
Capítulo 1

Fundamentos de
MOPROSOFT

1.1 Qué es MOPROSOFT

Con el avance en la tecnología digital, el software se ha vuelto a su vez más complejo, más poderoso y, por tanto, con más probabilidad de fracaso. Por esto, el proceso de desarrollo de software debe seguir ciertas reglas que nos permitan generar un producto que cumpla con todos sus requerimientos de manera confiable y al mínimo costo.

La ingeniería de software se define como la disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos de software que son desarrollados y modificados en tiempo y dentro del presupuesto definido. La ingeniería de software difiere de la programación tradicional en que se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecido para el proyecto; además esta ingeniería se preocupa por aspectos administrativos que quedan fuera del dominio normal de programación.¹

La ingeniería de software, definida por la IEEE como "la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software"², presenta a los desarrolladores de software las técnicas necesarias para reducir la posibilidad de fracaso en el producto software.

La ingeniería de software es una tecnología multicapa, la cual contiene elementos que pueden agruparse en varias capas.

En la capa básica se encuentran los componentes que permiten dar un enfoque de calidad al desarrollo completo, sobre éstos se encuentran los procesos, es decir, las guías que establecen los pasos probados para la generación de los productos que permiten la construcción del software.

El fundamento de la ingeniería de software es la capa del proceso. El proceso de la ingeniería de software es la unión que mantiene juntas las capas de tecnología y que permiten un desarrollo racional y oportuno de la ingeniería de software.

El proceso define un marco de trabajo para un conjunto de áreas clave del proceso que se deben establecer para la entrega de la tecnología de la ingeniería de software.

¹ Fairley, Richard. *Ingeniería de software*; McGraw-Hill. México, 1987, pág. 37.

² IEEE. Estándar IEEE 610.12

Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento.

Las herramientas de la ingeniería de software proporcionan un enfoque automático o semiautomático para el proceso y para los métodos.

El enfoque en capas, se ilustra en la figura 1.

- Herramientas. Proporcionan un soporte automático a los procesos y a los métodos.
- Métodos. Indican cómo construir técnicamente el software.
- Procesos. Son el fundamento de la ingeniería de software, junta los métodos y las herramientas.
- Un enfoque de calidad. Son la base o conocimientos de la ingeniería de software.

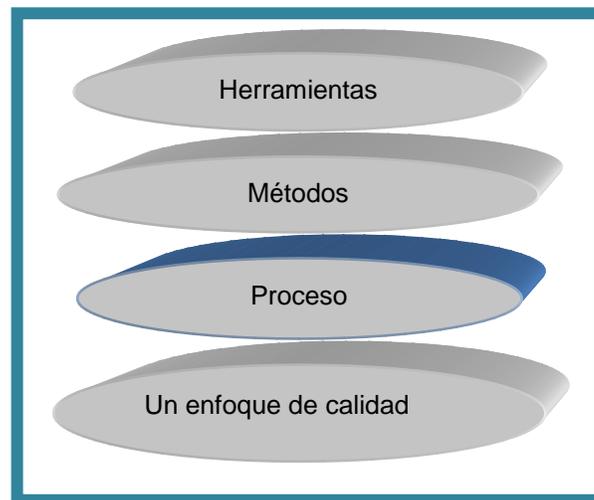


Figura 1. Capas en ingeniería de software.

Estos procesos pueden aplicarse siguiendo métodos específicos, los cuales dependen del tipo de software a desarrollarse. Para poder llevar a cabo todo esto, se utilizan herramientas que incluyen lenguajes de programación, software para controlar configuración, para controlar el proyecto, etc.

La teoría que apoya los componentes de cada una de estas capas es muy amplia; por lo que en este trabajo de tesis nos concentraremos únicamente en la capa de procesos.

Los procesos bien implementados llevan al éxito en el desarrollo de software y hablaremos de un modelo conocido como MOPROSOFT, que ha sido diseñado para la pequeña y mediana industria mexicana, permitiendo a las empresas mexicanas obtener una validación oficial del nivel de capacidad de madurez con que cuentan sus procesos.

El proceso para desarrollar software normalmente contiene las actividades de comunicación, planeación, modelado, construcción y arranque. Estas actividades pueden ejecutarse concurrentemente y su importancia dependerá de la fase de avance del proyecto de desarrollo de software.

Existen varios modelos del ciclo de vida de desarrollo de software que indican la secuencia en que el proceso se puede llevar a cabo. Sin importar cuál sea el tipo de ciclo de vida que se use, la calidad del sistema construido estará altamente influenciada por la calidad de los procesos usados para adquirir sus insumos, desarrollarlo y mantenerlo.

Un proceso se define como un conjunto de prácticas relacionadas entre sí, llevadas a cabo a través de roles, que utilizando recursos e insumos producen un satisfactorio de negocio para el cliente.

Cada equipo de trabajo adopta un proceso para desarrollar software, ya sea de manera consciente y estructurada, o inconsciente y resultado de la práctica diaria, para asegurar que estas actividades consigan sus objetivos.

Los modelos de procesos son guías que presentan las mejores prácticas para el desarrollo del producto en cuestión. Su propósito es guiar a las organizaciones en la selección de estrategias de mejora, determinando la madurez del proceso actual para identificar puntos importantes a atacar y mejorar tanto el proceso como la calidad de software.

La capacidad de un proceso se define como la medida estadística utilizada para evaluar cuantitativamente la posibilidad de que un proceso genere los productos o servicios para los cuales fue diseñado.

Por madurez de un proceso se entiende el avance que ha alcanzado el proceso en su capacidad para conseguir aquello para lo cual han sido creados, esto es, el proceso cumple totalmente con su objetivo.

La aplicación de modelos de procesos eventualmente lleva a la mejora de la calidad, debido a que se aumenta la capacidad de los procesos.

Es importante implementar modelos de procesos que han sido probados y que están basados en estándares de calidad de software, por mencionar algunos:

ISO 9001:2000, SW CMM, ISO /IEC TR 15504.

Esto porque:

- La implementación ordenada y sistemática nos permite evaluar cuantitativamente el nivel de madurez de los procesos de una organización, lo que permitirá saber dónde está e ir midiendo su avance en la eficiencia de producción.
- Las empresas que aplican modelos de procesos estándares tienen una mayor oportunidad de conseguir niveles altos de competencia internacional.

Por otro lado, los modelos de mejora de procesos crean conciencia y responsabilidad de los errores en el equipo de desarrollo, ayudan a conseguir la satisfacción de cliente, pues permiten entregar productos de calidad y a tiempo, reduciendo los costos de la producción.

Implementar un modelo de proceso significa enseñar a una organización desarrolladora de software, la manera en que el modelo debe usarse, adecuar el modelo a las necesidades particulares del negocio, probar su efectividad utilizándolo en la práctica diaria y posteriormente, descubrir las adecuaciones requeridas que permitirán una mejora continua en el desarrollo del software.

Un modelo se define como la descripción de un conjunto de reglas y criterios para definir un proceso o procedimiento que se utiliza para la realización de un servicio o producto.

Al hablar de implementación de modelos de procesos nos hacemos la siguiente pregunta:

¿Es posible aplicar modelos de procesos a una empresa desarrolladora de software altamente inmadura? La respuesta es sí. Sin embargo esta implementación debe llevarse a cabo siguiendo un orden y de manera progresiva. No es posible pasar del caos al estado óptimo en un solo paso.

Por otro lado y de manera muy importante es necesario tener en cuenta que los modelos deben ajustarse a la realidad de la cultura de la empresa en cuestión. Por esta razón, muchas veces los modelos internacionales no pueden aplicarse directamente en empresas latinoamericanas, sino que deben ajustarse a factores característicos de éstas, tales como tamaño, educación del equipo de trabajo y disponibilidad de alta tecnología y otros recursos.

Para la evaluación de un producto de software existen distintos modelos que manejan varios atributos que tienen que ver con la calidad de un producto.

A continuación se mencionarán algunos modelos importantes para la evaluación de la calidad del producto de software, de los cuales podemos destacar el modelo espiral, cascada y Boehm por ser los más utilizados y los más completos.

1.2 Metodología de desarrollo de software

Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Cada metodología de desarrollo de software tiene más o menos su propio enfoque para el desarrollo de software.

Por mencionar algunos:

- Modelo en cascada
- Espiral

1.3 Modelo en cascada

Los modelos prescriptivos de procesos se propusieron originalmente para ordenar el caos del desarrollo de software, estos modelos han traído consigo cierta cantidad de estructuras útiles para el trabajo en la ingeniería de software. Ver figura 2.

Cualquier organización de ingeniería de software debe describir un conjunto único de actividades dentro del marco de trabajo para los procesos de software que adopte.

También debe llenar cada actividad del marco de trabajo con un conjunto de acciones de ingeniería de software y definir cada acción en cuanto a un conjunto de tareas que identifique el trabajo que deben completarse para alcanzar las metas de desarrollo. Después la organización debe adaptar el modelo y ajustarlo a la naturaleza específica de cada proyecto, a las personas que lo realizarán y el ambiente en el que se ejecutará el trabajo.

“El modelo en cascada o ciclo de vida clásico, sugiere un enfoque sistemático, secuencial hacia el desarrollo del software, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del software terminado enfoque metodológico que ordena las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.”³

Estas cinco actividades son útiles durante el desarrollo de programas pequeños, la creación de grandes aplicaciones en la red, y en la ingeniería de sistemas basados en computadoras grandes y complejas.

Existen variantes de este modelo; especialmente destacamos la que hace uso de prototipos y en la que se establece un ciclo antes de llegar a la fase de mantenimiento, verificando que el sistema final esté libre de fallos.

En la vida real, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.

³ Pressman S. Roger; *Ingeniería de software*; McGraw-Hill, 6ta. edición. México, 2005, pág. 50.

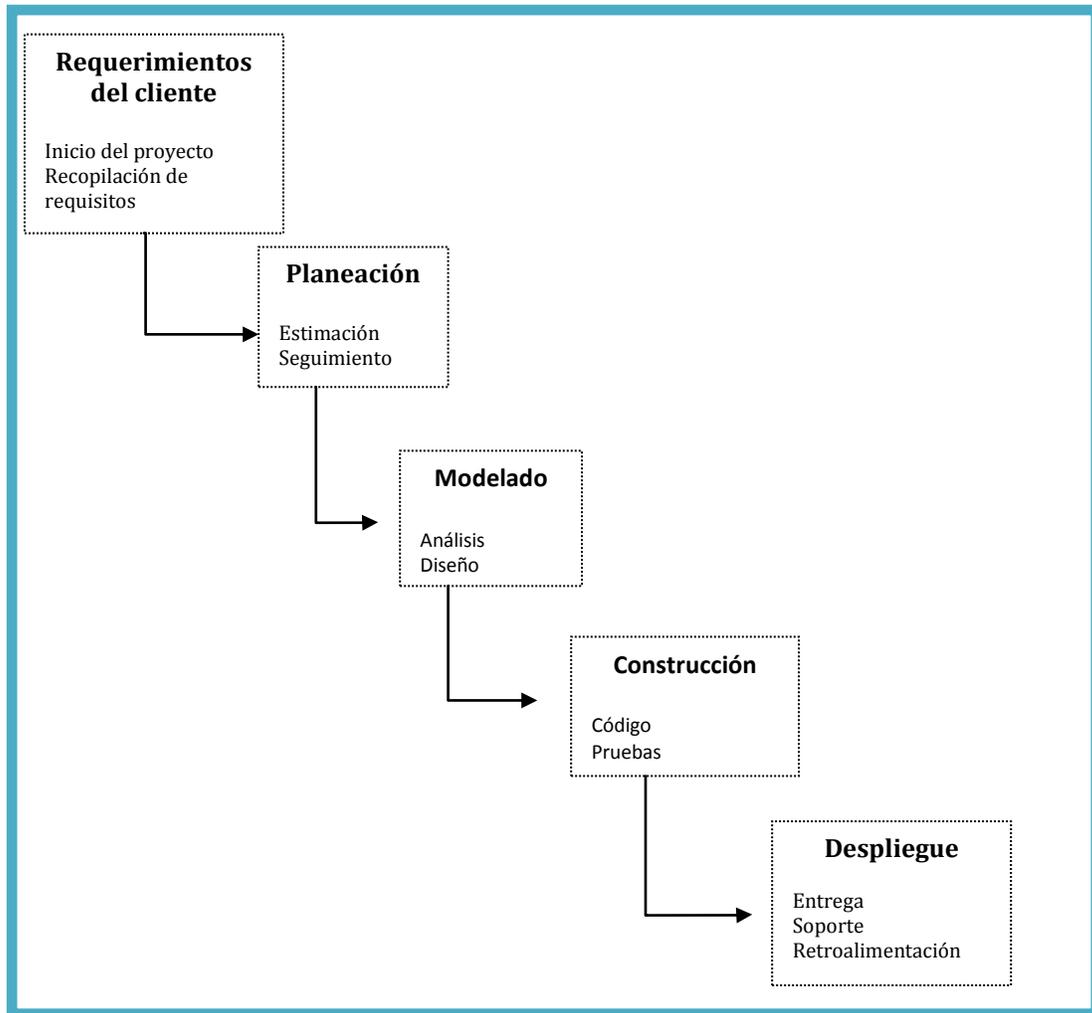


Figura 2. Modelo en cascada.

El proceso de creación del software tarda mucho tiempo, ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera. Esto es la base para que funcione bien.

1.4 Modelo en espiral

Es un modelo de tipo incremental, el cual combina elementos del modelo en cascada aplicado en forma iterativa.

Cuando se aplica el modelo en espiral, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, la entrega tal vez sea un documento del modelo o un prototipo. En las últimas iteraciones se producen versiones cada vez más completas del sistema desarrollado.

“El modelo en espiral es un enfoque realista para el desarrollo de software y de sistemas a gran escala. Como el software evoluciona conforme avanza el proceso, el desarrollador y el cliente entienden y reaccionan de la mejor manera ante los riesgos en cada etapa evolutiva.”⁴

Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Ver figura 3.

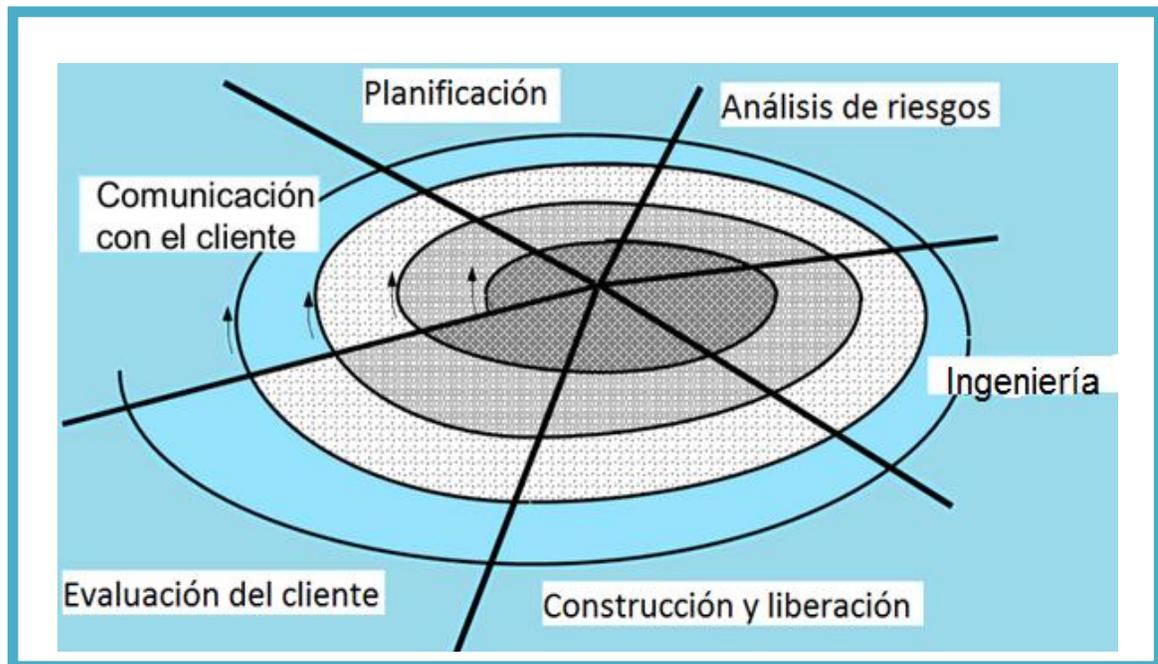


Figura 3. Modelo en espiral.

⁴ Pressman S. Roger; *Ingeniería de Software*; McGraw-Hill, 6ta. edición. México, 2005, pág. 58.

El modelo en espiral se divide en un número de actividades estructurales, también llamadas regiones de tareas.

Generalmente, existen seis regiones de tareas que se describen a continuación.

- Comunicación con el cliente. Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- Planificación. Las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.
- Análisis de riesgos. Las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.
- Ingeniería. Las tareas requeridas para construir una o más representaciones de la aplicación.
- Construcción y liberación. Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- Evaluación del cliente. Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software, creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.

El modelo en espiral, que propuso Boehm originalmente, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con aspectos controlados y sistemáticos del modelo en cascada.

La calidad del software es una compleja combinación de factores que varían entre diferentes aplicaciones y los distintos clientes que las solicitan.

A continuación se describen dos modelos que identifican los factores que afectan la calidad del software entre ellos el de Boehm y McCall.

1.5 Modelo de Boehm

El modelo presenta una jerarquía de características donde cada una de ellas contribuye a la calidad global.

Se centra en:

- Sus características operativas
- Su capacidad para soportar los cambios
- Su adaptabilidad a nuevos entornos
- La evaluación del desempeño del hardware

El modelo comienza con la utilidad general del software, afirmando que el software es útil, evitando pérdida de tiempo y dinero. Ver figura 4.

La utilidad puede considerarse en correspondencia a los tipos de usuarios que quedan involucrados. El primer tipo de usuarios queda satisfecho si el sistema hace lo que pretende que haga; el segundo tipo es aquel que utiliza el sistema luego de una actualización y el tercero, es el programador que mantiene el sistema.

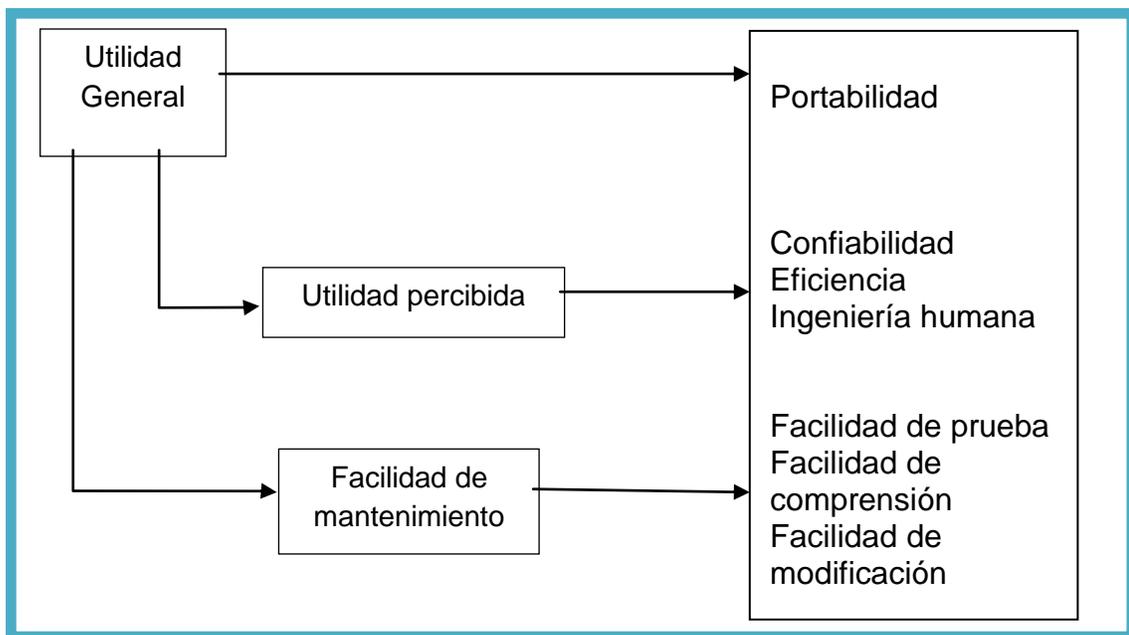


Figura 4. Características del modelo de Boehm.

“Otro aspecto muy importante para medir la calidad de un sistema consiste en establecer un esquema de graduación que cubra los siguientes factores.

- Facilidad de auditoría. Facilidad con que se puede comparar con los estándares establecidos.
- Exactitud. La precisión de los cálculos y el control del programa.
- Completitud. El grado en el que se ha conseguido la total implementación de las funciones requeridas.
- Concisión. Lo compacto que es el programa en términos de líneas de código.
- Consistencia. El uso de un diseño uniforme y de técnicas de documentación a lo largo del desarrollo del proyecto.
- Estandarización de los dato. El uso de estructuras de datos y de tipos de estándar a lo largo del programa.
- Tolerancia a errores. El daño que se produce cuando el programa encuentra un error.
- Eficiencia en la ejecución. El rendimiento en tiempo de ejecución de un programa.
- Modularidad. La independencia funcional de los componentes del programa.
- Facilidad de operación. Facilidad para operar el programa.
- Seguridad. La disponibilidad de mecanismos para controlar o proteger los programas y la información.
- Simplicidad. El grado en que un programa puede ser entendido sin dificultad.

- Facilidad de seguimiento. La posibilidad de seguir paso a paso las instrucciones del programa hacía atrás, hasta llegar a los requerimientos.”⁵

1.6 Modelo de McCall

Los factores se centran en tres aspectos importantes de un producto de software.

- Operación del producto. Sus características operativas.
- Revisión del producto. Su capacidad para soportar los cambios.
- Transición del producto. Su adaptabilidad a nuevos entornos.

Se toma como referencia los factores que afectan la calidad del software indicados en la figura 5.

- Facilidad de uso. Es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y de salida.
- Integridad. Es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.
- Corrección. Mide el grado en que un programa cumple con sus especificaciones y consigue los objetivos del usuario.
- Confiabilidad. Mide el grado en que se puede esperar que un programa lleve a cabo su función con la precisión requerida.
- Eficiencia. Mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones especificadas. “Grado con el que un producto de programación efectúa sus funciones, mediante un mínimo de recursos computacionales.” ⁶
- Facilidad de mantenimiento. Es el esfuerzo requerido para localizar y arreglar programas.

⁵ Zaldívar Zamorategui Orlando. *Apuntes Ingeniería de programación*; UNAM México, 2008, págs. 173-174.

⁶ Fairley, Richard. *Ingeniería de software*; McGraw-Hill, 6ta. edición. México, 1987, pág. 37.

- Facilidad de prueba. Es el esfuerzo requerido para probar un programa.
- Flexibilidad. Es el esfuerzo requerido para modificar un sistema operativo.
- Reusabilidad. Es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.
- Interoperación. Es el esfuerzo requerido para asociar un programa a otro sistema.
- Portabilidad. Es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistema a otro. Facilidad con la que un producto de programación puede ser transferido de un sistema de cómputo a otro o de un ambiente a otro.

	Facilidad de uso	¿Puedo ejecutarlo?
	Integridad	¿Es seguro?
Operación del producto	Corrección	¿Hace el software lo que yo quiero?
	Confiabilidad	¿Lo hace de forma exacta todo el tiempo?
	Eficiencia	¿Se ejecutará sobre mí HW lo mejor posible?
	Facilidad de mantenimiento	¿Puedo arreglarlo?
Revisión del producto	Facilidad de prueba	¿Puedo probarlo?
	Flexibilidad	¿Puedo modificarlo?
	Reusabilidad	¿Podré reutilizar parte del software?
Transición del producto	Interoperación	¿Podré comunicarlo con otros sistemas?
	Portabilidad	¿Podré utilizarlo en otra computadora?

Figura 5. Modelo de McCall.

Estos tres aspectos manejados por McCall, se definen a partir de la visión general del usuario del software, cada una involucra atributos de calidad conocidos como criterios de calidad, que son elementos que están presentes en un producto de software, éstos a su vez se miden por ciertos factores de algunas características del producto de software los cuales son indicadores de un determinado atributo.

A grandes rasgos sabemos qué atributos debe de cumplir el producto de software para cubrir las expectativas del usuario final, con esto aseguramos una buena calidad y así la innovación de productos basados en los procesos ya controlados.

La problemática general a la que se enfrenta el software es:

- ❖ Aumento constante de tamaño y complejidad de los programas.
- ❖ Carácter dinámico e iterativo a lo largo de su ciclo de vida, es decir que los programas de software a lo largo de su vida cambian o evolucionan de una versión a otra para mejorar las prestaciones con respecto a las anteriores.
- ❖ Dificultad de conseguir productos totalmente depurados, ya que en ningún caso un programa será perfecto.
- ❖ Se dedican elevados recursos monetarios a su mantenimiento, debido a la dificultad que los proyectos de software entrañan y a la normalización a la hora de realizar los proyectos.
- ❖ No suelen estar terminados en los plazos previstos, ni con los costos estipulados, ni cumpliendo los niveles deseables de los requisitos especificados por el usuario.
- ❖ Incrementos constantes de los costos de desarrollo debido entre otros, a niveles de productividad bajos.
- ❖ Los clientes tienen una alta dependencia de sus proveedores por ser en muchos casos aplicaciones a "la medida".
- ❖ Procesos artesanales de producción con escasez de herramientas.
- ❖ Insuficientes procedimientos normalizados para estipular y evaluar la productividad, costos y calidad.

Todo lo anterior puede concretarse en:

- ❖ Ausencia de especificaciones completas, coherentes y precisas previas por parte del cliente, así como posteriores por parte de los proveedores del software.
- ❖ Ausencia de la aplicación sistemática de métodos, procedimientos y normas de ingeniería del software.
- ❖ Escasez o ausencia de entornos integrados de programación.
- ❖ Escasez de uso de técnicas actuales y automatizadas para la gestión de proyectos.
- ❖ Escasez de personal con formación y experiencia en los nuevos métodos, normas y uso de entornos y utilidades de programación.
- ❖ Otros derivados del grado de desarrollo técnico y organizativo de cada compañía.
- ❖ El aumento de la complejidad de las aplicaciones sin la adopción de los procesos adecuados de desarrollo contribuye a obtener un producto de mayor calidad y a menor costo.

Tener implantados sistemas de calidad en la empresa, significa, no sólo el tener que instalar la metodología del sistema de calidad, sino también sistemas de información que controlen y coordinen el sistema, sistemas automáticos, sistemas documentales, etc. Por todo ello, la implantación de sistemas de calidad en cualquier organización debe implicar que también el software que empleen los posea, pues de ésta forma se evitarían defectos provenientes de los sistemas de información.

En el año 2004 se liberó en la Ciudad de México el modelo de mejora de procesos de desarrollo de software conocido como MOPROSOFT, el cual fue diseñado para la pequeña y mediana empresa mexicana (PyME) y a las áreas internas de desarrollo de software.

MOPROSOFT es una norma orgullosamente mexicana que fue creada por la Universidad Nacional Autónoma de México, a petición de la Secretaría de Economía en 2002, con el inicio del programa para el desarrollo de la industria del software (PROSOFT), cuyo objetivo es fortalecer a la industria de software en México, mediante la estrategia de promover el uso y aprovechamiento de la tecnología y de la información.

Durante el desarrollo de MOPROSOFT se contó con la participación de 48 organizaciones, entre las que se encuentran instituciones de gobierno, privadas, universidades y asociaciones, lo que nos da la certeza de contar con una norma diseñada por mexicanos y para atender la problemática mexicana.

El modelo pretende que las PyMEs mejoren la capacidad de crear productos de software, para ofrecer servicios de buena calidad e incrementar la competitividad hasta alcanzar niveles internacionales.

1.7 Estructura de MOPROSOFT

MOPROSOFT está estructurado de una manera muy sencilla, hay tres categorías que representan los niveles de una organización.

Alta dirección, Gerencia y Operación, cada categoría cuenta con procesos definidos, mismos que indican las actividades para lograr una implementación del modelo, los roles necesarios, productos que se deben generar, así como la forma de resguardar ésta información y, entendiendo que cada organización tiene sus propias particularidades, se cuenta con una sección de ajustes a cada proceso; en términos simples, podemos decir que MOPROSOFT proporciona paso a paso la implementación de un modelo de calidad en la organización para obtener así el producto de software deseado que cumpla con los requerimientos solicitados.

MOPROSOFT identifica los procesos y los clasifica en tres categorías. Ver figura 6.

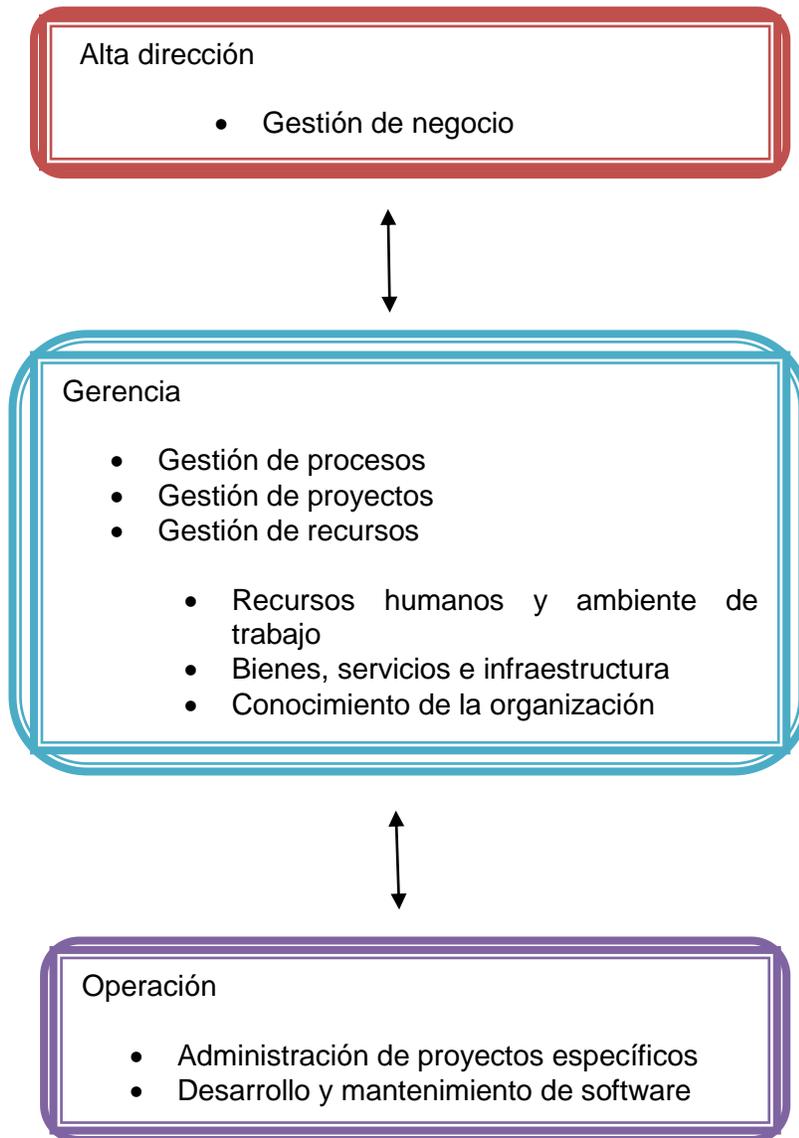


Figura 6. Categorías MOPROSOFT.

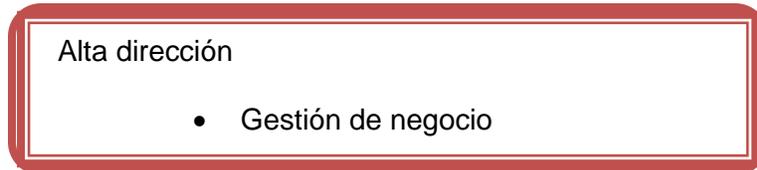
Categorías

1. Alta dirección

La categoría de la Alta dirección se ubica en los puestos directivos; en ella se define el rumbo de la empresa, la estrategia a seguir y la forma como se evaluará el cumplimiento de los objetivos.

La implantación de un plan estratégico nos ayuda a cubrir completamente el proceso de esta categoría.

La información generada en este proceso nos ayuda a contestar la pregunta: ¿Hacia dónde va la organización?

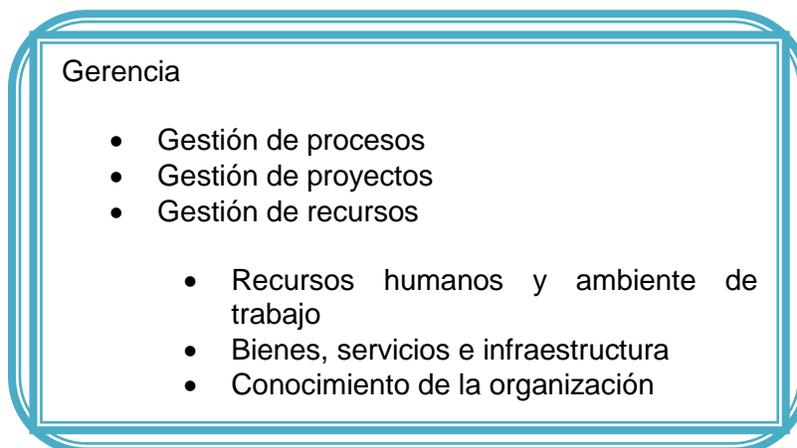


2. Gerencia

La categoría de Gerencia, se ubica en los puestos gerenciales de la organización y es la encargada de la administración de recursos, proyectos y procesos.

Es la categoría que brinda más información a la organización, porque nos indica por cada recurso humano o material, dónde se encuentra y nos contesta preguntas como:

¿Quién podría apoyarnos para un proyecto en la tecnología?, ¿dónde encuentro un formato de propuesta de venta del proyecto?, ¿quién es el responsable del proceso?, ¿qué cursos de capacitación se brindarán?



La categoría de Gerencia se divide en tres procesos:

- Gestión de procesos
- Gestión de proyectos
- Gestión de recursos

Además de tres subprocessos. Ver figura. 7

- Recursos humanos y ambiente de trabajo
- Bienes, servicios e infraestructura
- Conocimiento de la organización

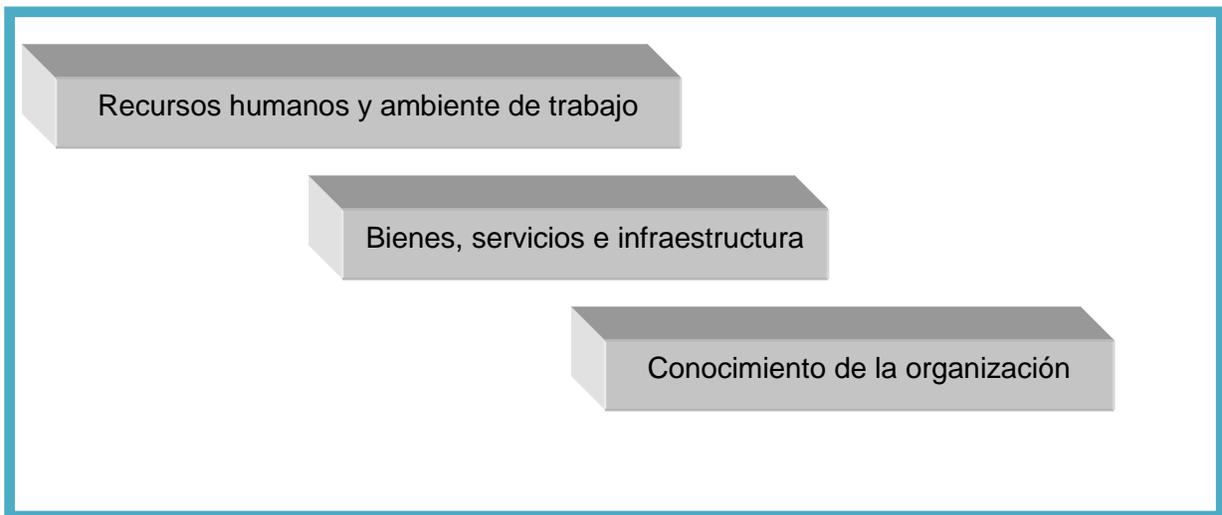


Figura 7. Subprocesos categoría de Gerencia.

El proceso de Gerencia de procesos tiene como propósito implantar los procesos de la organización en función de los todos los procesos identificados en el plan estratégico, así como definir, plantear e implementar las actividades de mejora de los mismos. La Gestión de proyectos es la llamada oficina de proyectos, en ella se administran los proyectos, buscando que contribuyan al cumplimiento de los objetivos estratégicos de la empresa y finalmente la Gestión de recursos, busca conseguir y brindar los recursos humanos, infraestructura, ambiente de

trabajo y proveedores, así como crear y mantener la base de conocimiento de la organización.

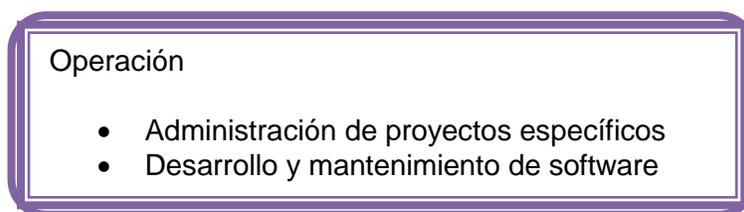
La finalidad de la categoría de Gerencia que engloba a los tres procesos es apoyar el cumplimiento de los objetivos del plan estratégico de la organización.

3. Operación

Por último, se encuentra la categoría de Operación, que es donde se realizan las actividades que permiten a la empresa existir, el desarrollo de los proyectos. Esta categoría soporta las actividades tanto de la administración de proyectos en específico, como el desarrollo de los mismos.

“Realiza las actividades de acuerdo a los elementos proporcionados por la categoría de Gerencia y entrega a ésta la información y productos ya generados.”⁷

Esta categoría cuenta con los siguientes procesos:



- Administración de proyectos específicos. Aquí se establecen y llevan a cabo de forma continua las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
- Desarrollo y mantenimiento de software. Es donde se llevan a cabo todas las actividades de análisis, diseño, construcción, integración y pruebas de los productos de software ya sea nuevo o modificado cumpliendo con los requisitos solicitados.

⁷ Oktaba Hanna, et al. *Modelo de procesos para la industria del software MOPROSOFT*. Versión 1.5. Secretaría de Economía. México, 2005, pág. 10.

Administración de proyectos específicos

El propósito de la administración de proyectos específicos es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

Se define cada etapa de la administración de un proyecto. Ver figura 8.

1. Planeación. Actividades cuya finalidad es obtener y mantener el plan de proyecto y el plan de desarrollo que regirán al proyecto específico, con base en la descripción del proyecto.
2. Realización. Consiste en llevar a cabo las actividades del plan del proyecto.
3. Evaluación y control. Consiste en asegurar que se cumplan los objetivos del proyecto. Se supervisa y evalúa el progreso para identificar desviaciones y realizar acciones correctivas cuando sea necesario.
4. Cierre. Consiste en entregar los productos de acuerdo a un protocolo de entrega y dar por concluido el ciclo.

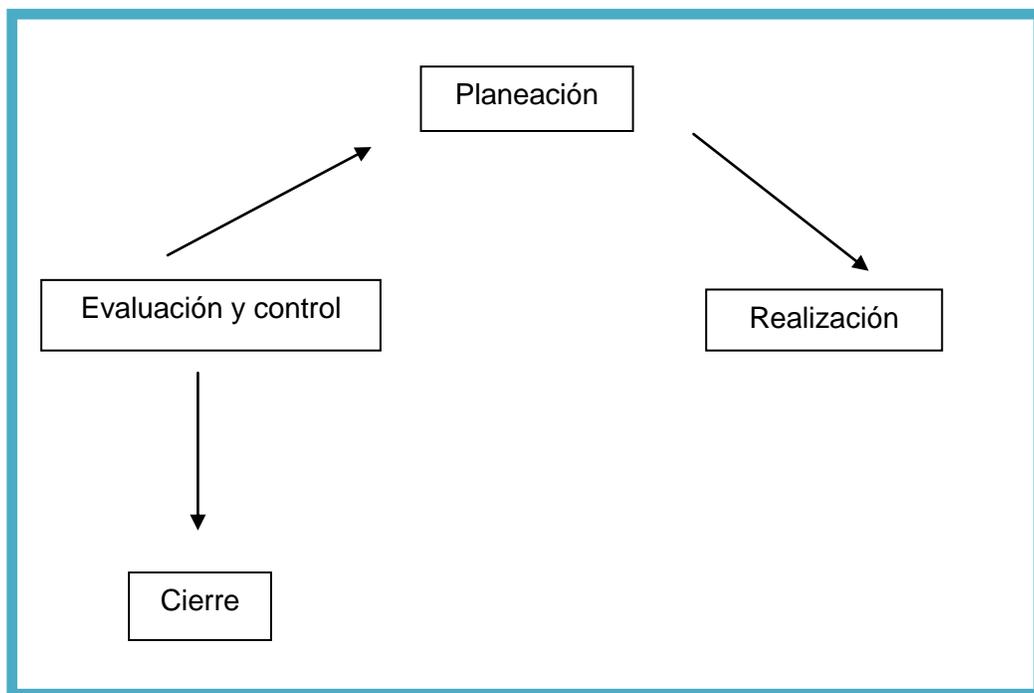


Figura 8. Etapas de la administración de un proyecto.

Desarrollo y mantenimiento de software

Aquí el propósito es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software cumpliendo con los requerimientos especificados.

El proceso de desarrollo y mantenimiento de software se compone de uno o más ciclos de desarrollo, ver figura 9.

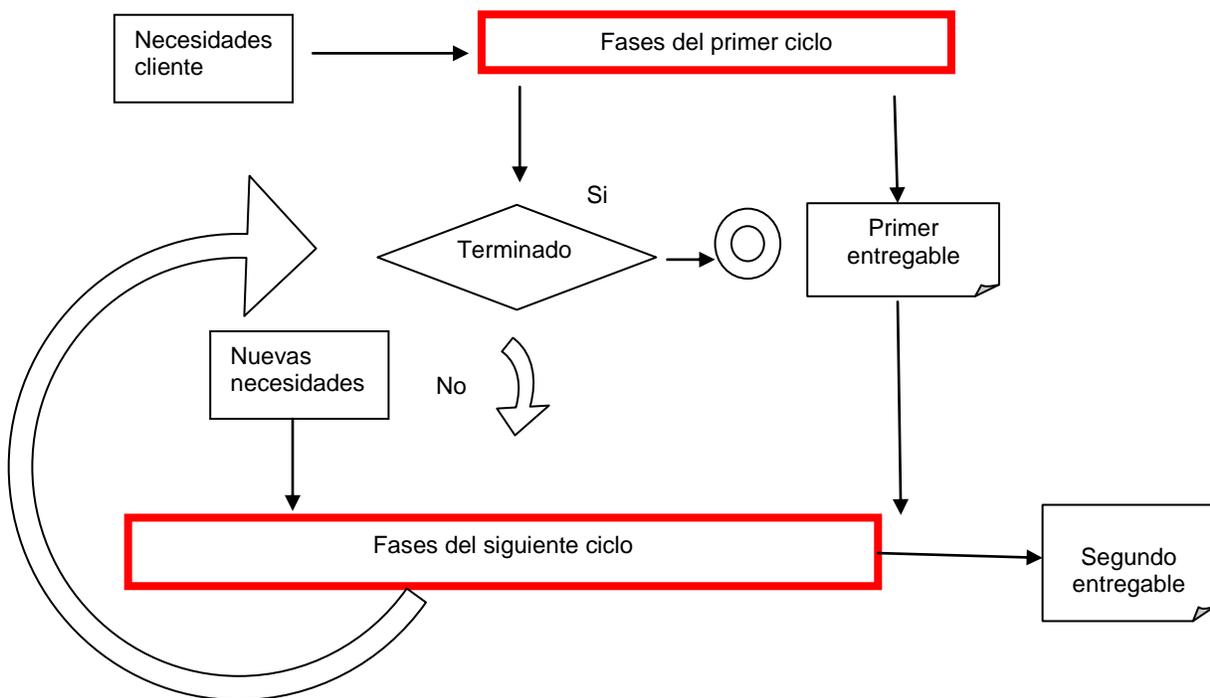


Figura 9. Etapas ciclo de desarrollo.

Fases de un ciclo de desarrollo

Es el conjunto de actividades sucesivas que los analistas, diseñadores y usuarios realizan, cada actividad requiere información de entrada, procesos y resultados, los cuales deben estar bien definidos.

Cada ciclo está formado por las siguientes etapas, ver figura 10.

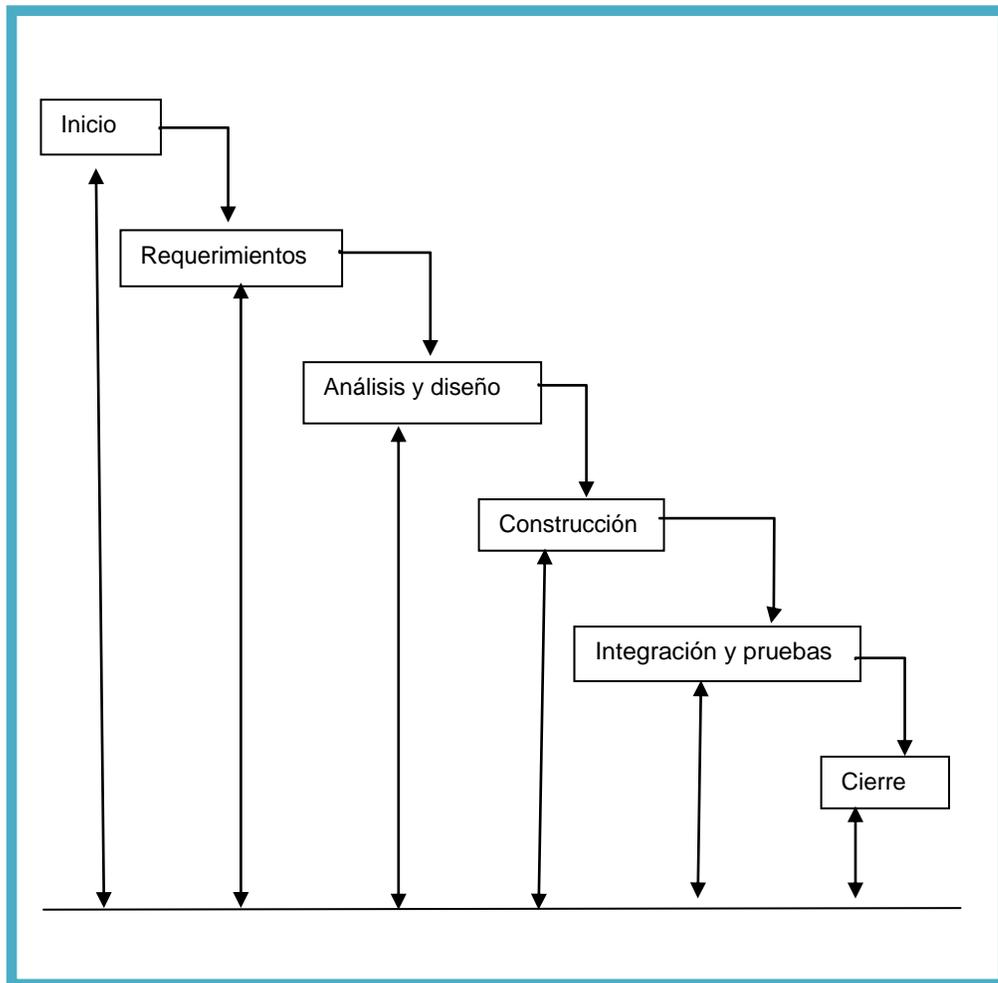


Figura 10. Etapas ciclo de vida.

- "Inicio. Revisión del plan de desarrollo por los miembros del equipo de trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.
- Requerimientos. Conjunto de actividades cuya finalidad es obtener la documentación de la especificación de requerimientos y plan de pruebas de sistema, para conseguir un entendimiento común entre el cliente y el proyecto.

- Análisis y diseño. Conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del análisis y diseño y plan de pruebas de integración.
- Construcción. Conjunto de actividades para producir componente(s) de software que correspondan al análisis y diseño, así como la realización de pruebas unitarias. Como resultado se obtienen el (los) componente(s) de software probados.
- Integración y pruebas. Conjunto de actividades para integrar y probar los componentes de software, basados en los planes de pruebas de integración y de sistema, con la finalidad de obtener el software que satisfaga los requerimientos especificados. Se genera la versión final del manual de usuario, manual de operación y manual de mantenimiento. Como resultado se obtiene el producto de software probado y documentado.
- Cierre. Integración final de la configuración de software generada en las fases para su entrega, identificación y documentación de las lecciones aprendidas, generación del reporte de mediciones y sugerencias de mejora. ”⁸

El mantenimiento del software es una frase que engloba todo y que se usa para denotar las distintas actividades de verificación sucedidas después de la liberación del producto. Las modificaciones se realizan para mejorar, adaptar y corregir errores en los productos de software. Mantener la calidad de un producto de software a través de ciclos sucesivos de modificaciones y actualizaciones es un aspecto de importancia fundamental durante el desarrollo de software.

“La calidad de un producto de software se puede degradar debido a parches y arreglos rápidos, cuando no se es cuidadoso en garantizar que el estilo de documentación y estándares de documentación se mantengan y, además, que los documentos de apoyo se actualicen para ilustrar las modificaciones.”⁹

En la tabla A, se muestran los nueve procesos de MOPROSOFT agrupados por categorías, así como su principal objetivo de cada una y su respectiva abreviatura para un mejor uso.

⁸ Oktaba Hanna, et al. *Modelo de procesos para la industria del software MOPROSOFT*. Versión 1.5. Secretaria de Economía. México, 2005, pág. 93.

⁹ Fairley, Richard. *Ingeniería de software*. McGraw-Hill. México, 1987, pág. 365.

Tabla A. Procesos de MOPROSOFT.

Categoría	Proceso	Objetivo
Alta dirección (DIR)	Gestión de negocio (GN)	Establecer la razón de ser de la organización, sus objetivos y condiciones para lograrlos, para lo cual es necesario contemplar las necesidades de los clientes, así como evaluar resultados para poder proponer algún cambio y permitir la mejora continua
Gerencia (GER)	Gestión de procesos (GPR)	Establecer los procesos de la organización en función de los procesos requeridos identificados en el plan estratégico Así como también definir, planificar e implementar las actividades de mejora de los mismos
Gerencia (GER)	Gestión de proyectos (GPY)	Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización
Gerencia (GER)	Gestión de recursos (GR)	Conseguir y dotar a la organización de recursos humanos, infraestructura y ambiente de trabajo, proveedores, también crear y mantener la base de conocimiento de la organización El objetivo aquí es apoyar con el cumplimiento del plan estratégico de la empresa
Gerencia (GER)	Recursos humanos y ambiente de trabajo (RHAT)	Proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la empresa
Gerencia (GER)	Bienes, servicios e infraestructura (BSI)	Proporcionar proveedores de bienes y servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos
Gerencia (GER)	Conocimiento de la organización (CO)	Mantener y administrar la base de conocimiento que contiene la información y los productos generados por la organización
Operación (OPE)	Administración de proyectos específicos	Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo
Operación (OPE)	Desarrollo y mantenimiento de software (DMS)	Realización sistemática de actividades de obtención de requisitos de análisis, diseño, construcción integración y pruebas de productos de software nuevos o modificados cumpliendo con los objetivos

1.8 Ventajas de utilizar MOPROSOFT

MOPROSOFT es un modelo fácil de entender y aplicar en una organización, ya que está dividido en tres partes o categorías, orientado en específico a un área determinada en una organización, la parte de Alta dirección, Gerencia y la Operación, permitiéndonos determinar las funciones que cada individuo debe realizar para el mejoramiento de los productos.

Cuenta con un patrón, el cual sirve de plantilla a los usuarios para documentar cada uno de los procesos realizados.

Estas documentaciones servirán al evaluador para determinar el cumplimiento de los nueve procesos mencionados anteriormente.

En seis meses una empresa puede elevar el nivel de todos sus procesos trabajando constantemente hasta controlarlos para determinar su capacidad y madurez logrando así, avanzar al siguiente nivel hasta alcanzar la certificación y ser capaz de buscar un plus e innovar sus productos de software.

Todo esto permite a la compañía ser reconocida para poder competir y buscar un crecimiento.

El punto más importante es que va dirigido a las PyMEs que se dediquen al desarrollo y mantenimiento de software o bien al departamento donde se desarrolle el software.

Entre las principales características de MOPROSOFT encontramos:

- Es específico para el desarrollo y mantenimiento del software
- Fácil de entender; los procesos están organizados en 3 categorías, que corresponden a la estructura de cualquier organización
- Se define como un conjunto de procesos, sólo 6 procesos principales y 3 subprocesos

- Es práctico y fácil de aplicar, sobre todo en organizaciones pequeñas.
- Está orientado a mejorar los procesos de una organización y no simplemente para una certificación
- Tiene un mecanismo de evaluación para saber el estado de la organización durante un cierto periodo
- Aplicable como norma mexicana

“Entre otros beneficios encontramos que:

Al adoptar y aplicar el modelo MOPROSOFT, las empresas incorporarán las mejores prácticas en gestión e ingeniería de software, lo cual les permitirá elevar la capacidad de las organizaciones para ofrecer servicios con calidad.

Es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros.

A continuación se mencionan la relación entre las principales características del modelo y las ventajas o beneficios para la implementación de MOPROSOFT.

Característica

1. Las categorías de procesos corresponden a niveles organizacionales de administración

Beneficios y ventajas

- Estructura de administración común en muchas organizaciones
- Facilita la comprensión del modelo
- Simplifica la relación entre el modelo de procesos y la organización.

Característica

2. Procesos integrados y relacionados

Beneficios y ventajas

- Cuenta únicamente con nueve procesos evitando la fragmentación que se presenta en otros modelos
- La relación entre procesos se establece mediante la identificación de los productos de trabajo de entrada y salida y la definición de las responsabilidades de los roles que participan en más de un proceso

Característica

3. Foco en producto y su capitalización

Beneficios y ventajas

- Se identifican y la caracterizan los productos y las actividades de verificación y validación a las que deben estar sometidos
- El proceso de conocimiento de la organización administra una base de conocimiento, que controla y asegura la disponibilidad de los productos de trabajo

Característica

4. Capacidad organizacional de Gestión de procesos

Beneficios y ventajas

- Un solo proceso, Gestión de procesos, establece la capacidad organizacional para la planeación, definición, implantación, evaluación y valoración de procesos

- Regido por las directrices de Gestión de negocio, lo que asegura la alineación con los objetivos

Característica

5. Capacidad organizacional de Gestión de proyectos

Beneficios y ventajas

- Se distingue entre la administración a nivel proyecto (Administración de proyecto específico) y la gestión del portafolio de proyectos de la organización (Gestión de proyectos)
- La Gestión de proyectos facilita la Identificación de iniciativas y proyectos; la provisión, asignación y reasignación de recursos a programas y proyectos; y el mantenimiento del balance del portafolio

Característica

6. Alineación con objetivos de negocio

Beneficios y ventajas

- El proceso de Gestión de negocio enfatiza la importancia de alinear todas las actividades de la organización a los objetivos del negocio a través de la elaboración, difusión, valoración y mejora del plan estratégico
- El plan estratégico sirve de guía a los demás procesos de la organización logrando de este modo una alineación explícita con los objetivos de negocio”¹⁰

¹⁰ <http://www.iteraproces.com>

1.9 Análisis de MOPROSOFT con otras normas

Como se había comentado en capítulos anteriores, MOPROSOFT se basó en los siguientes modelos:

- ISO 9001:2000 Sistemas de Gestión de Calidad-requisitos. Tabla B.
- SW-CMM. Tabla C.
- ISO/IEC TR 15504. Tabla D.

A continuación, se enlistan las ventajas y desventajas de cada modelo mencionado.

ISO 9001:2000

Es una norma internacional para evaluar la capacidad de la organización para poder cumplir con las especificaciones del cliente, los reglamentarios y los propios de una organización.

Se aplica a cualquier organización que quiera mejorar la calidad general de los productos, sistemas o servicios que provee, por lo que se aplica de modo directo a compañías y organizaciones de software.

Los requisitos del sistema de gestión de la calidad son complementarios a los requisitos del producto y se especifican en la norma ISO 9001:2000, son genéricos y aplicables a organizaciones de cualquier sector económico e industrial con independencia del producto que suministren y además hacen énfasis en el uso y aplicaciones técnicas del producto.

Ha adoptado un ciclo de “planear-hacer-revisar-actuar” que se aplica a los elementos de gestión de calidad de un proyecto de software. Dentro de un contexto de software, “planear” establece los objetivos, las actividades y tareas del proceso necesarios para conseguir un software de alta calidad y una satisfacción del cliente; “hacer” implementa el proceso de software; “revisar” monitorea y mide el proceso para asegurarse de que todos los requisitos hayan sido cumplidos; “actuar” inicia las actividades de mejoramiento del proceso de software, el cual tiene una continuidad de trabajo para mejorar el proceso.¹¹

En la tabla B se muestran las ventajas y desventajas del modelo.

¹¹ Pressman S. Roger; *Ingeniería de Software*; McGraw-Hill, 6ta. edición. México, 2005, pág. 38.

La desventaja más notable es que no es específico para la industria del software, aunque esta norma es muy conocida a nivel nacional, no es fácil de entender.

MOPROSOFT está dirigido a las PyMEs dedicadas al mantenimiento y desarrollo del software, es fácil de entender, se está conociendo y aplicando poco a poco y cada año se eleva el número de compañías que lo están adoptando.

Tabla B. Norma ISO 9001:2000.

ISO 9001:2000	
Ventajas	Desventajas
Cuenta con un mecanismo de certificación bien definido	No es específica para la industria del software
Es conocida y está disponible	No es fácil de entender
	No se aplica fácilmente
	No se define como un conjunto de procesos
	La mejora de procesos se enfoca básicamente a la calidad

SW-CMM

El modelo de madurez de la capacidad para el desarrollo de software (*Capability Maturity Model for Software, SW-CMM*) es un modelo de procesos para el desarrollo y mantenimiento de sistemas de software, diseñado sobre los siguientes criterios:¹²

- La calidad de un producto o sistema es consecuencia directa de los procesos empleados en su desarrollo.
- Las organizaciones que desarrollan software presentan un atributo denominado madurez, cuya medida es proporcional a los niveles de capacidad e institucionalización de los procesos que emplean en su trabajo.

¹² <http://es.wikipedia.org/wiki/SW-CMM>

- Fue diseñado a finales de los ochenta por Software Engineering Institute (SEI) a instancias del Congreso Norteamericano, como medio para evaluar a las empresas suministradoras de software para el Departamento de Defensa Norteamericano.

SW-CMM define 5 niveles de madurez para las organizaciones, en función de cuáles son los procesos que emplean en el desarrollo y mantenimiento de software y los grados de capacidad e institucionalización de cada uno; y puede emplearse como criterio para la evaluación de la madurez de la organización o bien de guía para la mejora de sus procesos.

Niveles de madurez definidos en SW-CMM

Nivel 1: Inicial

Los resultados de calidad obtenidos en este nivel son consecuencia de las personas y de las herramientas que las emplean. No de los procesos, porque no los hay o no se emplean.

Nivel 2: Repetible

Se considera un nivel 2 de madurez cuando se llevan a cabo prácticas básicas de Gestión de proyectos, de gestión de requisitos, control de versiones y de los trabajos realizados por subcontratistas. Los equipos de los proyectos pueden aprovechar las prácticas realizadas para aplicarlas en nuevos proyectos.

Nivel 3: Definido

Los procesos comunes para desarrollo y mantenimiento del software están documentados de manera suficiente en una biblioteca accesible a los equipos de desarrollo. Las personas han recibido la formación necesaria para comprender los procesos.

Nivel 4: Gestionado

La organización mide la calidad del producto y del proceso de forma cuantitativa con base a métricas establecidas.

La capacidad de los procesos empleados es previsible y el sistema de medición permite detectar si las variaciones de capacidad exceden los rangos aceptables para adoptar medidas correctivas.

Nivel 5: Optimizado

La mejora continua de los procesos afecta a toda la organización, que cuenta con medios para identificar las debilidades y reforzar la prevención de defectos. Se analizan de forma sistemática datos relativos a la eficacia de los procesos de software para analizar el costo y el beneficio de las adaptaciones y las mejoras.

Se analizan los defectos de los proyectos para determinar las causas y su mapeado sobre los procesos.

Es el nivel más alto de SW- CMM por el momento¹³.

Tabla C. Norma SW-CMM.

SW-CMM	
Ventajas	Desventajas
Específico para el desarrollo y mantenimiento del software	Es un modelo extranjero
Se define como un conjunto de áreas clave de procesos	No es fácil de aplicar ya que está pensado para organizaciones grandes
Tiene un modelo de evaluación	La mejora no se enfoca directamente a los objetivos de negocio
Desde el año 1998 comenzó a ser popular en México	La evaluación es costosa y no tiene periodo de vigencia
Hay organizaciones evaluadas	Se está abandonando a favor de CMMI

¹³ <http://es.wikipedia.org/wiki/SW-CMM>

Es un modelo dirigido a las empresas grandes, dedicadas a la industria del software, es un modelo extranjero que no es fácil de entender, además la evaluación de los procesos es muy costosa, y ya no se aplica mucho. Ver tabla C.

MOPROSOFT es un modelo totalmente mexicano, fácil de aplicar y entender, no es costoso, ya que sus procesos se dividen en tres partes dirigidos al área de Alta dirección, Gerencia y Operación de las PyMEs.

Su mecanismo de evaluación no es costoso, hay un periodo de vigencia, el cual después de un corto tiempo, obliga a la empresa a volver a certificarse obteniendo un nivel mayor asegurando el buen funcionamiento de un producto.

ISO/ IEC TR 15504

Estándar internacional IS ISO/ IEC 15504 – Tecnologías de la Información.

Es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

En 1991, dado el número creciente de métodos de evaluación de procesos disponibles y el uso de estas técnicas en áreas comerciales sensibles, la Organización de Estandarización internacional ISO aprueba la realización de un estudio al respecto de la necesidad de crear un estándar internacional para la evaluación de procesos.

Principales características:

- Define el modelo de referencia de procesos de software y de capacidades de procesos que constituyen la base para la evaluación de procesos de software.
- Tiene una arquitectura basada en dos dimensiones: de proceso y de capacidad de proceso.
- Define que todo modelo de evaluación de procesos debe definir: la dimensión de procesos, basándose en el modelo de procesos de referencia.
- La dimensión de la capacidad: niveles de capacidad y atributos de los procesos.

Los niveles de capacidad para todo modelo de evaluación de procesos pueden tener desde el 0 y al menos hasta el nivel 1 de los siguientes niveles de capacidad estándar:

- Nivel 0: Incompleto
- Nivel 1: Realizado
- Nivel 2: Gestionado
- Nivel 3: Establecido
- Nivel 4: Predecible
- Nivel 5: En optimización

Para cada nivel existen unos atributos de procesos estándar que ayudan a evaluar los niveles de capacidad.

Tabla D. Norma ISO/ IEC TR 15504.

ISO/ IEC TR 15504	
Ventajas	Desventajas
Es específico para el desarrollo y mantenimiento del software	No es práctico ni fácil de aplicar
Fácil de entender, son 24 procesos, en 16 páginas	Sólo tiene lineamientos para un mecanismo de evaluación
Se define como un conjunto de procesos	Aún no es norma internacional
Está orientado a mejorar los procesos, para cumplir con los objetivos del negocio	

Aunque la norma también está dirigida a la industria del software, una de sus desventajas es que no es fácil de aplicar y no es práctico debido que se conforma de 24 procesos que a diferencia de MOPROSOFT sólo maneja 9 procesos, los cuales están bien distribuidos en la organización para su entendimiento y aplicación. Ver tabla D.

A continuación se hace un análisis minucioso de las tres normas mencionadas con anterioridad en relación con MOPROSOFT, se observan las ventajas denominadas con la letra V, las desventajas con la letra D y ambas comparadas con otros modelos, con letra A.

Ver Tabla E. Características.

Ver Tabla F. Prácticas.

Ver Tabla G. Comparación de los tres modelos.

Tabla E. Características de MOPROSOFT.

Características deseadas/modelos	ISO 9001:2000	SW-CMM	ISO/ IEC TR 15504
Para software	D	V	V
Comprensible	D	D	V
Procesos	D	V	V
Práctico	D	D	D
Mejora de procesos orientado al objetivo de la organización	D	D	V
Evaluación con vigencia	V	A	D
Aplicable como norma	V	D	D

Podemos ver que al comparar los tres modelos con MOPROSOFT, éste cubre siete características, es específico para software, al igual que SW-CMM e ISO/ IEC TR 15504.

Es comprensible y práctico; está totalmente orientado para mejorar el objetivo de la organización, lo cual ISO 9001:2000 y SW-CMM no. Además de que su evaluación tiene vigencia de dos años.

Tabla F. Prácticas.

MOPROSOFT/ OTROS	SW-CMM	ISO/ IEC TR 15504
Roles involucrados y capacitación	D	D
Actividades	V	V
Diagrama de flujo de trabajo	D	D
Verificaciones y validaciones	A	D
Incorporación a la base de conocimiento	D	D
Recursos e infraestructura	D	D
Mediciones	V	D
Situaciones excepcionales	D	D
Lecciones aprendidas	A	D

En cuanto a las prácticas empleadas, MOPROSOFT cumple con los nueve puntos lo que a diferencia de la norma ISO/IEC TR 15504 sólo cubre la parte de actividades a realizar en cada proceso.

Le sigue SW-CMM con dos puntos a favor: actividades y mediciones; y entre las desventajas que encontramos: el punto de verificaciones y validaciones y las lecciones aprendidas.

1.10 Comparación con otros modelos

Tabla G. Comparación de modelos.

Características/ modelos	ISO 9001:2000	SW-CMM	ISO/IEC TR 15504	MOPROSOFT
Para SW	D	V	V	V
Comprensible	D	D	V	V
Procesos	D	V	V	V
Práctico	D	D	D	V
Mejora de procesos orientada al objetivo de la organización	D	D	V	V
Evaluación con vigencia	V	A	D	D
Aplica como norma	V	D	D	D

Se hace un análisis de los tres modelos sobre sus características más representativas. En primer lugar vemos que MOPROSOFT tiene cinco ventajas, seguido de ISO/ IEC TR 15504, al cumplir sólo cuatro, SW-CMM presenta dos ventajas, y en cuanto a la característica que se refiere a evaluación con vigencia se encuentra tanto con ventaja y desventaja.

ISO 9001:2000 tiene cinco puntos con desventajas.

Esto indica que MOPROSOFT cuenta con más ventajas sobre los tres modelos pues presenta pocas desventajas y se dirige a la pequeña y mediana empresa.

Se enlistan a continuación las principales características del modelo.

Características de MOPROSOFT

“Con toda esta comparación tenemos que con MOPROSOFT hay una gran ventaja en cuanto a lo siguiente:

- Son pocos procesos y abarcan a toda la organización en sus áreas de la Alta dirección, el área de Gestión y área de Operación.
- Los procesos están integrados y se relacionan entre sí.
- Hay roles específicos definidos para cada proceso.
- Se define el propósito, objetivos, indicadores y metas cuantitativas.
- Definición de los productos de entrada, salida e internos de cada proceso y sus características.
- Se definen los flujos de trabajo con las tareas, actividades, roles y los productos generados.
- Existe una base de conocimiento de la organización donde encontramos los productos generados, éstos se administran y se consultan.
- Hay una definición de las actividades para obtener las lecciones aprendidas y usarlas en proyectos futuros.
- Se define un mecanismo específico para la reacción a las situaciones excepcionales cuando se llevan a cabo cada una de las actividades.
- Se definen explícitamente las actividades de verificación, validación y pruebas para una buena calidad de los productos.
- Definición de guías de ajuste que se adaptan a las empresas, sin perder de vista los objetivos de dichos procesos.
- Los objetivos y metas cuantitativas nos ayudan a guiar a los proyectos y procesos y son los que se evalúan para saber la efectividad de cada proceso.

- Existen sugerencias de mejora a cada proceso, las cuales se reportan a los responsables del proceso gestión de procesos.
- Los procesos del modelo se ajustan de acuerdo al contexto de cada organización.¹⁴

La calidad es una capacidad para identificar las necesidades y expectativas de los clientes y demás partes interesadas para satisfacerlas, cumpliendo los requerimientos del producto o software desarrollado, esto adquiere cada vez más importancia en la gestión de las empresas de ahí que los gerentes reconozcan que se pueden obtener ventajas competitivas sustanciales mediante el desarrollo de sistemas de gestión de calidad.

Es clara la necesidad en nuestro país de aplicar métodos probados para desarrollo de software (esto es ingeniería de software) y la necesidad de seguir modelos de mejora para conseguir competencia internacional. La competencia internacional es una necesidad en el mundo globalizado actual para permanecer en el mercado.

Por ello, los sistemas de gestión de la calidad basados en las normas aquí mencionadas, han cobrado una gran importancia y muchas organizaciones se han decidido a tomar el camino de documentar e implementar sus procesos.

En este capítulo 1, Fundamentos de MOPROSOFT, se habló de la norma, su estructura y los puntos a favor y en contra de cada modelo en estudio, por todo el análisis se podría concluir que MOPROSOFT es el más adecuado por las ventajas que ofrece al aplicarse en las PyMEs. Es una manera simple de iniciar un camino a la internacionalización.

¹⁴ Apuntes curso *Consultores junior MOPROSOFT 2007*. IIMAS, UNAM, 2007.