



03063

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

28

POSGRADO EN CIENCIA E INGENIERIA DE LA COMPUTACION

**DESARROLLO DE UNA GUIA DE
INTERPRETACION Y COMPRENSION
DEL MODELO DE MADUREZ DE
CAPACIDADES (CMM), NIVEL 3.**

TESIS

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS**

**PRESENTA:
LEONARDO RAMIREZ FERNANDEZ**

DIRECTORA DE TESIS: DRA. HANNA OKTABA

MEXICO, D. F.

OCTUBRE, 2001



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos.

A mis padres y mis hermanos por haberme apoyado siempre.

A la Dra. Hanna Oktaba

Por su tiempo que dedicó a dirigir la tesis y por los conocimientos que nos enseñó durante la maestría.

Al jurado:

M. en C. Guadalupe Ibarguengoitia.

Dr. Fernando Gamboa Rodríguez.

M. en C. Javier García García.

M. en I. Reynaldo Alanis Cantú.

Por haber revisado el trabajo de tesis y brindarme sus comentarios.

A mis Amigos Angel y Gloria que siempre han estado conmigo.

A Erika por su Amistad.

CONTENIDO

Introducción		3
Capítulo 1	Estructura del Modelo de Madurez de capacidades (CMM)	6
1.1	¿Qué es CMM?	6
1.2	Historia	6
1.3	Representación gráfica de la estructura del CMM	7
1.4	Los niveles de madurez	9
1.5	Áreas claves de procesos (ACP)	9
1.6	Prácticas claves	11
Capítulo 2	Nivel 3 del Modelo de Madurez de Capacidades (CMM): Definido	13
2.1	Áreas claves de proceso del Nivel 3 Definido:	13
2.2	Metodología para el desarrollo del modelo gráfico	16
Capítulo 3	Enfoque en el Proceso de la Organización	18
3.1	Enfoque en el Proceso de la Organización	19
3.2	Modelo gráfico del Enfoque en el Proceso de la Organización (EPO)	28
3.3	Identificación de los grupos participantes	28
3.4	Diagrama de actividades	29
Capítulo 4	Definición del Proceso de la Organización	37
4.1	Definición del Proceso de la Organización.	38
4.2	Modelo gráfico Definición del Proceso de la Organización (DPO)	50
4.3	Identificación de los grupos participantes	50
4.4	Diagrama de actividades	51
Capítulo 5	Programa de Capacitación	57
5.1	Programa de Capacitación	58
5.2	Modelo gráfico Programa de Capacitación (PC)	69
5.3	Identificación de los grupos participantes	69
5.4	Diagrama de actividades	70

Capítulo 6	Administración Integrada de Software	78
6.1	Administración Integrada de Software	79
6.2	Modelo gráfico Administración Integrada de Software (AIS)	99
6.3	Identificación de los grupos participantes	99
6.4	Diagrama de actividades	100
Capítulo 7	Ingeniería de Productos de Software	110
7.1	Ingeniería de Productos de Software	111
7.2	Modelo gráfico Ingeniería de Productos de software (IPS)	133
7.3	Identificación de los grupos participantes	133
7.4	Diagrama de actividades	134
Capítulo 8	Coordinación Entre Grupos	143
8.1	Coordinación Entre Grupos	144
8.2	Modelo gráfico Coordinación Entre Grupos (CEG)	154
8.3	Identificación de los grupos participantes	154
8.4	Diagrama de actividades	155
Capítulo 9	Revisiones Entre Colegas	164
9.1	Revisiones Entre Colegas	165
9.2	Modelo gráfico Revisiones Entre Colegas (REC)	173
9.3	Identificación de los grupos participantes	173
9.4	Diagrama de actividades	174
Conclusiones		179
Bibliografía		182
Anexo 1		183
Anexo 2		193
Anexo 3		219

Introducción

En la actualidad los clientes exigen productos de software de mayor calidad, que cumplan sus necesidades y resuelvan sus problemas con eficacia y eficiencia. Esto ha generado que las empresas que se dedican a desarrollar software busquen la forma de mejorar su proceso de desarrollo de software, ya que por no tener un proceso de desarrollo de software eficaz, se originan los siguientes problemas:

- El incumplimiento de los requisitos del cliente y los usuarios finales al no haber realizado un correcto análisis de los mismos.
- Los costos de desarrollo del proyecto aumentan.
- No hay una asignación de recursos adecuada.
- No se cumplen los plazos del proyecto.
- Mala calidad en el producto final.

El departamento de defensa de los Estados Unidos al observar los grandes problemas que se presentaban en el desarrollo de software, decidió formar el Software Engineering Institute (SEI). Con el fin de establecer estándares de excelencia en la Ingeniería de software.

En los últimos años la industria del software ha estado implantando varios modelos que tratan de recaudar las mejores prácticas basadas en la experiencia colectiva de la industria de software internacional.

El objetivo de estos modelos es minimizar los problemas y ayudar a realizar eficientemente el planteamiento de los proyectos, mejorar la eficiencia y la productividad, desarrollar productos de mejor calidad y generar mayores utilidades.

Los modelos más importantes son el CMM (Capability Maturity Model), generado por el SEI, y el proyecto internacional ISO15504. En ambos casos el objetivo es dar a conocer las mejores prácticas de los procesos de software para que sean implantadas en las empresas que desarrollan software y que carecen de ellas.

Los modelos CMM y ISO15504 están presentados en texto en lenguaje natural (inglés) y carecen de una representación gráfica que los sintetice y que facilite su comprensión. Esto cobra mayor importancia dado que el volumen de la información que se transmite en cada uno de los modelos es considerada. Por ejemplo, el CMM ocupa 479 páginas. Asimilar e interpretar esta cantidad de información es un proceso lento y difícil. Por lo tanto este trabajo consiste en la elaboración de un modelo gráfico para CMM nivel 3 para lograr un rápido y más sencillo entendimiento del CMM.

En México, en los últimos años, las empresas de software trasnacionales empezaron a definir sus procesos siguiendo principalmente CMM. Debido a la falta de expertos

capacitados en este modelo en el país, estos esfuerzos requirieron de muchos recursos destinados a la capacitación y asimilación del conocimiento requerido.

Las empresas pequeñas y medianas, al no disponer de estos recursos, tienen dificultad de aprovechar las prácticas que propone CMM, al no tener fácil acceso a la información en español y en una forma más entendible.

Es por eso que el objetivo de este trabajo de tesis es proporcionar una guía que consiste en la traducción al español del nivel 3 de CMM y la generación del modelo gráfico que abstrae la información contenida en el texto, que ayudara a los usuarios, las empresas pequeñas y medianas que desarrollan software, a comprender y asimilar más fácilmente el propósito de cada una de las áreas claves de procesos del nivel 3 de CMM.

Para la construcción de esta guía se utilizaron los diagramas de actividades de UML (Unified Modeling Language) Lenguaje Unificado de Modelado, [G.Bocch, J.Rumbaugh, I.Jacobson 1999], para modelar gráficamente las áreas claves de proceso del CMM nivel 3 (Definido). Los diagramas de actividades indican el orden en que se harán las cosas, es decir definen las reglas esenciales de secuenciación que debemos de seguir. Además estos diagramas pueden manejar procesos paralelos, así como la posibilidad de indicar explícitamente los actores responsables por la realización de dichas actividades.

El modelo gráfico se elabora como un complemento del modelo textual para facilitar su comprensión y sintetizar los elementos cruciales del proceso. Este modelo gráfico puede ser utilizado como material de apoyo en la capacitación de empresas dedicadas al software que estén interesadas en alcanzar la madurez en sus procesos.

El alcance de este trabajo de tesis sólo cubre las áreas claves del nivel 3 de CMM, tanto su traducción como la generación del modelo gráfico de sus áreas claves de procesos. En trabajos futuros se dará a conocer y se harán pruebas para confirmar que el modelo gráfico ayuda a la comprensión del CMM nivel 3.

Este trabajo es una continuación del trabajo de tesis de maestría [Guía de Interpretación del Nivel 2 de CMM], realizado por Claudia Alquira Esquivel que modeló las áreas claves del nivel 2 de CMM, usando diagramas de actividades de UML.

En el primer capítulo se explica que es el CMM, como surgió, así como la estructura general del CMM, para que el lector se familiarice, comprenda y entienda los niveles del CMM así como sus prácticas claves.

En el capítulo dos se describirán brevemente las áreas claves del nivel 3 (Definido), así como el proceso general para desarrollar el modelo gráfico de cada una de las áreas clave.

Los capítulos 3,4,5,6,7,8,9 están dedicados a las áreas claves de procesos de **Enfoque en el proceso de la organización (EPO), Definición del Proceso de la Organización (DPO), Programa de Capacitación (PC), Administración Integrada de Software (AIS), Ingeniería de Productos de Software (IPS), Coordinación Entre Grupos (CEG), Revisiones Entre Colegas (REC)**, respectivamente. Estos capítulos se desarrollarán de la misma manera es decir cada capítulo contendrá la traducción de cada área clave de proceso así como su modelo gráfico, con la explicación correspondiente.

Al final se presentan las conclusiones de este trabajo.

El trabajo también incluye dos apéndices: el apéndice A tiene una explicación de lo que son los diagramas de actividades, y el apéndice B presenta el área clave de procesos Administración de Subcontratos de Software que pertenece al Nivel 2 del CMM. Se incluyó la interpretación de esta área clave para complementar lo que faltó del nivel 2 de CMM en la tesis de maestría “[Guía de Interpretación del Nivel 2 de CMM]”, realizado por Claudia Alquira Esquivel.

Capítulo 1 Estructura del Modelo de Madurez de capacidades (CMM)

1.1 ¿Qué es CMM ?

El CMM es un conjunto de prácticas importantes que deben ser implementadas por cualquier organización que está interesada en desarrollar software y mejorar la madurez de su proceso. Puede concebirse como el producto de la experiencia colectiva de muchos de los proyectos más exitosos de software, la cual se ha documentado en forma de un modelo. Su propósito es proporcionar recomendaciones a la comunidad del software sobre lo que debe hacerse para lograr proyectos exitosos y mejorar sus procesos continuamente. Publicado por el SEI, ha sido usado ampliamente por la comunidad del software para evaluar la madurez de los procesos del software usados en compañías y agencias, para desarrollar planes de mejoramiento, o como un manual de prácticas recomendables.

Por la estructura con la que fue diseñado el CMM puede ser usado por diferentes grupos, por ejemplo:

- Los equipos de análisis: Lo utilizan para identificar las fortalezas y debilidades en la organización.
- Los equipos de evaluación: Lo utilizan para identificar los riesgos de seleccionar, entre los diferentes contratistas y subcontratistas y para supervisar contratos.
- La administración: Lo utilizan los administradores superiores para entender las actividades que son necesarias para lograr un programa de mejoramiento de proceso de software de su organización.
- El personal técnico y grupos de mejoramiento de proceso: Lo utilizan como una guía para ayudarlos a definir el proceso de software de su organización.

1.2 Historia

El Instituto en Ingeniería de Software comenzó el desarrollo de un modelo de madurez de procesos, que ayudaría a las organizaciones a mejorar sus procesos de software. Este esfuerzo se inició porque el gobierno federal de los Estados Unidos, les solicitó un método para evaluar la capacidad de sus contratistas de software.

Después de muchos esfuerzos se creó el modelo CMM, este modelo está basado en conceptos de calidad total y de mejoramiento continuo, y fue elaborado por el Software Engineering Institute (SEI) de la Universidad Carnegie Mellon.

La versión 1.0 del Modelo de Madurez de Capacidades fue lanzada al público por el SEI en 1990. La versión 1.1, más estable, se publicó en 1993 y tuvo un éxito inmediato dentro de la comunidad internacional. Ampliamente usado por varios segmentos de la industria y organizaciones públicas, ha inspirado numerosas aplicaciones y adaptaciones del modelo. La versión 2 ha sido publicada hasta la fecha en versiones preliminares y ha circulado para revisión.

También es una herramienta ampliamente difundida, la cual ha seguido una evolución ordenada involucrando a la comunidad del software internacional. El modelo de CMM representa, sin ninguna duda, una de las mejores referencias disponibles de prácticas recomendadas para desarrollo o mantenimiento de software.

El modelo de madurez se encuentra descrito en los documentos que aparecen a continuación, cuya versión electrónica (en idioma Inglés) se encuentra disponible en el sitio WEB del SEI-Carnegie Mellon University.

- Capability Maturity Model, versión 1.1 (SEI/CMU-93-TR24)
- CMM Practices, versión 1.1 (SEI/CMU-93-TR25). Este último fue el que se uso para traducir las áreas claves de procesos del Nivel 3.

1.3 Representación gráfica de la estructura del CMM

La estructura del CMM se modeló usando diagramas de clases, ya que por medio de estos diagramas podemos describir los objetos y las relaciones que existen entre ellos.

Al elaborar el modelado del CMM se identificaron las siguientes clases:

- CMM
- NiveldeMadurez
- AreaclavedeProceso
- CaracteristicaComunes
- Practicaclave

El CMM está constituido por 5 cinco niveles de madurez, cada nivel de madurez tiene de cero a siete áreas claves de procesos. La clase `Areaclavedeproceso` tiene dos atributos, el primero es el Propósito que nos dice cual es el objetivo del área clave de procesos y el segundo atributo es Metas que nos dice que es lo que debe lograr la organización al haber implantado el área clave de procesos.

Las prácticas que deben ser realizadas por cada área clave de proceso se organizan en 5 características comunes. Estas están representadas en el modelo como una clase llamada CaracteristicasComunes y cada una de las características comunes son métodos de la clase, se indicaron como métodos ya que describen que es lo que se debe hacer para establecer una capacidad de proceso.

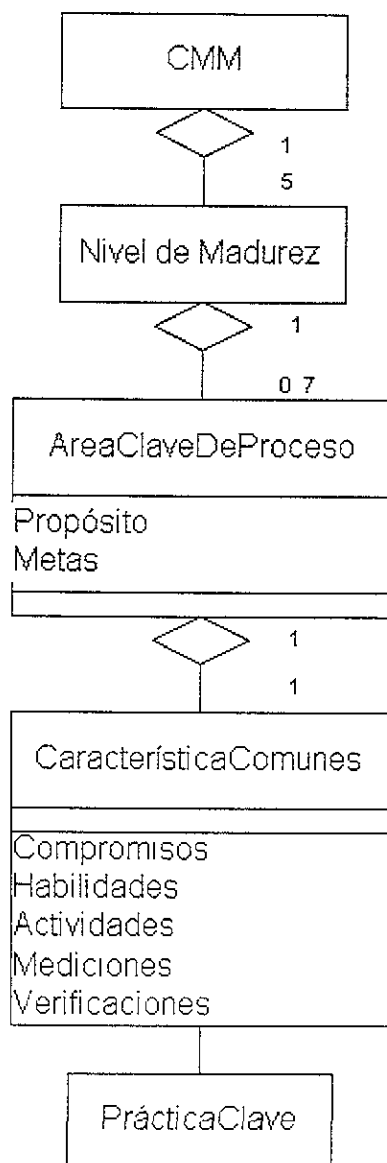


Figura 1.1 Estructura del CMM

1.4 Los niveles de madurez

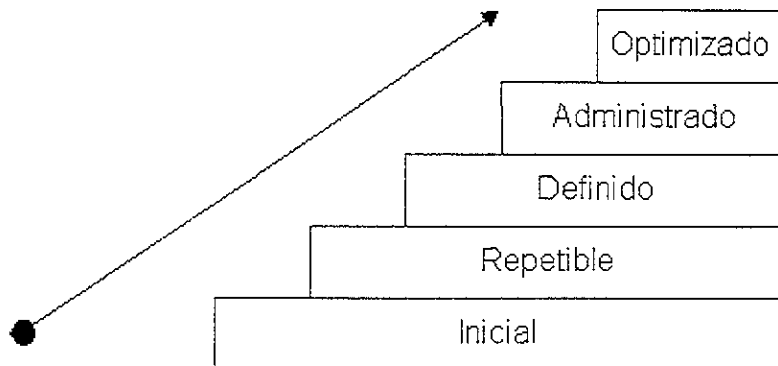
El CMM se organiza en cinco niveles de madurez, los cuales deben ser implementados, por una organización, para lograr el mejoramiento en sus procesos de software.

Estos niveles son:

- Inicial
- Repetible
- Definido
- Administrado
- Optimizado

Una organización que desea mejorar sus procesos de software por medio del CMM, tiene que alcanzar estos niveles en forma progresiva, continua y ascendente, es decir los niveles inferiores son la base de los niveles superiores. Por ejemplo este modelo recomienda que para llegar al nivel tres, la organización debió haber implantado correctamente las áreas claves de proceso del nivel dos, ya que si no se implantan correctamente puede generar problemas en la madurez del proceso e incluso fracasar en el intento.

Por esta característica de los niveles podemos representar al CMM como una escalera, donde cada uno de sus escalones representa un nivel de madurez.



1.5 Áreas claves de procesos (ACP)

Las áreas claves de procesos indican a las organizaciones, donde se deben mejorar sus procesos de software. Cada área clave propone llevar a cabo un conjunto de prácticas para lograr una serie de metas, que CMM considera importantes, para reforzar la capacidad en los procesos.

El CMM no describe todas las áreas de procesos involucradas en el desarrollo y el mantenimiento de software, solo las que considera “clave” para lograr un nivel de madurez. Las áreas claves consideradas en cada uno de los niveles son las siguientes.

Nivel 1

El proceso de software se caracteriza por ser caótico. Por caótico entendemos cuando los proyectos no se pueden planear y administrar, cuando no se lleva un control de los requisitos y versiones del proyecto. El éxito depende del esfuerzo individual y heroico.

No existen áreas clave de proceso en este nivel

Nivel 2

En el nivel 2 se realiza la implementación de los procesos básicos de la Administración de proyectos, control de requisitos y versiones del producto, con el objetivo de controlar los costos, la calendarización del trabajo, y su funcionalidad. La disciplina necesaria del proceso es repetir éxitos recientes en proyectos con aplicaciones similares.

Las áreas clave de proceso en el nivel 2 son:

- Administración de Requisitos.
- Planeación del Proyectos de Software.
- Seguimiento y Control del Proyecto de Software.
- Administración de Subcontratos de Software.
- Aseguramiento de Calidad de Software.
- Administración de Configuraciones de Software.

Nivel 3

En este nivel la organización define el proceso de software de la organización, las metodologías y herramientas estándar que se usarán en todos los proyectos que comience la organización.

El proceso de software para la administración y las actividades de ingeniería de software es documentado e integrado en un proceso de software estándar para la organización.

Todos los proyectos usan una versión aprobada, ajustada del proceso de software estándar de la organización para desarrollar y mantener software.

Las áreas clave de proceso en el nivel 3 son:

- Enfoque en el Procesos de la Organización.
- Definición del Proceso de la Organización.
- Programa de Capacitación.
- Administración Integrada de Software.
- Ingeniería de Productos de Software.
- Coordinación Entre Grupos.
- Revisiones Entre Colegas.

Nivel 4

Las áreas clave de proceso en el nivel 4 se enfocan en establecer un entendimiento cuantitativo tanto del proceso de software como de los productos que están siendo contruidos. Estas áreas clave son:

- Administración Cuantitativa de Procesos.
- Administración de Calidad de Software.

Nivel 5

Las áreas clave de proceso en el nivel 5 cubren los temas que tanto la organización como los proyectos deben atender para implementar el mejoramiento continuo. El mejoramiento continuo del proceso es habilitado por retroalimentación cuantitativa del proceso y de las ideas innovadoras y tecnológicas. Estas áreas clave son:

- Prevención de Defectos.
- Administración del Cambio Tecnológico.
- Administración del Cambio de Proceso.

Para alcanzar un nivel se deben alcanzar las áreas clave de proceso de ese nivel y en caso de que aplique, también las de los niveles inferiores.

1.6 Prácticas claves

Las prácticas clave describen la infraestructura y actividades que contribuyen en mayor medida, a la implementación e institucionalización efectiva del área clave de proceso.

El propósito de las prácticas clave es transmitir principios que se pueden aplicar a una gran variedad de proyectos y organizaciones. Es decir, CMM establece “qué” debe hacer una organización, pero deja que está proponga “cómo” lo llevará a cabo.

Cada práctica clave consiste en un enunciado simple, generalmente seguido por una descripción mas detallada, la cual conoce como subpráctica, y en algunos casos también es seguido por ejemplos. Las subprácticas pueden usarse como una guía para interpretar a la práctica clave en forma adecuada.

Como ya hemos mencionado, las prácticas claves, de cada área clave de procesos, están organizadas en 5 características comunes, las cuales indican si la implementación y la institucionalización de un proceso es efectivo, repetible y duradero. Estas características comunes son:

- **Compromisos:** Describen las acciones que la organización debe llevar a cabo para garantizar que el proceso sea establecido y duradero. En los compromisos generalmente se establecen las políticas de organización.
- **Habilidades:** Describen las precondiciones que deben existir en el proyecto u organización, para implementar el proceso de software en forma competente. Las habilidades típicamente se enfocan en los recursos, estructuras organizacionales, capacitación y orientación sobre el proceso.
- **Actividades:** Describen las actividades, roles y procedimientos necesarios para implementar un área clave de procesos. Las actividades involucran establecer planes y documentarlos, procedimientos, hacer revisiones periódicas y tomar acciones correctivas en caso de que sea necesario.
- **Mediciones y análisis:** Las mediciones y análisis se usan para determinar el estado del proceso. Estas mediciones se usarán para controlar y mejorar el proceso.
- **Verificaciones:** Las verificaciones se realizan para garantizar que las actividades se realicen de acuerdo al proceso estándar de la organización. Las verificaciones se realizan por medio de revisiones y auditorias hechas por la Administración Gerencial y el Grupo de aseguramiento de la calidad.

Capítulo 2 Nivel 3 del Modelo de Madurez de Capacidades (CMM): Definido

2.1 Áreas claves de proceso del Nivel 3 Definido:

El objetivo del nivel 3 de CMM será que la organización defina un conjunto de procesos, metodologías y herramientas estándar que serán aplicadas a todos los proyectos de software. Al final de este nivel los procesos deberán estar bien documentados y definidos, y todo el personal debió haber recibido la capacitación necesaria para aplicar el proceso estándar de la organización.

En los siguientes capítulos se tratarán con detalle cada una de las áreas claves del nivel 3, En este capítulo describiremos brevemente cada una de éstas.

Las áreas clave de procesos del Nivel 3 definido son:

Enfoque en el Proceso de la Organización (EPO). El objetivo de esta área clave de proceso será asignar las responsabilidades en la organización, para llevar a cabo las actividades de desarrollo, mantenimiento, evaluación y mejoramiento del proceso de software de la organización.

Definición del Proceso de la Organización (DPO). El objetivo de esta área clave de proceso es desarrollar y mantener los procesos de software de la organización así como los componentes del proceso de software como son: los ciclos de vida del proyecto, guías, bases de datos de los procesos de software y otros documentos relacionados con el proceso de software.

Programa de Capacitación (PC). El objetivo de esta área clave de proceso es que el personal desarrolle las habilidades y el conocimiento en cada proceso que aplica a su trabajo y realice su trabajo con eficiencia.

Administración Integrada de Software (AIS). El objetivo de esta área clave de proceso, es unir las actividades de ingeniería de software con las de administración, en un proceso definido y que será ajustado según el proceso de software de la organización.

Ingeniería de Productos de Software (IPS). El objetivo de esta área clave de proceso es realizar las tareas de ingeniería de software para desarrollar y mantener el software. Estas tareas son: analizar los requisitos, desarrollar los requisitos, diseñar el software, la implementación del código fuente, integrar los componentes y realizar las pruebas para verificar que cumplen y satisfacen las necesidades de los clientes.

Coordinación Entre Grupos (CEG). El objetivo de esta área clave de proceso será establecer los mecanismos para que el grupo de ingeniería de software tenga una comunicación activa con los demás grupos de ingeniería, para que el proyecto mejore y cumpla las necesidades del cliente eficientemente.

Revisiones Entre Colegas (REC). El objetivo de está área clave de proceso es llevar a cabo una revisión y evaluación de los productos de software para identificar los defectos lo mas pronto posible y eliminarlos, así como identificar las áreas en donde los cambios son necesarios.

La figura 2.1 es un diagrama de actividades que muestra la forma en que se realizará este trabajo de tesis, en cada capítulo se escribirá la traducción correspondiente de cada área clave de proceso y después se procederá con la elaboración del modelo gráfico de las características comunes de cada área clave.

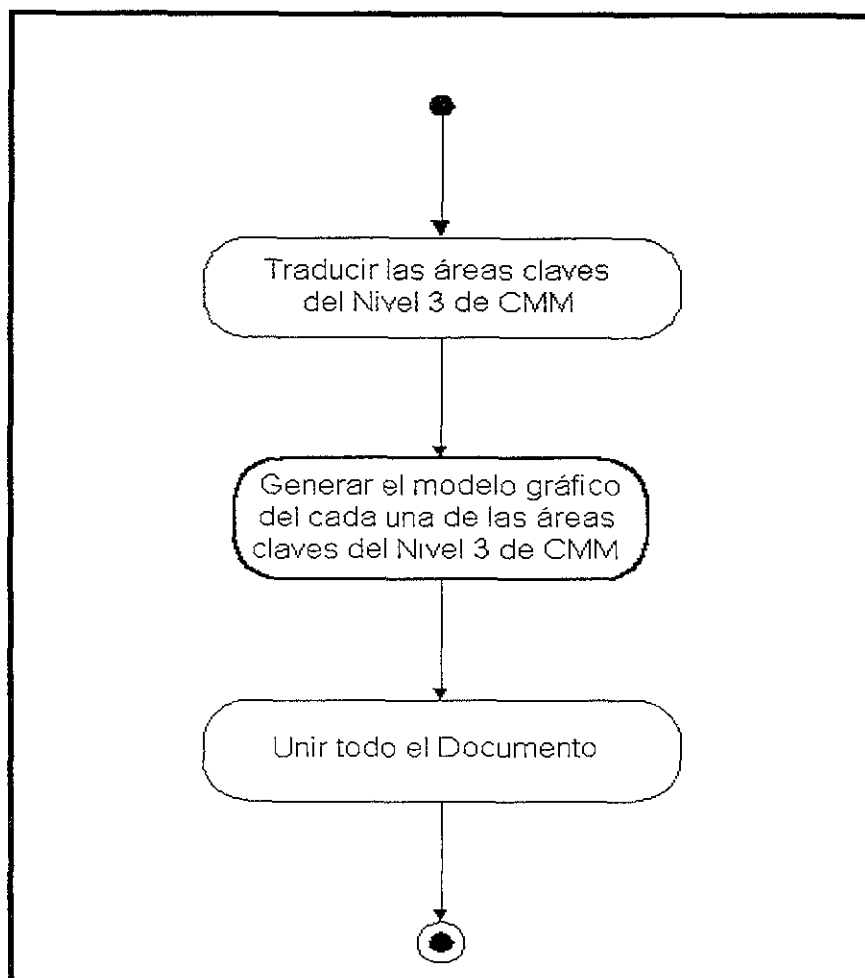


Figura 2.1 Elaboración de Tesis

La actividad Generar el modelo gráfico de cada una de las áreas claves del Nivel 3 de CMM se descompone en la figura 2.2.

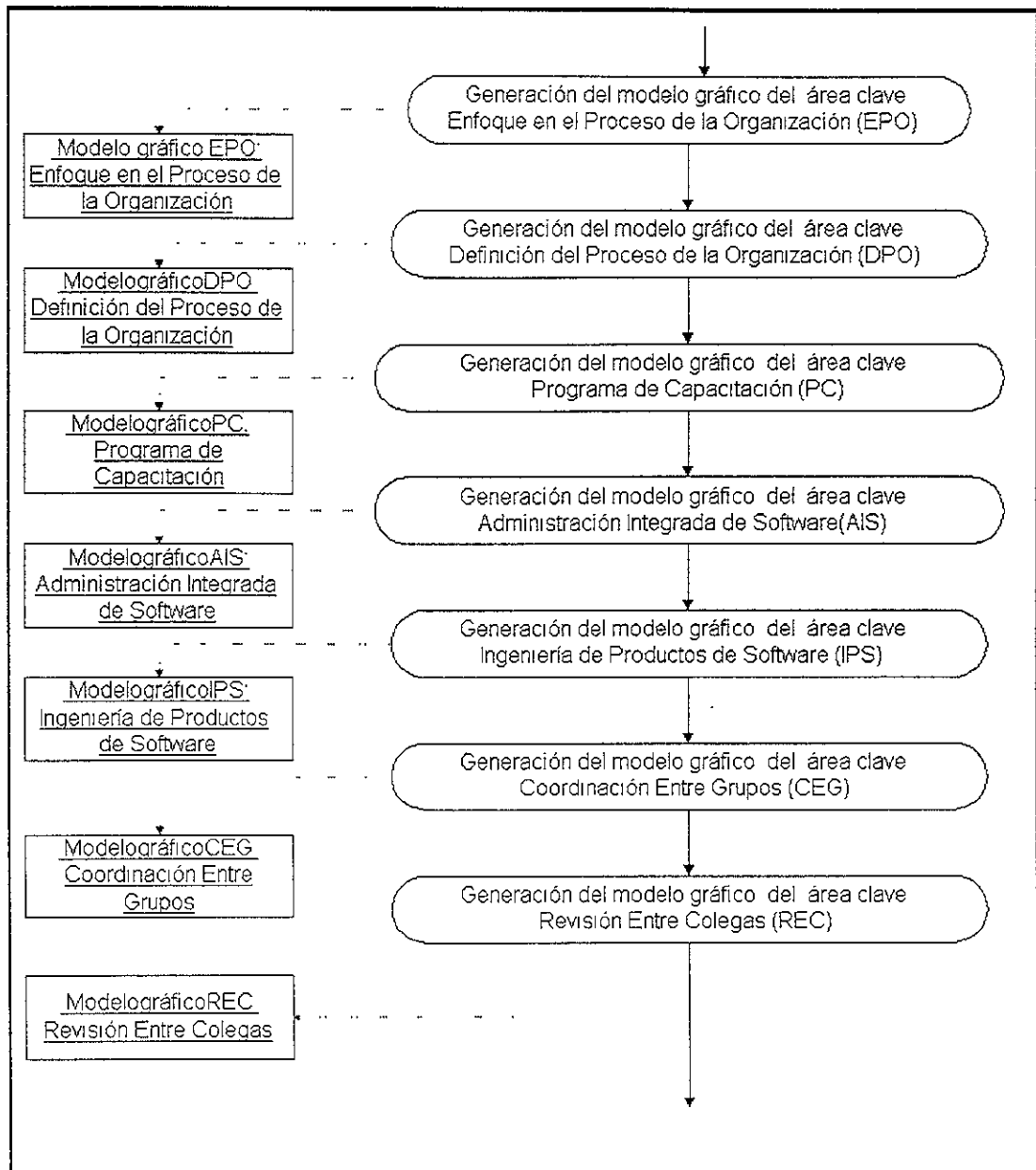


Figura 2.2 Generar el modelo gráfico de cada una de las áreas claves del Nivel 3 de CMM

2.2 Metodología para el desarrollo del modelo gráfico

Para desarrollar el modelo gráfico de cada área de procesos, el primer paso consistió en traducir todas las áreas claves de procesos del Nivel 3 de CMM al español, ya que no se cuenta con ninguna traducción de CMM al español.

El segundo paso fue analizar el texto e identificar los grupos que participan en cada área clave de procesos, así como las actividades que se deben de implementar, los objetos informativos que se producen en cada área de proceso, como son los planes, procedimientos, políticas, documentación relacionada al proceso de software de la organización, etc.

Una vez que se han identificado los grupos participantes, los objetos informativos y las actividades, se empieza con la elaboración de los diagramas de actividades, para mostrar las relaciones de flujo de control y las relaciones de flujo de objetos de información.

Para nuestro análisis dividimos el texto original de cada área clave de procesos en tres partes fundamentales.

- 1) El contexto: El contexto es una introducción que viene en cada área clave de proceso, que nos dice el propósito de cada área clave, que es lo que involucra y algunas recomendaciones que facilitaran la comprensión de cada área clave de procesos.
- 2) Las Metas: Estas nos dicen lo que la organización debe lograr al terminar de implantar cada área clave de procesos.
- 3) Las prácticas: Como se ha mencionado anteriormente en este trabajo las prácticas se organizan en 5 características comunes, que son compromisos, habilidades, actividades, medición y análisis, verificaciones. Cada una de estas características comunes es modelada en cada una de las áreas claves de procesos del Nivel 3. Las prácticas indican a la organización que es lo que debe llevar a cabo para lograr las metas del área clave de proceso.

El modelo gráfico comienza con una identificación en cada área de procesos, de los grupos involucrados, así como una descripción de su rol y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades. Cada diagrama de actividades tiene una explicación de cómo se deben seguir las actividades, quienes las realizan así como los objetos informativos que se generan después de cada actividad.

Antes de elaborar los diagramas de actividades del CMM nivel tres se hizo como ejercicio el modelado de la área clave de procesos Administración de Subcontratos de Software del Nivel 2 para comprender el proceso de cómo se iba a trabajar en las demás áreas claves de procesos.

Los capítulos del 3 al 9 están estructurados en dos partes [ver figura 1], la primera parte es la traducción de cada área clave de procesos del CMM. La segunda parte es el modelo gráfico de la área clave de procesos. Este inicia con una identificación en todo el texto original de los grupos participantes en la área clave de proceso y una descripción de cada grupo participante. Luego se comenzará con la explicación de los diagramas de actividades de cada una de las características comunes. Durante la elaboración del modelo gráfico se siguieron cada una de las siguientes convenciones.

1. En el texto original se subrayaron a los grupos participantes en el proceso.
2. Si un diagrama de actividades que está contenido en una figura se desglosa de una misma actividad que pertenece a otro diagrama, entonces el nombre de la figura es el mismo que el de la actividad.
3. Denotamos las características comunes con una letra mayúscula y un número. La letra mayúscula representa la practicas clave, por ejemplo Compromisos usará la letra C, Habilidades la letra H, Actividades la letra A, Mediciones y análisis la letra M y verificaciones con la letra V, y el número indicará la actividad correspondiente.

Ejemplo: para representar la Habilidad 1.

Constituir el grupo de
capacitación **H1**

4. Si una actividad representa más de una práctica clave, se usa una notación que propuso en [4tas. Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software cuyas autoras son Dra. Hanna y Claudia Esquivel], en la cual la actividad a desglosar tendrá mas grueso el contorno de la elipse que todas las demás actividades. Esto indicará que esta actividad se descompone en otro diagrama que muestra las actividades que están contenidas en otra práctica clave. Por ejemplo aquí la actividad 3 “Proceder a la capacitación de acuerdo al plan” será descompuesta en otro diagrama que nos mostrará que actividades están contenidas en esta actividad.

Proceder a la
capacitación de
acuerdo al plan **A3**

Capítulo 3 Enfoque en el Proceso de la Organización

En el Enfoque en el Proceso de la Organización se asignan las responsabilidades en la organización, para llevar a cabo las actividades de desarrollo, mantenimiento, evaluación y mejoramiento del proceso de software de la organización.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico del área clave Enfoque en el Proceso de la Organización que abstrae la información contenida en el texto.

El modelo gráfico incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

3.1 Enfoque en el Proceso de la Organización

Un área clave de proceso para el nivel 3: Definido

El propósito del Enfoque en el Proceso de la Organización, es establecer la responsabilidad organizacional por las actividades del proceso de software que mejoran la capacidad global de los procesos de software de la organización.

El Enfoque en el Proceso de la Organización involucra desarrollar y mantener una comprensión de los procesos de software de la organización y de los proyectos, además de coordinar las actividades para evaluar, desarrollar, mantener y mejorar estos procesos.

La organización provee compromisos y recursos a largo plazo para coordinar el desarrollo y mantenimiento de los procesos de software durante los proyectos de software actuales y futuros por medio de un grupo llamado el grupo de Ingeniería de procesos de software. Este grupo es responsable por las actividades de los procesos de la organización de software. Es específicamente responsable por el desarrollo y mantenimiento del proceso de software estándar de la organización y los elementos de proceso relacionados, (como se describe en el área clave de procesos Definición del Proceso de la Organización). Así como coordinar las actividades del proceso con los proyectos de software.

Metas

- | | |
|---------------|--|
| Meta 1 | Las actividades de desarrollo y mejoramiento del proceso de software son coordinadas por la organización. |
| Meta 2 | Las fortalezas y debilidades del uso los procesos de software se identifican con relación a un proceso estándar. |
| Meta 3 | Las actividades de desarrollo y mejoramiento del proceso son planeadas a nivel de la organización. |

Compromisos a realizar

- | | |
|---------------------|--|
| Compromiso 1 | <u>La organización</u> sigue una política escrita para coordinar las actividades de desarrollo y mejoramiento del proceso de software en toda la organización. |
|---------------------|--|

Esta política típicamente especifica:

1. Se establece un grupo que es responsable de las actividades del proceso de software y de coordinar estas actividades con los proyectos.
2. Los procesos de software usados por los proyectos son evaluados periódicamente para determinar sus fortalezas y debilidades.
3. Los procesos de software usados por los proyectos son ajustes apropiados del proceso de software estándar de la organización.

Refiérase a la actividad 1 del área clave de proceso Administración Integrada de Software para las prácticas que cubren el ajuste de los procesos de software estándar de la organización.

4. Las mejoras y la información útil acerca de los procesos de software de cada proyecto, las herramientas y métodos están disponibles para otros proyectos.

Compromiso 2

El Administrador Gerencial garantiza las actividades para el desarrollo y mejoramiento del proceso de software de la organización.

El Administrador Gerencial:

1. Demuestra al personal y los gerentes de la organización su compromiso con estas actividades del proceso de software.
2. Establece planes y compromisos a largo plazo para proveer financiamiento, personal y otros recursos.
3. Establece estrategias para administrar e implementar las actividades para el desarrollo y mejoramiento del proceso.

Compromiso 3

El Administrador Gerencial supervisa las actividades para el desarrollo y mejoramiento de los procesos de software de la organización.

El Administrador Gerencial:

1. Asegura que el proceso de software estándar de la organización apoya sus metas comerciales y estrategias.
2. Aconseja en el establecimiento de prioridades para el desarrollo y mejoramiento del proceso de software.
3. Participa en la definición de planes para el desarrollo y mejoramiento del proceso de software.
 - El Administrador Gerencial coordina los requerimientos y problemas del proceso de software con el personal de mas alto nivel y los gerentes.
 - El Administrador Gerencial coordina a los gerentes de la organización para afianzar el apoyo y la participación de ellos y del personal.

Habilidades a realizar

Habilidad 1

Existe un grupo que es responsable por las actividades del proceso de software de la organización.

Un grupo es una colección de departamentos, gerentes e individuos que tiene responsabilidades para un conjunto de tareas o actividades. Un grupo varia desde una persona asignada de tiempo parcial, varias personas asignadas de tiempo parcial de varios departamentos, hasta varias personas dedicadas de tiempo completo. Las consideraciones que debemos tener cuando se implementa un grupo incluyen las tareas y actividades asignadas, el tamaño del proyecto, la estructura y cultura de la organización. Algunos grupos como el de aseguramiento de la calidad de software se enfocan en las actividades del proyecto y otros, como el grupo de ingeniería de procesos de software, se enfocan en las actividades de toda la organización.

1. Donde sea posible este grupo está formado por una selección de profesionales técnicos que son asignados tiempo completo al grupo y posiblemente apoyados por otros de medio tiempo.

El ejemplo mas común de este grupo es el Grupo de Ingeniería de Procesos de Software (Grupo IPS).

2. Este grupo está formado por personal que representa la disciplina de ingeniería de software y otras disciplinas que están relacionadas con el software.

Ejemplos de disciplinas de ingeniería de software y otras disciplinas relacionadas incluyen:

- análisis de los requerimientos del software,
- diseño de software,
- codificación,
- pruebas de software,
- administración de la configuración de software, y
- aseguramiento de la calidad de software.

Habilidad 2

Se proveen los recursos y el financiamiento adecuado para las actividades del proceso de software de la organización.

1. Se comprometen personas expertas en áreas específicas para apoyar a este grupo.

Ejemplos de áreas específicas

- reuso de software,
- tecnologías de ingeniería de software asistida por computadora,
- medición, y
- desarrollo de cursos de capacitación.

Están disponibles las herramientas para apoyar las actividades del proceso de software de la organización.

Ejemplos de herramientas de apoyo incluyen:

- herramientas de análisis estadístico,
- herramientas de edición,

- sistemas manejadores de base de datos, y
- herramientas de modelado de procesos.

Habilidad 3

Los miembros del grupo responsable de las actividades del proceso de software de la organización reciben la capacitación requerida para realizar estas actividades.

Ejemplos de capacitación incluyen:

- prácticas de ingeniería de software;
- técnicas de control de procesos;
- administración del cambio de la organización;
- planeación, administración y monitoreo del proceso de software; y
- cambio tecnológico.

Refiérase al área clave de proceso del Programa de Capacitación.

Habilidad 4

Los miembros del grupo de ingeniería de software y otros grupos relacionados con el software reciben orientación sobre las actividades del proceso de software de la organización así como sus roles en estas actividades.

Refiérase al área clave de proceso del Programa de Capacitación.

Actividades a realizar

Actividad 1

El proceso de software es evaluado periódicamente y los planes de acción son desarrollados para responder a los hallazgos de la evaluación.

Las evaluaciones son típicamente conducidas cada 1-1/2 a 3 años.

Las evaluaciones abarcan todo el proceso de software utilizado en la organización, pero también se pueden muestrear las áreas de proceso y los proyectos.

Un ejemplo de un método para evaluar la capacidad del proceso de software de la organización es el método de evaluación del proceso de software.

El plan de acción identifica:

- cuales hallazgos de la evaluación serán abordados,
- guías para la implementación de los cambios resultantes de hallazgos, y
- los grupos o individuos responsables por implementar los cambios.

Actividad 2

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

Este plan:

1. Usa como entradas principales los planes de acción obtenidos de las evaluaciones de proceso de software y otras iniciativas de mejora de la organización.
2. Define las actividades a ser realizadas y el programa para estas actividades.
3. Especifica los grupos e individuos responsable para estas actividades.
4. Identifica los recursos requeridos, incluyendo personal y herramientas.
5. Se somete a revisiones entre colegas en la liberación inicial y siempre y cuando se hagan revisiones mayores.

Refiérase al área clave de procesos de Revisiones Entre Colegas.

6. Es revisado y acordado por los gerentes de software y el Administrador Gerencial de la organización.

Actividad 3

Las actividades para el desarrollo y mejoramiento de los procesos de software de los proyectos y de la organización son coordinadas a nivel de la organización.

Esta coordinación cubre el desarrollo y mejoramiento de:

1. Los procesos de software estándar de la organización.

Refiérase a las actividades 1 y 2 del área clave de procesos de la Definición del Proceso de la Organización para prácticas que cubren los procesos estándar de la organización.

2. Los procesos de software definidos de los proyectos.

Refiérase a las actividades 1 y 2 del área clave de procesos de la Administración Integrada de Software para prácticas que cubren los procesos de software definidos de los proyectos.

Actividad 4

El uso de la base de datos del proceso de software de la organización es coordinado a nivel organizacional.

La base de datos del proceso de software de la organización es usada para recolectar información sobre los procesos de software y el producto de software resultante de la organización y de los proyectos.

Refiérase a la actividad 5 del área clave de procesos de la Definición del Proceso de la Organización para prácticas que cubren la base de datos de los procesos de software de la organización.

Actividad 5

Los nuevos procesos, métodos y herramientas de uso limitado en la organización son monitoreados, evaluados y cuando sea apropiado, transferidos a otras partes de la organización.

Actividad 6

La capacitación en los procesos de software de los proyectos y de la organización es coordinada a través de la organización.

1. Se preparan los planes para la capacitación sobre temas relacionados con los procesos de software de los proyectos y de la organización.
2. Cuando sea apropiado la capacitación puede ser preparada y conducida por el grupo responsable de las actividades del proceso de software de la organización.

Refiérase a la área clave de procesos de Programa de Capacitación.

Actividad 7

Los grupos involucrados en la implementación de los procesos de software son informados de las actividades para el desarrollo y mejoramiento del proceso de software de los proyectos y de la organización.

Ejemplos de medios para informar e involucrar a estas personas incluyen:

- boletines de anuncios electrónicos sobre los procesos,
- asesorías sobre los procesos,
- grupos de trabajo,
- reuniones de intercambio de información,
- encuestas,
- equipos de mejoras de procesos, y
- discusiones informales.

Medición y análisis

Medición 1

Se hacen y se usan las mediciones para determinar el estado de las actividades de desarrollo y mejoramiento de los procesos de la organización.

Ejemplos de estas mediciones incluyen:

- el trabajo terminado, el esfuerzo y el capital gastado para las actividades de evaluación, desarrollo y mejoramiento de los procesos de la organización, comparado con los planes de tales actividades; y
- la evaluación de cada proceso de software, comparado con los resultados y las recomendaciones de las evaluaciones previas.

Verificación de la implementación

Verificación 1

Las actividades de desarrollo y mejoramiento del proceso de software son revisadas periódicamente con el Administrador Gerencial.

El propósito principal de las revisiones periódicas por el Administrador Gerencial es proveer una conciencia y visión de las actividades del proceso de software en un nivel apropiado de abstracción y de una manera oportuna. El tiempo entre las revisiones debe estar de acuerdo a las necesidades de la organización y puede prolongarse hasta que estén disponibles los mecanismos adecuados para el reporte de excepciones.

1. El progreso y el estado de las actividades para desarrollar y mejorar el proceso de software es revisado contra el plan.
2. Los conflictos y problemas no resueltos en los niveles inferiores se abordan.
3. Se asignan acciones a revisar, se revisan y se siguen hasta concluirlos.
4. Se prepara un resumen informativo de cada reunión y se distribuye a los grupos o individuos afectados.

3.2 Modelo gráfico del Enfoque en el Proceso de la Organización (EPO)

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en el proceso de EPO y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

3.3 Identificación de los grupos participantes

En el modelo textual del Enfoque en el Proceso de la Organización hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentamos una breve definición de cada uno de ellos.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de varios proyectos a la vez.

Grupo de Ingeniería de Procesos de Software (Grupo de IPS): Un grupo de especialistas que facilitan la definición, mantenimiento y mejora del proceso de software usado por la organización. En las prácticas clave, este grupo es genéricamente llamado “ el grupo responsable por las actividades de proceso de software de la organización”.

Grupo de Ingeniería de Software (Grupo de IS): Es la colección de individuos (tanto administradores, como personal técnico), que tienen la responsabilidad para desempeñar las actividades de mantenimiento y desarrollo de software (como análisis de requisitos, diseño, codificación y pruebas) para el proyecto. Los grupos relacionados al software, tal como el grupo de aseguramiento de calidad, el grupo de administración de configuraciones, el grupo de ingeniería de procesos de software, no están incluidos en el grupo de ingeniería de software.

Otros grupos relacionados: Es cualquier grupo involucrado en los procesos de software de la organización.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

Evaluador: Es el responsable de evaluar periódicamente el proceso de software.

3.4 Diagrama de actividades

Compromisos.

En esta área clave de procesos contamos con tres compromisos, el primero establece que la organización sigue una política organizacional escrita para coordinar las actividades de desarrollo y mejoramiento del proceso de software de la organización(PSO). El segundo establece que el administrador gerencial debe garantizar las actividades para el desarrollo y mejoramiento del PSO. El tercero establece que el administrador gerencial supervisa las actividades del PSO.

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que sirve para coordinar las actividades de desarrollo y mejoramiento de procesos.

La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 3.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para coordinar las actividades de desarrollo y mejoramiento del proceso de software queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Coordinar el Desarrollo y mejora del PSO”.

El compromiso 2 establece que el administrador gerencial debe garantizar las actividades para el desarrollo y mejoramiento del proceso de software. Este compromiso lo representamos como “Garantizar las actividades para el desarrollo y mejoramiento del PSO” que de acuerdo al texto es realizada por el Administrador Gerencial.

El compromiso 3 establece que el administrador gerencial supervisa las actividades. Este compromiso lo representamos como “Supervisar las actividades para el desarrollo y mejoramiento del PSO” que de acuerdo al texto es realizada por el Administrador Gerencial.

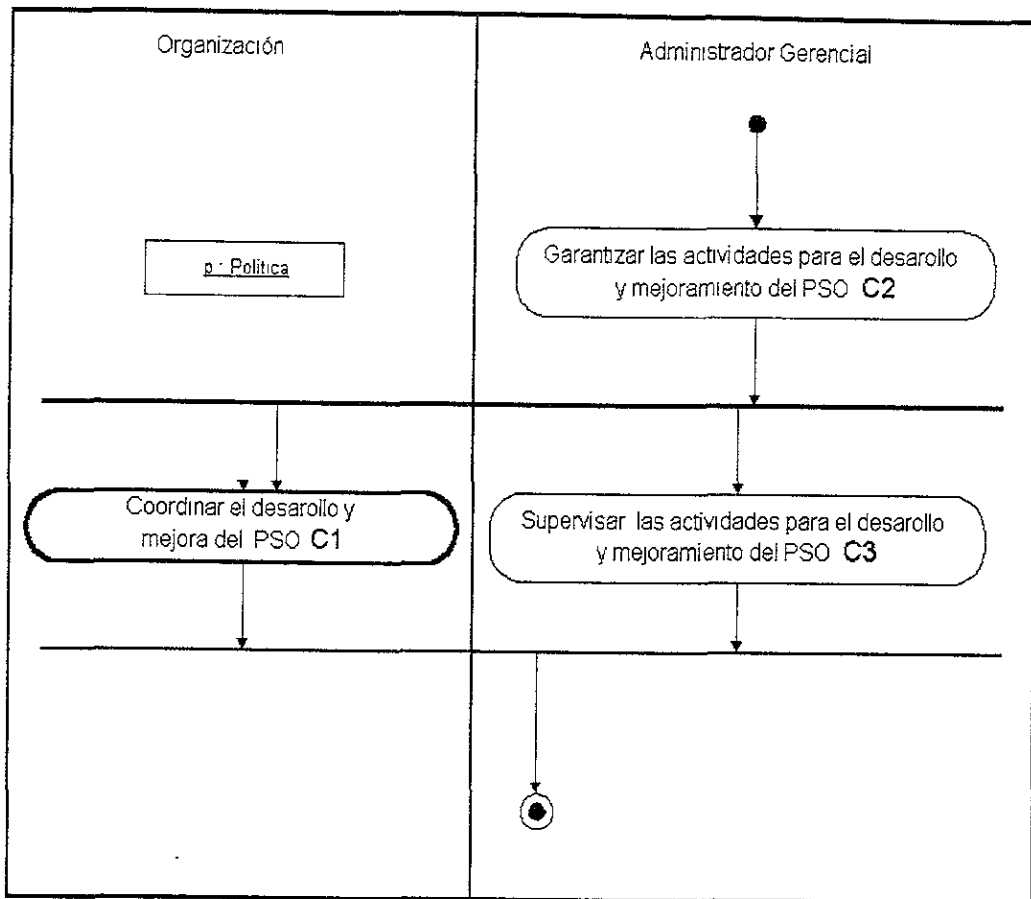


Figura 3.1 Compromisos del área clave de proceso: Enfoque en el Proceso de la Organización (EPO).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para Coordinar el desarrollo y mejora del Proceso de software de la Organización.

Para modelar este hecho, la actividad correspondiente se desglosa en otro diagrama presentado en la Figura 3.2.

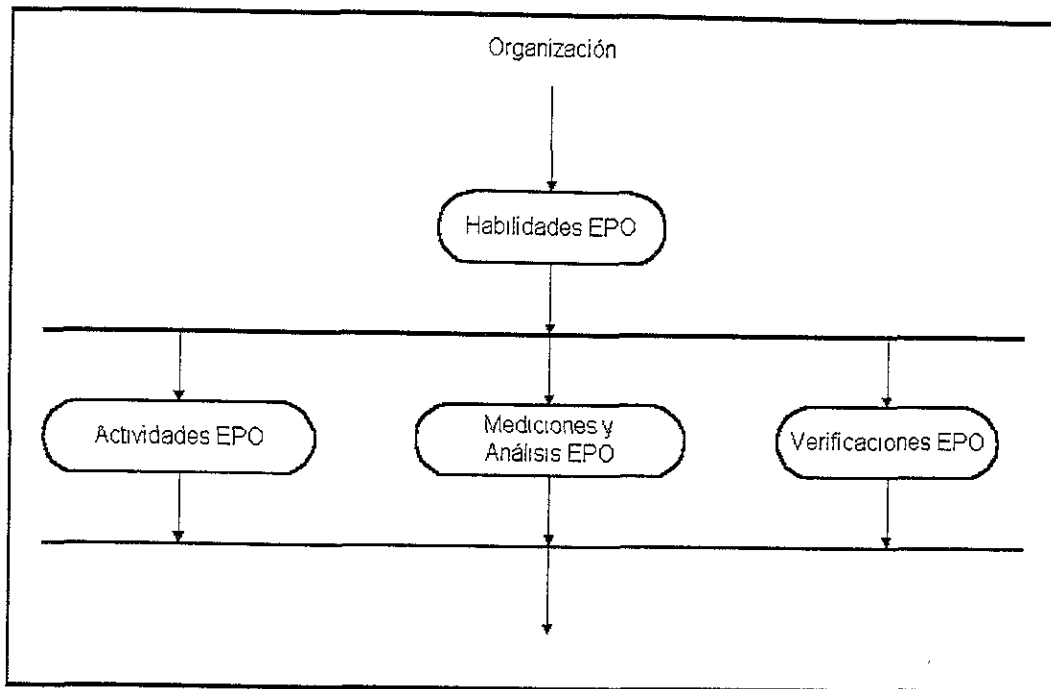


Figura 3.2 Coordinar el desarrollo y mejora del Proceso de software de la Organización (PSO).

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y/o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias del Enfoque en el Proceso de la Organización. Por otro lado, las actividades de verificación y medición con su análisis, son simultáneas a las de EPO.

Habilidades.

La habilidad 1 especifica que debe haber un grupo responsable de las actividades del proceso de software de la organización. Si no existe el grupo se debe de crear. Generalmente este grupo es el Grupo de IPS.

La habilidad 2 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo de Grupo de IPS, Grupo de IS y Otros grupos relacionados.

Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación u orientación.

Las dos últimas habilidades tratan la capacitación y orientación de los grupos involucrados en este proceso. La habilidad 3 especifica que el Grupo de IPS, el Grupo de IS y Otros grupos relacionados, reciben la capacitación requerida para las

actividades del proceso de software de la organización. Esta habilidad la representaremos con la actividad “Recibir capacitación”.

La habilidad 4 especifica que el Grupo de IPS, el Grupo de IS y Otros grupos relacionados. Reciben la orientación requerida sobre las actividades del proceso de software de la organización. Esta habilidad la representaremos con la actividad “Recibir orientación”.

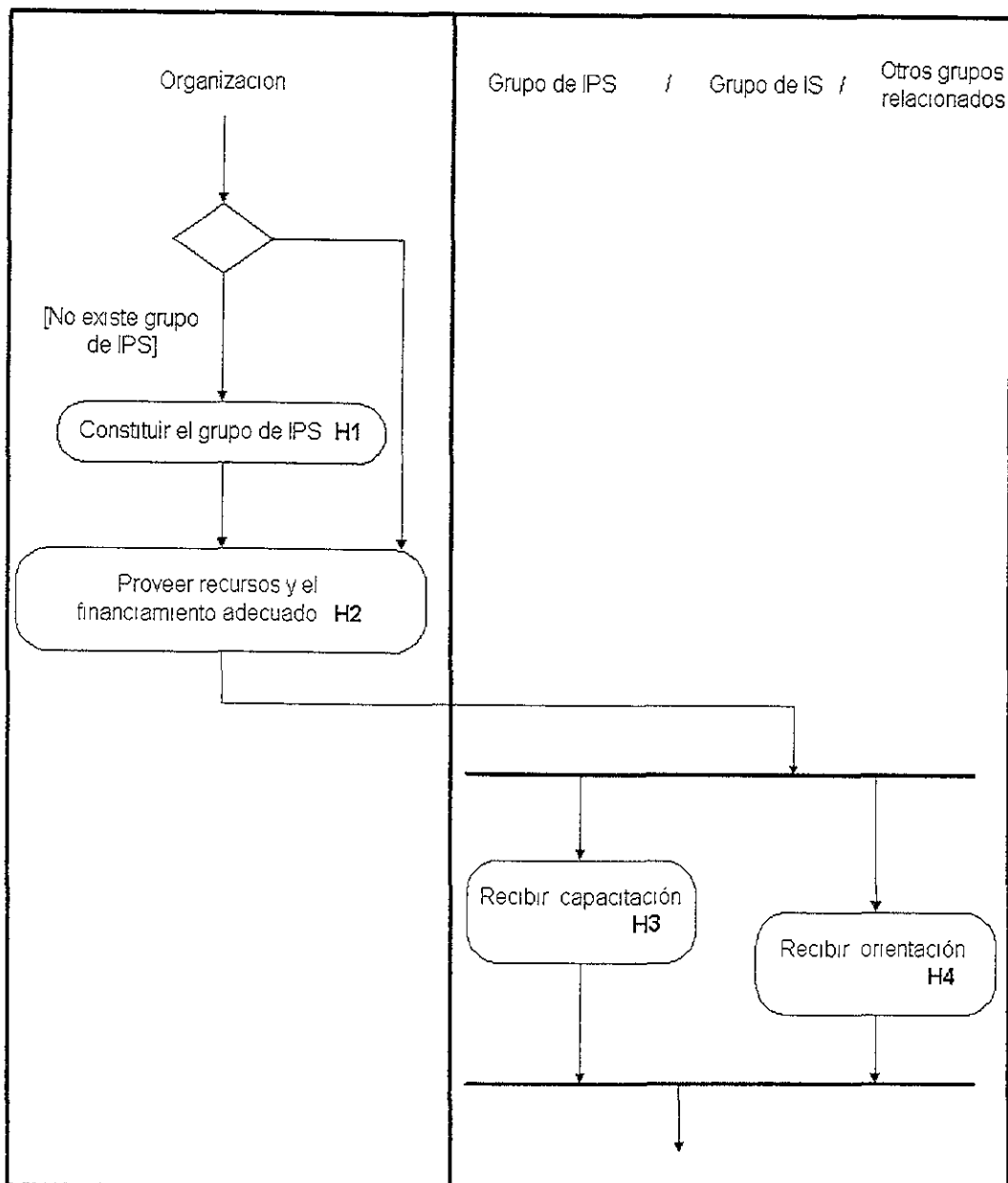


Figura 3.3 Habilidades del área clave de proceso: Enfoque en el Proceso de la Organización (EPO).

El diagrama de la Figura 3.3 presenta las actividades a realizarse para adquirir las habilidades para el EPO, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde. También en este diagrama es fácil observar que actividades pueden llevarse a cabo en forma paralela y el orden en que se realizan.

Actividades.

La actividad 1 especifica que el proceso de software es evaluado periódicamente y se deben desarrollar planes de acción para responder a los hallazgos de la evaluación. Esta actividad la representamos en el diagrama como “Evaluar periódicamente el proceso de software” Como en el texto no dice quien hace la evaluación nosotros proponemos introducir un rol de Evaluador. Esta actividad recibe como entrada al objeto *ps: Proceso de software*, el cual va a ser evaluado. Al término de esta actividad tendremos como salida a *pla: Planes de acción* que servirán para responder a los hallazgos de la evaluación.

La actividad 2 especifica que la organización desarrolla y mantiene un plan para las actividades de desarrollo y mejoramiento de su proceso de software.

Esta actividad la representamos como “Desarrollar y mantener un plan para las actividades de desarrollo de procesos” que de acuerdo al texto es realizada por la organización. Esta actividad recibe como entrada al objeto *pla: Planes de acción*, obtenidos de las evaluaciones de proceso de software y otras iniciativas de mejora de la organización. Al término de esta actividad tendremos como salida a *pla: Plan de las actividades*.

La actividad 3 especifica que las actividades para el desarrollo y mejoramiento de los procesos de software de los proyectos, son coordinadas por la organización, en forma compartida con el Grupo de IPS. Esta actividad la representamos como “Coordinar las actividades”.

La actividad 4 especifica que el uso de la base de datos del proceso de software de la organización es coordinado por la organización en forma compartida con el Grupo de IPS. Esta actividad la representamos como “Coordinar el uso de la base de datos”.

La actividad 5 especifica que se deben monitorear los nuevos procesos, métodos y herramientas y evaluarlos cuando sea apropiado. Esta actividad la realiza la organización en forma compartida con el Grupo de IPS. Esta actividad la representamos como “Monitorear los nuevos métodos y herramientas”

La actividad 6 especifica que se debe coordinar la capacitación en los procesos de software de los proyectos y de la organización. Esta actividad la realiza la organización en forma compartida con el Grupo de IPS. Esta actividad la representamos como “Coordinar la capacitación en los procesos de software”.

La actividad 7 especifica que los grupos involucrados sobre la implementación de los procesos de software son informados de las actividades para el desarrollo y mejoramiento de los procesos. Esta actividad la realiza la organización en forma compartida con el Grupo de IPS. Esta actividad la representamos como “Informar a los grupos involucrados”.

El diagrama de la Figura 3.4 presenta las actividades que se describieron anteriormente. También en este diagrama es fácil observar que actividades pueden llevarse acabo en forma paralela y el orden en que se realizan.

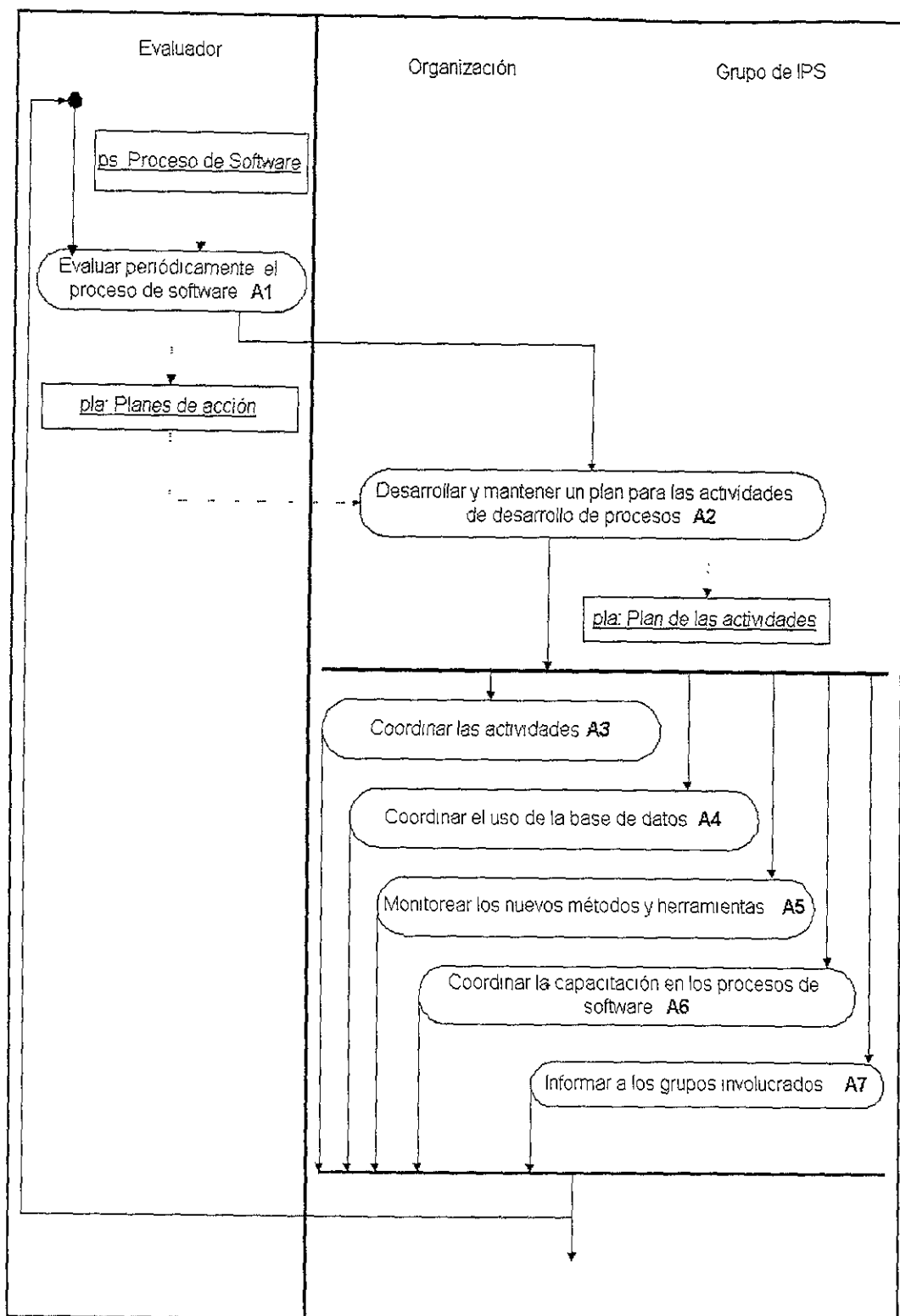


Figura 3.4 Actividades del área clave de proceso: Enfoque en el Proceso de la Organización (EPO).

Todas estas actividades se repiten de manera cíclica de una evaluación a otra.

Mediciones y análisis.

El texto correspondiente dice que deben de recabarse y usar las mediciones para determinar el estado de las actividades del EPO. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad compartida entre el Administrador Gerencial, el Evaluador y el Grupo de IPS, quienes utilizan estos datos para determinar el estado de las actividades del EPO. La actividad mencionada anteriormente queda representada en la figura 3.5

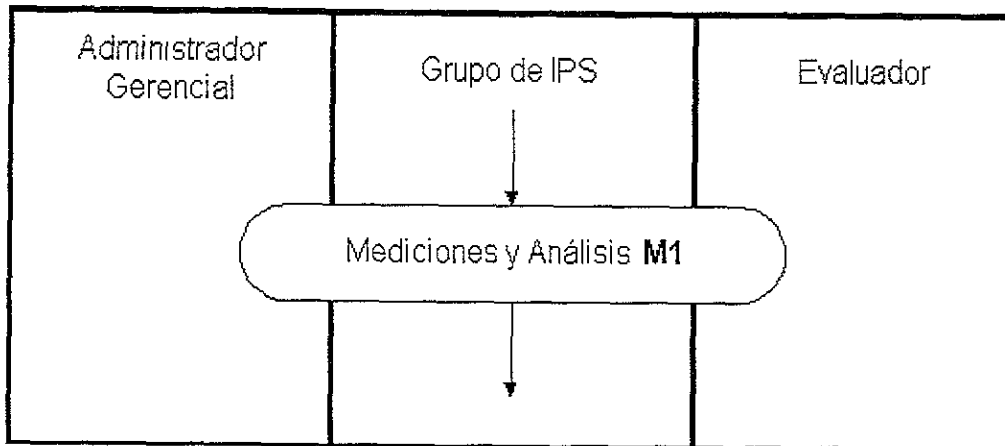


Figura 3.5 Medición y análisis del área clave de proceso: Enfoque en el Proceso de la Organización (EPO).

Verificaciones.

La verificación 1 especifica que las actividades de desarrollo y mejoramiento del proceso de software son revisadas periódicamente con el Administrador Gerencial, también proponemos que en esta actividad participa el Grupo de IPS. Lo cual mostramos en la Figura 3.6

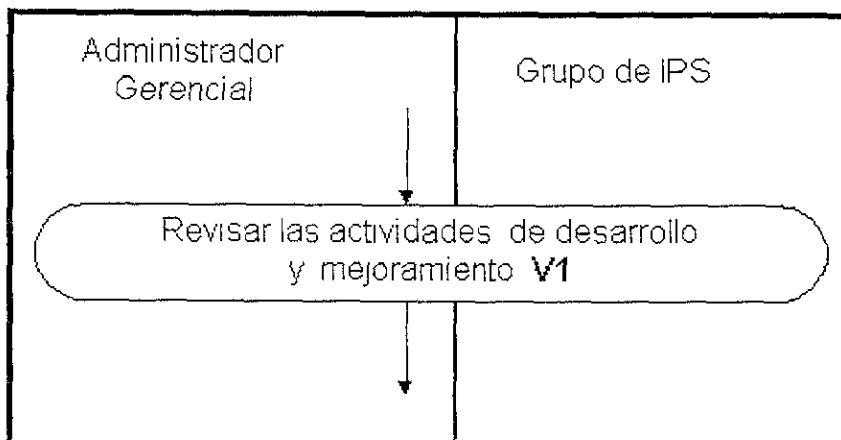


Figura 3.6 Verificaciones del área clave de proceso: Enfoque en el Proceso de la Organización (EPO).

Capítulo 4 Definición del Proceso de la Organización

En la Definición del Proceso de Software se desarrollarán los procesos de software de la organización así como los componentes del proceso de software como son: los ciclos de vida del proyecto, guías, y otros documentos relacionados con el proceso de software.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico del área clave Definición del Proceso de la Organización que abstraer la información contenida en el texto.

El modelo gráfico incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

4.1 Definición del Proceso de la Organización

Un área clave de proceso para el nivel 3: Definido

El propósito de la Definición del Proceso de la Organización es desarrollar y mantener un conjunto útil de elementos del proceso de software que mejoren el desempeño de los procesos a través de los proyectos y proveer una base para beneficios acumulativos y de largo plazo para la organización.

La Definición del Proceso de la Organización involucra desarrollar y mantener los procesos de software estándar de la organización junto con los elementos relacionados al proceso, tales como las descripciones de los ciclos de vida del software, guías y criterios de adaptación de procesos, la base de datos de los procesos de software de la organización y una colección de documentos relacionados al proceso de software.

Estos elementos pueden ser recolectados de varias maneras, dependiendo de la implementación de la Definición del Proceso de la Organización. Por ejemplo la descripción de los ciclos de vida del software puede ser una parte integral del proceso de software estándar de la organización o puede formar parte de una colección de documentos relacionados al proceso de software almacenado en la base de datos del proceso de software de la organización.

Los elementos del proceso de software de la organización están disponibles para su uso en el desarrollo, implementación y mantenimiento de los procesos de software definidos en los proyectos (las prácticas relacionadas con el desarrollo y mantenimiento del proceso de software definido en los proyectos es descrito en la área clave de procesos Administración Integrada de Software).

Metas

- | | |
|---------------|---|
| Meta 1 | Se desarrolla y mantiene un proceso de software estándar para la organización. |
| Meta 2 | La información relacionada con el uso del proceso estándar de la organización por los proyectos de software se recolecta, revisa y pone a disposición. |

Compromisos a realizar

Compromiso 1

La organización sigue una política escrita para desarrollar y mantener un proceso de software estándar y los elementos relacionados al proceso.

Los elementos del proceso de software de la organización incluyen:

- los procesos de software estándar de la organización,
- guías y criterios para el ajuste de los procesos de software,
- estándar para los proyectos de la organización,
- descripciones de los ciclos de vida de software aprobados para su uso,
- base de datos del proceso de software de la organización, y
- una colección de documentos relacionados al proceso de software, previamente desarrollada y disponible para su reuso.

Esta política típicamente especifica que:

1. Se define un proceso de software estándar para la organización.

El propósito principal del proceso estándar de software es compartir al máximo los elementos de proceso y proveer la habilidad para definir a un conjunto estándar de mediciones de proceso de los proyectos a nivel organizacional.

El proceso de software estándar de la organización puede contener múltiples procesos de software. Los múltiples procesos de software se pueden necesitar para responder a las necesidades de las diferentes aplicaciones, ciclos de vida, metodologías y herramientas que los proyectos del software pueden combinar de múltiples maneras.

2. Un proceso de software definido para un proyecto es una versión ajustada del proceso de software estándar de la organización.
3. Se les da mantenimiento a los elementos de proceso de software de la organización.

Refiérase a la actividad 1 de la área clave de proceso Administración Integrada de Software para prácticas que cubren el ajuste del proceso de software estándar de la organización

4. La información recolectada de los proyectos es organizada y usada para mejorar el proceso de software estándar de la organización.

Ejemplos de la información recolectada incluye:

- mediciones de productos y de procesos,
- lecciones aprendidas, y
- otra documentación relacionada con el proceso.

Habilidades a realizar

Habilidad 1

Se proveen los recursos y el financiamiento adecuado para desarrollar y mantener el proceso de software estándar de la organización y los elementos relacionados con los procesos.

1. El desarrollo y mantenimiento del proceso de software estándar y los elementos relacionados con los procesos es realizado o coordinado por el grupo responsable por las actividades del proceso de software de la organización (por ejemplo el Grupo de Ingeniería de Procesos de Software).

Refiérase al área clave de proceso Enfoque en el Proceso de la Organización para prácticas que cubre el grupo responsable por las actividades del proceso de software de la organización.

2. Se proporcionan herramientas para apoyar el desarrollo y mantenimiento del proceso.

Ejemplos de herramientas de apoyo incluyen:

- herramientas de publicación de escritorio,
- sistemas manejadores de base de datos, y
- herramientas de modelado de procesos.

Habilidad 2

Los individuos que desarrollan y mantienen el proceso de software estándar y los elementos relacionados con el proceso reciben la capacitación requerida para realizar estas actividades.

Ejemplos de capacitación incluyen:

- prácticas y métodos de ingeniería de software,
- métodos de análisis y documentación de los procesos, y
- modelado de procesos.

Refiérase a de la área clave de proceso Programa de Capacitación

Actividades a realizar

Actividad 1

El proceso de software estándar de la organización es desarrollado y mantenido de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El proceso de software estándar de la organización satisface las políticas de software, estándares de procesos y de productos que están definidos en la organización, si es apropiado.
2. El proceso de software estándar de la organización satisface los estándares de los procesos y productos que

son comúnmente impuestos sobre los proyectos de la organización por sus clientes, según lo apropiado.

3. Se incorporan métodos y herramientas de ingeniería de software dentro del proceso de software estándar de la organización, según lo apropiado.
4. Se describen las interfaces internas entre las disciplinas de software.

Ejemplos de disciplinas de ingeniería de software incluyen:

- análisis de los requerimientos del software,
- diseño de software,
- codificación,
- pruebas de software,
- administración de la configuración de software, y
- aseguramiento de la calidad de software

5. Se describen las interfaces externas de proceso entre el proceso de software y los procesos de otros grupos afectados.

Ejemplos de otros grupos afectados incluyen:

- ingeniería de sistemas,
- pruebas de sistema,
- administración de contratos, y
- apoyo de documentación.

6. Se documentan, se revisan y se prueban los cambios propuestos para el proceso de software estándar de la organización por el grupo responsable de las actividades del proceso de software estándar de la organización (por ejemplo el grupo de ingeniería de proceso de software) antes de que estos sean incorporados.

Ejemplos del origen de los cambios incluyen:

- los hallazgos y recomendaciones de las evaluaciones del proceso de software.

- los resultados de los ajustes a los proyectos del proceso de software estándar de la organización,
- las lecciones aprendidas de monitorear las actividades de proceso de software de los proyectos y de la organización,
- cambios propuestos por el personal y gerentes de la organización, y
- datos de mediciones de procesos y de productos que son analizados e interpretados.

7. Se definen planes para introducir los cambios en los procesos de software de proyectos en marcha según lo apropiado.
8. La descripción del proceso de software estándar de la organización es sometida a la revisión entre colegas cuando se desarrolla inicialmente y siempre y cuando se hagan cambios o adiciones significativas.

Refiérase al área clave de procesos Revisiones Entre Colegas.

9. La descripción del proceso de software estándar de la Organización es puesta bajo la administración de configuración.

Refiérase al área clave de proceso Administración de Configuración de Software

Actividad 2

El proceso de software estándar de la organización es documentado de acuerdo a los estándares establecidos de la organización.

Estos estándares típicamente especifican que

1. El proceso es descompuesto en sus elementos de procesos constituyentes hasta la granularidad necesaria para comprender y describir los procesos.

Cada elemento de proceso cubre un conjunto de actividades bien definidas, acotadas y estrechamente relacionadas.

Ejemplos de elementos de proceso incluyen:

- elementos de estimación de software,
- elementos de diseño de software,
- elementos de codificación, y
- elementos de revisión entre colegas.

2. Cada elemento de proceso es descrito y abarca:

- ☐ Los procedimientos, prácticas, métodos y tecnologías requeridas;
- ☐ Los estándares aplicables de procesos y productos;
- ☐ Las herramientas y recursos requeridos;
- ☐ Entradas;
- ☐ Los productos de trabajo de software producidos;
- ☐ Los productos de trabajo de software que deben ponerse bajo las revisiones entre colegas;
- ☐ Los criterios de entrega y terminación; y
- ☐ Los datos de productos y de proceso que deben ser recolectados.

3. Se describen las relaciones entre los elementos del proceso y abarcan:

- ☐ el orden,
- ☐ las interfaces, y
- ☐ las interdependencias.

Esta relación entre los elementos del proceso es a veces llamada como la arquitectura del proceso de software.

Actividad 3

Las descripciones de los ciclos de vida del software que son aprobados para su uso por los proyectos son documentadas y mantenidas.

Ejemplos de ciclos de vida incluyen:

- cascada,
- cascada traslapada,
- espiral,
- construcción serial, y
- prototipo simple / cascada traslapada.

1. Los ciclos de vida del software son compatibles con el proceso de software estándar de la organización.
2. Los cambios propuestos para las descripciones de los ciclos de vida son documentados, revisados y aprobados por el grupo responsable de las actividades del proceso de software de la organización (por ejemplo el grupo de ingeniería de procesos de software) antes de que estos sean incorporados.
3. Las descripciones de los ciclos de vida de software son sometidas a revisiones entre colegas cuando se documenta inicialmente y siempre y cuando se hayan hecho cambios o adiciones significativas.

Refiérase a la área clave de procesos de Revisión Entre Colegas.

4. Las descripciones de los ciclos de vida del software son administradas y controladas.

Administrado y controlado implica que la versión del producto de trabajo de software en un determinado momento (pasado o presente) es conocida y los cambios son incorporados y controlados.

Si se desea un mayor grado de control del que es aplicado por el administrado y controlado el producto de trabajo puede ser puesto bajo la disciplina total de la Administración de configuración como es descrita en el área clave de proceso Administración de Configuración de Software.

Actividad 4

Se desarrollan y mantienen guías y criterios para el ajuste de los proyectos del proceso de software estándar de la organización.

1. Las guías y criterios de ajuste cubren:
 - ☐ selección y ajuste de un ciclo de vida de software para el proyecto,
 - ☐ ajustes del proceso de software estándar de la organización para acomodar el ciclo de vida de software y las características del proyecto, y

Ejemplos de ajuste incluyen:

- ajustar el proceso para una nueva línea de productos o ambiente del servidor,
 - ajustar el proceso para un proyecto específico o una clase de proyecto, y
 - elaborar y añadir detalles al proceso de tal modo que el proceso de software definido del proyecto resultante pueda ser ejecutado.
- ☐ Estándares para la documentación del proceso de software definido para el proyecto.
 2. Los cambios propuestos para las guías y criterios de ajuste son documentados y aprobados por el grupo responsable por las actividades del proceso de software de la organización (por ejemplo el grupo de ingeniería de procesos de software) antes de que estos sean integrados.
 3. Las guías y criterios de ajuste son administrados y controlados.

Actividad 5

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

1. La base de datos es establecida para recolectar y hacer disponible datos sobre los procesos de software y los productos de trabajo de software resultantes.

Ejemplos de datos de procesos y productos de trabajo incluyen:

- estimaciones del tamaño, esfuerzo y costo de software;
- datos reales del tamaño, esfuerzo y costo de software;
- datos de productividad;
- mediciones de calidad;
- cobertura y eficiencia de revisiones entre colegas;
- cobertura y eficiencia de pruebas;
- mediciones de confiabilidad de software;
- número y gravedad de los defectos encontrados en los requerimientos del software; y
- número y gravedad de los defectos encontrados en el código del software.

2. Los datos ingresados a la base de datos son revisados para asegurar la integridad del contenido de la base de datos.

Además la base de datos contiene referencias de datos de mediciones e información relacionada y los datos necesarios para comprender e interpretar los datos de mediciones y evaluarlos en términos de lo razonable y aplicabilidad.

3. La base de datos es administrada y controlada.
4. El acceso de usuarios al contenido de la base de datos es controlado para asegurar completamente la integridad y exactitud de los datos.

El acceso es limitado para aquellos que tienen la necesidad de introducir, cambiar, ver, analizar o extraer datos.

Los datos delicados son protegidos y el acceso a estos datos es controlado apropiadamente.

Actividad 6

Se establece y se mantiene una colección de documentos relacionada con el proceso de software.

Ejemplos de documentos relacionados con el software incluyen:

- la descripción de procesos de software definidos en los proyectos,
- estándares de proyectos,
- procedimientos de proyectos,
- planes de desarrollo de software de proyectos,
- planes de medición de proyectos, y
- materiales de capacitación en procesos de proyectos.

1. Los elementos candidatos a la documentación son revisados y los elementos apropiados que puedan ser útiles en el futuro son incluidos en la colección de documentos.
2. Los elementos de la documentación son catalogados para un fácil acceso.
3. Las correcciones hechas a los elementos de la documentación son revisadas y el contenido de la colección de documentos es actualizado según lo adecuado.
4. El contenido de la colección de documentos es disponible para el uso por los proyectos de software y otros grupos de software relacionados.

Ejemplos de grupos de software relacionados incluyen:

- aseguramiento de calidad de software
- administración de configuración de software,
- pruebas de software, y
- apoyo de documentación.

5. El uso de cada elemento de la documentación es revisado periódicamente y los resultados son usados para mantener el contenido de la colección de documentos.
6. El contenido de la colección de documentos es administrado y controlado.

Medición y análisis

Medición 1

Se hacen mediciones y se usan para determinar el estado de las actividades de la Definición del Proceso de la Organización.

Ejemplos de mediciones incluyen:

- el estado de los hitos de la calendarización para el desarrollo y mantenimiento del proceso, y
- costo de las actividades de la definición del proceso.

Verificación de la implementación

Verificación 1

El grupo de aseguramiento de la calidad de software revisa y / o audita las actividades y productos de trabajo de la organización para desarrollar y mantener el proceso de software estándar de la organización y los elementos de proceso relacionados y reporta los resultados.

Refiérase al área clave de proceso Aseguramiento de la Calidad de Software

Como mínimo estas revisiones y /o auditorias verifican que:

1. Se han seguido los estándares apropiados en el desarrollo, documentación y mantenimiento en el proceso de software estándar de la organización y los elementos de proceso relacionados.
2. Los procesos de software estándar de la organización y los elementos de procesos son controlados y usados apropiadamente.

4.2 Modelo gráfico Definición del Proceso de la Organización (DPO)

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en el proceso de PSO y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

4.3 Identificación de los grupos participantes

En el modelo textual de la Definición del Proceso de la Organización (DPO) hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentamos una breve definición de cada uno de ellos.

Grupo de Aseguramiento de la Calidad: Revisa y / o audita las actividades y los productos de trabajo para desarrollar y mantener el proceso de software estándar de la organización y reporta los resultados.

Grupo de Ingeniería de Procesos de Software (Grupo de IPS): Un grupo de especialistas que facilitan la definición, mantenimiento y mejora del proceso de software usado por la organización. En las prácticas clave, este grupo es genéricamente llamado “ el grupo responsable por las actividades de proceso de software de la organización”.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

4.4 Diagrama de actividades

Compromisos.

En esta área clave de procesos contamos con un compromiso, este establece que la organización sigue una política organizacional escrita para desarrollar y mantener un Proceso de Software Estándar (PSE).

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que sirve para desarrollar y mantener el PSE. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 4.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para desarrollar y mantener el PSE queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Desarrollar y mantener un PSE C1”

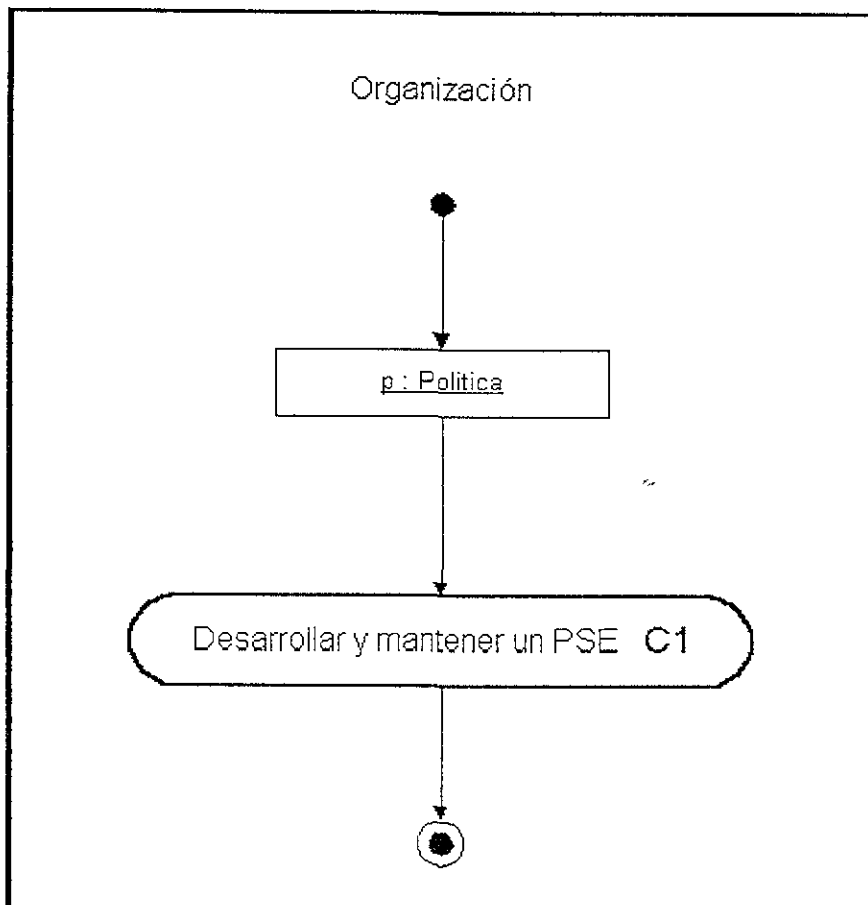


Figura 4.1 Compromiso del área clave de proceso: Definición del Proceso de la Organización (DPO).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para Desarrollar y mantener un PSE.

Para modelar este hecho, la actividad se puede desglosar en otro diagrama presentado en la Figura 4.2.

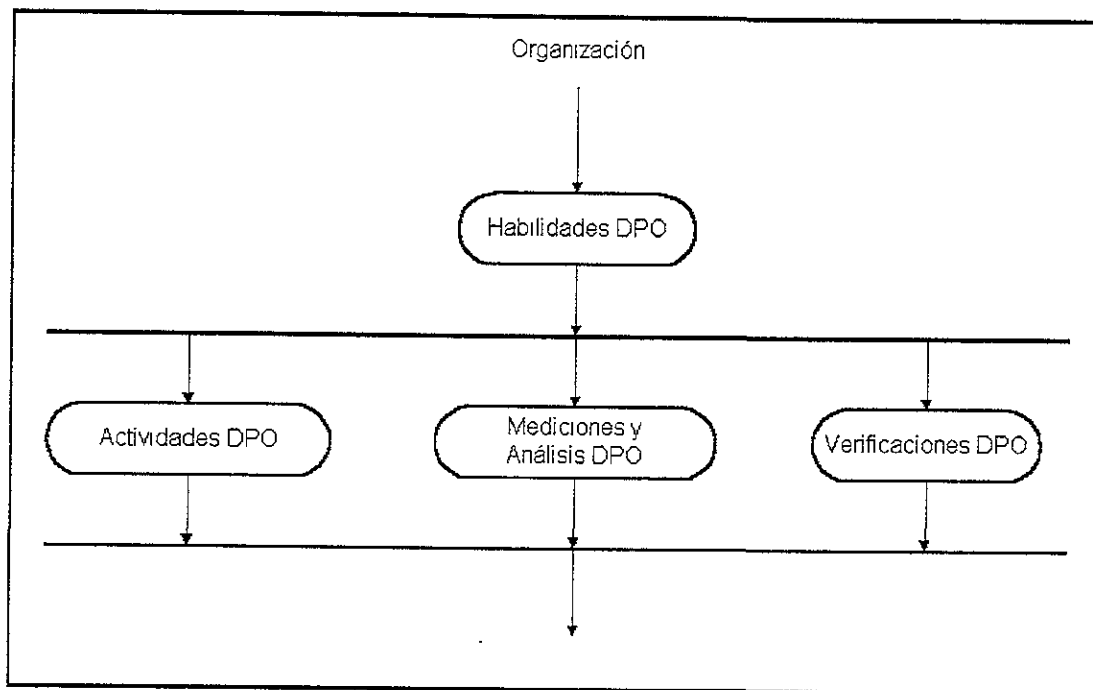


Figura 4.2 Desarrollar y mantener un Proceso de software estándar (PSE).

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y/o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de la Definición del Proceso de la Organización (DPO). Por otro lado, las actividades de verificación y medición con su análisis, son simultáneas a las de DPO.

Habilidades.

La habilidad 1 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo de la Organización. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado H1”.

La habilidad 2 habla de que los individuos que desarrollan y mantienen el PSE reciben la capacitación requerida, también se puede interpretar que está a cargo del Grupo de IPS. Esta habilidad la representamos con la actividad “Recibir capacitación H2”.

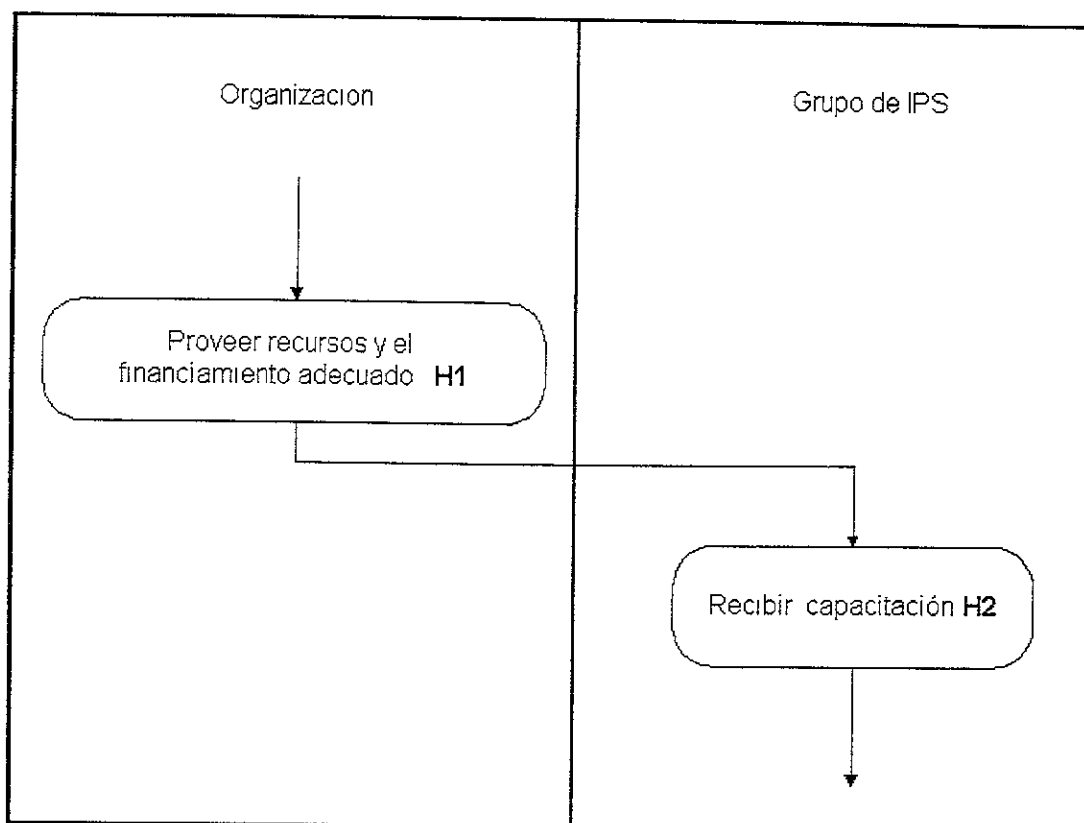


Figura 4.3 Habilidades del área clave de proceso: Definición del Proceso de la Organización (DPO).

El diagrama de la Figura 4.3 presenta las actividades a realizarse para adquirir las habilidades para el PSE, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde.

Actividades.

La actividad 1 especifica que el PSE de la organización es desarrollado y mantenido de acuerdo a un procedimiento documentado. Esta actividad la representamos en el diagrama como “Desarrollar y mantener el PSE A1”. En el texto no dice explícitamente quien realiza todas las actividades del PSE de la organización, pero por la existencia del Grupo IPS podemos concluir que debe ser su responsabilidad. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*. Al término de esta actividad tendremos como salida a *ps: Proceso de Software*.

La actividad 2 especifica que el PSE de la Organización es documentado de acuerdo a los estándares establecidos de la organización. Esta actividad la representamos como “Documentar el PSE A2”. Esta actividad recibe como entrada al objeto *ps: Proceso de software* que fue desarrollado anteriormente. Al término de esta actividad tendremos como salida a *psd: Proceso de Software Documentado*.

La actividad 3 especifica que las descripciones de los ciclos de software son documentadas y mantenidas, Esta actividad la representamos como “Documentar y mantener las descripciones de los ciclos de vida A3”. Al término de esta actividad tendremos como salida a *dcv: Descripción de los ciclos de vida*.

La actividad 4 especifica que se desarrollan guías y criterios para el ajuste de los proyectos del PSE de la organización. Esta actividad la representamos como “Desarrollar, mantener guías y criterios A4”.

La actividad 5 especifica que se establece y mantiene la base da datos del PSE de la organización. Esta actividad la representamos como “Establecer y mantener la base de datos del PSE A5”.

La actividad 6 especifica que se debe establecer y mantener una colección de documentos relacionados con el PSE. Esta actividad la representamos como “Establecer y mantener una colección de documentos A6”.

El diagrama de la figura 4.4 presenta las actividades que se describieron anteriormente. Por la naturaleza de la mejora continua de los procesos de las actividades arriba mencionadas de documentación, no pueden llevar un orden específico, por lo tanto en el diagrama representamos su ejecución de manera simultanea.

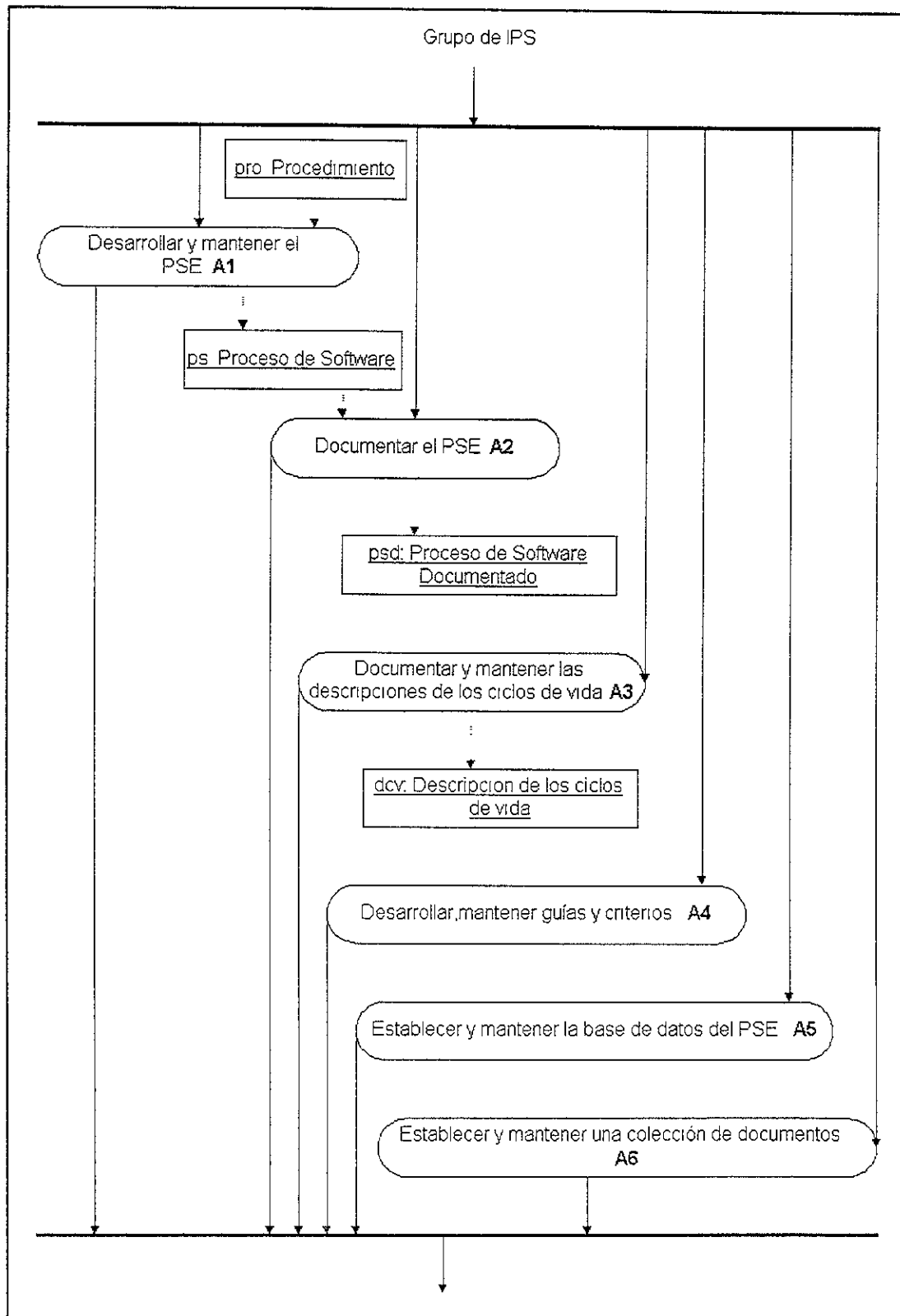


Figura 4.4 Actividades del área clave de proceso: Definición del Proceso de la Organización (DPO).

Mediciones y análisis.

El texto correspondiente dice que deben de recabarse y usar mediciones las para determinar el estado de las actividades del PSE. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad que lleva a cabo el Grupo de IPS. La actividad mencionada anteriormente queda representada en la figura 4.5

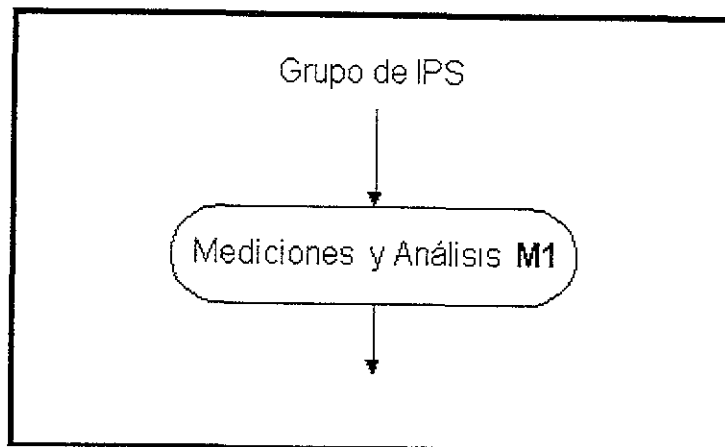


Figura 4.5 Medición y análisis del área clave de proceso: Definición del Proceso de la Organización (DPO).

Verificaciones.

La verificación 1 especifica que el grupo de aseguramiento de calidad revisa y / o audita las actividades y productos de trabajo para desarrollar y mantener el PSE de la organización. Esta verificación esta representada con la actividad de "Revisar y / o auditar las actividades y productos de trabajo V1". Lo cual mostramos en la Figura 4.6

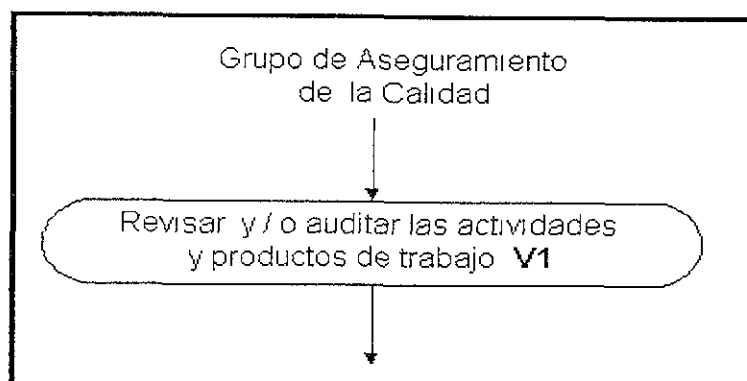


Figura 4.6 Verificaciones del área clave de proceso: Definición del Proceso de la Organización (DPO).

Capítulo 5 Programa de Capacitación

En el Programa de Capacitación (PC), el personal recibirá la capacitación adecuada para que desarrolle las habilidades y el conocimiento en cada proceso que aplica a su trabajo y realice su trabajo con eficiencia.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico del área clave Programa de Capacitación que abstrae la información contenida en el texto.

El modelo gráfico incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

5.1 Programa de Capacitación

Un área clave de proceso para el nivel 3: Definido

El propósito del área clave de procesos del Programa de Capacitación es desarrollar las habilidades y conocimientos de los individuos de tal forma que ellos puedan realizar sus roles eficaz y eficientemente.

El Programa de Capacitación involucra primero identificar la capacitación requerida por la organización, proyectos, e individuos y, después, desarrollar o procurar que la capacitación abarque las necesidades identificadas.

En cada proyecto de software se evalúan las necesidades actuales y futuras de habilidades y se determina como estas habilidades serán obtenidas. Algunas habilidades se adquieren eficaz y eficientemente a través de medios informales (ejemplos: entrenamiento en el trabajo y aprendizaje informal) mientras que otras habilidades necesitan medios de capacitación mas formales (ejemplos entrenamiento y auto-estudio guiado) para ser eficaz y eficientemente impartidas. Se seleccionan las formas y medios de capacitación apropiados.

Esta área de proceso cubre las prácticas para el grupo que realiza la función de capacitación. Las prácticas que identifican los tópicos específicos de capacitación (es decir conocimientos y habilidades requeridas) están contenidas en la característica común de habilidades para el desempeño en cada una de las áreas clave de procesos.

Metas

- | | |
|---------------|---|
| Meta 1 | Las actividades de capacitación son planeadas. |
| Meta 2 | Se provee la capacitación necesaria para desarrollar las habilidades y el conocimiento para realizar los roles técnicos y de administración de software. |
| Meta 3 | Los individuos en el grupo de ingeniería de software y los grupos relacionados reciben el entrenamiento necesario para realizar sus roles. |

Compromisos a realizar

- | | |
|---------------------|--|
| Compromiso 1 | La organización sigue una política escrita para satisfacer sus necesidades de capacitación. |
|---------------------|--|

Esta política especifica típicamente que:

1. Se identifican las habilidades y los conocimientos necesarios para cada rol técnico y de administración de software.
2. Los medios para impartir las habilidades y los conocimientos son identificados y aprobados.

Ejemplos de medios de capacitación aprobados incluyen:

- capacitación de aula,
- instrucción asistida por computadora,
- auto estudio guiado,
- programas formales de maestros y aprendices, y
- videos facilitadores.

3. La capacitación es proveída para construir la base técnica de la organización para cubrir las necesidades específicas de los proyectos y desarrollar las habilidades de los individuos.
4. La capacitación es desarrollada dentro de la organización u obtenida fuera de la organización cuando sea apropiado.

Ejemplos de fuentes externas de capacitación incluyen:

- capacitación entregada por el cliente,
- cursos de capacitación disponibles comercialmente,
- programas académicos,
- conferencias profesionales, y
- seminarios.

Habilidades a realizar

Habilidad 1

Existe un grupo responsable por el cumplimiento de las necesidades de capacitación de la organización.

Los miembros del grupo de capacitación pueden incluir instructores de tiempo completo o de medio

tiempo tomados de la organización, los miembros pueden también ser tomados de fuentes externas.

Un grupo es una colección de departamentos, gerentes e individuos que tiene responsabilidades para un conjunto de tareas o actividades. Un grupo está formado de una o varias personas asignadas medio tiempo y de diferentes departamentos, hasta varios individuos dedicados tiempo completo. Las consideraciones que debemos tener cuando se implementa un grupo incluyen las tareas y actividades asignadas, el tamaño del proyecto, la estructura y cultura de la organización. Algunos grupos como el de aseguramiento de la calidad de software se enfocan en las actividades del proyecto y otros como el grupo de ingeniería de procesos de software, se enfocan en las actividades de toda la organización.

Habilidad 2

Se provee de los recursos y del financiamiento adecuado para implementar el programa de capacitación.

Ejemplos de elementos del programa de capacitación:

- el plan de capacitación de la organización,
- materiales de capacitación,
- desarrollo y contratación de capacitación,
- llevar a cabo la capacitación,
- facilidades para la capacitación,
- evaluación de la capacitación, y
- el mantenimiento de registros de la capacitación.

1. Se designa un gerente para ser responsable por implementar el programa de capacitación de la organización.
2. Se hacen disponibles las herramientas para apoyar las actividades del programa de capacitación.

Ejemplos de herramientas de apoyo incluyen:

- estaciones de trabajo,

- herramientas de diseño de instrucción,
- programas de base de datos, y
- paquetería para desarrollar materiales de presentaciones.

3. Se preparan las instalaciones apropiadas para llevar a cabo la capacitación.

Las instalaciones de salas de clases deberían estar separadas del ambiente de trabajo de los estudiantes para eliminar las interrupciones.

Cuando sea apropiado la capacitación se lleva a cabo en escenarios que se parecen a las condiciones reales del trabajo e incluye las actividades que simulan las situaciones reales.

Habilidad 3

Los miembros del grupo de capacitación tienen las habilidades y el conocimiento necesario para realizar sus actividades de capacitación.

Ejemplos de formas de proveer estas habilidades y conocimientos incluyen:

- capacitación en técnicas de instrucción, y
- capacitación de perfeccionamiento del tema de la materia.

Habilidad 4

Los gerentes de software reciben orientación sobre el programa de capacitación.

Actividades a realizar

Actividad 1

Cada proyecto de software desarrolla y mantiene un plan de capacitación que especifica sus necesidades de capacitación.

El plan cubre:

1. El conjunto de habilidades necesarias y el momento en que estas necesidades serán requeridas.

2. Las habilidades para las cuales la capacitación es requerida y las habilidades que serán obtenidas por otros medios.

Algunas habilidades son eficaz y eficientemente impartidas a través de medios informales (por ejemplo capacitación y presentaciones informales, leyendo libros y revistas, pláticas, seminarios, capacitación en el trabajo y aprendizaje informal) mientras otras habilidades, para ser eficaz y eficientemente impartidas, necesitan estar basadas en medios mas formales de capacitación (por ejemplo capacitación de aula instrucciones asistidas por computadora, auto estudio guiado, facilitar videos, programas formales de maestros y aprendices.

3. La capacitación que se requiere, para quienes es necesaria y cuando sea requerida.

Refiérase a la característica común de habilidades a realizar en todas las otras áreas claves de proceso para ejemplos de necesidades específicas de entrenamiento.

Cuando sea apropiado, el entrenamiento de los individuos está unido a las responsabilidades de su trabajo y otras experiencias externas que reforzaran la capacitación dentro de un periodo razonable después de la capacitación.

4. Cómo será proporcionada la capacitación.

La capacitación puede ser proporcionada por el proyecto de software, por el grupo de capacitación de la organización o por una organización externa.

Ejemplos de capacitación apropiadamente hechas por el proyecto de software incluyen:

- capacitación en aplicaciones y requerimientos específicos del proyecto,
- capacitación en la arquitectura del proyecto de software, y
- otras capacitaciones que se realizan mas eficaz y eficientemente a nivel del proyecto.

Actividad 2

Se desarrolla y revisa el plan de capacitación de la organización de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El plan usa las necesidades de capacitación de los proyectos de software identificadas en sus planes de capacitación.
2. La capacitación especifica a ser provista se identifica basándose en las habilidades requeridas por la organización y el momento en que estas habilidades serán requeridas.
3. El plan de capacitación de la organización es revisado cuando sea apropiado para incorporar los cambios.
4. El plan de capacitación de la organización es revisado por los individuos afectados cuando es inicialmente liberado y siempre y cuando se hayan hecho revisiones mayores.

Ejemplos de individuos afectados incluyen:

- administrador gerencial,
- gerentes de software,
- gerentes de los grupos relacionados con el software.

5. El plan de capacitación de la organización es administrado y controlado.

Administrado y controlado implica que la versión del producto de trabajo de software en un determinado momento (pasado o presente) es conocida y los cambios son incorporados y controlados.

Si se desea un mayor grado de control del que es aplicado por el administrado y controlado el producto de trabajo puede ser puesto bajo la disciplina total de la Administración de configuración como es descrita en el área clave de proceso Administración de Configuración de Software.

6. El plan de capacitación de la organización está de manera oportuna disponible para los grupos o individuos afectados.

Ejemplos de grupos o individuos afectados incluyen:

- el administrador gerencial,
- el grupo de capacitación,
- los gerentes de grupos relacionados con el software,
- ingeniería de software (incluyendo todos los subgrupos tales como los de diseño de software),
- estimación de software,
- ingeniería de sistemas,
- pruebas de sistema,
- aseguramiento de la calidad de software,
- administración de configuración de software,
- administración de contratos, y
- apoyo de documentación.

Actividad 3

La capacitación para la organización es realizada de acuerdo con el plan de capacitación de la organización.

El plan cubre:

1. La capacitación requerida dentro de la organización en el momento en que sea necesaria.
2. La capacitación que será obtenida de fuentes externas y la capacitación que será proveída por el grupo de entrenamiento.

3. El financiamiento y recursos (incluyendo el personal, herramientas e instalaciones) necesarias para preparar y dirigir o procurar la capacitación.
4. Estándares para materiales de instrucción usados en cursos de capacitación desarrollados por el grupo de capacitación.
5. La programación para desarrollar y revisar los cursos de capacitación que serán realizados por el grupo de capacitación.
6. La programación para llevar a cabo la capacitación basados en las fechas acordadas con las necesidades expresadas y el numero estimado de estudiantes.
7. Los procedimientos para:
 - ☐ seleccionar los individuos quines recibirán la capacitación,
 - ☐ registros y participación en la capacitación,
 - ☐ mantenimiento en registros de capacitación provista, y
 - ☐ colección, revisiones y uso de evaluaciones de capacitación y otras retroalimentaciones de la capacitación.

Actividad 4

Se desarrollan los cursos de capacitación preparados al nivel de la organización de acuerdo a los estándares de la organización.

Estos estándares requieren que:

1. Se desarrolla una descripción de cada curso de capacitación.

Ejemplos de tópicos abarcados por la descripción incluyen:

- audiencia esperada,
- requisitos para participar,
- objetivos de la capacitación.
- duración de la capacitación,
- planes de clases.

- criterios para determinar el grado de satisfacción de los estudiantes,
- procedimientos para evaluaciones periódicas de la eficacia y la eficiencia de la capacitación, y
- consideraciones especiales tales como experimentación y pruebas de campo de cursos de capacitación, necesidades para entrenamientos de mejoras de la capacitación y oportunidades para llevar a cabo la capacitación.

2. Se revisan los materiales para los cursos de capacitación.

Ejemplos de individuos que revisan los materiales de capacitación incluyen:

- expertos en instrucción,
- expertos en el tema del curso, y
- estudiantes representativos de las sesiones experimentales del cursos bajo revisión.

3. Los materiales para la capacitación de los cursos son administrados y controlados.

Actividad 5

Se establece un procedimiento de renuncia a la capacitación el cual se usa para determinar si algún individuo ya posee el conocimiento y las habilidades necesarias para ejecutar sus roles designados.

Actividad 6

Se mantienen registros de la capacitación.

1. Se conservan los registros de todos los estudiantes que completaron satisfactoriamente cada curso de capacitación u otra actividad aprobada de capacitación.
2. Se conservan los registros de todos los estudiantes que completaron satisfactoriamente sus capacitaciones necesarias asignadas.
3. Los registros de los cursos completados satisfactoriamente se proporcionan para consideraciones en asignación del personal y administradores.

Medición y análisis

Medición 1

Se hacen mediciones y se usan para determinar el estado de las actividades del programa de capacitación.

Ejemplos de mediciones incluyen:

- asistencia real a cada curso de capacitación comparado con la asistencia proyectada,
- el progreso de los cursos de capacitación comparado con los planes de capacitación de los proyectos y de la organización, y
- el número de exhibiciones de capacitación aprobadas en el tiempo.

Medición 2

Se hacen mediciones y se usan para determinar la calidad del programa de capacitación.

Ejemplos de mediciones incluyen:

- resultados de pruebas post capacitación,
- revisiones de los cursos por parte de los estudiantes, y
- retroalimentación de los gerentes de software.

Verificación de la implementación

Verificación 1

Se revisan las actividades del programa de capacitación periódicamente con el Administrador Gerencial.

El propósito principal de las revisiones periódicas por el Administrador Gerencial es proveer una conciencia y visión de las actividades del proceso de software en un nivel apropiado de abstracción y de una manera oportuna. El tiempo entre las revisiones debe estar de acuerdo a las necesidades de la organización y puede prolongarse hasta que estén disponibles los mecanismos adecuados para el reporte de excepciones.

Refiérase a la verificación 1 del área clave de procesos de Seguimiento y Control del Proyecto de Software para prácticas que cubren el contenido típico de las revisiones de supervisión del Administrador Gerencial.

Verificación 2

El programa de capacitación es evaluado de manera independiente en forma periódica para verificar la consistencia con la relevancia para los requerimientos de la organización.

Verificación 3

Las actividades del programa de capacitación y productos de trabajo se revisan y / o auditan y los resultados se reportan.

Como mínimo las revisiones y / o auditorias verifican que:

1. Se sigue el proceso para desarrollar y verificar el plan de capacitación de la organización.
2. Se sigue el proceso para desarrollar y revisar los cursos de capacitación.
3. Los registros de capacitación se mantienen adecuadamente.
4. Los individuos designados para recibir capacitaciones específicas completan las capacitaciones.
5. Se sigue el plan de capacitación de la organización.

5.2 Modelo gráfico Programa de Capacitación (PC)

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en el programa de capacitación. y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

5.3 Identificación de los grupos participantes

En el modelo textual del Programa de capacitación hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave.

A continuación presentamos una breve definición de cada uno de ellos.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de varios proyectos a la vez.

Grupo de Aseguramiento de la Calidad: Revisa y / o audita las actividades del programa de capacitación la organización y reportar los resultados.

Gerente de Software: Cualquier gerente de un proyecto que tiene responsabilidad directa por el desarrollo y /o mantenimiento del software.

Grupo de capacitación. Se dice a la colección de individuos (tanto administradores, como personal) que es responsable por coordinar y poner en orden las actividades de capacitación para una organización. Este grupo típicamente prepara y dirige la mayoría de los cursos de capacitación y coordina el uso de otros medios de capacitación.

Evaluador: Es el responsable de evaluar periódicamente el programa de capacitación.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

Otros grupos relacionados: Es cualquier grupo involucrado en los procesos de software de la organización.

5.4 Diagrama de actividades.

Compromisos.

En esta área clave de procesos contamos con un compromiso, este establece que la organización sigue una política organizacional escrita para satisfacer sus necesidades de capacitación.

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que sirve para satisfacer sus necesidades de capacitación. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 5.1) como un objeto **p: Política**. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para satisfacer sus necesidades de capacitación queda expresado colocando a **p: Política** como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Satisfacer necesidades de capacitación C1”.

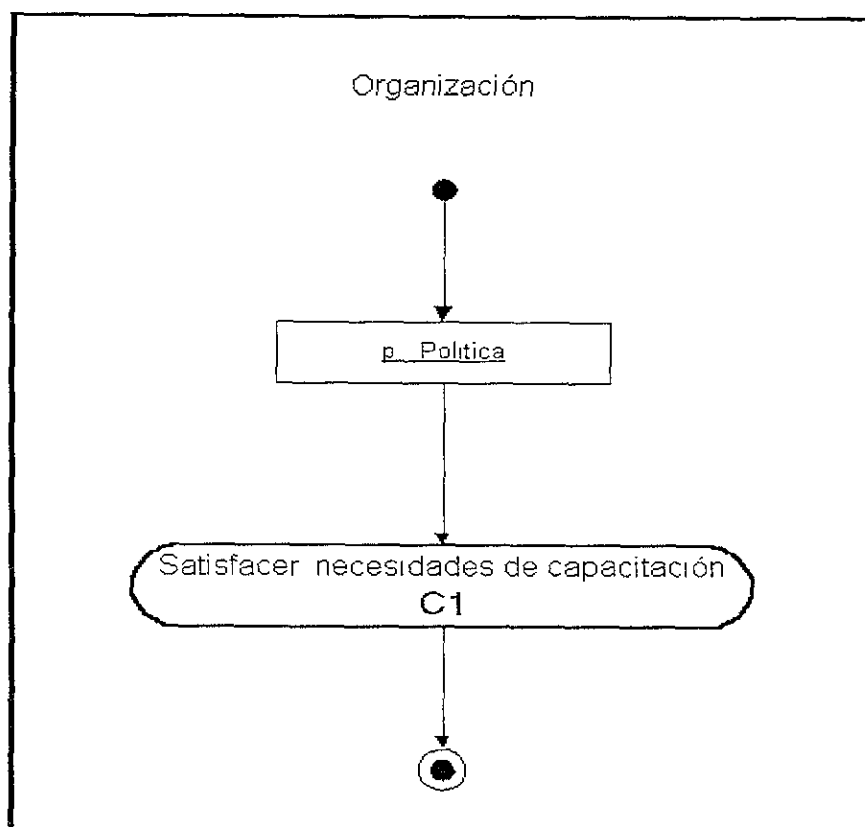


Figura 5.1 Compromisos del área clave de proceso: Programa de Capacitación (PC).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para satisfacer sus necesidades de Capacitación.

Para modelar este hecho, la actividad se puede desglosar en otro diagrama presentado en la Figura 5.2.

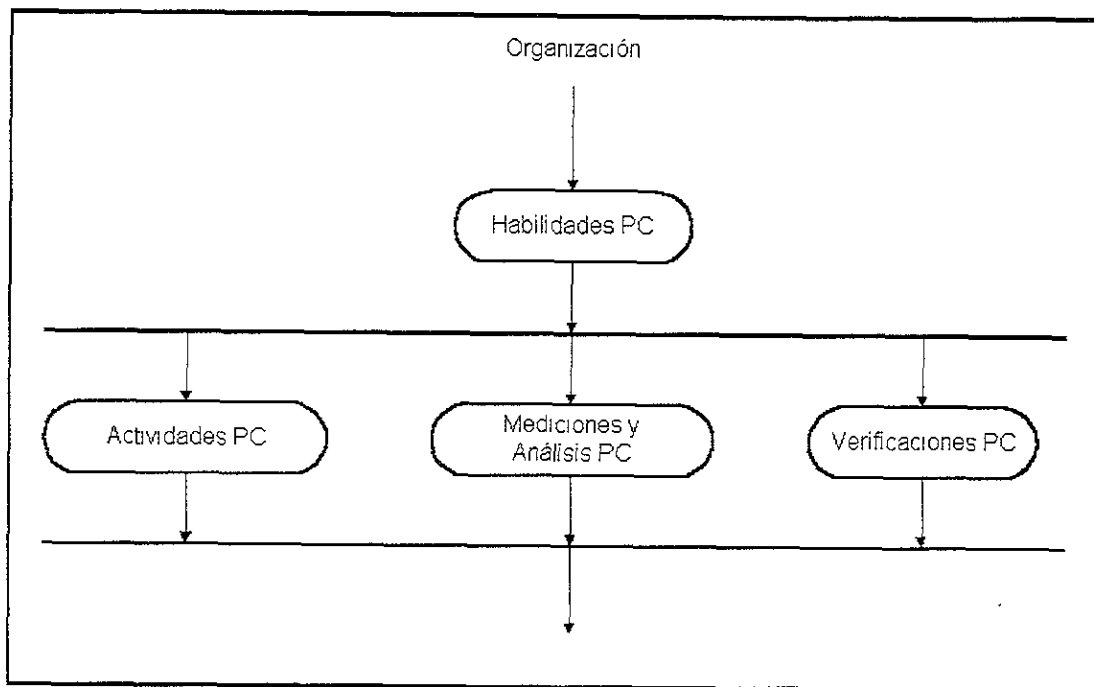


Figura 5.2 Satisfacer necesidades de capacitación.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias del Programa de capacitación. Por otro lado, las actividades de verificación y medición con su análisis, son simultáneas a las del Programa de capacitación.

Habilidades.

La habilidad 1 especifica que debe haber un grupo responsable de cumplir las necesidades de la capacitación de la organización. Si no existe el grupo se debe de crear. Generalmente este grupo es el grupo de capacitación.

La habilidad 2 habla de la provisión de recursos y financiamiento, que está a cargo de la organización. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación u orientación.

La habilidad 3 especifica que el Grupo de capacitación tiene el conocimiento necesario para realizar sus actividades de capacitación, para que tenga este conocimiento se le imparte capacitación a este grupo. Esta habilidad la representaremos con la actividad “Recibir capacitación”.

La habilidad 4 especifica que los gerentes de software reciben orientación sobre el programa de capacitación. Esta habilidad la representaremos con la actividad “Recibir orientación”.

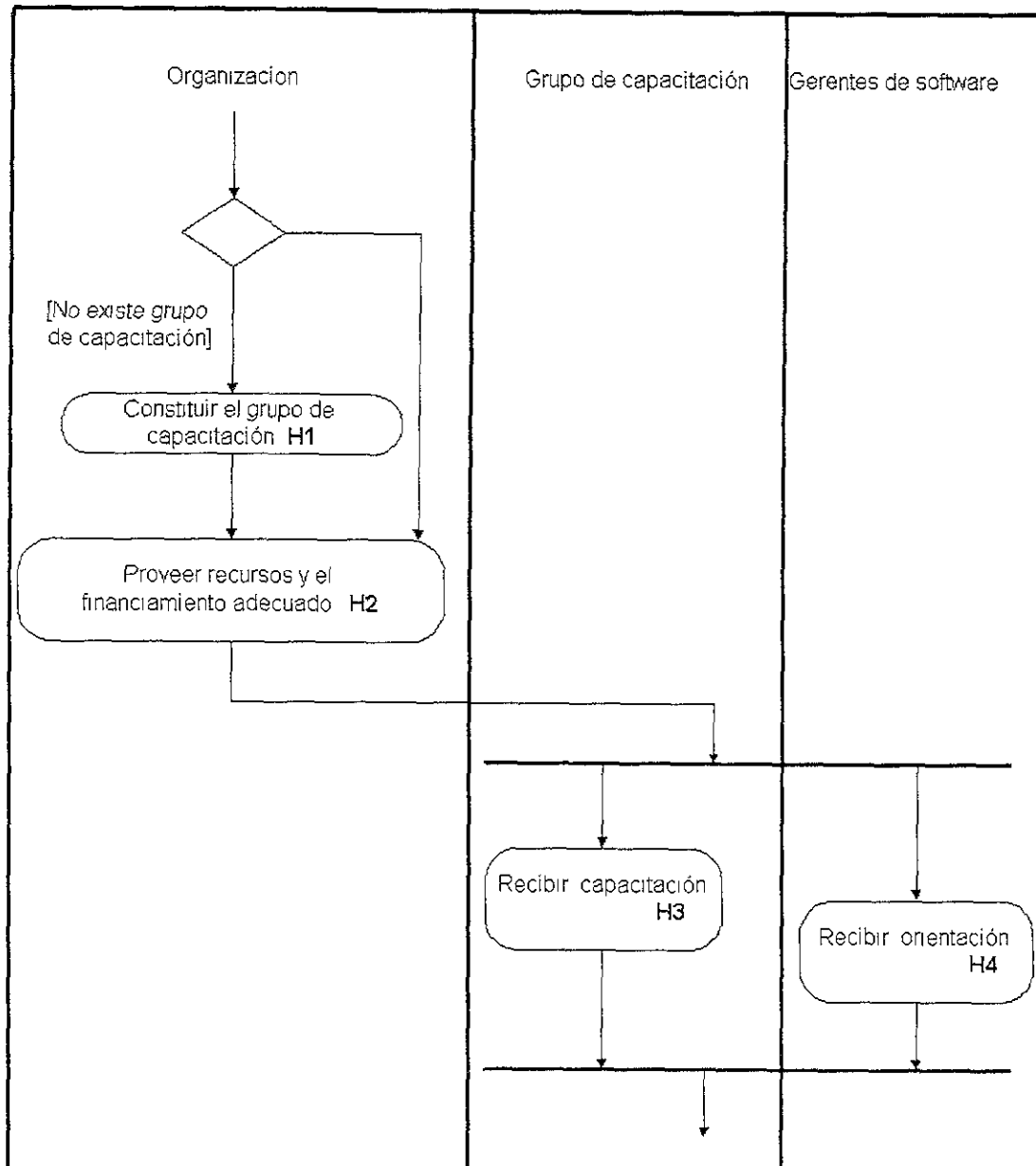


Figura 5.3 Habilidades del área clave de proceso: Programa de Capacitación (PC).

El diagrama de la figura 5.3 presenta las actividades a realizarse para adquirir las habilidades para el PC, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde.

Actividades.

La actividad 1 específica que por cada proyecto de software en curso se desarrolla y mantiene un plan de capacitación que especifica sus necesidades de capacitación. Esta actividad la representamos en el diagrama como “Desarrollar y revisar el plan de capacitación de cada proyecto A1”. Como en el texto no dice quien hace el plan de capacitación de cada proyecto nosotros proponemos que quien lo realiza son los gerentes de software y otros grupos relacionados con el software. Esta actividad recibe como entrada al objeto **pro: Proyecto de software**. Al término de esta actividad tendremos como salida a **placp: Plan de capacitación del Proyecto**.

La actividad 2 específica que se desarrolla y se revisa el plan de capacitación de la organización. Esta actividad la representamos como “Desarrollar y revisar el plan de capacitación la organización A2”. Como en el texto no dice quien desarrolla y revisa el plan de capacitación de la organización nosotros proponemos que quien lo realiza son los gerentes de software y otros grupos relacionados con el software, la organización y el administrador gerencial. Esta actividad recibe como entrada a los objetos **pro: Procedimiento documentado**, y **placp: Plan de capacitación del Proyecto**, que fue desarrollado anteriormente. Al término de esta actividad tendremos como salida **placo: Plan de capacitación de la organización [aprobado]**.

La actividad 3 específica que la capacitación debe realizarse de acuerdo al plan de capacitación de la organización. Dentro de esta actividad están contenidas las actividades 4,5,6 que se desglosaran en otro diagrama. Esta actividad la representamos como “Proceder a la capacitación de acuerdo al plan A3”. Como en el texto no dice quien realiza la capacitación de la organización nosotros proponemos que quien lo realiza es el grupo de capacitación. Esta actividad recibe como entrada al objeto **placo: Plan de capacitación de la organización [aprobado]**.

Las actividades mencionadas anteriormente quedan representadas en la Figura 5.4

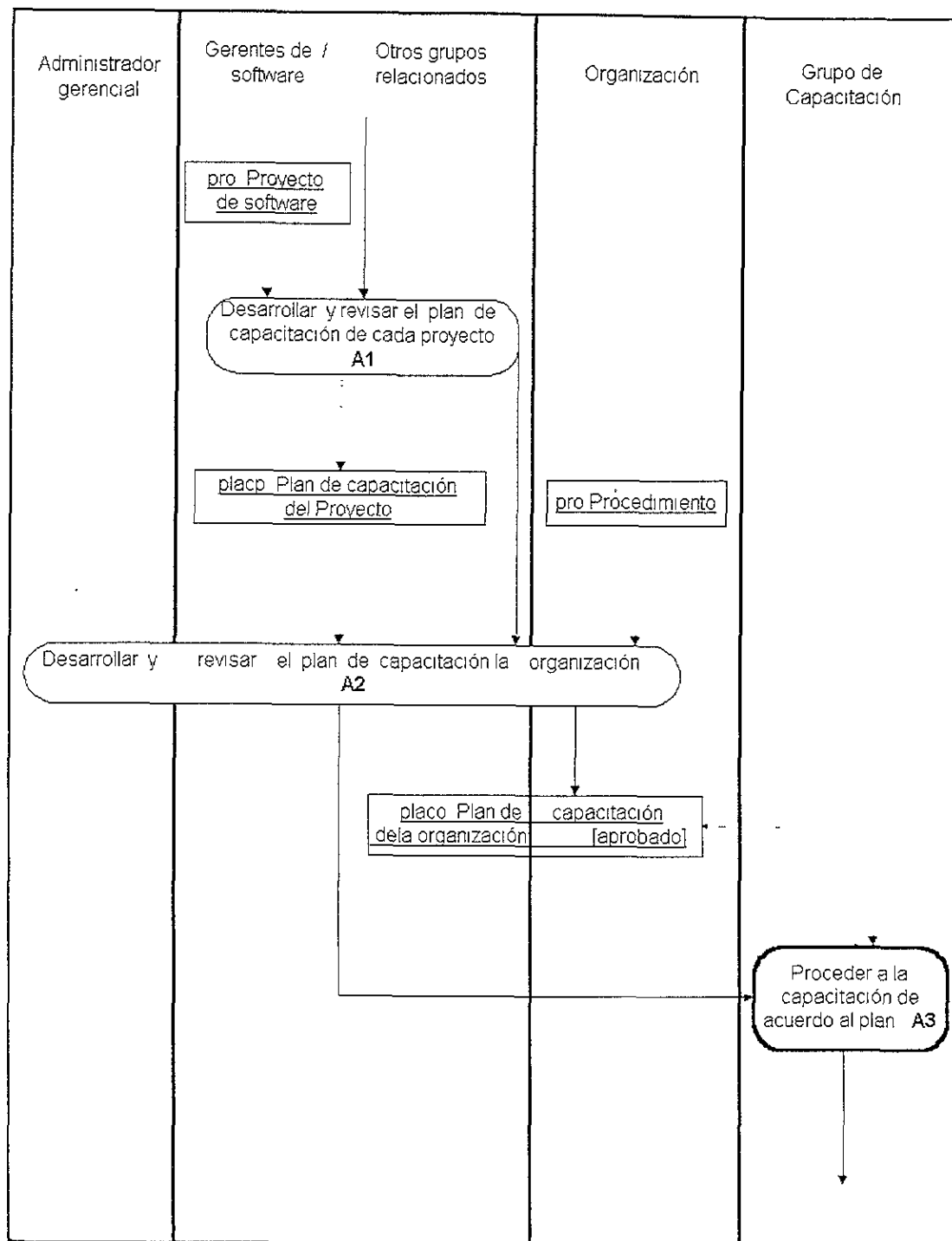


Figura 5.4 Actividades del área clave de proceso: Programa de Capacitación (PC).

La actividad 4 especifica que se desarrollan y mantiene los cursos de capacitación al nivel de la organización. Esta actividad la representamos como “Desarrollar y mantener los cursos de capacitación A4”. Como en el texto no dice quien realiza la

capacitación de la organización nosotros proponemos que quien lo realiza es el grupo de capacitación.

La actividad 5 especifica que se establece un procedimiento de renuncia a la capacitación para determinar si algún individuo ya posee el conocimiento y las habilidades necesarias para ejecutar sus roles. Esta actividad la representamos como “Verificar si los individuos adquirieron las habilidades requeridas A5”. Como en el texto no dice quien realiza la capacitación de la organización nosotros proponemos que quien lo realiza es el grupo de capacitación.

La actividad 6 especifica que se mantienen los registros de la capacitación. Esta actividad la representamos como “Mantener los registros de la capacitación A6”. Como en el texto no dice quien realiza la capacitación de la organización nosotros proponemos que quien lo realiza es el grupo de capacitación.

Las actividades mencionadas anteriormente quedan representadas en la figura 5.5

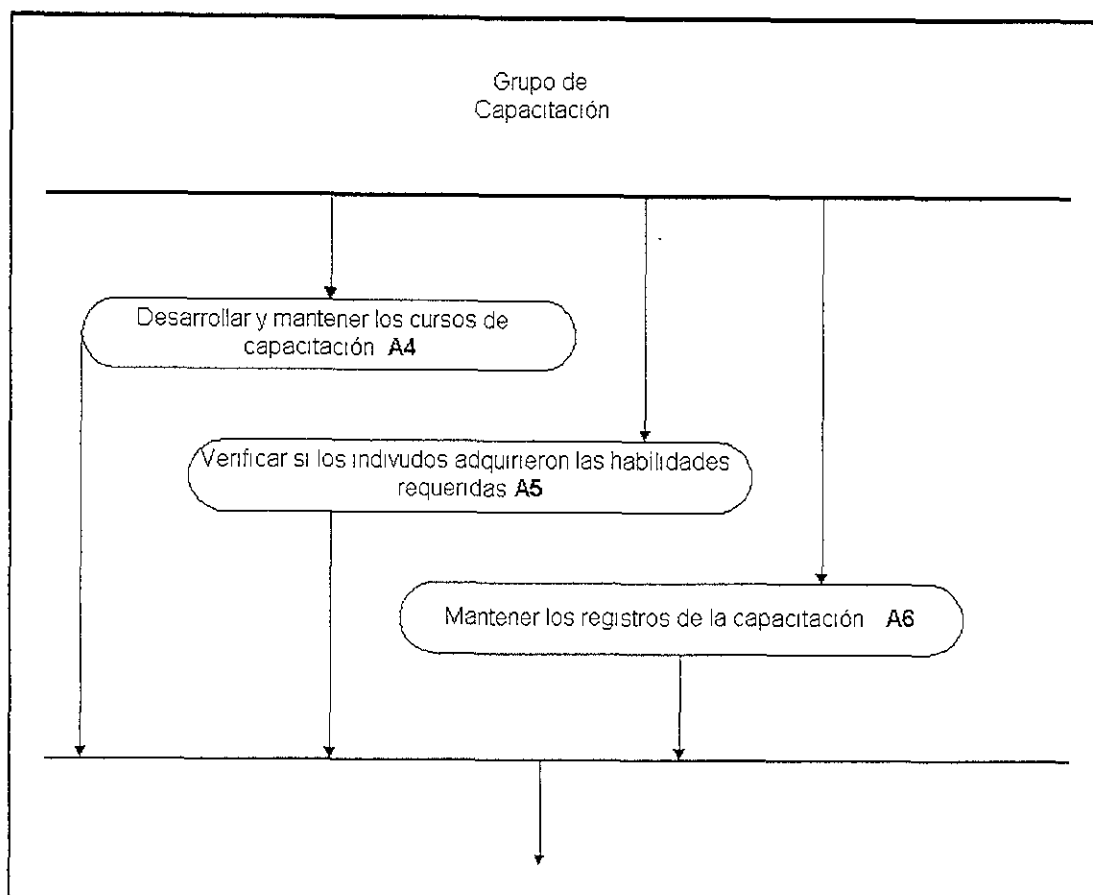


Figura 5.5: Proceder a la capacitación de acuerdo al plan.

El diagrama de la figura 5.5 presenta las actividades que se describieron anteriormente. También en este diagrama es fácil observar que actividades pueden llevarse a cabo en forma paralela en momentos adecuados.

Mediciones y análisis.

El texto correspondiente dice que deben de recabarse y usar las mediciones para determinar el estado de las actividades del PC. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad compartida entre el Administrador Gerencial y el Grupo de Capacitación. La actividad mencionada anteriormente queda representada en la figura 5.5

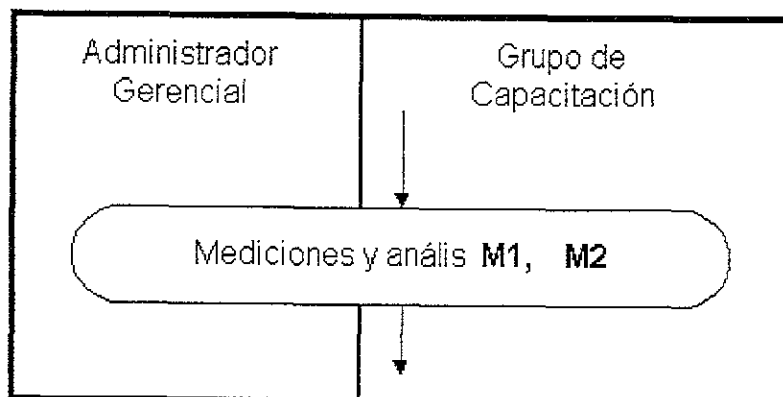


Figura 5.6 Medición y análisis del área clave de proceso: Programa de Capacitación (PC).

Verificaciones.

La verificación 1 especifica que las actividades para el PC serán revisadas periódicamente por el Administrador Gerencial. Esta verificación está representada con la actividad de “Revisar Actividades del PC V1”.

La verificación 2 especifica que el programa de capacitación es evaluado de manera independiente y en forma periódica para verificar la consistencia con la relevancia para los requerimientos de la organización. Como en el texto no dice quien realiza la verificación nosotros proponemos que quien lo realiza es un Evaluador independiente. Esta verificación está representada con la actividad de “Evaluar periódicamente el PC V2”.

Finalmente en la verificación 3 especifica que las actividades del programa de capacitación y productos de trabajo se revisan y / o auditan y los resultados se reportan. Como en el texto no dice quien realiza esta verificación nosotros proponemos que quien lo realiza es el grupo de aseguramiento de la calidad. Esta verificación está representada con la actividad de “Revisar y / o auditar las actividades y productos de trabajo V3”.

Las actividades mencionadas anteriormente quedan representadas en la figura 5.7.

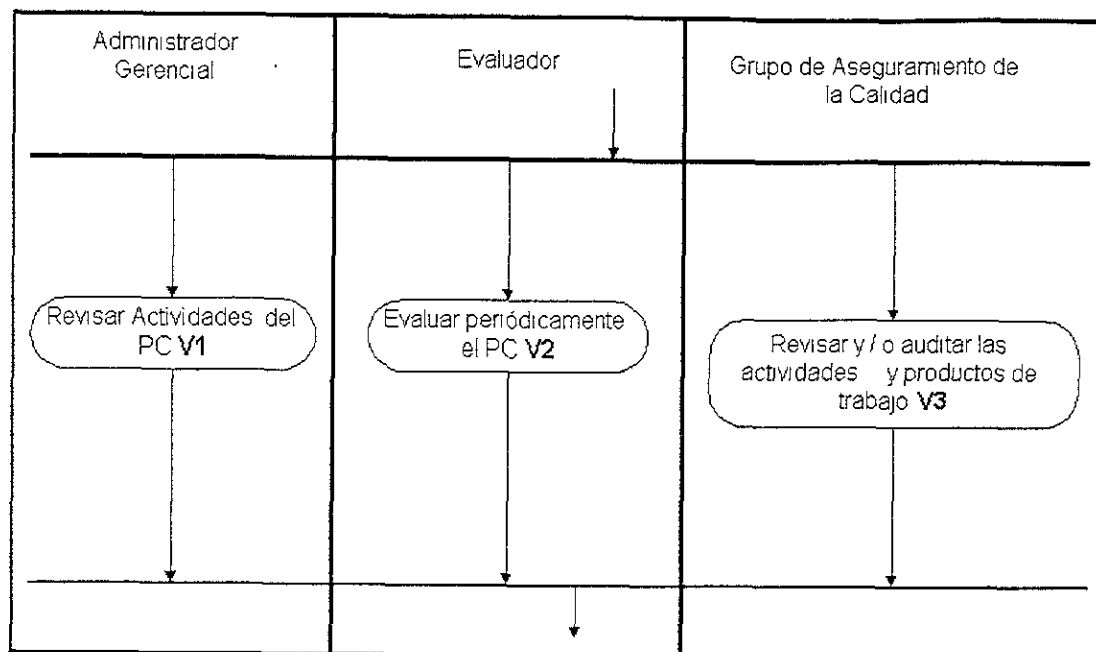


Figura 5.7 Verificaciones del área clave de proceso: Programa de Capacitación (PC).

Capítulo 6 Administración Integrada de Software

En la Administración Integrada de Software trata de unir las actividades de ingeniería de software con las de administración, en un proceso definido y que será ajustado del proceso de software de la organización.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico que abstrae la información contenida en el texto.

El modelo gráfico incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

Compromisos a realizar

Compromiso 1

El proyecto sigue una política de organización escrita que demanda que el proyecto de software sea planeado y administrado usando el proceso de software estándar de la organización y los elementos relacionados del proceso.

Refiérase al área clave de proceso de Definición del Proceso de la Organización para prácticas que cubren el proceso de software estándar de la organización y los elementos de proceso relacionados.

Esta política típicamente especifica que:

1. Cada proyecto documenta el proceso de software definido del proyecto ajustando el proceso de software estándar de la organización.
2. Las desviaciones del proyecto del proceso de software estándar de la organización son documentadas y aprobadas.
3. Cada proyecto realiza sus actividades de acuerdo con el proceso de software definido del proyecto.
4. Cada proyecto recopila y almacena los datos apropiados de mediciones del proyecto en la base de datos del proceso de software de la organización.

Refiérase a la actividad 5 del área clave de procesos Definición del Proceso de la Organización para prácticas que cubren la base de datos del proceso de software de la organización.

Habilidades a realizar

Habilidad 1

Se provee los recursos y el financiamiento adecuado para administrar el proyecto de software usando el proceso de software definido del proyecto.

Refiérase a la habilidad 3 del área de proceso Planificación del proyecto de software y la habilidad 3 del área clave de proceso Seguimiento y control del Proyecto de Software para prácticas que cubren recursos y el financiamiento para la planificación, seguimiento y control del proyecto de software.

Habilidad 2

Los individuos responsables para el desarrollo del proceso de software definido del proyecto reciben la capacitación adecuada en como ajustar el proceso de software estándar de la organización y usar los elementos relacionados del proceso.

Ejemplos de capacitación incluyen:

- uso de la base de datos del proceso de software,
- uso de los procesos de software estándar de la organización, y
- uso de guías y criterios para ajustar los proceso de software estándar de la organización para satisfacer las necesidades del proyecto de software.

Refiérase al área clave de proceso Programa de Capacitación.

Habilidad 3

Los gerentes de software reciben la capacitación adecuada para administrar los aspectos técnicos, administrativos y aspectos del personal del proyecto de software basándose en el proceso de software definido del proyecto.

Refiérase a la habilidad 4 del área clave de procesos Planificación del proyecto de software y la habilidad 4 del área clave de procesos Seguimiento y control del proyecto de software para prácticas que cubren la capacitación para la planificación, seguimiento y control del proyecto de software.

Ejemplos de capacitación incluyen:

- métodos y procedimientos para la estimación de software, planeación y seguimiento basándose en el proceso de software definido del proyecto; y
- métodos y procedimientos para identificar, administrar y comunicar riesgos de software.

Refiérase al área clave de proceso Programa de Capacitación.

Actividades a realizar

Actividad 1

Se desarrolla el proceso de software definido del proyecto adoptando el proceso de software estándar de la organización de acuerdo a los procedimientos documentados.

Refiérase a la actividad 2 del área clave de procesos Definición del Proceso de la Organización para prácticas que cubren los contenidos del proceso de software estándar de la organización.

Estos procedimientos típicamente especifican que:

1. Un ciclo de vida de software es:

- ☐ seleccionado de aquellos aprobados por la organización para satisfacer las restricciones contractuales y operacionales del proyecto;

Refiérase a la actividad 3 del área clave de proceso Definición del Proceso de la organización para prácticas que cubren ciclos de vida de software aprobados.

- ☐ modificado, si es necesario de forma permitida por las guías y criterios de adaptación de la organización; y
- ☐ documentado de acuerdo a los estándares de la documentación.

Refiérase a la actividad 4 del área de procesos Definición del proceso de software de la organización para prácticas que cubren las guías y criterios de adaptación de la organización.

2. Se documenta la descripción del proceso de software definido.

Refiérase a la actividad a del área de procesos Definición del proceso de software de la organización para prácticas que cubren los contenidos esperados de la definición de un proceso.

3. La adaptación del proceso de software estándar de la organización para el proyecto es revisado por el grupo responsable para coordinar las actividades del proceso de software de la organización (por ejemplo el grupo de Ingeniería de procesos) y es aprobado por la Administración General.

Refiérase a la actividad 6 del área clave de procesos Definición del Proceso de la Organización para prácticas que cubren los contenidos esperados de la definición del proceso.

- ☐ Permisos para desviaciones del proceso software estándar son documentados, revisados y aprobados por la Administración General.
4. Permisos para desviaciones de requisitos contractuales del proceso de software están documentados, revisados y aprobados por el Administrador Gerencial y el cliente del proyecto de software, según lo apropiado.

5. La descripción del proceso de software definido del proyecto es administrada y controlada.

Administrado y controlado implica que la versión del producto de trabajo de software en un determinado momento (pasado o presente) es conocida y los cambios son incorporados y controlados.

Si se desea un mayor grado de control del que es aplicado por el administrado y controlado el producto de trabajo puede ser puesto bajo la disciplina total de la Administración de configuración como es descrita en el área clave de proceso Administración de Configuración de Software.

Actividad 2

Se revisa cada proceso de software definido del proyecto de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Los cambios derivados por las siguientes causas son documentados y sistemáticamente revisados:
 - ☐ las lecciones aprendidas de seguir las actividades de software de los proyectos de la organización,
 - ☐ cambios propuestos por el proyecto de software, y
 - ☐ datos de mediciones de procesos y productos de trabajo.
2. Los cambios al proceso de software definido del proyecto son revisados y aprobados antes de que estos sean incorporados.

Ejemplos de individuos que revisan los cambios incluyen:

- miembros de los grupos responsables por las actividades del proceso de software de la organización (por ejemplo el grupo de ingeniería de procesos de software),

- los gerentes de software, y
- los gerentes de proyectos de software.

Ejemplos de individuos que aprueban los cambios incluyen:

- El gerente de software, y
- El gerente del proyecto de software.

Actividad 3

Se desarrolla y revisa el plan de desarrollo de software del proyecto, el cual describe el uso del proceso de software definido del proyecto de acuerdo a un procedimiento documentado.

Refiérase a las actividades 6 y 7 del área clave de procesos Planificación del proyecto de software y las actividades 1 y 2 del área clave de procesos Seguimiento y control del proyecto de software para prácticas que cubren el plan de desarrollo de software.

Actividad 4

Se administra el proyecto de software de acuerdo con el proceso de software definido del proyecto.

Refiérase a las áreas claves de procesos de Planificación del proyecto de software y Seguimiento y control del proyecto de software para prácticas básicas que cubren la Administración del Proyecto de Software.

El proceso de software definido del proyecto típicamente especifica que:

1. Se toman previsiones para obtener analizar y reportar los datos de las mediciones requeridas para administrar el proyecto de software.
2. Las actividades para estimación, planificación y seguimiento de software son ligadas a las tareas claves y a los productos de trabajo del proceso de software definido del proyecto.

3. Se establecen, documentan y se usan criterios de término y completitud para autorizar la iniciación y determinar la terminación de las tareas claves.
4. Se definen criterios documentados para indicar cuando replanear el proyecto de software.
5. Las lecciones técnicas y administrativas aprendidas se documentan y se almacenan en la colección de documentos relacionados con el proceso de software de la organización.

Refiérase a la actividad 6 del área clave de procesos de la Definición del Proceso de la organización para prácticas que cubren la colección de documentos relacionados con el proceso de software de la organización.

6. Las lecciones técnicas y administrativas aprendidas del seguimiento de las actividades de otros proyectos en la organización son sistemáticamente revisadas y usadas para estimar, planificar, seguir, y replanear el proyecto de software.
7. El plan de recursos humanos contempla las necesidades del proyecto de software por individuos con habilidades especiales y conocimiento del dominio de aplicación.
8. Se identifican las necesidades de capacitación y se documentan para ajustarse a las necesidades específicas del proyecto de software.

Refiérase a la actividad 1 del área clave de procesos Programa de capacitación para prácticas que cubren la identificación de las necesidades de capacitación del proyecto.

9. Los planes y procesos de software que se rigen al interactuar con otros grupos se ajustan para considerar disparidades con estos grupos y para otros problemas potenciales.

Ejemplos de disparidades y problemas incluyen:

- diferencias en la madurez del proceso,
- incompatibilidad de procesos, y
- varios factores de negocios.

Actividad 5

La base de datos del proceso de software de la organización se usa para planificación y estimación de software.

Refiérase a la actividad 5 de la área clave de proceso Definición del Proceso de la Organización para prácticas que cubren la base de datos del proceso de software de la organización.

1. La base de datos se usa como una fuente de datos para estimar, planificar y replanear un proyecto de software. Cuando sea posible se usan los datos de un proyecto de software similar.

Ejemplos de datos contenidos en la base de datos del proceso de software de la organización incluyen:

- el tamaño de los productos de trabajo de software,
- esfuerzo de software,
- calendarización,
- personal, y
- actividades técnicas.

2. Los valores de los parámetros usados para derivar estimaciones para el tamaño de software, esfuerzo, costo, calendarización y el uso de recursos críticos de computadoras se comparan con aquellos de otros proyectos de software para evaluar su validez.

- ☐ Se evalúan y registran las similitudes y diferencias con otros proyectos del mismo dominio de la aplicación y con la misma aproximación del diseño.

- ☐ Razones para semejanzas y diferencias entre los valores de los parámetros son registradas.
 - ☐ El razonamiento usado para juzgar la credibilidad de las estimaciones del proyecto es registrado.
3. El proyecto de software provee los datos adecuados de la planificación y replanificación de software y los datos de mediciones reales para su almacenamiento en la base de datos del proceso de software de la organización.

Ejemplos de datos almacenados por el proyecto de software incluyen.

- la descripción de la tarea,
- los supuestos,
- las estimaciones,
- las estimaciones revisadas,
- los datos de mediciones reales, y
- la información asociada requerida para reconstruir las estimaciones, evaluar su razonabilidad, y derivar estimaciones para un nuevo trabajo.

Actividad 6

Se administra el tamaño de los productos de trabajo de software (o el tamaño de los cambios en los productos de trabajo de software) de acuerdo a un proceso documentado.

Refiérase a la actividad 9 del área clave de proceso Planificación del proyecto de software y la actividad 5 del área clave de proceso Seguimiento y control del proyecto de software para prácticas básicas que cubren la planificación y el seguimiento del tamaño de los productos de trabajo de software.

Este procedimiento típicamente especifica que:

1. Un grupo, que es independiente del grupo de ingeniería de software, revisa los procedimientos para estimar el tamaño de los productos de trabajo de software y provee guías para usar datos históricos de la base de datos del proceso de software de la organización para hacer estimaciones.

Un ejemplo de un grupo independiente es un grupo de estimación de software.

Un ejemplo de un método para evaluar la credibilidad del tamaño de software es una comparación función por función respecto a un sistema terminado.

- ☐ Los individuos que preparan las estimaciones del tamaño, aseguran que los procedimientos y datos usados en las estimaciones sean apropiados.
 - ☐ Cuando la validación de un tamaño estimado es cuestionado, un equipo de los pares y expertos revisa la estimación.
2. Se aplica un factor de contingencia al tamaño estimado por cada elemento de software identificado como un riesgo de software.
- ☐ La razón para la contingencia se documenta.
 - ☐ Los riesgos asociados con reducir o eliminar la contingencia son evaluados y documentados.
3. Se identifican los componentes de software comercialmente disponibles o reutilizables.
- ☐ Se toman en cuenta las métricas de reuso para la reutilización de requerimientos, diseño, código, planes de prueba, y pruebas de procedimientos, etc.
 - ☐ El esfuerzo para modificar e incorporar componentes reutilizables se pondera contra las estimaciones del tamaño.
4. Factores que pueden afectar significativamente el tamaño de los productos de software son identificados y seguidos de cerca.
5. Se establece un umbral para cada elemento de software administrado, el cual cuando será excedido requiera de una acción.

Actividad 7

Se administran los esfuerzos y gastos del proyecto de software de acuerdo a un procedimiento documentado.

Refiérase a la actividad 10 del área clave de Proceso de Planificación del proyecto de software y la actividad 6 del área clave de procesos Seguimiento y control del proyecto de software para prácticas básicas que cubren la planificación y el seguimiento del esfuerzo y costo de software.

Este procedimiento típicamente especifica que:

1. Los modelos de esfuerzo, costos y perfil del personal de software, si son usados son ajustados al proyecto y usan datos históricos disponibles cuando sea apropiado.
2. Los datos de referencia y costo son ajustados para incorporar variables del proyecto.

Ejemplos de variables del proyecto incluyen:

- la localización geográfica de los grupos y las organizaciones del proyecto (por ejemplo los subcontratistas),
- el tamaño y complejidad del sistema,
- la estabilidad de los requerimientos,
- el ambiente del servidor para el desarrollo,
- el ambiente objetivo del sistema,
- la familiaridad y experiencia de los desarrolladores con la aplicación,
- la disponibilidad de los recursos, y
- otras restricciones especiales.

3. El esfuerzo y costo total de software se asigna a tareas o actividades administradas individualmente según lo requiera la administración efectiva del esfuerzo y costo.
4. Cuando se revisa el estado del esfuerzo y gasto de software y se comparan las estimaciones con los gastos actuales sobre el tiempo, y contra el trabajo completo contra el plan de desarrollo de software y se usan para refinar las estimaciones de esfuerzo y costo para el trabajo restante.

- ☐ Los valores de los parámetros de los modelos usados en la estimación del esfuerzo y costo del software son actualizados siempre y cuando se hagan cambios mayores a los requerimientos de software.
 - ☐ Cuando sea apropiado se usan los datos reales sobre la productividad del proyecto y otros gastos nuevos de software.
5. Se establece un umbral para el esfuerzo y gasto para cada tarea o actividad individual, el cual cuando será excedido, requiera una acción.

Actividad 8

Se administran los recursos críticos de la computadora del proyecto de acuerdo a un procedimiento documentado.

Refiérase a la actividad 11 del área clave de procesos Planificación de proyectos de software y la actividad 7 del área clave de proceso Seguimiento y control de proyectos de software para prácticas básicas que cubren la planificación y seguimiento de recursos críticos de la computadora.

Este procedimiento típicamente especifica que:

1. Se derivan las estimaciones para los recursos críticos de la computadora del proyecto, con base en la experiencia histórica, simulaciones, prototipos o análisis según corresponda.
 - ☐ Se documentan las fuentes y razones de las estimaciones.
 - ☐ Se evalúan y registran las similitudes y diferencias entre el proyecto y las fuentes de datos históricos en términos del dominio de la aplicación y la aproximación del diseño.
 - ☐ Se registran las razones usadas para juzgar la credibilidad de las estimaciones.
2. Se ajustan los recursos planificados de la computadora, los requisitos de sistema asignados al software, los requerimientos de software y / o el diseño de software

para cumplir con los requerimientos críticos de la computadora del proyecto.

3. Se asignan los recursos disponibles de la computadora a los componentes de software.
4. La capacidad disponible para los recursos críticos de las computadoras forma una capacidad de reserva para las estimaciones iniciales.
5. Se establece un umbral para cada recurso crítico de la computadora el cual cuando será excedido, requiera una acción.

Actividad 9

Se administran las dependencias críticas y rutas críticas de la calendarización del proyecto de software de acuerdo a un proceso documentado.

Refiérase a la actividad 12 del área clave de proceso de Planificación del proyecto del software, la actividad 8 del área clave de proceso Seguimiento y control del proyecto de software y la actividad 4 del área clave de proceso de Coordinación entre grupos para prácticas que cubren la negociación y el seguimiento de dependencias críticas.

Este procedimiento típicamente especifica que:

1. Se asignan hitos, tareas, compromisos, dependencias críticas, personal, gastos y revisiones en la calendarización de manera consistente con el proceso de software definido del proyecto.
 - ☐ La programación de software identifica tareas e hitos específicos cuyo término puede ser objetivamente determinado (es decir una determinación binaria si / no).

Se desarrollan diferentes niveles de detalle de la programación, apropiadamente ligados uno con otros, para acomodarse a las necesidades de los diferentes grupos e individuos.

2. Son definidas las dependencias críticas, negociadas y reflejadas en la programación del software.

Las dependencias críticas incluyen tanto aquellas dentro del grupo de ingeniería de software (es decir entre subgrupos) como el grupo de ingeniería de software y otros grupos afectados.

3. Son definidas y reflejadas las rutas críticas de la calendarización en la programación del software.
4. Las dependencias críticas y las rutas críticas de la programación son monitoreadas en forma regular.
5. Se establecen criterios específicos documentados basados en umbrales para cada ruta crítica, que cuando sean excedidos requieran de una acción.

Ejemplos de acciones incluyen:

- dirigir análisis y simulaciones para la función de equilibrio, calidad, costo, programación, personal y otros recursos;
- asignar contingencias y relajar la calendarización si hay disponibilidad;
- evaluar los efectos de las acciones contempladas en todas las rutas críticas; y
- hacer visibles las decisiones que afectan a los grupos.

Actividad 10

Se administran y documentan los riesgos de software identificados y evaluados, de acuerdo a un procedimiento documentado.

Refiérase a la actividad 13 del área clave de proceso de Planificación del proyecto de software y la actividad 10 del área clave de Seguimiento y control del proyecto de software para prácticas básicas que cubren la identificación y seguimiento de riesgos.

Ejemplos de riesgos de software a ser administrados incluyen posibilidades significativas de que el proyecto de software puede fallar en cumplir sus objetivos en áreas tales como:

- calendarización,
- gasto,
- funcionalidad,
- resultados o desempeño en tiempo real,
- fiabilidad o disponibilidad, y
- uso de recursos críticos de computadora.

Ejemplos de actividades para administrar riesgos incluyen

- identificar los objetivos de alto riesgo del proyecto;
- identificar los eventos que pueden introducir o incrementar los riesgos;
- prototipos o implementación de módulos de alto riesgo; y
- seguir minuciosamente los indicadores claves de los riesgos del proyecto.

Este procedimiento típicamente especifica que:

1. Se documenta y se usa un plan de administración de riesgos del proyecto para identificar y administrar los riesgos de software.

Ejemplos de elementos de un plan de administración de riesgos incluyen:

- recursos requeridos (incluyendo personal y herramientas);
- métodos de administración de riesgos (por ejemplo identificación, análisis, ordenación, planeación, seguimiento y resolución);
- lista de riesgos identificados (incluyendo evaluación, orden, estado y plan);
- calendarización de la administración de riesgos;
- responsabilidades y autoridades;

- métodos y la frecuencia de la comunicación del estado de los riesgos y actividades; y
- mediciones.

2. La planificación de contingencias está basado en el proceso de software definido del proyecto y se realiza a lo largo del ciclo de vida de software del proyecto.

Ejemplos de áreas cubiertas por las actividades de planificación de contingencias incluyen:

- identificación de opciones
- evaluación del impacto de las opciones
- la fiabilidad técnica de las opciones
- asignación de reservas de la administración y
- criterios de decisión sobre cuando proseguir una opción.

3. Se definen alternativas para cada uno de los riesgos de software, cuando sea posible junto con el criterio para seleccionar entre las alternativas.
4. La liberación inicial y las revisiones mayores para el plan de administración de riesgos de software son sometidas a la revisión entre colegas.

Refiérase al área clave de procesos de la Revisión Entre Colegas.

5. Se administra el plan de administración de riesgos de software.
6. Los riesgos de software son seguidos revaluados y replaneados en hitos selectos del proyecto, a los puntos de revisión de riesgos designados, y durante la planificación de cambios significativos que afectan al proyecto de software.

- ☐ Se revisan y corrigen las prioridades de los riesgos y los planes de administración de riesgos en éstos puntos de la reevaluación.

- ☐ Se usa la información obtenida de seguir los riesgos para refinar la evaluación de los riesgos y los planes de administración de riesgos.

7. El grupo de ingeniería de software, otros grupos e individuos son incluidos en las comunicaciones sobre los riesgos de software, el plan de administración de riesgos de software y los resultados de la mitigación de riesgos.

Ejemplo de grupos e individuos afectados incluyen:

- el cliente,
- los subcontratistas,
- usuarios finales,
- estimaciones de software,
- ingeniería de sistemas,
- pruebas del sistema,
- aseguramiento de la calidad de software,
- administración de configuración de software,
- administración de contrato, y
- apoyo de documentación.

Actividad 11

Se realizan periódicamente las revisiones del proyecto de software, para determinar las acciones necesarias para llevar el desempeño y los resultados del proyecto de software en línea con las necesidades actuales y proyectadas del negocio, el cliente y los usuarios finales según lo adecuado.

Ejemplos de acciones incluyen:

- la aceleración de la calendarización,
- cambiar los requerimientos del sistema en respuesta a los cambios en las oportunidades de mercado o las necesidades de los clientes y usuarios finales, y
- terminación del proyecto.

Los usuarios finales referidos en estas prácticas son los usuarios finales designados por el cliente o representantes de los usuarios finales.

Medición y análisis

Medición 1

Se hacen mediciones y se usan para determinar la eficiencia de las actividades de la Administración integrada de software.

Ejemplos de mediciones incluyen

- esfuerzo gastado en el tiempo de administrar el proyecto de software comparado con el plan;
- frecuencia, causas y magnitud del esfuerzo de re planificación;
- para cada riesgo de software identificado, el impacto inverso comprendido comparado con las pérdidas estimadas; y
- el número y magnitud los impactos adversos mayores no anticipados al proyecto del software seguidos en el tiempo.

Verificación de la implementación

Verificación 1

Son revisadas periódicamente las actividades para la administración del proyecto de software con el Administrador Gerencial.

Refiérase a la verificación 1 del área clave de procesos Seguimiento y control del proyecto de software para prácticas que cubren el contenido típico de las revisiones de supervisión del Administrador Gerencial.

Verificación 2

Las actividades para la administración del proyecto de software son revisadas con el Administrador del proyecto tanto periódicamente como en respuesta a eventos.

Refiérase a la verificación 2 del área clave de proceso de Seguimiento y control de proyectos para prácticas que cubren el típico contenido de las revisiones de seguimiento del Administrador del proyecto.

Verificación 3

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

Refiérase al área clave de proceso de Aseguramiento de calidad de software.

Como mínimo las revisiones y / o auditorías verifican:

1. Proceso para desarrollar y revisar el proceso de software definido.
2. El proceso para preparar el plan de desarrollo de software del proyecto y el plan de administración de riesgos de software.
3. Los procesos para administrar el proyecto de acuerdo con el proceso de software definido del proyecto.
4. Los procesos para recolectar y proveer los datos apropiados para la base de datos del proceso de software de la organización.
5. El proceso para usar la base de datos del proceso de software de la organización para apoyar las actividades de planeación, estimación y seguimiento del proyecto de software.

6.2 Modelo gráfico Administración Integrada de Software (AIS)

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en la Administración Integrada de Software y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

6.3 Identificación de los grupos participantes

En el modelo textual del (AIS) hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentamos una breve definición de cada uno de ellos.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de varios proyectos a la vez.

Gerente del Proyecto de Software: El gerente de software de proyecto es el individuo que trata con el Administrador del proyecto en términos de compromisos del software y quien controla todos los recursos del software para un proyecto.

Administrador del Proyecto: Es el responsable de las actividades y resultado del proyecto de software. Este administrador asigna las responsabilidades para los productos de trabajo y actividades de software, y participa en las revisiones de las actividades del AIS que se realizan en forma periódica o a causa de un evento.

Grupo de Aseguramiento de la Calidad: Revisa y / o audita las actividades y los productos de trabajo para administrar el proyecto de software y reporta resultados.

Grupo de Ingeniería de Procesos de Software (Grupo de IPS): Un grupo de especialistas que facilitan la definición, mantenimiento y mejora del proceso de software usado por la organización. En las prácticas clave, este grupo es genéricamente llamado “ el grupo responsable por las actividades de proceso de software de la organización”.

Gerente de Software: Cualquier gerente de un proyecto que tiene responsabilidad directa por el desarrollo y /o mantenimiento del software.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

Otros grupos relacionados: Es cualquier grupo involucrado en los procesos de software de la organización.

Responsables de Desarrollo del Proceso de Software del Proyecto (RDPSP): Son los individuos responsables para el desarrollo del proceso de software definido del proyecto.

Usuario Final: El individuo o grupo que usará el sistema para su uso operacional esperado cuando éste es entregado en su ambiente.

Cliente: El individuo u organización que es responsable por aceptar el producto y autorizar el pago a la organización desarrolladora.

6.4 Diagrama de actividades

Compromisos.

En esta área clave de procesos contamos con un compromiso, éste establece que el proyecto sigue una política organizacional escrita que demanda que el proyecto de software se planee y administre usando el PSE de la organización.

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que demanda que se planee y se administre el proyecto de software usando el PSE de la organización. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 6.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1 al igual que *prsofe: Proceso de software estándar*, el cual es usado para administrar y planear el proyecto. Este compromiso lo representamos como “Planear y administrar el proyecto de software C1”

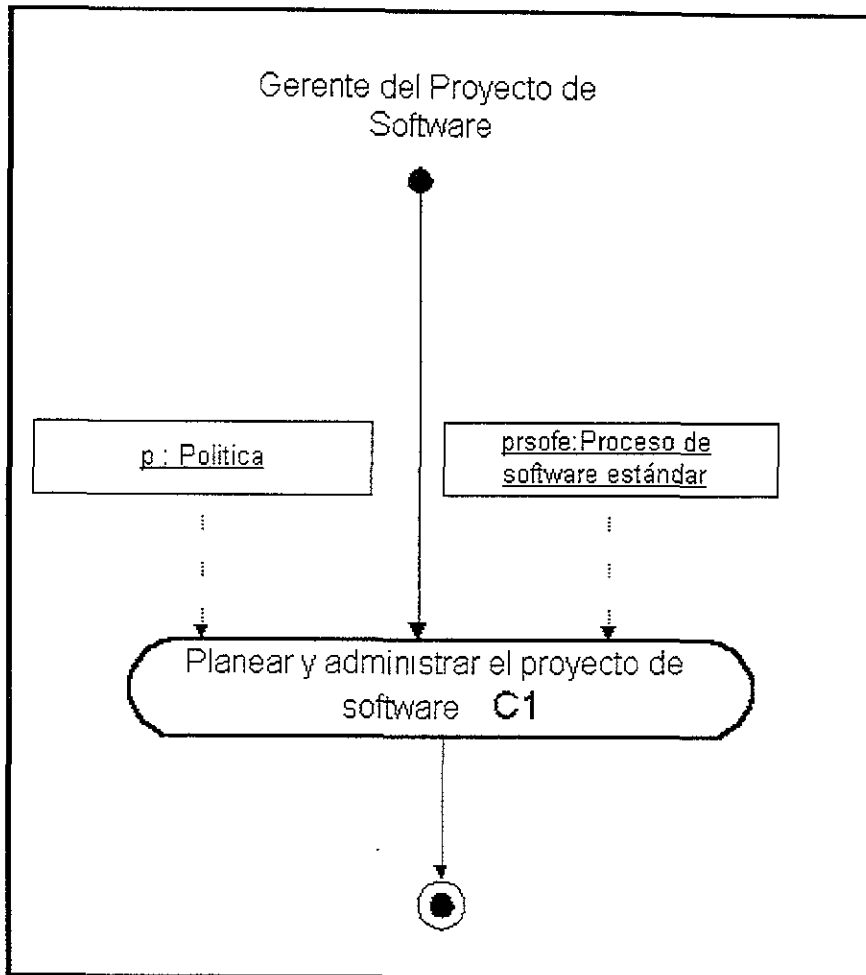


Figura 6.1: Compromiso del área clave de proceso: Administración Integrada de Software (AIS).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para la AIS.

Para modelar este hecho, la actividad se puede desglosar en otro diagrama presentado en la figura 6.2.

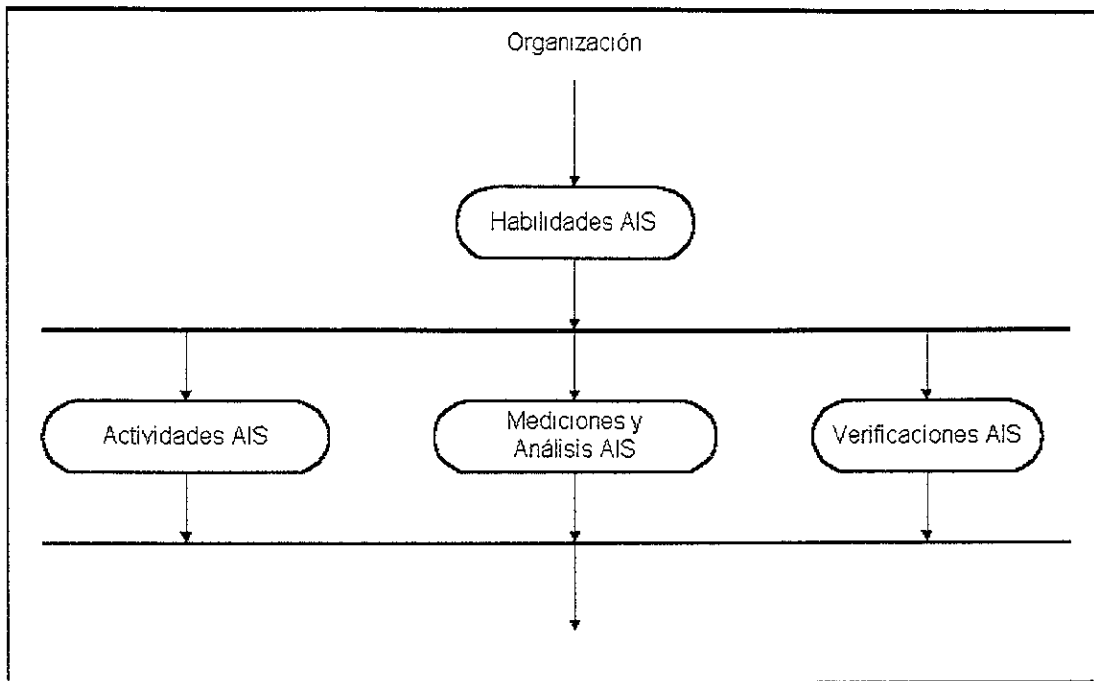


Figura 6.2: Planear y administrar el proyecto de software.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias del la (AIS). Por otro lado, las actividades de verificación y medición con su análisis, son simultaneas a las de AIS.

Habilidades.

La habilidad 1 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo de la Organización y del Gerente del Proyecto de Software. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado H1”.

La habilidad 2 habla de que los individuos responsables del desarrollo del proceso de software reciben la capacitación adecuada en como adecuar el proceso de software, también se puede interpretar que está a cargo del Grupo de IPS. Esta habilidad la representamos con la actividad “Recibir capacitación para adecuar el PSE H2”.

La habilidad 3 habla de que la administración de software recibe la capacitación adecuada para la administrar los aspectos técnicos, administrativos y del personal del proyecto. Esta habilidad la representamos con la actividad “Recibir capacitación para administración H3”.

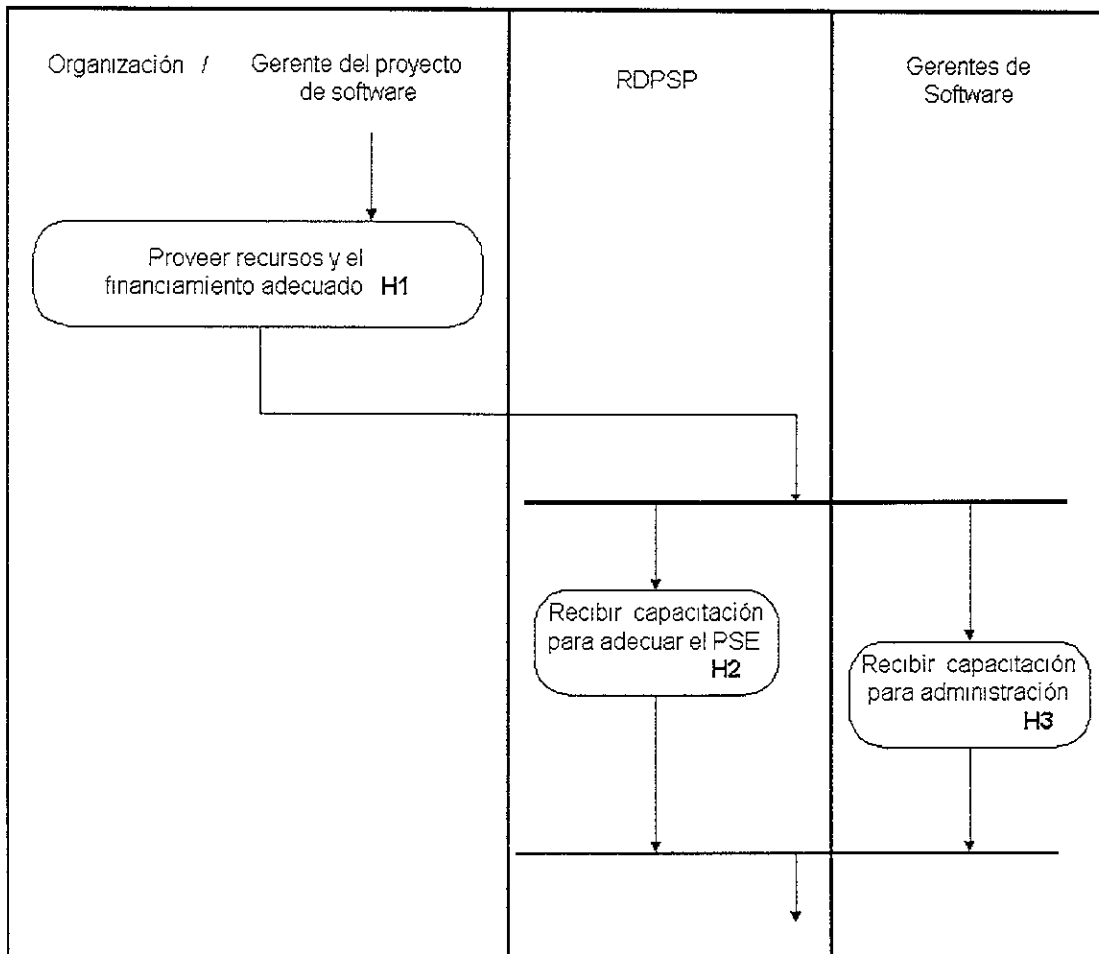


Figura 6.3: Habilidades del área clave de proceso: Administración Integrada de Software (AIS).

El diagrama de la Figura 6.3 presenta las actividades a realizarse para adquirir las habilidades para la AIS, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde.

Actividades.

La actividad 1 especifica que el proceso de software definido del proyecto se desarrolla adoptando el PSE de la organización de acuerdo a un procedimiento documentado. Esta actividad la representamos en el diagrama como “Desarrollar el proceso de software definido del proyecto A1”. Esta actividad recibe como entrada a los objetos *pro*: *Procedimiento documentado*, *pse*: *Proceso de software estándar*. Al término de esta actividad tendremos como salida a *prsofd*: *Proceso de software definido del proyecto*.

La actividad 2 especifica que cada proceso de software definido del proyecto se revisa de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Revisar cada proceso de software A2”. Esta actividad recibe como entrada a los objetos a *prsofd: Proceso de software definido del proyecto*, que fue desarrollado anteriormente y al objeto *pro: Procedimiento documentado*. Al término de esta actividad tendremos como salida a *prsofd: Proceso de software definido del proyecto [revisado]*.

La actividad 3 especifica que se desarrolla y revisa el plan de desarrollo de software del proyecto de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Desarrolla y revisar el plan de desarrollo A3”. Al término de esta actividad tendremos como salida a *plad: Plan de desarrollo*. Esta actividad está compuesta de la actividad 5 que se desglosara en la Figura 6.5.

La actividad 4 especifica que el proyecto de software es administrado de acuerdo al proceso de software definido del proyecto. Esta actividad la representamos como “Administrar el proyecto de software A4”. Esta actividad recibe como entrada al objeto *plad: Plan de desarrollo* que fue creado anteriormente en la actividad 3. Esta actividad también esta compuesta de las actividades 6,7,8,9,10, que se desglosaran en la Figura 6.6.

La actividad 11 especifica que las revisiones del proyecto se realizan periódicamente para determinar las acciones necesarias para llevar el desempeño y los resultados del proyecto. Esta actividad la representamos como “Realizar revisiones periódicas A11”.

Las actividades antes mencionadas se muestran a continuación en la figura 6.4

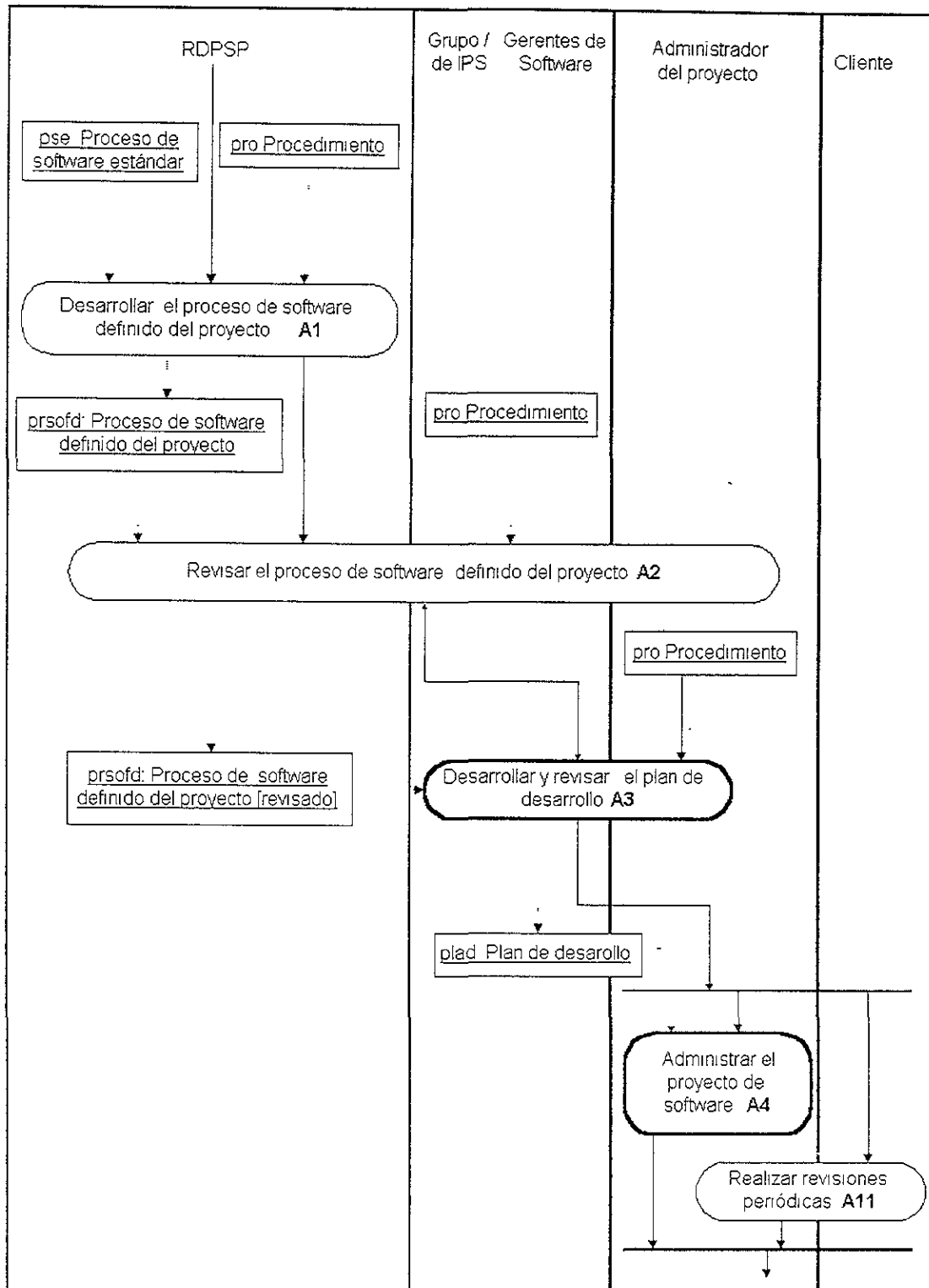


Figura 1.4: Actividades del área clave de proceso: Administración Integrada de Software (AIS).

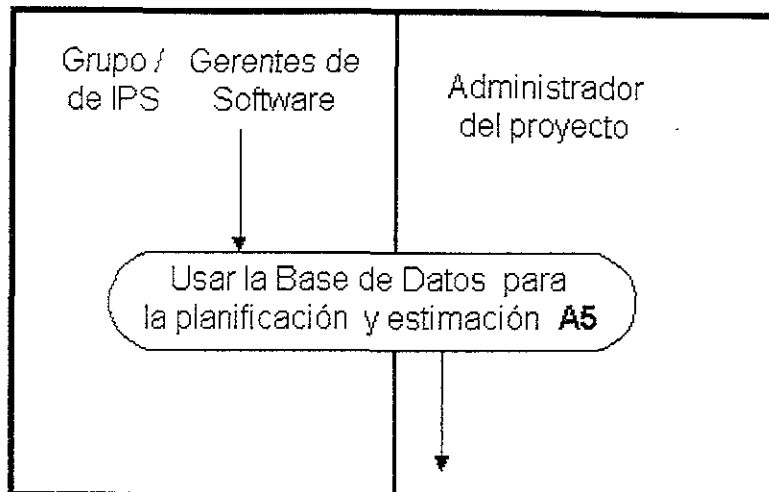


Figura 6.5: Desarrolla y revisar el plan de desarrollo.

La actividad 5 especifica que se establece que la base da datos del PSE de la organización es usada para planificación y estimación de software. Esta actividad la representamos como “Usar la Base de Datos para la planificación y estimación A5”.

La actividad 6 especifica que el tamaño de los productos de trabajo de software se administra de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Administrar el tamaño de los productos de trabajo A6”. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 7 especifica que los esfuerzos y costos del proyecto de software son administrados de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Administrar los esfuerzos y costos del proyecto A7”. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 8 especifica que los recursos críticos de la computadora del proyecto son administrados de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Administrar los recursos críticos de la computadora A8”. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 9 especifica que las dependencias criticas y rutas criticas de la programación son administradas de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Administrar las rutas criticas y dependencias criticas A9”. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 10 especifica que los riesgos de software son identificados, evaluados, documentados de acuerdo a un procedimiento documentado. Esta actividad la

representamos como “Identificar los riesgos de software A10”. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

El diagrama de la Figura 6.6 presenta las actividades que se describieron anteriormente. Por la naturaleza de la mejora continua de los procesos de las actividades arriba mencionadas de documentación, no pueden llevar un orden específico, por lo tanto en el diagrama representamos su ejecución de manera simultanea.

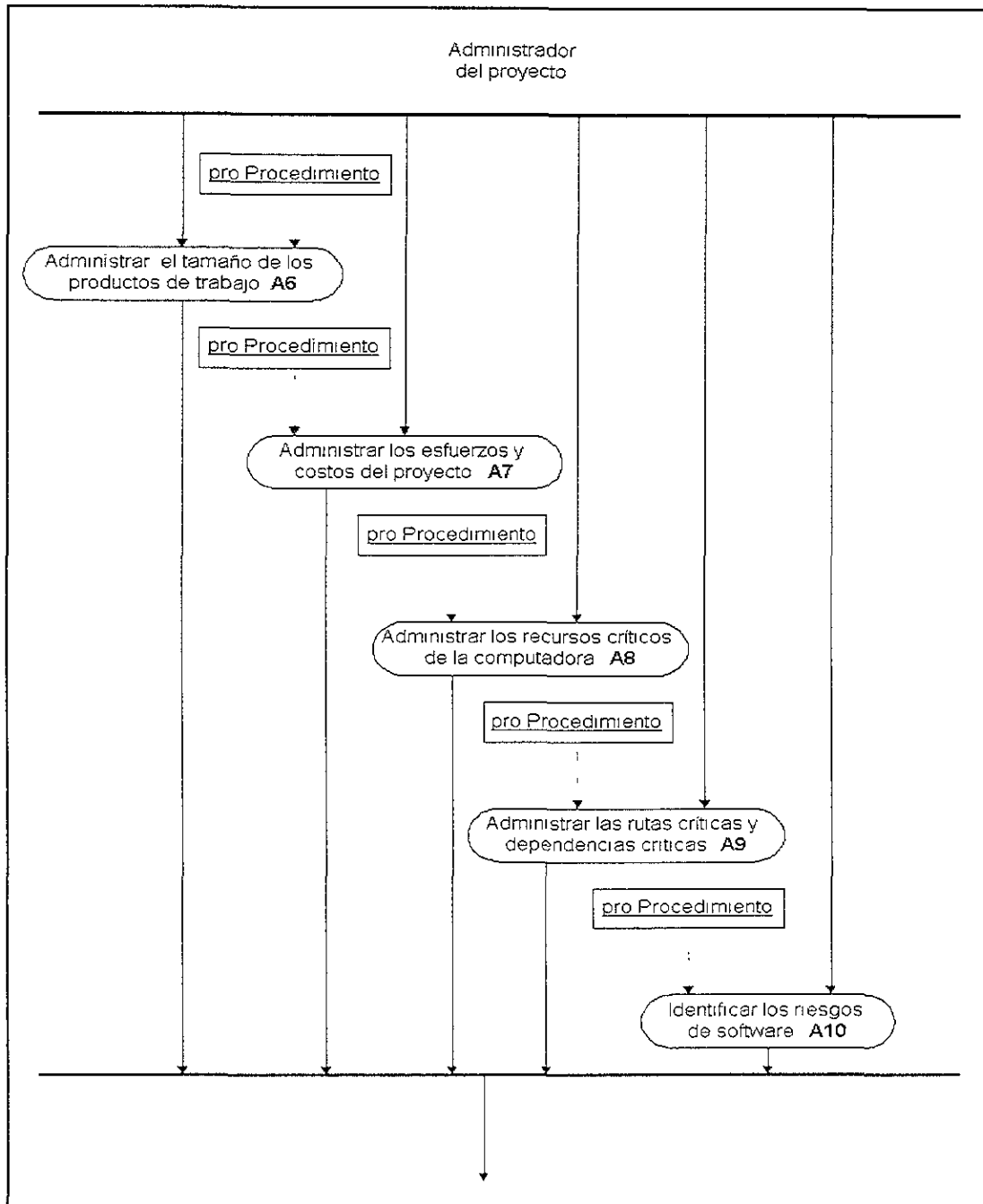


Figura 6.6: Administrar proyecto de software.

Mediciones y análisis.

El texto correspondiente dice que deben de recabarse y usar las mediciones para determinar el estado de las actividades de la AIS. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad compartida entre el Administrador Gerencial, Gerente del proyecto y el Grupo de aseguramiento de la calidad. La actividad mencionada anteriormente queda representada en la figura 6.7

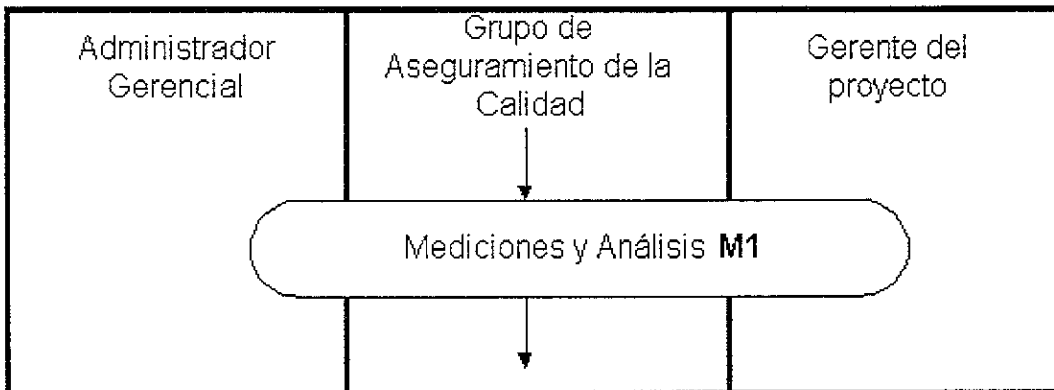


Figura 6.7 Mediciones y análisis del área clave de proceso: Administración Integrada de Software (AIS).

Verificaciones.

La verificación 1 especifica que las actividades para la administración del proyecto de software serán revisadas periódicamente por la administración gerencial y la verificación 2 por el administrador del proyecto. Las dos verificaciones las representamos con la actividad de “Revisar Actividades de la APS periódicamente V1, V2”

Finalmente en la verificación 3 el grupo de aseguramiento de calidad revisa y / o audita las actividades y productos de trabajo para administrar el proyecto de software y reporta los resultados. Esta verificación esta representada con la actividad de Revisar y / o auditar actividades y productos de trabajo”.

Las actividades mencionadas anteriormente quedan representadas en la figura 6.8

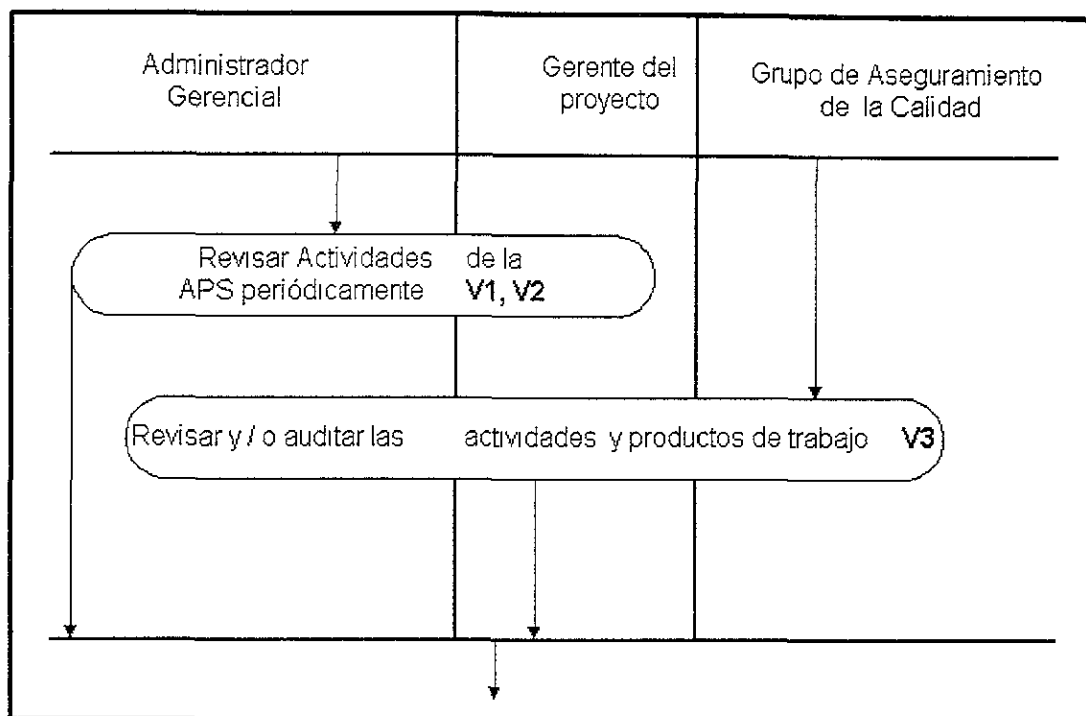


Figura 6.8 Verificaciones del área clave de proceso: Administración Integrada de Software (AIS).

Capítulo 7 Ingeniería de Productos de Software

En la Ingeniería de Productos de Software se realizan las tareas de ingeniería de software para desarrollar y mantener el software. Estas tareas son analizar los requisitos, desarrollar los requisitos, diseñar el software, la implementación del código fuente, integrar los componentes y realizar las pruebas para verificar que cumplen y satisfacen las necesidades de los clientes.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico que abstrae la información contenida en el texto.

El modelo gráfico incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

7.1 Ingeniería de Productos de Software

Una área clave de proceso para el nivel 3: Definido

El propósito de la Ingeniería de Productos de Software es realizar consistentemente un proceso de ingeniería bien definido que integra todas las actividades de ingeniería para producir productos de software correctos, consistentes, eficazmente y con eficiencia.

La ingeniería de productos de software involucra la realización de las tareas de ingeniería para construir y mantener el software usando el proceso de software definido del proyecto (el cual es descrito en la área clave de procesos Administración Integrada de Software) y los métodos y herramientas apropiadas.

Las tareas de ingeniería de software incluyen analizar los requisitos del sistema asignados al software (estos requisitos del sistema son descritos en la área clave de procesos Administración de Requisitos de Software), desarrollar los requisitos del software, desarrollar la arquitectura del software, diseñar el software, implementar el código del software, integrar los componentes de software y probar el software para verificar que satisface los requisitos especificados (es decir los requisitos del sistema asignados al software y los requisitos de software).

La documentación requerida para realizar las tareas de ingeniería de software (por ejemplo el documento de los requisitos de software, el documento de diseño de software, el plan de pruebas y los procedimientos de prueba) es desarrollada y revisada para asegurar que cada tarea abarca los resultados de las tareas predecesoras y que los resultados producidos sean apropiados para las tareas subsecuentes (incluyendo las tareas de operar y mantener el software). Cuando los cambios son aprobados, los productos de trabajo de software, los planes, compromisos, procesos y las actividades afectadas son revisadas para reflejar los cambios aprobados.

Metas

- | | |
|---------------|--|
| Meta 1 | Las tareas de ingeniería de software son definidas, integradas y realizadas consistentemente para producir el software. |
| Meta 2 | Los productos de trabajo de software se mantienen consistentes unos con otros. |

Compromisos a Realizar

Compromiso 1

El proyecto sigue una política de organización escrita para realizar las actividades de ingeniería de software.

Esta política típicamente especifica que:

1. Las tareas de ingeniería de software son realizadas de acuerdo con el proceso de software definido del proyecto.

Refiérase a las actividades 1 y 2 del área clave de proceso Administración Integrada de Software para prácticas que cubren el proceso de software definido del proyecto.

2. Se usan métodos y herramientas apropiadas para construir y mantener los productos de software.
3. Los planes de software, tareas y productos son trazables a los requisitos del sistema asignados al software.

Los Requisitos del sistema asignados al software son mencionados como "requisitos asignados" en estas prácticas.

Refiérase al área clave de proceso de Administración de Requisitos para prácticas que cubren los requisitos del sistema asignados al software.

Habilidades a realizar

Habilidad 1

Se proveen los recursos y el financiamiento adecuado para realizar las tareas de ingeniería de software.

1. Se disponen de individuos experimentados para realizar las diferentes tareas de ingeniería de software incluyendo:
 - ☐ análisis de los requisitos de software,
 - ☐ diseño de software,
 - ☐ codificación,
 - ☐ pruebas, y

□ mantenimiento de software.

2. Se hacen disponibles herramientas para apoyar las tareas de ingeniería de software.

Ejemplos de herramientas generales de apoyo:

- estaciones de trabajo,
- sistemas manejadores de bases de datos,
- herramientas de ayuda en línea,
- herramientas gráficas,
- herramientas de documentación interactivo, y
- sistemas de procesadores de palabras.

Ejemplos de herramientas de apoyo para el análisis de los requisitos de software incluyen:

- herramientas de seguimiento de requisitos,
- herramientas de especificación,
- herramientas de prototipos,
- herramientas de modelado, y
- herramientas de simulación.

Ejemplos de herramientas de apoyo para el diseño de software incluyen:

- herramientas de especificación,
- herramientas de prototipos,
- herramientas de simulación, y
- lenguajes de diseño de programas.

Ejemplos de herramientas de apoyo para la codificación incluyen:

- editores,
- compiladores,
- generadores de referencias cruzadas, y
- pretty printers.

Ejemplos de herramientas de apoyo para las pruebas de software incluyen:

- herramientas de administración de pruebas,
- generadores de pruebas,
- manejadores de prueba,
- perfiles de pruebas,
- depuradores simbólicos, y
- analizadores de cobertura de pruebas.

Habilidad 2

Los miembros del personal técnico de ingeniería de software, reciben la capacitación necesaria para realizar sus tareas técnicas.

Los miembros del personal técnico de ingeniería de software deben recibir la capacitación en el dominio de la aplicación.

Ejemplos de capacitación en el análisis de los requisitos de software incluyen:

- principios de análisis de requisitos de software;
- los requisitos de software existentes para cualquier software a ser mantenido;
- habilidades para entrevistar usuarios finales y expertos en el dominio de la aplicación para establecer los requisitos de software (es decir obtener los requisitos); y
- El uso de herramientas, métodos, convenciones y estándares seleccionados por proyecto, para analizar los requisitos de software.

Ejemplos de capacitación en el diseño de software incluyen:

- conceptos de diseño;
- los diseños existentes para cualquier software existente a ser mantenido; y

- el uso de herramientas, métodos, convenciones y estándares seleccionados por proyecto, para diseñar software.

Ejemplos de capacitación incluyen:

- los lenguajes de programación seleccionados;
- revisar el código fuente existente para cualquier software existente a ser mantenido;
- el uso de herramientas, métodos, convenciones y estándares seleccionados por el proyecto, para la programación; y
- técnicas de pruebas unitarias.

Ejemplos de capacitación en pruebas de software y otras técnicas de verificación incluyen:

- métodos de verificación (análisis, demostración, así como también otras técnicas de verificación);
- planificación de pruebas;
- el uso de herramientas, métodos, convenciones y estándares seleccionados por el proyecto, para probar y verificar el software;
- criterios de inicio y finalización de las pruebas; y
- medición de la cobertura de las pruebas.

Refiérase al área clave de procesos Programa de Capacitación.

Habilidad 3

Los miembros del personal técnico de ingeniería de software reciben orientación relacionada con las disciplinas de software.

Ejemplos de disciplinas de ingeniería de software relacionadas incluyen:

- análisis de requisitos de software,
- diseño de software,
- codificación.

- pruebas,
- administración de la configuración, y
- aseguramiento de la calidad de software.

Refiérase al área clave de procesos Programa de Capacitación

Habilidad 4

El Administrador del proyecto y todos los gerentes de software reciben orientación en aspectos técnicos del proyecto de software.

Ejemplos de orientación incluyen:

- métodos y herramientas de ingeniería de software,
- el dominio de la aplicación,
- software entregable y no entregable y sus productos de trabajo asociados, y
- guías sobre como administrar el proyecto usando los métodos y herramientas elegidas.

Refiérase al área clave de proceso Programa de Capacitación.

Actividades a realizar

Actividad 1

Se incorporan al proceso de software definido del proyecto, métodos y herramientas apropiadas de la ingeniería de software.

Refiérase a las actividades 1 y 2 del área clave de procesos de Administración Integrada de Software para prácticas que cubren el proceso de software definido del proyecto.

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

2. Se seleccionan métodos y herramientas apropiadas para su uso en el proyecto de software.

La selección de métodos y herramientas candidatas se hace con base en su aplicabilidad a los estándares de la organización, el proceso de software definido del proyecto, en base a las habilidades existentes, disponibilidad de capacitación, requisitos contractuales, poder, facilidad de uso, y servicio de soporte.

- Se documentada la razón para seleccionar una herramienta o método.
3. Se seleccionan y se usan los modelos de administración de la configuración apropiados al proyecto de software.

Ejemplos de modelos de administración de la configuración incluyen:

- modelos de check-out / check-in,
- modelos de composición,
- modelos de transición, y
- modelos de conjunto de cambios.

4. Las herramientas usadas para desarrollar y mantener los productos de software son puestas bajo la administración de la configuración.

Refiérase al área clave de procesos de Administración de la Configuración de Software.

Actividad 2

Se desarrollan, mantienen, documentan y verifican los requisitos de software analizando sistemáticamente los requisitos asignados de acuerdo al proceso de software definido del proyecto.

1. Los individuos involucrados en el desarrollo de los requisitos de software revisan los requisitos asignados para asegurar que los problemas que afecten el análisis de los requisitos de software sean identificados y resueltos.

Los requisitos de software cubren las funciones de software y desempeño, las interfaces tanto para hardware como para software y otros componentes del sistema (por ejemplo humanas).

2. Se usan métodos efectivos para el análisis de requisitos para identificar y derivar los requisitos de software.

Ejemplos de métodos para el análisis de requisitos incluyen:

- descomposición funcional,
- descomposición orientada a objetos,
- tradeoff,
- simulación,
- modelación,
- prototipos, y
- generación de escenarios.

3. Los resultados del análisis de requisitos y la razón para la alternativa seleccionada son documentados.
4. Los requisitos del software son analizados para asegurar que estos sean posibles y apropiados para implementar en el software, claramente establecidos, consistentes unos con otros, verificables y completos (cuando se consideran como un conjunto).
 - Los problemas con los requisitos de software son identificados y revisados con el grupo responsable de los requisitos del sistema, se hacen los cambios apropiados a los requisitos asignados y a los requisitos del software.

Refiérase al área clave de procesos de Administración de Requisitos.

5. Los requisitos de software son documentados.

6. El grupo responsable por las pruebas del sistema y aceptación del software analiza cada requisito de software para verificar si es que puede ser probado.
7. Son identificados y documentados los métodos para verificar y validar que cada requisito de software está satisfecho.

Ejemplos de métodos de verificación y validación incluyen:

- demostración,
- pruebas de sistema,
- pruebas de aceptación,
- análisis, e
- inspección.

8. El documento de los requisitos de software es sometido a las revisiones entre colegas antes de que se considere completo.

Refiérase al área clave de procesos de las Revisiones Entre Colegas.

9. El documento de los requisitos de software es revisado y aprobado.

Ejemplos de individuos que revisan y aprueban el documento de los requisitos de software incluyen;

- el administrador del proyecto,
- el administrador de ingeniería de sistemas,
- el administrador del proyecto de software, y
- el administrador de pruebas de software.

10. El documento de los requisitos de software es revisado con el cliente y los usuarios finales según lo apropiado.

Los usuarios finales mencionados en estas prácticas son usuarios finales designados por el cliente o representantes de los usuarios finales.

11. El documento de requisitos de software es puesto bajo la administración de configuración.

Refiérase al área clave de procesos de Administración de Configuración de Software.

12. Los requisitos de software son modificados apropiadamente toda vez que los requisitos asignados cambian.

Refiérase al área clave de procesos de Administración de Requisitos.

Actividad 3

Se desarrolla, mantiene, documenta y verifica el diseño del software de acuerdo al proceso de software definido del proyecto para adaptarse a los requisitos de software y formar un marco de referencia para la codificación.

El diseño de software consiste de la arquitectura de software y el diseño detallado del software.

1. Se revisan y se desarrollan los criterios de diseño.

Ejemplos de criterios de diseño incluyen:

- comprobable,
- adherencias a estándares de diseño,
- facilidad de construcción,
- simplicidad, y
- facilidad de planificación.

2. Los individuos involucrados en el diseño de software revisan los requisitos de software para asegurar que los problemas que afectan el diseño de software sean identificados y resueltos.
3. Se usan estándares de aplicación cuando sea lo apropiado.

Ejemplos de estándares de aplicación incluyen:

- estándares para interfaces de sistemas operativos,
- estándares para interfaces computadora-humano,
- estándares para interfaces para redes.

4. Son usados métodos efectivos para diseñar el software.

Ejemplos de métodos para diseñar software incluyen:

- prototipos,
- modelos estructurados,
- reuso de diseño,
- diseño orientado a objetos, y
- análisis esencial de sistemas.

5. La arquitectura del software se desarrolla tempranamente, dentro de las restricciones del ciclo de vida de software y la tecnología en uso.

La arquitectura del software establece el marco de referencia de alto nivel, interfaces internas y externas bien definidas.

6. La arquitectura del software es revisada para asegurar que los problemas de arquitectura que afectan el diseño detallado del software están identificados y resueltos.
7. El diseño detallado de software es desarrollado con base en la arquitectura del software.
8. El diseño de software (es decir la arquitectura de software y el diseño detallado) está documentado.

- La documentación del diseño de software cubre los componentes de software, las interfaces internas entre los componentes de software, y las interfaces de software con otros sistemas de software, con el hardware y otros componentes de sistemas (por ejemplo humanos).
9. El documento de diseño del software es sometido a las revisiones entre colegas antes de que el diseño sea considerado completo.

Refiérase al área clave de procesos de las Revisiones Entre Colegas.

10. El documento de diseño de software es puesto bajo la administración de configuración.

Refiérase al área clave de procesos Administración de Configuración de Software.

11. El documento de diseño de software es modificado adecuadamente cada vez que los requisitos de software cambian.

Actividad 4

El código del software es revisado, mantenido, documentado y verificado, de acuerdo al proceso de software definido del proyecto, para implementar los requisitos de software y el diseño de software.

1. Los individuos involucrados en la codificación revisan los requisitos de software y el diseño de software para asegurar que los problemas que afectan la codificación están identificados y resueltos.
2. Se usan métodos efectivos para codificar el software.

Ejemplos de métodos de programación incluyen:

- programación estructurada, y
- reuso de código.

3. La secuencia de desarrollo de las unidades de código está basada en un plan que toma en cuenta factores tales como puntos críticos, dificultad, problemas de pruebas e integración, y necesidades del cliente y los usuarios finales según lo adecuado.
4. Cada unidad de código es sometida a las Revisiones entre colegas y pruebas unitarias antes de que la unidad se considere completa.

Refiérase al área clave de procesos Revisiones Entre Colegas.

5. El código es puesto bajo la administración de configuración.

Refiérase al área clave de procesos de Administración de Configuración de Software.

6. El código es modificado adecuadamente cada vez que los requisitos y / o el diseño del software cambie.

Actividad 5

Las pruebas del software se realizan de acuerdo con el proceso de software definido del proyecto.

1. Se desarrollan y revisan criterios de pruebas con el cliente y los usuarios finales, según corresponda.
2. Se usan métodos efectivos para probar el software.
3. La disponibilidad de las pruebas es determinada con base en:
 - ☐ el nivel de pruebas efectuado.

Ejemplos de niveles de pruebas incluyen:

- pruebas unitarias,
- pruebas de integración,
- pruebas de sistema, y

- pruebas de aceptación.

- la estrategia de pruebas seleccionada y

Ejemplos de estrategias de pruebas incluyen.

- funcional (la caja negra),
- estructural (la caja blanca), y
- estadística.

- La cobertura de las pruebas a ser lograda:

Ejemplos de cobertura de pruebas incluyen:

- cobertura de instrucciones,
- cobertura de rutas,
- coberturas de ramificaciones, y
- perfiles de uso.

4. Se establecen y se usan criterios de finalización de pruebas, para cada nivel de pruebas de software.

Ejemplos de criterios para determinar la finalización de las pruebas incluyen:

- las unidades del software han completado con éxito las revisiones entre colegas del código y las pruebas unitarias antes de que estas comenzaran sus pruebas de integración,
- el software ha completado exitosamente las pruebas de integración antes de que estas entren a las pruebas del sistema, y
- una revisión de la finalización de las pruebas se sostiene antes de que el software entre a sus pruebas de aceptación.

5. Se realizan pruebas de regresión, según lo adecuado en cada nivel de prueba siempre y cuando el software que está siendo probado o su medio ambiente cambian.

6. El plan de pruebas, los procedimientos de pruebas y los casos de prueba son sometidos a las revisiones entre colegas antes de que estos sean considerados listos para su uso.

Refiérase al área clave de procesos de las Revisiones Entre Colegas.

7. Los planes de prueba, los procedimientos de prueba y los casos de prueba son administrados y controlados.

Administrado y controlado implica que la versión del producto de trabajo de software en un determinado momento (pasado o presente) es conocida y los cambios son incorporados y controlados.

Si se desea un mayor grado de control del que es aplicado por el administrado y controlado el producto de trabajo puede ser puesto bajo la disciplina total de la Administración de configuración como es descrita en el área clave de proceso Administración de Configuración de Software.

8. Los planes de prueba, los procedimientos de prueba y los casos de prueba, son modificados apropiadamente siempre y cuando cambien los requisitos asignados, los requisitos del software, el diseño de software o el código bajo prueba.

Actividad 6

Las pruebas de integración del software son planeadas y realizadas de acuerdo al proceso de software definido del proyecto.

1. Los planes para las pruebas de integración son documentados y están basados en el plan de desarrollo de software.
2. Los casos de pruebas de integración y procedimientos de prueba son revisados con los individuos responsables por los requisitos del software, diseño del software y las pruebas del sistema y aceptación.

3. Las pruebas de integración del software son realizadas contra una versión designada del documento de requisitos de software y el documento de diseño de software.

Actividad 7

Se planifican y realizan pruebas de sistema y de aceptación del software para demostrar que el software satisface sus requisitos.

Las pruebas del sistema son realizadas para asegurar que el software satisface los requisitos del software.

Las pruebas de aceptación son realizadas demostrar a los clientes y los usuarios finales que el software satisface los requisitos asignados.

1. Los recursos para las pruebas del software se asignan tempranamente para proveer la preparación adecuada para las pruebas.

Ejemplos de actividades requeridas para preparar las pruebas incluyen:

- preparación de la documentación de las pruebas,
- calendarización de los recursos de las pruebas,
- desarrollar manejadores de pruebas, y
- desarrollar simuladores.

2. Las pruebas del sistema y aceptación son documentadas en el plan de pruebas, el cual es revisado con, y aprobado por, el cliente y los usuarios finales según corresponda, el plan de pruebas cubre:

- ☐ todas las pruebas y verificaciones;
- ☐ las responsabilidades de la organización desarrolladora, los subcontratistas, los clientes y los usuarios finales según corresponda;
- ☐ instalaciones de prueba, equipo de pruebas y condiciones para apoyar las pruebas; y
- ☐ criterios de aceptación.

3. Los casos y procedimientos de prueba son planeados y preparados por el grupo de pruebas que es independiente de los desarrolladores de software.
4. Los casos de prueba son documentados y son revisados con y aprobados por el cliente y los usuarios finales según corresponda, antes de que las pruebas comiencen.
5. Las pruebas de software son realizadas contra una línea base del software y una línea base de la documentación de los requisitos asignados y los requisitos de software.
6. Los problemas identificados durante las pruebas son documentados y seguidos hasta terminarlos.

Refiérase a la actividad 9 del área clave de procesos Seguimiento y control de proyecto de software y la actividad 5 del área clave de procesos de Administración de Configuración de Software para prácticas que cubren la documentación y seguimiento de problemas.

7. Los resultados de las pruebas son documentados y usados como las bases para determinar cuando el software satisface sus requisitos.
8. Los resultados de las pruebas son administrados y controlados.

Actividad 8

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

1. Se usan métodos y herramientas apropiadas para desarrollar la documentación.

Ejemplos de métodos y herramientas incluyen:

- procesadores de texto,
- casos de estudio, y
- reuso de la documentación.

2. Especialistas en la documentación participan activamente en la planificación, desarrollo y mantenimiento la documentación
3. Se desarrollan y se hacen disponibles tempranamente las versiones preliminares de la documentación en el ciclo de vida del software para el cliente, usuarios finales y personal de mantenimiento del software, según lo adecuado para revisar y proveer la retroalimentación.

Ejemplos de documentación incluyen:

- documentación de capacitación,
- documentación en línea,
- manuales de usuarios,
- manual del operador, y
- manual de mantenimiento.

4. Las versiones finales de la documentación son verificadas contra la línea de base de software para las pruebas de aceptación del software.
5. La documentación se somete a las revisiones entre colegas.

Refiérase al área clave de procesos Revisiones Entre Colegas.

6. La documentación es administrada y controlada.
7. La documentación final es revisada y aprobada por el cliente y los usuarios finales y el personal de mantenimiento, según corresponda.

Actividad 9

Los defectos identificados en las revisiones entre colegas y las pruebas se recolectan y se analizan los datos de acuerdo con el proceso de software definido del proyecto.

Ejemplos de los tipos de datos a ser recolectados y analizados incluyen:

- descripción de los defectos,
- categoría del defecto,
- severidad del defecto,
- unidades que contiene el defecto,
- unidades afectadas por el defecto,
- actividades donde se introdujo el defecto,
- revisiones entre colegas o casos de prueba que identificaron el defecto,
- descripción del escenario que estaba corriendo cuando se identificó el defecto, y
- resultados esperados y los resultados reales que identificaron el defecto.

Actividad 10

Se mantiene la consistencia a lo largo de los productos de trabajo de software incluyendo los planes de software, descripción del proceso, requisitos asignados, requisitos de software, diseño de software, código, planes y procedimientos de pruebas.

1. Los productos de software son documentados y la documentación rápidamente se hace disponible.
2. Los requisitos de software, diseño, código y casos de prueba, son trazados hacia el origen de los cuales fueron derivados y a los productos de las siguientes actividades de ingeniería de software.
3. Se administra y controla la documentación donde se trazan los requisitos asignados a través de los requisitos de software, el diseño, el código y casos de pruebas.
4. Conforme la comprensión del software va mejorando, se proponen y analizan los cambios a los productos de trabajo de software, planes, procesos, descripciones de procesos y actividades.
 - ☐ El proyecto determina el impacto de los cambios antes de que los cambios se realicen.
 - ☐ Se aprueban e incorporan donde se necesitan los cambios a los requisitos asignados, antes de que se modifique cualquier producto de trabajo de software o actividad.
 - ☐ Son coordinados los cambios a todos los productos de software, planes, descripción de procesos y actividades.

- Los cambios son negociados con y comunicados a los grupos afectados.

Ejemplos de personas y grupos afectados incluyen:

- ingeniería de software,
- estimación de software,
- pruebas de sistema,
- aseguramiento de la calidad de software,
- administración de la configuración de software,
- administración de contrato, y
- apoyo de la documentación.

- Los cambios son seguidos hasta terminarlos.

Medición y análisis

Medición 1

Se hacen y se usan las mediciones para determinar la funcionalidad y calidad de los productos de software.

Ejemplos de mediciones incluyen:

- números, tipos y severidad de los defectos identificados en los productos de software seguidos, tanto acumulativamente como por etapas; y
- requisitos asignados resumidos por categoría (por ejemplo seguridad, configuración del sistema desempeño y confiabilidad) y trazados a los requisitos de software y casos de prueba del sistema.

Medición 2

Se hacen y se usan las mediciones para determinar el estado de las actividades de la ingeniería de productos de software.

Ejemplos de mediciones incluyen:

- el estado de cada requisito asignado a lo largo de la vida del proyecto;

- reportes de problemas por severidad y periodo de tiempo que estos están abiertos;
- cambios de las actividades para los requisitos asignados;
- esfuerzo para analizar cada uno de los cambios propuestos y totales acumulativos;
- número de cambios incorporados dentro de la línea base del software por categoría (por ejemplo interfaces, seguridad, configuración del sistema, desempeño y utilidad); y
- tamaño y costo para implementar y probar los cambios incorporados, incluyendo estimaciones iniciales y los tamaños y costos reales.

Verificación de la implementación

Verificación 1

Se revisan en forma periódica las actividades de la ingeniería de productos de software con el Administrador gerencial.

Refiérase a la verificación 1 del área clave de proceso Seguimiento y control de proyectos para prácticas que cubren el típico contenido de las revisiones de seguimiento del administrador gerencial.

Verificación 2

Se revisan en forma periódica y ante eventos que lo requieran las actividades de la ingeniería de productos de software con el Administrador del proyecto.

Refiérase a la verificación 2 del área clave de proceso de Seguimiento y control de proyectos para prácticas que cubren el típico contenido de las revisiones de seguimiento del administrador gerencial.

Verificación 3

El grupo de aseguramiento de la calidad de software revisa y / o audita las actividades y los productos de trabajo de ingeniería de productos de software e informa los resultados.

Refiérase al arrea clave de procesos de Aseguramiento de la calidad de software.

Como mínimo la revisión y /o auditoría verifica que:

1. Los requisitos de software son revisados para asegurar que ellos están.
 - ☐ completos,
 - ☐ correctos,
 - ☐ consistentes,
 - ☐ factibles, y
 - ☐ verificables.
2. Los criterios de finalización y cumplimiento para cada tarea de ingeniería de software sean satisfechos.
3. Los productos de software cumplen con los estándares y los requisitos especificados.
4. Se han realizado las pruebas requeridas.
5. Las pruebas de aceptación y de sistema de software se han realizado de acuerdo a los planes y procedimientos documentados.
6. Las pruebas satisfacen sus criterios de aceptación, tal como están documentados en el plan de pruebas de software.
7. Las pruebas se han terminado y registrado satisfactoriamente.
8. Los problemas y defectos detectados son documentados, seguidos y abordados.
9. Se realiza el seguimiento de los requisitos asignados, a través de los requisitos del software, el diseño, el código y los casos de prueba.
10. La documentación usada para operar y mantener el software es verificada contra una línea base de software y cualquier requisito asignado antes de que el producto de software sea liberado al cliente o usuarios finales.

7.2 Modelo gráfico Ingeniería de Productos de software (IPS)

La construcción del modelo gráfico se inicia con una identificación, en todo el texto original, de los grupos participantes en la Ingeniería de productos de software y, posteriormente, con una asignación de sus responsabilidades en los diagramas de actividades.

7.3 Identificación de los grupos participantes

En el modelo textual de la (IPS), hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave.

A continuación presentamos una breve definición de cada uno de ellos.

Grupo de Aseguramiento de la calidad: Revisa y / o audita las actividades y los productos de trabajo de ingeniería de productos de software y reporta resultados.

Grupo de Ingeniería de Software (Grupo de IS): Es la colección de individuos (tanto administradores, como personal técnico), que tienen la responsabilidad para desempeñar las actividades de mantenimiento y desarrollo de software (como análisis de requisitos, diseño, codificación y pruebas) para el proyecto. Los grupos relacionados al software, tal como el grupo de aseguramiento de calidad, el grupo de administración de configuraciones, el grupo de ingeniería de procesos de software, no están incluidos en el grupo de ingeniería de software.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de varios proyectos a la vez.

Administrador del Proyecto: Es el responsable de las actividades y resultado del proyecto de software. Este administrador asigna las responsabilidades para los productos de trabajo y actividades de software, y participa en las revisiones de las actividades del IPS.

Gerente del Proyecto de Software: El gerente de software de proyecto es el individuo que trata con el gerente del proyecto en términos de compromisos del software y quien controla todos los recursos del software para un proyecto.

Cliente: El individuo u organización que es responsable por aceptar el producto y autorizar el pago a la organización desarrolladora.

Usuario Final: El individuo o el grupo que usara el sistema para su uso operacional esperado cuando este es entregado en su ambiente.

Gerente de Software: Cualquier gerente de un proyecto que tiene responsabilidad directa por el desarrollo y /o mantenimiento del software.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

7.4 Diagrama de actividades

Compromisos.

En esta área clave de procesos contamos con un compromiso, éste establece que la organización sigue una política organizacional escrita para realizar las actividades de Ingeniería de software. Analizando el texto del compromiso 1, se deduce que la organización tiene una política. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 7.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para realizar las revisiones entre colegas, queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Realizar las actividades de Ingeniería de software C1”.

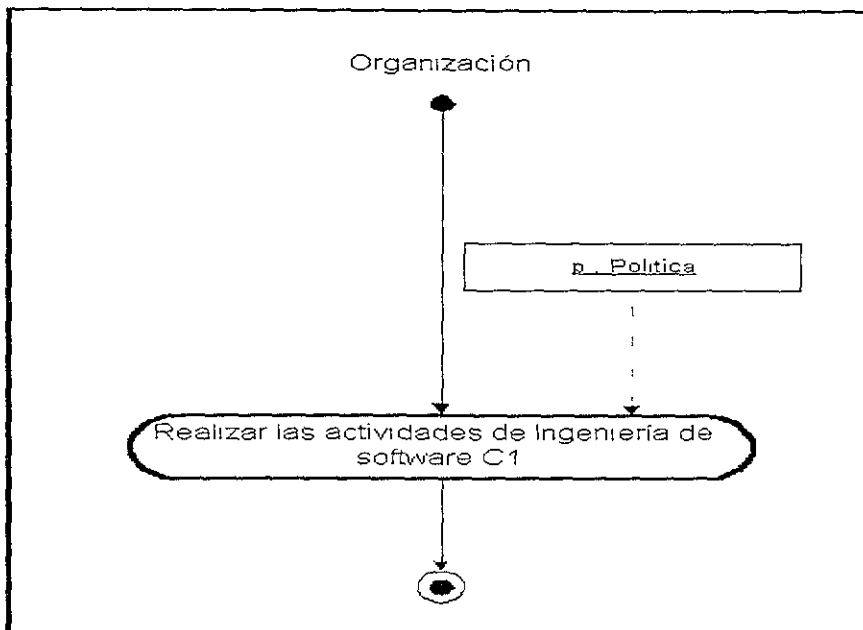


Figura 7.1 Compromiso del área clave de proceso: Ingeniería de Productos de Software (IPS).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes

grupos participantes deben tomar para realizar las actividades de Ingeniería de software.

Para modelar este hecho, la actividad se puede desglosar en otro diagrama presentado en la figura 7.2.

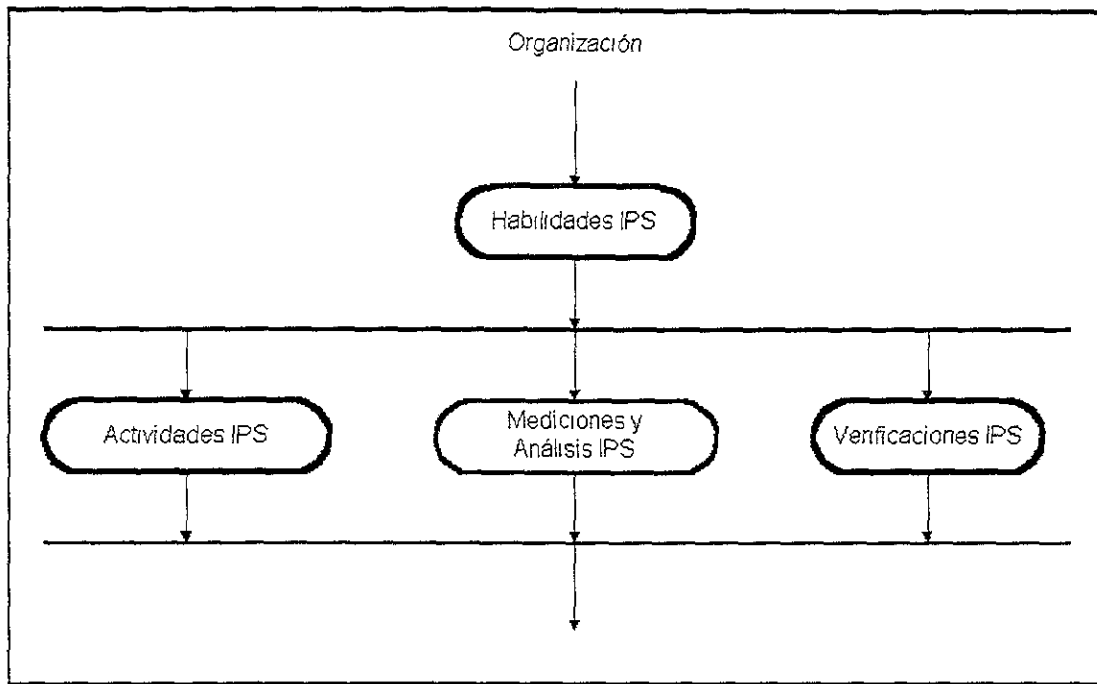


Figura 7.2 Realizar las actividades de Ingeniería de software.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de la Ingeniería de productos de software. Por otro lado, las actividades de verificación y medición con su análisis, son simultaneas a las de Ingeniería de productos de software.

Habilidades.

La habilidad 1 habla de la provisión de recursos y financiamiento, que está a cargo de la organización y del Gerente del Proyecto de software. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado H1” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación.

La habilidad 2 habla de que los Miembros del Personal Técnico de Ingeniería de Software (MPTIS), reciben la capacitación necesaria para realizar sus tareas técnicas. Esta habilidad la representamos con la actividad “Recibir capacitación en sus tareas H2 ”.

La habilidad 3 habla de que los Miembros del Personal Técnico de Ingeniería de Software reciben orientación relacionada con las disciplinas de software. Esta habilidad la representaremos con la actividad “Recibir orientación de las disciplinas de Software H3 ”.

La habilidad 4 habla de que el Administrador del proyecto y todos los gerentes de software reciben orientación en aspectos técnicos del proyecto de software.

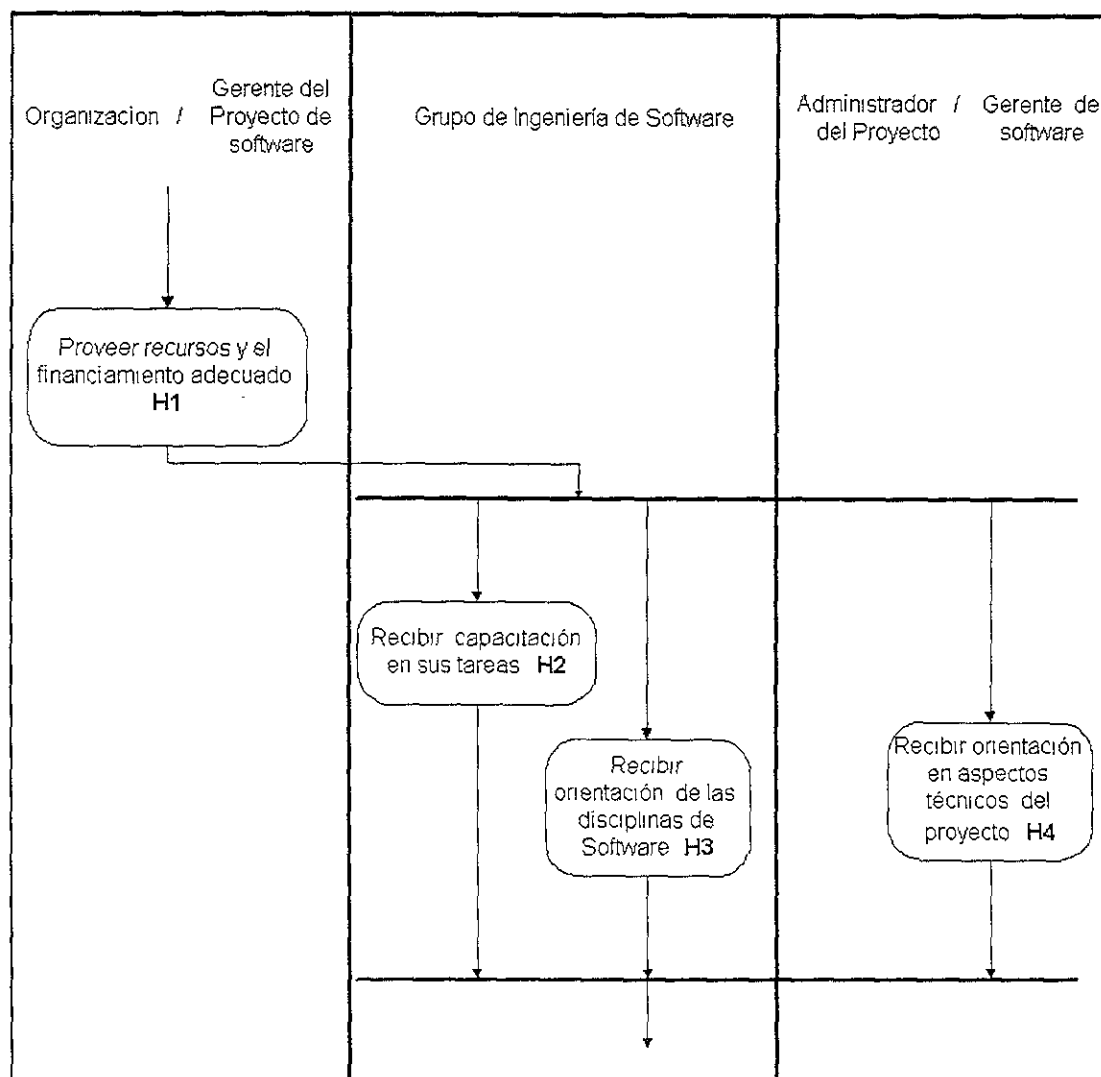


Figura 7.3: Habilidades del área clave de proceso: Ingeniería de Productos de Software (IPS).

El diagrama de la figura 7.3 presenta las actividades a realizarse para adquirir las habilidades para la IPS, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde.

Actividades.

La actividad 1 específica que se incorporan al proceso de software definido del proyecto, métodos y herramientas apropiadas de la ingeniería de software. Esta actividad la representamos en el diagrama como “Incorporar métodos y herramientas apropiadas de la ingeniería de software A1”. En el texto no dice explícitamente quien realiza todas las actividades del IPS de la organización, pero por la existencia del grupo IPS podemos concluir que debe ser su responsabilidad.

La actividad 2 específica que se deben desarrollar, mantener, documentar y verificar los requisitos de software analizando sistemáticamente los requisitos asignados de acuerdo al proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “Desarrollar, mantener, documentar y verificar los requisitos de software A2”. Al término de esta actividad tendremos como salida al objeto **rs: Requisitos de Software**. Esta actividad también es realizada por el cliente y los usuarios finales.

La actividad 3 específica que se deben desarrollar, mantener, documentar y verificar el diseño del software de acuerdo al proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “Desarrollar, mantener, documentar y verificar el diseño del software A3”. Esta actividad tiene como entrada al objeto **rs: Requisitos de Software**, y al final de esta actividad tendremos al objeto **ds: Diseño del Software**.

La actividad 4 específica que el código del software es revisado, mantenido, documentado y verificado, de acuerdo al proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “El código del software es revisado, mantenido, documentado y verificado A4”. Esta actividad tiene como entrada al objeto **ds: Diseño del Software**, y al final de esta actividad tendremos al objeto **cs: Código del Software**.

La actividad 5 específica que se deben realizar las pruebas del software de acuerdo con el proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “Realizar las pruebas de software A5”. Esta actividad tiene como entrada a los objetos **cs: Código del Software** y al objeto **rs: Requisitos de Software**. Esta actividad contiene a las actividades 6,7 que se mostraran en la figura 7.5. Las actividades 5,8 y 9 se realizan en paralelo. Al final de esta actividad se tiene al objeto **sofA:software aprobado**.

La actividad 8 específica que la documentación que será usada para operar y mantener el software es desarrollada y mantenida de acuerdo al proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “Desarrollar y mantener la documentación usada para operar y mantener el software A8”. Al término de esta actividad tendremos al objeto **dc: Documentación**.

La actividad 9 especifica que los defectos identificados en las revisiones entre colegas y las pruebas se recolectan y los datos se analizan de acuerdo con el proceso de software definido del proyecto. Esta actividad la representamos en el diagrama como “Se recolectan y analizan los datos de los defectos identificados A9”.

La actividad 10 especifica que se debe mantener la consistencia a lo largo de los productos de trabajo de software incluyendo los planes de software, descripción del proceso, requisitos asignados, requisitos de software, diseño de software, código, planes y procedimientos de pruebas. Esta actividad se realiza en paralelo con las actividades antes mencionadas excepto la actividad 1. Las actividades mencionadas anteriormente quedan representadas en la Figura 7.4

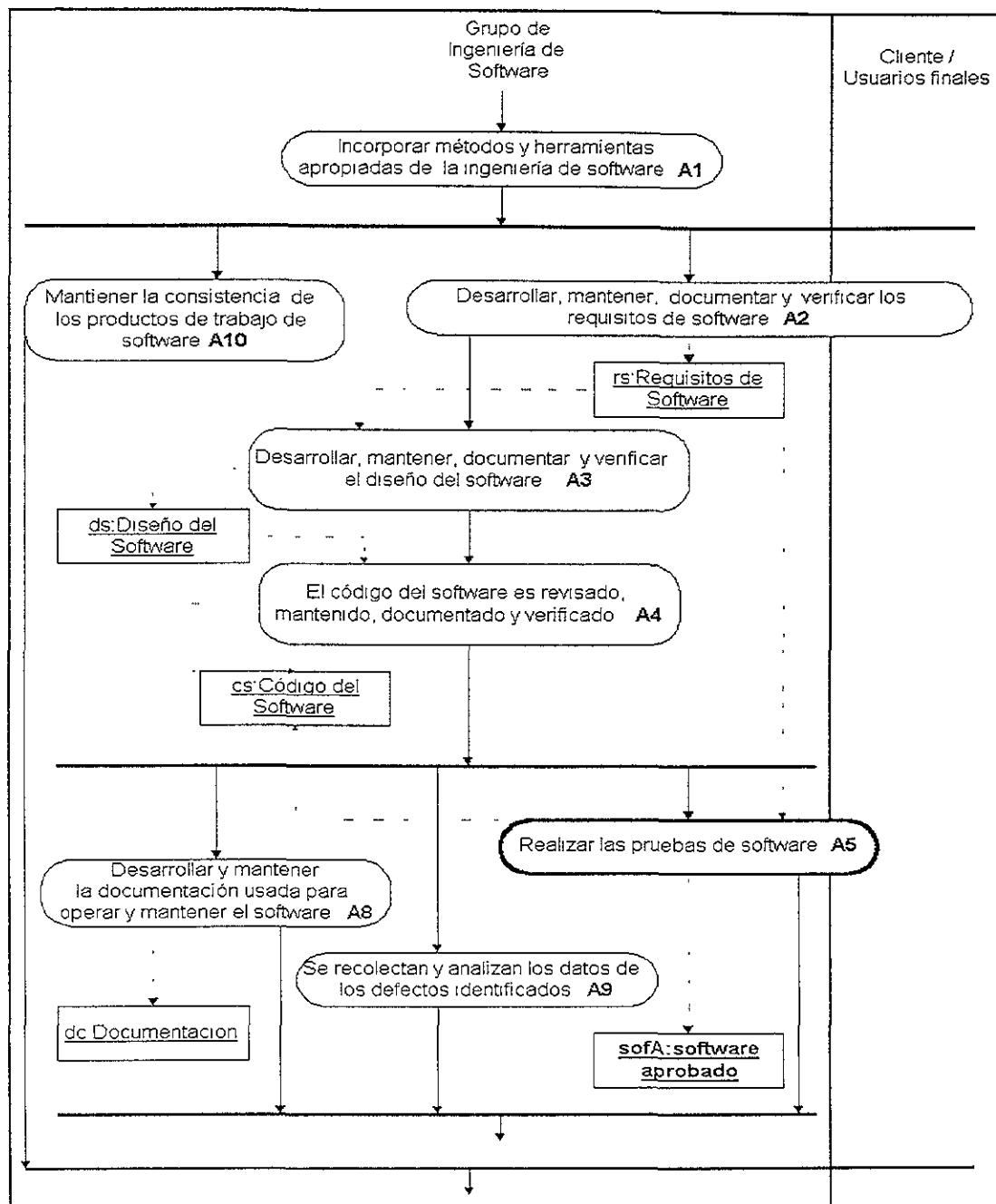


Figura 7.4: Actividades del área clave de proceso: Ingeniería de Productos de Software (IPS).

La actividad 6 especifica que las pruebas de integración del software son planeadas y realizadas de acuerdo al proceso de software definido del proyecto. Esta actividad tiene como entrada a los objetos **cs: Código del Software** y al objeto **rs: Requisitos de Software**. Esta actividad la representamos en el diagrama como “Realizar las pruebas de integración A6”. Al final de esta actividad tendremos al objeto **sof:software**.

La actividad 7 específica que se planifican y realizan pruebas de sistema y de aceptación del software para demostrar que el software satisface sus requisitos. Esta actividad la representamos en el diagrama como “Realizar las pruebas del sistema A7”. Al Comienzo de esta actividad tendremos al objeto sof:software, que es el que será probado por el cliente y los usuarios finales si cumple con los requisitos de estos junto con los Grupos de Ingeniería de software, se tendrá al final de esta actividad al objeto sofA:software aprobado.

Las actividades 6 y 7 mencionadas anteriormente quedan representadas en la Figura 7.5

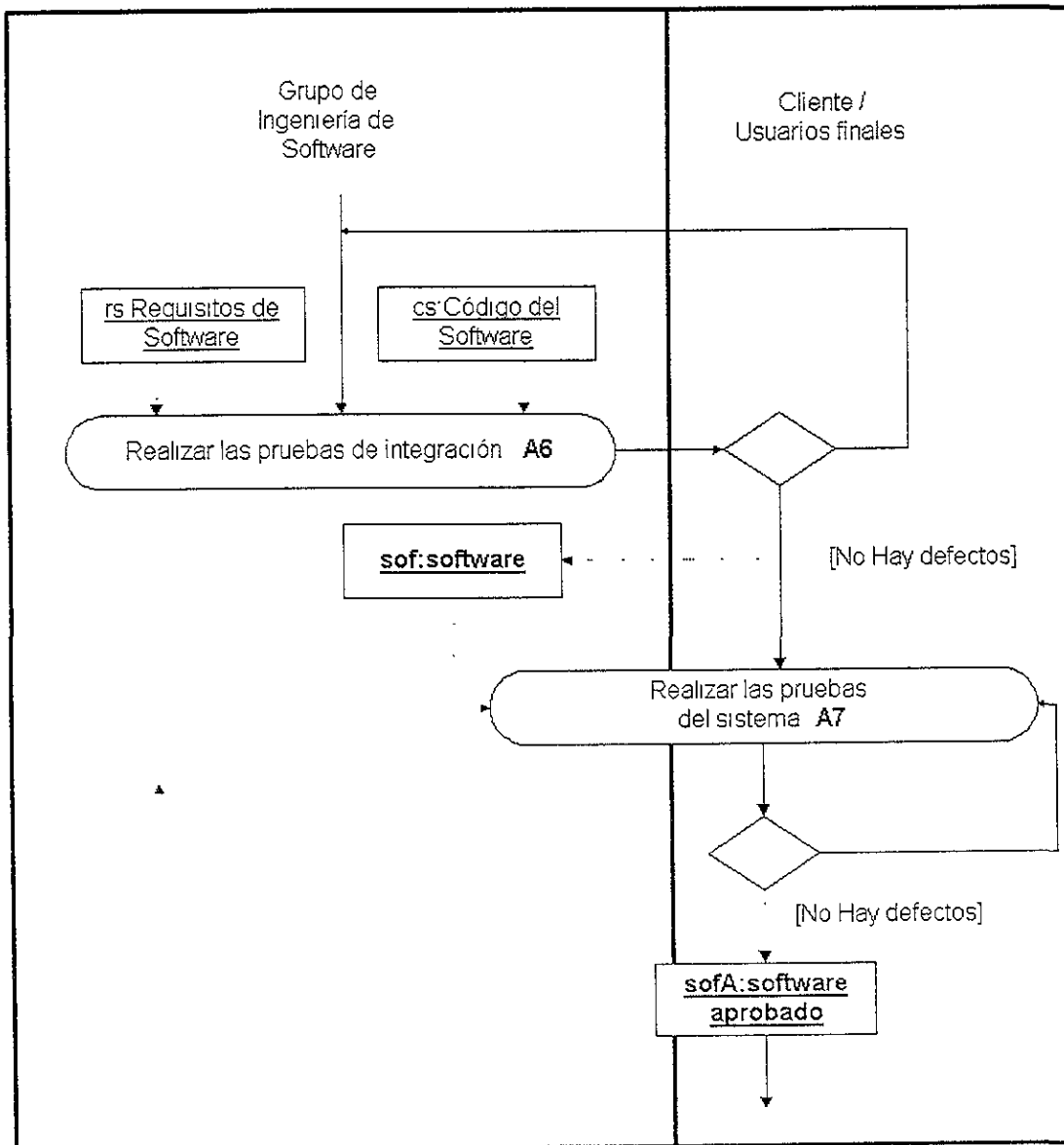


Figura 7.5: Realizar las pruebas de software.

Mediciones y análisis.

El texto correspondiente dice deben de recabarse y usar las mediciones para determinar el estado de las actividades de las IPS. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad que la realizan Grupo de Ingeniería de Software, Administrador del Proyecto y el Grupo de aseguramiento de calidad. La actividad mencionada anteriormente queda representada en la figura 7.6

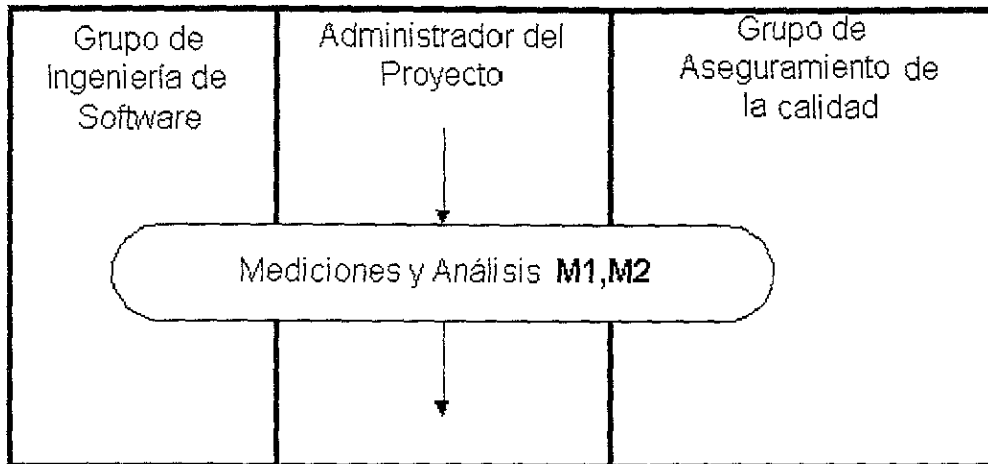


Figura 7.6 Mediciones y Análisis del área clave de proceso: Ingeniería de Productos de Software (IPS).

Verificaciones.

La verificación 1 especifica que se revisan en forma periódica las actividades de la ingeniería de productos de software con el Administrador gerencial.

La verificación 2 especifica que se revisan en forma periódica y ante eventos que lo requieran las actividades de la ingeniería de productos de software con el Administrador del proyecto.

Las verificaciones anteriores están representadas con la actividad de “Revisar Actividades para la IPS V1, V2”.

Finalmente en la verificación 3 especifica que las actividades y productos de trabajo se revisan y / o auditan y los resultados se reportan, quien la realiza es el grupo Aseguramiento de calidad. Esta verificación esta representada con la actividad de “Revisar y / o auditar las actividades y productos de trabajo V3”.

Las actividades mencionadas anteriormente quedan representadas en la Figura 7.7.

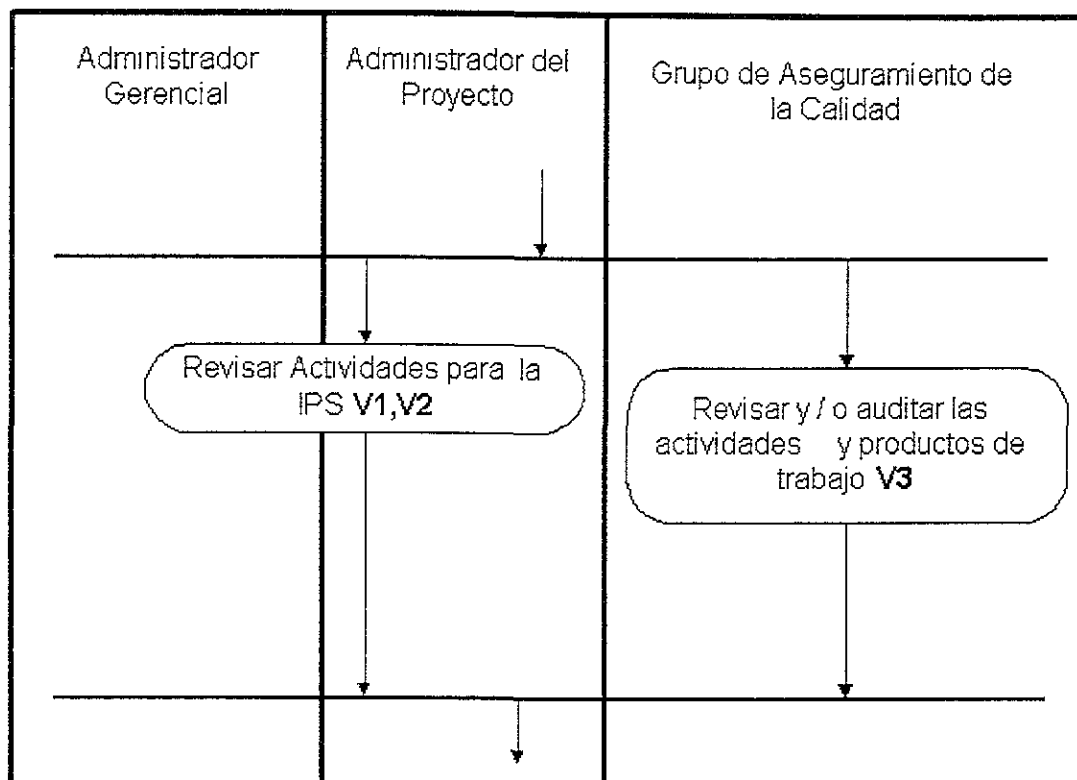


Figura 7.7 Verificaciones del área clave de proceso: Ingeniería de Productos de Software (IPS).

Capítulo 8 Coordinación Entre Grupos

En la Coordinación Entre Grupos se establecerán los mecanismos necesarios para que el grupo de ingeniería de software tenga una comunicación activa con los demás grupos de ingeniería.

El objetivo de este capítulo es presentar la traducción de esta área clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico que abstrae la información contenida en el texto.

8.1 Coordinación Entre Grupos

Un área clave de proceso para el nivel 3: definido

El propósito de la Coordinación Entre Grupos es establecer los medios para que el grupo de ingeniería de software participe activamente con los otros grupos de ingeniería, de manera que el proyecto mejore su habilidad de satisfacer las necesidades del cliente eficaz y eficientemente.

La Coordinación Entre Grupos involucra la participación de los grupos de ingeniería de software con otros grupos de ingeniería del proyecto, para cubrir los requisitos, objetivos y asuntos del sistema. Los representantes de los grupos de ingeniería del proyecto participan en el establecimiento de los requisitos, objetivos del sistema y planes de trabajo con el cliente y los usuarios finales, según lo apropiado. Estos requisitos, objetivos y planes llegan a ser las bases para todas las actividades de ingeniería.

Las interfaces e interacciones técnicas de trabajo entre los grupos son planeadas y administradas para asegurar la calidad e integridad del sistema completo. Se dirigen regularmente revisiones e intercambios técnicos con representantes del grupo de ingeniería del proyecto para asegurar que todos los grupos de ingeniería están conscientes del estado y los planes de todos los grupos y que los problemas del sistema y entre grupos reciben la atención apropiada.

Las prácticas específicas de software relacionadas a estas tareas de ingeniería son descritas en las áreas clave de proceso de Administración de Requisitos e Ingeniería de Productos de Software.

Metas

- | | |
|---------------|---|
| Meta 1 | Los requisitos del <u>cliente</u> son acordados por todos los grupos afectados. |
| Meta 2 | Los compromisos entre los <u>grupos de ingeniería</u> son acordados por todos los grupos afectados. |
| Meta 3 | Los <u>grupos de ingeniería</u> identifican, siguen y resuelven los problemas entre grupos. |

Compromisos a realizar

Compromiso 1

El proyecto sigue una política de organización escrita para establecer grupos de ingeniería interdisciplinarios.

Esta política típicamente especifica.

1. Los requisitos del sistema y los objetivos del proyecto son definidos y revisados por todos los grupos afectados.

Ejemplos de grupos afectados incluyen:

- ingeniería de software,
- estimación de software,
- pruebas del sistema,
- aseguramiento de la calidad de software,
- administración de configuración de software,
- administración de contrato, y
- apoyo de documentación.

2. Los grupos de ingeniería, coordinan sus planes y actividades.
3. Los gerentes son responsables por establecer y mantener un ambiente que facilite la interacción, coordinación, apoyo y trabajo de equipo entre los grupos de ingeniería del proyecto, entre el proyecto y los clientes o usuarios finales, según lo apropiado y a través de toda la organización.

Los usuarios finales referidos en estas prácticas son los usuarios finales designados por el cliente o representantes de los usuarios finales.

Habilidades a realizar

Habilidad 1

Se proveen los recursos y el financiamiento adecuado para coordinar las actividades de ingeniería de software con los otros grupos de ingeniería.

Habilidad 2

Las herramientas de apoyo usadas por los diferentes grupos de ingeniería son compatibles para establecer una comunicación y una coordinación efectiva.

Ejemplos de herramientas de apoyo que deben ser compatibles incluyen:

- sistemas de procesamiento de texto,
- sistema de bases de datos,
- herramientas graficas,
- programa de hojas de calculo,
- paquetes de seguimiento de problemas, y
- herramientas de administración de bibliotecas.

Habilidad 3

Todos los Administradores en la organización reciben la capacitación requerida de trabajo en equipo.

Ejemplos de capacitación incluyen:

- construcción de equipos;
- administración de equipos;
- estableciendo y promoviendo y facilitando el equipo de trabajo; y
- dinámicas de grupo.

Refiérase al área clave de procesos Programa de Capacitación.

Habilidad 4

Todos los líderes de tarea en cada grupo reciben la orientación en procesos, métodos y estándares usados por otros grupos de ingeniería.

Refiérase al área clave de procesos Programa de Capacitación.

Habilidad 5

Los miembros de los grupos de ingeniería reciben orientación en trabajar como un equipo.

Actividades a realizar

Actividad 1

El grupo de ingeniería de software y los otros grupos de ingeniería participan con el cliente y los usuarios finales según lo adecuado, para establecer los requisitos del sistema.

Específicamente estos grupos:

1. Definen las características críticas de los requisitos establecidos por los clientes y los usuarios finales, según lo apropiado.
2. Negocian las dependencias críticas.
3. Documentan los criterios de aceptación para cada producto entregado al cliente o a los usuarios finales.

Actividad 2

Representantes del grupo de ingeniería de software del proyecto trabajan con representantes de otros grupos de ingeniería para seguir y coordinar las actividades técnicas y resolver los problemas técnicos.

1. Los representantes de esos grupos siguen y coordinan las actividades técnicas mediante.
 - coordinación de la especificación y las revisiones y aprobaciones técnicas de los requisitos del sistema y el diseño del sistema;

Los requisitos del sistema y el diseño del sistema son típicamente la responsabilidad del grupo de ingeniería de sistemas, pero se espera que los representantes de otros grupos de ingeniería tengan una participación significativa en esas tareas.

Los requisitos del sistema y el diseño de sistemas incluyen:

- los requisitos globales del sistema.

- la configuración del sistema (es decir hardware, software y otros componentes del sistema),
- la asignación y traza de los requisitos a estos componentes del sistema, y
- la definición de las interfaces entre estos componentes del sistema.

- se proporciona la revisión y el análisis técnico del proyecto necesario para administrar y controlar los cambios a los requisitos del sistema y los objetivos del proyecto a través del ciclo de vida del proyecto.
- seguimiento y revisión de las actividades de diseño y desarrollo para hardware, software y otros componentes del sistema, y
- evaluación y desarrollo de recomendaciones para el seguimiento de los riesgos técnicos que involucran mas de un grupo de ingeniería.

Refiérase a la actividad 10 de la tarea clave de proceso de Administración Integrada de Software para prácticas que cubren la administración de riesgos

2. Los representantes de los grupos manejan los problemas técnicos:

- resolviendo los conflictos del proyecto y aclarando los requisitos del sistema y problemas de diseño.
- desarrollando recomendaciones conjuntas para resolver los problemas.
- atendiendo los problemas de procesos que abarcan a los grupos de ingeniería del proyecto

Actividad 3

Se usa un plan documentado para la comunicación de los compromisos entre grupos y para coordinar y seguir el trabajo realizado.

Este plan es:

1. La línea base para:
 - la calendarización del proyecto,
 - los aspectos técnicos y contractuales del proyecto,
 - la asignación de responsabilidades a los grupos de ingeniería,
2. Usado para coordinar las actividades entre los diferentes grupos de ingeniería.
3. Se hace disponible a los miembros de todos los grupos de ingeniería.
4. Se actualiza para incorporar todos los compromisos entre grupos y los cambios a estos compromisos.
5. Se actualiza conforme el trabajo progresa para reflejar el progreso y los cambios del plan a nivel del proyecto, particularmente cuando se completan los hitos mayores del proyecto y cuando los planes han cambiado significativamente.
6. Revisado y acordado por todos los grupos de ingeniería y el gerente de proyecto.

Actividad 4

Las dependencias críticas entre los grupos de ingeniería son identificadas y negociadas, y seguidas de acuerdo a un procedimiento documentado.

Refiérase a la actividad 9 del área clave de procesos Administración Integrada de Software para prácticas que cubren la administración de dependencias críticas.

Este procedimiento típicamente especifica que:

1. Cada dependencia critica es definida explícitamente, incluyendo:
 - ❑ los elementos a ser proporcionados,
 - ❑ quien los proporcionara,
 - ❑ cuando serán proporcionados,
 - ❑ los criterios de aceptación.
2. Las dependencias críticas son negociadas entre el grupo de ingeniera de software y otros grupos de ingeniería en el proyecto y la organización
3. Las fechas necesarias y fechas de disponibilidad de los elementos de las dependencias críticas son ligadas a la calendarización del proyecto y la programación del software.
4. Los acuerdos para cada dependencia critica son documentados revisados y aprobados tanto por el grupo receptor y el grupo responsable por proveer el elemento de la dependencia crítica.
5. Las dependencias críticas son seguidas regularmente y se toman acciones correctivas cuando sea apropiado.
 - ❑ Se compara el estado y el cumplimiento actual o proyectado contra el plan usado para coordinar los compromisos entre los grupos.
 - ❑ Los efectos de terminación temprana o atrasada son evaluados en cuanto a los impactos en las actividades e hitos futuros.
 - ❑ Los problemas actuales y potenciales son reportados a los gerentes apropiados.

Actividad 5

Los productos de trabajo producidos como entradas para otros grupos de ingeniería son revisados, por los representantes de los grupos receptores para asegurar que los productos de trabajo satisfacen sus necesidades.

Actividad 6

Los problemas entre grupos no resueltos por los representantes del grupo de ingeniería del proyecto son manejados de acuerdo a un procedimiento documentado.

Ejemplos de problemas entre grupos incluyen:

- calendarización incompatible,
- financiamiento inadecuado,
- riesgos técnicos,
- defectos en los requisitos y diseño del sistema, y
- problema del sistema.

Actividad 7

Los representantes del grupo de ingeniería del proyecto dirigen revisiones e intercambios técnicos periódicos.

En estas reuniones los participantes:

1. Proveen la visibilidad de las necesidades y deseos de los clientes y de los usuarios finales según lo adecuado.
2. Siguen las actividades técnicas del proyecto.
3. Aseguran que las interpretaciones e implementaciones que hacen los grupos con los requisitos técnicos estén de acuerdo con los requisitos del sistema.
4. Revisan los compromisos para determinar si estos han sido satisfechos.

Refiérase al área clave de procesos Seguimiento y control del proyecto de software para prácticas que cubren las revisiones.

5. Revisan los riesgos técnicos y otros problemas técnicos

Refiérase a la actividad 10 del área clave de procesos Administración Integrada de Software para prácticas que cubren la administración de los riesgos.

Medición y análisis

Medición 1

Se hacen y se usan para determinar el estado de las actividades de coordinación entre grupos.

Ejemplos de mediciones incluyen

- el esfuerzo real y otros recursos gastados por el grupo de ingeniería de software para apoyar a otros grupos de ingeniería;
- el esfuerzo real y otros recursos gastados por otros grupos de ingeniería en apoyar al grupo de ingeniería de software;
- terminación real de tareas e hitos específicos del grupo de ingeniería de software para apoyar las actividades de otros grupos de ingeniería; y
- terminación real de tareas e hitos específicos de otros grupos de ingeniería para apoyar las actividades del grupo de ingeniería de software.

Verificación e implementación

Verificación 1

Se revisan periódicamente las actividades para la coordinación entre grupos con la Administración superior.

Refiérase a la verificación 1 del área clave de proceso de Seguimiento y control de proyectos para prácticas que cubren el típico contenido de las revisiones de seguimiento de la Administración superior

Verificación 2

Las actividades para la coordinación entre grupos son revisadas con el Administrador del proyecto periódicamente y como respuesta a eventos.

Refiérase a la verificación 2 del área clave de proceso Seguimiento y control del proyecto de software para prácticas que cubren el contenido típico de revisiones de seguimiento de la Administración del proyecto.

Verificación 3

El grupo de aseguramiento de la calidad de software revisa y / o audita las actividades y los productos de trabajo para la coordinación entre grupos y reporta los resultados

Refiérase al área clave de procesos de Aseguramiento de calidad de software

Las responsabilidades del Aseguramiento de la calidad de software para esta área clave de procesos pueden ser incluidas dentro de la función de aseguramiento de calidad que cubre todos los grupos de ingeniería del proyecto.

Como mínimo las revisiones y / o auditorías verifican;

1. El procedimiento para identificar, negociar y seguir las dependencias críticas entre los grupos de ingeniería del proyecto.
2. El manejo de problemas entre grupos.

8.2 Modelo gráfico Coordinación Entre Grupos (CEG)

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en la coordinación entre grupos (CEG) y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

8.3 Identificación de los grupos participantes

En el modelo textual del (CEG), hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave.

A continuación presentamos una breve definición de cada uno de ellos.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de varios proyectos a la vez.

Grupo de Aseguramiento de la Calidad: Revisa y / o audita las actividades para la coordinación entre grupos y reporta los resultados.

Grupo de Ingeniería de Software (Grupo IS): Es la colección de individuos (tanto administradores, como personal técnico), que tienen la responsabilidad para desempeñar las actividades de mantenimiento y desarrollo de software (como análisis de requisitos, diseño, codificación y pruebas) para el proyecto. Los grupos relacionados al software, tal como el grupo de aseguramiento de calidad, el grupo de administración de configuraciones, el grupo de ingeniería de procesos de software, no están incluidos en el grupo de ingeniería de software.

Grupos de Ingeniería: Una colección de individuos (tanto gerentes, como personal técnico) representando una disciplina de la ingeniería. Ejemplos de disciplinas de ingeniería incluyen ingeniería de sistemas, ingeniería de hardware, pruebas del sistema, ingeniería de software, administración de configuración de software y aseguramiento de la calidad de software.

Cliente: El individuo u organización que es responsable por aceptar el producto y autorizar el pago a la organización desarrolladora.

Usuario Final: El individuo o el grupo que usara el sistema para su uso operacional esperado cuando éste es entregado en su ambiente.

Administradores: Un rol que abarca proporcionar dirección técnica y administrativa y controlar a los individuos que realizan tareas o actividades dentro del área de responsabilidad del administrador. Las funciones tradicionales de un administrador incluyen, planificación, recursos, organización, dirección y control de trabajo.

Administrador del Proyecto: Es el responsable de las actividades y resultado del proyecto de software. Este administrador participa en las revisiones de las actividades de la CEG, que se realizan en forma periódica o a causa de un evento.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

8.4 Diagrama de actividades

Compromisos

En esta área clave de procesos contamos con un compromiso, este establece que la organización sigue una política organizacional escrita para establecer equipos de ingeniería interdisciplinarios.

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que sirve para establecer grupos de ingeniería interdisciplinarios. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 8.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para establecer grupos de ingeniería interdisciplinarios, queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Establecer grupos de Ingeniería interdisciplinarios C1”.

Una vez establecidos los grupos se puede empezar la coordinación entre estos, representada en la Figura 8.1 por la actividad con el nombre “Coordinar grupos”. Esta actividad representa al resto de características comunes de esta área clave de procesos.

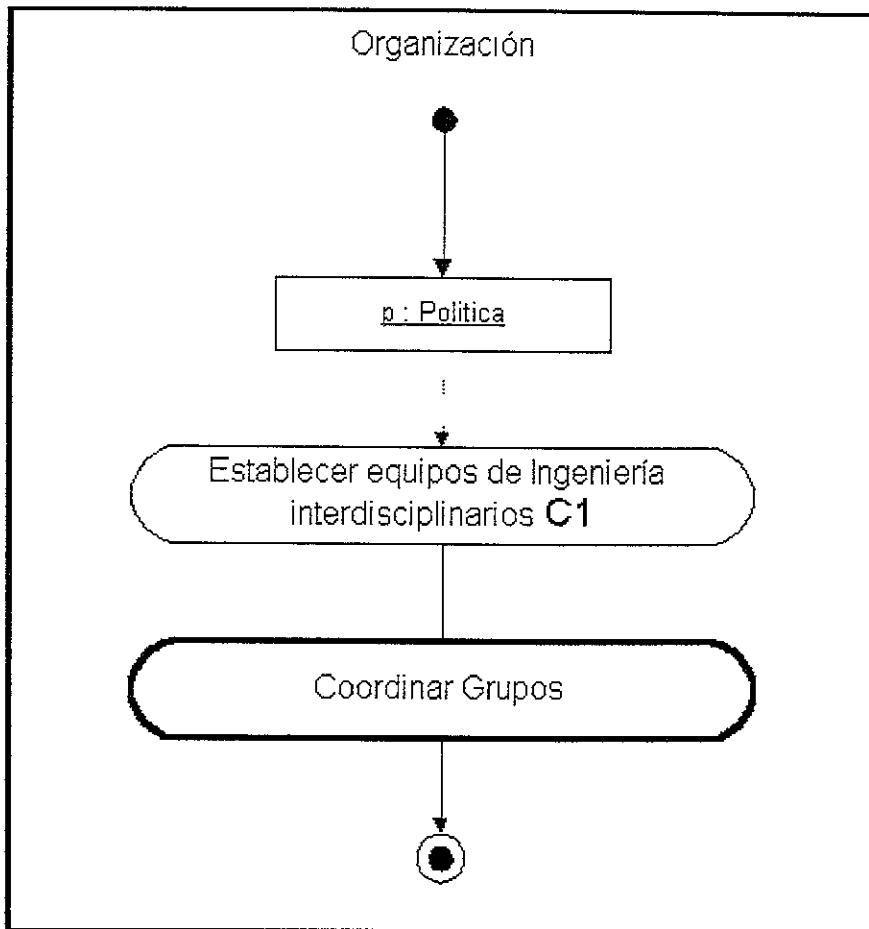


Figura 8.1: Compromisos del área clave de proceso Coordinación Entre Grupos (CEG).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para realizar la Coordinación entre grupos.

Para modelar este hecho, la actividad “Coordinar Grupos” se puede desglosar en otro diagrama presentado en la Figura 8.2.

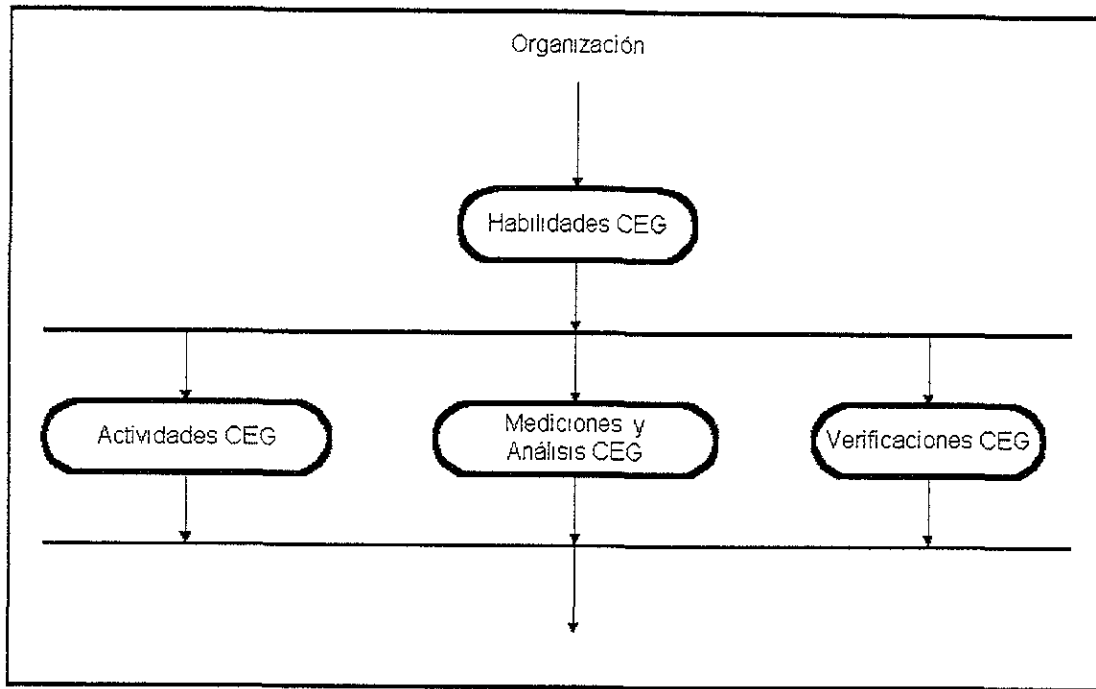


Figura 8.2: Coordinar Grupos.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de la Coordinación entre grupos. Por otro lado, las actividades de verificación y medición con su análisis, son simultaneas a las de Coordinación entre grupos.

Habilidades.

La habilidad 1 habla de la provisión de recursos y financiamiento, que está a cargo de la organización. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado H1” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación u orientación.

La habilidad 2 habla de que las herramientas de apoyo que se utilizan entre los diferentes grupos de ingeniería deben ser compatibles para establecer una comunicación y una coordinación efectiva. Esta habilidad la representamos con la actividad “Usar herramientas de apoyo compatibles H2”.

La habilidad 3 especifica que los Administradores de la organización reciben capacitación sobre trabajo en equipo. Esta habilidad la representaremos con la actividad “Recibir capacitación H3”.

La habilidad 4 especifica que todos los lideres reciben orientación en procesos, métodos y estándares usados por otros grupos de ingeniería. Esta habilidad la representaremos con la actividad “Recibir orientación H4”.

La habilidad 5 especifica que los miembros de los grupos de ingeniería reciben orientación en trabajar como un equipo. Esta habilidad la representaremos con la actividad “Recibir orientación en trabajar como un equipo H5”.

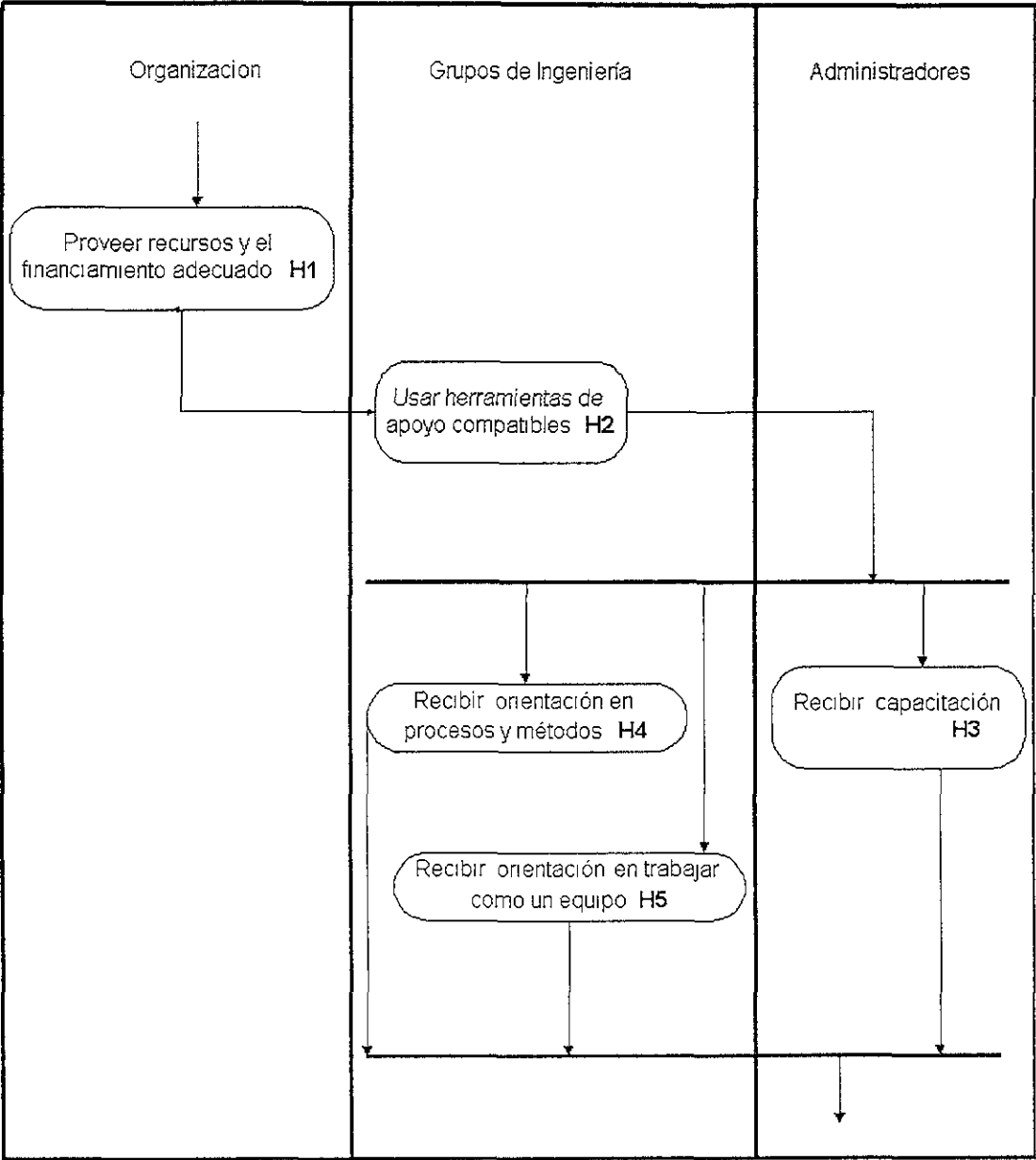


Figura 8.3 Habilidades del área clave de proceso Coordinación Entre Grupos (CEG).

El diagrama de la figura 8.3 presenta las actividades a realizarse para adquirir las habilidades para la CEG.

Actividades.

La actividad 1 específica que el Grupo de ingeniería de software y otros grupos de ingeniería participan con el cliente y los usuarios finales para establecer los requisitos del sistema. . Esta actividad la representamos en el diagrama como “Establecer los requisitos del sistema A1”. Al término de esta actividad tendremos como salida a **rs: Requisitos del Sistema**.

La actividad 2 específica que los representantes del Grupo de ingeniería de software del proyecto, trabajan con los representantes de otros grupos de Ingeniería para seguir y coordinar las actividades técnicas y resolver los problemas. Esta actividad la representamos como “Coordinar y seguir las actividades técnicas A2”. Dentro de esta actividad están contenidas las actividades 3,4,5,6 que se desglosaran en otro diagrama.

La actividad 7 específica que los representantes del Grupo de ingeniería del proyecto, dirigen revisiones e intercambios técnicos periódicos. Esta actividad la representamos como “Dirigir revisiones periódicas A7”. Las actividades mencionadas anteriormente quedan representadas en la figura 8.4

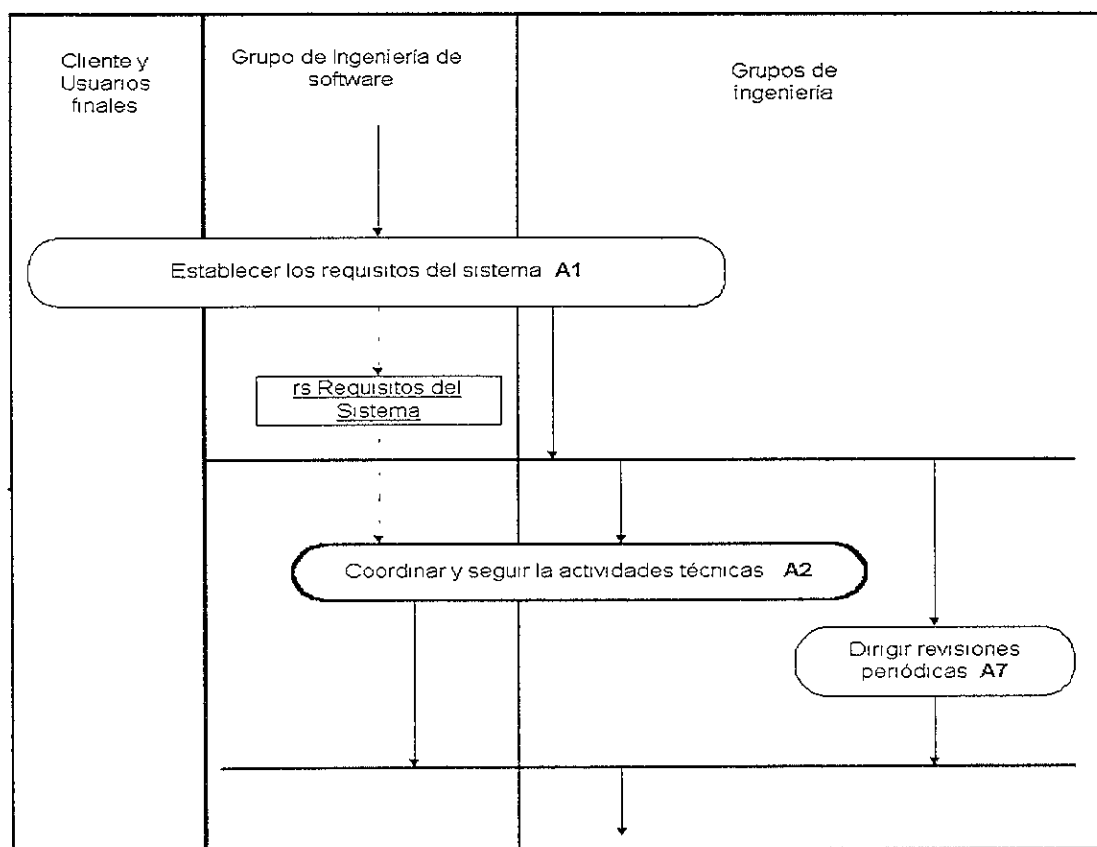


Figura 8.4 Actividades del área clave de proceso: Coordinación Entre Grupos (CEG).

La actividad 3 especifica que se usa un plan documentado para coordinar y seguir el trabajo realizado. Esta actividad la representamos como “Coordinar y seguir el trabajo realizado A3”. Esta actividad tiene como entrada al objeto **plad: Plan documentado**. Como en el texto no dice quien realiza la coordinación y el seguimiento del trabajo nosotros proponemos que quien lo realiza son el Grupo de ingeniería de software, y otros Grupos de Ingeniería.

La actividad 4 especifica que las dependencias críticas entre los grupos de ingeniería son identificadas y negociadas y seguidas de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Identificar y negociar las dependencias críticas A4”. Esta actividad tiene como entrada al objeto **pro: Procedimiento documentado**. Como en el texto no dice quien identifica y negocia las dependencias criticas, nosotros proponemos que quien lo realiza son el Grupo de Ingeniería de software, y otros Grupos de Ingeniería.

La actividad 5 especifica que se revisan los productos de trabajo producidos para asegurar que los productos de trabajo satisfacen sus necesidades. Esta actividad la representamos como “Revisar los productos de trabajo A5”. Esta actividad la realizan los representantes de los Grupos de ingeniería.

La actividad 6 especifica que los problemas entre grupos no resueltos son manejados de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Manejar los problemas no resueltos A6”. Esta actividad tiene como objeto de entrada a **pro: Procedimiento documentado**, que es utilizado para resolver los problemas. Esta actividad la realizan los representantes de los grupos de ingeniería.

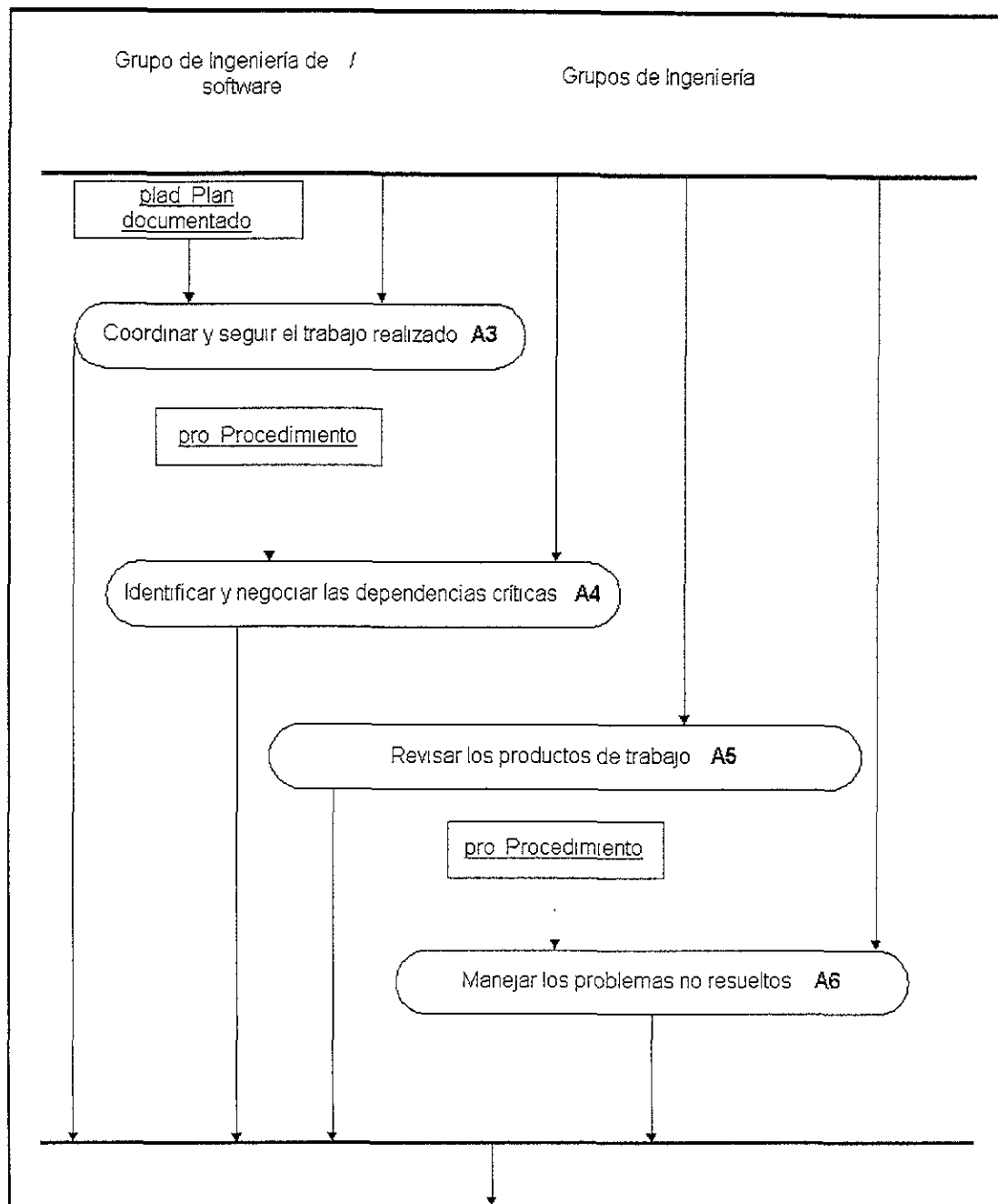


Figura 8.5: Coordinar y seguir las actividades técnicas.

El diagrama de la figura 8.5 presenta las actividades que se describieron anteriormente. También en este diagrama es fácil observar que actividades pueden llevarse a cabo en forma paralela y el orden en que se realizan.

Mediciones y análisis.

El texto correspondiente a las mediciones y análisis es muy escueto. Se dice que deben de recabarse y usar mediciones para determinar el estado de las actividades de la CEG. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad que realizan los Grupos de Ingeniería.

La actividad mencionada anteriormente queda representada en la figura 8.6

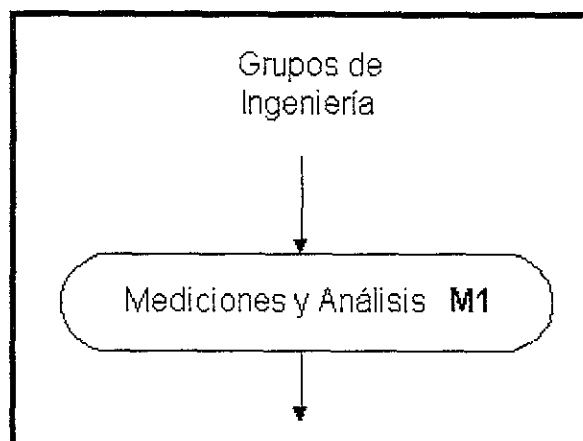


Figura 8.6 Mediciones del área clave de proceso Coordinación Entre Grupos (CEG).

Verificaciones

La verificación 1 especifica que las actividades para la CEG serán revisadas periódicamente con la Administración gerencial. Esta verificación está representada con la actividad de “Revisar Actividades para la CEG V1”.

La verificación 2 especifica que las actividades para la CEG serán revisadas periódicamente con el Administrador del proyecto. Esta verificación esta representada con la actividad de “Revisar Actividades para la CEG V2”.

Finalmente en la verificación 3 especifica que las actividades y productos de trabajo se revisan y / o auditan y los resultados se reportan, quien la realiza es el grupo Aseguramiento de calidad. Esta verificación esta representada con la actividad de “Revisar y / o auditar las actividades y productos de trabajo V3”.

Las verificaciones mencionadas anteriormente quedan representadas en la figura 8.7

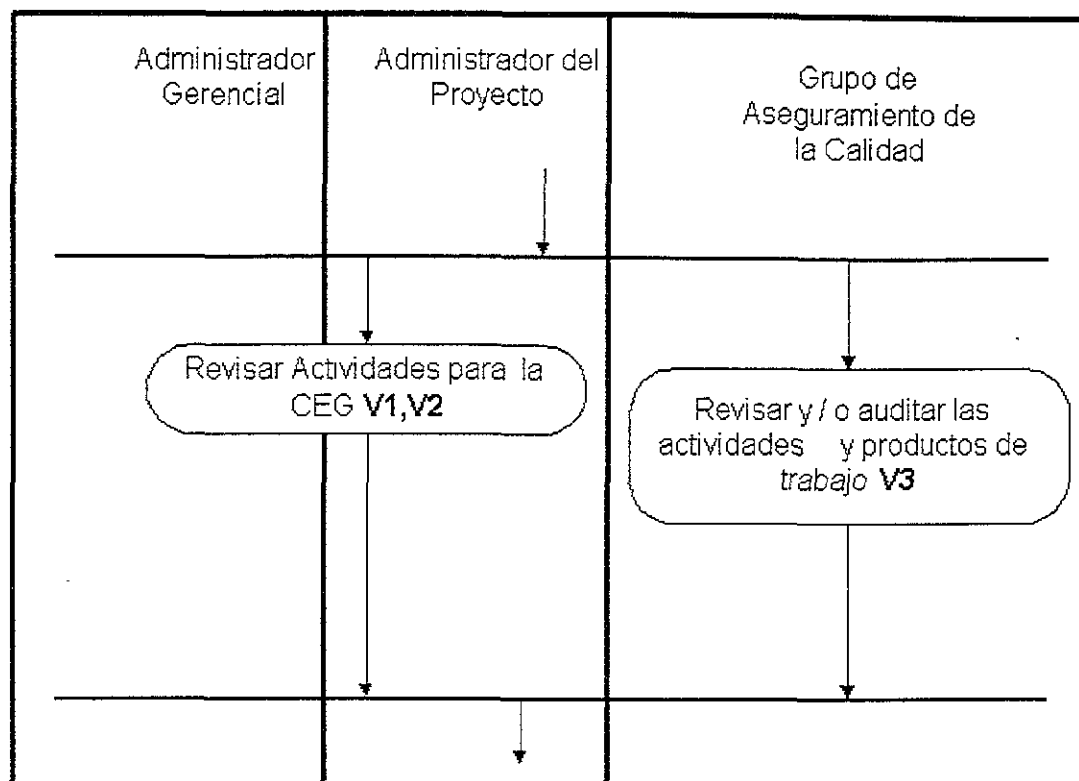


Figura 8.7 Verificaciones del área clave de proceso Coordinación Entre Grupos (CEG).

Capítulo 9 Revisiones Entre Colegas

En las Revisiones Entre Colegas se llevará a cabo una revisión y evaluación de los productos de software para identificar los defectos lo mas pronto posible y eliminarlos, así como identificar las áreas en donde los cambios son necesarios.

El objetivo de esté capítulo es presentar la traducción de está area clave de procesos [CMM Practices, versión 1.1 (SEI/CMU-93-TR25).] y la generación del modelo gráfico que abstrae la información contenida en el texto.

9.1 Revisiones Entre Colegas

Una área clave de proceso para el nivel 3 Definido

El propósito de las Revisiones entre colegas es eliminar los defectos de los productos de software tempranamente y eficazmente. Un efecto colateral importante es desarrollar un mejor entendimiento de los productos de trabajo de software y de los defectos que pueden ser prevenidos.

Las Revisiones entre colegas involucran un examen metódico de los productos de trabajo de software por los colegas de los productores para identificar defectos y áreas donde los cambios son necesarios. Los productos específicos que se someterán a las Revisiones entre colegas son identificados en el proceso de software definido del proyecto y planificados como parte de las actividades de planificación del proyecto de software, como se describe en la Administración Integrada de Software.

Esta área clave de procesos cubre las prácticas para realizar las Revisiones Entre Colegas. Las prácticas de identificación de los productos de trabajo de software específicos que han de someterse a las Revisiones Entre Colegas, están contenidas en las áreas claves de procesos que describen el desarrollo y mantenimiento de cada producto de trabajo de software.

Metas

- | | |
|---------------|--|
| Meta 1 | Las actividades de las Revisiones entre colegas son planificadas. |
| Meta 2 | Los defectos en los productos de trabajo de software son identificados y removidos. |

Compromisos a realizar

- | | |
|---------------------|--|
| Compromiso 1 | Los proyectos siguen una política de organización escrita para realizar las Revisiones entre colegas. |
|---------------------|--|

Esta política típicamente especifica que:

1. La organización identifica un conjunto estándar de productos de trabajo de software que serán sometidos a las Revisiones entre colegas.

2. Cada proyecto identifica los productos de trabajo de software que serán sometidos a las Revisiones entre colegas.

Refiérase a la actividad 1 del área clave de procesos de Administración Integrada de Software y la actividad 2 del área clave de procesos de Definición del Proceso de la Organización para prácticas que cubren la identificación de productos de trabajo que serán sometidos a las Revisiones Entre Colegas.

Ejemplos de productos de trabajo de software incluyen:

- software operacional y software de apoyo,
- productos de trabajo de software entregables y no entregables,
- software (por ejemplo el código fuente) y productos de trabajo que no son software (por ejemplo documentos), y
- descripción de procesos.

3. Las Revisiones entre colegas son dirigidas por los líderes capacitados en las Revisiones entre colegas.
4. Las Revisiones entre colegas se enfocan en los productos de trabajo de software que están siendo revisados y no en el productor.
5. Los resultados de las Revisiones entre colegas no son usados por la administración para evaluar el desempeño de los individuos.

Habilidades a realizar

Habilidad 1

Se proveen los recursos y el financiamiento adecuado para realizar las Revisiones entre colegas en cada producto de trabajo de software a ser revisado.

Se proveen recursos y financiamiento para:

1. Preparar y distribuir los materiales de las Revisiones entre colegas.
2. Dirigir las Revisiones entre colegas.
3. Revisar los materiales.
4. Participar en las Revisiones entre colegas y en cualquier revisión de seguimiento requerida, basada en los defectos identificados en la Revisiones entre colegas.
5. Monitorear la reconstrucción de los productos de software basado en los defectos identificados en las Revisiones entre colegas.
6. Recolectar y reportar los datos resultantes de las Revisiones entre colegas.

Habilidad 2

Los Líderes de las Revisiones Entre Colegas reciben la capacitación necesaria sobre como dirigir las Revisiones entre colegas.

Ejemplos de capacitación incluyen:

- los objetivos, principios y métodos de las Revisiones entre colegas;
- planificación y organización de las Revisiones entre colegas;
- evaluación de los criterios de inicio y finalización para una Revisión entre colegas;
- dirigir y facilitar una Revisión entre colegas;
- reportar los resultados de una Revisión entre colegas;
- seguimiento y confirmación de las correcciones, para que corresponda dirigir las acciones identificadas en una Revisión entre colegas; y
- recolectar y reportar los datos requeridos para una Revisión entre colegas.

Refiérase al área clave de procesos Programa de Capacitación

Habilidad 3

Los revisores participantes en las Revisiones entre colegas reciben la capacitación requerida en los objetivos, principios y métodos de las Revisiones entre colegas.

Ejemplos de capacitación incluyen:

- tipos de Revisiones entre colegas (por ejemplo revisiones de requisitos de software, diseño de software, código y procedimientos de prueba de software);
- los objetivos, principios y métodos de las Revisiones entre colegas;
- roles de los revisores; y
- estimación del esfuerzo por preparar y participar en las Revisiones entre colegas.

Refiérase al área clave de procesos Programa de Capacitación.

Actividades realizadas

Actividad 1

Se planean las Revisiones entre colegas y los planes son documentados.

Estos planes:

1. Identifican los productos de trabajo de software que serán sometidos a las Revisiones entre colegas.
 - Los productos de trabajo de software seleccionados incluyen el conjunto identificado en el Proceso de software estándar de la organización.

Refiérase a la actividad 2 del área clave de procesos Enfoque en el Proceso de la Organización para prácticas que cubren el proceso de software estándar de la organización.

2. Especifican la calendarización de las Revisiones entre colegas.

Para Revisiones entre colegas que están programadas en el futuro cercano se identifican los líderes de las Revisiones entre colegas y los otros revisores.

Actividad 2

Se realizan las Revisiones entre colegas de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las Revisiones entre colegas son planeadas y guiadas por líderes capacitados en las Revisiones entre colegas.
2. Se distribuyen los materiales de revisión a los revisores por adelantado, para que así puedan prepararse adecuadamente para las Revisiones entre colegas.

Los materiales de revisión deben incluir las entradas relevantes para el desarrollo de las Revisiones entre colegas de productos de trabajo de software.

Ejemplos de entradas relevantes incluyen:

- los objetivos de los productos de trabajo de software,
- los estándares aplicables,
- los requisitos relevantes para un módulo de diseño o,
- el diseño detallado para un modulo de código.

3. Los revisores tienen roles asignados en las Revisiones entre colegas.

4. Se especifican y se aplican los criterios de inicio y terminación de las Revisiones entre colegas.
 - Los problemas para cumplir estos criterios son reportados al gerente correspondiente.
5. Se usan listas de verificación para identificar criterios usados en la revisión de los productos de trabajo de software de una forma consistente.
 - Las listas de verificación son adecuadas al tipo específico de producto de trabajo y las Revisiones entre colegas.

Ejemplos de elementos abordados al adaptar las listas de verificación incluyen:

- compatibilidad con estándares y procedimientos,
- integridad,
- exactitud,
- reglas de construcción, y
- mantenimiento.

- Las listas de verificación son revisadas por los desarrollares de las mismas y usuarios potenciales.
6. Se da seguimiento a las acciones identificadas en las Revisiones entre colegas hasta su resolución.
 7. La terminación exitosa de las Revisiones entre colegas, incluyendo el replanteamiento para cubrir los elementos identificados en las Revisiones entre colegas, es usado como un criterio de terminación para la tarea asociada.

Actividad 3

Se registran los datos sobre la conducción y los resultados de las Revisiones entre colegas.

Ejemplos de datos incluyen:

- identificación del producto de trabajo de software revisado,
- tamaño del producto de trabajo de software,
- tamaño y composición del equipo de revisión,
- tiempo de preparación por revisor,

- duración de la reunión de revisión,
- Tipos y número de defectos encontrados y arreglados, y
- el esfuerzo de las correcciones.

Medición y análisis

Medición 1

Se hacen y se usan las mediciones para determinar el estado de las actividades de las Revisiones entre colegas.

Ejemplos de mediciones incluyen:

- número de Revisiones entre colegas realizadas, comparado con los planes,
- el esfuerzo global gastado en revisiones comparado con los planes, y
- número de productos de trabajo de software revisados comparado con los planes.

Verificación e implementación

Verificación 1

El grupo de aseguramiento de la calidad de software revisa y /o audita las actividades y los productos de trabajo para las Revisiones entre colegas y reporta los resultados.

Refiérase al área clave de proceso de Aseguramiento de la calidad de software.

Como mínimo las revisiones y / o auditorias verifican que:

1. Las Revisiones entre colegas planificadas, sean efectuadas.
2. Los lideres de las Revisiones entre colegas están capacitados adecuadamente para sus roles.
3. Los revisores están propiamente capacitados o tienen experiencia en sus roles.
4. Se sigue el proceso para prepara las Revisiones entre colegas, conducir las Revisiones entre colegas y ejecutar las acciones de seguimiento.

9.2 Modelo gráfico Revisiones Entre Colegas (REC)

La construcción del modelo gráfico se inicia con una identificación, en todo el texto original, de los grupos participantes en las Revisiones entre colegas (REC) y, posteriormente, con una asignación de sus responsabilidades en los diagramas de actividades.

9.3 Identificación de los grupos participantes

En el modelo textual del (REC), hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentamos una breve definición de cada uno de ellos.

Grupo de Aseguramiento de la Calidad: Revisan y /o auditan las actividades para las Revisiones Entre Colegas y reporta los resultados.

Líder de las Revisiones Entre Colegas (LREC): Un individuo específicamente especializado y calificado para planificar, organizar y ejecutar una Revisión Entre Colegas.

Los revisores: Son los individuos que participan en los diferentes tipos de Revisiones entre colegas (por ejemplo revisiones de requisitos de software, diseño de software, código y procedimientos de prueba de software).

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

9.4 Diagrama de actividades

Compromisos

En esta área clave de procesos contamos con un compromiso, este establece que la organización sigue una política organizacional escrita para realizar las Revisiones entre colegas. Analizando el texto del compromiso 1, se deduce que la organización tiene una política. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 9.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para realizar las revisiones entre colegas, queda expresado colocando a *p: Política* como objeto de entrada del compromiso C1. Este compromiso lo representamos como “Realizar la Revisión entre colegas C1”.

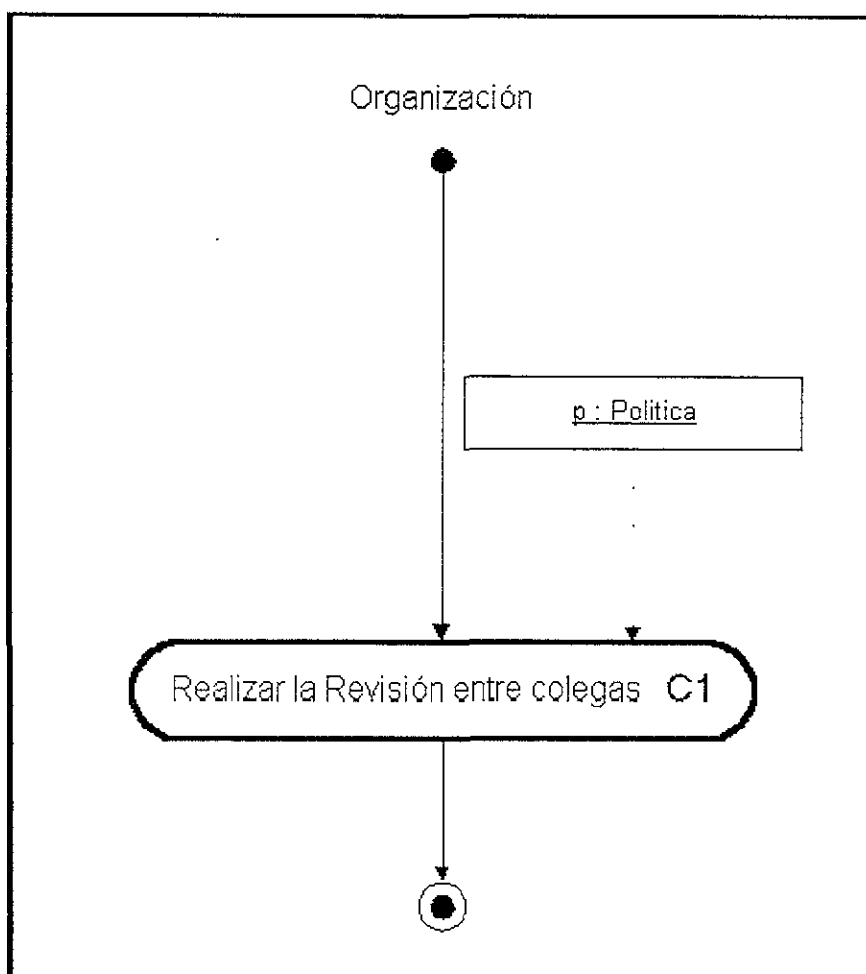


Figura 9.1: Compromisos del área clave de proceso: Revisión Entre Colegas (REC).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para realizar las Revisiones entre colegas.

Para modelar este hecho, la actividad se puede desglosar en otro diagrama presentado en la Figura 9.2.

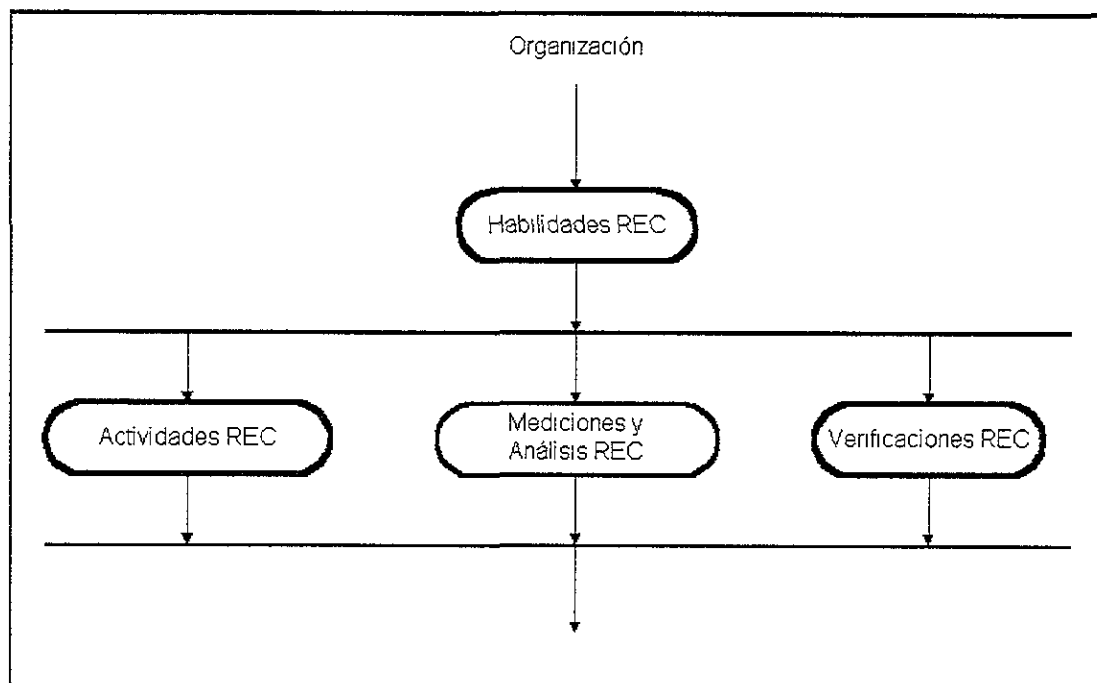


Figura 9.2: Realizar la Revisión Entre Colegas.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de las Revisiones entre colegas. Por otro lado, las actividades de verificación y medición con su análisis, son simultaneas a las de las Revisiones entre colegas..

Habilidades

La habilidad 1 habla de la provisión de recursos y financiamiento, que está a cargo de la organización. Esta habilidad la representamos con la actividad “Proveer recursos y el financiamiento adecuado H1” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación.

La habilidad 2 habla de que los lideres de las Revisiones entre colegas reciben la capacitación necesaria sobre como dirigir las Revisiones entre colegas. Esta habilidad la representamos con la actividad “Recibir capacitación sobre como dirigir las REC H2”.

La habilidad 3 habla de que los revisores participantes en las Revisiones entre colegas reciben la capacitación requerida en los objetivos, principios y métodos de las Revisiones entre colegas. Esta habilidad la representaremos con la actividad “Recibir capacitación en los objetivos y métodos de las REC H3 ”.

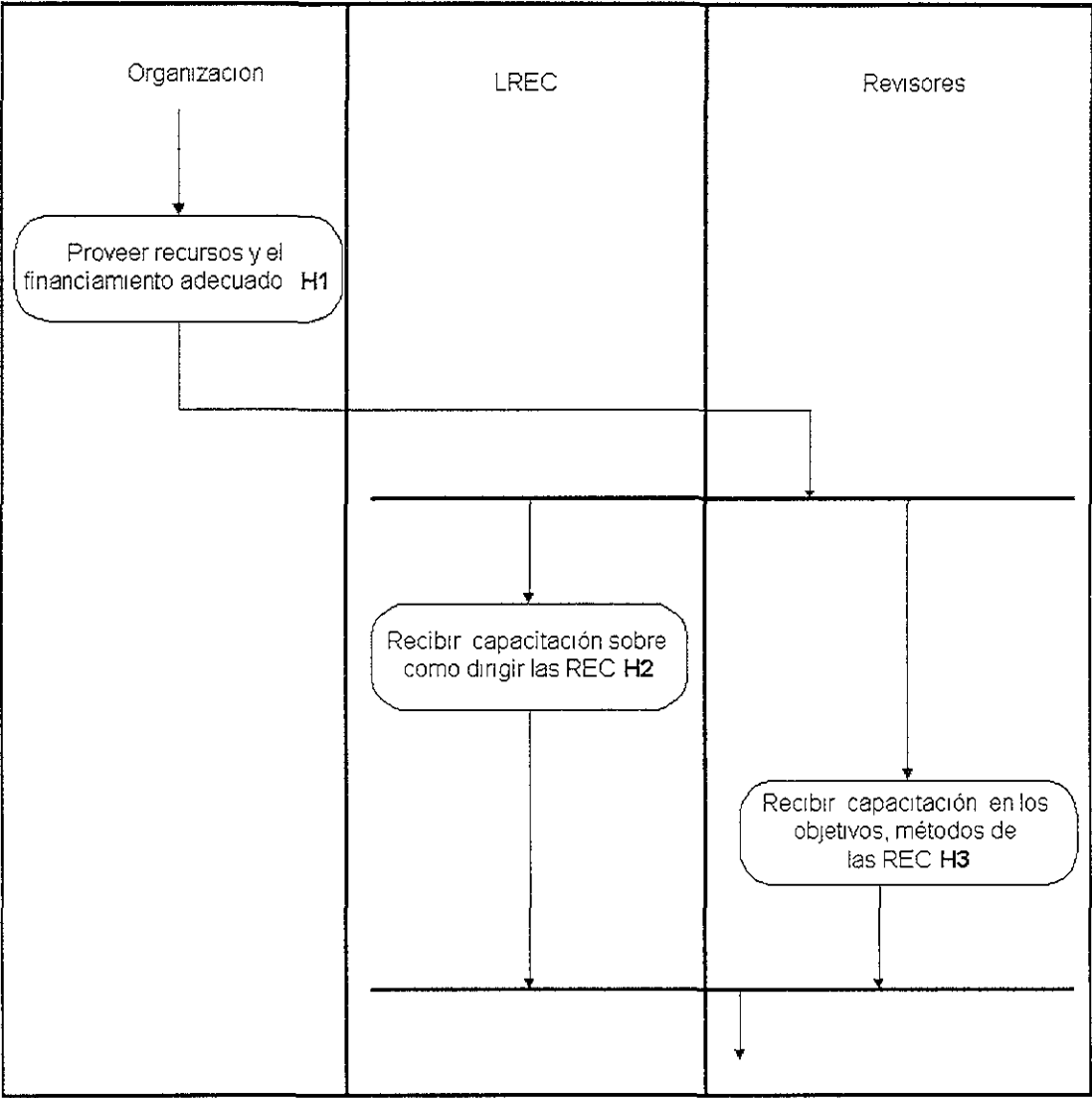


Figura 9.3: Habilidades del área clave de proceso: Revisión Entre Colegas (REC).

El diagrama de la Figura 9.3 presenta las actividades a realizarse para adquirir las habilidades para la REC, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde.

Actividades

La actividad 1 específica que Se planean las Revisiones entre colegas. Esta actividad la representamos en el diagrama como “Planear las REC A1”. Al término de esta actividad tendremos como salida a **plad: Planes documentados**. Como en el texto no dice quien realiza la planeación y documenta los planes, nosotros proponemos que quien lo realiza es el Líder de las Revisiones entre colegas, los revisores y la organización como un acuerdo.

La actividad 2 específica que se realizan las Revisiones entre colegas de acuerdo a un procedimiento documentado. Esta actividad la representamos en el diagrama como “Realizar las REC A2”. Esta actividad tiene como entrada al objeto **pro:Procedimiento documentado**. Como en el texto no dice quien realiza las Revisiones entre colegas, nosotros proponemos que quien lo realiza es el Líder de las Revisiones entre colegas y los revisores.

La actividad 3 específica que Se registran los datos sobre la conducción y los resultados de las Revisiones entre Colegas. Esta actividad la representamos en el diagrama como “Registrar los resultados de las REC A3”. Como en el texto no dice registra los datos sobre la conducción y resultados de las Revisiones entre colegas, nosotros proponemos que quien lo realiza es el Líder de las Revisiones entre colegas.

Las actividades mencionadas anteriormente quedan representadas en la figura 9.4

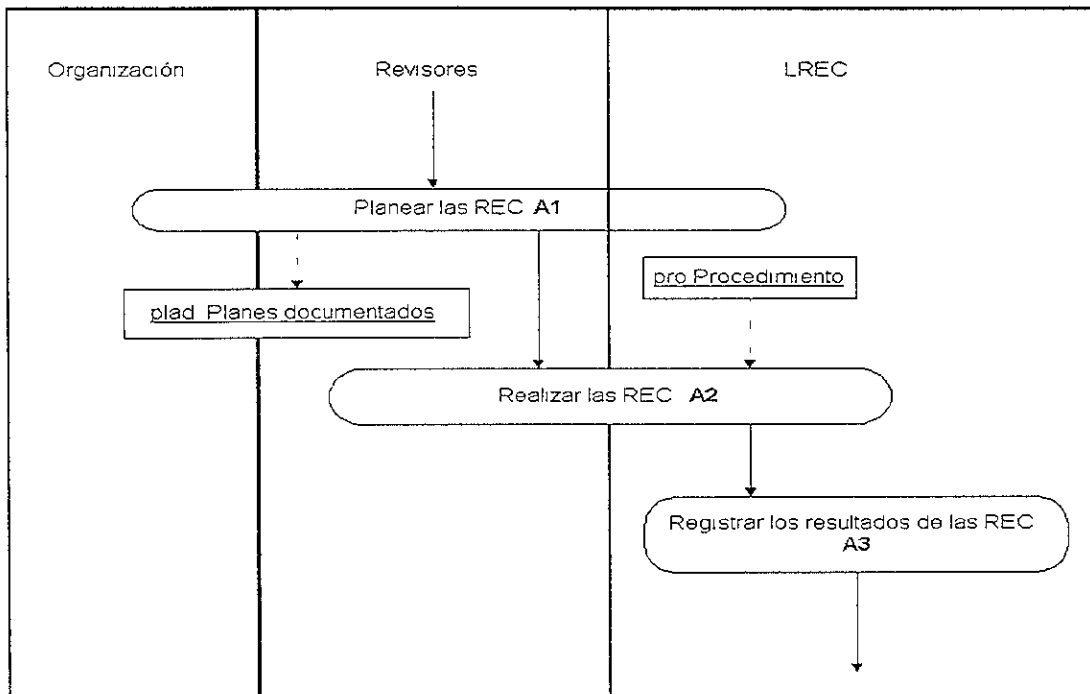


Figura 9.4: Actividades del área clave de proceso: Revisión Entre Colegas (REC).

Mediciones y análisis

El texto correspondiente dice que se deben de recabar y usar las mediciones para determinar el estado de las actividades de las REC. Nos atreveremos a proponer que la actividad mediciones y análisis es una actividad que la realiza el Líder de las REC. La actividad mencionada anteriormente queda representada en la figura 9.5

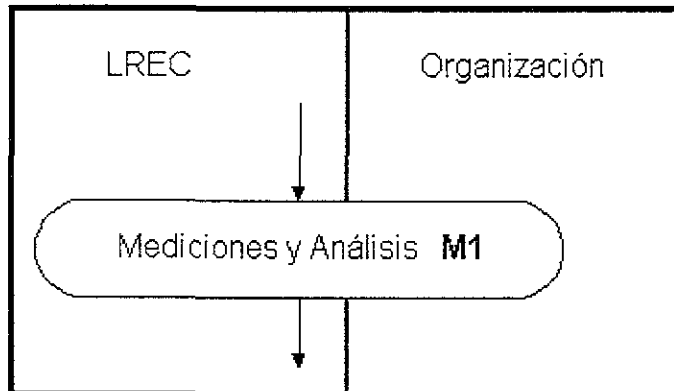


Figura 9.5 Medición y Análisis del área clave de proceso: Revisión Entre Colegas (REC).

Verificaciones

La verificación 1 especifica que las actividades y productos de trabajo de las Revisiones entre colegas, se revisan y / o auditan y los resultados se reportan, Estas verificación la realiza el Grupo de aseguramiento de calidad. Está verificación esta representada con la actividad de “Revisar y / o auditar las actividades y productos de trabajo V1”. La actividad mencionada anteriormente queda representada en la figura 9.6

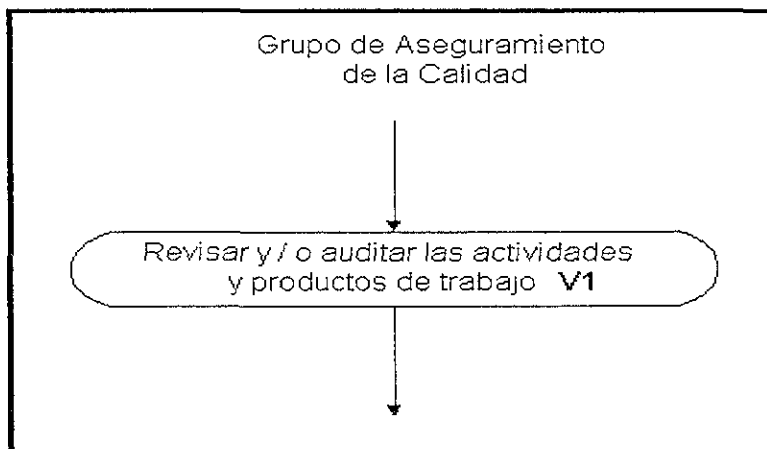


Figura 9.6 Verificaciones del área clave de proceso: Revisión Entre Colegas (REC).

Conclusiones

Para la elaboración de la tesis, lo primero que tuve que realizar fue comprender el CMM y en especial el nivel 3 de CMM. El modelo CMM ayuda a las organizaciones que desarrollan software a mejorar sus procesos, ya que un proceso inadecuado puede tener graves consecuencias, lo cual puede significar la diferencia entre el éxito y el fracaso.

Actualmente no se cuenta con traducciones al español de CMM, así que el primer paso para la elaboración de esta tesis, fue traducir todas las áreas claves del nivel 3 de CMM y el área clave de procesos Administración de Subcontratos de Software del nivel 2.

El área clave de procesos Administración de subcontratos de software, fue elaborada con el fin de completar el nivel 2 de CMM.

Una vez concluida la traducción de las áreas claves de procesos, se procedió con la elaboración del modelo gráfico del nivel 3 de CMM.

El modelo gráfico fue generado con la finalidad de que el usuario o las empresas que se dedican a desarrollar software, visualizaran y comprendieran mejor las prácticas claves de CMM, ya que el texto resulta muy difícil de asimilar.

Para el modelado se utilizó el Lenguaje Unificado de Modelado UML (Unified Modeling Language), por las siguientes razones:

- Actualmente se está popularizando como un lenguaje común de las personas que se dedican a desarrollar software.
- Es un lenguaje estándar que puede utilizarse para visualizar, especificar y documentar.
- La semántica de UML está bien definida, gracias a eso un usuario puede escribir un modelo en UML y otro usuario puede interpretar ese modelo sin ambigüedad.

El primer modelo gráfico que se realizó fue de la área clave Administración de subcontratos de software del nivel 2 de CMM. Este primer modelo nos sirvió como un ejercicio previo para establecer la metodología de trabajo. Una vez comprendido se aplicó el mismo proceso a las áreas claves del nivel 3 de CMM.

Para modelar las áreas claves se utilizó una extensión de los diagramas de actividades de UML, ya que permite hacer más claros y sencillos los diagramas que se elaboraron.

La extensión consiste de dos elementos que son:

1. Las responsabilidades compartidas.- Se utilizaron para mostrar que una actividad puede ser realizada por más de un responsable en la organización. Es decir representaríamos responsabilidades compartidas sobre una actividad.
2. El desglose de actividades.- Lo utilizamos en el trabajo para representar gráficamente las actividades anidadas, donde una actividad más compleja se desglosa en otro diagrama.

El modelo gráfico incluye la presentación de los grupos participantes, la descripción de sus roles, los diagramas de actividades de cada área clave, así como su explicación, de esta forma logramos abstraer y sintetizar el texto de CMM nivel 3.

Este trabajo cumple con el alcance de la tesis, ya que se generaron los diagramas de actividades de todas las áreas claves del nivel 3 de CMM, con estos diagramas de actividades el usuario y las organizaciones podrán:

- Asimilar más fácilmente las áreas claves de procesos del nivel 3 de CMM.
- Comprender el objetivo de cada área clave.
- Quiénes son los grupos que desarrollan estas actividades.
- El orden en que deben realizarse.
- Qué productos se obtienen después de cada actividad.

Por último, esta guía nos permite una difusión fácil y accesible del CMM nivel 3 para cualquier empresa o usuario, lo cual permite e incrementa sus probabilidades de éxito al tratar de implantar el nivel 3 de CMM.

Los beneficios de implantar el nivel 3 de CMM son:

- La organización ha definido sus procesos, métodos y herramientas estándar que se usaran en todos los proyectos.
- Los procesos junto con los elementos relacionados están documentados y puestos a disposición.
- El personal ha recibido la capacitación necesaria para comprender y realizar con eficiencia el proceso estándar de la organización.
- Se realizan las tareas de Ingeniería de software para desarrollar y mantener el software de acuerdo al proceso de software.
- Se establecen los medios de comunicación entre los diferentes departamentos y proyectos, con esto la organización asegura que exista una visión común del producto,

- Se realizan revisiones tempranas para mejorar la calidad de los productos y eliminar defectos tempranamente.

Finalmente podemos concluir que estamos satisfechos de este trabajo, ya que con esta guía se ayudará a la gente que se dedica a desarrollar software, a comprender y aprovechar las prácticas que propone CMM.

Trabajos a Futuro.

- Darlo a conocer. La guía de interpretación y comprensión del CMM nivel 3, estará disponible en un sitio web, para que sea consultada.

Hacer pruebas para confirmar que el modelo gráfico ayuda a la comprensión. Para tales pruebas, se podrán elaborar cursos de comprensión en diferentes empresas, que se dediquen a elaborar software, y proporcionar esta guía a estudiantes que en sus proyectos quieran utilizar CMM.

Bibliografía

CMM Practices, versión 1.1 (SEI/CMU-93-TR25).

La pagina del SEI. <http://www.sei.cmu.edu/>

Grady Booch - James Rumbaugh - Ivar Jacobson,
El lenguaje Unificado de modelado.
Addison Wesley 1999.

Martin Fowler - Kendall Scott,
UML Gota a gota.
Addison Wesley 1999.

Hanna Oktaba - Claudia Alquira Esquivel.
Process Diagrams: An extension of UML Activity Diagrams for the Modeling
of Software Processes.

Claudia Alquira Esquivel, "Guía de interpretación del nivel 2 de CMM"
Tesis de maestría en ciencia de la computación, UNAM 2000.

Anexo 1

En este anexo se presenta una introducción a los diagramas de actividades.
Al comienzo de este anexo se incluye una tabla con los símbolos utilizados en estos diagramas.

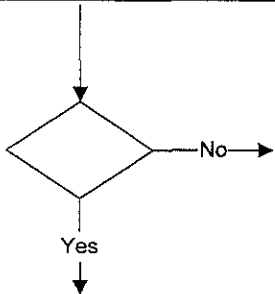
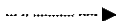
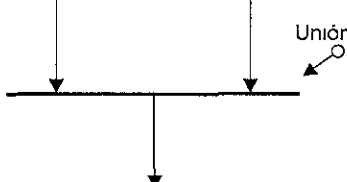
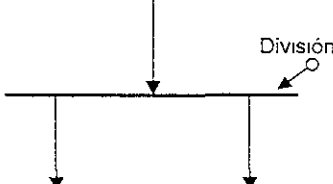
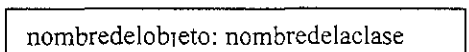
	Inicio.
	Fin.
	Bifurcación. Especifica caminos alternativos.
	Actividad.
	Actividad a desglosar.
	Flujo de Objetos.
	Transición.
	Unión Una unión puede tener dos o más transiciones de entrada y una transición de salida.
	División. Una división puede tener una transición de entrada y dos o más transiciones de salida.
	Objeto

Tabla de símbolos utilizados en los Diagramas de Actividades.

Diagramas de Actividades

Se presentará una breve explicación de los elementos de los diagramas de actividades. Este tipo de diagrama está incluido en UML (Lenguaje de Modelado Unificado).

Un diagrama de actividades es:

- Un diagrama de flujo que muestra el flujo de control entre las actividades a lo largo del tiempo.
- Estos diagramas son útiles para la descripción del comportamiento que tiene una gran cantidad de procesos en paralelo.
- Permite seleccionar el orden en el que se harán las cosas, define las reglas de secuenciación.
- Son útiles para visualizar, especificar, construir y documentar la dinámica de los objetos.

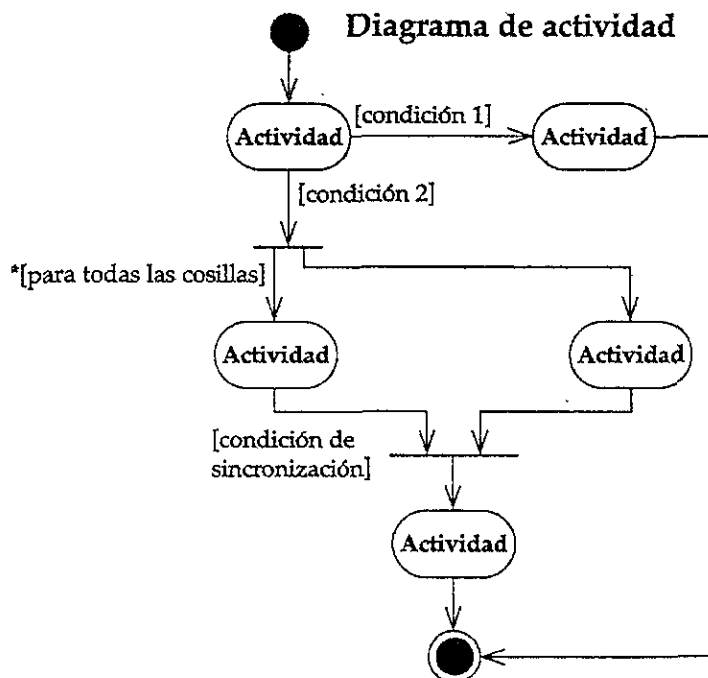


Figura 1. Ejemplo de Diagrama de actividades.

Actividades.

En un diagrama conceptual una actividad es una tarea que debe ser realizada, por alguien ya sea un ser humano, un grupo, una computadora, etc.

Cuando una actividad finaliza, el flujo de control pasa inmediatamente a la siguiente actividad. Se especifica este flujo por medio del uso de transiciones. Las transiciones se usan para mostrar el camino a seguir de una actividad a la siguiente actividad.

En UML una transición se representa con una línea dirigida y una actividad como una elipse como veremos a continuación.

Ejemplo: En la figura 2 el diagrama de actividades contiene dos actividades que van a ser realizadas por la organización. Como podemos ver la organización primero debe realizar la actividad “Establecer equipos de Ingeniería interdisciplinarios” una vez que realice esta actividad continuará con la siguiente actividad que será “Coordinar Grupos”.

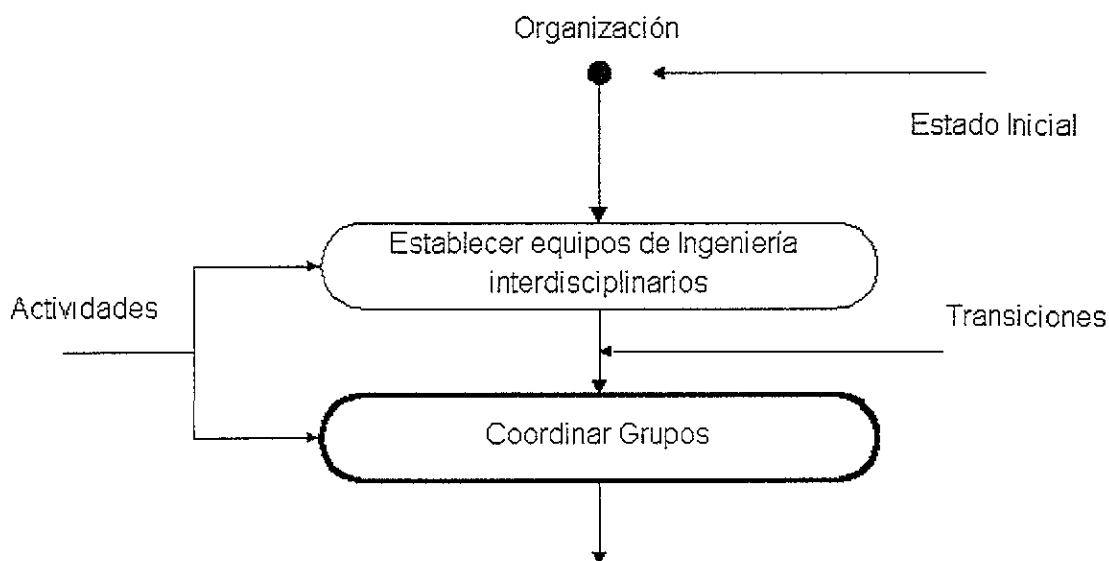


Figura 2 Transiciones.

Bifurcación.

Las transiciones secuenciales son muy comunes, pero habrá ocasiones en que para modelar un flujo de control necesitemos incluir una bifurcación que especificará caminos alternativos según el resultado de una condición.

En UML una bifurcación se representa con un rombo. Una bifurcación tiene una transición de entrada y dos o más de salida. En cada transición de salida se coloca una condición, que se evalúa solo una vez al entrar en la bifurcación.

Ejemplo: En la figura 3 diagrama la organización debe verificar si existe un grupo de capacitación para poder realizar la actividad “Proveer recursos y el financiamiento adecuado”. Si no existe el grupo de capacitación debe realizar la actividad “Constituir el grupo de capacitación” y si ya existe realizará la actividad “Proveer recursos y el financiamiento adecuado”

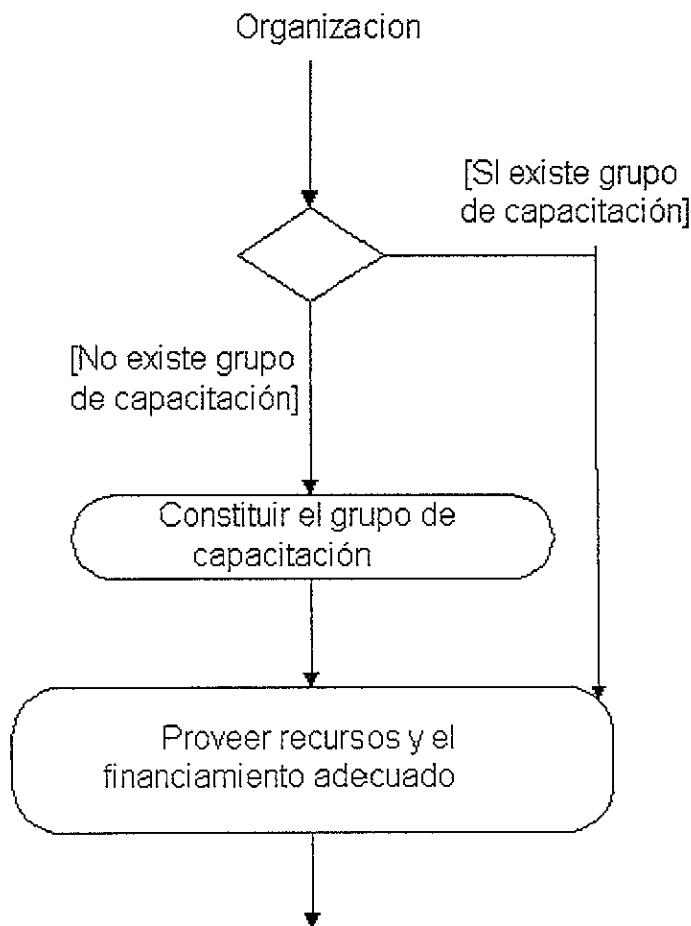


Figura 3 Bifurcación.

División y Unión.

Las transiciones secuenciales y las bifurcaciones se utilizan mas a menudo pero en los diagramas de actividades también es posible encontrar flujos concurrentes.

En UML se utiliza una barra de sincronización para especificar la división y unión de estos flujos concurrentes.

La división representa la separación de un flujo de control sencillo en dos o más flujos de control concurrentes. Una división puede tener una transición de entrada y dos o más transiciones de salida, cada una de las cuales representa un flujo de control independiente.

Después de la división, las actividades asociadas a cada uno de estos caminos actúan en paralelo.

Una unión representa la sincronización de dos o más flujos de control concurrentes. Una unión puede tener dos o más transiciones de entrada y una transición de salida. Antes de llegar a la unión, las actividades asociadas con cada uno de los caminos continúan en paralelo, como vemos en la figura 4.

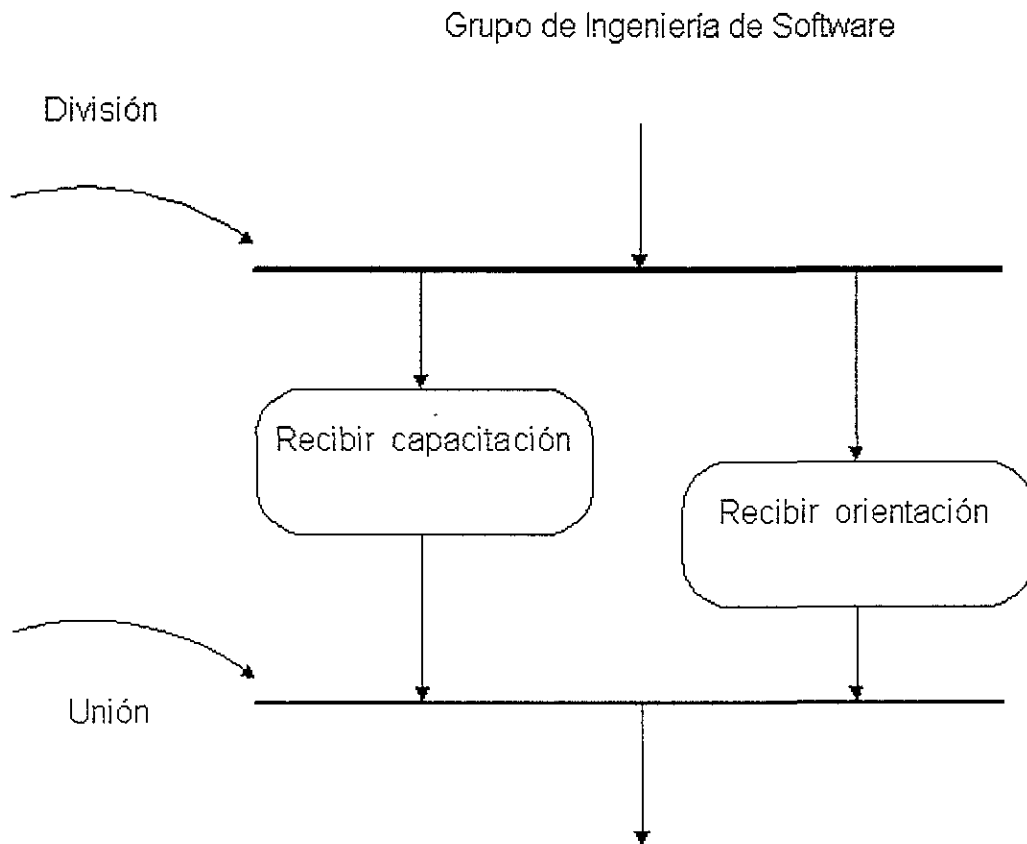


Figura 4 División y Unión.

Carriles (Swimlanes).

Los diagramas de actividades se pueden dividir en grupos, donde cada uno de estos grupos es representado por un responsable de la organización, quién tiene que realizar las actividades contenidas en dicho grupo. En UML, estos grupos son llamados carriles porque visualmente cada grupo es separado de sus vecinos por una línea vertical.

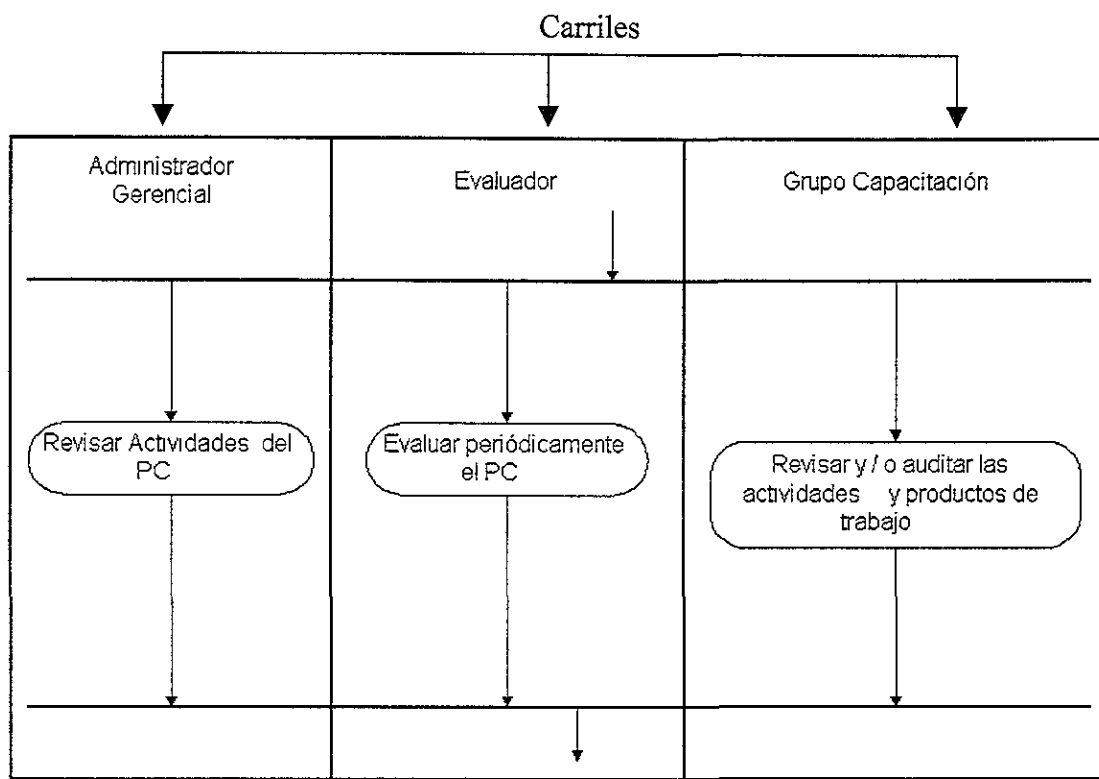


Figura 5 Carriles.

Flujo de objetos.

En el flujo de control de un diagrama de actividades, también podemos especificar los objetos implicados en el diagrama de actividades. Como se muestra en la figura 6 se pueden especificar los objetos implicados en un diagrama de actividades colocándolos en el diagrama, conectados con una dependencia a la actividad o transición que los crea, los destruye o los modifica. Este uso de las relaciones de dependencia y de los objetos se denomina flujos de objetos, por que representa la participación de un objeto en un flujo de control.

En UML, un objeto se representa por un rectángulo con un identificador subrayado, el cual está constituido por el nombre del objeto, dos puntos y el nombre de la clase.

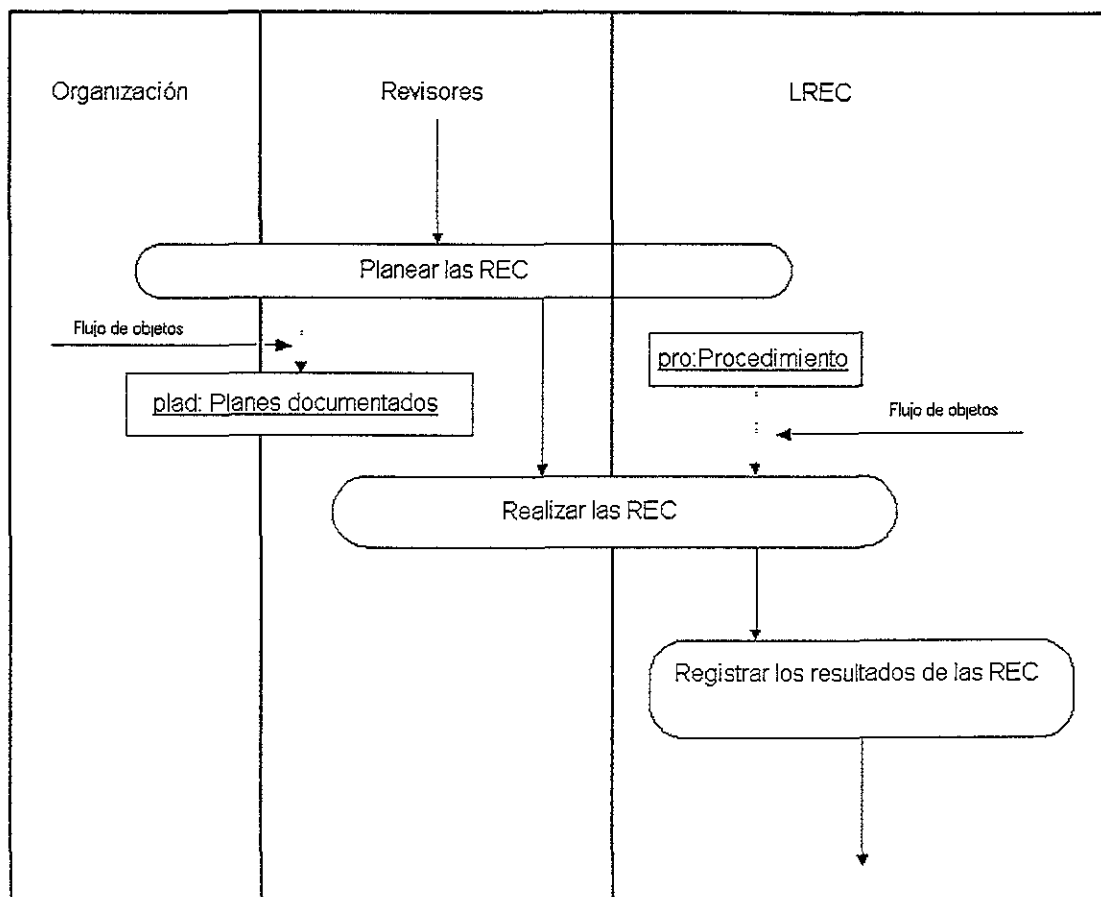


Figura 6 Flujo de Objetos.

Extensión a los diagramas de actividades.

Los elementos de los diagramas de actividades, que hemos explicado en las secciones anteriores, son adecuados para modelar gráficamente las áreas claves de procesos del nivel 3 de CMM. En este trabajo se utilizaron dos elementos mas en estos diagramas que se propusieron en [4tas. Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software cuyas autoras son Dra. Hanna Oktaba y Claudia Esquivel], que en esta sección explicaremos.

Desglose de actividades.

Algunas veces una actividad puede ser vista como un elemento compuesto, cuyo flujo de control se compone de otras actividades, para hacer explícita esta característica, se utiliza el desglose de actividades, es decir, representar gráficamente cuando una actividad será desglosada y que el diagrama correspondiente será presentado posteriormente. Así tendremos la facilidad de mostrar en forma gráfica diferentes niveles de abstracción.

La actividad a desglosar tendrá mas grueso el contorno de la elipse que todas las demás actividades, esto indicará que esta actividad se descompone en otro diagrama que muestra las actividades que están contenidas en esa actividad.

El desglose de una actividad es útil para comprender y entender en forma progresiva flujos de trabajo mas complicados. Por lo tanto en este trabajo se utilizaran cuando sea necesario el desglose de actividades para dar un mejor entendimiento y comprensión del CMM.

Por ejemplo en la figura 7, la actividad A3 tiene la elipse mas gruesa eso indica que esta actividad se desglosará en otro diagrama que nos muestra las actividades que están contenidas en la actividad A3. Este diagrama los vemos en la figura 8.

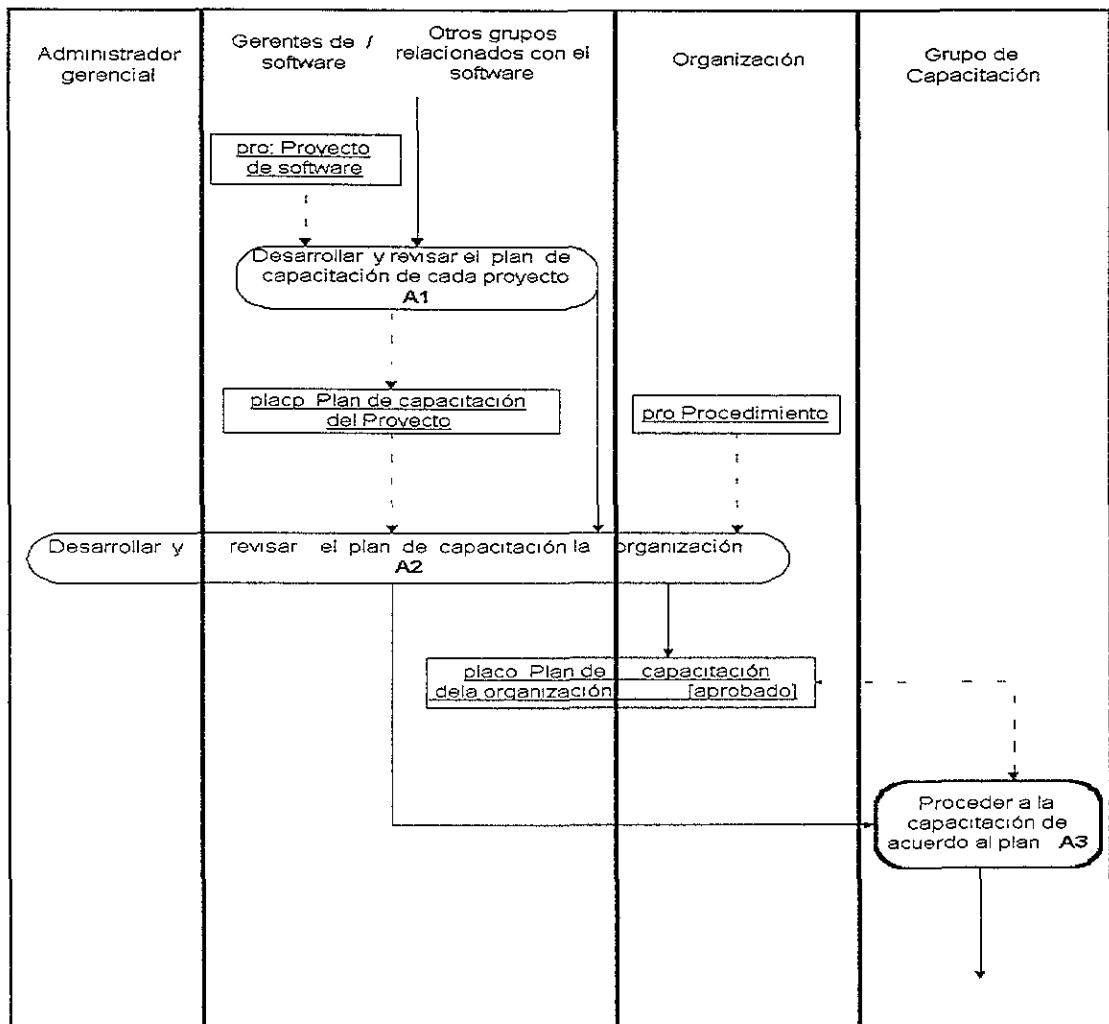


Figura 7 Indica que la Actividad A3 será desglosada en otro diagrama.

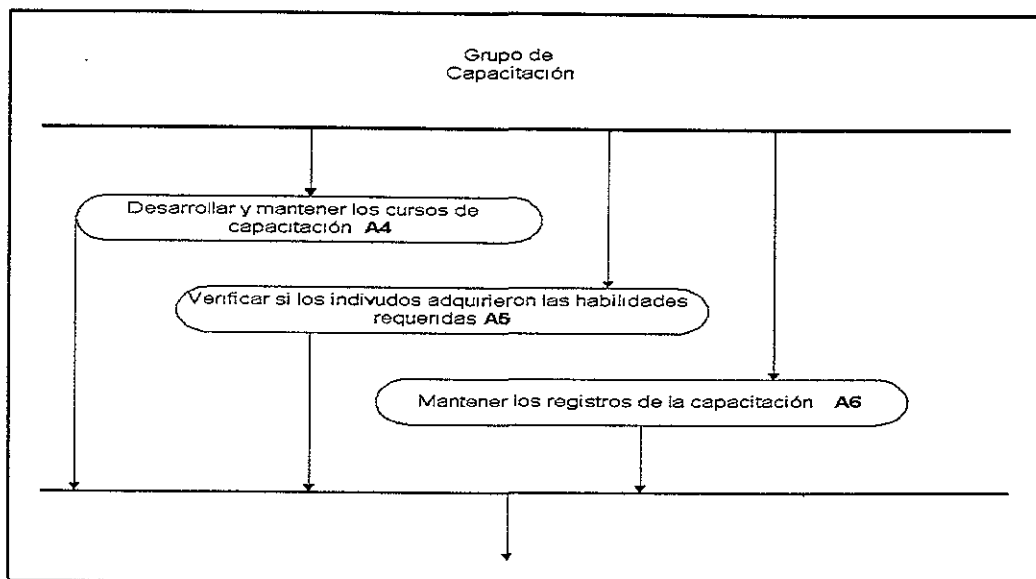


Figura 8 Actividad 3 desglosada.

Responsabilidades compartidas.

Se ha mencionado anteriormente que cada actividad pertenece solamente a un carril. Hay veces que es útil permitir que una actividad pueda cruzar uno o ambos límites del carril, esto significa que esta actividad puede ser realizada por mas de un responsable de una organización. Es decir representaremos responsabilidades compartidas sobre una actividad.

Por ejemplo en la figura 9 para el caso de las mediciones hay veces que esta responsabilidad la comparten dos o mas individuos de la organización, es decir las mediciones las llevan acabo El Administrador Gerencial, y el Grupo de Capacitación. Por tal caso esta representación gráfica es más adecuada. Ver figura 9.

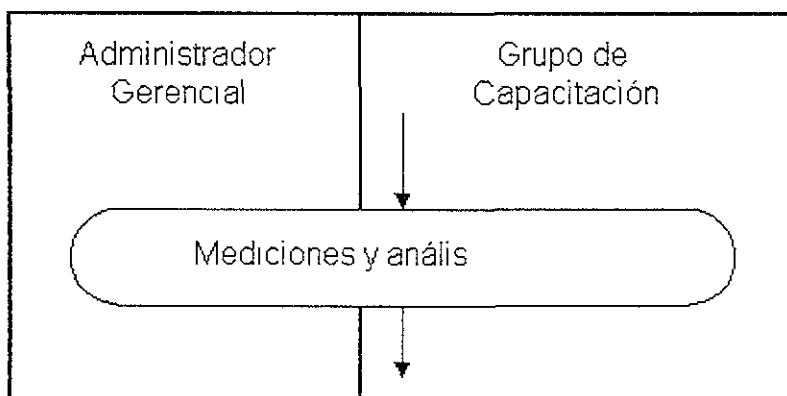


Figura 9 Responsabilidades compartidas.

Anexo 2

En este anexo se presenta la traducción del área clave de proceso Administración de Subcontratos de Software del nivel 2 de CMM, así como su modelo gráfico que incluye la presentación de los grupos participantes, así como una descripción de sus roles y se concluye con la explicación de los diagramas de actividades que conforman el modelo gráfico.

Administración de Subcontratos de Software

Un área clave de proceso para el nivel 2: Repetible

El propósito de la Administración de Subcontratos de Software es seleccionar subcontratistas de software calificados y administrarlos efectivamente.

La Administración de Subcontratos de Software involucra la selección del subcontratista, establecer compromisos con el subcontratista, y el seguimiento y revisión del desempeño y resultados del subcontratista.

Estas practicas cubren la Administración de Subcontratos de Software (solamente), así como también la administración de un componente de software de un subcontrato que incluye software, hardware y posiblemente otros componentes de sistema.

El subcontratista es seleccionado con base en su habilidad para desarrollar el trabajo. Muchos factores pueden contribuir para la decisión de subcontratar una porción del trabajo del contratista. Los subcontratistas pueden ser seleccionados sobre la base de alianzas comerciales estratégicas, o bien por razones técnicas. Las practicas de esta área clave de proceso abarcan el proceso tradicional de adquisición asociado con subcontratar una porción definida del trabajo con otra organización.

Al subcontratar, se crea un acuerdo documentado cubriendo tanto los requerimientos técnicos como no técnicos (por ejemplo: fechas de entrega) y es usado como base para administrar el subcontrato. El trabajo a ser realizado por el subcontratista y los planes para el trabajo son documentados. Los estándares que son para ser seguidos por el subcontratista son compatibles con los estándares del contratista.

Las actividades de planificación de software, seguimiento, y control para el trabajo subcontratado son ejecutadas por el subcontratista. El contratista asegura que esas actividades de planificación, seguimiento y control son ejecutadas apropiadamente y que los productos de software entregados por el subcontratista satisfacen sus criterios de aceptación. El contratista trabaja con el subcontratista para administrar las interfaces de productos y procesos.

Metas

Meta 1

El contratista selecciona subcontratistas de software calificados.

- Meta 2** El contratista y el subcontratista acuerdan compromisos recíprocos.
- Meta 3** El contratista y el subcontratista de software mantienen una comunicación continua.
- Meta 4** El contratista hace un seguimiento al subcontratista de software, de los resultados reales y del desempeño en relación con sus compromisos.

Compromisos a realizar

- Compromiso 1** El proyecto sigue una política de organización escrita para la administración del subcontrato de software.

Esta política, típicamente, especifica:

1. Se usan estándares y procedimientos documentados para la selección y la administración de los subcontratistas de software.
2. Los acuerdos contractuales son la base para la administración del subcontrato.
3. Los cambios en el subcontrato se hacen con la participación y acuerdo del contratista y el subcontratista.

- Compromiso 2** Se designa un administrador de subcontratos para hacerse responsable de establecer y administrar el subcontrato de software.

1. El administrador del subcontrato tiene experiencia en ingeniería de software o tiene personas asignadas que tienen este conocimiento y experiencia.
2. El administrador del subcontrato es responsable de coordinar el alcance técnico del trabajo a ser subcontratado y los términos y condiciones del subcontrato con las partes afectadas.

El grupo de ingeniería de sistemas y el grupo de ingeniería de software del proyecto definen el alcance técnico del trabajo a ser subcontratado.

La función de los grupos apropiados del negocio, tales como los compradores, los de finanzas y los jurídicos es establecer y controlar los términos y condiciones del subcontrato.

3. El administrador del subcontrato es responsable por:

- ☐ seleccionar al subcontratista de software
- ☐ administrar el subcontrato y
- ☐ organizar el apoyo post-subcontrato de los productos subcontratados.

Habilidades a realizar

Habilidad 1

Se proveen los recursos y el financiamiento adecuado para seleccionar al subcontratista y administrar el subcontrato.

1. Se asignan responsabilidades específicas a los administradores del software y otros individuos por administrar el subcontrato.
2. Se ponen a disposición herramientas para apoyar la administración del subcontrato.

Ejemplos de herramientas de apoyo incluyen:

- modelos de estimación,
- hojas de cálculo, y
- programas de administración y programación de proyectos.

Habilidad 2

Los administradores del software y otros individuos que están involucrados en el establecimiento y administración de los subcontratos están capacitados para realizar esas actividades.

Ejemplos de capacitación incluyen:

- preparación y planificación para la subcontratación de software,

- evaluación de la capacidad del proceso de software de un postor del subcontrato,
- evaluación de las estimaciones y planes de un postor del subcontrato,
- selección de un subcontratista, y
- administración de un subcontrato.

Habilidad 3

Los administradores del software y otros individuos que están involucrados en la administración del subcontrato de software reciben orientación en los aspectos técnicos del subcontrato.

Ejemplos de orientación incluyen:

- dominio de aplicación,
- tecnologías de software en aplicación,
- herramientas de software en uso,
- metodologías en uso,
- estándares en uso, y
- procedimientos en uso.

Actividades a realizar

Actividad 1

El trabajo a ser subcontratado es definido y planeado de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las actividades y productos de software a ser subcontratados son seleccionados con base en una valoración equilibrada de acuerdo a las características técnicas y no técnicas del proyecto.
 - Las funciones o subsistemas a ser subcontratados son seleccionados para emparejar las habilidades y capacidades de potenciales subcontratistas.
 - La especificación de los productos y actividades de software a ser subcontratadas es determinada basándose en un análisis sistemático y una partición

adecuada de los requerimientos de sistema y de software.

2. La especificación del trabajo a ser subcontratado, los estándares y procedimientos a ser seguidos se derivan del proyecto

- ☐ la orden de trabajo,
- ☐ los requerimientos de sistema asignados al software,
- ☐ los requerimientos de software,
- ☐ el plan de desarrollo del software y,
- ☐ los estándares de software y procedimientos.

3. Una orden de trabajo es:

- ☐ preparada,
- ☐ revisada,
- ☐ acordada,

Ejemplos de individuos que revisan y acuerdan la orden de trabajo del subcontratista incluyen:

- el administrador del proyecto,
- el administrador del proyecto de software,
- los administradores de software responsables,
- el administrador de configuración de software,
- el administrador de aseguramiento de calidad del software, y
- el administrador de subcontratos.

- ☐ revisada cuando es necesario, y
- ☐ administrada y controlada.

Administrado y controlado implica que la versión del producto de trabajo de software en un determinado momento (pasado o presente) es conocida y los cambios son incorporados y controlados.

Si se desea un mayor grado de control del que es aplicado por el administrado y controlado el producto de trabajo puede ser puesto bajo la disciplina total de la Administración de configuración como es descrita en el área clave de proceso Administración de Configuración de software.

4. Se prepara un plan para seleccionar un subcontratista de manera concurrente con la orden de trabajo.

Actividad 2

El subcontratista es seleccionado, con base en una evaluación de la capacidad del candidato al subcontrato de realizar el trabajo, de acuerdo a un procedimiento documentado.

Este procedimiento cubre la evaluación de:

1. Propuestas enviadas para el subcontrato planificado.
2. Registros de desempeño en trabajos similares, si están disponibles.
3. La ubicación geográfica de los candidatos al subcontrato.
4. Las capacidades de administración e ingeniería de software.
5. El personal disponible para realizar el trabajo.
6. Experiencia previa en aplicaciones similares, incluyendo experiencia en software del equipo administrador de software del subcontratista.
7. Recursos disponibles.

Ejemplos de recursos incluyen:

- medios,
- hardware,
- software, y
- capacitación.

Actividad 3

El acuerdo contractual entre el contratista y los subcontratistas se usa como base para administrar el subcontrato.

El acuerdo contractual documenta:

1. Los términos y condiciones.
2. La orden de trabajo.

3. Los requerimientos para los productos a ser desarrollados.
4. La lista de dependencias entre el subcontratista y el contratista
5. Los productos subcontratados que serán entregados al contratista.

Ejemplo de productos incluyen:

- código fuente,
- plan de desarrollo de software,
- ambiente de simulación,
- documentación del diseño, y
- el plan de pruebas de aceptación.

6. Las condiciones bajo las cuales se someterán a revisiones los productos.
7. Los procedimientos y criterios de aceptación para ser usados en la evaluación del producto subcontratado antes de la aceptación por el contratista.
8. Los procedimientos y criterios de evaluación a ser usados por el contratante para monitorear y evaluar el desempeño del subcontratista.

Actividad 4

Un plan documentado de desarrollo del software subcontratado es revisado y aprobado por el contratista.

1. Este plan de desarrollo de software cubre (directamente o por referencia) los detalles apropiados del plan de desarrollo de software del contratante.

Actividad 5

Un plan de desarrollo de software documentado y aprobado se usa para el seguimiento de las actividades de software y comunicar su estado.

Actividad 6

Los cambios a la orden de trabajo, los términos y condiciones del subcontrato y otros acuerdos son resueltos de acuerdo a un procedimiento documentado.

1. Este procedimiento típicamente especifica que están involucrados todos los grupos afectados, tanto del contratista como del subcontratista.

Actividad 7

La administración del contratista conduce revisiones periódicas, de estado / coordinación con la administración del subcontratista.

1. Se le entrega al subcontratista la visión de las necesidades y deseos del cliente del producto y de los usuarios finales si corresponde.
2. Se revisan, contra el plan de desarrollo del software del subcontratista, el desempeño técnico, costos, personal y programación.
3. Se revisan los recursos computacionales designados como críticos para el proyecto; la contribución del subcontratista a las estimaciones actuales es seguida y comparada con las estimaciones para cada componente de software documentado en el plan de desarrollo de software del subcontratista.
4. Se abordan las dependencias críticas y compromisos entre el grupo de ingeniería de software del subcontratista y otros grupos del subcontratista.
5. Se abordan las dependencias y acuerdos críticos entre el contratista y el subcontratista.
 - ☐ Se revisan los compromisos del subcontratista con el contratista y viceversa.
6. Se abordan no conformidades con el subcontrato.
7. Se abordan los riesgos de proyecto que involucran el trabajo del subcontratista.
8. Se abordan conflictos y problemas no resueltos internamente por el subcontratista.
9. Se asignan elementos de acción, se revisan y se siguen hasta que se terminen.

Actividad 8

En forma periódica se realizan intercambios y revisiones técnicas, con el subcontratista de software.

Estas revisiones:

1. Proveen al subcontratista de visibilidad de las necesidades y deseos del cliente y los usuarios finales, según corresponda.
2. Monitorear las actividades técnicas del subcontratista
3. Verifican que la interpretación e implementación de los requisitos técnicos, por parte del subcontratista, estén de acuerdo con los requisitos del contratista.
4. Verifican que los acuerdos se están cumpliendo.
5. Verifican que los aspectos técnicos se están resolviendo de forma oportuna.

Actividad 9

Se conducen revisiones formales para abarcar los resultados y logros de la ingeniería de software del subcontratista en hitos selectos y de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las revisiones son preplanificadas y documentadas en la orden de trabajo.
2. Las revisiones abordan los acuerdos para planes y estado de las actividades de software.
3. Se identifican y documentan los problemas significativos, los elementos de acción y las decisiones.
4. Se abordan los riesgos del software.
5. El plan de desarrollo del software del subcontratista se refina, según lo apropiado.

Actividad 10

El grupo de aseguramiento de la calidad de software del contratista monitorea las actividades de aseguramiento de la calidad de software del subcontratista de acuerdo a un procedimiento documentado

Este procedimiento típicamente especifica que:

1. Los planes, recursos, procedimientos y estándares para aseguramiento de calidad del subcontratista, son revisados periódicamente para asegurar que ellos son

adecuados para monitorear el desempeño del subcontratista.

2. Se conducen revisiones regulares del subcontratista para asegurar que se están siguiendo los procedimientos y estándares aprobados.
 - El grupo de aseguramiento de calidad de software del contratante revisa las actividades y productos de ingeniería de software del subcontratista
 - El grupo de aseguramiento de calidad de software del contratista audita los registros de aseguramiento de calidad de software del subcontratista, según corresponda.
3. Los registros de las actividades de aseguramiento de calidad del subcontratista son auditados periódicamente para evaluar cuan bien se están siguiendo los planes, estándares y procedimientos de aseguramiento de calidad de software.

Actividad 11

El grupo de administración de la configuración del contratante monitorea las actividades de administración de la configuración del subcontratista, de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Se revisan los estándares, planes, recursos y procedimientos de la administración de configuración del subcontratista para asegurar que son adecuados.
2. El contratista y el subcontratista coordinan sus actividades en materias relacionadas a la administración de la configuración para asegurar que los productos del subcontratista pueden ser rápidamente integrados o incorporados en el ambiente del proyecto del contratista.
3. La biblioteca de líneas base de software del subcontratista es revisada periódicamente para evaluar que tan bien son seguidos los procedimientos para la administración de configuración del software y que tan efectivos son ellos en administrar la línea base de software.

Actividad 12

El contratista conduce pruebas de aceptación como parte de la entrega de los productos de software del subcontratista de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Están definidos, revisados y aprobados los criterios de aceptación para cada producto.
2. Los resultados de las pruebas de aceptación son documentados.
3. Se planifican acciones para cada producto de software que no pasa sus pruebas de aceptación.

Actividad 13

Se evalúa, periódicamente, el desempeño de subcontratista de software y la evaluación es revisada con el subcontratista.

La evaluación del desempeño del subcontratista provee una oportunidad para el subcontratista de obtener retroalimentación en ellas sobre si está o no satisfaciendo las necesidades de sus clientes (es decir, las necesidades del contratante). Un mecanismo tal como las revisiones de premio por cuota de desempeño proveen este tipo de retroalimentación, como opuesto a las revisiones periódicas técnicas y de coordinación que ocurren a través del proyecto. La documentación de esas evaluaciones también actúa como una entrada para actividades futuras de selección de subcontratistas

Medición y análisis

Medición 1

Se realizan y se usan las mediciones para determinar el estado de las actividades de la administración del subcontrato de software.

Ejemplos de mediciones:

- costos de las actividades para la administración del subcontrato en comparación con el plan,

- tiempos de entrega reales para productos subcontratados en comparación con el plan,
- tiempos de entrega del contratante al subcontratista en comparación con el plan.

Verificación de la implementación

Verificación 1

Se revisan en forma periódica las actividades para la administración del subcontrato de software, con el Administrador gerencial.

El objetivo principal de las revisiones periódicas con el Administrador gerencial es proporcionarle una visión general y señales de alerta sobre las actividades del proceso de software de manera oportuna. Los lapsos de tiempo entre las revisiones son de acuerdo a las necesidades de la organización y pueden ser muy largos si existe un mecanismo adecuado para reportar situaciones excepcionales

Verificación 2

Se revisan en forma periódica y ante eventos que lo requieran las actividades para la administración del subcontrato de software con el Administrador del proyecto.

Verificación 3

El grupo de aseguramiento de calidad del software, revisa y/o audita las actividades y los productos del trabajo para administrar los Subcontratos del software, y reporta los resultados.

Esta revisión y / o auditoria verifica:

1. Las actividades para seleccionar al subcontratista.
2. Las actividades para administrar el subcontrato.
3. Las actividades para la coordinación de las actividades de administración de la configuración del contratante y el subcontratista.

4. La revisión de la conducción de lo planificado con el subcontratista.
5. La conducción de revisiones que establecen el término de aspectos claves del proyecto para el subcontratista.
6. El proceso de aceptación de los productos de software del subcontratista.

Modelo gráfico de la Administración de Subcontratos de Software (ASS).

La construcción del modelo grafico se inicia con una identificación, en todo el texto original, de los grupos participantes en el proceso de ASS y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

Identificación de los grupos participantes.

En el modelo textual de la administración de subcontratos de software hemos identificado a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentamos una breve definición de cada uno de ellos.

Administrador de Subcontrato: Gerente en la organización del contratante que tiene responsabilidad directa por administrar y manejar uno o más subcontratos.

Administrador Gerencial: Es un rol de administración de un nivel alto en una organización, que tiene como objetivo principal mantener el éxito de la organización a largo plazo. En general un Administrador Gerencial para ingeniería tendrá la responsabilidad de supervisar varios proyectos a la vez.

Administradores de Software: Algún administrador, a nivel proyecto u organización, que tiene la responsabilidad directa o total para el desarrollo y / o mantenimiento del software. Es responsable de administrar los aspectos técnicos y del personal del proyecto de software.

Contratista: Es el responsable de seleccionar al subcontratista, acordar con el los compromisos y de revisar y aprobar el plan documentado de desarrollo del software subcontratado, así como de conducir revisiones periódicas con el subcontratista y de realizar las pruebas de aceptación.

Subcontratista: Un individuo, sociedad, corporación, o asociación que es contratada por una organización, es el responsable por la entrega de los productos de software subcontratados.

Otros individuos: Son los demás involucrados en el Subcontrato de Software.

Grupo de Aseguramiento de Calidad: Revisa y / o audita las actividades y los productos de trabajo para administrar los subcontratos y reporta resultados.

Organización: Es el conjunto de los grupos mencionados anteriormente y de otros grupos involucrados.

Diagrama de actividades

Compromisos.

En esta área clave de procesos contamos con dos compromisos, el primero establece que se debe seguir una política organizacional escrita para la Administración del Subcontrato de software y el segundo es designar un Administrador de Subcontratos.

Analizando el texto del compromiso 1, se deduce que la organización tiene una política escrita que guía la Administración del Subcontrato de Software. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Figura 1.1) como un objeto *p: Política*. Este objeto fue probablemente generado por una actividad de definición de política la cual, según el texto del compromiso queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir una política para la Administración del Subcontrato de Software queda expresado colocando a *p: Política* como objeto de entrada de la actividad de la Administración del Subcontrato de Software (ASS). Este compromiso lo representamos como “Administrar el Subcontrato de Software C1”.

El compromiso 2 dice que se designa un Administrador de Subcontratos que será responsable de establecer y administrar el subcontrato, Este compromiso lo representamos con la actividad Designar un Administrador de Subcontratos y se propone como una actividad realizada por el Administrador Gerencial, aunque el texto no le dice explícitamente pero creemos que es una aseveración congruente.

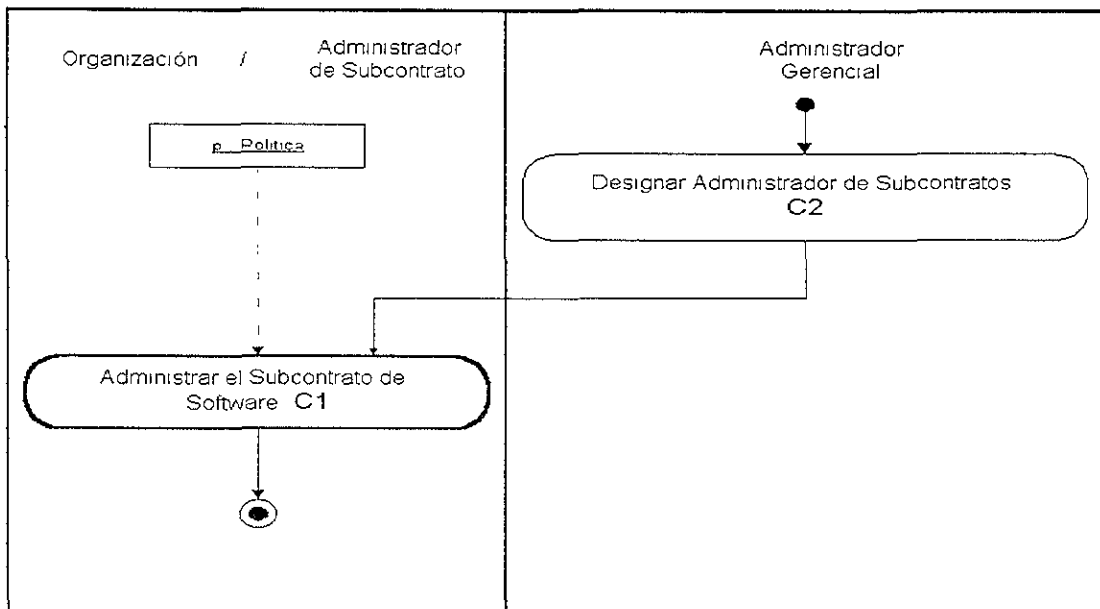


Figura 1.1: Compromisos del área clave de proceso: Administración de Subcontratos de Software (ASS).

Las características siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben tomar para la Administración de Subcontratos de Software. Para modelar este hecho, la actividad de Administración de Subcontratos de Software de la figura 1.1, se puede desglosar en otro diagrama presentado en la figura 1.2.

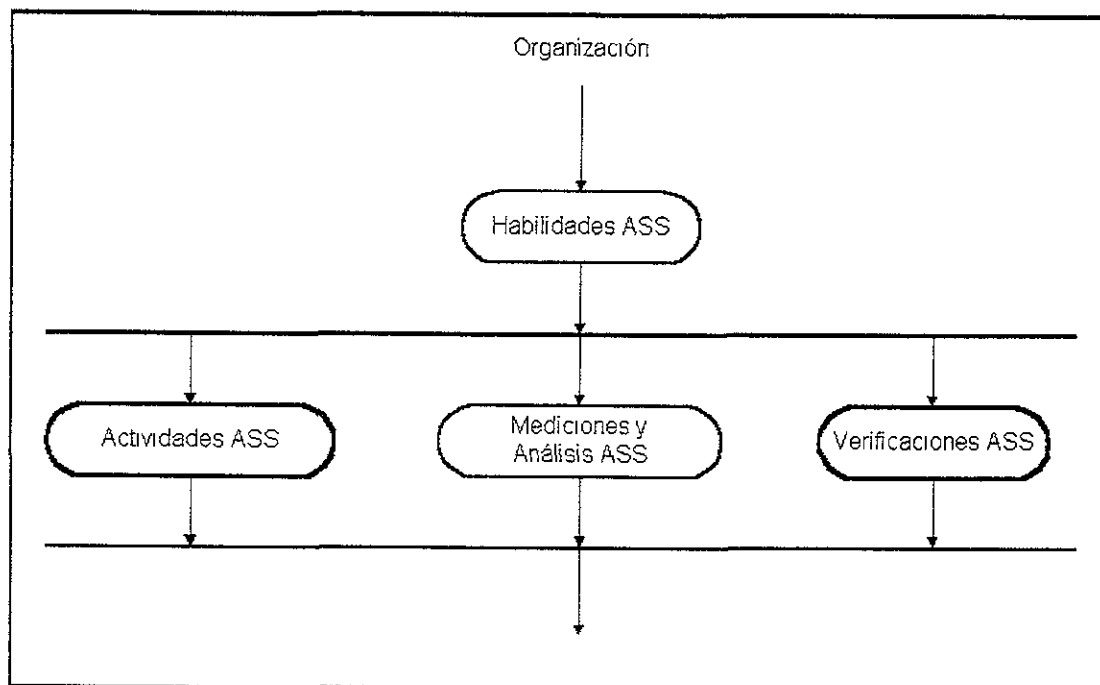


Figura 1.2: Administrar el Subcontrato de Software.

Este diagrama introduce una actividad por cada característica común mencionada y además expresa el orden de ejecución y /o el paralelismo de ellas. Queda claro que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de la Administración de Subcontratos de Software (ASS). Por otro lado, las actividades de verificación y medición con su análisis, son simultaneas a las de ASS.

Habilidades.

La habilidad 1 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo del Administrador de Subcontrato. Esta habilidad la representamos con la actividad “Asignar recursos y financiamiento” para la ASS y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación u orientación.

Las dos ultimas habilidades tratan la capacitación y orientación de los grupos involucrados en este proceso, la habilidad 2 especifica que el administrador de software y otros individuos se capacitan en el establecimiento y la administración de

subcontratos. Esta habilidad la representaremos con la actividad “Capacitación en la ASS”.

La habilidad 3 dice que el administrador de software y otros individuos reciben orientación en los aspectos técnicos del subcontrato. Esta habilidad la representaremos con la actividad “Recibir Orientación”.

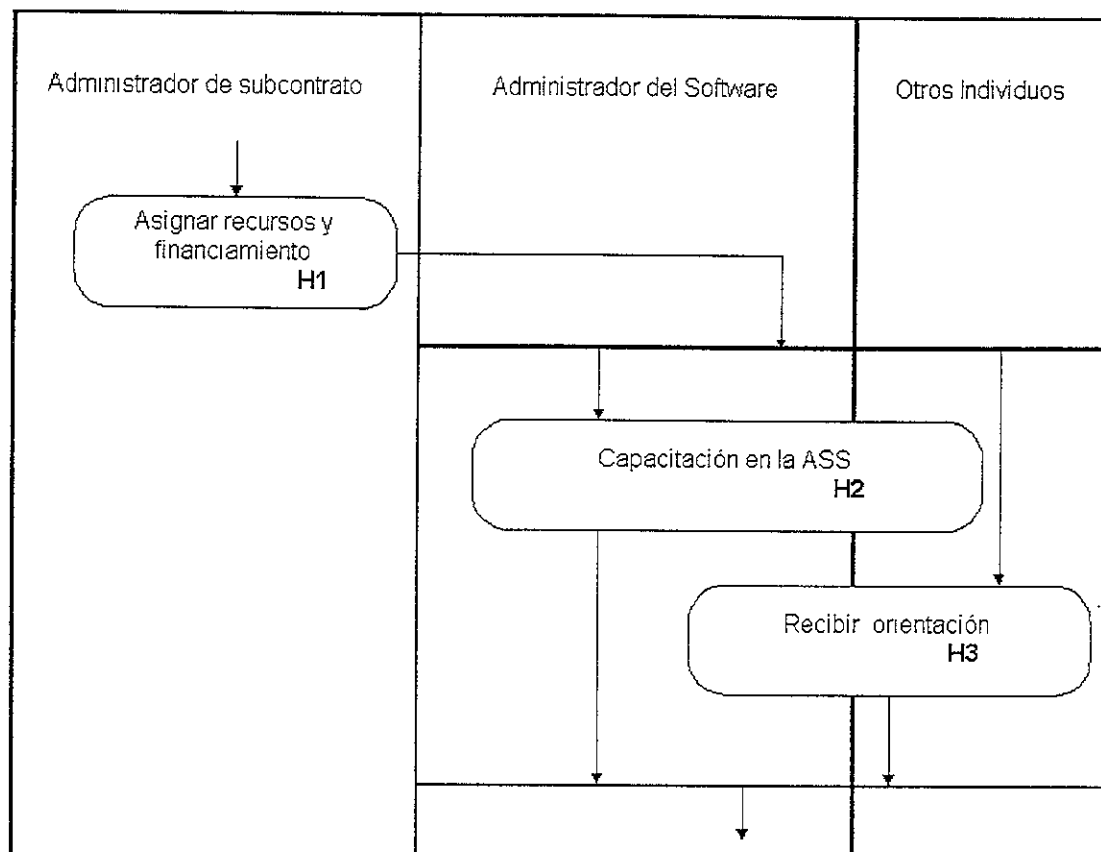


Figura 1.3 Habilidades del área clave de proceso: Administración de Subcontratos de Software (ASS).

El diagrama de la figura 1.3 presenta las actividades a realizarse para adquirir las habilidades para la ASS, donde cada una de las habilidades es mapeada en forma directa a la actividad que corresponde. También en este diagrama es fácil observar que actividades pueden llevarse acabo en forma paralela y el orden en que se realizan.

Actividades.

La actividad 1 especifica que el trabajo a ser subcontratado deber ser definido y planeado. Esta actividad la representamos en el diagrama como “Definir y planear el trabajo” se propone como una actividad realizada por el Administrador de Subcontrato, aunque el texto no le dice explícitamente pero creemos que es una

aseveración congruente. Esta actividad recibe como entrada al objeto **pro: Procedimiento** documentado, que servirá para definir y planear el trabajo. Al término de esta actividad tendremos como salida a **ot: Orden de Trabajo** que fue revisada y acordada por el Administrador de Software en forma compartida con otros individuos, y el **pl: Plan**, este plan se genera para seleccionar a un subcontratista.

La actividad 2 especifica que se debe seleccionar un subcontratista. Esta actividad la representamos como “Seleccionar subcontratista” se propone como una actividad realizada por el Administrador de Subcontrato, aunque el texto no le dice explícitamente pero creemos que es una aseveración congruente. Esta actividad recibe como entrada al objeto **pro: Procedimiento documentado**, que servirá para evaluar al subcontratista y a los objetos **pl: Plan** y **ot: Orden de Trabajo**, ya que con este plan se seleccionara un subcontratista.

La actividad 3 la vamos a dividir en dos actividades. La primera especifica que se debe llegar a un acuerdo entre el contratista y el subcontratista, y según el texto es realizada por el contratista en forma compartida con el subcontratista.

Al término de esta actividad tendremos como salida a **ac: Acuerdo**, ya que este sirve como base para la siguiente actividad, además de que en este acuerdo se documenta los términos y condiciones, la orden de trabajo así como los requerimientos.

La segunda actividad especifica que se debe de administrar el subcontrato. Esta actividad la representamos como “Administrar subcontrato” que de acuerdo al texto es realizada por el contratista en forma compartida con el subcontratista. Esta actividad recibe como entrada al objeto **ac: Acuerdo**. Consideramos que de la actividad 4 a la 12 (revisar, aprobar, seguir un plan de desarrollo de software, cambios a la orden de trabajo y las revisiones periódicas). Por lo tanto quedara desglosada en el diagrama de la figura 1.5.

La actividad 13 especifica que se debe de evaluar periódicamente el desempeño del subcontratista. Esta actividad la representamos como “Evaluar desempeño” que de acuerdo al texto es realizada por el contratista en forma compartida con el subcontratista.

El diagrama de la figura 1.4 presenta las actividades que se describieron anteriormente.

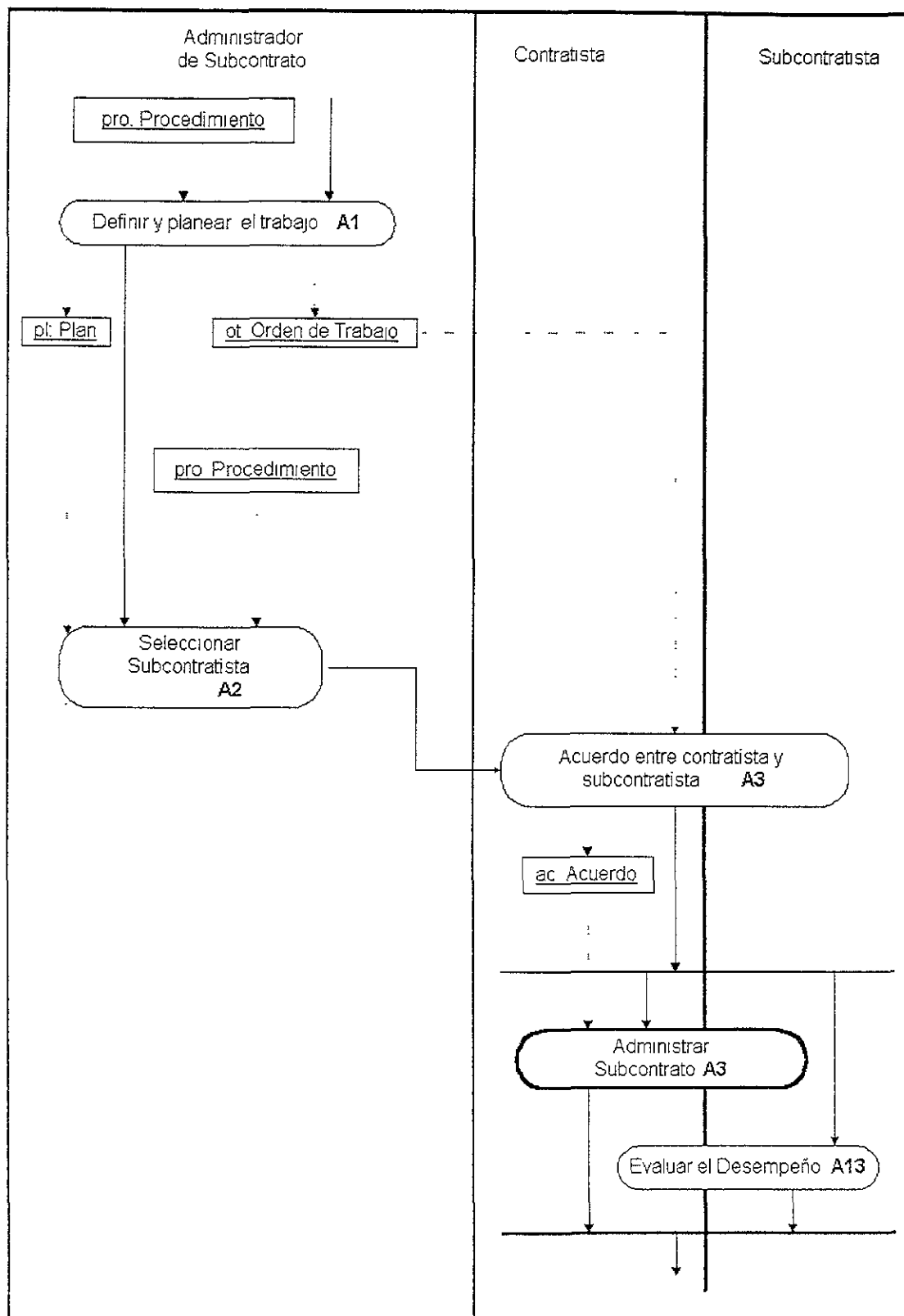


Figura 1.4 Actividades del área clave de proceso: Administración de Subcontratos de Software (ASS).

La actividad 4 especifica que un plan documentado de desarrollo del software subcontratado es revisado y aprobado por el contratista. Esta actividad la representamos como “Revisar y aprobar el plan de desarrolló” que de acuerdo al texto es realizada por el contratista. Esta actividad recibe como entrada al objeto **pld: Plan de desarrollo**. Este cubre directamente o por referencia los detalles apropiados del plan de desarrollo de software del contratante. Al término de esta actividad tendremos como salida a **pld: Plan de desarrollo** que fue aprobada por el contratista.

La actividad 5 especifica que un plan documentado de desarrollo del software subcontratado se usa para el seguimiento de las actividades. Esta actividad la representamos como “Usar para el seguimiento de las actividades y comunicar estado” que de acuerdo al texto es realizada por el Contratista, en forma compartida con el Subcontratista. Esta actividad recibe como entrada al objeto **pld: Plan de desarrollo aprobado**, este cubre directamente o por referencia los detalles apropiados del plan de desarrollo de software del contratante.

La actividad 6 especifica que los cambios a la orden de trabajo, los términos y condiciones del subcontrato son resueltos de acuerdo a un procedimiento documentado. Esta actividad la representamos como “Resolver cambios” que de acuerdo al texto es realizada por el contratista, en forma compartida con el Subcontratista. Esta actividad recibe como entrada al objeto **pro: Procedimiento documentado**.

Las actividades 7,8,9,10,11 especifican que se deben llevar a cabo revisiones periódicas, estas actividades las representamos como “Conducir revisiones periódicas”, estas se desglosaran en otro diagrama que presentaremos mas adelante (figura 1.6).

La actividad 12 especifica que se deben conducir pruebas de aceptación como parte de la entrega de los productos de software del subcontratista. Esta actividad la representamos como “Realizar pruebas de aceptación” que de acuerdo, al texto es realizada por el contratista, y se ejecuta al finalizar las actividades del subcontrato.

El diagrama de la figura 1.5 presenta las actividades que se describieron anteriormente. También en este diagrama es fácil observar que actividades pueden llevarse acabo en forma paralela y el orden en que se realizan.

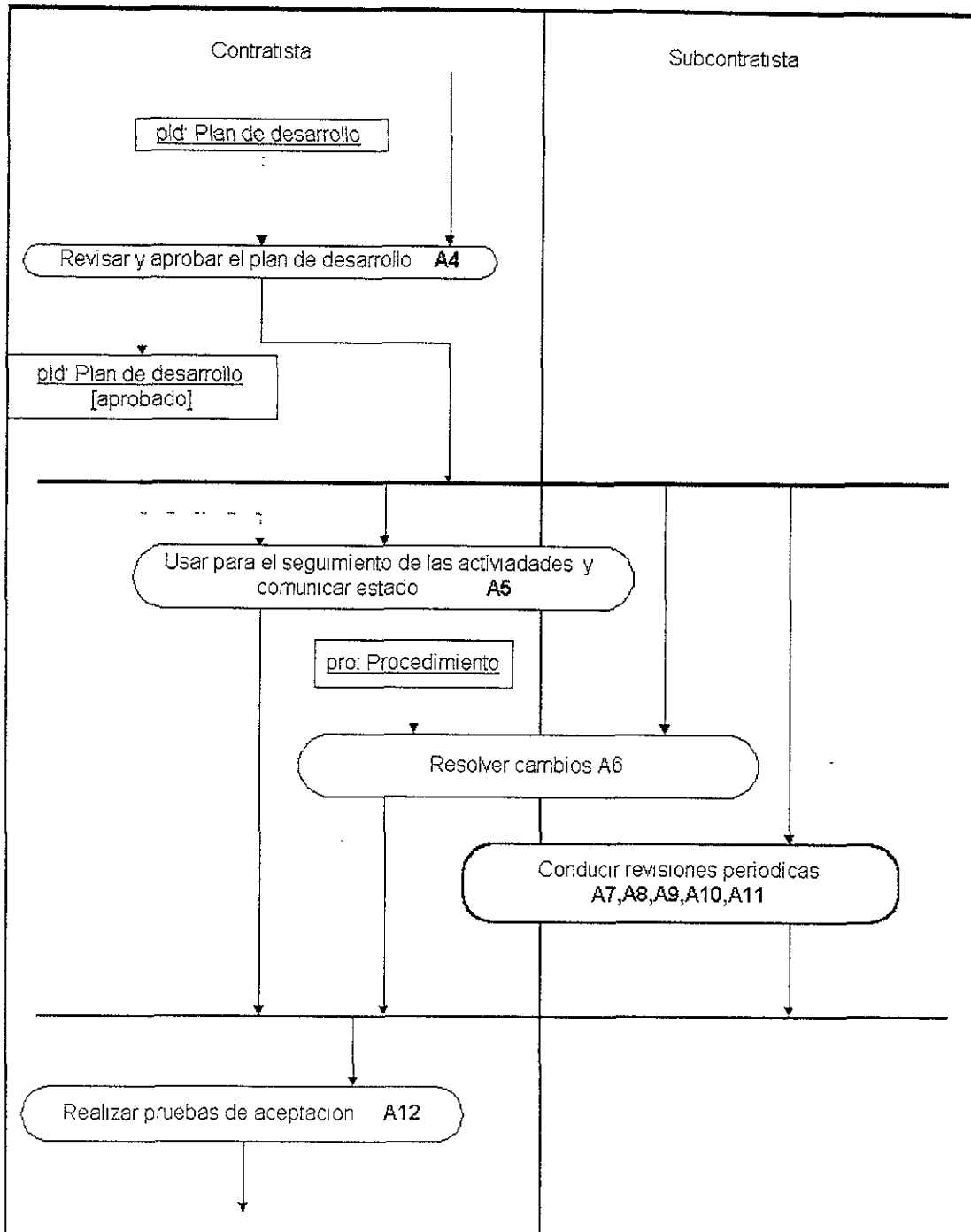


Figura 1.5 Administrar Subcontrato.

La actividad 7 especifica que se deben de realizar revisiones periódicas. Esta actividad la representamos como “Revisar estado / coordinación” que de acuerdo al texto es realizada por el Contratista, en forma compartida con el Subcontratista.

La actividad 8 especifica que se deben realizar intercambios y revisiones técnicas. Esta actividad la representamos como “Realizar cambios y revisiones técnicas” que de acuerdo al texto es realizada por el contratista, en forma compartida con el Subcontratista.

La actividad 9 especifica que se deben conducir revisiones formales. Esta actividad la representamos como “Conducir revisiones formales” que de acuerdo al texto es realizada por el contratista, en forma compartida con el Subcontratista. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 10 especifica que se deben monitorear las actividades de aseguramiento de la calidad de software. Esta actividad la representamos como “Monitorear las actividades de aseguramiento de calidad” que de acuerdo al texto es realizada por el contratista, en forma compartida con el Subcontratista. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

La actividad 11 especifica que se deben monitorear las actividades de administración de la configuración. Esta actividad la representamos como “Monitorear las actividades de administración” que de acuerdo al texto es realizada por el contratista, en forma compartida con el Subcontratista. Esta actividad recibe como entrada al objeto *pro: Procedimiento documentado*.

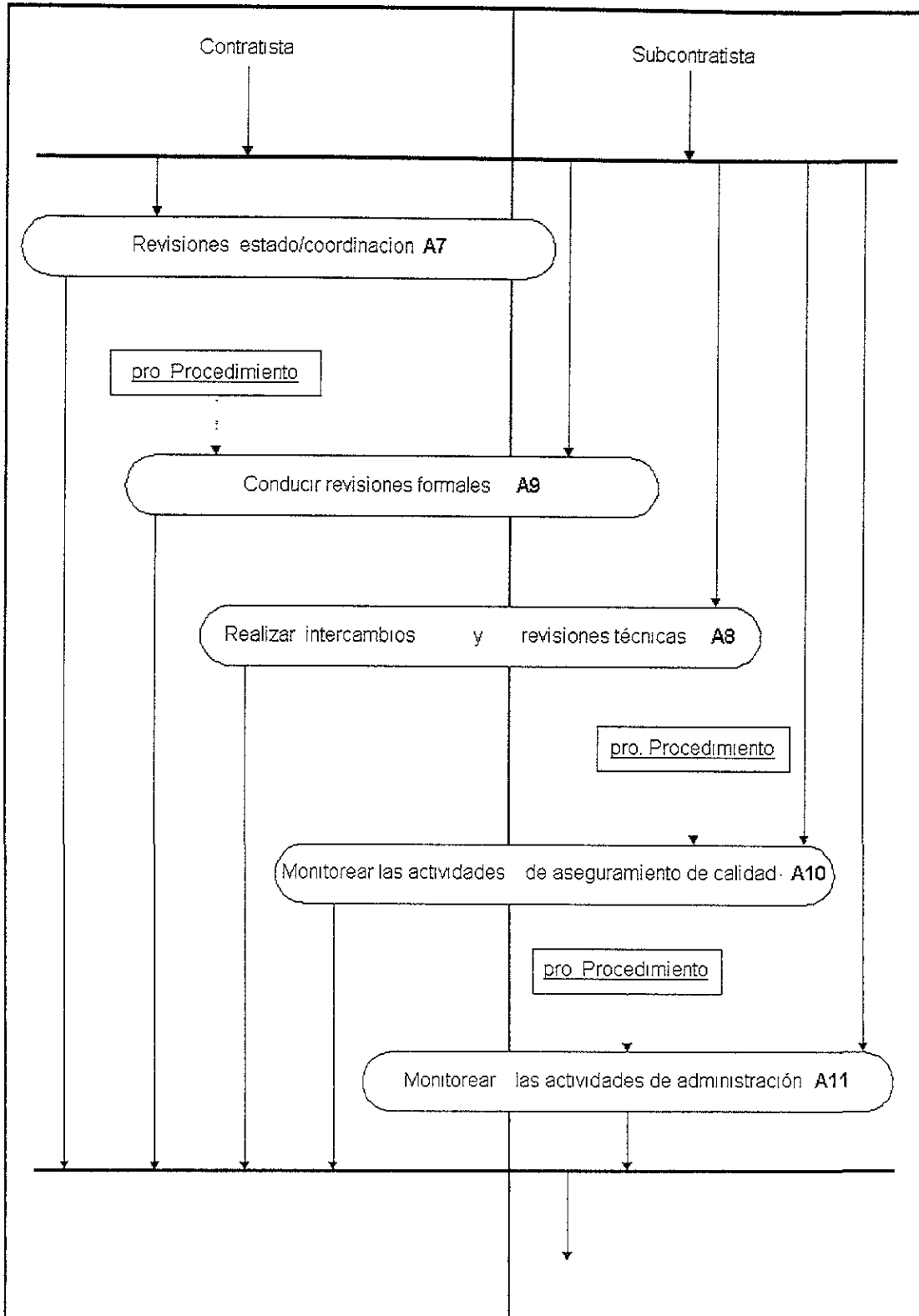


Figura 1.6 Conducir revisiones periódicas.

El diagrama de la figura 1.6 presenta las actividades que se describieron anteriormente. También en este diagrama es fácil observar que actividades pueden llevarse a cabo en forma paralela y el orden en que se realizan.

Mediciones y análisis.

El texto correspondiente dice deben de recabarse y usar las mediciones para determinar el estado de las actividades de la ASS. Nos atreveremos a proponer (Figura 1.7) que la actividad mediciones y análisis es una actividad compartida entre el Contratista, El administrador y el Grupo de Aseguramiento de Calidad, quienes utilizan estos datos para determinar el estado de las actividades de la ASS.

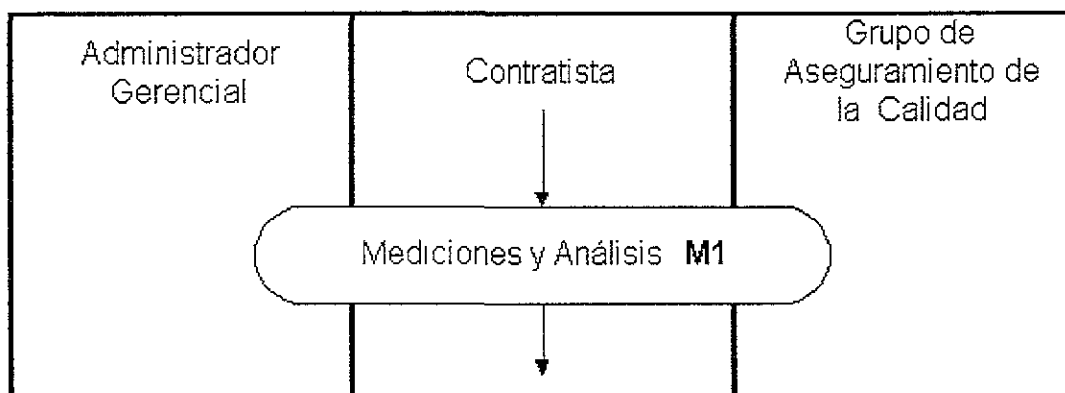


Figura 1.7 Mediciones y Análisis del área clave de proceso: Administración de Subcontratos de Software (ASS).

Verificaciones.

La verificación 1 especifica que las actividades para la administración del subcontrato de software serán revisadas periódicamente por la administración gerencial y la verificación 2 por el administrador del proyecto. Las dos verificaciones las representamos con la actividad de “Revisar Actividades ASS”

Finalmente en la verificación 3 el grupo de aseguramiento de calidad revisa y / o audita las actividades y productos de trabajo para ASS y reporta los resultados. Esta verificación esta representada con la actividad de Revisar y / o auditar actividades y productos de trabajo”.

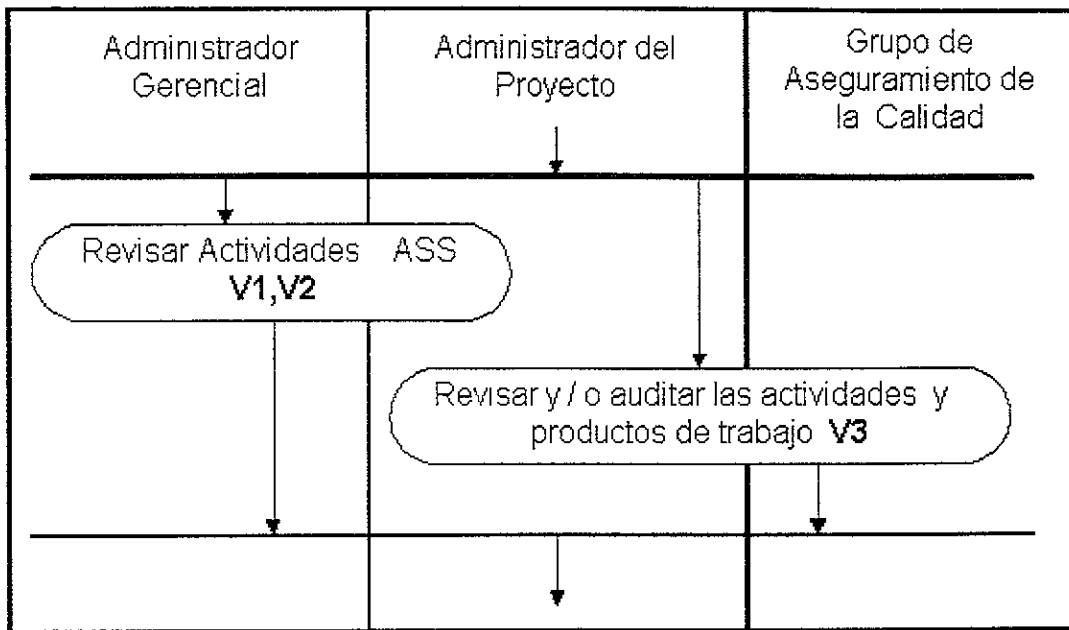


Figura 1.8 Verificaciones del área clave de proceso: Administración de Subcontratos de Software (ASS).

El diagrama de la Figura 1.8 presenta las actividades que se describieron anteriormente.

Anexo 3

En este anexo se presenta un glosario de abreviaciones que se utilizan en este trabajo de tesis, así como sus respectivos significados.

SEI.	Software Engineering Institute (Instituto de Ingeniería de Software).
CMM.	Capability Maturity Model. (Modelo de Madurez de Capacidades).
UML.	Unified Modeling Language. (Lenguaje Unificado de Modelado).
EPO.	Enfoque en el Proceso de la Organización.
DPO.	Definición del Proceso de la Organización.
PC.	Programa de Capacitación.
AIS.	Administración Integrada de Software.
IPS.	Ingeniería de Productos de Software.
CEG.	Coordinación Entre Grupos
REC.	Revisiones Entre Colegas.
ACP.	Áreas claves de procesos.
PSO.	Proceso de Software de la Organización.
PSE.	Proceso de Software Estándar.
RDPSP.	Responsables de Desarrollo del Proceso de Software del Proyecto.
MPTIS.	Miembros del Personal Técnico de Ingeniería de Software.
LREC.	Líder de las Revisiones Entre Colegas
Grupo de IPS.	Grupo de Ingeniería de Procesos de Software.
Grupo de IS.	Grupo de Ingeniería de Software.