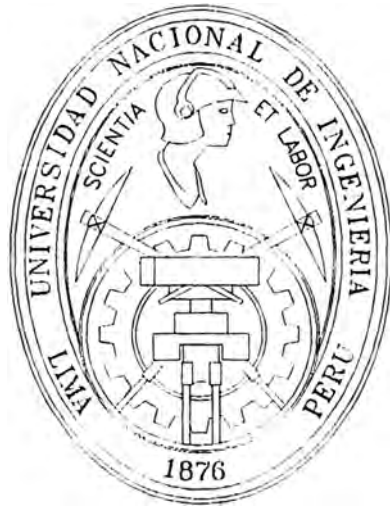


# Universidad Nacional de Ingeniería

FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS



## **EL MODELO DE CAPACIDAD DE MADURACION (NIVEL 2) APLICADO EN EL DESARROLLO DE SOFTWARE**

### **INFORME DE INGENIERIA**

**Para Optar el Título Profesional de:**

### **INGENIERO DE SISTEMAS**

**CARLOS ALBERTO FERNANDO WURST CALLE**

**Lima - Perú  
1998**

## **DEDICATORIA**

Esta tesis se la dedico a mi padre, que en paz descansa, quien con sus enseñanzas y ejemplo me heredo los mas altos valores humanos y profesionales.

## **AGRADECIMIENTOS**

A mi hermano Ingeniero Ricardo Wurst, quien con su ejemplo, gran ayuda y voluntad, me incentivo a lograr el desarrollo de esta tesis.

A mi esposa Lilly que siempre me apoyo moralmente durante el tiempo que estuve avocado a la confección de la tesis, y a mis hijos Erika, Christian y Sebastián que tuvieron paciencia y comprensión, al dedicarme a esta tarea durante sus fines de semana y vacaciones.

## **SUMARIO**

### **PRESENTACION DEL TEMA**

El tema presentado es un informe que trata de resumir la experiencia de la aplicación del modelo de maduración de capacidad (CMM), en una organización dedicada a la producción de software.

Este modelo fue inicialmente desarrollado por el departamento de defensa de Estados Unidos, para mejorar la capacidad de producción de artefactos militares, y posteriormente ha sido aplicado en la empresa privada, especialmente en empresas dedicadas al desarrollo y mantenimiento de productos de software.

### **PROCEDIMIENTOS ADOPTADOS**

La aplicación de este modelo específicamente es un proceso gradual, que aun no culmina. Inicialmente fue muy difícil adecuarse a los procesos y procedimientos adoptados, y daba la impresión de que lo único que producía su aplicación, era el incremento en los plazos de entrega. Sin embargo luego de un tiempo de su aplicación, se pudo demostrar sus bondades y resultados positivos.

## **DESCRIPTORES TEMATICOS**

- **CMM - Capability Maturity Model for Software**

Traducción en español : Modelo de Capacidad de Maduración

Es una estructura que describe los elementos claves de un efectivo proceso de software.

El CMM describe un camino de mejoramiento evolutivo, desde un proceso inmaduro hasta un proceso disciplinado y maduro.

- **Desarrollo de Software**

Cambios de funcionalidad o Creación de nuevas funcionalidades en Sistemas o Aplicaciones, a través de modificaciones, o mejoras en el Software.

- **Proyecto de Desarrollo de Software**

Se refiere a Proyectos cuyo objetivo involucra el Desarrollo de Software.

Los Proyectos de Desarrollo de Software, no necesariamente, se refieren a grandes cambios en los Sistemas, o el desarrollo de grandes aplicaciones. Los cambios pequeños, también están catalogados como proyectos, para los efectos del CMM.

- **Ciclo de Vida de Desarrollo de Software**

Se refiere al transcurso del Desarrollo de Software, dividiéndolo en etapas de desarrollo. En el ambiente objeto de este informe el Ciclo de Vida de Desarrollo de Software se dividió en : Iniciación, Definición, Construcción, Verificación e Implementación.

- **Proceso de Software**

El Proceso de Software se refiere a la infraestructura implementada, para mantener el Desarrollo de Software en un nivel específico.

Existen dos tipos de Proceso de Software

1. **Estándar de Proceso de Software para la Organización**

Específicamente en el Nivel 3 de CMM , el Estándar de Proceso de Software para la Organización, es un compendio de las buenas prácticas utilizadas, en Nivel 2 , a nivel de áreas o departamentos, la cual se aplicará a toda la Organización.

2. **Proceso de Software Definido para el Proyecto**

El Proceso de Software Definido para el Proyecto se obtiene del Estándar de Proceso de Software para la Organización, para ser aplicado en un proyecto específico

## **RESULTADOS Y CONCLUSIONES**

El tema presentado puede que contenga un grado de excepticismo, para los que estamos acostumbrados a desarrollar software en forma rápida y aparentemente eficiente. El desarrollo de software generalmente ha dependido del trabajo heroico de algunos buenos técnicos. La aplicación de este modelo demostró que los procesos son aun mas importantes que las habilidades individuales.

<b>INTRODUCCION.....</b>	<b>10</b>
<b>CAPITULO I - MARCO TEORICO.....</b>	<b>12</b>
ANTECEDENTES .....	12
ALCANCES .....	14
CMM - DESCRIPCIÓN GENERAL.....	15
1.1) NIVEL DE MADURACION 2 : PROCESO REPETITIVO .....	26
1.1.1) <i>Gestión de Requerimientos (Requirements Management (RM))</i> .....	27
1.1.2) <i>Planeamiento de Proyectos de Software (Software Project Plan( SPP))</i> .....	31
1.1.3) <i>Seguimiento de Proyectos de Software (Project Tracking and Oversight (PTO))</i> .....	36
1.1.4) <i>Evaluación de Calidad de Software (Software Quality Assurance (SQA))</i> .....	42
1.1.5) <i>Gestión de Configuración del Software (Software Configuration Management (SCM))</i> .....	46
1.1.6) <i>Gestión de Subcontratistas de Software (Software Subcontract Management SSM)</i> .....	50
1.2) NIVEL DE MADURACION 3 : EL PROCESO DEFINIDO .....	55
1.2.1) <i>Enfoque de Proceso de Organización (Organization Process Focus (OPF))</i> .....	56
1.2.2) <i>Definición del Proceso de la Organización (Organization Process Definition (OPD))</i> .....	61
1.2.3) <i>Programa de Entrenamiento (Training Program (TP))</i> .....	64
1.2.4) <i>Gestión de Software Integrado (Integrated Software Management (ISM))</i> .....	66
1.2.5) <i>Ingeniería de Productos de Software (Software Product Engineering (SPE))</i> .....	69
1.2.6) <i>Coordinación Intergrupos (Intergroup Coordination (IC))</i> .....	72
1.2.7) <i>Revisión entre Colegas (Peer Reviews (PR))</i> .....	76
1.3) NIVEL DE MADURACION 4 - EL PROCESO DE GESTIÓN.....	78
1.3.1) <i>Gestión de Proceso Cuantitativo (Quantitative Process Management (QPM))</i> .....	79
1.3.2) <i>Gestión de Calidad de Software (Software Quality Management (SQM))</i> .....	83
1.4) NIVEL DE MADURACION 5 - EL PROCESO DE OPTIMIZACION.....	86
1.4.1) <i>Prevención de Defectos (Defect Prevention (DP))</i> .....	87
1.4.2) <i>Gestión de Cambios de Tecnología (Technology Change Management (TCM))</i> .....	90
1.4.3) <i>Gestión de Cambios de Proceso (Process Change Management (PCM))</i> .....	93
<b>CAPITULO II - ENTORNO DE APLICACION DEL MODELO .....</b>	<b>96</b>
<b>CAPITULO III - PREPARACION A NIVEL EMPRESA CMM NIVEL 2 .....</b>	<b>99</b>
3.1 GENERALIDADES .....	99
3.2 GUÍAS PARA LA CREACIÓN DE PROCEDIMIENTOS Y MANUALES Y SISTEMAS UTILIZADOS .....	100
1) <i>Sistema Láser para Control de Requerimientos</i> .....	101
2) <i>Requerimientos de Negocio</i> .....	104
3) <i>Especificaciones Funcionales</i> .....	112
4) <i>Listas de Chequeos</i> .....	119
5) <i>Guía del Plan de Evaluación de Calidad de Software</i> .....	128
6) <i>Lista de Chequeo del Ciclo de Vida del Desarrollo de Software</i> .....	131
7) <i>Guía de Plan de Gestión de Configuración de Software</i> .....	148



8) <i>Mantenimiento y mejoras de Software (M&amp;MS) - Para proyectos de menos de 40 días de esfuerzo o menores a US\$ 50,000</i> .....	154
9) <i>Manual de desarrollo de Software (SDH Software development Handbook) - Para proyectos de mas de 40 días de esfuerzo o mayores a US\$ 50,000</i> .....	156
10) <i>Software de Gestión de Proyectos (Workbench)</i> .....	157
11) <i>Política de Contratación de Subcontratistas</i> .....	159
<b>CAPITULO IV - APLICACION PRÁCTICA DE CMM NIVEL 2</b> .....	<b>160</b>
4.1 GENERALIDADES .....	160
4.2 ORGANIZACIÓN.....	162
4.3 DESCRIPCIÓN DE PROCESOS - CICLO DE VIDA DE DESARROLLO .....	164
4.3.1 <i>Iniciación</i> : .....	171
4.3.2 <i>Definición</i> .....	174
4.3.3 <i>Construcción</i> .....	179
4.3.4 <i>Verificación</i> .....	182
4.3.5 <i>Implementación</i> .....	186
4.4 FUNCIONES DE SOPORTE DE PRODUCCIÓN .....	187
4.5 PROCEDIMIENTOS Y MANUALES .....	196
<b>CAPITULO V - ANALISIS COSTO/BENEFICIO</b> .....	<b>198</b>
<b>COSTOS</b> .....	<b>198</b>
<i>Contratación de Personal</i> .....	<i>198</i>
<i>Entrenamientos</i> : .....	<i>198</i>
<i>Certificación de CMM Nivel 2</i> : .....	<i>199</i>
<i>Desarrollo de Sistemas de Soporte</i> .....	<i>199</i>
<i>Adquisición de Software</i> .....	<i>199</i>
<i>Viajes</i> : .....	<i>199</i>
<i>Desarrollo de Sistemas de Soporte</i> .....	<i>200</i>
<b>BENEFICIOS</b> .....	<b>202</b>
<b>CONCLUSIONES</b> .....	<b>205</b>
<b>RECOMENDACIONES</b> .....	<b>207</b>
<b>GLOSARIO DE TERMINOS</b> .....	<b>209</b>
<b>BIBLIOGRAFIA</b> .....	<b>211</b>

## **INTRODUCCION**

Esta tesis tuvo como propósito presentar una aplicación práctica del modelo de capacidad de maduración, y con dicha presentación, poder evaluar su funcionalidad así como la factibilidad de su aplicación a través de los diversos niveles que el modelo ofrece, y por último determinar los beneficios obtenidos en su aplicación.

Se presentara inicialmente un marco teórico del modelo, necesario para poder relacionar esta base teórica con los distintos procedimientos que fueron aplicados para alcanzar el nivel 2 del modelo.

En el capítulo 2 se revisa el entorno de aplicación del modelo, una descripción de la organización, y del departamento donde el autor implemento el modelo.

En el capítulo 3, se presentan los cambios realizados en la organización que fueron requeridos para la aplicación del modelo, así como las guías y procedimientos desarrolladas por el equipo del proyecto, para ser implementadas en los diversos departamentos y areas de la organización.

En el capítulo 4, se describe como se transfirió las directivas de las guías y procedimientos, en procesos adecuados a las características tanto de tecnología, como de infraestructura organizacional, del departamento de soporte HP, en el cual el autor desarrollo sus actividades profesionales.

Para finalizar, se presenta un resumen del análisis Costo/Beneficio. Si bien los costos pueden ser cuantificados, los beneficios son en cierta forma cualitativos dadas las características evolutivas del modelo.

## **CAPITULO I - MARCO TEORICO**

### ***Antecedentes***

El Instituto de Ingeniería de Software, desarrollo una version inicial de un modelo de maduración y un cuestionario de maduración a requerimiento del Gobierno.

A través del desarrollo del modelo y el cuestionario, el SEI ha puesto atención en las sugerencias de practicantes quienes estan envueltos en desarrollo y mejoramiento de los procesos de software. El objetivo ha sido el de proveer un modelo que

- Esta basado en prácticas actuales
- Refleja lo mejor del estado de la práctica
- Refleja las necesidades de individuos ejecutando mejoramiento de procesos de software, evaluaciones de procesos de software, o evaluaciones de capacidades de software.
- Esta documentado
- Esta disponible públicamente

Conocimiento adicional en la maduración de proceso de software, ha sido alcanzado, desde la versión inicial del modelo de maduración. Este conocimiento ha sido alcanzado

- Estudiando Organizaciones no dedicadas al Software

- Ejecutando y observando evaluaciones de proceso de software y evaluaciones de capacidad de software
- Solicitando y analizando requerimientos de cambio al modelo
- Participando en reuniones y conferencias con representantes del Gobierno e Industria
- Solicitando opiniones de los representantes de la Industria y Gobierno

Utilizando este conocimiento adicional, el Modelo de Capacidad de Maduración y sus prácticas han sido revisadas, Creando CMM V1.1.

## ***Alcances***

En este capítulo se presentará una descripción teórica del modelo, incidiendo especialmente en el Nivel 2 (Repetitivo) , el cual es el nivel objeto del desarrollo del presente informe.

Sin embargo, aunque sin contemplar todas las claves prácticas de los demás niveles, la presentación de los KPAs, objetivos y actividades relacionadas de estos, nos permiten visualizar el contexto global del modelo, y nos dan una idea de como se presentarían en sus futuras aplicaciones prácticas.

## ***CMM - Descripción General***

El CMM (Capability Maturity Model) para “Software”, cuya traducción en español, sería “Modelo de capacidad de maduración” fue producido por el SEI (Software Engineering Institute), esta en uso a nivel mundial para mejorar la forma en que el Software es construido y mantenido.

El CMM es un modelo que describe como las prácticas de ingeniería de software en una organización evolucionan bajo ciertas condiciones

1. El trabajo ejecutado es organizado y visualizado como un proceso
2. La evolución del proceso es manejado sistemáticamente

Como otros modelos, el CMM es abstracto, pero es basado en experiencia.

Es mas, el CMM es un compendio de prácticas de ingeniería de software, en evolución para el mejoramiento continuo de los procesos utilizados para desarrollar y mantener software.

El CMM en si, tiene internamente una fuerza de motivación de cambio, en contraste con estándares estáticos tales como ISO 9001 para sistemas de Gestión de calidad (International Organization for Standardization, 1987), el cual es efectivo proveiendo un nivel mínimo de buena práctica, bajo la cual ninguna organización debería caer. En contraste el CMM es un estándar progresivo con una dimension dinámica que lleva a la organización a mejorar sus actuales prácticas de software en forma continuada.

La Evolución del proceso de ingeniería de software, enfatiza el aspecto de mejoramiento por etapas.

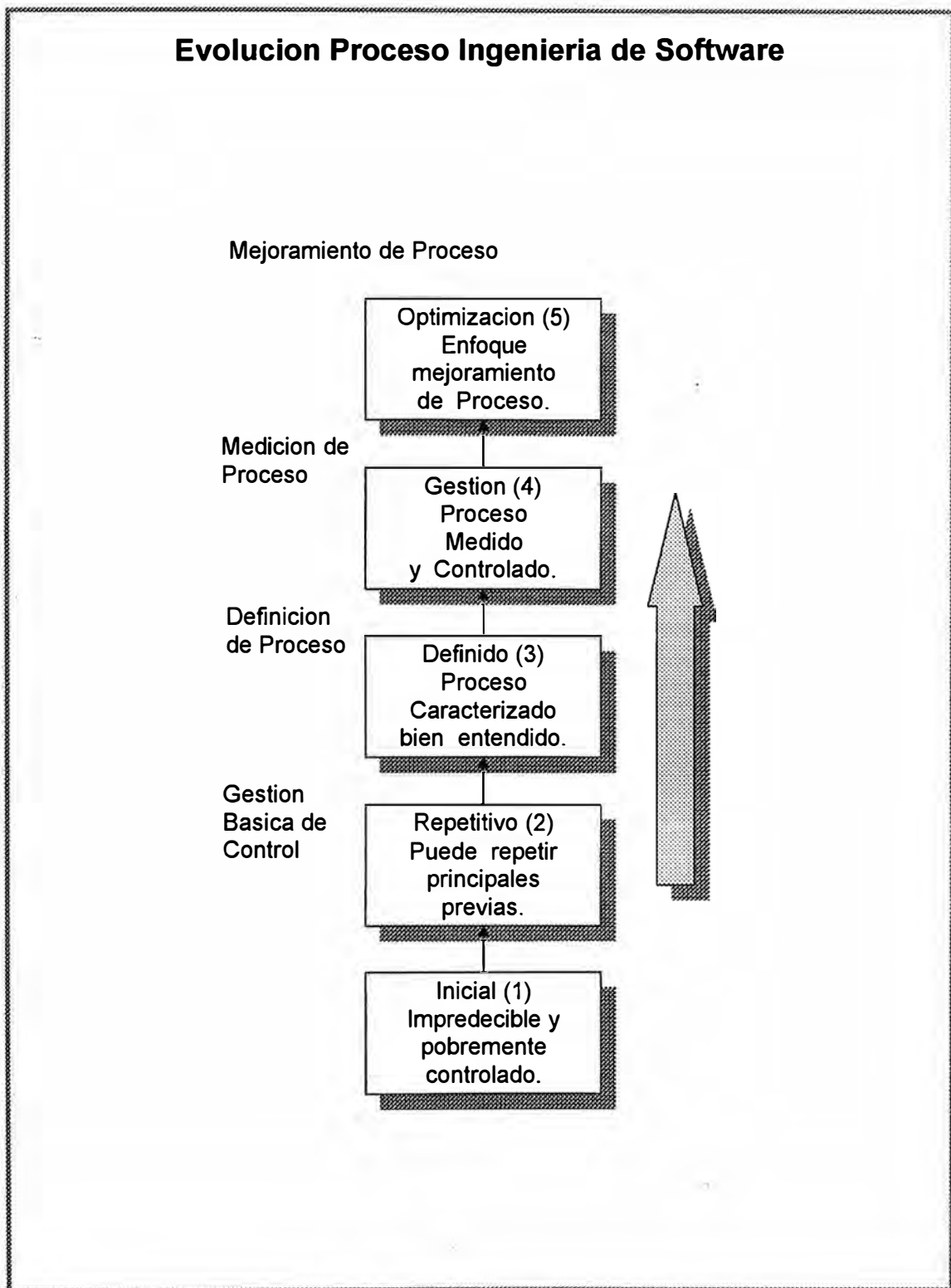
Las cinco etapas o niveles de maduración de procesos de software, mas marcadas son las siguientes

**TABLA I**

NIVEL	
<b>1-INICIAL</b>	IMPREDECIBLEYPOBREMENTECONTROLADO
<b>2-REPETITIVO</b>	PUEDEREPETIRSEPRINCIPALESTAREAS/CONTROLBASICO
<b>3-DEFINIDO</b>	PROCESO CARACTERIZADO Y BIEN ENTENDIDO / DEFINICION DE PROCESOS
<b>4-GESTIÓN</b>	PROCESOSMEDIDOSYCONTROLADOS/MEDICIONESDEPROCESOS
<b>5 OPTIMIZACION</b>	- INCIDENCIAENMEJORAMIENTODEPROCESOS



**FIGURA 1**



Cada nivel de maduración también puede ser definido operacionalmente en el sentido de que este esta compuesto por un conjunto de prácticas.

Las organizaciones que han alcanzado el nivel de maduración ejecutaran estas prácticas, o sus equivalentes, rutinaria y efectivamente. Estos conjuntos de prácticas, específicos a un nivel de maduración son llamados KPAs (Key Process Areas) Areas Claves de Proceso. A lo largo de los niveles se definen de 2 a 5 KPAs, y en total existen 18 KPAs. El nivel inicial no tiene KPAs.

**TABLA II**

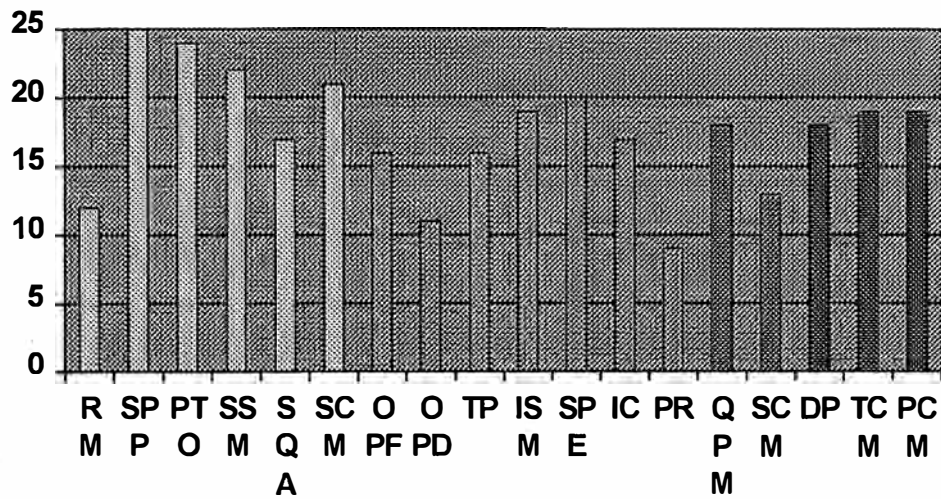
Nivel	Nombre de KPA	Abreviatura de KPA
<b>5-OPTIMIZADO</b>	Prevención de Defectos	Defect Prevention (DP)
	Gestión de cambios de tecnología	Technology Change Management (TCM)
	Gestión de cambios de Proceso	Process Change Management (PCM)
<b>4-GERENCIADO</b>	Gestión de proceso cuantitativo	Quantitative Process Management (QPM)
	Gestión de calidad del Software	Software Quality Management (SQM)
<b>3-DEFINIDO</b>	Enfoque de proceso de la organización	Organization Process Focus (OPF)
	Programa de Entrenamiento	Training Program (TP)
	Gestión de Software Integrado	Integrated Software Management (ISM)
	Ingeniería de Productos de Software	Software Product Engineering (SPE)
	Coordinación Intergrupos	Intergroup Coordination (IC)
	Revisión de Colegas	Peer Reviews (PR)
	<b>2-REPETITIVO</b>	Gestión de Requerimientos
<b>2-REPETITIVO</b>	Planeamiento de Proyectos de Software	Software Project Planning (SPP)
	Seguimiento de Proyectos de Software	Software Project Tracking and Oversight (PTO)
	Evaluación de Calidad de Software	Software Quality Assurance (SQA)
	Gestión de Configuración de Software	Software Configuration Management (SCM)
	Gestión de Subcontratistas de Software	Software Subcontract Management (SSM)

Cada KPA esta compuesto por claves prácticas, cuya aplicación, indica que el KPA es implementado en una organización.

El siguiente gráfico muestra el número de “claves prácticas” por cada KPA.

**FIGURA 2**

*Numero de claves practicas por KPA*



Hay otros dos importantes elementos estructurales, a considerar en la aplicación del CMM: Características comunes y objetivos o “goals”. Las prácticas claves están agrupadas en cinco categorías que el CMM llama “Common features”. Cada KPA tiene los cinco tipos de “common features” (características comunes) y al menos una clave práctica por cada ventaja común.

Las características comunes (colección de claves prácticas), permite la implementación de los objetivos de los KPAs. Los nombres de las ventajas comunes son :

- Compromiso de ejecución (CO)

El compromiso de ejecución describe las acciones que la organización debe tomar para asegurar que el proceso es establecido, y se mantendrá. El compromiso de ejecución, típicamente envuelve el establecimiento de políticas de la organización, y apoyo de la alta gerencia.

- Habilidad para ejecución (AB)

La habilidad de ejecución describe las precondiciones que debe existir en el proyecto u organización para implementar el proceso de software competentemente. La habilidad de ejecución típicamente envuelve, recursos, estructuras organizacionales y entrenamiento.

- Actividades ejecutadas (AC)

Las actividades ejecutadas describe los roles y procedimientos necesarios para implementar un KPA. Actividades ejecutadas típicamente envuelve establecer planes y procedimientos, ejecutar el trabajo, hacerle seguimiento a este, y tomar acciones correctivas cuando es necesario.

- Medición y análisis (ME)

Medición y Análisis describe la necesidad de medir el proceso y analizar estas mediciones. La Medición y Análisis típicamente incluye ejemplos de las mediciones que podrían haberse tomado para determinar el estado y efectividad de las actividades ejecutadas.

- Verificación de Implementación (VI)

Verificación e Implementación describe los pasos para asegurar que las actividades son ejecutadas de acuerdo al proceso que se ha establecido. La Verificación típicamente se refiere a las revisiones y auditorías por la gerencia y el equipo de “Evaluación de calidad de software”.

Con esta división en características comunes, el CMM separa los pasos de mejoramiento de procesos desde las acciones necesarias para mantener ellas y hacer de ellas la manera natural de hacer negocios. Es más, cuatro de los cinco tipos de “ventajas comunes” apuntan a institucionalizar las acciones de proceso, implementadas en el quinto tipo.

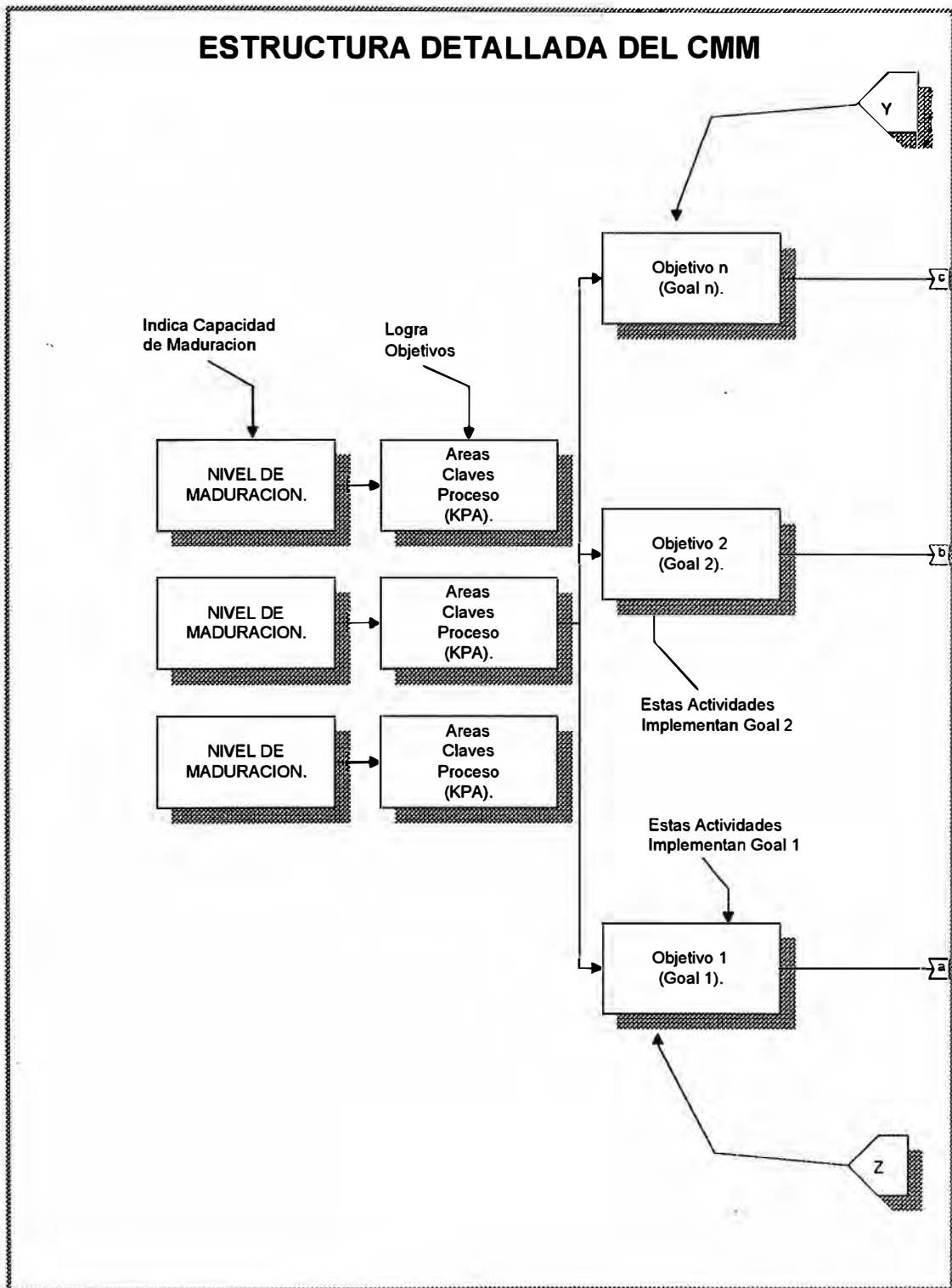
El CMM toma un comportamiento evolucionario para gerenciar cambios en el trabajo de procesos. Se ha demostrado (basado en la historia) que el mejoramiento en la manera que el trabajo es hecho no pasa por adoptar comportamientos revolucionarios, los cuales automáticamente se convierten en permanentes. Las experiencias de muchas organizaciones que han tratado mejoramiento de procesos muestra que la mayoría de los cambios, a pesar que los procesos nuevos trabajan mejor que los viejos, se requiere esfuerzos especiales para hacerlos durar.

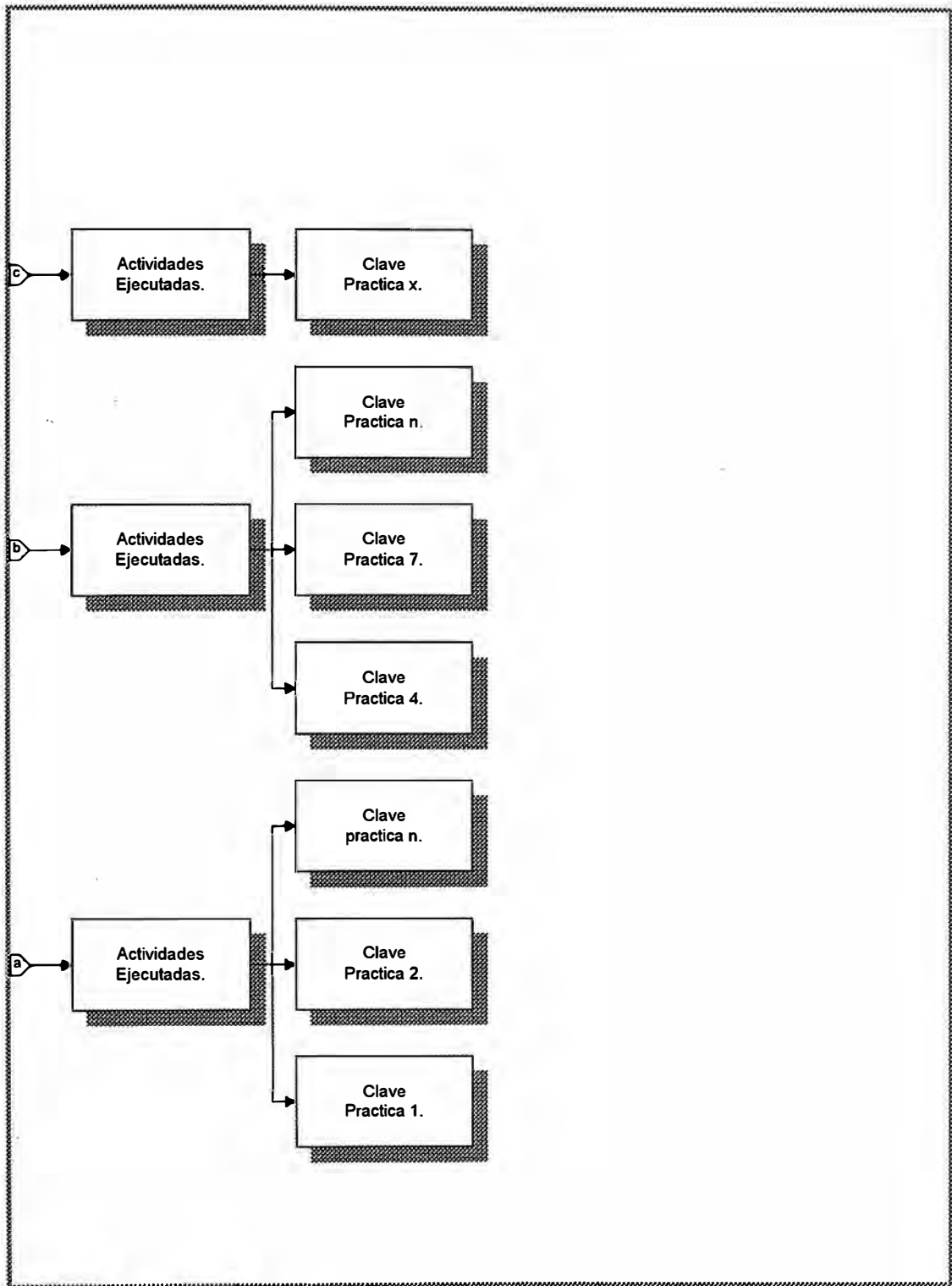
Las claves prácticas de “compromiso de ejecución” demuestran que la organización está dispuesta a hacer del KPA asociado una manera normal de negocio. La práctica de “compromiso de ejecución”, es usualmente una política de la organización firmada por la alta

gerencia. La práctica “Habilidad para ejecución” asegura los recursos (usualmente dinero) y tiempo, que estarían disponibles para llevar adelante las otras prácticas, estando satisfechas las otras condiciones, tales como entrenamiento. Las características comunes : Medición y análisis asegura que el estado de la práctica del KPA es conocido cuantitativamente. Y la característica común de Verificación de implementación corresponde a una revisión regular por la gerencia, y usualmente por el departamento de control de calidad (SQA), para asegurar que la implementación del KPA, tiene el efecto deseado y para ver si es necesaria la gestión de la gerencia para solucionar los problemas de implementación.

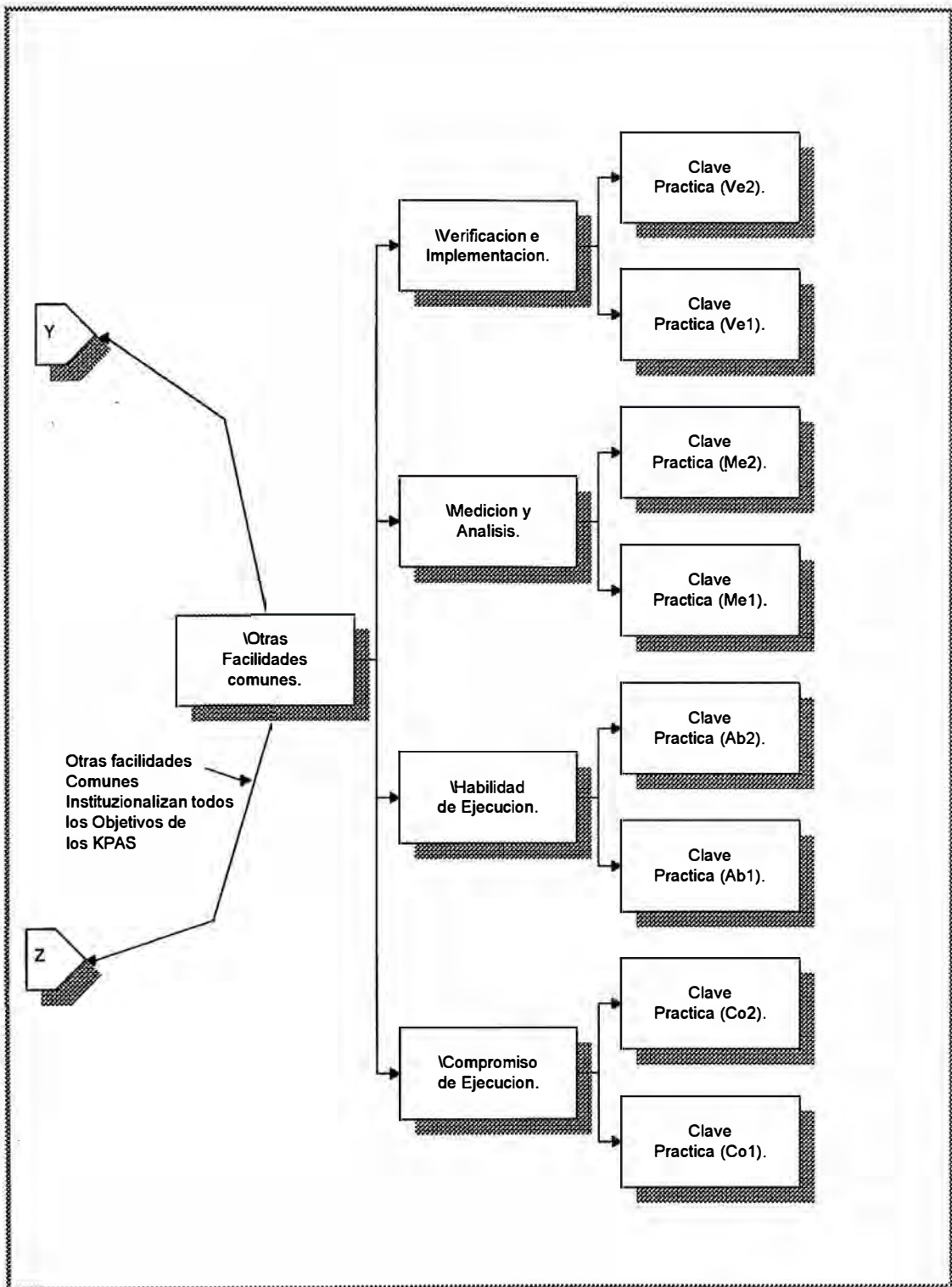
Las “Actividades ejecutadas”, sugieren acciones que el personal técnico y gerentes podrían tomar, para llevar a cabo el planeamiento, seguimiento, o entrenamiento en esa área del proceso. Sin las actividades ejecutadas, las otras “características comunes” no podrían ser institucionalizadas.

FIGURA 3









### **1.1) NIVEL DE MADURACION 2 : PROCESO REPETITIVO**

Cuando una organización alcanza el Nivel 2, su proceso de software es repetitivo y bajo un control de manejo básico. Los gerentes de proyectos son capaces de hacer estimados razonables, planes de proyectos, seguimientos y controlar el avance del proyecto a través de estos planes consistentemente.

Las mejores prácticas de software son acumuladas a nivel de proyecto, y hay una diferencia palpable en el estilo de trabajo de organizaciones de nivel 1.

En este nivel, las lecciones aprendidas no tienen que volverse a aprender y la organización ha establecido una base sólida de práctica de mejorar su proceso.

### **1.1.1) Gestión de Requerimientos (Requirements Management (RM) )**

El KPA (RM) viene primero en la lista de todos los KPAs. La razón por esta prioridad es que el control de los requerimientos puede llegar a ser el factor más crucial en estabilización del Nivel 1 del proceso de software, por lo cual este éxito a su vez llega a ser repetitivo.

El RM tiene el enfoque de Gestión más que de Ingeniería, debido a que los problemas que se reflejan en las evaluaciones se deben más a la falta de control de las personas que dieron los requerimientos al grupo de software, más que el grupo con la habilidad para analizar requerimientos.

Sin ese control, parecería que los requerimientos pueden provenir desde casi cualquier parte - Gerencia General, Clientes, Mercadeo, etc. - al grupo de software, quienes se espera implementen estos, no importa cuál es el impacto en el producto.

El KPA RM tiene únicamente dos objetivos (“goals”), y 12 prácticas, pero su brevedad puede ser engañosa: Un RM efectivo es fácil de decir, pero difícil de lograr.

#### ***OBJETIVO 1 :***

“ Los Requerimientos de sistema, asignados al software, son controlados, para establecer una “línea base” (línea base) para uso de ingeniería de software y Gestión”.

Debido a que esta frase se debe usar, en forma genérica para cualquier tipo de requerimientos (Complejos procesos de manufactura, etc.), la frase “requerimientos del sistema asignados al software”, para aquellos requerimientos, que no incluye componentes de manufactura, excepto computadoras, la frase “requerimientos del sistema asignados al software”, se debería decir “requerimientos de software”.

Los requerimientos, son revisados y aprobados por el grupo de ingeniería de software, y son colocados en una “línea base”, la cual entonces maneja, Ingeniería de software y Gestión.

#### ***OBJETIVO 2 :***

“Planeamientos de software, productos y actividades, se mantienen consistentes con los requerimientos de software”. Los requerimientos cambian durante la vida del proyecto, y el objetivo 2, es la manera como debemos mantener los cambios en los requerimientos, esto es , utilizando sucesivas “lineas bases”.

El termino “línea base”, implica que todas las versiones de un documento son identificadas, y la versión actual es controlada por una autoridad responsable.

#### ***PRACTICAS :***

Compromiso de ejecución 1 : (CO-1)

“El proyecto sigue una política de la organización, escrita para manejo de los requerimientos de software.”

Habilidad para ejecución 1 : (AB-1)

“Para cada proyecto, se establece responsabilidad para analizar los requerimientos de software, y asignar estos al Hardware, Software y otros componentes del sistema.”

Habilidad para ejecución 2 : (AB-2)

“Los requerimientos asignados son documentados”

Habilidad para ejecución 3 : (AB-3)

“Recursos y fondos adecuados deben ser proveídos para la Gestión de los requerimientos asignados.”

Habilidad para ejecución 4 : (AB-4)

“Los miembros de grupo de ingeniería de software y otros grupos relacionados con el software son entrenados para ejecutar sus actividades de Gestión de recursos”.

Actividad para ejecución 1 : (AC-1)

“El grupo de ingeniería de software revisa los requerimientos asignados antes de que ellos son incorporados en el proyecto de software”.

Actividad para ejecución 2 : (AC-2)

“El grupo de ingeniería de software utiliza los requerimientos asignados como la base para el planeamiento del software, productos, y actividades”.

Actividad para ejecución 3 : (AC-3)

“Los cambios para los requerimientos asignados, son revisados e incorporados en el proyecto de software”.

**Medición y Análisis 1 : (MA-1)**

“Las mediciones son hechas y utilizadas para determinar el estado de las actividades para Gestión de los requerimientos asignados”.

**Verificación de implementación 1 : (VI-1)**

“Las actividades para Gestión de requerimientos asignados son revisadas por la alta gerencia periódicamente”.

**Verificación de Implementación 2 : (VI-2)**

“Las actividades para Gestión de los requerimientos asignados son revisadas con el Gerente de proyecto, periódica y eventualmente”.

**Verificación de Implementación 3 : (VI-3)**

“El grupo de Evaluación de calidad del software (SQA), revisa y/o audita las actividades y productos para Gestión de los requerimientos asignados y reporta los resultados”.

### **1.1.2) Planeamiento de Proyectos de Software (Software Project Plan( SPP))**

Planeamiento de proyectos de software, el siguiente KPA en nivel 2, tiene 3 objetivos y 25 claves prácticas, entre estas tiene 15 actividades, más que cualquier otro KPA; este enfoca varios problemas en los proyectos de software.

#### ***OBJETIVO 1 :***

“Las Estimaciones de software son documentadas para su uso en planeamiento y seguimiento del proyecto de software”.

Se refiere a tamaño, cronograma, esfuerzo, etc.

Este objetivo, apunta directamente a otro KPA. El de “Seguimiento del proyecto (“Project Tracking and Oversight” PTO), donde estos estimados son usados también.

Estos estimados son documentados en un archivo de data, y utilizado en el Plan de desarrollo de software.

#### ***OBJETIVO 2 :***

“Las actividades de los proyectos de software y compromisos son planificadas y documentadas”.

Es común incluir actividades en un plan de proyecto, pero no es común incluir compromisos.

Y los compromisos envueltos deberían estar basados en los estimados de software (Objetivo 1), de otra manera los compromisos no serán realistas.

#### ***OBJETIVO 3***

“Los individuos y grupos afectados están de acuerdo de sus compromisos relacionados con el proyecto de software”.

Sin el compromiso de los acuerdos, el plan los estimados, y el cronograma no son operativos.

***RELACION OBJETIVO 1 -ACTIVIDADES:***

Existen cinco actividades asociadas con el OBJETIVO - 1.

Actividad 9 :

Relacionada con los estimados de tamaño de los productos de trabajo de software, los estimados de tamaño son la base para otros estimados.

En la terminología CMM, los “productos de trabajo de software”, no se refieren únicamente al código del software, si no que es la salida de cualquier etapa en el proyecto, inclusive productos que nunca serán entregados. Los “productos de software” son aquellos ítems que serán entregados.

La actividad 9, dice que los estimados de tamaño de los productos (intermedios y finales), así como los estimados de cambios y sus tamaños, son derivados de acuerdo a un procedimiento documentado.

Los estimados de tamaño, ya sea en líneas de código fuente, puntos de función, número de paginas, o alguna otra medición, son hechas y registradas. El procedimiento debería cubrir los principales productos de trabajo (incluyendo los no entregados), asunciones por cada estimado, y debería requerir que los estimados de tamaño son revisados y se esta de acuerdo con estos.

Las otras actividades 10, 11 y 12 bajo este Objetivo, son similares a la prácticade estimación de tamaño. Todas estas se refieren a un procedimiento documentado.



La actividad 10, se refiere a estimados de esfuerzo y costo. Un elemento de este procedimiento es que el estimado de esfuerzo y costo, están relacionados al estimado de tamaño.

La actividad 11 se refiere a seguir un procedimiento documentado para hacer estimaciones de recursos críticos de computadores necesarios para el proyecto. Recursos críticos de computadoras pueden ser memoria, CPU, comunicaciones o capacidad de entrada/salida.

Esta actividad es mas relevante, en proyectos en los cuales alguna pieza de hardware, es crítica para el proyecto, ej. “Chips” de computadoras, en un sistema de aviación.

La actividad 12, la cuarta actividad bajo el objetivo - 1, tiene que ver con estimaciones del cronograma del proyecto a través de un procedimiento documentado. Algunos de los ítems que incluye el procedimiento son : comparación con proyectos similares, usando apropiada duración de tareas, dependencias críticas (en proveedores, en otras tareas, quizás en entrega de componentes) y, por supuesto la relación del cronograma con el estimado crítico de tamaño.

Los cronogramas forzados o dictatoriales, son un problema y una forma de vida, no solo en el mundo del software sino también en la sociedad industrial. No podemos evitar las fechas de entrega dictatoriales, así también, debemos considerar el proceso de compromiso. La principal parte de la capacidad del nivel 2 apunta al manejo de condiciones que van contra los cronogramas impuestos.

La actividad 15, se refiere a que todas las actividades de estimación deben estar basadas en los datos históricos de la organización.

## ***RELACION OBJETIVO 2 - ACTIVIDADES :***

El Objetivo 2 requiere de 6 de las 15 actividades definidas en este KPA.

### **Actividad 2**

El Planeamiento de proyectos de software es desarrollado en las etapas iniciales y en paralelo con el planeamiento global del proyecto.

Notar que el CMM, asume grandes desarrollos de la organización, donde el proyecto de software es solo una parte de un proyecto mas grande.

### **Actividad 5**

El proyecto deberá identificar o usar un apropiado ciclo de vida del software.

### **Actividad 14 :**

El ambiente para el soporte del desarrollo del proyecto (Facilidades de ingeniería de software y herramientas) son planificadas.

### **Actividad 8**

Se relaciona con “Gestión de configuración de software”. El proyecto identificara el conjunto de productos de trabajo - intermedios y finales - sobre los cuales el proyecto necesita mantener control.

### **Actividad 13**

Se relaciona con evaluación de riesgos. No solo riesgos técnicos (como factibilidad de ingeniería), sino también riesgos de programación (debido a costos, cronogramas o conflictos de recursos) son identificados, evaluados y documentados.

#### Actividad 7

El plan del proyecto de software es documentado. Esta actividad sirve para asegurarse que la colección de planes acumuladas por estas actividades son realmente recolectadas.

#### ***RELACION OBJETIVO 3 - ACTIVIDADES :***

Las actividades del Objetivo - 3 constituyen, el valor agregado del CMM por una buena práctica en el planeamiento del proyecto.

#### Actividad 1

El grupo de ingeniería de software participa en la propuesta del proyecto.

La voz del desarrollador requiere ser oída en la fase temprana del proyecto.

#### Actividad 3

Asegura que el planeamiento continua a través de la vida del proyecto y que el grupo de ingeniería de software revisa los niveles del plan de proyecto. El grupo de software debe estar envuelto en la replanificación, en caso de cambios.

#### Actividad 4

Los compromisos entre el grupo de software y grupos fuera de la organización son revisados por la alta gerencia de acuerdo a un procedimiento documentado.

La intención de esta práctica es que los compromisos con los clientes están basados en planes razonables. En organizaciones con procesos de software casi caóticos, es frecuente que la alta

gerencia quienes deben hacer compromisos externos los hagan con bases no razonables en estimados y prácticas de planificación, y esto puede ocasionar cronogramas no cumplidos y productos fallidos.

### **1.1.3) Seguimiento de Proyectos de Software (Project Tracking and Oversight (PTO))**

El uso de las prácticas agrupadas bajo esta área de procesos, provee visibilidad en las actividades y estado del proyecto en si. Estas prácticas permiten al proyecto hacer el seguimiento de sus actividades y tomar acciones de control.

#### ***OBJETIVO 1 :***

Los resultados reales y desenvolvimiento son comparados contra los planes de software

#### ***OBJETIVO 2 :***

Acciones correctivas son tomadas y gerenciadas hasta su solución, cuando los resultados reales y desenvolvimiento se desvían significativamente de los planes de software.

#### ***OBJETIVO 3 :***

Los cambios en los compromisos de software son acordados por los grupos e individuos afectados.

#### ***RELACION DE OBJETIVOS CON PRÁCTICAS DE HABILIDAD PARA EJECUCION :***

Este KPA considera 5 prácticas de “Habilidad de ejecución” (AB).

##### **Habilidad 1**

El plan de desarrollo de software es documentado y aprobado.

Parecería redundante, con la Actividad 7 del KPA SPP, el cual dice que el plan es documentado. Pero el Plan de desarrollo de software es un prerrequisito para ejecutar el Seguimiento del Proyecto, por eso aquí se refiere como una Habilidad de ejecución.

#### Habilidad 2 :

Se refiere a la responsabilidad relacionada con los productos de trabajo, la cual es asignada por el Gerente de Proyecto.

#### Habilidad 3

Se refiere a que debe haber recursos adecuados para las actividades de seguimiento.

#### Habilidad 4:

Los gerentes de software son entrenados en Gestión de los aspectos técnicos y de personal del proyecto.

#### Habilidad 5

Gerentes de primera línea reciben orientación en los aspectos técnicos del proyecto de software.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

Objetivo 1 esta soportado por 10 actividades de las 13 que tiene este KPA.

#### Actividad 1

Un plan de desarrollo de software documentado es usado para el seguimiento de las actividades de software, y comunicación de su estado.

En esta actividad vemos que el plan no es solo la base para el seguimiento del progreso, sea la necesidad de actividades, estimados, productos de trabajo, o puntos de finalizacion de etapas (milestones) , sino también es usado para la comunicación del estado. Esto implica que el plan es un documento vivo, actualizado con los cambios y distribuido a los grupos afectados.

#### Actividad 12 :

Se refiere a que el grupo de ingeniería de software, conduce su propia revisión interna para seguir el progreso técnico, los planes, y su desenvolvimiento, y los verifica contra el plan de desarrollo de software.

Existen actividades separadas en el CMM, por cada uno de los parámetros estimados en el Plan del Proyecto de Software (KPA SPP).

Actividad 5 : Tamaño

Actividad 6 : Esfuerzo y costo

Actividad 7 : Recursos críticos de computadora

Actividad 8 : Cronograma del proyecto

Actividad 9 : Actividades técnicas de ingeniería de software

Actividad 10: Riesgos de software, técnicos y programáticos

Actividad 11 :

Se refiere a mediciones de lo real, esto es, medidas de desenvolvimiento y datos utilizados para planificación, a ser registrados.

Actividad 13

Se refiere a revisión formal externa y documentada, siendo ésta registrada y almacenada.

### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

El objetivo 2 es soportado por muchas de las mismas actividades del Objetivo 1. Pero el Objetivo 2 remarca la acción correctiva a ser tomada cuando ocurren desviaciones en tamaño (Actividad 5), esfuerzo y costo (Actividad 6), recursos críticos de computadoras (actividad 7), Cronograma del software (Actividad 9), y expectativas de riesgo (Actividad 10).

Para el punto de vista de evaluación, el mismo conjunto de actividades ejecutadas son consideradas pero con respecto a acciones correctivas tomadas. Cambios en factores de tamaño, personal, costos, recursos críticos de computadoras, y compromisos son negociados. Para las actividades de ingeniería de software, los problemas en los productos de trabajo, generan reportes de resultados, a los cuales se les hace seguimiento hasta que son resueltos.

Finalmente, bajo el objetivo 2 se incluye la revisión del plan de desarrollo de software, a través de un procedimiento documentado. El procedimiento es como una lista de chequeo que ayuda a hacer de la actividad una práctica uniforme. Los cambios en los compromisos son reflejados en el plan actualizado, el cual es gerenciado y controlado.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

El Objetivo 3 del KPA PTO, es simétrico al tercer objetivo del KPA SPP, en lo que se refiere a que este concierne al tipo de protocolo que anima los otros dos objetivos.

Sin el acuerdo de compromisos por los grupos afectados, el planeamiento y documentación de los otros dos objetivos no será efectivo.

#### **Actividad 3**

Los compromisos relacionados a los individuos y grupos fuera de la organización son revisados con la alta gerencia, de acuerdo a un procedimiento documentado.



Actividad 4 :

Los cambios aprobados en el Proyecto de software, son comunicados a los grupos afectados.

Como se indica previamente, en la actividad 1, el Plan de software, es usado para comunicar el estado. La actividad 4 requiere mas. Que cada uno afectado por los compromisos cambiados debería saber acerca de estos cambios aprobados.

### ***PROCESO DE COMPROMISO***

En una organización, los compromisos son hechos y acordados por individuos, pero también debe haber un soporte de la organización y una cultura de hacer compromisos. El soporte y cultura es lo que significa el PROCESO DE COMPROMISO.

El proceso descansa en los siguientes dos principios

- 1) Una alta actitud de compromiso
- 2) Prácticas de la organización de alcanzar pequeños y grandes compromisos

Hay un proceso de compromisos de software por proyectos y un Sistema de gerencia que provee para ambos, proyectos y compromisos a largo plazo de la organización.

El sistema de Gestión debe permitir revisiones de planes de proyecto, prògresos de proyectos, y también revisión de planes de operación, los cuales generalmente marchan fuera de fase con los puntos de finalización de etapas del proyecto (milestones). Problemas por recursos y fechas de

entrega aparecerán en las diferentes fases, por lo que debe haber un proceso para levantar estos temas en conflicto y resolverlos.

Esto implica un cambio radical del comportamiento, ya que el levantamiento de estos problemas, debe ser una norma. Generalmente las organizaciones tienden a esconder los problemas. Por ejemplo, nadie quiere admitir en una revisión del proyecto que el o su grupo estarán tarde, o al menos nadie quiere ser el primero en admitirlo, debido a que tan pronto otro grupo confiese que estará tarde, entonces uno podrá culpar la tardanza de su propio grupo al primer grupo. Esto es llamado el fenómeno del “Ultimo mentiroso”.

El Sistema de Gestión, tiene que ofrecer un ambiente donde se espera que la gente levante temas, y que donde hay un proceso para resolver estos (libre de culpas).

#### **1.1.4) Evaluación de Calidad de Software (Software Quality Assurance (SQA))**

SQA provee un chequeo independiente de que el proyecto esta siguiendo su proceso y da visibilidad gerencial al proceso. SQA avisa a la alta gerencia cuando las acciones de control de seguimiento del proyecto son inadecuadas.

##### ***OBJETIVO 1 :***

Las actividades de SQA son planeadas.

##### ***OBJETIVO 2 :***

La adherencia de los productos de trabajo de software y actividades a los estándares, procedimientos, y requerimientos es verificada objetivamente.

##### ***OBJETIVO 3 :***

Los grupos e individuos afectados, son informados de los resultados del SQA.

##### ***OBJETIVO 4:***

Los temas que no pueden ser resueltos con el proyecto de software, son escalados a la alta gerencia.

##### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

Actividad 1:

El plan de SQA para el proyecto es preparado a través de un procedimiento documentado.

El planeamiento del SQA, es hecho en paralelo con el planeamiento del software. Como otros tipos de planes en CMM, el plan de SQA es gerenciado y controlado.

#### Actividad 2

El Plan de Evaluación de Calidad (SQAP), gobierna las actividades del SQA.

Las actividades a ser consideradas en el plan de SQA son las siguientes :

- Fondos y recursos para el grupo SQA
- Que actividades y productos de trabajo serán evaluadas
- Estándares y procedimientos a ser revisadas
- Procedimiento para escalar temas no cumplidos.

#### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

#### Actividad 3

El grupo SQA participa en la preparación y revisión del Plan de proyecto de software y los estándares y procedimientos a ser usados.

Esta actividad le da una sabor de “naturaleza de colaboración”. No es simplemente una inspección sino una revisión de consultoria.

#### Actividad 4

Se refiere a revisión de actividades.

#### Actividad 5 :

Auditoria de productos de trabajo, por el grupo de SQA.

Los productos a ser entregados, son evaluados antes de la entrega al cliente. Los productos de trabajo son evaluados de acuerdo a requerimientos contractuales así como estándares y procedimientos.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

En objetivos 3 y 4, vemos mas de naturaleza colaboradora del estilo del CMM para el SQA.

#### **Actividad 6**

El grupo de SQA reporta resultados periódicamente.

Generalmente el SQA es percibido por los desarrolladores, como que adiciona poco valor. La actividad 6 trata de remediar esto, requiriendo comunicación regular acerca de los resultados del SQA a los ingenieros de software.

La implicancia es que la ingeniería de software es un cliente de SQA, y una práctica como la actividad 6 da al SQA la opción de explicar sus beneficios y conseguir opiniones de sus clientes sobre su desenvolvimiento.

#### **Actividad 8**

Menciona que el grupo de SQA revisa periódicamente sus actividades y descubrimientos con los clientes del grupo SQA apropiadamente.

### ***RELACION OBJETIVO 4 - ACTIVIDADES:***

El objetivo 4 menciona esencialmente, que los temas no resueltos son manejados lo mas cerca de su origen en el proyecto y es escalado solo cuando ellos no pueden ser resueltos en el proyecto.

#### **Actividad 7**

Las desviaciones identificadas en las revisiones y auditorias (De actividades 4 y 5) son manejadas de acuerdo a un procedimiento documentado.

Solo cuando la desviación no puede ser resuelta a nivel del Gerente del proyecto, SQA levanta el tema con gerentes de mayor jerarquía.

Por supuesto, esto es hecho a través de un procedimiento, el cual implica que hay una rutina del proceso de escalamiento. El proceso incluye revisión del tema con la alta gerencia hasta la resolución (notar que el proceso se supone que trabajara: se espera resolución). También los ítems no cumplidos son documentados, y el documento es gerenciado y controlado.

### **1.1.5) Gestión de Configuración del Software (Software Configuration Management (SCM))**

La Gestión de Configuración, se refiere al control del producto, ya sea la entrega final o un artefacto intermedio del proyecto. Hasta el mas simple de los productos esta supuesto a tener muchos componentes, y cada componente puede tener muchas versiones, por lo que todas las versiones de las piezas y el producto resultante necesitan ser controlados.

#### ***OBJETIVO 1 :***

Las actividades SCM son planeadas.

#### ***OBJETIVO 2 :***

Algunos productos de trabajo de software son identificados, controlados y disponibles.

El plan de SCM determina que tipo de productos de trabajo de software están bajo Gestión de Configuración, esto es, son identificados, controlados, y disponibles.

Los productos de trabajo, son guardados, en lo que se llama “línea base”, un depositario, donde ellos estarán disponibles.

En Gestión de Configuración, se consideran tres principios

- 1) Los ítems tiene un identificador común
- 2) Los ítems son controlados (Estado de uso)
- 3) Los ítems están disponibles

***OBJETIVO 3 :***

Los cambios a los productos de trabajo de software identificados son controlados.

Nuevas versiones de un producto de software, son insertados en la “línea base”, y la versión obsoleta es movida a un archivo.

***OBJETIVO 4 :***

Los grupos e individuos afectados son informados del estado y contenido de las “líneas bases” de software.

Es considerado como un prerequisite para hacer que los productos esten disponibles.

***RELACION OBJETIVO 1 - ACTIVIDADES:***

Actividad 1

Existe un plan de SCM por cada proyecto a través de un procedimiento. Este plan es iniciado tempranamente y es desarrollado al mismo tiempo que el plan del proyecto. El plan es “gerenciado y controlado” revisado y aprobado por los grupos afectados.

Actividad 2

Todas las actividades SCM en el proyecto, son desarrolladas a través del plan.

***RELACION OBJETIVO 2 - ACTIVIDADES:***

Actividad 3



Un sistema de librería de Gestión de configuración es establecida como almacenamiento de las “líneas bases” de software.

Esta actividad parece implicar, que debiera existir una herramienta, sin embargo la palabra “sistema”, implica que esta función también puede ser llevada con un sistema manual.

#### Actividad 4

Especifica identificación de productos de software. Esta actividad implica un esquema de identificación y un identificador único por cada ítem único.

#### Actividad 7

Los productos de la librería de software “línea base” son creados y su versión es controlada a través de un procedimiento documentado.

El procedimiento generalmente implica una Reunion de Control de Configuración de Software (SCCB “Software configuration Control Board”). Este es un grupo que revisa los requerimientos de cambios, autoriza cambios a los ítems de configuración y “líneas bases”, y autoriza construcción de productos.

#### Actividad 8

El estado de los ítems de configuración son registrados de acuerdo a un procedimiento documentado.

El registro incluye el estado y contenido de todos los ítems., versiones pasadas y actuales, y la historia completa de cambios de todos los ítems, de tal manera que todas las versiones previas sean recuperadas.

Actividad 10 :

Ejecución de auditorías de “líneas bases” de acuerdo a un procedimiento documentado.

No solo el contenido del “línea base”, sino las facilidades y estructura del sistema de librería, son evaluadas en conjunto con la integridad del “línea base”.

las actividades 8 y 10 están relacionadas con el objetivo, en el sentido de hacer disponibles los ítems de configuración.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

Actividad 5

Los cambios de requerimientos y reportes de problemas para todos los ítems de configuración son iniciados, registrados, revisados, aprobados, y se les hace seguimiento de acuerdo a un procedimiento documentado.

Actividad 6

Los cambios a las “líneas bases” son llevados a través de un procedimiento documentado.

Bajo el procedimiento, la aprobación de estos cambios será hecho por la propia autoridad (SCCB), la entrada y salida de ítems de la “línea base” deberá preservar la integridad de la “línea base”, y se deberán efectuar pruebas de regresión.

### ***RELACION OBJETIVO 4 - ACTIVIDADES:***

Relacionado a mantener informado a grupos e individuos de las actividades SCM.

Actividad 9

Reportes estándar de SCM, actividades y contenidos de líneas bases de Software, están disponibles para los grupos afectados.

La clase de reportes son : Minutas de SCCB (Software configuration control board), resumen de requerimientos de cambios, reportes de problemas, historia de revisiones, así como resultados de auditoria, ayudan a proveer una visión del proceso SCM para la organización. Así como en actividad similar en SQA, la función de SCM puede utilizar esta práctica para reportar a sus clientes en la organización.

### **1.1.6) Gestión de Subcontratistas de Software (Software Subcontract Management SSM)**

· El SSM, puede ser considerado como la aplicación de los otros KPAs a subcontratistas.

#### ***OBJETIVO 1 :***

· El contratista principal selecciona subcontratistas de software calificados.

#### ***OBJETIVO 2:***

El contratista principal y el subcontratista de software están de acuerdo con los compromisos de cada uno de ellos.

#### ***OBJETIVO 3 :***

El contratista principal y el subcontratista de software, mantienen una comunicación constante.

#### ***OBJETIVO 4:***

El contratista principal hace un seguimiento de los resultados y del desenvolvimiento del subcontratista de software contra sus compromisos.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

##### **Actividad 1**

Se relaciona con el uso de un procedimiento en la definición y planificación del trabajo a ser subcontratado.

Esta actividad aplica solo para software y pasa en un proyecto de software.

El procedimiento envuelve y presupone los requerimientos de software, El plan de desarrollo de software, y todos los componentes de dicho plan (Ej. Planes de SQA y SCM). Como resultado de esta actividad, hay un plan para seleccionar un subcontratista basado en estos ítems en el procedimiento.

Actividad 2 :

EL subcontratista es seleccionado a través de una evaluación.

Esta actividad también presupone un procedimiento. Esta vez para seleccionar un subcontratista basado en una evaluación de las habilidades requeridas para satisfacer el plan.

El procedimiento considerara, entre otras características, su previo desenvolvimiento, proximidad geografica, Capacidad de Ingeniería de software, y personal disponible.

### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

Actividad 3 :

El contrato es la base para gerenciar los compromisos.

Esta parece una prácticanormal de negocios, pero existen muchos casos donde el contrato no registra los acuerdos informales hechos por las personas en ambos lados, ya sea inicialmente o mientras el proyecto esta en marcha.

Esta informalidad y los malentendidos resultantes, son las razones porque las actividades 4 y 6 están asociadas con este objetivo.

Actividad 4

El subcontratista tiene un plan de desarrollo de software documentado, y requiere de la revisión y aprobación del contratista principal.

#### Actividad 6

Especifica que un procedimiento gobierna, cambios en los compromisos de los subcontratistas, o en las condiciones y términos del contrato.

- Las prácticas de esta actividad son similares a aquellas relacionadas a cambios en los requerimientos (Gestión de requerimientos), o cambios de compromisos en el KPA PTO.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

#### Actividad 7 :

El contratista principal conduce revisiones periódicas con el subcontratista a nivel gerencial.

#### Actividad 8

EL contratista principal realiza revisiones técnicas periódicas.

La actividad 7 asegura que los niveles de gerencia tengan revisiones periódicas, y la actividad 8 que los niveles técnicos tengan también revisiones e intercambios periódicos.

La revisión de gerencia es a nivel de temas programáticos (Costos, cronogramas, finanzas), así como riesgos y dependencias críticas. Los ítems son asignados y seguidos hasta su culminación.

Las reuniones técnicas se enfocan en compromisos y de asegurarse que la implementación de los requerimientos es correcta.

#### Actividad 9

Revisiones formales de acuerdos reflejados en el contrato.

Estas revisiones formales son conducidas por un procedimiento documentado. Los compromisos y riesgos así como los resultados son investigados, las decisiones son hechas y registradas, y las acciones son documentadas.

Los resultados de estas revisiones deberían ser coleccionadas y registradas por el contratista principal, y esta información de “estado” acumulado debería ser evaluada periódicamente. La información del desenvolvimiento sería reusada en la selección de futuros subcontratistas.

***RELACION OBJETIVO 4 - ACTIVIDADES:***

Actividad 13

El desenvolvimiento del subcontratista es evaluado, y este es informado.

Actividad 9

Revisiones formales a través de procedimiento.

Actividad 10 :

El SQA del contratista principal supervisa las actividades de SQA del subcontratista

Actividad 11

El SCM del contratista principal supervisa las actividades de SCM del Subcontratista.

Las actividades 10 y 11 son simétricas: ellas requieren la inspección de parte de los grupos de SQA y SCM del contratista principal, respectivamente a las actividades de SQA y SCM del subcontratista de acuerdo a un procedimiento documentado.

El uso de un procedimiento implica que la revisión es mas que casual y tiene un contenido regular y sistemático.

#### Actividad 12

EL contratista principal conduce una prueba de aceptación de los productos entregados por el subcontratista, a través de un procedimiento documentado.

. El proceso de aceptación, se refiere a un criterio de aceptación acordado por ambos, el contratista principal y el subcontratista, registrando los resultados de la prueba y planes de acción para remediar los defectos revelados en el producto.

En lo que se refiere a la modalidad de subcontratos de personal técnico de otra compañía, para trabajar dentro la organización, al lado del personal del contratista principal, (Modalidad “Body-shopping”), el KPA SSM no pareciera encajar muy bien. Las revisiones de desenvolvimiento son hechas sobre el individuo, y no sobre la compañía proveedora. La interfase entre compañías en el caso del “Body-shopping” es personal, entre individuos mas que entre organizaciones, haciendo dificultosa la aplicación del KPA SSM.



## **1.2) NIVEL DE MADURACION 3 : EL PROCESO DEFINIDO**

En el nivel de maduración 3 las mejores prácticas de proyectos han sido generalizadas para el uso de la organización como un todo.

Las mejores prácticas han sido codificadas en lo que el CMM llama el estándar de la organización, y los procesos definidos de software.

Los procesos definidos no restringen proyectos. Por el contrario, los proyectos ahora tendrán maneras explícitas de adaptar el conocimiento acumulado que todos los proyectos encontraron valioso.

### **1.2.1) Enfoque de Proceso de Organización (Organization Process Focus (OPF))**

#### ***OBJETIVO 1***

El desarrollo del proceso de software y las actividades de mejoramiento, son coordinadas a través de la organización.

Se refiere a la coordinación de actividades para el desarrollo y mejoramiento de los procesos en todos los niveles de la organización y a través de todas sus funciones.

#### ***OBJETIVO 2 :***

Las fuerzas y debilidades de un proceso de software utilizadas son identificadas en relación a un procesos estándar.

#### ***OBJETIVO 3 :***

Los procesos de desarrollo y actividades de mejoramiento a nivel de la organización son planeadas.

Como en el nivel 2, si hay un proyecto, debe haber un plan de proyecto. Para este proyecto a nivel de toda la compañía, existen planes de mejoramientos de proceso.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

Actividad 3

Coordinación explícita, a nivel de la organización, de actividades de mejoramiento de proceso (SPI - Software Improvement Process) para proyectos de software y para la organización.

Esto se realiza, cuidando el estándar de procesos de software así como el proyecto de proceso de software definido.

El texto de la actividad 3 se refiere a dos otros KPA del nivel 3, Definición de la Organización del Proceso (Organization Process Definition OPD) y a Gestión de Software Integrado (Integrated Software Management ISM).

Estos tres KPAs (OPF, OPD y ISM), están interrelacionados en sus prácticas. Juntos estos KPAs conciernen a dos tipos genéricos de procesos de software en la visión CMM : “El Estándar de la Organización”, y el “Proyecto definido”.

El proceso de software estándar de la organización es una descripción del proceso general de software, que cada proyecto en la compañía se espera que siga.

El proyecto de proceso de software definido es adecuado de los estándares de procesos de software de la organización, para que encaje en los requerimientos técnicos y no-técnicos. El proyecto de proceso de software definido contendría, o apuntaría a elementos específicos del proceso enmarcado en el procesos de la organización : estándares, procedimientos, metodologías, y funciones (Como SQA, SCM, etc.)

#### Actividad 6

Coordinación de entrenamiento.

Para la implementación y uso de los procesos de software de la organización y del proyecto.

El entrenamiento beneficia a toda la organización, y la coordinación para la organización es hecha a través de este objetivo.

Actividad 7 :

Los grupos envueltos en la implementación de los procesos de software, son informados acerca de las actividades que se llevan a cabo a través de toda la organización y de los proyectos de software, para desarrollo y mejora de los procesos de software.

La actividad 7 puede ser considerada como la función de publicidad, del grupo de Ingeniería de software.

Actividad 4 :

Coordinación de uso.

La actividad 4 menciona que hay una base de datos de procesos de la organización, y que su uso es coordinado a través de toda la organización. El depósito de esta actividad contiene los datos recolectados, desde medidas de actividades de proceso de software, y productos de trabajo, mas cualquier información necesaria para interpretar estos datos.

Actividad 5

Supervisión.

Puede ser considerada como la función de transferencia tecnológica del grupo de ingeniería de software: Nuevas tecnologías -procesos, métodos, y herramientas - en uso limitado en la organización son supervisadas y evaluados. Así las tecnologías que son valiosas para ponerlas en práctica, son transferidas para su uso generalizado.

### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

#### Actividad 1:

El Proceso de Software es evaluado periódicamente.

· La evaluación es una forma de métodos de apreciación. La evaluación es una auto-apreciación. La organización utiliza su mismo personal, para efectuar la evaluación. Los resultados de la evaluación son propiedad de la organización, y ella puede escoger, hacerlos conocidos a externos o no.

El otro tipo de apreciación es una auditoria, donde un equipo de externos determina los resultados del proceso, pero los resultados son reportados a terceros, y fuera de la organización. Generalmente este tercero es un cliente que está considerando la aprobación del contrato con la compañía que está siendo evaluada.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

#### Actividad 1:

Los planes de acción dirigen los resultados de las evaluaciones periódicas.

Un plan de acción a diferencia de un plan de proyecto, dirige los temas levantados en una evaluación. El plan contiene los lineamientos generales así como detalles - la estrategia y tácticas -, de como los temas de procesos serán dirigidos. (Objetivos, recursos, y responsabilidades de la gente envuelta).

#### Actividad 2

La organización desarrolla y mantiene un plan para las actividades de desarrollo y mejoramiento de su proceso de software.

Da ciertas guías en los contenidos del plan de acción: Cronogramas para todo el ciclo de mejoramiento, prioridades de los temas a ser dirigidos, y recursos a ser usados.

### **1.2.2) Definición del Proceso de la Organización (Organization Process Definition (OPD))**

#### ***·OBJETIVO 1***

Un proceso de software estándar para la organización es desarrollado y mantenido.

#### ***·OBJETIVO 2:***

La información relacionada al uso del proceso de software estándar de la organización en los proyectos de software es recolectada, revisada y puesta a disposición.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

##### **Actividad 1**

El desarrollo y mantenimiento del estándar del procesos de software de la organización se ha convertido en un conjunto de rutinas o etapas en un procedimiento documentado.

Además de otros ítems, el procedimiento gobierna el uso de políticas y estándares impuestos en la organización (generalmente por el cliente, o por terceros, como autoridades gubernamentales o regulatorias).

El procedimiento indica, como los cambios propuestos al estándar de proceso de software son revisados, aprobados, e implementados en proyectos en marcha. El procedimiento también indica revisiones entre colegas (peer reviews), de la descripción del estándar del proceso de software.

Las personas que realizan las revisiones, deben ser personas experimentadas en la definición del proceso de software en un área diferente de la organización, o ingenieros de software de diversas disciplinas que utilizaran el proceso estándar.

#### Actividad 2:

El estándar de proceso de software es documentado de acuerdo a los mismos estándares aplicados a otros estándares de la organización.

Los estándares típicos deberían describir los elementos del proceso (estimación, diseño, etc.) y estándares aplicables para procesos y productos, responsabilidades de las personas que aplican el proceso, y como los elementos del proceso se relacionan unos a otros.

#### Actividad 3

Los ciclos de vida del software aprobados para el uso en los proyectos son descritos y sus descripciones son mantenidas.

#### Actividad 4

Guías y criterios para los proyectos son gerenciadas y controladas.

### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

#### Actividad 5

La base de datos del proceso de software y de productos de trabajo de la organización es mantenida.

Ejemplos de data de producto : Tamaños, defectos descubiertos.



Ejemplos de data de proceso : Revisiones entre colegas y eficiencia, tasa de descubrimiento de defectos, pruebas y eficiencia. La base de datos contiene tanto estimados como datos reales. Y hay un control apropiado de la data ingresada (Ejemplo : Chequeos de razonabilidad).

#### Actividad 6

Los documentos del proceso de software son recolectados y mantenidos como activos del proceso. Estos documentos incluirían proyectos de procesos definidos de software (Estos son ejemplos de como se ajustan en el proceso de la organización), y los otros documentos del proyecto (planes, procedimientos, materiales de entrenamiento). La librería de proceso debería dar fácil acceso a los grupos de software relacionados a través de un catalogo, y los mantenedores de la librería deberán obtener opiniones y retroalimentacion en su uso para mejorar su contenido.

### **1.2.3) Programa de Entrenamiento (Training Program (TP))**

En el nivel 3 una organización tiene su proceso de software bajo control: Es estable y sus resultados son repetitivos. Debido a esta estabilidad, la organización puede hacer ahora cambios de proceso a nivel de la organización. El programa de entrenamiento es uno de estos, en el son hechos cambios generalizados.

#### ***OBJETIVO 1 :***

Las actividades de entrenamiento son planeadas. Se espera ver un plan de entrenamiento en marcha, cubriendo las necesidades de la organización.

#### ***OBJETIVO 2 :***

Se provee entrenamiento para el desarrollo de habilidades y conocimientos necesarios para ejecutar roles de Gestión de software y técnico.

#### ***OBJETIVO 3 :***

Los individuos en el grupo de ingeniería de software y grupos relacionados de software reciben el entrenamiento necesario para ejecutar sus roles.

***RELACION OBJETIVO 1 - ACTIVIDADES:***

Actividad 1 :

Cada proyecto, desarrolla y mantiene un plan de entrenamiento. Esta actividad provee una lista de alto nivel de todas las cosas a considerar en el plan de entrenamiento.

Actividad 2 :

El Plan de entrenamiento de la organización es desarrollado y revisado.

***RELACION OBJETIVO 2 - ACTIVIDADES:***

Actividad 3

El entrenamiento de la organización es ejecutado de acuerdo al plan de entrenamiento.

Actividad 4

Los cursos de entrenamiento preparados a nivel de la organización, son desarrollados y mantenidos de acuerdo a los estándares de la organización.

Entre otras cosas, los estándares deberían especificar por ejemplo : objetivos del curso y prerequisites, como el entrenamiento va a ser evaluado, y como los materiales del cursos son revisados (quizás por expertos en entrenamiento y expertos en el tema}.

***RELACION OBJETIVO 3 - ACTIVIDADES:***

Actividad 5

Un procedimiento para determinar si los individuos ya tienen el conocimiento y habilidades que ellos necesitan.

Esta actividad es como una cláusula de escape para la organización. Una persona puede no necesitar un tipo de entrenamiento, por ejemplo si ella ya ha obtenido dicho conocimiento en otra compañía. Pero debe haber una manera sistemática y efectiva para determinar la posesión de esta habilidad.

#### Actividad 6

Los registros de entrenamiento son mantenidos.

El valor del CMM aquí es que los registros de entrenamiento están disponibles para asignar los trabajos al personal.

#### **1.2.4) Gestión de Software Integrado (Integrated Software Management (ISM))**

El KPA ISM dice que un proyecto es planeado, seguido, y gerenciado al mas alto nivel de maduración mirando primero el estándar de proceso de software de la organización, y entonces adaptando este proceso genérico.

##### ***.OBJETIVO 1 :***

El proyecto con Procesos de software definido es una versión a la medida del estándar del proceso de software.

##### ***OBJETIVO 2 :***

El proyecto es planeado y gerenciado de acuerdo al proceso de software definido para los proyectos.

##### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

###### **Actividad 1**

Desarrolla el proceso de software definido para proyectos, ajustando a la medida el proceso estándar de la organización y a través de un procedimiento documentado.

El procedimiento se refiere a los siguientes ítems típicos

Un apropiado ciclo de vida es seleccionado del proceso estándar de la organización y modificado con la ayuda de una guía para este propósito.

La descripción del proceso del proyecto es registrada y revisada por el Grupo de ingeniería de Software.

###### **Actividad 2**

Revisa el proyecto de software definido usando un procedimiento documentado.

***RELACION OBJETIVO 2 - ACTIVIDADES:***

**Actividad 3**

El plan de software del proyecto es desarrollado y revisado a través de un procedimiento documentado.

El procedimiento deriva el plan de software para el proceso de software definido del proyecto.

**Actividad 5 :**

Los datos para la estimación de los parámetros del proyecto (tamaño, costo, esfuerzo, etc.) son tomados desde la base de datos del proceso de la organización, y son hechas comparaciones razonables con datos de proyectos similares.

**Actividad 4 :**

El proyecto es gerenciado a través de proceso de software definido del proyecto.

**Actividad 6, 7 y 8 :**

Todos los factores cuantitativos estimados en “nivel 2”, tamaño de productos de trabajo, esfuerzo y costo de software, y recursos críticos de computadora, son manejados a través de un procedimiento documentado.

En nivel 2 estos factores de estimación fueron seguidos por actividades de PTO. Ahora en nivel 3 hay suficiente experiencia para utilizar procesos establecidos para manejar estos factores cuantitativos y esta experiencia es codificada en procedimientos.

**Actividad 9 :**

Gestión de camino crítico, y dependencia crítica. El procedimiento especifica que el cumplimiento de tareas, derivados de un procesos de software definido, es caracterizado por no ser ambiguo.

Actividad 10 :

Relacionado con Riesgos de proyecto, y su identificación, evaluación, documentación, y Gestión a través de un procedimiento documentado.

Alcanzando nivel 3, la organización ha codificado su práctica de Gestión de riesgos, a pesar de el riesgo es una cosa abstracta (Un riesgo no existe, excepto como una posibilidad).

En nivel 3, la evaluación del impacto de riesgo, como los riesgos son supervisados, y que acciones de contingencia serán tomadas, si la supervisión muestra que el riesgo, puede suceder, formaría parte del procedimiento.

Esta actividad también incluye la comunicación de los riesgos, planeamiento de riesgos y supervisión de estos incluyendo la participación del cliente.

### **1.2.5) Ingeniería de Productos de Software (Software Product Engineering (SPE))**

En este KPA, el CMM menciona que es lo que normalmente reconoce como Ingeniería de Software : Análisis de requerimientos, diseño, codificación, y prueba.

Este es el KPA donde realmente se desarrolla la Ingeniería y se produce software.

#### ***OBJETIVO 1***

Las tareas de ingeniería de software son definidas, integradas y consistentemente ejecutadas para producir software.

#### ***OBJETIVO 2:***

Los productos de trabajo de software se mantienen consistentes unos con otros.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

Actividad 1 :

Métodos y herramientas de ingeniería de software apropiadas son integradas en el proyecto.

Parte de esta integración, es la documentación de la racionalidad para seleccionar los métodos y herramientas y poner estas herramientas bajo la Gestión de Configuración.

Actividad 2 :

Los requerimientos de software son desarrollados, mantenidos, documentados, y verificados sistemáticamente analizando los requerimientos asignados de acuerdo al proceso de software definido del proyecto.



### Actividad 3

El diseño del software es desarrollado, mantenido, documentado, y verificado de acuerdo al proceso de software definido del proyecto, para acomodar los requerimientos de software y formar la base para la codificación.

### Actividad 4

La codificación del software es desarrollado, mantenido, documentado y verificado, de acuerdo al proceso de software definido del proyecto, para implementar los requerimientos de software y diseño de software. En esta actividad así como en la 2 y 3, son usados métodos efectivos de programación, y los desarrolladores revisan la salida de los requerimientos y las etapas de diseño para resolver temas.

Para estas tres actividades, requerimientos, diseño y codificación, la salida es bajo Gestión de Configuración y requiere revisión de colegas.

### Actividad 5,6 y 7

Pruebas de software son ejecutadas de acuerdo al procesos de software definido para proyectos.

Este proceso debería cubrir, en adición a otras “sub-prácticas”, el como determinar si la prueba es adecuada (Incluyendo niveles de pruebas :Unitario, Integrado, de Sistema, etc.), estrategia de pruebas, (Caja blanca, caja negra, etc.), Que cubre la prueba, y criterios de pruebas. Las pruebas de regresión son hechas a nivel de cada prueba, cuando un cambio es hecho al software o su ambiente. Y los planes de prueba así como los casos de prueba son documentados.

#### Actividad 6

Las pruebas de integración son planeadas y ejecutadas de acuerdo al proceso de software definido para proyectos.

#### Actividad 7

Para pruebas de sistemas y aceptación, estas son planeadas y ejecutadas para demostrar que el software satisface sus requerimientos.

Las pruebas de sistemas y aceptación, son preparadas por un grupo independiente a los desarrolladores. Los problemas identificados en las pruebas son documentados y seguidos hasta su resolución, y todos los resultados de las pruebas son registradas.

#### Actividad 8 :

La documentación que será usada para operar y mantener el software es desarrollada y mantenida de acuerdo al proceso de software definido para el proyecto.

Métodos y herramientas apropiadas, así como especialistas en documentación están envueltos.

#### Actividad 9

Los datos y los defectos identificados en revisiones de colegas y pruebas son recolectados y analizados de acuerdo al proceso de software definido para el proyecto. Los tipos de defectos de datos incluyen : Donde el defecto fue descubierto (En cual producto de trabajo de software y que paso en el proceso), el escenario de pruebas que estaba siendo corrido, la severidad del defecto, etc. Estos datos son otra fuente para la visión cuantitativa de Nivel 4 y son un Activo para la organización.

***RELACION OBJETIVO 2 - ACTIVIDADES:***

Actividad 10 :

Se mantiene consistencia entre productos de trabajo de software.

### **1.2.6 Coordinación Intergrupos (Intergroup Coordination (IC))**

El propósito de la Coordinación Intergrupos es la de proveer un mecanismo por el cual los grupos de ingeniería de software participen con los otros grupos de ingeniería. El resultado debería ser la de satisfacer todas las necesidades de los clientes con respecto a todo el proyecto, incluyendo el software.

#### ***OBJETIVO 1 :***

Los requerimientos del cliente están acordados por todos los grupos afectados. Como Cliente también se refiere al Usuario final.

#### ***OBJETIVO 2 :***

Los compromisos entre los grupos de ingeniería están acordados por los grupos afectados.

#### ***OBJETIVO 3 :***

Los grupos de ingeniería identifican, hacen seguimiento, y resuelven temas de intergrupos.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

##### **Actividad 1**

El grupo de ingeniería de software y los otros grupos de ingeniería de software participan con el cliente y usuarios finales, como sea apropiado, para establecer los requerimientos del sistema.

Esta participación también incluye dependencias críticas entre grupos y los criterios de aceptación del producto por el cliente.

## ***RELACION OBJETIVO 2 - ACTIVIDADES:***

### **Actividad 3**

Un plan documentado es usado para comunicar compromisos intergrupos, para coordinar y hacer seguimiento del trabajo ejecutado.

Todos los grupos afectados registran sus compromisos de cronograma en el plan, para ítems técnicos y contractuales, y sus responsabilidades de unos a otros. Y el plan, incluyendo revisiones de este, debería acordarse por el gerente del proyecto y los grupos afectados.

### **Actividad 4**

El proceso de identificación, negociación, y seguimiento de dependencias críticas entre los grupos de ingeniería es tan rutinario, que este es llevado a cabo a través de un procedimiento documentado.

El procedimiento gobernará las dependencias críticas entre el plan de coordinación y especificara como las dependencias son negociadas, como los acuerdos en estos son documentados, revisados, y aprobados, como se les hace seguimiento y como son tomadas las acciones correctivas.

### **Actividad 5**

Los productos de trabajo recibidos por otros grupos de trabajo, distintos al grupo de ingeniería revisan los productos de trabajo ingresados para asegurarse que sus necesidades están satisfechas.

El criterio para las revisiones de los productos de trabajos son documentados y acordados en el plan de coordinación.

El significado principal de estos dos primeros objetivos, es que no debe haber sorpresas entre los grupos cooperando en un proyecto porque la interfase entre los grupos y su interacción es manejada por un buen protocolo o etiqueta.

Uno como miembro del grupo participante se espera que levante temas con antelación por el bien de todo el proyecto. A pesar de que ese tema causara que el cronograma se retrase, esta persona no será culpada.

La coordinación entre grupos no podría ser efectiva en una cultura donde culpar al que da el mensaje, sea la reacción normal por las malas noticias.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

#### **Actividad 2 :**

El grupo de ingeniería de software y otros grupos trabajan juntos, quizás a través de representantes, para supervisar y coordinar actividades técnicas y resolver temas técnicos.

El enfoque de la actividad 2 es en los requerimientos y diseño del sistema así como también en riesgos técnicos, y perturbaciones en temas técnicos.

Para temas no resueltos por los grupos o representantes, la actividad 6 provee un proceso de decisión que es invocado por el procedimiento.

#### **Actividad 6**

Temas de intergrupos no resueltos por representantes de individuos son manejados por un procedimiento documentado.

## Actividad 7

Los grupos de Ingeniería conducen revisiones periódicas e intercambios. Un lineamiento de estas reuniones podría incluir, necesidades de clientes , usuarios asi como que también están ellos satisfechos. También la Agenda podría incluir el estado de actividades técnicas, compromisos, y estado de riesgos y requerimientos.

### **1.2.7) Revisión entre Colegas (Peer Reviews (PR))**

El KPA PR es la única práctica específica de ingeniería de software, que tiene su propia área de proceso en CMM.

La razón para este énfasis en el CMM, es la casi universalmente experiencia de los ingenieros de software que las revisiones entre Colegas son efectivas para prevenir defectos que pudieran pasar a etapas posteriores del ciclo de vida de desarrollo.

El termino “revisión entre colegas” (Peer review), corresponde a cualquier examen de un producto de trabajo, por un compañero o colega del que lo produjo, esto es, revisiones de cada nivel de formalidad, desde un chequeo de escritorio por un colega hasta una altamente formal inspección de software.

#### ***OBJETIVO 1 :***

Las actividades de revisión entre colegas son planeadas.

#### ***OBJETIVO 2 :***

Los defectos en los productos de trabajo son identificados y removidos.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES:***

##### **Actividad 1**

La revisión entre colegas es planeada, y el plan es documentado.

Los estándares de proceso de software de la organización especificara el conjunto de productos de trabajo típicamente revisados. Así mismo, el cronograma para las revisiones, los nombres de los supervisores y revisores serán incluidos.



## ***RELACION OBJETIVO 2 - ACTIVIDADES:***

### **Actividad 2**

Las revisiones entre colegas son ejecutadas de acuerdo a un procedimiento documentado.

El procedimiento especifica las lecciones principales que la industria de software ha aprendido acerca de la conducción de revisiones entre colegas.

Las revisiones tienen un Supervisor (Líder), que es entrenado para esta actividad.

El material para ser revisado es distribuido, con anticipación para que los revisores tengan adecuada preparación y tiempo.

### **Actividad 3:**

Los datos en la conducción y resultados de una revisión entre colegas es registrada.

Los datos cubren factores como horas de preparación, horas de tiempo de revisión por pagina de documento, tamaño y composición del equipo de revisión. También, los datos registrados son relacionados con el producto revisado y el número y tipo de defectos encontrados y eventualmente corregidos.

Los datos defectuosos, nunca debe ser usada para evaluar el desenvolvimiento de un individuo.

### **1.3) NIVEL DE MADURACION 4 - EL PROCESO DE GESTIÓN**

En el nivel 2 las mejores prácticas están en los proyectos. En el nivel 3 la organización había dominado la técnica de propagar las mejoras prácticas a través de la organización. Las mejores prácticas de los proyectos fueron usadas para definir e implementar un proceso estable para la organización.

Ahora todos los activos acumulados de proceso de prácticas de nivel 2 y nivel 3 son utilizados por la organización de nivel 4 para dar soporte a proyectos con un entendimiento cuantitativo y un proceso estable.

### **1.3.1) Gestión de Proceso Cuantitativo (Quantitative Process Management (QPM))**

#### ***OBJETIVO 1 :***

Las actividades de Gestión de procesos cuantitativos son planeadas

#### ***OBJETIVO 2 :***

El desenvolvimiento del proceso definido de software del proyecto es controlado cuantitativamente.

#### ***OBJETIVO 3:***

La capacidad del proceso de un proceso de software de la organización es conocido en términos cuantitativos.

Se toman mediciones de desenvolvimiento de proceso. Cuando los datos medidos indican una variación que debería ser controlada, se efectúan ajustes al proceso casi a tiempo real. El resultado es alcanzar control cuantitativo de desenvolvimiento.

Desde este control de cada proyecto, una “línea base” de parámetros de control es acumulado en proyectos de todo tipo.

En nivel 4, la organización integra estos parámetros de control sobre todos los proyectos para alcanzar una definición cuantitativa de la capacidad del proceso de la organización. Esta definición cuantitativa de capacidad reside en la base de datos del proceso de software de la organización. Y los datos de capacidad de proceso de la organización es usada por proyectos para definir sus propios objetivos de desenvolvimiento, los cuales finalmente serán reflejados en mejoras de procesos a través del KPA Gestión de cambios de proceso en nivel 5.

## ***RELACION OBJETIVO 1 - ACTIVIDADES:***

### **Actividad 1**

Existe un procedimiento para el desarrollo de un plan. Los datos de desenvolvimiento y otros proyectos se obtienen de la base de datos de proceso (mantenida por el KPA OPD de nivel 3).

Los objetivos de desenvolvimiento para el proyecto, el programa de mediciones a nivel de la organización (basada en los datos obtenidos por el KPA OPD), y de la estrategia de los planes de negocio.

El plan de proyecto para el proceso de calidad también viene de una cadena de eventos procesos estándar de software de la organización adecuado para dar el proceso de software definido del proyecto, con el manejo posterior del plan de desarrollo de proyectos de software.

El plan de desarrollo provee los objetivos del proyecto para calidad de software, productividad, y reducciones de los tiempos del ciclo de desarrollo.

### **Actividad 2 :**

La actividad 2 menciona que la Gestión cuantitativa del proyecto de su proceso a través de datos esta de acuerdo con el plan QPM de la actividad 1.

Las actividades de QPM incluyen la especificación de como las etapas del proceso seran instrumentadas (que y como medir), quien es responsable, los recursos que ellos usaran, y el cronograma que seguirán.

### **Actividad 3**

Se refiere a la estrategia de recolección de datos a ser seguida y que análisis de datos cuantitativos serán hechos. Algunos elementos de esta actividad son las relaciones entre tareas del proceso y productos, los puntos de recolección de datos en las tareas, y los puntos del

proceso al cual el control será aplicado. Estos elementos están basados en el proceso definido de software del proyecto.

***RELACION OBJETIVO 2 - ACTIVIDADES :***

**Actividad 4 :**

Los datos medidos utilizados para el control son recolectados a través de un procedimiento.

En el nivel 4, los objetivos de medición sirven no solamente al proyecto, sino a toda la organización, por lo tanto al menos algunos datos, deben ser consistentes entre los proyectos.

El procedimiento debería especificar la definición de las métricas, como estas serán analizadas e interpretadas para su uso, y los puntos de recolección a través del ciclo de vida (incluyendo post-desarrollo).

**Actividad 5**

El proceso definido de software del proyecto es controlado. El procedimiento de la actividad 5 especifica el análisis de los datos y técnicas a ser utilizadas, el valor esperado y la desviación de los datos a ser recolectados, los límites de control para correcciones de desviaciones en medidas, y la comparación de medidas y desviaciones.

**Actividad 6 :**

Se refiere al reporte de los resultados de la Gestión del proceso cuantitativo del proyecto. Estos datos del proceso deben ser diseminados para ser valiosos a nivel de la organización.

***RELACION OBJETIVO 3 - ACTIVIDADES :***

**Actividad 7 :**

Desarrollo y mantenimiento de la “línea base” de la capacidad del proceso estándar (La mas alta expresión del proceso de software de la organización).

La “línea base” del desenvolvimiento del proceso del proyecto resultó desde el análisis y control cuantitativo bajo actividad 5 del Objetivo 2.

Estas “líneas bases del desenvolvimiento” del proceso son recolectadas en la base de datos del proceso de software de la organización y utilizadas para calcular la “línea base” de la capacidad del proceso de la organización.

### **1.3.2) Gestión de Calidad de Software (Software Quality Management (SQM))**

Un entendimiento cuantitativo de la calidad de los productos de software y el logro del incremento de la calidad del producto.

#### ***OBJETIVO 1 :***

Las actividades de la Gestión de Calidad del proyecto de software son planeadas.

#### ***OBJETIVO 2 :***

Objetivos medibles para la calidad del producto de software y sus prioridades son definidas.

#### ***OBJETIVO 3 :***

El progreso actual de alcanzar los objetivos de calidad para los productos de software es cuantificado y gerenciado.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

##### **Actividad 1**

El plan de calidad de software es desarrollado y mantenido a través de un procedimiento.

El procedimiento especifica, entre otras cosas que se escribe un plan de calidad para el proyecto y se ha revisado por los grupos afectados, aquellos que tienen una participación en esa calidad.

##### **Actividad 2**

El proyecto sigue el plan para la Gestión de la Calidad de los productos de software de la actividad 1.

### ***RELACION OBJETIVO 2 - ACTIVIDADES:***

#### **Actividad 3**

Los objetivos de calidad del producto son definidos para el proyecto. Las características de la calidad del producto derivan de los requerimientos de software.

Estos requerimientos de calidad se convierten en objetivos de calidad numéricos, cuando los valores de los datos describiendo estas características de calidad son producidos del plan de mediciones.

#### **Actividad 5**

Asigna apropiados objetivos de calidad de producto a los subcontratistas, y estos objetivos son una nueva “entrada” en la práctica de la Gestión de Subcontratistas.

### ***RELACION OBJETIVO 3 - ACTIVIDADES :***

#### **Actividad 4**

La calidad del producto es medida y comparada esporádicamente con los objetivos de calidad.

Al inicio de cada tarea, el equipo encargado de la tarea revisa los objetivos de calidad del producto, determina, como la tarea esta relacionada con estos objetivos, activa los planes para alcanzar los objetivos, y revisa cualquier cambio de proceso asociado.

Entonces en cada etapa del ciclo de vida, las características de calidad de los productos resultantes son medidas y comparadas con los objetivos de calidad. Si los objetivos de calidad no son alcanzados, se toman las acciones especificadas en la actividad 4. Si hay un conflicto



entre los objetivos de calidad, entonces un análisis de costo es hecho para alcanzar los objetivos de calidad, y las prioridades de los objetivos son decididos basado en la estrategia de la compañía y preferencias de clientes o usuarios finales.

#### **1.4) NIVEL DE MADURACION 5 - EL PROCESO DE OPTIMIZACION**

Al alcanzar el nivel 5, el mecanismo del proceso de la organización opera rutinariamente para entregar sus productos de software.

Estos mecanismos trabajan internamente y corresponden a la manera normal de hacer negocio.

El proceso de producción de software es virtualmente automático; hay incluso un proceso para manejar desviaciones en los procesos.

Las organizaciones en nivel 5 , se concentran en cambios del proceso de software por razones de estrategia de negocio : Ventajas de competencia en calidad, productividad, y reducción de tiempo. Cuando los mecanismos de producción son efectivos, eficientes, y bajo control, la organización puede enfocarse en mejorar estos.

#### **1.4.1) Prevención de Defectos (Defect Prevention (DP))**

##### ***OBJETIVO 1 :***

Las actividades de prevención de defectos son planeadas.

##### ***OBJETIVO 2 :***

Las causas comunes de los defectos son identificadas.

##### ***OBJETIVO 3 :***

Las causas comunes de los defectos son priorizadas y sistemáticamente eliminadas.

##### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

###### **Actividad 1**

El proyecto desarrolla un plan para sus propias actividades de prevención de defectos.

###### **Actividad 2**

Al inicio de cada tarea de software, los miembros del equipo perteneciente a la tarea, se reúnen para preparar las actividades de la tarea y actividades de prevención de defectos.

La agenda de la reunión utilizara como “entrada” los objetivos de calidad del producto (Del KPA SQM), traducidos a nivel de tareas. La agenda de la reunión también incluirá “salidas” (o ejemplos de “salidas”) a ser producidos, como las “salidas” de la tarea serán evaluadas, y como verificar que el proceso de software de la tarea es seguido. Habrá también una lista de errores típicamente producidos y sugerencias de como prevenir estos.

### ***RELACION OBJETIVO 2 - ACTIVIDADES :***

Una causa común de un defecto, es una propiedad regular del proceso involucrado, a diferencia de una causa especial. Existen dos actividades para alcanzar la identificación de los defectos por causas comunes.

#### **Actividad 3:**

Especifica reuniones de análisis de causas para búsqueda de defectos y un método de identificación, a ser llevadas a cabo por un proceso, definido a través de un procedimiento.

El equipo de la tarea conduce estas reuniones al final de la tarea, cuando se produce el defecto, durante la tarea, periódicamente durante la tarea, y después de la entrega del producto al cliente.

#### **Actividad 5**

Los resultados de una reunión de análisis de causas son documentadas, registradas y seguidas a través de los equipos, coordinando actividades de prevención de defectos.

Se hace seguimiento de las acciones para prevenir defectos, por su estado y resultados.

### ***RELACION OBJETIVO 3 - ACTIVIDADES:***

#### **Actividad 4**

Los equipos coordinando las actividades de prevención de defectos mantienen reuniones para revisar las acciones propuestas y como implementar estas.

La “salida” de estas reuniones es la selección de propuestas a ser convertidas en acciones, registrando la racionalidad de las decisiones, y asignando las responsabilidades para la ejecución de las acciones. Las reuniones también efectúan seguimiento del “estado” de las acciones y registran los resultados de los experimentos de prevención de defectos.

#### Actividad 6

Un procedimiento documentado, para revisión de los estándares de software de la organización.

#### Actividad 7

Un procedimiento documentado, para revisión del proceso de software definido del proyecto.

Estas revisiones resultan de acciones exitosas de prevención de defectos de la actividad 4.

#### Actividad 8

El “estado” y resultado de las actividades de prevención de defectos, a través de la organización, son informadas a los grupos relacionados con el software.

Todos estos grupos estarán participando de alguna manera en acciones de prevención de defectos, y deberán ser informados del progreso y resultados, especialmente porque los efectos probablemente tomarán un tiempo en aparecer.

#### **1.4.2) Gestión de Cambios de Tecnología (Technology Change Management (TCM))**

En nivel 5, la organización se visualiza a si misma a través de los procesos. Esto significa que la organización entiende cuanto contribuye el proceso a su capacidad de producción de software.

Ventajas competitivas y salud financiera son alcanzadas no solo por adquisiciones o reorganizaciones, sino también por un continuo y sistemático mejoramiento del proceso. La organización de nivel 5 entiende como mejorar su capacidad de software, mejorando continuamente su proceso.

Este entendimiento de como gerenciar cambios es reflejado en los nombres de los últimos dos KPAs Gestión de Cambios de Tecnología y Gestión de Cambios de Proceso.

##### ***OBJETIVO 1 :***

La incorporación de cambios de tecnología es planeada.

##### ***OBJETIVO 2 :***

Las nuevas tecnologías son evaluadas para determinar sus efectos en calidad y productividad.

##### ***OBJETIVO 3 :***

Nuevas tecnologías apropiadas son transferidas en las prácticas normales en toda la organización.

##### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

###### **Actividad 1**

La organización desarrolla y mantiene un plan para la Gestion de Cambios de Tecnología. El plan para la Gestión de Cambios de Tecnología gobierna el proceso para dicho cambio. No se

refiere a cambios particulares, los cuales son llevados a cabo por otras actividades. Hay también un grupo a cargo de la Gestión de Cambios de Tecnología.

El plan estratégico para la Gestión de Cambios de Tecnología especifica : Los recursos y responsabilidades, la estrategia técnica para la automatización de partes del proceso de software estándar de la organización, y los procedimientos para Gestión de cambios. El plan describirá también el camino que seguirá la organización para seleccionar áreas de proceso para automatización, para estimar la vida económica de una tecnología, como decidir si hacer o comprar, y como evaluar tecnologías candidatas.

### ***RELACION OBJETIVO 2 - ACTIVIDADES :***

#### Actividad 2

Las actividades 2 y 4 están relacionadas con la identificación de las áreas, donde nuevas tecnologías serian beneficiosas. En la actividad 2 el grupo TCM, trabaja con proyectos para identificar estas áreas: El grupo solicita sugerencias y también conduce búsquedas, de nuevas tecnologías disponibles en el comercio.

#### Actividad 4

En la actividad 4, el grupo TCM escoge donde aplicar nueva tecnología, mirando el estándar de proceso de software de la organización sistemáticamente. El análisis del grupo incluirá los factores económicos, que áreas de proceso están envueltas, los ingresos esperados en el proceso, y prioridades para los posibles cambios. Los datos cuantitativos disponibles del nivel 4 alimentan este análisis. El grupo recomendará donde se deberían considerar esfuerzos pilotos.

#### Actividad 5:

Las tecnologías son seleccionadas y obtenidas para la organización y sus proyectos de acuerdo a un procedimiento documentado. El típico procedimiento especifica que ellos analizan requerimientos para nuevas tecnologías y conducen análisis preliminares de costo beneficio

#### Actividad 6

Especifica que en la transferencia técnica de una tecnología particular, se utilizan proyectos pilotos, cuando sea apropiado. La actividad 6 implica que para la organización de nivel 5, no es una norma hacer cambios al proceso; generalmente se efectuaran pequeños cambios bajo control de un plan piloto.



### **1.4.3) Gestión de Cambios de Proceso (Process Change Management (PCM))**

#### ***OBJETIVO 1 :***

El continuo mejoramiento del proceso es planeado.

#### ***OBJETIVO 2 :***

La participación de las actividades de mejoramiento del proceso de software de la organización se efectúa en toda la organización.

#### ***OBJETIVO 3 :***

El estándar del proceso de software de la organización y el proceso definido de los proyectos es mejorado continuamente.

#### ***RELACION OBJETIVO 1 - ACTIVIDADES :***

Actividad 2:

Un grupo responsable por el mejoramiento del proceso de software coordina el trabajo.

Actividad 3

La organización utiliza un conjunto de estándares de pasos (procesos) codificados en un procedimiento para desarrollar y mantener el plan para las mejoras del proceso de software.

El plan es a nivel de toda la organización, y es basado en los planes y estrategia de negocios de la compañía, y en mediciones de satisfacción de clientes. El plan es revisado por colegas y gerentes afectados.

#### Actividad 4

Las actividades de mejoramiento de proceso son conducidas a través de un plan. El plan típico debe considerar, Objetivos, prioridades entre los objetivos, y recursos. Las responsabilidades para las actividades son asignadas, especialmente a los equipos.

#### ***RELACION OBJETIVO 2 - ACTIVIDADES :***

##### Actividad 1:

Se establece un programa para mejoramiento de proceso de software.

##### Actividad 6:

Las personas trabajan en equipos para desarrollar mejoramientos de procesos, esencialmente una práctica de planeamiento. Los objetivos de mejoramiento, cuantitativamente expresados, son parte del plan, junto con compromisos de soporte financiero y aprobaciones de los gerentes afectados y del grupo manteniendo las descripciones del proceso de software definido.

##### Actividad 10

Las personas encargadas de hacer el mejoramiento del software, personal técnico y gerentes, esencialmente, cada uno recibe opiniones de los esfuerzos de mejoramiento del proceso.

#### ***RELACION OBJETIVO 3 - ACTIVIDADES :***

##### Actividad 6

Se refiere a un procedimiento para el manejo de propuestas de mejoramiento de proceso.

El procedimiento especifica como las propuestas son entregadas y como cada propuesta es evaluada y decidida.

Las propuestas son implementadas sin un esfuerzo piloto, son ingresadas en el plan de mejoramiento de proceso, y los equipos de mejoramiento son asignados para implementar estas.

Se hace un seguimiento al estado de las mejoras a ser implementadas y los resultados son registrados y reportados a las personas que tuvieron la idea original porque cada uno enfoca las mejoras como parte de su propia contribución.

Otras propuestas (Aquellas que tienen tareas piloto) son manejadas en la Actividad 7.

#### Actividad 8

En esta actividad se considera la parte de salida de este KPA. Un proceso, reflejado en un procedimiento, para implementación de mejoras. Los elementos principales de este procedimiento son : Estimación y recursos, especificando como el cambio resultante en la desenvolvimiento del proceso será medido y seguido, y decidiendo en cuando y como proveer consultas y soporte.

#### Actividad 9

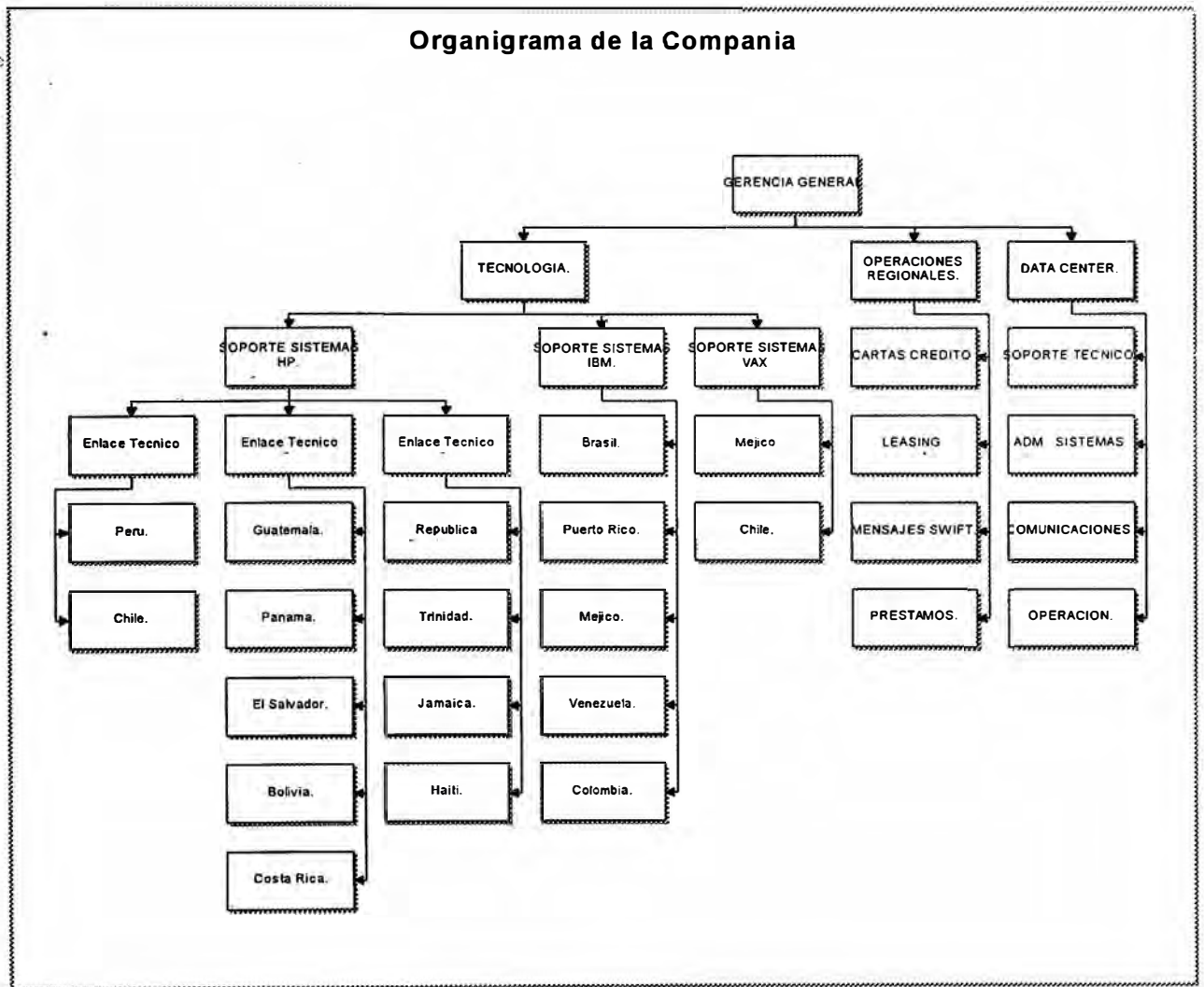
Especifica que se mantienen registros de las actividades de mejoramiento de proceso. El registro envuelto contiene estado de esfuerzos actuales así como crónicas de esfuerzos pasados. Estos registros reflejan la capacidad de cambio del proceso de la organización y su ciclo de tiempo del cambio de proceso.

## **CAPITULO II - ENTORNO DE APLICACION DEL MODELO**

El modelo CMM, se aplicó específicamente en una organización que forma parte de una corporación bancaria a nivel mundial, la cual presta servicios de mantenimiento y desarrollo del software, para las sucursales bancarias de Latino América y el Caribe.

La organización en el tiempo en que el proyecto de CMM nivel 2 se desarrolló, estaba estructurada de la siguiente forma (Ver Organigrama en Figura 4).

FIGURA 4



El modelo se aplicó a nivel de toda la organización, y para ser utilizado en los mantenimientos de los Sistemas existentes, así como en los desarrollos de nuevos sistemas.

Durante la primera etapa del proyecto CMM nivel 2, toda la organización fue entrenada en el modelo, y se trabajó en la confección de procedimientos y guías, para la aplicación del modelo. El equipo del proyecto se dedicó a desarrollar las guías, que soportaban los KPAs del modelo, y en base a estas guías, cada área de la organización debería confeccionar procedimientos específicos que cumplan las guías y por lo tanto cumplan los KPAs de CMM nivel 2.

Por otro lado, la organización debió someterse a un examen de evaluación, a todo nivel, a cargo del personal del SEI (Software Engineering Institute), con el cual se determina si la organización está o no en el nivel 2.

Es necesario mencionar, que dicho examen debe ser aprobado por cada una de las áreas. Basta que una área no alcance el nivel, para que toda la organización no pueda subir al nivel 2. Por esta razón, la importancia de entrenamiento a todo nivel de la organización.

Mis responsabilidades durante la implementación de CMM nivel 2, era la de enlace técnico para las sucursales de Perú y Chile.

Las tareas que se nos encomendó, fue la de desarrollar procedimientos y manuales del departamento, de tal manera que el proceso de desarrollo y mantenimiento de software se efectúe de acuerdo a los lineamientos de CMM Nivel 2.

Adicionalmente se efectuó un cambio en la organización de mi departamento, que debió estar sincronizado con los procedimientos desarrollados.

## **CAPITULO III - PREPARACION A NIVEL EMPRESA CMM NIVEL 2**

### **3.1 Generalidades**

Para el proyecto CMM nivel 2 se formo un grupo que reportaba directamente a la Gerencia General, el cual se encargo del entrenamiento del personal, así como de la preparación de guías para la creación de Procedimientos y Manuales, para cada departamento de la organización.

El grupo del proyecto, asesoraba la confección de los procedimientos y Manuales de cada departamento, en base a las guías previamente desarrolladas.

Las guías permitían adecuar la realidad de cada departamento, en base a la realidad de cada uno de ellos.

El grupo debía también revisar los borradores de los procedimientos y aprobarlos.

Asi mismo el grupo debió encargarse de coordinar los entrenamientos de los KPAs de nivel 2 a todo el personal de la Empresa. Estos entrenamientos tenían como objetivo, inculcar al personal, el modelo CMM, con el fin de que puedan responder correctamente en la evaluación a ser ejecutada por el personal del SEI (Software Engineering Institute).

### 3.2 Guías para la creación de Procedimientos y Manuales y Sistemas utilizados

La siguiente tabla muestra la relación de guías desarrolladas, para la creación de procedimientos y manuales en los distintos departamentos de la empresa, y su relación con cada uno de los KPAs del CMM nivel 2

**TABLA III**

<b>Kpa Nivel 2</b>	<b>#doc.</b>	<b>Descripcion de documento</b>
<b>Gestión de Requerimientos (RM)</b>	<b>1</b>	<ul style="list-style-type: none"> <li>• <b>Sistema Laser para control de requerimientos</b></li> </ul>
	<b>2</b>	<ul style="list-style-type: none"> <li>• <b>Requerimientos de Negocio</b></li> </ul>
	<b>3</b>	<ul style="list-style-type: none"> <li>• <b>Especificaciones Funcionales</b></li> </ul>
	<b>4</b>	<ul style="list-style-type: none"> <li>• <b>Listas de Chequeo</b></li> </ul>
<b>Evaluación de Calidad (SQA)</b>	<b>5</b>	<ul style="list-style-type: none"> <li>• <b>Guía del Plan de Evaluación de Calidad de Software</b></li> </ul>
	<b>6</b>	<ul style="list-style-type: none"> <li>• <b>Lista de Chequeo para Ciclo de Vida de Desarrollo de Software</b></li> </ul>
<b>Gestión de Configuración (SCM)</b>	<b>7</b>	<ul style="list-style-type: none"> <li>• <b>Guía del Plan de Gestión de Configuración de Software</b></li> </ul>
<b>Planeamiento de Proyectos de Software (PTO)</b>	<b>8</b>	<ul style="list-style-type: none"> <li>• <b>Mantenimiento y mejoras de Software (M&amp;MS) - Para proyectos de menos de 40 días esfuerzo o menores a US\$ 50,000</b></li> </ul>
	<b>9</b>	<ul style="list-style-type: none"> <li>• <b>Manual de desarrollo de Software (SDH Software development Handbook) - Para proyectos de mas de 40 días de esfuerzo o mayores a US\$ 50,000</b></li> </ul>
<b>Seguimiento de Proyectos de Software (SPP)</b>	<b>10</b>	<ul style="list-style-type: none"> <li>• <b>Software de Gestión de Proyectos (Workbench)</b></li> </ul>
<b>Gestión de Subcontratistas de Software (SSM)</b>	<b>11</b>	<ul style="list-style-type: none"> <li>• <b>Política de Contratación de Sub Contratistas</b></li> </ul>



Debido a la extensión de cada una de estas guías, solo describimos resumidamente cada una de ellas.

### **1) Sistema Láser para Control de Requerimientos**

#### ***Descripción General :***

El Sistema “Láser”, es un sistema diseñado para registrar los requerimientos de cambios o mejoras de software (RFS).

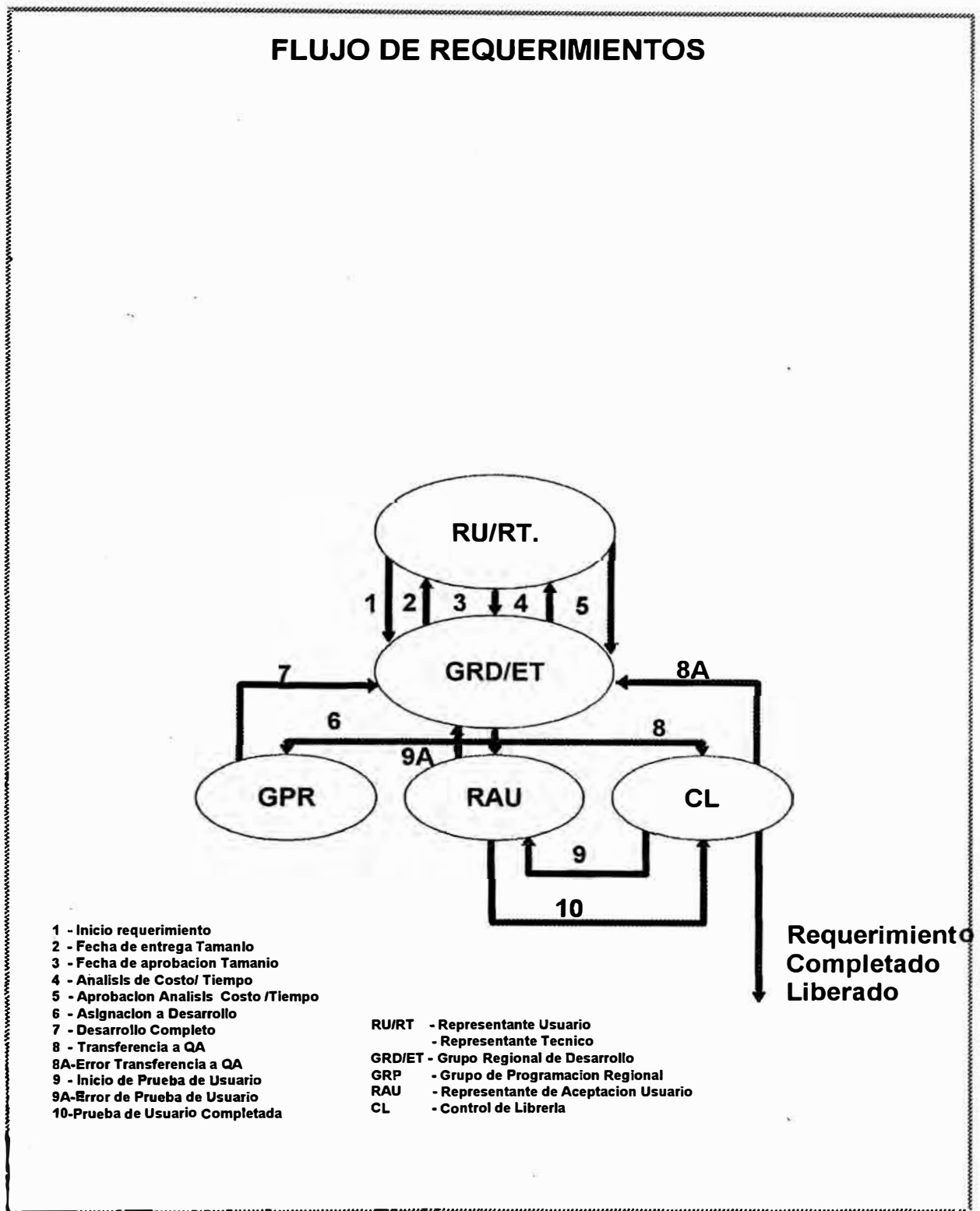
El Sistema esta desarrollado sobre una plataforma VAX, y conectada a través de la red de la compañía, para ser accesible, desde cualquier país.

El sistema presenta un mecanismo de aprobación por cada etapa de proceso del “RFS”. El esquema adjunto muestra las distintas etapas de proceso, y los puntos de aprobación.

El sistema permite registrar con un Código único el requerimiento, y se puede complementar y enriquecer a lo largo del ciclo de vida del desarrollo, con especificaciones de usuarios, análisis costos/beneficios, fechas de entrega, recursos a ser utilizados, criterios de aceptación, tipos de riesgos, prioridades, entre otros conceptos.

En la figura 5 adjunta, se puede esquematiza el flujo de requerimientos.

**FIGURA 5**



### ***Flujo del Requerimiento desde el Origen hasta su Liberación***

El Representante de Usuario (RU) o el Representante técnico (RT), envía el requerimiento al Grupo Regional de Desarrollo (GRD) o al enlace técnico (ET).

El Grupo Regional de Desarrollo (GRD) ingresa la fecha en la cual informara el tamaño del proyecto, y luego de que el Representante de Usuario (RU) aprueba dicha fecha, el GRD entrega un tiempo estimado y costos para completar este requerimiento.

El RU aprueba el estimado y lo envía de regreso al GRD para su desarrollo.

El GRD asigna el requerimiento al Grupo Regional de Programación (GRP) para efectuar el desarrollo en ese requerimiento.

Cuando el GRP completa el requerimiento, el GRD envía este al Control de Librería (CL), para su transferencia a QA (Evaluación de Calidad).

El Control de la Librería (CL), efectúa la catalogación y envía al Representante de Aceptación de Usuario (RAU) para la prueba de Usuario y comentarios.

Cuando el RAU prueba satisfactoriamente el requerimiento, el lo transmite al Control de la Librería (CL), para liberación y uso por el país.

Si el RAU tiene un problema con el requerimiento, el envía es de regreso al Grupo Regional de Desarrollo (GRD), quien en turno envía este al Grupo Regional de Programación (GRP) para corrección, y el requerimiento regresa a través del ciclo completo nuevamente.

## **2) Requerimientos de Negocio**

### ***Descripción General:***

Este documento, deberá ser llenado por los Representantes de Negocio, de los países, y permite determinar, conceptualmente el objetivo del requerimiento, así como los alcances, e impactos del cambio en el negocio.

En la figura 6 adjunta, se muestra el formato de requerimiento de negocio, y las instrucciones de llenado del mismo.



<b>III. LISTA DE CHEQUEO DE REQUERIMIENTOS PRINCIPALES</b>		
<b>Tipo</b>	<b>Requerido</b>	<b>Comentarios/Referencias</b>
1. Aprobación Gerente Regional Producto	<input type="checkbox"/> Si <input type="checkbox"/> No	
2. Análisis Costo/Beneficio	<input type="checkbox"/> Si <input type="checkbox"/> No	
3. Legal/Regulatorio	<input type="checkbox"/> Si <input type="checkbox"/> No	
4. Contingencia	<input type="checkbox"/> Si <input type="checkbox"/> No	
5. Interdependencias con otros productos o sistemas	<input type="checkbox"/> Si <input type="checkbox"/> No	
6. Reconciliación con otros sistemas y/o aplicaciones	<input type="checkbox"/> Si <input type="checkbox"/> No	
7. Reportes	<input type="checkbox"/> Si <input type="checkbox"/> No	
8. Nuevo Hardware	<input type="checkbox"/> Si <input type="checkbox"/> No	
9. Adquisición de Software	<input type="checkbox"/> Si <input type="checkbox"/> No	
10. Desviaciones a estándares corporativos	<input type="checkbox"/> Si <input type="checkbox"/> No	
11. Pantallas	<input type="checkbox"/> Si <input type="checkbox"/> No	
12. Documentación de Usuario	<input type="checkbox"/> Si <input type="checkbox"/> No	
13. Acuerdos de nivel de servicio	<input type="checkbox"/> Si <input type="checkbox"/> No	
14. Seguridad de Información	<input type="checkbox"/> Si <input type="checkbox"/> No	
15. Otros		

<b>IV. DETALLE DE DEFINICION DE FUNCION (Requerido para cada función)</b>	
1. Nombre de Función y/o número	
2. Tipo de Función:	<input type="checkbox"/> Batch <input type="checkbox"/> On-line
3. Propósito de Función:	
4. Flujo y Descripción del Proceso	4.1      Entrada: 4.2      Proceso (Lógica General del Proceso): 4.3      Salida: 4.4      Consideraciones de Integridad:
5. Requerimientos de Data	5.1      Descripción de Data (Atributos y Dependencias): 5.2      Validación de Data y Criterios de Aceptación : 5.3      Volúmenes de Data (Tasa de Crecimiento, expansión):
6. Factores de Tiempo:	6.1      Frecuencia de Ejecución: 6.2      Tiempo de respuesta esperada:: 6.3      Otros temas de desenvolvimiento:
7. Controles de Operación	
8. Controles de Seguridad:	
9. Requerimientos de Hardware:	
10. Requerimientos Legales/regulatorios:	
11. Requerimientos de Contingencia:	
12. Interfases con Otros Sistemas :	12.1      Tipo (Tiempo Real o Batch): 12.2      Consideraciones de Tiempo: 12.3      Totales de Control 12.4      Controles Manuales
13. Comentarios:	

**Instrucciones de llenado :**

<b>Part</b>	<b>Número de campo/nombre</b>	<b>Descripción</b>
<b>I</b>	<b>Información de Identificación de Requerimiento</b>	
	1. Número de Control de Requerimiento	Este es el número asignado por LASER o la correspondiente herramienta de iniciación y sistema de seguimiento.
	2. Versión	Número de versión de la actual especificación. Esta es necesaria con el fin de asegurarse que la versión mas actualizada es la especificación aplicable a ser usada en el desarrollo y la aprobada mas recientemente. Adicionalmente, este es un paso inicial relacionado al Gestión de configuración del documento.
	3. Fecha	La fecha que el requerimiento fue iniciado.
	4. Ambiente	La plataforma de Hardware en la cual el Sistema/aplicación corre, por ejemplo: VAX, IBM,PC,etc.
	5. Sistema/Aplicación	El nombre del sistema o la aplicación. Ejemplo : Contabilidad, Prestamos, etc.
	6. Nombre de Area de Negocios	Nombre Operacional o de función del área requiriendo el cambio. Ejemplo : Control Financiero, Administración de Créditos. FX, etc.
	7. Usuario/Gerente del Negocio	Gerente del Area funcional a cargo del área definida en número 6 arriba.
	8. País	Nombre del País requiriendo el cambio.
	9. Nombre del Producto y Familia de Producto.	Nombre del producto y/o familia de producto. Ej: Gestión de Cash, etc.
	10. Prioridad de Requerimiento/Secuencia de implementación	Explicado por si mismo. Sin embargo, cuando se esta requiriendo un servicio de ala prioridad, tener en cuenta el impacto operacional del negocio, y el trabajo actual en progreso. Cuando sea aplicable, indicar la secuencia de implementación para este requerimiento. Esto puede implicar que otros requerimientos pendientes serán postpuestos o retrasados.
	11. Beneficios	Seleccionar una de las cuatro categorías proveídas en este campo.
	12. Propósito de requerimiento	Seleccionar una de las cuatro categorías proveídas en este campo.
	13. Preparado por	Nombre y número de teléfono de la persona que preparo la especificación.
	14. Fecha	Fecha en la cual la especificación fue preparada
	15. Aprobación	Nombre y Aprobación del oficial encargado de la aprobación
	16. Fecha	Fecha de aprobación de la especificación



<b>Part</b>	<b>Número de campo/nombre</b>	<b>Descripción</b>
<b>II</b>	<b>Alcances y propósitos del requerimiento</b>	
	1. Situación Actual	Resumir el proceso actual o situación a ser afectada por el cambio requerido.
	2. Objetivo del Cambio Requerido	Descripción completa del objetivo del cambio y su impacto asociado en otras áreas.
	3. Solución Propuesta	Describir en términos de negocio la solución propuesta o cambios a ser implementados.
	4. Espacio Adicional	Si existe espacio adicional es requerido, indicar un número de referencia para la continuación de cualquier información de arriba.
<b>III</b>	<b>Lista de Chequeo de Requerimientos Principales</b>	
	1. Aprobación del Gerente Regional de Productos	Indicar si la aprobación del Gerente Regional de Productos es requerida. Esto es aplicable para todas las mejoras.
	2. Análisis Costo/Beneficio	Indicar si el Análisis Costo/Beneficio es requerido.
	3. Legal/Regulatorio	Indicar si el cambio es el resultado de un requerimiento Legal/ Regulatorio.
	4. Contingencia	Indicar si el Plan de Contingencia es afectado
	5. Inter-dependencias con otros productos y Sistemas	Indicar si existe dependencias con otros productos o sistemas y si se requiere considerarlo para este requerimiento.
	6. Reconciliación con otros Sistemas/Aplicaciones	Indicar si se requiere efectuar reconciliaciones con otros sistemas.
	7. Definición de Reportes	Indicar si son generados nuevos reportes o si los reportes actuales son afectados. Incluir el Layout del Reporte.
	8. Nuevo Hardware	Indicar si el nuevo Hardware es requerido.
	9. Adquisición de Software	Indicar si el nuevo Software es requerido. Ejemplo : Automatización de Oficina, Comunicaciones, Herramientas de Desarrollo, Bases de Datos, etc.
	10. Desviaciones en los Estándares de la Corporación	Indicar si una desviación es requerida para la implementación de los cambios de Software requerido, como el de desarrollo de unidad(es) de Operación, incapaces de cumplir con los Estándares de la Corporación.
	11. Pantallas	Indicar si la definición de pantallas son afectadas o no. Incluir el Layout de la pantalla.
	12. Documentación de Usuario	Indicar si la documentación de Usuario necesita actualización o no.
	13. Acuerdos de Nivel de Servicio	Indicar si se requiere o no Acuerdos de Nivel de Servicio. o si los existentes necesitan ser actualizados.
	14. Seguridad de la Información	Indicar si alguno de los requerimientos identificados por los Estándares de Seguridad de Información afectaran este requerimiento de cambio.
	15. Otros	Indicar cualquier otro tema que pueda afectar la implementación del cambio requerido.

<b>Part</b>	<b>Número de campo/nombre</b>	<b>Descripción</b>
<b>IV</b>	<b>Detalle de la Definición de Función</b>	Esta sección deberá ser preparada por cualquier función incluida en el requerimiento
	1. Nombre y/o Número de Función	Nombre y/o Número de Función.
	2. Tipo de Función	Indicar si la función es ejecutada en línea o "batch"
	3. Propósito de la Función	Explicación detallada para implementar esta función.
	4. Descripción de Proceso y Flujo	Detallar la descripción del proceso para la función incluyendo su efecto en otras funciones actuales y flujo de los requerimientos.
	4.1 Entrada	Indicar que procesos o data será usada como entrada de la función.
	4.2 Proceso	Indicar como las entradas en los procesos y/o datos manejaran los cambios requeridos. Las consideraciones adicionales deberían ser dadas a la lógica de proceso, afectando las interfases a otros sistemas y/o aplicaciones.
	4.3 Salida	Describir los datos de salida o nuevos procesos como resultado de 4.2, esto puede incluir pantallas nuevas o modificadas, reportes, e interfases a otros sistemas y/o aplicaciones.
	4.4 Consideraciones de Migración	Identificar cualquier consideración especial que debe ser considerada en la implementación del cambio requerido. esto incluye: Prioridad de Implementación, Procesos periódicos claves, requerimientos de contingencia, entrenamientos especiales, actualización de procesos de Data Center, etc.
	5. Requerimientos de Datos	Definición detallada de todos los campos de datos en uso para esa función.
	5.1. Descripción de Datos	Describir detalladamente los atributos de los datos y las dependencias.
	5.2. Validación de Datos y Criterios de Aceptación	Describir el criterio de validación para todos los campos. situaciones de integridad, rangos de tolerancia, márgenes de error, etc. Adicionalmente, definir los criterios de aceptación de las pruebas. Estas serán utilizadas para desarrollar el plan de prueba unitaria, y plan de pruebas de sistemas integrado.
	5.3. Volúmenes de datos	Describir el volumen aproximado de transacciones y o reportes esperados por el periodo corriente (Año, Trimestre, mes, etc.) Esto será usado para determinar los requerimientos de almacenamiento y su efecto en la desenvolvimiento del sistema.
	6. Factores de Tiempo	Identificar todos los elementos de tiempo los cuales serán afectados por este cambio. Cualquiera de estos requerimientos afectaran los Criterios de Aceptación del Usuario, y el desarrollo del plan de pruebas.

<b>Part</b>	<b>Número de campo/nombre</b>	<b>Descripción</b>
	6.1. Frecuencia de Ejecución	Indicar el número de veces este cambio será ejecutado. por ejemplo, los requerimientos en línea son usualmente ejecutados según la demanda, los reportes también pueden ser ejecutados según la demanda o en una frecuencia periódica (Anualmente, Trimestralmente, Mensualmente, etc.)
	6.2. Tiempo de Respuesta Esperada	Indicar el Tiempo de respuesta para la función o transacción. Para transacciones En línea esto es expresado en número de segundos, mientras que para reportes o funciones en batch puede tomar mas tiempo, est es también conocido como el tiempo de corrida.
	6.3. Otros Temas de Desarrollo	Cualquier otro tema relacionado con tiempo no considerado.
	7. Controles de Operación	Indicar si se debe establecer controles de operación adicionales o no para esa función, o si esta función afecta los controles actuales de operación y la acción requerida.
	8. Controles de Seguridad	Indicar si se necesita implementar controles adicionales de seguridad, esto incluye actualización de perfiles de usuario, derechos de acceso, esto puede afectar la clasificación de la información (restringida, confidencial, interna, publica).
	9. Requerimientos de Hardware	Detallar la definición de los requerimientos de nuevo hardware. Por ejemplo : Impresoras, módem, etc. Si no es el caso, indicar No.
	10. Requerimientos Legales regulatorios	Descripción detallada o referencia(s) completa(s) de los requerimientos Legales/regulatorios. Incluir una copia de la regulación con la(s) especificación(es).
	11. Requerimientos de Contingencia	Descripción detallada de requerimientos nuevos de contingencia. Indicar como el plan de contingencia será afectado y reprogramar una nueva prueba (Si es aplicable).
	12. Interfases con otros Sistemas	Indicar la(s) interfase(s) requeridas a otro(s) sistema(s). Si no es requeridas interfases indicar esto como "No".
	12.1. Tipo	Indicar si es tiempo real (En línea) o Batch.
	12.2. Consideraciones de Tiempo	Indicar si hay o no restricciones de tiempo o ventanas entre las cuales la interfase debe ser producida o ejecutada.
	12.3. Totales de Control	Indicar si se consideran generar totales de control. con el fin de asegurarse que la interfase ha sido generado apropiadamente.
	12.4. Totales Manuales	Indicar si tiene que ser implementados controles manuales, junto con el producto de software con el fin de asegurarse que la interfase ha sido generada apropiadamente.
	13. Comentarios	Indicar cualquier otro requerimiento no considerado

### **3) Especificaciones Funcionales**

#### ***Descripción General:***

Este documento, deberá ser llenado por los Representantes de Negocio, de los países, y permite especificar funcionalmente el requerimiento, desde el punto de vista del Usuario.

Este documento servirá como base para la confección del Análisis de Impacto y Diseño Técnico.

En la figura 7 adjunta se muestra el Formato de Especificaciones Funcionales.

**FIGURA 7**

**ESPECIFICACIONES FUNCIONALES - FORMATO**

<b>SECCION I. Iniciación de Requerimiento</b>	
Número de Control de requerimiento	Versión:
Nombre de Area de Negocio:	Gerente de Negocio:
Nombre de Producto y Familia de Producto:	Justificación:
Preparado Por:	Fecha:

<b>Promotor (es)</b>	
1)	2)

<b>Aprobaciones de Especificaciones</b>			
<b>Nombre</b>	<b>Titulo Funcional</b>	<b>Firma</b>	<b>Fecha</b>

<b>Alcances y Propósito del requerimiento</b>
<b>ALCANCE</b>
<b>PROPOSITO</b>

**ESPECIFICACIONES FUNCIONALES - FORMATO (CONT.)**

<b>NIVEL DE RIESGO DEL NEGOCIO:</b>
<b>FECHA CRÍTICA:</b>

<b>Equipo del Proyecto</b>			
<b>Nombre</b>	<b>Localización</b>	<b>Telef.</b>	<b>Responsabilidades</b>

<b>Lista de Escalamiento</b>			
<b>Razón/Evento</b>	<b>Contactos</b>	<b>Telef.</b>	<b>Secuencia</b>

**ESPECIFICACIONES FUNCIONALES - FORMATO (CONT.)**

<b>SECCION II - Narrativa de Sumario</b>	
<b>Sumario del Requerimiento:</b>	
<b>Objetivos del Software/Cambios requeridos y Adiciones al Sistema actual :</b>	
<b>Descripción del Sistema Actual</b>	
<b>Hardware</b>	
<b>Software</b>	
<b>Telecomunicaciones</b>	
<b>Funcionalidad</b>	
<b>Ambiente de la Aplicación requerida</b>	
<b>Conectividad:</b>	
<b>Estación de Trabajo</b>	
<b>Software Sistema</b>	
<b>Software Aplicación</b>	
<b>Operación DC</b>	
<b>Otros</b>	

<b>Comunidad De Usuarios</b>		
<b>Tipo Usuario</b>	<b>Localización</b>	<b>Requerimientos</b>
<b>SECCION III - Funciones</b>		
<b>Sumario de Requerimientos Principales</b>		
<b>Tipo</b>	<b>Requerido</b>	<b>Comentarios/Referencias</b>
<b>Aprobación Gerente de Producto:</b>		
<b>Análisis Costo/Beneficio:</b>		
<b>Legal/regulatorio:</b>		
<b>Contingencia:</b>		
<b>Sistemas de Interfases:</b>		
<b>Reconciliaciones otros Sistemas:</b>		
<b>Reportes:</b>		
<b>Nuevo Hardware:</b>		
<b>Adquisición de Software</b>		



<b>Detalle de Definición de Funciones</b>	
<b>Nombre de Función/Número:</b>	
<b>Tipo de Función:</b>	<b>Batch</b>  <b>En Línea</b>
<b>Propósito de la Función:</b>	
<b>Descripción del Proceso:</b>	
<b>Descripción de Datos (Atributos):</b>	
<b>Dependencias de Datos y Relaciones:</b>	
<b>Volúmenes de Datos (expansión, tasa de crecimiento):</b>	
<b>Criterio de Edición/Validación (Rangos de Tolerancia):</b>	
<b>Frecuencia de Ejecución:</b>	
<b>Tiempo de respuesta esperado, Otros temas de desenvolvimiento:</b>	
<b>Controles de Operación:</b>	
<b>Controles de Seguridad:</b>	
<b>Requerimientos de Hardware:</b>	
<b>Requerimientos legales/regulatorios:</b>	
<b>Requerimientos de Contingencia:</b>	
<b>Interfases con Otros Sistemas:</b>	

<b>Detalle de Definición de Funciones (Cont.)</b>	
<b>Lógica general del Proceso:</b>	
<b>Comentarios:</b>	

#### **4) Listas de Chequeos**

##### ***Descripción General***

Para facilitar el cumplimiento de los procedimientos implementados en la Gestión de Requerimientos, se desarrollaron listas de chequeo de los diversos documentos, a generarse durante el ciclo de vida de desarrollo de software.

Estas listas permiten la evaluación sistemática de la calidad del proceso, con el fin de detectar y corregir cualquier deficiencia, en dichos procesos.

Las siguientes son las listas creadas :

- Lista de Chequeo - Requerimientos de Negocio
- Lista de Chequeo - Especificaciones Funcionales
- Lista de Chequeo - Especificaciones Técnicas
- Lista de Chequeo - Inspección de Código
- Lista de Chequeo - Plan de Pruebas de Sistemas

**FIGURA 8**

## LISTA DE CHEQUEO - REQUERIMIENTOS DE NEGOCIO

Nombre de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

**Propósito:** *Los requerimientos de negocio deben ser definidos y documentados así el producto resultante entregado puede ser verificado y/o probado contra ellos.*

Check No.	Actividad Ejecutada	Si	No	N/A
	<b><i>El Documento de requerimientos de negocio.....</i></b>			
1	Especifica el propósito y alcance del sistema propuesto ?			
2	Contiene una descripción del proceso?			
3	Especifica que datos son requeridos para el proceso?			
4	Especifica con que frecuencia el proceso necesita ser corrido o ejecutado?			
5	Especifica los resultados esperados del proceso?			
6	Especifica el desenvolvimiento del sistema?			
7	Identifica otros departamentos y grupos a ser impactados?			
8	Incluye un análisis de Costo/Beneficio?			
9	Especifica cualquier restricción de Costos o cronograma?			
10	Describe el impacto del negocio por el sistema propuesto?			
11	Considera los requerimientos de contingencia (Si es requerido)?			
12	Considera los requerimientos legales, regulatorios y de auditoria?			
13	Describe los procesos de reconciliación con sistemas existentes?			
14	Especifica los volúmenes estimados de transacciones?			
15	Describe las interfases requeridas con otros sistemas?			
16	Especifica los requerimientos de reportes?			
17	Tiene las aprobaciones necesarias del Gerente de Producto?			

# LISTA DE CHEQUEO - ESPECIFICACIONES FUNCIONALES

Número de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

## Propósito:

*El objetivo de las especificaciones funcionales es la de trasladar los requerimientos de negocio en términos que puedan ser entendidos por el analista del sistema/aplicación. Esta describe que va hacer el sistema, y que restricciones va a haber, como necesidades de tiempo de respuesta y desenvolvimiento, en adición con interfases externas a otros sistemas.*

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>El documento de Especificaciones funcionales...</i>			
1	Establece los objetivos del sistema y los propósitos técnicos?			
2	Describe la lógica general del proceso?			
3	Establece o define la plataforma técnica y el sistema de archivo a ser usado?			
4	Especifica las funciones principales del sistema?			
5	Describe los requerimientos detallados para cada función?			
6	Justifica en detalle cualquier desviación de los Estándares de la Corporación?			
7	Especifica los requerimientos de Hardware y software ?			
8	Identifica los clientes y/o usuarios finales?			
9	Describe todas las responsabilidades asignadas en el proyecto?			
10	Describe el plan de contingencia?			
11	Detalla los requerimientos de seguridad?			
12	Define los aspectos legales y regulatorios?			
13	Especifica volúmenes de transacciones anticipados?			
14	Especifica requerimientos de retención de datos?			
15	Detalla cualquier requerimiento de licencias?			
16	Define cualquier restricción de desenvolvimiento?			
17	Especifica la disponibilidad del sistema, tiempo de respuesta y requerimientos de desenvolvimiento?			
18	Define sistemas de control requeridos (Reportes para auditoria, restricciones de ingreso al sistema, etc.)?			
19	Define requerimientos de respaldos/reinicio/recuperación? Incluye requerimientos de duración de recuperación?			
20	Define entidades de datos?			
21	Define relaciones de datos?			
22	Define atributos de datos?			

## LISTA DE CHEQUEO - ESPECIFICACIONES FUNCIONALES (Cont.)

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>El documento de Especificaciones funcionales....</i>			
23	Define población de datos?			
24	Define expandibilidad de datos?			
25	Define requerimientos de edición/validación de datos?			
26	Define interfases de usuarios (Diseño de pantallas, diseño de reportes, ediciones "defaults", etc.)?			
27	Define interfases entre-sistemas (propósito, frecuencia, requerimientos de datos)?			
28	Establece los beneficios del sistema propuesto?			
29	Identifica cualquier requerimiento de Identificación de Usuario?			

# LISTA DE CHEQUEO - ESPECIFICACIONES TECNICAS

Nombre de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

**Propósito:**

*El objetivo de las Especificaciones Técnicas es la de trasladar las Especificaciones Funcionales en términos que reflejen precisamente las Especificaciones Funcionales expresadas en términos que el analista de sistemas y programador puedan entender.*

Check No.	Actividad Ejecutada	Si	No	N/A
	<b><i>El documento de especificaciones técnicas...</i></b>			
1	Detalla que funciones van a ser ejecutadas?			
2	Detalla que requerimientos están siendo satisfechos con cada función?			
3	Describe quien o que inicia/gatilla cada función y que datos son necesarios para empezar estos?			
4	Describe que pasos son necesarios para ejecutar cada función?			
5	Define que datos son pasados a otras funciones/transacciones?			
6	Define que salidas resultan de cada proceso/función (archivos, bases de datos, pantallas, reportes, etc.)?			
7	Define volúmenes de datos por función?			
8	Considera tiempo de respuesta y desenvolvimiento?			
9	Identifica restricciones de datos a/desde fuentes externas?			
10	Define protocolos de comunicaciones?			
11	Define formatos de datos y posibles conversiones de datos?			
12	Describe restricciones de acceso de usuario?			
13	Define número máximo de usuarios concurrentes?			
14	Considera actualización simultánea de datos?			
15	Describe y considera datos sensitivos y procesos que requieren consideraciones de seguridad?			
16	Define requerimientos de auditoria?			
17	Define niveles de seguridad?			
18	Define especificaciones de recuperación?			
19	Explica reconciliación con Sistemas existentes?			
20	Especifica hardware, sistema y software de comunicación?			
21	Contiene estructura lógica de datos en la forma de modelos de datos detallados?			
22	Define funciones del sistema (En-linea vs. Batch)?			
23	Describe donde/como los archivos de datos y/o codigo del sistema puede ser reusado?			

## LISTA DE CHEQUEO - ESPECIFICACIONES TECNICAS (Cont.)

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>El documento de especificaciones técnicas...</i>			
24	Describe cualquier producto o herramientas de terceros a ser usado, incluyendo cualquier necesidad de seguridad adicional, documentación de instalación, y licencias?			
25	Detalla diseño de base de datos y archivos? El diseño coincide con las especificaciones funcionales?			
26	Identifica programas (En-línea y Batch) a ser usados?			
27	Define especificaciones de reinicio/recuperación ? Los requerimientos coinciden?			
28	Considera/resuelve requerimientos legales/regulatorios/política interna?			
29	Considera requerimientos de seguridad (Datos, comunicaciones, software, perfiles de usuario)?			
30	Considera requerimientos de contingencia?			
31	Define procedimientos de reconciliación con otros sistemas?			
32	Considera temas de conversión de datos?			
33	Contiene un diagrama de la red?			



# LISTA DE CHEQUEO - INSPECCIÓN DE CODIGO

Nombre de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>Inspección de Código ...</i>			
1	El equipo de inspección ha sido identificado?			
2	Los materiales de inspección han sido identificados y presentados al equipo de inspección?			
3	Las reuniones de inspección han sido concertadas/llevadas a cabo ?			
4	El código esta suficientemente comentado?			
5	Los defectos han sido registrados por tipo (diseño, sintaxis, código de retorno, comentarios, logica, estandards, requerimientos, mensajes, etc.)?			
6	Los defectos (Si existen) han sido considerados/corregidos?			
7	Una re-inspección ha sido concertada/llevada a cabo?			

# LISTA DE CHEQUEO - PLAN DE PRUEBAS UNITARIAS

Nombre de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

**Propósito:**

*Los dos objetivos de las pruebas unitarias - remover defectos y demostración de desenvolvimiento.*

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>El Plan de Pruebas Unitarias...</i>			
1	Provee una descripción detallada por cada unidad?			
2	Contiene las opciones de prueba por cada unidad?			
3	Provee datos de prueba?			
4	Considera pruebas de interfases?			
5	-Número de parámetros de entrada = número de argumentos?			
6	- Coinciden os atributos de argumentos y parámetros?			
7	-Las definiciones de variables globales coinciden a través de los módulos?			
8	-Atributos de archivo son correctos?			
9	-Instrucciones OPEN/CLOSE correctos?			
10	-Archivos abiertos antes de uso?			
11	-Especificaciones de formato coincide con las instrucciones de Entrada/Salida?			
12	-El tamaño del "Buffer" coincide con la longitud del registro?			
13	-Se manejan condiciones de Fin de archivo?			
14	-Manejo de errores de Entrada/Salida?			
15	-Errores textuales en salidas?			
16	Considera pruebas de borde:			
17	-Chequeo elementos en loops, arreglos?			
18	Considera Integridad de datos:			
19	-Cálculos son correctos?			
20	-Tipos de datos similares son comparados?			
21	Considera manejo de errores:			
22	-Mensajes de error ilegibles?			
23	-Procesamiento de condiciones de excepciones es correcta?			
24	-Error notificado corresponde a error encontrado?			
25	-La descripción de error contiene suficiente información para búsqueda?			
26	-El manejo del error ocurre antes de la intervención del sistema?			

# LISTA DE CHEQUEO - PLAN DE PRUEBAS DE SISTEMAS

Nombre de Proyecto:	Fecha Inicio proyecto:
Aplicación/Sistema:	Ambiente:
Gerente de proyecto:	Teléfono:
Contacto Alternante:	Teléfono:
Número de LASER:	Fecha de Revisión
Nombre de Revisor:	Firma:

**Propósito:**

*La prueba de sistemas asegura que los diversos componentes del sistema puedan ser ejecutados como un sistema integrado.*

Check No.	Actividad Ejecutada	Si	No	N/A
	<i>El plan de pruebas de Sistemas/Integración...</i>			
1	Provee una descripción detallada de los objetivos del plan?			
2	Provee las opciones de la prueba?			
3	Provee datos de prueba?			
4	Describe el ambiente de la prueba?			
5	Contiene procedimientos detallados incluyendo..			
6	-Propósito?			
7	-Orden de los módulos a ser probados?			
8	Cada modulo es chequeado para ...:			
9	-Integridad de Interfase?			
10	-Validez Funcional?			
11	-Contenido de la Información?			
12	-Desenvolvimiento?			
13	Incluye pruebas de recuperación?			
14	Incluye pruebas de seguridad?			
15	Incluye pruebas de situaciones límites?			
16	Incluye prueba de desenvolvimiento?			
17	Contiene datos de pruebas?			
18	Describe los resultados esperados?			
19	Incluye roles y responsabilidades?			

## **5) Guía del Plan de Evaluación de Calidad de Software**

La guía del Plan de Evaluación de Calidad de Software o SQAP (Software Quality Assurance Plan), fue desarrollado para servir como herramienta de consulta en la confección de los Planes de Evaluación de Calidad de Software para los distintos departamentos y áreas de la organización.

El documento Plan de Evaluación de calidad de Software, presenta el siguiente contenido :

### **Acerca de este Documento**

#### **Identificación e Historia de Cambios**

Sumario

Política

## **II DEFINICION**

1.0 Propósito

2.0 Documentación Generada

2.1 Objetivo

2.2 Documentos requeridos de Ingeniería de Software

2.3 Documentos adicionales generados

2.4 Referencia de Documentos externos

3.0 Responsabilidades y Roles de la Gerencia

### 3.1 Organización

### 3.2 Tareas

### 3.3. Responsabilidades

### 4.0 Políticas, Estándares, prácticas y métricas

### 4.1 Propósito

### 4.2 Políticas, Estándares y Prácticas

### 4.3 Métricas

### 5.0 Consideraciones de Evaluación de Calidad de Software (Revisiones y Auditorias)

### 5.1 Objetivo

### 5.2 Requerimientos de Revisión

### 5.3 Pruebas

### 6.0 Comunicación del Plan de Evaluación de Calidad de Software (SQAP)

### 7.0 Criterio de Aceptación y Liberación

### 7.1 Criterio de Aceptación

### 7.2 Criterio de liberación

### 8.0 Gestión de Configuración de Software

### 9.0 Herramientas, técnicas y metodologías

### 10.0 Gestión de Subcontratistas

### 11.0 Colección de Registros /Mantenimiento / Retención

12.0 Entrenamiento

13.0 Gestión de Riesgo

14.0 Glosario de Términos

15.0 Apéndice

## **6) Lista de Chequeo del Ciclo de Vida del Desarrollo de Software**

Para facilitar las tareas en el area de “SQA”, en cuanto a la revisión de los diversos requerimientos de documentación, y cumplimiento de las tareas en cada una de las etapas del Ciclo de Vida de Desarrollo de Software, se creo la “Lista de Chequeo del Ciclo de Vida de Desarrollo de Software”. Este documento debe ser llenado por el Gerente del Proyecto de Software, y cada una de las etapas contiene una línea, a ser utilizada por el “SQA”, para ser completada, una vez se determine que la etapa en cuestión esta conforme.

FIGURA 9

## LISTA DE CHEQUEO DEL CICLO DE VIDA DE DESARROLLO DE SOFTWARE

Fase:		INICIACION (4.1)	Completo			Justificación por respuestas NO o No aplicable			
C/P	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	A	Acción Correctiva / Control Compensatorio	F e c h a O b j e t o	Ri e s g o	Res p o n s a b l e
1	M&MS 4.1.1	<b>Creación de RS</b> a) El Requerimiento de Servicio (RS) ha sido <b>formalizado</b> en LASER o en un formato controlado ?				<i>Impresión original del Láser en folder</i>			
2	M&MS 4.1.1 (4) CMMRM (Ac 1, 2, 3)	b) Si el RS es una mejora, ha sido aprobado por el Gerente Regional del producto?				<i>Láser o E.mail conteniendo la aprobación del GRP</i>			
3	M&MS 4.1.2, 5.1.1.1- 5.1.1.4 CMMRM (Ac 1, 2, 3)	<b>Revisión RS</b> a) El RS contiene toda la información obligatoria?				<i>Especificación funcional completada Impacto Operacional completado Criterio de Aceptación completado</i>			



Fase:		<b>INICIACION (4.1) (Cont.)</b>	Completo			Justificación por respuestas NO o No aplicable			
C/P	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	A	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Responsable
4	<i>M&amp;MS</i> 4.1.2 (2)	<p>b) Si el RS esta correcto, el Enlace técnico (ET) envió una fecha para conclusión del plan de proyecto (Fecha de evaluación de tamaño) al Representante del Negocio (RN) en los cinco días laborables (A través de E.mail o LASER)?</p> <p>Si no, (Incompleto o no claro), el RS fue retornado al RN con comentarios (a través de Email o Láser)?</p>				<i>Láser o Email en Folder</i>			
5	<i>CMMRM</i> (Ab 2) <i>CMMSCM</i> (Ac 2; 3)	<p><b>Línea base de Requerimientos</b></p> <p>Al RS, definido arriba, se ha aplicado "línea base" en el Folder del Proyecto y registrado en el Log de Control de Cambios del Proyecto (LCCP) ?</p>				<i>Todas las entregas han sido ingresadas en el LCCP</i>			
6	<i>CMMSQA</i> (Ac 2, 4, 5)	<p><b>Evaluación de Calidad</b></p> <p>La Gerencia del grupo de desarrollo se aseguro que las actividades del CCFD fueron ejecutadas?</p>				<i>El ET verifica que las actividades fueron hechas correctamente. Se requiere nombre e inicial para este punto de verificación.</i>			

G/P	Fase:	DEFINICION (4.2)	Completo			Justificaciones para respuestas NO o NA			
	Referencia	Actividad Ejecutada / Requerimiento del proceso	S	N	A	Acción Correctiva / Control Compensatorio	F e c h a O b j e t.	Rie sgo	Re sp.
1	M&MS 4.2.1, 5.2.1 CMMSCM (Ac 5, 6)	<b>Análisis de Impacto</b> a) Existe un documento “Análisis de Impacto” para el RS cubriendo todas las áreas definidas?				Llenado del formato de Análisis de Impacto			
2	M&MS 4.2.1, 5.2.2	b) Existe un “Plan de prueba unitario” (PPU) documentado para el RS?				PPT para todos los cambios - Plan requerido			
3	M&MS 4.2.1, 5.2.3	c) Existe un “Plan de Prueba de Sistema” documentado para el RS ?				Riesgo técnico alto - Plan requerido			
4	M&MS 4.2.1, 5.2.4	d) Existe un “Plan de Retorno” para RS con un nivel de Alto riesgo de negocio ?				Plan de Retorno no requerido si el Impacto Operacional/Negocio es de nivel alto			
5	M&MS 4.2.2 CMMSP (Ac 1, 3, 4, 5, 7, 8, 13)	<b>Planeamiento del Proyecto</b> a) Existe un cronograma de Plan de proyecto para el RS?				Herramienta de Cronograma de Proyecto			

C/P	Fase: Referencia	DEFINICION (4.2) (Cont.) Actividad Ejecutada / Requerimiento del proceso	Completo			Justificaciones para respuestas NO o NA			
			S	N	A	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
6	M&MS 4.2.2 (1), 5.2.5	b) El cronograma del plan de proyecto contiene esfuerzo planeado y recursos asignados por cada actividad ?				Herramienta de Cronograma de Proyecto			
7	M&MS 4.2.2 (3)	c) La fecha de Prueba de Aceptación del Usuario (PAU) fue entregada al RN ?				Láser o E.mail en el folder del Proyecto			
8	M&MS 4.2.3	<b>Obtener Aprobación del Proyecto</b> El ET recibió aprobación del proyecto, y la fecha de entrega del PAU ?				Láser or E.mail en el folder del proyecto			
9	CMMSCM (Ac 2)	<b>Gestión de Configuración</b>  Al Cronograma del Proyecto se le ha aplicado "línea base" en el folder del proyecto, y registrado en el LCCP ?				Impresión del cronograma y Log de Control de cambios del proyecto (LCCP)			

C/P	Fase:	DEFINICION (4.2) (Cont.)	Completo			Justificaciones para respuestas NO o NA			
	Referencia	Actividad Ejecutada / Requerimiento del proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
10	CMM SQA (Ac 2, 4, 5) CMM SCM (Ac 2, 3)	<b>Evaluación de Calidad</b> La gerencia del grupo de desarrollo ha verificado las entregas para esta fase? a) Ha sido preparada de acuerdo a los procedimientos aprobados? b) Se aplico "línea base" en el folder del proyecto, y registrado en el LCCP ? (Ver la sección de archivo de requerimientos del proyecto en las siguientes paginas)				<i>El ET revisa que todas las entregas (y evidencias) para esta fase están en el folder del proyecto y que las actividades fueron hechas de acuerdo a procedimientos establecidos.</i> <i>El nombre e inicial del ET es requerido para este punto de chequeo. Verificado por el QA interno.</i>			

	<b>Fase:</b>	<b>CONSTRUCCION (4.3)</b>	<b>Completo</b>			<b>Justificación por respuestas N o NA</b>			
<b>C/P</b>	<b>Referencia</b>	<b>Actividad Ejecutada / Requerimiento de Proceso</b>	<b>S</b>	<b>N</b>	<b>NA</b>	<b>Acción Correctiva / Control Compensatorio</b>	<b>Fecha Objeto</b>	<b>Riesgo</b>	<b>Resp.</b>
<b>1</b>	<i>M&amp;MS</i> 4.3.1	<b>Especificaciones Técnicas detalladas</b> a) El ET ha revisado el Análisis de Impacto y Planes de Prueba para determinar los cambios a los componentes del sistema ?				<i>Formato de Análisis de Impacto y lista de componentes (Temas técnicos); Especificaciones</i>			
<b>2</b>	<i>M&amp;MS</i> 4.3.1, 5.3.1	b) Se han aprobado las especificaciones técnicas (Tech Spec) ?				<i>Especificaciones de programas en Folder del proyecto. El ET debe aprobar las especificaciones por el programador.</i>			
<b>3</b>	<i>M&amp;MS</i> 4.3.2, 4.3.6 (1)	<b>Codificación</b> a) Se ha ejecutado la inspección del código para asegurarse que los cambios al código fuente reflejan las especificaciones de programa (Cambios de código coinciden con la especificación)?				<i>No alto riesgo técnico - Comparador en el folder del proyecto o referencia de la localización del comparador en un medio electrónico. Alto riesgo técnico - Inspecciones de código, evidencia de elementos revisados y comparadores de código.</i>			
<b>4</b>	<i>M&amp;MS</i> 2.2.3	b) El nuevo código ha sido revisado por un individuo distinto a la persona que lo escribió ?				<i>Evidencia de revisión del código.</i>			

Fase:		CONSTRUCCION (4.3 (Cont.))	Completo			Justificación por respuestas N o NA			
C/P	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
5	M&MS 2.2.3	<b>Control de Código Fuente</b> a) El control de código fuente de producción reside fuera del área de desarrollo ?				Impresión del Email de control.			
6	M&MS 4.3.6 (2), 2.2.3	b) Los cambios al código fuente son hechos a través del Procedimiento de control de cambios por el ambiente del sistema?				Chequeo Independiente (Email indicando que los programas han sido catalogados)			
7	M&MS 2.2.3	c) El nuevo código fuente esta conforme con los "estándares de programación y/o convención de nombres"?				Cambios hechos por Los Estándares de codificación.			
8	M&MS 4.3.3	<b>Prueba Unitaria</b> a) La prueba unitaria fue ejecutada para todos los cambios de acuerdo al Plan de Pruebas Unitario (PPU)?				Evidencia de resultados de las pruebas unitarias			
9	SM&E 2.2.5, 5.3.3	b) Los resultados de la prueba unitaria fueron documentados?				Evidencia de resultados de las pruebas unitarias.			

<b>Fase:</b>		<b>CONSTRUCCION (4.3 (Cont.))</b>	<b>Completo</b>			<b>Justificación por respuestas N o NA</b>			
<b>C/P</b>	<b>Referencia</b>	<b>Actividad Ejecutada / Requerimiento de Proceso</b>	<b>S</b>	<b>N</b>	<b>NA</b>	<b>Acción Correctiva / Control Compensatorio</b>	<b>F e c h a O b j e t .</b>	<b>Rie s g o</b>	<b>Re s p.</b>
10		c) Las discrepancias de la Prueba unitaria fueron registradas en el Log de defectos ?				<i>Log de Defectos.</i>			
11		d) Las discrepancias de la prueba unitaria fueron resueltas (PPU o código fuente corregido y re-probado)?				<i>Evidencia de Resultados de pruebas unitarias de regresión.</i>			
12	<i>SM&amp;E 4.3.4</i>	<b>Prueba de Sistemas</b> a) La prueba de Sistema fue ejecutada para todos los cambios según el plan de prueba de sistema (PPS)?				<i>Evidencia de los resultados de la prueba de sistema para proyectos de alto riesgo técnico.</i>			
13	<i>MM&amp;S 5.3.5</i>	b) Los resultados de la Prueba de Sistemas fueron documentados?				<i>Evidencia de Resultados de la prueba de Sistemas, para proyectos de alto riesgo técnico.</i>			
14		c) Las discrepancias de la Prueba de Sistema fueron registradas en el Log de Defectos ?				<i>Log de Defectos.</i>			

	<i>Fase:</i>	<b>CONSTRUCCION (4.3)(Cont.)</b>	<b>Completo</b>			<b>Justificación por respuestas N o NA</b>			
<b>C/P</b>	<b>Referencia</b>	<b>Actividad Ejecutada / Requerimiento de Proceso</b>	<b>S</b>	<b>N</b>	<b>NA</b>	<b>Acción Correctiva / Control Compensatorio</b>	<b>Fecha Objeto</b>	<b>Riesgo</b>	<b>Resp.</b>
<b>15</b>		d) Las discrepancias de la prueba de Sistemas fueron resueltas (PPS o código fuente corregido y reprobado)?				<i>Evidencia de resultados de las pruebas de regresión del sistema.</i>			
<b>16</b>	<i>M&amp;MS</i> 4.3.5	<b>Actualización de Documentación Técnica</b> a) Los documentos son actualizados para todos los sistemas afectados, Ejemplo : Especificaciones de Programa, y otros, según sea aplicable?				<i>Responsabilidad del ET/Revisiones internas de SQA</i>			



CP	Fase:	CONSTRUCCION (4.3)(Cont.)	Completo			Justificación por respuestas N o NA			
	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
17	M&MS 4.3.6	<p><b>Autorización de Transferencia de software a PAU (Prueba de Aceptación de Usuario)</b></p> <p>a) La transferencia de Software fue debidamente autorizada de acuerdo al Procedimiento de Control de Cambios de Sistemas/Aplicaciones?</p> <p>b) Estas entregas asociadas fueron revisadas por el ET antes de que el Software sea transferido?</p>				<p><i>Email evidenciando la transferencia al ambiente de QA, del encargado del Control de cambios.</i></p> <p><i>El ET chequea las siguientes entregas: Código actualizado, resultados de las pruebas unitarias y resultados de las pruebas de sistema.</i></p>			
18	M&MS 4.3.7	<p><b>Actualización de la Documentación de Usuario</b></p> <p>Todos los documentos relacionados con el usuario del sistema han sido actualizados?</p>				<i>ET/Responsabilidad de Usuario</i>			
19	CMMSCM (Ac 2)	<p><b>Gestión de Configuración</b></p> <p>El grupo se adhiere a su Plan de Gestión de Configuración de Software(PGCS) ?</p>				<i>LCCP en el Folder</i>			

		<b>Fase:</b>	<b>CONSTRUCCION (4.3)(Cont.)</b>			<b>Justificación por respuestas N o NA</b>			
<b>C/P</b>	<b>Referencia</b>	<b>Actividad Ejecutada / Requerimiento de Proceso</b>	<b>S</b>	<b>N</b>	<b>NA</b>	<b>Acción Correctiva / Control Compensatorio</b>	<b>F e c h a O b j e t.</b>	<b>Rie sgo o</b>	<b>Re sp.</b>
<b>20</b>	<i>CMM SQA (Ac 2,4, 5)</i>	<b>Evaluación de Calidad</b> La Gerencia del Grupo de Desarrollo ha verificado que las actividades y entregas para esta fase : a) han sido preparadas de acuerdo con procedimientos aprobados? b) Están en el folder del proyecto? c) Han sido revisadas?				<i>El ET revisa que todas las entregas y evidencias para esta fase están en el folder del proyecto, y que las actividades fueron hechas de acuerdo a procedimientos establecidos. Nombre e inicial del ET se requiere n este punto de chequeo.Revision interna de QA hecha.</i>			

C/P	Fase:	VERIFICACION (4.4)	Completo			Justificación para respuestas N o NA			
	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
1	M&MS 4.4.1, 5.4.1	<b>Preparar el Plan de Prueba de Aceptación de Usuario (PPAU)</b> a) El Usuario ha definido el alcance del PPAU y los resultados esperados, y el plan ha sido aprobado?				<i>Responsabilidad Usuario/Representante de Negocio Pais (RNP) o Representante de Negocio Regional (RNR), basado en nivel de riesgo : Bajo Riesgo : RNP Alto Riesgo : RNR</i>			
2	CMM SQA (Ac 2, 4, 5)	b) A una copia del PPAU se le ha aplicado "línea base" en el Folder del Proyecto, y registrada en el LCCP?				<i>Responsabilidad Usuario/Representante de Negocio Pais (RNP) o Representante de Negocio Regional (RNR), basado en nivel de riesgo : Bajo Riesgo : RNP Alto Riesgo : RNR</i>			
3	M&MS 4.4.2	<b>Ejecución de Prueba de Aceptación de Usuario</b> a) El PPAU fue ejecutado por el usuario y los resultados fueron documentados?				<i>Responsabilidad Usuario/Representante de Negocio Pais (RNP)</i>			
4	M&MS 4.4.2 (2)	b) Cualquiera de los temas encontrados en el PPAU (Discrepancias de la prueba), fueron documentados por el Usuario y el RN y enviados al ET?				<i>Formulario de Retorno de Pruebas de Aceptacion de Usuarios o Email</i>			
5	M&MS 4.4.2 (3)	c) Las discrepancias de la prueba fueron registradas en el Log de defectos?				<i>Log de Defectos</i>			

C/P	Fase:	VERIFICACION (4.4) (Cont.)	Completo			Justificación para respuestas N o NA			
	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	F e c h a O b j e t.	Rie sgo	Re sp.
6	M&MS 4.4.2	d) Todos los temas del PPAU han sido resueltos?				EM del usuario indicando que el Software esta Correcto			
7	M&MS 4.4.2	e) Todos los temas del PPAU han sido reprobados?				Representante de negocio del país/Responsabilidad del Usuario			
8	CMMSQA (Ac 2, 4, 5)	<b>Gestión de Configuración</b>  A los resultados de las actividades aplicables se les ha aplicado "línea base" en el Folder del Proyecto y registradas en el LCCP?				Log de Control de cambios del proyecto.			
9	CMMSCM (Ac 2)	<b>Evaluación de Calidad</b> La gerencia del grupo de desarrollo ha verificado que las actividades y entregas para esta fase :  a) Han sido preparadas de acuerdo a procedimientos aprobados? b) Están en el folder del Proyecto? c) Han sido revisados?				El ET revisa que todas las entregas y evidencias para esta fase están en el folder del proyecto, y que las actividades fueron hechas de acuerdo a procedimientos establecidos. Nombre e inicial del ET se requiere n este punto de chequeo.Revision interna de QA hecha.			

C/P	Fase:	IMPLEMENTACIÓN (4.5)	Completo			Justificación para respuestas N o NA			
	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
1	M&MS 4.5.1	<b>Instalación del Ambiente de Producción</b> a) Todas las discrepancias fueron resueltas (Plan de pruebas o código fuente corregido y reprobado)?				<i>Láser y Aprobación del Usuario del paquete a ser promovido a producción/Responsabilidad del Gerente de implementación.</i>			
2	M&MS 4.5.1	b) El Usuario autorizo la transferencia del Software al Ambiente de Producción de acuerdo al Procedimiento de Control de Cambios del sistema ?				<i>Aprobación del Usuario del paquete a ser promovido a producción. Responsabilidad del Gerente de Implementación.</i>			
3	CMM SCM (Ac 2)	<b>Gestión de Configuración</b> a) Los resultados de las actividades aplicables arriba han sido puestas en el folder del proyecto, de acuerdo al LCCP?				<i>Log del Control de cambios del proyecto.</i>			
4	M&MS 2.1.6	b) Todos los Items del Folder del Proyecto han sido preparados para ser almacenados?				<i>Gavetero de Archivo.</i>			

C/P	Fase:	IMPLEMENTACIÓN (4.5) (Cont.)	Completo			Justificación para respuestas N o NA			
	Referencia	Actividad Ejecutada / Requerimiento de Proceso	S	N	NA	Acción Correctiva / Control Compensatorio	Fecha Objeto	Riesgo	Resp.
5	CMMISQA (Ac 2, 4, 5)	<b>Evaluación de Calidad</b> La gerencia del Grupo de Desarrollo ha verificado que las entregas y actividades para esta fase : a) Han sido preparadas de acuerdo a procedimientos aprobados? b) Están en el Folder del Proyecto? c) Han sido revisadas?				<i>El ET revisa que todas las entregas y evidencias para esta fase están en el folder del proyecto, y que las actividades fueron hechas de acuerdo a procedimientos establecidos.            Nombre e inicial del ET se requiere en este punto de chequeo.</i>			

C/P	Referencia	Actividad Ejecutada / Requerimiento de Proceso	Completo			Todas estas preguntas deben ser respondidas despues que el proyecto ha terminado. Justificación para respuestas N o NA			
			S	N	NA	Acción Correctiva / Control Compensatorio	F e c h a O b j e t o	Rie sgo	Re sp.
1	M&MS CMMPTO (Ac 1)	<b>Revisión de Gerencia</b>  La gerencia ha revisado los planes (cronogramas) y actividades?				Aplicable según el proyecto			
2	M&MS (2.1.3) CMMPTO (Ac 1,2)	<b>Seguimiento del Proyecto</b>  a) Se ha reportado al Gerente del Proyecto el esfuerzo real gastado por el proyecto ?				deltas			
3	M&MS 2.1.3	b) Todos los cambios a los compromisos previos han sido documentados y todas las partes están de acuerdo?				EM			
4	M&MS 2.1.3	c) Los estados del requerimiento y todos sus cambios están documentados?				Las referencias se imprimirán en un Láser final.			

<b>Proceso:</b>		<b>PROCESO DE GESTIÓN</b>	<b>Completo</b>			<i>Todas estas preguntas deben ser respondidas despues que el proyecto ha terminado.</i>			
						<b>Justificación para respuestas N o NA</b>			
<b>C/P</b>	<b>Referencia</b>	<b>Actividad Ejecutada / Requerimiento de Proceso</b>	<b>S</b>	<b>N</b>	<b>NA</b>	<b>Acción Correctiva / Control Compensatorio</b>	<b>Fecha Objetivo</b>	<b>Riesgo</b>	<b>Resp.</b>
5	M&MS 2.1.3	d) Si una desviación significativa en el Cronograma del Plan del Proyecto (Resultados actuales) ha sido detectada, se ha identificado el problema y tomado acción correctiva?				Aplicable según el proyecto.  "Significante" será definido en el Plan de evaluación de calidad.			
6	M&MS 2.1.4	e) Se han recolectado el conjunto de métricas básicas y se han entregado a SQA?				Métricas mensuales			
7	M&MS 2.1.6	<b>Archivo del proyecto</b> a) Existe un folder del proyecto para este requerimiento?							
8	M&MS 2.1.6	b) El folder para el proyecto completo contiene evidencias / referencias de todos las entregas y documentación relacionada?							
9	M&MS 2.1.6	c) El folder del proyecto ha sido revisado si esta completo por el Gerente del Proyecto.?							



## 7) Guía de Plan de Gestión de Configuración de Software

La Guía del Plan de Gestión de Configuración de Software, permitirá el desarrollo de planes específicos en cada una de las áreas de la organización.

A continuación el contenido del plan

### Acerca de este Documento

Se explica en que consiste el documento, por quienes fue desarrollado, el objetivo y alcances del mismo.

### Historia de Cambios

Detalla la fecha del cambio el número de versión, por quien fue cambiada y la descripción del cambio.

Indica también por quien deberá ser revisado y aprobado el cambio. Cada cambio deberá producir un número de versión.

### HISTORIA DE CAMBIOS

Fecha	Versión	Por	Descripción

### Aprobaciones

Firma y sello de las personas que aprueban el documento.

El gerente del departamento, y el Coordinador del Proyecto CMM.

## **1. Sumario de Políticas**

### **1.1 Racionalidad**

Descripción de la política de la organización, que menciona el requerimiento de un plan de Gestión de Configuración de software.

### **1.2 Alcance**

Alcance de la aplicación del plan.

### **1.3 Política**

El objetivo de este plan es de asegurarse que las siguientes políticas son seguidas:

- Los Items de Configuración deben ser identificados. El responsable por la identificación de los Items de Configuración es el “CCB” (Change Control Board).
- Los Items de Configuración deben ser controlados. Para cualquier proyecto que envuelve un Item de Configuración, se deben planear actividades de Gestión de Configuración de Software (GCS), responsabilidades para estas actividades explícitamente asignadas, e implementadas a lo largo del ciclo de vida del proyecto.
- El “Estado de Items de Configuración” o control de versiones debe ser mantenido y contiene suficiente detalle para seguir todos los cambios realizados al ítem de configuración durante el Ciclo de Vida de Desarrollo de Software.

El cumplimiento con la política deberá ser revisada y auditada en una base regular.

#### **1.4 Descripción del Documento**

Esta sección identifica el Plan de Gestión de Configuración de Software (PGCS) e identifica la guía general bajo la cual este será desarrollado y será usado. Las siguientes secciones describen el Plan en detalle. Este describe porque el GCS es ejecutado, como es ejecutado en el grupo o área , y como el proceso en si es controlado y cambiado según las condiciones.

#### **1.5 Rango de Control**

Explica el rango de aplicación y control del plan. Indicando el nombre del área o grupo de la organización.

#### **1.6 Convenciones**

- Nombre con que se llamara el documento
- Tipo de proyectos que se aplicara el plan.
- Formato del documento

#### **1.7 Cumplimiento de Políticas**

Describe como se va a supervisar el cumplimiento del plan.

#### **1.8 Reporte de problema**

Describe como se va a reportar los problemas presentados con la aplicación de los procedimientos presentados en este documento.

#### **1.9 Mantenimiento de Políticas**

Indica la frecuencia de revisión de la política.

## **2.0 Preparándose para el cambio**

### **2.1 Propósito y Alcance del Plan de Gestión de Configuración de Software**

Propósito y alcances del documento.

### **2.2 Definiciones y Acrónimos**

Lista de Abreviaciones y acronismos.

### **2.3 Referencias**

Documentos utilizados como referencias, para la confección del presente documento.

### **2.4 Aprobaciones**

Indica quien o quienes deben aprobar el documento

## **3. 0 Responsabilidades y Roles de la Organización**

### **3.1 Estructura de la Organización**

Descripción de la estructura de la Organización, donde se aplicara el plan.

## **4. 0 Ciclo de vida del desarrollo del Software y puntos de Gestión de Configuración de Software**

### **4.1 Adecuación de Ciclo de Vida regular y Gestión de Configuración de Software**

Describe los puntos de control en el ciclo de vida de un proyecto, en los cuales intrevienen las tareas de Gestión de Configuración de software

## **4.2 Relación con otros Cronogramas**

La Gestión de Configuración de Software es una parte integral de Ciclo de Vida de Desarrollo de Software. Esta afecta actividades como : Desarrollo de software, Pruebas de Sistemas, Pruebas de Aceptación de Usuario, e Implementación a Producción.

Se explica como afectaran a los Cronogramas del proyecto, las actividades de Gestión de Configuración de Software.

## **5.0 Gestión del Cambio**

### **5.1 Identificación de Items de Configuración por requerimiento**

Identifica los Items de Configuración a ser Controlados.

### **5.2 Items de Configuración adicionales**

Identifica los Items de Configuración, que son cambiados esporádicamente, pero que también requerirán de ser controlados. Tales como manuales, formatos, y procedimientos.

### **5.3 Metodología de Gestión de Cambio**

Detalla los procedimientos que permiten cumplir con la Gestión de Cambios de los Items de Configuración.

#### **5.3.1 Ambiente de Hardware**

#### **5.3.2 Ambiente de la Aplicación**

### **5.4 Reuniones de Control de Cambios**

Explica el procedimiento utilizado para llevar a cabo las reuniones de control de cambios.

## **6.0 Soporte de Cambios**

### **6.1 Identificación de recursos**

Indica los recursos de Software, Hardware, Conectividad así como las necesidades de entrenamiento al personal para soportar los cambios.

## **7.0 Reporte de Cambio**

### **7.1 Reporte de Estado**

Identifica los Items de Configuración entregadas en cada etapa del ciclo de vida del desarrollo, la herramienta utilizada para controlar el Item, y una explicación del proceso.

#### **7.1.1 Métricas de Gestión de Configuración de Software**

#### **7.1.2 Sistemas Automatizados**

#### **7.1.3 Seguimiento de Elementos de Datos**

## **8.0 Elementos de éxito de Gestión de Cambio**

Describe los procesos y elementos que aseguran el éxito en la Gestión del Cambio.

### **8.1 Respaldo y Recuperación**

### **8.2 Auditorias y Revisiones**

#### **8.2.1 Revisiones Internas**

#### **8.2.2 Revisiones Externas**

## **9.0 Tareas administrativas de Gestión de Configuración de Software**

### **9.1 Pertenencia**

## **10.0 Mejoramiento de Proceso**

## **11.0 Procedimientos e instrucciones de trabajo**

## **12.0 Apéndices**

### **8) Mantenimiento y mejoras de Software (M&MS) - Para proyectos de menos de 40 días de esfuerzo o menores a US\$ 50,000**

#### ***Descripción General***

Este documento es un manual con directivas a nivel de la organización, que contiene todos los documentos y procesos requeridos a lo largo del ciclo de vida del desarrollo de proyectos considerados menores, y catalogados así por la cantidad de esfuerzo o el costo que conlleva su implementación.

Este documento básicamente es un compendio lógico de todos los documentos explicados anteriormente, según la etapa de desarrollo.

## **9) Manual de desarrollo de Software (SDH Software development Handbook) - Para**

**proyectos de mas de 40 días de esfuerzo o mayores a US\$ 50,000**

### ***Descripción general***

Este manual de desarrollo es la versión del documento MM&S, pero para proyectos extensos .

En dichos proyectos, existen documentos adicionales que no son requeridos, tales como

- Permiso de Construcción
- Plan de Proyecto
- Estudio de Factibilidad
- Aprobación de Presupuesto
- Análisis de Costo/Beneficio

Desde que este informe tiene como finalidad la de proveer una idea general de la aplicación practica, la descripción de estos manuales a nivel detalle, no ofrecen mayor información para el objetivo del trabajo.



## **10) Software de Gestión de Proyectos (Workbench)**

### ***Descripción general***

La organización decidió la utilización de la aplicación PC Project Workbench, la cual permite la planificación de proyectos, y el seguimiento de los mismos a través de la generación de “views” standards a ser utilizados por los diferentes departamentos.

El procedimiento es el siguiente

#### **Planeamiento**

1) Se completan los tiempos estimados en horas y los recursos utilizados por cada una de las etapas del ciclo de vida de desarrollo.

2) Se adiciona el proyecto al Grupo de proyectos.

3) Se ejecuta la opción “Autoschedule”, la cual calcula las fecha de entrega en cada etapa basado en :

- Fecha de inicio del proyecto
- Esfuerzos en horas estimados por etapa
- Recursos por etapa y horas de trabajo diarias asignados al recurso
- Calendario de fines de semana y feriados

4) Se ejecuta la opción “Baseline” la cual sirve para “congelar” las fechas estimadas calculadas en 3.

5) Se informa las Fechas de entrega al Usuario.

#### Seguimiento

1) A través del software “Workbench”, se generara una hoja de control semanal por Recurso / Proyecto, con los tiempos estimados a ser completados por dicho recurso en dicha semana.

2) ) El recurso asignado a cada etapa del proyecto, deberá completar las horas reales utilizadas en el proyecto, en la hoja de control semanal.

3) Las horas reales trabajadas se ingresaran al sistema, permitiendo así obtener las desviaciones entre los tiempos estimados y reales.

4) Se deberá mantener reuniones periódicas y eventuales, con el grupo de desarrollo para efectuar el seguimiento de cada uno de los proyectos, y determinar las razones de los atrasos.

El software permite así relacionar diferentes proyectos, que utilizan los mismos recursos, y así controlar la carga de trabajo de cada recurso, generando gráficos con perfiles de disponibilidad de recurso.

## **11) Política de Contratación de Subcontratistas**

### ***Descripción General***

La Organización definió políticas de contratación, de los Subcontratistas para cumplir con el KPA SSM (Software Subcontract Management).

Dado que según este KPA, las Empresas Subcontratistas deben seguir los lineamientos del mismo nivel de CMM en la que se encuentra la empresa contratante, se determinó que por contrato, las empresas subcontratistas se comprometen bajo contrato a seguir los procesos, procedimientos y estándares determinados por la Organización.

Se definió un proceso de Contratación, el cual requiere de un concurso, entre Empresas que se presentan, con el fin de determinar la mas adecuada para el trabajo en cuestión.

## **CAPITULO IV - APLICACION PRÁCTICA DE CMM NIVEL 2**

### **4.1 Generalidades**

El departamento de Soporte de Sistemas HP, se encarga de dar mantenimiento y desarrollo de nuevas aplicaciones sobre un Sistema Bancario integrado, desarrollado en lenguaje “Business Basic”.

Originalmente dicho sistema se desarrollo en equipos Basic Four, luego con el soporte del desarrollo de un emulador del lenguaje, se pudo migrar a equipos Prime, los cuales tenían mayor capacidad. Finalmente se migro a equipos HP, siendo estos los que actualmente se están utilizando.

El sistema ha sido mantenido a lo largo de los años, por cambios relacionados a requerimientos de los países, mejoras de diseño y la creación de nuevas versiones del software (Releases), las cuales incluían mejoras y adiciones de nuevos módulos, según las estrategias del negocio, cambios en las políticas de la Corporación o mejoras de las tecnologías, entre otros.

Inicialmente dicho Sistema era mantenido por los países, los cuales tenían una versión del Software, y existía un equipo de analistas/programadores, que instalaba estas nuevas versiones creadas en un laboratorio de software, y daba soporte de estas, así como también incluía los cambios locales necesarios.

Posteriormente con la implementaron de la Regionalización, el Sistema paso a ser mantenido por una Compañía de Soporte de Software en Florida Estados Unidos, perteneciente a la Corporación.

La Corporación contrato personal con experiencia en el Sistema de los países, y por ello se pudo soportar relativamente bien el Sistema. Existían enlaces técnicos que se encargaban de grupos de países.

Sin embargo, con el transcurso del tiempo y las necesidades de los países , los requerimientos de cambios y mejoras se incrementaron, haciendo mas dificil el mantenimiento con este tipo de organización, y sobre todo sin procesos claros de desarrollo.

El cambio en la organización, el incremento de personal, y la creación de los procedimientos de procesos de software, para alcanzar el nivel 2, ayudo al departamento a mejorar el servicio con sus clientes, los países.

## **4.2 Organización**

La organización del departamento de soporte del sistema HP se reestructuró, con el fin de poder soportar el aumento de requerimientos, y adecuarlo al modelo CMM nivel 2.

Se subcontrato a una Compañía de Software, para desarrollar y mantener los requerimientos, así como para dar soporte de los procesos de producción.

Se crearon tres gerencias (Desarrollo, Producción e Implementación), las cuales coordinarían y supervisarían las actividades de los subcontratistas.

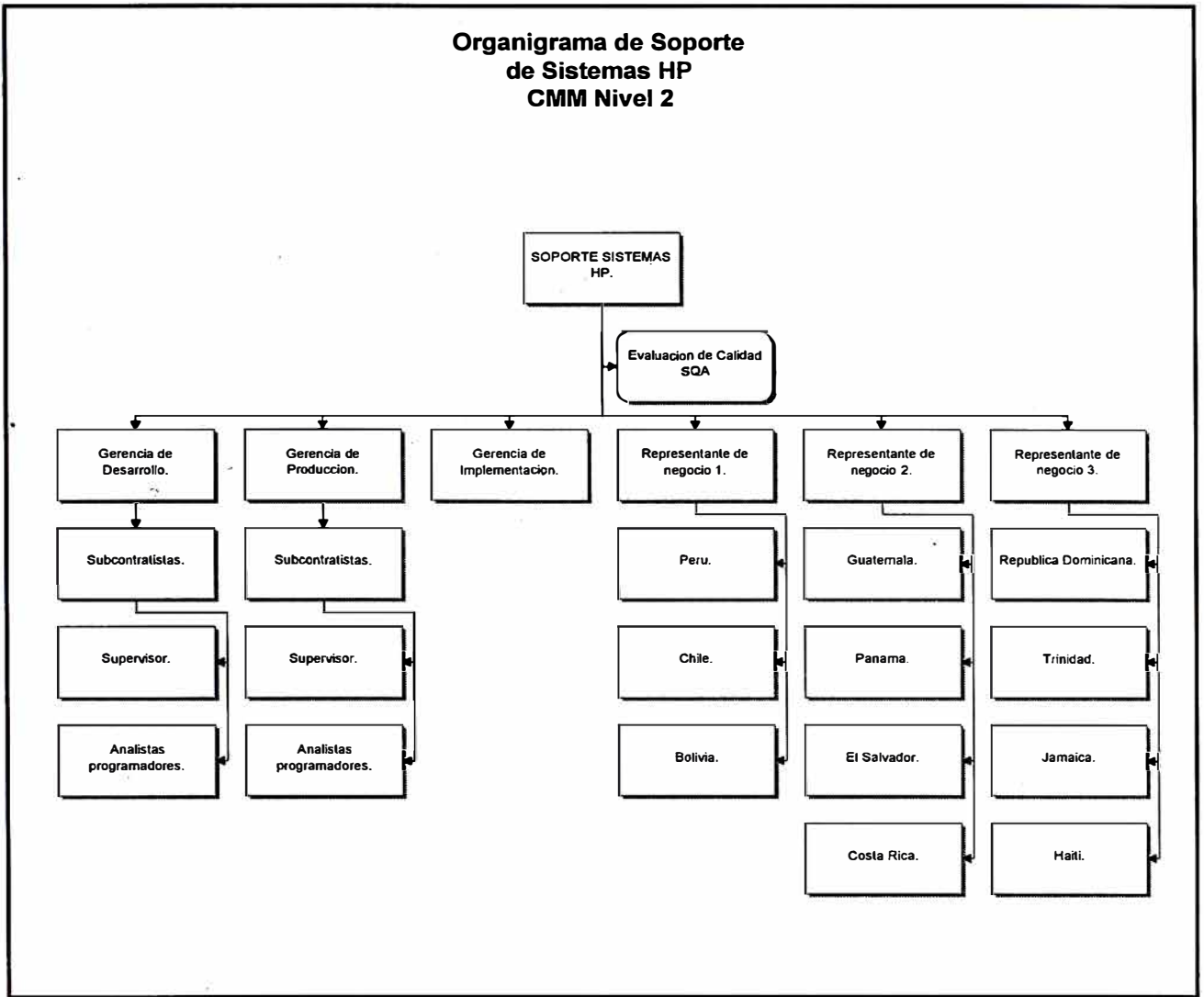
Se crearon las posiciones de “Representantes del negocio”, quienes servirían de enlace entre los usuarios de los países, y las áreas técnicas, traduciendo los requerimientos de los países en especificaciones funcionales, que permitirían el análisis de impacto técnico de dichos requerimientos.

El área de producción, se encargaría de resolver los problemas y errores presentados durante la ejecución de los programas en producción. Dado que dichos procesos son críticos, el personal subcontratado, debía mantener un servicio de 24 horas, incluyendo turnos nocturnos, para los procesos de fin de día, y posiciones en las oficinas de la Compañía.

El área de desarrollo, se encargaría de todo tipo de requerimiento o mejora en el sistema, y estaría localizada en oficinas fuera de la compañía.

El área de implementación, permitiría mantener la formalización de todo desarrollo culminado, en el proceso de transferencia a producción.

**FIGURA 10**



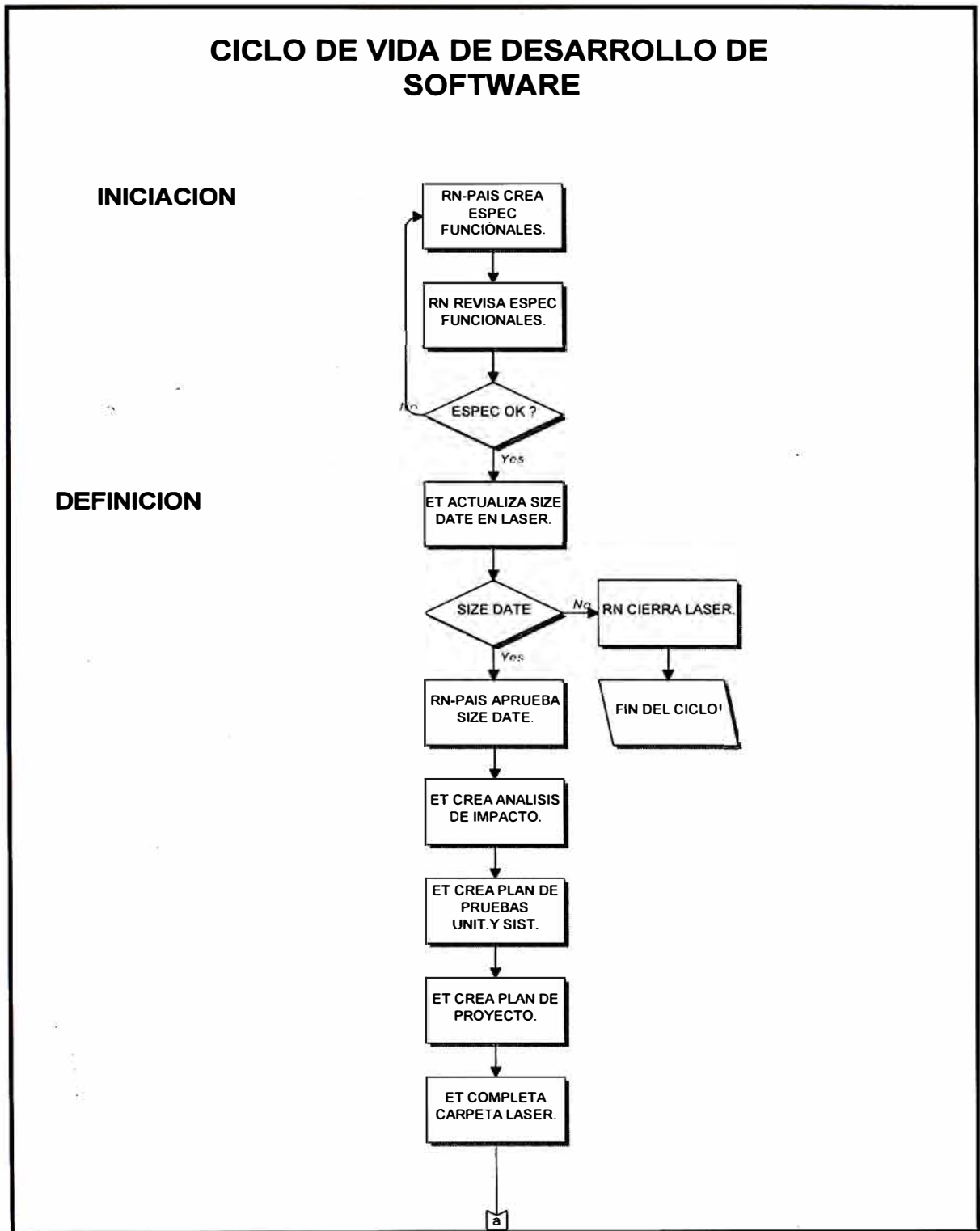
### **4.3 Descripción de Procesos - Ciclo de vida de desarrollo**

El ciclo de vida de desarrollo de software, impuesto por la organización, para cualquier desarrollo debió ser interpretado por el Área de soporte de Sistemas HP, y adecuado a las características propias de los requerimientos, y a los recursos disponibles.

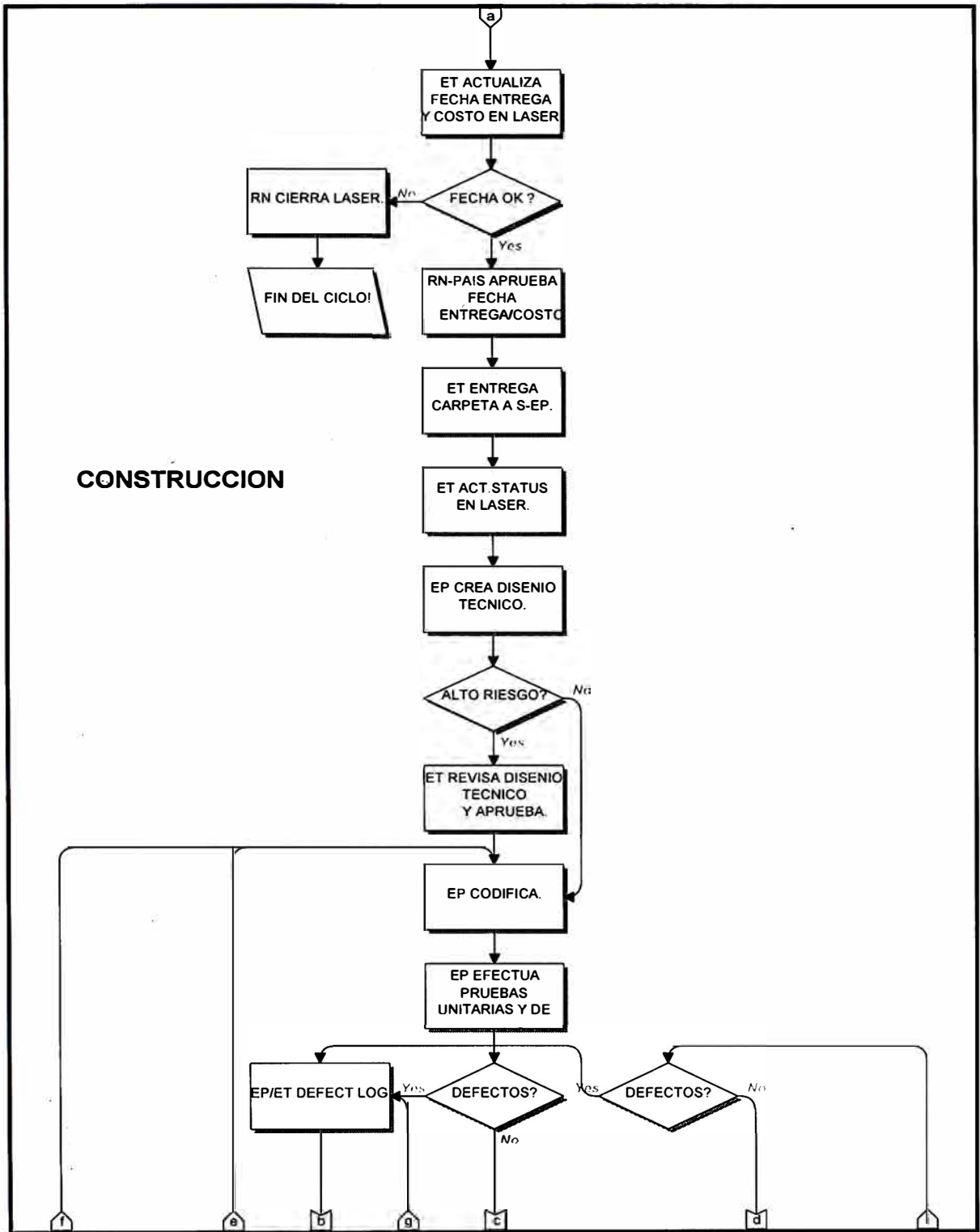
El diagrama a continuación esquematiza la interpretación del ciclo de vida de desarrollo de software, en el departamento de soporte de sistemas HP.

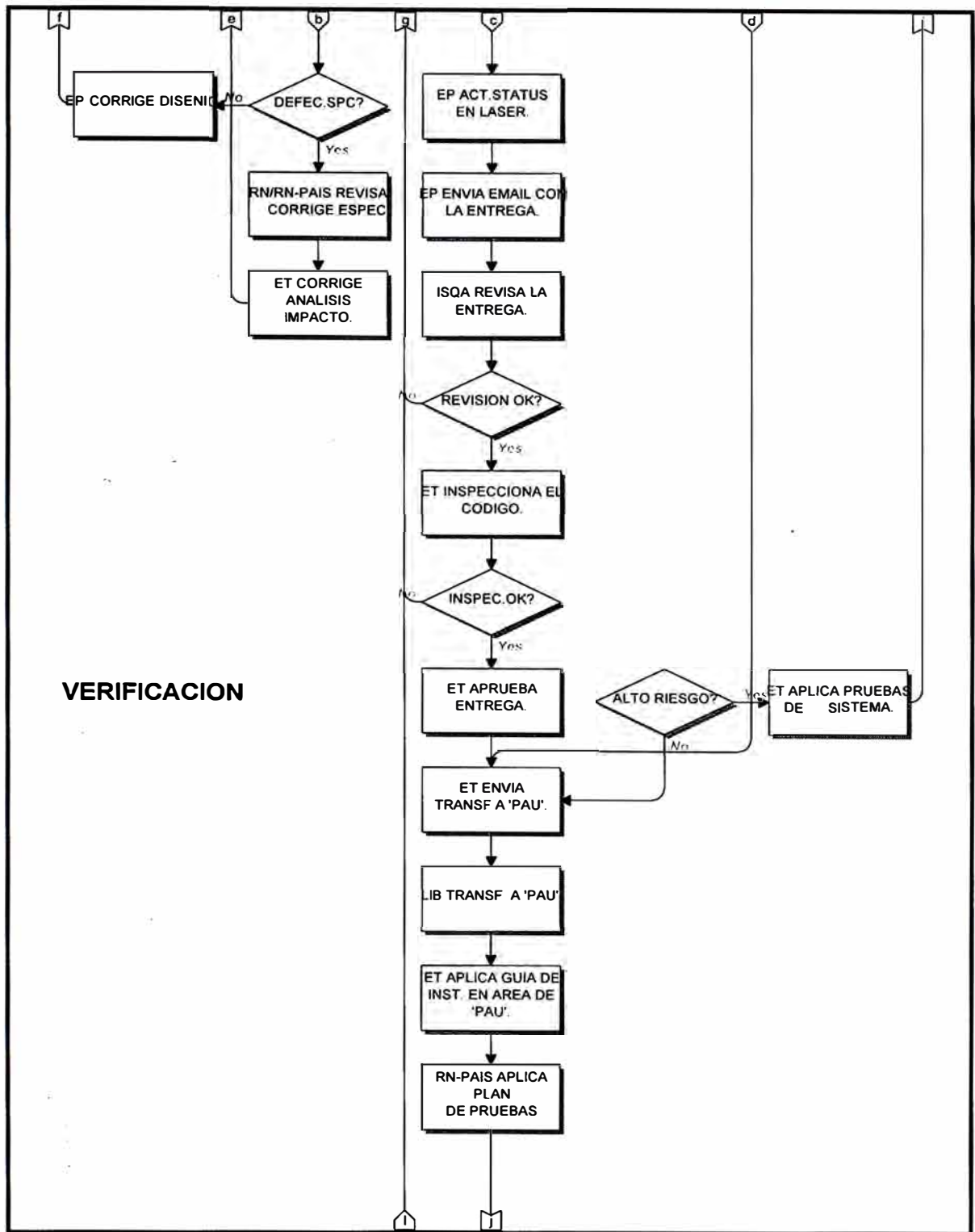


FIGURA 11

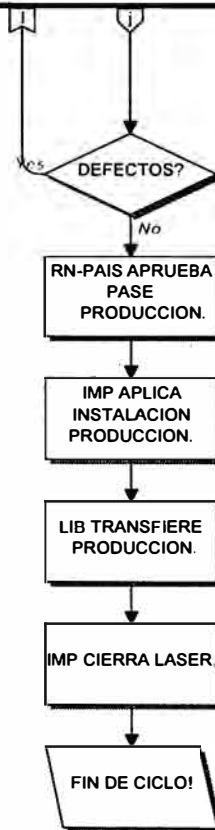


**CONSTRUCCION**





## IMPLEMENTACION



### Funciones involucradas en ciclo de vida de desarrollo de software

RN-PAIS: REPRESENTANTE NEGOCIO PAIS  
RN: REPRESENTANTE DE NEGOCIO  
ET : ENLACE TECNICO  
EP : EMPRESA PROGRAMACION  
S-EP: SUPERVISOR EMPRESA PROGRAMACION  
IMP: COORDINADOR DE IMPLEMENTACION  
LIB: ENCARGADO DE LIBRERIA  
ISQA: ENCARGADO DE EVALUACION DE CALIDAD INTERNO

Toda la documentación mencionada en los próximos puntos, son consideradas Items de Configuración, susceptibles de ser cambiadas pero sus cambios controlados por versiones. Se creo la carpeta por proyecto, la cual mantendría las diversas versiones de los Items de Configuración, según la etapa en que el proyecto se encuentre. Esta carpeta contiene el documento “Tabla de versiones”, donde se va indicando las fechas de cada una de las versiones generadas según el Item generado o cambiado.

**FIGURA 12**

**TABLA DE CONTROL DE VERSIONES**

INFORMACION DE IDENTIFICACION DEL PROYECTO					
1. País		2. Gerente de Proyecto (sig. & sello)		3. Desarrollador	
4. Número LASER		5. Descripción LASER		6. Fecha Inicio Proyecto	
DOCUMENTOS LINEA BASED			FECHAS DE REVISION		
Nombre	Respons	1	2	Coment	
<b>Inicio</b>					
Inicio de requerimiento (LASER origi	Usuari				
país	país				
Especificaciones Funcionales	Usuari				
país/E	país/E				
Lista de Chequeo esta fase	ET/ISC				
<b>Definición</b>					
Análisis de Impacto y Lista de compo	ET/ISC				
Plan de pruebas unitarias (Requeridos	ET/ISC				
todo riesgo)	ET/ISC				
Plan de pruebas de sistemas (Req. para	ET/ISC				
riesgo técnico)	ET/ISC				
Plan de retomo (Req. para Alto riesgo)	ET/ISC				
Calendario/Plan de Proyecto	ET/ISC				
Lista de Chequeo de esta fase	ET/ISC				
<b>Construcción</b>					
Especificación técnica de programas (p	Subcon				
para todo riesgo)	Subcon				
Comparador de código	Subcon				
Resultados de pruebas unitarias (req. /	Subcon				
riesgo)	Subcon				
Resultados de pruebas de sistemas (req	ET/IS				
riesgo)	ET/IS				
Log de seguimiento de defectos (Prue	Subc./				
Unitaria, Prueba de Sistemas)	Subc./				
Evidencia de transferencia a Prueba de	Usuari				
Aceptación de Usuarios (Email)	QA				
Lista de Chequeo de esta fase	ET/ISC				
<b>Verificación</b>					
Plan de Pruebas de Usuario	Usuari				
Resultados de Pruebas de Usuarios	Usuari				
país	país				
Log de seguimiento de defectos de Pru	ET/ISC				
Usuarios	ET/ISC				
Evidencia de éxito en resultados de pr	Usuari				
Usuario (Email))	país/E				
Lista de Chequeo de esta fase	ET/ISC				
<b>Implementación</b>					
Aprobación de pase a producción (Em	Usuari				
Plan de proyecto actualizado	or Impl				
Última impresión del Láser	ET/ISC				
Lista de Chequeo de esta fase	ET/ISC				
<b>Comentarios:</b>					
_____					
_____					
_____					
_____					
_____					
_____					
_____					
_____					
_____					
_____					
Revisado Por Internal SQA:					
Revisado Por External SQA:					

### **4.3.1 Iniciación :**

Los representantes del negocio que están localizados en los países tienen como función la de traducir los requerimientos de los usuarios finales, en especificaciones funcionales, que puedan a su vez ser utilizadas, como base para la confección de las especificaciones técnicas.

Los representantes de negocio de los países, deberán abrir un Láser, adjuntando el documento de requerimiento de negocio y las especificaciones funcionales.

Los representantes de negocio regionales, que están localizados en el centro regional de la organización, tiene como función la de evaluar los requerimientos de negocio y especificaciones funcionales, recibidos por el país, con el fin de determinar si esta completo y cumple con todas las aprobaciones necesarias.

Para el caso de proyectos cuyo esfuerzo es de mas de 40 días, o mas de US\$50,000, el representante de negocio, junto con el usuario final deberá confeccionar el plan de pruebas de aceptación de usuario.

Una vez que el Láser esta correcto, el Representante de negocio regional lo aprueba marcando así el línea base del requerimiento. Así mismo determinara si el Láser pertenece al Area de desarrollo (Para los casos de mejoras o adiciones al Software actual) o al Area de Producción (Para los casos de correcciones de defectos).

Los documentos/Sistemas utilizados en esta etapa de iniciación son

- **Sistema Láser**

Ver descripción en capítulo 4.

- **Requerimientos de negocio**

Ver formularios en capítulo 4.

- **Especificaciones Funcionales**

Las especificaciones funcionales, tienen un formato libre, pero expresan en lenguaje de negocio el requerimiento. Por mas detalles ver capítulo 4. Documento 3.

- **Plan de Pruebas de Aceptación de Usuario**

El plan de pruebas de aceptación de Usuario (PPAU), es de completa responsabilidad del Usuario, y en el se deberán desarrollar los casos de prueba, que el Usuario considera deberán ser probados, según la funcionalidad que maneja con el sistema.

Los casos de pruebas deberán indicar los valores en los distintos campos en las pantallas de ingreso, y los resultados esperados, en los reportes y pantallas.



### **4.3.2 Definición**

Cuando el Láser pertenece al Area de Producción, este requerimiento es procesado inmediatamente, y por técnicos locales. Generalmente estos Lasers, son correcciones cuyo esfuerzo es menor.

Para el caso de los Lasers pertenecientes al Area de Desarrollo, el proceso es mas completo.

El Enlace Técnico, revisa el Láser entregado por la Representante de Negocio, y efectúa el Análisis de Impacto, se determinan los programas impactados en el requerimiento, obteniendo el esfuerzo estimado total para la completacion del requerimiento.

A través del programa “Work-bench” se produce un Plan de proyecto, con el esfuerzo y fechas de cada etapa, se asigna un programador para la tarea de codificación y pruebas unitarias, en base a las disponibilidades de recursos según los Lasers en proceso, el “Work-bench”, produce las fechas automáticamente, fijando el plan del proyecto.

Documentos/Sistemas utilizados en esta etapa :

### **Sistema Láser**

E sistema Láser deberá ser actualizado con las aprobaciones de fechas de entrega y “status” de cada uno de los proyectos.

### **Análisis de Impacto**

El Análisis de Impacto, corresponden al análisis preliminar, donde se determina :

- El tipo de riesgo técnico involucrado.

- Los componentes impactados en el requerimiento
- Estimación del esfuerzo requerido

### **Sistema de Planeamiento de Proyecto (Work-bench)**

Para el desarrollo del plan del proyecto, se usa el Sistema Work-bench, el cual permite agrupar los distintos proyectos (Lasers), con el recurso asignado, y poder así controlar la carga de trabajo de cada uno de los recursos, y a su vez, poder determinar fechas en las que cada uno de los recursos estará disponible.

El Work-bench, efectúa un “Auto-cronograma”, el cual considera feriados y días de vacaciones, de cada uno de los recursos, así como las horas diarias de trabajo. según los esfuerzos por cada una de las etapas del ciclo de vida del proyecto.

### **Plan de Pruebas Unitarias y de Sistemas (Si es aplicable)**

El plan de pruebas unitarias, es requerido para cualquier desarrollo, e involucra un plan de las pruebas individuales a ser efectuadas a cada uno de los componentes cambiados o desarrollados.

Para proyectos considerados de “Alto riesgo técnico”, adicionalmente a las pruebas unitarias se deberá efectuar pruebas integradas o de sistemas, las cuales consisten en probar los componentes cambiados y nuevos, conjuntamente con los componentes que no cambiaron, pero que forman parte de la aplicación o sistema cambiado. La ejecución de las corridas “Batch”, de fin de día o de mes, y la generación de archivos de interfases son algunos ejemplos.

### **4.3.3 Construcción**

Una vez completado el Análisis de Impacto, y el Plan de proyecto, el requerimiento se entrega a la empresa de servicios de programación, para que se analice y diseñe el cambio.

La empresa de servicio deberá verificar que los tiempos de esfuerzo estimados y las fecha de entrega son aceptables. En caso de no ser así, deberá informar en un plazo no mayor de cinco días útiles, con el fin de ajustar las fechas en caso necesario.

Para proyectos de cambios y desarrollos mas sofisticados, el analista-programador deberá entregar el diseño técnico para revisión y aprobación.

Una vez culminado el desarrollo, la empresa de servicios de programación deberá completar y entregar la siguiente documentación, localizándola dentro de la plataforma de desarrollo, en un directorio con nombre el Código del Láser.

#### **Documentación Técnica**

La documentación técnica, deberá ser desarrollada por cada programa modificado o nuevo, incluyendo los siguientes componentes

- Descripción de programa
- Diagrama de flujo
- Layouts de archivos
- Diseño de Pantallas
- Diseño de Reportes

## **Formularios con las listas de programas y CRCs**

Este formulario servirá, para promover los programas involucrados, a las áreas de pruebas de Usuarios. El formulario mostrara los CRC anteriores y nuevos. Este CRC se utiliza para mantener un control de los cambios de los programas.

## **Comparador de los cambios efectuados**

Los cambios a los programas existentes deberán ser presentados en un comparador, que mostrara las diferencias entre la versión original, y la versión cambiada. Las líneas de comentarios relativas a los estándares especificados en el documento “Procedimiento y Estándares” se mostraran en este comparador.

## **Listado de programas nuevos**

Los programas nuevos deberán llevar los estándares especificados en el documento “Procedimientos y Estándares”.

## **Resultados de Pruebas Unitarias y de Sistemas**

Los resultados obtenidas durante las pruebas, así como evidencias de dichos resultados, deberán ser presentados.

## **Guía de Instalación**

Actividades de instalación necesarios para la ejecución de programas nuevos y cambiados, que deberán ser efectuadas por el Data Center y por el Usuario

## **Sistema Láser**

El Sistema Láser será utilizado para informar que el desarrollo ha sido culminado, y que su entrega ha sido completada.

#### **4.3.4 Verificación**

La etapa de verificación es responsabilidad del Usuario y Representante de Negocio. Una vez recibida la documentación de construcción del subcontratista el “Enlace Técnico”, revisa y aprueba dicha documentación, y envía la autorización de transferencia a las áreas de pruebas de Usuarios los programas cambiados y nuevos, así como también, informa la guía de instalación de dichos cambios.

La transferencia al área de pruebas es realizada por un área de la organización independiente encargada del control de cambios. Dicha área a través de un sistema de catalogación, verifica los CRC antiguos y nuevos de cada uno de los componentes o programas, y copia estos a las áreas de pruebas de usuarios. Evidencia de estas transferencias, deberá ser incluida en la carpeta del proyecto.

Luego de esta transferencia, el requerimiento esta disponible para las pruebas de Usuarios. Los usuarios deberán aplicar el plan de pruebas de usuario desarrollado, y evaluar si los resultados obtenidos son los esperados.

Los defectos encontrados durante estas pruebas deberán ser informados por escrito (Email) al “Enlace Técnico”, el cual deberá incluirlas en el Log de Seguimiento de defectos, y deberá devolver al supervisor de la empresa de programación, para su corrección.

Los defectos encontrados serán clasificados de acuerdo a su origen como

- Error de Especificaciones Funcionales (SPC)
- Error de Diseño/ codificación (BUG)
- Errores compuestos (Especificaciones funcionales y de diseño/codificación) (BSP)

La empresa de programación deberá efectuar entregas completas incluyendo la corrección del defecto reportado, en directorios internos (SPC,BUG o BSP) según sea el caso, dentro del directorio inicial. Así la estructura de entregas será como sigue :

## Directorio de Entregas

|

|-->Código de Láser

|

|---> BUG

|

|---> BSP ....etc.

Los documentos/Sistemas actualizados dentro de esta etapa son

### **Evidencia de resultados de pruebas de usuario**

Esta evidencia de resultados de la prueba deberá mantenerse en el país, para efectos de auditoria.

### **Log de seguimiento de defectos**

El Log de seguimiento de defectos, deberá ser completado con todos los defectos, encontrados, incluyendo las fechas de resolución, y pruebas efectuadas antes de la corrección.



**Informe que los resultados de las pruebas y la aprobación respectiva**

Este informe es requerido para la implementación a producción del proyecto.

### **4.3.5 Implementación**

Una vez culminadas satisfactoriamente las pruebas de aceptación de Usuario, el Representante de Negocio del País, enviara la aprobación para la transferencia a producción.

#### **4.4 Funciones de Soporte de Producción**

Las funciones principales de soporte de producción en el departamento de soporte de sistemas HP son :

- Soporte de la aplicación en producción (En línea/Fin de día)
- Reparación de Datos de producción
- Ejecución de requerimientos de mantenimiento de la aplicación
- Detección temprana de errores y mantenimiento preventivo
- Mejoramiento continuo del software

La función de reparación será ejecutada por los miembros de soporte a producción, bajo la supervisión y control del Gerente de Producción.

La definición de termino “Reparación” esta relacionado con un Login a la maquina de producción por un técnico autorizado con la finalidad de resolver o investigar cualquier problema reportado. El propósito de la “reparación” es la de conseguir las actividades normales del negocio en el menor tiempo posible.

El usuario (Ya sea un usuario del país, un representante de negocio o un operador del “Data Center”) es responsable en reconocer y reportar los problemas de producción, cuando ellos ocurren.

La identificación de cualquier problema de producción es en la mayoría de las situaciones, una tarea directa, debido a que cada vez que un programa falla, el siguiente mensaje Estandar será mostrado :

**ERROR : ## LINE : ##### PROGRAM : Ppppppp USER : bbb x---x dd.mmm yy**

**\*\* PLEASE, NOTIFY TO DATA CENTER IMMEDIATELY \*\*\***

Donde:

Ppppppp = Nombre de programa

bbb = Número de Sucursal

x---x = Nombre de Usuario

dd = Día

mmm = Mes

aa = Año

La información mostrada en el mensaje de error es una facilidad de identificación que deberá ser usada por el Usuario para reportar problemas.

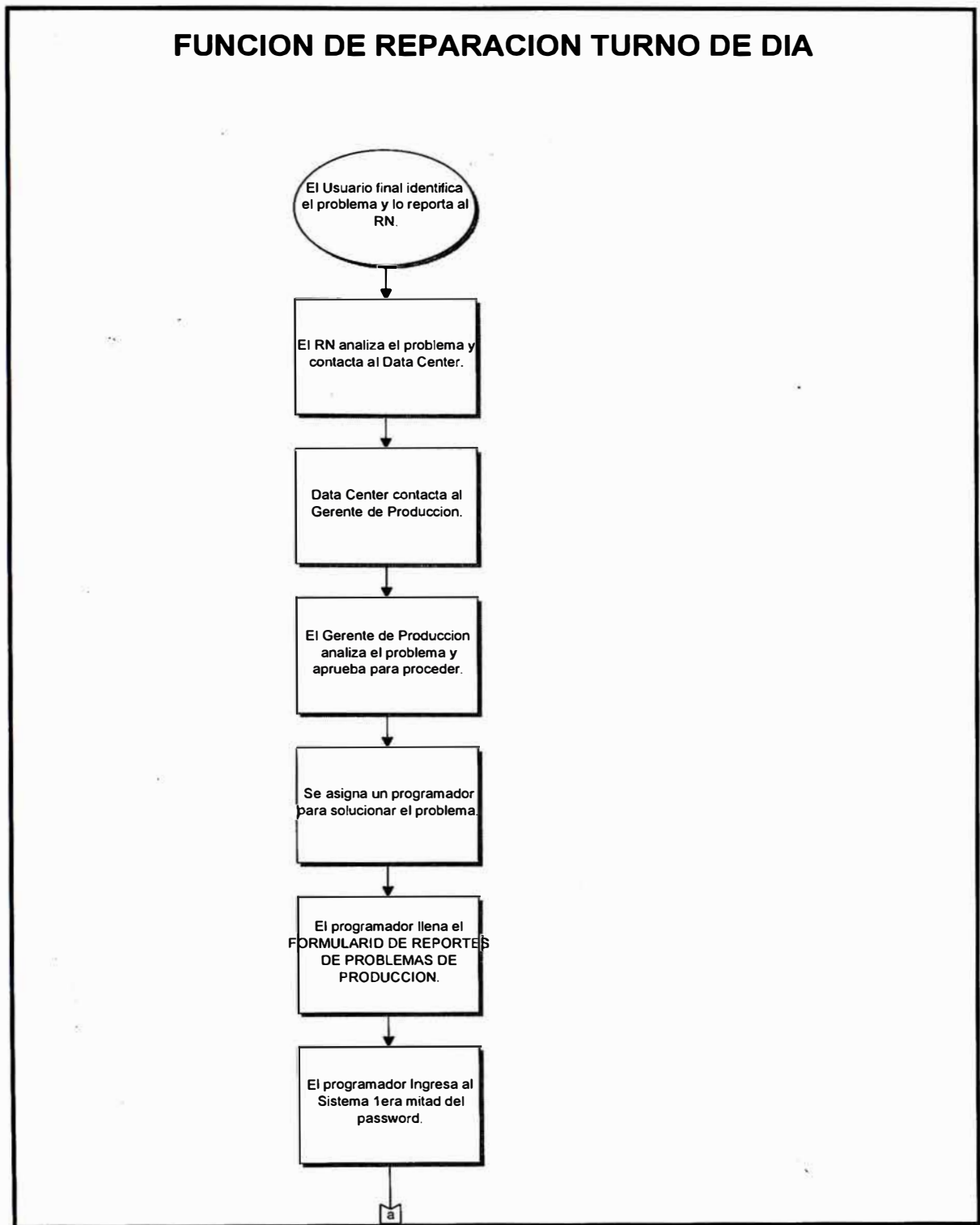
En otros casos, la identificación de una condición de error es hecha por una observación de las condiciones del ambiente, como un comportamiento errático de los programas de la aplicación, condiciones de “loop”, procesamiento incompleto, discrepancias en resultados de transacciones, etc. En esos casos el Usuario deberá coleccionar toda la información pertinente acerca de los síntomas del problema y detalles acerca de la secuencia de eventos que precedieron al problema.

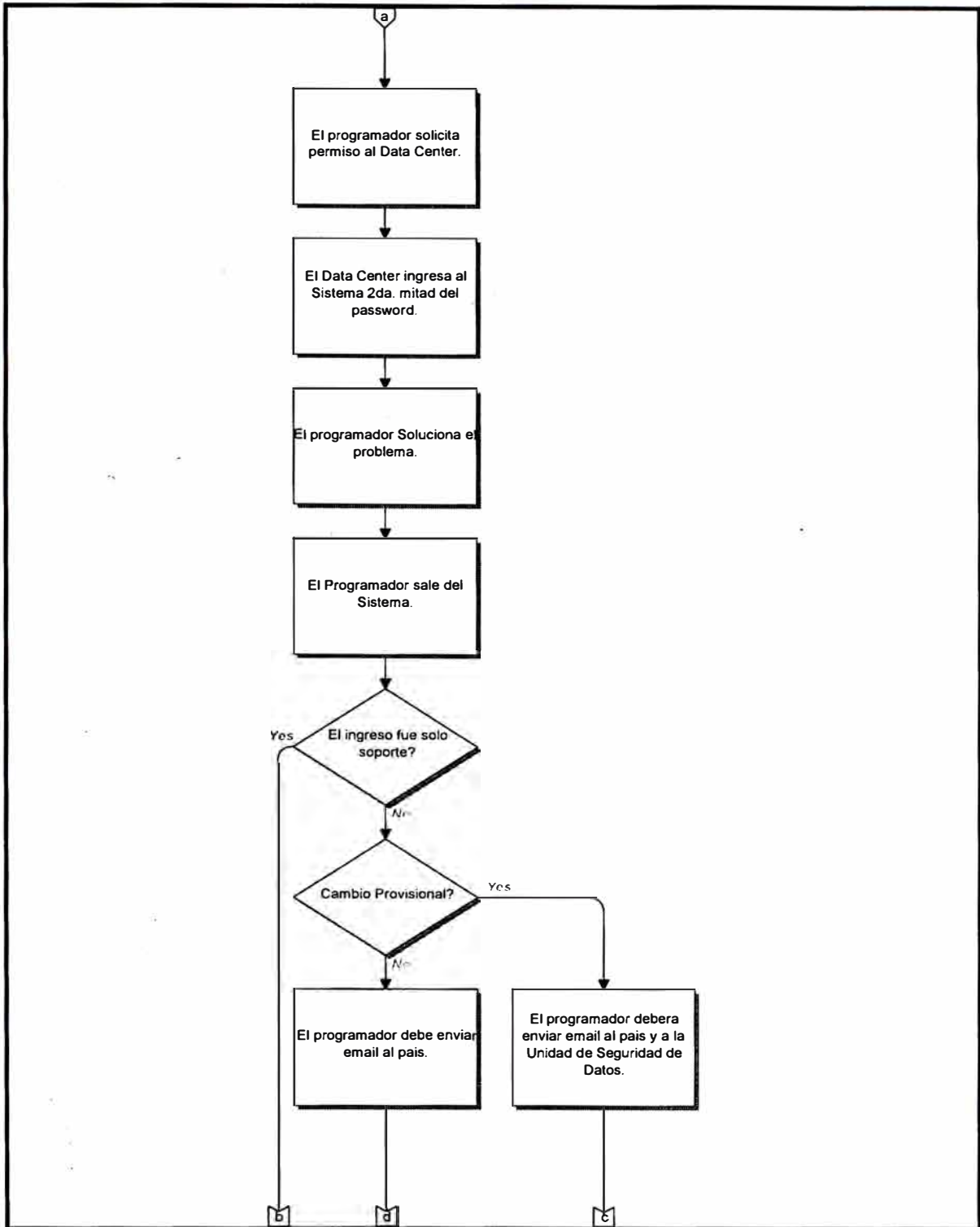
En cualquier evento, los usuarios finales y/o los representantes de negocios del país deberán incluir una descripción detallada de la transacción o actividad bancaria que está siendo ejecutada al momento que el error ocurrió.

El gerente de producción deberá determinar si el problema debe ser considerado como una emergencia real o si este podría ser arreglado como un proceso normal de mantenimiento a través de "LASER". Esta decisión es normalmente basada según el grado de impacto en el negocio.

**FIGURA 13**

**Procedimientos de Reparación (09:00AM a 06:00PM)**





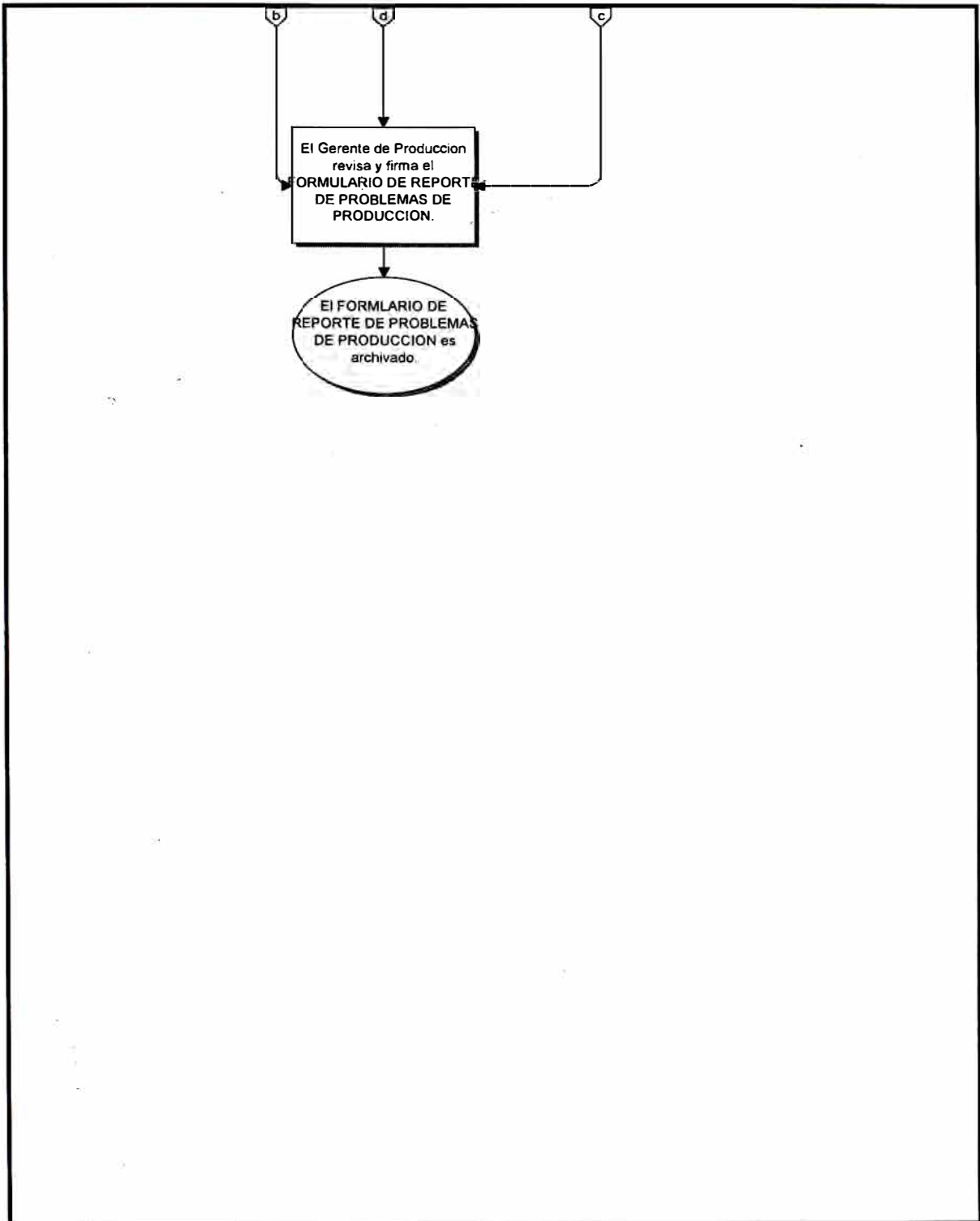
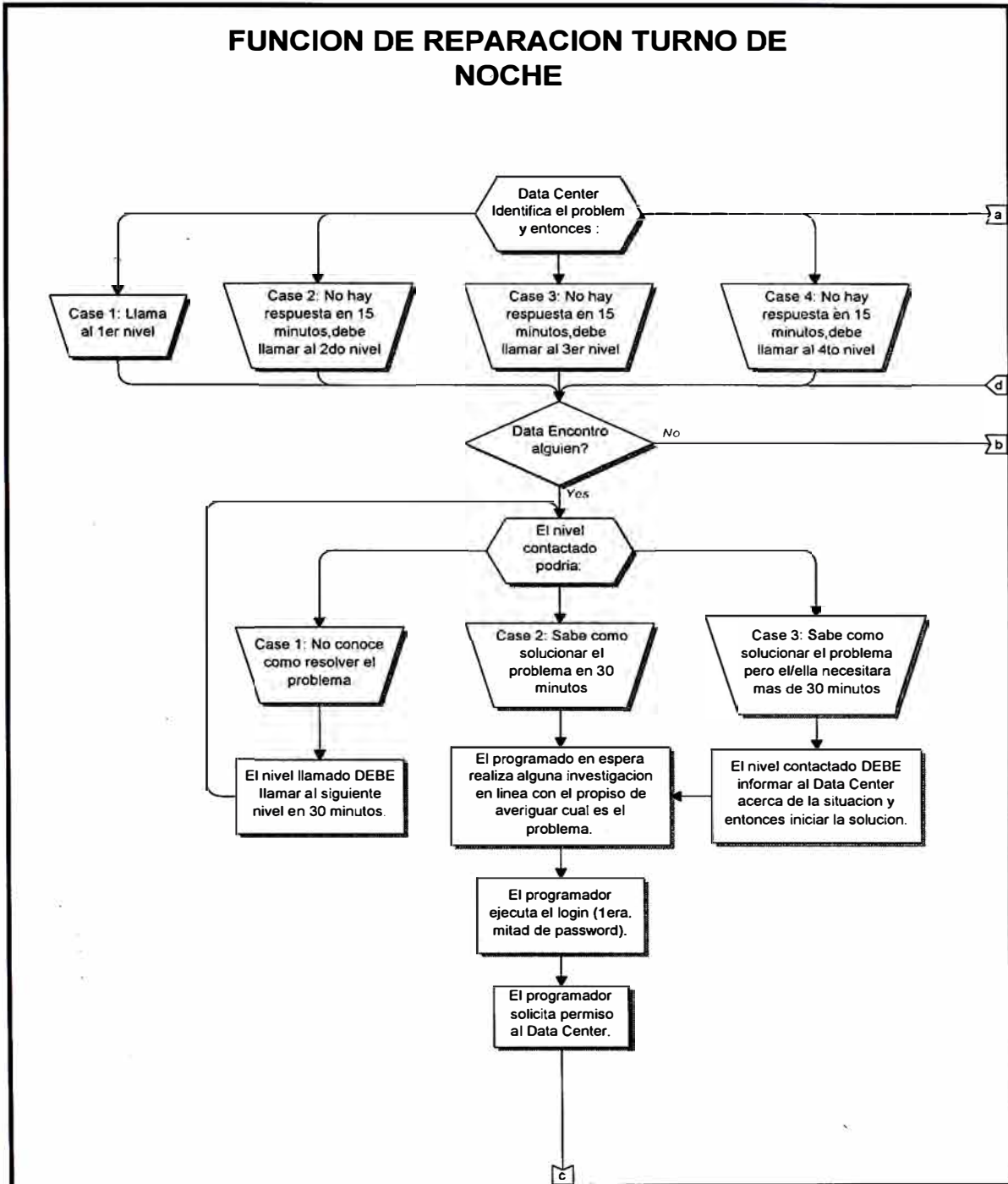
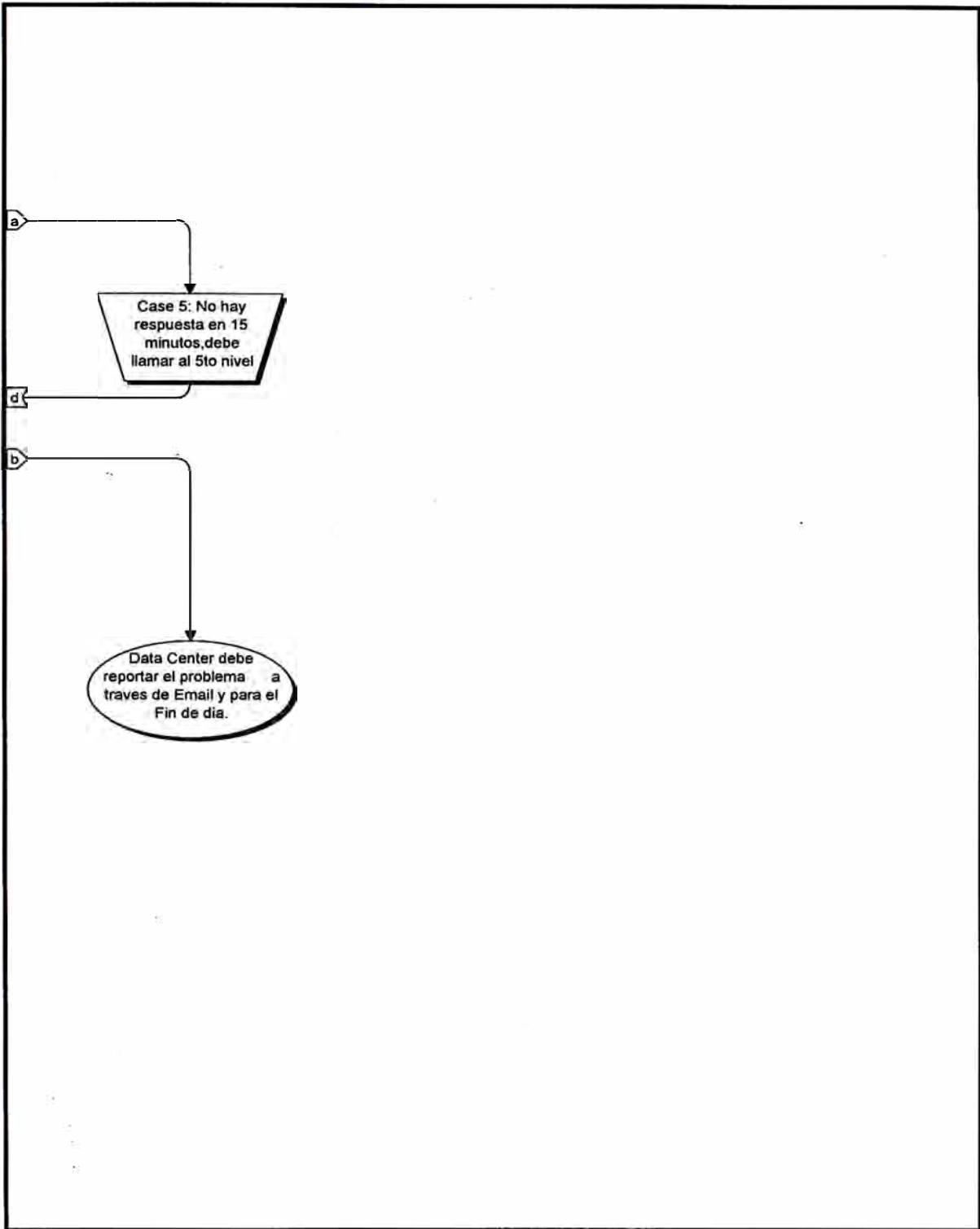


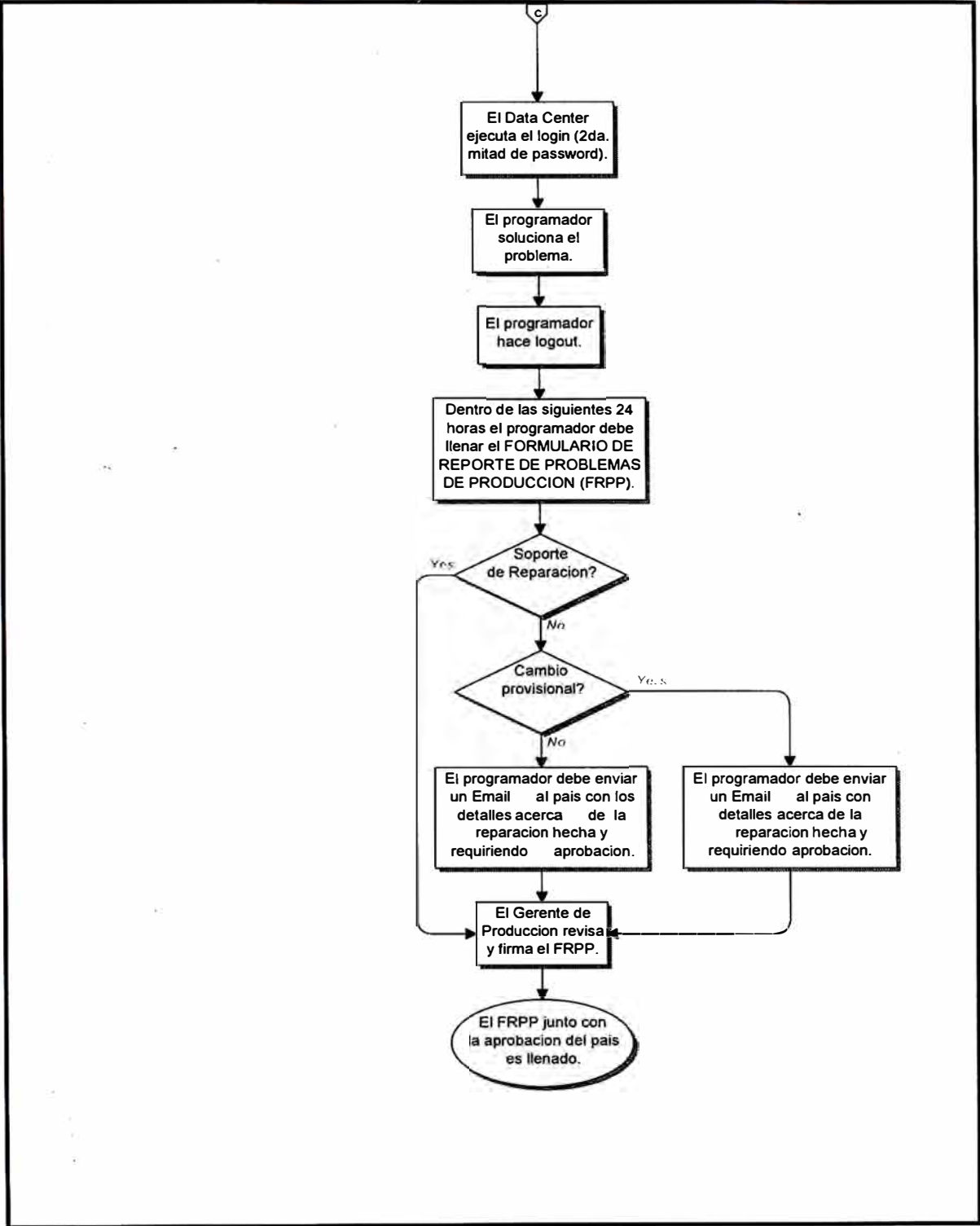


FIGURA 14

Procedimientos de Reparación para Proceso "Batch" (06:00PM a 09:00 AM ET)







#### **4.5 Procedimientos y Manuales**

Durante el proyecto de CMM nivel 2 , el departamento de soporte de sistemas HP, creo diversos manuales de acuerdo a las guías de desarrollo de procedimientos y manuales entregados por el Equipo del Proyecto CMM Nivel 2.

Los principales manuales han sido previamente descritos anteriormente.

La lista de manuales y procedimientos es la siguiente

**TABLA IV**

**DOCUMENTOS DEL DEPARTAMENTO DE SOPORTE DE SISTEMAS HP**

#Doc.	KPA Nivel	Descripción Documento
1	SCM	Procedimientos y Estándares
2	SCM,PTO	Procedimientos unidad de desarrollo
3	SCM,PTO	Procedimientos unidad de Producción
4	SCM	Procedimientos unidad de Implementación
5	SQA	Plan de Evaluación de Calidad
6	SCM	Plan de Gestión de Configuración de Software
7	SCM	Tabla de entregas por versiones
8	SCM	Procedimiento de Meetings de Control de Cambios
9	PTO	Plan de Pruebas de Sistemas
10	PTO	Resultados de Pruebas de Sistemas
11	SCM	Cambios en requerimientos
12	SCM	Mejoramiento de Procesos de Software
13	RM	Inventario de Documentación de requerimientos
14	SCM	Lista de chequeo de folder
15	SCM	Documentación de la organización del folder
16	PTO	Análisis de Impacto
19	RM	Especificaciones Técnicas)
20	RM	Plan Anual de Soporte de Software
21	RM	Formatos de requerimientos de negocio y especificaciones funcionales
22	PTO	Post-Implementation Customer Survey form for S
23	PTO	Hoja de Trabajo de Proyectos de Software
24	PTO	Hoja de Trabajo de esfuerzos de proyecto
25	RM	SDH Lista de Chequeo de Requerimientos de negocio
26	SQA	SDH Lista de Chequeo de Inspección de código SQA
27	SQA	SDH Lista de Chequeo de Especificaciones funcionales SQA
28	SQA	SDH lista de chequeo de planes de pruebas integrales y de sistemas
29	SQA	SDH Lista de chequeo de especificaciones técnicas SQA
30	SQA	SDH Lista de chequeo de Plan de pruebas unitarias SQA
31	SCM	Log de defectos

## **CAPITULO V - ANALISIS COSTO/BENEFICIO**

### **COSTOS**

Los Costos incurridos por la organización para alcanzar el nivel 2 del Modelo de capacidad de maduración (CMM), estuvieron distribuidos en los siguientes rubros

#### **Contratación de Personal**

- Contrataciones de Consultores especializados del Instituto de Ingeniería de Software(SEI).
- Contratación de Staff adicional, con experiencia en este tipo de proyectos, para hacerse cargo de la preparación de la evaluación/examen de la organización.
- Contratación de Escritores Técnicos, que deberían ayudar escribiendo los nuevos procedimientos desarrollados.

#### **Entrenamientos :**

- Presentaciones a la alta gerencia del modelo
- Cursos a Consultores y Staff de la Empresa, sobre los distintos KPAs, que conforma el Nivel 2:
  1. Requirements Management (RM)
  2. Software Configuration Management (SCM)
  3. Software Subcontract Management

4. Software Quality Assurance Management (SQA)
5. Software Project management (SPM)

#### **Certificación de CMM Nivel 2 :**

- La evaluación o examen para certificar el nivel 2, que lo dan los miembros del instituto de ingeniería de software (Software Engineering Institute), tiene un costo alto, pues es una semana en la cual dichos miembros evalúan a todo el personal, sobre como realizan sus actividades y como estas están cumpliéndose según el modelo.

#### **Desarrollo de Sistemas de Soporte**

- Se tuvieron que desarrollar Sistemas de control de requerimientos, que permitiera llevar un inventario controlado de los requerimientos de desarrollo, de tal manera de cumplir los objetivos de mantener una formalización de estos.

#### **Adquisición de Software**

- Se adquirieron paquetes comerciales, para el Gestión de proyectos (cronograma/recursos, etc.), a ser instalados en todas las estaciones de gerentes de proyectos.

#### **Viajes :**

- El equipo del proyecto visito otras organizaciones, que ya alcanzaron nivel 2 en CMM, con el fin de absorber sus experiencias.

- Se visito a los países de América Latina con fines de entrenamiento a los representantes del negocio (RN), básicamente en lo relacionado al KPA "RM" (Requirements Management).
- Formularios de Requerimientos de negocios, y especificaciones funcionales.

### **Desarrollo de Sistemas de Soporte**

- Se tuvieron que desarrollar Sistemas de control de requerimientos, que permitiera llevar un inventario controlado de los requerimientos de desarrollo, de tal manera de cumplir los objetivos de mantener una formalización de estos.



**TABLA V**

<b>Rubro</b>	<b>Costo (US\$)</b>
<b><u>Contratación de Personal</u></b>	
<b>Consultores SEI (2x8meses)</b>	160,000
<b>Staff experimentado (4x8meses)</b>	200,000
<b>Escritores técnicos (7x8meses)</b>	280,000
	=====
<b>Sub Total</b>	640,000
<b><u>Entrenamientos</u></b>	
<b>Presentaciones alta gerencia</b>	10,000
<b>Requirements Management (RM)</b>	50,000
<b>Software Configuration Man.(SCM)</b>	50,000
<b>Software Subcontract Man. (SSM)</b>	50,000
<b>Software Quality Assurance (SQA)</b>	50,000
<b>Software Project Management (SPM)</b>	150,000
	=====
<b>Sub Total</b>	360,000
<b>Certificación de CMM nivel 2</b>	300,000
<b>Adquisición de Software</b>	
<b>Project Workbench (20 licencias)</b>	100,000
<b>Viajes</b>	100,000
<b>Desarrollos de Sistemas de Soporte</b>	
<b>Sistema Láser</b>	30,000
<b>Total</b>	<b>1,530,000</b>

## **BENEFICIOS**

Los beneficios obtenidos en la aplicación de este modelo, van mas allá del entorno del proyecto de alcanzar el nivel 2. Este modelo, como su nombre lo indica es un modelo de maduración.

Quiere decir que cada nivel alcanzado ofrece beneficios, por si mismo pero a su vez abre el camino para alcanzar el nivel inmediato superior, constituyendo esta característica un gran beneficio.

No se trata de un modelo estático, si no que contempla un dinamismo que es en síntesis lo que lo hace valioso y beneficioso para cualquier organización, y especialmente para organizaciones dedicadas al desarrollo de proyectos de software.

Luego de estar trabajando en el nivel 2 por mas de un año, se pueden catalogar los beneficios logrados, en los siguientes

- Reducción del Costo de Desarrollo en un 50% por :
  1. Reducción de discrepancias entre especificaciones funcionales y diseño técnico
  2. Reducción de errores de construcción por mejoramiento sustancial de los planes de pruebas unitarias y de sistemas
  3. Mayor control en recursos utilizados, planificando y haciendo seguimiento de cada proyecto.
- Aumento de calidad de los productos de Software entregados

Los cambios y mejoras en los sistemas reflejan en forma mas precisa los requerimientos del negocio.

o

- Mejora en la Documentación de los Sistemas y Cambios de Software

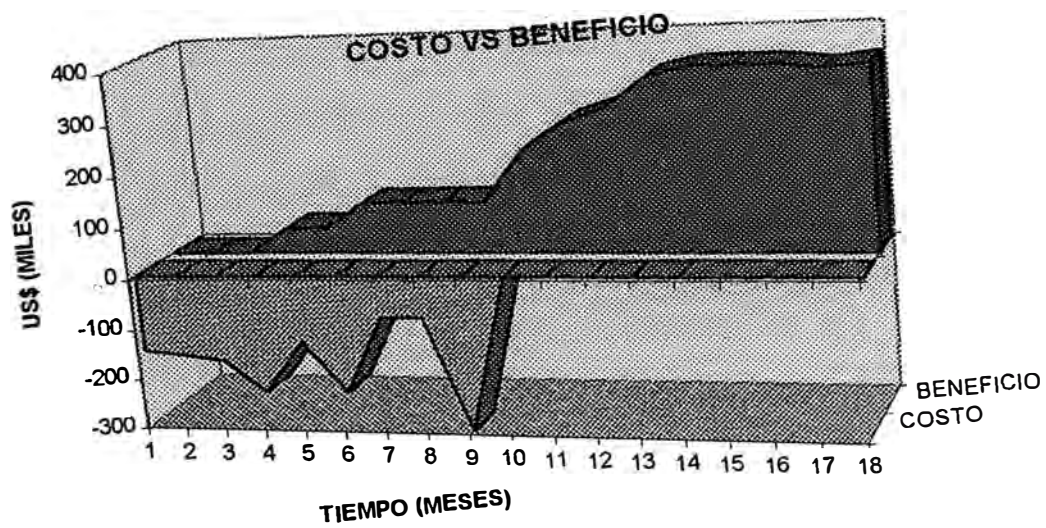
Los sistemas y cambios son mejor documentados, generándose una invaluable información, a ser usada en proyectos futuros.

- Recolección de mediciones

Diversas métricas, están siendo registradas sobre las diferentes etapas del ciclo de vida de desarrollo. Estas métricas serán posteriormente utilizadas en procesos estadísticos, para la generación de información valiosa que permitirá la mejora de los procesos, conforme se alcancen los niveles siguientes del modelo.

La figura 15 muestra una gráfica entre los costos aplicados por mes y los beneficios obtenidos, por la aplicación del modelo nivel 2. Nótese que los beneficios se reflejan mayormente luego de que la organización fue certificada en el nivel 2.

**FIGURA 15**



## CONCLUSIONES

En este documento considero he transmitido en forma gradual mi experiencia profesional de la aplicación del modelo de capacidad de maduración, presentando inicialmente la parte teórica del modelo en su versión completa, para luego pasar a describir el caso particular de la aplicación del modelo para alcanzar el nivel 2.

- El nivel 2 (Repetitivo), es considerado como base para iniciar una maduración completa y lograr los siguientes niveles, los cuales deberían en rigor ser menos dificultosos de alcanzar. El inicio del proceso es mas dificultoso, pues se parte de un nivel de desarrollo de software, en el cual no existen procesos definidos, o dichos procesos no contemplan en su totalidad los factores que permiten el control completo del ciclo de vida de desarrollo de sistemas.
- La base lograda en nivel 2, permite por ejemplo la obtención y registro de mediciones de todo tipo tales como estimaciones de tamaños de proyectos, defectos encontrados en cada etapa y el tiempo utilizado para sus correcciones, etc. Así como una librería de proyectos, todos desarrollados con el mismo proceso. Es lograr tener una misma unidad de medición en desarrollo de sistemas.
- Los niveles siguientes se nutren en parte de las mediciones y librerías que han sido capturadas en el nivel 2. El proceso de maduración entonces continua, aunque oficialmente la organización se encuentre en el nivel 2. Conforme se aumenta la experiencia repetitiva, se aumenta la información a ser utilizada en niveles superiores.

- La evolución es posible y toma tiempo. Hay una aproximación sistemática para mejorar la manera que el software es construido y mantenido.
- Considero que el modelo de Capacidad de Maduración (CMM), es un modelo que podría ser aplicado en la Empresa Peruana, considerando que el modelo presenta niveles que permiten adecuarse a una diversidad de presupuestos, alcanzando el nivel permitido por cada caso particular.

## RECOMENDACIONES

- Visualizar el modelo CMM, como una forma de trabajo, y no como una serie de formatos que obstruyen la velocidad del desarrollo de software. Mi experiencia personal en la aplicación del modelo, es que inicialmente se produce un fenómeno natural de rechazo al cambio, y al incremento de tareas, producto de la implementación de los procedimientos, pero gradualmente, estos van tomando sentido, hasta que se notan cambios de calidad en el software entregado, y a su vez se reducen los tiempos de entrega al detectar defectos en las fases iniciales del desarrollo.
- Forzar el cumplimiento de los procedimientos en todos los niveles. Los procedimientos definidos no son útiles si estos no se implementan. La implementación de los procedimientos es la clave del éxito. Y para ello, se deberá forzar su cumplimiento, a través de incentivos, e inclusive algún tipo de mecanismo de presión.
- La certificación del pase de nivel, es sin duda el acontecimiento más importante en todo este proceso. En esta certificación, participan todos los niveles de la organización, y todos tienen que tener claro el modelo, como se está aplicando, en cada caso particular, y que objetivo del CMM se estaría cubriendo. Especial atención se deberá considerar para la transmisión de los conocimientos del modelo. Se deberán efectuar simulacros de evaluaciones, para asegurarse que el personal está completamente mentalizado con el tema. Estos simulacros a su vez permiten

visualizar los niveles alcanzados por cada departamento, permitiendo así, incidir en las áreas mas débiles.

- Por ultimo, y como una recomendación personal sobre este tema, es la de darle al modelo flexibilidad. Como menciono al inicio, este modelo como cualquier otro, es abstracto, pues pretende abarcar una diversidad de situaciones, en patrones constantes, sin embargo su aplicación variará según la realidad del ambiente donde se utilice. Tal vez en países en vías de desarrollo, se deba ajustar algunos de los KPAs al país, y hacerlo menos estricto en ciertos aspectos. Sin embargo estos cambios se deberán definir y documentar desde el inicio. Podemos semejarlo a algun juego. Las reglas se deben definir al principio, para no afectar los resultados de cada participante durante el transcurso del juego. Es así, que para la aplicación del modelo en el caso particular de la organización objeto de este informe, se desarrollo una visualización propia del modelo, que es la que he tratado de mostrar.



## GLOSARIO DE TERMINOS

TERMINO	DESCRIPCION
AB	ABILITY TO PERFORM
AC	ACTIVITIES TO PERFORM
AUTOSCHEDULE	GENERACION AUTOMATICA DE CRONOGRAMA
BSP	ERROR PROGRAMACION Y ESPECIFICACIONES
BUG	ERROR PROGRAMACION
CL	CONTROL DE LIBRERIA
CLAVE PRACTICA	COMPONENTE DE KPA
CMM	CAPABILITY MATURITY MODEL
CO	COMMITMENT TO PERFORM
COMMON FEATURES	VENTAJAS COMUNES
DATA CENTER	CENTRO DE DATOS REGIONAL
DP	DEFECT PREVENTION
ET	ENLACE TECNICO
FRPP	FORM. REPORTES DE PROBLEMAS DE PRODUCCION
GOALS	OBJETIVOS
GRD	GRUPO REGIONAL DE DESARROLLO
GRP	GRUPO DE PROGRAMACION REGIONAL
HARDWARE	COMPONENTE DURO
HP	HEWLETT PACKARD
IC	INTERGROUP COORDINATION
ISM	INTEGRATED SOFTWARE MANAGEMENT
KPA	KEY PROCESS AREA
LASER	SISTEMA DE CONTROL DE REQUERIMIENTOS
LINEA BASE	BASELINE - SE FIJA UNA BASE
LCCP	LOG DE CONROL DE CAMBIOS DEL PROYECTO
M&MS	MANTENIMIENTO Y MEJORA DE SOFTWARE
ME	MEASUREMENT AND ANALYSIS
MILESTONE	PUNTO DE CONTROL
OPD	ORGANIZATION PROCESS DEFINITION
OPF	ORGANIZATION PROCESS FOCUS
PCM	PROCESS CHANGE MANAGEMENT
PPAU	PLAN DE PRUEBAS DE ACEPTACION DE USUARIO
PR	PEER REVIEW
PTO	PROJECT TRACKING AND OVERSIGHT
QA	QUALITY ASSURANCE
QPM	QUANTITATIVE PROCESS MANAGEMENT
RAU	REPRESENTANTE DE ACEPTACION DE USUARIO
RFS	REQUEST FOR SERVICE
RM	REQUIREMENTS MANAGEMENT
RN	REPRESENTANTE DE NEGOCIO
RT	REPRESENTANTE TECNICO
RU	REPRESENTANTE DE USUARIO
SCCB	SOFTWARE CHANGE CONTROL BOARD

<b>TERMINO</b>	<b>DESCRIPCION</b>
<b>SCM</b>	SOFTWARE CONFIGURATION MANAGEMENT
<b>SDH</b>	SOFTWARE DEVELOPMENT HANDBOOK
<b>SDLC</b>	SOFTWARE DEVELOPMENT LIFE CYCLE
<b>SEI</b>	SOFTWARE ENGINEERING INSTITUTE
<b>SOFTWARE</b>	COMPONENTE BLANDO
<b>SPC</b>	ERROR DE ESPECIFICACION
<b>SPE</b>	SOFTWARE PRODUCT ENGINEERING
<b>SPP</b>	SOFTWARE PROJECT PLAN
<b>SQA</b>	SOFTWARE QUALITY ASSURANCE
<b>SQA</b>	SOFTWARE QUALITY ASSURANCE
<b>SQM</b>	SOFTWARE QUALITY MANAGEMENT
<b>SSM</b>	SOFTWARE SUBCONTRACT MANAGEMENT
<b>TCM</b>	TECHNOLOGY CHANGE MANAGEMENT
<b>TP</b>	TRAINING PROGRAM
<b>VI</b>	VERIFICATION AND IMPLEMENTATION
<b>WORK-BENCH</b>	SISTEMA DE CONTROL DE PROYECTOS

## BIBLIOGRAFIA

Nombre	Autor	Editorial	Año
Key Practices of the Capability Maturity Model Versión 1.1	Mark C.Paulk Charles V.Weber Suzanne M.Garcia Mary Beth Chrissis Marilyn Bush	Software Engineering Institute Carnegie Mellon University Pittsburgh Pennsylvania 15213	1993
A Guide to CMM Understanding the Capability Maturity Model for Software	Kenneth M, Dymond	Process Inc US P.O Box 1988 Annapolis Maryland 21401 USA	1995