

## CAPÍTULO 5. MODELO DE CAPACIDAD DE MADUREZ

Ya que el problema fundamental de las organizaciones de software es su inhabilidad para administrar sus procesos. El CMM para Software (CMM-SW) se convierte en una guía que nos ayudará a ganar el control sobre estos procesos y así desarrollar y mantener un mejor software. La meta a alcanzar será la evolución hacia una cultura de excelencia tanto en la Ingeniería como en la Administración de software.

El CMM incluye prácticas de planeación, ingeniería y administración de desarrollo y mantenimiento de software. Si se siguen estas prácticas aumentará la habilidad con que una organización podrá alcanzar metas como costo, programa, funcionalidad y calidad de producto.

El propósito de CMM es el guiar a las organizaciones en la selección de estrategias de mejora determinando la madurez del proceso actual e identificando los puntos importantes que se deben atacar para así mejorar tanto el proceso como la calidad del software.

Ahora, ¿por qué confiar en CMM? El modelo de capacidad de Madurez está basado en prácticas *reales*, refleja las *mejores* prácticas en el área, también refleja la necesidad de los individuos de llevar a cabo una *mejora* en el proceso de software, al igual que la *valoración* del proceso de software. Y para finalizar, CMM está documentado y es público [PAULK, 1994]

## 5.1 Antecedentes

Para poder entender el origen del CMM es necesario regresar a 1986 cuando el Software Engineering Institute (SEI) junto con MITRE Corporation buscaron mejorar el proceso de software y comenzaron a desarrollar un Marco de Trabajo que llamaron Proceso de Madurez. Éste está basado en el concepto de la Administración de la Calidad Total (TQM), el cual cuenta con cinco etapas evolutivas hacia una implementación de prácticas de calidad. CMM es una aplicación de TQM para software.

Pero no es sino hasta 1989 que W. Humphrey, el entonces Director del SEI, publica el libro *Administrando el Proceso de Software* (Managing the Software Process). En esta obra ya encontramos un Marco de Trabajo definido por cinco **niveles de madurez**. Este Marco presentaba dos métodos (Valoración del proceso de software y la evaluación de la capacidad del software) y un cuestionario de madurez para valorar la madurez del proceso de software.

Se involucran diferentes empresas y académicos relacionados al área de Ingeniería de Software y fue en 1991 cuando ese Marco de Trabajo evoluciona para convertirse en el Modelo de Capacidad de Madurez (CMM). Este modelo presenta un conjunto de prácticas divididas en dieciocho **Áreas Clave de Proceso** que han demostrado acrecentar la **capacidad** de los procesos de software. El CMM siguiendo su propia filosofía continúa y continuará evolucionando siempre hacia un mejoramiento continuo.

Acorde a este modelo, el control de procesos y la excelencia en Ingeniería y la Administración del Software podrán ser alcanzados, si se puede mejorar un conjunto finito de actividades que conformen la base central de un área clave. Al enfocarnos en

este pequeño conjunto de actividades y trabajando agresivamente para cubrirlas en su totalidad, la organización podrá mejorar de una manera estable y gradual en lo que respecta a sus procesos de software.

Es necesario entonces poder determinar el *grado de madurez* de los *procesos actuales* e identificar los *puntos clave* en donde se deberá enfocar la atención de mejora para así lograr por una parte calidad en el software y por la otra el mejoramiento de los procesos.

### **CMM es...**

- Una estrategia de mejora.
- Una señalización de deficiencias dentro de una organización.
- Una guía para poder avanzar hacia una cultura de calidad.

### **CMM no es...**

- Una solución rápida, sino gradual.
- Un *checklist* que puede ser utilizado en todos los ambientes, aunque las prácticas detalladas en el CMM sirven como guía para tomar decisiones.

## **5.2 Términos Importantes**

Como punto de partida se darán una serie de definiciones manejadas por el SEI [PAULK, 1994]:

¿Qué es un **proceso de software**?

Conjunto de actividades, métodos, prácticas y transformaciones para desarrollar y mantener software y productos asociados.

¿Qué es la **capacidad de un proceso**?

*El rango de resultados esperados que se pueden obtener tras seguir un proceso.*

¿Qué es la **madurez** de un proceso de software?

*Es el punto hasta el cual un determinado proceso es explícitamente definido, administrado, medido, controlado y efectivo.*

La madurez es el potencial de crecimiento de la capacidad del proceso, también nos indica tanto la riqueza del proceso de software de una organización, como la consistencia con la cual es aplicado en los proyectos a lo largo de la misma.

La madurez de un proceso implica que la capacidad del proceso de software ha crecido. *Específicamente debe ser: definido, documentado, entrenado, practicado, soportado, mantenido, controlado, verificado, validado, medido, y capaz de mejorar.*

¿Qué es un **Nivel de Madurez**?

*Plataforma bien definida desde la cual podremos obtener un proceso maduro de software.*

A medida que una organización de software adquiere madurez en su proceso de software, ésta lo institucionaliza a través de políticas, estándares y estructuras

organizacionales. La institucionalización conlleva a la construcción de una infraestructura y a una cultura corporativa que soporte los métodos, prácticas y procedimientos.

Estos niveles definen una escala para medir la madurez y evaluar la capacidad de los procesos de software. Los niveles ayudan a la empresa a dar prioridades en el esfuerzo de mejora.

#### ¿Qué es una **Actividad**?

*Cualquier paso o función que se realiza (mental o física) para alcanzar algún objetivo.*

Incluyendo todo el trabajo realizado para realizar las tareas del proyecto y la organización.

#### ¿Qué es un **Área Clave de Proceso** (Key Process Area; KPA)?

*Grupo de actividades relacionadas que cuando se llevan a cabo en conjunto alcanzan un conjunto de metas (consideradas importantes para aumentar la capacidad del proceso)*

Las KPAs son al proceso de software lo que los cimientos a una casa. Cada una de las dieciocho pertenece a uno y solo uno de los cinco niveles de madurez. Como resultado se obtiene un despliegue de los procesos de software que son efectivos, usables y aplicados constantemente a lo largo de la organización.

#### ¿Qué significa **Institucionalizar**?

*Es edificar una infraestructura y una cultura que soporte los métodos, las prácticas y los procesos para que estos sean la manera REAL de hacer negocios.*

La infraestructura es el marco de trabajo de una organización o sistema, incluyendo estructuras organizacionales, políticas, estándares, entrenamiento, facilidades y herramientas que apoyan el desempeño. Gracias a esta institucionalización se puede alcanzar una mejora continua.

Por último sería bueno identificar la diferencia entre un proceso maduro de uno eficaz. Los dos tipos de procesos son evaluados dentro de su propio contexto. La **madurez** del proceso se puede evaluar comparándolo a un modelo (ejemplo: CMM) La **efectividad** de un proceso se puede determinar solo con respecto a los objetivos de negocio de la organización. Cabe resaltar que la calidad de un producto de software está fuertemente determinado por la calidad del proceso implementado para su desarrollo y mantenimiento. Un proceso de software efectivo reúne tanto personas, herramientas y métodos en un ente integrado. [PAULK, 1994]

### **5.3 Estructura**

Pensemos en CMM como un Modelo basado en el conocimiento, en donde si uno no aprende a gatear jamás podrá llegar a correr. CMM cuenta con cinco niveles de Madurez. Niveles que son progresivos y no autónomos. Estos niveles están organizados conforme a su importancia (prioridad)

Una organización está formada por todo un equipo de personas. Y sólo involucrando e informando a cada una de ellas es que podremos transmitir, mejorar y aumentar el conocimiento ya adquirido.

Imaginemos un edificio (figura 5.1) Cada piso representa una plataforma bien definida (nivel de madurez). Al tener ambos pies en el piso podremos intentar alcanzar el siguiente. Es decir uno a la vez, sin saltos.

Ahora, los pisos de este edificio pueden ser muy altos por lo tanto necesitaremos de plataformas intermedias para poder ayudarnos a alcanzar el siguiente piso (nuestra próxima meta) A estas plataformas las llamaremos KPAs (nuestras metas intermedias)

Pero, ¿qué debemos hacer para poder subir estas plataformas? Necesitaremos de escalones (Prácticas Clave) para alcanzar cada una de esas plataformas intermedias. El número de escalones entre cada plataforma no es fijo. Esas Prácticas Clave podrían ser muchas así que por conveniencia las dividiremos en cinco grupos, de acuerdo a su objetivo. Estos grupos se llaman Características Comunes (common features) Los últimos dos elementos nos permitirán transmitir el logro de nuestro avance a todos nuestros viejos y nuevos compañeros. Y como resultado tendremos una institucionalización de actividades realmente útiles.

Veamos ahora la estructura de CMM. Partiendo desde el más alto nivel nos encontramos con los niveles de madurez. Estos nos indican la capacidad de nuestro proceso. Los niveles contienen áreas clave de proceso (KPAs) que nos permitirán alcanzar ciertas metas consideradas importantes para la mejora del proceso. Las KPAs se encuentran organizadas en cinco distintas características comunes, las cuales buscan la implementación o la institucionalización de las KPAs. Estas características contienen a su vez prácticas clave que describen la infraestructura o las actividades que se deben realizar para satisfacer a determinada KPA (figura 5.2)

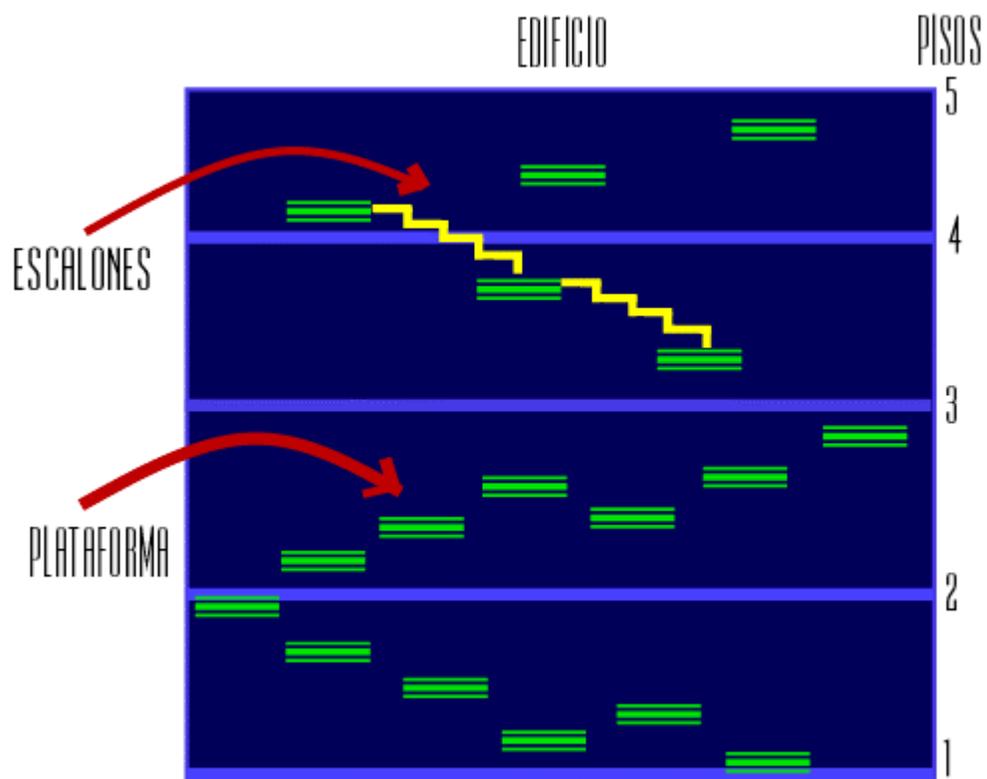


Figura 5.1 Semejanza de CMM con un edificio

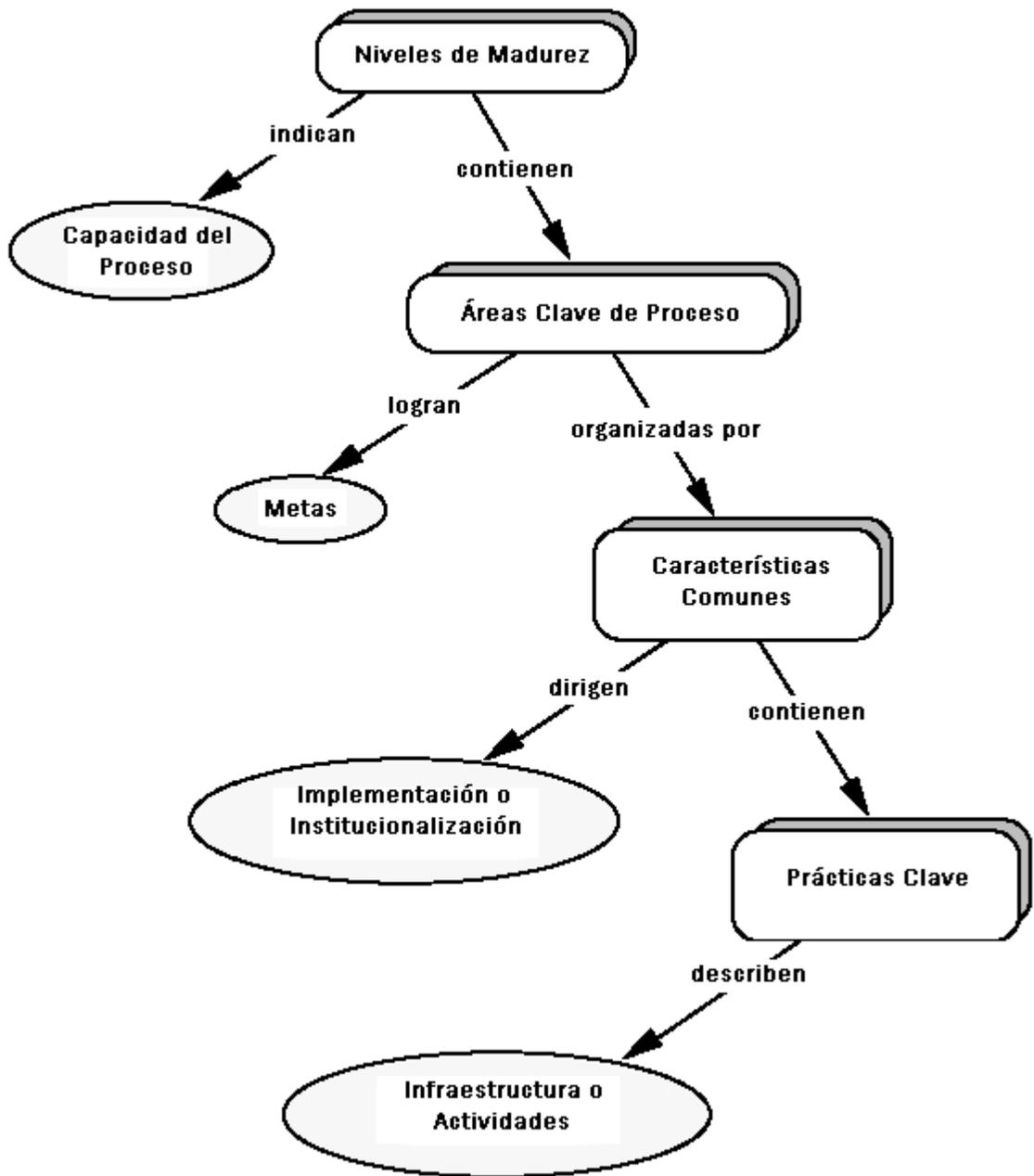


Figura 5.2 Estructura de CMM [SEI; 1994]

### 5.3.1 Niveles de Madurez

La estructura de más alto nivel del CMM son los cinco niveles de Madurez.

Recordemos la definición de Nivel de Madurez:

*Plataforma bien definida desde la cual podremos obtener un proceso **definido, administrado, medido, controlado y efectivo** de software.*

Cada uno de estos niveles nos indicará que tan capaz es un proceso. Gracias a esto podremos determinar el resultado del próximo proyecto que la empresa decida realizar, figura 5.3.

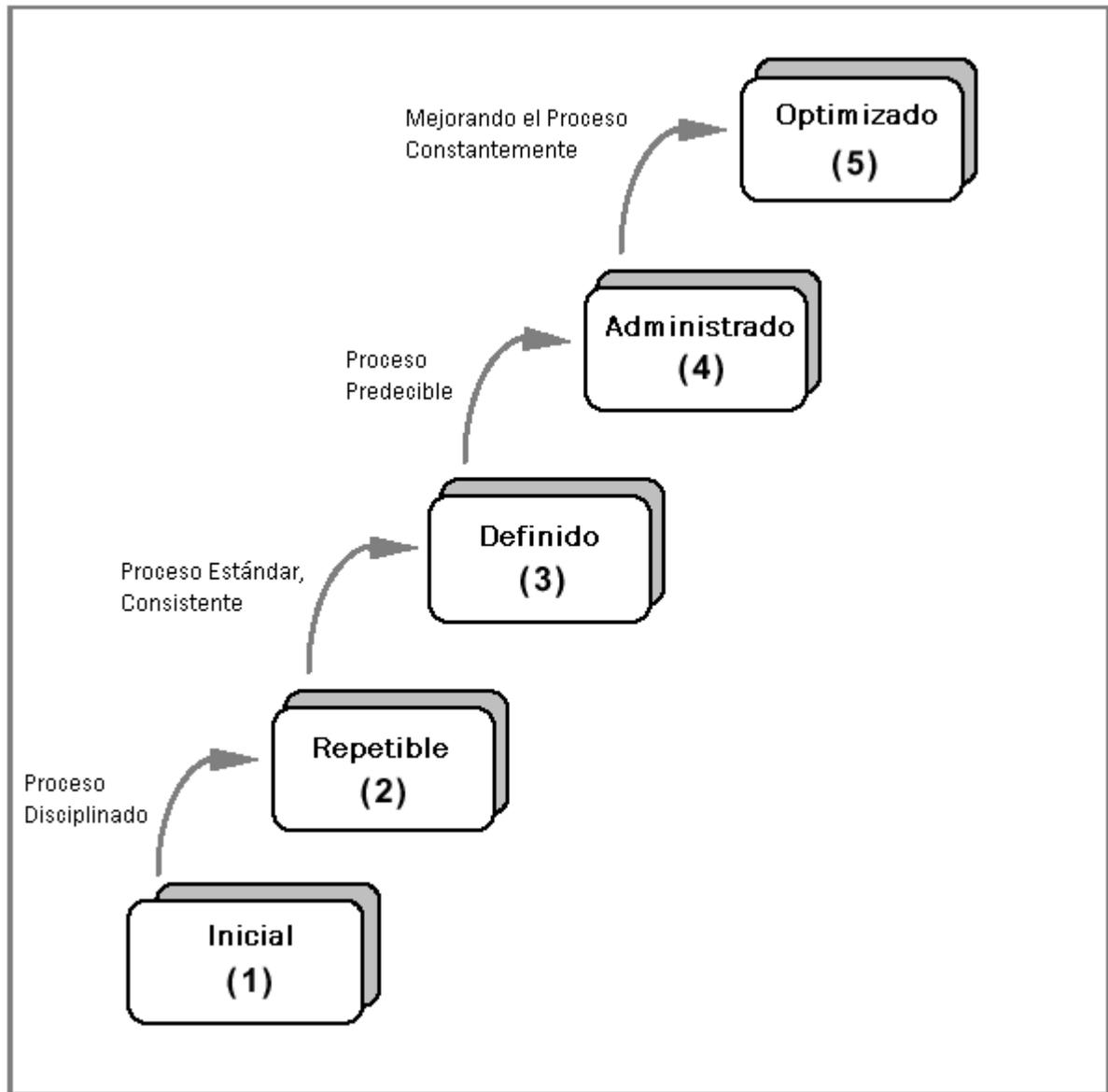


Figura 5.3 Niveles de Madurez de los Procesos de Software [SEI; 1994]

### 5.3.1.1 Nivel 1: Inicial

El primer nivel es el Inicial, que en realidad es el punto base sin valor. Una empresa estará ubicada en el nivel inicial si su proceso es ad-hoc o caótico. Esto quiere decir que realmente no existe un ambiente *estable* en el cual se pueda desarrollar o mantener software. En este nivel tendremos un número de entradas, seguidas por cierto proceso que realmente no estaba documentado, ni se documenta. Esto último está representado en la figura 5.4



Figura 5.4 **Visibilidad y Madurez del Proceso de Software Inicial [SEI; 1994]**

En este nivel lo normal es no alcanzar las metas definidas ni en tiempo, ni costo, ni recursos planeados. Y si esto llegara a ocurrir, sería gracias a personas *excepcionales* dentro del equipo de trabajo. Suponiendo que gracias a esfuerzos heroicos se logra tener éxito en el proyecto, lo más probable será que ese éxito no se podrá volver a repetir a menos que las mismas personas *excepcionales* se encuentren involucradas nuevamente.

Supongamos que dentro de nuestra estimación en cuanto a tiempo y costo definimos un Objetivo N. Dentro del Nivel 1 la probabilidad de que cumplamos ese objetivo es casi nula. La figura 5.5 nos muestra como la mayoría de las veces el Objetivo N es sobrepasado. También podemos notar que el área estimada es muy dispersa, es decir no muy certera.

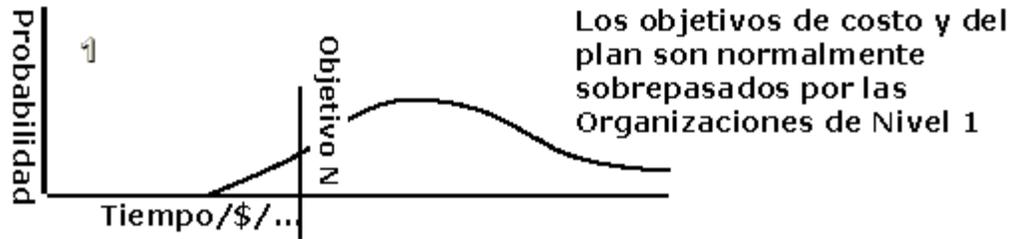


Figura 5.5 Capacidad del proceso Nivel 1 [SEI; 1994]

La Figura 5.6 nos muestra otra cara de cómo se realiza el *proceso* dentro de las Organizaciones de Nivel 1. Tenemos como entrada las necesidades del cliente. Estas pasan por una transformación no definida para dar como resultado un producto. Ahora este producto puede o no satisfacer las necesidades del cliente y puede o no haber cumplido con los estimados propuestos por la Organización que otorga el servicio.



Figura 5.6 Nivel 1

Conclusión: la capacidad es una cualidad de las personas más no de la organización. Se alcanza el propósito del proceso de manera inconsistente. No es planeado ni lleva un seguimiento.

### 5.3.1.2 Nivel 2: Repetible

Para poder dejar atrás el Nivel 1 la Organización debe empezar a documentar su proceso, empezamos a *guardar* información. Así pues, si una empresa cuenta con políticas que le permitan administrar un proyecto de software y a su vez cuenta con procedimientos para verificar que esas políticas son implementadas, se ubicaría en el nivel dos. Lo que nos muestra la figura 5.7 es una entrada que aunque pasa por ciertas cajas negras (ignoramos lo que pasa dentro de ellas) al menos ya son cajas definidas. Al término de cada caja tenemos una revisión donde se podrá detectar si el proceso está funcionando. Se puede actuar ante los problemas en cuanto se sabe de ellos. El cliente puede saber acerca del status del proyecto al finalizar cada una de las cajas.

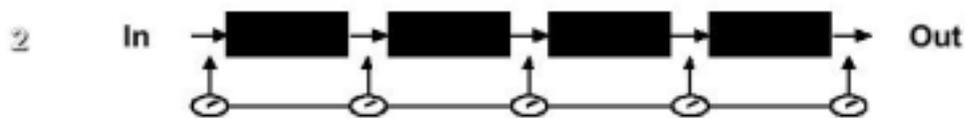


Figura 5.7 Visibilidad y Madurez del Proceso de Software Nivel 2 [SEI; 1994]

El nombre de este nivel nos da una idea de la meta principal para poder ubicarnos en este nivel. El ser repetible, se refiere a poder *repetir* éxitos anteriores. Lo cual implica que debemos tener ya cierta experiencia y una gama de proyectos previos donde el éxito ha sido obtenido. Los proyectos entonces implementan procesos efectivos, que están bien definidos, documentados, practicados, entrenados, medidos, se llevan a cabo pero que pueden ser mejorados.

En este nivel los compromisos definidos en cuanto tiempo, costo y recursos son más factibles, reales. Para lograr esto es necesario llevar un recuento en cada proyecto acerca de costos, tiempo y funcionalidad. Si se presentara el caso, los problemas para cumplir estos compromisos son detectados en su momento. En la figura 5.8 podemos notar que el Objetivo N ha sido recorrido (en  $a$  unidades) hacia la derecha a diferencia del Nivel 1, lo que significa que nos tardaremos más tiempo, nos costará más. Esto podría causar una mala impresión, parecería que no hubo mejora sino al contrario que hubo pérdida. Pero la realidad es que lo que logramos en el Nivel 2 fue ser más precisos en cuanto a nuestra estimación. En esta ocasión la estimación es más factible, y como resultado es más probable que la alcancemos.

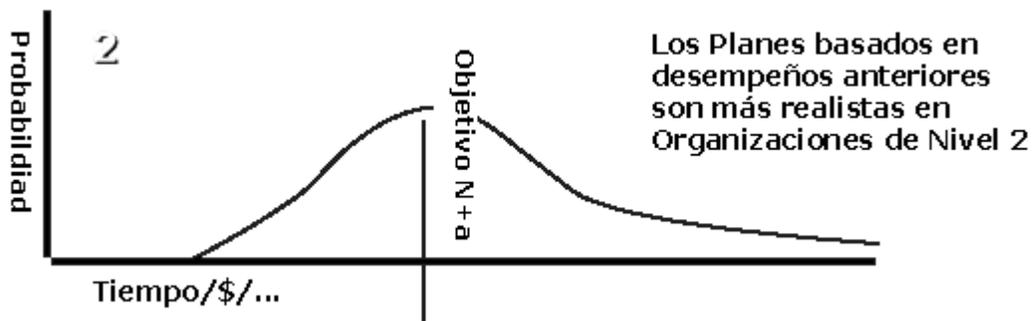


Figura 5.8 Capacidad del proceso Nivel 2 [SEI; 1994]

Los requerimientos de software y sus productos desarrollados están bien definidos en este nivel.

Es necesario tener políticas a nivel organización que permitan llevar a cabo los proyectos con los mejores procesos de administración. Pero, aun los procesos entre proyecto y proyecto podrán variar.

La figura 5.9 señala ya un proyecto definido junto con procesos de administración. Como salida obtenemos un producto bien definido.

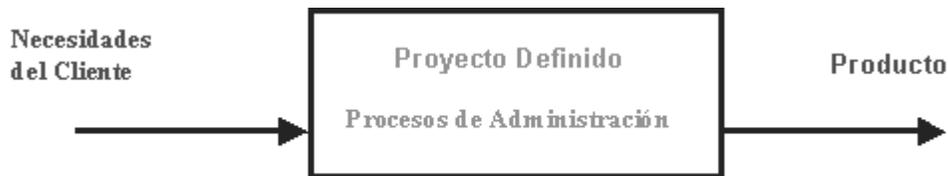


Figura 5.9 Nivel 2

Conclusión, la capacidad del proceso de software de una organización nivel 2 tiene disciplina, ya que el proyecto de software involucra planeación y seguimiento; es un proceso documentado. El proceso es estable y los éxitos anteriores pueden *repetirse*. Pero aún no contamos con métricas para servicios, solamente para productos.

### 5.3.1.3 Nivel 3: Definido

En el Nivel 3 nos olvidamos de las cajas negras. En el Nivel 3 contamos con un *proceso de software estándar de la organización* para desarrollar o mantener el software. Éste está documentado y es implementado a lo largo de toda la organización en distintos proyectos. Este proceso es la unión de prácticas de ingeniería de software y de administración de procesos. La figura 5.10 nos muestra el interior de estas cajas. Como consecuencia se incrementa el número de accesos para conocer la situación del proyecto tanto por la organización como por el cliente.

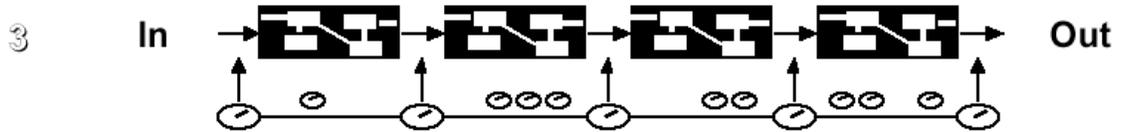


Figura 5.10 **Visibilidad y Madurez del Proceso de software Nivel 3 [SEI; 1994]**

Algunos puntos clave del nivel 3 [PAULK; 1994] son:

1. Los procesos se implementan y actualizan para ayudar a los administradores de proyectos y staff técnico a desempeñarse más efectivamente
2. Para estandarizar se ocupan prácticas de ingeniería de software
3. Un grupo dentro de la organización está encargado de las actividades del proceso de software.
4. La organización cuenta con un programa de capacitación para que todos los miembros de la organización cuenten con el conocimiento y las habilidades requeridas para desempeñar completamente sus roles.
5. El proceso de software estándar de la organización se amolda a cada proyecto para así determinar el *proceso de software definido del proyecto* (su propio proceso de software) Este contendrá información acerca de cómo saber que ya está listo el producto, entradas, estándares y procedimientos para desarrollar el trabajo, mecanismos de verificación, salidas y un criterio de terminación.

En el Nivel 3 se recorre nuestro objetivo un poco a la izquierda lo cual indica reducción en tiempo, costo y otras variantes. La probabilidad de acertar en nuestros estimado es muy precisa (figura 5.11)

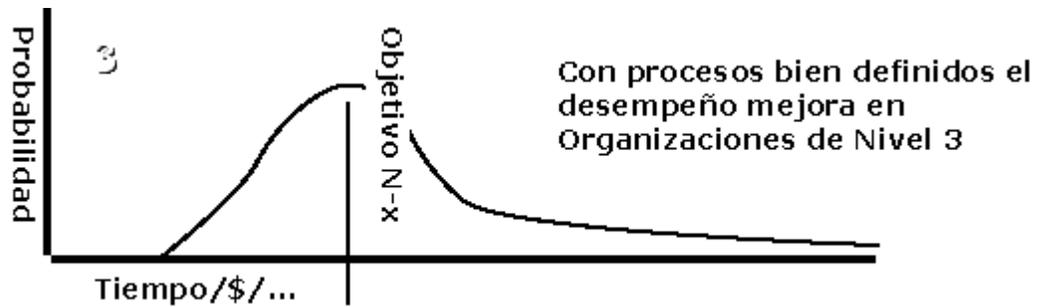


Figura 5.11 Capacidad de proceso Nivel 3 [SEI; 1994]

En el Nivel 3 agregamos a nuestro proceso un enfoque Organizacional y procesos de Ingeniería, como lo podemos visualizar en la figura 5.12.

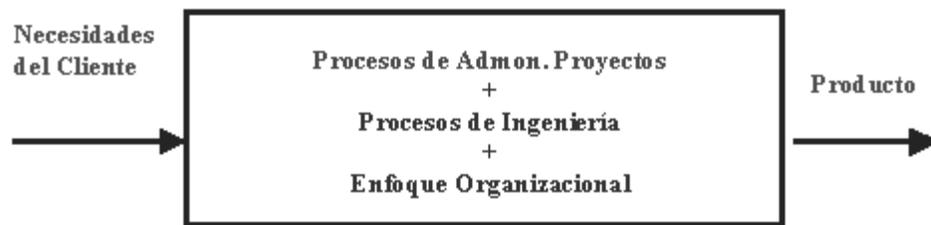


Figura 5.12 Nivel 3

Conclusión, el Nivel 3 es estándar y consistente, ya que gracias a las prácticas de ingeniería de software y a las de administración de proyectos el proceso es estable y repetible. La capacidad se logra basándose en el entendimiento de las actividades, roles y responsabilidades en un proceso de software bien definido. La administración ahora puede prepararse con anterioridad para afrontar riesgos posibles. En este Nivel se cuenta con planes y programas de mejora aunque no necesariamente se les da un seguimiento. La medición no es sólo de productos sino también de servicios.

#### 5.3.1.4 Nivel 4: Administrado

En el Nivel 4 hacemos uso de todos los datos que hemos recolectado. Convertimos datos en información relevante para la organización para así poder identificar que era lo que estaba mal.

Este nivel podría llamarse “cuantitativo” ya que en él cualquier decisión es respaldada por una base cuantitativa. Medimos el progreso y los problemas. Y mientras aumentamos la probabilidad de ser más precisos en nuestros estimados, reducimos la variabilidad (incertidumbre) de nuestro proceso. El cliente tendrá un entendimiento *medible* tanto de la capacidad del proceso como del riesgo que éste implica, incluso antes de que el proyecto inicie.

En este nivel la organización fija metas de calidad tanto del proceso como del producto. Existe un programa de medición dentro de la organización que es aplicado a lo largo de todos los proyectos, midiendo así la productividad y la calidad.

Al mismo tiempo, la organización cuenta con un repositorio de información donde almacena información relevante de proyectos anteriores que podría reutilizarse en proyectos futuros.

En la figura 5.13 se logra apreciar que no sólo se toma información del proyecto sino que también se ingresa información; existe una retroalimentación.

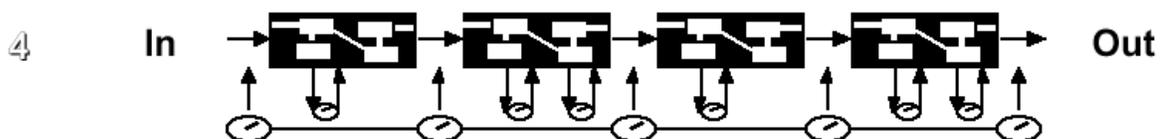


Figura 5.13 Visibilidad y Madurez del Proceso de software nivel 4 [SEI; 1994]

Todos los procesos de software son medidos. En este nivel formamos la parte cuantitativa de la organización para poder evaluar los proyectos, los procesos y los productos. Esto no quiere decir que en niveles anteriores no se obtuvo información cuantitativa. Puede ser que sí se guardo este tipo de información, pero no es sino hasta el nivel 4 que le damos un significado a esta información valiosa.

Todas esas mediciones nos marcan distintos límites (inferiores y superiores) de como debería llevarse a cabo nuestro proyecto. Así que si nuestros números caen fuera de ese rango de valores posibles (dado por proyectos anteriores) podremos identificar ya sea mejores prácticas o, en su caso, malas.

En la figura 5.14 observamos que seguimos reduciendo nuestro objetivo y la probabilidad de ser certeros es aun mayor.



Figura 5.14 Capacidad del proceso nivel 4 [SEI; 1994]

En el Nivel 4 agregamos una retroalimentación cuantitativa a nuestro proceso. Esto lo podemos observar en la figura 5.15

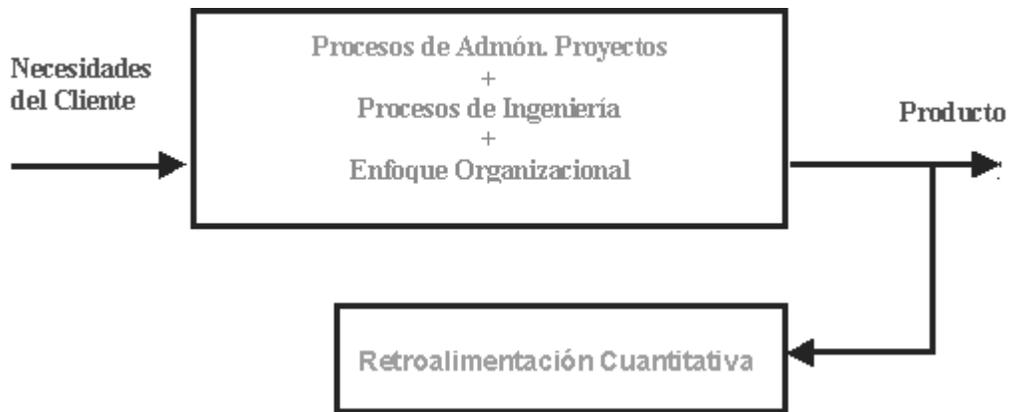


Figura 5.15 Nivel 4

Conclusión, el nivel 4 es cuantificable y predecible. Todo gracias a que el proceso, a la par que el producto y servicios, es medido y opera dentro de un límite cuantificable. Se cumple con planes y programas de mejora. Se hace una distinción entre los procesos principales y los de apoyo. Se pueden observar cadenas cliente-servidor a lo largo del proceso. Como resultado obtenemos un producto de alta calidad.

### 5.3.1.5 Nivel 5: Optimizado

En el nivel 5 la organización se dedica a mejorar continuamente su proceso probando nuevas maneras de construcción de software, pero siempre de una manera controlada. Para lograr esto es necesario poder localizar tanto debilidades como fortalezas. Al analizarlas podremos mejorar nuestro proceso y así prevenir defectos. Si un defecto es encontrado, entonces es revisado o sustituido, como lo podemos observar en la figura 5.16. Los defectos son analizados con el fin de determinar sus causas. El objetivo de esta actividad es el prevenir y evitar la recurrencia de estos defectos. Una vez encontradas las



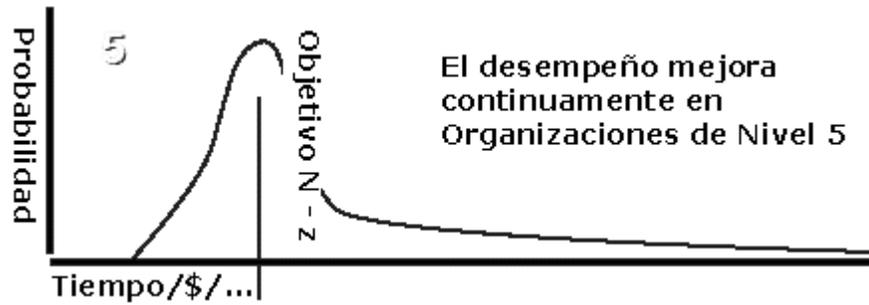


Figura 5.17 Capacidad de Proceso nivel 5 [SEI; 1994]

Es en este Nivel 5 que agregamos una fase más, la optimización del proceso. Y es así como cerramos el círculo de nuestro proceso. Lo anterior lo podemos ver gráficamente en la figura 5.18

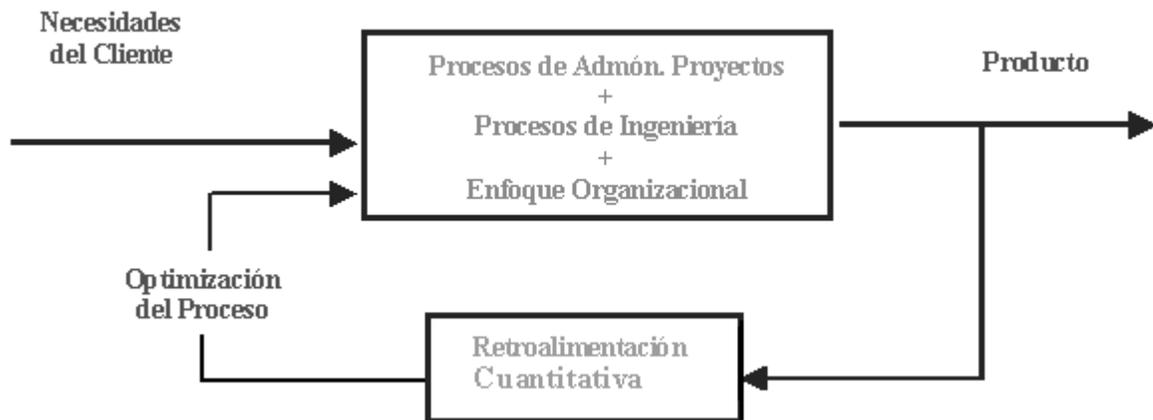


Figura 5.18 Nivel 5

Conclusión, el nivel optimizado se dedica al mejoramiento continuo de su proceso (capacidad) a la par de su madurez. Este mejoramiento se da gracias al uso o implementación de nuevas tecnologías o métodos. Obtenemos un cumplimiento total de objetivos de calidad. Los ciclos de mejora continua son identificables. Los indicadores de

desempeño competitivos son comparados contra los mejores en su clase. Existe una relación causal entre la mejora de calidad y el desempeño financiero de la compañía.

Como nota adicional, sería bueno destacar que el alcanzar el nivel 5 no significa que la organización ya no tenga una meta superior a la cual aspirar. Es más, si la organización no persiste en su mejoramiento continuo ésta podría bajar a un nivel inferior de la escala de CMM. Por lo tanto, como moraleja: si lo ganado no se cuida, ni se trabaja, se pierde.

### **5.3.2 Áreas de Procesos Clave (KPAs)**

El CMM fue creado pensando completamente en el desarrollo y manutención de software. Al armar éste modelo se reunieron las prácticas de Ingeniería de software más exitosas. Se definieron un conjunto de áreas de proceso como claves para poder lograr un mejoramiento en el proceso de software.

Un nivel de madurez será obtenido si y sólo si todas las KPAs comprendidas por él han sido cubiertas satisfactoriamente. Ahora bien, para saber si una KPA ha sido cubierta basta con que las metas especificadas en ella se hayan alcanzado.

El único nivel de madurez que no tiene KPAs a cubrir es el nivel Inicial. Es entonces a partir del nivel 2, el Repetible, que se definen un conjunto de KPAs por nivel, figura 5.19. En total son dieciocho KPAs que conforma al CMM. Estas KPAs pueden ser distribuidas de acuerdo a tres categorías de proceso (tabla 5.1): gerencia, organizacional e ingeniería.

La categoría que corresponde a la Gerencia se refiere a la planeación del proyecto de software, administración, etc. La Organizacional abarca temas como capacitación e

infraestructura entre otros. Por último la de Ingeniería toma temas como lo es el análisis de requerimientos, diseño, codificación, pruebas, etc.

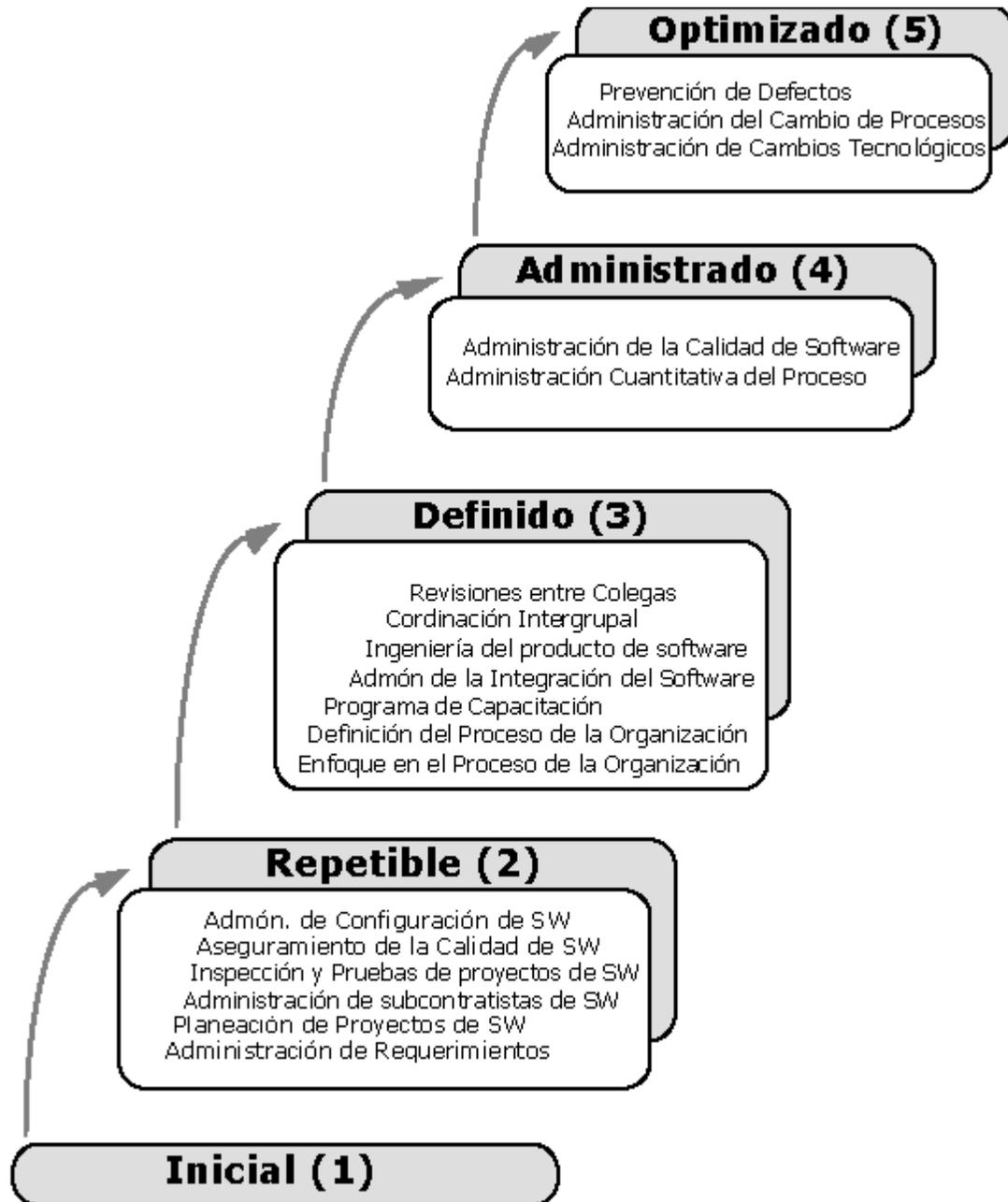


Figura 5.19 KPA's por Nivel de Madurez [PAULK; 1994]

**Tabla 5.1** KPAs por Categoría de Proceso

<b>Categorías de Proceso por nivel</b>	<b>Gerencia</b> (Planeación del proyecto de software, administración, etc.)	<b>Organizacional</b> (Capacitación, infraestructura, etc.)	<b>Ingeniería</b> (Análisis de requerimientos, diseño, codificación, pruebas, etc.)
<b>5</b>		Administración de cambios tecnológicos	
	Administración del cambio de procesos		Prevención de defectos
<b>4</b>	Administración cuantitativa del proceso		Administración de la calidad de software
<b>3</b>	-Administración de la integración del software -Coordinación intergrupal	-Enfoque en el proceso de la organización -Definición del proceso de la organización -Programa de capacitación	-Ingeniería del producto de software -Revisión entre colegas
<b>2</b>	-Admón. de Requerimientos -Planeación de proyectos de software -Seguimiento y supervisión de proyectos de software -Administración de subcontratistas de software -Aseguramiento de la calidad de software -Administración de la configuración de software		
<b>1</b>	Procesos Ad-hoc		

[PAULK; 1994]

### 5.3.3 Características Comunes

Para poder alcanzar las metas definidas por KPA, CMM nos brinda un conjunto de prácticas clave. Estas prácticas representan la infraestructura o las actividades que contribuyen en mayor medida en la implementación e institucionalización de las Áreas Clave de Proceso.

Por facilidad CMM agrupa a las prácticas clave en cinco grupos. De acuerdo a CMM v1.1 éstas son:

1. Compromiso.

Describe las Acciones que la organización debe realizar para asegurar que el proceso sea establecido y persistirá. Normalmente involucra el establecimiento de políticas organizacionales y liderazgo.

2. Habilidades Necesarias

Describe las condiciones previas que deben existir en el proyecto u organización para poder implementar el proceso competentemente. Normalmente involucra recursos, estructuras organizacionales y capacitación.

3. Actividades Realizadas

Describe las actividades, roles, y procedimientos necesarios para implementar una KPA. Normalmente involucra el establecer planes y procedimientos, realizar el trabajo, darle seguimiento, y tomar acciones correctivas según sea necesario.

#### 4. Medición y Análisis

Describe las prácticas básicas de medición que son necesarias para determinar el estatus en relación al proceso. Estas mediciones son utilizadas para controlar y mejorar el proceso. Normalmente incluyen ejemplos de métricas que pueden tomarse.

#### 5. Verificación e Implantación

Describen los pasos para asegurarse que las actividades son llevadas a cabo de acuerdo con el proceso que se ha establecido. Normalmente abarca revisiones y auditorias por parte de la administración y aseguramiento de la calidad de software.

### 5.4 Roles

Para poder escalar niveles de madurez es necesario aumentar la capacidad de nuestra organización. Es decir el rango de resultados esperados que se pueden obtener tras seguir un proceso debe reducirse. Rangos en cuanto tiempo, costos y recursos.

Cada **rol** (una persona puede jugar varios roles, dependiendo del tamaño del proyecto u organización) deberá cumplir con sus responsabilidades:

- Debe de existir un compromiso por parte de la *Gerencia Superior*.
- Los *Gerentes* deberán planear, organizar, dirigir, y controlar el trabajo dentro de su área.
- Los *Líderes de Proyecto* deberán dirigir, controlar, administrar, y regular un proyecto para construir sistemas de software o hardware/software. Es el responsable ante el cliente.

- El *Líder de Proyectos de Software* será el responsable y controlará todas las actividades y recursos de software de un proyecto.
- El *Líder de Software de Primera Línea* funge como el responsable directo de la administración del staff y actividades de una unidad organizacional (departamento, equipo de proyecto) de ingenieros de software y staff relacionado. Esto incluye el proveer dirección técnica, administrar el personal y salarios.
- El *Líder de Tareas de Software* cubre el rol de líder de un equipo técnico para una tarea específica, tiene la responsabilidad técnica y provee dirección técnica al staff de trabajo dedicado a determinada tarea.
- El *Staff* (no gerentes) es responsable de realizar una función asignada, tales como desarrollo de software o administración de configuración de software.
- El *Staff de Ingeniería de Software* son las personas técnicas de software (no gerentes) como analistas, programadores e ingenieros. Ellos realizan el desarrollo de software y las actividades de mantenimiento del proyecto.

## 5.5 Grupos

Los grupos básicos que se deben cubrir:

- Grupo de Ingeniería de Software
  - Conformado por: Gerentes y Staff Técnico
  - Responsable de: Actividades de desarrollo y manutención de software.
- Grupo de Ingeniería de procesos de software

- Conformado por: Especialistas
  - Responsable de: Facilitar la definición, mantenimiento y mejoramiento del proceso de software usado por la organización.
- Grupo de Ingeniería de Sistema
  - Conformado por: Gerentes y Staff Técnico
  - Responsable de: Especificar los requerimientos del sistema, asignar y definir las interfaces de los recursos necesarios y monitorear el diseño y desarrollo de los componentes.
- Grupo de Pruebas de Sistema
  - Conformado por: Gerentes y Staff Técnico
  - Responsable de: Planear y realizar las pruebas del sistema del Software para determinar si es que el producto de software satisface sus requerimientos.
- Grupo de Aseguramiento de Calidad de Software
  - Conformado por: Gerentes y Staff Técnico
  - Responsable de: Planear e implementar las actividades de calidad de los proyectos para asegurar que los pasos del proceso de software y estándares sean seguidos.
- Grupo de Administración de Configuración de Software
  - Conformado por: Gerentes y Staff Técnico
  - Responsable de: Planear, coordinar, e implementar las actividades de administración de la configuración para los proyectos de software.
- Grupo de Entrenamiento
  - Conformado por: Gerentes y Staff

- Responsable de: Coordinar y arreglar las actividades de entrenamiento de una organización. Preparan y conducen la mayoría de los cursos de entrenamiento y coordinan el uso de otras vías de entrenamiento.

Después de nombrar cada uno de los roles y grupos en los que se basa CMM y de especificar sus responsabilidades, es importante que estos / éstas se cubran.

También es muy cierto que no es necesario que una persona cumpla con sólo un rol, o que forme parte de un solo grupo. El tamaño de la organización determinará esto. Una persona puede cubrir varios roles o ser parte de distintos equipos. Lo importante es que exista alguien responsable de cubrir las actividades de cada uno de los roles o grupos.

## **5.6 Apreciaciones de Procesos**

### **5.6.1 Valoraciones y Evaluaciones**

Básicamente el SEI divide en dos las apreciaciones para identificar la madurez de una organización en la ejecución de su proceso de software:

1. Valoración del Proceso de Software (Assessment)
2. Evaluación de la Capacidad de Software

Las *valoraciones del proceso del software* se ocupan para:

- Determinar el estado del proceso de software actual de la organización
- Para determinar los asuntos de mayor prioridad en cuanto al proceso que enfrenta una organización y
- Para obtener el apoyo de la organización para la mejora del proceso de software.

Las *evaluaciones de la capacidad del software* se utilizan para:

- Identificar contratistas que sean calificados para realizar el trabajo de software o
- Para monitorear el estado del proceso de software empleado en un esfuerzo existente de software

Los pasos comunes tanto para una evaluación como para una valoración se observan en la figura 5.20.

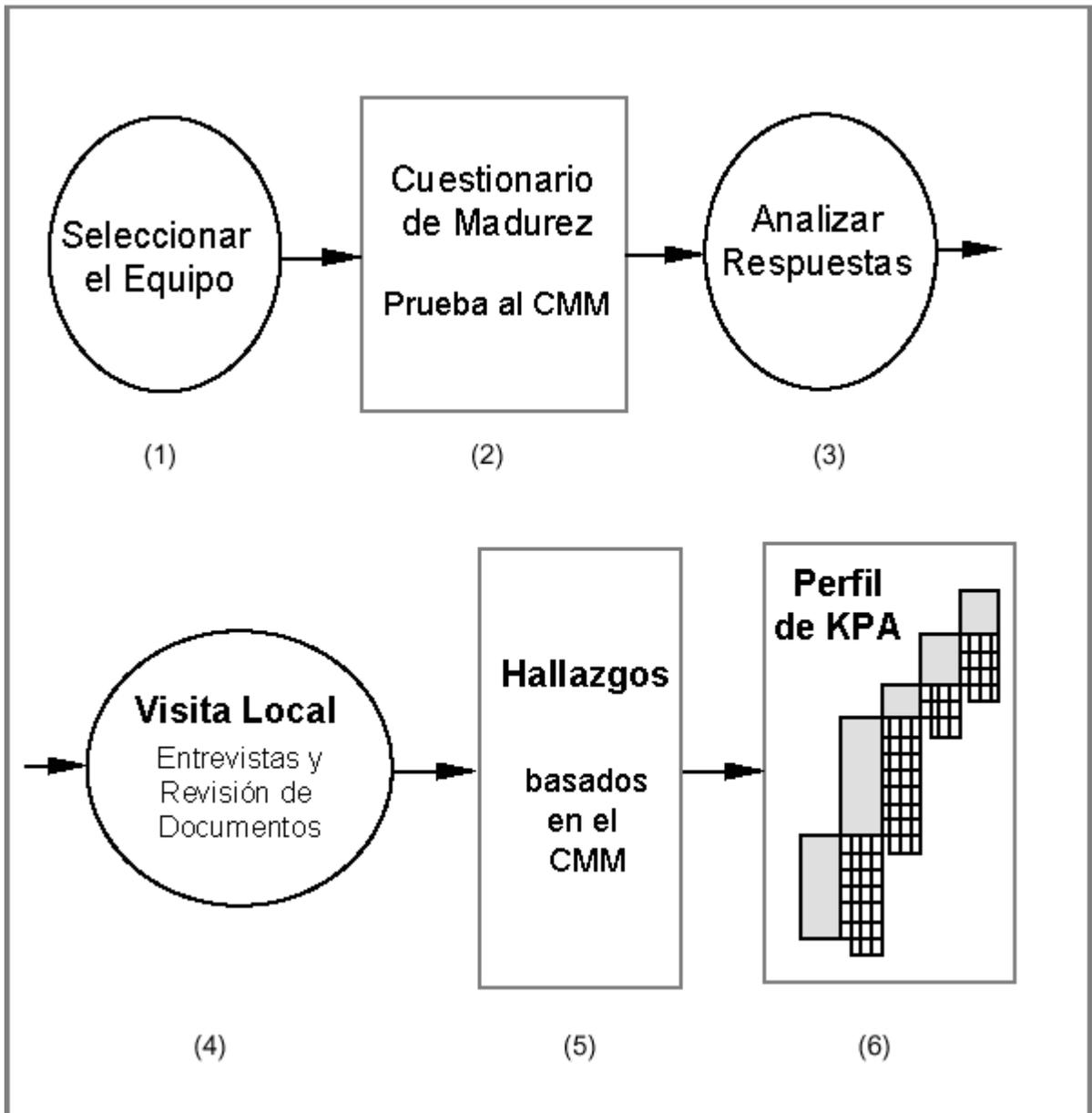


Figura 5.20 Pasos Comunes para Evaluaciones y Valoraciones [CMM; 1993]

Pasos:

1. Seleccionar el Equipo, éste deberá estar conformado por asesores reconocidos por el SEI como capaces de realizar evaluaciones o valoraciones según sea el caso. Este grupo de personas cuentan con un alto conocimiento en prácticas de ingeniería de software.
2. Localmente se contesta el cuestionario de madurez, tomando a CMM como una guía para poder realizar esta tarea
3. El equipo de asesores analiza las respuestas
4. El equipo de asesores se traslada físicamente a la organización. Durante su estancia se llevan a cabo entrevistas al igual que se hace una revisión de documentos. Todo con el afán de comprender el proceso de software que sigue la organización. Si alguna práctica difiriera notablemente de aquellas definidas dentro del CMM el equipo entrevistado debe dar una explicación para sostener la práctica propia.
5. Una vez que toda la información ha sido recabada el equipo evalúa y determina aquellas fortalezas y puntos de mejora para la organización.
6. Por último, se realiza un estudio detallado de las KPAs y de las metas que no fueron cubiertas. Es posible que aun cuando una KPA obtenga un puntaje satisfactorio se puedan identificar puntos de mejora.

### **5.6.2 Métodos de Apreciación basados en el CMM (Appraisal Methods)**

El SEI proporciona un juego de herramientas para diagnosticar, llamado apreciación, para aquellas organizaciones que desean saber el estado de su propio o el de otro proceso de software en relación al CMM para software V 1.1. El término apreciación como es utilizado dentro del SEI incluye tanto las evaluaciones como las valoraciones. Ambas se enfocan en el proceso de desarrollo de software de la organización.

Para determinar el estado de los procesos propios, el método apropiado a utilizar es una valoración tal como lo es la Apreciación Basada en CMM para la Mejora Interna del Proceso (**CBA IPI**). Para determinar el estado del proceso de un externo, el método apropiado a implementar es una evaluación, tal como la Evaluación de Capacidad de Software (**SCE**).

Para asegurarnos que ambos métodos sean rigurosos y sólidos y el programa del proceso sea desarrollado y mantenido se utiliza el marco de trabajo de la Evaluación de CMM (CAF). Éste define los estándares que un método de apreciación debe reunir.

El SEI entrena y autoriza a personas calificadas para dirigir las apreciaciones, en otras palabras, en la Apreciación Basada en CMM para la Mejora Interna del Proceso (CBA IPI), y para utilizar los materiales del SEI al conducir las apreciaciones. El SEI no valida ni certifica los resultados de la apreciación.

## 5.7 Beneficios del CMM

*“Todos los modelos son erróneos; algunos modelos son útiles”*

*George Box*

Recordemos que CMM es una herramienta que ayuda a las organizaciones de software para mejorar sus procesos. Por igual, ayuda a organizaciones que se dedican a la adquisición, a seleccionar o administrar a sus contratistas de software.

El propósito de CMM es describir buenas prácticas de administración y de ingeniería estructuradas por un marco maduro de trabajo.

Algunos de los beneficios de utilizar CMM como marco de trabajo para un mejoramiento continuo son:

1. CMM ayuda a formar una visión compartida lo que significa mejoramiento de procesos a nivel organización.
2. Establece un lenguaje común al hablar acerca del proceso de software.
3. Define un conjunto de prioridades para atacar los problemas de software.
4. Provee una estructura conceptual para mejorar la administración y desarrollo de productos de software en una manera disciplinada y consistente.
5. Aumenta la posibilidad de que una organización de software alcance sus metas de costos, calidad y productividad de una manera disciplinada y consistente.

[PAULK, 1994]

CMM identifica las características de un proceso efectivo de software pero, la organización madura atiende los asuntos esenciales para un proyecto exitoso, incluyendo gente y tecnología, al igual que al proceso.

Cabe resaltar que para poder interpretar correctamente CMM a un ambiente específico, es necesario utilizar inteligencia, experiencia y conocimientos. Esa interpretación debe basarse en las necesidades de negocio, los objetivos de la organización y los proyectos.

Para ser más específicos, al aplicar CMM y a razón que la organización acrecenta su madurez es notorio el avance en la posibilidad de predecir, en el control, y en la eficiencia.

La primera mejora que se presenta es la posibilidad de predecir. Con esto nos referimos a que a lo largo que aumenta la madurez en una organización, la definición de resultados objetivo se vuelve más certera. Es decir, los resultados planeados se asemejan más a los resultados reales.

También se observa una mejora en el control. Aumenta el control que se tiene alrededor de los resultados objetivo. Es decir, si se presentara un proyecto similar a uno ya enfrentado es probable que se ocupe la experiencia ya adquirida y así reducir el rango (tiempo, costo, entre otros) de entrega.

Otra mejora en la eficiencia ya que los resultados objetivo se perfeccionan si la madurez de la organización se incrementa. Esto se ve reflejado en distintos campos. El tiempo de desarrollo se acorta, los costos se reducen, la calidad y productividad aumentan, ya que al ocupar procesos más efectivos, se reduce el número de errores generados. Al no perder tiempo en re-trabajo (arreglando esos errores) se reduce el tiempo de desarrollo.

[CMU/SEI 93-TR-24]

### 5.7.1 Algunos Estudios

A continuación se presentarán los resultados de algunos estudios acerca de los beneficios que brinda CMM. Lo importante de ellos son los número que se dan, ya que esto nos permite medir las mejoras para así evaluarlas.

- I. Según Herb Krasner, “Self-Assessment Experience at Lockheed” presentado en el Tercer Workshop Annual del SEPG en noviembre de 1990 el número de defectos/ KSLOC<sup>1</sup> enviados de una organización nivel 1 es de 30 mientras que el de una de nivel 3 es de 3.5, y éste es superado por una de nivel 5 que es de 0.5.
- II. Acorde con Stan Rifkin, “The Business Case for Software Process Improvement” presentado en la Quinta reunión nacional del SEPG en 1993, el número de defectos en el producto liberado de una organización nivel 1 es de 61, en el de una nivel 3 es de 7 y en el de una nivel 5 es de 1.
- III. El estudio realizado por John Vu, Boeing “Software Process Improvement Journey” presentado en SEPG’97 marca que una organización que implementa CMM mejora en cuatro aspectos:
  1. Al utilizar datos históricos la precisión en los estimados de software aumenta.
  2. El número de horas se reduce mientras que la productividad aumenta en un 62%.
  3. El número de errores que se presentan tras la liberación del producto disminuye mientras se escalan los distintos niveles de madurez.
  4. El tiempo de ciclo (cycle time) se reduce en un 36%.

---

<sup>1</sup> KSLOC, por sus siglas en inglés se refiere a miles de líneas de código de software

Después de revisar estos datos podemos concluir que aplicar un modelo de mejoramiento de procesos como lo es CMM puede brindar grandes beneficios. Pero siempre hay que tener en mente que para poder llegar a un nivel 5 es necesario tiempo, compromiso y perseverancia. El crear una cultura dirigida hacia el mejoramiento continuo no es fácil, y más si los resultados no son inmediatos. Por lo tanto, no hay que perder de vista nuestra meta y seguir para poder disfrutar de los prometedores beneficios.