

63063
7



Universidad Nacional Autónoma de México

Unidad Académica de los Ciclos Profesionales
y de Posgrado

Modelado de áreas clave de procesos para CMM nivel 2

T e s i s
Que para obtener el grado de:
Maestra en Ciencias de la Computación
presenta:

Claudia Alquicira Esquivel

Directora de tesis: Dra. Hanna Oktaba

México, D.F.

2000

631432



Universidad Nacional
Autónoma de México



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 esta 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.

Dedico este trabajo a mi familia:

Rafa y Tere

Rosi, Feli, Tere y Mari

Paola, Anita, Rafa y Cheli

Fermin . . .

Agradecimientos.

A la Dra. Hanna Oktaba por su tiempo y paciencia para dirigir este trabajo.

Al Dr. Pablo Barrera Sánchez le agradezco su apoyo incondicional durante mis estudios de maestría.

Al jurado, Dra. Hanna Oktaba, Dr. Pablo Barrera Sánchez, M. en C. Guadalupe Ibarguengoita González, M. en C. Gloria Quintanilla Osorio y Dr. José de Jesús Cabrera Molina, por haber aceptado revisar este trabajo.

Al Dr. Jesús López Estrada por las facilidades y el equipo de cómputo, que me proporcionó a través del Laboratorio de Posgrado del Departamento de Matemáticas de la Facultad de Ciencias, para mis estudios de maestría y la realización de esta tesis.

Al Consejo Nacional de Ciencia y Tecnología por haberme otorgado una beca para realizar mis estudios de maestría.

A la Dirección General de Estudios de Posgrado por haberme otorgado una beca complementaria para realizar mis estudios de maestría.

Indice

Introducción	1
1. Estructura del Modelo de Madurez de Capacidad (CMM)	5
1.1. Historia	6
1.2. Representación gráfica de la estructura general del CMM	6
1.3. Niveles de madurez	7
1.4. Áreas clave de procesos (ACP)	9
1.4.1. Plantilla de las áreas clave de procesos	10
1.5. Prácticas clave	14
2. Nivel 2 del Modelo de Madurez de Capacidad (CMM): Repetible	15
2.1. Áreas clave de procesos del nivel 2	16
2.2. Proceso general para el modelo gráfico	17
2.2.1. Notación adicional	18
2.3. Modelo gráfico del nivel 2	20
3. Administración de requisitos	23
3.1. Modelo textual de la administración de requisitos (AdR)	24
3.2. Modelo gráfico de la administración de requisitos (AdR)	30
3.2.1. Identificación de los grupos participantes	30
3.2.2. Diagramas de actividades	31
4. Planeación del proyecto de software	39
4.1. Modelo textual de la planeación del proyecto de software (PPS)	40
4.2. Modelo gráfico de la planeación del proyecto de software (PPS)	52
4.2.1. Identificación de los grupos participantes	52
4.2.2. Diagramas de actividades	53
5. Seguimiento y control del proyecto de software	65
5.1. Modelo textual del seguimiento y control del proyecto de software (SCPS)	66
5.2. Modelo gráfico del seguimiento y control del proyecto de software (SCPS)	75
5.2.1. Identificación de los grupos participantes	75
5.2.2. Diagramas de actividades	76
6. Aseguramiento de calidad de software	87
6.1. Modelo textual del aseguramiento de calidad de software (AC)	88
6.2. Modelo gráfico del aseguramiento de calidad de software (AC)	95
6.2.1. Identificación de los grupos participantes	95
6.2.2. Diagramas de actividades	96

7. Administración de configuraciones de software	105
7.1. Modelo textual de la administración de configuraciones (ACS)	106
7.2. Modelo gráfico de la administración de configuraciones (ACS)	115
7.2.1. Identificación de los grupos participantes	115
7.2.2. Diagramas de actividades	116
8. Conclusiones	125
Anexos.	
Anexo I. Diagramas de actividades	127
Anexo II. Cuestionarios de apoyo para las áreas clave de procesos	139
Referencias	213

Introducción

Las presiones del mercado y las necesidades internas están empujando a las empresas que desarrollen software hacia la búsqueda de mejoras en sus procesos. Para apoyar este esfuerzo en los últimos años se han creado varios modelos que tratan de recaudar las mejores prácticas en la administración y el desarrollo de los proyectos de software. Los más importantes de estos modelos son: el CMM [CMM, 1994], generado en los EU, y el resultante del proyecto SPICE [El Emam et al., 1998], fruto del esfuerzo internacional de la estandarización ISO. 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 carecen de ellas.

Los modelos mencionados están presentados en un texto estructurado como descripciones en el lenguaje natural (inglés), sin embargo carecen de una representación gráfica que los sintetice y que facilite su comprensión. Esto cobra mayor importancia dado el volumen de la información que se transmite en cada uno de los modelos. Por ejemplo, la descripción del CMM en [CMM, 1994] ocupa 220 páginas. Asimilar e interpretar esta cantidad de información es un proceso lento y difícil.

Por otro lado, el modelado orientado a objetos y el lenguaje unificado de modelado UML [Rumbaugh et al., 1999] se están popularizando como el lenguaje común de los que desarrollan software. En particular el UML ofrece una gama muy interesante de notaciones gráficas para expresar diferentes aspectos de lo que se modela. ¿Porqué no explorar esta riqueza gráfica para modelar procesos de software? ¿Porqué no presentar a las mejores prácticas de la administración y el desarrollo de software en el lenguaje que los profesionistas de esta área manejan? Estas preguntas llevaron a dos casos de estudio de utilizar modelado con objetos y UML. El primero consistió en modelar de manera general el proceso de software [Oktaba and Ibarguengoitia, 1998] y el segundo sirvió para documentar en forma de patrones la recolección de datos para las métricas básicas de software [Oktaba and Ibarguengoitia, 1999].

En este trabajo utilizaremos los diagramas de actividades de UML [Rumbaugh et al., 1999] para modelar gráficamente las áreas clave de proceso del CMM nivel 2, excepto la Administración de Subcontratistas. Los diagramas de actividades, según las sugerencias de [Booch et al., 1999], pueden ser aplicados al modelado de flujo de trabajo en procesos de negocio. La forma concisa de reflejar el flujo de control e información entre las actividades, el paralelismo y los puntos de sincronización, así como la posibilidad de introducir explícitamente a los actores responsables por la realización de las actividades nos sugirieron que estos diagramas también pueden ser útiles para el modelado de los procesos de software.

El objetivo principal de usar el modelado gráfico para los procesos de software es encontrar la manera de abstraer y sintetizar los elementos cruciales de un proceso en un diagrama. Los diagramas de actividades nos ofrecen una vista que refleja a la vez la estructura y el comportamiento dinámico del proceso al igual que los grupos de actores involucrados en él.

El objetivo de esta tesis es construir un modelo gráfico, usando diagramas de actividades de UML, de las áreas clave de procesos del CMM nivel 2. El modelo gráfico está pensado como complemento del modelo textual para facilitar su comprensión de manera gradual e integrada. Por lo que el modelo textual en conjunto con el gráfico pueden ser utilizados como material de apoyo en la capacitación de empresas dedicadas al software interesadas en mejorar sus procesos a nivel 2 de CMM

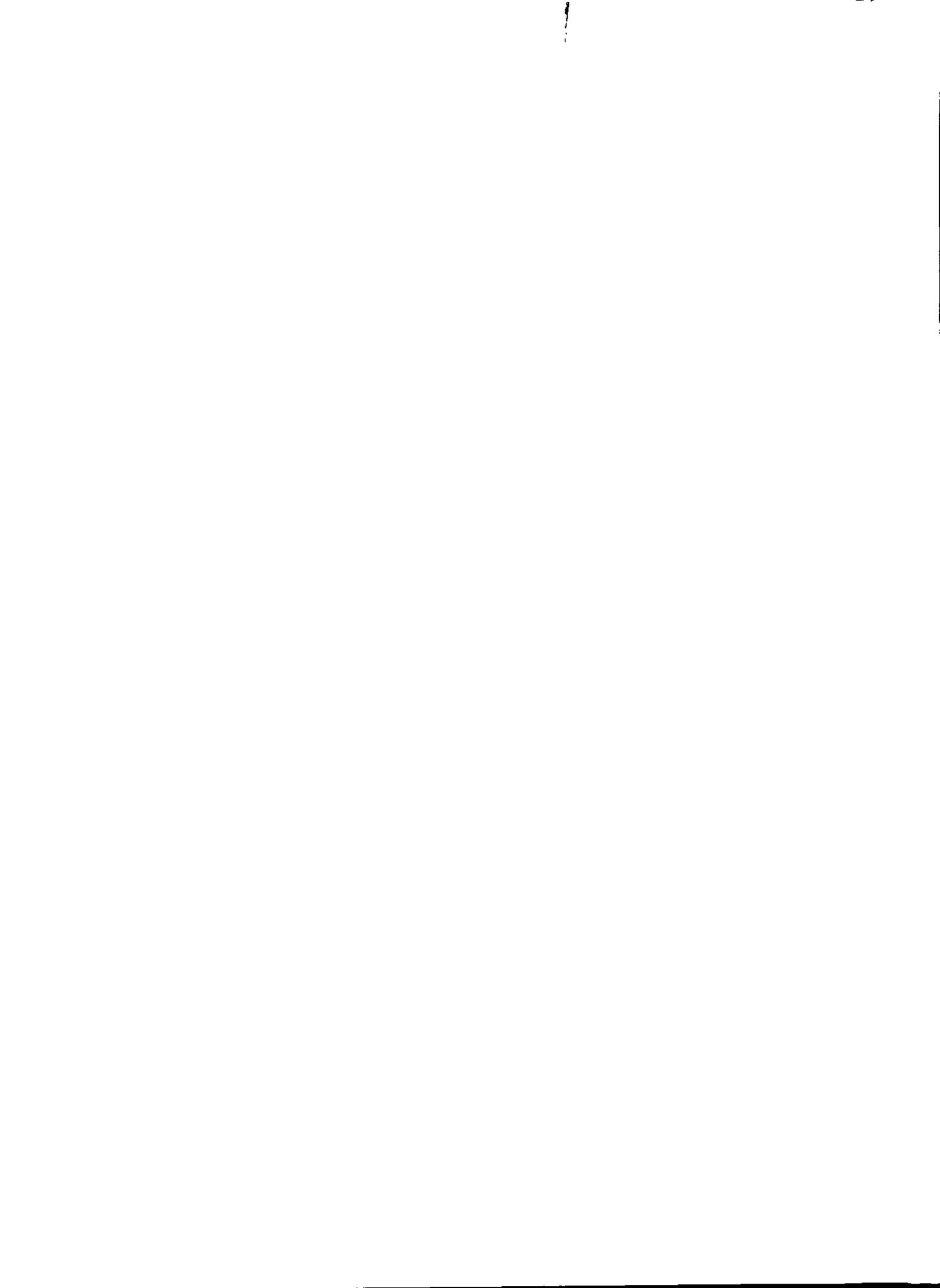
En el primer capítulo se presenta la estructura general del modelo de CMM (Modelo de Madurez de Capacidad), para familiarizar al lector tanto de la estructura como de los términos elementales de este modelo. Lo que permitirá proporcionar un ambiente adecuado para comprender de forma natural las áreas clave de procesos del CMM, que se modelarán gráficamente.

El capítulo dos se enfoca en el nivel 2 de CMM. En primer instancia se describen brevemente las áreas clave de procesos de este nivel. Posteriormente se presenta el proceso de modelado gráfico general que se sigue para construir el modelo gráfico de cada una de las áreas clave. Finalmente se presenta la descripción del modelo gráfico general del nivel 2.

Los capítulos del tres al siete están dedicados a las áreas clave de procesos de Administración de Requisitos, Planeación del Proyecto de Software, Seguimiento y Control del Proyecto de Software, Aseguramiento de Calidad de Software y Administración de Configuraciones de Software respectivamente. Estos capítulos se desarrollan con la misma estructura. Para apreciar la correspondencia entre el modelo textual y el gráfico, primero se presenta la traducción del área clave de procesos y después se describe el modelo gráfico de la área clave correspondiente.

En el capítulo ocho se presentan las conclusiones obtenidas durante el proceso de realización de los capítulos anteriores.

Finalmente se concluye con dos anexos, en el primero se describen brevemente los diagramas de actividades, desde un punto de vista práctico. En el segundo se incluyen cinco cuestionarios, que complementan a los capítulos del 3 al 7 respectivamente, en caso que el contenido de estos capítulos sea utilizado como material de capacitación.



Capítulo 1

Estructura del Modelo de Madurez de Capacidad (CMM).

El CMM para Software es un modelo que describe los elementos clave de un proceso de software eficaz. Este modelo propone, a las organizaciones dedicadas al software, una guía de buenas prácticas para mejorar sus procesos de software en forma continua, evolutiva y ascendente. Podemos decir que CMM es un modelo abstracto, en el sentido que no especifica en forma explícita “cómo” se implementa un proceso de software en una organización, sino que sólo establece “qué” debe de cumplir este proceso.

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

1. Los equipos de análisis para identificar fortalezas y debilidades en la organización.
2. Los equipos de evaluación para identificar los riesgos de seleccionar, entre los diferentes contratistas, para otorgar un negocio y para supervisar contratos.
3. La administración de niveles superiores para entender que actividades son necesarias para lanzar un programa de mejoramiento de proceso de software en su organización.
4. El personal técnico y grupos de mejoramiento de proceso como una guía para ayudarlos a definir y mejorar el proceso de software en su organización.

En este capítulo, primero presentaremos una breve historia del CMM y posteriormente describiremos su estructura, porque consideramos que la estructura de este modelo forma parte importante en el desarrollo de los siguientes capítulos.

1.1. Historia

En noviembre de 1986, el Instituto en Ingeniería de Software (SEI) y la Corporación MITRE empezaron el desarrollo de un esquema de proceso de madurez, que ayudaría a las organizaciones a mejorar sus procesos de software. Este esfuerzo se inicio porque el gobierno federal de los Estados Unidos, les solicitó un método para evaluar la capacidad de sus contratistas de software.

En septiembre de 1987, el SEI publicó una descripción breve del esquema de proceso de madurez de software [Humphrey, 1987], que posteriormente fue ampliado en el libro de Humphrey llamado "Administrando el Proceso del Software" [Humphrey, 1989]. Al mismo tiempo se desarrollaron dos métodos y una encuesta para evaluar la madurez del proceso de software.

Con la experiencia adquirida en el esquema de proceso de madurez, el SEI desarrolló el Modelo de Madurez de Capacidad para software (CMM). La versión inicial del CMM fue revisada y usada por la comunidad del software durante 1991 y 1992 y la versión 1.1 de CMM fue liberada en noviembre de 1993. Esta última versión es en la que basaremos nuestro trabajo.

1.2. Representación gráfica de la estructura general del CMM

El CMM está estructurado en cinco niveles de madurez y cada uno de estos niveles, a excepción del primer nivel, está constituido por un resumen y por áreas clave de procesos. A su vez cada una de estas áreas está integrada por un contexto, metas y practicas clave. Finalmente las prácticas clave están organizadas en Compromisos, Habilidades, Actividades, Mediciones y análisis y Verificaciones. Esta estructura del CMM la podemos modelar con el diagrama de clases presentado en la figura 1.1. En este diagrama las palabras "está constituido", "está integrada", "están organizadas" son representadas con el símbolo de agregación¹, el cual es una línea dirigida con punta en forma de rombo.

¹ El símbolo de agregación forma parte de los diagramas de clases. Modela la relación "Todo/Parte" en donde la clase, señalada por el rombo, representa una cosa más grande (todo) la cual esta constituida de cosas pequeñas (partes). Estas partes están representadas por la clase que está en el otro extremo de la línea dirigida.

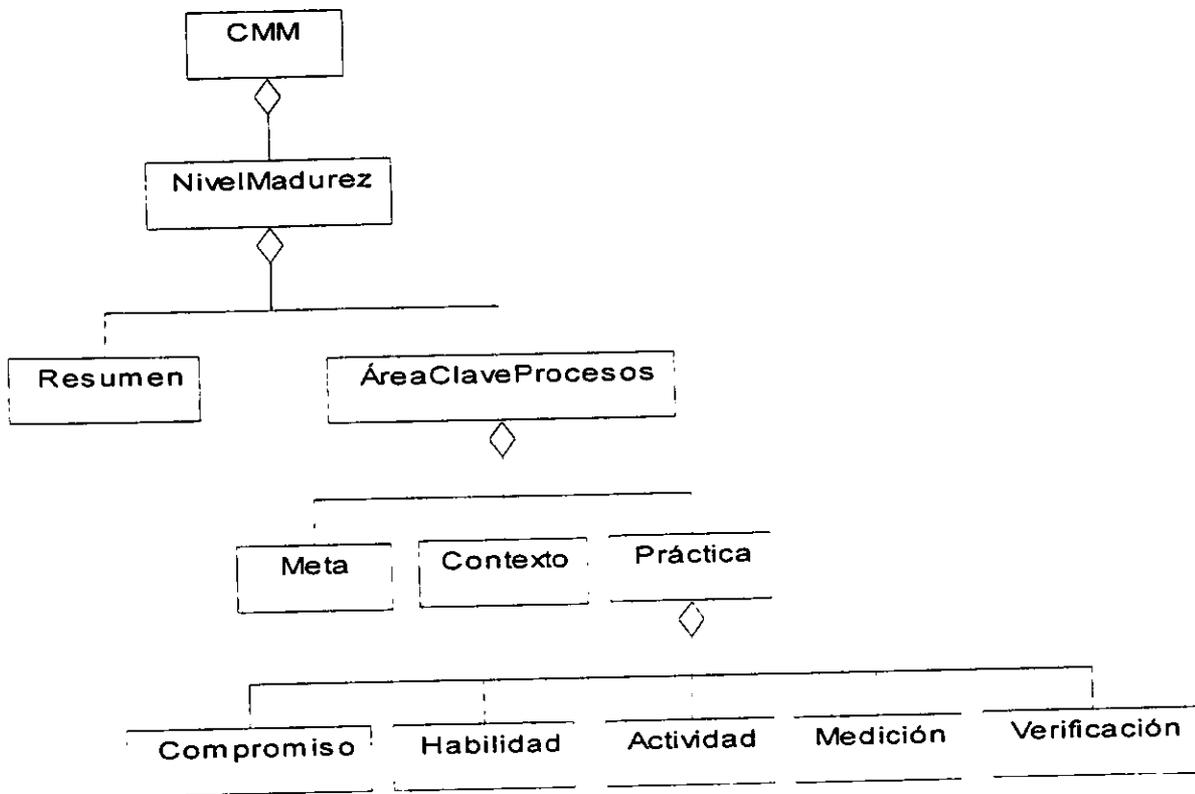


Fig. 1.1: Representación gráfica de la estructura de CMM con un diagrama de clases

En las siguientes secciones, describiremos a los niveles de madurez, áreas clave de procesos y prácticas clave, en forma más amplia.

1.3. Niveles de madurez

Como mencionamos en sección anterior, el CMM está estructurado en cinco niveles de madurez, los cuales deben ser alcanzados, por una organización, en forma ascendente. Estos niveles son:

- Inicial
- Repetible
- Definido
- Administrado
- Optimizando

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 el cimiento de los superiores. Por ejemplo, este modelo recomienda que para llegar al nivel tres, la organización debió obtener el nivel dos anteriormente. Por esta característica de los niveles, podemos representar al CMM como una escalera (vista frontal), donde cada uno de sus escalones representará un nivel de madurez (Fig. 1.2).

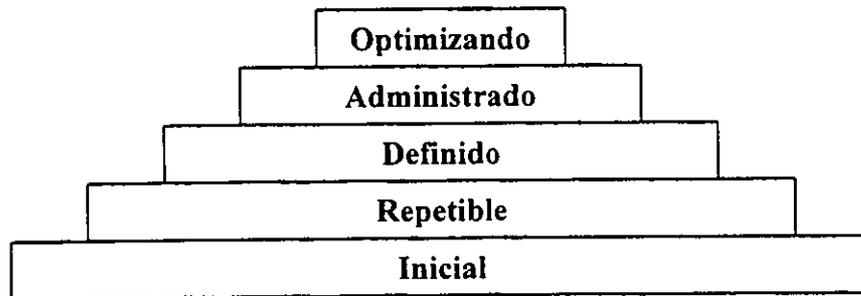


Fig. 1.2: Niveles de madurez

Cada nivel de madurez indica un nivel de capacidad en el proceso de software. Esta capacidad en el proceso provee un medio de predecir la mayoría de los resultados del próximo proyecto que la organización emprenda. A continuación presentaremos una breve descripción de cada uno de los niveles de madurez.

Nivel 1. Inicial

El proceso del software se caracteriza como 'ad hoc' y ocasionalmente caótico. Pocos procesos son definidos, y el éxito de los proyectos depende del esfuerzo individual y heroico.

Nivel 2. Repetible

Los procesos básicos de administración del proyecto están establecidos para seguir costo, calendario, y funcionalidad. La disciplina necesaria del proceso es repetir éxitos recientes en proyectos con aplicaciones similares.

Nivel 3. Definido

El proceso del software para la administración y actividades de diseño es documentado, estandarizado e integrado en un proceso del 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.

Nivel 4. Administrado

Las medidas detalladas del proceso del software y calidad del producto son reunidas. Tanto el proceso del software como los productos son cuantitativamente entendidos y controlados.

Nivel 5. Optimizando

El mejoramiento continuo del proceso es habilitado por retroalimentación cuantitativa del proceso y de las ideas innovadoras y tecnologías.

1.4. Áreas clave de procesos (ACP)

Las áreas clave de procesos indican a una organización, donde debe 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 medio, para alcanzar estas metas, podría variar entre proyectos, pero todas deben ser obtenidas para considerar que la organización ha implantado un área clave de procesos.

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

- **Inicial**
- **Repetible**
 - Administración de requisitos
 - Planeación del proyecto de software
 - Seguimiento y control del proyecto de software
 - Administración de subcontratistas de software
 - Aseguramiento de calidad de software
 - Administración de configuraciones de software
- **Definido**
 - Enfoque del proceso de organización
 - Definición del proceso de organización
 - Programa de capacitación
 - Administración integral de software
 - Ingeniería de producto de software
 - Coordinación entre grupos
 - Revisiones por pares
- **Administrado**
 - Administración cuantitativa de procesos
 - Administración de calidad de software
- **Optimizando**
 - Prevención de defectos
 - Administración de cambio de tecnología
 - Administración de cambio de proceso

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

- Habilidad 2 **Se proveen los recursos y financiamiento adecuados para <X>.**
1. <Subprácticas para la Habilidad 2...>
 2. Las herramientas de soporte para <X> se hacen disponibles.

Ejemplos de herramientas de <X>:
<ejemplos de herramientas>

- Habilidad 3 **<Roles> se capacitan <para realizar sus actividades de X>.**
- o
- <Roles> reciben la capacitación necesaria <para realizar sus actividades de X>.**

Ejemplos de capacitación:
<ejemplos de capacitación>

- Habilidad 4 **<Roles> reciben orientación en <X>.**

Ejemplos de orientación:
<ejemplos de orientación>

Actividades

- Actividad 1 **< Actividad realizada en el Área Clave de Procesos X>.**
1. <Subpráctica para la Actividad 1, posiblemente involucra a diferentes grupos>

Ejemplos de grupos involucrados:
<lista de grupos involucrados>

2. <Subprácticas adicionales para la Actividad 1 ...>
3. <Los productos de trabajo de software, en caso de ser conveniente,> son puestos bajo la administración de configuraciones.

Referirse al área clave de proceso de Administración de Configuraciones de Software.

- Actividad 2 **< Actividad realizada en el Área Clave de Procesos X> de acuerdo a un procedimiento documentado.**

Este procedimiento típicamente especifica que:

1. <Subprácticas para la Actividad 2, posiblemente con referencia(s) cruzadas a práctica(s) clave de otra área clave de proceso.>

Referirse a la Actividad N del área clave de proceso <Z> para practicas <relacionadas a la actividad 2.1>

2. <Subprácticas adicionales para la Actividad 2 ...>
3. <Los productos de trabajo de software> se someten a una revisión entre colegas <de acuerdo a criterios adecuados>.

Referirse al área clave de proceso de Revisiones entre Colegas.

4. Se administra y se controla <los productos de trabajo de software, en caso de ser conveniente>.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos Administración de Configuraciones de Software.

...

Mediciones y Análisis

Medición 1 Las mediciones se definen y se usan para determinar el estado de las actividades de <X>.

Ejemplos de mediciones:
<ejemplos de mediciones>

Verificaciones.

Verificación 1 Las actividades para <X> se revisan periódicamente con el administrador gerencial.

1. <Subprácticas para la Verificación 1 ...>

Verificación 2 Las actividades para <X> se revisan con el administrador del proyecto periódicamente y/o a causa de un evento.

1. <Subprácticas para la Verificación 2 ...>

Verificación 3 El grupo de aseguramiento de calidad de software revisa y/o audita las actividades y productos de trabajo para <X> y reporta los resultados.

Referirse al área clave de procesos de Aseguramiento de Calidad de Software

Al menos, estas revisiones y/o auditorias verifican que:

1. <Subprácticas para la Verificación 3 ...>

En algunas ACP, esta plantilla no puede ser utilizada en forma íntegra. Por ejemplo en Aseguramiento de Calidad de Software³, en la Verificación 3, el grupo de aseguramiento de calidad no puede auditarse así mismo, por lo que en este caso el texto de esta verificación cambia de acuerdo al contexto de esta ACP.

Si analizamos la plantilla de las ACP, podemos observar los siguientes puntos:

- Inicia con el nombre de la ACP en un tamaño de letra mayor que el resto del texto y en negritas. Le sigue un renglón escrito también en negritas, en cual se especifica a que nivel de madurez pertenece.
- Los párrafos, que siguen después del título, especifican el objetivo del ACP así como resumen de los aspectos relevantes del área y en algunos casos incluyen conceptos. A este conjunto de párrafos le llamaremos Contexto.
- El texto, que se localiza entre el contexto y el título de Compromisos, son las Metas.
- El texto restante son la prácticas clave, que están organizadas por sus características comunes, las cuales son: Compromisos, Habilidades, Actividades, Mediciones y análisis, y Verificaciones.
- Tanto la metas como las prácticas clave están delimitadas por títulos en negritas, que contienen el nombre de cada una de éstas. Estos títulos son: Metas, Compromisos, Habilidades, Actividades, Mediciones y análisis, y Verificaciones.
- Tanto las metas como las prácticas clave se identifican por medio de su nombre y un número de secuencia. Por ejemplo si existen tres metas, cada una de estas será etiquetada por Meta 1, Meta 2 y Meta 3.
- El texto de las metas y las prácticas clave está escrito en letras negritas y se localiza a continuación de sus etiquetas.
- El texto de las prácticas clave puede ser seguido por las subprácticas, las cuales están enumeradas, y escritas con el mismo tipo y tamaño de letra pero no en negritas. Generalmente las subprácticas inician con un preámbulo, por ejemplo, en los compromisos que se refieren a la existencia de una política las subprácticas inician "Esta política típicamente especifica que:".
- Algunas veces, después de las prácticas clave o entre el texto de las subprácticas podrían localizarse textos enmarcados, escritos en letra de menor tamaño. Estos textos pueden contener conceptos, referencias cruzadas o ejemplos.

³ Área clave de procesos del nivel 2.

1.5. Prácticas clave

El propósito de las prácticas clave es transmitir principios que se pueden aplicar a una gran variedad de proyectos y organizaciones. Por lo que CMM describe estos principios y la organización los adopta y adecua de acuerdo a su cultura y experiencia de sus administradores y personal técnico. Es decir, CMM establece “que” debe hacer una organización, pero deja que esta proponga “como” lo llevará a cabo.

Cada práctica clave consiste en un enunciado simple, generalmente es seguido por una descripción más detallada, la cual se conoce como subpráctica, y en algunos casos también es seguido por ejemplos y referencias cruzadas. Estas prácticas clave describen la infraestructura y las actividades que contribuyen a una implementación e institucionalización de una ACP. Las subprácticas pueden usarse como una guía para interpretar a la práctica clave en forma adecuada.

Como ya hemos mencionado, las prácticas clave, que describen al área clave de procesos, están organizadas por características comunes. Estas características comunes son:

- **Compromisos:** Describen las acciones que la organización debe tomar para garantizar que el proceso sea establecido y duradero. En los compromisos generalmente se establecen políticas organizacionales y liderazgos.
- **Habilidades:** Describen las precondiciones que deben de existir en el proyecto u organización, para implementar el proceso de software en forma competente. Las habilidades involucran típicamente el establecimiento de liderazgos, de recursos y de estructuras organizacionales, así como de capacitación.
- **Actividades:** Describen las actividades, roles y procedimientos necesarios para implementar un ACP. Las actividades involucra establecer planes y procedimientos, realizar el trabajo, dar seguimiento y tomar acciones correctivas en caso necesario.
- **Mediciones y análisis:** Especifican que se deben definir las mediciones básicas necesarias para determinar el estado del proceso. Estas mediciones se usarán para controlar y mejorar el proceso. Esta práctica clave generalmente tiene asociada ejemplos de mediciones que podrían ser consideradas.
- **Verificaciones:** Describen los pasos para garantizar que las actividades se realicen de acuerdo al proceso establecido. Las verificaciones se realizan por medio de revisiones y auditorías hechas por la administración y grupo de aseguramiento de calidad.

Las prácticas clave agrupadas en Actividades describen que debe ser implementado para cubrir un área clave de procesos, mientras que las otras prácticas forman la base para que esto pueda ser posible.

Capítulo 2

Nivel 2 del Modelo de Madurez de Capacidad (CMM): Repetible

Durante la implantación del nivel 2 de CMM, la organización debe establecer políticas para administrar el proyecto de software. Una vez que la organización haya logrado este nivel, debe de ser capaz de planear y administrar sus proyectos apoyándose en la experiencia de proyectos exitosos y similares. Por lo que los compromisos se aceptan de acuerdo a los resultados observados en proyectos previos. La administración del proyecto da seguimiento a costos calendario y funcionalidad, e identifica los problemas en el momento en que surgen, los cuales se tratan de resolver dentro del proyecto, pero cuando no es posible se pasan a los niveles altos de la administración.

En este capítulo, presentaremos una descripción de las áreas clave de procesos del nivel 2¹, puesto que el objetivo de esta tesis es proporcionar un modelo gráfico de estas áreas clave. En segundo lugar explicaremos el proceso del trabajo que seguimos para construir este modelo gráfico. Finalmente propondremos un diagrama de actividades que representa la relación entre áreas del nivel Repetible.

¹ Excepto el área clave de procesos de Administración de Subcontratistas de Software porque esta área clave no la modelaremos gráficamente, puesto que consideramos que esta área no es vital para las empresas o simplemente carecen de ella.

2.1. Áreas clave de procesos del nivel 2

A pesar que, en los siguientes capítulos trataremos a las áreas clave de procesos del nivel 2 en forma más profunda. En esta sección decidimos describir brevemente cada una ellas, porque nos darán los elementos necesarios para presentar, al final de este capítulo, el diagrama de actividades que modela la relación entre las áreas clave de este nivel.

Las áreas clave del nivel Repetible son:

- **Administración de requisitos (AdR).** Esta área clave establece un acuerdo común entre el cliente y el equipo de proyecto de software sobre los requisitos del cliente que serán abordados en el proyecto de software. Este acuerdo además de ser establecido debe ser vigente durante el ciclo de vida del proyecto. Por lo que será indispensable revisar los requisitos con el cliente, y al mismo tiempo generar la documentación necesaria sobre ellos y establecer el control de cambios.
- **Planeación del proyecto de software (PPS).** En esta área clave se establecen planes adecuados para realizar ingeniería de software y para administrar el proyecto de software.

La planeación inicia con una descripción de trabajo, las restricciones y las metas, que limitan y acotan el proyecto de software. En el proceso de la planeación de software se estima el tamaño de los productos de trabajo de software y los recursos necesarios para generar un calendario, identificar y evaluar los riesgos del software, y negociar los compromisos. Esta planeación culmina con la generación de un plan de desarrollo de software.

- **Seguimiento y control del proyecto de software.** Esta área proporciona una visión adecuada del avance real, de suerte que la administración pueda tomar acciones efectivas cuando el cumplimiento del proyecto de software se desvíe significativamente de los planes.

En esta área clave se plantea dar seguimiento y revisar los logros y resultados durante la realización del proyecto, para llevar a cabo esto se utilizará el plan de desarrollo de software. En caso de que existan desviaciones con respecto al plan se tendrán que tomar acciones correctivas. Estas acciones podrían ser entre otras, revisar los logros reales y replanear el resto del trabajo.

- **Aseguramiento de calidad de software.** Esta área clave provee a los administradores la visión adecuada del proceso usado por el proyecto de software y de los productos que se están construyendo.

Las prácticas clave de esta área sugieren realizar revisiones y auditorías a los productos y actividades de software, para verificar que se está cumpliendo con los estándares y procedimientos correspondientes.

- **Administración de Configuraciones de Software.** Esta área establece y mantiene la integridad de los productos de software durante el ciclo de vida de un proyecto de software.

Para conservar la integridad de los productos de trabajo se identifica la configuración de software (productos de trabajo seleccionados y sus descripciones) en fechas específicas, se controlan los cambios y se da seguimiento a la configuración.

2.2. Proceso general para el modelo gráfico

Para desarrollar el modelo gráfico de cada área clave de procesos aplicaremos la “ingeniería inversa” analizando el modelo textual para identificar elementos de diagramas de actividades de UML² tales como:

- Grupos de actores involucrados
- Actividades
- Objetos de información significativos
- Relaciones de flujo de control
- Relaciones de flujo de objetos de información

Los elementos identificados se plasmarán en diagramas de actividades³ de UML. Para facilitar el análisis dividimos el contenido del modelo textual de cada área clave en tres partes:

- **Contexto:** Son los primeros párrafos del modelo textual que definen el objetivo del ACP a tratar e introducen una serie de definiciones que facilitan la comprensión del resto del texto.
- **Metas.** Es la definición concisa de lo que se debe de lograr como resultado del proceso del área clave a tratar.
- **Prácticas.** Son los compromisos, habilidades, actividades, mediciones y su análisis así como verificaciones descritas como prácticas clave que una organización debe de llevar a cabo para lograr las metas del área clave.

²UML significa en Inglés *Unified Modeling Language*.

³Cabe notar que el término *actividad* en diagramas de UML tiene un sentido más amplio que en el modelo textual, y se refiere a una ejecución progresiva de cualquier tipo de acciones [Rumbaugh et al., 1999]. Una breve descripción de los diagramas de actividades se presenta en el Anexo 1.

Como se puede apreciar, el contexto ofrece la definición del problema que se está atacando y el vocabulario indispensable para su comprensión. Las metas son los objetivos específicos del proceso del ACP, las cuales, sirven de guía para el modelado del proceso y para su implantación en las organizaciones. Las prácticas constituyen la parte central del modelo. Son ellas que tienen que ser comprendidas y adoptadas por las organizaciones que quieren lograr las metas del ACP. Esta es la razón por la cual la parte de prácticas queremos enriquecer con un modelo gráfico, complementario al texto, para facilitar su comprensión y asimilación.

El modelado gráfico iniciará con una identificación, en todo el texto original, de los grupos de actores participantes en el proceso del ACP y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades. En la descripción de cada ACP se identifica y se resalta en letras *itálicas* los grupos participantes que son relevantes en el proceso.

Los capítulos del 3 al 7 básicamente los estructuramos en dos partes. En la primera presentamos la traducción del área clave a tratar, a la cual llamaremos modelo textual. En la segunda parte hacemos referencia al modelo gráfico, el cual iniciamos con una breve definición de cada uno de los grupos participantes identificados y continuamos con la explicación de los diagramas de actividades que modelan las prácticas clave del área correspondiente. Durante la explicación del modelo gráfico seguimos las siguientes convenciones:

- Los grupos participantes se escriben en letras *itálicas* y la primera letra en mayúscula.
- Los nombres de los objetos se escriben en letras *itálicas*.
- Los nombres de las actividades se escriben en letras *itálicas*, la primera letra en mayúscula y entre comillas.
- Si un diagrama de actividades, que está contenido en una figura, es el desglose de una actividad que pertenece a otro diagrama entonces el nombre de la figura es el mismo que el de la actividad desglosada.
- El primer número que identifica a una figura se refiere al capítulo al que pertenece y el segundo es el número secuencial de la figura.

2.2.1. Notación adicional

Para realizar el modelo gráfico utilizamos básicamente los diagramas de actividades, aunque en algunas ocasiones nos apoyaremos en los diagramas de clases. En el Anexo 1, describimos en forma compacta los diagramas de actividades y algunas modificaciones a estos, que nos parecieron adecuados para este trabajo. Además de los elementos explicados

en este Anexo I, incluimos en el modelo gráfico una notación⁴ especial para las actividades, que hace una referencia rápida entre las prácticas clave y las actividades. Esta notación la describimos en los siguientes puntos:

- La notación se localizará en la parte inferior derecha del rectángulo redondeado.
- Esta notación se escribirá en letras negritas.
- Denotaremos a las prácticas clave con una letra mayúscula y un subíndice. La letra mayúscula sólo podrá adquirir los valores de C, H, A, M y V, que representan a los Compromisos, Habilidades, Actividades, Mediciones y análisis, y Verificaciones respectivamente. El subíndice es un número natural, que indica el número de secuencia de las características comunes. Por ejemplo con A_{15} estamos denotando a la Actividad 15 de alguna área clave. Si suponemos que nos referimos a la Planeación del Proyecto de Software, esta actividad dice que se registran los datos de planeación de software, la cual modelamos como (ver Fig. 4.4):

Registrar datos PPS
 A_{15}

- Si una actividad representa a más de una práctica clave, se escribirá una lista de las prácticas involucradas en una renglón separadas por comas. En caso de ser prácticas consecutivas se denotarán poniendo en primer lugar la práctica inicial seguida de tres puntos y la practica final. Por ejemplo en la Figura 4.5 tenemos a “*Identificar elementos PPS*”, la cual representa a las Actividades 5 y de la 8 a la 14 de PPS.

Identificar elementos PPS
 $A_5, A_8 \dots A_{14}$

- Denotaremos con el símbolo matemático de pertenencia “ \in ”, para indicar que un conjunto de prácticas clave está contenidas en otra práctica clave. Por ejemplo en la figura 4.4, observamos:

Planear proyecto de software
 $A_4 \dots A_{14} \in A_2$

Esto quiere decir que la Actividad 2 se modela con “*Planear proyecto de software*”, y que además agrupa de la Actividad 4 a la 14.

Consideramos que esta notación será útil para obtener una referencia rápida entre prácticas clave y actividades en los modelos gráfico de cada una de las áreas clave.

⁴ Esta notación no la incluimos en el Anexo I, porque consideramos que es una notación que surge de las condiciones particulares del modelado de las prácticas clave, por lo que podría no ser aplicable en otros casos.

2.3. Modelo gráfico del nivel 2

En esta sección presentaremos el diagrama de actividades, que relaciona a las ACP del nivel 2 en forma gráfica (Fig. 2.1). En este diagrama modelaremos a cada una de las áreas clave con una actividad, las cuales identificaremos con su nombre abreviado⁵.

Una organización a nivel 2, realiza mejoras a los procesos del proyecto, por lo que consideramos que la Actividad "Solicitar producto" sea la primera en realizarse y esta a cargo del *Cliente*, pues para iniciar un proyecto alguien debió solicitarlo. Una vez solicitado un producto la *Organización* y el *Cliente* establecen un acuerdo sobre los requisitos del proyecto, es decir se debe llevar a cabo las prácticas propuestas en la Administración de Requisitos (AdR). Este hecho lo representamos con "AdR", de esta actividad obtenemos los requisitos del proyecto, los cuales representamos con el objeto de salida *r:Requisito*.

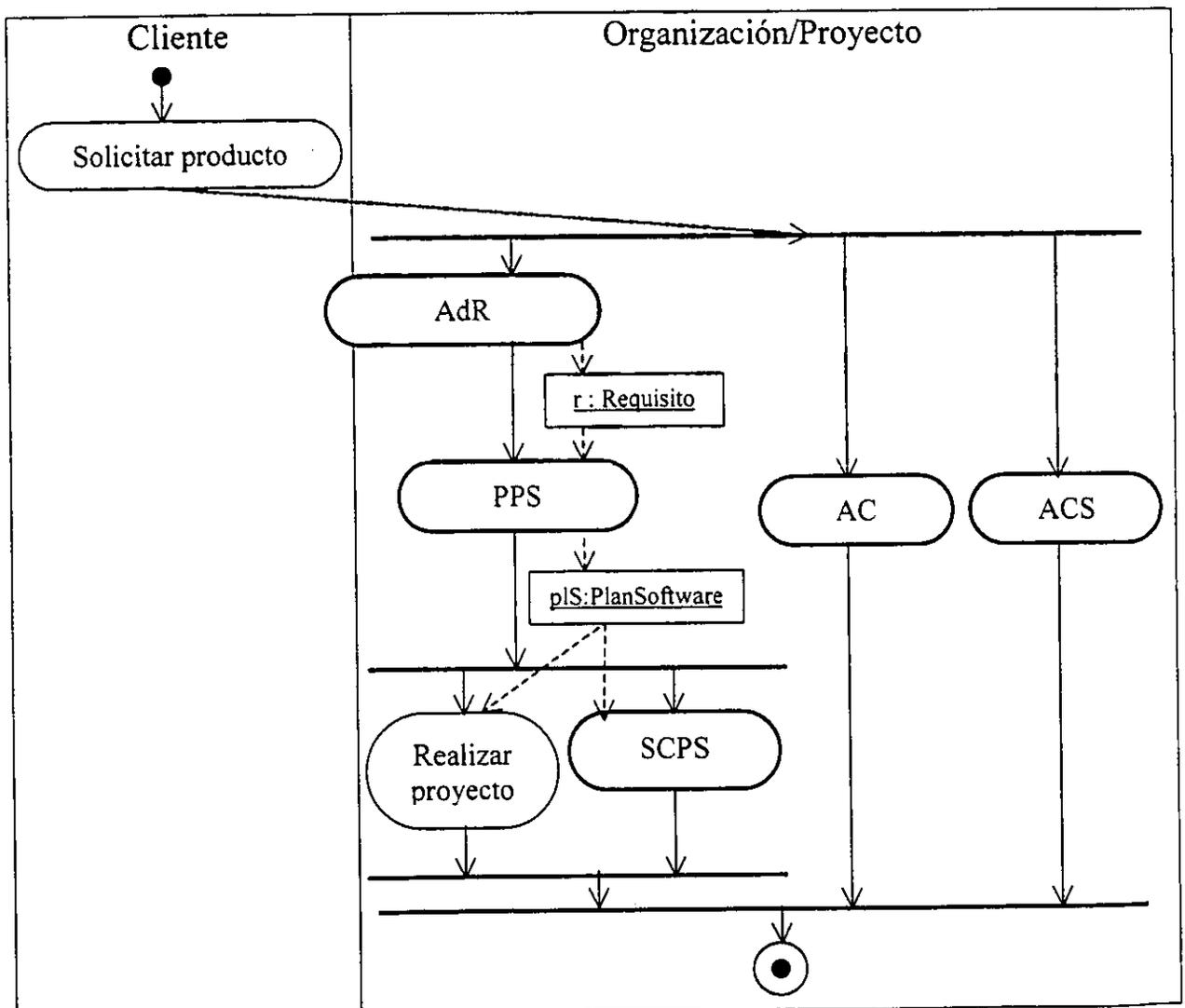


Fig. 2.1: Modelo gráfico general del nivel 2.

⁵ Los nombres abreviados de las áreas clave de procesos los puede consultar en la sección 2.1.

Los requisitos del proyecto son la base de la Planeación del Proyecto de Software (PPS), por lo que entra *r:Requisito* a la actividad "PPS". Como sabemos de la planeación obtenemos el plan de desarrollo de software. Este plan es utilizado por el Seguimiento y Control del Proyecto de Software (SCPS) y para realizar el proyecto de Software. Lo que en el diagrama modelamos con las actividades "Realizar proyecto" y "SCPS" que se llevan a cabo en forma simultánea y con el flujo del objeto *plS:PlanSoftware* que va de "PPS" a "Realizar proyecto" y "SCPS".

La Administración de Configuraciones de Software (ACS) mantiene la integridad de los productos de software durante el ciclo de vida de un proyecto. Mientras que el Aseguramiento de Calidad de Software (AC) revisa y audita productos y actividades de software. Estos productos y actividades son producidos durante la implantación de la administración de requisitos, la planeación, realización y seguimiento del proyecto. Por lo que la trayectoria constituida por "AdR", "PPS", "SCPS" y "Realizar proyecto" se lleva a cabo en forma simultánea con las actividades "AC" y "ACS".

Los miembros de la organización asignadas al proyecto es el grupo responsable de las actividades que modelan a las ACP y a la realización del proyecto, a este grupo lo llamamos *Organización/proyecto*. La actividad "AdR" además de ser realizada por el proyecto también recae bajo la responsabilidad del *Cliente*, porque entre ambos acuerdan los requisitos del proyecto.

Las actividades con rectángulos redondeados más gruesos (fig. 2.1), están en el primer nivel de abstracción que queremos presentar. En los capítulos del 3 al 7 mostraremos el desglose de cada una de ellas.

Capítulo 3

Administración de requisitos

La administración de requisitos es la primer área clave de procesos del nivel 2 de CMM. En esta área se establece un acuerdo sobre los requisitos del proyecto entre el cliente y el grupo asignado al proyecto. Este acuerdo constituye la base para estimar, planear, desarrollar y dar seguimiento al proyecto durante su ciclo de vida. Por lo que es necesario documentar y establecer un control de cambio de este acuerdo.

En este capítulo presentaremos la traducción de esta área clave de procesos [CMM, 1994], posteriormente iniciamos la descripción del modelo gráfico con la presentación de los grupos participantes y la concluimos con la explicación de los diagramas de actividades, que constituyen el modelo gráfico.

3.1. Modelo textual de la Administración de Requisitos (AdR)

El propósito de la administración de requisitos es establecer un acuerdo común entre el *cliente* y el *equipo de proyecto de software* sobre los requisitos del cliente que serán abordados en el proyecto de software.

La administración de requisitos involucra establecer y mantener un acuerdo con el *cliente* sobre los requisitos para el proyecto de software. Este acuerdo es conocido como “requisitos del sistema asociados al software”. El “cliente” podría ser interpretado como el grupo de ingeniería del sistema, el grupo de mercadotecnia, otra organización interna o un cliente externo. El acuerdo cubre tanto requisitos técnicos como no técnicos (por ejemplo fecha de entrega). El acuerdo constituye la base para estimar, planear, desarrollar y dar seguimiento a las actividades del proyecto de software a lo largo del ciclo de vida del software.

La asignación de los requisitos del sistema al software, hardware y otros componentes del sistema (por ejemplo humanos) podrían ser realizados por un grupo externo al grupo de ingeniería de software (por ejemplo el grupo de ingeniería del sistema) y el grupo de ingeniería de software podría no tener el control directo de esta asignación. Dentro de las restricciones del proyecto, el *grupo de ingeniería de software* sigue los pasos apropiados para asegurar que los requisitos del sistema asociados al software sean documentados y controlados.

Para lograr este control, el *grupo de ingeniería de software* revisa los requisitos iniciales y revisados del sistema asociados al software para resolver problemas antes de que estos sean incorporados al proyecto de software. Cuando los requisitos del sistema asociados al software se cambian, los planes, productos y actividades de software afectados se adaptan para seguir siendo consistentes con los requisitos actualizados.

Metas

Meta 1

Controlar los requisitos del sistema asociados al software para establecer una línea base para ingeniería de software y para uso administrativo.

Meta 2

Mantener consistencia entre los planes, productos y actividades de software con los requisitos del sistema asociados al software.

Compromisos

Compromiso 1

El proyecto sigue una política organizacional escrita para administrar los requisitos del sistema asociados al software.

En estas practicas, se hace referencia a los requisitos del sistema asociados al software como "requisitos asociados".

Los requisitos asociados es el subconjunto de los requisitos del sistema que serán implementados en los componentes del sistema de software. Los requisitos asociados son una entrada primaria al plan de desarrollo del software. El análisis de requisitos de software elabora, refina y documenta los requisitos asociados y termina en los requisitos del software que son documentados.

Esta política especifica típicamente que:

1. Los requisitos asociados se documentan.
2. Los administradores del software y otros grupos involucrados revisan los requisitos asociados.

Ejemplos de grupos involucrados:

sistema de prueba,
ingeniería de software (incluyendo todos los subgrupos, por ejemplo diseño de software),
ingeniería de sistemas,
aseguramiento de calidad de software,
administración de configuraciones de software y
soporte de documentación.

3. Los planes, productos de trabajo y actividades de software se cambian para ser consistentes con los cambios a los requisitos asociados.

Habilidades

Habilidad 1

Para cada proyecto, se asigna la responsabilidad de analizar los requisitos del sistema que corresponden al hardware, software u otros componentes del sistema.

El análisis y asignación de los requisitos del sistema no es responsabilidad del grupo de ingeniería de software pero es prerequisite para su trabajo.

La responsabilidad cubre:

1. Administrar y documentar los requisitos del sistema y su asignación a lo largo de la vida del proyecto.
2. Efectuar cambios a los requisitos del sistema y su asignación.

Habilidad 2

Se documentan los requisitos asociados.

Los requisitos asociados incluyen:

1. Los requisitos no técnicos (es decir los acuerdos, condiciones, y/o los términos contractuales) que afectan y determinan las actividades del proyecto del software.

Ejemplos de acuerdos, condiciones y términos contractuales:
productos por entregar,
fechas de entrega, y
fechas y/o eventos predeterminados (*milestones*).

2. Los requisitos técnicos para el software.

Ejemplos de requisitos técnicos:
funcionalidades del usuario final, operador, soporte, o de integración;
requisitos de desempeño;
restricciones de diseño;
lenguaje de programación; y
requisitos de interfase.

3. Los criterios de aceptación que serán usados para validar que los productos del software satisfacen los requisitos asociados.

Habilidad 3

Se proveen los recursos y financiamiento adecuado para administrar los requisitos asociados.

1. La administración de los requisitos se asigna a individuos que tienen experiencia y especialización en el dominio de aplicación y en ingeniería de software.
2. Se proporcionan las herramientas para apoyar las actividades para administrar los requisitos.

Ejemplos de herramientas de apoyo:
hojas de cálculo,
herramientas para administración de configuraciones,
herramientas de seguimiento, y
herramientas para administración de documentos de texto.

Habilidad 4

Se capacita a los miembros del grupo de ingeniería de software y otros grupos relacionados al software para realizar sus actividades de administración de requisitos.

Ejemplos de capacitación:
los métodos, estándares, y procedimientos usados por el proyecto, y
el dominio de aplicación.

Actividades

Actividad 1

El **grupo de ingeniería de software** revisa los requisitos asociados antes de que ellos sean incorporados al proyecto de software.

1. Se identifican los requisitos asociados incompletos y omitidos.
2. Se revisan los requisitos asociados para determinar si ellos son:
 - factibles y apropiados para implementarlos en el software,
 - establecidos claramente y propiamente,
 - consistentes entre sí, y
 - verificables.
3. Cada requisito asociado, identificado como uno con problemas potenciales, se revisa con el responsable del grupo que analiza y asigna los requisitos del sistema y se hacen los cambios necesarios.
4. Los compromisos resultantes de los requisitos asociados se negocian con los grupos involucrados.

Ejemplos de grupos involucrados:
ingeniería de software (incluyendo todos los subgrupos, por ejemplo diseño de software),
estimación de software,
ingeniería de sistemas,
prueba de sistema,
aseguramiento de calidad de software,
administración de configuración de software,
administración de contrato, y
soporte de documentación.

Referirse a la Actividad 6 del área clave de procesos de Planeación del Proyecto de Software para cubrir la negociación de los compromisos.

Actividad 2

El **grupo de ingeniería de software** usa los requisitos asociados como la base para los planes, productos de trabajo y actividades del software.

Los requisitos asociados:

1. Se administran y se controlan.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos Administración de Configuraciones de Software.

2. Son la base para el plan de desarrollo de software.
3. Son la base para desarrollar los requisitos de software.

Actividad 3

Se revisan e incorporan los cambios a los requisitos asociados al proyecto del software.

1. Se valora el impacto a los compromisos existentes, y se negocian los cambios si proceden.
 - Los cambios a los compromisos hechos a individuos y grupos externos a la organización se revisan con el administrador gerencial.

Referirse a la Actividad 4 del área clave de procesos de Planeación del Proyecto de Software y la Actividad 3 de Seguimiento y Control del Proyecto de Software que cubren la aceptación de compromisos externos a la organización.

- Los cambios a los compromisos con la organización se negocian con los grupos involucrados.

Referirse a las Actividades 5, 6, 7 y 8 del área clave de procesos de Seguimiento y Control del Proyecto de Software para cubrir la negociación de los cambios a los compromisos.

2. Los cambios que necesitan ser hechos a los planes de software, productos de trabajo, y actividades resultantes de los cambios a los requisitos se:
 - identifican,
 - evalúan,
 - valoran con respecto al riesgo,
 - documentan,
 - planean,
 - comunican a los grupos e individuos involucrados, y
 - siguen hasta su conclusión.

Mediciones y análisis

Medición 1

Se definen, se recaudan y se usan mediciones para determinar el estado de las actividades de la administración de los requisitos asociados.

Ejemplos de mediciones:

- estado de cada uno de los requisitos asociados;
- actividad de cambio para los requisitos asociados; y
- número acumulado de cambios a los requisitos asociados incluyendo el número total de cambios propuestos, pendientes, aprobados e incorporados en línea base del sistema.

Verificaciones

Verificación 1

Las actividades para la administración de los requisitos asociados se revisan periódicamente con el administrador gerencial.

El principal propósito de las revisiones periódicas con el administrador gerencial es proveerle la información relevante sobre las actividades del proceso de software en un nivel de abstracción apropiado y a tiempo. La periodicidad de las revisiones tiene que ser acorde con las necesidades de la organización y ser sustituida por los mecanismos adecuados para reportar situaciones de excepción.

Referirse a la Verificación 1 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador gerencial.

Verificación 2

Las actividades para la administración de los requisitos asociados se revisan con el administrador del proyecto periódicamente y/o a causa de un evento.

Referirse a la Verificación 2 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador de proyecto.

Verificación 3

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

Referirse al área clave de procesos de Aseguramiento de Calidad de Software.

Estas revisiones y/o auditorias verifican al menos que:

1. Se revisen los requisitos asociados y se resuelvan los problemas antes de que sean entregados al grupo de ingeniería de software.
2. Los planes de software, productos de trabajo y actividades se revisen apropiadamente cuando los requisitos asociados cambian.
3. Los cambios a los compromisos resultantes de los cambios a los requisitos asociados sean negociados con los grupos involucrados.

3.2. Modelo gráfico de la Administración de Requisitos (AdR).

El modelado gráfico se iniciará con una identificación, en todo el texto original, de los grupos de actores participantes en el proceso de AdR y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

3.2.1. Identificación de los grupos participantes.

En el modelo textual de la administración de requisitos hemos identificado y resaltado en letras itálicas 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.

Cliente. Es el que define los requisitos para un producto de software, que él solicita a un proveedor.

Responsable de Analizar Requisitos (AnR). Es el responsable del análisis de requisitos del sistema que se asocian al hardware, software u otros componentes.

Grupo de Ingeniería de Software (IS). Revisa, documenta e incorpora los requisitos asociados a software. En el caso de cambio de requisitos modifica los planes, productos de trabajo y actividades para mantener la consistencia con los requisitos modificados.

Administrador¹. Crea las condiciones para el Análisis de Requisitos y revisa las actividades de la administración de los requisitos asociados a software.

Grupo de Aseguramiento de Calidad (AC). Revisa y/o audita las actividades y productos de trabajo para administrar los requisitos asociados a software.

Organización². Es el conjunto de los grupos mencionados anteriormente excepto Cliente. Estos grupos proporcionan el servicio de desarrollo de software para el cliente.

¹En el texto original se habla de dos niveles de administradores gerencial (*señior*) y de proyecto. Para fines de este capítulo ubicaremos ambos grupos bajo el mismo nombre de Administrador.

²En este grupo también están incluidos el grupo del proyecto de software y otros grupos relacionados al software.

3.2.2. Diagramas de actividades

Compromisos

La parte de prácticas del modelo textual inicia con la sección de Compromisos. Analizando el texto del único compromiso, se deduce que la organización tiene una política escrita que guíe la administración de requisitos de los proyectos. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Fig.3.1) con el objeto *p:Política*. Este objeto fue probablemente generado por una actividad de definición de política, la cuál, según el texto del compromiso, queda implícita y por lo tanto no aparece en el diagrama.

El compromiso de seguir la política de AdR en cada proyecto queda expresado colocando a *p:Política* como objeto de entrada de la actividad de “*Administrar requisitos*” (Fig. 3.1). Cabe resaltar, que “*Administrar requisitos*” se lleva a cabo en forma compartida entre el *Cliente* y la *Organización*. Esta actividad la modelamos de esta manera, porque la administración de requisitos involucra establecer y mantener un acuerdo con el cliente sobre los requisitos del proyecto de software³.

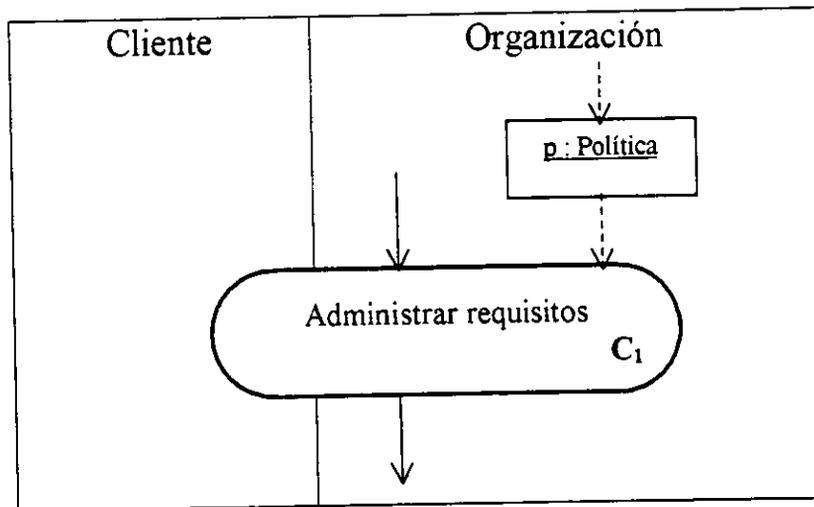


Fig. 3.1: Compromisos

La Fig.3.1 ofrece una vista global de las relaciones entre el *Cliente* y la *Organización* con respecto a la política de AdR y su implicación para los proyectos.

Las secciones que siguen: **Habilidades**, **Actividades**, **Mediciones y Análisis**, y **Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben de tomar para administrar requisitos en un proyecto. Para modelar este hecho, la actividad de “*Administrar requisitos*” de la Fig.3.1 se desglosa en el diagrama presentado en la Fig.3.2.

³ Argumento tomado del segundo párrafo del contexto del área clave de procesos de Administración de requisitos.

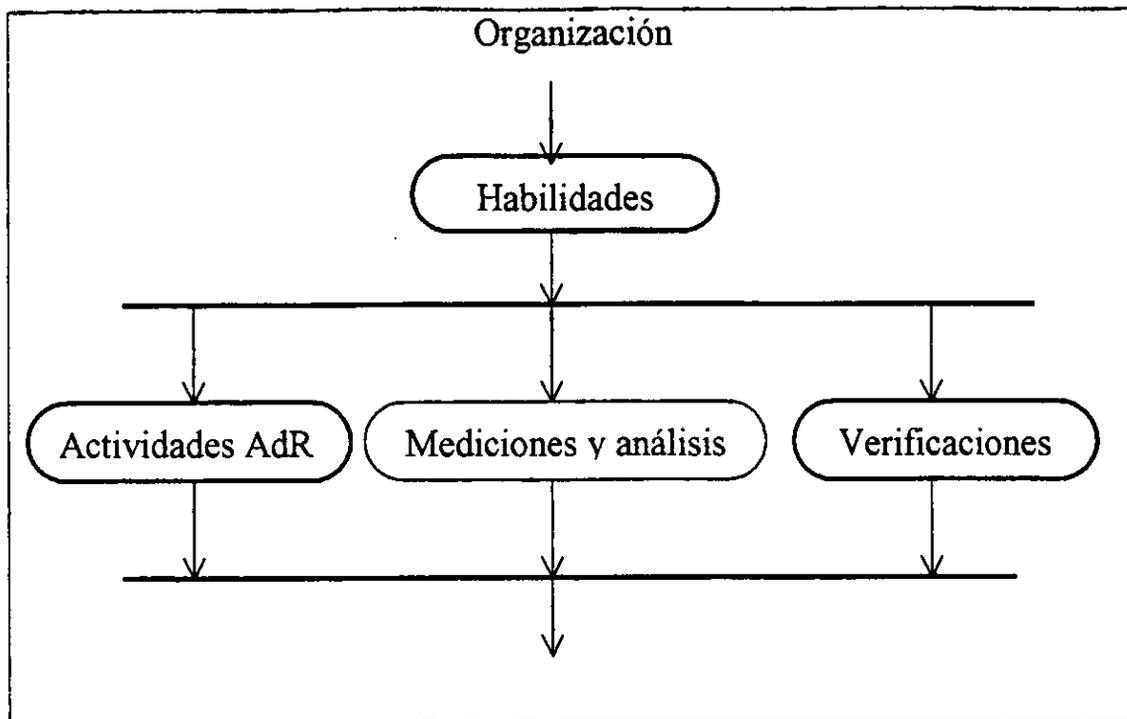


Fig. 3.2: Administrar requisitos

Este diagrama introduce una actividad por cada sección mencionada y, además, expresa el orden de ejecución y/o el paralelismo entre ellas. Queda claro, que las habilidades tienen que ser adquiridas antes de que sean realizadas las actividades propias de la administración de requisitos. Por otro lado, las actividades de verificación y medición, con su análisis, son simultáneas a la de AdR, y tienen la misma duración que el proyecto.

Habilidades

La habilidad 1 del modelo textual habla de la asignación de la responsabilidad de analizar los requisitos para cada proyecto. Aunque no es explícito quien hace la asignación, parece natural que sea alguien del nivel administrativo, y que esta responsabilidad recaerá en alguna persona, la cual llamaremos *Responsable de Analizar Requisitos (AnR)*.

La Fig.3.3 muestra la actividad de “*Asignar responsable*” como una de las primeras que tiene que realizar un *Administrador* antes de las demás actividades relacionadas con la adquisición de habilidades. La habilidad 3 habla de la provisión de recursos y financiamiento. Modelamos esta habilidad con la actividad “*Asignar recursos*”, también se puede interpretar que está a cargo del *Administrador*, y es la que constituye la base para el buen funcionamiento de la AdR. Ambas actividades, “*Asignar responsable*” y “*Asignar recursos*”, son actividades que se pueden realizar en forma simultánea.

El texto de la habilidad 2 menciona solamente que los requisitos son documentados. Aquí, otra vez se nos obliga a interpretar las intenciones implícitas del texto. Nos parece razonable presentar gráficamente la actividad de “Analizar y documentar requisitos”, compartida entre el *Cliente* y el *Responsable de AnR*, cuyo objetivo es generar una versión acordada del documento de requisitos. Finalmente, la cuarta habilidad habla de la capacitación en AdR del grupo de IS y otros grupos relacionados. Esta habilidad se presenta en la Fig. 3.3 como una actividad de “Capacitar en AdR” que puede realizarse en paralelo con la actividad “Analizar y documentar requisitos”. Para simplificar el diagrama omitimos otros grupos relacionados a los cuales también se podría aplicar la misma actividad. Cuando las cuatro actividades relacionadas con la adquisición de habilidades se terminan, incluyendo la generación del documento de requisitos, se pueden iniciar las actividades propias de la administración de requisitos para el proyecto.

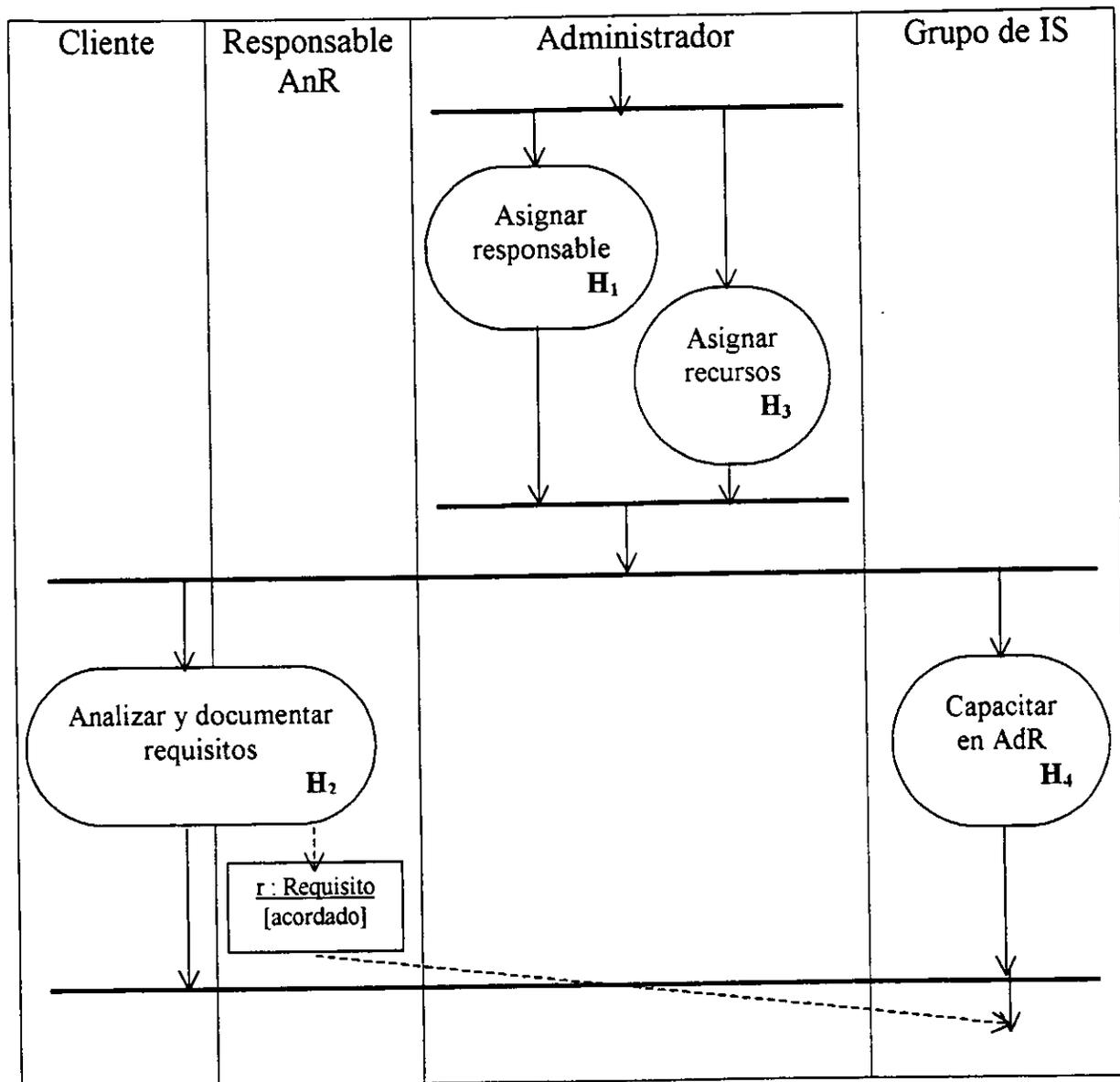


Fig. 3.3: Habilidades

El diagrama de la Fig. 3.3 identifica de manera clara la distribución de las actividades entre diversos grupos involucrados con respecto a la adquisición de las habilidades para la AdR; también, es fácil observar qué actividades pueden llevarse a cabo de manera paralela.

Actividades

Esta sección es integrada por tres actividades. Las dos primeras se refieren a la revisión y la incorporación de los requisitos realizadas por el *grupo de IS*. La actividad 1 se mapea directamente con la actividad de “Revisar requisitos” y la actividad 2 con la de “Incorporar requisitos”, lo podemos observar en la Fig. 3.5. Además el texto de la actividad 1 establece que los requisitos no pueden ser incorporados antes de que sean revisados por el *grupo de IS*.

Los requisitos acordados, que son el producto de la actividad de “Analizar y documentar requisitos” (Fig. 3.3), son el documento de entrada de “Revisar requisitos” y una vez revisados por el *grupo de IS* sirven de entrada a la actividad de “Incorporar requisitos”.

Según el modelo textual esta actividad usa los requisitos como base para los planes del software, productos de trabajo y actividades. En nuestro diagrama el resultado de esta incorporación lo representamos con el objeto *pr:Producto*. Para facilitar la representación gráfica, hemos creado a la clase *Producto* como una generalización de las subclases *PlanSoftware*, *ProductoTrabajo* y *ActividadSoftware* (Fig.3.4) para englobar el efecto de la incorporación de los requisitos en cualquiera de ellos.

La actividad 3 dice que los cambios a los requisitos son revisados e incorporados al proyecto de software. En nuestro diagrama (Fig. 3.5) esto se representa con las actividades “Revisar cambios” e “Incorporar cambios” del *grupo de IS*. Para que los cambios se revisen e incorporen alguien tuvo que solicitarlos. Como esta parte queda implícita en la formulación de la actividad 3, nos parece que quedará más claro si en la representación gráfica añadimos las actividades de “Solicitar cambios” y “Analizar y documentar cambios” los cuales generarán una nueva versión acordada de requisitos, la cual puede ser revisada e integrada al proyecto.

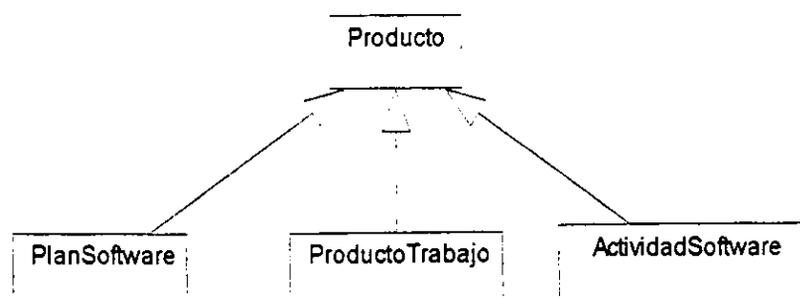


Fig. 3.4: Superclase *Producto*

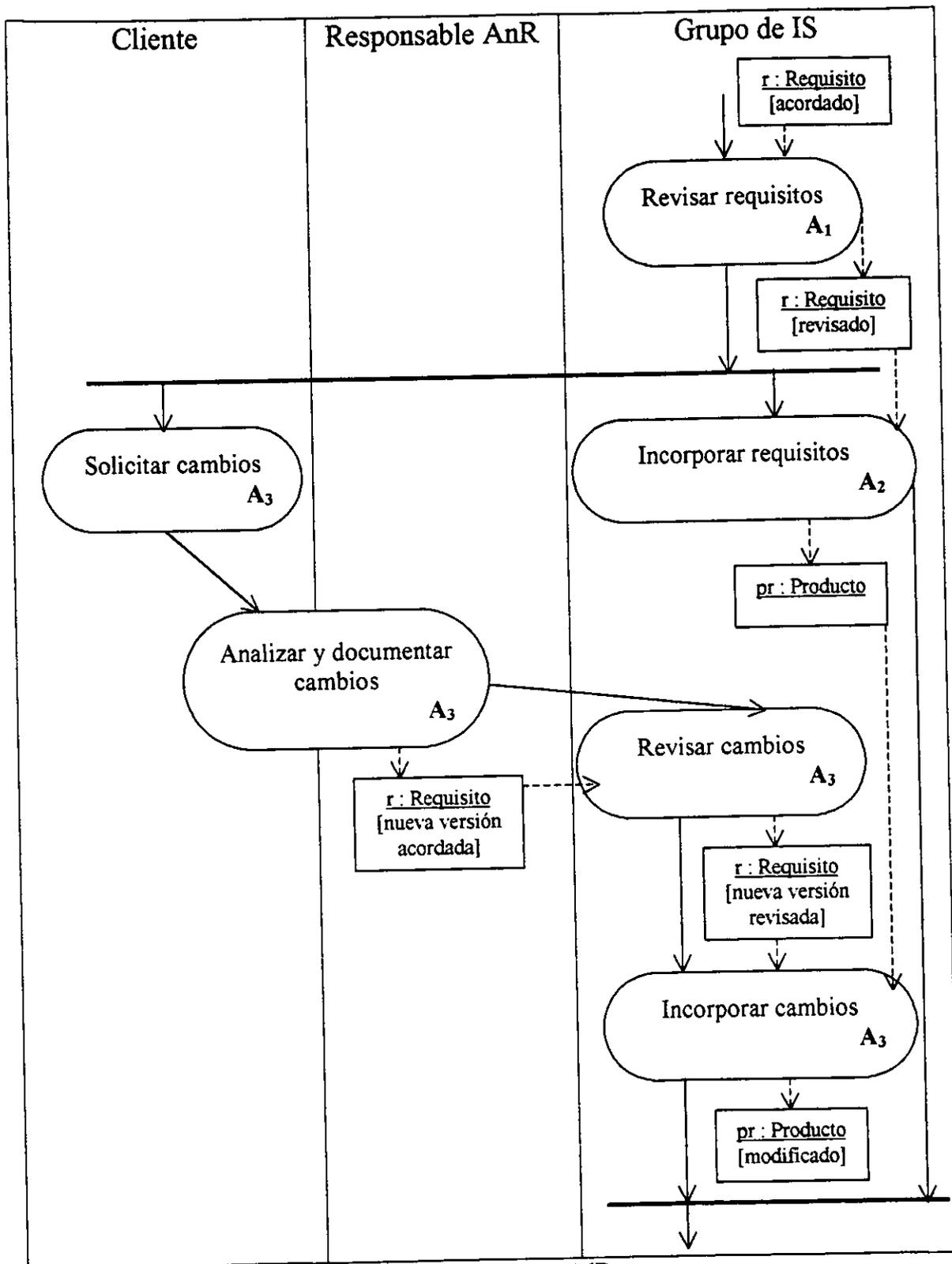


Fig. 3.5: Actividades AdR

El diagrama de la Fig. 3.5 pone en evidencia la importancia de los requisitos para la generación del producto, así como del control de cambios a los requisitos y su impacto en el proceso de producción.

Mediciones y análisis.

El texto correspondiente a la medición y análisis es muy escueto. Se dice que deben de recabarse y usar mediciones para determinar el estado de las actividades de la AdR. Sin embargo, falta precisar quienes son responsables por efectuar dichas actividades. Conociendo todo el contexto del modelo textual de AdR nos atrevemos a proponer (Fig.3.6) que la actividad de "Mediciones y análisis" es una actividad compartida entre el Grupo de IS, que debe ser responsable por recaudar datos de las mediciones, y el Administrador y el Grupo de Aseguramiento de Calidad (AC), quienes utilizan estos datos para determinar el estado de las actividades de la AdR. Previamente, en la Fig. 3.2, ya quedó plasmado el paralelismo de las actividades de medición y análisis con las de AdR y de las verificaciones.

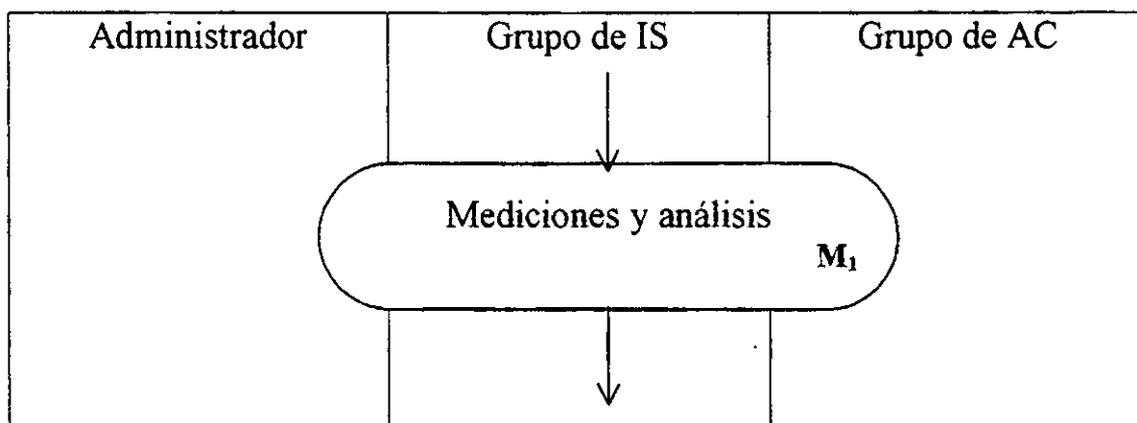


Fig. 3.6: Mediciones y análisis

El diagrama, que proponemos (Fig. 3.6), le añade la interpretación de las responsabilidades a la actividad de "Mediciones y análisis" con respecto al modelo textual, lo que a nuestro juicio facilita entender en que grupos debe de recaer.

Verificaciones

La verificación 1 especifica que las actividades de administrar requisitos serán revisadas periódicamente por el administrador gerencial y la verificación 2 por el administrador de proyecto. Como habíamos mencionado, en el modelo gráfico ambos administradores son considerados como el grupo *Administrador* por la similitud de sus responsabilidades. Las dos verificaciones las representamos (Fig. 3.7) con la actividad de "Revisar actividades AdR", la cual recae en la responsabilidad del grupo *Administrador*.

La revisión la aplica el grupo *Administrador* a las actividades de AdR realizadas por el Grupo de IS. Por esta razón, en el diagrama la actividad "Revisar actividades AdR" aparece como compartida entre ambos grupos involucrados.

Finalmente, en la verificación 3 el grupo de AC revisa y/o audita las actividades y productos de trabajo para AdR y reporta resultados. Esta verificación está representada con la actividad de "Revisar y/o auditar actividades y producto de trabajo" compartida entre el Grupo de AC y Grupo de IS.

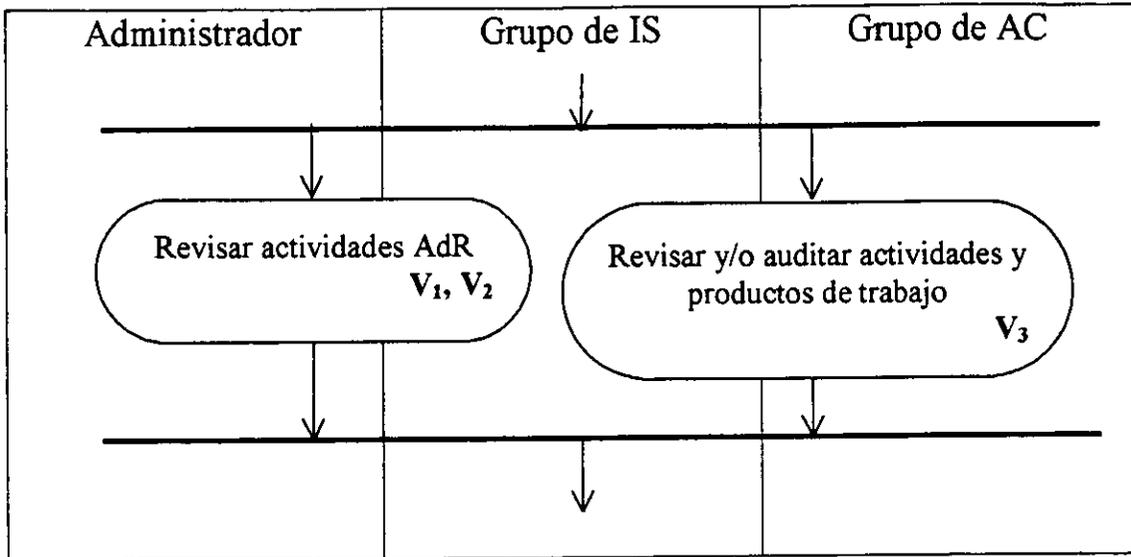


Fig. 3.7: Verificaciones

Capítulo 4

Planeación del proyecto de software

La planeación del proyecto establece los planes adecuados para realizar ingeniería de software y para administrar el proyecto. El proceso de planeación incluye la estimación del tamaño de los productos de trabajo y de los recursos necesarios para generar un calendario, identificar y evaluar los riesgos del software, así como negociar los compromisos.

En este capítulo presentaremos la traducción de esta área clave de procesos [CMM, 1994], posteriormente iniciamos la descripción del modelo gráfico con la presentación de los grupos participantes y la concluimos con la explicación de los diagramas de actividades, que constituyen el modelo gráfico.

4.1. Modelo textual de la Planeación del Proyecto de Software (PPS)

El propósito de la Planeación del Proyecto de Software es establecer planes adecuados para realizar ingeniería de software y para administrar el proyecto de software.

La Planeación del Proyecto de Software involucra desarrollar estimaciones del trabajo a ser realizado, estableciendo los compromisos necesarios, y definiendo el plan de trabajo.

La planeación de software inicia con una descripción del trabajo a ser realizado y de las restricciones y metas que definen y limitan el proyecto de software (establecidos en las prácticas del área clave de procesos de Administración de Requisitos). El proceso de planeación de software incluye los pasos para estimar el tamaño de los productos de trabajo de software y los recursos necesarios, generar un calendario, identificar y evaluar los riesgos del software, y negociar los compromisos. Puede ser necesario iterar estos pasos para establecer el plan para el proyecto de software (es decir, el plan de desarrollo de software).

Este plan provee la base para realizar y administrar las actividades del proyecto de software y aborda los compromisos con el cliente del proyecto de software de acuerdo a los recursos, las restricciones y las capacidades del proyecto.

Metas

Meta 1

Documentar las estimaciones respecto al software, para su uso en la planeación y seguimiento del proyecto del software.

Meta 2

Planear y documentar las actividades y compromisos del proyecto de software.

Meta 3

Los grupos e individuos involucrados aceptan sus compromisos relacionados al proyecto del software.

Compromisos

Compromiso 1

Se designa un administrador de proyecto de software, para que sea responsable de negociar los compromisos y realizar el plan de desarrollo del proyecto de software.

Compromiso 2

El proyecto sigue una política organizacional escrita para planear un proyecto de software.

Esta política típicamente especifica que:

1. Los requisitos del sistema se usan como la base para la planeación del proyecto del software.

Referirse a la Actividad 2 del área clave de procesos de Administración de Requisitos.

2. Los compromisos del proyecto del software se negocian entre:
 - el administrador de proyecto,
 - el administrador de proyecto de software, y,
 - los otros administradores de software.
3. La participación de otros grupos de ingeniería en las actividades de software se negocia con estos grupos y se documenta.

Ejemplos de otros grupos de ingeniería:
ingeniería de sistemas,
ingeniería de hardware, y,
pruebas del sistema.

4. Los grupos involucrados revisan del proyecto de software:
 - las estimaciones de tamaño del software,
 - las estimaciones de esfuerzo y costo,
 - la calendarización, y
 - otros compromisos.

Ejemplos de grupos involucrados:
ingeniería de software (incluye todos sus subgrupos, por ejemplo diseño de software),
estimación del software,
ingeniería de sistemas,
pruebas del sistema,
aseguramiento de calidad de software,
administración de configuración de software,
administración de contrato, y,
soporte de documentación.

5. El administrador gerencial revisa todos los compromisos del proyecto de software hechos a individuos y grupos externos a la organización.
6. El plan de desarrollo de software del proyecto se administra y se controla.

El término "plan de desarrollo de software" se usa en estas prácticas para referirse al plan completo para administrar el proyecto de software. El uso del término "desarrollo" no está pensado para excluir proyectos de mantenimiento o soporte y debe ser interpretado apropiadamente en el contexto del proyecto individual.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en una forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos de Administración de Configuraciones de Software.

Habilidades

Habilidad 1

Para el proyecto de software existe una descripción de trabajo documentada y aprobada.

1. La descripción de trabajo cubre:
 - alcance del trabajo,
 - metas y objetivos técnicos,
 - identificación de clientes y de usuarios finales,

Los usuarios finales mencionados en estas prácticas son los clientes o representantes de los usuarios finales.

- estándares impuestos,
- responsabilidades asignadas,
- restricciones de costo y de calendario y metas,
- dependencias entre el proyecto de software y otras organizaciones,

Ejemplos de otras organizaciones:

el cliente,
subcontratistas, y
compañeros de trabajo.

- restricciones de recursos y metas, y
 - otras restricciones y metas para desarrollo y/o mantenimiento.
2. La descripción de trabajo es revisada por:
 - el administrador de proyecto,
 - el administrador de proyecto de software,
 - otros administradores de software, y
 - otros grupos involucrados.
 3. La descripción de trabajo es administrada y controlada.

Habilidad 2

Se asignan las responsabilidades para realizar el plan de desarrollo de software.

1. El administrador de proyecto de software, en forma directa o delegada, coordina la planeación del proyecto de software.
2. Las responsabilidades de los productos de trabajo y actividades de software se dividen y se asignan a los administradores de software en forma fácil de seguir.

Ejemplos de productos de trabajo de software:

- productos que serán entregados a los clientes externos o usuarios finales;
- productos que serán usados por otros grupos de ingeniería; y
- productos para uso interno del grupo de ingeniería de software.

Habilidad 3

Se proveen los recursos y financiamiento adecuados para planear el proyecto de software.

1. Se dispone de individuos con experiencia en el dominio de aplicación del proyecto de software, para realizar el plan de desarrollo de software.
2. Las herramientas de soporte de las actividades de la planeación del proyecto de software se hacen disponibles.

Ejemplos de herramientas de soporte:

- hojas de cálculo,
- modelos de estimación, y
- programas de planeación de proyecto y calendarización.

Habilidad 4

Los administradores de software, ingenieros de software, y otros individuos involucrados en la planeación del proyecto de software son capacitados en los procedimientos de estimación y planeación de software aplicables a sus áreas de responsabilidad.

Actividades

Actividad 1

El grupo de ingeniería de software participa en el equipo de propuesta del proyecto.

1. Se involucra al grupo de ingeniería de software en:
 - preparación y presentación de la propuesta,
 - discusiones y presentaciones de aclaraciones, y
 - negociación de cambios a los compromisos que afectan al proyecto de software.
2. El grupo de ingeniería de software revisa los compromisos del proyecto.

Ejemplos de compromisos del proyecto:
las metas y objetivos técnicos;
la solución técnica del software y del sistema;
el presupuesto, calendario, y recursos; y
los estándares y procedimientos.

Actividad 2

La planeación del proyecto de software se inicia en las primeras etapas de, y en paralelo con, la planeación del proyecto global.

Actividad 3

El grupo de ingeniería de software participa con otros grupos involucrados en la planeación del proyecto global durante el tiempo de vida del proyecto.

1. El grupo de ingeniería de software revisa los planes del nivel correspondientes al proyecto de software.

Actividad 4

Se revisan los compromisos del proyecto de software hechos a individuos y grupos externos a la organización con el administrador gerencial de acuerdo a un procedimiento documentado.

Actividad 5

Se identifica o se define un ciclo de vida de software con etapas predefinidas de tamaño administrable.

Ejemplos de ciclos de vida de software:
cascada,
cascada con traslape,
espiral,
construcción en serie, y
prototipo único/ cascada con traslape.

Actividad 6

Se realiza el plan de desarrollo del proyecto de software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El plan de desarrollo de software se basa y sigue:
 - los estándares del cliente;
 - los estándares del proyecto;
 - la descripción aprobada de trabajo; y
 - los requisitos.

2. Los planes para los grupos relacionados al software y otros grupos de ingeniería involucrados en las actividades del grupo de ingeniería de software se negocian con estos grupos, los esfuerzos de soporte se presupuestan, y los acuerdos se documentan.

Ejemplos de grupos relacionados al software:
aseguramiento de calidad de software,
administración de configuraciones de software, y
soporte de documentación.

Ejemplos de otros grupos de ingeniería:
ingeniería de sistemas,
ingeniería de hardware, y
prueba de sistemas.

3. Los planes para la participación del grupo de ingeniería de software en las actividades de otros grupos relacionados al software y otros grupos de ingeniería se negocian con estos grupos, los esfuerzos de soporte se presupuestan, y los acuerdos se documentan.
4. El plan de desarrollo del software es revisado por:
 - el administrador de proyecto,
 - el administrador de proyecto de software,
 - los otros administradores de software, y
 - otros grupos involucrados.
5. El plan de desarrollo del software es administrado y controlado.

Actividad 7

Se documenta el plan para el proyecto de software.

En las prácticas clave, este plan o colección de planes se refiere como el plan de desarrollo de software.

Referirse a la Actividad 1 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que conciernen al uso del plan de desarrollo de software durante el proyecto.

El plan de desarrollo de software cubre:

1. El propósito, alcance, metas y objetivos del proyecto de software.
2. La selección de un ciclo de vida de software.
3. La identificación de los procedimientos, métodos y estándares seleccionados para desarrollar y/o mantener el software.

Ejemplos de estándares y procedimientos de software:
planeación de desarrollo de software,
administración de configuraciones de software,
aseguramiento de calidad de software,
diseño de software,
seguimiento y resolución de problemas, y
mediciones de software.

4. La identificación de los productos de trabajo de software a ser desarrollados.
5. Las estimaciones de tamaño de los productos de trabajo de software y algunos cambios a estos.
6. Las estimaciones del esfuerzo y costos del proyecto de software.
7. El uso estimado de los recursos computacionales críticos.
8. El calendario del proyecto del software, incluyendo la identificación de las fechas o eventos predeterminados (*milestones*) y revisiones.
9. La identificación y evaluación de los riesgos del proyecto de software.
10. Planes para los servicios de ingeniería de software y herramientas de soporte del proyecto de software.

Actividad 8

Se identifican los productos de trabajo de software que son necesarios para establecer y mantener el control del proyecto de software.

Referirse a la Actividad 4 del área clave de procesos de Administración de Configuraciones de Software.

Actividad 9

Se obtienen las estimaciones para el tamaño de los productos de trabajo de software (o cambios al tamaño de los productos de trabajo de software) de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las estimaciones de tamaño se hacen para todos los productos de trabajo de software y actividades.

Ejemplos de mediciones de tamaño de software:

puntos de función,
puntos característicos,
líneas de código,
número de requisitos, y
número de páginas.

Ejemplos de tipos de productos de trabajo y actividades, para los cuales las estimaciones de tamaño se hacen:

software operacional y de soporte,
productos de trabajo entregables y no entregables,
productos de trabajo de software y los que no pertenecen al software (por ejemplo, los documentos), y
actividades para desarrollar, verificar y validar productos de trabajo.

2. Los productos de trabajo de software se desglosan a la granularidad necesaria para alcanzar los objetivos estimados.
3. Se usan los datos históricos donde estén disponibles.
4. Los supuestos de las estimaciones de tamaño se documentan.
5. Las estimaciones de tamaño se documentan, se revisan, y se acuerdan.

Ejemplos de grupos e individuos que revisan y acuerdan las estimaciones de tamaño:
el administrador de proyecto,
el administrador de software de proyecto, y
los otros administradores de software.

Actividad 10

Se obtienen las estimaciones del esfuerzo y costos del proyecto de software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las estimaciones de esfuerzo y costos del proyecto de software están relacionadas con las estimaciones de tamaño de los productos de trabajo de software (o tamaño de los cambios).
2. En caso de su disponibilidad se usan los datos de productividad (históricos y/o actuales) para las estimaciones. Las fuentes y la justificación de estos datos deben estar documentadas.
 - Siempre que sea posible, los datos de productividad y costo provienen de los proyectos de la organización.
 - Los datos de productividad y costo toman en cuenta el esfuerzo y costos significantes que se utilizan para hacer los productos de trabajo de software.

Ejemplos de costos significativos que se utilizan para hacer los productos de trabajo de software:
gastos de trabajo,
gastos generales,
gastos de viaje, y
costos de uso de computadoras.

3. Las estimaciones del esfuerzo, de personal y de costo se basan en la experiencia.
 - Los proyectos similares deben usarse cuando sea posible.
 - El tiempo de actividades se estima por fase.
 - Se prepara la distribución de las estimaciones del esfuerzo, del personal y del costo en el ciclo del software.
4. Las estimaciones y las suposiciones se documentan, se revisan y se acuerdan.

Actividad 11

Se obtienen las estimaciones para los recursos computacionales críticos de acuerdo a un procedimiento documentado.

Los recursos computacionales críticos podrían estar en el ambiente *host* (anfitrión), en el ambiente de integración y prueba, en el ambiente destino, o en alguna combinación de estos.

Este procedimiento típicamente especifica que:

1. Se identifican los recursos computacionales críticos para el proyecto.

Ejemplos de recursos computacionales críticos:
capacidad de memoria,
uso de procesador, y
capacidad del canal de comunicaciones.

2. Las estimaciones para los recursos computacionales críticos están relacionadas con las estimaciones de:
 - el tamaño de los productos de trabajo de software,
 - la carga de proceso operacional, y
 - el tráfico de las comunicaciones.
3. Las estimaciones para los recursos computacionales críticos se documentan, se revisan, y se acuerdan.

Actividad 12

Se genera el calendario del proyecto de software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El calendario del software se relaciona a:
 - el tamaño estimado de los productos de trabajo de software (o el tamaño de los cambios), y
 - el esfuerzo y costos del software.
2. El calendario del software se basa en experiencias pasadas.
 - Se usan los proyectos similares cuando sea posible.
3. El calendario del software acomoda las fechas de *milestones*¹ definidos, fechas de dependencias críticas, y otras restricciones.
4. Las actividades del calendario de software son de duración adecuada y los *milestones*¹ son de una separación de tiempo apropiada para soportar la precisión de la medición del progreso.
5. Las suposiciones tomadas para la calendarización se documentan.
6. El calendario de software se documenta, se revisa y se acuerda.

¹ En esta ocasión no traducimos *milestone* por problemas de redacción. *Milestone* es una fecha o evento previamente seleccionado.

Actividad 13

Se identifican, se evalúan y se documentan los riesgos del software asociados con el costo, los recursos, el calendario y los aspectos técnicos del proyecto.

1. Los riesgos se analizan y se ordenan por importancia en su impacto potencial al proyecto.
2. Las contingencias para los riesgos se identifican.

Ejemplos de contingencias:

margen de tolerancia en la calendarización,
planes alternativos de personal, y
planes alternativos para equipo de cómputo adicional.

Actividad 14

Se preparan los planes de los servicios para la ingeniería de software del proyecto y herramientas de soporte.

1. Las estimaciones de los requisitos de capacidad para estos servicios y herramientas de soporte se basan en las estimaciones de tamaño de los productos de trabajo de software y otras características.

Ejemplos de servicios y herramientas de soporte de desarrollo de software:

computadoras y periféricos para desarrollo de software,
computadoras y periféricos para prueba de software,
software para el ambiente de la computadora destino, y
otro software de soporte.

2. Las responsabilidades se asignan y los compromisos se negocian para proporcionar o desarrollar estos servicios y herramientas de soporte.
3. Los planes se revisan por todos los grupos involucrados.

Actividad 15

Se registran los datos de planeación de software.

1. La información registrada incluye las estimaciones y la información necesaria para reconstruir las estimaciones y evaluaciones.
2. Los datos de planeación de software se administran y se controlan.

Mediciones y Análisis

Medición 1

Las mediciones se definen y se usan para determinar el estado de las actividades de la planeación del software.

Ejemplos de mediciones:

*milestones*² concluidos de las actividades de la planeación del proyecto de software comparados con el plan; y
trabajo terminado, esfuerzo y financiamiento utilizados en las actividades de la planeación del proyecto de software comparados con el plan.

Verificaciones

Verificación 1

Las actividades para la planeación del proyecto de software se revisan periódicamente 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.

1. Se revisa el cumplimiento técnico, de costo, del personal y de calendario.
2. Se analizan los conflictos y los problemas no resueltos en los niveles inferiores.
3. Se analizan los riesgos del proyecto de software.
4. Se asignan, se revisan y se siguen las acciones acordadas hasta su cierre.
5. Se prepara un reporte resumido de cada reunión y se distribuye a los individuos y grupos involucrados.

Verificación 2

Las actividades para la planeación del proyecto de software se revisan con el administrador de proyecto periódicamente y/o a causa de un evento.

1. Los grupos involucrados son representados.
2. El estado y los resultados actuales de las actividades de la planeación del proyecto de software se revisan contra el informe de trabajo del proyecto de software y los requisitos.
3. Se analizan las dependencias entre grupos.
4. Se analizan los conflictos y los problemas no resueltos en los niveles inferiores.
5. Se analizan los riesgos del proyecto de software.
6. Se asignan, se revisan y se siguen las acciones acordadas hasta su cierre.
7. Se prepara un reporte resumido de cada reunión y se distribuye a los individuos y grupos involucrados.

² *Milestone*: Evento o fecha previamente acordada.

Verificación 3

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

Referirse al área clave de procesos de Aseguramiento de Calidad de Software.

Las revisiones y/o auditorías verifican al menos:

1. Las actividades de estimación y planeación de software.
2. Las actividades de planeación y aceptación de los compromisos del proyecto.
3. Las actividades de preparación del plan de desarrollo de software.
4. Los estándares usados para preparar el plan de desarrollo de software.
5. El contenido del plan de desarrollo de software.

4.2. Modelo gráfico de la Planeación del Proyecto de Software (PPS)

El modelado gráfico iniciará con una identificación en todo el texto original, de los grupos de actores participantes en el proceso de PPS y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

4.2.1. Identificación de los grupos participantes.

En el modelo textual de la planeación del proyecto de software hemos identificado y resaltado en letras itálicas a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentaremos una breve definición de cada uno de ellos.

Administrador de proyecto: Es el responsable de negociar los compromisos y de realizar el plan de desarrollo de software.

Grupo de Ingeniería de Software (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 proceso de ingeniería de software no están incluidos en el grupo de ingeniería de software.

Este grupo participa en el equipo de propuesta del proyecto y en la planeación del proyecto global durante el tiempo de vida del proyecto.

Ingeniero de software: Miembro del grupo de ingeniería de software.

Administrador gerencial: Un rol de administración en 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 tendría la responsabilidad de varios proyectos a la vez.

Grupo de Aseguramiento de Calidad (AC). Es el responsable de revisar y/o auditar las actividades y productos de trabajo para planear el proyecto de software y reporta resultados.

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

4.2.2. Diagramas de actividades

Compromisos

La parte de prácticas del modelo textual inicia con las agrupadas en la característica común de **Compromisos**. En esta área clave contamos con dos compromisos. El primero establece un liderazgo y el segundo una política organizacional.

El compromiso 1 dice que se designa un administrador de proyecto de software, él será responsable de negociar los compromisos y realizar el plan de desarrollo del proyecto de software. Este compromiso lo representamos con la actividad "Designar administrador de proyecto" (Fig. 4.1), y se propone como una actividad realizada por el *Administrador gerencial*, aunque en el texto no lo dice explícitamente pero creemos que es una aseveración congruente.

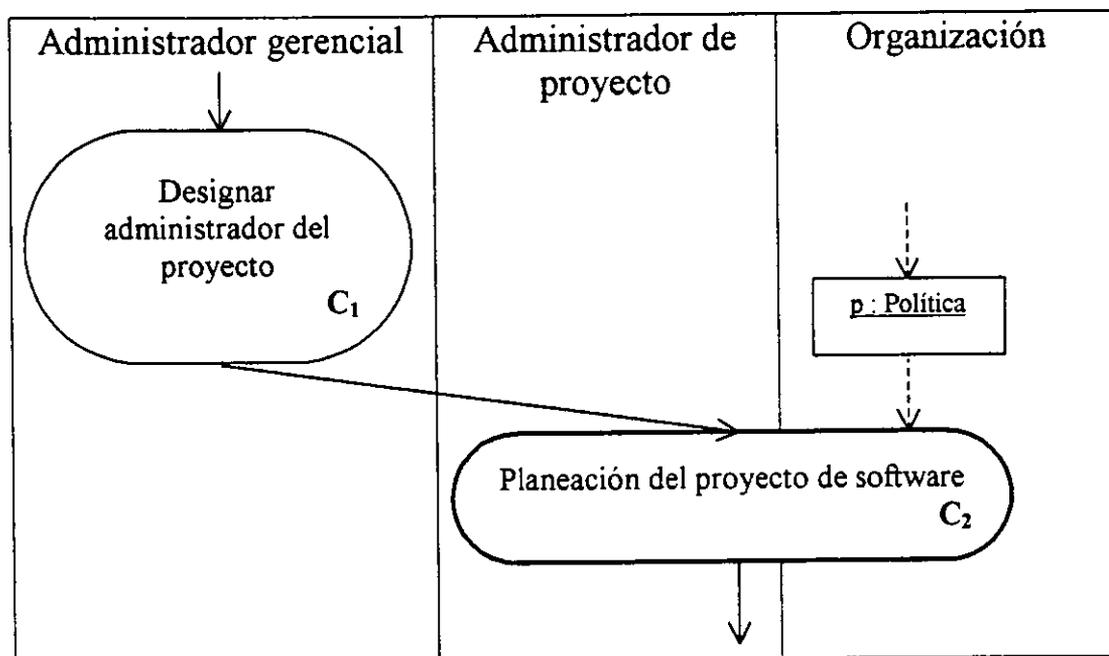


Fig. 4.1: Compromisos

Del texto del compromiso 2, se deduce que la organización tiene una política escrita que guíe la planeación del proyecto de software. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Fig.4.1) como un objeto *p:Política*. Este objeto fue probablemente generado por una actividad de definición de política, la cuál, de acuerdo al texto del compromiso, queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir la política para planear el proyecto de software queda expresado colocando como objeto de entrada a *p:Política* a la actividad de "Planeación del proyecto de software". Esta actividad se representa en el diagrama, como una actividad compartida entre el *Administrador de proyecto* y la *Organización*. Consideramos que ambos realizan esta actividad, porque mientras el *Administrador de proyectos* coordina la PPS, la *Organización* proporciona los elementos necesarios para realizar efectivamente esta actividad.

La Fig. 4.1 ofrece una síntesis gráfica del área clave de procesos PPS y, a la vez, nos da una vista global de las relaciones entre el grupo *Administrador*³ y el resto de la *Organización* con respecto a la política de PPS.

Una vez que hemos explicado los **Compromisos**, nos falta explicar las características comunes restantes, **Habilidades**, **Actividades**, **Mediciones y Análisis** y **Verificaciones**, son las que realmente describen las acciones que los diferentes grupos participantes deben de realizar la planeación del proyecto de software. Para modelar este hecho, la actividad de “*Planeación del proyecto de software*” de la Fig. 4.1 se desglosa en el diagrama presentado en la Fig. 4.2.

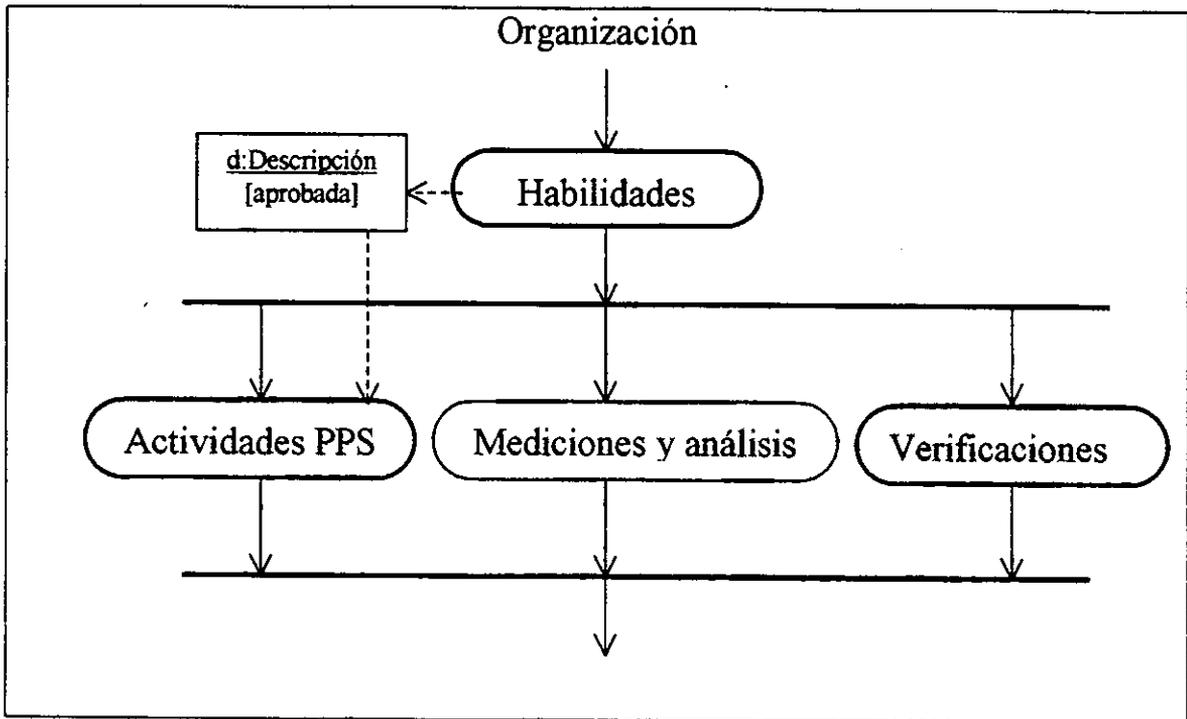


Fig. 4.2: Planeación del proyecto de software

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

³ En el grupo *Administrador* están incluidos el *administrador gerencial* y el *administrador de proyecto*.

Habilidades

Esta sección está integrada por cuatro habilidades. La primera de ellas especifica que existe una descripción de trabajo documentada y aprobada. En esta ocasión no la vamos a representar como en otros casos parecidos, es decir, no vamos a considerar que esta descripción existe de manera implícita. Esta decisión, la tomamos al analizar las subprácticas de la habilidad 1, en las cuales explica las características de la descripción de trabajo. De acuerdo a estas características, la descripción de trabajo debe crearse, documentarse y aprobarse para cada uno de los proyectos de la organización. Por lo que la habilidad 1 la modelamos, (Fig. 4.3), con la actividad "Crear, documentar y aprobar descripción de trabajo". De esta actividad obtenemos la descripción de trabajo para el proyecto de software, que representamos con el objeto *d:Descripción* en estado de aprobado. Creemos que esta actividad es realizada en forma compartida entre el *Administrador de proyecto*, que es responsable de la planeación de proyecto, y la *Grupo de IS*, que es idóneo para proporcionar información contenida en la descripción.

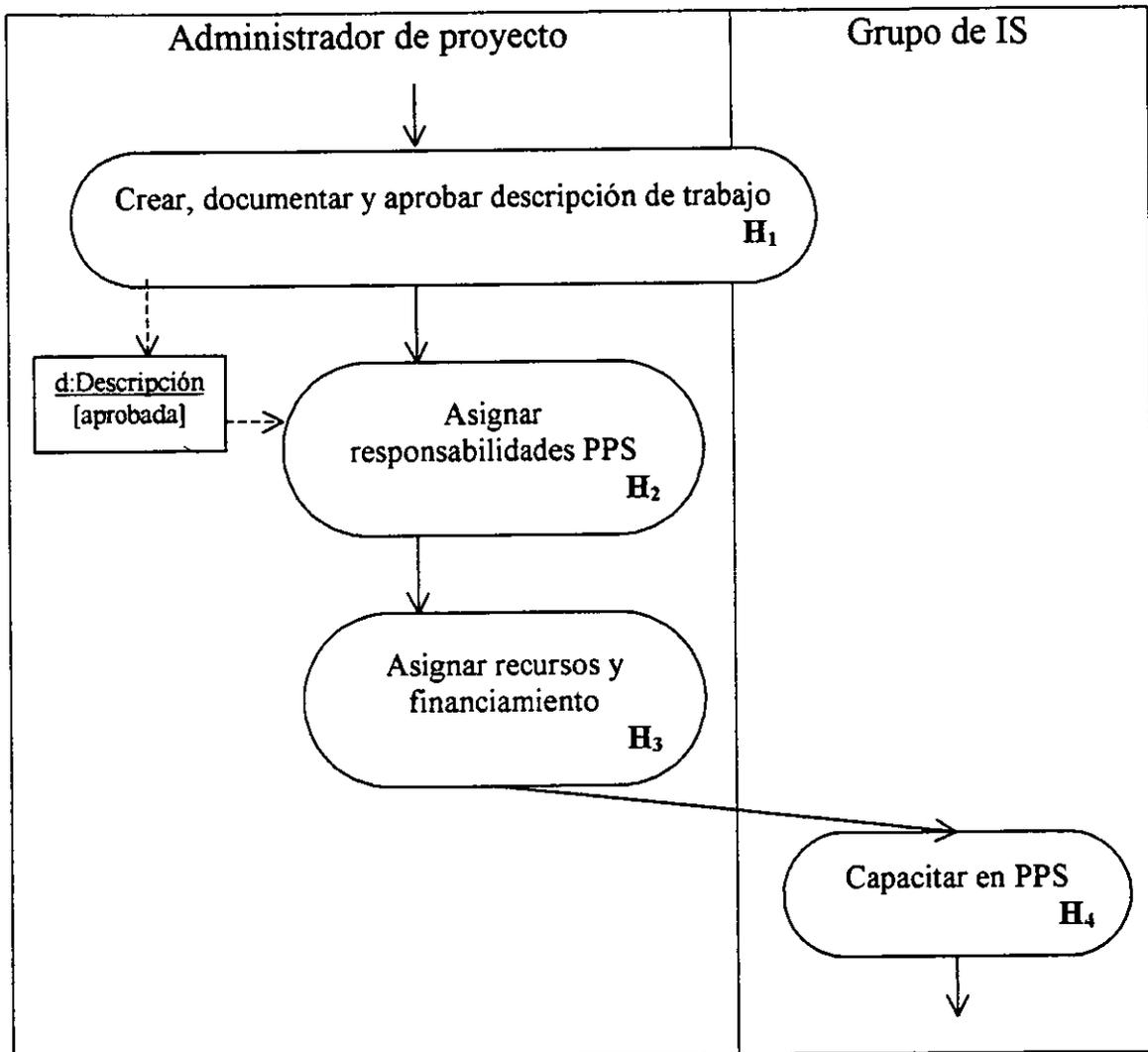


Fig. 4.3: Habilidades

Una vez aprobada la descripción de trabajo, esta es utilizada por el *Administrador de proyecto* para asignar responsabilidades para realizar el plan de desarrollo de software. Lo anterior es especificado en la habilidad 2 y lo representamos por la actividad “*Asignar responsabilidades PPS*” (Fig. 4.3). El flujo del objeto *d:Descripción* a la actividad “*Asignar responsabilidades PPS*” representa su uso en esta actividad. El modelo textual no establece explícitamente que el *Administrador de proyecto* realiza actividad, pero nos parece razonable que sea de esta forma (Fig. 4.3).

La habilidad 3 habla de la provisión de recursos y financiamiento para planear el proyecto de software, también se puede interpretar que está a cargo del *Administrador de proyecto*. Esta habilidad la modelamos con la actividad “*Asignar recursos y financiamiento*” y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación.

La última habilidad trata la capacitación de los administradores de software, ingenieros de software y otros individuos involucrados, en los procedimientos de estimación y planeación de software aplicables a sus áreas de responsabilidad. Para representar esta habilidad sólo consideramos al *Grupo de IS*, por lo que en el diagrama este grupo lleva a cabo la actividad “*Capacitar en PPS*”.

El diagrama de la Fig. 4.3 presenta las actividades a realizarse para adquirir las habilidades para la PPS, donde cada una de las habilidades es mapeada en forma directa a la actividad que le corresponde. Una característica especial de este diagrama, es que todas las actividades se realizan en forma secuencial.

Actividades

El modelo textual de esta sección está constituido por 15 actividades, las cuales modelaremos en tres diagramas para una mejor representación.

La actividad 1 dice que el *Grupo de IS* participa en el equipo de propuesta del proyecto. Esta actividad la representamos con “*Desarrollar propuesta del proyecto*” (Fig. 4.4). A pesar que en el texto de la actividad no expresa explícitamente que se documenta una propuesta como tal, en nuestro modelo decidimos hacerlo explícito. En el texto del modelo de esta actividad queda plasmado en forma literal, que “*Desarrollar propuesta del proyecto*” es realizada en forma compartida entre el *Administrador de proyecto* y el *Grupo de IS*.

La actividad 2 dice que la planeación del proyecto de software inicia en las primeras etapas de, y en paralelo con, la planeación del proyecto global. Lo anterior lo modelamos con la ejecución en forma simultánea de las actividades “*Planear proyecto de software*” y “*Planear proyecto global*”. Ambas actividades reciben como entrada el objeto *pro:Propuesta*.

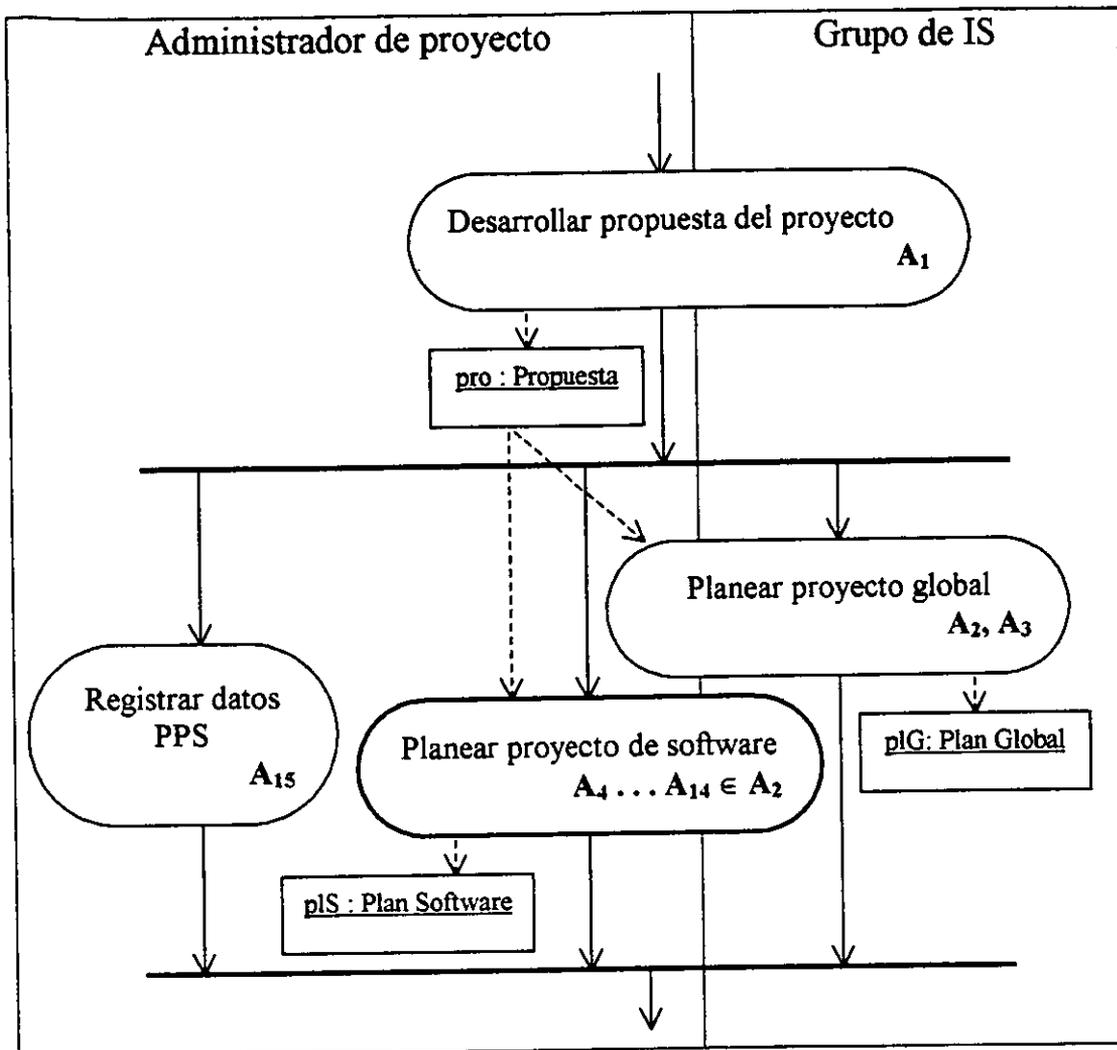


Fig. 4.4: Actividades PPS

La actividad 3 plantea que el *Grupo de IS* participa con otros grupos involucrados en la planeación del proyecto global durante el tiempo de vida del proyecto. Esta actividad la representamos (Fig. 4.4) con el hecho que “*Planear proyecto global*” se lleva a cabo en forma compartida entre el *Administrador de proyecto* y el *Grupo de IS*. Por similitud suponemos que “*Planear proyecto de software*” se realiza en forma compartida por ambos grupos. En el diagrama no incluimos a otros grupos involucrados en la planeación global por razones de legibilidad.

Analizando el texto de la sección de **Actividades**, encontramos que de la actividad 4 a la 14 están contenidas en la actividad 2 que introduce en forma general la planeación del proyecto de software y su relación con la planeación global. La inclusión de estas subactividades en la actividad 2 la modelamos con la expresión $A_4 \dots A_{14} \in A_2$ (Fig. 4.4).

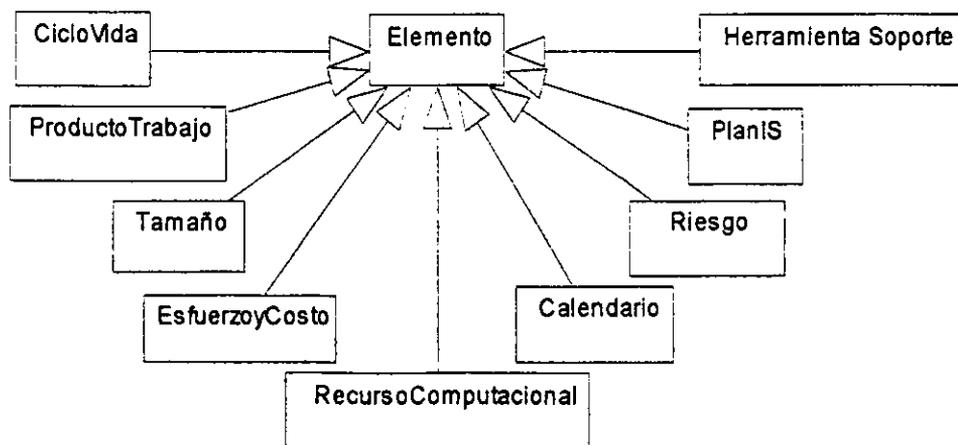
Finalmente durante la planeación del proyecto de software se registran datos de esta actividad, hecho que se expresa en la actividad 15. En el diagrama (Fig. 4.4), lo plasmamos con la actividad "Registrar datos PPS", que se realiza en paralelo con "Planear proyecto de software".

El diagrama de la Fig. 4.4 muestra un vista compacta, que resume las actividades de PPS.

El diagrama de actividades de la Fig. 4.5 desglosa "Planear proyecto de software"⁴. Como habíamos mencionado en este diagrama representamos las actividades de la actividad 4 a la 14.

La actividad 4 establece que se revisan los compromisos del proyecto de software, hechos a individuos y grupos externos a la organización, con el *Administrador gerencial* de acuerdo a un procedimiento documentado. Esta actividad, la representamos con "Revisar compromisos", que tiene como objetos de entrada *c:Compromisos* y *pr:Procedimiento*, que de alguna manera se conocen implícitamente y un objeto de salida *c:Compromisos* en estado de revisado. La actividad "Revisar compromisos" se realiza en paralelo con la actividad "Identificar elementos PPS". Esta última no mapea directamente alguna actividad del modelo textual, pero si con ella representamos la actividad 5 y de la actividad 8 a la 14. De "Identificar elementos PPS" obtenemos los elemento de la PPS, que representamos con el objeto *e:Elemento*.

Los elementos que se identifican en la actividad "Identificar elementos PPS" son: ciclo de vida (A₅), productos de trabajo de software (A₈), estimaciones del tamaño de los productos de trabajo de software (A₉), estimaciones de esfuerzo y costos (A₁₀), estimaciones de los recursos computacionales (A₁₁), calendario (A₁₂), riesgos de software (A₁₃), planes de los servicios para IS (A₁₄) y herramientas de soporte (A₁₄). Estos elementos los modelamos como una especialización de objetos y la observamos en el siguiente diagrama de clases.



⁴ Esta actividad es del diagrama de actividades de la Fig. 4.4.

La actividad 6 dice que se realiza el plan de desarrollo del proyecto de software de acuerdo a un procedimiento documentado. Esta actividad la representamos en forma directa con “Realizar plan de desarrollo”, que tiene como objeto de entrada *pr:Procedimiento*. En las subprácticas de esta actividad (A_6) se especifica que entre otros documentos el plan de desarrollo se basa y sigue en la descripción aprobada de trabajo. En el diagrama de la Fig. 4.5, este hecho lo representamos con el objeto de entrada *d:Descripción* que recibe “Realizar plan de desarrollo”. El otro objeto de entrada a esta actividad es *c:Compromiso* en estado de revisado, a pesar que en el modelo textual no hace referencia a este hecho, pero creemos que los compromisos deben ser considerados para la realización del plan de desarrollo. Al termino de esta actividad obtenemos un plan de desarrollo, que representamos con el objeto *pIS:PlanSoftware*.

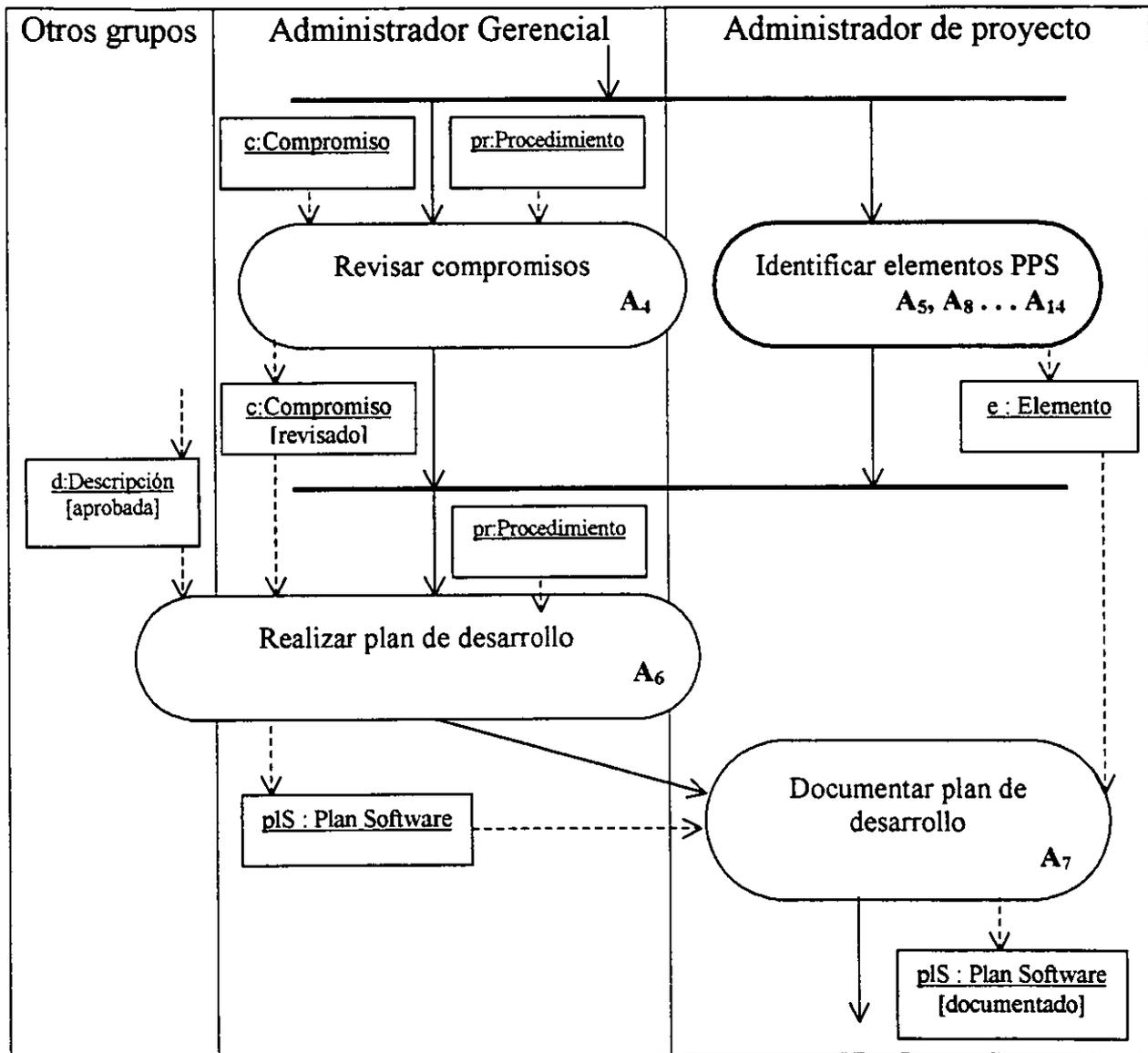


Fig. 4.5: Planear proyecto de software

La actividad 7 establece que se documenta el plan para el proyecto de software. Modelamos esta actividad con "*Documentar plan de desarrollo*", que recibe como objetos de entrada a *plS:PlanSoftware* y a *e:Elemento*, este último contiene la información necesaria para documentar el plan. Al terminar esta actividad obtenemos el objeto *plS:PlanSoftware* en estado de documentado⁵.

Las dos últimas actividades, modeladas en el diagrama de la figura 4.5, "*Realizar plan de desarrollo*" y "*Documentar plan de desarrollo*" se ejecutan en forma secuencial, y a partir de ellas obtenemos el plan de desarrollo, uno de los objetivos principales de esta área clave de procesos.

El siguiente diagrama (Fig. 4.6) desglosa la actividad "*Identificar elementos PPS*"⁶. Como habíamos mencionado en este diagrama modelaremos la actividad 5, y las actividades de la 8 a la 14.

El texto de la actividad 5, dice que se identifica o se define un ciclo de vida de software. Esta actividad la representamos con "*Identificar o definir ciclo de vida*", de la cual obtenemos el objeto *cv:CicloVida*, (Fig. 4.6).

De acuerdo al ciclo de vida elegido se identifican los productos de trabajo de software para establecer el control del proyecto, (*A₈*)⁷. Esto se modela con el flujo del objeto de *cv:CicloVida* que entra a la actividad "*Identificar productos de trabajo de software*" y de la cual se obtiene el objeto *pt:ProductoTrabajo*.

Una vez identificados los productos de trabajo se obtienen las estimaciones del tamaño de estos productos (*A₉*), que se representa con la entrada del objeto *pt:ProductoTrabajo* a "*Obtener estimaciones de tamaño de productos de software*". De esta actividad se obtiene el *t:Tamaño*, el cual contiene las estimaciones del tamaño de los productos de software,

La actividad 11 del modelo textual la mapeamos con "*Obtener estimaciones de recursos computacionales*". La cual para llevarse a cabo necesita el tamaño de los productos de trabajo, por lo que en el diagrama (Fig. 4.6), esta actividad recibe como objeto de entrada a *t:Tamaño* y tiene como objeto de salida a *rc:RecursoComputacional*.

Para estimar el esfuerzo y costo, (*A₁₀*), se requiere de las estimaciones de los recursos computacionales y del tamaño de los productos, por lo que en el diagrama (Fig. 4.6), "*Obtener estimaciones de esfuerzo y costo*" tiene como objeto de entrada a los objetos *t:Tamaño* y *rc:RecursoComputacional*. Cuando finaliza esta actividad se obtiene al objeto *ec:EsfuerzoCosto*.

⁵ El flujo del objeto *plS:PlanSoftware* en estado de documentado continua en la Fig. 5.3 de *Habilidades del áreas clave de procesos de Seguimiento y Control del Proyecto de Software (SCPS)*.

⁶ Esta actividad es del diagrama de actividades de la Fig. 4.5.

⁷ Texto tomado de modelo textual de la Actividad 8.

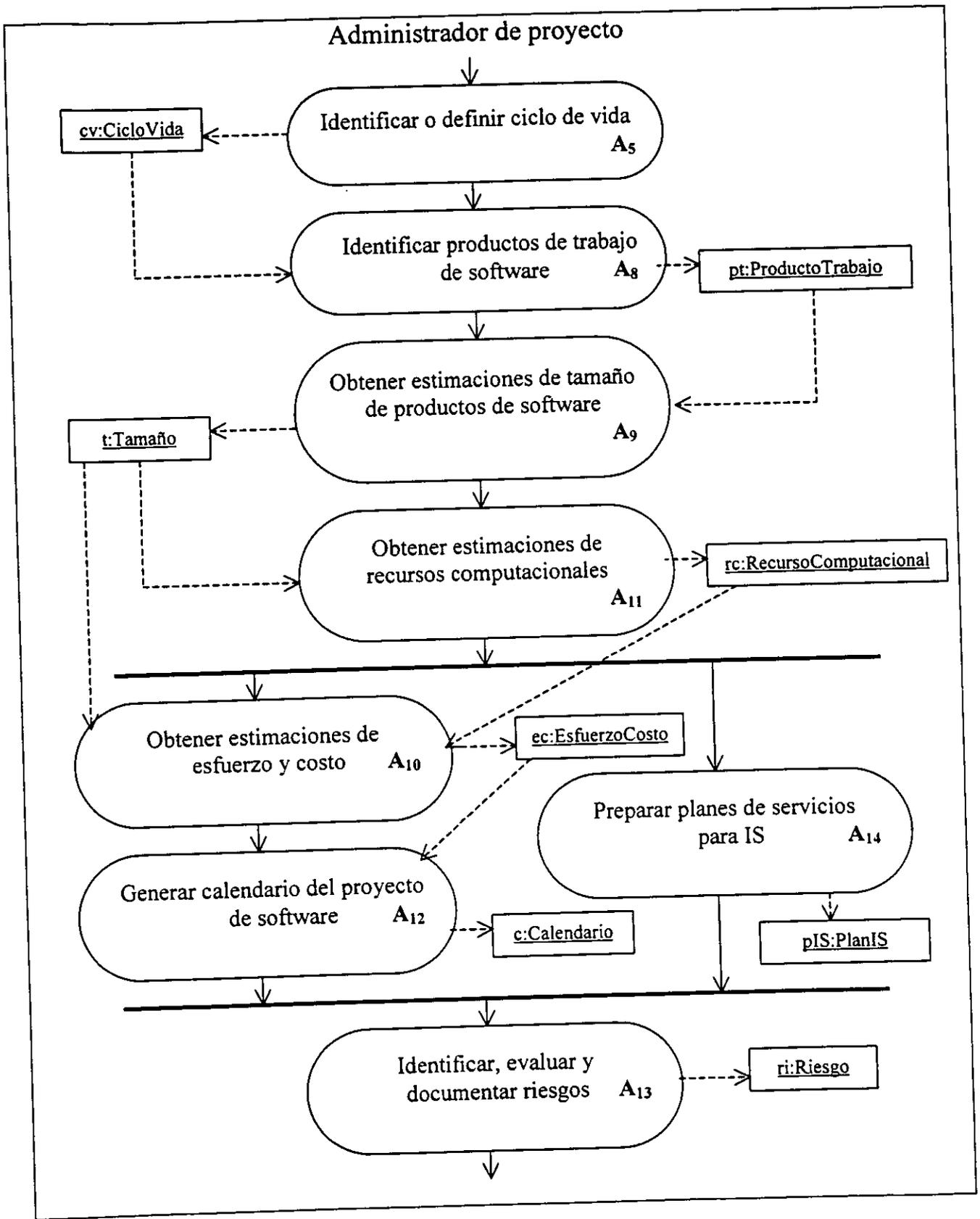


Fig. 4.6: Identificar elementos PPS

Con la estimación de esfuerzo y costo tenemos el conocimiento necesario para generar el calendario del proyecto de software; por lo que se lleva a cabo la actividad “*Generar calendario del proyecto de software*” y así obtener el calendario que modelamos con el objeto *c:Calendario*.

Las actividades “*Obtener estimaciones de esfuerzo y costo*” y “*Generar calendario del proyecto de software*” se realizan en forma secuencial y consideramos que ambas actividades se llevan a cabo en forma paralela a “*Preparar planes de servicios para IS*”. De esta actividad obtenemos el objeto *pIS:PlanIS*.

Finalmente con los elementos de PPS conocidos hasta este momento podemos identificar, evaluar y documentar los riesgos, hecho que representamos con la actividad “*Identificar, evaluar y documentar riesgos*” (A₁₃), ver figura 4.6.

Si analizamos el texto de las actividades de la 9 a la 12, encontraremos que se trata de obtener un elemento de PPS (tamaño de producto de trabajo de software, esfuerzo y costos, recursos computacionales críticos y calendario), por medio de un procedimiento. Las primeras tres actividades se refieren a una estimación y la última a una definición. Estas actividades las mapeamos en forma directa (Fig. 4.6), pero omitimos al objeto entrada *pr:Procedimiento* a cada una de estas actividades para obtener una mejor legibilidad del diagrama.

Mediciones y análisis

El texto correspondiente a la medición y análisis es muy escueto. Se dice que deben de definirse y usarse mediciones para determinar el estado de las actividades de la PPS. Sin embargo, falta precisar quienes son los responsables por efectuar dichas actividades. Conociendo todo el contexto del modelo textual de la PPS nos atrevemos a proponer el diagrama de la (Fig. 4.7), en donde la actividad de “*Mediciones y análisis*” es una actividad compartida entre el *Administrador de proyecto*, que debe ser responsable por recaudar datos de las mediciones, y el *Administrador gerencial* y el *Grupo de Aseguramiento de Calidad (AC)*, quienes utilizan estos datos para determinar el estado de las actividades de la PPS. Previamente, en la Fig. 4.2, ya quedó plasmado el paralelismo de las actividades de medición y análisis con las de PPS y de las verificaciones.

El diagrama, que proponemos (Fig. 4.7), le añade la interpretación de las responsabilidades a la actividad de “*Mediciones y análisis*” con respecto al modelo textual, lo que a nuestro juicio facilita entender en que grupos debe de recaer.

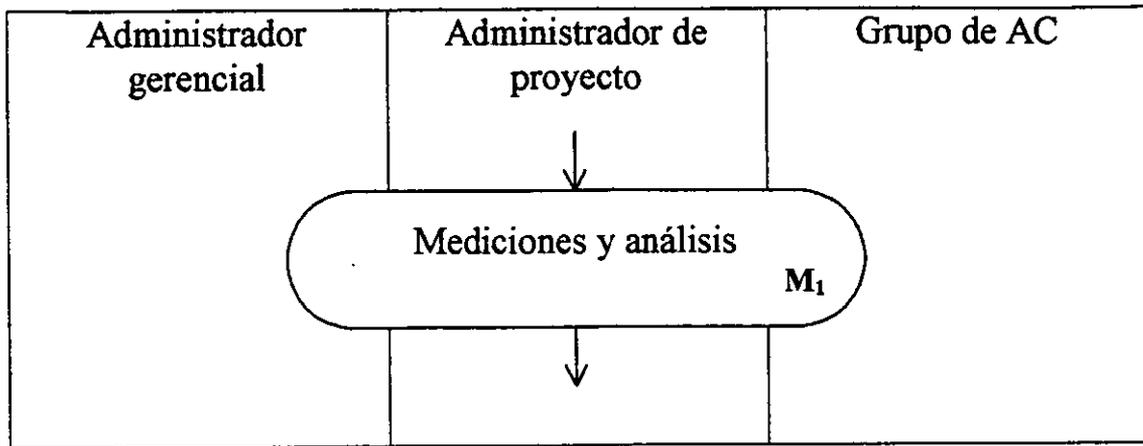


Fig. 4.7: Mediciones y análisis

Verificaciones

La verificación 1 especifica que las actividades para la planeación del proyecto de software serán revisadas periódicamente por el *Administrador gerencial*. Modelamos esta práctica con la actividad "Revisar actividades PPS" realizada por el *Administrador de proyecto*, que es el responsable de esta área clave, y por el *Administrador gerencial*, como lo indica el texto.

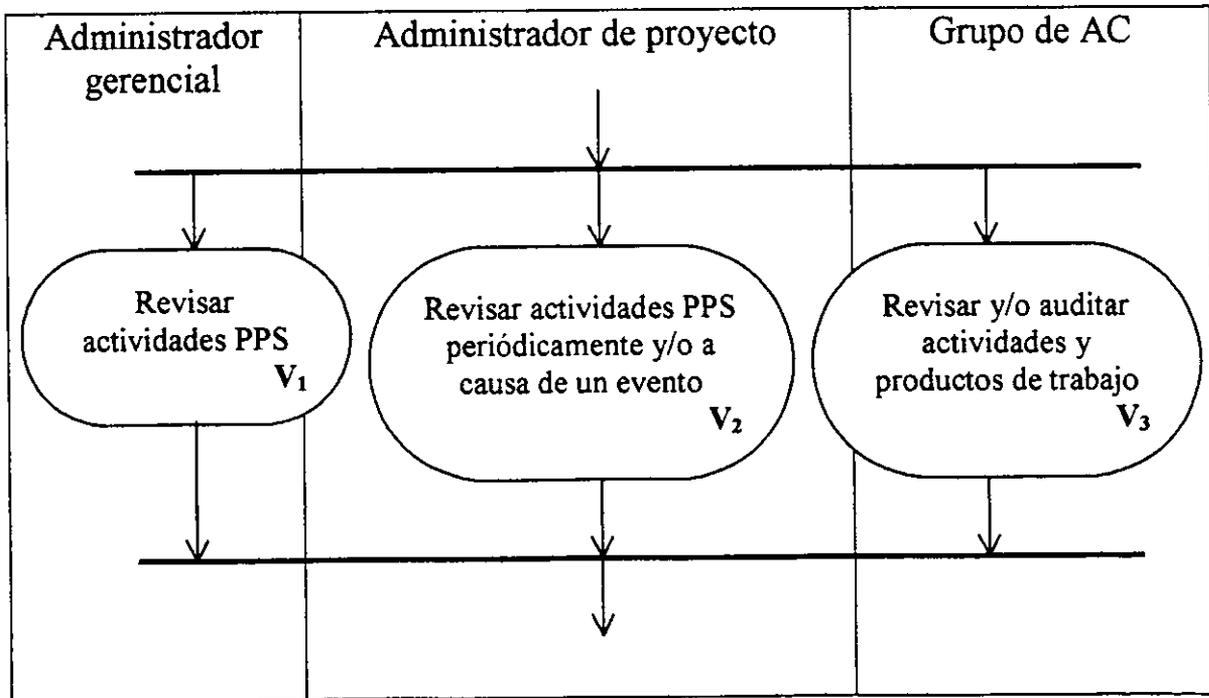


Fig. 4.8: Verificaciones

La verificación 2, que es similar a la primera, dice que las actividades para la planeación del proyecto se revisan con el *Administrador de proyecto* periódicamente y/o a causa de un evento, la cual mapeamos en forma directa a “*Revisar actividades PPS periódicamente y/o a causa de un evento*” cuya responsabilidad de llevarla a cabo corresponde al *Administrador de proyecto*⁸.

Finalmente, en la verificación 3 el *grupo de AC* revisa y/o audita las actividades y productos de trabajo para PPS y reporta resultados. Esta verificación está representada con la actividad de “*Revisar y/o auditar actividades y producto de trabajo*” compartida entre el *Administrador de proyecto*, por ser el responsable de la PPS, y el *Grupo de AC*, como lo indica explícitamente el texto.

⁸ De las subprácticas de la Verificación 2 podemos deducir que las actividades de la PPS se revisan con el *Administrador de proyecto* y los grupos involucrados en estas actividades. Pero por legibilidad en el diagrama omitimos a estos grupos

Capítulo 5

Seguimiento y control del proyecto de software

En el seguimiento y control del proyecto se da seguimiento y se revisa los logros y resultados del software contra las estimaciones, compromisos, y planes documentados, y en caso de ser necesario se ajustan estos planes tomando como base los logros y resultados reales.

En este capítulo presentaremos la traducción de esta área clave de procesos [CMM, 1994], posteriormente iniciamos la descripción del modelo gráfico con la presentación de los grupos participantes y la concluimos con la explicación de los diagramas de actividades, que constituyen el modelo gráfico.

5.1. Modelo textual del Seguimiento y Control del Proyecto de Software (SCPS)

El propósito de Seguimiento y Control del Proyecto de Software (SCPS) es proporcionar una visión adecuada del avance real, de suerte que la administración pueda tomar acciones efectivas cuando el cumplimiento del proyecto de software se desvíe significativamente de los planes.

El Seguimiento y Control del Proyecto de Software involucra dar seguimiento y revisar los logros y resultados del software contra las estimaciones, compromisos, y planes documentados, y ajustar estos planes basados en los logros y resultados reales.

Un plan documentado para el proyecto de software (es decir, el plan de desarrollo de software, como se describe en la área clave de procesos de Planeación del Proyecto de Software) se usa como la base para dar seguimiento a las actividades del software, comunicar su estado y revisar planes. La administración controla las actividades de software. El avance se determina comparando el tamaño de software, esfuerzo, costo, y calendario reales con el plan cuando los productos de trabajo seleccionados son terminados y en las fechas y/o eventos seleccionados. Cuando se determina que los planes del proyecto de software no se cumplen, se toman acciones correctivas. Estas acciones podrían incluir revisiones al plan de desarrollo de software para reflejar los logros reales y replanear el trabajo restante o tomar acciones para mejorar el cumplimiento.

Metas

Meta 1

Dar seguimiento a los resultados y desempeño reales contra los planes de software.

Meta 2

Tomar y administrar acciones correctivas, cuando los resultados y desempeño reales se desvían significativamente de los planes del software, hasta su resolución.

Meta 3

Acordar los cambios a los compromisos del software por los grupos e individuos involucrados.

Compromisos

Compromiso 1

Se designa un administrador de proyecto de software que sea responsable de las actividades y resultados del proyecto de software.

Compromiso 2

El proyecto sigue una política organizacional escrita para administrar el proyecto de software.

Esta política típicamente especifica que:

1. Un plan de desarrollo de software documentado se usa y se mantiene, como la base para dar seguimiento al proyecto de software.
2. El administrador de proyecto siempre está informado del estado y de los problemas del proyecto de software.
3. Cuando el plan de software no está lográndose, ni ajustándose al desempeño y tampoco a los planes se toman acciones correctivas.
4. Los cambios a los compromisos se realizan con la participación y acuerdo de los grupos involucrados.

Ejemplos de grupos involucrados:

ingeniería de software (incluye todos los subgrupos; por ejemplo diseño de software),
estimación de software,
ingeniería de sistemas,
pruebas del sistema,
aseguramiento de calidad de software,
administración de configuraciones de software,
administración de contratos, y
soporte de documentación.

5. El administrador gerencial revisa todos los cambios a los compromisos y los nuevos compromisos del proyecto de software hechos a individuos y a grupos externos a la organización.

Habilidades

Habilidad 1

Se documenta y aprueba un plan de desarrollo de software para el proyecto de software.

Referirse a las actividades 6 y 7 del área clave de procesos de Planeación del Proyectos de Software para prácticas que cubre el plan de desarrollo de software.

Habilidad 2

El administrador de proyecto de software asigna explícitamente las responsabilidades para los productos de trabajo y actividades de software.

Las responsabilidades asignadas cubren:

1. Los productos de trabajo de software a ser desarrollados o los servicios a ser proporcionados.
2. El esfuerzo y costo para estas actividades del software.
3. El calendario para estas actividades de software.
4. El presupuesto para estas actividades de software.

Habilidad 3

Se proveen los recursos y financiamiento adecuados para dar seguimiento al proyecto de software.

1. Los administradores de software y los líderes de las tareas de software se les asignan responsabilidades para dar seguimiento al proyecto de software.
2. Las herramientas de soporte al seguimiento de software se hacen disponibles.

Ejemplos de herramientas de soporte:
hojas de cálculo y
programas de calendarización/planeación del proyecto.

Habilidad 4

Se capacitan a los administradores de software para administrar los aspectos técnicos y del personal del proyecto del software.

Por ejemplo la capacitación incluye:
administrar los proyectos técnicos;
seguir y controlar el tamaño, esfuerzo, costo, y calendario del software; y
administrar personal.

Habilidad 5

Los administradores de software de primera-línea reciben orientación en los aspectos técnicos del proyecto del software.

La orientación incluye:
los procedimientos y estándares de ingeniería de software del proyecto, y
el dominio de aplicación del proyecto.

Actividades

Actividad 1

Se usa un plan de desarrollo de software documentado para dar seguimiento a las actividades y comunicar el estado del software.

Referirse a la Actividad 7 del área clave de procesos de Planeación del Proyecto de Software que cubren el contenido del plan de desarrollo de software.

Este plan de desarrollo de software está:

1. Actualizado de acuerdo al avance del trabajo para reflejar los logros, en especial cuando un evento/fecha previamente seleccionada acontece.
2. Disponible a:
 - el grupo de ingeniería de software (incluyendo a todos los subgrupos, por ejemplo, diseño de software),
 - los administradores de software,
 - el administrador de proyecto,
 - administrador gerencial, y
 - otros grupos involucrados.

Actividad 2

Se revisa el plan de desarrollo de software del proyecto de acuerdo a un procedimiento documentado.

Referirse a la Actividad 6 del área clave de procesos de Planeación de Proyectos de Software para las prácticas que cubren las actividades para producir el plan de desarrollo de software.

Este procedimiento típicamente especifica que:

1. El plan de desarrollo de software se revisa, para incorporar detalles y cambios al plan, particularmente cuando los planes cambian significativamente.

Interdependencias entre los requisitos del sistema, restricciones de diseño, recursos, costos, y calendario necesitan ser reflejadas en todos los cambios del plan.

2. El plan de desarrollo de software se actualiza para incorporar todos los compromisos nuevos del proyecto de software y cambios a los compromisos.
3. El plan de desarrollo de software se examina en cada revisión.
4. El plan de desarrollo de software se administra y se controla.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos Administración de Configuraciones de Software.

Actividad 3

Se revisan los compromisos del proyecto de software y los cambios a los compromisos hechos a individuos y grupos externos a la organización con el administrador gerencial de acuerdo a un procedimiento documentado.

Actividad 4

Los cambios aprobados a los compromisos que afectan al proyecto de software se comunican a los miembros del grupo de ingeniería y otros grupos relacionados al software.

Ejemplos de otros grupos relacionados:
aseguramiento de calidad de software,
administración de configuraciones de software, y
soporte de documentación.

Actividad 5

Se da seguimiento al tamaño de los productos de trabajo de software (o tamaño de los cambios a los productos de trabajo de software), y se toman acciones correctivas en caso necesario.

Referirse a la Actividad 9 del área clave de procesos de Planeación de Proyectos de Software para las prácticas que cubren el cálculo de las estimaciones de tamaño.

1. Se da seguimiento al tamaño de todos los principales productos de trabajo (o tamaño de los cambios).
2. El tamaño real de código (generado, probado completamente, y entregado) se compara con las estimaciones documentadas en el plan de desarrollo de software.
3. Las unidades reales de documentación entregadas se comparan con las estimaciones documentadas en el plan de desarrollo de software.
4. El tamaño global proyectado de los productos de trabajo de software (estimaciones combinadas con las reales) se refina, se controla, y se ajusta con regularidad.
5. Los cambios a la estimación del tamaño de los productos de trabajo del software que afectan a los compromisos del software se negocian con los grupos involucrados y se documentan.

Actividad 6

Se da seguimiento al esfuerzo y costos del proyecto de software, y se toman acciones correctivas en caso necesario.

Referirse a la Actividad 10 del área clave de procesos de Planeación de Proyectos de Software para las prácticas que cubren el cálculo de las estimaciones de costo.

1. Los gastos reales de esfuerzo, costos fuera de tiempo y de trabajo terminado se compara con las estimaciones documentadas en el plan de desarrollo para identificar subestimaciones y sobreestimaciones.
2. Los costos de software se calculan y se comparan con las estimaciones documentadas en el plan de desarrollo de software.
3. El esfuerzo y la asignación de personal se comparan con las estimaciones documentadas en el plan de desarrollo de software.
4. Los cambios de personal y de otros costos de software, que afectan a los compromisos, se negocian con los grupos involucrados y se documentan.

Actividad 7

Se da seguimiento a los recursos computacionales críticos del proyecto, y se toman acciones correctivas en caso necesario.

Referirse a la Actividad 11 del área clave de procesos de Planeación de Proyectos de Software para las prácticas que cubren las estimaciones de los recursos computacionales.

1. El uso real y proyectado de los recursos computacionales críticos del proyecto se analizan y se comparan con las estimaciones para cada componente principal de software como se documentó en el plan de desarrollo de software.
2. Los cambios a las estimaciones de los recursos computacionales críticos que afectan a los compromisos de software se negocian con los grupos involucrados y se documentan.

Actividad 8

Se da seguimiento al calendario del proyecto de software, y se toman acciones correctivas en caso necesario.

Referirse a la Actividad 12 del área clave de procesos de Planeación de Proyectos de Software para las prácticas que cubren la definición del calendario.

1. La conclusión de las actividades de software, *milestones*¹, y otros compromisos se comparan con el plan de desarrollo de software.
2. Las consecuencias por conclusión tardía o temprana de las actividades de software, *milestones*¹, y otros compromisos se evalúan, para conocer el impacto en las actividades y *milestones*¹ futuros.
3. Las revisiones al calendario de software que afectan a los compromisos se negocian con los grupos involucrados y se documentan.

Actividad 9

Se da seguimiento a las actividades técnicas de ingeniería de software, y se toman acciones correctivas en caso necesario.

1. Los miembros del grupo de ingeniería de software reportan con regularidad su estado técnico a su administrador de primera línea.
2. El contenido de los entregables de software se compara con los planes documentados del plan de desarrollo de software.
3. Los problemas identificados en algún producto de trabajo de software se reportan y se documentan.
4. Los reportes de problemas se siguen hasta su cierre.

Actividad 10

Se da seguimiento a los riesgos del software asociados con costo, recursos, calendario y aspectos técnicos del proyecto.

Referirse a la Actividad 13 del área clave de procesos de Planeación del Proyecto de Software para las prácticas que cubren la identificación de riesgos.

1. Las prioridades de los riesgos y las contingencias para los riesgos se ajustan como información adicional para ser disponible.
2. Se revisan con regularidad las áreas de alto riesgo con el administrador de proyecto.

¹ *Milestone*: Evento o fecha previamente acordada.

Actividad 11

Se registran los datos de mediciones reales y los datos de replaneación para el proyecto de software.

Referirse a la Actividad 15 del área clave de procesos de Planeación del Proyecto de Software para las prácticas que cubren el registro de datos del proyecto.

1. La información registrada incluye las estimaciones y la información necesaria para reconstruir las estimaciones y verificar su validez.
2. Los datos de replaneación del software se administran y controlan.
3. Los datos de planeación, de replaneación, y de mediciones reales se archivan para su uso en proyectos en curso o futuros.

Actividad 12

El grupo de ingeniería de software realiza revisiones periódicas internas para dar seguimiento al avance técnico, planes, realización, y resultados contra el plan de desarrollo de software.

Estas revisiones se realizan entre:

1. Los administradores de primera línea y líderes de tareas de software.
2. El administrador de proyecto de software, los administradores de software de primera línea, y otros administradores de software.

Actividad 13

Se realizan revisiones formales para conocer el cumplimiento y resultados del proyecto de software en un evento o fecha previamente seleccionados de acuerdo a un procedimiento documentado.

Estas revisiones:

1. Se planean en fechas significativas del calendario del proyecto de software, tal como el inicio y término de etapas seleccionadas.
2. Se realizan con el cliente, el usuario final, y grupos involucrados dentro de la organización.

Los usuarios finales, en estas prácticas, son los clientes designados como usuarios finales o representantes de los usuarios finales.

3. Usan materiales que se revisan y se aprueban por los administradores de software responsables.
4. Analizan los compromisos, planes y estado de las actividades de software.
5. Terminan con la identificación y documentación de problemas, acciones a realizar, y decisiones significativas.
6. Analizan los riesgos del proyecto de software.
7. Terminan con el refinamiento del plan de desarrollo de software, en caso necesario.

Mediciones y análisis

Medición 1

Las mediciones se definen y se usan para determinar el estado de las actividades del seguimiento y control del proyecto de software.

Ejemplos de mediciones:

esfuerzo y otros recursos empleados para realizar las actividades de seguimiento y control; y cambios del plan de desarrollo de software, que incluyen cambios a las estimaciones de tamaño de los productos de trabajo, estimaciones de costo de software, estimaciones de recursos computacionales críticos, y calendario.

Verificaciones

Verificación 1

Las actividades para el seguimiento y control del proyecto de software se revisan periódicamente 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.

1. Se revisa el cumplimiento técnico, de costo, del personal y de calendario.
2. Se analizan los conflictos y los problemas no resueltos en los niveles inferiores.
3. Se analizan los riesgos del proyecto de software.
4. Se asignan, se revisan y se siguen las acciones acordadas hasta su cierre.
5. Se prepara un reporte resumido de cada reunión y se distribuye a los grupos involucrados.

Verificación 2

Las actividades para el seguimiento y control del proyecto de software se revisan con el administrador de proyecto periódicamente y/o a causa de un evento.

1. Los grupos involucrados son representados.
2. El cumplimiento técnico, de costo, del personal y de calendario se revisa contra el plan de desarrollo de software.
3. Se revisa el uso de los recursos computacionales críticos; las estimaciones actuales y el uso real de estos recursos computacionales críticos se comparan con las estimaciones originales y se informa el resultado.
4. Se analizan las dependencias entre grupos.

5. Se analizan los conflictos y los problemas no resueltos en los niveles inferiores.
6. Se analizan los riesgos del proyecto de software.
7. Se asignan, se revisan y se siguen las acciones acordadas hasta su cierre.
8. Se prepara un reporte resumido de cada reunión y se distribuye entre los grupos involucrados.

Verificación 3

El grupo de aseguramiento de calidad de software revisa y/o audita las actividades y productos de trabajo para seguir y controlar el proyecto del software y reporta resultados.

Referirse al área clave de procesos de Aseguramiento de Calidad de Software.

Las revisiones y/o auditorias verifican, al menos:

1. Las actividades de examinar y revisar compromisos.
2. Las actividades de revisar el plan de desarrollo de software.
3. El contenido del plan de desarrollo de software revisado.
4. Las actividades para dar seguimiento, al costo, calendario, riesgos, restricciones de diseño y técnicas, funcionalidad y desarrollo del proyecto de software.
5. Las actividades para llevar a cabo las revisiones técnicas planeadas y de administración.

5.2. Modelo gráfico del Seguimiento y Control del Proyecto de Software (SCPS)

El modelado gráfico se iniciará con una identificación, en todo el texto original, de los grupos de actores participantes en el proceso de SCPS y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

5.2.1. Identificación de los grupos participantes

En el modelo textual del seguimiento y control del proyecto de software hemos identificado y resaltado en letras itálicas a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentaremos una breve definición de cada uno de ellos.

Administrador de proyecto: Es el responsable de las actividades y resultados 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 SPCS que se realizan en forma periódica o a causa de un evento.

Administrador 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.

Administrador de software de primera línea: Un administrador, que tiene la responsabilidad, en forma directa, del personal (incluye proveer dirección técnica, y dirigir personal y definir su salario) y de las actividades de una unidad organizacional simple (por ejemplo un departamento o equipo de proyecto) de ingenieros de software o personal relacionado.

Administrador gerencial: Un rol de administración en 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 tendría la responsabilidad de varios proyectos a la vez. Participa en la revisión periódica de las actividades del SCPS.

Grupo de Ingeniería de Software (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 proceso de ingeniería de software no están incluidos en el grupo de ingeniería de software.

Este grupo realiza revisiones periódicas internas para dar seguimiento al avance técnico, planes, realización, y resultados contra el plan de desarrollo de software.

Grupo de Aseguramiento de Calidad (AC). Es el responsable de revisar y/o auditar las actividades y productos de trabajo para seguir y controlar el proyecto de software y reporta resultados.

Organización. Es el conjunto de los grupos mencionados anteriormente.

5.2.2. Diagramas de actividades

Compromisos

En esta área clave contamos con dos compromisos. El primero establece un liderazgo y el segundo una política organizacional.

El compromiso 1 dice que se designa un administrador de proyecto de software, quien será responsable de las actividades y resultados del proyecto de software. Este compromiso lo representamos con la actividad "Asignar administrador de proyecto de software" (Fig. 5.1), y se propone como una actividad realizada por el *Administrador gerencial*, aunque en el texto no lo dice explícitamente pero creemos que es una aseveración congruente.

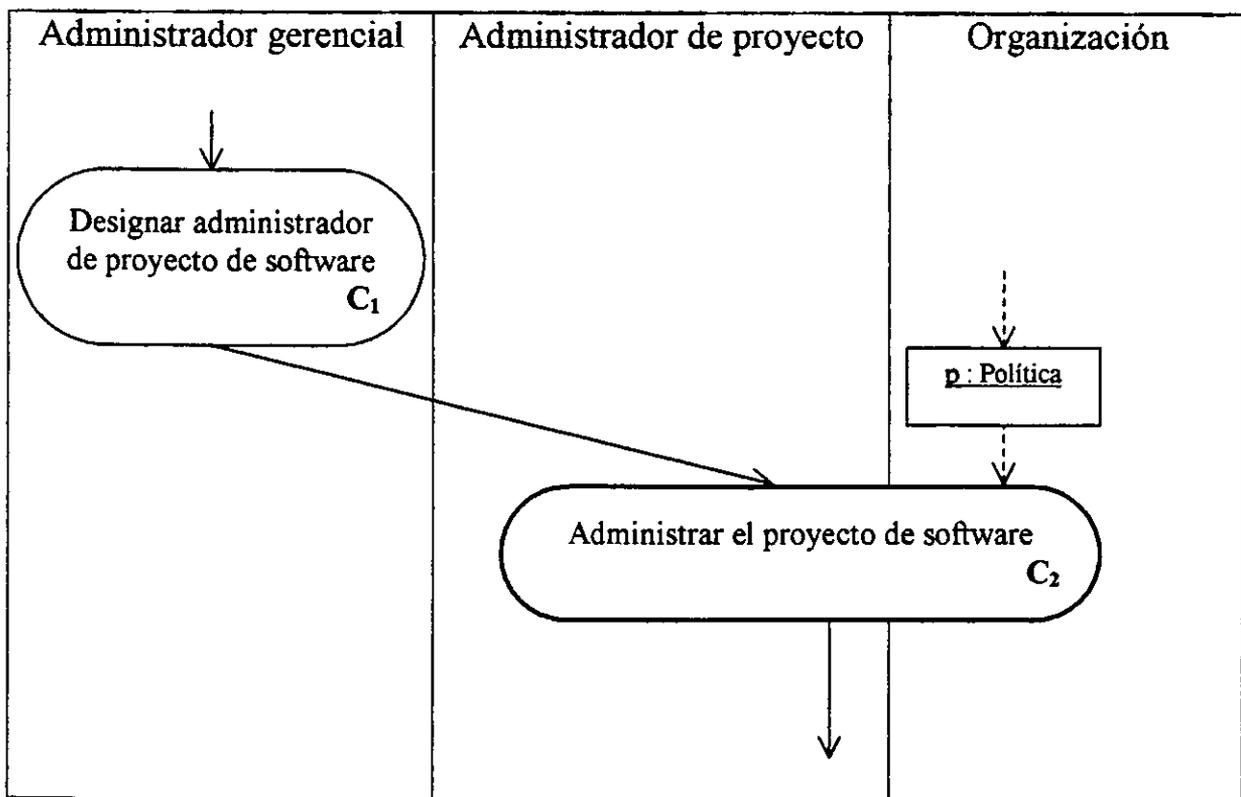


Fig. 5.1: Compromisos

Analizando el texto del compromiso 2, se deduce que la organización tiene una política escrita que guíe la administración del proyecto de software. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Fig. 5.1) como un objeto *p:Política*. Este objeto fue probablemente generado por una actividad de definición de política, la cuál, según el texto del compromiso, queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir la política para administrar el proyecto de software queda expresado colocando a *p:Política* como objeto de entrada de la actividad de *Administrar el proyecto de software*. La actividad *Administrar el proyecto de software* se representa en el diagrama, como una actividad compartida entre el *Administrador de proyecto* y la *Organización*. Consideramos que ambos realizan esta actividad porque un grupo depende del otro para realizar efectivamente esta actividad.

La Fig. 5.1 ofrece una síntesis gráfica del área clave de procesos SCPS y, a la vez, nos da una vista global de las relaciones entre el *Administrador* y el resto de la *Organización* con respecto a la política de SCPS y su implicación en el seguimiento y control de los proyectos.

Las características comunes siguientes: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben de tomar para administrar el proyecto de software. Para modelar este hecho, la actividad de *Administrar el proyecto de software* de la Fig. 5.1 puede desglosarse en otro diagrama presentado en la Fig. 5.2.

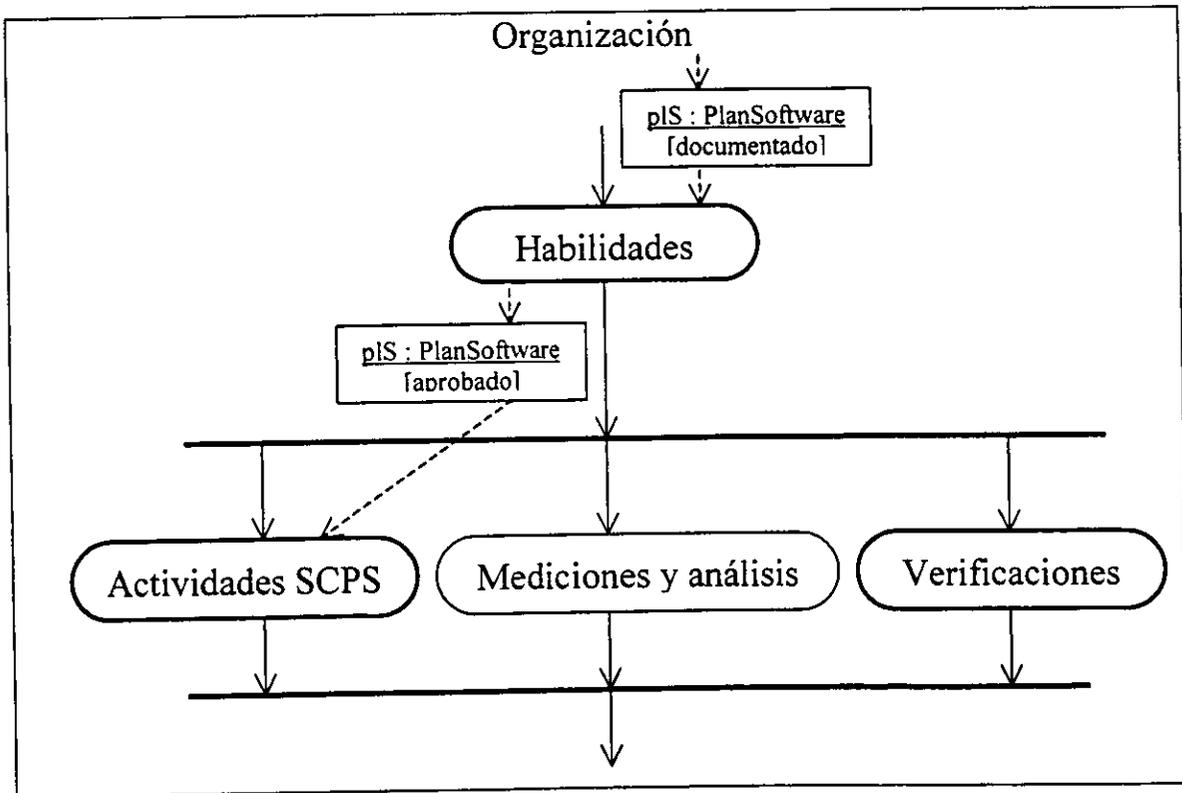


Fig. 5.2: Administrar el proyecto de software

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 seguimiento y control del proyecto de software. Por otro lado, las actividades de verificación y medición con su análisis, son simultáneas a la de SCPS.

Habilidades

La habilidad 1 del modelo textual se refiere a la aprobación de un plan de desarrollo para el proyecto de software. Este texto hace referencia a un plan de desarrollo de software, el cual se representa en el diagrama de la Fig. 5.3 con el objeto *plS:PlanSoftware*. Este objeto es creado por la actividad “*Realizar plan de desarrollo*” en estado de borrador. Posteriormente es modificado por la actividad “*Documentar plan de desarrollo*” y lo cambia a estado de documentado, pero aun en estado de borrador. Ambas actividades, tanto la que creó como la que documentó a *plS:PlanSoftware* son las actividades 6 y 7 del área clave de procesos de Planeación de Proyectos de Software (PPS), las cuales modelamos en la figura 4.5 del capítulo anterior.

La habilidad 1 queda plasmada en el diagrama de actividades de la Fig. 5.3 por la actividad “*Aprobar plan de desarrollo de software*”. Esta actividad recibe como entrada al objeto *plS:PlanSoftware* y le cambia el estado a aprobado. A pesar que en el texto no dice explícitamente quien realiza esta actividad, nos parece congruente que sea realizada en forma compartida por el *Administrador gerencial* y el *Administrador de proyecto*.

Una vez aprobado el plan de desarrollo, éste es utilizado por el administrador de proyecto para asignar responsabilidades para los productos y actividades de software. Lo anterior es especificado por la habilidad 2 y lo representamos por la actividad “*Asignar responsabilidades*” (Fig. 5.3). El flujo del objeto *plS:PlanSoftware* a la actividad “*Asignar responsabilidades*” representa su uso para realizar esta actividad, pero el objeto del plan no es modificado. Como mencionamos al inicio de este párrafo, el modelo textual establece explícitamente que esta actividad es realizada por el *Administrador de proyecto*, hecho que plasmamos en la Fig. 5.3.

La habilidad 3 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo del *Administrador de proyecto*. Esta habilidad la representamos con la actividad “*Asignar recursos y financiamiento para SCPS*” 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 4 especifica que el *Administrador de software* se capacita en administración de aspectos técnicos y del personal del proyecto de software, mientras que la habilidad 5 dice que el *Administrador de software de primera línea* recibe orientación en los aspectos técnicos del proyecto de software. La primera habilidad la representamos con la actividad “*Capacitar en SCPS*” y la segunda con la actividad “*Recibir orientación*”.

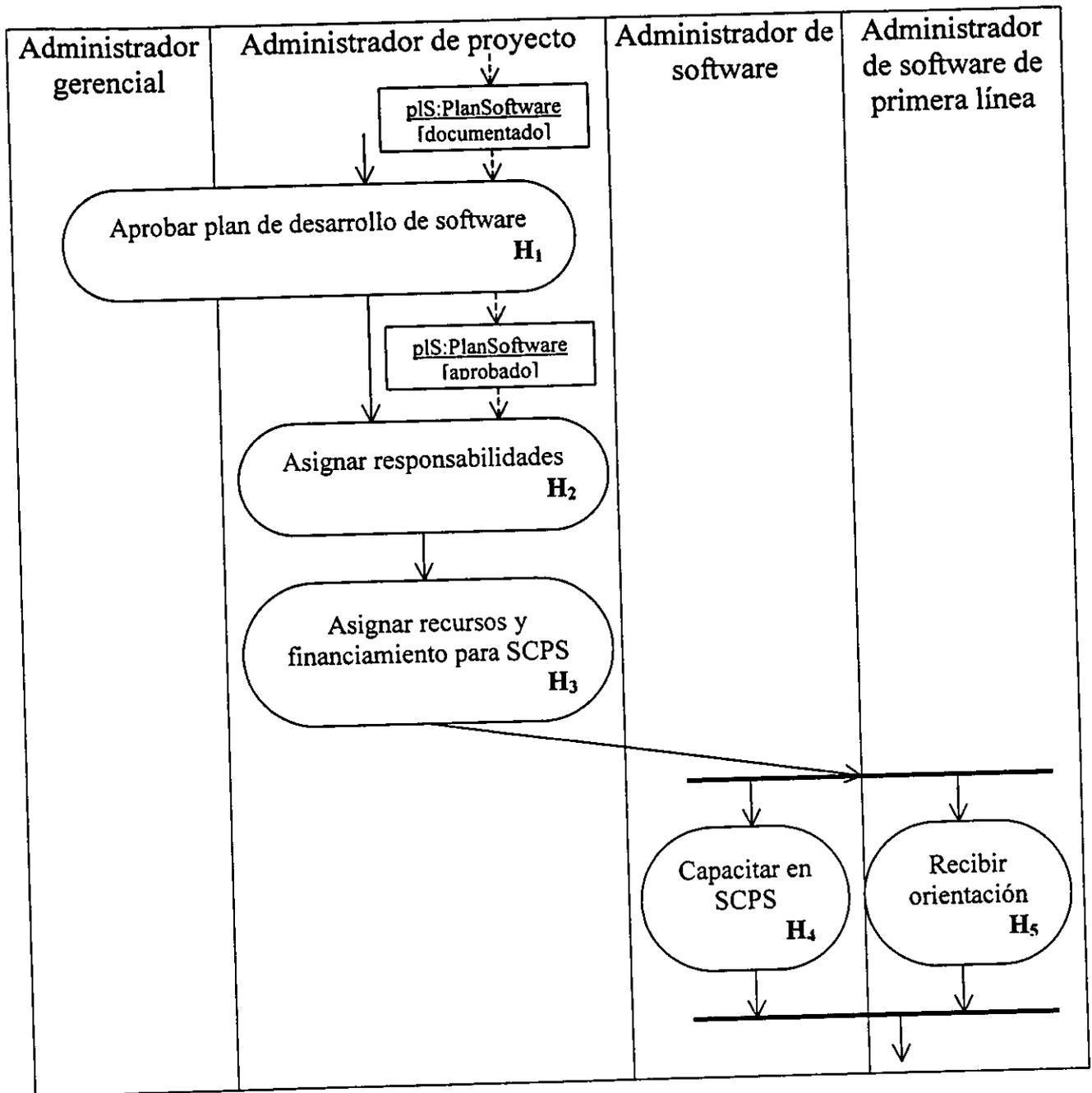


Fig. 5.3: Habilidades

El diagrama de la Fig. 5.3 presenta las actividades a realizarse para adquirir las habilidades para el SCPS, donde cada una de las habilidades es mapeada en forma directa a la actividad que le 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 dice que un plan de desarrollo de software documentado se usa para dar seguimiento a las actividades y comunicar el estado del software. La explicación de cómo representamos esta actividad la haremos en partes. En la primera nos referimos a la representación explícita, es decir en el diagrama de la Fig. 5.4, mapeamos en forma directa a la actividad "Dar seguimiento a actividades y comunicar estado del software". Esta actividad recibe como entrada al objeto *plS:PlanSoftware*, el cual proviene de la actividad "Aprobar plan de desarrollo de software" de la sección de **Habilidades**² (Fig. 5.3). La segunda parte consiste en explicar la representación implícita de esta actividad, de acuerdo al texto del modelo correspondiente a **Actividades**, consideramos que de la actividad 5 a la 10 (seguimiento del tamaño de los productos de trabajo, del esfuerzo, de los recursos computacionales críticos, del calendario, de las actividades técnicas de ingeniería de software y de los riesgos del software) están contenidas en la actividad 1. Esto lo representamos en diagrama (Fig. 5.4) con la expresión $A_5 \dots A_{10} \in A_1$.

La actividad 2 especifica que el plan de desarrollo se revisa de acuerdo a un procedimiento documentado. Esta actividad la representamos en el diagrama con "Revisar plan de desarrollo de software", la cual tiene como entrada al objeto *plS:PlanSoftware*, que será revisado, y además al objeto *pr:Proce*³, se refiere al procedimiento que de alguna forma ya se cuenta con él.

La actividad 3 y 4 se refieren a los compromisos y a sus posibles cambios. La primera dice que se revisan los compromisos del proyecto de software y los cambios a los compromisos hechos a individuos y grupos externos a la organización, con el administrador gerencial de acuerdo a un procedimiento documentado. Esta actividad se representa en el diagrama con "Revisar compromisos y cambios a los compromisos", que de acuerdo al texto, es realizada por el *Administrador gerencial* en forma compartida con el *Administrador de proyecto*. Si esta actividad va a revisar los cambios a los compromisos, de alguna forma debe de existir una solicitud de cambio. Pero para no complicar este diagrama, sólo representaremos este hecho al incluir como entrada a esta actividad un objeto *ca:Cambio* en estado de solicitado. Y si este cambio es aceptado esta actividad tendrá como salida a *ca:Cambio* en estado de aprobado.

Una vez revisados y aprobados los cambios a compromisos se comunican al *grupo de IS* y a otros grupos relacionados al software, esto lo especifica la actividad 4. En el diagrama (Fig. 5.4) lo plasmamos con la actividad "Comunicar cambios a compromiso". Estas dos últimas actividades se realizan en forma secuencial, y este flujo de actividades se realiza en forma paralela a las actividades "Dar seguimiento a actividades y comunicar estado del software", "Revisar plan de desarrollo de software", "Registrar datos de mediciones reales y de replaneación" y "Realizar revisiones", estas dos últimas a continuación las explicaremos

² En la sección de **Habilidades** se explica el origen del objeto *plS:PlanSoftware*.

³ En el diagrama de actividades de la Fig. 5.4, modelamos el texto "de acuerdo a un procedimiento documentado" con el objeto *pr:Proce*. Donde el nombre de la clase *Proce* es una abreviación de la clase *Procedimiento*.

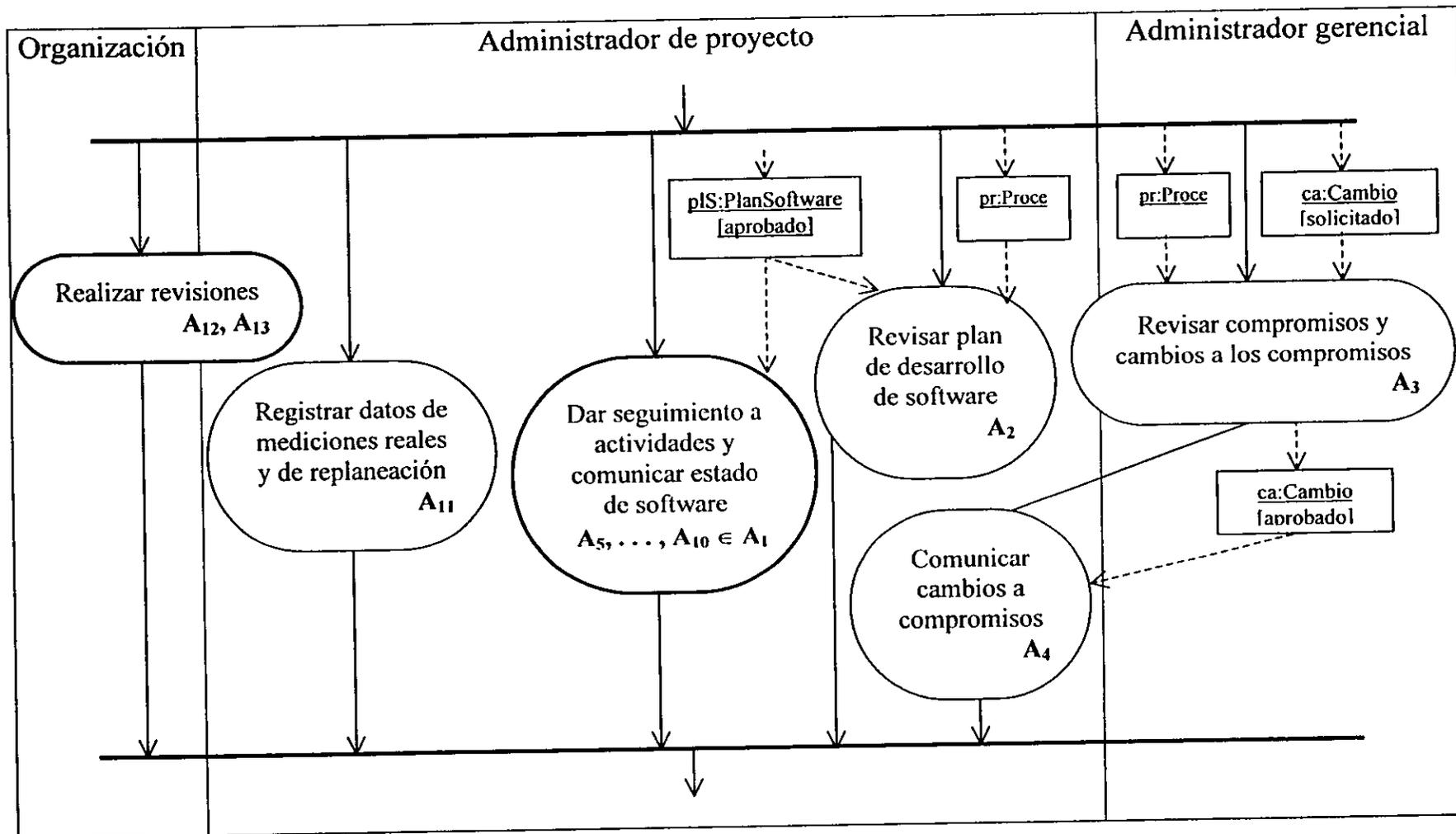


Fig. 5.4: Actividades SCPS

En el texto de la actividad 11 establece que se registran los datos de mediciones reales y de replaneación para el proyecto de software, esta actividad se mapea en forma directa a “Registrar datos de mediciones reales y de replaneación”. A pesar de que no se dice explícitamente en el texto es razonable pensar que es una actividad propia del *Administrador de proyecto*.

La última actividad por explicar de este diagrama (Fig. 5.4) es “Realizar revisiones”. En esta actividad agrupamos las actividades 12 y 13 por tratarse de actividades similares. Por lo que el desglose de esta actividad lo presentaremos en el siguiente diagrama (Fig. 5.5).

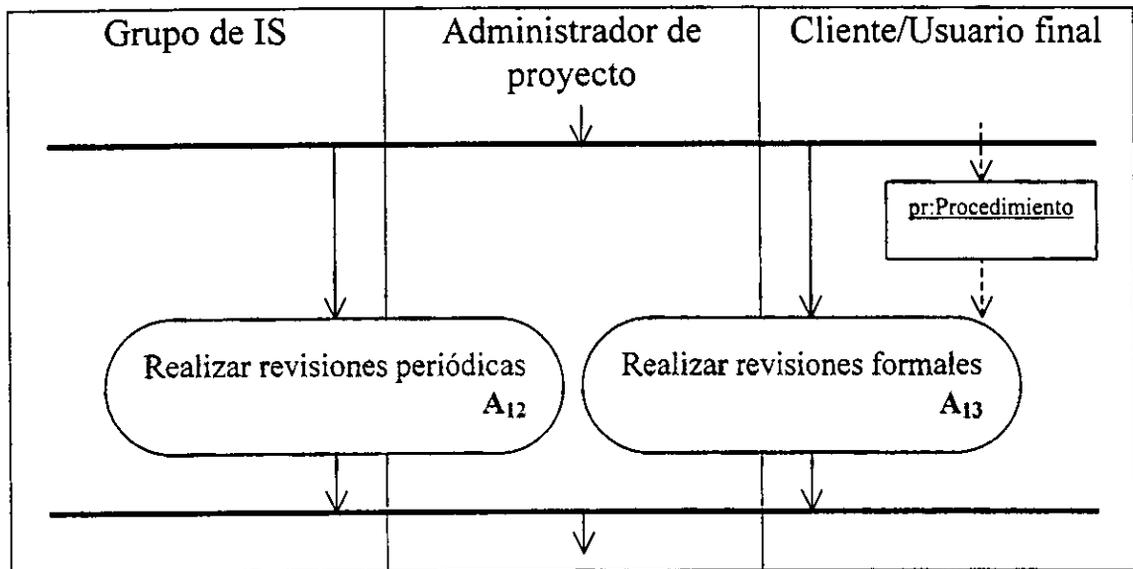


Fig.5.5: Realizar revisiones

La actividad 12 dice que el *grupo de IS* lleva acabo revisiones periódicas para dar seguimiento al avance técnico, planes, realización, y resultados contra el plan de desarrollo de software. Esta actividad la representamos en el diagrama (Fig. 5.5) con la actividad “Realizar revisiones periódicas”. En el texto especifica explícitamente que esta actividad es realizada por el grupo de IS, pero por el contexto de esta área clave, proponemos que sea en forma compartida con el *Administrador de proyecto*.

La actividad 13 especifica que se realizan revisiones formales para conocer los logros y resultados del proyecto de software en un evento o fecha previamente seleccionados de acuerdo a un procedimiento documentado. Esta actividad la representamos en el diagrama con la actividad “Realizar revisiones formales”. La subpráctica nos dice que esta actividad se realiza con el cliente o usuario final, y nuevamente por el contexto, podemos deducir que se lleva acabo en forma compartida con el *Administrador de proyecto*.

El diagrama de actividades de la Fig. 5.6 desglosa “Dar seguimiento a actividades y comunicar estado del software”⁴. Como habíamos mencionado en este diagrama representaremos las actividades de la 5 a la 10.

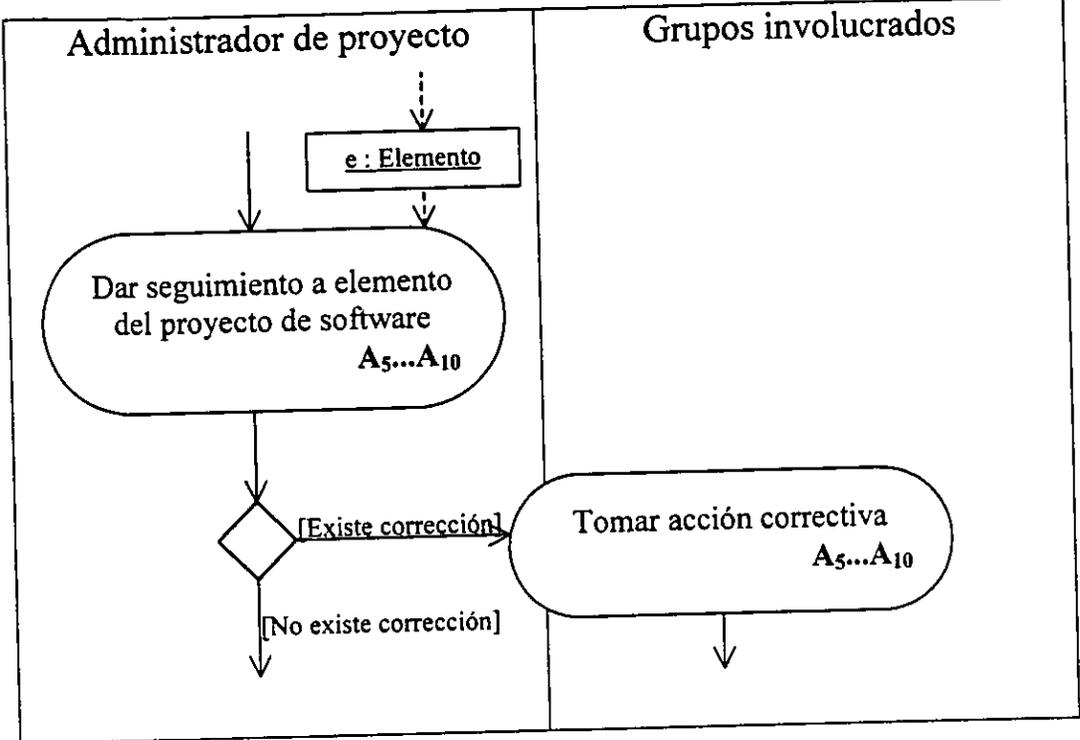


Fig. 5.6: Dar seguimiento a actividades y comunicar estado de software

Si observamos la estructura del texto de la actividad 5 a la 10, encontraremos que se trata de dar seguimiento a cada elemento del plan de desarrollo, y se toman acciones correctivas de ser necesario. La primera parte de esta estructura es representada por la actividad “Dar seguimiento a elemento del proyecto de software”, y la parte final la representamos con una selección. Si existe una corrección entonces se realiza la actividad “Tomar acción correctiva” por el grupo involucrado en caso contrario continuamos con el flujo de actividades. Los elementos a los que se les da seguimiento son: tamaño (A₅), esfuerzo y costos (A₆), recurso computacional (A₇), calendario (A₈), actividades técnicas de IS (A₉) y riesgo de software (A₁₀). Estos elementos los modelamos como una especialización de objetos y la observamos en la Fig. 5.7.

⁴ Esta actividad es del diagrama de actividades de la Fig. 5.4.

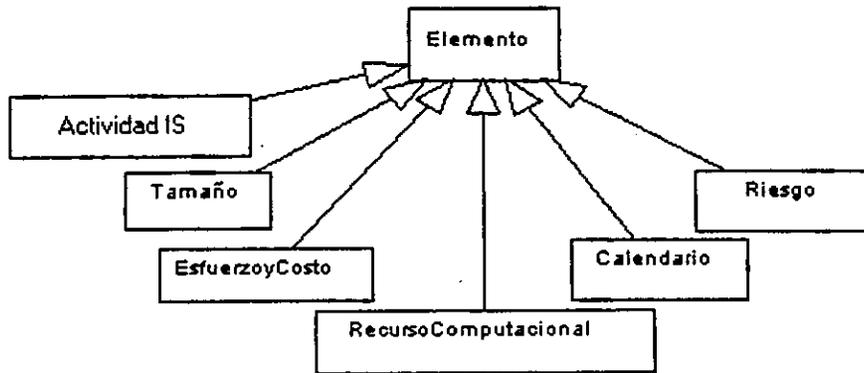


Fig. 5.7: Elemento.

Mediciones y análisis

El texto correspondiente a la medición y análisis es muy escueto. Se dice que deben de definirse y usar mediciones para determinar el estado de las actividades del SCPS. Sin embargo, falta precisar quienes son responsables por efectuar dichas actividades. Conociendo todo el contexto del modelo textual de SCPS nos atrevemos a proponer (Fig.5.8) que la actividad de “*Mediciones y análisis*” es una actividad compartida entre el *Administrador de proyecto*, que debe ser responsable por recaudar datos de las mediciones, y el *Administrador gerencial* y el *Grupo de Aseguramiento de Calidad (AC)*, quienes utilizan estos datos para determinar el estado de las actividades del SCPS. Previamente, en la Fig. 5.2, ya quedó plasmado el paralelismo de las actividades de medición y análisis con las de SCPS y de las verificaciones.

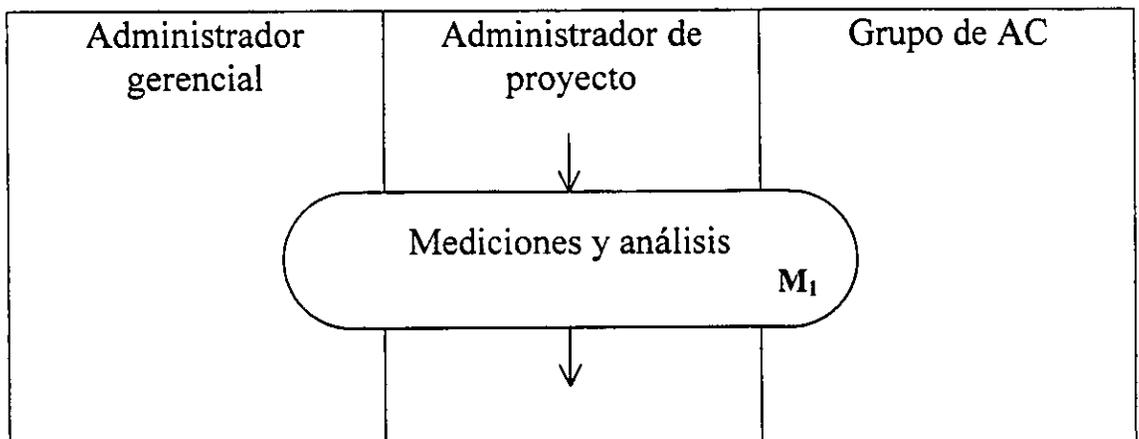


Fig. 5.8: Mediciones y análisis

El diagrama, que proponemos (Fig. 5.8), le añade la interpretación de las responsabilidades a la actividad de "Mediciones y análisis" con respecto al modelo textual, lo que a nuestro juicio facilita entender en que grupos debe de recaer.

Verificaciones

La verificación 1 especifica que las actividades para el seguimiento y control del proyecto de software se revisan periódicamente con el *Administrador gerencial*. Modelamos esta práctica con la actividad "Revisar actividades SCPS" realizada en forma compartida entre el *Administrador de proyecto*, que es el responsable de esta área clave, y el *Administrador gerencial*, como lo indica el modelo textual (Fig. 5.9).

La verificación 2 dice que las actividades para el seguimiento y control del proyecto se revisan con el *Administrador de proyecto* periódicamente y/o a causa de un evento. Esta verificación la mapeamos en forma directa a "Revisar actividades SCPS periódicamente y/o a causa de un evento", cuya responsabilidad de llevarla a cabo corresponde al *Administrador de proyecto*⁵.

Finalmente, en la verificación 3 el *grupo de AC* revisa y/o audita las actividades y productos de trabajo para SCPS y reporta resultados. Esta verificación la representamos con la actividad de "Revisar y/o auditar actividades y producto de trabajo" compartida entre el *Grupo de AC* y *Administrador de proyecto*.

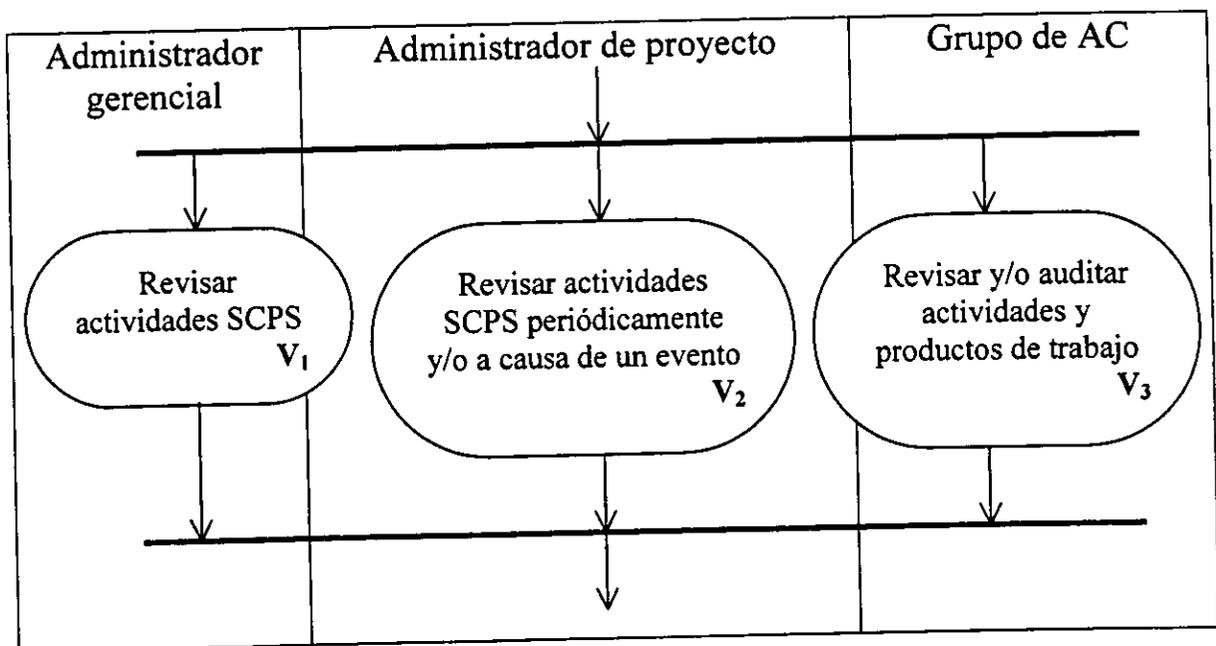


Fig. 5.9: Verificaciones

⁵ De las subprácticas de la Verificación 2 podemos deducir que las actividades del seguimiento y control del proyecto de software se revisan con el *Administrador de proyecto* y los grupos involucrados en estas actividades. Pero por legibilidad en el diagrama omitimos a estos grupos.

Capítulo 6

Aseguramiento de calidad de software

El aseguramiento de calidad provee a los administradores la visión adecuada del proceso usado por el proyecto de software y de los productos que se están construyendo. Las prácticas clave de esta área sugieren realizar revisiones y auditorías a los productos y actividades de software, para verificar que se está cumpliendo con los estándares y procedimientos correspondientes.

En este capítulo presentaremos la traducción de esta área clave de procesos [CMM, 1994], posteriormente iniciamos la descripción del modelo gráfico con la presentación de los grupos participantes y la concluimos con la explicación de los diagramas de actividades, que constituyen el modelo gráfico.

6.1. Modelo textual del Aseguramiento de Calidad de Software (AC)

El propósito del Aseguramiento de Calidad de Software (AC) es proveer a los administradores la visión adecuada del proceso usado por el proyecto de software y de los productos que se están construyendo.

El Aseguramiento de Calidad de Software involucra revisar y auditar los productos y actividades del software, para verificar que cumplen los procedimientos y estándares aplicables y proveer los resultados de estas revisiones y auditorías a los administradores de proyecto de software y a otros involucrados.

El grupo de aseguramiento de calidad de software trabaja en el proyecto de software durante las primeras etapas para establecer planes, estándares y procedimientos que añadirán valor al proyecto de software, y para satisfacer las restricciones del proyecto y las políticas de la organización. Con su participación para establecer los planes, estándares, y procedimientos, el grupo de aseguramiento de calidad de software ayuda a garantizar que éstos se ajusten a las necesidades del proyecto y verifica que serán útiles para realizar revisiones y auditorías durante el ciclo de vida del software. El grupo de aseguramiento de calidad de software revisa las actividades del proyecto y audita los productos de trabajo de software durante el ciclo de vida y provee a los administradores una visión adecuada con respecto al apego a los planes, estándares, y procedimientos establecidos.

Los problemas encontrados se tratan de resolver, en primera instancia, en el proyecto de software. Los problemas no resueltos en el proyecto son escalados, para su solución, por el grupo de aseguramiento de calidad a un nivel apropiado de la administración.

Esta área clave de procesos cubre las practicas que realiza el grupo de aseguramiento de calidad. Las prácticas identifican las actividades específicas y productos de trabajo que el grupo de aseguramiento de calidad de software revisa y/o audita, las cuales, generalmente están contenidas en la sección de Verificación de las otras áreas clave de procesos.

Metas

Meta 1

Planear las actividades de aseguramiento de calidad del software.

Meta 2

Verificar objetivamente el apego de productos y actividades del software a los estándares, procedimientos, y requisitos que apliquen.

Meta 3

Informar a los grupos e individuos involucrados de las actividades y resultados del aseguramiento de calidad del software.

Meta 4

Los problemas referentes al incumplimiento, que no pueden resolverse dentro del proyecto de software, son resueltos por el administrador gerencial.

Compromisos

Compromiso 1

El proyecto sigue una política organizacional escrita para llevar a cabo el aseguramiento de calidad de software.

Esta política típicamente especifica que:

1. El AC se realiza en todos los proyectos de software.
2. El grupo de AC tiene un canal de comunicación con el administrador gerencial, que es independiente:
 - del administrador de proyecto,
 - del grupo de ingeniería de software, y
 - de otros grupos relacionados al software.

Ejemplos de otros grupos relacionados:
administración de configuraciones de software, y
soporte de documentación.

La organización debe determinar la estructura organizacional que dará soporte a las actividades que requieren independencia, como el AC, de acuerdo a sus estrategias y ambiente de negocios.

La independencia debería:

dar libertad organizacional a los individuos para realizar el rol de AC. para que pueden ser "ojos y oídos" del administrador gerencial dentro del proyecto de software.

proteger a los individuos, que realizan el rol de AC, de la evaluación de su desempeño por el administrador del proyecto de software; y

dar confianza al administrador gerencial, que se le comunicará la información objetiva sobre los procesos y productos del proyecto de software.

3. El administrador gerencial revisa periódicamente las actividades y resultados del AC.

Habilidades

Habilidad 1

Existe un grupo responsable de coordinar y llevar a cabo el AC del proyecto (es decir el grupo de AC).

Un grupo es la colección de departamentos, administradores, e individuos que comparten la responsabilidad sobre un conjunto de tareas o actividades. Un grupo podría variar de un individuo asignado parte de su tiempo laboral, a varios individuos de diferentes departamentos asignados parte de su tiempo laboral, o a varios individuos asignados de tiempo completo. Las consideraciones, para formar un grupo, incluyen las tareas o actividades asignadas, el tamaño del proyecto, la estructura y cultura organizacional. Algunos grupos, como el grupo de aseguramiento de calidad, están enfocados en las actividades del proyecto, y otros, como el grupo de ingeniería de procesos de software, están enfocados en las actividades de la organización.

Habilidad 2

Se proveen los recursos y financiamiento adecuados para realizar las actividades de AC.

1. Se asigna un administrador con responsabilidades específicas para las actividades de AC del proyecto.
2. Un administrador gerencial, quién tiene conocimiento sobre el rol de AC y tiene la autoridad para tomar acciones de control adecuadas, es asignado para recibir y actuar en los puntos incumplidos del software.
 - Todos los administradores en la cadena de información del AC al administrador gerencial tienen conocimiento del rol, responsabilidades, y autoridad del AC.
3. Las herramientas de soporte de las actividades del AC son disponibles.

Ejemplos de herramientas de soporte:
estaciones de trabajo,
programas de bases de datos,
hojas de cálculo, y
herramientas para auditorías.

Habilidad 3

Los miembros del grupo de AC se capacitan para realizar sus actividades de AC.

Ejemplos de capacitación:
habilidades y prácticas de ingeniería de software;
roles y responsabilidades del grupo de ingeniería de software y otros grupos relacionados al software;
estándares, procedimientos y métodos para el proyecto de software;
dominio de aplicación del proyecto de software;
objetivos, procedimientos, y métodos;
participación del grupo de AC en las actividades de software;
uso eficaz de los métodos y herramientas de AC; y
comunicación entre personas.

Habilidad 4

Los miembros del proyecto de software reciben orientación sobre el rol, responsabilidades, autoridad e importancia del grupo de AC.

Actividades

Actividad 1

Se prepara un plan de AC para el proyecto del software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El plan de AC se desarrolla en las etapas iniciales de, y en paralelo con, la planeación global del proyecto.
2. Los grupos e individuos involucrados revisan el plan de AC.

Ejemplos de grupos e individuos involucrados:

administrador del proyecto de software;
otros administradores de software;
administrador de proyecto;
representante de AC del cliente;
administrador gerencial, a quien el grupo de AC informa sobre los problemas no resueltos; y
grupo de ingeniería de software (incluye todos los subgrupos, por ejemplo diseño de software así como los líderes de tareas de software).

3. Se administra y se controla el plan AC.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos de Administración de Configuraciones de Software.

Actividad 2

Las actividades del grupo de AC se realizan de acuerdo con el plan de AC.

El plan cubre:

1. Las responsabilidades y autoridad del grupo de AC.
2. Los requisitos de los recursos para el grupo de AC (incluyendo personal, herramientas, y facilidades).
3. El calendario y financiamiento de las actividades del grupo de AC del proyecto.
4. La participación del grupo de AC para establecer el plan de desarrollo de software, estándares, y procedimientos para el proyecto.
5. Las evaluaciones a ser realizadas por el grupo de AC.

Ejemplos de productos y actividades a ser evaluados:

software en funcionamiento y de soporte,
productos entregables y no entregables,
productos de software y los que no pertenecen al software (por ejemplo, documentos),
desarrollo de software y actividades de verificación de productos (por ejemplo, llevar a cabo los casos de prueba), y
actividades para crear el producto.

6. Las auditorías y revisiones a ser conducidas por el grupo de AC.
7. Los estándares y procedimientos del proyecto a ser usados como la base para las revisiones y auditorías del grupo de AC.
8. Los procedimientos para documentar y dar seguimiento a los problemas hasta su cierre.

Estos procedimientos podrían ser incluidos como parte del plan o como referencia a otros documentos donde ellos estén contenidos.

9. La documentación que el grupo de AC deberá generar.
10. El método y frecuencia para proveer retroalimentación al grupo de ingeniería de software y a otros grupos relacionados al software, sobre las actividades de AC.

Actividad 3

El grupo de AC participa en la preparación y revisión del plan de desarrollo, estándares y procedimientos del proyecto de software.

1. El grupo de aseguramiento de calidad consulta y revisa los planes, estándares, y procedimientos de acuerdo a:
 - el cumplimiento de la política organizacional,
 - el cumplimiento de los estándares y requisitos impuestos externamente,
 - lo adecuado de los estándares para ser usados en el proyecto,
 - los temas que deberían ser considerados en el plan de desarrollo de software, y
 - otras áreas definidas por el proyecto.
2. El grupo de AC verifica que los planes, estándares, y procedimientos están disponibles y puedan ser usados para revisar y auditar el proyecto de software.

Actividad 4

El grupo de AC revisa las actividades de ingeniería de software para verificar su cumplimiento.

1. Se evalúan las actividades, los estándares de software y procedimientos contra el plan de desarrollo de software.

Referirse a la característica común de Verificación en las otras áreas claves de procesos para las practicas que cubren las revisiones y auditorías realizadas por el grupo de AC.

2. Las desviaciones se identifican, documentan, y se siguen hasta su cierre.
3. Las correcciones se verifican.

Actividad 5

El grupo de AC audita los productos de trabajo de software designados para verificar su conformidad.

1. Los productos de software se evalúan antes de que sean entregados al cliente.
2. Los productos de trabajo de software se evalúan contra los estándares, procedimientos, y requisitos contractuales.
3. Las desviaciones se identifican, documentan, y se siguen hasta resolverse.
4. Las correcciones se verifican.

Actividad 6

El grupo de AC reporta periódicamente los resultados de sus actividades al grupo de ingeniería de software.

Actividad 7

Las desviaciones identificadas en las actividades y en los productos de trabajo de software se documentan y se manejan de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las desviaciones del plan de desarrollo de software, de los estándares y de procedimientos del proyecto se documentan y, cuando es posible, se resuelven con los líderes de tareas de software, administradores de software, o administradores de proyecto.
2. Las desviaciones del plan de desarrollo de software, de los estándares y de los procedimientos no resueltos con los líderes de tareas de software, administradores de software, o administrador de proyecto se documentan y se presentan al administrador gerencial asignado para recibir estos elementos de no conformidad.
3. Los elementos de no conformidad presentados al administrador gerencial se revisan periódicamente hasta que sean resueltos.
4. Se administra y se controla la documentación de los elementos de no conformidad.

Actividad 8

El grupo de AC dirige periódicamente revisiones de sus actividades y hallazgos con el personal de AC del cliente, en caso de que aplique.

Mediciones y análisis

Medición 1

Las mediciones se definen y se usan para determinar el estado de las actividades del AC.

Ejemplos de mediciones:

*milestones*¹ concluidos de las actividades de AC comparados con el plan;
trabajo terminado, esfuerzo y financiamiento utilizados en las actividades de AC comparados con el plan; y
número de productos auditados y de actividades revisadas comparado al plan.

Verificaciones

Verificación 1

Las actividades del AC se revisan periódicamente 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.

Referirse a la Verificación 1 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador gerencial.

Verificación 2

Las actividades del AC se revisan con el administrador del proyecto periódicamente y/o a causa de un evento.

Referirse a la Verificación 2 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador de proyecto.

Verificación 3

Los expertos independientes del grupo de AC revisan periódicamente las actividades y productos de trabajo de software del grupo de AC del proyecto.

¹ *Milestone*: Evento o fecha previamente acordada.

6.2. Modelo gráfico del Aseguramiento de Calidad de Software (AC).

El modelado gráfico se iniciará con una identificación, en todo el texto original, de los grupos de actores participantes en el proceso de AC y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

6.2.1. Identificación de los grupos participantes

En el modelo textual del aseguramiento de calidad de software hemos identificado y resaltado en letras itálicas a los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentaremos una breve definición de cada uno de ellos.

Grupo de Aseguramiento de Calidad (AC). Este grupo trabaja en el proyecto de software durante las primeras etapas para establecer planes, estándares, y procedimientos que añadirán valor al proyecto de software. También revisa las actividades del proyecto y audita los productos de trabajo de software durante el ciclo de vida y provee a los administradores una visión adecuada.

Miembros del proyecto de software. Es el personal de la organización involucrado en el proyecto de software (por ejemplo, administradores, ingenieros de software, etc.)

Grupo de Ingeniería de Software (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 proceso de ingeniería de software no están incluidos en el grupo de ingeniería de software.

Personal de AC del cliente. Es el grupo que realiza periódicamente revisiones de las actividades del grupo de AC de la organización.

Administrador de proyecto: Participa en las revisiones de las actividades del AC que se realizan en forma periódica o a causa de un evento.

Administrador gerencial: Un rol de administración en 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 tendría la responsabilidad de varios proyectos a la vez. Participa en la revisión periódica de las actividades del AC.

Expertos independientes del grupo de AC. Personal externo o de la organización especializado en AC, que revisa las actividades y productos de trabajo del grupo de AC.

Organización. Es el conjunto de grupos relacionados al proyecto de software incluyendo a los grupos anteriormente mencionados.

6.2.2. Diagramas de actividades

Compromisos

En el texto se establece el compromiso de la organización desarrolladora de software de seguir una política organizacional durante la realización del aseguramiento de calidad de software. Consideramos que esta política es propia de la organización, es decir, ya existe y es aplicada en cada proyecto. En el diagrama (Fig. 6.1) representamos la existencia previa de esta política con el objeto *p:Política*.

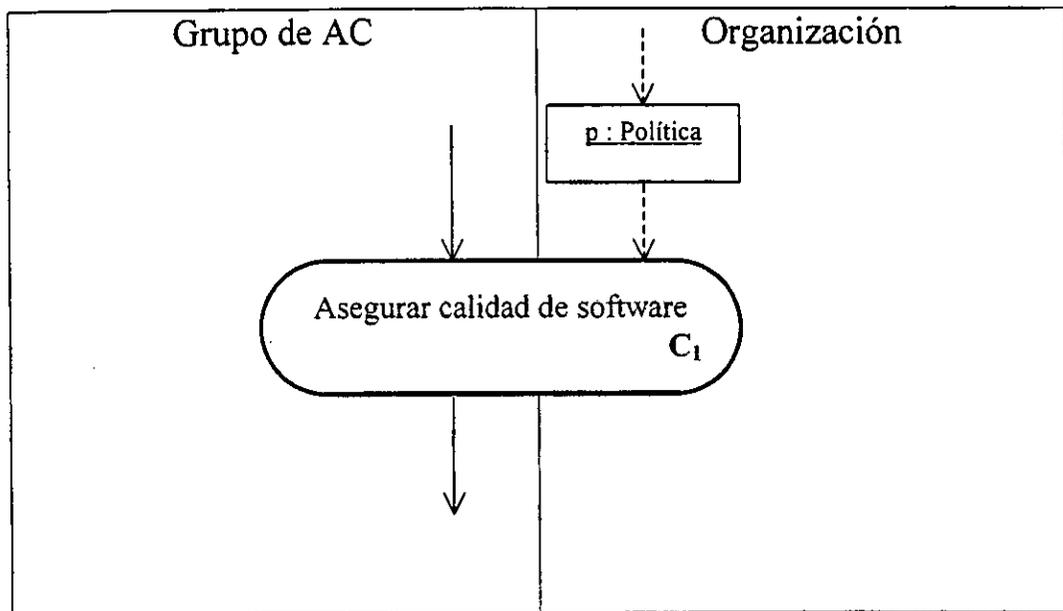


Fig. 6.1: Compromisos

Como habíamos mencionado, el compromiso 1 especifica que el proyecto sigue una política organizacional para el aseguramiento de calidad. Lo anterior lo representamos en el diagrama con la actividad "Asegurar calidad de software" al recibir como objeto de entrada a *p:Política*.

Las características comunes del área clave de procesos del AC que siguen: **Habilidades, Actividades, Mediciones y Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben de tomar para el aseguramiento de calidad del proyecto de software. Para modelar este hecho, la actividad de "Asegurar calidad de software" de la Fig.6.1 puede desglosarse en otro diagrama presentado en la Fig.6.2.

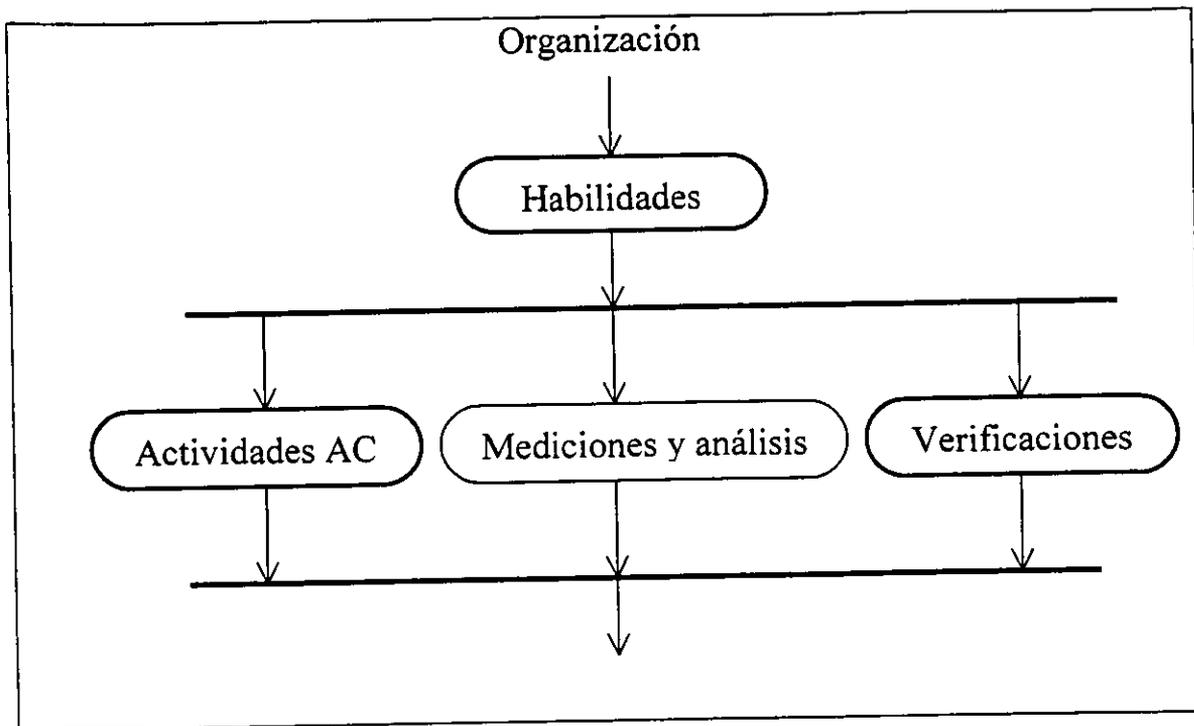


Fig. 6.2: Asegurar calidad de software

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 aseguramiento de calidad de software. Por otro lado, las actividades de verificación y medición con su análisis, son simultáneas a la de AC.

Habilidades.

La habilidad 1 del modelo textual habla de la existencia de un grupo de AC. En el diagrama de la Fig. 6.3 representamos este hecho con una decisión: si existe el *grupo de AC* entonces no realizamos ninguna actividad, pero si este grupo no existe entonces se lleva a cabo la actividad "*Constituir grupo de AC*". Representamos esta habilidad para asegurar la existencia del *grupo de AC*. Parece razonable pensar que el responsable de esta constitución es el Administrador, a pesar que el texto no lo dice explícitamente.

La habilidad 2 trata la asignación de los recursos y financiamiento para las actividades propias del AC. La cual modelamos con la actividad "*Asignar recursos y financiamiento*". En este caso, podemos deducir de la subpráctica que esta actividad la realiza el grupo *Administrador*. Esta actividad se realiza antes de proporcionar algún tipo de capacitación y orientación.

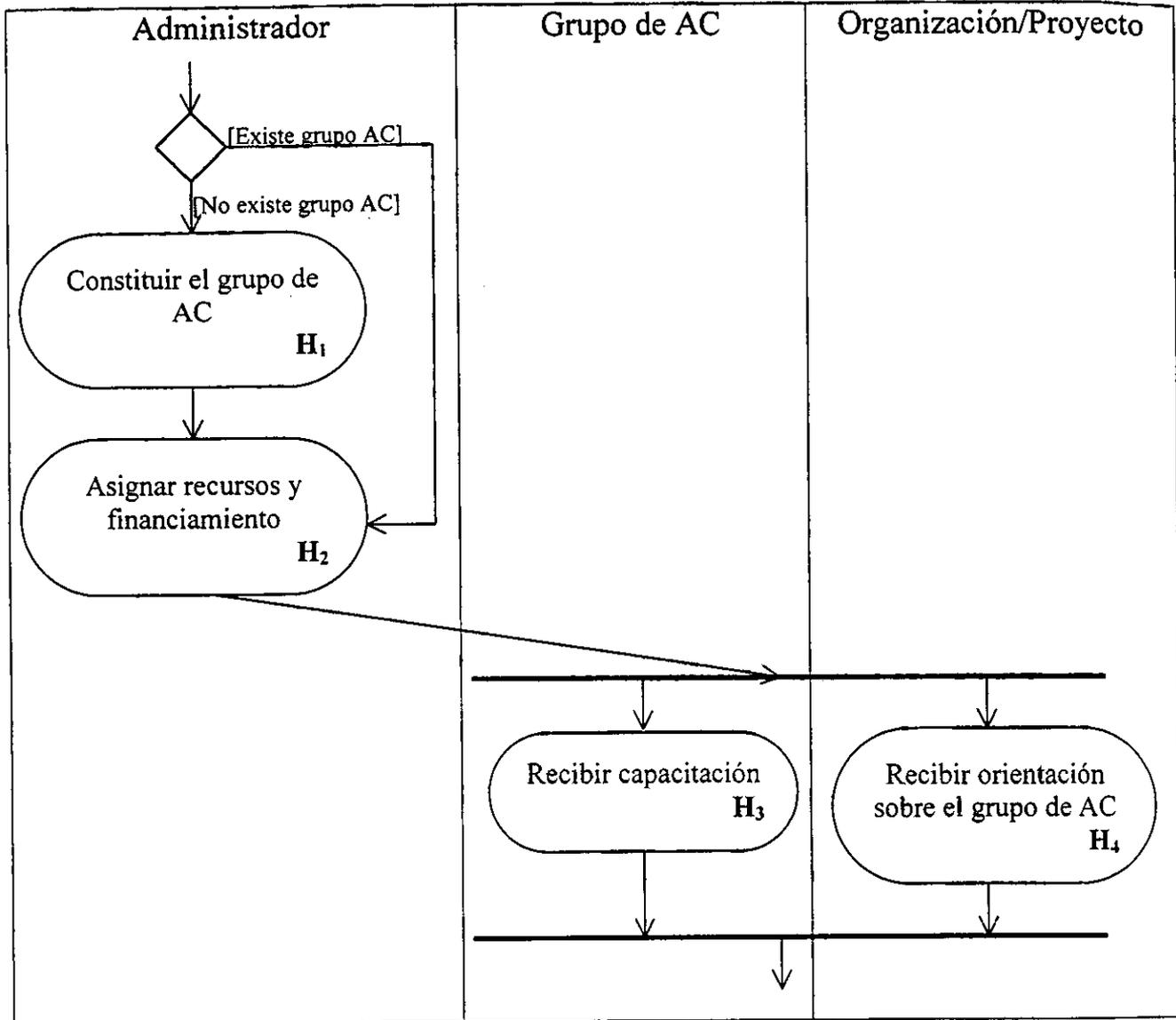


Fig. 6.3: Habilidades

La habilidad 3 especifica que el grupo de AC recibe capacitación para desempeñar sus actividades de AC. Esta habilidad se representa en el diagrama (Fig. 6.3) por la actividad "Recibir capacitación", esta actividad es realizada por el grupo de AC como lo especifica el modelo textual.

Finalmente la habilidad 4 habla de que los miembros del proyecto de software reciben orientación sobre el rol del *grupo de AC*. Esta habilidad la mapeamos en la actividad "Recibir orientación sobre el grupo de AC". Por cada proyecto, la organización debe realizar esta actividad, aunque parezca tedioso hacerlo. En la práctica sabemos que los miembros del proyecto pueden molestarse o irritarse cuando el grupo de AC realiza sus actividades². Por lo que es vital llevar a cabo esta actividad para conservar la armonía en el proyecto.

² Experiencia observada en el curso de Ingeniería de Software Orientada a Objetos de la Maestría en Ciencia e Ingeniería de la Computación.

Estas dos últimas actividades pueden ser realizadas en forma simultánea, como se representa en la Fig.6.3.

El diagrama de la Fig. 6.3 presenta las actividades a realizarse para adquirir las habilidades para el AC, donde cada una de las habilidades es mapeada en forma directa. También, en este diagrama, es fácil observar que actividades pueden llevarse a cabo en forma paralela y el orden en que se realizan.

Actividades

El modelo textual de esta sección está compuesto por ocho actividades, las cuales modelaremos en dos diagramas.

En el texto de la actividad 1 dice que se prepara el plan de AC del proyecto de software de acuerdo a un procedimiento documentado. Esta actividad la modelamos, en la Fig. 6.4, con "*Preparar plan de AC*", la cual es realizada por el *grupo de AC*. El texto menciona que esta actividad se realiza de acuerdo a un procedimiento documentado, lo que representamos en el diagrama con el flujo del objeto *pr:Procedimiento*. Esta actividad tiene como propósito obtener un plan de AC, lo que representamos con el objeto *plAC:PlanAC* que proviene de "*Preparar plan de AC*".

El texto de la actividad 3, establece que el *grupo de AC* participa en la preparación y revisión del plan de desarrollo de software, estándares y procedimientos del proyecto. Esta actividad se mapea en forma directa a "*Preparar y revisar plan de desarrollo, estándares y procedimientos de software*", la cual es realizada por el grupo de AC y personal de la organización asignado al proyecto.

La actividad "*Preparar plan de AC*", consideramos que se realiza en forma simultánea a la actividad "*Preparar y revisar plan de desarrollo, estándares y procedimientos de software*", lo que representamos en el diagrama (Fig. 6.4). Ambas actividades se llevan a cabo antes que cualquier actividad; puesto que ambas preparan elementos importantes para el resto de las actividades.

La actividad 2, expresa que el *grupo de AC* realiza sus actividades de acuerdo al plan de AC. Esta actividad la representamos con "*Realizar actividades AC*", llevada a cabo por el *grupo de AC*. En el texto dice explícitamente que esta actividad usa el plan de AC, hecho que visualizamos, diagrama de la Fig. 6.4, con el flujo del objeto *plAC:PlanAC*.

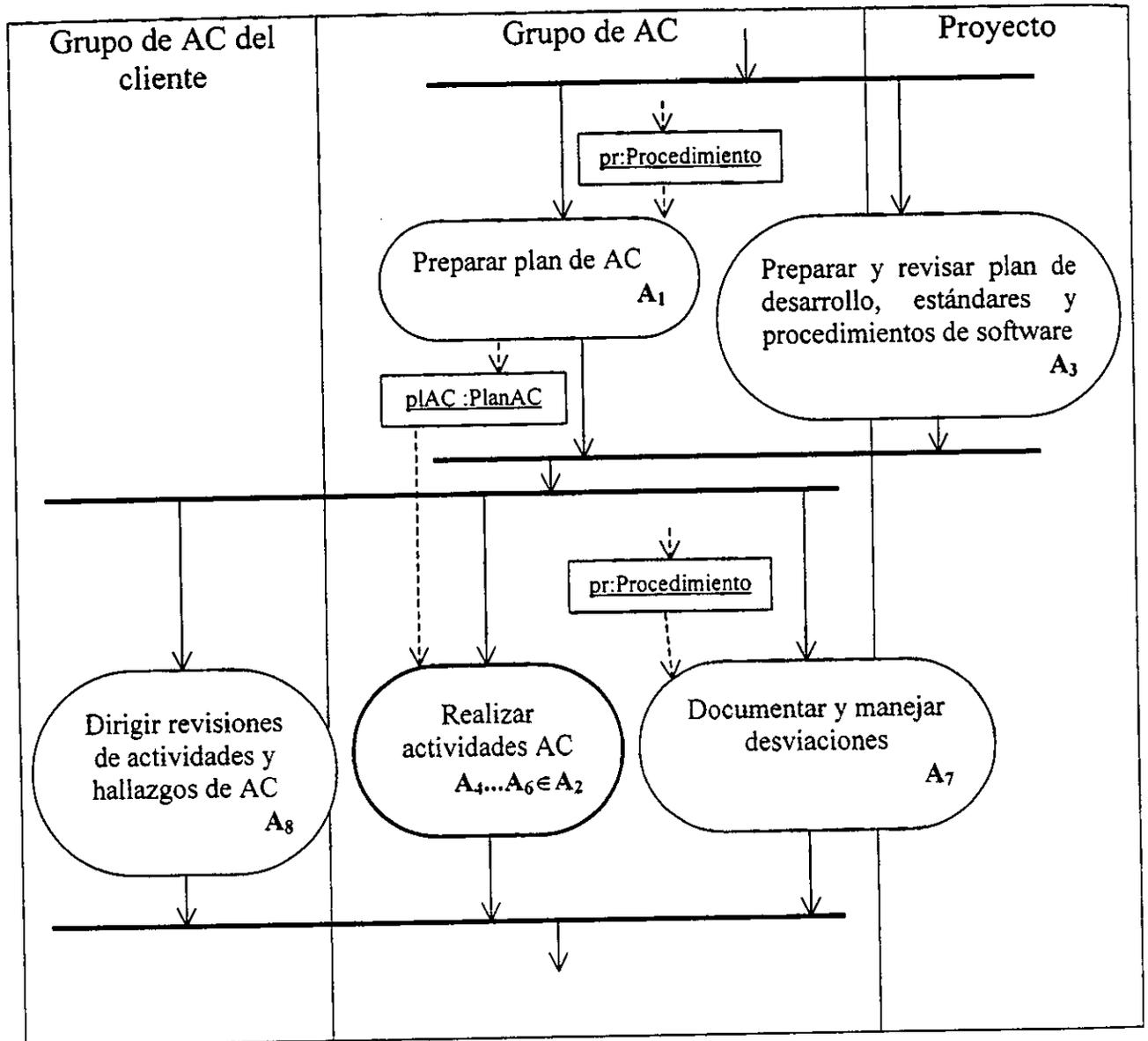


Fig. 6.4: Actividades AC

La actividad 7 especifica que las desviaciones en las actividades y en los productos de trabajo de software son administradas de acuerdo a un procedimiento documentado. En la parte de subprácticas de esta actividad, nos explica que es realizada por el *grupo de AC* en conjunto con líderes de tareas, administradores de software o administradores de proyecto. Por lo anterior, esta actividad queda plasmada en el diagrama con "*Documentar y manejar desviaciones*", realizada en forma compartida entre el *grupo de AC* y *Proyecto*. Con el grupo *Proyecto* de alguna manera estamos generalizando a los grupos participantes en el proyecto de software. De acuerdo a cada actividad, este grupo representa en forma abstracta a los grupos que intervienen en dicha actividad. Sólo falta comentar que esta actividad se realiza de acuerdo a un procedimiento documentado, que en el diagrama lo observamos con el objeto *pr:Procedimiento*.

La última actividad, que interviene en forma explícita en este diagrama, es la ocho. La cual especifica, que el grupo de AC realiza periódicamente revisiones tanto de sus actividades como de sus hallazgos con el personal de AC del cliente, en caso de que aplique. Este texto es muy claro, por lo que en el diagrama queda representado por la actividad "Dirigir revisiones de actividades y hallazgos de AC"; que se realiza en forma compartida por el grupo de AC del cliente y el grupo de AC del proyecto.

Definitivamente el centro de esta sección es la actividad "Realizar actividades AC", la que se realiza en forma paralela a las actividades "Documentar y manejar desviaciones" y "Dirigir revisiones de actividades y resultados de AC".

La actividad "Realizar actividades AC", de la figura 6.4, se desglosa en el diagrama de la Fig. 6.5; es decir esta actividad involucra realizar las actividades 4,5 y 6 del modelo textual. Para justificar esta representación, nos basaremos en la subpráctica de la actividad 2. En esta subpráctica, plantea los puntos que cubre el plan de AC. Entre estos puntos tenemos el sexto, que se refiere a las auditorias y revisiones a ser realizadas por el grupo de AC, este punto justifica porque incluimos a las actividades 4 y 5. Mientras el décimo punto se refiere al método y frecuencia para proveer retroalimentación al grupo de IS y otros grupos relacionados al software. Este punto justifica porque la actividad 6 la integramos a la actividad 2.

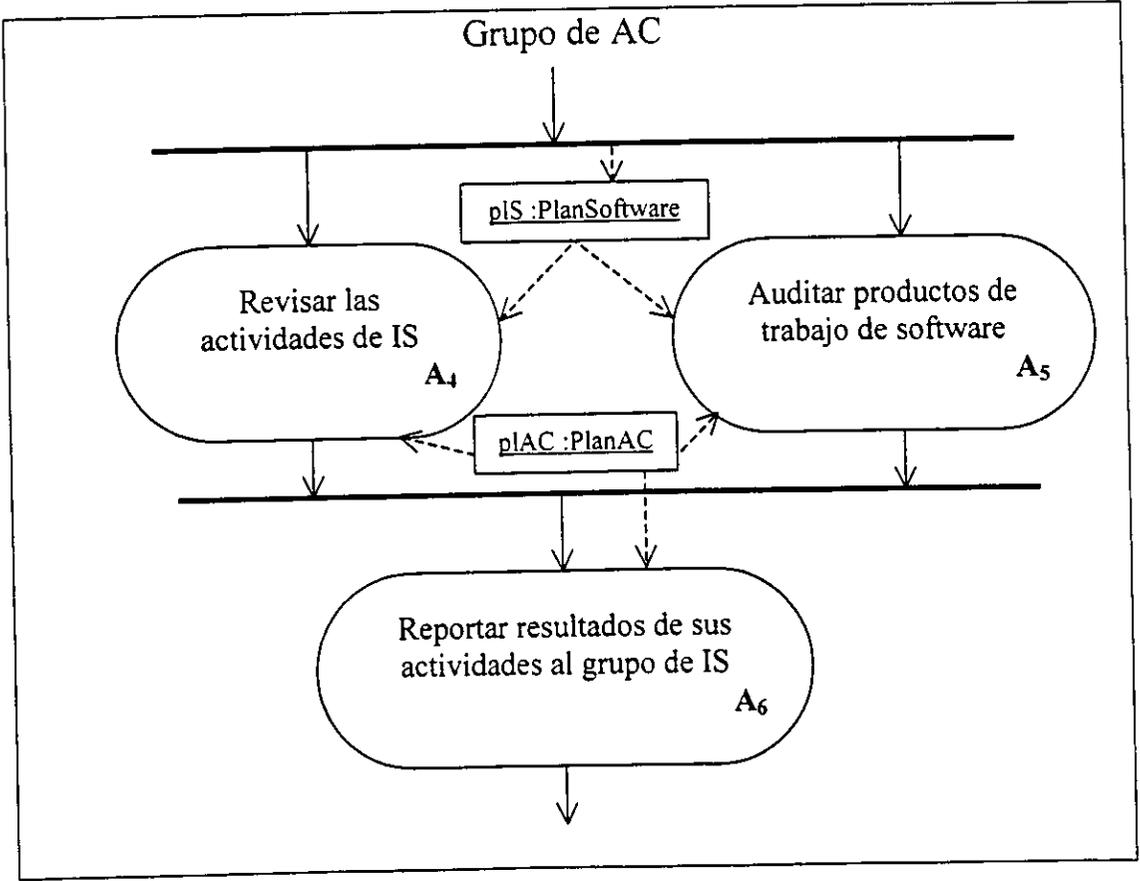


Fig. 6.5: Realizar actividades de AC

Capítulo 7

Administración de configuraciones de software

Esta área establece y mantiene la integridad de los productos de software durante el ciclo de vida de un proyecto de software. Para conservar la integridad de los productos de trabajo se identifica la configuración de software (productos de trabajo seleccionados y sus descripciones) en fechas específicas, se controlan los cambios y se da seguimiento a la configuración.

En este capítulo presentaremos la traducción de esta área clave de procesos [CMM, 1994], posteriormente iniciamos la descripción del modelo gráfico con la presentación de los grupos participantes y la concluimos con la explicación de los diagramas de actividades, que constituyen el modelo gráfico.

7.1. Modelo textual de la Administración de Configuraciones de Software (ACS) del nivel 2 de CMM.

El propósito de la Administración de Configuraciones de Software (ACS) es establecer y mantener la integridad de los productos de software durante el ciclo de vida de un proyecto de software.

La ACS involucra la identificación de la configuración del software (es decir, productos de trabajo de software seleccionados y sus descripciones) en fechas específicas, el control sistemático de los cambios a la configuración, y mantenimiento de la integridad y seguimiento de la configuración durante el ciclo de vida del software. Los productos de trabajo que son puestos bajo la ACS incluyen los productos que son entregados al cliente (por ejemplo, el documento de requisitos del software y el código) y las entidades que son identificadas o requeridas para crear estos productos (por ejemplo, el compilador).

Se define una biblioteca de líneas base de software, la cual contiene las líneas base conforme se van desarrollando. Los cambios a las líneas base y a los productos de software liberados a partir de la biblioteca de líneas base se controlan sistemáticamente mediante las funciones de control de cambios y auditoría de la configuración de la administración de configuraciones de software.

Esta área clave de procesos cubre las prácticas para desarrollar la función de ACS. Las prácticas que identifican entidades/idades específicas de la configuración están contenidas en las áreas clave de procesos que describen el desarrollo y mantenimiento de cada entidad/idad de configuración.

Metas

Meta 1

Planear las actividades de administración de configuraciones de software.

Meta 2

Identificar, controlar y hacer disponibles los productos de trabajo de software seleccionados.

Meta 3

Controlar los cambios a los productos de trabajo de software seleccionados.

Meta 4

Informar el estado y contenido de las líneas base de software a los grupos e individuos involucrados.

Compromisos

Compromiso 1

El proyecto sigue una política organizacional escrita para llevar a cabo la administración de configuraciones de software (ACS).

Esta política especifica que:

1. La responsabilidad para la ACS para cada proyecto se asigna explícitamente.
2. La ACS se lleva a cabo durante el ciclo de vida del proyecto.
3. La ACS se lleva a cabo para productos de software a ser entregados externamente, para productos de trabajo internos seleccionados y para herramientas de soporte usadas dentro del proyecto (por ejemplo, los compiladores).
4. Los proyectos establecen o tienen acceso a un almacén para guardar entidades/unidades y los registros asociados a la ACS.

En estas practicas, la "biblioteca de líneas base de software" se refiere al contenido de este almacén. Se considera al "sistema de almacén de administración de configuraciones" como el conjunto de herramientas y procedimientos para tener acceso a este almacén.

Las entidades de configuración son los productos de trabajo que son puestos bajo la administración de configuraciones y tratados como una entidad simple.

Las entidades de configuración son típicamente descompuestos en componentes de configuración, los componentes de configuración son típicamente descompuestos en unidades. En un sistema hardware/software, todo lo que se refiere a software puede ser considerado como una entidad de configuración simple, o el software podría ser descompuesto en múltiples entidades de configuración. En estas prácticas el término "entidades/unidades de configuración" se usa para referirse a los elementos bajo la administración de configuración.

5. Las líneas base de software y las actividades de la ACS se auditan en forma periódica.

Habilidades

Habilidad 1

Existe o se constituye un comité que tiene la autoridad para administrar las líneas base del proyecto de software (es decir, un comité de control de configuraciones de software -CCCS).

El CCCS:

1. Autoriza el establecimiento de las líneas base de software y la identificación de las entidades/unidades de configuración.
2. Representa los intereses del administrador de proyecto y de todos los grupos que podrían estar involucrados por los cambios a las líneas base de software.

Ejemplos de grupos involucrados:

aseguramiento de calidad de hardware,
administración de configuración de hardware,
ingeniería de hardware,
ingeniería de manufactura,
ingeniería de software (incluyendo todos sus subgrupos, como diseño de software),
ingeniería de sistemas,
pruebas del sistema,
aseguramiento de calidad de software,
administración de configuración de software,
administración de contrato, y
soporte de documentación.

3. Revisa y autoriza los cambios a las líneas base de software.
4. Autoriza la creación de productos a partir de la biblioteca de las líneas base de software.

Habilidad 2

Existe un grupo que es responsable de coordinar y llevar a cabo la ACS para el proyecto (es decir el grupo de ACS).

Un grupo es la colección de departamentos, administradores, e individuos que comparten la responsabilidad sobre un conjunto de tareas o actividades. Un grupo podría variar de un individuo asignado parte de su tiempo laboral, a varios individuos de diferentes departamentos asignados parte de su tiempo laboral, o a varios individuos asignados de tiempo completo. Las consideraciones, para formar un grupo, incluyen las tareas o actividades asignadas, el tamaño del proyecto, la estructura y cultura organizacional. Algunos grupos, como el grupo de aseguramiento de calidad, están enfocados en las actividades del proyecto, y otros, como el grupo de ingeniería de procesos de software, están enfocados en las actividades de la organización.

El grupo de ACS coordina o realiza:

1. La creación y administración de la biblioteca de las líneas base de software del proyecto.
2. El desarrollo, mantenimiento, y distribución de los planes, estándares y procedimientos de la ACS.
3. La identificación del conjunto de productos de trabajo a ser puestos bajo la ACS.

Un producto de trabajo es algún artefacto para definir, mantener, o usar en un proceso de software.

4. La administración del acceso a la biblioteca de líneas base de software.
5. La actualización de las líneas base de software.
6. La creación de los productos a partir de la biblioteca de líneas base de software.
7. El registro de las acciones de la ACS.
8. La producción y distribución de los reportes de la ACS.

Habilidad 3

Se proveen los recursos y el financiamiento adecuados para realizar las actividades de ACS.

1. Un administrador con responsabilidades específicas se asigna para la ACS.
2. Las herramientas para dar soporte a las actividades de ACS se hacen disponibles.

Ejemplos de herramientas de soporte:
estaciones de trabajo,
programas de bases de datos, y
herramientas para la administración de configuraciones.

Habilidad 4

Se capacitan a los miembros del grupo de ACS en los objetivos, procedimientos y métodos para desempeñar las actividades de ACS.

Ejemplos de capacitación:
estándares, procedimientos, y métodos para la ACS; y
herramientas para ACS.

Habilidad 5

Se capacitan a los miembros del grupo de ingeniería de software y otros grupos relacionados al software para realizar sus actividades de ACS.

Ejemplos de otros grupos relacionados al software:
aseguramiento de calidad de software, y
soporte para documentación.

Ejemplos de capacitación:
los estándares, procedimientos, y métodos, que serán utilizados por el grupo de ingeniería y otros grupos relacionados al software, para realizar las actividades de ACS; y
el rol, responsabilidades, y autoridad del grupo de ACS.

Actividades

Actividad 1

Se prepara un plan de ACS para cada proyecto de software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El plan de ACS se desarrolla en las primeras etapas de, y en paralelo con, la planeación del proyecto.
2. El plan de ACS se revisa por los grupos involucrados.
3. El plan de ACS es administrado y controlado.

“Administrado y controlado” implica conocer la versión (es decir control de versión) del producto de trabajo en uso a un tiempo dado (pasado o presente) e incorporar los cambios en forma controlada (es decir control de cambio).

Si “administrado y controlado” implica un grado mayor de control, el producto de trabajo puede ser puesto bajo una disciplina completa de administración de configuraciones, como se describe en el área clave de procesos Administración de Configuraciones de Software.

Actividad 2

Se usa un plan de ACS documentado y aprobado como base para realizar las actividades de ACS.

El plan cubre:

1. Las actividades de la ACS a ser realizadas, el calendario de actividades, las responsabilidades asignadas, y los recursos requeridos (incluyendo personal, herramientas, y facilidades computacionales).
2. Los requisitos y las actividades de la ACS a ser realizados por el grupo de ingeniería de software y otros grupos relacionados al software.

Actividad 3

Se establece un sistema bibliotecario de administración de configuraciones como un almacén para las líneas base de software.

Este sistema bibliotecario:

1. Soporta múltiples niveles de control de ACS.

Ejemplos de situaciones que llevan a múltiples niveles de control:
diferencias en los niveles de control necesarias en diferentes fases del ciclo de vida (por ejemplo, dar mayor control a un producto maduro),
diferencias en los niveles de control necesarias para sistemas únicamente de software contra sistemas de software y hardware.

2. Almacena y recupera las entidades/unidades de configuración.
3. Comparte y transfiere las entidades/unidades de configuración entre los grupos involucrados y entre los niveles de control en la biblioteca.
4. Ayuda en el uso de los estándares de productos para las entidades/unidades de configuración.

5. Almacena y recupera las versiones archivadas de las entidades/unidades de configuración.
6. Ayuda a garantizar la creación correcta de productos a partir de la biblioteca de líneas base de software.
7. Almacena, actualiza y recupera los registros de la ACS.
8. Da soporte a la producción de los reportes de ACS.
9. Da mantenimiento a la estructura y contenido de la biblioteca.

Ejemplos de funciones de mantenimiento de la biblioteca:
respaldos/restablecimiento de archivos de la biblioteca, y
recuperación de los errores de la biblioteca.

Actividad 4

Se identifican los productos de trabajo de software que serán puestos bajo la administración de configuraciones.

1. Las entidades/unidades de configuración se seleccionan de acuerdo a criterios documentados.

Ejemplos de productos de trabajo de software que podrían ser identificados como entidades/unidades de configuración:

- documentación relacionada a procesos (por ejemplo, planes, estándares o procedimientos),
- requisitos de software,
- diseño de software,
- unidades de código de software,
- procedimientos de prueba de software,
- sistema de software creado para las actividades de prueba del software,
- sistema de software creado para la entrega a los clientes o usuarios finales,
- compiladores, y
- otras herramientas de soporte.

2. Las entidades/unidades de configuración se les asignan identificadores únicos.
3. Se especifican las características de cada entidad/unidad de configuración.
4. Se especifican las líneas base de software a las cuales pertenece cada entidad/unidad de configuración.
5. Se especifica el momento del desarrollo en que cada entidad/unidad se pone bajo la administración de configuraciones.
6. Se identifica la persona responsable de cada entidad/unidad de configuración (por ejemplo el propietario, desde el punto de vista de la administración de configuración).

Actividad 5

Las solicitudes de cambio y reportes de problemas para todas las entidades/unidades de configuración se inicializan, se registran, se revisan, se aprueban y se siguen de acuerdo a un procedimiento documentado.

Actividad 6

Los cambios a las líneas base se controlan de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las revisiones y/o pruebas de regresión se realizan para garantizar que los cambios no han causado efectos no deseados en las líneas base.
2. Solamente las entidades/unidades de configuración que sean aprobadas por el CCCS se integran a la biblioteca de líneas base de software.
3. Las entidades/unidades de configuración se revisan al integrarse y al extraerse para mantener la biblioteca de líneas base de software correcta e íntegra.

Ejemplos de pasos para integrar o extraer;
verificar que las revisiones están autorizadas,
crear un registro de cambios,
mantener una copia de los cambios,
actualizar la biblioteca de las líneas base de software, y
archivar las líneas base reemplazadas.

Actividad 7

Se crean los productos a partir de la biblioteca de las líneas base de software y su liberación se controla de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. El CCCS autoriza la creación de productos procedentes de la biblioteca de líneas base de software.
2. Los productos de uso interno y externo se crean solamente de entidades/unidades de configuración a partir de la biblioteca de líneas base de software.

Actividad 8

El estado de las entidades/unidades de la configuración se registra de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. Las acciones de la administración de configuraciones se registran con suficiente detalle para que el contenido y el estado de cada entidad/unidad de configuración se conozca y versiones previas puedan ser recuperadas.
2. Se mantiene el estado actual e historia (es decir, cambios y otras acciones) de cada entidad/unidad de configuración.

Actividad 9

Se elaboran los reportes estandarizados que documentan las actividades de la ACS y el contenido de las líneas base de software y se ponen a disposición de los grupos e individuos involucrados.

Ejemplos de reportes:

minutas de reuniones del CCCS,
resumen y estado de las solicitudes de cambio,
resumen y estado de los reportes de problemas (incluyendo soluciones),
resumen de los cambios hechos a las líneas base de software,
historial de revisiones de las entidades/unidades de configuración,
estado de las líneas base, y
resultados de las auditorías a las líneas base de software.

Actividad 10

Se realizan auditorías a las líneas base del software de acuerdo a un procedimiento documentado.

Este procedimiento típicamente especifica que:

1. La auditoría se prepara adecuadamente.
2. Se evalúa la integridad de las líneas base.
3. Se revisa la estructura y las facilidades del sistema bibliotecario de la administración de configuración.
4. Se verifica si el contenido de la biblioteca de las líneas base de software es correcto y está completo.
5. Se verifica el cumplimiento de los estándares y procedimientos de la ACS.
6. Los resultados de la auditoría se reportan al administrador del proyecto de software.
7. Las acciones resultantes de las auditorías son seguidas hasta su término.

Mediciones y análisis

Medición 1

Las mediciones se definen y se usan para determinar el estado de las actividades de la ACS.

Ejemplos de mediciones:

número de solicitudes de cambio procesadas por unidad de tiempo;
*milestones*¹ concluidos de las actividades de ACS comparadas con el plan; y
trabajo terminado, esfuerzo y recursos utilizados en las actividades de ACS.

¹ *Milestone*: Evento o fecha previamente acordada.

Verificaciones

Verificación 1

Las actividades de la ACS se revisan periódicamente 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.

Referirse a la Verificación 1 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador gerencial.

Verificación 2

Las actividades de ACS se revisan con el administrador del proyecto periódicamente o a causa de un evento.

Referirse a la Verificación 2 del área clave de procesos de Seguimiento y Control del Proyecto de Software para las prácticas que cubren el contenido típico de las revisiones del administrador de proyecto.

Verificación 3

El grupo de ACS audita periódicamente las líneas base para verificar su conformidad con la documentación que las define.

Verificación 4

El grupo de aseguramiento de calidad de software revisa y/o audita las actividades y productos de trabajo de la ACS y reporta resultados.

Referirse al área clave de procesos de Aseguramiento de Calidad de Software.

Las revisiones y/o auditorías verifican al menos:

1. El cumplimiento de los estándares y procedimientos por:
 - el grupo de ACS,
 - el CCCS,
 - el grupo de ingeniería de software, y
 - otros grupos relacionados al software.
2. La realización de las auditorías periódicas a las líneas base de software.

7.2. Modelo gráfico de la Administración Configuraciones de Software (ACS).

El modelado gráfico se iniciará con una identificación, en todo el texto original, de los grupos de actores participantes en el proceso de ACS y posteriormente con una asignación de sus responsabilidades en los diagramas de actividades.

7.2.1. Identificación de los grupos participantes

En la descripción de la administración de configuraciones hemos identificado y resaltado en letras itálicas los grupos participantes que son relevantes en el proceso de esta área clave. A continuación presentaremos una breve definición de cada uno de ellos.

Grupo de Administración de Configuraciones de Software (ACS). Este grupo establece y mantiene la integridad de los productos de trabajo durante el ciclo de vida de un proyecto de software.

Comité de Control de Configuraciones de Software (CCCS). Es un comité que tiene la autoridad para administrar las líneas base de software del proyecto, efectuar el control de cambios a las entidades/unidades de configuración y las actividades propias de ACS.

Grupo de Ingeniería de Software (IS). Es la colección de individuos (tanto administradores como personal técnico) que tienen la responsabilidad de realizar el desarrollo del software y actividades de mantenimiento para un proyecto.

Administrador². Crea las condiciones para la administración de configuraciones y revisa las actividades de esta área clave de procesos.

Grupo de Aseguramiento de Calidad (AC). Revisa y/o audita las actividades y productos de trabajo para la administración de configuraciones.

Organización. Es el conjunto de los grupos mencionados anteriormente.

²En el texto original se habla de dos niveles de administradores gerencial (*senior*) y de proyecto. Para fines de este capítulo ubicaremos ambos grupos bajo el mismo nombre de *Administrador*.

7.2.2. Diagramas de actividades

Compromisos

La parte de prácticas del modelo textual inicia con la sección de Compromisos. Analizando el texto del único compromiso, se deduce que la organización tiene una política escrita que guíe la administración de configuraciones de los proyectos. La existencia previa de la política organizacional queda plasmada en el diagrama de actividades (Fig.7.1) como un rectángulo del objeto *p:Política*. Este objeto fue probablemente generado por una actividad de definición de política, la cuál, según el texto del compromiso, queda implícita y por lo tanto no aparece en el diagrama. El compromiso de seguir la política de ACS en cada proyecto queda expresado colocando como objeto de entrada, *p:Política*, a la actividad de "Administrar configuraciones".

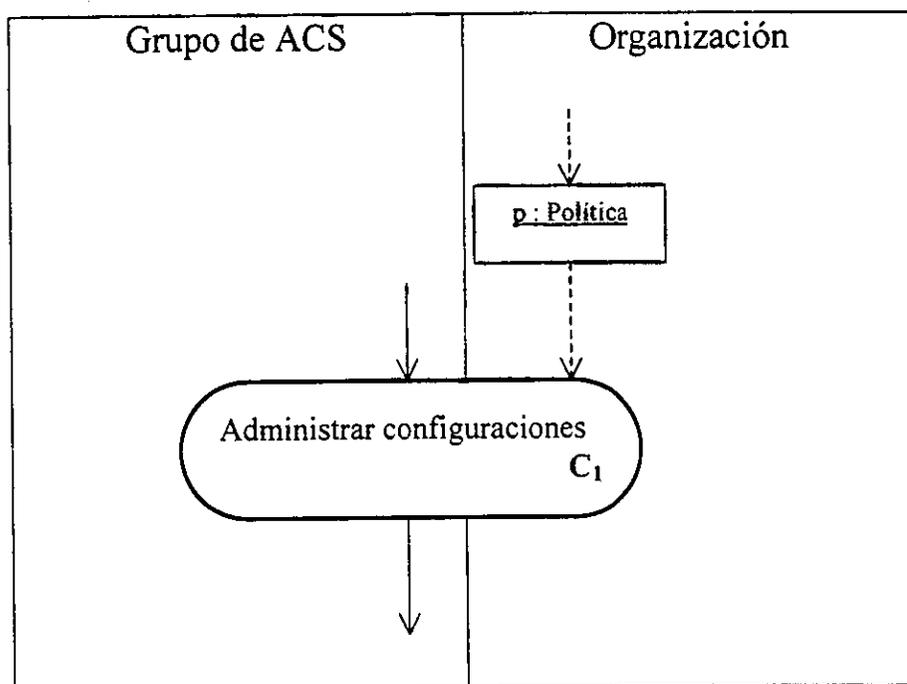


Fig. 7.1 : Compromisos

Las secciones que la siguen: **Habilidades**, **Actividades**, **Mediciones** y **Análisis y Verificaciones** son las que realmente describen las acciones que los diferentes grupos participantes deben de tomar para administrar configuraciones en un proyecto. Para modelar este hecho, la actividad de "Administrar configuraciones" de la Fig.7.1 puede desglosarse en otro diagrama presentado en la Fig.7.2.

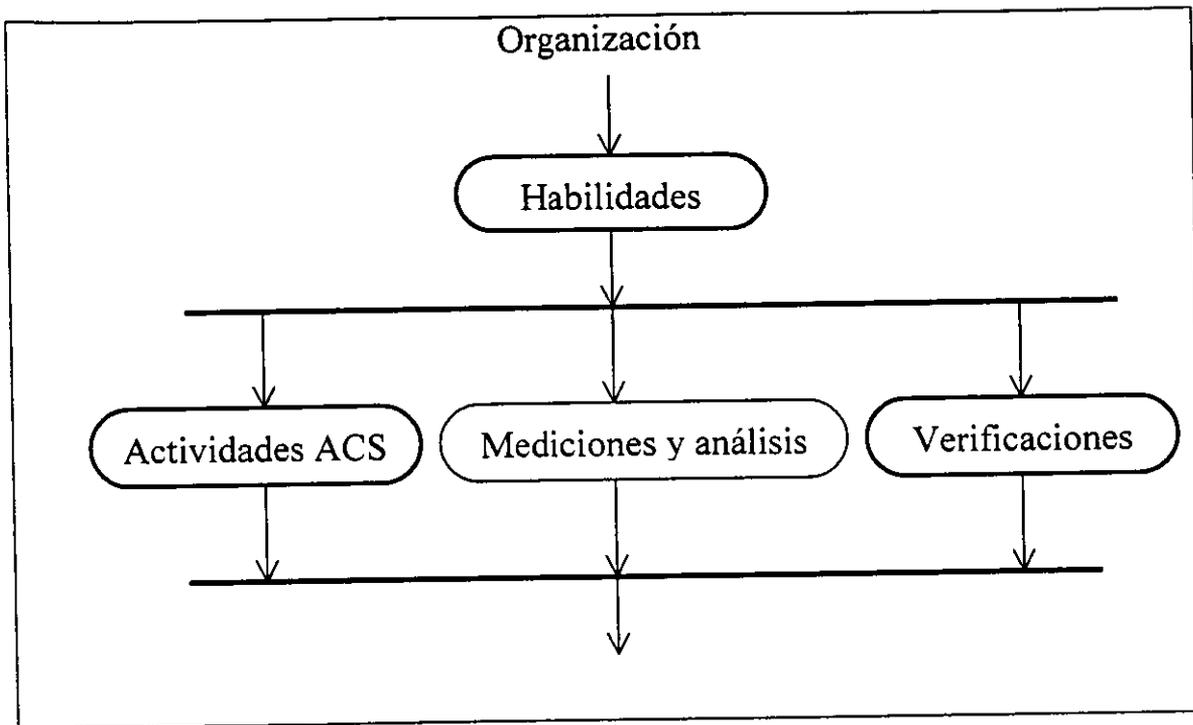


Fig. 7.2: Administrar configuraciones

Este diagrama introduce una actividad por cada sección 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 configuraciones. Por otro lado, las actividades de verificación y medición, con su análisis, son simultáneas a la de ACS, y tienen la misma duración que el proyecto.

Habilidades

La habilidad 1 del modelo textual habla que existe o se constituye un *Comité de Control De Configuración de Software (CCCS)*. En el diagrama de la Fig. 7.3 representamos este hecho con una decisión: si existe el CCCS entonces no realizamos ninguna actividad, pero si este comité no existe entonces se lleva a cabo la actividad "*Constituir CCCS*". En la habilidad 2 se dice que existe un *grupo de ACS*. Esta habilidad también la modelamos con una decisión: si existe el *grupo de ACS* entonces no realizamos ninguna actividad, pero si este grupo no existe entonces se realiza la actividad "*Constituir grupo ACS*". Representamos esta habilidad de esta forma para asegurar la existencia del *grupo de ACS*. Consideramos que ambas decisiones se realizan en forma simultánea. Nos parece razonable que el *grupo Administrador* realice ambas actividades aunque esto no queda explícito en el texto.

La habilidad 3 habla de la provisión de recursos y financiamiento, también se puede interpretar que está a cargo del *Administrador*. Esta habilidad la representamos con la actividad "*Asignar recursos*" y ésta se lleva a cabo antes de proporcionar algún tipo de capacitación.

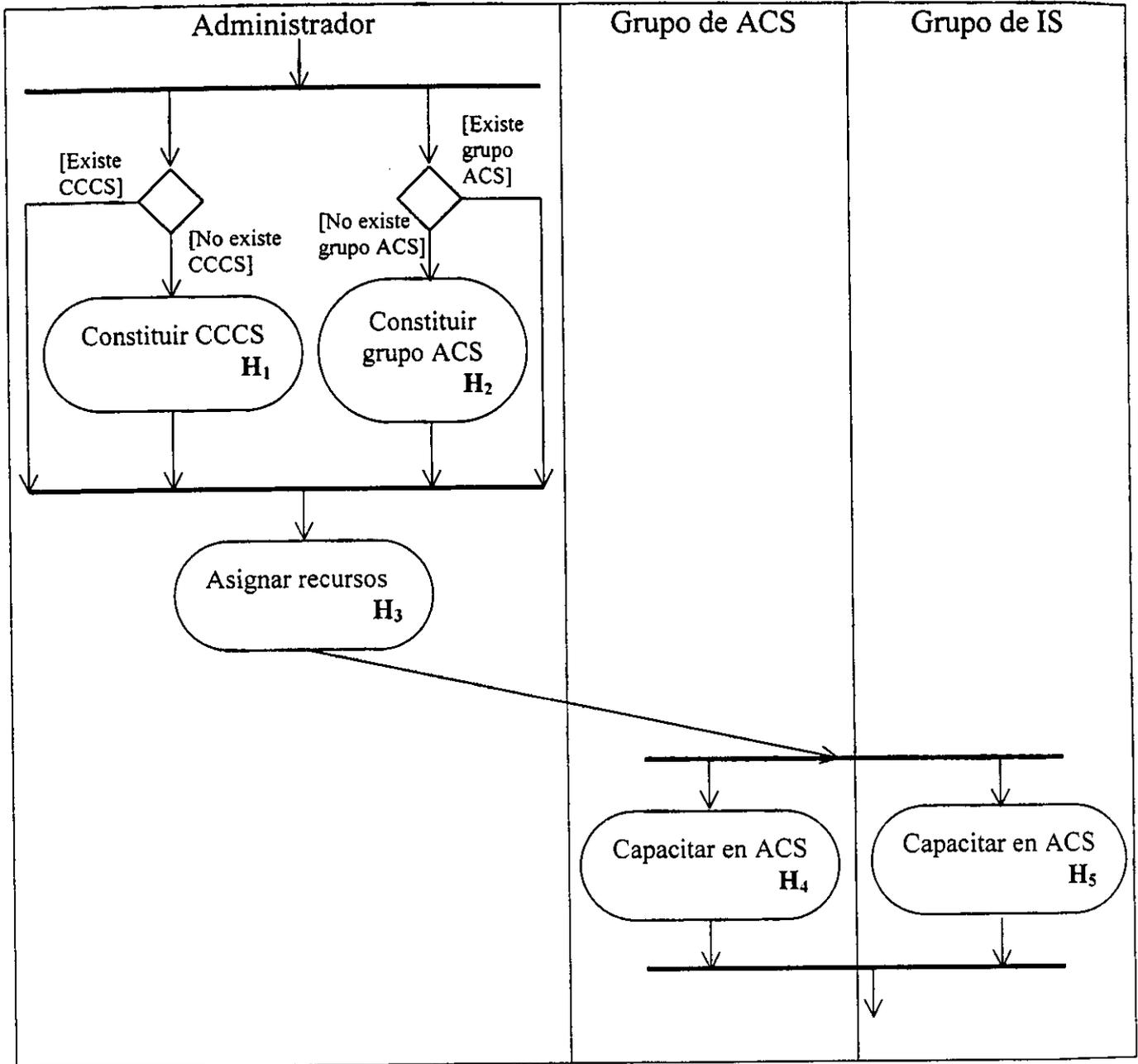


Fig. 7.3: Habilidades

Las dos últimas habilidades tratan la capacitación de los grupos involucrados en este proceso. La habilidad 4 especifica que el *grupo de ACS* es capacitado en objetivos, procedimientos y métodos para realizar las actividades de ACS, mientras que la habilidad 5 dice que se capacita el *grupo de IS* y otros grupos relacionados. En el diagrama (fig. 7.3) no consideramos este último grupo para no complicarlo. Ambas habilidades las representamos con la actividad "Capacitar en ACS", pero esta actividad no es compartida entre los *grupos de ACS y de IS*. Colocamos dos actividades en forma simultanea porque "Capacitar en ACS" toma su propio significado de acuerdo al grupo al que aplica.

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

Esta parte del modelo textual está integrada por diez actividades. Para hacer su representación gráfica más legible las colocamos en tres diagramas.

El texto de la actividad 1 dice que un plan de ACS es preparado de acuerdo a un procedimiento documentado y la primera parte de la actividad 2 menciona que este plan es documentado y aprobado. Conjuntando ambos textos los mapeamos a las dos actividades, "Preparar y documentar plan ACS" y "Aprobar plan ACS". La primera, "Preparar y documentar plan ACS", quedó asignada como responsabilidad del grupo de ACS. Esta actividad se realiza por medio de un procedimiento existente, el cual queda plasmado en el diagrama (fig. 7.4) con el objeto *pr:Procedimiento*. De esta actividad se obtiene un plan de ACS en versión de borrador.

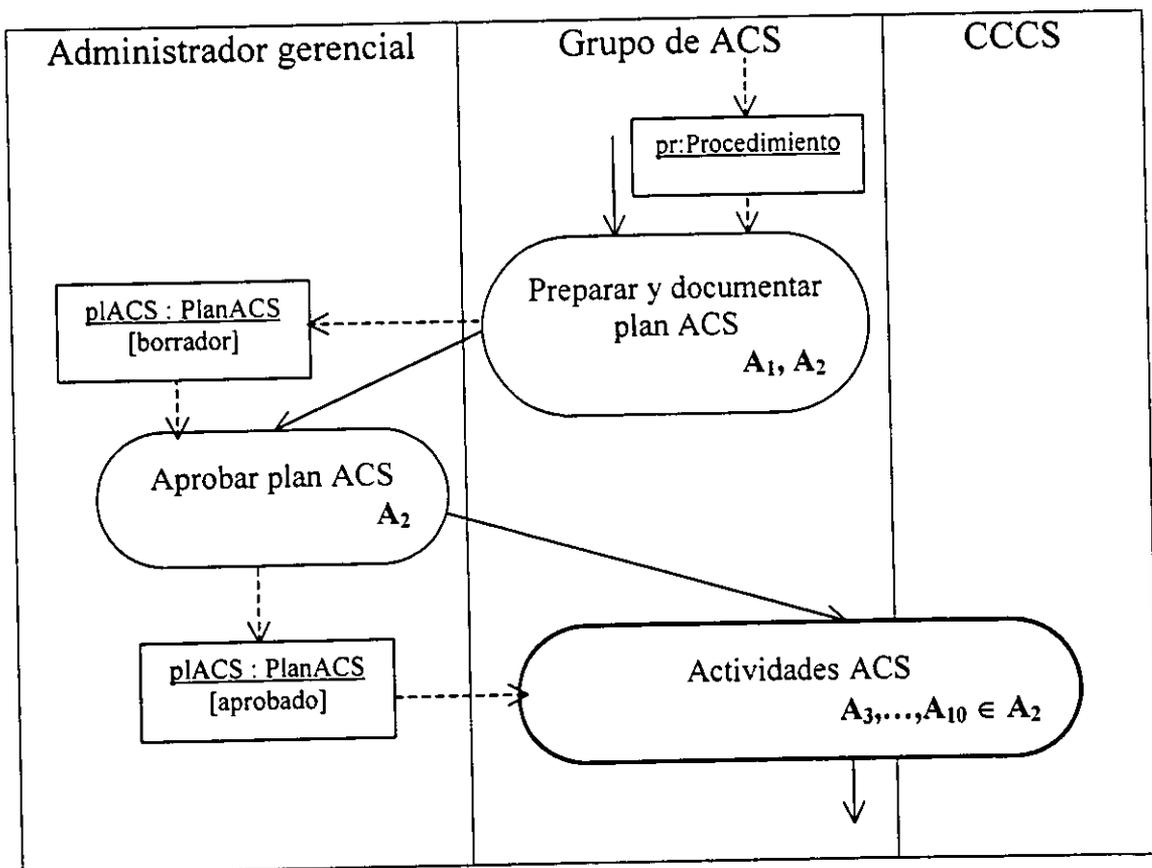


Fig. 7.4: Actividades ACS

Como ya habíamos mencionado anteriormente, la actividad 2 especifica que este plan de ACS debe ser aprobado, lo cual representamos en el diagrama con la actividad “*Aprobar plan ACS*” que tiene como entrada el plan en borrador y como salida el plan aprobado. Esta actividad debería ser responsabilidad de algún grupo o individuo con la autoridad de hacerlo, como en el texto no dice explícitamente, proponemos que sea alguien de la administración de alto nivel, por ejemplo el *Administrador gerencial*.

La última parte de la actividad 2 dice que este plan es usado como la base para realizar las actividades de ACS. Esto lo plasmamos en el diagrama poniendo como objeto de entrada a *plACS:PlanACS* en estado de aprobado a la actividad “*Actividades ACS*”. Esta última representa las ocho actividades restantes. Cabe mencionar que la actividad “*Actividades ACS*” se realiza en forma compartida entre el *grupo de ACS*, que es el responsable directo de la ACS, y el *CCCS*, que es responsable de administrar las líneas base del proyecto.

El diagrama de actividades de la fig. 7.5 desglosa “*Actividades ACS*” del diagrama anterior (fig. 7.4) y representa las actividades de tres a la diez del modelo textual.

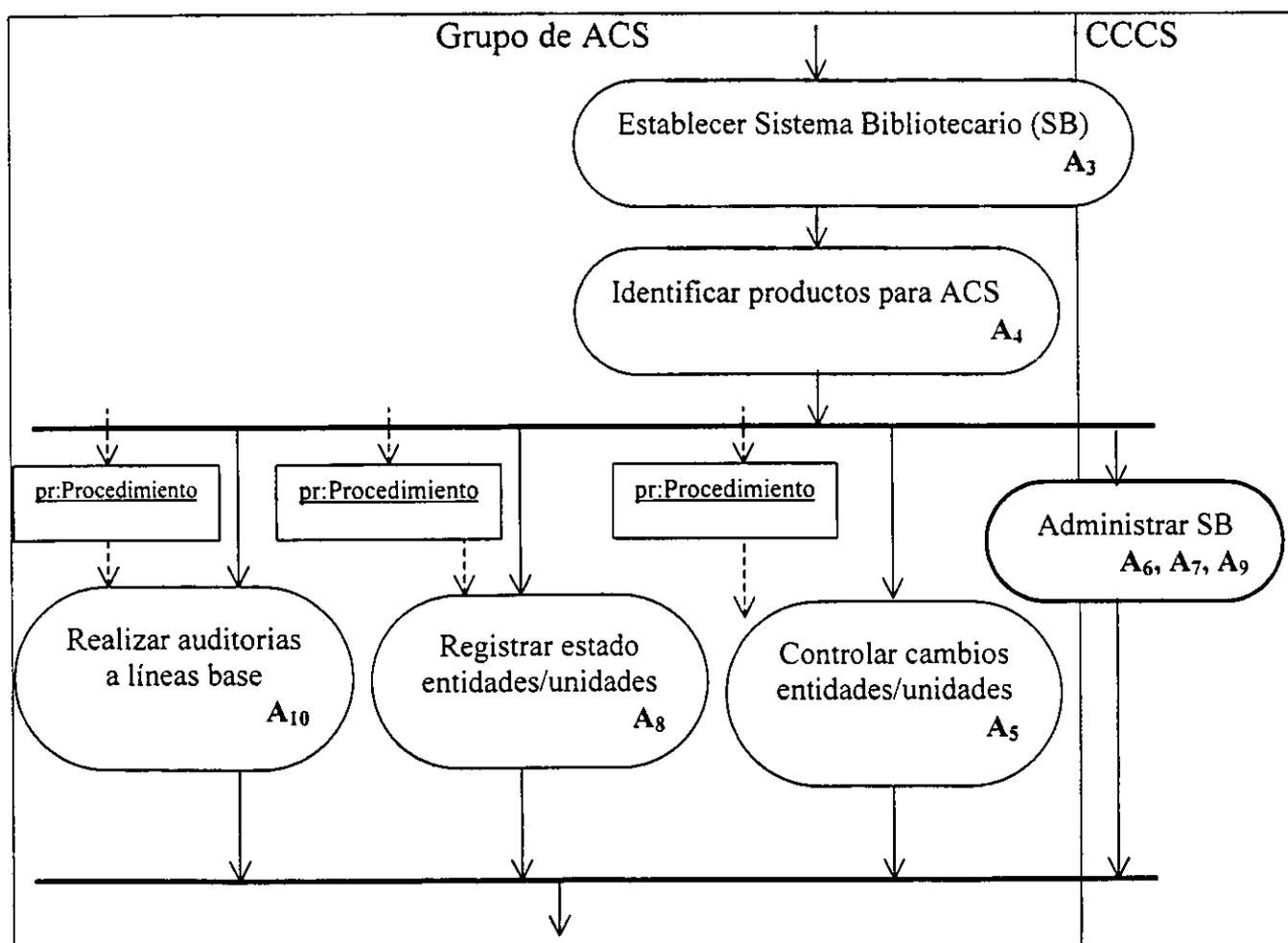


Fig. 7.5: Actividades ACS

La actividad 3 se mapea directamente con la actividad “*Establecer sistema bibliotecario (SB)*”, como responsabilidad compartida entre el *grupo de ACS* y *CCCS*. De forma similar la actividad 4 es representada por “*Identificar productos para ACS*” la cual se realiza una vez establecido el sistema bibliotecario.

Así como estas actividades se mapearon directamente, las actividades restantes de esta sección también se representarán de esta forma. Para mayor legibilidad del diagrama (fig.7.5) agrupamos las actividades “*Controlar cambios líneas base*”, “*Crear y liberar productos SB*” y “*Desarrollar y propagar reportes de actividades ACS y líneas base*” en la actividad “*Administrar SB*”. Las actividades que se realizan en forma simultánea a “*Administrar SB*” son “*Controlar cambios entidades/unidades*”, “*Realizar auditorias a líneas base*” y “*Registrar estado entidades/unidades*”. El desglose de la actividad “*Administrar SB*” lo presentamos en la fig. 7.6.

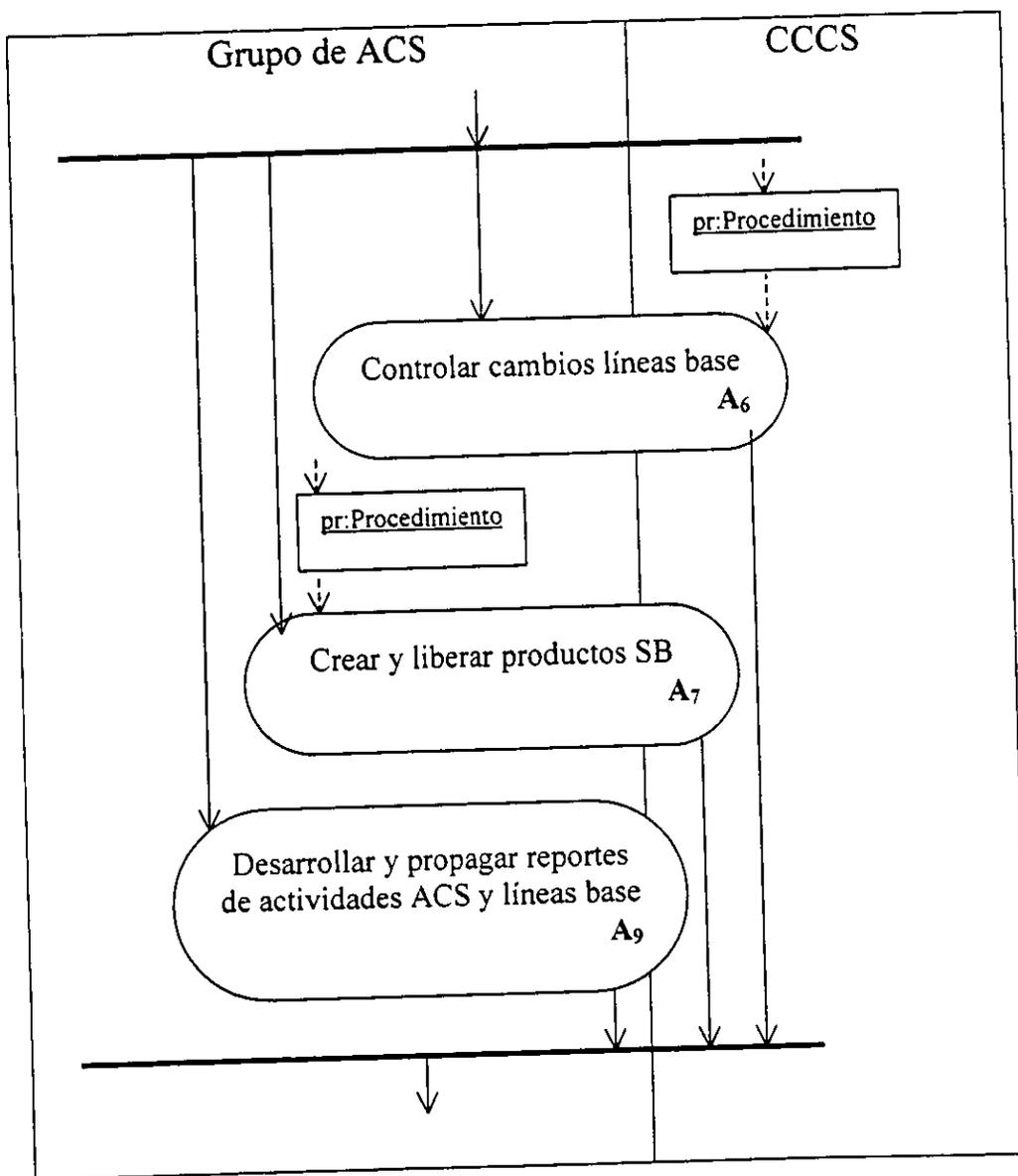


Fig.7.6: Administrar SB

Finalmente cabe aclarar que la responsabilidad de realizar “*Administrar SB*” es compartida entre el *grupo de ACS* y *CCCS*, puesto que las actividades que la integran tienen esta característica y también son actividades que se realizan en forma simultánea.

Mediciones y análisis.

El texto correspondiente a la medición y análisis es muy escueto. Se dice que deben de definirse y usar medidas para determinar el estado de las actividades de la ACS. Sin embargo, falta precisar quienes son responsables por efectuar dichas actividades. Conociendo todo el contexto del modelo textual de ACS nos atrevemos a proponer (Fig.7.7) que la actividad de “*Mediciones y análisis*” es una actividad compartida entre el *grupo de ACS*, que debe ser responsable por recaudar datos de las mediciones, y el *CCCS* y el *Grupo de Aseguramiento de Calidad (AC)*, quienes utilizan estos datos para determinar el estado de las actividades de la ACS. Previamente, en la Fig. 7.2, ya quedó plasmado el paralelismo de las actividades de medición y análisis con las de ACS y de las verificaciones.

El diagrama, que proponemos (Fig. 7.7), le añade la interpretación de las responsabilidades a la actividad de “*Mediciones y análisis*” con respecto al modelo textual, lo que a nuestro juicio facilita entender en que grupos debe de recaer.

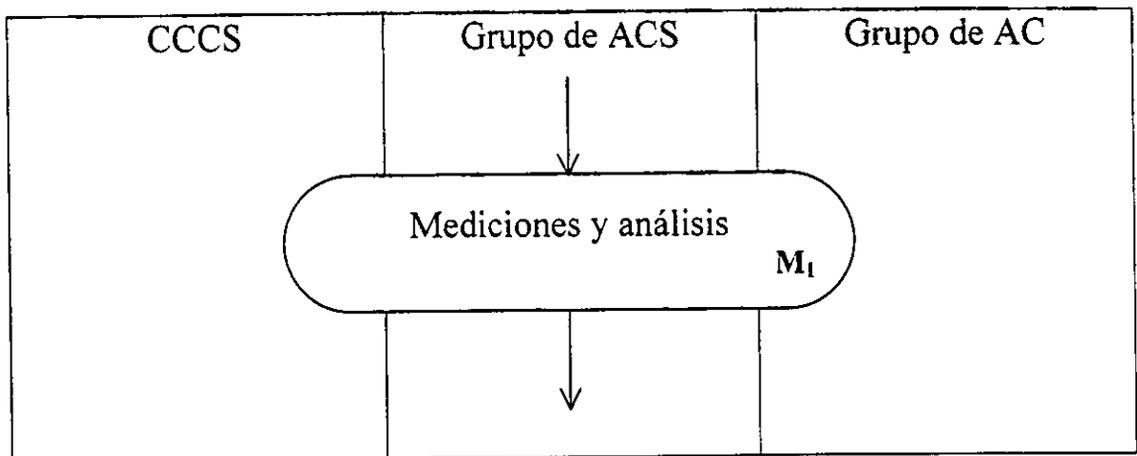


Fig. 7.7: Mediciones y análisis

Verificaciones

La verificación 1 especifica que las actividades de administrar configuración serán revisadas periódicamente por el administrador gerencial y la verificación 2 por el administrador de proyecto. Como habíamos mencionado, en el modelo gráfico ambos administradores son considerados como el grupo *Administrador* por la similitud de sus

responsabilidades. Las dos verificaciones las representamos (Fig. 7.8) como la actividad de "Revisar actividades ACS".

La verificación 3 especifica que el grupo de ACS audita las líneas base. Esta verificación la mapeamos directamente a la actividad "Auditar líneas base" y como especifica el texto es realizada por el grupo de ACS.

Finalmente, en la verificación 4 el grupo de AC revisa y/o audita las actividades y productos de trabajo para ACS y reporta resultados. Esta verificación está representada con la actividad de "Revisar y/o auditar actividades y productos" compartida entre el grupo de ACS, grupo de AC, CCCS y grupo de IS.

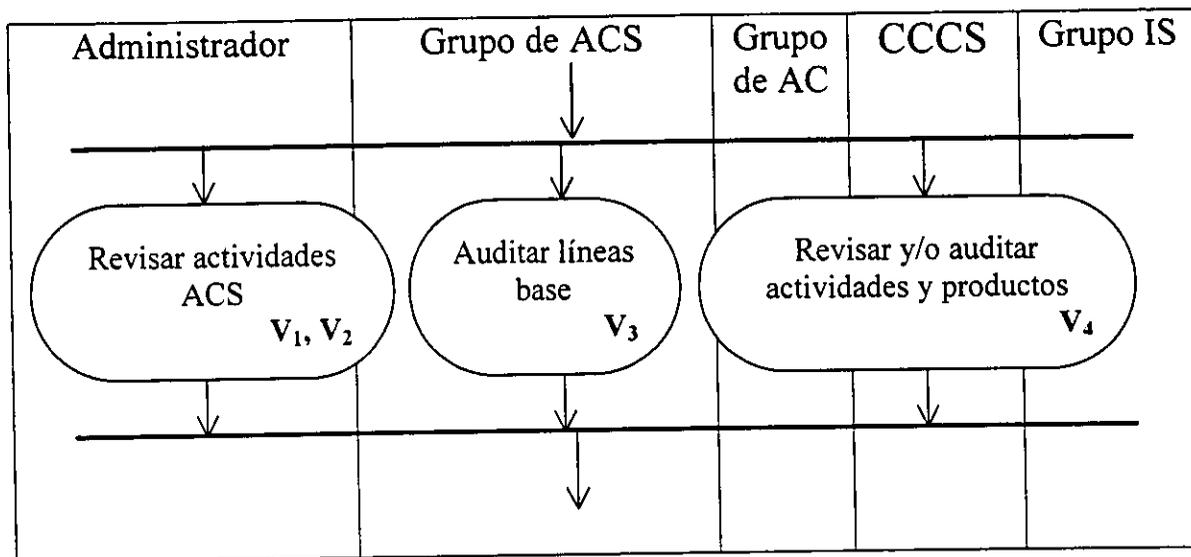


Fig. 7.8: Verificaciones

Capítulo 8

Conclusiones

El inicio de este trabajo fue difícil, porque el primer paso que tenía que dar era entender lo básico de CMM y en especial el nivel 2. Este primer contacto me resultó un poco decepcionante, pues considero que el texto se torna muy árido en las primeras lecturas.

El primer modelo gráfico que obtuvimos fue el del área clave de procesos de Administración de Requisitos, el cual estaba basado en los diagramas de clases, casos de uso y diagramas de interacción. Trabajamos en esta dirección hasta que dirigimos nuestra atención hacia los diagramas de actividades. Estos diagramas nos parecieron muy atractivos porque conjuntaban roles, actividades, objetos, flujo de trabajo y flujo de objetos.

Basándonos en los diagramas de actividades, realizamos otro modelo gráfico de la Administración de Requisitos, este modelo gráfico lo presentamos en el Segundo Encuentro Nacional de Computación [Oktaba and Alquicira, 1999-1]. Durante la obtención de esta representación gráfica, establecimos un proceso general para modelar las áreas clave, así como una extensión a los diagramas de actividades, que obtuvimos de forma natural. Una vez establecida la forma de trabajo se aplicó el mismo proceso al resto de las áreas clave del nivel 2, en especial la segunda fue la de Administración de Configuraciones de Software, que presentamos en el 4º. Seminario Estratégico "Seminario de Calidad en Ingeniería de Software" [Oktaba and Alquicira, 1999-2]. Cuando terminamos de aplicar el proceso de modelado a las áreas clave, ya habíamos obtenido cierta claridad de la relación que existe entre ellas, por lo que realizamos un diagrama de actividades para representar gráficamente esta relación del nivel 2.

Como parte del objetivo inicial de esta tesis, propusimos que el material generado sirviera de apoyo para la capacitación de empresas pequeñas, que quisieran mejorar sus procesos a nivel 2 de CMM. Por lo que utilizamos los capítulos del tres al cinco, que corresponden a la modelación de las áreas clave, como material para realizar, en dos ocasiones, un taller de interpretación del nivel 2 de CMM, apoyado por cinco cuestionarios para cada área clave. Estos cuestionarios los incluimos en el Anexo 2 de este trabajo. En el primer taller que llevamos a cabo asistieron alrededor de 20 personas que provenían de seis empresas distintas y en el segundo asistieron alrededor de 25 personas que pertenecían a una sola empresa. De la experiencia adquirida en este taller, podemos decir que las personas asistentes se apropiaron del conocimiento de CMM, además se dieron cuenta de sus fortalezas y de sus debilidades. Por lo que esto podría implicar, que esta área podría ser una vía de comunicación entre el sector académico y sector productivo.

A partir del desarrollo de este trabajo surgió una propuesta de hacer una extensión de los diagramas de actividades para el modelado de procesos de software, la cual está en vía de investigación. Durante el desarrollo del taller de interpretación surgió, por parte de los participantes, el interés de un trabajo similar para el nivel 3, el cual podría ser un tema para otra tesis. Actualmente ya está disponible el CMM integrado, el cual también es un modelo textual en inglés, que también se podría tratar de modelar con el proceso establecido en esta tesis.

Finalmente cabe mencionar, que a pesar de un inicio incierto podemos afirmar que estamos muy satisfechos del resultado de este trabajo, porque hemos podido comprobar que un trabajo realizado en términos académicos también puede ser útil para el sector productivo.

Anexo I

Diagramas de actividades

En este anexo presentamos una breve explicación de los elementos de los diagramas de actividades, este tipo de diagrama es uno de los cinco que están incluidos en UML (Lenguaje de Modelado Unificado). Un diagrama de actividades muestra el flujo de control de una actividad a otra y se usa para modelar aspectos dinámicos de un sistema.

Un diagrama de actividades es un caso especial de una máquina de estados, donde los estados son las actividades, y como consecuencia estos diagramas heredaron las decisiones, uniones y bifurcaciones. Las transiciones, carriles y flujo de objetos también forman parte de estos diagramas.

En las primeras secciones de este anexo explicaremos los elementos de los diagramas de actividades y en la última presentaremos dos extensiones, que proponemos, a la notación de los diagramas de actividades.

I.1 Actividades

Una actividad es una ejecución progresiva de cualquier tipo de acciones. En UML se modela por medio de un rectángulo redondeado con una etiqueta que contiene el nombre de la actividad. Por ejemplo, dos actividades que podemos realizar en la mañana para el desayuno son “*Preparar alimento*” y “*Preparar bebida*”, que mostramos en la figura I.1:

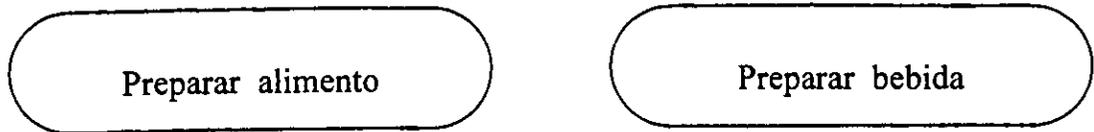


Fig. I.1: Actividades

Sabemos intuitivamente, que ambas actividades, involucran realizar una serie de simples pasos para llevarlas a cabo, es decir, si el alimento que queremos preparar es un emparedado podemos imaginar los pasos que debemos seguir para obtener este alimento. Por lo que ahora nos parecerá natural decir que una actividad está constituida de sentencias atómicas ejecutables que terminan en un cambio de estado del sistema o en la obtención de un valor.

Las siguientes observaciones sobre las actividades consideremos que son importantes:

- Un caso especial de una actividad es una acción, que es una sentencia atómica ejecutable. En otras palabras una acción es una actividad que no puede ser descompuesta.
- Una actividad puede representar un diagrama de actividades.

De la segunda observación deducimos que podemos modelar a diferentes niveles de abstracción en un diagrama de actividades. Es decir un diagrama de actividades puede sintetizarse o expandirse de acuerdo al nivel de comprensión deseado.

I.2 Transiciones

Cuando una actividad termina el flujo de control pasa inmediatamente a la siguiente actividad. Especificamos este flujo por medio del uso de transiciones para modelar la trayectoria de una actividad a la siguiente. En UML, se representa esta transición con una simple línea dirigida. Como observamos en el siguiente diagrama.

En el diagrama de actividades de la figura I.2, estamos modelando las actividades que posiblemente realizamos para desayunar. La primera actividad que llevamos a cabo es "*Poner la mesa*" y una vez finalizada, el flujo de control pasa inmediatamente a "*Preparar y consumir desayuno*", hecho que modelamos por medio de una transición. Al terminar esta última actividad mencionada realizamos "*Recoger la mesa*". Estas tres actividades las realizamos en forma secuencial.

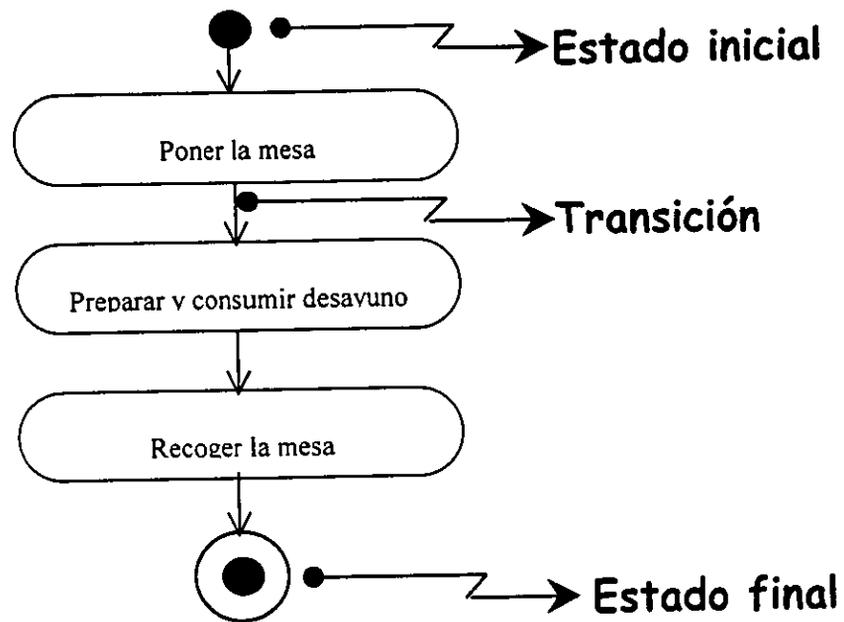


Fig. I.2 Transiciones

Un flujo de control tiene que empezar y terminar en algún momento, a menos que sea un flujo infinito, en este caso tendríamos que iniciar pero no finalizar. En la figura I.2 podemos observar que el estado inicial lo denotamos con un círculo lleno mientras que el estado final con un círculo lleno dentro de un círculo.

I.3 Bifurcaciones y uniones

Las transiciones simples secuenciales son las trayectorias más comunes que podemos encontrar en un diagrama de actividades. Sin embargo, podríamos tener la necesidad de modelar trayectorias que sean concurrentes o trayectorias simultáneas. En UML se usa la barra de sincronización para especificar las bifurcaciones y uniones de trayectorias simultáneas. Una barra de sincronización la representamos por una línea horizontal o vertical gruesa.

Modificaremos el ejemplo que presentamos en la sección anterior, al considerar que nuestro desayuno consiste en un alimento sólido y una bebida. Dada esta suposición, podríamos preparar un alimento y después consumirlo, y mientras realizamos esta secuencia de actividades, podríamos preparar una bebida y consumirla. En el diagrama de la figura I.3 mostramos el diagrama modificado de acuerdo a esta suposición.

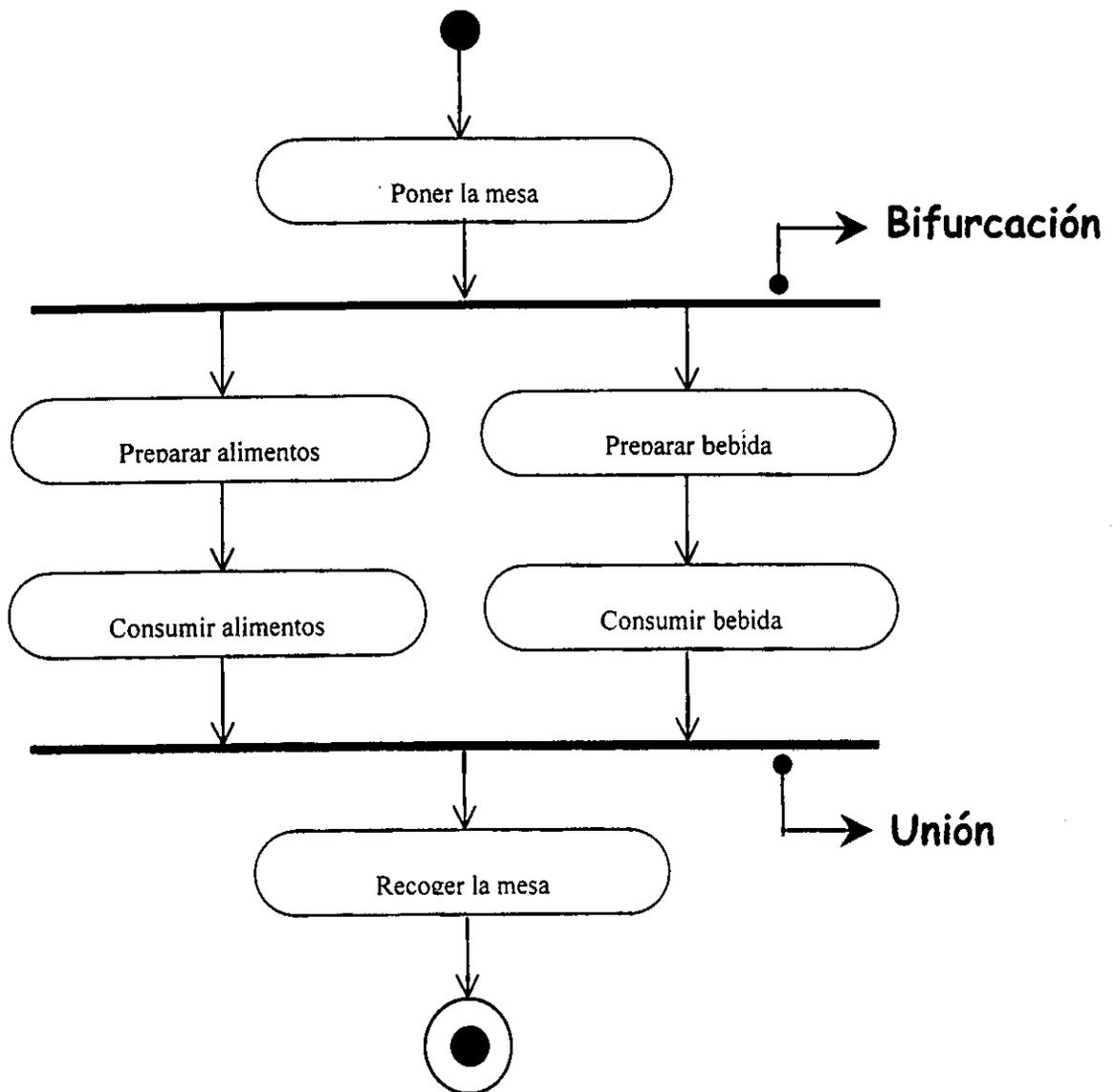


Fig. I.3: Bifurcaciones y uniones

Al terminar *"Poner la mesa"*, la transición de salida de esta actividad se bifurca en dos trayectorias, que en los diagramas de actividades de UML modelamos con una barra de sincronización, la cual conocemos como bifurcación. De esta barra salen estas dos trayectorias, en la primera se llevan a cabo en forma secuencial las actividades *"Preparar alimento"* y *"Consumir alimento"*, y la segunda está conformada por *"Preparar bebida"* y *"Consumir bebida"*. Una vez realizadas y hasta que finalicen ambas trayectorias, podemos continuar con la siguiente actividad, *"Recoger la mesa"*. Esto nuevamente lo modelamos con una barra de sincronización, en este caso esta barra representa una unión, la cual recibe dos transiciones y las transforma en una sola (Fig. I.3).

Para utilizar en forma correcta las bifurcaciones y uniones debemos tener en mente las siguientes consideraciones.

- A cada bifurcación le corresponde una unión.
- Una bifurcación tiene una transición de entrada y dos o más de salida.
- Una unión tiene dos o más transiciones de entrada y una de salida.
- Las bifurcaciones y uniones deben estar balanceadas, es decir el número de transiciones que salen de una bifurcación debe ser el mismo número de transiciones que entran a su unión correspondiente. Es decir, las trayectorias que inician en una bifurcación deben terminar en la unión que le corresponde a esta.

I.4 Decisiones

Como en los diagramas de flujo, también los diagramas de actividades incluyen las decisiones. Una decisión modela trayectorias alternativas tomando como base alguna expresión booleana. En UML, una decisión se representa con un rombo.

Para ejemplificar este elemento de los diagramas de actividades, seguiremos utilizando nuestro ejemplo. En esta ocasión, consideramos que al consumir la bebida preparada lo hacemos en porciones. Con esta suposición modificaremos el diagrama de la figura I.3 para obtener el de la figura I.4 y sólo explicaremos los cambios a este último diagrama.

Una vez terminada la actividad "*Preparar bebida*" (Fig. I.4) llevamos a cabo "*Consumir porción*", al finalizar esta actividad, continuamos con una decisión. En una de las transiciones de salida de esta decisión evaluamos la expresión **Desea otra porción**, en caso de ser afirmativa volvemos a realizar "*Consumir porción*" y en caso negativo vamos a la siguiente transición de salida. Evaluamos la expresión asociada, que será afirmativa por ser la negación de la expresión anterior, por lo que el flujo continuará en la unión.

Las características principales de las decisiones son:

- Las decisiones se utilizan para modelar trayectorias alternativas o repetición de trayectorias.
- Las decisiones tienen una transición de entrada y dos o más de salida.
- Cada una de las transiciones de salida tiene asociada una expresión booleana.

- UML no recomienda un lenguaje especial para estas expresiones. Se podría utilizar texto estructurado o la sintaxis y semántica de un lenguaje de programación específico.
- Las expresiones deben ser excluyentes entre sí y no deben ser ambiguas, para evitar ciclos innecesarios.

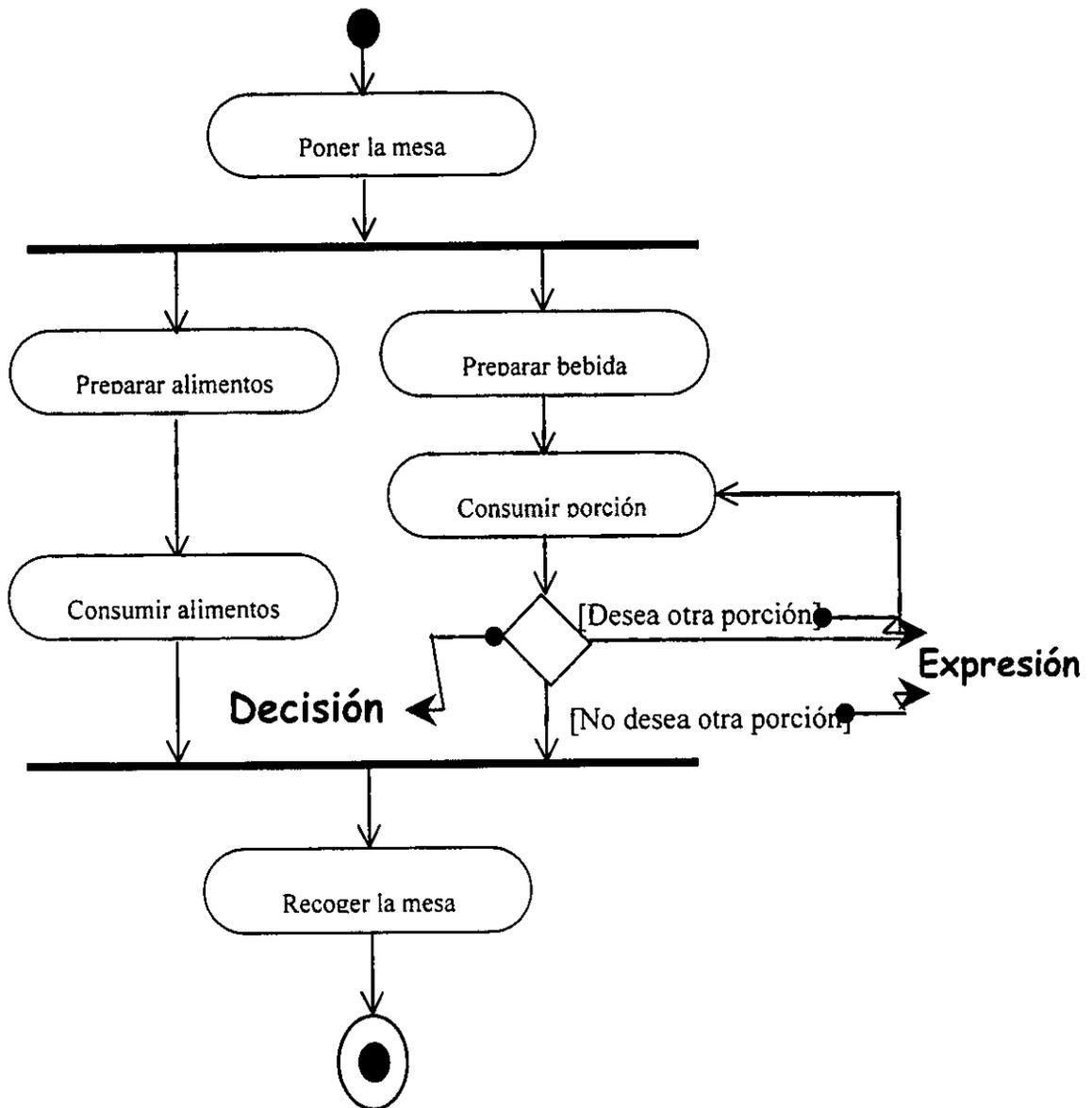


Fig. 1.4: Decisiones

I.5 Flujo de objetos

Los objetos pueden estar involucrados en un diagrama de actividades. Un objeto modela en forma abstracta a un ente o cosa del mundo real. 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 a la que pertenece el objeto. El nombre del objeto se escribe en letras minúsculas y el de la clase inicia con una letra mayúscula.

Como observamos en la figura I.5, con los objetos podemos modelar cosas que están involucradas con las actividades. Por ejemplo de la actividad "Preparar alimento" obtenemos un alimento que representamos con el objeto *a:Alimento* y en forma similar "Preparar bebida" tiene como objeto de salida a *b:Bebida*.

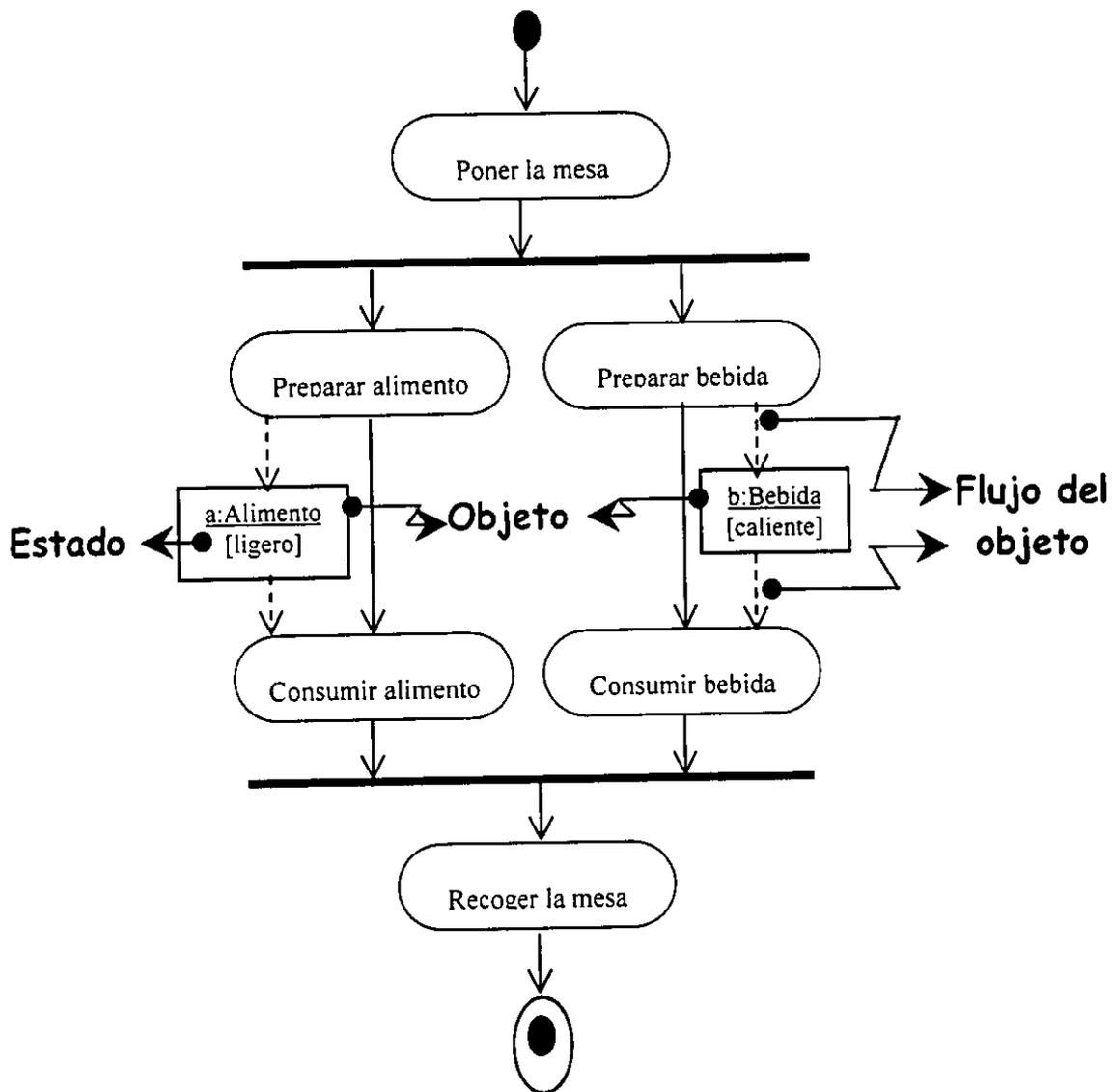


Fig. I.5: Flujo de objetos

Cuando una actividad crea, modifica, utiliza o destruye un objeto decimos que se lleva a cabo un flujo de objetos y modelamos este flujo de objetos por medio de líneas dirigidas punteadas.

En un diagrama de actividades, también podemos mostrar el estado y atributos de un objeto. En la figura I.5 el objeto *a:Alimento* tiene el estado de ligero, mientras que *b:Bebida* el de caliente. En ambos objetos especificamos los atributos.

Como observamos en la figura, el estado de un objeto se escribe debajo del identificador de este y entre paréntesis cuadrados. Los atributos de un objeto se escriben en un compartimento debajo del nombre y atributos.

I.6 Carriles

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 están contenidos en un rectángulo y cada grupo es separado de sus vecinos por una línea vertical.

En nuestro ejemplo (Fig. I.6), la organización asociada a estas actividades es un matrimonio. A los individuos, que constituyen esta organización, los nombraremos como *Esposa* y *Esposo*. La *Esposa* inicia las actividades, al poner la mesa. Una vez puesta, ella prepara el alimento mientras él prepara la bebida y hasta que ambos terminen pueden consumirlos. Posteriormente, la *Esposa* y el *Esposo* consumen, en forma simultánea, tanto el alimento como la bebida y hasta que los dos finalicen, él podrá levantar la mesa.

El diagrama de la figura I.6, lo hemos dividido en dos grupos, como ya hemos mencionado. La *Esposa* es responsable por el grupo de actividades constituido por "*Poner la mesa*", "*Preparar alimento*", "*Consumir alimento*" y "*Consumir bebida*". El *Esposo* lo es por "*Preparar bebida*", "*Consumir alimento*", "*Consumir bebida*" y "*Recoger la mesa*".

A continuación mencionamos las características sobresalientes de los carriles.

- Los carriles dividen al diagrama de actividades en grupos.
- Un carril puede ser representado por una entidad del mundo real.
- Cada actividad pertenece solamente a un carril.
- Las transiciones y las barras de sincronización pueden cruzar los carriles.

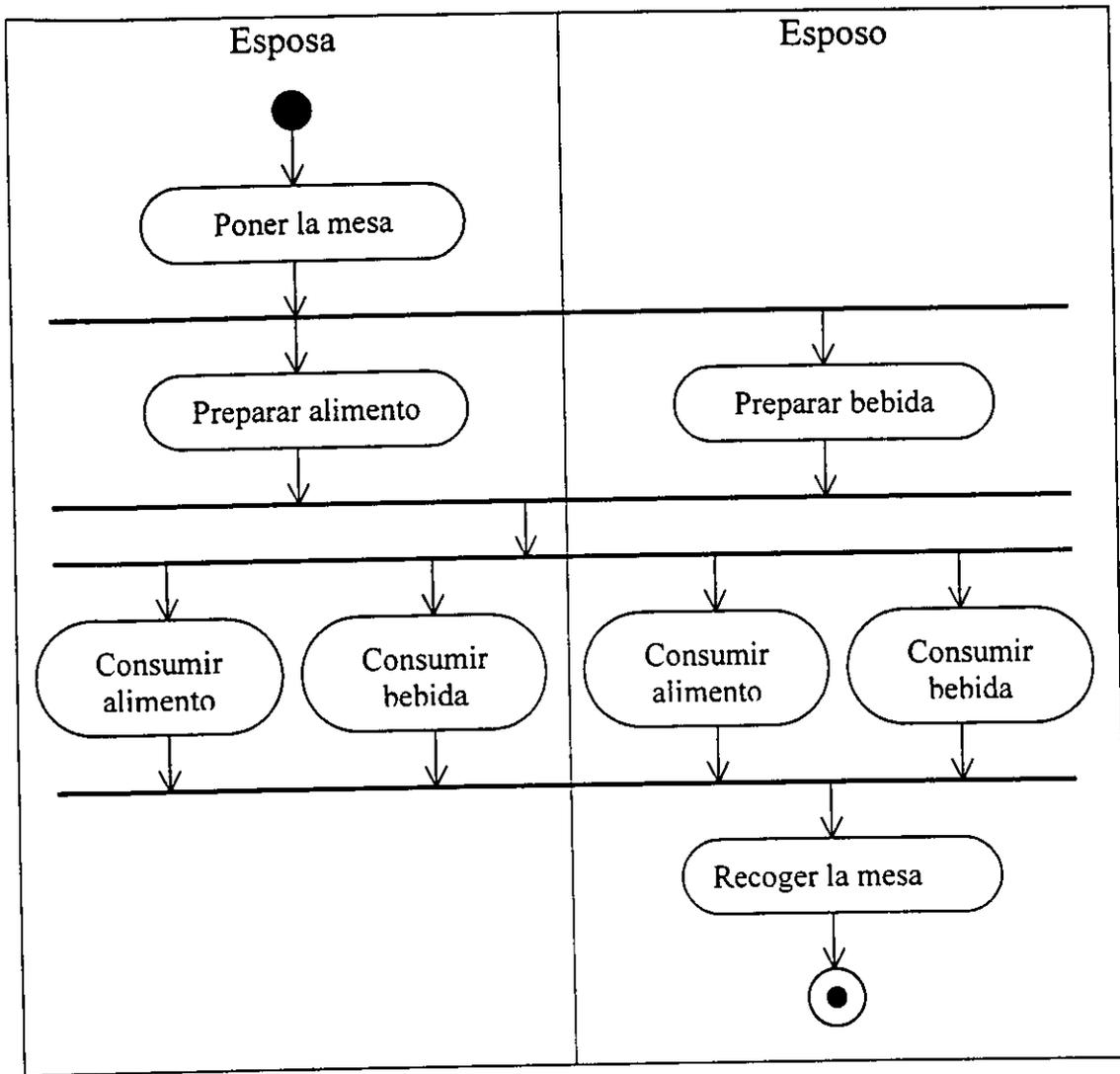


Fig. I.6: Carriles

I.7 Extensión a los diagramas de actividades.

Los elementos de los diagramas de actividades, que hemos explicado en las secciones anteriores, nos parecieron adecuados para modelar gráficamente las áreas clave de procesos del nivel 2 del CMM. Pero durante el desarrollo de este trabajo surgieron, de forma natural, dos elementos más a estos diagramas, en esta sección los explicaremos.

I.7.1 Desglose de actividades

En la sección I.1 mencionamos que una actividad puede representar a su vez un diagrama de actividades. En nuestro caso, queremos hacer explícita esta característica, 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. Para representar el desglose de una actividad lo haremos engrosando su rectángulo redondeado.

Por ejemplo, del diagrama de actividades de la figura I.1, desglosaremos la actividad "Preparar y consumir desayuno" y en la figura I.7 presentamos el diagrama con esta nueva notación gráfica.

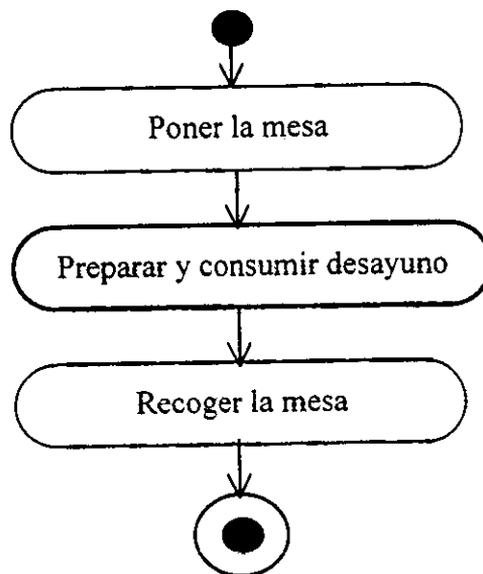


Fig. I.7: Desglose de actividades

Un posible diagrama de actividades de "Preparar y consumir desayuno", lo presentamos en la figura I.8. En este caso por ser un ejemplo sencillo el diagrama completo sin anidamiento está en la figura I.3. Pero el desglose de una actividad lo consideramos útil para comprender y entender en forma progresiva flujos de trabajo más complicados.

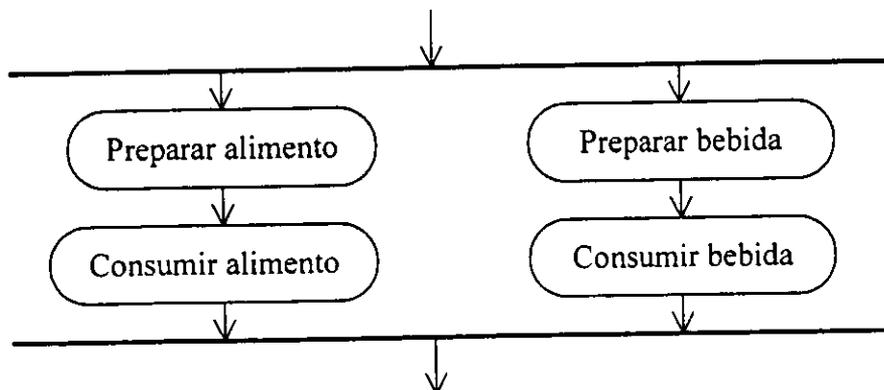


Fig. I.8: Preparar y consumir desayuno

I.7.2 Responsabilidades compartidas

En la sección I.6, que se refiere a los carriles, mencionamos que cada actividad pertenece solamente a un carril. Pero consideramos útil permitir que una actividad pueda cruzar uno o ambos límites de un carril, esto significaría que esta actividad puede ser realizada por más de un responsable de una organización. Es decir representaríamos responsabilidades compartidas sobre una actividad.

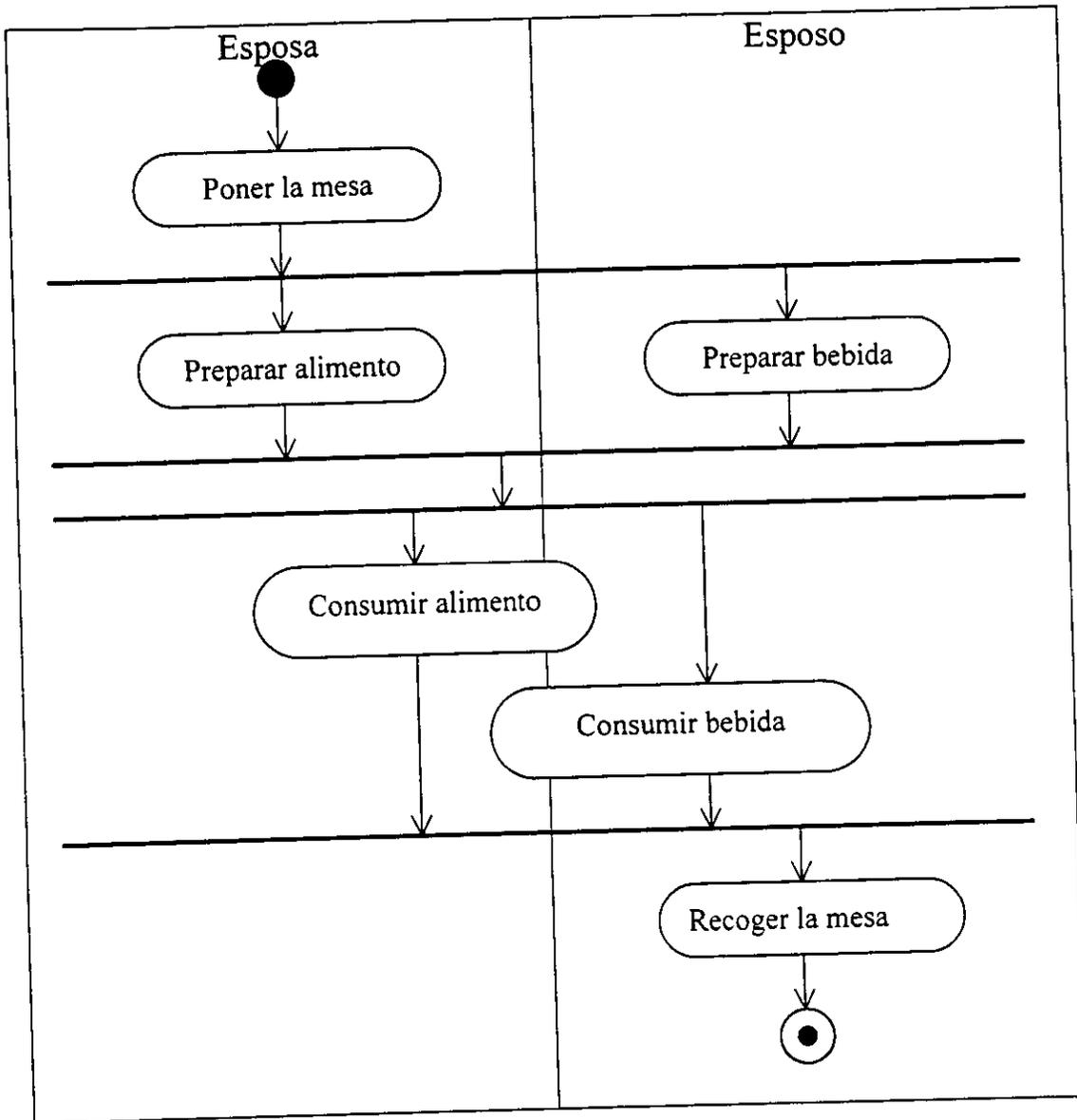


Fig. I.9: Responsabilidades compartidas

En el ejemplo que presentamos en la sección I.6, consideramos que la *Esposa* y el *Esposo* consumen un alimento y una bebida al mismo tiempo. Esto lo modelamos en el diagrama de la figura I.6, colocamos en forma simultánea cuatro actividades, "Consumir alimento" y "Consumir bebida" en el carril de la *Esposa* y otras dos con los mismos nombres en el carril del *Esposo*. Este diagrama de alguna forma es redundante, pero fue

construido de acuerdo a la sintaxis de los carriles. Con esta extensión simplificaremos el diagrama y solo utilizaremos dos actividades compartidas entre la Esposa y el Esposo, como lo observamos en la figura I.9.

Esta representación gráfica, para este caso, nos parece más adecuada, puesto que los miembros de este matrimonio comparten la responsabilidad para realizar ambas actividades.

Anexo II

Cuestionarios de apoyo para las áreas clave de procesos

En este anexo presentamos cinco cuestionarios, que proponemos como material complementario a los capítulos del 3 al 7, que corresponden a las siguientes áreas clave:

- Administración de requisitos.
- Planeación del proyecto de software.
- Seguimiento y control del proyecto de software.
- Aseguramiento de calidad de software.
- Administración de configuraciones de software.

Cada uno de estos cuestionarios inicia con una carátula en donde se especifica el área clave de procesos a la que pertenece, datos de la persona que contestará el cuestionario y el contenido. Para cada uno de los puntos citados en el contenido se hacen las preguntas básicas ¿cómo lo haces ahora? y ¿cómo te gustaría hacerlo?. La primer pregunta se completa con algunas preguntas adicionales que solo sirven de apoyo y no necesariamente se tienen que contestar individualmente.

Taller de interpretación de CMM nivel 2

Cuestionario No. 1

Administración de Requisitos

Fecha: _____ .

Nombre: _____ .

Empresa: _____ .

Contenido:

Administrar requisitos

Asignar responsable

Asignar recursos

Analizar y documentar requisitos

Capacitar en AdR

Revisar requisitos

Incorporar requisitos

Revisar e incorporar cambios

Mediciones y análisis

Revisar actividades AdR

Revisar y/o auditar actividades y productos

Compromiso 1: Administrar requisitos

El proyecto sigue una política organizacional escrita para administrar requisitos.

¿Cómo lo haces ahora?

- a. ¿Existe una política de AdR implícita, explícita o no existe?
- b. ¿Quién o qué grupos siguen esta política?
- c. ¿Qué puntos están incluidos en esta política?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

- a. ¿Cuáles serían los puntos importantes de esta política?
- b. ¿Quién o qué grupo deberían seguir esta política?

Habilidad 1: Asignar responsable.

Para cada proyecto, se asigna la responsabilidad de analizar los requisitos del sistema que corresponden al hardware, software u otros componentes del sistema.

¿Cómo lo haces ahora?

- a. ¿Quién asigna al responsable de AnR (analizar los requisitos)?
- b. ¿Qué actividades realiza el responsable AnR?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

.....
.....
.....

Habilidad 3:Asignar recursos.

Se provee de recursos y financiamiento adecuados para administrar los requisitos.

.....

¿Cómo lo haces ahora?

- a. ¿Quién asigna estos recursos y financiamiento?
- b. ¿Qué tipo de recursos se proveen para la AdR?

c. No lo hago: _____

.....

¿Cómo te gustaría hacerlo?

.....
.....

Habilidad 2: Analizar y documentar requisitos.

Se documenta los requisitos técnicos, no técnicos, así como los criterios de aceptación de los productos de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo realiza esta actividad?
- b. ¿Cuáles son los requisitos que analizan y documentan?
- c. ¿Cómo lo logras? (Herramientas de apoyo, esquema previamente establecido, etc.)

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 4: Capacitar en AdR.

Los miembros del equipo de ingeniería de software y otros grupos relacionados al software se capacitan para administrar requisitos.

¿Cómo lo haces ahora?

- a. ¿Existe una capacitación formal o informal?
- b. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 1: Revisar requisitos.

El grupo de ingeniería de software revisa los requisitos antes de que estos sean incorporados al proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo revisa los requisitos antes de ser incorporados al proyecto de software?
- b. ¿Cuáles son los pasos para llevar a cabo esta revisión? (Referirse a las subprácticas de la actividad 1)

c. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Actividad 2: Incorporar requisitos.

El grupo de ingeniería de software usa los requisitos como la base para los planes, productos de trabajo y actividades.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo incorpora los requisitos al proyecto de software?
- b. ¿Cuáles son los pasos para incorporar los requisitos al proyecto?
- c. ¿Se documenta esta incorporación?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 3: Revisar e incorporar cambios.

Se revisan e incorporan los cambios a los requisitos al proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo puede solicitar un cambio a los requisitos?
- b. ¿Quién o qué grupo es responsable de revisar e incorporar cambios a los requisitos?
- c. ¿Quién autoriza estos cambios?
- d. ¿Qué pasos se siguen para revisar e incorporar cambios?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Medición 1: Mediciones y análisis

Las mediciones se definen y se usan para determinar el estado de las actividades de la administración de los requisitos.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de definir las mediciones?
- b. ¿Qué mediciones se registran?
- c. ¿Cómo se registran estas mediciones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Quién o qué grupo es responsable de registrar las mediciones?
- e. ¿Quién o qué grupo usa estas mediciones?
- f. ¿En que actividades se usan estas mediciones?

g. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 1 y 2: Revisar actividades AdR.

Las actividades para la administración de los requisitos se revisa periódicamente con el administrador.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan las actividades de la administración de requisitos?
- b. ¿Qué puntos se revisan?
- c. ¿Cómo se registran los resultados de estas revisiones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿A quién o qué grupo se da a conocer los resultados de estas revisiones?
- e. ¿Quién o qué grupo usa la información de estas revisiones?
- f. ¿En que actividades se usa la información de estas revisiones?

g. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 3: Revisar y/o auditar actividades y productos de trabajo.

El grupo de aseguramiento de calidad revisa y/o audita las actividades y productos de trabajo para administrar los requisitos.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan y/o auditan las actividades y productos de trabajo para administrar requisitos?
- b. ¿Qué puntos se revisan y/o auditan?
- c. ¿Cómo se registran los resultados de estas revisiones y/o auditorias? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿A quién o qué grupo se da a conocer los resultados de estas revisiones y/o auditorias?
- e. ¿Quién o qué grupo usa la información de estas revisiones?
- f. ¿En que actividades se usa la información de estas revisiones?

g. No lo hago: _____

¿Cómo te gustaría hacerlo?

Taller de interpretación de CMM nivel 2

Cuestionario No. 2

Planeación del Proyecto de Software

Fecha: _____ .

Nombre: _____ .

Empresa: _____ .

Contenido:

- Designar un administrador de proyecto de software
- Planeación del proyecto de software
- Crear, documentar y aprobar descripción de trabajo
- Asignar responsabilidades PPS
- Asignar recursos y financiamiento
- Capacitar en PPS
- Desarrollar propuesta del proyecto
- Revisar compromisos
- Identificar elementos PPS
- Realizar plan de desarrollo
- Documentar plan de desarrollo
- Registrar datos PPS
- Mediciones y análisis
- Revisar actividades PPS
- Revisar y/o auditar actividades y productos

Compromiso 1: Designar un administrador de proyecto de software.

Se designa un administrador de proyecto de software, para que sea responsable de negociar los compromisos y realizar el plan de desarrollo del proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Existe un responsable de negociar los compromisos y/o de realizar el plan de desarrollo? ¿Qué nombramiento recibe este responsable?
- b. ¿Quién designa a este responsable?
- c. ¿Qué actividades realiza este responsable (administrador de proyecto de software)?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Compromiso 2: Planeación del proyecto de software.

El proyecto sigue una política organizacional escrita para planear un proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Existe una política de PPS implícita, explícita o no existe?
- b. ¿Quién o quiénes siguen esta política?
- c. ¿Qué puntos están incluidos en esta política?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

- a. ¿Cuáles serían los puntos importantes de esta política?
- b. ¿Quién o qué grupos deberían seguir esta política?

Habilidad 1: Crear, documentar y aprobar descripción de trabajo.

Para el proyecto de software existe una descripción de trabajo documentada y aprobada.

¿Cómo lo haces ahora?

- a. ¿Realizan una descripción de trabajo en la planeación del proyecto?.
- b. Si se realiza esta descripción, ¿se documenta, se revisa y se aprueba? o ¿Cuál es el proceso para generar esta descripción?
- c. ¿Quién o que grupos son los responsables de documentar, revisar y aprobar esta descripción?
- d. ¿Qué puntos cubre esta descripción de trabajo?

e. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Habilidad 2: Asignar responsabilidades PPS.

Se asignan las responsabilidades para realizar el plan de desarrollo de software.

¿Cómo lo haces ahora?

- a. ¿Quién realiza esta actividad?
- b. ¿En quién o en qué grupos recaen estas responsabilidades?
- c. ¿Cómo se logra? (Herramientas de apoyo, esquema previamente establecido, etc.)

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 3: Asignar recursos y financiamiento.

Se proveen los recursos y financiamiento adecuado para planear el proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Quién asigna estos recursos y financiamiento?
- b. ¿Qué tipo de recursos se proveen para la PPS?
- c. ¿Qué herramientas de soporte, para las actividades de PPS, se proveen?
- d. ¿Qué criterios se utilizan para realizar esta asignación?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 4: Capacitar en PPS.

Los administradores de software, ingenieros de software, y otros individuos involucrados en la planeación del proyecto de software son capacitados en los procedimientos de estimación y planeación de software aplicables a sus áreas de responsabilidad.

¿Cómo lo haces ahora?

- a. ¿Existe una capacitación formal o informal?
- b. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

.....
.....

Actividad 1: Desarrollar propuesta del proyecto.

El grupo de ingeniería de software participa en el equipo de propuesta del proyecto.

.....

¿Cómo lo haces ahora?

- a. ¿Existe un equipo para desarrollar la propuesta del proyecto de software?
- b. ¿Qué grupos participan en el equipo de propuesta?
- c. ¿Qué puntos contiene esta propuesta?

d. No lo hago: _____ .

.....

¿Cómo te gustaría hacerlo?

.....
.....

Actividad 4: Revisar compromisos

Se revisan los compromisos del proyecto de software hechos a individuos y grupos externos a la organización con el administrador gerencial de acuerdo a un procedimiento documentado.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo revisa los compromisos del proyecto de software?
- b. ¿Cuáles son los pasos para revisar los compromisos del proyecto de software?
- c. ¿Se documenta esta revisión? ¿Cómo realizas esta documentación?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 6: Realizar plan de desarrollo.

Se realiza el plan de desarrollo del proyecto de software de acuerdo a un procedimiento documentado.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo realiza el plan de desarrollo de software?
- b. ¿Cuáles son los pasos para realizar el plan de desarrollo de software?
- c. ¿Cuáles son los puntos importantes del plan de desarrollo de software?
- d. ¿Existe un procedimiento para realizar este plan?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 7: Documentar plan de desarrollo.

Se documenta el plan para el proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Cuáles son los puntos que se documentan en el plan de desarrollo de software?
- b. ¿Quién o qué grupo documenta el plan de desarrollo de software?
- c. ¿Cuáles son los pasos para documentar el plan de desarrollo de software?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 15: Registrar datos PPS.

Se registran los datos de planeación de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo registra los datos de la PPS?
- b. ¿Qué datos de la PPS se registran?
- c. ¿Cómo se utilizan estos datos?
- d. ¿Qué herramientas de apoyo se utilizan para registrar los datos de PPS?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Medición 1: Mediciones y análisis

Las mediciones se definen y se usan para determinar el estado de las actividades de la planeación del software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de definir las mediciones?
- b. ¿Qué mediciones se registran?
- c. ¿Cómo se registran estas mediciones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Quién o qué grupo es responsable de registrar las mediciones?
- e. ¿Quién o qué grupo usa estas mediciones?
- f. ¿En que actividades se usan estas mediciones?

g. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 3: Revisar y/o auditar actividades y productos de trabajo.

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

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan y/o auditan las actividades y productos de trabajo para administrar requisitos?
- b. ¿Qué puntos se revisan y/o auditan?
- c. ¿Cómo se registran los resultados de estas revisiones y/o auditorias? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿A quién o qué grupo se da a conocer los resultados de estas revisiones y/o auditorias?
- e. ¿Quién o qué grupo usa la información de estas revisiones?
- f. ¿En que actividades se usa la información de estas revisiones?

g. No lo hago: _____

¿Cómo te gustaría hacerlo?

Taller de interpretación de CMM nivel 2

Cuestionario No. 3

Seguimiento y Control del Proyecto de Software

Fecha: _____ .

Nombre: _____ .

Empresa: _____ .

Contenido:

- Designar un administrador de proyecto de software
- Administrar el proyecto de software
- Aprobar plan de desarrollo de software
- Asignar responsabilidades
- Asignar recursos y financiamiento para SCPS
- Capacitar en SCPS
- Recibir orientación
- Dar seguimiento a actividades y comunicar estado de software
- Revisar plan de desarrollo de software
- Revisar compromisos y cambios a los compromisos
- Comunicar cambios a compromisos
- Registrar datos de mediciones reales y de replaneación
- Realizar revisiones periódicas
- Realizar revisiones formales
- Mediciones y análisis
- Revisar actividades SCPS
- Revisar y/o auditar actividades y productos

.....
.....
.....

Compromiso 1: Designar administrador de proyecto de software.

Se designa un administrador de proyecto de software que sea responsable de las actividades y resultados del proyecto de software.

.....

¿Cómo lo haces ahora?

a. En tu organización, ¿Quién es el responsable de las actividades y resultados del proyecto?

b. No lo hago: _____

.....

¿Cómo te gustaría hacerlo?

.....
.....

Compromiso 2: Administrar el proyecto de software.

El proyecto sigue una política organizacional escrita para administrar el proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Existe una política de PPS implícita, explícita o no existe?
- b. ¿Qué puntos integran esta política?
- c. ¿Quién o qué grupos siguen esta política?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

- a. ¿Cuáles serían los puntos importantes de esta política?
- b. ¿Quién o quiénes deberían seguir esta política?

Habilidad 1: Aprobar plan desarrollo de software

Se documenta y aprueba un plan de desarrollo de software para el proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Existe una aprobación formal del plan de desarrollo de software?
- b. ¿Qué puntos se documentan?
- c. ¿Quién o qué grupo aprueba el plan de desarrollo de software?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 2: Asignar responsabilidades SCPS.

El administrador de proyecto de software asigna explícitamente las responsabilidades para los productos de trabajo y actividades de software.

¿Cómo lo haces ahora?

- a. ¿Se realiza una asignación explícita de las responsabilidades sobre los productos de trabajo y actividades de software?
- b. ¿Cómo lo logras? (Herramientas de apoyo, asignación previamente establecida, etc.)

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 3: Asignar recursos y financiamiento para SCPS.

Se proveen los recursos y financiamiento adecuado para dar seguimiento al proyecto del software.

¿Cómo lo haces ahora?

- a. ¿Quién asigna estos recursos y financiamiento?
- b. ¿Qué recursos y herramientas se proveen para el SCPS?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 4: Capacitar en SCPS.

Se capacitan los administradores del software para administrar los aspectos técnicos y del personal del proyecto del software.

¿Cómo lo haces ahora?

- a. ¿Existe una capacitación formal o informal?
- b. ¿Quién recibe esta capacitación?
- c. ¿En qué etapa del proyecto se realiza esta capacitación?
- d. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 5: Recibir orientación.

Los administradores de software de primera-línea reciben orientación en los aspectos técnicos del proyecto del software.

¿Cómo lo haces ahora?

- a. ¿Existe una orientación formal o informal?
- b. ¿Quién recibe esta orientación?
- c. ¿Cómo se logra esta orientación?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Actividad 1, 5...10: Dar seguimiento a actividades y comunicar estado de software

Se usa un plan de desarrollo de software documentado para dar seguimiento a las actividades y comunicar el estado del software. Se lleva a cabo el seguimiento del tamaño de los productos de trabajo, del esfuerzo, de los recursos computacionales críticos, del calendario, de las actividades técnicas de ingeniería de software y de los riesgos del software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de dar seguimiento a las actividades del proyecto de software?
- b. Selección los elementos de la planeación del proyecto de software a los que se da seguimiento:
 - Actividades de IS
 - Tamaño de los productos de trabajo
 - Esfuerzo y costo
 - Recursos computacionales críticos
 - Calendario del proyecto
 - Riesgos
 - Otro _____
- c. ¿A quién se da a conocer el resultado de esta actividad?
- d. ¿Cómo se logra? (Herramientas de apoyo, formatos, etc.)

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 2: Revisar plan de desarrollo de software

Se revisa el plan de desarrollo de software del proyecto de acuerdo a un procedimiento documentado.

¿Cómo lo haces ahora?

- a. ¿Existe un procedimiento documentado para modificar el plan de desarrollo?
- b. ¿Quién o qué grupo modifica el plan de desarrollo de software?
- c. ¿Qué aspecto o puntos se modifican?
- d. ¿Cómo se documenta esta modificación?
- e. ¿A quién o a quiénes revisan las modificaciones?

f. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 4: Comunicar cambios a compromisos

Los cambios aprobados a los compromisos que afectan al proyecto del software se comunican a los miembros del grupo de ingeniería y otros grupos relacionados al software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo comunica los cambios a los compromisos?
- b. ¿Cuál es el mecanismo para comunicar estos cambios?
- c. ¿A qué grupos se les comunica estos cambios?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 11: Registrar datos de mediciones reales y de replaneación.

Se registran los datos de mediciones reales y los datos de replaneación para el proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo realiza un registro datos de mediciones reales y de replaneación?
- b. ¿Que datos se registran?
- c. ¿Cuál es el uso de esta información?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 12: Realizar revisiones periódicas.

El grupo de ingeniería de software realiza revisiones periódicas internas para dar seguimiento al avance técnico, planes, realización, y resultados contra el plan de desarrollo de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo realiza revisiones periódicas internas para dar seguimiento al proyecto?
- b. ¿A qué puntos o aspectos se revisan?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 13: Realizar revisiones formales.

Se realizan revisiones formales para conocer los logros y resultados del proyecto de software en un evento o fecha previamente seleccionados de acuerdo a un procedimiento documentado.

¿Cómo lo haces ahora?

- a. ¿Existe un procedimiento documentado para realizar estas revisiones formales?
- b. ¿Quién o qué grupo realiza revisiones formales para conocer los logros y resultados del proyecto de software?
- c. ¿Qué puntos o aspectos se revisan?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Medición 1: Mediciones y análisis

Las mediciones se definen y se usan para determinar el estado de las actividades del seguimiento y control del proyecto de software.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de definir las mediciones?
- b. ¿Qué mediciones se registran?
- c. ¿Cómo se registran estas mediciones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Quién o qué grupo es responsable de registrar las mediciones?
- e. ¿Para que se usan estas mediciones?

f. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 1 y 2: Revisar actividades SCPS.

Las actividades para el seguimiento y control del proyecto de software se revisan periódicamente con el administrador.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan las actividades del SCPS?
- b. ¿Qué puntos se revisan?
- c. ¿Cómo se registran los resultados de estas revisiones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Para que se usan los resultados de estas revisiones?

e. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Verificación 3: Revisar y/o auditar actividades y productos de trabajo.

El grupo de aseguramiento de calidad de software revisa y/o audita las actividades y productos de trabajo para seguir y controlar el proyecto del software y reporta resultados.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan y/o auditan las actividades y productos de trabajo para seguir y controlar el proyecto?
- b. ¿Qué puntos se revisan y/o auditan?
- c. ¿Cómo se registran los resultados de estas revisiones y/o auditorias? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Para que se utilizan los resultados de estas revisiones y/o auditorias?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Taller de interpretación de CMM nivel 2

Cuestionario No. 4

Aseguramiento de Calidad de Software

Fecha: _____ .

Nombre: _____ .

Empresa: _____ .

Contenido:

- Asegurar calidad de software
- Constituir el grupo de AC
- Asignar recursos y financiamiento
- Recibir capacitación
- Recibir orientación sobre el grupo de AC
- Preparar plan de AC
- Preparar y revisar plan de desarrollo, estándares y procedimientos de software
- Realizar actividades AC
- Documentar y manejar desviaciones
- Dirigir revisiones de actividades y hallazgos de AC
- Mediciones y análisis
- Revisar actividades AC
- Revisar las actividades y productos de trabajo

Compromiso 1: Asegurar calidad de software

El proyecto sigue una política organizacional escrita para llevar a cabo el aseguramiento de calidad de software.

¿Cómo lo haces ahora?

- a. ¿Existe una política de AC implícita, explícita o no existe?
- b. ¿Qué puntos integran esta política?
- c. ¿Quién o qué grupos siguen esta política?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

- a. ¿Cuáles serían los puntos importantes de esta política?
 - b. ¿Quién o quiénes deberían seguir esta política?
-
-

.....

Habilidad 1: Constituir el grupo de AC.

Existe un grupo responsable de coordinar y llevar a cabo el AC del proyecto (es decir el grupo de AC).

.....

¿Cómo lo haces ahora?

- a. En tu organización, ¿existe una persona o grupo que sea responsable de coordinar y llevar a cabo el AC? ¿Que nombramiento recibe?

b. No lo hago: _____ .

.....

¿Cómo te gustaría hacerlo?

.....

Habilidad 2: Asignar recursos y financiamiento.

Se proveen los recursos y financiamiento adecuados para realizar las actividades de AC.

¿Cómo lo haces ahora?

- a. ¿Quién asigna estos recursos y financiamiento?
- b. ¿Qué recursos se proveen?
- c. ¿Qué herramientas de soporte, para las actividades de AC, se proveen?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Habilidad 3: Recibir capacitación.

Los miembros del grupo de AC se capacitan para realizar sus actividades de AC.

¿Cómo lo haces ahora?

- a. ¿Existe una capacitación formal o informal?
- b. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)
- c. ¿En qué puntos se centra esta capacitación? (ver subprácticas de la Habilidad 3)

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 4: Recibir orientación sobre el grupo de AC.

Los miembros del proyecto de software reciben orientación sobre el rol, responsabilidades, autoridad e importancia del grupo de AC.

¿Cómo lo haces ahora?

- a. ¿Existe una orientación formal o informal?
- b. ¿Quién recibe esta orientación?
- c. ¿Qué autoridad tiene el grupo de AC?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Actividad 1: Preparar plan de AC

Se prepara un plan de AC para el proyecto del software de acuerdo a un procedimiento documentado.

¿Cómo lo haces ahora?

- a. ¿En que momento se prepara el plan de AC?
- b. ¿Quién o qué grupo revisa el plan de AC?
- c. ¿Qué puntos se consideran en el plan de AC?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 3: Preparar y revisar plan de desarrollo, estándares y procedimientos de software

El grupo de AC participa en la preparación y revisión del plan de desarrollo, estándares y procedimientos del proyecto de software.

¿Cómo lo haces ahora?

- a. ¿En qué elementos del proyecto de software el grupo de AC de tu organización participa en su preparación y revisión?
- plan de desarrollo de software,
 - estándares,
 - procedimientos,
 - otros elementos del proyecto. ¿Cuáles?
- b. ¿En qué consiste la participación del grupo de AC con respecto a la preparación y revisión del plan de desarrollo de software, estándares y procedimientos del proyecto?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Actividad 2,4,5,6: Realizar actividades AC.

Las actividades del grupo de AC se realizan de acuerdo con el plan de AC.

¿Cómo lo haces ahora?

- a. ¿El grupo de AC revisa las actividades de ingeniería de software? ¿Qué aspectos o puntos revisa? ¿Qué acciones realiza para lograr esta revisión?
- b. ¿El grupo de AC audita los productos de trabajo de software? ¿Cómo logra esta auditoría?
- c. ¿El grupo de AC reporta periódicamente los resultados de sus actividades? ¿A quién o a que grupo reporta estos resultados?

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

.....
.....
.....

Actividad 7: Documentar y manejar desviaciones

Las desviaciones identificadas en las actividades y en los productos de trabajo de software se documentan y se manejan de acuerdo a un procedimiento documentado.

.....

¿Cómo lo haces ahora?

- a. ¿Qué acciones se realizan en el manejo de las desviaciones de las actividades y en los productos de trabajo de software?
- b. Además del grupo de AC ¿Quién o qué grupos intervienen en este manejo?

c. No lo hago: _____

.....

¿Cómo te gustaría hacerlo?

.....
.....

Actividad 8: Dirigir revisiones de actividades y hallazgos de AC.

El grupo de AC dirige periódicamente revisiones de sus actividades y hallazgos con el personal de AC del cliente, en caso de que aplique.

¿Cómo lo haces ahora?

- a. ¿En algún proyecto, el cliente ha contado con personal propio de AC?
- b. ¿Qué puntos o aspectos se revisan?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

.....
.....

Medición 1: Mediciones y análisis

Las mediciones se definen y se usan para determinar el estado de las actividades del AC.

.....

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de definir las mediciones?
- b. ¿Qué mediciones se registran?
- c. ¿Cómo se registran estas mediciones? (Herramientas de apoyo, formatos especiales, etc.)

d. No lo hago: _____ .

.....

¿Cómo te gustaría hacerlo?

.....
.....

Verificación 1 y 2: Revisar actividades AC.

Las actividades del AC se revisan periódicamente con el administrador.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan las actividades del AC?
- b. ¿Qué puntos se revisan?
- c. ¿Cómo se registran los resultados de estas revisiones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Cómo se utilizan los resultados de estas revisiones?

e. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 3: Revisar las actividades y productos de trabajo.

Los expertos independientes del grupo de AC revisan periódicamente las actividades y productos de trabajo de software del grupo de AC del proyecto.

¿Cómo lo haces ahora?

- a. ¿Qué puntos o aspectos se revisan las actividades y productos de trabajo de software del grupo de AC?
- b. ¿Cómo se registran los resultados de estas revisiones? (Herramientas de apoyo, formatos especiales, etc.)
- c. ¿Cómo se utilizan los resultados de estas revisiones?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Taller de interpretación de CMM nivel 2

Cuestionario No. 5

Administración de Configuraciones de Software

Fecha: _____ .

Nombre: _____ .

Empresa: _____ .

Contenido:

- Administrar configuraciones.
- Constituir grupo de ACS.
- Constituir grupo de ACS.
- Asignar recursos y financiamiento.
- Capacitar en ACS.
- Capacitar en ACS.
- Preparar plan de ACS. Aprobar plan ACS.
- Actividades ACS.
- Mediciones y análisis.
- Revisar actividades ACS.
- Auditar líneas base.
- Revisar y/o auditar actividades y productos.

.....

Compromiso 1: Administrar configuraciones

El proyecto sigue una política organizacional escrita para llevar a cabo la administración de configuraciones de software (ACS).

.....

¿Cómo lo haces ahora?

- a. ¿Existe una política de ACS implícita, explícita o no existe?
- b. ¿Qué puntos integran esta política?
- c. ¿Quién o qué grupos siguen esta política?

d. No lo hago: _____

.....

¿Cómo te gustaría hacerlo?

- a. ¿Cuáles serían los puntos importantes de esta política?
- b. ¿Quién o quiénes deberían seguir esta política?

.....

Habilidad 1: Constituir CCCS.

Existe o se constituye un comité que tiene la autoridad para administrar las líneas base del proyecto de software (es decir, un comité de control de configuración de software -CCCS).

¿Cómo lo haces ahora?

- a. En tu organización, ¿existe un comité o grupo que tenga la autoridad para administrar las líneas base del proyecto de software? ¿Que nombramiento recibe?
- b. ¿Cuáles son las obligaciones [tareas, acciones, actividades] de este comité o grupo?

c. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Habilidad 2: Constituir grupo de ACS.

Existe un grupo que es responsable de coordinar y llevar a cabo la ACS para el proyecto (es decir el grupo de ACS).

¿Cómo lo haces ahora?

- a. En tu organización, ¿existe una persona o grupo que sea responsable de coordinar y llevar a cabo la ACS para el proyecto? ¿Que nombramiento recibe?
- b. ¿Cuáles son las obligaciones [tareas, acciones, actividades] de esta persona o grupo?

c. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 3: Asignar recursos y financiamiento.

Se proveen los recursos y el financiamiento adecuado para realizar las actividades de ACS.

¿Cómo lo haces ahora?

- a. ¿Quién asigna estos recursos y financiamiento?
- b. ¿Qué recursos se proveen?
- c. ¿Qué herramientas de soporte, para las actividades de ACS, se proveen?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 4: Capacitar en ACS.

Se capacitan los miembros del grupo de ACS en los objetivos, procedimientos y métodos para realizar las actividades de ACS.

¿Cómo lo haces ahora?

- a. ¿Existe una capacitación formal o informal?
- b. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)
- c. ¿En qué puntos se centra esta capacitación? (ver subprácticas de la Habilidad 4)

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Habilidad 5: Capacitar en ACS.

Se capacitan los miembros del grupo de ingeniería de software y otros grupos relacionados al software para realizar sus actividades de ACS.

¿Cómo lo haces ahora?

- a. Además del grupo de ACS, ¿quién o qué grupos se capacitan en ACS?
- b. ¿Existe una capacitación formal o informal?
- c. ¿Cómo se logra esta capacitación? (Experiencias dentro de la empresa, expertos externos a la empresa)
- d. ¿En qué puntos se centra esta capacitación? (ver subprácticas de la Habilidad 5)

e. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

.....

Actividad 1,2: Preparar plan de ACS. Aprobar plan ACS. Actividades ACS

Actividad 1: Se prepara un plan de ACS para cada proyecto del software de acuerdo a un procedimiento documentado.

Actividad 2: Se usa un plan de ACS documentado y aprobado como base para realizar las actividades de ACS.

.....

¿Cómo lo haces ahora?

- a. ¿En que momento se prepara el plan de ACS?
- b. ¿Quién o qué grupo revisa el plan de ACS?
- c. ¿Qué puntos se consideran en el plan de ACS?

d. No lo hago: _____

.....

¿Cómo te gustaría hacerlo?

.....

Actividad 3,...,10: Actividades ACS

Se usa un plan de ACS documentado y aprobado como base para realizar las actividades de ACS.

¿Cómo lo haces ahora?

- a. ¿Quién o que grupos establecen el sistema bibliotecario? ¿Cuál es el alcance de SB? (Referirse a las subprácticas de la Actividad 3)
 - b. ¿Cuáles son los criterios para identificar los productos de trabajo de software que serán puestos bajo la ACS? ¿Qué otras actividades (acciones) se involucran en esta identificación?
 - c. ¿Cómo se administran [manejan] las solicitudes de cambios y reportes de las entidades/unidades de configuración?
 - d. ¿Cómo se controlan los cambios a las líneas base?
 - e. ¿Qué pasos sigue la ACS de los productos que están en la biblioteca de las líneas base de software? [¿Cómo se controlan los productos que están en la biblioteca de las líneas base de software?]
 - f. ¿Cómo se registra el estado de las entidades/unidades de la configuración?
 - g. ¿Qué tipo de reportes se realizan para documentar las actividades de ACS y el contenido de las líneas base? ¿Quién o que grupos tienen acceso a estos reportes?
 - h. ¿Cuál es el procedimiento para realizar las auditorías a las líneas base de software?

 - i. No lo hago: _____
-

¿Cómo te gustaría hacerlo?

Medición 1: Mediciones y análisis

Las mediciones se definen y se usan para determinar el estado de las actividades del ACS.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupo es responsable de definir las mediciones?
- b. ¿Qué mediciones se registran?
- c. ¿Cómo se registran estas mediciones? (Herramientas de apoyo, formatos especiales, etc.)

d. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Verificación 1 y 2: **Revisar actividades ACS.**

Las actividades del ACS se revisan periódicamente con el administrador.

¿Cómo lo haces ahora?

- a. ¿Quién o qué grupos revisan las actividades del ACS?
- b. ¿Qué puntos se revisan?
- c. ¿Cómo se registran los resultados de estas revisiones? (Herramientas de apoyo, formatos especiales, etc.)
- d. ¿Cómo se utilizan los resultados de estas revisiones?

e. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Verificación 3: Auditar líneas base.

El grupo de ACS audita periódicamente las líneas base para verificar que ellas conforman la documentación que las define.

¿Cómo lo haces ahora?

- a. ¿Qué puntos o aspectos se auditan a las líneas base? ¿Quién o que grupo realiza esta auditoría?
- b. ¿Cómo se registran los resultados de estas auditorías? (Herramientas de apoyo, formatos especiales, etc.)
- c. ¿Cómo se utilizan los resultados de estas auditorías?

d. No lo hago: _____

¿Cómo te gustaría hacerlo?

Verificación 4: Revisar y/o auditar actividades y productos.

El grupo de aseguramiento de calidad de software revisa y/o audita las actividades y productos de trabajo de la ACS y reporta resultados.

¿Cómo lo haces ahora?

- e. ¿Qué puntos o aspectos se revisan (o se auditan) de las actividades y de los productos de trabajo de software de la ACS?
- f. ¿Cómo se registran los resultados de estas revisiones (o auditorías)? (Herramientas de apoyo, formatos especiales, etc.)
- g. ¿Cómo se utilizan los resultados de estas revisiones (o auditorías)?

h. No lo hago: _____ .

¿Cómo te gustaría hacerlo?

Referencias

[Booch et al., 1999]

Booch, G., Rumbaugh, J. and Jacobson, I., *"The Unified Modeling Language, user Guide"*, Addison-Wesley, 1999.

[CMM, 1994]

"CMM, The Capability Maturity Model", Carnegie Mellon University, Software Engineering Institute Series in Software Engineering, Addison-Wesley, 1994.

[El Emam et al., 1998]

El Emam K., J.-N. Drouin and W. Melo, *"SPICE The Theory and Practice of Software Process Improvement and Capability Determination"*, IEEE Computer Society, 1998.

[Humphrey, 1987]

Humphrey W. S., *"Characterizing the Software Process: A Maturity Framework"*, Software Engineering Institute, CMU/SEI-87-TR-11, DTIC Number ADA182895, June 1987.

[Humphrey, 1989]

Humphrey W. S., *"Managing the Software Process"*, Addison-Wesley, 1989.

[Oktaba and Ibargüengoitia, 1998]

Oktaba, H. and Ibargüengoitia González, G., *"Software Processes Modeled with Objects: Static View"*, Computación y Sistemas, Iberoamerican Journal of Computer Science, Centro de Investigación en Computación, Instituto Politécnico Nacional, México, 1, 4 (1998), 228-238.

[Oktaba and Ibargüengoitia, 1999]

Oktaba, H. and Ibargüengoitia González, G., *"Basic Measures Patterns for Software Process Improvement"*, aceptado para el 9th International Software Quality Conference, 6-8 de Octubre de 1999, Boston.

[Oktaba and Alquicira, 1999-1]

Oktaba, H. and Alquicira Esquivel, C., *"Modelo gráfico de la Administración de Requisitos del SW-CMM[®] nivel 2"*, memorias del Segundo Encuentro Nacional de Computación (ENC'99), 12-15 de Septiembre de 1999, Pachuca, Hgo.

[Oktaba and Alquicira, 1999-2]

Oktaba, H. and Alquicira Esquivel, C., *"Modelo gráfico de la Administración de Configuraciones del SW-CMM[®] nivel 2"*, memorias del 4^o. Seminario Estratégico "Seminario de Calidad en Ingeniería de Software", 11-13 de mayo de 1999, Mexico, D.F.

[Rumbaugh et al., 1999]

Rumbaugh, J., I. Jacobson and G. Booch, *"The Unified Modeling Language, Reference Manual"*, Addison-Wesley, 1999.