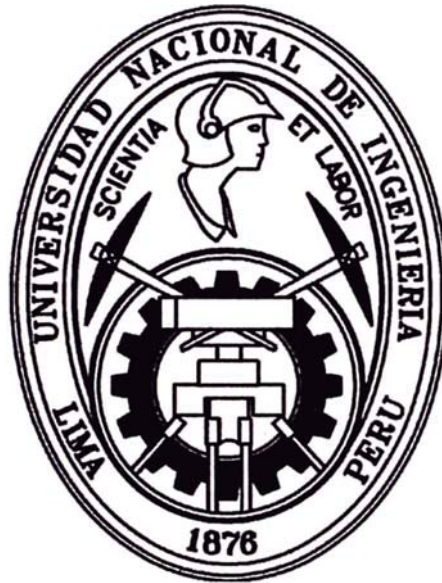


**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS**



**MODELO DE MADUREZ DE CAPACIDADES DE  
PROCESO Y APRENDIZAJE PARA SOFTWARE**

**INFORME DE SUFICIENCIA**

**Para optar el título Profesional de:  
INGENIERO DE SISTEMAS**

**Delfín Napoleón Espejo Donayre**

**LIMA – PERU  
2002**

## INDICE

<b>DESCRITORES TEMATICOS</b> .....	<b>3</b>
<b>RESUMEN EJECUTIVO</b> .....	<b>4</b>
<b>INTRODUCCION</b> .....	<b>6</b>
<b>1. CAPITULO I: ANTECEDENTES</b> .....	<b>7</b>
1.1    DIAGNÓSTICO ESTRATÉGICO .....	7
1.1.1 <i>Análisis Interno</i> .....	7
1.1.2 <i>Análisis Externo</i> .....	8
1.2    DIAGNÓSTICO FUNCIONAL.....	9
<b>2. CAPITULO II: MARCO TEORICO</b> .....	<b>11</b>
2.1    LAS PERSONAS, PROCESOS Y ORGANIZACIONES.....	11
2.2    EL MARCO DE MADUREZ DE PROCESOS .....	12
2.2.1 <i>Organizaciones de Software con Procesos Inmaduros vs Maduros</i> .....	13
2.2.2 <i>Conceptos Fundamentales sobre Madurez de Procesos</i> .....	15
2.2.3 <i>Comportamiento Característico de los Niveles de Madurez</i> .....	16
2.2.4 <i>Los Cinco Niveles de Madurez del Proceso de Software</i> .....	20
2.2.5 <i>Visibilidad dentro del Proceso de Software</i> .....	22
2.2.6 <i>La Capacidad de Proceso y la Predicción de la Ejecución</i> .....	25
2.2.7 <i>Saltando los Niveles de Madurez</i> .....	27
2.2.8 <i>Estructura interna de los niveles de madurez</i> .....	29
2.2.9 <i>Áreas de Proceso Clave</i> .....	30
2.2.10 <i>Características Comunes</i> .....	38
2.3    LAS ORGANIZACIONES ABIERTAS AL APRENDIZAJE.....	39
2.3.1 <i>Las cinco disciplinas para el aprendizaje</i> .....	41
2.3.2 <i>La Arquitectura de las Organizaciones Abiertas al Aprendizaje</i> .....	43
<b>3. CAPITULO III: PROCESO DE TOMA DE DECISIONES</b> .....	<b>46</b>
3.1    PLANTEAMIENTO DEL PROBLEMA .....	46
3.2    ALTERNATIVAS DE SOLUCIÓN.....	49
3.2.1 <i>El Modelo de Madurez de Capacidades de Proceso</i> .....	49
3.2.2 <i>El Modelo de Organizaciones Abiertas al Aprendizaje de Peter Senge</i> .....	51

3.3	METODOLOGÍA DE LA SOLUCIÓN .....	52
3.3.1	<i>El Proceso de Software de la Organización</i> .....	54
3.3.2	<i>Estructura Organizacional</i> .....	60
3.3.3	<i>Prácticas Claves Nivel 2</i> .....	61
3.3.4	<i>Prácticas Claves Nivel 3</i> .....	75
3.3.5	<i>Prácticas Claves Nivel 4</i> .....	90
3.3.6	<i>Prácticas Claves Nivel 5</i> .....	96
3.4	TOMA DE DECISIONES .....	104
3.5	ESTRATEGIAS ADOPTADAS.....	105
<b>4.</b>	<b>CAPITULO IV: EVALUACIÓN DE RESULTADOS .....</b>	<b>107</b>
4.1	ANTES .....	107
4.2	DESPUÉS.....	107
<b>5.</b>	<b>CAPITULO V: CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>109</b>
5.1	CONCLUSIONES.....	109
5.2	RECOMENDACIONES.....	110
	<b>BIBLIOGRAFIA .....</b>	<b>112</b>
	<b>ANEXOS.....</b>	<b>113</b>
	ANEXO 1: TABLA RESUMEN DE LA EVALUACIÓN EN EL NIVEL 2 .....	113
	ANEXO 2: PRÁCTICAS CLAVES DE LAS CAPACIDADES DE APRENDIZAJE .....	114
	ANEXO 3: CICLO DE VIDA DEL PROCESO UNIFICADO.....	116

## **DESCRITORES TEMÁTICOS**

- Modelo de madurez de capacidades
- Proceso de desarrollo de software
- Organización abierta al aprendizaje
- Organización inteligente
- Calidad del Software
- Productividad en el desarrollo de software
- Aprendizaje organizacional

## RESUMEN EJECUTIVO

### **Descripción del Problema**

Los grupos de personas que realizan actividades de desarrollo de software, tienen problemas más complejos que cualquier grupo u organización al incluir a la tecnología, que cambia rápidamente, como insumo y producto. Estos inconvenientes van desde el ámbito personal hasta el organizacional, incluyendo problemas de comunicación, adaptación, poco conocimiento y experiencia, falta de valores, problemas de trabajo en equipo, liderazgo, roles no definidos, procesos mal estructurados, etc. Estos problemas dan como resultado aplicaciones que no cumplen con toda la funcionalidad esperada, no satisfacen al usuario, tienen muchos errores, el tiempo de desarrollo y el presupuesto exceden grandemente el estimado, difíciles de hacer cambios, sin apropiada documentación, etc.

La problemática descrita ha sido estudiada por organizaciones como el SEI (Software Engineering Institute) y el MIT (Massachusetts Institute of Technology) por varios años. El SEI en estudios realizados hace una década, determinó que el 85% de las organizaciones que desarrollan software se encuentran en un nivel de madurez incipiente, donde predomina la falta de comunicación y los procesos no definidos. Este porcentaje ha disminuido poco en este tiempo, actualmente se estima una reducción de tan solo unos 10 puntos. Por otra parte, el MIT ha estudiado la problemática de las organizaciones desde un enfoque sistémico, determinando que los problemas tienen su origen en los valores, actitudes y capacidad de aprendizaje de las personas de la organización.

### **Solución**

El presente trabajo trata cómo las organizaciones de software pueden superar estos problemas usando un modelo orientado a los procesos unido con otro orientado al aprendizaje organizacional. Se define un marco de trabajo que permite determinar el estado actual o nivel de madurez de los procesos y personas, y propone una

forma de evolución a los niveles superiores, teniendo siempre como base el mejoramiento de las capacidades de aprendizaje de las personas en todos los niveles; para finalmente transformarla en una organización abierta al aprendizaje, adaptable a los cambios tecnológicos y con procesos que se mejoran constantemente.

De esta manera una organización o equipo que desarrolla software aprenderá a trabajar en grupo, a dialogar, desarrollando habilidades de comunicación, resolución de conflictos, capacidades reflexivas que permiten ver las causas reales de los problemas, es decir, desarrollará sus capacidades de aprendizaje, también tendrá definido el proceso de desarrollo de software y los roles involucrados que serán conocidos por todas las personas, medirá dicho proceso para mejorarlo e investigará la Tecnología de Información y las mejores prácticas de negocio para su uso en los sistemas a desarrollar y como herramientas interna, produciendo así, software de alta calidad, aceptables por el usuario y sin excederse en el presupuesto y tiempo estimado.

### **Conclusión**

El modelo de madurez de capacidades junto con las disciplinas de aprendizaje permiten mejorar las capacidades de las personas y los procesos de la organización. Se considera a la organización como un sistema formado por elementos; como son las personas, los procesos, la infraestructura, y la tecnología. Estos elementos al interactuar determinan la calidad de los productos desarrollados. El elemento principal es la persona y su interacción interna dada por los procesos. La persona es quien define los planes, políticas y procesos de la organización, y a su vez ejecuta dichas actividades. La persona tiene habilidades, competencias y valores que son usados en los procesos que se ejecutan sobre una infraestructura utilizando la tecnología como herramienta. Para evolucionar a los siguientes niveles de madurez, las personas tienen que cambiar su forma de pensar, aprendiendo constantemente y adquiriendo nuevas capacidades que le van a permitir realizar cambios permanentes y mejorar los procesos de la organización.

## INTRODUCCIÓN

El Modelo de Madurez de Capacidades (CMM) es un marco que describe los elementos claves de un proceso de software efectivo. El CMM describe un camino de mejoramiento evolutivo desde un proceso ad hoc e inmaduro hacia un proceso maduro y disciplinado.

El CMM cubre prácticas para planeamiento, ingeniería, y gestión de desarrollo y mantenimiento de software. Cuando son seguidas, estas prácticas claves mejoran la habilidad de la organización de llegar a sus metas, en cuanto a costo, programación, funcionalidad y calidad del producto.

El CMM establece un instrumento de medida contra el que es posible juzgar, en una forma repetitiva, la madurez del proceso de software de una organización y compararla con el estado de la práctica de la industria. También puede ser usado por una organización para planear mejoramientos a su proceso de software.

El Modelo de Madurez de Capacidades para software provee a las organizaciones de software con los lineamientos para determinar cómo ganar control de sus procesos para el desarrollo y mantenimiento de software y cómo evolucionar rumbo a una cultura de Ingeniería de Software y hacia la excelencia en la administración. El CMM fue diseñado para guiar a las organizaciones de software en la selección de las estrategias para el mejoramiento del proceso mediante la determinación de la madurez de los procesos actuales y de la identificación de algunos de los problemas más críticos con respecto a la calidad del software y el mejoramiento del proceso. Enfocándose en un conjunto limitado de actividades y trabajando agresivamente para realizar dichas actividades, una organización puede mejorar constantemente su proceso de software para hacer posible ganancias en las capacidades del proceso de software de forma continua.

# CAPITULO I: ANTECEDENTES

## 1.1 Diagnóstico Estratégico

### 1.1.1 Análisis Interno

Las organizaciones que desarrollan software, adolecen de muchos problemas internamente, usando tecnología que es muy cambiante. Estos inconvenientes se dan en todos los niveles abarcando el ámbito personal y organizacional, incluyendo problemas de comunicación, adaptación, competencias insuficiente, falta de valores, problemas de trabajo en equipo, liderazgo, roles no definidos y procesos mal estructurados.

Los problemas internos en una organizaciones de desarrollo de software se pueden resumir en:

- Poca o mala comunicación en el equipo
- Los errores en el proyecto nunca se informan hacia niveles superiores, hasta que es demasiado tarde.
- Trabajo individualista, poca integración del equipo.
- Muchos conceptos o términos no están bien definidos.
- Roles y responsabilidades no están bien delimitados.
- Se usan métodos y herramientas muy complejas.
- Los resultados de unas actividades no son muy útiles para las siguientes actividades.
- El personal que participa tiene poca experiencia en las Tecnologías de Información usadas.
- Personal de desarrollo trabaja bajo mucha presión.
- Alta rotación del personal.



### 1.1.2 Análisis Externo

Las organizaciones de software crean aplicaciones informáticas para sus clientes, que pueden ser otro departamento dentro de la misma empresa u otra empresa.

Dichas aplicaciones informáticas terminan con muchos problemas, como por ejemplo: no cumplen con toda la funcionalidad esperada, no satisface al usuario, tiene muchos errores, no son rápidos, el tiempo de desarrollo ha excedido grandemente el estimado, es muy difícil hacer cambios al software, no existe una documentación apropiada, el conocimiento se encuentra en las personas que participaron en el desarrollo.

Parte de estos problemas son consecuencia de los cambios constantes en los requisitos del software por parte del cliente o falta de compromiso de los directivos y de las personas participantes en el proyecto.

Los problemas percibidos por el cliente de una organización de desarrollo de software se pueden resumir en:

- El software no cumple con los requerimientos del cliente.
- El software es de baja calidad, muchos errores.
- El software provee poca utilidad al cliente.
- El mantenimiento del software es muy costoso.
- Proyecto excede los tiempos y presupuestos estimados.
- La inversión en el software es muy alta.
- El mantenimiento y mejoramiento del software depende de los desarrolladores.

Los problemas externos percibidos por la organizaciones de desarrollo de software son:

- Cambios constantes en los requerimientos.
- El proyecto no tiene el respaldo total de la alta gerencia.
- Falta de compromiso de las personas participantes en el proyecto.

## 1.2 Diagnóstico Funcional

Analizaremos a las organizaciones de software agrupándolas según como trabajan o funcionan para lograr sus objetivos, es decir, basadas en el grado en que sus procesos son conocidos, controlados, medidos y mejorados.

El primer grupo tienen sus procesos de software como una entidad amorfa – una caja negra – y la visibilidad dentro de los procesos del proyecto esta limitada. Desde que la división de las actividades es definida pobremente, los administradores tienen un periodo extremadamente difícil para establecer el estado de progreso de los proyectos y de las actividades.

Los requerimientos fluyen dentro del proceso de software de manera descontrolada. El desarrollo de software es frecuentemente visto como “magia negra”, principalmente por los administradores que no están familiarizados con el software.

En el segundo grupo los requerimientos de los clientes y los trabajos producidos están controlados, y se establecen practicas básicas de administración de proyectos. En ocasiones definidas estos controles de la administración permiten tener visibilidad dentro del proyecto.

El proceso de construcción de software puede ser visto como una sucesión de cajas negras que permiten tener visibilidad administrativa en puntos de transición como lo son los flujos de actividades entre las diferentes cajas. Aún así la administración puede que no conozca los detalles de qué está sucediendo dentro de la caja, los productos de los procesos y los puntos de verificación que permiten determinar que el proceso que se está trabajando es definido y conocido. La administración reacciona a los problemas según vayan ocurriendo.

En el tercer grupo, son visibles las estructuras internas de las cajas. La estructura interna representa el medio por el cual los procesos estándares de software definidos de diferentes proyectos se han aplicado a proyectos específicos. Los ingenieros y los administradores comprenden sus roles y responsabilidades dentro del proceso y además saben cómo sus actividades interactúan en el nivel de detalle apropiado. La administración se prepara proactivamente para los riesgos que se presenten. Individuos externos al

proyecto pueden obtener estados de actualización precisa y rápida debido a que los procesos definidos proporcionan gran visibilidad dentro de las actividades del proyecto.

En el cuarto grupo los procesos del software definidos son cualitativamente instrumentados y controlados. Los administradores son capaces de medir los progresos y los problemas. Tienen un objetivo, bases cualitativas para la toma de decisiones. Su habilidad de predecir los resultados crece constantemente de forma más precisa así como la variabilidad en el proceso crece en forma reducida.

En el quinto grupo, se intenta continuamente con nuevas y mejoradas formas de construir software, de una forma controlada, con el fin de mejorar la productividad y la calidad.

El cambio disciplinado es el medio de vida por el cual las actividades ineficientes o defectuosas son identificadas y revisadas o eliminadas. La comprensión se extiende a través de procesos existentes y dentro de los efectos de cambios potenciales a los procesos. Los administradores son capaces de estimar y por eso de seguir cuantitativamente el impacto y la efectividad de los cambios.

## **CAPITULO II: MARCO TEORICO**

### **2.1 Las Personas, Procesos y Organizaciones**

“La productividad y la calidad de los productos están directamente relacionados con la calidad del proceso usado para desarrollarlo.”

“Las capacidades y competencias de las personas están directamente relacionadas con la adaptación rápida a los nuevos modelos organizacionales y nuevas tecnologías.”

“Las transformaciones personales puede originar cambios organizacionales profundos, que a su vez cambian las actitudes de las personas, aspiraciones y conductas que afectan a los procesos, estrategias y prácticas”

Después de dos décadas de promesas incumplidas acerca de obtener productividad y calidad en base a la aplicación de nuevas metodologías y tecnologías de software, las organizaciones se han dado cuenta que su principal problema es la incapacidad de administrar el proceso de software. Los beneficios de las mejores herramientas y métodos no pueden ser obtenidos en un ambiente donde los proyectos son indisciplinados y caóticos. En muchas organizaciones, los proyectos frecuentemente terminan muy tarde y duplican el presupuesto planeado. En tales circunstancias, la organización no proporciona la infraestructura y ayuda necesaria para superar estos problemas.

Aún en organizaciones indisciplinadas, algunos proyectos de software producen resultados excelentes. Estos proyectos son exitosos, generalmente, por el heroico esfuerzo de un equipo dedicado o por la elevada competencia de algunos, en vez de, repetir métodos que han sido probados en una organización con procesos de software maduros. En ausencia de un proceso

de software que abarque toda la organización, la repetición de los resultados depende enteramente en conseguir a los mismos individuos para el siguiente proyecto. Basar el éxito solamente en la disponibilidad de los individuos no provee la base para el mejoramiento de la productividad y calidad en toda la organización.

Por otra parte, la mayoría de las actitudes y competencias de las personas que participan en los proyectos, son consecuencia de la cultura organizacional desarrollada en muchos años y en la cual las personas se desempeñan. Pueden existir personas muy competentes que al no encontrar un ambiente adecuado para seguir desarrollándose se retiran a otras organizaciones, y al no estar debidamente documentado sus actividades realizadas se pierde parte del conocimiento y experiencia adquirida. Aun más, los directores o jefes de proyectos al darse cuenta que no tienen el control del proyecto buscan responsables externos, encontrándose frecuentemente al cliente como parte del problema y el cliente haciendo lo mismo, encuentra a la organización como responsable, formándose un círculo vicioso, en el cual se genera una desconfianza por parte del cliente y una desmotivación y presión sobre los desarrolladores. En ausencia de una infraestructura y procesos que permite explotar las competencias de las personas y mejorarlas, las actitudes serán defensivas, el grado de compromiso será menor, la rotación del personal aumentará y el éxito dependerá del esfuerzo y las competencias de algunas pocas personas.

El mejoramiento continuo puede ocurrir solo cuando se orientan los esfuerzos hacia la construcción de una infraestructura basada en procesos con prácticas efectivas de Ingeniería de software, administración, y aprendizaje continuo que mejoren las competencias de las personas.

## **2.2 El Marco de Madurez de Procesos**

El Modelo de Madurez de Capacidades (CMM) de Proceso es un marco que describe los elementos claves de un proceso de software efectivo. El CMM describe un camino de mejoramiento evolutivo desde un proceso ad hoc e inmaduro hacia un proceso maduro y disciplinado.

El CMM cubre prácticas para planeamiento, ingeniería, y gestión de desarrollo y mantenimiento de software. Cuando son seguidas, estas prácticas claves mejoran la habilidad de la organización de llegar a sus metas, en cuanto a costo, programación, funcionalidad y calidad del producto.

El CMM establece un instrumento de medida contra el que es posible juzgar, en una forma repetitiva, la madurez del proceso de software de una organización y compararla con el estado de la práctica de la industria. También puede ser usado por una organización para planear mejoramientos a su proceso de software.

El Modelo de Madurez de Capacidades para software provee a las organizaciones de software los lineamientos para determinar cómo ganar control de sus procesos para el desarrollo y mantenimiento de software y cómo evolucionar rumbo a una cultura de Ingeniería de Software y hacia la excelencia en la administración. El CMM fue diseñado para guiar a las organizaciones de software en la selección de las estrategias para el mejoramiento del proceso mediante la determinación de la madurez de los procesos actuales y de la identificación de algunos de los problemas más críticos con respecto a la calidad del software y el mejoramiento del proceso. Enfocándose en un conjunto limitado de actividades y trabajando agresivamente para realizar dichas actividades, una organización puede mejorar constantemente su proceso de software para hacer posible ganancias en las capacidades del proceso de software de forma continua.

### **2.2.1 Organizaciones de Software con Procesos Inmaduros vs Maduros**

La asignación de objetivos para el mejoramiento de procesos requieren de un entendimiento sobre la diferencia de organizaciones maduras e inmaduras. En las organizaciones de software inmaduras, los procesos de software son generalmente improvisados por los administradores de proyecto durante el curso del proyecto. Aun si un proceso de software es especificado, este no es rigurosamente ejecutado. Una organización de software inmadura es reactiva, y los administradores de proyectos están frecuentemente orientados a la

resolución inmediata de crisis. Los cronogramas y presupuestos son frecuentemente excedidos por que ellos no están basados en estimaciones reales. Cuando una fecha esta comprometida, la funcionalidad y calidad del producto usualmente se sacrifican para alcanzar los tiempos del cronograma.

En una organización de software inmadura, no hay objetivos basados para certificar la calidad de los productos o para resolver los problemas del producto o procesos. De allí que la calidad de los productos es difícil de predecir. Actividades que intentan mejorar la calidad, tales como las revisiones y pruebas, son frecuentemente reducidas o eliminadas cuando no hay mucho tiempo para cumplir con el cronograma del proyecto.

En el otro lado, una organización de software madura, posee una habilidad que abarca a toda la organización para administrar el proceso de desarrollo y mantenimiento de software. El proceso de software es rigurosamente comunicado a los empleados existentes como a los nuevos y las actividades de trabajo son llevados a cabo según el proceso planeado. Los procesos definidos son adecuados para su uso y consistentes con la forma en que el trabajo actualmente se realiza. Estos procesos definidos son modificados cuando es necesario, y mejoramientos son desarrollados a través de controladas pruebas-piloto y análisis de costo/beneficio. Los roles y responsabilidades junto con el proceso definido son claros en todo el proyecto y en toda la organización.

En una organización de software madura, los administradores, monitorean la calidad de los productos de software y la satisfacción del cliente. Hay un objetivo, cuantificar variables para certificar la calidad de los productos y analizar los problemas de los procesos y productos. Los cronogramas y presupuestos son basados en datos históricos y son realistas. Los resultados esperados de costos, cronogramas, funcionalidad y calidad del producto son usualmente obtenidos. En general un proceso disciplinado, es seguido consistentemente porque todos los participantes entienden el valor de lo que hacen, a la vez que una necesaria infraestructura existe para dar soporte a los procesos.

Razonando sobre estas observaciones acerca de organizaciones de software maduras e inmaduras, se requiere construir un marco de madurez de proceso de software.

### **2.2.2 Conceptos Fundamentales sobre Madurez de Procesos**

Según el diccionario Webster, un proceso es “..un sistema de operaciones para producir algo...una serie de acciones, cambios o funciones que obtienen un fin o resultado”. La IEEE define un proceso como “un secuencia de pasos ejecutados para obtener un objetivo”. Un proceso de software puede ser definido como un conjunto de actividades, métodos y practicas y transformaciones que las personas usan para desarrollar y mantener software y los productos asociados (p.e. planes de proyectos, documentos de diseños, código, casos de pruebas y manuales de usuario). Mientras una organización madura, el proceso de software llega a ser mas definido, mas consistentemente implementado en toda la organización.

La capacidad de proceso de software describe el rango de resultados esperados que pueden ser obtenidos siguiendo un proceso de software. La capacidad de proceso de software de una organización provee una forma de predecir los resultados esperados para el siguiente proyecto de software que la organización lleve a cabo.

El rendimiento del proceso de software representa los actuales resultados obtenidos al haber seguido un proceso de software. El rendimiento del proceso de software se orienta en los resultados obtenidos mientras que la capacidad de proceso de software se orienta en los resultados esperados. Basado en los atributos de un proyecto especifico y el contexto en el cual es conducido, el actual rendimiento del proyecto puede que no refleje totalmente la capacidad de proceso de una organización. La capacidad del proyecto puede estar restringida por su entorno. Por ejemplo, cambios radicales en la aplicación o tecnología puede colocar al personal del proyecto en una curva de aprendizaje que cause que sus capacidades y rendimiento del proyecto disminuyan la capacidad de proceso de toda la organización.



Madurez del proceso de software es la medida en el cual un proceso específico es explícitamente definido, administrado, medido, controlado y efectivo. Madurez implica un potencial para crecer en capacidad e indica lo efectivo que son los procesos de software de la organización y la forma consistente de como se está aplicando en los proyectos de toda la organización. El proceso de software es muy bien entendido en toda la organización, usualmente a través de documentos y entrenamientos, y el proceso es continuamente monitoreado y mejorado por sus usuarios. La capacidad de un proceso maduro es conocida. La madurez del proceso de software implica que la productividad y calidad resultantes del proceso de software de la organización pueden ser mejoradas en el tiempo a través de mejoras consistentes en las disciplinas obtenidas de usar su proceso de software.

Mientras una organización obtiene madurez en su proceso de software, institucionaliza su proceso de software vía políticas, estándares y estructuras organizacionales. La institucionalización conlleva a la construcción de una infraestructura y cultura corporativa que apoya a las practicas, métodos y procedimientos del negocio de tal forma que duren después de quienes originalmente las definieron.

#### Niveles de Madurez

Un nivel de madurez es una plataforma de evolución bien-definida para lograr un proceso de software maduro. Cada nivel de madurez indica un nivel en la capacidad del proceso, como se ilustró en Figura 2.1. Por ejemplo, en el Nivel 2 la capacidad de proceso de una organización ha sido elevada desde especializada (ad-hoc) a disciplinada estableciendo control sobre la dirección de los proyectos.

### **2.2.3 Comportamiento Característico de los Niveles de Madurez**

Los niveles de madurez del 2 al 5 pueden ser caracterizados a través de las actividades ejecutadas por la organización para establecer o mejorar el proceso de software, por las actividades ejecutadas en cada proyecto y por los resultados de las capacidades del proceso en los proyectos. Un

comportamiento característico del nivel 1 es incluido para establecer una base de comparación para el mejoramiento del proceso en los niveles superiores.

#### El Nivel Inicial

En el Nivel Inicial, la organización no proporciona un ambiente apropiado para desarrollar y mantener el software. Cuando a una organización carece de practicas de dirección, los beneficios de las buenas prácticas de ingeniería de software son minados por la planificación ineficaz y los sistemas de compromiso manejados por reacción.

Durante una crisis, los proyectos típicamente abandonan los procedimientos planeados y se orientan a la codificación y pruebas. El éxito depende completamente de tener un excepcional gerente y un equipo de software eficaz. De vez en cuando, los gerentes del software capaces y fuertes pueden resistir las presiones para tomar los atajos en el proceso de software; pero cuando ellos dejan el proyecto, su influencia de estabilización se va con ellos. Incluso un proceso de ingeniería robusto no pueda superar la inestabilidad creada por la ausencia de prácticas de dirección.

La capacidad de proceso de software en las organizaciones de Nivel 1 es imprevisible porque el proceso del software constantemente se ha cambiado o modificado durante el progreso del trabajo (es decir, el proceso es ad-hoc). Los horarios, presupuestos, la funcionalidad, y la calidad del producto son generalmente imprevisible. La actuación depende de las capacidades de los individuos y varía con sus habilidades innatas, el conocimiento y motivaciones. Hay pocos procesos de software estable, y el rendimiento sólo puede predecirse por el individuo en lugar de la capacidad de la organización.

#### Nivel 2 - El Nivel Repetible

En el Nivel Repetible, se establecen las políticas para dirigir un proyecto de software, así como, los procedimientos para llevar a cabo esas políticas. La planeación y administración de los nuevos proyectos esta basado en la experiencia de proyectos similares. El objetivo logrado en el Nivel 2 es institucionalizar la dirección eficaz de los procesos para proyectos de software que permiten a las organizaciones repetir las prácticas exitosas que

desarrollaron en los proyectos más tempranos, aunque procesos específicos llevados a cabo por los proyectos pueden diferir. Un proceso eficaz puede ser caracterizado como experto, documentado, robusto, especializado, moderado, y capaz de mejorar.

Los proyectos en las organizaciones de Nivel 2 han instalado el control básico en la administración del desarrollo de software. Los compromisos del proyecto son realistas y están basados en los resultados observados de los proyectos anteriores y en los requisitos del proyecto actual. El gerente de software de un proyecto realiza un seguimiento al costo, horarios, y la funcionalidad; se identifican los problemas cuando ellos se originan. Los requisitos del software y los productos de trabajo desarrollados para satisfacerlos son la línea base, y su integridad se controla. Los estándares del proyecto de software se definen, y la organización asegura que ellos sean fielmente seguidos.

La capacidad de proceso de software de las organizaciones de Nivel 2 pueden resumirse como disciplinado porque la planeación y seguimiento del proyecto de software son estables y pueden repetirse los éxitos anteriores. El proceso del proyecto está bajo el mando eficaz de un sistema de administración de proyectos, siguiendo planes realistas basados en la actuación de proyectos anteriores.

### Nivel 3 - El Nivel Definido

En el Nivel Definido, el proceso normal para desarrollar y mantener software en la organización se documenta, esto incluye ambos, la ingeniería de software y la administración de proyectos, y estos procesos se integran en un todo coherente. Este proceso normal es referido a lo largo del CMM como el proceso de software estándar de la organización. Los procesos establecidos en el Nivel 3 son usados (y cambiados, especializados) para ayudar a los gerentes y el personal técnico realicen su trabajo más eficazmente. La organización explota eficazmente las prácticas de ingeniería de software cuando estandariza su proceso de software. Hay un grupo que es responsable por las actividades de proceso de software de la organización, por ejemplo el grupo de procesos de ingeniería de software, o SEPG. Un programa de entrenamiento en toda la organización se lleva a cabo para

asegurar que el personal y gerentes tienen el conocimiento y las habilidades que exigen cumplir sus papeles asignados.

Los proyectos adaptan el proceso de software estándar de la organización para desarrollar su propio proceso de software definido, el cual que considera las características propias del proyecto. Este proceso adaptado es referido en el CMM como el proceso de software definido del proyecto. Un proceso de software definido contiene un conjunto coherente e integrado de procesos de ingeniería de software y administración. Un proceso bien-definido puede caracterizarse por incluir criterios entendibles, entradas, estándares y procedimientos para realizar el trabajo, mecanismos de comprobación (como las revisiones de par), salidas, y criterio de realización. Debido a que el proceso de software está bien definido, la dirección tiene una buena visión del progreso técnico en todos los proyectos.

La capacidad de proceso de software de las organizaciones del Nivel 3 puede resumirse como estándar y consistente, porque ambas actividades de ingeniería de software y administración son estables y repetibles. Se tiene establecido líneas del producto, costos, horarios, y funcionalidad que están bajo control y a su vez se hace un seguimiento de la calidad de software. Esta capacidad del proceso está basada en un común entendimiento, en toda la organización, de las actividades, roles y responsabilidades del proceso de software definido.

#### Nivel 4 - El Nivel Administrado

En el Nivel Administrado, la organización fija las metas de calidad cuantitativas para productos y procesos de software. La productividad y calidad son medidos para las actividades importantes del proceso de software en todos los proyectos como parte de un programa de medición organizacional. Una Base de Datos de proceso de software organizacional se usa para coleccionar y analizar los datos disponibles de los procesos de software definidos de los proyectos. Los procesos del software son instrumentados con mediciones bien definidas y consistentes. Estas mediciones establecen la base cuantitativa para evaluar los procesos y productos de software de los proyectos.

Los proyectos logran el control sobre sus productos y procesos estrechando la variación de sus rendimientos de proceso para que caigan dentro de los límites cuantitativo aceptables. Las variaciones significantes en el rendimiento del proceso pueden ser distinguidas de la variación al azar (el ruido), particularmente dentro de las líneas de producto establecidas. Los riesgos involucrados en el recorrido de la curva de aprendizaje de un nuevo dominio de la aplicación son conocidos y cuidadosamente manejada.

La capacidad de proceso de software de las organizaciones del Nivel 4 puede resumirse como predecible porque el proceso es moderado y opera dentro de los límites mensurables. Este nivel de capacidad del proceso permite a una organización predecir las tendencias en el proceso y calidad del producto dentro de los límites cuantitativos. Cuando estos límites se exceden, se toman acciones para corregir la situación. La calidad de los productos del software son predecibles y alta.

#### Nivel 5 - El Nivel Optimizado

En el Nivel Optimizado, la organización entera se enfoca en el continuo mejoramiento del proceso. La organización tiene los medios para identificar las debilidades y fortalezas del proceso proactivamente con la meta de prevenir la ocurrencia de defectos. Los datos en la efectividad del proceso de software se usan para realizar análisis de beneficio-costos de nuevas tecnologías y proponer cambios al proceso de software de la organización. Las innovaciones que explotan mejor las prácticas de ingeniería de software se identifican y transfieren a lo largo de la organización.

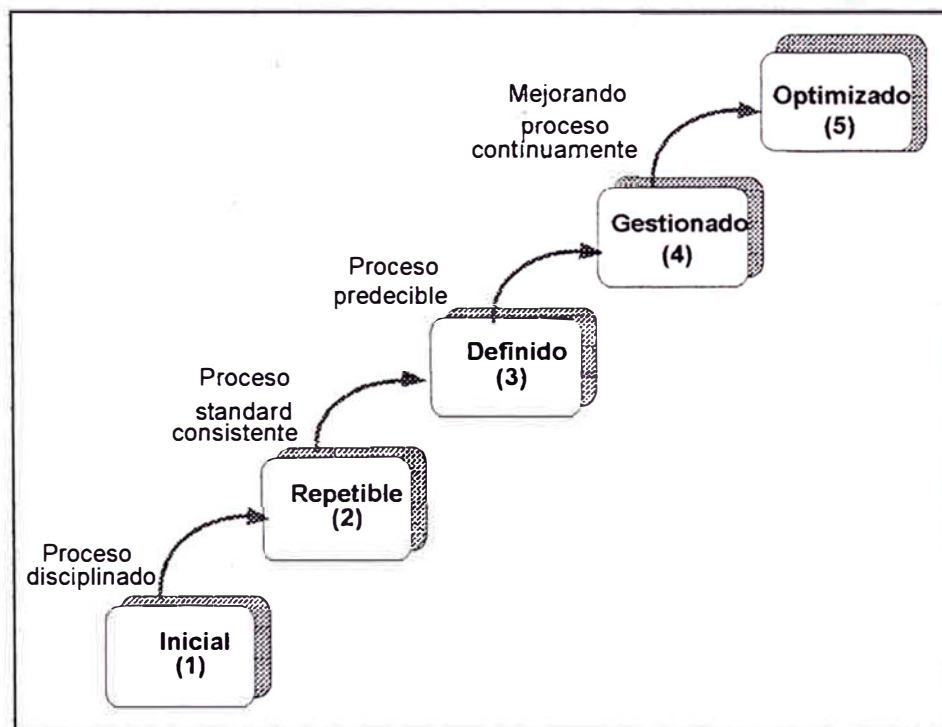
Los equipos de proyecto de software en las organizaciones de Nivel 5 analizan los defectos para determinar sus causas. Se evalúan los procesos de software para prevenir que se repitan los tipos conocidos de defectos, y las lecciones aprendidas se diseminan a los otros proyectos.

#### **2.2.4 Los Cinco Niveles de Madurez del Proceso de Software**

El mejoramiento continuo de proceso esta basado en muchos pero pequeños pasos de evolución mas que innovaciones revolucionarias. El CMM provee un

marco para organizar estos pasos de evolución dentro de cinco niveles de madurez, que ponen las bases para sucesivos mejoramientos continuos de procesos. Estos cinco niveles de madurez definen una escala para medir la madurez de los procesos de software de una organización y para evaluar su capacidad de proceso de software. Los niveles también ayudan a la organización a priorizar sus esfuerzos de mejoramiento.

Un nivel de madurez es un estado de evolución bien definido en el cual se va obtener un proceso de software maduro. Cada nivel de madurez provee una capa que es la base para un mejoramiento de proceso continuo. Cada nivel compromete un conjunto de objetivos de proceso que cuando son satisfechos estabilizan un importante componente del proceso de software. Obteniendo cada nivel del marco de madurez, se establece un diferente componente en el proceso de software resultando un incremento en la capacidad del proceso de software.



**Figura 2.1** Los cinco niveles de madurez de proceso de software

Organizando el CMM en cinco niveles como se muestra en la figura 2.1 se priorizan las acciones para el incremento de la madurez del proceso de software. La flecha en la figura indica el tipo de capacidad de proceso que esta siendo institucionalizado en la organización en cada paso del marco de

madurez. Las siguientes características de los cinco niveles de madurez resaltan los cambios primarios de proceso hechos en cada nivel.

1) Inicial El proceso de software es caracterizado por ser especializado y ocasionalmente caótico. Pocos procesos están definidos y el éxito depende del esfuerzo individual.

2) Repetible Los procesos de administración de proyectos básicos son establecidos para un seguimiento del costo, cronograma y funcionalidad. La necesaria disciplina en el proceso es para repetir los éxitos de los proyectos anteriores de similares aplicaciones.

3) Definido El proceso de software tanto para la administración e ingeniería esta documentado, estandarizado e integrado dentro de un proceso de software estándar para toda la organización. Todos los proyectos usan una versión aprobada y personalizada del proceso de software estándar de la organización para el desarrollo y mantenimiento de software.

4) Administrada Mediciones detallada del proceso de software y la calidad del producto son coleccionadas. Ambas, el proceso de software y producto son cuantitativamente entendidas y controladas.

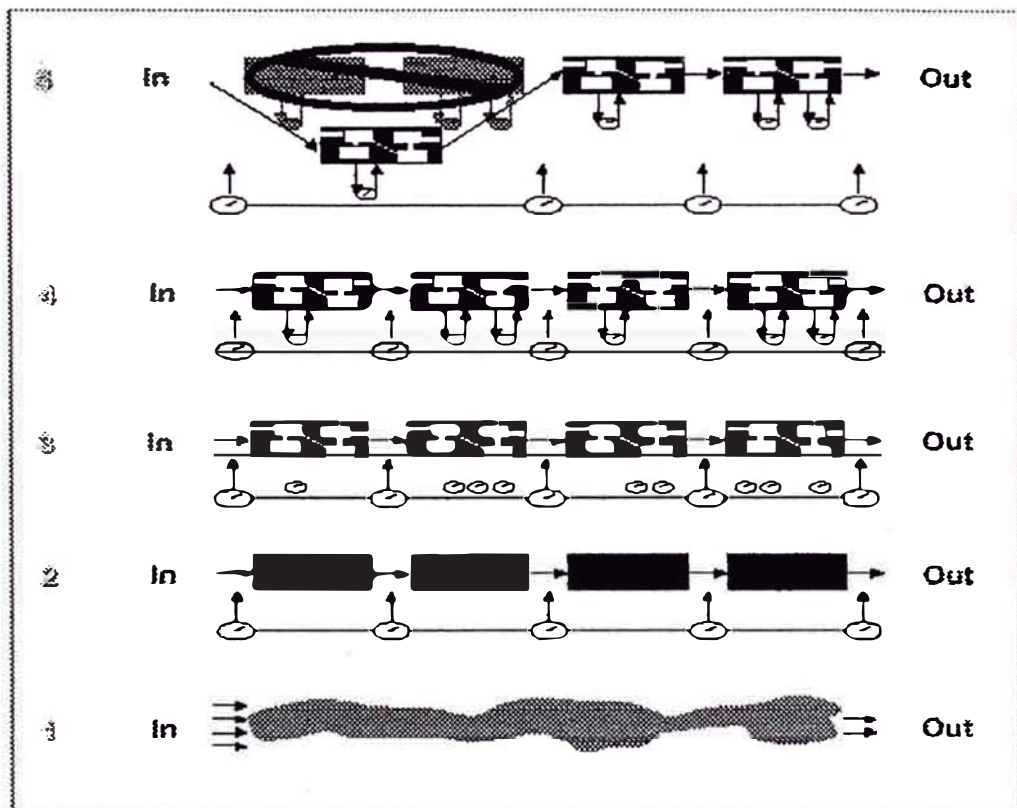
5) Optimizada El mejoramiento del proceso continuo es habilitado para una retroalimentación cuantitativa desde los procesos hasta las innovaciones de ideas y tecnologías en proyectos pilotos.

#### **2.2.5 Visibilidad dentro del Proceso de Software**

Los ingenieros de Software tienen una idea detallada acerca del estado del proyecto porque ellos tienen la información de primera mano sobre el estado y ejecución del proyecto.

Sin embargo, en grandes proyectos la idea detallada usualmente es obtenida del personal con experiencia en su área responsable. Esto visto desde fuera del proyecto, sin una observación de primera mano, tal como la de un administrador senior, disminuye la visibilidad dentro del proceso del proyecto.

La figura 2.2 ilustra el nivel de visibilidad dentro del proyecto en cada nivel de madurez del proceso. Cada nivel de madurez alcanzado provee gradualmente una mejor visibilidad dentro del proceso del proyecto.



**Figura 2.2 Una visión Administrativa de la Visibilidad dentro del Proceso de Software en cada Nivel de Madurez**

En el Nivel 1, el proceso de software es como entidad amorfa – una caja negra – y la visibilidad dentro del proceso del proyecto es limitada. Desde que los escenarios de las actividades están pobremente definidas, los administradores tiene una gran dificultad para establecer el estado actual del progreso y actividades del proyecto. Los requerimientos fluyen dentro del proceso de software de una manera descontrolada y de resultados inesperados. El desarrollo de software es visto como magia negra, especialmente por los administradores que no están familiarizados con el software.

En el Nivel 2, los requerimientos de cliente y los productos producidos están controlados, y practicas básicas de administración de proyectos han sido



establecidas. Estos controles de administración permiten visibilidad dentro del proyecto en ocasiones establecidas. El proceso de construir software es visto como una sucesión de cajas negras que permiten al administrador una visibilidad en los puntos de transición mientras las actividades se realizan dentro de las cajas (hitos del proyecto). A pesar que la administración no puede conocer los detalles de lo que está sucediendo dentro de las cajas, el resultado del proceso y los puntos de control para confirmar que el proceso esta caminando son identificados y conocidos. La administración reacciona conforme surgen los problemas.

En el Nivel 3, la estructura interna de las cajas, por ejemplo, las tareas dentro del proceso de software definido, es visible. La estructura interna representa la forma como el proceso de software estándar de la organización ha sido aplicado a un específico proyecto. Ambos, los administradores e ingenieros entienden sus roles y responsabilidades dentro del proceso y como sus actividades interactúan en el apropiado nivel de detalle. Los administradores proactivamente se preparan para los riesgos que puedan surgir. Individuos externos al proyecto pueden obtener rápidamente y en forma segura el estado actualizado del proyecto debido a que el proceso permite una gran visibilidad dentro de las actividades del proyecto.

En el Nivel 4, el proceso de software definido son instrumentados y controlados cuantitativamente. Los administradores son capaces de medir el progreso y los problemas. Ellos tienen un objetivo y bases cuantitativas para tomar decisiones. Sus habilidades para predecir resultados crecen constantemente mientras que la variabilidad en el proceso crece muy poco.

En el Nivel 5, formas nuevas y mejoradas para construir software son continuamente probadas de una manera controlada, para mejorar la calidad y productividad. Los cambios disciplinados son una forma de vida mientras que las actividades ineficientes o defectos propensos son identificados y reemplazados o revisados. La capacidad de entendimiento se extiende mas allá de los procesos existentes y abarca los efectos de cambios potenciales en el proceso. Los administradores son capaces de estimar y luego hacer un seguimiento cuantitativo del impacto y efectividad del cambio.

### **2.2.6 La Capacidad de Proceso y la Predicción de la Ejecución**

La madurez de un proceso de software de una organización ayuda a predecir la habilidad del proyecto para cumplir con sus objetivos. Los proyectos en organizaciones de Nivel 1 experimentan muchas variaciones en los costos, cronogramas, funcionalidad y calidad. Como se ilustra en la Figura 2.3, tres mejoras para alcanzar los objetivos finales son observados mientras madura el proceso de software de una organización.

Primero, mientras aumenta la madurez, la diferencia entre los objetivos planeados y los resultados actuales disminuye a través del proyecto. Las organizaciones del Nivel 1 frecuentemente no cumplen con sus fechas programadas originalmente y se exceden ampliamente, sin embargo, las organizaciones de Nivel 5 podrían ser capaces de cumplir con las fechas estimadas con una seguridad considerable. Esto es porque las organizaciones de Nivel 5 usan un proceso de software cuidadosamente construido operando con parámetros conocidos, y la selección de la fecha final está basada en muchos datos que poseen sobre su proceso y su ejecución al aplicarlo. (Esto está ilustrado en la Figura 2.4 como que tanto el área bajo la curva se acerca a la derecha de la línea).

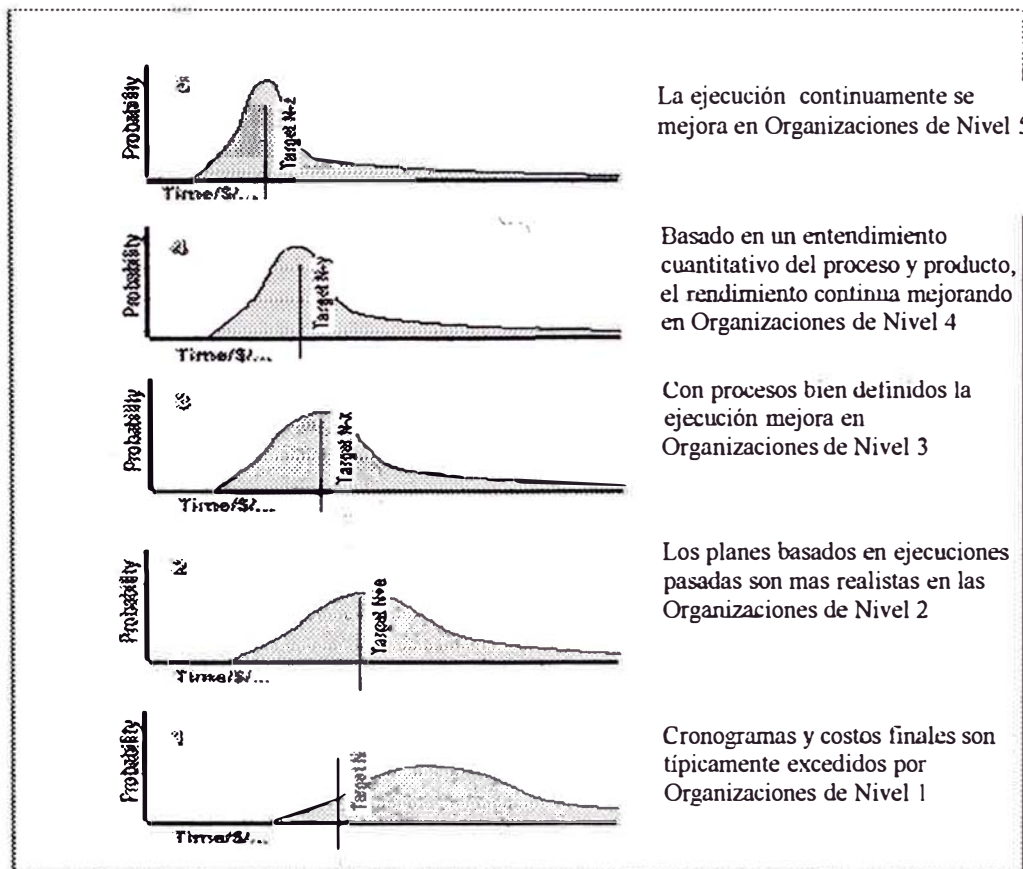


Figura 2.3 La Capacidad de Proceso a medida que se incrementa el Nivel de Madurez

Segundo, mientras se incrementa la madurez, la variabilidad de los resultados actuales sobre los objetivos disminuye. Por ejemplo, en el Nivel 1 las fechas de entrega para proyectos de similar tamaño son impredecibles y muy variable. Proyectos similares en organizaciones de Nivel 5 estarán terminando con un rango más pequeño. Esta estrecha variación ocurre en el nivel de madurez más alto debido a que virtualmente todos los proyectos son ejecutados con parámetros controlados. (Esto está ilustrado en la Figura 2.3 como que tanto el área bajo la curva se concentra cerca de la línea objetivo)

Tercero, los objetivos finales mejoran mientras el nivel de madurez incrementa. Es decir, mientras la organización madura, los costos disminuyen, el tiempo de desarrollo llega a ser más corto, y la productividad y calidad incrementa. En una organización de Nivel 1, el tiempo de desarrollo puede ser muy largo debido a la cantidad de re-trabajo que debe ejecutarse para corregir los errores. En contraste, las organizaciones de Nivel 5 usan un proceso continuo de mejoramiento y técnicas de prevención de errores para

incrementar la eficiencia del proceso y eliminar el costo del re-trabajo, permitiendo que el tiempo de desarrollo sea mas corto. (Esto se ilustra en la Figura 2.3 como un desplazamiento horizontal desde su origen de la línea objetivo)

Los mejoras en la predicción de los resultados del proyecto representados en la Figura 2.3 asume que el resultado del proyecto de software llega ha ser mas predecible mientras que el ruido o re-trabajo, se remueva del proceso de software. Sistemas sin precedentes complican el cuadro desde que nueva tecnología y aplicaciones disminuyen la capacidad de proceso al incrementar la variabilidad. Aun en el caso de Sistemas sin precedentes, las prácticas de administración e ingeniería características de organizaciones maduras, ayudan a identificar y direccionar los problemas mas tempranamente durante el ciclo de desarrollo que si ellos lo hubieran detectado en organizaciones menos maduras. Detecciones tempranas de los defectos contribuyen a la estabilidad y rendimiento del proyecto al eliminar el re-trabajo durante las fases finales. La Administración de Riesgos es una parte integral de la Administración de Proyecto en una organización madura. En algunos casos un proceso maduro, significa que los “fallas” del proyecto son identificadas tempranamente durante el ciclo de vida del software y las inversiones en causas perdidas (re-trabajo) son minimizadas.

### **2.2.7 Saltando los Niveles de Madurez**

Cada Nivel de Madurez construye la base seguir a los siguientes niveles a través de una implementación de procesos efectivos y eficientes. Las organizaciones pueden, sin embargo, provechosamente usar los procesos descritos en el nivel de madurez siguiente donde ellos se encuentran. Los procesos de ingeniería, tales como, análisis de requerimientos, diseño, codificación, pruebas, no son discutidos en el CMM hasta el Nivel 3, sin embargo, las organizaciones de Nivel 1 pueden usarlas. Organizaciones de Nivel 1 ó 2 pueden ejecutar la Revisión de Pares (Nivel 3), hacer Análisis de Pareto (Nivel 4), o pilotos para nuevas tecnologías (Nivel 5) de una manera provechosa. Cuando se propone los pasos que una organización debería ejecutar para pasar del Nivel 1 al Nivel 2, frecuentemente una de las

recomendaciones es establecer un grupo de proceso de ingeniería de software, el cual es un atributo de las organizaciones de Nivel 3. A pesar que las mediciones es el foco del Nivel 4, es también una parte integral de los niveles de madurez mas bajos.

Estos procesos no pueden alcanzar su total potencialidad hasta que su apropiada base esta lista. La Revisión de Pares no puede ser totalmente efectiva al menos que ella sea consistentemente implementada. Los niveles de madurez describen los problemas predominantes en un nivel. Los problemas predominantes del Nivel 1 son directivos; otros problemas tienden a estar enmascarados por las dificultades en el planeamiento y administración de los proyectos de software.

Saltear los niveles es contraproducente porque cada nivel forma la base necesaria para alcanzar el siguiente nivel. El CMM identifica los niveles que una organización debe desarrollar para establecer una cultura de excelencia de ingeniería de software.

Una organización de Nivel 1 que esta tratando de implementar un proceso definido (Nivel 3) antes de haber establecido un proceso repetible (Nivel 2) resulta usualmente insatisfactorio porque los administradores de proyecto están saturados por las presiones de los costos y cronogramas. Esta es la razón fundamental para la orientación sobre administración antes que la ingeniería de procesos. Puede parecer fácil definir e implementar un proceso de ingeniería que un proceso de administración (especialmente para los ojos de las personas técnicas), pero sin una disciplina de gestión, la ingeniería de proceso es sacrificada por las presiones del cronograma y costo.

Una organización que esta tratando de implementar in proceso gestionado (Nivel 4) sin la base de un proceso definido resulta usualmente insatisfactorio porque no hay bases comunes para interpretar las mediciones sin un proceso definido. Mientras los datos son coleccionados por proyectos individuales, pocas mediciones tiene un significado útil para otros proyectos, y ellos no incrementan significativamente el entendimiento del proceso de software. Es

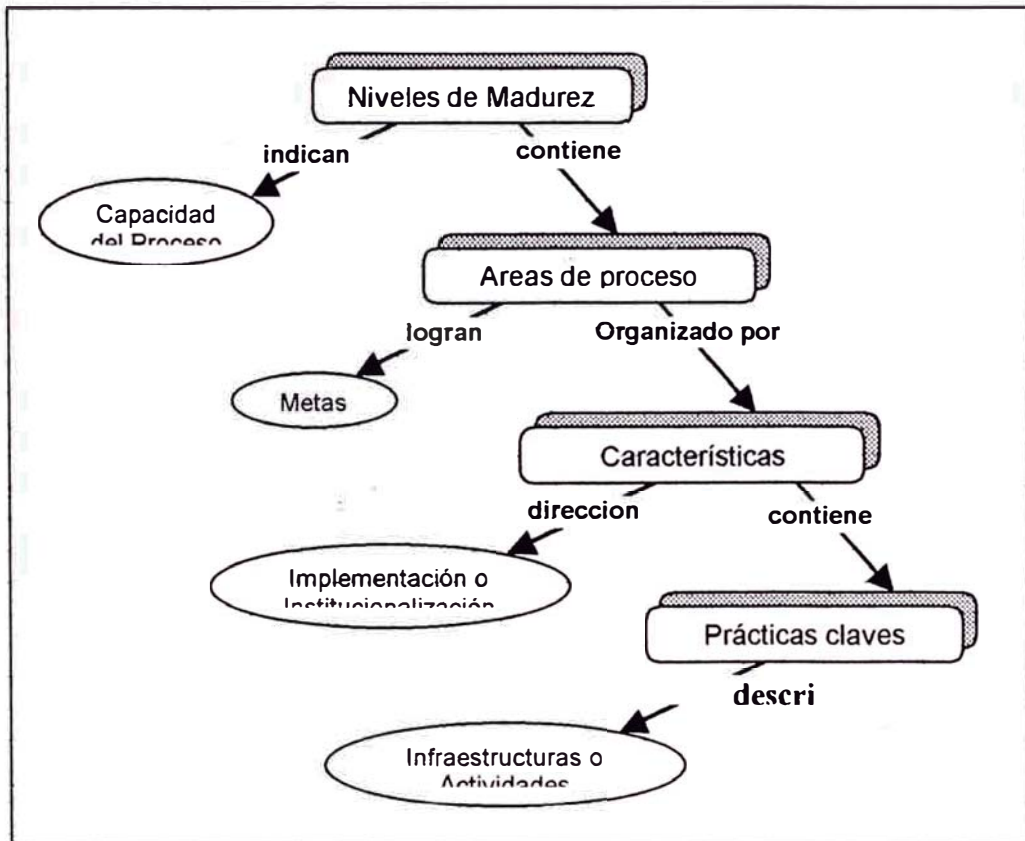
difícil identificar mediciones útiles con la ausencia de procesos definidos porque la variación de los procesos está siendo medida.

Una organización que está tratando de implementar un proceso optimizado (Nivel 5) sin la base de un proceso gestionado (Nivel 4) fallará porque falta el entendimiento del impacto en el cambio de proceso. Sin un control en el proceso hay mucho ruido en los datos para determinar objetivamente si un mejoramiento de un proceso específico tiene efecto. Decisiones pueden degenerar en una guerra porque una pequeña base cuantitativa existe para tomar decisiones racionales y basadas en información.

Los esfuerzos de mejoramiento de procesos deberían enfocarse en las necesidades de la organización dentro del contexto de su entorno de negocio. La habilidad para implementar los procesos de los niveles de madurez altos no implica que se salten los niveles de madurez

#### **2.2.8 Estructura interna de los niveles de madurez**

Cada nivel de madurez se ha descompuesto en partes. Con la excepción del Nivel 1, la descomposición de cada nivel de madurez va desde los resúmenes abstractos de cada nivel bajando hasta su definición operacional en prácticas claves, como se muestra en la Figura 2.4. Cada nivel de madurez está compuesto en varias áreas de proceso clave. Cada área de proceso clave está organizada en cinco secciones llamadas características comunes. Las características comunes especifican las prácticas claves que, cuando colectivamente se ejecutan, se logran las metas del área de proceso clave.



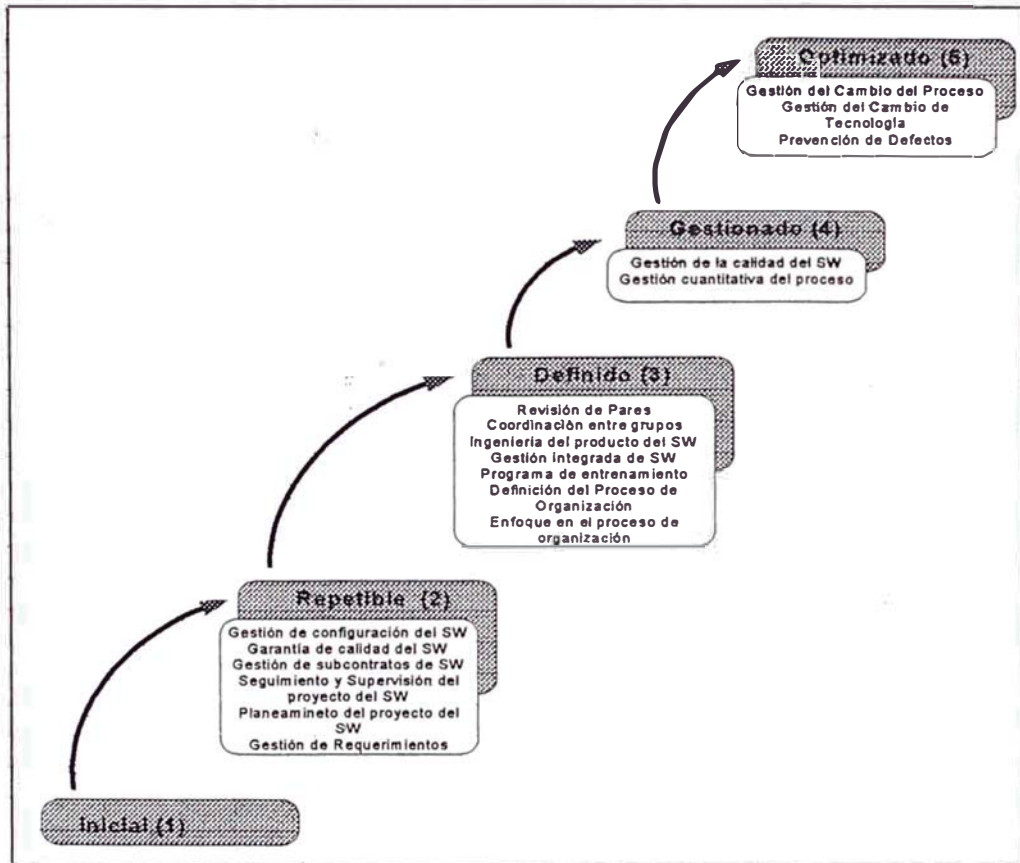
**Figura 2.4 La Estructura del CMM**

### 2.2.9 Áreas de Proceso Clave

Salvo el Nivel 1, cada nivel de madurez se descompone en algunos áreas de proceso que indican las áreas que una organización debe enfocarse para mejorar su proceso de software. Las áreas de proceso clave identifican los problemas que deben ser enfrentados para lograr un nivel de madurez.

Cada área del proceso importante identifica un conjunto de actividades relacionadas que, cuando se realizan colectivamente, logra un juego de metas considerado importante para reforzar la capacidad de proceso. Las áreas de proceso clave se han definido para residir en un solo nivel de madurez como se muestra en la Figura 2.5. El camino para lograr las metas de un área de proceso clave pueden diferir por proyectos basado en las diferencias de los dominios de la aplicación o entornos. No obstante, todas las metas de una área de proceso clave deben alcanzarse por la organización para satisfacer el área de proceso clave. Cuando las metas de un área de proceso clave son cumplidas puede decirse que la organización tiene

institucionalizado la capacidad de proceso caracterizada por el área de proceso clave.



**Figura 2.5 Las Áreas de Proceso Clave por Nivel de Madurez**

El adjetivo "clave" implica que hay áreas de proceso (y procesos) que no son importantes para lograr un nivel de madurez. El CMM no describe todas las áreas de proceso en detalle que involucran el desarrollo y manteniendo de software. Ciertas áreas del proceso se han identificado como importantes para la capacidad del proceso; éstos son los áreas descritas en el CMM.

Aunque otros problemas afectan el rendimiento del proceso, las áreas del proceso clave fueron identificadas debido a su efectividad en el mejoramiento en la capacidad de proceso de software de una organización. Ellos pueden ser considerados los requisitos para lograr un nivel de madurez. La Figura 2.5 muestra las áreas del proceso importantes para cada nivel de madurez. Para lograr un nivel de madurez, las áreas de proceso claves del nivel deben satisfacerse. Para satisfacer una área de proceso clave, cada uno de las metas para el área de proceso clave deben satisfacerse. Las metas resumen



las prácticas claves de un área de proceso clave y pueden usarse para determinar si una organización o proyecto ha llevado a cabo el área de proceso clave eficazmente. Las metas significan el alcance, límites, e intento de cada área de proceso clave.

Las prácticas específicas a ser ejecutadas en cada área de proceso clave evolucionarán al mismo tiempo que la organización logra niveles más altos de madurez de proceso. Por ejemplo, muchas de las capacidades de estimación de proyectos descritas en el área de proceso clave en la gestión de Proyectos de Software en el Nivel 2 deben evolucionar para manejar los datos adicionales disponibles en los Niveles 3, 4, y 5. La Gestión de Software Integrado en el Nivel 3 es la evolución del Planeamiento de Proyectos de Software y del Seguimiento de Proyecto de Software en el Nivel 2 ya que el proyecto es administrado usando un proceso de software definido.

Las áreas de proceso clave del CMM representan una manera de describir cómo las organizaciones maduran. Éstos áreas de proceso clave fueron definidas basándose en muchos años de experiencia en ingeniería y Administración en el desarrollo de software y sobre cinco años de experiencia con las evaluaciones de proceso y capacidades de software.

Las áreas de proceso clave en el Nivel 2 se orientan en las problemas del proyecto de software relacionados con el establecimiento básico del control en la dirección del proyecto. Las descripciones de cada una de las áreas de proceso clave para Nivel 2 se dan a continuación:

El propósito de la Gestión de Requisitos es establecer un común entendimiento entre el cliente y el proyecto de software sobre los requisitos del cliente que serán asignados al proyecto de software. Este acuerdo con el cliente es la base para la planeación (como se describe en la Planificación del Proyecto de Software) y administración (como se describe en Seguimiento del Proyecto de Software) del proyecto de software. El control en la relación con el cliente depende de seguir un eficaz proceso de control de cambios (como se describe en la gestión de la Configuración del Software).

El propósito del Planeamiento del Proyecto de Software es establecer razonables planes para realizar las actividades de ingeniería de software y para administrar el proyecto de software. Estos planes son la base necesaria para la administración del proyecto de software (como se describe en el Seguimiento del Proyecto de Software). Sin planes realistas, la dirección eficaz del proyecto no puede llevarse a cabo.

El propósito del Seguimiento y Supervisión del Proyecto de Software es establecer la visibilidad adecuada en el progreso real para que la dirección pueda tomar las acciones eficaces cuando el rendimiento del proyecto del software se desvía significativamente de los planes de software.

El propósito de la Gestión de Subcontratación de Software es seleccionar a subcontratantes de software calificados y administrarlos eficazmente. Combina lo concerniente a la Gestión de Requisitos, Planificación de Proyectos de Software, y Seguimiento de Proyectos de Software para un control básico de la dirección, junto con la coordinación necesaria del Aseguramiento de la Calidad del Software y la Gestión de Configuración de Software, y aplica estos controles a los subcontratantes de una manera apropiada.

El propósito del Aseguramiento de la Calidad de Software es proporcionar a la dirección con la visibilidad apropiada dentro del proceso que está siendo usado por el proyecto de software y de los productos que están siendo construidos. El Aseguramiento de la Calidad de Software es una parte integral de muchos de los procesos de ingeniería de software y gestión.

El propósito de la Gestión de Configuración de Software es establecer y mantener la integridad de los productos del proyecto a lo largo del ciclo de vida del proyecto de software. La Gestión de Configuración de Software es una parte integral de muchos de los procesos de ingeniería de software y gestión.

Las áreas de proceso clave en el Nivel 3 direccionan ambos problemas, de los proyectos y organizacionales, mientras que la organización establece una

infraestructura que institucionaliza los procesos eficaz de ingeniería de software y de gestión para todos los proyectos. Las descripciones de cada una de las áreas de proceso clave para el Nivel 3 se dan a continuación:

El propósito del Enfoque en el Proceso de la Organización es establecer la responsabilidad organizacional para las actividades de proceso de software que mejoren las capacidades de proceso de software de toda la organización. El resultado primario de las actividades del Enfoque en el Proceso de la Organización es un conjunto de elementos de proceso de software que se describen en La Definición del Proceso de la Organización. Estos recursos se usan por los proyectos de software, como es descrito en la Gestión de Software Integrado.

El propósito de la Definición del Proceso de la Organización es desarrollar y mantener un conjunto utilizable de elementos del proceso de software que mejoren el rendimiento del proceso en los proyectos y proporcione una base para acumular los beneficios a largo plazo de la organización. Estos elementos proporcionan una base estable que pueden ser institucionalizados vía los mecanismos de entrenamiento que se describen en el Programa de Entrenamiento.

El propósito del Programa de Entrenamiento es desarrollar las habilidades y conocimientos de las personas para que ellos pueden realizar sus papeles eficaz y eficientemente. Entrenar es una responsabilidad organizacional, pero los proyectos de software deben identificar las habilidades que necesitan y deben proporcionar el entrenamiento necesario cuando los requerimientos del proyecto sean únicas.

El propósito de la Gestión de Software Integrada es integrar las actividades de ingeniería de software y de dirección en un coherente proceso de software definido que se especifica de acuerdo al proceso de software estándar de la organización y los elementos de proceso relacionados que son descritos en la Definición del Proceso de la Organización. Esta especificación está basada en el ambiente del negocio y las necesidades técnicas del proyecto, como se describe en la Ingeniería de Producto de Software. La Gestión de Software

Integrada evoluciona del Planeamiento del Proyecto de Software y del Seguimiento del Proyecto de Software en el Nivel 2.

El propósito de la ingeniería de Producto de Software es de ejecutar consistentemente un proceso de ingeniería bien definido que integra todas las actividades de ingeniería de software para producir el producto de software correcto, consistentemente, eficazmente y eficientemente. La Ingeniería de Producto de software describe las actividades técnicas del proyecto, por ejemplo, requisitos del análisis, diseño, código, y prueba.

El propósito de la Coordinación de Entre Grupos es establecer un medio para el grupo de ingeniería de software que para participa activamente con los otros grupos de ingeniería para que el proyecto puede satisfacer mejor las necesidades del cliente eficazmente y eficientemente. La Coordinación Entre Grupos es el aspecto interdisciplinario de La Gestión del Software Integrada que se extiende más allá de la ingeniería del software; no sólo debería de ser el proceso de software integrado, sino que la interacción de los grupos de ingeniería de Software con otros grupos debe ser coordinado y controlado.

El propósito de Revisiones Par es quitar los defectos de los productos de software durante el trabajo de manera temprana y eficazmente. Un efecto importante del corolario es desarrollar un mejor entendimiento del producto de software y de los defectos que pueden prevenirse. La revisión par es un método importante y eficaz de ingeniería que se convoca en la ingeniería de Producto de Software.

Las áreas de proceso clave en el Nivel 4 se enfocan en establecer un cuantitativo entendiendo de ambos, el proceso de software y el producto de software que se está construyendo. Las dos áreas de proceso clave a este nivel, la gestión del Proceso Cuantitativo y la Gestión de Calidad de Software, son altamente muy interdependiente, como se describe abajo:

El propósito de la Gestión del Proceso Cuantitativo es controlar la ejecución del proceso del proyecto del software cuantitativamente. La ejecución del proceso representa los resultados reales logrados al seguir un proceso del

software. El enfoque está en identificar las causas especiales de variación dentro de un proceso mesurablemente estable y corregido, así como apropiada, las circunstancias que manejaron la transición. La Gestión del Proceso Cuantitativo agrega un programa de la medición comprensivo a las prácticas de Definición del Proceso de la Organización, Gestión de Software Integrada, la Coordinación Entre Grupos, y Revisiones Par.

El propósito de la Gestión de Calidad de Software es desarrollar un entendimiento cuantitativo de la calidad del software del producto del proyecto y lograr las metas de calidad específicas. La Gestión de Calidad de software aplica un programa de medición comprensivo a los productos de software descritos en la Ingeniería de Producto de Software.

Las áreas del proceso clave del Nivel 5 cubre los problemas que ambos la organización y los proyectos deben dirigirse para llevar a cabo un continuo y medible mejora de proceso de software. Las descripciones de cada uno de las áreas del proceso clave para el Nivel 5 se describen a continuación:

El propósito de la Prevención de Defecto es identificar las causas de defectos y evitar su repetición. El proyecto de software analiza los defectos, identifica sus causas, y cambia su proceso del software definido, como está descrito en la Gestión de Software Integrada. Los cambios del proceso de valor son transmitidos a otros proyectos de software, tal como se describe en la Gestión de Cambio de Proceso.

El propósito de la Gestión de Cambio de Tecnología es identificar las nuevas tecnologías beneficiosas (es decir, herramientas, métodos, y procesos) y transferirlas a la organización de una manera ordenada, como está descrito en la Gestión de Cambio de Proceso. El enfoque de la Gestión de Cambio de Tecnología está en realizar la innovación eficazmente en un mundo cambiante.

El propósito de la Gestión de Cambio de Proceso es mejorar continuamente los procesos de software usados en la organización con el intento de mejorar la calidad del software, la productividad creciente, y disminuir el tiempo del

ciclo de desarrollo del producto. La Gestión de Cambio de proceso toma las mejoras en la Prevención de Defecto y las mejoras innovadoras de la Gestión de Cambio de Tecnología y los hace disponible a la organización entera.

Cada área de proceso clave está descrita en términos de prácticas claves que contribuyen a satisfacer sus metas. Las prácticas claves describen la infraestructura y actividades que más contribuyen a una eficaz implementación e institucionalización del área del proceso clave.

Cada práctica clave consiste en una sola frase, a menudo seguida de un descripción más detallada que puede incluir ejemplos. Estas prácticas clave, también llamado prácticas claves de alto-nivel, declaran las políticas fundamentales, procedimientos, y actividades para el área del proceso clave. Los componentes de la descripción detallada frecuentemente son llamadas subprácticas. La Figura 2.6 proporciona un ejemplo de la estructura que es la base de la práctica clave para el área de proceso clave “Planificación de Proyecto de Software”.

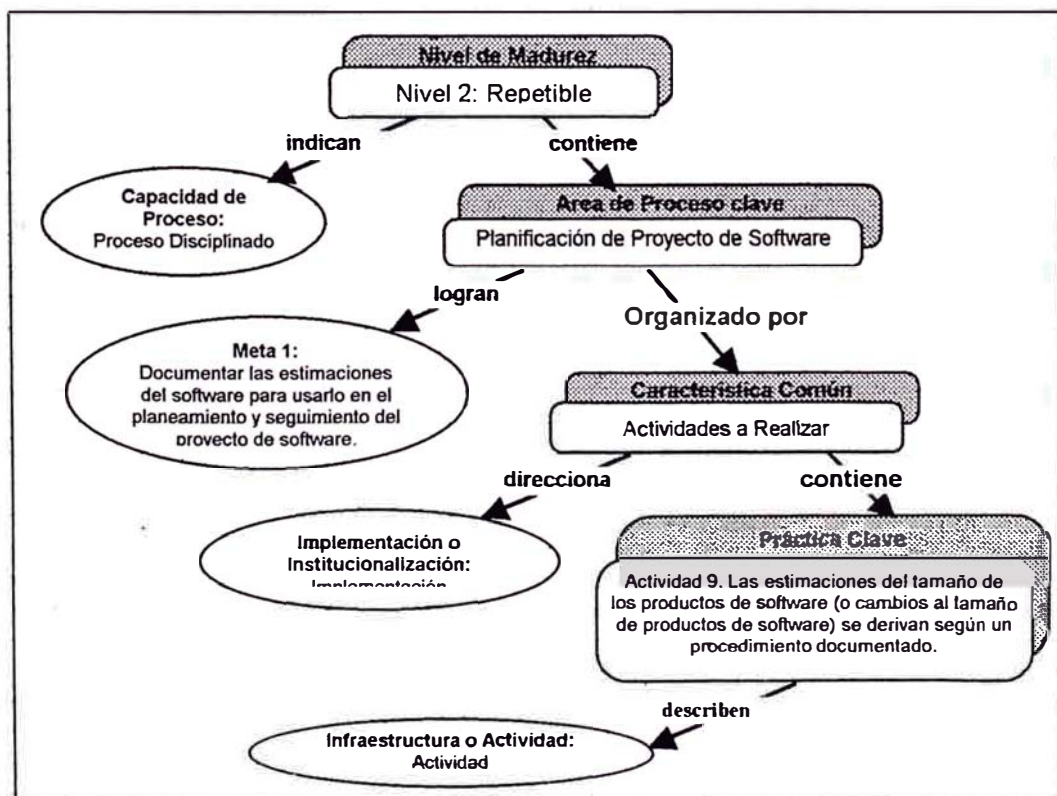


Figura 2.6 Construyendo la Estructura del CMM: Un Ejemplo de una Práctica Clave

Como se ilustra en la Figura 2.6, asegurar el logro consistente de la meta, de documentar los planes para la planeación y seguimiento del proyecto, la organización debe establecer un procedimiento documentado para derivar las estimaciones de tamaño del software. Si estas estimaciones no se desarrollan con un procedimiento documentado, ellos pueden variar ampliamente como las diferencias en las suposiciones a primera vista. La descripción detallada de lo que se esperaría de tal procedimiento incluye el uso de datos históricos, suposiciones documentadas, y revisión de las estimaciones. Este criterio guía el juicio de seguir un razonable procedimiento de estimación de tamaño.

#### **2.2.10 Características Comunes**

Las áreas del proceso clave son organizadas por Características Comunes. Las Características comunes son atributos que indican si la aplicación e institucionalización de un área del proceso clave es eficaz, repetible, y duradera. Los cinco Características comunes se listan a continuación:

##### **El Compromiso a Ejecutar**

El compromiso a Ejecutar describe las acciones que la organización debe tomar para asegurar que el proceso se establezca y sea duradero. El Compromiso a Ejecutar típicamente involucra el establecimiento de las políticas organizacionales y el patrocinio de la dirección superior.

##### **La Habilidad para Ejecutar**

La Habilidad para ejecutar describe las condiciones previas que deben existir en el proyecto u organización para llevar a cabo competentemente el proceso de software. La habilidad para ejecutar típicamente involucra recursos, las estructuras organizacionales, y entrenamiento.

##### **Las Actividades a Ejecutar**

Las Actividades a Ejecutar describen los roles y procedimientos necesarios para implementar un área de proceso clave. Las actividades a ejecutar típicamente involucran el establecimiento de planes y procedimientos, ejecución del trabajo, seguimiento, y la toma de acciones correctivas necesarias.

### **Medición y Análisis**

La medición y el Análisis describe la necesidad de medir el proceso y analizar las mediciones. La medición y el Análisis incluye los ejemplos típicos de las mediciones que podrían tomarse para determinar el estado y la efectividad de las Actividades a ejecutar.

### **Verificando la Implementación**

Verificando la Implementación describe los pasos para asegurar que las actividades se realizan conforme al proceso que se ha establecido. La comprobación típicamente abarca revisiones y auditorías de la dirección y del Aseguramiento de la Calidad de Software.

Las prácticas de la característica común "Actividades a Ejecutar" describen eso que debe llevarse a cabo para establecer una capacidad de proceso. Las otras prácticas, tomadas en conjunto, forman la base para que la organización institucionalice las prácticas descritas en la característica común "Actividades a Ejecutar".

## **2.3 Las Organizaciones Abiertas al Aprendizaje**

Las organizaciones que desarrollan software elaboran proyectos para cambiar o automatizar parte de los procesos de sus clientes. Para llevar a cabo el proyecto, el equipo de desarrollo de software aprende diferentes aspectos de la organización del cliente. El equipo aprende de los requerimientos del cliente, de las limitaciones de la arquitectura a implementar, de los procesos a automatizar, y hasta cómo usar las nuevas tecnologías.

Pero existe otro tipo de aprendizaje que experimentan las personas de una organización y se da cuando los miembros del equipo adquieren nuevas aptitudes que alteran sus actos y entendimiento. Al desarrollar nuevas aptitudes, también desarrollan una nueva percepción. Con el tiempo, a medida que la gente aprende a ver y experimentar el mundo de otra manera, las nuevas creencias cobran forma, lo cual permite a su vez el desarrollo de nuevas aptitudes. Este ciclo de aprendizaje profundo constituye la esencia de

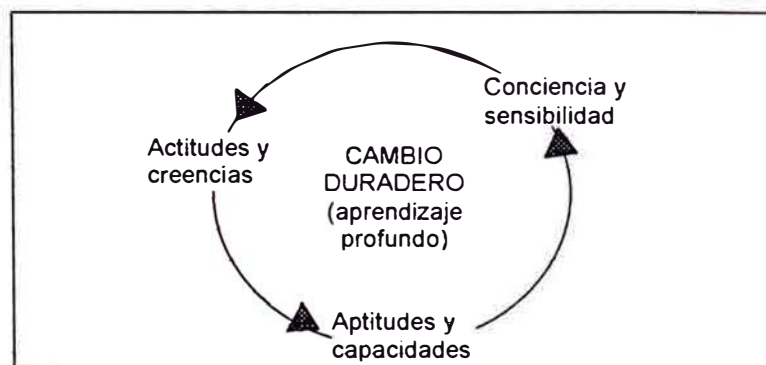


la organización inteligente: no solo el desarrollo de nuevas aptitudes, sino de nuevas perspectivas mentales, tanto individuales como colectivas.

Las características de estas nuevas aptitudes incluyen la capacidad de los individuos, equipos y organizaciones para orientarse hacia sus auténticos intereses, la capacidad para reflexionar sobre premisas profundas y pautas de conducta, tanto individual como colectivamente, desarrollo de aptitudes para el dialogo, capacidad de ver los sistemas y fuerzas que están en juego. (El pensamiento sistémico es vital para adquirirlas.)

Al desarrollarse nuevas aptitudes, el mundo que vemos es diferente, por ejemplo, a medida que nos perfeccionamos en el pensamiento sistémico, comenzamos a ver las estructuras que impulsan la conducta. En situaciones donde antes hubiéramos culpado a los demás, ahora sabemos que existen fuerzas que los han impulsado a actuar de esa manera. Asimismo, teniendo presentes nuestro modelos mentales, somos conscientes de los modos en que continuamente construimos nuestra visión del mundo. A medida que practicamos las disciplinas de dominio personal y visión compartida, tenemos mayor conciencia de la presencia o ausencia de espíritu de una empresa.

Gradualmente, las nuevas percepciones se asimilan reflejándose en cambios básicos en actitudes y creencias. Estos cambios no se producen rápidamente, pero representan una profunda modificación en la cultura de una organización. Las creencias profundas pueden cambiar cuando cambia la experiencia, no basta con proclamar los cambios, y de esta manera si se produce un cambio cultural.



**Figura 2.7 El Ciclo de Aprendizaje Profundo**

En una organización abierta al aprendizaje se deja de creer que una persona debe tener el control para ser eficaz, estamos dispuestos a revelar nuestra incertidumbre, nuestra ignorancia y nuestras limitaciones, sabiendo que son requisitos esenciales para aprender, pues liberan nuestra capacidad innata para la curiosidad, el asombro y la experimentación. Las personas tienen mayor confianza, basada en la experiencia directa del poder de la gente que vive con integridad, apertura, compromiso e inteligencia colectiva, en contraste con culturas empresariales tradicionales basadas en la fragmentación, el secreto y el miedo.

### **2.3.1 Las cinco disciplinas para el aprendizaje**

Las cinco disciplinas de aprendizaje son prácticas que permiten crear organizaciones con capacidades de aprender. Estas cinco disciplinas son: maestría personal, modelos mentales, visión compartida, aprendizaje en equipo y pensamiento sistémico.

La *Maestría Personal* es una disciplina de aspiración que busca formular un cuadro coherente de los resultados que mas desea el individuo (su visión personal), junto con una evaluación realista del estado actual de su vida (su realidad). Aprender a cultivar las tensiones entre visión y realidad puede ampliar la capacidad de la persona para hacer mejores elecciones y alcanzar más de los resultados que ha elegido.

Los *Modelos Mentales* es una disciplina de reflexión y destrezas de indagación que se propone desarrollar conciencia de las actitudes y percepciones que influyen en el pensamiento y la interacción. Reflexionando continuamente sobre estas representaciones interiores del mundo, hablando de ellas y volviéndolas a considerar, se puede adquirir mayor capacidad para gobernar las propias acciones y decisiones.

La *Visión Compartida* es una disciplina colectiva que establece un foco de propósito común. Se aprende a cultivar un sentido de compromiso en un grupo u organización desarrollando imágenes compartidas del futuro que se

busca, y los principios y prácticas por los cuales uno espera guiarse para llegar allí.

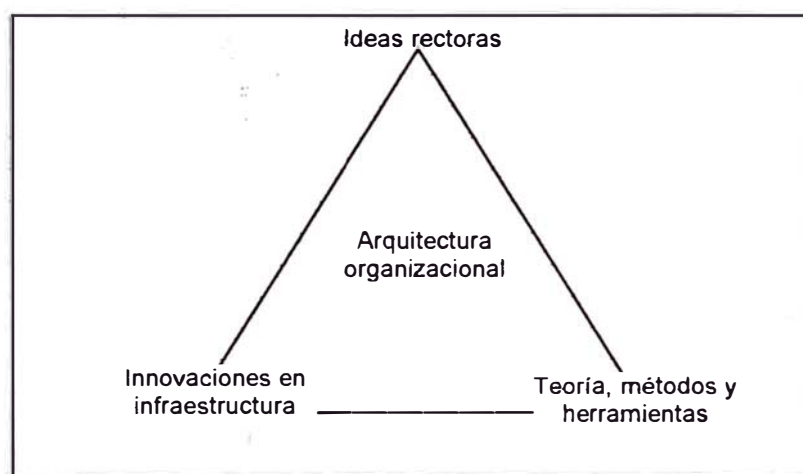
El Aprendizaje en Equipo es una disciplina de interacción en grupo. Por medio de técnicas como diálogo y hábil discusión los equipos transforman su pensamiento colectivo y aprenden a movilizar sus energías y acciones a fin de alcanzar metas comunes y una inteligencia y habilidad mayores que la suma de los talentos individuales de sus miembros.

El Pensamiento Sistémico es una disciplina donde se aprende a entender mejor la interdependencia y el cambio, y por tanto a entendernos mas eficientemente con las fuerzas que norman las consecuencias de nuestros actos. EL pensamiento sistémico se basa en un creciente cuerpo de teoría relativa al comportamiento de la retroalimentación y la complejidad, es decir, las tendencias innatas de un sistema que llevan a crecimiento o estabilidad.

Las organizaciones son producto de la manera como las personas piensan e interactúan. Para cambiar las organizaciones es necesario dar a las personas la oportunidad de cambiar su forma de pensar y de interactuar. Esto no se puede lograr aumentando el entrenamiento ni mediante enfoques administrativos de mando y control. Ninguna persona por si sola, aun cuando sea muy carismático, puede entrenar u ordenar a otros que alteren sus actitudes, creencias, habilidades, capacidades, percepciones o nivel de compromiso. Pro el contrario, la práctica de aprendizaje organizacional desarrolla actividades tangibles: nuevas ideas de administrar, innovaciones en infraestructura y nuevos métodos gerenciales y herramientas para cambiar la manera como las personas realiza su trabajo. Cuando se les da la oportunidad de participar de tomar parte en estas nuevas actividades, los individuos desarrollan una capacidad perdurable de cambio. El proceso recompensará a la organización con niveles muchos mas altos de diversidad, compromiso, innovación y talento.

### 2.3.2 La Arquitectura de las Organizaciones Abiertas al Aprendizaje

El ciclo de aprendizaje profundo es difícil de iniciar. Las aptitudes que suponen nuevos modos de pensar e interactuar necesitan tiempo para madurar. Las nuevas sensibilidades y percepciones de nuestro mundo constituyen un subproducto del crecimiento y el cambio prolongados. Las creencias profundas no son como interruptores que pueden encenderse y apagarse.



**Figura 2.8 Arquitectura de las Organizaciones Abiertas al Aprendizaje**

#### **Ideas Rectoras**

Las ideas rectoras comienzan con la visión, los valores y el propósito que se propone la organización y que desean crear sus integrantes. Las ideas rectoras principales sugieren que las relaciones son más fundamentales que las cosas, y que la totalidad son más importantes que las partes. Las partes no son absolutas, surgen cuando observamos e interactuamos con el fenómeno que estamos observando. En el ámbito organizacional muchas personas están condicionadas para ver nuestras organizaciones como "cosas" y no como sistemas de interacción. Buscamos soluciones que arreglen nuestros problemas como si fueran externos y se pudieran corregir sin arreglar aquello que está en nuestro interior y condujo a su creación. En consecuencia caemos inevitablemente en una espiral incesante de soluciones superficiales, aumentando las dificultades de largo plazo, y en una creciente sensación de impotencia. La idea rectora de la primacía de la totalidad puede constituir el primer paso para ayudar a romper con este círculo vicioso. Otra

idea rectora importante es considerar a la persona como parte de una comunidad y que la persona tiene importancia por su relación con otras personas de quienes podemos aprender y cambiar, y así abrir nuevas posibilidades para ser mas plenamente nosotros mismos. Finalmente otra idea rectora es reconocer la capacidad del lenguaje para renovar nuestra interpretación de la experiencia, y así permitimos suscitar nuevas realidades.

### **Teorías, métodos y Herramientas**

Sin teorías, métodos y herramientas no es posible cambiar la forma de pensar de las personas y crear organizaciones abiertas al aprendizaje. Los métodos y herramientas basados en teorías permiten conocer las situaciones bajo las cuales estas pueden funcionar o fallar, es decir, conocemos las limitaciones de las herramientas. Los métodos y herramientas deberían de ayudar a desarrollar las aptitudes que caracterizan a las organizaciones abiertas al aprendizaje: aspiración, reflexión y dialogo, conceptualización.

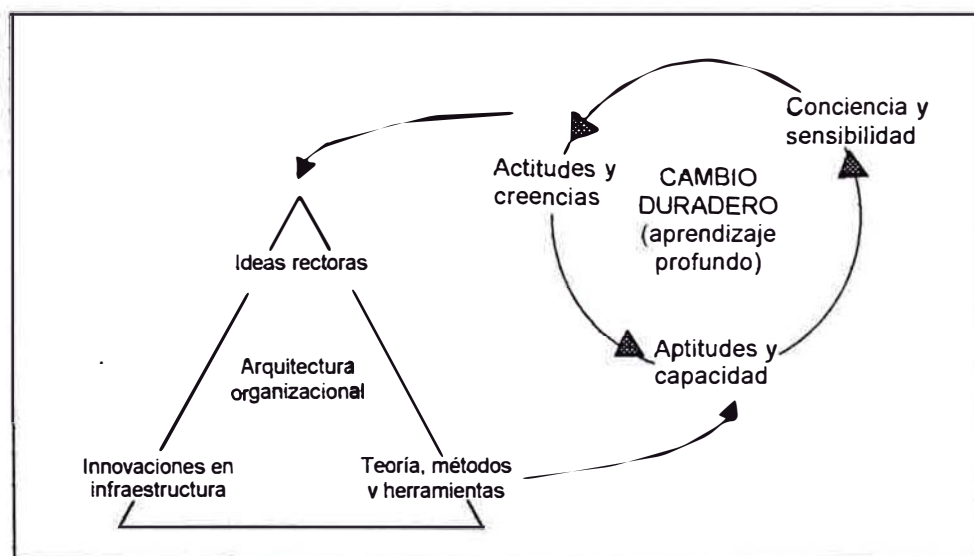
### **Innovaciones en Infraestructura**

La infraestructura es el medio por el cual una organización obtiene recursos disponibles para respaldar a la gente en su trabajo. Las organizaciones abiertas al aprendizaje deben elaborar y mejorar mecanismos infraestructurales para que la gente disponga de los recursos que necesita: tiempo, respaldo gerencial, dinero, información, contacto con colegas y demás. Las innovaciones infraestructurales abarcan una amplia gama de cambios, entre ellas nuevas estructuras empresariales, nuevos diseños de los procesos, nuevos sistemas de remuneración, redes de información. Sin innovaciones en infraestructuras, las ideas inspiradoras y las herramientas eficientes carecen de credibilidad porque las personas no tienen oportunidades ni recursos para seguir su visión ni aplicar las herramientas.

El triangulo de la arquitectura organizacional representa la forma mas tangible de los esfuerzos (el símbolo del triangulo representa la estructura física). En cambio el circulo representa el ciclo de aprendizaje basado en las disciplinas (el circulo no posee comienzo ni fin, es un símbolo del movimiento continuo). La clave de la actividad esta en el triangulo, la causalidad central del cambios esta en el circulo. Ambos ejercen una continua influencia mutua. Juntos

representa los cambios tangibles y sutiles que supone la construcción de organizaciones abiertas al aprendizaje.

Tendemos a suponer, que lo mas tangible es lo mas sustancial y que lo intangible es insustancial. De hecho, sucede lo contrario. Las ideas rectoras que ha expresado una generación de directivos pueden ser modificadas por otras. La infraestructura que se desarrolla hoy puede ser modificada mañana. Un conjunto de herramientas y métodos puede ser suplantado por un nuevo conjunto. La misma razón por la cual nos concentramos en el triángulo –el lugar donde podemos introducir los cambios- también sugiere que esos cambios pueden ser efímeros. En cambio, el ciclo de aprendizaje profundo, que parece tan evanescente e incierto a primera vista, es duradero. Una vez que comenzamos a asimilar el pensamiento sistémico como una manera de ver el mundo, no podemos “desprendernos” del ciclo. Una vez que aprendemos a distinguir nuevos prejuicios de los “datos” en que se basan, cobramos conciencia permanente de nuestro pensamiento. Una vez que comenzamos a operar con un sentido genuino de la visión, adquirimos una comprensión permanente de la diferencia entre reaccionar y crear. Una vez que un grupo ha participado en un verdadero dialogo, sus miembros no lo olvidan. Los cambios que se producen en el ciclo de aprendizaje profundo suelen ser irreversibles.



**Figura 2.9 El ciclo de las Organizaciones Abiertas al Aprendizaje**

## CAPITULO III: PROCESO DE TOMA DE DECISIONES

### 3.1 Planteamiento del Problema

Los grupos de personas que realizan actividades de desarrollo de software, tienen mayores problemas que cualquier grupo u organización. Estos inconvenientes van desde el ámbito personal hasta el organizacional, incluyendo problemas de comunicación, adaptación, competencias, trabajo en equipo, liderazgo, roles no definidos, procesos mal estructurados, cambio constante de la tecnología. Estos problemas dan como resultado aplicaciones que no cumplen con toda la funcionalidad esperada, no satisfacen al usuario, tienen muchos errores, el tiempo de desarrollo y el presupuesto han excedido grandemente el tiempo estimado, difíciles de hacer cambios, sin apropiada documentación, etc.

Desarrollar software confiable, que funcionen como se esperaba, que sea entregado a tiempo y dentro del presupuesto es un esfuerzo difícil para muchas organizaciones. Mientras que los proyectos de software continúan creciendo en tamaño e importancia, estos problemas llegan a ser incontrolables, afectando grandemente a los clientes y a las personas que desarrollan software. Los clientes se dan cuenta de la gran inversión que han hecho cuando el proyecto está muy avanzado y por otro lado los desarrolladores han realizado un gran esfuerzo trabajando bajo mucha presión, en un entorno hostil, y con muy poca credibilidad por parte del cliente.

Estos problemas pueden agruparse dentro los siguientes elementos claves de la organización:

- Procesos
- Personas

- Infraestructura
- Tecnología y Técnicas

### **Procesos**

Los procesos no están documentados o no se siguen o no son conocidos por todos los involucrados. Las actividades de un proceso se llevan a cabo de manera ineficiente generando un re-trabajo. Los objetivos de los procesos contienen inclinaciones de beneficio personal mas que grupal u organizacional. Los roles de las personas no están bien definidos dentro de los procesos originando confusión, males entendidos y re-trabajo. Las actividades definidas en los procesos no generan valor agregado o tienen poca utilidad para las siguientes actividades.

### **Personas**

Las capacidades de las personas no son adecuadas para los roles asignados. Estas deficiencia de capacidades abarca desde los conocimientos técnicos hasta la experiencia en las actividades que realizan. Las actitudes de las personas son individualistas, afectando el trabajo en equipo y el aprendizaje en grupo basado en el reconocimiento de los errores. Puede que las personas sean muy competentes técnicamente pero con muy poca intención de colaboración y hasta con poca veracidad en la información que entregan a los miembros del equipo, desde los desarrolladores a los jefes de proyecto, de los jefes de proyecto a los gerentes de proyecto y de éstos a los clientes.

### **Infraestructura**

La infraestructura de la organización está basada en políticas que incentivan el éxito de las áreas en forma individual, originando que las personas se enfoquen en cumplir solo con ciertas actividades impidiendo la integración de los procesos. La infraestructura no permite cambiar fácilmente los procesos de la organización, ni que las personas aprendan de las experiencias de otras.

### **Tecnología y Técnicas**



Las tecnologías que usan las personas para ejecutar su trabajo y sobre la cual el software a desarrollar, es muy cambiante, comprendiendo lenguajes de programación, arquitecturas, sistemas operativos, etc. Los desarrolladores y administradores de proyectos no usan técnicas eficientes, que permitan la comunicación y ejecución de las actividades de Ingeniería y gestión de proyectos.

A continuación se muestra una lista de los principales problemas en el Desarrollo de Software asociados a los elementos claves de la organización.

Item	Problema	Elemento Clave
1	Procesos no documentados	Proceso
2	Procesos no conocidos por todos los involucrados	Proceso
3	Roles y responsabilidades no están bien delimitados.	Proceso
4	Las actividades definidas en los procesos no generan valor agregado o no son muy útiles para las siguientes actividades.	Proceso
5	Personal de desarrollo trabaja bajo mucha presión.	Proceso
6	Las capacidades de las personas no son adecuadas para los roles asignados.	Persona
7	Los errores en el proyecto nunca se informan hacia niveles superiores, hasta que es demasiado tarde.	Persona
8	Trabajo individualista, poca integración del equipo.	Persona, Infraestructura
9	Alta rotación del personal.	Infraestructura
10	Se usan métodos y herramientas muy complejas.	Personas, Técnicas
11	El personal que participa tiene poca experiencia en las Tecnologías de Información usadas.	Personas, Tecnología
12	Muchos conceptos o términos no están bien definidos.	Técnicas
13	El mantenimiento y mejoramiento del software es dependiente de los desarrolladores.	Técnicas

**Tabla 3.1 Principales Problemas en el Desarrollo de Software relacionados con los elementos claves de una organización**

De la tabla anterior se puede observar que cerca del 80% de los problemas están relacionados con los procesos y las personas.

### **3.2 Alternativas de Solución**

Los principales problemas están relacionados con los procesos y las personas. Para mejorar el desarrollo de software tiene que cambiar la forma de pensar de las personas para que estas cambien las estrategias y puedan crear nuevas capacidades para un cambio continuo. Este cambio continuo solo se puede dar cuando la organización esta capacitada para aprender de sus propias experiencias y de las mejores practicas desarrolladas por otras organizaciones.

Las alternativas de solución que se analizan son las siguientes:

Mejorar los Procesos, usando el Modelo de Madurez de Capacidades de Proceso (SEI).

Mejorar la capacidad de aprendizaje de la organización, usando el modelo de Organizaciones Abiertas al Aprendizaje (MIT).

#### **3.2.1 El Modelo de Madurez de Capacidades de Proceso**

A pesar que los ingenieros de software y administradores frecuentemente conocen en detalle sus problemas, ellos pueden no estar de acuerdo en cuales de los mejoramientos son los más importantes. Sin una estrategia organizada para el mejoramiento, es difícil tener consenso entre la administración y el personal profesional de qué actividades de mejoramiento se llevaran primero. Para obtener los mejores resultados del esfuerzo de mejoramiento de procesos, es necesario diseñar un camino de evolución que incremente la madurez del proceso de software de la organización en fases. El marco de madurez del proceso de software ordena estas fases de tal forma que las mejoras en un nivel proveen las bases para subir al siguiente. De esta manera, una estrategia de mejoramiento obtenida del marco de madurez de proceso de software provee una ruta para un mejoramiento continuo del proceso.

Esta estrategia guía el avance e identifica las deficiencias solo de procesos de la organización, no intenta proveer un rápido arreglo a los problemas del proyecto, es decir, es de mediano plazo.

El Modelo de Madurez de Capacidades para software provee a las organizaciones de software de una guía de como obtener control sobre sus procesos para el desarrollo y mantenimiento de software y como dirigirse hacia una cultura de ingeniería de software y excelencia en la administración. El CMM fue diseñado para guiar a las organizaciones de software en la selección de las estrategias de mejoramiento de procesos, determinando la madurez actual del proceso e identificando los pocos casos más críticos del mejoramiento del proceso y calidad de software. Orientándose sobre un conjunto limitado de actividades y trabajando agresivamente para obtenerla, una organización puede rápidamente mejorar su proceso de software en toda la organización para habilitar un continuo mejoramiento en la capacidad de proceso de software.

La estructura organizada del CMM esta basada en principios de calidad de producto que han existido hace 60 años. Estos principios han sido adaptados por el SEI dentro de un marco de trabajo de madurez que establece una administración de proyectos e ingeniería basadas en el control cuantitativo del proceso de software el cual es la base para el mejoramiento continuo del proceso.

Item	Ventaja	Desventaja
1	Define la situación actual o nivel de madurez del proceso	Identifica las deficiencias solo de procesos
2	Define una camino de evolución para el mejoramiento continuo de los Procesos	Estrategia de mediano plazo
3	Esta basada en las mejores practicas de negocio en Desarrollo de Software	No define como hacerlo sino que hacer

### **3.2.2 El Modelo de Organizaciones Abiertas al Aprendizaje de Peter Senge**

Una organización abierta al aprendizaje se concentra en los factores principales que no pueden ignorarse y sincroniza las estrategias para el éxito de la empresa. Aunque cada entorno empresarial es único, todas las organizaciones desarrollan aptitudes de acuerdo con ciertos lineamientos genéricos. Algunos cambios son de largo plazo, otros pueden iniciarse con rapidez, pero solo asumen importancia duradera en conjunción con otros cambios más lentos. Algunos cambios se pueden lograr directamente, otros son subproductos de otros esfuerzos.

En una organización abierta al aprendizaje, los equipos de personas están bajo ciclos de aprendizaje profundo, donde los miembros adquieren nuevas aptitudes que alteran sus actos y entendimiento que con el tiempo forman nuevas creencias y premisas lo cual permite desarrollar otras nuevas aptitudes. Estas aptitudes se basan en disciplinas llamadas de aprendizaje que consta en la capacidad de las personas para orientarse hacia lo que desean, en la capacidad de reflexionar sobre premisas profundas y pautas de conducta, en la capacidad de sostener conversaciones inteligentes, capacidad de ver los sistemas y fuerzas que están en juego, lo que parece simple desde una perspectiva individual resulta complejo cuando se ve desde la perspectiva de otros.

La construcción de organizaciones abiertas al aprendizaje es el resultado de la práctica constante de las disciplinas de aprendizaje que son parte del ciclo de aprendizaje profundo, y que suceden dentro de un marco, una arquitectura de ideas rectoras, innovaciones en infraestructura y en teoría, métodos y herramientas. Estas ideas rectoras abordan las necesidades superiores de autoestima y autorrealización, consideran que estas ideas evolucionan en el tiempo, que las relaciones son más importantes que las partes y que el lenguaje puede ayudar a describir otras realidades. Las teorías son la base de las herramientas y nos dicen las situaciones bajo el cual pueden funcionar. Para alentar el aprendizaje en una organización se requiere de cambiar la infraestructura, para que las personas dispongan de los recursos que necesita, como el tiempo, información, respaldo gerencial, dinero, contactos con colegas.

Item	Ventaja	Desventaja
1	Orientado a los cambios permanentes en la organización, basado en el aprendizaje organizacional.	Estrategia de mediano y largo plazo.
2	Utiliza herramientas y métodos basadas en teorías.	Identifica solo las deficiencias de las personas y sus interacciones.
3	Crea organizaciones y personas con capacidades de aprender constantemente.	

### 3.3 Metodología de la Solución

Para mejorar las capacidades de proceso se requiere de identificar el estado actual o madurez de los procesos de una organización de desarrollo de software, para luego implementar programas que llevarán a cabo las practicas de mejoramiento de proceso. Estos cambios profundos requieren de mejorar las capacidades de las personas, los programas de mejoramiento requieren de un alto nivel de integración y de cambios en la forma de pensar de las personas, para que estos puedan entender los factores limitativos y que estén mas abiertos al aprendizaje.

Las personas involucradas, tanto gerentes como personal técnico, serán capacitadas para entender el modelo de madurez y las actividades necesarias para implementar un programa de mejoramiento. Al mismo tiempo aprenderán técnicas para mejorar sus objetivos personales y grupales, entiendan los modelos mentales, compartan una misma visión, mejoren la comunicación de equipo, y usen el pensamiento sistémico.

Los cambios se realizaran en las áreas de proceso de clave que son expresadas en un único nivel de madurez. Sin embargo, hay relación entre las áreas de proceso clave y los mejoramientos en una especifica área técnica o de administración. Esta relación se muestra en la figura 3.1. Las organizaciones pueden trabajar en las áreas de proceso clave de un nivel alto antes de que ellas hayan alcanzado los niveles de madurez mas bajos, y

deben de atender a las áreas de proceso clave mas bajas aún cuando hayan alcanzado las áreas de proceso clave mas altas.

Las disciplinas de aprendizaje proporcionan las técnicas y herramientas que darán soporte a las evoluciones de un nivel a otro y dentro de cada nivel. En la figura 3.1 se representan sobre fondo gris. Estas disciplinas facilitarán el aprendizaje, empezando por crear aspiraciones en las personas, definiendo sus objetivos personales y hacia donde quieren llegar, para luego formar una visión común con la organización y mejorar las aptitudes de comunicación y trabajo en equipo, y que en el tiempo las acciones tomadas tendrán sus resultados que serán analizados usando el pensamiento sistémico. El uso de estas disciplinas se dan en los diferentes niveles de madurez de procesos, como prácticas clave adicionales dentro de las actividades de las áreas ya existentes.

<b>Categorías de Proceso Niveles</b>	<b>Gestión</b>	<b>Organización</b>	<b>Ingeniería</b>
5 Optimizado		Administración de Cambio de Tecnología Administración de Cambio de Proceso	Prevención de Defectos Pensamiento Sistémico
4 Administrado	Administración de Procesos Cuantitativos		Administración de Calidad de Software
3 Definido	Administración de Software Integrado Coordinación de Grupos Aprendizaje en Equipo	Orientación al Proceso de la Organización Definición del Proceso de la Organización Programa de Entrenamiento Desarrollo de Visión Compartida Desarrollo de	Ingeniería del producto de Software Revisión de Pares

		Vision Personal Desarrollo de nuevos Modelos Mentales	
2 Repetible	Gestión de Requerimientos Planeamiento de Proyectos de Software Seguimiento y Supervisión de Proyectos de Software Gestión de Subcontratación de Software Gestión de Configuración de Software		
1 Inicial	Proceso Ad-Hoc		

**Figura 3.1 Las Áreas de Proceso Clave asignadas a las Categorías de Proceso**

### 3.3.1 El Proceso de Software de la Organización

La definición del proceso de software es fundamental para obtener los mas altos niveles de madurez. Un concepto fundamental de definición de proceso en el CMM es el proceso de software estándar de la organización, que es la definición operacional de los procesos básicos que guían el establecimiento de un proceso de software común a través de los proyectos de software de la organización. Describe los elementos del proceso de software fundamentales que cada proyecto de software espera para integrarlo dentro de su proceso de software definido. También describe las relaciones (p.e. ordenamiento e interfaces) entre estos elementos de software. Establece un camino consistente para ejecutar las actividades de software a través de la organización que es esencial para una estabilidad y mejoramiento.

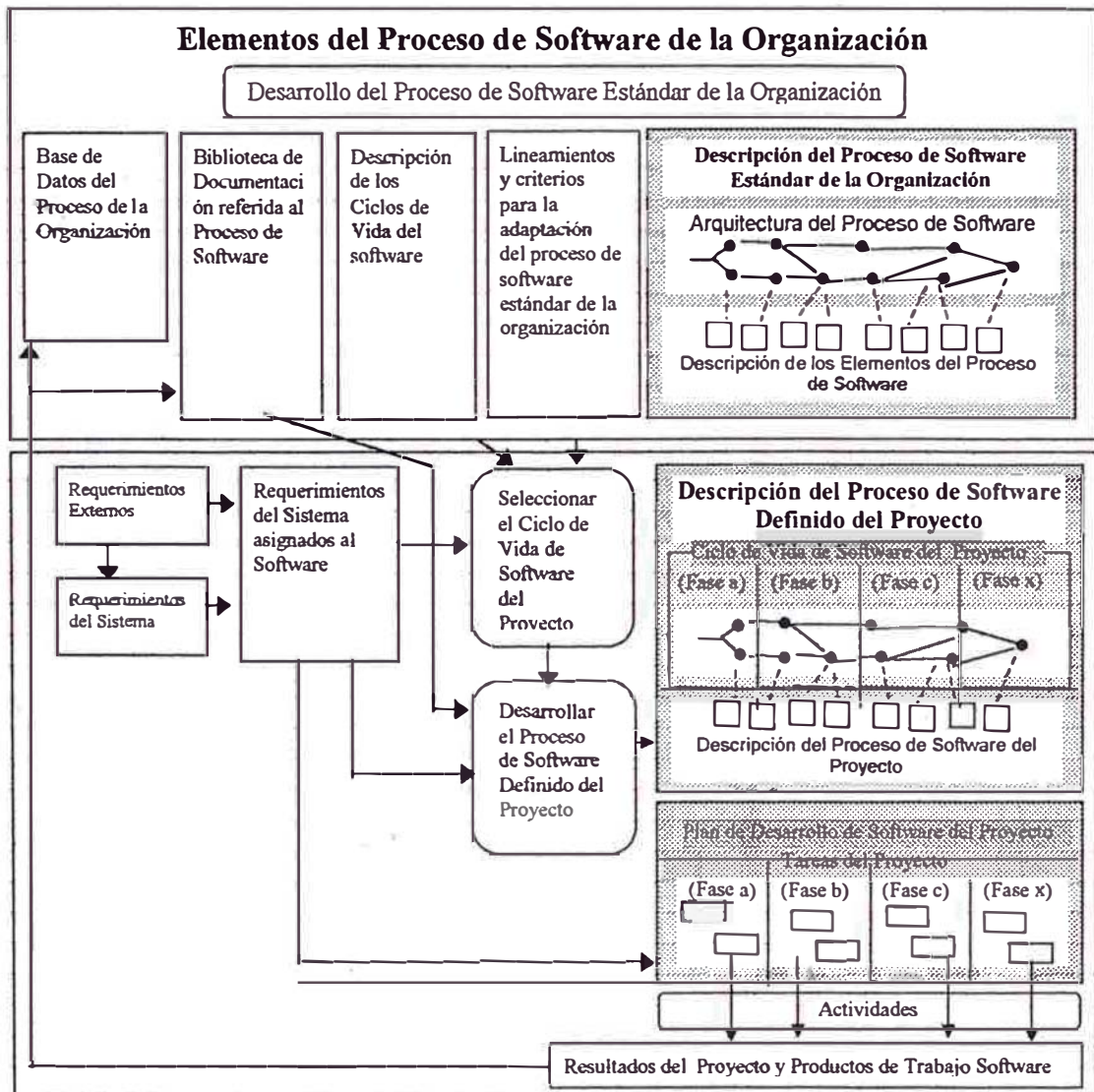
A nivel de la organización, el proceso de software estándar de la organización necesita estar descrito, administrado, controlado y mejorado de una manera formal. A nivel del proyecto, está en la usabilidad del proceso de software

definido del proyecto y el valor que el adiciona al proyecto. El proceso de software definido del proyecto es la definición operacional del proceso de software usado por el proyecto. El proceso de software definido del proyecto es un proceso de software bien caracterizado y entendido, descrito en términos de estándares de software, procedimientos, herramientas y métodos. Es desarrollado para adaptar el proceso de software estándar de la organización de tal manera que encaje las características específicas del proyecto.

Un concepto fundamental es que los procesos pueden ser desarrollados y mantenidos de una manera similar en que los productos son desarrollados y mantenidos. De allí que el proceso tiene:

- Requerimientos que defina lo que el proceso va ha realizar.
- Una arquitectura y diseño que provean información de cómo el proceso será definido.
- Implementación del diseño del proceso en un proyecto o en una situación organizacional.
- Validación de la descripción del proceso vía mediciones, y
- Distribución del proceso dentro de la operación general de la organización o proyecto para el cual el proceso ha sido desarrollado.





**Figura 3.2 Los Elementos del Proceso de Software de la Organización**

La organización establece y mantiene un conjunto de elementos de proceso de software, que incluye:

- El proceso de software estándar de la organización (que incluye la arquitectura y elementos).
- La descripción del ciclo de vida del software aprobado para su uso.
- Los lineamientos y criterios para la adaptación del proceso de software estándar de la organización.
- La Base de Datos del proceso de la organización, y
- La Biblioteca de Documentación referida al Proceso de Software.

Estos elementos están disponibles para uso del proyecto en el desarrollo, mantenimiento e implementación de su proceso de software definido.

### **Proceso de software estándar de la organización**

El proceso de software estándar de la organización es la definición operacional del proceso básico, que guía el establecimiento de un proceso de software común en los proyectos de software de la organización. Describe los elementos de proceso de software que cada proyecto de software espera incorporar dentro de su proceso de software definido. También define las relaciones entre estos elementos del proceso, que algunas veces es referida como la "arquitectura del proceso de software".

Cada elemento del proceso cubre un conjunto de actividades relacionadas, bien definidas, y limitado (p.e. elementos de la estimación de software, elementos del diseño de software, elementos de codificación, elementos de revisión de pares). Las descripciones de los elementos del proceso deben ser plantillas a ser llenadas o completadas.

### **El ciclo de vida del software**

El ciclo de vida del software es el tiempo que empieza cuando el producto software es concebido y termina cuando ya no está disponible para su uso. El ciclo de vida normalmente incluye una fase conceptual, requerimientos, diseño, implementación, pruebas, instalación, operación y mantenimiento y en algunos casos una fase de desinstalación. El ciclo de vida del software puede no ser único para diferentes productos de software, de allí que la organización debe identificar mas de un ciclo de vida para usarlos en los proyectos. El ciclo de vida del software debe estar disponible junto con el proceso de software estándar de la organización para desarrollar el proceso de software definido del proyecto.

### **La base de datos del proceso de software de la organización**

La base de datos del proceso de software de la organización es una base de datos establecida para coleccionar y hacer disponible la data a los procesos de software y de los productos de trabajo de software resultantes. La base de datos contiene los datos de mediciones actuales y la información necesaria

para entender la medición de los datos y las evaluaciones para posteriores consideraciones.

### **Biblioteca de documentación referida al proceso de software**

Una biblioteca de documentación referida al proceso de software es establecida para almacenar los documentos de procesos que podrían ser usados por futuros proyectos, especialmente si están referidos al proceso estándar de software de la organización, y hacerlo disponible a toda la organización. Esta biblioteca puede contener temas tales como procesos de software definido de proyectos, procedimientos, métodos, estándares, planes de desarrollo de software, planes de medición y materiales de entrenamiento de procesos. Esta biblioteca es un recurso importante que puede ayudar a reducir la cantidad de esfuerzo requerido para iniciar un nuevo proyecto, proporcionando ejemplos de proyectos exitosos como punto de inicio.

### **Proceso de Software Estándar Definido del Proyecto**

La descripción del Proceso de Software Estándar Definido del Proyecto es la definición operacional del proceso de software usado por el proyecto, que está bien caracterizado y entendido por los integrantes del proyecto, está descrito en términos de estándares, procedimientos, herramientas y métodos. Este proceso es desarrollado adaptando el proceso de software estándar de la organización para que encaje con las características específicas del proyecto, y seleccionando un ciclo de vida de los que han sido aprobados por la organización.

El Proceso de Software Estándar Definido del Proyecto provee la base para el planeamiento, ejecución y el mejoramiento de las actividades de los administradores y del personal técnico el ejecutar las tareas del proyecto. Es posible que un proyecto tenga mas de un proceso de software definido (p.e. uno para las parte operacional del software y otro para el soporte de las pruebas).

### **Fases**

Una fase es una partición del esfuerzo del software que es de un tamaño manejable y que representa un conjunto significativo y medible de actividades

relacionados las cuales son ejecutadas en un proyecto. Una fase es considerada usualmente como una división del ciclo de vida del proyecto y frecuentemente terminada con una revisión formal antes de pasar a la siguiente fase.

### **Tareas**

El trabajo a ser ejecutado es dividido en tareas. Una tarea es una unidad de trabajo bien definida dentro del proceso de software que provee administrativamente un punto de chequeo visible del estado del proyecto. Las tareas tienen criterios para iniciar o precondiciones y criterios de terminación o postcondiciones. Dentro del contexto de definición de proceso una tarea es un componente bien definido de un proceso definido. Todas las tareas pueden ser consideradas actividades, pero no todas las actividades están suficientemente bien definidas para ser consideradas tareas. Es por eso que en el nivel 2 se evita el término *Tarea* y se usa el de *actividad*.

### **Actividades**

Una actividad es cualquier paso o función ejecutada, físico o mental hacia la obtención de un objetivo. Las actividades incluyen todo el trabajo que los administradores y personal técnico hacen para ejecutar las tareas del proyecto y de la organización.

### **Productos de Trabajo de Software (resultados del proyecto)**

El resultado de las actividades y tareas consisten principalmente de productos de trabajo de software. Un producto de trabajo de software es cualquier artefacto creado como parte de la definición, mantenimiento o uso del proceso de software, que incluye las descripciones del proceso, planes, procedimientos, programas de computadora y documentación asociada, la cual puede o no ser entregada al cliente o usuario final. Los productos de trabajo llegan a ser la entrada al siguiente paso en el proceso o proveen información a archivar para proyectos futuros. El conjunto de productos de trabajo de software que son entregados al cliente o usuario final son referidos como productos de software.

### **Plan de Desarrollo del Software**

La descripción del Proceso de Software Definido del Proyecto no es suficiente para ejecutarse directamente. Sin embargo, la descripción usualmente identifica cosas tales como los roles y los tipos de productos de trabajo de software necesarios para ejecutar una tarea, el no especifica las personas quienes asumirán los roles, los productos de trabajo de software específicos que serán creados, ni el cronograma para ejecutar las tareas y actividades.

### **3.3.2 Estructura Organizacional**

#### **Grupo de Ingeniería de Software**

El Grupo de Ingeniería de Software está formado tanto por gerentes como personal técnico quienes tienen responsabilidad para las actividades de desarrollo de software y de mantenimiento (p.e. Análisis de requerimientos, diseño, código, y pruebas) para un proyecto.

#### **Grupo de Proceso de Ingeniería de Software**

El grupo de proceso de Ingeniería de Software es el grupo de especialistas quienes facilitan la definición, mantenimiento y mejora del proceso de software usado por la organización. En las prácticas claves, este grupo es generalmente referido como “el grupo responsable de las actividades del proceso de software de la organización”.

#### **Grupo de Ingeniería de Sistemas**

El grupo de Ingeniería de Sistemas está formado por gerentes y personal técnico quienes tienen la responsabilidad de especificar los requerimientos del sistema; asignar requerimientos para el hardware, el software y otros componentes; especificar interfaces entre el hardware, software y otros componentes; y monitorear el diseño y desarrollo de estos componentes para asegurar la conformidad con sus especificaciones.

#### **Grupo de Pruebas**

Este grupo está formado por gerentes y personal técnico quienes tienen responsabilidad en la planeación y el desarrollo de las pruebas del software de sistemas independientes para determinar si el producto de software satisface los requerimientos.

### **Grupo de Aseguramiento de la Calidad**

Formado por gerentes y personal técnico quienes planean e implementan el aseguramiento de la calidad de los proyectos para asegurar que los pasos y estándares del proceso de software sean seguidos.

### **Grupo de Gestión de la Configuración del Software**

Formado por gerentes y personal técnico quienes tienen la responsabilidad de planear, coordinar e implementar las actividades de gestión de la configuración del software en los proyectos.

### **Grupo de Entrenamiento**

Es un conjunto de individuos (gerentes y personal técnico) quienes son responsables de coordinar y arreglar las actividades de entrenamiento para una organización. Este grupo típicamente prepara y conduce la mayoría de los cursos de entrenamiento y coordina el uso de otras formas de entrenamiento.

## **3.3.3 Prácticas Claves Nivel 2**

### **Área de proceso clave: Gestión de Requerimientos**

El propósito es establecer un entendimiento común entre el cliente y los requerimientos que han de ser implementados en el proyecto de software. Este acuerdo está referido a cómo los requerimientos del sistema están asignados al software y cubre tanto los requerimientos técnicos como los no técnicos. Este acuerdo forma la base para la estimación, planeamiento, ejecución y seguimiento de las actividades del proyecto de software a través de todo su ciclo de vida. Si los requerimientos son cambiados, los planes de software, productos de Software y las actividades afectadas deben ser ajustadas para guardar consistencia con los requerimientos actualizados.

#### **Metas**

1: Los requerimientos de sistema son controlados para establecer una línea base para la ingeniería de Software y uso gerencial.

2: Mantener la consistencia entre los planes de software, productos, actividades y requerimientos.

#### Compromisos a ejecutar

1: El proyecto sigue una política organizacional escrita para administrar los requerimientos especificados en el proyecto de software. Se requiere que los requerimientos sean documentados, revisados por el gerente de software y otros grupos afectados.

#### Habilidades para ejecutar

1: Para cada proyecto se establece una responsabilidad en el análisis de los requerimientos del sistema y la ubicación de ellos en componentes de software, hardware y otros.

2: Los requerimientos identificados son documentados. Esto incluye a requerimientos no técnicos, requerimientos técnicos y los criterios de aceptación que será usados para validar que el producto de software satisfaga los requerimientos.

3: Se debe contar con los recursos adecuados para la gestión de requerimientos tales como personal experto en el dominio de la aplicación y herramientas para soportar estas actividades.

4: Capacitación de grupos de ingeniería de Software y otros grupos relacionados para realizar actividades de gestión de requerimientos.

#### Actividades ejecutadas

1: El grupo de ingeniería de software revisa los requerimientos ubicados antes de que éstos sean incorporados en el proyecto.

2: El grupo de ingeniería de software usa los requerimientos identificados como una base para los planes de software, productos del trabajo y actividades.

3: Los cambios de requerimientos son revisados e incorporados en el proyecto de software. En donde el impacto de los compromisos existentes es fijado y los cambios son negociados apropiadamente.

#### Mediciones y análisis

1: Las mediciones son realizadas para determinar el estado de las actividades para la gestión de requerimientos.

#### Verificación de la implementación

1: Las actividades para la gestión de requerimientos son revisadas con la gerencia senior sobre una base periódica.

2: Las actividades para la gestión de requerimientos son revisadas por la gerencia del proyecto sobre una base periódica y manejada por eventos.

3: El grupo de aseguramiento de la calidad de software revisa y/o audita las actividades y productos de trabajo para la gestión de requerimientos y reporta los resultados.

#### **Área de proceso clave: Planeamiento del Proyecto de Software**

El propósito del Planeamiento del Proyecto de Software es establecer planes razonables para la realización de Ingeniería de Software y la gestión del proyecto de software. El planeamiento involucra realizar estimaciones para el trabajo a ser realizado estableciendo compromisos necesarios y definiendo el plan de trabajo a realizarse. El proceso de planeamiento incluye pasos para estimar el tamaño de los productos de trabajo de software y los recursos necesarios para producir un cronograma e identificar, fijar riesgos de software y negociar compromisos. La iteración entre estos pasos puede ser necesaria para establecer un plan para el proyecto de software.

#### Metas

1: Las estimaciones de Software son documentadas para ser usadas en el planeamiento y seguimiento del proyecto de software.

2: Las actividades y compromisos de un proyecto de software son planeadas y documentadas.

3: Los grupos o individuos afectados aceptan sus compromisos relacionados con el proyecto de software.

#### Compromisos a ejecutar

1: Asignación de un gerente del proyecto de software para responsabilizarse de la negociación de compromisos y desarrollo del plan de desarrollo del proyecto de software.



2: El proyecto sigue una política organizacional escrita para el planeamiento del proyecto de software.

#### Habilidades para ejecutar

1: Debe existir un informe de trabajo documentado y aprobado para el proyecto de software.

2: Se asignan responsabilidades para el desarrollo del plan de desarrollo del software.

3: Se proveen fondos y recursos adecuados para el planeamiento del proyecto de software

4: Los gerentes de software, ingenieros de software y otros individuos involucrados en el planeamiento del proyecto de software son entrenados en estimación de software y planeamiento de procedimiento aplicables a sus áreas de responsabilidad.

#### Actividades ejecutadas

1: El grupo de Ingeniería de Software debe participar en la preparación de la propuesta, clarificación de discusiones además de la revisión y negociación de cambios para compromisos que afectan al proyecto.

2: El planeamiento del proyecto de software es iniciado en las etapas iniciales y en paralelo con el planeamiento general del proyecto de software.

3: El grupo de Ingeniería de software participa con otros grupos afectados en el planeamiento general del proyecto durante el ciclo de vida del mismo.

4: Los compromisos del proyecto de software hechos a individuos o grupos externos a la organización son revisados por el gerente senior de acuerdo a un procedimiento documentado.

5: Se identifica o define un ciclo de vida de software con etapas predefinidas y de tamaño manejable.

6: El plan de desarrollo del proyecto se lleva de acuerdo a un procedimiento documentado. Está basado según los estándares del cliente, del proyecto, un informe de trabajo aprobado y requerimientos identificados.

7: El plan para un proyecto de software es documentado. Esto cubre el propósito, ámbito, metas y objetivos del proyecto de software; selección del ciclo de vida, identificación de métodos y estándares para el desarrollo y/o mantenimiento del software, tamaño estimado, estimación de esfuerzos y

costos, estimación de recursos computacionales críticos, identificación de riesgos, entre otros.

8: Se identifican productos de trabajo de software necesarios para establecer y mantener el control del proyecto de software.

9: Las estimaciones para el tamaño del producto de trabajo de software son derivadas de acuerdo a un procedimiento documentado.

10: Estimaciones de esfuerzo para el proyecto de software y costos son derivadas de acuerdo a la documentación.

11: Las estimaciones de recursos computacionales críticos son derivadas de acuerdo a procedimientos documentados.

12: Determinación del cronograma del proyecto de software en base a un procedimiento documentado, éste debe estar relacionado al tamaño estimado para el producto de trabajo de software, esfuerzo y costos.

13: Se identifican y documentan los riesgos del software asociado a los costos, recursos, cronograma y aspectos técnicos del proyecto.

14: Se preparan planes para facilitar la ingeniería de software del proyecto y soporte de herramientas.

15: Los datos para el planeamiento de software son registrados. La información grabada incluye los estimados y la información asociada para reconstruir los estimados y límites razonables.

#### Medición y análisis

1: Las mediciones son realizadas y usadas para determinar el estado de las actividades del planeamiento del software.

#### Verificación de la implementación

1: Las actividades para el planeamiento del proyecto de Software son revisadas por la gerencia senior sobre una base periódica.

2: Las actividades para el planeamiento del proyecto de software son revisadas por la gerencia del proyecto sobre una base periódica y manejada por eventos.

3: El grupo de aseguramiento de la calidad de software revisa y/o audita las actividades y productos de trabajo para el planeamiento del proyecto de software y reporta los resultados.

### **Área de proceso clave: Seguimiento y Supervisión del Proyecto de Software**

El propósito es proporcionar una visibilidad adecuada del progreso actual de tal forma que la gerencia pueda tomar acciones efectivas cuando la ejecución del proyecto de software se desvíe significativamente de los planes del proyecto de software.

Se debe tener un plan documentado del proyecto para usarlo como base para el seguimiento de las actividades de software, comunicando el estado y revisión de los planes. El avance es determinado básicamente comparando el tamaño real del software, esfuerzo, costo y tiempo con el plan del proyecto.

#### **Metas**

- 1: Los resultados y ejecución actuales son comparados contra los planes de software.
- 2: Tomar y manejar las acciones correctivas cuando los resultados se desvíen significativamente de los planes de Software.
- 3: Cambios a los compromisos de software son aceptados por los grupos e individuos afectados.

#### **Compromisos a ejecutar**

1: Un Gerente de proyecto de Software se designa para ser responsable de las actividades y resultados del proyecto de software.

2: El proyecto sigue una política organizacional escrita para la gestión del proyecto de Software:

Un Plan de Desarrollo documentado se usa y se mantiene como base para rastrear el proyecto.

El Gerente del proyecto se mantiene informado del estado y temas concernientes al proyecto.

Acciones correctivas son tomadas cuando el plan de desarrollo de software no es alcanzado

Los cambios en los compromisos de Software se hacen con la participación y acuerdos de los grupos afectados.

#### **Habilidades para ejecutar**

1. Aprobar y documentar un plan de desarrollo de software.

- 2: El Gerente de del proyecto de software asigna los responsables para las actividades y los productos de trabajo de software.
- 3: Se proveen recursos adecuados y consolidados para el seguimiento del proyecto de software.
- 4: Los gerentes del software son entrenados en el manejo de aspectos técnicos y de personal.
- 5 : Los gerentes del software de primera línea reciben orientación en aspectos técnicos del proyecto de software.

#### Actividades ejecutadas

- 1: Un plan de desarrollo de software documentado es usado para el seguimiento de las actividades de software y la comunicación de su estado.
- 2: El plan de desarrollo del proyecto de software se revisa según un procedimiento documentado. El plan de desarrollo se revisa para incorporar refinamientos y cambios, especialmente cuando este cambia significativamente. El plan de desarrollo se actualiza para incorporar todo nuevo compromiso del proyecto o cambio de compromisos.
- 3: Compromisos y cambios a los compromisos hechos a individuos y grupos externos a la organización son revisados con el gerente senior de acuerdo a un procedimiento documentado.
- 4: Cambios a los compromisos aprobados que afectan al proyecto de software son comunicados a los grupos de ingeniería de software y a otros grupos relacionados.
- 5: El tamaño de los productos del trabajo de software es seguido y se toman acciones correctivas de ser necesarias.
- 6: Se hace un seguimiento al esfuerzo y el costo de proyecto de software y se toman acciones correctivas de ser necesarias.
- 7: Se hace un seguimiento a los recursos computacionales críticos y se toman acciones correctivas de ser necesarias.
- 8: Se hace un seguimiento a los cronogramas del proyecto de software y se toman acciones correctivas de ser necesarias.
- 9: Se hace un seguimiento a las actividades técnicas de ingeniería de software y se toman acciones correctivas de ser necesarias.
- 10: Se hace un seguimiento a los riesgos del software asociados con el costo, recursos, cronogramas y aspectos técnicos del proyecto.

11: Se registran datos de mediciones actuales y de datos modificados para el proyecto de software.

12: El grupo de ingeniería de software dirige las revisiones periódicas internas para hacer el seguimiento del progreso técnico, planes, ejecución contra el plan de desarrollo de software

13: Se conducen revisiones formales para dirigir los logros y resultados hacia los objetivos del proyecto.

#### Mediciones y análisis

1: Las mediciones son realizadas y usadas para determinar el estado de las actividades de seguimiento y supervisión del proyecto de software.

#### Verificación de la implementación

1: Las actividades de seguimiento y supervisión del proyecto de software son revisadas por la gerencia Senior sobre una base periódica.

2: Las actividades para seguimiento y supervisión del proyecto de software son revisadas por la gerencia del proyecto sobre una base periódica y manejada por eventos.

3: El grupo de seguimiento y supervisión del proyecto de software revisa y/o audita las actividades y productos de trabajo para Seguimiento y supervisión del proyecto de software y reporta los resultados.

#### **Área de proceso clave: Aseguramiento de la Calidad de Software (SQA)**

Provee a la gerencia de apropiada visibilidad dentro del proceso que está siendo usado por el proyecto de software y de los productos que están siendo construidos. Implica revisar y auditar los productos y las actividades para verificar que concuerdan con los procedimientos y estándares aplicables. El equipo de Garantía de calidad trabaja con el proyecto en sus etapas iniciales para establecer planes, estándares procedimientos que agregarán valor al proyecto, satisfarán las restricciones del proyecto y las políticas de la organización. El grupo de Aseguramiento de calidad de software revisa las actividades del proyecto y audita los productos del trabajo a lo largo del ciclo de vida

#### Metas

- 1: Las actividades del Aseguramiento de calidad del Software son planeadas.
- 2: La adherencia de los productos de software y las actividades para los estándares aplicables, procedimientos y requerimientos, es objetivamente verificada.
- 3: Los grupos e individuos afectados son informados de las actividades y resultados del SQA.
- 4: Los problemas de disconformidad que no se puedan resolver en el proyecto de software son dirigidos a la gerencia senior.

#### Compromisos a ejecutar

- 1: El proyecto sigue una política organizacional escrita para implementar SQA.
- 2: El grupo de SQA tiene un canal del reporte para la gerencia senior que es independiente de: el gerente del proyecto, el grupo de ingeniería de software del proyecto y de los otros grupos relacionados.

#### Habilidades para ejecutar

- 1: Existe un grupo que es responsable de coordinar e implementar SQA en el proyecto.
- 2: Recursos y fondos adecuados son proveídos para realizar las actividades del SQA.
- 3: Los miembros del grupo de SQA son entrenados para realizar sus actividades.
- 4: Los miembros del grupo de software reciben orientación en el rol, responsabilidades, autoridad y valor del grupo de SQA.

#### Actividades ejecutadas

- 1: Un plan de SQA es preparado para el proyecto de software de acuerdo a un procedimiento documentado.
- 2: Las actividades del grupo de SQA se realizan en concordancia con el plan SQA.
- 3: El grupo de SQA participa en la preparación y revisión del plan de desarrollo de software del proyecto, estándares y procedimientos.
- 4: El grupo de SQA revisa las actividades de ingeniería de software para verificar la conformidad.

5: El grupo de SQA audita los productos del trabajo de software designado para verificar conformidad.

6: El grupo de SQA periódicamente reporta los resultados de sus actividades al grupo de ingeniería de software.

7: Las desviaciones identificadas en las actividades de software y los productos de trabajo del software son documentadas y manejadas de acuerdo a un procedimiento documentado.

8: El grupo de SQA conduce revisiones periódicas de sus actividades y hallazgos con el personal de SQA del cliente, mientras sea apropiado.

#### Mediciones y análisis

1: Se realizan mediciones y son usadas para determinar el costo y el estado del cronograma de las actividades de SQA.

#### Verificación de la implementación

1: Las actividades de SQA son revisadas con la gerencia senior en una base periódica.

2: Las actividades de SQA son revisadas con el gerente del proyecto en una base periódica y basada en eventos.

3: Expertos independientes del grupo de SQA periódicamente revisan las actividades y los productos del trabajo de software del grupo del grupo SQA del proyecto.

#### **Área de proceso clave: Gestión de la Configuración del Software (SCM).**

El propósito de la gestión de configuración de software es establecer y mantener la integridad de los productos del proyecto de software a lo largo del ciclo de vida del proyecto. Involucra la identificación de la configuración del software en puntos dados en el tiempo, los cambios sistemáticamente controlados de la configuración y el mantenimiento de la integridad e identificación de la configuración a lo largo del ciclo de vida.

#### Metas

1: Las actividades de gestión de la configuración de software son planeadas.

2: Los productos del trabajo del software seleccionados son identificados, controlados y disponibles.

3: Los cambios a productos de trabajo de software identificados son controlados.

4: Los grupos e individuos afectados son informados del estado y contenido de las bases del software.

#### Compromisos a ejecutar

1: El proyecto sigue una política organizacional escrita para implementar SCM.

#### Habilidades para ejecutar

1: Un panel con la autoridad de manejar las bases del software del proyecto existe o es establecido.

2: Un grupo responsable de coordinar e implementar SCM para el proyecto existe.

3: Adecuados recursos y fondos son proveídos para realizar las actividades de SCM.

4: Los miembros del grupo de SCM son entrenados en los objetivos, procedimientos y métodos para realizar sus actividades de SCM.

5: Los miembros del grupo de ingeniería de software y otros grupos relacionados con software son entrenados para realizar sus actividades de SCM.

#### Actividades ejecutadas

1: Un plan de SCM es preparado para cada proyecto de software de acuerdo a un procedimiento documentado.

2: Un plan documentado y aprobado de SCM es usado como base para realizar las actividades de SCM.

3: Un sistema de librería de la gestión de configuración es establecido como un repositorio para las bases del software.

4: Los productos del trabajo del software que serán puestos bajo gestión de configuración son identificados.

5: Los pedidos de cambio y los reportes de problemas de todos los elementos o unidades de la configuración son iniciados, registrados, revisados, aprobados y rastreados de acuerdo a un procedimiento documentado.



6: Los cambios a las bases son controlados de acuerdo a un procedimiento documentado.

7: Los productos de la librería de las bases del software son creados y su descarga es controlado de acuerdo a un procedimiento documentado.

8: El estado de la configuración de los elementos o unidades son registrados de acuerdo a un procedimiento documentado.

9: Reportes estándar que documentan las actividades de SCM y los contenidos de las bases del software son desarrollados y puestos a disposición de los grupos e individuos afectados.

10: Las auditorias a las bases del software son conducidas de acuerdo a un procedimiento documentado.

#### Mediciones y análisis

1: Se realiza mediciones y son usadas para determinar el estado de las actividades de SCM.

#### Verificación de la implementación

1: Las actividades de SCM son revisadas con la gerencia senior en una base periódica.

2: Las actividades de SCM son revisadas con el gerente del proyecto en una base periódica y manejada por eventos.

3: El grupo de SCM periódicamente audita las bases del software para verificar que estén conformes con la documentación que las define.

4: El grupo de SQA revisa y/o audita las actividades y los productos de trabajo de SCM y reporta los resultados.

#### **Área de proceso clave: Gestión de Subcontratos de Software**

La gestión de subcontratos de software involucra la selección de un subcontratante de software, estableciendo los compromisos con el subcontratante y rastreando y revisando el rendimiento y los resultados del subcontratante.

El subcontratante se selecciona según su habilidad de realizar el trabajo.

La planeación del software, seguimiento y vigilancia de las actividades para el trabajo subcontratado se realizan por el subcontratante.

### Metas

- 1: El contratista principal selecciona a los subcontratantes calificados para el software.
- 2: El contratista principal y el subcontratante del software acuerdan sus compromisos.
- 3: El contratista principal y el subcontratante mantienen una comunicación constante.
- 4: El contratista principal rastrea los resultados reales del subcontratante y su rendimiento, versus los compromisos.

### Compromisos a ejecutar

- 1: El proyecto sigue una política organizacional documentada para administrar el subcontrato de software.
- 2: Se designa un gerente de subcontrato para que sea el responsable de establecer y administrar el subcontrato de software.

### Habilidades para ejecutar

- 1: Se proveen recursos adecuados y financiamiento para seleccionar al subcontratante de software y administrar el subcontrato.
- 2: Los gerentes de software y otros individuos involucrados en establecer y administrar el subcontrato de software son entrenados para desarrollar estas actividades.
- 3: Los gerentes de software y otros individuos involucrados en administrar el subcontrato de software reciben orientación en los aspectos técnicos del subcontrato.

### Actividades ejecutadas

- 1: El trabajo a ser subcontratado se define y planea según un procedimiento documentado.
- 2: El subcontratante de software se selecciona basado en una evaluación de la habilidad de los postores del subcontrato para realizar el trabajo, según un procedimiento documentado.
- 3: El acuerdo contractual entre el contratista principal y el subcontratante de software se usa como base para administrar el subcontrato.

- 4: El plan de desarrollo de software documentado del subcontratante es revisado y aprobado por el contratista principal.
- 5: El plan de desarrollo de software documentado y aprobado del subcontratante es usado para el seguimiento de las actividades de software y el estado de la comunicación.
- 6: Los cambios al informe de trabajo del subcontratante de software, términos del subcontrato y condiciones, y otros compromisos son resueltos de acuerdo a un procedimiento documentado.
- 7: La gerencia del contratista principal guía revisiones de coordinación/estado periódicas con la gerencia del subcontratante de software.
- 8: Se sostienen revisiones técnicas periódicas con el subcontratante de software.
- 9: Se conducen revisiones formales para dirigir los logros de ingeniería del subcontratante de software y los resultados en hitos seleccionados de acuerdo a un procedimiento seleccionados.
- 10: El grupo de aseguramiento de la calidad del contratista principal de software monitorea las actividades del aseguramiento de calidad del subcontratante de software de acuerdo a un procedimiento documentado.
- 11: El grupo de gestión de la configuración de software del contratista principal de software monitorea las actividades de la gestión de la configuración de software del subcontratante de software de acuerdo a un procedimiento documentado.
- 12: El contratista principal conduce pruebas de aceptación como parte de la entrega del producto del subcontratante de software de acuerdo a un procedimiento documentado.
- 13: El rendimiento del subcontratante de software es evaluado sobre un base periódica, y la evaluación es revisada con el subcontratante.

#### Mediciones y análisis

- 1: Se realizan mediciones y son usadas para determinar el estado de las actividades para la gestión de subcontratos de software.

#### Verificación de la implementación

- 1: Las actividades para la gestión del subcontrato de software son revisadas por la gerencia senior sobre una base periódica.

- 2: Las actividades para la gestión del subcontrato de software son revisadas por el gerente del proyecto sobre una base periódica y manejada por eventos.
- 3: El grupo de aseguramiento de la calidad revisa y/o audita las actividades y el producto del trabajo para la gestión del subcontrato de software y reporta los resultados.

### **3.3.4 Prácticas Claves Nivel 3**

#### **Área de proceso clave: Enfoque de Proceso de la Organización**

El propósito del Enfoque del Proceso de la Organización es establecer las responsabilidades organizacionales para las actividades del proceso de software que mejoren todas las capacidades del proceso de software de la organización. El enfoque del proceso de la organización involucra el desarrollar y mantener el entendimiento de los procesos de software de los proyectos y de la organización y coordinar las actividades para asignar, desarrollar, mantener y mejorar dichos procesos.

La organización provee los compromisos y los recursos necesarios para coordinar el desarrollo y mantenimiento de los procesos de software, a través de los proyectos de software actuales y futuros mediante un grupo, como lo es el grupo de ingeniería de software

#### **Metas**

- 1: Las actividades de mejoramiento y desarrollo del proceso de software son coordinadas a lo largo de la organización
- 2: Las fortalezas y debilidades del proceso de software usado, son identificadas en relación con un estándar de procesos
- 3: Las actividades de mejoramiento y desarrollo de procesos al nivel organizacional son planeadas.

#### **Compromisos a ejecutar**

- 1: La organización sigue una política organizacional escrita para la coordinación las actividades de mejoramiento y desarrollo de procesos de software a lo largo de la organización.
- 2: La Gerencia Senior patrocina las actividades de la organización para el mejoramiento y desarrollo del proceso de software

3: La Gerencia Senior supervisa las actividades de la organización para el mejoramiento y desarrollo de los procesos de software

#### Habilidades para ejecutar

1: Existe un grupo responsable de las actividades de los procesos de software de la organización.

2: Fondos y recursos adecuados son proveídos para las actividades de los procesos de software de la organización

3: Los miembros del grupo responsable de las actividades de procesos de software de la organización reciben el entrenamiento requerido para realizar estas actividades

4: Los miembros del grupo de ingeniería de software y otros grupos relacionados reciben orientación sobre las actividades de procesos de software de la organización y sus roles en estas actividades

#### Actividades ejecutadas

1: El proceso de software es evaluado periódicamente y se desarrollan planes de acción para direccionar los hallazgos de la evaluación.

2: La organización desarrolla y mantiene un plan para sus actividades de mejoramiento y desarrollo de procesos de software.

3: Las actividades de la organización y del proyecto para mejorar y desarrollar sus procesos de software son coordinados a nivel organizacional.

4: El uso de la base de datos de procesos de la organización es coordinado al nivel organizacional.

La base de datos del proceso de software de la organización es utilizada para recolectar información de los procesos de software y los productos resultantes de software de la organización y del proyecto.

5: Los nuevos procesos, métodos y herramientas en uso limitado en la organización son monitoreados, evaluados y, si es apropiado, son transferidos a otras partes de la organización

6: El entrenamiento para los procesos de software de la organización y del proyecto es coordinado a lo largo de la organización

7: Los grupos involucrados en implementar los procesos de software son informados de las actividades de la organización y de los proyectos para la mejora y desarrollo del proceso de software

#### Mediciones y análisis

1: Las mediciones son hechas y utilizadas para determinar el estado de las actividades de mejoramiento y desarrollo de los procesos de la organización

#### Verificación de la implementación

1: Las actividades de mejoramiento y desarrollo de procesos de software son revisadas con la Gerencia Senior en una base periódica

#### **Área de proceso clave: Definición del Proceso de la Organización**

El Propósito es desarrollar y mantener un conjunto utilizable de recursos de procesos de software que mejoren el desempeño del proceso en los proyectos y mantengan una base para beneficios acumulativos, a largo plazo para la organización.

La Definición del Proceso de la Organización involucra:

Desarrollo y mantenimiento del proceso de software estándar de la organización, junto con las cualidades relativas al proceso, como la descripción del ciclo de vida de software, guías de medición de procesos, la base de datos de procesos de software de la organización, y una librería, de documentación relativa al proceso de software.

#### Metas

1: Se desarrolla y mantiene un proceso de software estándar para la organización.

2: Se reúne, revisa y se hace disponible la Información correspondiente uso del proceso de software estándar de la organización mediante los proyectos del software

#### Compromisos a ejecutar

1: La organización sigue una política escrita para desarrollar y mantener un proceso del software estándar y los recursos del proceso relacionados.

#### Habilidades para ejecutar

1: Se proporcionan recursos adecuados y fondos para el mejoramiento y desarrollo del proceso de software estándar de la organización y los recursos del proceso relacionados.

2: Quienes desarrollan y mantienen los procesos de software estándar de la organización y los recursos de procesos relacionados reciben el entrenamiento requerido para realizar estas actividades.

#### Actividades ejecutadas

1: El proceso de software estándar de la organización se desarrolla y mantiene según un procedimiento documentado.

2: El proceso de software estándar de la organización se documenta según los estándares establecidos por la organización.

3: Las descripciones de ciclos de vida de software que son aprobadas para su uso por los proyectos se documentan y mantienen.

4: Se desarrollan y mantienen pautas y criterios para la elaboración de proyectos de software estándar de la organización.

5: Se establece y mantiene la base de datos de procesos de software de la organización.

6: Se establece y mantiene una librería de documentación de procesos de software relacionados.

#### Mediciones y análisis

1: Las mediciones son hechas y utilizadas para determinar el estado de las actividades de definición del Proceso de la Organización.

#### Verificación de la implementación

1: El grupo de Aseguramiento de Calidad de Software revisa y/o audita las actividades de la organización y los productos de trabajo para desarrollar y mantener el proceso de software estándar de la organización y los recursos de procesos relacionados e informa los resultados.

#### **Área de proceso clave: Programa de Entrenamiento**

El propósito es desarrollar las habilidades y conocimiento de los individuos para que puedan realizar sus roles efectiva y eficientemente.

El Programa de Entrenamiento involucra primero identificar el entrenamiento necesario para la organización, proyectos, e individuos, desarrollando o procurando el entrenamiento para dirigirse a las necesidades identificadas. Cada proyecto de software evalúa las habilidades actuales y futuras necesarias y determina cuando estas habilidades se obtendrán. Algunas habilidades son impartidas efectiva y eficientemente a través de medios informales, mientras que otras habilidades necesitan medios de entrenamiento más formales para ser impartidas efectiva y eficientemente.

#### Metas

- 1: se planean Las actividades de entrenamiento.
- 2: Se proporciona entrenamiento para desarrollar las habilidades y conocimiento necesarios para realizar la administración del software y los roles técnicos.
- 3: Los miembros del grupo de Ingeniería del Software y los grupos de software relacionados reciben el entrenamiento necesario para cumplir sus roles.

#### Compromisos a ejecutar

- 1: La organización sigue una política escrita para encontrar sus necesidades de entrenamiento

#### Habilidades para ejecutar

- 1: Existe un grupo responsable para el cumplimiento de las necesidades de entrenamiento de la organización
- 2: Se proporcionan recursos adecuados y financiamiento para implementar el Programa de Entrenamiento.
- 3: Los miembros del grupo de entrenamiento tienen las habilidades y conocimiento necesarios para desempeñar sus actividades de entrenamiento.
- 4: Los Gerentes de Software reciben orientación en el Programa de Entrenamiento.

#### Actividades ejecutadas

- 1: Cada proyecto de software desarrolla y mantiene un plan de entrenamiento que especifica sus necesidades de entrenamiento.



2: El plan de entrenamiento de la organización se desarrolla y revisa según un procedimiento documentado.

3: El entrenamiento para la organización se lleva a cabo de acuerdo con el Plan de Entrenamiento de la organización

4: Los cursos de entrenamiento preparados al nivel de la organización se desarrollan y mantienen según los estándares de la organización.

5: Un procedimiento de renuncia para el entrenamiento requerido se establece y se usa para determinar si los individuos ya poseen el conocimiento y las habilidades requeridas para realizar sus roles designados.

6: Se mantienen registros de entrenamiento.

#### Mediciones y análisis

1: Las mediciones se realizan y usan para determinar el estado de las actividades del Programa de Entrenamiento.

2: Las mediciones se realizan y usan para determinar la calidad del programa de entrenamiento.

#### Verificación de la implementación

1: Las actividades del Programa de Entrenamiento son revisadas con la Gerencia Senior en una base periódica.

2: El Programa de Entrenamiento es evaluado independientemente en una base periódica para consistencia con relevancia a las necesidades de la organización.

3: Las actividades del Programa de Entrenamiento y los productos del trabajo son revisados y/o auditados y los resultados son informados.

#### **Área de proceso clave: Gestión de Software Integrado**

El propósito de Gestión del Software Integrado es integrar el software que se diseña y las actividades de gestión en un proceso del software coherente, definido que se adapta al proceso de software estándar de la organización y a los recursos de procesos relacionados, que son descritos en la Definición del Proceso de la Organización.

La gestión del software integrado involucra el desarrollo y gestión del proyecto de software que usa el proceso de software definido. El proceso del software

definido del proyecto se adapta del proceso estándar de la organización para dirigir las características específicas del proyecto.

Las prácticas básicas para estimación, planeamiento, y seguimiento de un proyecto de software se describe en las áreas de proceso clave: Planeamiento del Proyecto de Software y Seguimiento y Supervisión del proyecto de software. Ellos enfocan en reconocer los problemas cuando éstos ocurren y ajustan los planes y/o actúan para dirigirlos.

#### Metas

- 1: El proceso del software definido del proyecto es una versión adaptada del proceso del software estándar de la organización.
- 2: El proyecto se planea y gestiona según el proceso de software definido del proyecto.

#### Compromisos a ejecutar

- 1: El proyecto sigue una política organizacional escrita que requiere que el proyecto de software se planee y gestione usando el software estándar de la organización y los recursos de proceso relacionados.

#### Habilidades para ejecutar

- 1: Recursos adecuados y financiamiento son proporcionados para gestionar el proyecto de software que usa el proceso de software definido del proyecto.
- 2: Los individuos responsables de desarrollar el proceso de software definido del proyecto reciben el entrenamiento requerido en cómo adaptar el proceso de software estándar de la organización y usar los recursos de proceso relacionados.
- 3: Los gerentes de software reciben el entrenamiento requerido en la gestión de los aspectos técnicos, administrativos, y de personal del proyecto de software, basados en el proceso del software definido del proyecto.

#### Actividades ejecutadas

- 1: El proceso de software definido del proyecto es desarrollado adaptando el proceso del software estándar de la organización según procedimiento documentado.

- 2: El proceso de software definido de cada proyecto se revisa según un procedimiento documentado.
- 3: El plan de desarrollo de software del proyecto que describe el uso del proceso del software definido de los proyectos es desarrollado y es revisado según un procedimiento documentado.
- 4: El proyecto de software es gestionado de acuerdo al proceso de software definido del proyecto.
- 5: La base de datos del proceso de software de la organización se usa para el planeamiento y estimación del software.
- 6: El tamaño de los productos de software (o el tamaño de los cambios a los productos del software) es manejado de acuerdo a un procedimiento documentado.
- 7: El esfuerzo y costos de software del proyecto son gestionados de acuerdo a un procedimiento documentado.
- 8: Los recursos críticos de cómputo del proyecto se gestionan según un procedimiento documentado.
- 9: Las dependencias y rutas críticas del cronograma del software del proyecto son gestionados según un procedimiento documentado.
- 10: Los riesgos del software del proyecto se identifican, evalúan, documentan y gestionan según un procedimiento documentado.
- 11: Revisiones del proyecto de software se realizan periódicamente para determinar las acciones necesarias para proporcionar el rendimiento y resultados del proyecto de software en línea con las necesidades actuales y las proyectadas del negocio, cliente y usuarios finales.

#### Mediciones y análisis

1: Las mediciones son hechas y usadas para determinar la efectividad de las actividades de Gestión de software integrada.

Como ejemplos de dimensiones pueden ser:

El esfuerzo consumido sobre el tiempo para gestionar el proyecto de software, comparado con el plan.

La frecuencia, causas y magnitud del esfuerzo de volver a planear.

#### Verificación de la implementación

- 1: Las actividades para gestión del proyecto de software son revisadas con la Gerencia Senior en un base periódica
- 2: Las actividades para gestión del proyecto de software son revisadas con la Gerencia Senior tanto en una base periódica y manejada por eventos.
- 3: El grupo de aseguramiento de la calidad revisa y/o audita las actividades y productos para gestionar el software del proyecto e informe los resultados.

### **Área de proceso clave: Ingeniería del Producto de Software**

El propósito de Ingeniería de Producto de Software es realizar consistentemente un proceso de ingeniería bien definido que integre todas las actividades de ingeniería de software para producir productos de software correctos, consistentes, efectiva y eficientemente.

Debe considerarse también que Ingeniería del Producto de Software incluye tareas como el análisis de los requisitos del sistema, junto con el desarrollo de la arquitectura del software, la codificación, la integración de componentes, y las pruebas de software, verificando que satisfaga los requerimientos.

La documentación del software ( donde se diseñan las tareas, los requisitos del software, plan de la prueba, y procedimientos de la prueba) se desarrolla y revisa para asegurar que cada tarea sea consecuente con los resultados de tareas predecesoras y los resultados producidos son apropiados para las tareas siguientes incluyendo las tareas de operación y mantenimiento de software.

Cuando los cambios son aprobados los productos de software afectados, planes, compromisos, procesos y actividades son revisados para reflejar los cambios aprobados.

#### **Metas**

- 1: Las tareas involucradas en la ingeniería del producto de software se definen, se integran, de forma consistente, para desarrollar un producto de software.
- 2: Los productos de software deben ser consistentes entre sí.

#### **Compromisos a ejecutar**

- 1: El proyecto sigue una política organizacional para realizar las actividades del proyecto de software.

Esta política especifica típicamente:

Las tareas de ingeniería de software son desarrolladas de acuerdo al proceso del software definido del proyecto.

Se usan métodos y herramientas apropiadas para construir y mantener los productos del software.

Los planes, tareas, y productos de software, son identificables con respecto a los requisitos del sistema asignados al software.

Habilidades para ejecutar

1: Se proveen los recursos adecuados y financiamiento, que permitan realizar las tareas de ingeniería de software.

2: Los miembros del personal técnico de ingeniería de software reciben el entrenamiento requerido para realizar sus asignaciones técnicas.

3: Los miembros del personal técnico reciben orientación en algunas disciplinas relacionadas a la ingeniería de software.

4: El jefe del proyecto y todos los jefes involucrados reciben orientación en los aspectos técnicos del proyecto de software.

Actividades Ejecutadas

1: Los métodos y herramientas de ingeniería de software apropiados son integrados al proceso de desarrollo del software definido del proyecto.

2: Los requisitos del software son desarrollados, mantenidos, documentados, y verificados, analizando los requisitos asignados sistemáticamente según el proceso del software definido del proyecto.

3: El diseño del software es desarrollado, mantenido, documentado, y verificado, según el proceso del software definido del proyecto, para adecuarse a los requerimientos del software y formar el marco de trabajo para la codificación..

4: El código del software es desarrollado, mantenido, documentado, y verificado, según el proceso definido en el proyecto, para implementar los requisitos del software y el diseño del mismo.

5: Las pruebas de software son desarrolladas según el proceso definido en el proyecto de software.

6: Las pruebas de integración de software son planeadas y desarrolladas según el proceso definido en el proyecto de software.

7: Pruebas del sistema y aceptación del software se realizan para demostrar que el software satisface sus requerimientos.

8: La documentación que se utilizara para operar y mantener el software es desarrollada y mantenida según el proceso definido en el proyecto de software.

9: Los datos con defectos identificados en las revisiones del par y probados son reunidos y analizados según el proceso de software definido del proyecto. Los tipos de datos a ser coleccionados y analizados son: la descripción del defecto, la categoría del defecto, la severidad del defecto, unidades que contienen el defecto, unidades afectadas por el defecto, actividad dónde el defecto fue introducido, revisión del par o casos de prueba que identificaron el defecto, los resultados reales que identificaron el defecto.

10: La consistencia es mantenida a través del trabajo con los productos de software, incluyendo los planes de software, las descripciones del proceso, los requisitos asignados, los requisitos del software, la codificación, los planes y procedimientos de la prueba.

#### Mediciones y análisis

1: Las medidas son hechas y utilizadas para determinar la funcionalidad y calidad de un producto de software.

2: Las medidas son hechas y utilizadas para determinar el status de las actividades de ingeniería de software.

#### Verificación de la implementación

1: Las actividades del proceso de ingeniería de software son revisadas por el Gerente Senior, en una base periódica.

2: Las actividades del proceso de ingeniería de software son revisadas por el administrador del proyecto, en una base periódica.

3: Las revisiones del grupo y/o auditorías para el aseguramiento de la calidad de software deben ser reportadas en los informes de resultados

#### **Área de proceso clave: Coordinación entre Grupos**

El objetivo es establecer una comunicación fluida entre los grupos de ingeniería de software. El propósito de Coordinación entre Grupos es establecer los medios necesarios para que el grupo de ingeniería de software

participe activamente con los otros grupos de ingeniería, con el objetivo de que el proyecto pueda satisfacer las necesidades del cliente eficazmente. Se planean las interfaces activas técnicas e interacciones entre los grupos y se manejan para asegurar la calidad e integridad del todo el sistema. Se dirigen revisiones técnicas e intercambios regularmente con representantes del proyecto para asegurar que todos los grupos de la ingeniería son conscientes del estado y planes de todos los grupos.

#### Metas

- 1: Los requerimientos del cliente son acordados por todos los grupos afectados.
- 2: Los compromisos entre los grupos de Ingeniería son acordados por todos los grupos afectados.
- 3: Los grupos de ingeniería identifican, siguen, y resuelven los problemas entre grupos.

#### Compromisos a ejecutar

- 1: El proyecto sigue una política organizacional establecida por los equipos de ingeniería interdisciplinarios.

#### Habilidades para ejecutar

- 1: Se proveen los recursos adecuados y financiamiento, que permitan realizar las tareas de ingeniería de software, coordinando con los grupos de ingeniería.
- 2: Las herramientas de apoyo usadas por los diferentes grupos de ingeniería son compatibles, para permitir una comunicación y coordinación efectiva.
- 3: Todos los gerentes en la organización reciben la capacitación requerida para el trabajo en equipo.
- 4: Todos los líderes de tareas en cada grupo de ingeniería reciben orientación en los procesos, métodos, y estándares usados por los otros grupos de ingeniería.
- 5: Los miembros de los grupos de ingeniería reciben la orientación de trabajo en equipo.

#### Actividades ejecutadas

- 1: El grupo de ingeniería de software y los otros grupos de ingeniería participan con el cliente y usuarios finales para establecer los requerimientos del sistema.
- 2: Los representantes del grupo de ingeniería de software del proyecto trabajan con los representantes de los otros grupos de ingeniería, para supervisar y coordinar las actividades técnicas y resolver los problemas técnicos.
- 3: Un plan documentado se usa para comunicar los compromisos entre grupos, coordinar y seguir el trabajo realizado.
- 4: Se identifican las dependencias críticas entre los grupos de ingeniería, se negocian, y se rastrean según un procedimiento documentado.
- 5: Los resultados producidos como entrada a otros grupos de ingeniería son revisados por representantes de los grupos receptores para asegurar que los productos satisfacen sus necesidades.
- 6: Los temas no resueltos por los representantes individuales de los grupos de ingeniería se manejan según un procedimiento documentado.
- 7: Los representantes del grupo de ingeniería del proyecto dirigen revisiones técnicas e intercambios periódicos.

#### Mediciones y análisis

- 1: Las mediciones son hechas y utilizadas para determinar el estado de las actividades de la coordinación entre grupos.

#### Verificación de la implementación

- 1: Las actividades de coordinación entre grupos son revisadas por el gerente senior del proyecto, en una base periódica.
- 2: Las actividades de coordinación entre grupos son revisadas por el gerente del proyecto, tanto en una base periódica como manejada por eventos.
- 3: Los grupos de aseguramiento de calidad de software hacen las revisiones y/o auditorías a las actividades y productos para la coordinación entre grupos e informa los resultados.

#### **Área de proceso clave: Revisión de Pares**

El propósito de la revisión de pares es remover temprana y eficientemente los defectos de los productos de trabajo de software. Un importante corolario es



desarrollar un mejor entendimiento de los productos de trabajo de software y de los defectos que pueden ser prevenidos.

Revisión de pares involucra un examen metódico de los productos de trabajo de software para identificar defectos y áreas donde los cambios sean necesarios. Los productos específicos que pasan por una revisión son identificados en el proceso de software definido del proyecto y programados como parte de las actividades de planeamiento del proyecto de software.

### Metas

1: Se planean las actividades de la revisión de pares.

2: Se identifican y remueven los defectos en los productos de trabajo de software.

### Compromisos a ejecutar

1: El proyecto sigue una política organizacional escrita para realizar las revisiones.

Esta política típicamente especifica lo siguiente:

La organización identifica un estándar para un conjunto de productos de trabajo de software que van a pasar por revisiones.

Cada proyecto identifica los productos de trabajo de software que van a sufrir revisiones.

Las revisiones son dirigidas por líderes entrenados en revisiones.

Las revisiones se enfocan en productos de trabajo de software y no en las personas que los realizaron, es decir no se evalúa la ejecución de las personas.

### Habilidades para ejecutar

1: Se proveen adecuados recursos y fondos para poder realizar las revisiones, para:

Preparar y distribuir materiales de revisión

Liderar la revisión

Revisar los materiales

Participar en la revisión y cualquier revisión posterior requerida basada en los defectos detectados en la revisión.

Monitorear la puesta al día de los productos de trabajo de software basados en los defectos encontrados en la revisión.

Recolectar y reportar los datos resultantes de la revisión.

2: Los líderes de la revisión reciben el entrenamiento adecuado en cómo liderar las revisiones.

3: Las personas que participan en las revisiones reciben adecuado entrenamiento en los objetivos, principios y métodos de las revisiones.

#### Actividades ejecutadas

1: Las revisiones son planeadas y los planes son documentados. Estos planes:

Identifican los productos de trabajo de software que van a ser revisados.

Especifican el cronograma para las revisiones. Para las revisiones que son programadas para un futuro cercano, se identifican a los líderes y a otros participantes.

2: Las revisiones son realizadas de acuerdo a un procedimiento documentado. Este procedimiento típicamente especifica que:

Las revisiones sean planeadas y dirigidas por líderes entrenados.

Los materiales de revisión sean distribuidos previamente a las personas que va a hacer la revisión de tal manera que puedan prepararse adecuadamente.

Los revisores hayan asignado roles en las revisiones.

Los criterios de prontitud y finalización de las revisiones sean especificados, reforzados, y sean reportados a los directores apropiados.

Listas de chequeo sean usadas para identificar los criterios de revisión de los productos de trabajo de software de una manera consistente.

Se haga un seguimiento a las acciones identificadas en las revisiones hasta que sean resueltas.

La culminación exitosa de las revisiones, incluyendo la puesta al día de los elementos identificados en las revisiones, sea usada como un criterio de terminación para la tarea asociada.

3: Se registran los datos de la conducción y los resultados de la revisión.

#### Mediciones y análisis

1: Las mediciones son hechas y usadas para determinar el estado de las actividades de revisión. Ejemplos de mediciones incluyen:

Número de revisiones realizadas comparadas con el plan

Esfuerzo promedio utilizado en las revisiones comparadas con el plan

Número de productos de trabajo revisados comparados con el plan

Verificación de la implementación

1: El grupo de aseguramiento de la calidad de software revisa y/o audita las actividades y productos de trabajo para las revisiones y reporta los resultados.

Como mínimo, las revisiones o auditorías verifican que:

Las revisiones planeadas sean conducidas.

Los líderes sean entrenados adecuadamente para sus roles.

Los revisores sean entrenados apropiadamente o tengan experiencia en sus roles.

El proceso de preparación, conducción y realización de las revisiones sea seguido.

El reporte de revisión sea completo, preciso y a tiempo.

### 3.3.5 Prácticas Claves Nivel 4

#### Área de proceso clave: Gestión Cuantitativa del Proceso

El propósito de la gestión cuantitativa del proceso es controlar cuantitativamente la ejecución del proceso, el cual representa los resultados actuales alcanzados por el correspondiente proceso de software.

La gestión cuantitativa del proceso involucra el establecimiento de objetivos para la ejecución del proceso de software definido del proyecto, tomando medidas de la ejecución del proceso, analizando esas medidas y haciendo ajustes para mantener la ejecución dentro de los límites aceptables. Cuando la ejecución del proceso es establecida dentro de los límites aceptables, el proceso de definición de software del proyecto, las mediciones asociadas y los límites aceptables son establecidas como una base y usadas para controlar cuantitativamente la ejecución del proceso.

La organización recolecta datos sobre la ejecución de los proyectos de software y usa esta data para representar la capacidad del proceso en los estándares de la organización. Capacidad del proceso describe un rango de resultados esperados como consecuencia de seguir un proceso de software.

Los datos de la capacidad del proceso son, por turnos, usados por los proyectos de software para establecer y revisar sus objetivos en la ejecución de procesos y para analizar la ejecución de los procesos de software definidos del proyecto.

#### Metas

- 1: Se planean las actividades de la gestión cuantitativa del proceso.
- 2: La ejecución del proceso de software definido del proyecto es controlada cuantitativamente.
- 3: La capacidad del proceso de los procesos de software estándares para la organización es conocida en términos cuantitativos.

#### Compromisos a ejecutar

- 1: El proyecto sigue una política organizacional escrita para realizar las mediciones y el controlar cuantitativamente la ejecución del proceso de software definido del proyecto. Esta política especifica que:

Cada proyecto implementa un plan documentado para llevar el proceso de software definido a un control cuantitativo.

La data susceptible a un desempeño individual es protegida y el acceso a esa data es controlada apropiadamente

- 2: La organización sigue una política escrita para analizar la capacidad del proceso de todos los procesos de software estándares de la organización. Esta política especifica:

Las mediciones de la ejecución del proceso de un proyecto son analizadas para establecer y mantener una línea base de capacidad del proceso para los procesos de software estándares de la organización.

La línea base de capacidad del proceso para el proceso de software estándar de la organización es usada por los proyectos de software para establecer las metas de ejecución de los procesos.

#### Habilidades para ejecutar

- 1: Hay un grupo que es responsable de coordinar las actividades de la gestión cuantitativa del proceso.
- 2: Recursos y fondos adecuados son provistos a las actividades de gestión cuantitativa del proceso.

3: Existe soporte para recolección, grabación y análisis de datos para procesos seleccionados y mediciones de productos.

4: Los individuos que implementan o soportan la gestión cuantitativa del proceso reciben el entrenamiento requerido para realizar estas actividades.

5: Los miembros del grupo de ingeniería de software y otros grupos relacionados reciben orientación acerca de los objetivos y el valor del manejo de proceso cuantitativo.

#### Actividades ejecutadas

1: El plan de proyectos de software para la gestión cuantitativa del proceso es desarrollado de acuerdo a un procedimiento documentado.

Este procedimiento especifica que:

El plan de gestión cuantitativa del proceso esté basado en:

Los objetivos estratégicos de la organización para calidad de productos, productividad y ciclo de tiempo de desarrollo del producto.

Programa de mediciones de la organización.

Proceso de software estándar de la organización

Los objetivos del proyecto para la calidad de los productos de software, productividad y ciclo de tiempo de desarrollo del producto.

Ejecución de la medición de procesos de software definidos para otros proyectos.

La descripción del proceso de software definido del proyecto

El plan sea sometido a revisiones de par.

El plan sea revisado por el grupo responsable de las actividades del proceso de software.

El plan sea manejado y controlado.

2: Las actividades de la gestión cuantitativa del proceso son realizadas en concordancia con el plan de gestión cuantitativa del proceso. El plan cubre:

Los objetivos de las actividades de gestión cuantitativa del proceso

Las tareas de software y otras actividades de software que van a ser medidas y analizadas

Los instrumentos del proceso de software definidos para el proyecto.

Las actividades a ser realizadas y sus respectivos cronogramas.

Los grupos e individuos responsables.

Los recursos requeridos para realizar las actividades incluyendo staff y herramientas.

Los procedimientos a ser seguidos.

3: La estrategia de recolección de datos y análisis cuantitativo a ser realizado se determina basándose en el proceso de definición de software del proyecto, que incluye:

Las tareas, actividades y sus relaciones entre ellas.

Los productos de trabajo de software y sus relaciones entre ellos.

Los puntos de control del proceso y los puntos de recolección de datos.

4: Los datos de medición usados para controlar cuantitativamente el proceso de software definido para el proyecto son recolectados de acuerdo a un procedimiento documentado.

5: El proceso de software definido del proyecto sea analizado y llevado bajo un control cuantitativo de acuerdo a un procedimiento documentado.

6: Se preparan y distribuyen los reportes que documentan los resultados de las actividades del manejo cuantitativo del proceso

7: La línea base de capacidad del proceso para el proceso de software estándar es establecido y mantenido de acuerdo a un procedimiento documentado.

#### Mediciones y análisis

1: Mediciones son realizadas y usadas para determinar el estado de las actividades de la gestión cuantitativa del proceso.

#### Verificación de la implementación

1: Las actividades del manejo cuantitativo de proceso son revisadas con el gerente.

2: Las actividades del proyecto de software para la gestión cuantitativa del proceso son revisadas con el gerente del proyecto periódicamente y basándose en acontecimientos.

3: El grupo de garantía de la calidad de software revisa y/o audita las actividades y productos de trabajo para la gestión cuantitativa del proceso y reporta resultados. Como mínimo, las revisiones y/o auditorías verifican que: Los planes y procedimientos para gestión cuantitativa del proceso sean seguidos.

La recolección y análisis de la data de la gestión cuantitativa del proceso sean realizadas como son requeridas, incluyendo la verificación de que:

La data necesaria exista y sea recolectada

La data recolectada exista y soporte los objetivos del programa de medición de la organización

Los datos sean recolectados en un punto correcto del ciclo de vida

El costo de recolección de datos sea justificado por la utilidad de la misma

Los datos sean exactos, correctos y oportunos

La confidencialidad de los datos sea protegida apropiadamente

### **Área de proceso clave: Gestión de la Calidad de Software**

El propósito de la Gestión de Calidad de Software es desarrollar una comprensión cuantitativa de la calidad de los productos de software del proyecto y lograr las metas de calidad específicas. La Gestión de Calidad de Software involucra la definición de las metas de calidad para los productos del software, estableciendo planes para lograr estas metas, supervisando y ajustando los planes de software, productos de trabajo de software, actividades, y metas de calidad para satisfacer las necesidades y deseos del cliente y usuario terminal para productos de alta calidad. Para que estas metas puedan lograrse, la organización establece estrategias y planes, y el proyecto ajusta su proceso definido del software, lograr para la realización de dichas metas.

#### **Metas**

- 1: La actividades de la gestión de calidad del software del proyecto son planeadas.
- 2: Se miden y definen las metas para alcanzar la calidad de producto de software y sus prioridades.
- 3: Se cuantifica y gestiona el progreso real hacia el logro de las metas de calidad para los productos del software.

#### **Compromisos a ejecutar**

- 1: El proyecto sigue una política organizacional escrita para la gestión de calidad del software.

#### Habilidades para ejecutar

- 1: Recursos adecuados y financiamiento son proporcionados para la gestión de calidad de los productos de software.
- 2: Los encargados de implementación y soporte de gestión de calidad de software reciben el entrenamiento requerido para ejecutar sus actividades.
- 3: Los miembros del grupo de ingeniería de software y otros grupos relacionados al software reciben el entrenamiento requerido en gestión de calidad de software.

#### Actividades ejecutadas

- 1: El plan de calidad del software del proyecto se desarrolla y mantiene de acuerdo a un Procedimiento documentado.
- 2: El plan de calidad de software del proyecto es la base de las actividades de gestión de calidad de software del proyecto.
- 3: Se definen, supervisan y revisan las metas de calidad cuantitativas del proyecto para los productos del software a lo largo del ciclo de vida de software.
- 4: La calidad del producto del software del proyecto es medida, analizada, comparada a las metas de calidad cuantitativa de los productos en una base de manejo de eventos.
- 5: Se asignan las metas de calidad cuantitativas del proyecto de software para los productos apropiadamente a los subcontratantes que entregan los productos del software al proyecto.

#### Mediciones y análisis

- 1: Las mediciones son hechas y usadas para determinar el estado de las actividades de gestión de calidad de software.

#### Verificación de la implementación

- 1: Se revisan las actividades para la gestión de calidad de software con la Gerencia Senior en un periodo base.
- 2: Se revisan las actividades para la gestión de calidad de software con la gerencia del proyecto tanto en una base periódica como una base manejada por eventos.



3: Las revisiones del grupo de aseguramiento de calidad de software y/o auditorías de las actividades y productos de trabajo para la gestión de calidad de software y reportes de los resultados.

### **3.3.6 Prácticas Claves Nivel 5**

#### **Área de proceso clave: Prevención de Defectos**

El propósito de Prevención de Defectos es identificar la causa de defectos e impedir que se repitan. La prevención de defectos involucra el análisis de defectos que se encontraron en el pasado y tomando las acciones específicas para prevenir la ocurrencia de esos tipos de defectos en el futuro. Los defectos se pueden haber identificado en otros proyectos así como en fases más tempranas o tareas del proyecto actual. Las actividades de prevención de defectos también son un mecanismo para difundir las lecciones aprendidas a otros proyectos. Se analizan las tendencias para rastrear los tipos de defectos que se han encontrado e identificar defectos que son probables a repetirse. Basado en una comprensión del proceso del software definido del proyecto y cómo se lleva a cabo (como se describe en las áreas de proceso clave Gestión Integrada de Software e Ingeniería del Producto), las causas de los defectos y las implicaciones de los defectos para las actividades futuras es determinante. El proyecto y la organización toman las acciones específicas para prevenir repetición de los defectos. Algunas de las acciones organizacionales pueden manejarse como lo descrito en el área clave del proceso Gestión de Cambio de Proceso.

#### **Metas**

- 1: Las actividades de prevención de defectos son planeadas.
- 2: Se buscan e identifican las causas comunes de defectos.
- 3: Se priorizan y eliminan automáticamente las causas comunes de defectos.

#### **Compromisos a ejecutar**

- 1: La organización sigue una política escrita para las actividades de prevención de defectos.
- 2: El Proyecto sigue una política organizacional escrita para las actividades de prevención de defecto.

#### Habilidades para ejecutar

- 1: Existe un equipo a nivel de organización para coordinar las actividades de prevención de defecto.
- 2: Un equipo coordina las actividades de prevención de defecto para los proyectos del software. Normalmente se asignan miembros del equipo que coordina las actividades de prevención de defecto a este equipo sobre un tiempo medio base y que tienen otras actividades de ingeniería de software que como su responsabilidad primaria.
- 3: Se provee de recursos adecuados y financiamiento para las actividades de prevención de defecto a nivel de proyecto y organización.
- 4: Miembros del grupo de ingeniería de software y otros grupos relacionados al software reciben el entrenamiento requerido para realizar sus actividades de prevención de defecto.

#### Actividades ejecutadas

- 1: Plan de prevención de defectos para el desarrollo y mantenimiento del Proyecto de Software.
- 2: Al principio de una tarea del software, los miembros del equipo que realizan la tarea se reúnen para preparar las actividades de esa tarea y las actividades de prevención de defecto relacionadas.
- 3: Las reuniones de análisis causal se realizan según un procedimiento documentado.
- 4: Cada uno de los equipos asignados para coordinar las actividades de prevención de defectos se reúnen periódicamente para revisar y coordinar la implementación de las propuestas de acción de las reuniones de análisis causal.
- 5: Los datos de prevención de defecto son documentados y rastreados por los equipos que coordinan las actividades de prevención de defecto.
- 6: Las Revisiones al proceso de software estándar de la organización que son el resultado de las acciones de prevención de defectos son incorporadas según un procedimiento documentado.
- 7: Las Revisiones al proceso de software definido en el proyecto que son el resultado de las acciones de prevención de defectos son incorporadas según un procedimiento documentado.

8: Los integrantes del grupo de Ingeniería del software y de los grupos de software-relacionados reciben retroalimentación del estado y los resultados de las actividades de prevención de defecto de la organización y proyecto periódicamente.

#### Mediciones y análisis

1: Las mediciones se realizan y usan para determinar el estado de las actividades de prevención de defecto.

#### Verificación de la implementación

1: Se revisan las actividades de la organización para la prevención del defecto con la Gerencia Senior periódicamente. El principal propósito de las revisiones periódicas por la Gerencia Senior es proporcionar el conocimiento y visión de las actividades de proceso de software a un nivel apropiado de abstracción y de una manera oportuna. El tiempo entre las revisiones debe satisfacer las necesidades de la organización y puede ser largo, con tal de que los mecanismos adecuados para reportar excepciones estén disponibles.

2: Se revisa el software de las actividades del proyecto para la prevención de defectos con el gerente del proyecto, esto se hace periódicamente y manejado por eventos.

3: El grupo de aseguramiento de la calidad revisa y/o audita las actividades y trabaja productos para la prevención de defectos y reporte de resultados.

#### **Área de proceso clave: Gestión del Cambio de Tecnología**

El propósito de la Gestión del Cambio de Tecnología es identificar nuevas tecnologías (por ejemplo: herramientas, métodos y procesos) y rastrearlas dentro de la organización de una forma ordenada. Incluye identificación, selección y evaluación de nuevas tecnologías e incorporación de nuevas y efectivas tecnologías dentro de la organización. El objetivo es mejorar la calidad del software, incrementar la productividad, y reducir el ciclo de vida para el desarrollo de productos. La organización establece un grupo que trabaje con los proyectos de software para introducir y evaluar nuevas tecnologías y manejar los cambios de las tecnologías existentes. Manteniendo un cuidado en las innovaciones tecnológicas relacionadas con el software y sistemáticamente evaluando y experimentando con ellas, la

organización selecciona tecnologías apropiadas para mejorar la calidad de su software y la productividad de las actividades de su software.

#### Metas

- 1: La incorporación de los cambios de tecnologías son planeados.
- 2: Las nuevas tecnologías son evaluadas para determinar su efecto en la calidad y la productividad.
- 3: Nuevas tecnologías apropiadas son transferidas a prácticas normales a través de la organización.

#### Compromisos a ejecutar

- 1: La organización sigue una política escrita para mejorar su capacidad tecnológica.
- 2: La gerencia senior auspicia las actividades de la organización para la Gestión del Cambio de Tecnología.
- 3: La gerencia senior vigila las actividades de la Gestión del Cambio de Tecnología de la organización.

#### Habilidades para ejecutar

- 1: Existe un grupo responsable para las actividades de la Gestión del Cambio de Tecnología. Un grupo es el conjunto de departamentos, gerentes e individuos que tienen responsabilidad para un conjunto de tareas o actividades. Las consideraciones cuando se implementa un grupo incluye tareas o actividades asignadas, el tamaño del proyecto, la estructura organizacional, y la cultura organizacional. Algunos grupos, como el grupo de aseguramiento de la calidad de software, están enfocados en actividades de proyecto y otras actividades, así como el grupo del proceso de ingeniería de software está enfocado en las actividades de la organización.
- 2: Se provee recursos adecuados y financiamiento para establecer y proveer un grupo responsable para las actividades de la Gestión del Cambio de Tecnología de la organización.
- 3: Existe el soporte para recolectar y analizar los datos requeridos para evaluar los cambios de tecnología. El resultado del análisis de datos es presentado en formatos que transfieren apropiadamente el contenido de la información, por ejemplo gráficos. Este soporte incluye la habilidad de:

- Registrar procesos seleccionados y datos del producto automáticamente.
- Soportar análisis de datos.
- Mostrar datos seleccionados.

4: Los datos apropiados en el proceso de software y los productos del trabajo de software están disponibles para soportar análisis desarrollados para evaluar y seleccionar los cambios de tecnología.

5: Los miembros de un grupo responsable para las actividades de la Gestión del Cambio de Tecnología de la organización reciben entrenamiento requerido para desarrollar sus actividades.

#### Actividades ejecutadas

1: La organización desarrolla y mantiene un plan para la Gestión del Cambio de Tecnología.

2: El grupo responsable para las actividades de la Gestión del Cambio de Tecnología de la organización trabaja con los proyectos de software identificando áreas de cambio de tecnología.

3: Los gerentes de software y el personal técnico están informados de las nuevas tecnologías.

4: El grupo responsable para la Gestión del Cambio de Tecnología de la organización analiza sistemáticamente el proceso estándar de software de la organización para identificar las áreas que necesitan o podrían beneficiarse de la nueva tecnología.

5: Las tecnologías son seleccionadas y adquiridas para la organización y los proyectos de software, de acuerdo con el procedimiento documentado.

6: Se conducen esfuerzos pilotos para el mejoramiento de la tecnología, donde sea apropiado, antes de que la nueva tecnología sea introducida dentro de la práctica normal.

7: Nuevas tecnologías apropiadas son incorporadas dentro del proceso estándar de software de la organización de acuerdo con el proceso documentado.

#### Mediciones y análisis

1: Las mediciones son realizadas y usadas para determinar el estado de las actividades de la organización para la Gestión del Cambio de Tecnología.

Ejemplos de la medición incluyen:

- La actividad del cambio de tecnología global; y
- El efecto de la implementación del cambio de tecnología, comparada con las metas.

**Verificación de la implementación**

1: Las actividades de la organización para la Gestión del Cambio de Tecnología son revisadas con la gerencia senior sobre una base periódica.

El propósito primario de las revisiones periódicas es el de proveer conocimiento de y visión dentro de las actividades del proceso de software en un nivel apropiado de abstracción y de una forma temporal. Los tiempos entre las revisiones deberían encontrarse con las necesidades de la organización y serían prolongadas, en tanto los mecanismos adecuados para los reportes de excepción estén disponibles.

2: El grupo de aseguramiento de calidad del software revisa y/o audita las actividades y los productos del trabajo para la Gestión del Cambio de Tecnología y reporta los resultados.

### **Área de proceso clave: Gestión del Cambio de Proceso**

El propósito de la Gestión del Cambio de Proceso es mejorar continuamente los procesos de software usados en la organización con la intención de mejorar la calidad del software, incrementar la productividad y reducir el ciclo de vida para el desarrollo de productos. Involucra la definición de los objetivos del mejoramiento del proceso y con el auspicio de la gerencia senior, proactivamente y sistemáticamente, identifica, evalúa e implementa los mejoramientos al proceso estándar de software de la organización y a los procesos de software definidos de los proyectos sobre una base continua. Programas de entrenamiento e incentivo son establecidos para permitir y alentar a todos en la organización a participar en las actividades de mejoramiento de procesos.

**Metas**

1: El mejoramiento de los procesos continuos es planeado.

2: La participación en las actividades de mejoramiento de los procesos de software es a lo ancho de la organización.

3: El proceso estándar de software de la organización y los procesos estándares definidos de software son mejorados continuamente.

#### Compromisos a realizar

1: La organización sigue una política escrita para la implementación de los mejoramientos del proceso de software.

2: La gerencia senior auspicia las actividades de la organización para el mejoramiento del proceso de software.

#### Habilidades para ejecutar

1: Se provee recursos adecuados y financiamiento para las actividades de mejoramiento del proceso.

2: Los gerentes de software reciben el entrenamiento requerido en mejoramiento del proceso de software. Ejemplos de este entrenamiento incluye:

- Gestionar el cambio tecnológico y organizacional,
- Construcción de equipos, y
- Habilidades de trabajo en grupo aplicado al mejoramiento del proceso continuo.

3: Los gerentes y el personal técnico del grupo de ingeniería de software y otros grupos relacionados reciben entrenamiento requerido en mejoramiento del proceso de software.

Ejemplos de grupos relacionados al software incluyen:

- Garantía de la calidad del software,
- Gestión de la configuración del software, y
- Soporte a la documentación.

Ejemplos de entrenamiento incluyen:

- Los principios del mejoramiento del proceso y la calidad, y
- Los procedimientos para proponer mejoramientos de proceso.

4: La gerencia senior recibe entrenamiento requerido en mejoramiento del proceso de software. Ejemplos de este entrenamiento incluye:

- Estadísticas y evaluación comparativa,

- Configuración y seguimiento de objetivos para el mejoramiento del proceso, y
- Motivación y construcción de equipos en un ambiente de mejoramiento de proceso continuo.

#### Actividades ejecutadas

1: Se establece un programa de mejoramiento del proceso de software, el cuál autoriza a los miembros de la organización a mejorar los procesos de la organización.

2: El grupo responsable para las actividades del proceso de software de la organización (por ejemplo el grupo de ingeniería del proceso) coordina las actividades del mejoramiento del proceso de software.

3: La organización desarrolla y mantiene un plan para el mejoramiento del proceso de software de acuerdo a un procedimiento documentado.

4: Las actividades del mejoramiento del proceso de software están desarrolladas de acuerdo con el plan de mejoramiento del proceso de software.

5: Los propósitos del mejoramiento del proceso de software son manejados de acuerdo a un procedimiento documentado.

6: Los miembros de la organización participan activamente en equipos para desarrollar el mejoramiento del proceso de software en las áreas de procesos asignadas.

7: Donde sea apropiado, son instaladas las mejoras del proceso de software sobre un piloto base para determinar sus beneficios y efectividad antes de que sean introducidos dentro de las prácticas normales.

8: Cuando se decide transferir una mejora de proceso de software a una práctica normal, la mejora es implementada de acuerdo al procedimiento documentado.

9: Se mantiene registros de las actividades del mejoramiento del proceso de software.

10: Los gerentes de software y el personal técnico reciben la retroalimentación sobre el estado y los resultados de las actividades del mejoramiento del proceso de software sobre una base manejada por eventos.

#### Mediciones y análisis



1: Las mediciones son realizadas y usadas para determinar el estado de las actividades del mejoramiento del proceso de software.

#### Verificación de la implementación

1: Las actividades para el mejoramiento del proceso de software son revisadas con la gerencia senior sobre una base periódica. El propósito primario de las revisiones periódicas es el de proveer conocimiento de y visión dentro de las actividades del proceso de software en un nivel apropiado de abstracción y de una forma temporal. Los tiempos entre las revisiones deberían encontrar las necesidades de la organización y serían prolongadas, en tanto los mecanismos adecuados para los reportes de excepción estén disponibles. Estas revisiones:

Resumen la participación en las actividades para la Gestión del Cambio de Proceso.

Evalúan el rendimiento del proceso.

Identifican los cambios necesarios de metas.

Resuelven características.

Aprueban revisiones al plan para la Gestión del Cambio de Proceso, como apropiadas.

2: El grupo de aseguramiento de calidad del software revisa y/o audita las actividades y los productos del trabajo para la Gestión del Cambio de Proceso y reporta los resultados.

### **3.4 Toma de Decisiones**

El Modelo de Madurez de Capacidades de Proceso establece un conjunto de criterios que describen las características de las organizaciones de software que poseen procesos maduros. Estos criterios pueden ser usados por las organizaciones para mejorar su proceso de desarrollo y mantenimiento de software, o por organizaciones gubernamentales o comerciales para evaluar los riesgos de contratar a una compañía que ejecute un proyecto de software. Las evaluaciones del proceso de software se orientan en identificar las prioridades en el mejoramiento del proceso de software de la organización. Un equipo de evaluación usa el CMM para guiarlos en la identificación y priorización de estas mejoras. Estos resultados de la evaluación junto con las

prácticas claves del CMM son usadas (por un grupo de ingeniería de proceso de software, por ejemplo) para planear y mejorar la estrategia de la organización.

Las técnicas de aprendizaje organizacional son consideradas como soporte en varias de las áreas de proceso clave priorizadas. Estas técnicas ayudarán a obtener estos cambios organizacionales de una manera eficaz, influenciando en las aptitudes de las personas.

### **3.5 Estrategias Adoptadas**

Las estrategias a realizar para implementar el modelo de madurez de capacidades de proceso y aprendizaje, vistas desde una óptica general son:  
Evaluación para identificar las fuerzas y las debilidades en la organización de desarrollo de software, usando los niveles de madurez.

Evaluación para identificar las aptitudes y capacidades del personal técnico y administrativo.

Gerentes y personal técnico serán capacitados para entender las actividades necesarias a planear y llevar a cabo un programa de mejoramiento de proceso de software en su organización, usando el CMM.

Gerentes y personal técnico serán capacitados para entender y usar las técnicas y herramientas de aprendizaje organizacional, que servirán de soporte durante toda la evolución en los niveles de madurez.

Se organizarán grupos que trabajarán en proyecto pilotos que usarán el CMM como una guía para ayudarles a definir y mejorar el proceso de software en la organización.

Los pasos a seguir se resumen en la selección de un equipo que realizará una evaluación de la situación actual en cuanto al proceso de software, se analizará y conducirán entrevistas para entender el proceso, comparará si cumple con las prácticas claves del CMM.

El primer paso es seleccionar un equipo. Este equipo será entrenado en los conceptos fundamentales del CMM y en las cinco disciplinas del aprendizaje organizacional, así como, en los métodos de evaluación. Los miembros del equipo podrían ser profesionales con conocimientos en administración e ingeniería de software.

El segundo paso es tener información sobre el sitio a ser evaluado, completando un cuestionario de madurez y otros instrumentos de diagnóstico. Una vez que esta actividad es terminada, el equipo de evaluación realiza un análisis de respuesta, el cual concuerda las respuestas del cuestionario e identifica esas áreas donde una investigación adicional es garantizada. Las áreas a ser investigadas corresponden a las áreas de proceso clave del CMM.

El equipo se encuentra listo para visitar el sitio a ser evaluado. El equipo conduce entrevistas y revisa documentación para ganar un entendimiento sobre el proceso de software seguido por el sitio evaluado. Las áreas de proceso clave y las prácticas clave del CMM guían a los miembros del equipo en los cuestionamientos, entendimiento, revisión y sintetización de la información obtenida en las entrevistas y documentos. El equipo aplica un criterio profesional para decidir si la implementación en el sitio del área de proceso clave satisface los objetivos del área de proceso clave. Cuando hay una gran diferencia entre las prácticas claves del CMM y las prácticas del sitio, el equipo debe documentar su criterio para juzgar dicha área de proceso clave.

Al final del periodo de visita al sitio, el equipo produce una lista de resultados que identifica las fuerzas y debilidades del proceso de software de la organización. Estos resultados serán la base para las recomendaciones de mejoramiento del proceso.

Finalmente, el equipo prepara un perfil del área de proceso clave, que muestra las áreas donde la organización tiene, y no tiene, satisfecho los objetivos del área de proceso clave.

## **CAPITULO IV: EVALUACIÓN DE RESULTADOS**

### **4.1 Antes**

Las mayoría de las organizaciones de desarrollo de software son organizaciones que tienen sus procesos de desarrollo de software inmaduras. Los procesos de software son generalmente improvisados por los administradores de proyecto durante el curso del proyecto. Aun si un proceso de software es especificado, este no es rigurosamente ejecutado. Una organización de software inmadura es reactiva, y los administradores de proyectos están frecuentemente orientados a la resolución inmediata de crisis. Los cronogramas y presupuestos son frecuentemente excedidos por que ellos no están basados en estimaciones reales. Cuando una fecha esta comprometida, la funcionalidad y calidad del producto usualmente se sacrifican para alcanzar los tiempos del cronograma.

En una organización de software inmadura, no hay objetivos basados para certificar la calidad de los productos o para resolver los problemas del proceso o productos. De allí que la calidad de los productos es difícil de predecir. Actividades que intentan mejorar la calidad, tales como las revisiones y pruebas, son frecuentemente reducidas o eliminadas cuando el no hay mucho tiempo para cumplir con el cronograma del proyecto.

### **4.2 Después**

Las organizaciones de software madura, poseen una habilidad que abarca a toda la organización para administrar el proceso en el desarrollo y mantenimiento de software. El proceso de software es rigurosamente comunicado a los empleados existentes como a los nuevos y las actividades de trabajo son llevados a cabo según el proceso planeado. Los procesos definidos son adecuados para su uso y consistentes con la forma en que el

trabajo actualmente se realiza. Estos procesos definidos son modificados cuando es necesario, y mejoramientos son desarrollados a través de controladas pruebas-piloto y análisis de costo/beneficio. Los roles y responsabilidades junto con el proceso definido son claros en todo el proyecto y en toda la organización.

En una organización de software madura, los administradores, monitorean la calidad de los productos de software y la satisfacción del cliente. Hay un objetivo, cuantificar variables para certificar la calidad de los productos y analizar los problemas de los procesos y productos. Los cronogramas y presupuestos son basados en datos históricos y son realistas. Los resultados esperados de costos, cronogramas, funcionalidad y calidad del producto son usualmente obtenidos. En general un proceso disciplinado, es seguido consistentemente porque todos los participantes entienden el valor de lo que hacen, a la vez que una necesaria infraestructura existe para dar soporte a los procesos.

## **CAPITULO V: CONCLUSIONES Y RECOMENDACIONES**

### **5.1 Conclusiones**

El modelo de madurez de capacidades de proceso provee una estructura conceptual para el mejoramiento de la administración y desarrollo de productos de software en una forma disciplinada y consistente. Este modelo no garantiza que los productos de software serán satisfactoriamente contruidos o que todos los problemas en la ingeniería de software se resuelvan. El CMM identifica las prácticas de un proceso de software maduro y provee ejemplos de estas prácticas, pero no significa que sea determinativo ni exhaustivo. El CMM identifica las características de un proceso de software efectivo, pero la madurez de una organización está referida a todos los temas esenciales para un proyecto exitoso, incluyendo las personas, la tecnología, así como el proceso.

Por otra parte, las disciplinas de aprendizaje organizacional provee herramientas para que cualquier organización, no solo de desarrollo de software, pueda mejorar la capacidad de aprendizaje grupal y organizacional, permitiendo que los proyectos de cambio de una organización puedan ejecutarse de una manera efectiva.

El CMM puede usar las herramientas de las disciplinas de aprendizaje organizacional como prácticas complementarias, permitiendo mejorar las capacidades de las personas, en cuanto a su desarrollo personal, compartiendo una misma visión en la organización, mejorando la comunicación de los equipos, entendiendo los modelos mentales, e identificando las causas de los problemas usando el pensamiento sistémico. Para evolucionar a los siguientes niveles de madurez, las personas tienen que cambiar su forma de pensar, aprendiendo constantemente y adquiriendo nuevas capacidades que le van ha permitir realizar cambios permanentes y de esta manera ayudar al mejoramiento de los procesos de la organización.

## 5.2 Recomendaciones

Para que una organización de desarrollo de software pueda mejorar su productividad y tener productos de software de una alta calidad, necesita de un marco que le permita evaluar el estado actual y le proporcione una guía para mejorar su proceso de desarrollo de software. Una organización tiene que alcanzar por lo menos el Nivel 3 de madurez de proceso para poder producir software de calidad. Esto implica tener identificado por lo menos un Ciclo de Vida de Desarrollo que se adecue al tipo de producto que está construyendo, un proceso estándar para toda la organización, estándares para las diferentes actividades de desarrollo y un modelo para conducir los programas de mejoramiento. Por lo tanto, para llegar al Nivel 3 se recomienda usar lo siguiente

- El Proceso Unificado de Rational (RUP), de Ivar Jacobson, Grady Booch y James Rumbaugh.
- Ciclo de Vida del Software Iterativo e incremental.
- El Unified Modeling Lenguaje (UML), de Ivar Jacobson, Grady Booch y James Rumbaugh.
- El Modelo IDEAL, para conducir programas de mejoramiento.

El Proceso Unificado de Desarrollo de Software ha sido creado por los mismos autores del UML y permite a las organizaciones alcanzar el Nivel 3 del CMM, ya que usa un mismo lenguaje de modelamiento para las actividades de ingeniería, el UML, está manejado por Casos de Uso y provee un Ciclo de Vida de Software Iterativo e incremental. Los Casos de Uso proveen una forma para documentar los requerimientos del cliente que sirven de base para el desarrollo, pruebas y planes de iteración del proyecto, permitiendo una adecuada Gestión de Requerimientos. El Proceso Unificado es un proceso iterativo que provee oportunidades para mejorar el proceso en la siguiente iteración. El Proceso Unificado representa un proceso que puede ser adaptado a muchos tipos de proyecto.

El ciclo de vida del software es la base sobre el cual se harán adaptaciones según requieran los proyectos. Este ciclo de vida está basado en el ciclo de vida del *proceso unificado extendido*. Cada ciclo de vida se desarrolla en el tiempo. Este tiempo a su vez se divide en cuatro fases, como se muestra en

el Anexo 3. Dentro de cada fase, los administradores pueden descomponer adicionalmente el trabajo – en iteraciones con sus incrementos resultantes. Cada fase termina en un hito. Cada hito se determina por la disponibilidad de un conjunto de artefactos (modelos, o documentos) que han sido desarrollados hasta alcanzar un estado predefinido. Los hitos permiten a la dirección y a los desarrolladores controlar el progreso del trabajo. En cada fase se ejecutan las actividades de requisitos, análisis, diseño, implementación y pruebas, en proporciones diferentes, según se requiera. Cada fase se divide normalmente en iteraciones o mini-proyectos. Una iteración típica pasa por los flujos de trabajo básicos (requisitos, análisis, diseño, implementación, prueba).

El Modelo IDEAL es un modelo para el mejoramiento organizacional que sirva como una mapa para iniciar, planear y guiar acciones de mejoramiento. Este modelo para programas de mejoramiento ha crecido en muchos años de experiencia y lecciones aprendidas sobre programas de mejoramiento de proceso de software. Este modelo es un ciclo de vida para organizar las fases de un programa de mejoramiento. El modelo IDEAL define un enfoque sistemático de cinco fases para el mejoramiento continuo de proceso: Inicial, Diagnóstico, Establecimiento, Acción y Aprendizaje. Estas fases direccionan las tareas de gestión del proyecto. La fase inicial establece el soporte y responsabilidades. La fase de diagnóstico identifica los problemas a ser resueltos. La fase de establecimiento selecciona un plan y especifica las actividades. La fase de acción diseña, pilotea, implementa e institucionaliza las mejoras. La fase de aprendizaje identifica las mejoras en las actividades del modelo IDEAL.



## Bibliografía

- “Key Practices of the Capability Maturity Model – Versión 1.1”, Software Engineering Institute, Mark C. Paulk.
- “Capability Maturity Model for Software – Versión 1.1”, Software Engineering Institute, Mark C. Paulk.
- “La Quinta Disciplina en la Practica”, MIT, Peter Senge.
- “Automatización, Reingeniería en los procesos de negocios a través de la simulación”, Gregory A. Hansen.
- “El Proceso Unificado de Desarrollo de Software”, Ivar Jacobson, Grady Booch, James Rumbaugh.

## ANEXOS

**ANEXO 1: Tabla Resumen de la Evaluación en el Nivel 2**

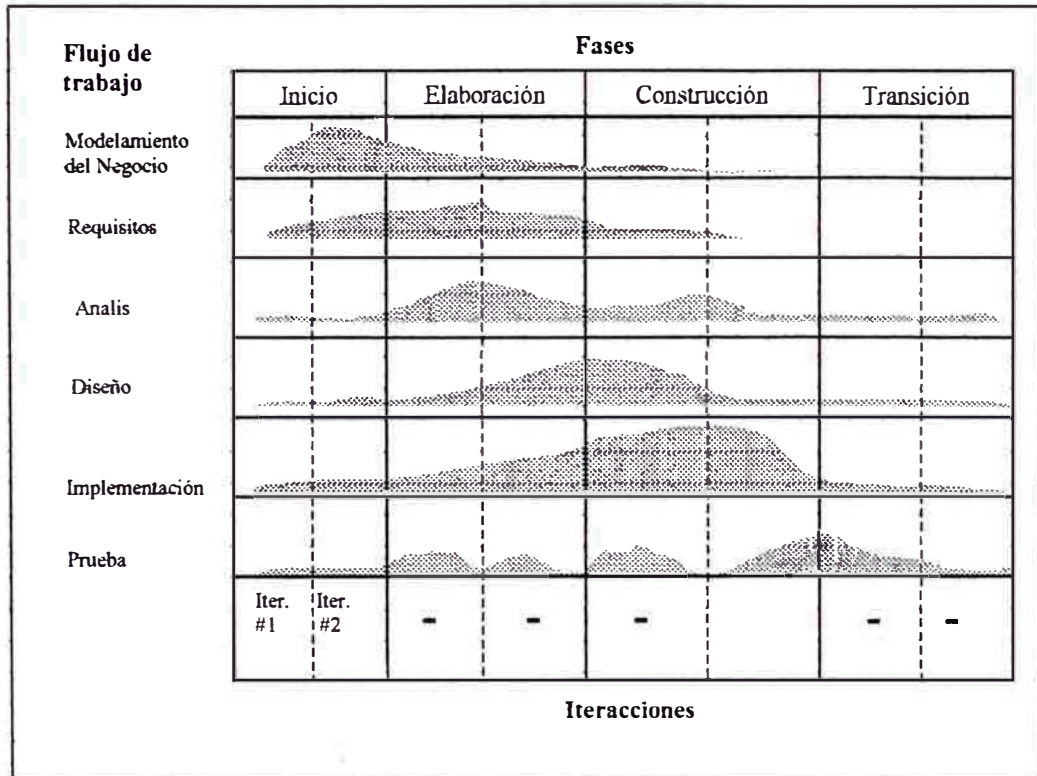
Nivel	Área de Proceso Clave	Meta	Grado de Satisfacción
2	Gestión de Requerimientos	<p>Requerimientos de sistema son controlados para establecer una línea base para la ingeniería de Software y la administración de proyectos.</p> <p>Mantener la consistencia entre los planes de software, productos, actividades y requerimientos.</p>	
	Planeamiento de Proyectos de Software	<p>Estimaciones de Software documentadas para ser usadas en el planeamiento y seguimiento del proyecto de software.</p> <p>Actividades y compromisos de un proyecto de software son planeadas y documentadas.</p> <p>Los grupos o individuos afectados aceptan sus compromisos relacionados con el proyecto de software.</p>	
	Seguimiento y Supervisión de Proyectos de Software	<p>Resultados y ejecución actuales son comparados contra los planes de software.</p> <p>Tomar las acciones correctivas cuando los resultados se desvíen significativamente de los planes de Software.</p> <p>Cambios a los compromisos de software son aceptados por los grupos e individuos afectados.</p>	
	Aseguramiento de la Calidad del Software	<p>Las actividades del Aseguramiento de calidad del Software son planeadas.</p> <p>La adherencia de los productos de software y las actividades para los aplicar estándares, procedimientos y requerimientos, es objetivamente verificada.</p> <p>Los grupos e individuos afectados son informados de las actividades y resultados del SQA.</p> <p>Los problemas de disconformidad que no se puedan resolver en el proyecto de software son dirigidos a la gerencia senior.</p>	
	Gestión de Configuración de Software	<p>Las actividades de gestión de la configuración de software son planeadas.</p> <p>Los productos del trabajo del software seleccionados son identificados, controlados y disponibles.</p> <p>Los cambios a productos de trabajo de software identificados son controlados.</p> <p>Los grupos e individuos afectados son informados del estado y contenido de las bases del software.</p>	

		<p>ciclos y arquetipos. Crear talleres de soluciones reales usando el pensamiento sistémico: Contar historia. Analizar la historia. Optando por un arquetipo. Estrategias y ramificaciones. Su propia versión de la historia</p>
--	--	--

## ANEXO 2: Prácticas Claves de las Capacidades de Aprendizaje

Nivel	Disciplina	Practica Clave
3	Dominio Personal	<p>Definición de la visión personal. Definir lo que la persona desea crear consigo mismo y con su entorno.            Crear Resultado (Como se ve, Como se siente)            Reflexionar sobre cada respuesta            Describir visión personal (Autoimagen, posesiones, hogar, salud, relaciones, trabajo, comunidad, propósito en la vida)            Expandir y clarificar visión (lo aceptaría?, que obtendría con eso?)            Determinación de Valores Laborales y Personales más significativos (Los 10 que más valoro, eliminación, elaboración de visión sobre los 3 valores más importantes).            Capacitar a entrenadores en visiones personales.            Revisión de la visión (revisión periódica de la realidad actual vs logros).            Determinación de políticas que impiden la lealtad a la verdad (lealtad al puesto, lealtad a la paz, política de amnistía).            Creación de un departamento de “Transformación y Descubrimiento”</p>
3	Modelo Mentales	<p>Análisis de creencias usadas en las inferencias. Taller de Escalera de Inferencias: Nuestras creencias afectan los datos que seleccionaremos la próxima vez.            Análisis de supuestos usados en las conversaciones.            Taller de Columna Izquierda (lo que pensaba, lo que dijimos).            Resolución de problemas usando equilibrio entre indagación y alegato. Uso de protocolos para el mejoramiento del alegato (que hacer, que decir).            Protocolos para mejorar la indagación. Protocolos para momentos de atasco.            Mejorar la visión sobre las personas claves de la organización. Usar la técnica Escribiendo a las lealtades.            Rotación de roles para tener perspectivas múltiples.            Creación de escenarios.            Definición del tema, propósito.            Comprendiendo las fuerzas impulsadoras.            Desarrollo de tramas.            Estrategia, ensayo y conversación.</p>
3	Visión Compartida	<p>Elaboración de visión, valores y propósito (Dirigentes tratan a los empleados como voluntarios).            Elaboración de ideas rectoras.</p>
3	Aprendizaje en Equipo	<p>Mejoramiento de la conversación.            Creación de laboratorios de aprendizaje.</p>
5	Pensamiento Sistémico	<p>Crear narraciones: acontecimientos, pautas de conducta, estructura sistémica, modelos mentales.            Crear talleres para exploración de narraciones.            Capacitación en Pensamiento sistémico: eslabones y</p>

### ANEXO 3: Ciclo de Vida del Proceso Unificado



En la fase de inicio el criterio fundamental es la viabilidad, que se lleva a cabo mediante la identificación de los riesgos críticos, la creación de un arquitectura candidata basada en un subconjunto clave de los requisitos, la realización de una estimación de los costos, esfuerzo, cronograma y calidad del producto. En esta fase se inicia el análisis del negocio que justifica que el proyecto vale la pena.

En la fase de elaboración, el criterio esencial es la capacidad de construir el sistema dentro de los límites económico, que se lleva a cabo mediante la preparación del plan del proyecto con suficiente detalle como para guiar la fase de construcción y la terminación del análisis del negocio.

En la fase de construcción, el criterio esencial es un sistema capaz de una operatividad inicial en el entorno del usuario, esto se lleva a cabo mediante una serie de iteraciones, que llevan a incrementos y entregas periódicas.

En la fase de transición, el criterio esencial es un sistema que alcanza una operatividad final, llevado a cabo mediante la modificación del producto para subsanar los problemas y la corrección de defectos.