mercado interno, mediante la integración digital de cadenas de suministro, creando una demanda de trámites y servicios gubernamentales.

"En el gobierno y empresas sin papeles (se busca que) el software signifique un ahorro en dinero a las industrias y no sea una inversión, sino que eficiente los procesos en las empresas", agregó.

Otro lineamiento será fortalecer a la industria local. Aquí, dijo, pensando en el desarrollo de la industria nacional se apoyará el software libre, fortalecerlo mediante financiamiento para capital de trabajo, capacitación y certificación de sus sistemas de calidad de procesos.

Como sexto punto está el alcanzar niveles internacionales en capacidad de procesos.

"Trabajaremos en impulsar una entidad local de certificación, fomentar a la investigación y desarrollo, así como crear el Premio Nacional de Calidad en Tecnologías de Información", señaló.

Y por último está el promover la construcción de infraestructura física y de telecomunicaciones.

El documento, dijo, incluye un diagnóstico de la industria, hacia dónde va y cómo se necesita.

Manos a la obra

Ruiz indicó que a pesar de que todavía no se ha anunciado formalmente ya están trabajando en ciertas líneas de acción hacia una industria de tecnología en México, y un ejemplo es que en Nuevo León se trabaja cercanamente con un parque tecnológico que estará en la ciudad de Monterrey.

"Este parque, a diferencia de otros está dentro de la ciudad, cerca de un campo de golf, universidades y zonas residenciales altas. Con un enfoque más de universidades y desarrollos de investigación", señaló.

Así, dijo, es como hay que pensar en estos parques, porque no se pueden pensar fuera del núcleo de la ciudad, porque trabajarán en ellos ingenieros de alta tecnología de cierto nivel.

Añadió otro parque en el que se trabaja está en Jalisco, en el cual se están pensando enfocar al cluster que tienen ellos en el Valle del Silicio Mexicano, es decir que buscan estar muy de la mano con el desarrollo de los productos.

Adicionalmente, dijo, otras acciones que realizan es invertir en digitalización de cadenas comerciales, para lo cual desarrollan software para pequeños comercios de abarrotes y los enseñan a trabajar en él.

"Generamos muestras pilotos exitosos en diversos municipios que se irán pasando a otras cadenas, para ir reconvirtiendo al comercio a la industria de los servicios. También queremos trabajar con un municipio para hacer de forma digital los pagos de sus servicios y convertirlo en programa piloto exitoso", indicó Ruiz.]

11.- Publicaciones / Desarrollan en México fábricas de software / 25 de abril del 2000

Invertirán 18 mdd en capacitación.

Por: Novedades

Invertirán 18 mdd en capacitación.

Crearán una plataforma sólida de técnicos especialistas.

Líderes empresariales de la informática promueven el desarrollo de un complejo industrial del Software en México, con una inversión de unos 18 millones de dólares sólo en capacitación.

Roberto Solís, líder programa “Fomento al Desarrollo de la Industria del Software”, dijo que éste es auspiciado desde hace un año por empresas desarrolladas nacionales y el corporativo mexicano de Microsoft.

“Se trata de crear una plataforma sólida de técnicos especialistas que aliente a los inversionistas a realizar sus desarrollos en cómputo dentro del el país”, explicó Eva María García, ejecutiva de Microsoft.

Para Rafael Bernal, miembro de la asociación Mexicana de la Industria de las Tecnologías de la Información (AMITI), las fábricas de software tienen alta rentabilidad y su principal soporte es el capital humano.

Al respecto Solís expuso que en Estados Unidos un curso de capacitación igual al que Microsoft ofrece en el marco del programa tiene un costo para los estudiantes de 60 mil dólares.

Para la AMITI, la promoción de fábricas de software en México está en sus prioridades ya que es una de las actividades de la Industria que mayor impacto económico tiene en los países que lo albergan.

“Es un segmento con una enorme capacidad de generación de empleos altamente remunerados y con un excelente factor multiplicador, lo que hace crecer la economía de los lugares que promueven este negocio” dijo Bernal.

Estas fábricas, agregó, tienen además la ventaja de ser “industria verde”, pues no generan desechos industriales y dan a pequeñas empresas la posibilidad de participar con productos terminados de gran valor agregado.

Reconocido como el maquilador por excelencia de software, ha sabido explotar este negocio y está exportando hasta cuatro mil millones de dólares al año por este concepto. Pero la enorme demanda para la maquila de productos de cómputo y el déficit mundial de profesional técnicos está agotando su capacidad de servicios y elevando los costos del trabajo de sus programadores.

El sueldo promedio de un programador en la India es de 500 dólares mensuales, estimaron los especialistas.

Según Bernal, “en México, el terreno de los desarrolladores de software está poco explorado y existen muy pocas industrias abocadas a esta tarea”. “Los más conocidos son Softtek y Serinf, también existen fábricas de software que “son privadas”, esto es, que la tiene una empresa para sus propios desarrollos, como Ddémesis, adquirida por GE Capital, y Tecnosys, de IBM”, agregó.

“Entendemos que hay intentos de ir al exterior de parte de otras compañías de cierto tamaño y hay algunos casos de pequeñas compañías que están haciendo la posible a través de contactos personales” apuntó.

La consultora Select, filial de International Data Corporation, considera que la actividad de las empresas desarrolladas de software en México económicamente es aún incipiente.

El software a la medida carece de demanda importante entre las empresas mexicanas ya que sus costos son elevados, dijo Alejandro Florean Gerente de software e Internet Working de Select.

“Pese a ello, durante 1999 generaron importantes utilidades estas se debieron a las necesidades que tenían las empresas de adecuar sus sistemas al año 2000”.

Según las proyecciones de Bernal, una fábrica de software con unos tres mil empleados podría ser rentable en dos o tres años con una inversión de los 10 millones de dólares.

El programa de Fomento de Desarrollo de la Industria de Software está dirigido a la capacitación de estudiantes de escuelas públicas de educación tecnológica del central estado mexicano de Guanajuato.

Contempla en su fase inicial la capacitación de 300 estudiantes como desarrolladores de informática y pretende promover la inversión directa de capitales en el estado con su consecuente generación de empleos.

A un año de haberse instrumentado, el programa cuenta con cinco células de desarrollo de software financiadas por industriales, donde los estudiantes se desenvuelven sobre proyectos reales.

Referencia:

<http://clubs.terra.com.mx/cienciaytecnologia/articulos/InvMexcyt/4775/>

<http://www.vermic.com/productos.htm>

<http://www.cem.itesm.mx/asociaciones/aesc/symposium2.html>

11.1. Información del X Symposium Internacional de Sistemas Computacionales realizado por la Asociación de Estudiantes de Sistemas Computacionales

Empresa: Aztec Tech Games

Ponente: Ing. Mario Rodríguez

Puesto: Director General

Conferencia: Modelos 3D: del papel a la batalla

Resumen: Aztec Tech Games es una empresa 100% mexicana que está en el proceso de elaboración de su primer video juego con presencia internacional. La conferencia trata de cómo llevar a un personaje del boceto en papel hasta su montaje final en el juego.

Empresa: Softtek

Ponente: Ing. Roberto Montelongo

Conferencia: Fábricas de Software

Resumen: Softtek es una empresa ampliamente reconocida, especialmente por ser una de las empresas más fuertes en la fabricación de software. En esta conferencia nos darán una breve reseña de lo que hace una empresa como fábrica de software y sus necesidades.

Referencia:

<http://www.cem.itesm.mx/asociaciones/aesc/symposium2.html>

12.- Conclusiones

Después de haber realizado el presente trabajo de investigación podemos llegar a las siguientes conclusiones:

- La implementación de una fabrica de software, solo es posible cuando se logran completar con éxito las capas superiores de los diferentes modelos como CMM, Triullum, Spice entre otros.
- En México podemos desarrollar paquetes, Buscar la generalización del producto y posteriormente buscar mercados internos y luego, externos (exportar).
- Con la implementación de metodologías para el uso de Fabricas de Software, se puede lograr:
 1. Reducción de Costos.
 2. Medios efectivos para controlar recursos económicos, laborales y organizacionales.
 3. Estandarización de políticas y procedimientos dentro de una organización.
- En México todavía no existe una cultura de organización y sobretodo de visualización de problemas por procesos.
- En México las pocas aplicaciones y esfuerzos encaminados a llevar a cabo fabricas de software, están basado solo en productos de Microsoft.
- Otros países similares al nuestro lo han demostrado: Israel, India, Chile, España, Costa Rica
- A mediano y largo plazo, la posibilidad de independizarse de los desarrollos y productos basados en Microsoft, se lograría mediante campañas mercado lógicas que promuevan el uso de otro tipo de aplicaciones como Linux, Uníx, Mac entre otros.
- También sería conveniente que la industria mexicana y el sector público, creyera más en si misma, por otro lado sería conveniente que viéramos qué desarrollar (cosas útiles) para la industria, los negocios, el gobierno
- Visualizando el futuro inmediato, con el crecimiento de países como China, India y Pakistan en el mundo informático, el Gobierno Mexicano deberá prestar atención a este tipo de esfuerzos y crear organismos descentralizados que guíen a las empresas Mexicana.
- Debemos concentrar en los que dirigen centros de investigación, el dinero para apoyar la investigación que se encuentra fragmentado, en comités, comisiones, consejos, y finalmente, establecer incubadoras para brindar apoyo a micro-empresas
- Se pudo observar la carencia de herramientas, políticas, procedimientos y metodologías en algunas organizaciones y empresas mexicanas.

- La "Métrica" es una rama de la computación (dentro de "Ingeniería de Programas" o Ingeniería de Software) que estudia cómo hacer mediciones, hay libros, reuniones y congresos. Desgraciadamente, esto no la convierte en ciencia.- Esta medición se inventó en los Estado Unidos y hay un Instituto (Software Engineering Institute) que la practica como negocio. Se está tratando de introducir este tipo de medición (y de negocio) a México.
- Finalmente se recomienda buscar primero al cliente y luego hacer el producto, para no hacer productos para la vitrina de exhibición y porque el cliente sabe lo que necesita, nosotros no.