

Tema 2. Ingeniería de software. El proceso

Tema 2. Ingeniería de software. El proceso	i
2.1.- Definición de Ingeniería del Software.	15
2.2.- Los procesos para la construcción del software.....	16
2.2.1.- La norma IEEE 1074 [Piattini 2003]	18
2.2.2.- Norma ISO 12207 – 1 [Piattini 1996].....	21
2.2.3.- La norma ISO/IEC TR 15504-2 [Piattini 2003].....	25
2.3.- Evaluación del proceso software.....	27
2.3.1.- Integración del Modelo de Capacidad de Madurez (CMMI).....	28

Índice de Figuras

Figura 2.1.- Secciones lógicas para los procesos en el estándar IEEE 1074.....	19
Figura 2.2.- Agrupación de procesos de la norma ISO 12207 - 1	21
Figura 2.3.- Procesos de la ISO/IEC 15504-2 y su alineamiento con la ISO 12207-1	25
Figura 2.4. Procesos de la norma ISO/IEC 15504-2 (rev. 2003)	27
Figura 2.5. Relación con los procesos de evaluación.....	28
Figura 2.6. Áreas de proceso por categoría para el CMMI continuo.	29
Figura 2.7. PAs por nivel de madurez del modelo CMMI discreto.	31

2.1.- Definición de Ingeniería del Software.

El desarrollo de sistemas de software complejos no es una actividad trivial, que pueda abordarse sin una preparación previa. El considerar que un proyecto de desarrollo de software podía abordarse como cualquier otro ha llevado a una serie de problemas que limitan nuestra capacidad de aprovechar los recursos que el hardware pone a nuestra disposición.

Los problemas tradicionales en el desarrollo de software no van a desaparecer de la noche a la mañana, pero identificarlos y conocer sus causas es el único método que nos puede llevar hacia su solución.

No existe una fórmula mágica para solucionar estos problemas, pero combinando métodos aplicables a cada una de las fases del desarrollo de software, construyendo herramientas para automatizar estos métodos, utilizando técnicas para garantizar la calidad de los productos desarrollados y coordinando todas las personas involucradas en el desarrollo de un proyecto, podremos avanzar mucho en la solución de estos problemas. De ello se encarga la disciplina llamada *Ingeniería del Software*.

Una de las primeras definiciones que se dio de la ingeniería del software propuesta por Fritz Bauer es la siguiente:

Ingeniería del software es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico, que sea fiable y funcione de manera eficiente sobre máquinas reales.[Nau69]

Esta definición no es más que un establecimiento de principios para resolver la crisis del software de la que hemos hablado y para el que la definición propone a la ingeniería del software como la respuesta definitiva.

El IEEE [IEEE93] ha desarrollado una definición más completa según la cual:

Ingeniería del software: La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software.

En esta definición está implícita la existencia de fases por las que se debe hacer pasar al software de manera sistemática, y lo que es más complicado, de forma cuantificable. Es decir, la ingeniería del software nos debe proporcionar una metodología de desarrollo de software que nos precise en cada momento que pasos debemos seguir para construir un software económico y de calidad. Aún más nos debe proporcionar la forma de averiguar en qué parte del proceso estamos y cuan bien lo estamos haciendo.

La primera aproximación a esta idea se hace a través de una serie de paradigmas que tratan de describir las fases que debe atravesar el proceso de desarrollo de software desde que éste es concebido hasta que deja de usarse. A dichos paradigmas se les denomina CICLOS DE VIDA DEL SOFTWARE y serán analizados posteriormente. Esta aproximación surge del análisis del proceso de Software desde la globalidad al detalle con esta perspectiva también se plantean otras ideas como las de proceso o estándares que no buscan una descripción detallada de las actividades si no su identificación. Pero esta perspectiva no es la única posible, se plantean así las metodologías que tienen históricamente una perspectiva menos amplia del problema pero mayor nivel de detalle en la descripción de las actividades.

En cualquier caso, las fronteras entre estos conceptos se mezclan con frecuencia. En general es interesante el planteamiento que propone Piattini al respecto utilizando la definición de ciclo de

vida del IEEE como una serie de procesos que se deben realizar sin definir el detalle de cómo llevarlos a cabo. Posteriormente introduce el concepto de metodología como las estrategias que definen con precisión los pasos que se deben seguir en cada momento. Aclara, sin embargo, que no existe una metodología estándar que pueda aplicarse al desarrollo de cualquier proyecto software sino una serie de ellas que además, cada empresa debería refinar para adecuarla a su particular idiosincrasia.

En cualquier caso debe quedar claro que la ingeniería del software busca formalizar la secuencia de pasos que hay que dar para completar con éxito el desarrollo de un software de calidad y que, aunque en este sentido no está todo dicho, si existen una serie de normas y estrategias que se deben seguir.

A lo largo de su historia se han propuesto distintos ciclos de vida del software para los que se definen distintas fases e interrelaciones entre ellas. En este proceso han surgido distintos términos como metodología, proyectos, ciclo de vida, procesos, fases, actividades, tareas, procedimientos, métodos, técnicas o herramientas que, según que autor sigamos, pueden ser usados con interpretaciones ligeramente distintas, a veces algunos términos son usados como sinónimos y en otros casos se les dan acepciones distintas. Todo esto es fruto de que el proceso de sistematización todavía se encuentra en fase de desarrollo y conviene tenerlo muy presente cuando leamos bibliografía de distintos autores.

Aunque a continuación intentaremos clarificar estos términos no deben entenderse como definiciones formales aceptadas por todo el mundo sino como un intento de establecer una jerarquía útil entre los distintos conceptos.

Las herramientas son el concepto más claro y obvio están constituidas por cualquier software que proporcione soporte automático a cualquier actividad relacionada con la ingeniería del software. Con frecuencia las herramientas se centran en algún aspecto concreto y facilitan la ejecución de una o varias técnicas de forma que es preciso utilizar varias para abarcar todas las actividades del desarrollo del software.

Las técnicas se centran principalmente en un conjunto de gráficos con apoyos textuales formales y determinan el formato de los productos resultantes de cada tarea. Una idea importante a tener presente es que la misma técnica puede usarse con distintos objetivos.

Los métodos y procedimientos determinan el modo en el que se aplican las técnicas y el detalle de los elementos de los que se debe partir y los productos resultantes de aplicar dicho método o procedimiento. Indican cómo construir técnicamente el software, y abarcan una amplia serie de tareas que incluyen la planificación y estimación de proyectos, el análisis de requisitos, el diseño de estructuras de datos, programas y procedimientos, la codificación, las pruebas y el mantenimiento. Los métodos introducen frecuentemente una notación específica para la tarea en cuestión y una serie de criterios de calidad. Según los autores podemos encontrar que un método engloba a procedimientos o viceversa o bien encontrar que se usan como sinónimos.

Metodologías, ciclos de vida y proceso del software. Estos son los conceptos más amplios que nos vamos a encontrar. De nuevo y según el caso se incluyen unos a otros o se utilizan como sinónimos. En todos los casos estos conceptos precisan una serie de fases por las que se ha de pasar en el desarrollo del software y según los autores cada uno de estos conceptos incluye más o menos precisión en la definición de dichas fases.

2.2.- Los procesos para la construcción del software

Como hemos discutido con anterioridad la construcción del software incluye una serie de actividades que comienzan a estandarizarse. En esta sección empezaremos por una visión simple

propuesta por Pressman de las fases técnicas inexcusables para la construcción del software. Posteriormente veremos un par de normas que amplían considerablemente esta visión, la IEEE 1074 y la ISO 12207-1. En ambos casos se pretende detectar todas las actividades precisas para la construcción del software y veremos, a través de su evolución histórica como la lista tiende a ampliarse. Estas listas no sólo describen actividades directamente orientadas a la construcción del software sino también a dar soporte a tal actividad y a las empresas dedicadas a tal empeño.

Posteriormente analizaremos distintas estrategias que, centradas en el desarrollo del software, organizan las tareas implicadas en dicha actividad proponiendo distintos paradigmas de ejecución y por tanto de desarrollo del software. Tales paradigmas se conocen como los ciclos de vida del software.

Pressman (5 ed pp.15-16) entiende la construcción del software como un proceso dividido en tres fases en cada una de las cuales se responde a alguna de las preguntas que un proceso de ingeniería debe plantear y resolver:

- ¿Cuál es el problema a resolver?
- ¿Cuáles son las características de la entidad que se utiliza para resolver este problema?
- ¿Cómo se realizará la entidad?
- ¿Qué enfoque se va a utilizar para no contemplar los errores que se cometieron en el diseño y en la construcción de la entidad?
- ¿Cómo se sostendrá la entidad cuando los usuarios soliciten correcciones, adaptaciones y mejoras de la entidad?

Estas fases serían la de definición, la de desarrollo y la de mantenimiento

Fase de definición.

La fase de definición se centra en el **qué**. Durante esta fase, se intenta identificar: ¿qué información es la que tiene que ser procesada?; ¿qué función y rendimiento son los que se esperan?; ¿qué restricciones de diseño existen?; ¿qué interfaces deben utilizarse?; ¿qué sistema operativo y soporte hardware van a ser utilizados?; ¿qué criterios de validación se necesitan para conseguir que el sistema final sea correcto?

En esta fase se pueden identificar dos actividades: *Análisis del sistema* que define el papel de cada elemento relacionado con el sistema informático que se pretende desarrollar, precisando cuál es el papel del software dentro de ese sistema. *Análisis de requisitos del software* que proporciona el ámbito del software, su relación con el resto de componentes del sistema, pero antes de empezar a desarrollar es necesario hacer una definición más detallada de la función del software. La comprensión del producto a desarrollar, alcanzado en estas actividades, es básica para la realización de una buena *Planificación*, con la que organizar las tareas que se llevarán a cabo en la realización del proyecto.

Existen dos formas de realizar el análisis y refinamiento de los requisitos del software. Por una parte, se puede hacer un análisis formal del ámbito de la información para establecer modelos del flujo y la estructura de la información. Luego se amplían unos modelos para convertirlos en una especificación del software. La otra alternativa consiste en construir un prototipo del software, que será evaluado por el cliente para intentar consolidar los requisitos. Los requisitos de rendimiento y las limitaciones de recursos se traducen en directivas para la fase de diseño.

El análisis y definición de los requisitos es una tarea que debe llevarse a cabo conjuntamente por el desarrollador de software y por el cliente. La especificación de requisitos del software es el documento que se produce como resultado de esta etapa.

Fase de desarrollo

Durante el desarrollo un ingeniero de software intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura software, cómo han de implementarse los detalles procedimentales, cómo han de caracterizarse los interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de realizarse la prueba. En este caso se definen también tres actividades que serían el diseño del software, la codificación y las pruebas.

Fase de mantenimiento

Esta fase se centra en el cambio asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. En esta fase se definen las siguientes actividades

Corrección: asociada a cambios debidos a errores detectados por el cliente en la aplicación cuando ésta ya se encuentra operativa. *Adaptación*: cuando se introducen modificaciones en el programa original para adaptarlo a cambios en su entorno original. *Mejora*: en esta actividad el usuario descubre funcionalidades adicionales que quiere incorporar y que llevan al sistema más allá de sus especificaciones iniciales y la *Prevención* que busca mejorar el programa a fin de que sea más fácil mejorarlo, adaptarlo o corregir sus defectos.

Además estas actividades desarrolladas a lo largo de las tres fases se complementan con un número de actividades protectoras. Seguimiento y control del proyecto de software, revisiones técnicas formales, garantía de calidad del software, gestión de configuración del software, preparación y producción de documentos, gestión de reutilización, mediciones y gestión de riesgos.

2.2.1.- La norma IEEE 1074 [Piattini 2003]

Este estándar proporciona el conjunto de actividades que constituyen los procesos que son obligatorios para el desarrollo y mantenimiento de software. El estándar está organizado en 17 procesos, que comprenden un total de 65 actividades.

Los procesos se dividen en cuatro secciones lógicas, como se muestra en la Figura 2.2.

1. Proceso de Modelo del Ciclo de Vida Software (véase apartados 1.3.2 en adelante). Existen muchas variables que afectan a la selección por parte de una empresa de un modelo de ciclo de vida software, como por ejemplo el tipo de producto (interactivo, batch, procesamiento de transacciones, etc.), restricciones. Aunque el estándar no establece ni define un ciclo de vida específico o sus metodologías subyacentes, sí requiere que se seleccione y utilice un modelo de ciclo de vida.

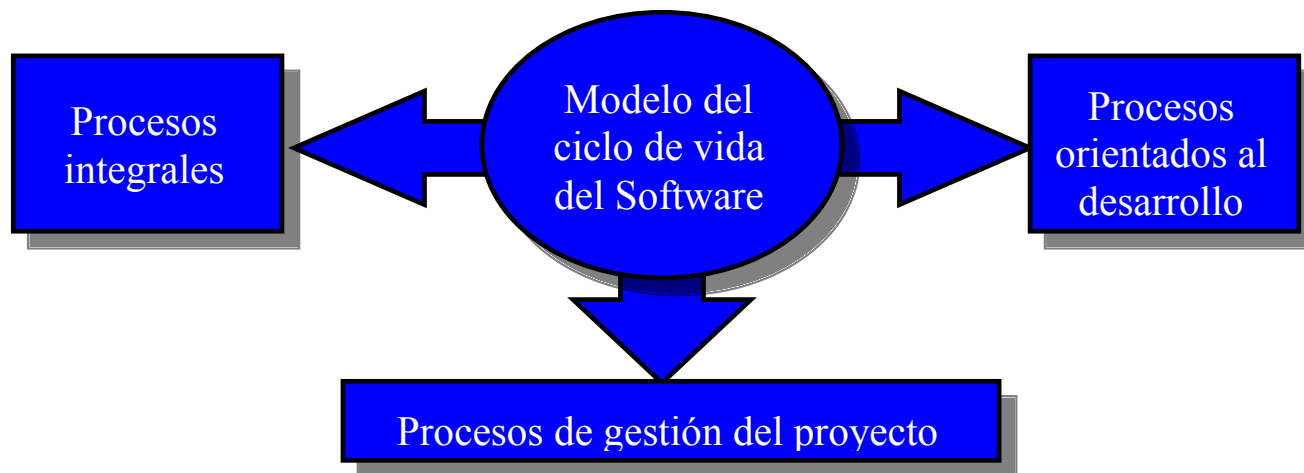


Figura 2.1.- Secciones lógicas para los procesos en el estándar IEEE 1074

Este proceso proporciona las actividades que se necesitan para identificar los modelos de ciclo de vida software candidatos y para seleccionar aquel modelo que se vaya a utilizar. Es posible que el mismo modelo de ciclo de vida se pueda aplicar a diferentes proyectos (por ejemplo, cascada); sin embargo aunque todos los proyectos sigan el modelo en cascada, cada proyecto deberá tener su propio ciclo de vida software en función del conjunto de actividades concretas que se vayan a realizar.

2. **Procesos de Gestión del Proyecto.** Son los procesos que inician, supervisan y controlan los proyectos software a lo largo del ciclo de vida software. Está formado por el Proceso de Iniciación del Proyecto, el Proceso de Supervisión y Control del Proyecto, y el Proceso de Gestión de la Calidad Software
 - El Proceso de Iniciación del Proyecto consiste en aquellas actividades que crean y mantienen el marco del proyecto. Es decir, se crea el ciclo de vida software del proyecto y se establecen los planes para gestionar el proyecto.
 - El Proceso de Supervisión y Control del Proyecto es un proceso iterativo de seguimiento, informes y gestión de costes, calendarios, problemas y rendimiento de un proyecto a lo largo de su ciclo de vida. El progreso de un proyecto se revisa y mide en los hitos establecidos en el plan del proyecto software,
 - El Proceso de Gestión de la Calidad Software se utiliza para tratar la planificación y administración del programa de Aseguramiento de la Calidad Software. Además, trata la satisfacción del cliente y los programas internos de mejora de la calidad.

Las actividades dentro del Proceso de Supervisión y Control del Proyecto, y del Proceso de Gestión de la Calidad Software, se realizan durante toda la vida del proyecto para asegurar el nivel apropiado de gestión del proyecto y de cumplimiento con las actividades obligatorias.

3. **Procesos Orientados al Desarrollo.** Comprenden los procesos que se realizan antes (Procesos de Pre-Desarrollo), durante (Procesos de Desarrollo) y después (Procesos de Post-Desarrollo) del desarrollo software.

Los Procesos de Pre-Desarrollo incluyen el Proceso de Exploración del Concepto y el Proceso de Asignación del Sistema.

- El Proceso de Exploración del Concepto examina los requisitos a nivel del sistema y produce un Informe de Necesidades que inicia el Proceso de Asignación del Sistema o el Proceso de Requisitos. Incluye la identificación de una idea o necesidad, su evaluación y refinamiento, y, una vez obtenidos los límites, la generación de un Informe de Necesidades para desarrollar el sistema.
- El Proceso de Asignación del Sistema realiza la asignación de requisitos al software y al hardware. Este proceso no se aplica cuando el proyecto sólo requiere desarrollo software.

Los Procesos de Desarrollo incluyen el Proceso de Requisitos, el Proceso de Diseño y el Proceso de Implementación.

- El Proceso de Requisitos incluye aquellas actividades dirigidas al desarrollo de los requisitos software. En el desarrollo de un sistema que contenga tanto componentes hardware como software, el Proceso de Requisitos sigue al desarrollo de los requisitos totales del sistema y a la asignación funcional de

dichos requisitos del sistema al hardware y al software.

- Durante el Proceso de Diseño se toman las principales decisiones para determinar la estructura del sistema. El objetivo del Proceso de Diseño es desarrollar una representación coherente y bien organizada del sistema software que cumpla los Requisitos Software. El Proceso de Diseño proyecta el ‘qué hacer’ de las especificaciones de requisitos en el ‘cómo hacerlo’ de las especificaciones de diseño. El diseño arquitectónico se centra en las funciones y estructura de los componentes software que comprenden el sistema software. El diseño detallado se centra en las estructuras de datos y algoritmos que se utilizan dentro de cada componente software.
- El Proceso de Implementación transforma la representación del diseño detallado de un producto software en una conversión del lenguaje de programación. Este proceso produce el código fuente, la base de datos y la documentación que constituye la manifestación física del diseño. Además, se integran el código y la base de datos. La salida de este proceso debe ser el objeto de todas las pruebas y validaciones subsiguientes.

Los Procesos de Post-Desarrollo incluyen el Proceso de Instalación, el Proceso de Explotación y Soporte, el Proceso de Mantenimiento y el Proceso de Retirada.

- El Proceso de Instalación consiste en el transporte e instalación de un sistema software desde el entorno de desarrollo hasta el entorno objeto. Incluye las modificaciones de software necesarias, la verificación en el entorno objeto y la aceptación del cliente. Durante este proceso, el software a entregar se instala, verifica y supervisa. Este esfuerzo culmina con la aceptación formal del cliente.
 - El Proceso de Explotación y Soporte implica la operación del usuario del sistema y el soporte continuo. El soporte incluye proporcionar asistencia técnica, consulta del usuario y registro de las peticiones de soporte del usuario.
 - El Proceso de Mantenimiento está relacionado con la resolución de los errores y fallos del software.
 - El Proceso de Retirada implica la eliminación de un sistema existente de su soporte o uso activo por cese de su explotación o soporte, o bien por su sustitución por un nuevo sistema o una nueva versión actualizada del sistema existente.
4. Procesos Integrales. Son aquellos procesos que se necesitan para completar con éxito las actividades de un proyecto. Estos procesos se utilizan para asegurar la terminación y calidad de las funciones del proyecto. Está formado por el Proceso de Verificación y Validación, el Proceso de Gestión de la Configuración Software, el Proceso de Desarrollo de la Documentación y el Proceso de Formación.
- El Proceso de Verificación y Validación incluye la planificación y realización de las tareas de Verificación (revisiones, auditorías de configuración y auditorías de calidad) y de Validación (todas las fases de pruebas) que se realizan a lo largo del ciclo de vida software para asegurar que se satisfacen todos los requisitos.
 - El Proceso de Gestión de Configuración Software identifica los elementos en un proyecto de desarrollo software y proporciona el control de los elementos identificados y la generación de informes de estado para que la dirección tenga visibilidad del ciclo de vida software.

- El Proceso de Desarrollo de la Documentación es el conjunto de actividades que planifican, diseñan, implementan, editan, producen, distribuyen y mantienen aquellos documentos necesitados por los desarrolladores y usuarios.
- El Proceso de Formación es esencial para los desarrolladores, plantilla de soporte técnico y clientes. El desarrollo de productos software de calidad depende ampliamente del conocimiento y habilidad de las personas.

2.2.2.- Norma ISO 12207 – 1 [Piattini 1996]

La norma ISO 12207 –1 define una serie de actividades que se realizan en la construcción del software agrupadas en cinco procesos principales, ocho de soporte y cuatro más de la organización. La norma no fomenta sin embargo ningún modelo concreto de ciclo de vida, gestión del software o método de ingeniería ni prescribe ni cómo realizar las actividades ni cómo organizarlas.

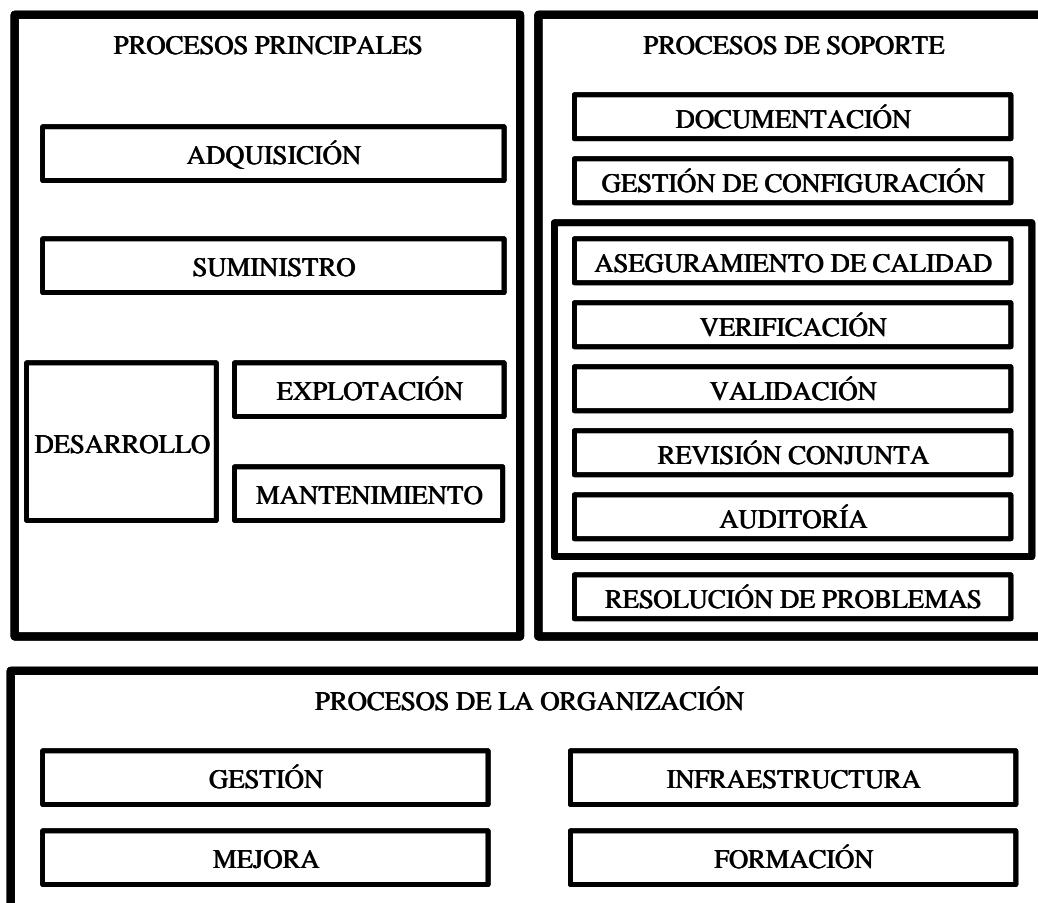


Figura 2.2.- Agrupación de procesos de la norma ISO 12207 - 1

2.2.2.1.- *Procesos principales*

Los procesos principales son aquellos que resultan útiles a las personas que inician o realizan el desarrollo, la explotación o el mantenimiento del software durante su ciclo de vida. Estas personas son los compradores, los suministradores, el personal de desarrollo, los usuarios y el personal de mantenimiento del software. Los procesos principales son:

2.2.2.1.1.- Proceso de adquisición

Este proceso contiene las actividades y las tareas que el comprador, el cliente o el usuario

realizan para adquirir un sistema o un producto software. También incluye la preparación y publicación de una solicitud de ofertas, la selección del suministrador del software y la gestión de los procesos desde la adquisición hasta la aceptación del producto.

2.2.2.1.2.- Proceso de suministro

Este proceso contiene las actividades y tareas que el suministrador realiza. Se inicia con la decisión de preparar una propuesta para responder a una petición de un comprador o por la firma de un contrato con el comprador para proporcionarle un producto software. También trata la identificación de los procedimientos y de los recursos necesarios para gestionar y garantizar el éxito del proyecto, incluyendo el desarrollo de los planes del proyecto y la ejecución de dichos planes hasta la entrega del producto software al comprador.

2.2.2.1.3.- Proceso de desarrollo

Este proceso contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas e instalación y aceptación. Debido al interés que tiene este proceso, se resumen a continuación sus principales actividades:

- Análisis de requisitos del sistema: se especifican los requisitos del sistema, incluyendo las funciones y las capacidades que debe incluir, los requisitos de seguridad, de interacción hombre-máquina, de interfaces, de operaciones y de mantenimiento, las restricciones aplicables al diseño y los requisitos para su aceptación.
- Diseño de la arquitectura del sistema: se establece la arquitectura de alto nivel, la cual identificará los principales componentes de hardware y de software, así como las operaciones manuales del sistema.
- Análisis de los requisitos del software: se establecen y se documentan dichos requisitos, incluyendo la especificación de las características de calidad que debe cumplir el sistema, tales como: especificaciones funcionales y de la capacidad (por ejemplo, el rendimiento del sistema o aplicación, sus características físicas, etc.), interfaces externas, requisitos de aceptación, especificaciones de seguridad y protección (de la información, de daños personales, etc.), especificaciones de interacción hombre-máquina (tales como, operaciones manuales, restricciones personales, por ejemplo «sólo los directores podrán acceder a la aplicación X», etc.), definición de los datos que se van a manejar y requisitos de bases de datos (por ejemplo, la aplicación se hará con un gestor de bases de datos relacional), requisitos de instalación y de aceptación del software entregado (por ejemplo, la aplicación se ha de instalar en un entorno PC, y sólo se aceptará si el tiempo de respuesta para la operación X es menor de dos segundos), requisitos de explotación y de ejecución, y requisitos de mantenimiento del usuario.
- Diseño de la arquitectura del software: el desarrollador debe transformar los requisitos del software en una arquitectura o estructura de alto nivel que identifica sus componentes principales. También se elabora una versión preliminar de los manuales de usuario, se definen y documentan los requisitos que deben cumplir las pruebas de estos componentes y se planifica la integración del software.
- Diseño detallado del software: se realiza un diseño detallado para cada componente software, incluidas las bases de datos. Así mismo se actualizan los manuales de usuario, se definen y documentan los requisitos que deben cumplir las pruebas de estos componentes detallados y se planifican las pruebas unitarias del software.
- Codificación y prueba del software: se desarrollan y se documentan los distintos componentes software y las bases de datos. Posteriormente se prueba cada uno de los mismos para asegurar que satisfacen los requisitos. Además, se actualizan los manuales de

usuario.

- Integración del software: se integran los componentes del software y se prueban según sea necesario. También se actualizan los manuales de usuario.
- Prueba del software: el desarrollador lleva a cabo la prueba de cualificación en función de los requisitos especificados.
- Integración del sistema: se integran los elementos software y hardware junto con las operaciones manuales.
- Prueba del sistema: análoga a la del software, pero se llevará a cabo de acuerdo con los requisitos de cualificación especificados para el sistema (el software integrado con el resto de componentes del mismo: hardware, componentes físicos, etc.).
- Instalación del software en el entorno de explotación final donde vaya a funcionar. Cuando el software nuevo reemplace a un sistema existente, es recomendable mantener ambos sistemas en funcionamiento paralelo durante un período razonable de tiempo para comprobar el funcionamiento correcto del nuevo sistema.
- Soporte del proceso de aceptación del software: el desarrollador debe dar su apoyo a la revisión de aceptación y a la prueba del software por parte del comprador.

2.2.2.1.4.- Proceso de explotación

Este proceso incluye la explotación del software y el soporte operativo a los usuarios. Debido a que esta explotación está integrada en la del sistema, las actividades y tareas de este proceso se aplican al sistema completo. También se denomina proceso de operación.

2.2.2.1.5.- Proceso de mantenimiento

Este proceso aparece cuando el software necesita modificaciones, ya sea en el código o en la documentación asociada, debido a un error, una deficiencia, un problema o la necesidad de mejora o adaptación. El objetivo es modificar el software existente manteniendo su consistencia. Este proceso puede incluir también las actividades de migración a un nuevo entorno operativo y las de retirada del software.

2.2.2.2.- Procesos de soporte

Estos procesos sirven de apoyo al resto y se aplican en cualquier punto del ciclo de vida. Los procesos de soporte son los siguientes:

2.2.2.2.1.- Proceso de documentación

Este proceso registra la información producida por un proceso o actividad del ciclo de vida. Incluye todo el conjunto de actividades que permiten planificar, diseñar, desarrollar, producir, editar, distribuir y mantener los documentos necesarios para todas las personas involucradas: directores, ingenieros, personal de desarrollo, usuarios del sistema o software, etc.

2.2.2.2.2.- Proceso de gestión de la configuración

Este proceso aplica ciertos procedimientos administrativos y técnicos durante todo el ciclo de vida del sistema para:

- Identificar, definir y establecer la línea base de los elementos de configuración del software de un sistema.
- Controlar las modificaciones (por ejemplo, la modificación X al programa Y fue hecha por la persona Z) y las versiones de los elementos (por ejemplo, la última versión del programa X es la 1.4).

- Registrar e informar sobre el estado de los elementos y las peticiones de modificación.
- Asegurar la compleción, la consistencia y la corrección de los elementos.
- Controlar el almacenamiento, la manipulación y la entrega de los elementos.

2.2.2.2.3.- Proceso de aseguramiento de la calidad

Este proceso aporta una confianza en la que los procesos y los productos software del ciclo de vida cumplen con los requisitos especificados y se ajustan a los planes establecidos. El aseguramiento de la calidad puede ser interno o externo dependiendo de si la calidad del producto o proceso debe demostrarse a los directivos del proveedor o del comprador. El aseguramiento de la calidad puede utilizar los resultados de otros procesos de apoyo, como la verificación, la validación, las revisiones conjuntas, las auditorías y las actividades de resolución de problemas.

2.2.2.2.4.- Proceso de verificación

Este proceso determina si los requisitos de un sistema o del software están completos y son correctos, y si los productos software de cada fase del ciclo de vida cumplen los requisitos o condiciones impuestos sobre ellos en las fases previas (por ejemplo, si el código es coherente con el diseño). Este proceso puede ser responsabilidad de una organización de servicios, en cuyo caso, se denomina proceso de verificación independiente.

2.2.2.2.5.- Proceso de validación

Este proceso sirve para determinar si el sistema o software final cumplen con los requisitos previstos para su uso. Al igual que el anterior, este proceso puede ser ejecutado por una organización de servicios, en cuyo caso se denomina proceso de validación independiente.

2.2.2.2.6.- Proceso de revisión conjunta

Este proceso sirve para evaluar el estado del software y sus productos en una actividad del ciclo de vida o una fase de un proyecto. Las revisiones conjuntas se celebran tanto a nivel de gestión como a nivel técnico del proyecto a lo largo de todo su ciclo de vida.

2.2.2.2.7.- Proceso de auditoría

Este proceso permite determinar, en los hitos predeterminados, si se han cumplido los requisitos, los planes y el contrato.

2.2.2.2.8.- Proceso de resolución de problemas

Este proceso permite analizar y eliminar los problemas (disconformidades con los requisitos o el contrato) descubiertos durante el desarrollo, la explotación, el mantenimiento u otro proceso. El objetivo es aportar un medio oportuno y documentado para asegurar que todos los problemas descubiertos se analizan y solucionan.

2.2.2.3.- Procesos de la organización

Estos procesos los emplea una organización para llevar a cabo funciones tales como la gestión, la formación del personal o la mejora del proceso. Ayudan a establecer, implementar y mejorar la organización consiguiendo una organización más efectiva. Se llevan a cabo normalmente a nivel organizativo, fuera del ámbito de proyectos y contratos específicos.

2.2.2.3.1.- Proceso de gestión

Este proceso contiene las actividades y las tareas genéricas que puede emplear cualquier organización que tenga que gestionar su(s) proceso(s), por lo que incluye actividades como la planificación, el seguimiento y control, y la revisión y evaluación.

2.2.2.3.2.- Proceso de infraestructura

Este proceso establece la infraestructura necesaria para cualquier otro proceso, lo que incluye hardware, software, herramientas, técnicas, normas e instalaciones para el desarrollo, la explotación o el mantenimiento.

2.2.2.3.3.- Proceso de mejora

Este proceso sirve para establecer, valorar, medir, controlar y mejorar los procesos del ciclo de vida del software.

2.2.2.3.4.- Proceso de formación

Este proceso sirve para proporcionar y mantener al personal formado, lo que incluye el desarrollo del material de formación, así como la implementación del plan de formación.

2.2.3.- La norma ISO/IEC TR 15504-2 [Piattini 2003]

Los procesos de esta norma están fuertemente alineados con la anterior y de hecho ésta se puede considerar una extensión de la anterior en el sentido de que incorpora todo lo dicho para la anterior y añade o amplía procesos. Esta norma tiene además un carácter bidimensional ya que además de definir los procesos también expresa la capacidad que una empresa ha logrado en el desarrollo de dicho proceso. Esta dimensión será analizada con mayor nivel de detalle cuando se trate el CMM y el CMMI.

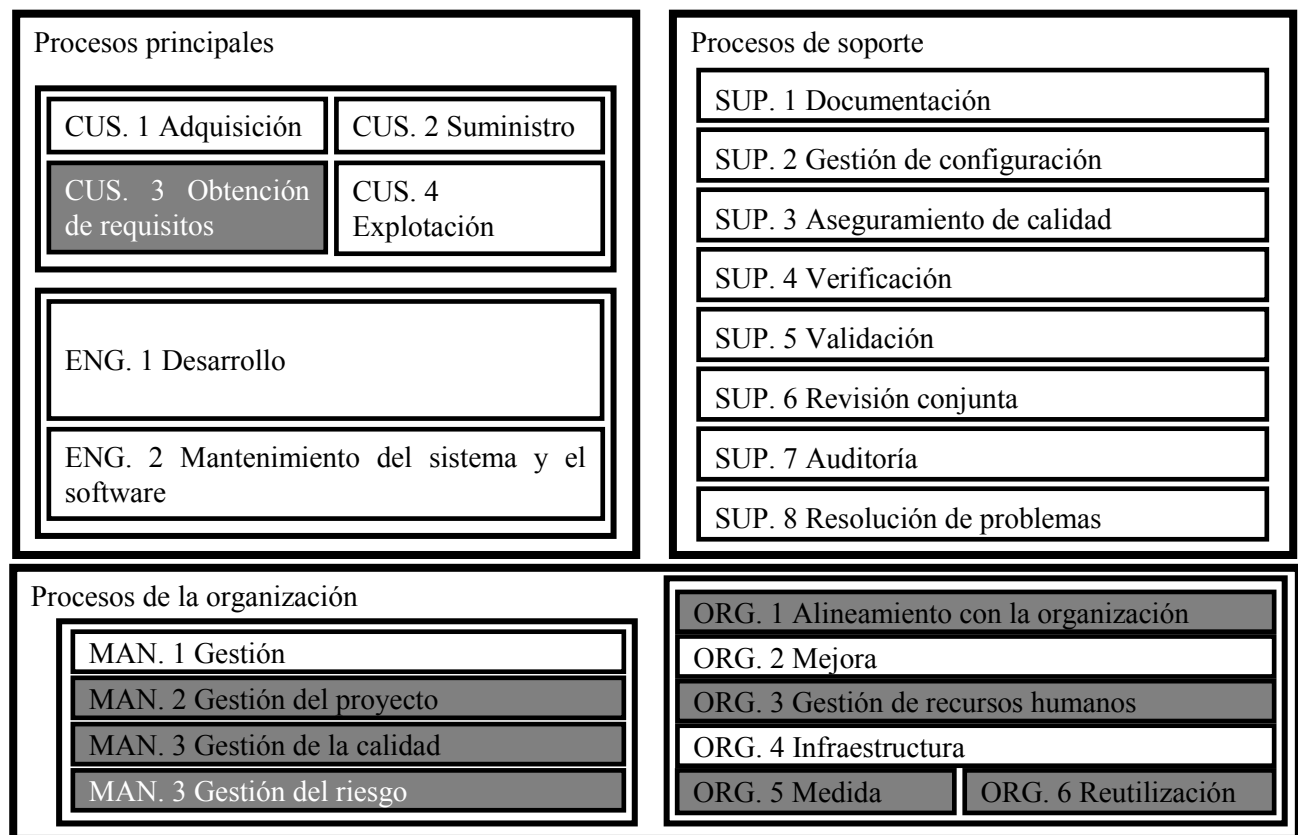


Figura 2.3.- Procesos de la ISO/IEC 15504-2 y su alineamiento con la ISO 12207-1

Los procesos que esta norma añade con respecto a la anterior son:

El proceso de obtención de requisitos, que se añade a los principales y tiene como objetivo reunir, procesar y seguir la evolución de las necesidades y requisitos del cliente a lo largo

de toda la vida del producto software y/o servicio, así como establecer una línea de base de requisitos que permita definir los entregables software requeridos.

Proceso de Alineamiento de la Organización. Se añade a los procesos de la organización y su propósito es asegurar que los individuos en la organización comparten una visión, cultura y comprensión común de los objetivos del negocio para autorizarlos a funcionar de forma efectiva. Aunque la reingeniería del negocio y la Gestión de la Calidad Total tienen un alcance mucho mayor que el de proceso software, la mejora de proceso software ocurre en un contexto de negocio y, para tener éxito, debe tratar los objetivos del negocio.

Proceso de Gestión de Recursos Humanos. Se añade a los procesos de la organización y su propósito es proporcionar a la organización y a los proyectos con individuos que posean las habilidades y el conocimiento para realizar sus roles de forma efectiva y para trabajar juntos como un grupo.

Proceso de Medida. Se añade a los procesos de la organización y su propósito es recoger y analizar los datos relativos a los productos desarrollados y a los procesos implementados dentro de la unidad organizativa, para soportar la gestión efectiva de los procesos y para demostrar de forma objetiva la calidad de los productos.

Proceso de Reutilización. Se añade a los procesos de la organización y su propósito es promover y facilitar la reutilización de productos de trabajo software nuevos y existentes desde una perspectiva de la organización y del producto/proyecto.

El proceso de gestión de los procesos de la organización se descompone ahora en:

Proceso de Gestión (proceso básico). El propósito del Proceso de Gestión es organizar y controlar la iniciación y la realización de cualquier proceso o función dentro de la organización para lograr sus objetivos y los objetivos de negocio de la organización de manera efectiva.

Proceso de Gestión del Proyecto (proceso nuevo). El propósito del Proceso de Gestión del Proyecto (véase el Capítulo 5) es identificar, establecer, coordinar y controlar las actividades, tareas y recursos necesarios para que un proyecto produzca un producto y/o servicio que cumplan los requisitos.

Proceso de Gestión de la Calidad (proceso nuevo). El propósito del Proceso de Gestión de la Calidad (véase el Capítulo 12) es controlar la calidad de los productos y/o servicios del proyecto y asegurar que satisfacen al cliente. El proceso implica el establecimiento de un enfoque sobre el control de la calidad del producto y del proceso tanto a nivel del proyecto como a nivel de la organización.

Proceso de Gestión del Riesgo (proceso nuevo). El propósito del Proceso de Gestión del Riesgo es identificar y reducir continuamente los riesgos en un proyecto a lo largo de su ciclo de vida. El proceso implica el establecimiento de un enfoque en el control de los riesgos tanto a nivel del proyecto como a nivel de la organización.

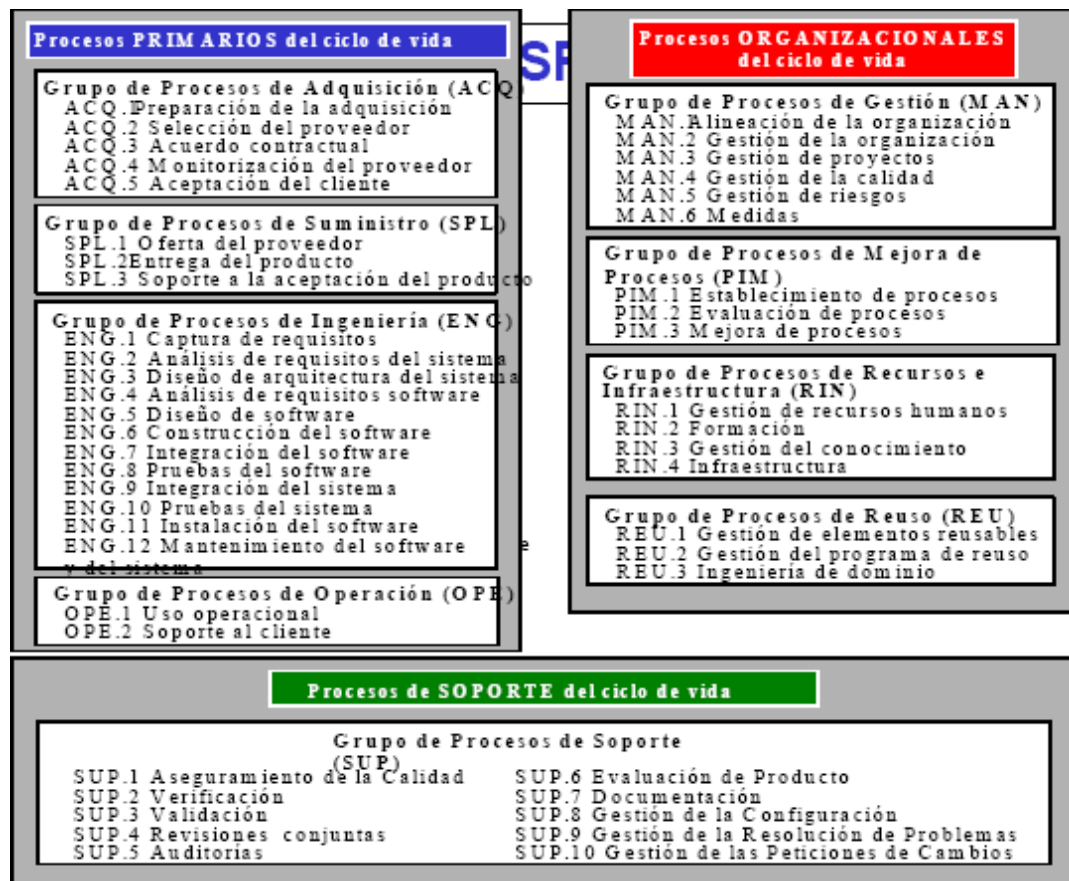


Figura 2.4. Procesos de la norma ISO/IEC 15504-2 (rev. 2003)

2.3.- Evaluación del proceso software.

La crisis del software y los problemas generados por el software heredado, planteados en temas anteriores, dan cuenta detallada de los problemas generados por desarrollar software sin seguir un proceso o una metodología orientados a la calidad. El desarrollo de software ad hoc supone un riesgo importante para las empresas que no pueden conocer el progreso de sus proyectos ni predecir su calidad, que en general será baja. Como consecuencia los presupuestos del proyecto no se cumplen, los requisitos o las restricciones del sistema no se alcanzan y el proyecto fracasa con el efecto inmediato de la insatisfacción del cliente.

De todo lo anterior se dedujo un interés en las empresas por encontrar y seguir algún proceso que les garantizara los resultados. A pesar de este interés la aplicación de los procesos y su adaptación a las empresas y sus políticas no resulta trivial. En muchos casos las empresas conocen los modelos y reconocen su utilidad pero los aplican de forma incompleta e inconsistente según sus necesidades más acuciantes. Como consecuencia en muchos casos las empresas llegan a desconocer en qué punto se encuentran sus procesos de forma que sus objetivos no se centran en el proceso sino en saber si lo siguen y adaptarlo de forma consistente a sus propias estrategias.

Surgen de este modo una serie de modelos orientados a guiar a las empresas en su evolución desde su estado actual, que debe ayudar a determinar el modelo, hasta uno evolucionado que garantice la optimización en la producción de software de calidad. El modelo guiará a la empresa en la mejora del proceso y, según su evolución, hasta le permitirá decidir el camino que prefiera seguir. La relación entre los procesos y los métodos de evaluación se muestra en la siguiente figura.

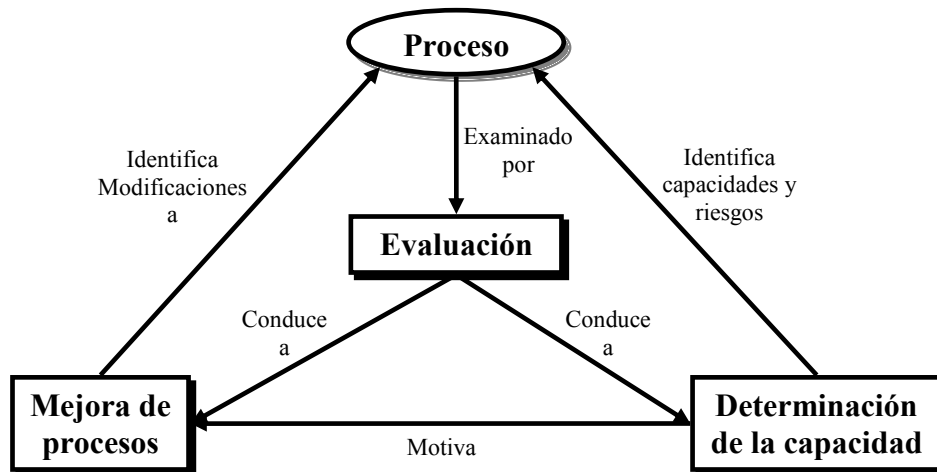


Figura 2.5. Relación con los procesos de evaluación

2.3.1.- Integración del Modelo de Capacidad de Madurez (CMMI)

El modelo CMM ha evolucionado actualmente al CMMI (Capability Maturity Model Integration) [Pressman06]. Este modelo presenta en contra de lo que sucedía con su primera versión dos formas diferentes: un modelo continuo y un modelo discreto. En el modelo continuo se evalúa las distintas áreas del proceso en 6 niveles de capacidad. El CMMI v1.2 propone 22 áreas del proceso (PA) divididas en 4 categorías (ver tabla 2.6) cada una de las cuales puede estar en cualquiera de los 6 niveles de capacidad mencionados y que detallamos a continuación:

Nivel 0: Incompleto. El área del proceso (por ejemplo, la gestión de requisitos) aún no se realiza o todavía no alcanza todas las metas y objetivos definidos para el nivel 1 de capacidad.

Nivel 1: Realizado. Todas las metas específicas del área del proceso (como las definió el CMMI) han sido satisfechas. Las tareas de trabajo requeridas para producir el producto específico han sido realizadas.

Nivel 2: Administrado. Todos los criterios del nivel 1 han sido satisfechos. Además, todo el trabajo asociado con el área de proceso se ajusta a una política organizacional definida; toda la gente que ejecuta el trabajo tiene acceso a recursos adecuados para realizar su labor; los clientes están implicados de manera activa en el área de proceso, cuando esto se requiere; todas las tareas de trabajo y productos están "monitorizados, controlados y revisados; y son evaluados de acuerdo a la descripción del proceso" [CMM02].

Nivel 3: Definido. Todos los criterios del nivel 2 se han cumplido. Además, el proceso está "adaptado al conjunto de procesos estándar de la organización, de acuerdo con las políticas de adaptación de esta misma, y contribuye a la información de los productos del trabajo, mediciones y otras mejoras del proceso para los activos del proceso organizacional" [CMM02].

Nivel 4: Administrado en forma cuantitativa. Todos los criterios del nivel 3 han sido cumplidos. Además, el área del proceso se controla y mejora mediante mediciones y evaluación cuantitativa. "Los objetivos cuantitativos para la calidad y el desempeño del proceso están establecidos y se utilizan como un criterio para administrar el proceso" [CMM02].

Nivel 5: Mejorado. Todos los criterios del nivel 4 han sido satisfechos. Además, el área del proceso "se adapta y mejora mediante el uso de medios cuantitativos (estadísticos) para conocer las necesidades cambiantes del cliente y mejorar de manera continua la eficacia del área del proceso que se está considerando" [CMM02].

Categoría 1	The Process Management process areas of CMMI are as follows
área 1	Organizational Process Focus
área 2	Organizational Process Definition
área 3	Organizational Training
área 4	Organizational Process Performance
área 5	Organizational Innovation and Deployment
Categoría 2	The Project Management process areas of CMMI are as follows
área 6	Project Planning
área 7	Project Monitoring and Control
área 8	Supplier Agreement Management
área 9	Integrated Project Management + IPPD (or Integrated Project Management)
área 10	Risk Management
área 11	Quantitative Project Management
Categoría 3	The Engineering process areas of CMMI are as follows
área 12	Requirements Development
área 13	Requirements Management
área 14	Technical Solution
área 15	Product Integration
área 16	Verification
área 17	Validation
Categoría 4	The Support process areas of CMMI are as follows
área 18	Configuration Management
área 19	Process and Product Quality Assurance
área 20	Measurement and Analysis
área 21	Decision Analysis and Resolution
área 22	Causal Analysis and Resolution

Figura 2.6. Áreas de proceso por categoría para el CMMI continuo.

El CMMI define cada área del proceso en función de "metas específicas" (ME) y de las "prácticas específicas" (PE) requeridas para alcanzar dichas metas. Las metas específicas establecen las características que deben existir para que las actividades implicadas por un área de proceso sean efectivas. Las prácticas específicas convierten una meta en un conjunto de actividades relacionadas con el proceso.

Por ejemplo, para la planificación del proyecto (en la tabla PA 6) las metas específicas (ME) y sus prácticas específicas asociadas (PE) que se han definido son [CMM02]:

ME 1 Establecer estimaciones

- PE 1.1 -1 Estimar el alcance del proyecto.
- PE 1.2-1 Establecer estimaciones para los atributos del producto y las tareas del trabajo.
- PE 1.3- 1 Definir el ciclo de vida del proyecto.
- PE 1.4- 1 Determinar estimaciones de esfuerzo y costo.

ME 2 Desarrollar un plan de proyecto

- PE 2.1- 1 Establecer el presupuesto y el programa.
- PE 2.2- 1 Identificar los riesgos del proyecto.
- PE 2.3- 1 Planear la gestión de los datos.
- PE 2.4- 1 Planear los recursos del proyecto.
- PE 2.5- 1 Planear los conocimientos y habilidades que se requieren.
- PE 2.6- 1 Planear la participación del cliente.
- PE 2.7-1 Establecer el plan de proyecto.

ME 3 Comprometerse con la planificación.

- PE 3.1-1 Revisar los planes que afectan al proyecto.
- PE 3.2-1 Conciliar el trabajo y los niveles de recursos.
- PE 3.3-1 Comprometerse con la planificación.

Además de las metas y prácticas específicas, el CMMI también define una serie de cinco

metas genéricas (MG) y prácticas genéricas (PG) relacionadas con cada área del proceso. Cada una de las metas genéricas corresponde a uno de los cinco niveles de capacidad. Por lo tanto, para lograr un nivel de capacidad particular se debe alcanzar la meta genérica para ese nivel y las prácticas genéricas que corresponden a esa meta. Para ilustrar lo anterior, a continuación se enumeran las metas genéricas (MG) y las prácticas genéricas (PG) para el área del proceso de planeación del proyecto [CMM02]:

MG 1 Alcanzar las metas específicas

PG 1.1 Realizar prácticas base.

MG 2 institucionalizar un proceso de gestión

PG 2.1 Establecer una política organizacional.

PG 2.2 Planear el proceso.

PG 2.3 Proporcionar recursos.

PG 2.4 Asignar responsabilidades.

PG 2.5 Capacitar gente.

PG 2.6 Manejar configuraciones.

PG 2.7 Identificar y hacer participar a clientes.

PG 2.8 Monitorizar y controlar el proceso.

PG 2.9 Evaluar la adherencia de un modo objetivo.

PG 2.10 Revisar el estatus con un alto grado de gestión.

MG 3 Institucionalizar un proceso definido

PG 3.1 Establecer un proceso definido.

PG 3.2 Recolectar información de la mejoría.

MG 4 Institucionalizar un proceso manejado en forma cuantitativa

PG 4.1 Establecer objetivos cuantitativos para el proceso.

PG 4.2 Estabilizar el desempeño del subproceso.

MG 5 Institucionalizar un proceso de mejoramiento.

PG 5.1 Asegurar la mejora continua del proceso.

PG 5.2 Corregir las causas de los problemas desde la raíz

El modelo discreto del CMMI define las mismas áreas, metas y prácticas específicas del proceso que el modelo continuo. En él, sin embargo, sólo están definidas las Metas Generales 2 y 3 tal y como se han descrito para el modelo continuo. La principal diferencia es que el modelo discreto establece cinco niveles de madurez, en vez de cinco niveles de capacidad. Para lograr un nivel de madurez se deben conseguir metas y prácticas específicas relacionadas con un conjunto de áreas del proceso. Para el nivel de madurez 2 se deben cumplir además las metas generales 2 para cada una de las áreas de proceso de ese nivel y para el nivel 3 y subsiguientes se deben satisfacer también las metas generales 3 para cada área de proceso presente en el nivel. De esta forma el nivel de madurez 2 implica realizar las metas generales 2 para 7 áreas de proceso mientras que el nivel 3 implica cumplir las metas generales 2 y 3 para 18 áreas de proceso. Los niveles 4 y 5 sólo añaden respectivamente 2 áreas de proceso cada uno para las que, por supuesto, se deben cumplir las metas generales 2 y 3.

La relación de las áreas del proceso presentes en cada nivel se muestra en la figura 2.7

Nivel	Enfoque	Áreas del proceso
De optimización	Mejora continua del proceso	Innovación organizacional y despliegue. Análisis causal y resolución
Gestionado de modo cuantitativo	Gestión cuantitativa	Ejecución del proceso organizacional. Gestión cuantitativa del proyecto
Definido	Estandarización del proceso	Desarrollo de requisitos
		Solución técnica
		Integración del producto
		Verificación

		Validación
		Enfoque del proceso organizacional
		Definición del proceso organizacional
		Capacitación organizacional
		Gestión integrada del proyecto
		Gestión integrada del proveedor
		Gestión del riesgo
		Análisis y resolución de la decisión
		Ambiente organizacional para la integración
		Equipo integrado
Gestionado	Gestión básica del proyecto	Gestión de requisitos
		Planificación del proyecto
		Monitorización y control del proyecto
		Gestión de acuerdos del proveedor
		Medición y análisis
		Aseguramiento de la calidad del producto y del proceso
Ejecutado		Gestión de la configuración

Figura 2.7. PAs por nivel de madurez del modelo CMMI discreto.