



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA  
COMPUTACIÓN

**REPRESENTACIÓN DEL MODELO DE PROCESOS PARA LA  
INDUSTRIA DE SOFTWARE (MOPROSOFT) EN EL  
METAMODELO DE INGENIERÍA DE PROCESOS, SOFTWARE  
Y SISTEMAS (SPEM) 2.0**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

***MAESTRO EN INGENIERÍA  
(COMPUTACIÓN)***

**P R E S E N T A:**

**DANIEL NAVA NAVA**

**DIRECTOR DE TESIS: DRA. HANNA OKTABA**

México, D.F.

2010.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# Índice

1 INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	2
1.2 PROPUESTA DE SOLUCIÓN. ....	4
1.3 OBJETIVO GENERAL.....	4
1.4 OBJETIVOS ESPECÍFICOS.....	4
1.5 ALCANCE.....	5
1.6 METODOLOGÍA DE TRABAJO.....	5
1.7 HERRAMIENTAS UTILIZADAS.....	6
1.8 RESULTADOS.....	6
1.9 ORGANIZACIÓN DE LA TESIS. ....	7
2 MODELO DE PROCESOS PARA LA INDUSTRIA DE SOFTWARE (MOPROSOFT) 8	
2.1 ANTECEDENTES.....	8
2.2 PROPÓSITO DE MOPROSOFT.....	9
2.3 DEFINICIONES BÁSICAS. ....	10
2.4 PATRÓN DE PROCESOS.....	12
2.5 DESCRIPCIÓN DEL MODELO.....	13
3 SOFTWARE & SYSTEMS PROCESS ENGINEERING METAMODEL (SPEM) 2.0.. 17	
3.1 CARACTERISTICAS GENERALES.....	17
3.2 COMO SE UTILIZA EL METAMODELO SPEM 2.0.....	20
3.2.1 ORGANIZACIÓN DE SPEM 2.0.....	23
3.3 ORGANIZACIÓN DE UN REPOSITORIO EN SPEM 2.0.....	26
3.3.1 JERARQUIA DE DESGLOSE DE TRABAJO.....	35
3.4 HERRAMIENTAS DISPONIBLES PARA SPEM 2.0.....	36
3.5 DESCRIPCIÓN DE LA HERRAMIENTA ECLIPSE PROCESS FRAMEWORK COMPOSER (EPFC).....	37
4 MODELADO POR NIVEL DE CAPACIDAD DE MOPROSOFT.....	39
4.1 ESTRATEGIA PARA EL MODELADO SPEM-MOPROSOFT.....	39
4.2 MAPEO SPEM-MOPROSOFT.....	41
4.3 CONVENCIONES PARA EL NOMBRADO DE LOS DIFERENTES ELEMENTOS SPEM-MOPROSOFT.....	42
4.4 ANÁLISIS DEL PROCESO DE GESTIÓN DE NEGOCIO Y MODELADO EN SPEM 2.0.....	44
4.5 INTEGRACIÓN DE TODOS LOS PROCESOS.....	52
4.6 MODELADO DE LOS NIVELES 1, 2, 3 Y 4 DE MOPROSOFT.....	53
4.6.1 MODELADO PARA EL NIVEL 2 DE MOPROSOFT.....	53
4.6.2 MODELADO PARA EL NIVEL 3 DE MOPROSOFT.....	56
4.6.3 MODELADO PARA EL NIVEL 4 DE MOPROSOFT.....	57
5 ANÁLISIS DEL MODELADO POR NIVEL DE CAPACIDAD Y PRESENTACIÓN DE RESULTADOS.....	58
5.1 REPORTE DEL MODELADO SPEM-MOPROSOFT POR NIVELES DE CAPACIDAD.....	59

5.1.1 NIVEL DE CAPACIDAD 1.....	59
5.1.2 NIVEL DE CAPACIDAD 2.....	65
5.1.3 NIVEL DE CAPACIDAD 3.....	66
5.2 ANÁLISIS DEL MODELADO SPEM-COMPETISOFT.....	68
5.3 COMPARACIÓN ENTRE LOS MODELADOS SPEM-MOPROSOFT Y SPEM-COMPETISOFT.....	69
5.4 VENTAJAS Y DESVENTAJAS DE AMBOS MODELADOS.....	70
6 CONCLUSIONES Y TRABAJO A FUTURO.....	71
7 REFERENCIAS .....	73
8 APÉNDICE A.....	75
8.1 MANUAL DE ECLIPSE PROCESS FRAMEWORK COMPOSER (EPFC).....	75
8.2 CONTENIDO DE MÉTODO.....	75
8.2.1 CATEGORIZACIÓN DE ELEMENTOS DE MÉTODO.....	77
8.2.2 CREANDO PROCESOS.....	78
9 GLOSARIO.....	81

# 1 INTRODUCCIÓN.

En los últimos años se ha visto cómo la industria de software tiene un efecto transversal en la economía de un país. Dado el gran potencial con que cuenta México para desarrollar esta industria, la Secretaría de Economía, en coordinación con organismos empresariales y empresas del sector, se ven en la necesidad de crear una serie de estrategias para que las empresas nacionales cuenten con lo más altos estándares de calidad que les permita participar exitosamente en este mercado tan competitivo y exigente (ProSoft, 2009).

Debido a esto, la Facultad de Ciencias de la UNAM, con un equipo dirigido por la Dra. Hanna Oktaba, a petición de esta misma secretaría, define el Modelo de Procesos para la industria de Software en México (MoProSoft), el cual constituye la base de la norma mexicana para la industria de desarrollo y mantenimiento de Software NMX-I-059/02-NYCE-2005. (NYCE, 2009).

Este modelo, a diferencia de otros como CMMI (Capability Maturity Model® Integration) (SEI, 2009), está dirigido a las pequeñas y medianas empresas de desarrollo y/o mantenimiento de sistemas. Incorpora las mejores prácticas y surge bajo las premisas: fácil de entender, fácil de aplicar, adopción no costosa y que represente una base para la adopción de otros modelos tales como el ya mencionado CMMI, ISO 9000:2000 (Ventura, 2006).

Asimismo, MoProSoft es un modelo flexible es decir, capaz de adaptarse al nivel de madurez de las organizaciones que lo implementan, ya sea omitiendo o modificando productos, actividades o procesos. Por tal motivo adquiere mucha importancia contar con un modelo que permita hacer instancias del mismo y poder adaptarlo a las necesidades, según se requiera.

Por otro lado SPEM (*Software & Systems Process Engineering*

*Metamodel*) versión 2.0 (SPEM, 2007) es un estándar internacional para dar soporte al desarrollo, gestión y administración de procesos de software.

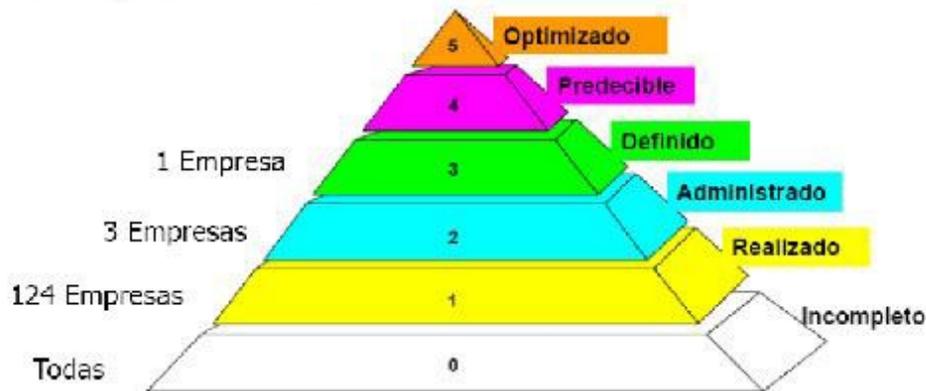
Al aplicar los conceptos de este estándar de procesos de software genérico se pueden construir mediante instanciación los modelos de procesos de software.

### 1.1 PLANTEAMIENTO DEL PROBLEMA.

Las Micro, Pequeñas y Medianas empresas (MiPyME's) representan más del 99 % de empresas en México, y constituyen la base de la economía, convirtiéndose en potenciales creadoras de empleo y reactivadoras de la misma (Contacto PyME, 2009).

En este grupo existen más de 2500 empresas de software (Rosas, 2006). De ellas, sólo un pequeño porcentaje se ha certificado en la norma mexicana MoProSoft (fig. 1)

#### No. Empresas MoProSoft



<http://www.nyce.org.mx/dictamenes.htm>

Figura 1 – Número de empresas MoProSoft

Como se observa en la figura 1, el total de empresas certificadas es 128. Esto indica que, por un lado, el problema puede ser la norma y, por el otro, que las empresas no tienen interés en certificarse. Por lo

tanto en este trabajo se analizara lo referente a la norma, el cual es el siguiente:

MoProSoft es un “modelo descriptivo” que surge bajo las premisas “fácil de entender, fácil de aplicar y adopción no costosa”; sin embargo, dada la flexibilidad del modelo y debido a que está descrito mediante lenguaje natural, en ocasiones se tienen interpretaciones erradas que dificulten obtener las premisas antes mencionadas.

Estas malas interpretaciones indican que la semántica del modelo en ciertas partes es ambigua, pero además, hay varias preguntas que surgen debido a como está el modelo estructurado, las cuales son las siguientes:

- ¿El nivel de detalle o granularidad en MoProSoft es el adecuado para aplicarlo?
- ¿Se pueden identificar claramente cuáles son las actividades que debe desempeñar un rol?
- ¿Los procesos de MoProSoft, están bien estructurados?

Por otro lado, la Secretaría de Economía planteo 7 estrategias básicas para lograr los objetivos del Programa de Desarrollo de la Industria de Software, PROSOFT.

De una de estas estrategias, específicamente la sexta, “Alcanzar niveles internacionales en capacidad de procesos” surge el Modelo de Procesos para la industria de Software, MoProSoft (Este tema se desarrolla en el punto 2.1 de la presente tesis).

Al conocer que MoProSoft tiene este antecedente, debemos de preguntarnos lo siguiente:

- ¿Las empresas que implementan MoProSoft en su

versión actual, alcanzan el nivel internacional en capacidad de procesos planteado en las estrategias del fondo Prosoft? (ProSoft, 2009).

## **1.2 PROPUESTA DE SOLUCIÓN.**

Se propone crear el modelado por niveles de capacidad de MoProSoft (modelo descriptivo) en SPEM 2.0 (modelo formal) y documentarlo en EPFC.

Este modelado considerará los procesos, subprocessos, actividades, tareas, roles, productos, y las relaciones existentes entre ellos.

Así mismo, se pretende compararlo con otro modelado realizado para Competisoft, con la finalidad de verificar que el modelado SPEM-MoProSoft esté bien estructurado y que esté acorde al estándar internacional SPEM 2.0.

Y finalmente, verificar que el nivel de detalle alcanzado es el adecuado.

## **1.3 OBJETIVO GENERAL.**

Representación del Modelo de Procesos para la Industria de Software (MoProSoft) en el Metamodelo de Ingeniería de Procesos, Software y Sistemas (SPEM) 2.0 y documentarlo.

## **1.4 OBJETIVOS ESPECÍFICOS.**

- Crear un modelado mediante SPEM 2.0 de MoProSoft por nivel de capacidad. Los niveles de capacidad considerados en la versión coloreada de MoProSoft y que serán modelados mediante SPEM 2.0 son los siguientes:

<b>Nivel</b>	<b>Capacidad de proceso</b>
1	Realizado

2	Gestionado
3	Establecido
4	Predecible

- Realizar una revisión y comparación de las herramientas existentes tanto en el mercado como en el ámbito académico que nos permitan crear, representar el modelado realizado y seleccionar la que mejor se adecue a nuestros objetivos.
- Revisión de la consistencia de cada uno de los modelados de los niveles de capacidad de MoProSoft.
- Realizar un análisis de los resultados, y compararlo con otro modelado (SPEM-Competisoft), definiendo ventajas y desventajas de cada uno de ellos.
- Documentar el modelado en Eclipse Process Framework Composer.

### **1.5 ALCANCE.**

Se creará un modelado mediante SPEM 2.0 para cada uno de los niveles de capacidad MoProSoft.

Se hará una comparación de los modelados SPEM-MoProSoft y SPEM-Competisoft.

### **1.6 METODOLOGÍA DE TRABAJO.**

- Estudio del metamodelo SPEM 2.0.
- Estudio de la estructura de MoProSoft
- Creación de la representación de un proceso del modelo MoProSoft en SPEM 2.0
- Verificación de la representación de un proceso del modelo MoProSoft en SPEM 2.0 con un experto.
- Modelado del nivel de capacidad 1,2,3 y 4 de MoProSoft en SPEM 2.0.

- Análisis y selección de una herramienta, de las existentes en el mercado y en ámbito escolar que nos sirvan para representar el modelado SPEM-MoProSoft.
- Estudio de la herramienta seleccionada.
- Representación del nivel de capacidad 1,2,3 y 4 de MoProSoft en la herramienta seleccionada.
- Verificación del modelado y representación del nivel 1,2,3 y 4 de capacidad de MoProSoft en SPEM 2.0 con un experto.
- Publicación de los resultados, tesis.
- Creación de sitio en <http://www.mcc.unam.mx/moprosoft/tesis> que contenga la versión electrónica de MoProSoft, así como los elementos usados y generados en la presente tesis.

## 1.7 HERRAMIENTAS UTILIZADAS.

Para el presente trabajo se hizo un análisis de las diferentes herramientas que se apegan al estándar SPEM 2.0 y como resultado la herramienta a utilizar es:

- *Eclipse Process Framework Composer* (Kroll, 2009).

## 1.8 RESULTADOS.

Se logró modelar cada uno de los niveles de capacidad de MoProSoft en SPEM 2.0.

Se solucionaron los defectos encontrados de MoProSoft durante el modelado y estandarización, así también, los 8 defectos encontrados en la tesis “*Utilización de redes de **Petri** coloreadas para la validación de la consistencia del Modelo de Procesos para la industria de software (MoProSoft)*” (Ramirez, 2007).

Se generó por cada nivel de capacidad un plugin, el cual estará disponible junto con la herramienta EPFC para su uso inmediato.

## 1.9 ORGANIZACIÓN DE LA TESIS.

Este trabajo de tesis está estructurado en 6 capítulos, 1 apéndice y las referencias bibliográficas.

En el segundo capítulo se describe el Modelo de Procesos para la Industria de Software (MoProSoft). Se explican sus antecedentes, conceptos básicos que serán usados en el modelado, descripción del modelo.

En el tercer capítulo se habla del Metamodelo de Ingeniería de procesos, software y sistemas (SPEM) 2.0. Se describen características generales, como funciona el metamodelo, como está estructurado, la jerarquía de desglose de trabajo y la herramienta Eclipse Process Framework para administrar procesos.

En el capítulo 4 se realiza el modelado por niveles de capacidad de MoProSoft en SPEM 2.0, para ello se define una estrategia para el modelado SPEM-MoProSoft, se genera un mapeo entre los elementos de MoProSoft y los conceptos de SPEM 2.0, se explica como se genera el modelado de un proceso y como se hace la integración de todos.

En el capítulo 5, se hace un análisis de cada uno de los modelados por nivel de capacidad, describiendo los defectos encontrados y como han sido solucionados. Así también, se hace una comparación de los modelados SPEM-MoProSoft y SPEM-Competisoft.

En el capítulo 6 se describen las conclusiones a las que se llegó con el presente trabajo y se identifican los trabajos a futuro.

En el apéndice se describen los pasos de como se puede generar el modelado en la herramienta *Eclipse Process Framework Composer*.

## **2 MODELO DE PROCESOS PARA LA INDUSTRIA DE SOFTWARE (MOPROSOFT)**

### **2.1 ANTECEDENTES.**

La industria del software se ha convertido en una actividad económica muy importante a nivel mundial.

Esta industria se caracteriza por generar un alto valor agregado, no ser contaminante, contribuir con la modernización del gobierno, propiciar innovación tecnológica y generar empleos bien remunerados.

Es evidente que este sector sigue estando dominado por los países desarrollados, sin embargo, en los últimos años se ha observado que la creciente demanda no puede ser satisfecha con la oferta interna. Tal es el caso de que empresas europeas y estadounidenses calculan que existe un déficit de 41,000 desarrolladores de software (Marchal, 2009). Por lo tanto se observa cada vez más que la producción de software está siendo realizada en países en desarrollo.

Dada la situación económica y geográfica de México, se presenta una enorme oportunidad de desarrollo por lo que resulta imperativo alentar el crecimiento de este sector. Por tal motivo, la Secretaría de Economía crea el Programa de Desarrollo de la Industria de Software, PROSOFT, el cual tiene por objetivo “Lograr una producción anual de software de 5,000 millones de dólares, alcanzar el promedio mundial de gasto en tecnologías de información y además, convertir a México en el líder latinoamericano de desarrollo de software y contenidos digitales en español.” (ProSoft, 2009).

Para lograrlo se plantearon 7 estrategias básicas:

1. Promover las exportaciones y la atracción de inversiones.
2. Educación y formación de personal competente en el desarrollo de software, en cantidad y calidad convenientes.

3. Contar con un marco legal promotor de la industria.
4. Desarrollar el mercado interno.
5. Fortalecer a la industria local.
6. Alcanzar niveles internacionales en capacidad de procesos.
7. Promover acciones conjuntas con los gobiernos estatales y construir infraestructura.

Para alcanzar niveles internacionales en capacidad de procesos, se propone la definición de modelos de procesos y de evaluación apropiados para la industria del software mexicana. Para lograrlo se define un grupo de trabajo integrado por representantes de la industria de desarrollo de software y de las instituciones de gobierno, apoyados por consultores en la materia y por representantes de la academia, encabezados por la Dra. Hanna Oktaba, con el objetivo de crear un modelo de procesos de software, tomando en cuenta la realidad de las organizaciones mexicanas, incorporando las mejores prácticas de la industria y considerando otros modelos de procesos tales como SW CMM, PSP, TSP, ISO/IEC TR 15504 e ISO 9000.

De esta manera, en diciembre del 2002, surge la primera versión del Modelo de Procesos para la Industria de Software (MoProSoft). En mayo del 2003 se genera la versión v1.1 y para agosto del 2005 se libera la versión v1.3, vigente actualmente, y base de la Norma mexicana NMX-I-059/02-NYCE-2005, publicada en el Diario Oficial de la Federación el 15 de agosto del 2005 (Ventura, 2006).

## **2.2 PROPÓSITO DE MOPROSOFT.**

Este modelo, a diferencia de otros como CMMI (*Capability Maturity Model® Integration*), está dirigido a las pequeñas y medianas empresas y/o áreas internas de desarrollo y/o mantenimiento de sistemas. Incorpora las mejores prácticas en gestión e ingeniería de software. Surge bajo las premisas: fácil de entender, fácil de aplicar,

adopción no costosa y que represente una base para la adopción exitosa de otros modelos tales como CMMI o ISO 9000:2000.

MoProSoft está enfocado en procesos y pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Para lograr esos objetivos básicamente responde a las preguntas ¿qué hacer?, ¿quién lo hace? y ¿con qué recursos? (Ventura, 2006).

## **2.3 DEFINICIONES BÁSICAS.**

A continuación se presentan algunas definiciones básicas (Oktaba, 2005) para la comprensión del modelo y de este trabajo de tesis.

### **Proceso**

Conjunto de prácticas relacionadas entre sí, llevadas a cabo a través de roles y por elementos automatizados, que utilizando recursos y a partir de insumos producen un satisfactor de negocio para el cliente.

### **Categoría de procesos**

Un conjunto de procesos que abordan la misma área general de actividad dentro de una organización.

### **Propósito**

Objetivos generales medibles y resultados esperados de la implantación efectiva del proceso.

### **Descripción**

Descripción general de las actividades y productos que componen el flujo de trabajo del proceso.

**Objetivos**

Fin a que se dirige o encamina una acción u operación.

**Indicadores**

Mecanismo que sirve para mostrar o significar una cosa con evidencias y hechos.

**Metas cuantitativas**

Valor numérico o rango de satisfacción por indicador.

**Responsabilidad y autoridad**

Responsabilidad es el rol principal responsable por la ejecución del proceso. Autoridad es el rol responsable por validar la ejecución del proceso y el cumplimiento de su propósito.

**Subprocesos**

Lista de procesos de los cuales se compone el proceso en cuestión.

**Producto entrada, productos de salida y productos internos**

Cualquier elemento que se genera en un proceso.

**Práctica**

Un conjunto de elementos, tales como actividades, roles, infraestructura y mediciones, que al llevarse a cabo describen la ejecución de un proceso.

**Rol**

Es responsable por un conjunto de actividades de uno o más procesos. Un rol puede ser asumido por una o más personas de tiempo parcial o completo.

**Actividad**

Conjunto de tareas específicas asignadas para su realización a uno o más roles.

**Verificación**

Actividad para confirmar que el producto refleja propiamente los requerimientos especificados para él.

**Validación**

Actividad para confirmar que el producto resultante es capaz de satisfacer los requerimientos para su aplicación especificada o uso previsto.

**Flujo de trabajo**

Esquema que expresa las relaciones entre las actividades de un proceso. Una relación puede ser secuencial, paralela, cíclica, de selección o anidada.

Los siguientes elementos: incorporación a la base de conocimiento, recursos de infraestructura, mediciones, capacitación, referencias bibliográficas, situaciones excepcionales y lecciones aprendidas no se definirán, ni se usarán en el mapeo debido a que son informativos (Oktaba, 2005).

**2.4 PATRÓN DE PROCESOS.**

Un patrón es una descripción de una solución genérica a un problema común, a partir del cual se puede resolver un problema específico. Un patrón de procesos describe un propuesta probada exitosamente y/o una serie de acciones para el desarrollo de software. En el caso de MoProSoft, se utiliza un patrón de procesos para definir

un esquema de elementos para la documentación de los procesos.

El patrón utilizado tiene la siguiente estructura:

- Definición general del proceso. En esta sección se indica su nombre, categoría, propósito, descripción general de sus actividades, objetivos, indicadores, metas cuantitativas, responsabilidad y autoridad, subprocesos, procesos relacionados, entradas, salidas, productos internos y referencias bibliográficas.
- Prácticas. Aquí se indican los roles involucrados en el proceso y la capacitación requerida, se describen las actividades en detalle asociándolas a los objetivos del proceso, se presenta un diagrama de flujo de trabajo, se describen las verificaciones y validaciones, se listan los productos que se incorporan a la base de conocimiento, se identifican los recursos de infraestructura necesarios para apoyar a las actividades, se establecen las mediciones del proceso, así como las prácticas para la capacitación, manejo de situaciones excepcionales y uso de lecciones aprendidas.
- Guías de ajuste. Aquí se sugieren las modificaciones al proceso que no deben afectar a los objetivos del mismo.

## **2.5 DESCRIPCIÓN DEL MODELO.**

MoProSoft se clasifica en tres categorías, agrupando los procesos que abordan la misma área general de actividad dentro de una organización. Estas son Alta Dirección, Gerencia y Operación.

La categoría de Alta Dirección contiene al proceso de Gestión de Negocio. En esta se definen los lineamientos que deberán seguir los procesos de la categoría de Gerencia.

A su vez, Gestión de Negocio se retroalimenta con la información

generada por la Gerencia. La categoría de Gerencia contiene a los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos. Adentro de los cuales se definen los procesos, proyectos y recursos requeridos para cumplir con los objetivos y estrategias especificados en el Plan Estratégico de la organización, generado durante la Gestión de Negocio.

Dentro de la Gestión de Recursos encontramos los subprocesos Recursos Humanos y Ambiente de Trabajo, Bienes Servicios e Infraestructura y Conocimiento de la Organización.

Dentro de la categoría de Operación se encuentran el proceso de Administración de Proyectos Específicos y el proceso de Desarrollo y Mantenimiento de Software. Con estos se pretende realizar las actividades definidas y con los recursos proporcionados por los procesos pertenecientes a la categoría de Gerencia.

A continuación se presenta la fig. 2 categorías de los procesos:

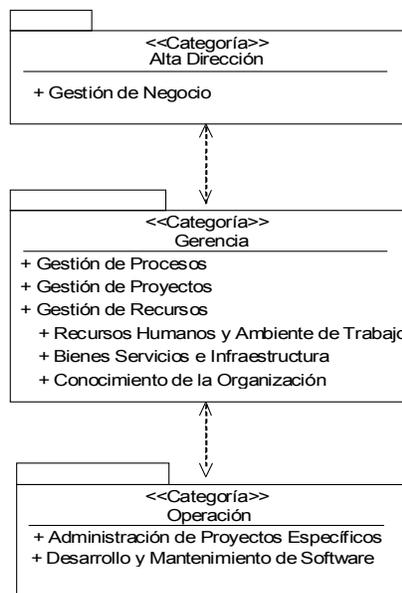


Figura 2.- Categorías de procesos

Como se puede observar, MoProSoft se compone de 9 procesos, a continuación se detalla el propósito de los mismos:

<b>Proceso</b>	<b>Descripción</b>
DIR. 1, Gestión de Negocio	<p>Establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua.</p> <p>Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.</p>
GES. 1, Gestión de Procesos	<p>Establecer los procesos de la organización, en función de los procesos requeridos identificados en el plan estratégico. Así como definir, planificar, e implantar las actividades de mejora en los mismos.</p>
GES. 2, Gestión de Proyectos	<p>Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.</p>
GES. 3, Gestión de Recursos	<p>Conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la base de conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del plan estratégico de la organización.</p>
GES. 3.1, Recursos	<p>Proporcionar los recursos humanos</p>

Humanos y Ambiente de Trabajo	adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.
GES. 3.2, Bienes, Servicios e Infraestructura.	Proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.
GES 3.3 Conocimiento de la Organización.	Mantener disponible y administrar la base de conocimiento que contiene la información y los productos generados por la organización.
OPE. 1, Administración de Proyectos Específicos.	Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
OPE. 2, Desarrollo y Mantenimiento de Software.	Realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados.

Para implementar este modelo, la organización, en caso de que no tenga procesos establecidos o documentados, deberá generar una instancia de cada uno de los procesos considerando los lineamientos establecidos en el propio documento de MoProSoft (Oktaba, 2005).

En caso de que la organización ya cuente con procesos establecidos o documentados, se deberá realizar un mapeo de procesos, en el cuál se identifique las coincidencias y discrepancias entre los procesos propios de la empresa y los definidos por MoProSoft, y deberá ajustarlos al modelo.

### 3 SOFTWARE & SYSTEMS PROCESS ENGINEERING METAMODEL (SPEM) 2.0.

#### 3.1 CARACTERISTICAS GENERALES.

SPEM 2.0 (SPEM, 2007) es un metamodelo para ingeniería de procesos y un marco de trabajo conceptual que provee los conceptos necesarios para modelar, documentar, presentar, publicar, gestionar, intercambiar y realizar métodos y procesos de software.

Este es un estándar del *Object Management Group* (OMG) (OMG, 2009), el cual se encuentra dentro de la ingeniería de procesos, que es un área de la ingeniería de software dedicada a la “definición, implementación, medición y mejora de los procesos de ingeniería de software”.

SPEM 2.0 sirve para definir procesos de desarrollo de software, sistemas y sus componentes. Su alcance se limita a los elementos mínimos necesarios para definir dichos procesos sin añadir características específicas de un dominio o disciplina particular. Sirve para métodos y procesos de diferentes estilos, culturas, niveles de formalismo, o modelos de ciclos de vida. No es un lenguaje de modelado de procesos en general, ya que está orientado a los procesos de software. No provee conceptos propios para modelado del comportamiento, pero incluye mecanismos para encajar el método externo elegido. Por ejemplo diagrama de actividad de UML (Lenguaje de Modelado Unificado 2 (UML, 2009)).

La idea central de SPEM 2.0 para representar procesos está basada en tres elementos básicos: **rol**, **producto de trabajo** y **tarea**. Esta idea es que un modelo de proceso consiste, básicamente, en decir quién (**rol**) realiza qué (**tarea**) para, a partir de unas entradas (**productos de trabajo**), obtener unas salidas (**productos de trabajo**)

(fig. 3).

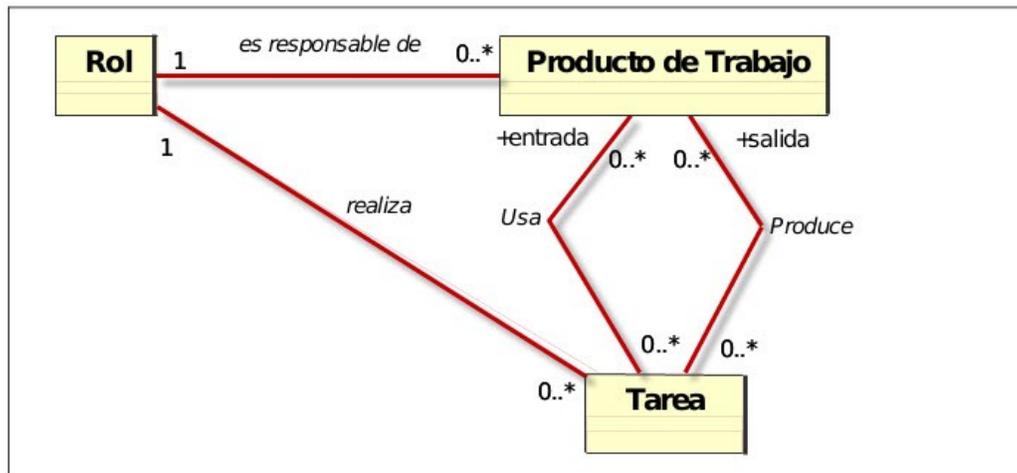


Figura 3. Relaciones entre elementos básicos SPEM 2.0

#### Al trabajar con SPEM 2.0 existen 4 escenarios fundamentales:

- Crear un repositorio de “Contenidos de método” reutilizables, es decir, una colección organizada de roles, tareas, productos de trabajo, guías, fragmentos de método y procesos, etc. Esto en sí solo ya es un valor para cualquier organización software, ya que supone disponer de un repositorio de conocimiento sobre procesos en un formato estandarizado. Además, el repositorio basado en SPEM 2.0 es una base de conocimiento ideal para la formación en procesos.
- Dar soporte al desarrollo, gestión y crecimiento de procesos software. Esto implica combinar, reutilizar y extender los elementos de método anteriores para configurar los procesos que sirven para guiar los proyectos. Desde fragmentos de proceso elementales se puede llegar a generar todo un proceso completo o toda una metodología, incluyendo varios procesos. SPEM 2.0 ayuda a los equipos de desarrollo a definir o seleccionar un proceso, incluyendo opciones para

que los mismos métodos puedan ser aplicados de forma diferente en momentos distintos o en proyectos distintos, para comprender claramente como se relacionan unas tareas con otras, o para que los procesos puedan ser vistos como flujos de trabajo o como estructuras de desglose de trabajo (WBS), según interese en cada momento. Para lo anterior, SPEM 2.0 soporta la creación sistemática de procesos basada en la reutilización de contenidos de método. También provee el fundamento conceptual para que los ingenieros de procesos y gestores de proyectos seleccionen, adapten y rápidamente ensamblen procesos para sus proyectos concretos. El ensamblado rápido de procesos es posible gracias a la implementación de catálogos de procesos predefinidos, “trozos“ de proceso o patrones de proceso.

- Establecer un marco de trabajo general de la organización a partir de los procesos y los elementos definidos anteriormente. Para ello, SPEM 2.0 permite dar soporte al despliegue del contenido de método y proceso que justo se necesita en cada caso, teniendo en cuenta que ningún proyecto es exactamente como el anterior y nunca exactamente el mismo proceso software se ejecuta dos veces. SPEM 2.0 incorpora conceptos para:
  - Reutilización de procesos o patrones de procesos.
  - Variabilidad, procesos que incluyen partes alternativas configurables.
  - Particularización, los usuarios definen sus propias extensiones, omisiones y puntos de variabilidad sobre procesos estándares reutilizados.
- Generar plantillas para planes de proyecto concretos. Esto supone que a partir de ahora los jefes de proyecto pueden

contar con mucha más información y disponible de manera automática a la hora de definir los planes de los proyectos. Para darles auténtico valor, las definiciones de los procesos deben ser desplegadas en formatos que permitan su realización automática (sistemas de gestión de proyectos y recursos, motores de flujos de trabajo, etc). Para ello, SPEM 2.0 incluye estructuras de definición de procesos que permiten expresar cómo un proceso será realizado de forma automática con estos sistemas. Ejemplos de ello son las iteraciones (una o varias definiciones de trabajo serán repetidas varias veces en un proyecto) y las ocurrencias múltiples (varias instancias de una definición de trabajo pueden llevarse a cabo a la vez de forma paralela).

### **3.2 COMO SE UTILIZA EL METAMODELO SPEM 2.0**

Para trabajar con SPEM 2.0 debemos de distinguir dos grupos al momento de definir una metodología.

El primero de los grupos es el *Method Content* (contenido de método); hay que poblar éste con *Content Elements* (elementos de contenido), es decir, los elementos primarios o constructores básicos (fig. 4).

El segundo grupo es *Procesos*, para generarlos hay que combinar y reutilizar los elementos primarios estructurándolos respetando la relación que se muestra en la fig. 3, agrupar las *tareas* en *actividades* y éstas a su vez en *procesos* (fig 4).

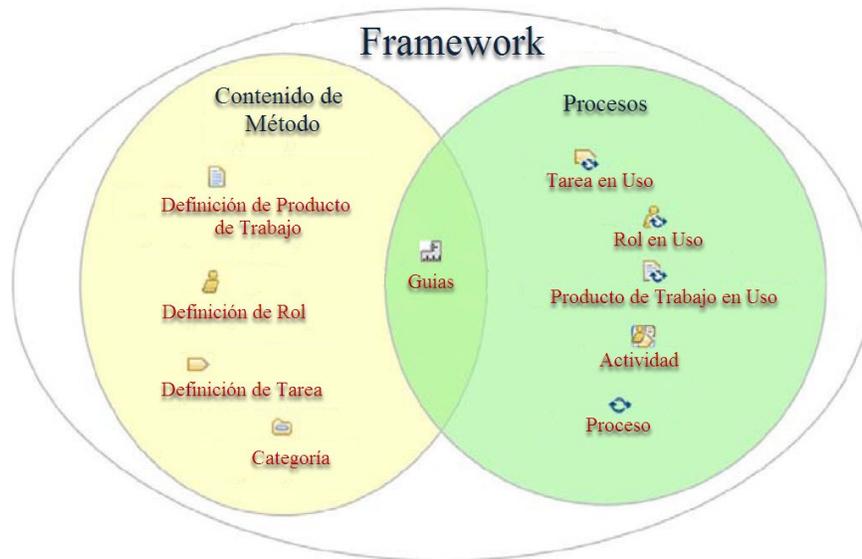


Figura 4.- Elementos básicos y procesos

Al relacionar y estructurar estos dos grupos, podemos generar cualquier metodología (fig 5).

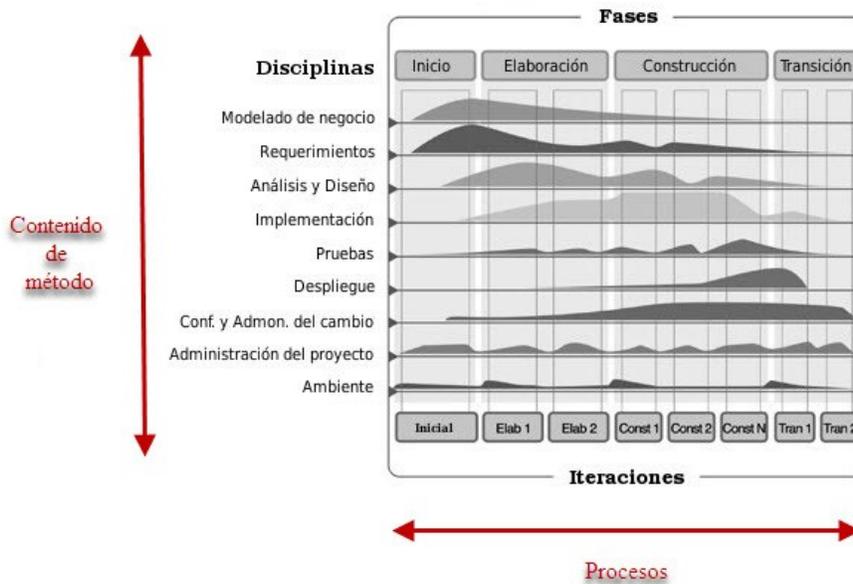


Figura 5.- Representación de un modelo usando elementos básicos y procesos

Además de la separación clara entre la definición de contenidos de método y su aplicación en procesos, ya comentada, otras características avanzadas del metamodelo SPEM 2.0 son:

- Mantenimiento consistente de muchos procesos alternativos. Para ello, SPEM 2.0 incluye: i) un conjunto extendido de interrelaciones de reutilización y variabilidad con semántica de herencia y orientación a aspectos; ii) conceptos de patrones de proceso, y iii) *plugins* de métodos. Estas opciones permiten tener diferentes variantes de procesos específicos, basados en los mismos contenidos de método y estructuras de procesos, pero aplicados con diferente detalle y escala.
- Muchos ciclos de vida diferentes. SPEM 2.0 permite trabajar con distintos tipos de ciclos de vida del software: Cascada, Iterativo, Incremental, Evolutivo, etc. Para ello incluye un conjunto de atributos que permiten especificar aspectos temporales para los elementos de proceso que luego pueden ser asociados a los planes de proyectos. Un ejemplo de atributo para clases de ciclos de vida es la propiedad Iteración, que permite representar que la ejecución de una o varias descripciones de trabajo se puedan repetir más de una vez.
- Variabilidad y extensibilidad. Para lo cual SPEM 2.0 incluye un mecanismo de *plugins* de dos tipos: a) *Method plugins* para particularizar y adaptar contenidos de método sin modificar el original; y b) *Process plugins* para procesos, pudiendo añadir o sustituir elementos de trabajo en el WBS (estructura de desglose de trabajo del proceso) sin afectar al original.
- *Patrones de proceso*. Son bloques o trozos de *proceso* reutilizables para crear nuevos *procesos*. La selección y aplicación de un patrón de proceso puede ser hecha de dos formas: a) puede ser copiado y modificado, permitiendo

individualizar el contenido del patrón según las necesidades de cada momento; o b) puede ser aplicado por medio del mecanismo de *Actividad en Uso*, que es una forma avanzada de reutilizar estructuras de proceso. Una *Actividad en Uso* define tipos de interrelaciones para que cuando el patrón esté siendo revisado o modificado, todos los cambios se reflejen automáticamente en todos los procesos en que se aplica el patrón.

- Componentes de proceso. Son piezas de proceso sustituibles y reutilizables basadas en los principios de encapsulación y caja negra. No se especifica la descripción de trabajo interna del componente, sino que sólo se especifican los productos de trabajo de entrada y salida que habrá mediante los llamados puertos de productos de trabajo. Esta opción de SPEM 2.0 permite manejar las situaciones en que un proyecto requiere que partes del proceso no sean decididas hasta la ejecución o nunca, como por ejemplo *outsourcing*.

### **3.2.1 ORGANIZACIÓN DE SPEM 2.0.**

SPEM 2.0, está organizado en 7 paquetes (fig. 6). Cada paquete es una unidad lógica que extiende los paquetes de los que depende, proveyendo estructuras y capacidades adicionales.

Los paquetes definidos en un nivel inferior pueden ser realizados en una implementación parcial de SPEM 2.0 sin necesidad de los paquetes de los niveles superiores. Como regla general, cada clase del metamodelo (constructor) se incluye en el paquete del nivel más inferior posible. En algunos casos, las clases se extienden (especializan) en paquetes de nivel superior vía el mecanismo “merge” (idéntico al de UML 2) para incluir más propiedades e interrelaciones.

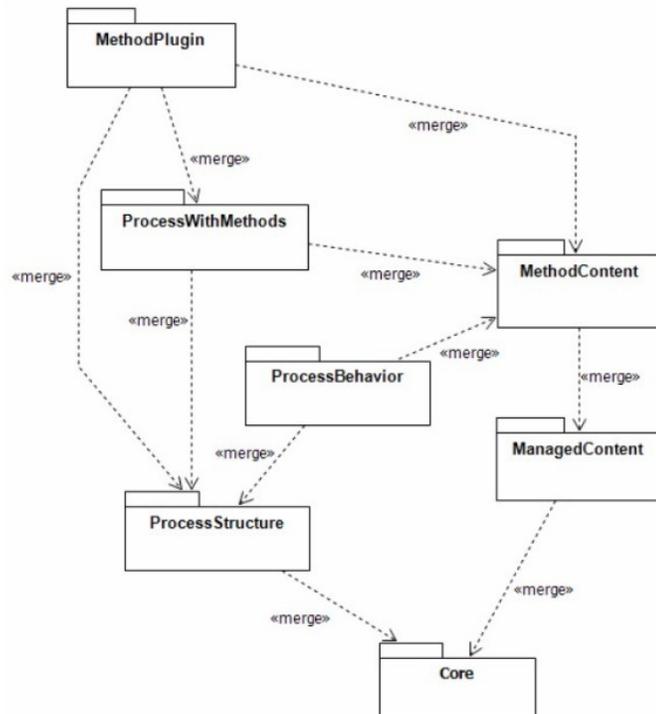


Figura 6.- Paquetes de SPEM 2.0

A continuación se describen los contenidos de cada paquete.

### **Core**

Contiene las clases y abstracciones que sirven de base para las clases de los demás paquetes. Provee dos capacidades básicas: a) crear evaluaciones definidas por el usuario (*kinds*) que permiten establecer tipos diferentes entre las instancias de una clase; y b) una colección de clases abstractas para definir el trabajo.

### **Process Structure**

Define la base para la creación de modelos de proceso (MP) flexibles y sencillos, es decir, define la estructura de desglose de trabajo estática mediante anidamiento de actividades y dependencias de precedencia entre ellas. Dicha estructura también incluye referencias a la lista de Roles que realizan cada actividad y a los Productos de

Trabajo que son entradas y/o salidas. También provee capacidades para reutilización mediante ensamblado de procesos usando conjuntos de actividades enlazadas de forma dinámica.

### ***Process Behavior***

Permite extender las estructuras del paquete anterior con modelos de comportamiento externos: diagramas de actividad de UML 2 (comportamiento de proceso), máquina de estados (ciclo de vida de un producto de trabajo), etc. En vez de incluir un mecanismo propio para representar el comportamiento, se opta por reutilizar los ya existentes de UML.

### ***Managed Content***

Permite incorporar y gestionar descripciones en lenguaje natural, documentos y otras informaciones útiles para la comprensión por humanos. Esta posibilidad es muy interesante y útil porque ciertos valores y culturas no pueden ser formalizados con modelos, sólo pueden ser capturados con documentación en lenguaje natural. Hay libertad total para combinar modelos estructurales de proceso con contenidos en lenguaje natural. Así, un proceso puede estar formado sólo por una colección de guías definiendo buenas prácticas (esto es especialmente apto en el caso de métodos ágiles poco estructurados); sólo una estructura de actividades sin ningún tipo de documento de texto; o una combinación interrelacionada de ambas cosas.

### ***Method Content***

Incluye los conceptos para crear elementos de método, que son los *items* elementales que sirven de base para el ensamblado de procesos, metodologías, ciclos de vida, etc. Los principales elementos de método se derivan del patrón básico de SPEM 2.0: alguien (rol) hace

algo (tarea) para obtener algo (producto de trabajo) basándose o ayudándose en algo (guía). Por tanto, los elementos de método permiten describir, con el detalle necesario, cómo se alcanzan los objetivos del proceso haciendo qué tareas por qué roles usando qué recursos y obteniendo qué resultados. Los procesos reutilizan y relacionan entre sí los elementos de método de distintas maneras para diferentes tipos de proyectos.

### ***Process with Methods***

Incorpora nuevas estructuras para poder integrar los procesos definidos con el paquete “*Process Structure*” con los elementos de método (instancias del paquete *Method Content*). Al asociar elementos de método a partes específicas de procesos, se crean nuevas clases (tarea en uso, rol en uso, etc.) que heredan de los elementos de método pero pueden tener cambios individualizados.

### ***Method Plugin***

Incorpora conceptos (*Method Plugin*, *Process Component*, *Variability*, etc.) para diseñar y gestionar bibliotecas o repositorios de contenidos de método y de procesos, que sean mantenibles a gran escala, reutilizables y configurables. Con estos conceptos los ingenieros de procesos pueden definir una o varias Configuraciones de Método de cada proceso. Esto permite tener vistas diferentes de un mismo proceso adaptadas a distintas audiencias o tipos de usuarios.

## **3.3 ORGANIZACIÓN DE UN REPOSITORIO EN SPEM 2.0.**

Un repositorio o biblioteca de métodos y procesos en SPEM 2.0 es una colección de uno o más plugins y una o varias configuraciones. Cada *plugin* se almacena en un directorio diferente e incluye contenido

de método y procesos, (fig. 4). El apartado contenido de método está formado por paquetes de contenido, categorías estándar y categorías personalizadas; y el apartado de Procesos contiene patrones de proceso y procesos de despliegue. Un ejemplo de toda esta estructura se muestra en la figura 7.

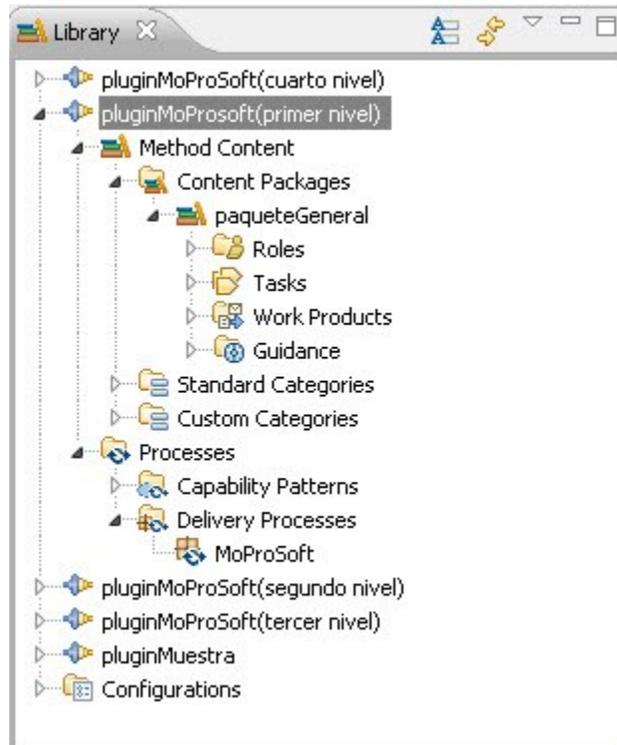


Figura 7.- Estructura de SPEM 2.0

### **Contenido de método**

El contenido de método puede ser organizado a voluntad del usuario mediante una jerarquía de paquetes de contenido (*Content Package*), cada uno de los cuales puede incluir roles, tareas, productos de trabajo y guías, estos son llamados elementos de contenido. A continuación se hará una descripción de cada uno de ellos.

### ***Tareas***

Una Tarea describe una unidad de trabajo que se asigna y se administra, es decir, es la unidad atómica de trabajo para definir procesos. Su granularidad es de unas pocas horas a unos pocos días, afectando a unos pocos productos de trabajo y vinculando a unos pocos roles. Es un Elemento de Método que define el trabajo realizado por roles.

### ***Roles***

Un Rol define un conjunto de habilidades, competencias y responsabilidades relacionadas, de un individuo o de un grupo. No se deben confundir roles con personas, ya que la vinculación entre personas y roles se realiza durante la planificación del proyecto y puede ocurrir que un individuo desempeñe varios roles o que un rol sea desempeñado por varios individuos. Un rol es un Elemento de Método usado en las Tareas para señalar quienes las realizan.

### ***Productos de trabajo***

Un Producto de Trabajo es consumido, producido o modificado por las Tareas. Un Producto de Trabajo puede estar asociado con otros Productos de Trabajo mediante asociaciones de los siguientes tipos:

- Composición, cuando las instancias de un Producto de Trabajo sirven para componer instancias de otro Producto de Trabajo. Por ejemplo los “Actores” se emplean para componer “Casos de uso”.
- Agregación, si un Producto de Trabajo está formado por agregación de otros. Por ejemplo el “Manual de usuario” incluye el “Manual de instalación”.
- Es impactado por, cuando un Producto de Trabajo impacta en otro, es decir, si los cambios del primero obligan a cambiar el segundo. Por ejemplo si cambia el “Modelo de casos de uso”, es

necesario adaptar a dicho cambio la “Realización de casos de uso”.

Existen tres tipos predefinidos (especializaciones) de Productos de Trabajo:

- Artefacto: De naturaleza tangible (modelo, documento, código, archivos, etc). Un artefacto puede estar formado por otros artefactos más simples.
- Entregable: Provee una descripción y definición para empaquetar otros productos de trabajo con fines de entrega a un cliente interno o externo. Representa una salida de un proceso que tiene valor para un usuario, cliente u otro participante .
- Resultado (*Outcome*): De naturaleza intangible (estado), o que no está formalmente definido.

### **Guías**

Una guía o instrucción es un elemento de método (o de proceso) que provee información adicional relacionada con otros elementos. Por ejemplo: ayuda o información sobre cómo trabaja un rol, cómo crear un producto de trabajo, cómo usar una herramienta o cómo realizar una tarea.

SPEM 2.0 tiene predefinidos los siguientes tipos de guías:

- Activo Reutilizable: Provee una solución a un problema para un contexto dado. Incluye reglas o instrucciones sobre cómo utilizarlo.
- Concepto: Resumen de ideas clave asociadas con principios básicos subyacentes. Refieren a tópicos más generales que las directrices y abarcan varios productos de trabajo y/o actividades. Por ejemplo, “iterativo e incremental”.
- Definición de Término: Definición de un término, concepto o

idea relevante. Sirve para generar una especie de glosario automático. Se relacionan con Elementos de Contenido mediante su aparición en las descripciones textuales.

- Directriz: Provee detalle adicional sobre cómo realizar una tarea o grupo de tareas, o provee información adicional, reglas y recomendaciones sobre productos de trabajo. Además, puede incluir detalles sobre: las mejores prácticas y aproximaciones diferentes para hacer un trabajo; cómo usar cierto tipo de producto de trabajo; los subtipos y variantes de un artefacto y su evolución a lo largo del tiempo; habilidades que los roles deben adquirir o mejorar; medidas del progreso o madurez, etc.
- Documentación: Versión especial de Concepto que ha sido revisada o publicada externamente y que puede ser leída y comprendida de forma aislada. Por ejemplo, una instancia típica, parcialmente completada, de uno o más productos de trabajo o descripción del escenario en que una tarea debe ser realizada.
- Guía de Herramienta: Explica el uso de una cierta herramienta en el contexto de cierto trabajo, o de forma independiente.
- Guía para la Estimación: Indicaciones para estimar el esfuerzo asociado con cierto trabajo, incluyendo consideraciones sobre cómo hacer la estimación y las métricas a utilizar.
- Hoja de Ruta: Describe en forma de camino lineal, cómo suele llevarse a cabo una actividad o proceso complejos. Provee información sobre cómo las actividades y tareas se relacionan entre sí a lo largo del tiempo. Sólo pueden estar asociados a Actividades y Procesos.
- Informe: Plantilla predefinida de un resultado que se obtiene de forma automática mediante alguna herramienta.

- Lista de Comprobación: Identifica una serie de elementos que deben ser completados o verificados.
- Material de Soporte: Comodín para utilizar cuando se está en un caso que no encaja en ninguno de los demás tipos de guías.
- Plantilla: Establece la tabla de contenidos, secciones, cabeceras y formato estandarizado predefinido de un artefacto (documentos, modelos, etc.). Puede incluir descripciones sobre cómo usar y completar cada parte.
- Práctica: Manera o estrategia predefinida de hacer un trabajo que tiene un impacto positivo sobre la calidad de un producto de trabajo o de un proceso. Resume aspectos que pueden impactar en diferentes partes de un método o proceso. Ejemplos: “gestionar riesgos”, “verificar continuamente la calidad”, “desarrollar centrado en la arquitectura”, “desarrollar basado en componentes”.

### **Categorías**

Una Categoría es un elemento de contenido, o de proceso, usado para clasificar o agrupar dichos elementos con base en los criterios que desee el ingeniero de procesos. SPEM 2.0 distingue dos clases de categorías:

- Estándar: Vienen predefinidas en SPEM 2.0 y son las siguientes:
  - Conjunto de roles, sirven para agrupar roles que tienen algo en común.
  - Disciplina, permiten clasificar o agrupar el trabajo (tareas).
  - Dominio, permiten establecer una jerarquía de dominios, para clasificar productos de trabajo, con tantos niveles como se desee.

Herramienta, sirve para clasificar o agrupar guías de herramientas.

- Personalizada: Sirven para que el ingeniero de procesos pueda definir otras categorías nuevas.

### ***Hito***

Representa uno o varios eventos significativos para el desarrollo de un proyecto, como por ejemplo:

- Una decisión importante.
- La conclusión de un entregable.
- La conclusión de una fase.

### ***Actividad***

En SPEM 2.0 la descomposición del trabajo en distintos niveles de detalle se realiza mediante el concepto de Actividad, de forma que cualquier elemento de desglose de trabajo con estructura interna (es decir, que incluye elementos de desglose) recibe el nombre de Actividad, independiente del nivel de desglose, es decir, del tamaño del fragmento de trabajo representado.

Dado que SPEM 2.0 se ayuda de la generalización proporcionada de UML 2 por medio de los estereotipos y por medio de ésta una Actividad puede usarse de las siguientes formas:

- Representar una unidad de trabajo general en un proceso.
- Puede tener estructura interna formada por agregación de elementos de desglose, que pueden ser de varios tipos, no solo de trabajo: actividades más simples (anidamiento), elementos de método en uso (roles en uso, productos de trabajo en uso), e hitos.

Por la primera de las dos formas anteriores en SPEM 2.0 tiene predefinidos varios tipos especiales de Actividades: Iteración, Fase y

Proceso. Puesto que Actividad es un concepto genérico, que incluye a proceso como especialización, en SPEM 2.0 no se respeta la nomenclatura tradicional de los estándares ISO de Ingeniería del software.

### ***Fases e Iteraciones***

Como ya se ha indicado, las fases e iteraciones son dos tipos especiales de actividades (adicionales a los procesos).

Una Fase representa un periodo de tiempo que es significativo para un proyecto, y que acaba con un punto de control de administración importante, un hito o un conjunto de entregables concluidos. En la práctica, en SPEM 2.0 una fase es una actividad que tiene el valor falso en la propiedad “Es repetible”.

Por el contrario, una Iteración (*Iteration*) representa un conjunto de actividades anidadas que se repiten más de una vez. Sirve para organizar ciclos repetitivos de trabajo. En la práctica, en SPEM 2.0 una iteración es una actividad que tiene el valor verdadero en la propiedad “Es repetible”.

### ***Procesos***

Tal como se mostró con la fig. 4 con SPEM 2.0 se distinguen dos etapas a la hora de implementar un proceso o metodología: primero se puebla el Contenido de Método, cuyos conceptos han sido presentados anteriormente, y en segundo lugar, se combinan y reutilizan dichos elementos para ensamblar actividades y procesos.

En SPEM 2.0, un Proceso es un tipo de Actividad que describe una estructura para tipos particulares de proyectos o partes de ellos.

Mediante un proceso de SPEM 2.0 se pueden representar distintos tipos de métodos de ingeniería del software: un proceso, un modelo de procesos completo, un ciclo de vida con sus diversos procesos, o una

metodología completa.

Los procesos software definen cómo serán ejecutados los proyectos software.

### **Tipos de procesos**

En SPEM 2.0 existen dos clases principales de procesos:

- *Patrón de Proceso*: Es un “fragmento de proceso” que describe un grupo de actividades reutilizable como solución a algún tipo de problema o situación habitual. Se definen para poder ser empleados más de una vez en uno o varios procesos o con fines de organización. Se pueden almacenar en una jerarquía de Paquetes de Proceso (*Process Package*).
- *Proceso de Despliegue (Delivery Process)*: Describe una aproximación completa e integrada para realizar un tipo específico de proyecto, abarcando un ciclo de vida completo de desarrollo o mantenimiento. Sirven como plantillas para planificar y ejecutar los proyectos. En un Proceso de Despliegue se ensamblan Patrones de Proceso y Elementos en Uso (tareas, roles y productos de trabajo en uso). Ejemplos: “proceso unificado de Rational (RUP)”, “programación extrema (XP)” .

### **Reutilización y variabilidad**

La organización en *plugins* permite que se puedan reutilizar los elementos de contenido y los procesos definidos en una Biblioteca (*Library*). Dicha reutilización se puede realizar de dos maneras:

a) Al crear un *plugin* nuevo se puede referenciar a otros *plugins*.

b) Usar de forma directa el contenido de un *plugin* desde otro *plugin* diferente.

SPEM 2.0 contempla 5 tipos de variabilidad entre elementos de

contenido:

- No Aplicable.
- Contribuye.
- Amplía
- Reemplaza.
- Amplía y Reemplaza.

### **Configuraciones de Método**

Una Configuración de Método es una selección de contenidos de los *plugins* de una Biblioteca de Métodos de forma que se limita el espacio visual de la biblioteca al subconjunto seleccionado.

Una Configuración de Método define un subconjunto lógico de una biblioteca de métodos mediante la selección (filtro) de los paquetes deseados, tanto de contenido de método como de procesos.

Mediante el uso de configuraciones se seleccionan los contenidos de cada uno de esos tipos de proyectos específicos.

#### **3.3.1 JERARQUIA DE DESGLOSE DE TRABAJO.**

Un repositorio o biblioteca de métodos y procesos en SPEM 2.0 es una colección de uno o más *plugins* y una o varias configuraciones. Cada *plugin* se almacena en un directorio de disco diferente e incluye contenido de método y procesos. El contenido de método puede ser organizado a voluntad del usuario mediante una jerarquía de paquetes de contenido (*Content Package*), cada uno de los cuales puede incluir roles, tareas, productos de trabajo y guías, estos son llamados elementos de contenido.

En SPEM 2.0 la *jerarquía de desglose del trabajo*, es decir, los conceptos existentes para representar el esfuerzo a realizar a distintos niveles de detalle, son los siguientes (del más general al

más particular) (fig. 8)

- *Proceso de despliegue.*
- *Patrón de proceso*
- *Actividad*
- *Tarea*
- *Producto de trabajo*

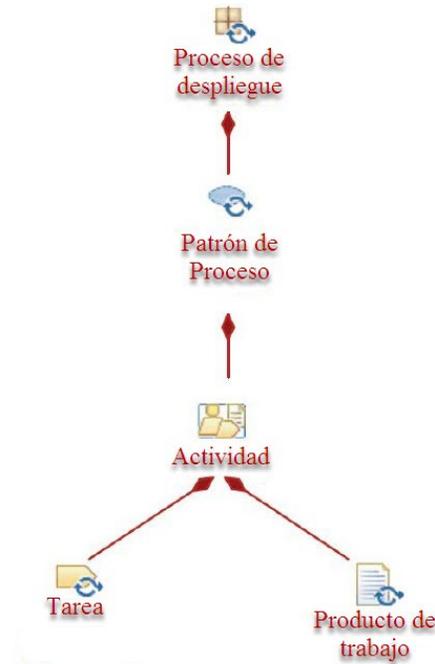


Figura 8. Conceptos para representar la jerarquía de desglose de trabajo

### 3.4 HERRAMIENTAS DISPONIBLES PARA SPEM 2.0

Hay diferentes herramientas que sirven de apoyo para representar y editar los elementos de SPEM 2.0, a continuación se muestra una tabla comparativa.

Herramienta	¿Cumple con el estándar?	Gratuita	Exportar A WEB Y XML	Documen- tación
<i>Eclipse Process Framework</i>	Si	Si	Si	Si

<i>Composer (EPFC)</i>				
<i>IBM Rational Method Composer</i>	Si	No	Si	Si
<i>Osellus IRIS</i>	Si	No	Si	Si

Todas las herramientas cumplen con el estándar SPEM 2.0, dado que han sido desarrolladas para este estándar, de las tres herramientas se escogió EPFC principalmente por que tiene lo mínimo necesario para representar un proceso en SPEM, ciertamente las demás herramientas al igual que EPFC cumplen con esta característica, pero además le han agregado muchos elementos. Por ejemplo, exportar a imagen cualquier parte del proceso, seleccionar que partes de los procesos se quieren exportar a web y a xml, han modelado sus propias metodologías y te proporcionan el plugin generado para usarlo, etc.

Todas estas características incrementan o le dan un costo a la herramienta, pero por la característica principal ya antes mencionada por la que se escoge EPFC, hay que agregarle que además esta herramienta es gratuita, de esta manera se siguen respetando las premisas de MoProSoft, fácil de entender, fácil de aplicar y adopción no costosa.

### **3.5 DESCRIPCIÓN DE LA HERRAMIENTA ECLIPSE PROCESS FRAMEWORK COMPOSER (EPFC).**

*Eclipse Process Framework Composer*, en adelante EPFC, es una herramienta gratuita, desarrollada dentro del entorno ECLIPSE, que sirve para editar fragmentos de método, procesos o metodologías, y generar automáticamente la documentación adecuada en formato para la web. Dichos fragmentos se almacenan en formato XML y, al

estar basados en el estándar SPEM 2.0, pueden ser reutilizados por cada vez más herramientas CASE. En realidad, EPFC utiliza la "*Unified Method Architecture*" (UMA), que a su vez está basada en el metamodelo SPEM 2.0 (Ruiz, 2008).

En resumen, EPFC es un editor de procesos SPEM 2.0, que incluye opciones adicionales para publicar de forma automática sitios web (Kroll, 2009).

Algunos modelos que han sido estandarizados y documentados con la herramienta EPFC son los siguientes:

- OpenUP
- Scrum
- XP
- COMPETISOFT

Para representar el modelado de MoProSoft se usó EPFC, por lo que la mayoría de las figuras de este artículo fueron generadas por esta herramienta.

## 4 MODELADO POR NIVEL DE CAPACIDAD DE MOPROSOFT.

### 4.1 ESTRATEGIA PARA EL MODELADO SPEM-MOPROSOFT.

Para trabajar con SPEM 2.0 se deben distinguir dos grupos al momento de modelar una metodología; Primero, hay que poblar el *Method Content* (contenido de método) con *Content Elements* (elementos de contenido), es decir, los elementos primarios o básicos. Segundo, hay que combinar y reutilizar dichos elementos para generar procesos.

Para poder realizar este trabajo, se empezó con el análisis para determinar qué elemento de MoProSoft se mapeaba a qué concepto de SPEM 2.0. Para poder generar este mapeo se tomaron en cuenta las definiciones de los conceptos de SPEM 2.0 y con ello se identificó qué concepto de SPEM 2.0 tenía el mismo objetivo que el elemento de MoProSoft. Al finalizar este análisis se generó la tabla 1 (ver punto 4.2).

Para poder estructurar los procesos de MoProSoft y hacer que respeten las relaciones entre los conceptos de SPEM 2.0, se tomó como base la jerarquía de desglose de trabajo mostrado en la fig. 8.

Basándose en esta jerarquía se prosiguió con el modelado de MoProSoft, para lo cual se determinaron las siguientes reglas:

- En la numeración de actividades y/o tareas solo se puede tener un máximo de cuatro niveles de desglose.
- En ocasiones se necesitan que las actividades estén más detalladas, por lo tanto, las actividades se tienen que dividir en subactividades antes de relacionarlos con el concepto “tarea” de SPEM 2.0.
- En MoProSoft se define al elemento “Actividad” como “Un conjunto de tareas”, pero, como tal, el elemento “Tarea” no existe explícitamente. En SPEM 2.0, el concepto “Tarea” debe

de existir, ya que solo a él se le puede relacionar con los demás elementos básicos. Tomando en cuenta estas condiciones, a las actividades se les agrego el concepto de SPEM 2.0 “tarea” y con ello se determina que se va a hacer.

- Los elementos primarios de SPEM 2.0: roles y productos de trabajo para agregarlos a una “tarea”, primero se hará el análisis de si el rol que se indica es el encargado de realizar la tarea y si los productos de trabajo de entrada y/o salida son los que se necesitan o se producen de la tarea, respectivamente.

Con ayuda de la herramienta EPFC, el mapeo de la tabla 1 y las reglas mencionadas previamente, se realizó el modelado de MoProSoft (nivel de capacidad uno, Realizado). Un ejemplo de este modelado se presenta en la fig. 9:



Figura 9 Modelado de procesos de MoProSoft a SPEM 2.0 usando EPFC

## 4.2 MAPEO SPEM-MOPROSOFT.

Tabla 1 Mapeo de MoProSoft a SPEM 2.0

<b>MoProSoft – elemento</b>	<b>SPEM 2.0-Concepto</b>	<b>Descripción</b>
Modelo de procesos	Proceso de despliegue.	
Categoría	Jerarquizando los procesos.	Respetando la jerarquía de desglose de trabajo.
Acrónimo	Nombre de Presentación.	Propiedad que tienen los conceptos de SPEM 2.0.
Propósito	Propósito (objetivo).	Propiedad que tienen los conceptos de SPEM 2.0.
Descripción	Descripción principal.	Propiedad que tienen los conceptos de SPEM 2.0.
Objetivos	<i>Milestone.</i>	
Responsabilidad y autoridad	Rol.	
Subprocesos (opcionales)	Patrón de proceso.	
Procesos relacionados	No definido	En este caso SPEM 2.0 no maneja algo parecido explícitamente, pero se puede ver implícitamente con las relaciones de los productos de trabajo.
Producto de entrada	Producto de trabajo.	Indicando que es de entrada.
Producto salida	Producto de trabajo	Indicado que es de salida.
Producto intermedio	Producto de trabajo	No indicando si es de entrada y/o salida.
Rol	Rol	
Actividad	Actividad	
Elementos dentro de la actividad	Tareas	
Diagramas de flujo de trabajo	Diagrama de flujo de trabajo	
Verificaciones y validaciones	Tareas	

#### 4.3 CONVENCIONES PARA EL NOMBRADO DE LOS DIFERENTES ELEMENTOS SPEM-MOPROSOFT.

Para la presente tesis los elementos de MoProSoft estarán en negritas y los elementos de SPEM en itálicas.

La convención a usar para los *patrones de procesos* (**categorías de procesos** de MoProSoft), es la siguiente:

- *Name*: Nombre completo de la **categoría de proceso** seguido de la abreviación en mayúsculas entre paréntesis, por ejemplo: Alta Dirección (DIR).
- *Presentation Name*: Nombre completo de la **categoría de proceso**, por ejemplo Alta Dirección.

La convención a usar para los *patrones de procesos* (**procesos** de MoProSoft), es la siguiente:

- *Name*: Tres letras en mayúsculas representativas del **proceso**, seguido de un punto y número, por ejemplo: DIR.1. El número representa la posición que ocupa con respecto a procesos de la misma **categoría de proceso**.
- *Presentation Name*: Nombre completo del **proceso**, por ejemplo Gestión de Negocio.

La convención a usar para las *actividades*, es la siguiente:

- *Name*: Las tres letras en minúsculas representativas del **proceso**, seguido del número, por ejemplo: dir1. A continuación la letra "A" mayúscula con el número de la *actividad*, será permitido solo 4 niveles en el número. Por ejemplo dir1A1.6.1.
- *Presentation Name*: Las tres letras en minúsculas representativas del **proceso**, seguido del número, por ejemplo: dir1. A continuación la letra "A" mayúscula con el número de la *actividad*, por ejemplo dir1A1.6.1, seguido de una oración la cual deberá iniciar con un verbo en infinitivo.

La convención a usar para las *tareas*, es la siguiente:

- *Name*: Las tres letras en minúsculas representativas del **proceso**, seguido del número, por ejemplo: dir1. A continuación la palabra “TareaA” con el número de la *actividad*, será permitido solo 4 niveles en el número. Por ejemplo dir1TareaA1.6.1
- *Presentation Name*: Las tres letras en minúsculas representativas del **proceso**, seguido del número, por ejemplo: dir1. A continuación la palabra “TareaA” con el número de la *actividad*, por ejemplo dir1TareaA1.6.1, seguido de una oración la cual deberá iniciar con un verbo en infinitivo.

#### 4.4 ANÁLISIS DEL PROCESO DE GESTIÓN DE NEGOCIO Y MODELADO EN SPEM 2.0.

Para realizar el modelado de este **proceso**, se hará uso de los elementos antes mencionados: el mapeo SPEM-MoProSoft en el punto 4.2, parte de la jerarquía de desglose de trabajo (fig. 8) y algunas de las reglas descritas en el punto 4.1. De tal manera que el **proceso** debe quedar estructurado de la siguiente forma (fig. 10).



Figura 10.- Estructura de un proceso en SPEM 2.0 vista tareas, roles y productos de trabajo, respectivamente.

Para estructurar el **proceso** Gestión de Negocio de esta manera, primero se identifican los *elementos de contenido* (*tareas, roles y productos de trabajo*) de la versión coloreada de MoProSoft usando el elemento mapeo SPEM-MoProsoft. Para ello se hace un análisis del contenido de cada una de las **actividades**, un ejemplo de esto es el siguiente (fig 11)

RGN	<p>A1.2. Entender la situación actual.</p> <ul style="list-style-type: none"> <li>• Análisis del entorno – identificación de oportunidades y amenazas con base en: necesidades de los clientes, información sobre competidores, tendencias tecnológicas, etc.</li> <li>• Análisis de la situación interna - identificación de las fortalezas y debilidades con base en: análisis financieros, identificación de recursos, entre otras.</li> </ul>
-----	---

Figura 11.- **Actividad** de MoProSoft del **proceso** de Gestión de Negocio.

El primero de los elementos de contenido que se buscan identificar son las *tareas* (si es que hubiese más de una), en este caso son dos, la primera es “Realizar el análisis del entorno” y la segunda es “Realizar el análisis de la situación interna”.

Como segundo elementos de contenido se buscan cuales son los *roles* encargados de realizar cada una de estas *tareas*, en el caso específico de esta **actividad** solo es uno para las dos *tareas* que es el “Responsable de Gestión de Negocio (RGN)”, por lo que se considera que este *rol* es el encargado de realizar las dos *tareas*, pero además de esto, se revisa la definición de este **rol** (fig 12).

Responsable de Gestión de Negocio	RGN	Conocimiento de las actividades necesarias para definir e implantar exitosamente el proceso de Gestión de Negocio.
-----------------------------------	-----	--

Figura 12.- Rol Responsable de Gestión de Negocio

Dado que este **rol** es el encargado de “definir” e “implantar” el **proceso**, pues para hacerlo debe de conocer la situación de la organización en la que se encuentra, es decir, debe hacer un análisis de la misma. Por lo que la asignación de estas *tareas* a este *rol* es adecuada.

Como tercer elemento de contenido se buscan cuales son los *productos de trabajo* que se necesitan para realizar cada una de las dos *tareas*.

Para la *tarea* “Realizar el análisis del entorno”, se hace un análisis de la

descripción de la **tarea** y dado que no nos dice específicamente cuales son los **productos de trabajo** de entrada, internos o de salida, se podría pensar que no los hay. Pero al revisar en los **productos de trabajo** de entrada, internos y de salida del **proceso** en general (Gestión de Negocio), se encuentra que existe el siguiente **producto de trabajo**(fig 13):

Factores Externos (tendencias tecnológicas, clientes y competidores)	Externa
--	---------

Figura 13.- **Producto de trabajo** Factores Externos.

Este producto se necesita como entrada de la **tarea**, dado que a partir de él se hace el análisis del entorno.

No hay **productos de trabajo** internos, dado que el objetivo de la **actividad** es “Entender la situación actual”.

Dado que la **actividad** en sí es “Entender la situación actual”, no se producen **productos de trabajo** de salida.

Para la **tarea** “Realizar el análisis de la situación interna”, se hace un análisis de la descripción de la **tarea**, la cual nos indica que necesitamos “análisis financieros, identificación de recursos, entre otros”, por lo que al revisar en los **productos de trabajo** de entrada, internos y de salida del **proceso** en general (Gestión de Negocio), se encuentra que existe el siguiente **producto de trabajo**(fig 14):

Reportes Financieros	Organización
----------------------	--------------

Figura 14.- **Producto de trabajo** Reportes Financieros

Este **producto de trabajo** se necesita como entrada de la **tarea**, dado que a partir de él se hace parte del análisis de la situación interna.

No hay **productos de trabajo** internos, dado que el objetivo de la **actividad** es “Entender la situación actual”.

Y dado que la **actividad** en sí es “Entender la situación actual”, no se

producen **productos de trabajo** de salida.

Este análisis para identificar cada uno de los elementos de método (elementos primarios), se hizo con cada una de las **actividades**, para lo cual mostrare algunos casos que tuvieron un principal interés. Por ejemplo cada uno de los elementos del **producto de trabajo** “Plan Estratégico” se tomaron igual como *productos de trabajo*, es el caso en la **actividad** A1.1 Articular, documentar o actualizar la Misión, Visión y Valores (fig 15).

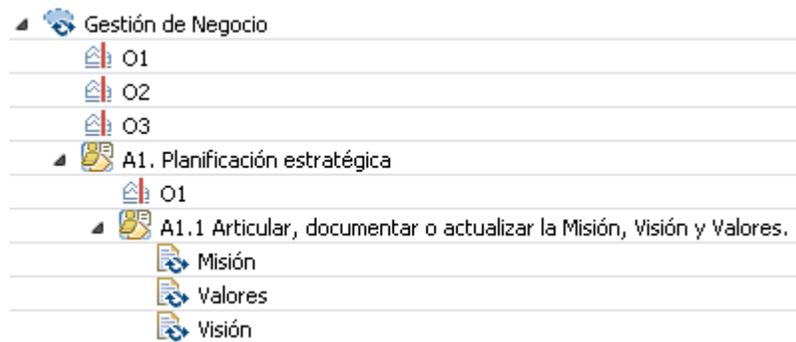


Figura 15.- Subproductos pertenecientes al producto de trabajo Plan Estratégico.

Al representar de esta manera a estos *productos de trabajo*, se cumple el objetivo de la **actividad** A1.10 Integrar y documentar el Plan Estratégico, ya que como entrada están todos los *subproductos* y como salida quedan agrupados en uno solo, que es el Plan Estratégico (fig 16).

▲ A1.10 Integrar y documentar el Plan Estratégico	
▲ Plan estratégico	Output
Cartera de Proyectos	Mandatory Input, Ou...
Estrategia de Recursos	Mandatory Input, Ou...
Estrategias	Mandatory Input, Ou...
Estructura de la Organización	Mandatory Input, Ou...
Misión	Mandatory Input, Ou...
Objetivos	Mandatory Input, Ou...
Periodicidad de Valoración	Mandatory Input, Ou...
Plan de Comunicación con el Cliente	Mandatory Input, Ou...
Presupuesto	Mandatory Input, Ou...
Procesos Requeridos	Mandatory Input, Ou...
Valores	Mandatory Input, Ou...
Visión	Mandatory Input, Ou...

Figura 16.- Integración de los subproductos de trabajo del producto de trabajo Plan Estratégico, ordenados alfabéticamente.

Con respecto a los *roles*, se indica que **rol o roles** son los responsables primarios de la tarea y quienes son los responsables adicionales o secundarios, en el caso de que el **rol** sea de apoyo solamente.

Ya que se hizo el análisis de los elementos primarios, se hace uso de la idea principal de SPEM 2.0 (ver fig 3) para relacionarlos, de tal forma que en la *tarea* estén relacionados de la siguiente forma: qué *rol o roles* son los encargados de realizar la *tarea* y qué *productos de trabajo* de entrada se necesitan y de salida se producen de la misma (fig 17).

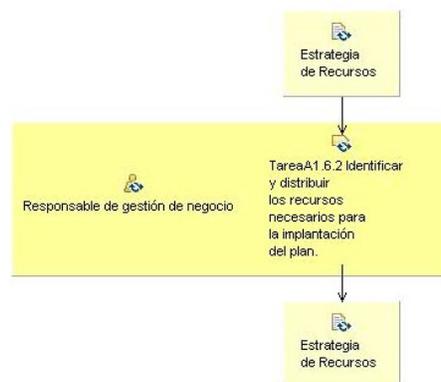


Figura 17.- Relación de elementos principales (Tarea – Rol – Productos de trabajo)

Ya que se cuenta con todas las relaciones entre los elementos

principales, se prosigue ahora a formar *actividades y procesos*.

Primero se generó una estructura general tomando como base la jerarquía de desglose de trabajo (fig 8), dicha estructura nos va a servir en los diferentes niveles de capacidad de MoProSoft, es decir, se toman las **actividades** principales del **proceso** que en el caso del proceso Gestión de Negocio son 3.

Dependiendo del nivel de capacidad, si en las actividades principales no hay *tareas*, de cualquier forma esta estructura existirá, así, de esta forma indicamos que en los niveles posteriores existirán *actividades y tareas* por realizar. Un ejemplo de ello es la **actividad** A3 del proceso Gestión de Negocio de la siguiente figura 18, que en el nivel de capacidad 1 de MoProSoft no cuenta con **actividades y/o tareas**.

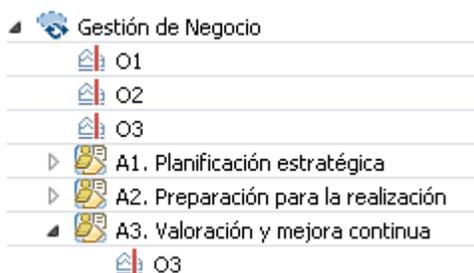


Figura 18.- Estructura general para el proceso Gestión de Negocio

Ya que se cuenta con la estructura *patrón de proceso* “Gestión de Negocio” junto con sus 3 *actividades*, se indican los *objetivos* generales del proceso y a partir de ellos se indica en cada una de las *actividades* de esta estructura general, cuáles de estos *objetivos* se deben de cumplir (ver fig 19).

Además, se hace el análisis por cada **actividad** si el nivel de detalle es suficiente para saber cuál es la *tarea* a realizar, si no lo fuese se hace un desglose, de tal manera que quede al final una sola *tarea* por *actividad*, un ejemplo de esto es la figura 19.

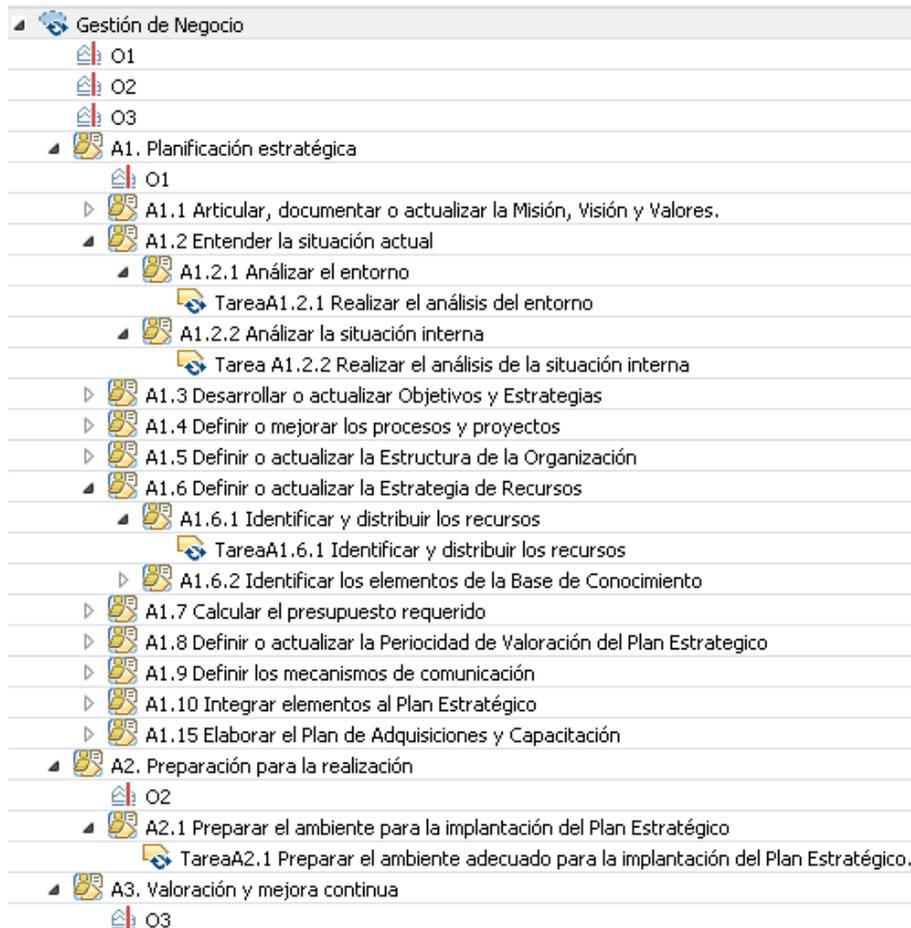


Figura 19.- Proceso Gestión de Negocio (nivel de capacidad 1)

Antes de continuar con el siguiente punto, se deben mencionar casos específicos de los otros **procesos** que en el **proceso** Gestión de Negocio no se presentaron, tal es el caso del **proceso** “Desarrollo y Mantenimiento de Software (OPE2)”, donde se encuentra la **actividad** A2.2 Documentar o modificar la Especificación de Requerimientos (fig 20).

En esta **actividad** los **roles** involucrados son 4: Analista (AN), Cliente (CL), Usuario (US) y Diseñador de interfaz de Usuario (DU). Al describir la **actividad** no se indica de manera específica qué **rol** debe realizar qué **tarea**.

La forma de solucionarlo es hacer un análisis de la descripción de la

**actividad** y las descripciones de lo que puede realizar cada uno de los **roles**, es decir, las capacidades y habilidades.

AN CL US DU	<p>A2.2. Documentar o modificar la <i>Especificación de Requerimientos</i>.</p> <ul style="list-style-type: none"> <li>Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos.</li> <li>Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.</li> <li>Elaborar o modificar el prototipo de la interfaz con el usuario.</li> <li>Generar o actualizar la <i>Especificación de Requerimientos</i>.</li> </ul>
----------------------	---

Figura 20.- Actividad A2.2 (OPE2) con 4 roles

Después de realizar el análisis de cada una de las descripciones tanto de las **tareas** como de los **roles** e identificar cada una de los elementos primarios y combinarlos, el resultado es el siguiente (fig 21):

▲	A2.2 Documentar o modificar la especificación de requerimientos
▲	A2.2.1 Identificar y consultar fuentes de información
	Tarea A2.2.1 Identificar y consultar fuentes de información
▲	A2.2.2 Analizar los requerimientos
	Tarea A2.2.2 Analizar los requerimientos
▲	A2.2.3 Elaborar o modificar el prototipo de la interfaz
	Tarea A2.2.3 Elaborar o modificar el prototipo de la interfaz
▲	A2.2.4 Generar o actualizar la especificación de requerimientos
	Tarea A2.2.4 Generar o actualizar la especificación de requerimientos
▶	A2.10 Documentar manual de usuario
▶	A2.13 Incorporar elementos a la configuración de software

Figura 21.- Modelado de la actividad A2.2 (OPE2) – vista tareas

Y la relación de los roles queda de la siguiente forma (fig 22):

▲	A2.2 Documentar o modificar la especificación de requerimientos
▲	A2.2.1 Identificar y consultar fuentes de información
	Analista
	Cliente
	Usuario
▲	A2.2.2 Analizar los requerimientos
	Analista
	Cliente
	Usuario
▲	A2.2.3 Elaborar o modificar el prototipo de la interfaz
	Diseñador de interfaz de usuario
▲	A2.2.4 Generar o actualizar la especificación de requerimientos
	Analista

Figura 22.- Modelado de la actividad A2.2 (OPE2) – vista roles

Por lo que de esta manera, se especifica de manera clara cuantos y cuales *roles* participan en la realización de cada una de las *tareas*.

#### 4.5 INTEGRACIÓN DE TODOS LOS PROCESOS.

El trabajo que se realizó para el **proceso** Gestión de Negocio, es el mismo que se realiza con cada uno de los 9 **procesos** de MoProSoft.

Para llevar a cabo la integración de todos los **procesos** se hace uso completo del mapeo SPEM-MoProSoft en el punto 4.2, de la jerarquía de desglose de trabajo (fig. 8) y las reglas descritas en el punto 4.1.

Para integrar se crea en un principio el *proceso de despliegue* MoProSoft, que está representado por el ícono . A este *proceso de despliegue* se le agregan de forma jerárquica cada uno de los diferentes *patrones de procesos*  generados de manera similar al **proceso** de Gestión de Negocio.

Un ejemplo como quedan los *procesos*, se muestra en la siguiente figura.



Figura 23.- Los 9 procesos de MoProSoft

En la figura 23, se muestran los 9 procesos de MoProSoft estructurados de forma jerárquica.

#### 4.6 MODELADO DE LOS NIVELES 1, 2, 3 Y 4 DE MOPROSOFT.

El modelado del nivel de capacidad 1 se llevó a cabo como se menciona en los puntos anteriores. Ya que se tiene este modelado, se extiende a partir de él, el nivel de capacidad 2 y a partir de “contribuciones locales” se van agregando las nuevas *actividades y tareas* a este nivel.

Este procedimiento se repite con cada uno de los *procesos* posteriores, de tal manera que con base en el nivel anterior se hacen contribuciones locales, es decir, en otros términos hacemos una herencia del nivel anterior.

Al armar cada uno de los niveles de esta manera, si se llegara a necesitar realizar un cambio en uno de los primeros niveles, dicho cambio se ve reflejado en los niveles superiores de forma automática al momento de guardar.

##### 4.6.1 MODELADO PARA EL NIVEL 2 DE MOPROSOFT.

Para el modelado del nivel de capacidad 2 de MoProSoft se hizo una reutilización del nivel de capacidad 1, a partir de una variabilidad del tipo “*Extends*”. De esta forma por medio de contribuciones locales se agregan las *actividades, tareas y productos de trabajo* que el nivel de capacidad 2 de MoProSoft adiciona.

En seguida se listan los **productos de trabajo, actividades y tareas** que se agregaron al nivel de capacidad 1:

- El Plan de Comunicación e Implantación .- las actividades y tareas que los generan, validan y corrigen.
- Los Reportes de Verificación y los Reportes de Validación .- las actividades y tareas que los generan.
- Productos de trabajo .- las actividades y tareas de verificación,

validación y corrección de todos los productos de trabajo

- La Asignación de Recursos .- las actividades y tareas que los generan, verifican y validan.
- Plan de Procesos .- la actividad y tarea de seguimiento.
- Supervisar el control de riesgos de acuerdo al Plan de Manejo de Riesgos de procesos .- la actividad y tarea.
- El Reporte de Seguimiento .- las actividades y tareas que los generan, verifican y validan.
- Las Acciones Correctivas o Preventivas .- las actividades y tareas que los generan, verifican y validan.
- Los Comentarios y Quejas del Cliente .- las actividades y tareas que los generan, verifican y validan.
- Los Mecanismos de Comunicación con los Clientes .- las actividades y tareas que los generan, verifican y validan.
- El Reporte de Recursos Humanos Disponibles, Capacitación y Ambiente de Trabajo .- las actividades y tareas que los generan, validan y/o corrigen.
- El Reporte de Bienes, Servicios e Infraestructura .- las actividades y tareas que los generan, validan y/o corrigen.
- El Reporte del Estado de la Base de Conocimiento .- las actividades y tareas que los generan, validan y/o corrigen.
- El Reporte de Ambiente de Trabajo .- las actividades y tareas que lo generan, validan y/o corrigen.
- El Reporte de Capacitación .- las actividades y tareas que lo generan, validan y/o corrigen.
- La Evaluación de Desempeño .- las actividades y tareas que lo generan, validan y/o corrigen.
- La Encuesta sobre el Ambiente de Trabajo .- las actividades y tareas que lo generan, validan y/o corrigen.
- El Registro de Mantenimiento .- las actividades y tareas que lo

generan, validan y/o corrigen.

- Evaluación periódica de la satisfacción del solicitante por el bien o servicio adquirido, y su registro en el Catálogo de Proveedores .- la actividad y tarea.
- Dar seguimiento al Plan de Mantenimiento .- la actividad y la tarea.
- El Reporte de Actividades .- las actividades y tareas que lo generan, validan y/o corrigen.
- La Solicitud de Cambios .- las actividades y las tareas que la generan, validan y/o corrigen.
- Las Acciones Correctivas para corregir una desviación o defecto con respecto al cumplimiento del Plan del Proyecto y Plan de Desarrollo .- las actividades y tareas que los generan, validan y/o corrigen.
- La Minuta .- las actividades y tareas que las generan, validan y/o corrigen.
- Actualización del Plan de Proyecto y del Plan de Desarrollo a causa de una Solicitud de Cambios por parte del cliente o Acciones Correctivas o Preventivas .- Actividad y tarea.
- La tarea de efectuar el cierre con subcontratistas de acuerdo al contrato establecido.
- El Registro de Rastreo .- las actividades y tareas que lo generan, actualizan, validan y/o corrigen.
- El Plan de Pruebas del Sistema .- las actividades y tareas que lo generan, validan y/o corrigen.
- El Reporte de Pruebas del Sistema .- las actividades y tareas que lo generan, validan y/o corrigen.
- El Plan de Pruebas de Integración .- las actividades y tareas que lo generan, validan y/o corrigen.
- El Reporte de Pruebas de Integración .- las actividades y tareas

que lo generan, validan y/o corrigen.

- El Manual de Mantenimiento .- las actividades y tareas que lo generan, validan y/o corrigen.
- La tarea de aplicación de pruebas unitarias.
- La tarea de realización de pruebas del sistema.

#### **4.6.2 MODELADO PARA EL NIVEL 3 DE MOPROSOFT.**

Para el modelado del nivel de capacidad 3 de MoProSoft se hizo una reutilización del nivel de capacidad 2, a partir de una variabilidad del tipo “Extends”. De esta forma por medio de contribuciones locales se agregan las *actividades, tareas y productos de trabajo* que el nivel de capacidad 3 de MoProSoft adiciona.

En seguida se listan los **productos de trabajo, actividades y tareas** que se agregaron al nivel de capacidad 2:

- Los Reportes Cuantitativos y Cualitativos, los procesos y proyectos .- las actividades y tareas que los generan, verifican y validan.
- El Plan de Medición de Procesos.
- El Reporte de Acciones Correctivas o Preventivas Relacionadas con los Clientes .- las actividades y tareas que lo generan, verifican y validan.
- Las Propuestas Tecnológicas .- las actividades y tareas que las generan, verifican y validan.
- Los Indicadores del Plan Estratégico.
- Los Reportes de Mediciones y Sugerencias de Mejora de todos los procesos .- las actividades y tareas que los generan, verifican y validan.
- Las Lecciones Aprendidas .- las actividades y tareas que las generan, consultan e incorporan a la Base de Conocimiento.
- Los Reportes de Valoración .- las actividades y tareas que los

generan, verifican y validan.

- El Plan de Acciones, Plan de Mejora y Reporte de Evaluación .- las actividades que los generan, verifican, validan y les dan seguimiento.
- El Proceso Específico del Plan de Desarrollo y las actividades que lo generan.
- La actividad y tarea de A1.18 del proceso de Administración de Proyectos Específicos.

#### **4.6.3 MODELADO PARA EL NIVEL 4 DE MOPROSOFT.**

Para el modelado del nivel de capacidad 4 de MoProSoft se hizo una reutilización del nivel de capacidad 3, a partir de una variabilidad del tipo "Extends". De esta forma por medio de contribuciones locales se agregan las *actividades, tareas y productos de trabajo* que el nivel de capacidad 4 de MoProSoft adiciona.

En seguida se listan los **productos de trabajo, actividades y tareas** que se agregaron al nivel de capacidad 3:

- Mediciones y las Metas Cuantitativas de todos los productos .- las actividades y tareas que los producen.
- Propuestas de Mejora de todos los productos .- las actividades que los producen y validan.

## 5 ANÁLISIS DEL MODELADO POR NIVEL DE CAPACIDAD Y PRESENTACIÓN DE RESULTADOS.

Cada uno de los modelados generados por nivel de capacidad de MoProSoft siguen cumpliendo con la norma misma, dado que lo que se realizó es un desglose de cada uno de los **procesos**, de tal manera que la relación fuese 1 a 1, es decir, conocer específicamente que *tarea* va a realizar cada uno de los *roles* y por lo tanto conocer que *productos de trabajo* de entrada se necesitan y qué *productos de trabajo* de salida se van a producir. Algunas de las modificaciones que fueron necesarias hacer, fue el nombrado y/o el desglose de las *actividades* y/o *tareas*, esto debido al análisis realizado en cada una de ellas. Al realizar este análisis se encontraron diferentes defectos, los cuales se mencionarán más adelante.

Además, para completar dicho análisis se analizarán los defectos encontrados en la tesis “Utilización de redes de **Petri** coloreadas para la validación de la consistencia del Modelo de Procesos para la industria de software (**MoProSoft**)” (Ramírez, 2007) que en lo siguiente se nombrará “Petri-MoProSoft”, la cual validó la consistencia de los procesos de MoProSoft y encontró 8 defectos.

Cada uno de estos defectos serán retomados en el nivel de capacidad en el que se encontraron y se definirá si al realizar el análisis para el modelado SPEM-MoProSoft también se presentó y como se solucionó. Si no se presentó el defecto por la forma en que se fueron estructurando los elementos de SPEM 2.0, se determinará si se solucionó con el modelado completo de SPEM-MoProSoft.

## 5.1 REPORTE DEL MODELADO SPEM-MOPROSOFT POR NIVELES DE CAPACIDAD.

### 5.1.1 NIVEL DE CAPACIDAD 1.

En este nivel se definió toda la estructura que soporta a los demás niveles. Al realizar el análisis de cada una de las **actividades** y **tareas**, para poder definir las y relacionarlas con los roles y productos de trabajo se encontró lo siguiente:

1. El definir a cada uno de los elementos del *Plan Estratégico* como *producto de trabajo*, dio como resultado que **tareas** como la de “Integrar o documentar el *Plan Estratégico*” cumplan su objetivo.
2. Al producto de trabajo *Plan de Adquisiciones y Capacitación* se le agregó un sufijo que corresponde a la abreviación del *proceso* en el que se genera, debido a que tiene el mismo nombre en diferentes *procesos* de los que sale, es decir, es un producto de salida. Esto es debido a que en la actividad A1.2 del **proceso** Gestión de Recursos, los 4 *productos de trabajo* *Plan de Adquisiciones y Capacitación* son necesarios como entrada, de esta forma se soluciona la ambigüedad presentada al momento de estar modelando MoProSoft (fig 24).

*Figura 24.- Sufijo en productos de trabajo*

3. En la **actividad** A1.1 del **proceso** Recursos Humanos y Ambiente de Trabajo se menciona que se debe hacer una revisión de las acciones correctivas, pero dicho **producto de trabajo** no se genera en este nivel, se genera en el nivel de capacidad 2 (fig 25).

*Figura 25.- A1.1 del proceso Recursos Humanos y Ambiente de trabajo*

Por lo tanto, después de realizar el análisis de las entradas y

salidas de las *actividades y tareas*, en el modelado se quitó y quedó de la siguiente forma (fig 26):

*Figura 26.- Modelado en SPEM 2.0 de la A1.1 del proceso Recursos Humanos y Ambiente de Trabajo*

Es decir, no existe dicho *producto de trabajo*, ni como entrada, ni como salida, por lo que podemos concluir que con esta modificación se soluciona.

Esto mismo sucede en la **actividad** A1.1 del proceso Bienes, Servicios e Infraestructura.

4. El defecto número 1 de la tesis Petri-MoProSoft indica lo siguiente:

En el análisis que se realizó para el modelado SPEM-MoProSoft, efectivamente el *Plan de Procesos* no es usado totalmente en ninguno de los **procesos**, pero si parcialmente, debido a que uno de sus *productos de trabajo* que es el *Plan de Adquisiciones y Capacitación (GES1)* es necesario para varias *tareas* del mismo **proceso** *Gestión de Procesos* y algunas otras del **proceso** *Gestión de Recursos* (ver fig 27).

Figura 27.- Producto de trabajo Plan de Procesos

Por lo que cuando se estaba modelando en SPEM 2.0 al llamar a un *producto de trabajo* con el mismo nombre en los diferentes *procesos*, como es el caso del “Plan de Adquisiciones y Capacitación” y que dicho *producto de trabajo* es diferente dado el *proceso* que lo genera, se puede pensar que es uno solo y llegar a interpretar de forma errónea. Al agregarle los calificativos a estos *productos de trabajo*, se soluciona esto. Cabe mencionar

que en la tesis Petri-MoProSoft se indica que la granularidad del producto de trabajo no es la adecuada, y dado que ese es uno de los elementos que se ha tratado de enfatizar en la tesis, y se soluciona desglosando cada uno de los *productos de trabajo* en los *productos de trabajo* que sean necesarios.

5. El defecto número 3 de la tesis Petri-MoProSoft indica lo siguiente:

Efectivamente en la versión coloreada de MoProSoft no indica a el Plan de Procesos como entrada, pero dado el análisis para realizar el modelado y la forma en la que se estructura a los *elementos de contenido*, de forma automática se coloca como entrada del *proceso* mismo. Es decir, dado que se necesita como entrada de las **tareas** pertenecientes al **proceso** *Gestión de Procesos* queda como entrada de todo el **proceso**, por lo que, quedó solucionado.

6. El defecto número 4 de la tesis Petri-MoProSoft indica lo siguiente:

Este defecto que en la tesis Petri-MoProSoft se menciona, es debido a que en las **tareas** en que se necesitan estos **productos de trabajo**, no se menciona que son requeridos explícitamente.

En el modelado SPEM-MoProSoft, se hizo el análisis de los textos que describen a las **actividades** y a las **tareas**. Con dicho análisis se logró identificar en que **tareas** y **actividades** no era claro lo que se necesitaba, éste es uno de los casos.

El resultado es que el **producto de trabajo** *Factores Externos* es requerido en la **actividad** A1.2.1 Analizar el entorno, que pertenece al **proceso** *Gestión de Negocio*, y el producto **de**

**trabajo Reportes Financieros** es requerido en la **actividad A1.2.2** Analizar la situación interna, que pertenece al **proceso** Gestión de Negocio (fig 28), por lo tanto, este defecto fue mal identificado por falta de claridad, pero con el modelado realizado queda solucionado.

*Figura 28.- Productos de trabajo Factores Externos y Reportes Financieros.*

7. El defecto número 5 de la tesis Petri-MoProSoft indica lo siguiente:

Ya que el **producto de trabajo Asignación de Recursos** no es generado por ninguna actividad en el nivel de capacidad 1, lo que se decidió hacer es quitar la *actividad A1.3.1* (junto con la *tarea y roles* asociados) y ponerla en el nivel de capacidad 2, con esto queda solucionado.

8. El defecto número 7 de la tesis Petri-MoProSoft indica lo siguiente:

Así como lo plantea no es muy claro, debido a que el **producto de trabajo Plan de Proyectos** contiene una “Descripción de las actividades para gestionar los proyectos externos e internos” y el

**producto de trabajo** *Plan de Proyecto* es otro documento que contiene todos los elementos de un proyecto, por lo que en el modelado SPEM-MoProSoft, esta **actividad** queda desglosada de la siguiente manera (fig 29):

Figura 29.- Desglose de actividad A2.2, vista productos de trabajo

Al desglosar de esta manera las *actividades*, *tareas* y a consecuencia de ello los *productos de trabajo*, se puede observar claramente cuales son los *productos de trabajo* obligatorios de entrada y cuales son los productos de trabajo de salida.

9. El defecto número 8 de la tesis Petri-MoProSoft indica lo siguiente:

En el análisis que se realizó para generar el modelado del nivel de capacidad 1 de MoProSoft en SPEM 2.0, se indica que este defecto fue mal detectado, debido a que el **producto de trabajo** *Configuración de Software* es requerido en varias *tareas* del **proceso** *Desarrollo y Mantenimiento de Software*, como lo indica la siguiente figura (fig 30).

Figura 30.- Producto de trabajo Configuración de Software en el nivel de capacidad 1

### 5.1.2 NIVEL DE CAPACIDAD 2.

1. En el **proceso** Gestión de Procesos, se agregó la **actividad** A1.3.1, junto con sus **tareas**, **roles** y **productos de trabajo** asociados. Esta solución se tomó por que la **actividad** estaba en

el nivel de capacidad 1, pero el **producto de trabajo** que requería es generado en el nivel de capacidad 2, por lo que se decidió pasar la **actividad** a este nivel.

2. El defecto número 6 de la tesis Petri-MoProSoft indica lo siguiente:

Efectivamente en la versión coloreada de MoProSoft se indica en el **proceso** Gestión de Negocio a el **producto de trabajo** *Plan de Comunicación e Implatación* que tiene como destino el **proceso** Gestión de Proyectos, pero dado el análisis para realizar el modelado y la misma forma en la que se estructura a los *elementos de contenido*, de forma automática se colocan como entrada o salida de los **procesos** mismos, es decir, si se necesitara como entrada de las *tareas* pertenecientes al *proceso* *Gestión de Proyectos* al finalizar el modelado, quedaría como entrada de todo el *proceso*, por lo que, queda solucionado.

### 5.1.3 NIVEL DE CAPACIDAD 3.

1. El defecto número 2 de la tesis Petri-MoProSoft indica lo siguiente:

Al realizar el análisis de que **productos de trabajo** de entrada

son necesarios en la **actividad** A1.2 de Gestión de Recursos, efectivamente el **producto de trabajo** Plan de Adquisiciones y Capacitación (GES1), es necesario, así como los demás **productos de trabajo** que tienen el mismo nombre, esto ha sido explicado en el punto 2 del nivel de capacidad 1 en el punto 5.1.1. Por lo que se considera que al identificar bien cada uno de los *productos de trabajo* (indicando el proceso en que se generan) y saber de donde provienen se soluciona.

## **5.2 ANÁLISIS DEL MODELADO SPEM-COMPETISOFT.**

Competisoft (Piattini, 2008) es un modelo que surgió como una modificación de MoProSoft. El libro (Piattini, 2008) contiene una representación del modelo de procesos de Competisoft en EPFC. Ambos modelos son bastante parecidos sin embargo sus modelados en SPEM 2.0 tienen diferencias. A continuación se comparan los dos mapeos e identifican ventajas y desventajas de cada uno.

Al hacer un análisis del modelado realizado para Competisoft, se pudo definir de forma general las siguientes observaciones:

- Se realizó una categoría personalizada que agrupara todos los procesos, llamada “Competisoft”.
- Cada uno de los procesos del modelo Competisoft, se mapearon a “patrones de procesos” (concepto de SPEM 2.0).
- Las actividades del modelo Competisoft, que se derivan de los procesos, se mapearon a “patrones de procesos”.
- Las subactividades del modelo se mapearon a “Tareas”.

Un ejemplo del modelado que se generó para Competisoft se presenta en la fig 31:

Figura 31.- Modelado SPEM-Competisoft

## **5.3 COMPARACIÓN ENTRE LOS MODELADOS SPEM-MOPROSOFT Y SPEM-COMPETISOFT.**

Al terminar de analizar el modelado que se realizó para Competisoft y observar parte de éste mismo (fig 31), se hace evidente lo siguiente: una de las características principales e importantes de SPEM

2.0 que no se respetó en el mapeo de Competisoft fue la jerarquía de desglose de trabajo (fig. 8), debido a que se inhibió a las actividades, es decir, se relacionó directamente una tarea con un proceso, y con ello no se considero que “En el diseño de un modelo de Procesos de Software se deben respetar las relaciones entre los diferentes conceptos definidos en SPEM.” (Ruiz, 2008).

De la misma forma se analizó el modelado que se generó para MoProSoft. Un ejemplo de dicho modelado se presenta en la fig 32.

*Figura 32.- Modelado SPEM-MoProSoft*

Al comparar estas dos figuras con la figura 8, se puede observar que el modelado realizado para MoProSoft respeta la jerarquía de desglose de trabajo y que el modelado de Competisoft, no.

#### **5.4 VENTAJAS Y DESVENTAJAS DE AMBOS MODELADOS.**

##### **Ventajas, modelado SPEM-Competisoft**

Este modelado se apega al modelo MoProSoft, es decir, las relaciones entre los diferentes elementos del modelo son respetadas.

##### **Ventajas, modelado SPEM-MoProSoft**

Este modelado respeta la jerarquía de desglose de trabajo, por lo tanto, el apego al metamodelo es considerable.

Se observa una mejora en compresión y claridad del modelo, debido al desglose y relaciones hechas entre tareas, roles y productos de trabajo.

Se agregaron las verificaciones y validaciones como tareas, es decir, forman parte de la actividad que las cita.

##### **Desventajas, modelado SPEM-Competisoft**

Este modelado no respeta la jerarquía de desglose de trabajo.

No tiene actividades.

No se puede poner estatus de entrada/salida a los productos de trabajo de una actividad.

**Desventajas, modelado SPEM-MoProSoft**

Se tuvieron que agregar elementos (subactividades, validaciones y verificaciones como tareas, etc.) para alcanzar el desglose y detalle deseado.

No se apega al Modelo MoProSoft, a consecuencia de los elementos agregados.

## **6 CONCLUSIONES Y TRABAJO A FUTURO.**

Hay que enfatizar la idea de que es difícil mejorar cualquier proceso mientras éste no sea estandarizado. El análisis que se presenta en esta tesis permite afirmar que MoProSoft en su versión 1.3 con respecto al estándar SPEM 2.0 no lo estaba, por lo que este trabajo se enfocó en estandarizar y modelar un modelo de procesos de software, en este caso MoProSoft, en SPEM 2.0.

Para lograrlo, se desarrolló un modelado por cada nivel de capacidad de MoProSoft en SPEM 2.0 y para representar cada uno de ellos se eligió la herramienta EPFC, debido a que nos permite modelar, estructurar, gestionar y publicar cualquier proceso de software que esté modelado por SPEM 2.0.

Para trabajar el modelo se definió en una primera instancia la estructura con la que se trabajaría en los diferentes niveles de capacidad, respetando el mapeo SPEM-MoProSoft, las reglas para modelar y la estructura de desglose de trabajo. De esta manera, por medio de la variabilidad de SPEM 2.0 se fue generando de manera progresiva el modelo. Al analizar las descripciones de las actividades y/o tareas, y estar formando los procesos se detectaron varios de los defectos mencionados. Esto ayudó a que al momento de revisar los defectos encontrados en la tesis Petri-MoProSoft la mayoría de ellos estuvieran resueltos.

Al modelar cada uno de los niveles de capacidad de MoProSoft en SPEM 2.0 y representarlos en EPFC, podemos concluir que se ha mejorado la calidad del proceso mismo, así como también se facilita la comunicación, distribución y uso del mismo.

Esto se afirma por las diferentes formas en las que se puede exportar el

modelo, de este modo el costo de la incorporación del nuevo personal a una MiPyME podría disminuir considerablemente y el costo de realizar las nuevas adecuaciones al modelo dado un nuevo proyecto, deben ser mínimas, dado que se encuentra en formato electrónico el cual se encontrara en <http://www.mcc.unam.mx/moprosoft/tesis> y se cuenta con la herramienta de forma gratuita.

Considerando todos estos argumentos, se puede decir que se logró cumplir con los objetivos planteados.

### **Trabajo Futuro**

Entre las posibles mejoras al trabajo se puede mencionar lo siguiente:

- Implementar el modelado SPEM-MoProSoft en MiPyME's, por medio de los plugin's (uno por cada nivel de capacidad) generados y extraer información que ayude a mejorar el proceso mismo.
- Que las MiPyME's o entidades del ámbito académico generen un repositorio de las diferentes instancias y adecuaciones que se generen del mismo modelado SPEM-MoProSoft.
- De los modelos exportados en xml de diferentes instancias basadas en el modelado SPEM-MoProSoft, hacer un análisis para extraer ciertos patrones de procesos que en proyectos parecidos se presenten.

## 7 REFERENCIAS

- (ProSoft, 2009) - Prosoft, SE (2009, Septiembre, 29) , “Programa para el desarrollo de la industria del software (Prosoft)”, [En línea], Disponible en: <http://www.economia.gob.mx/?P=1128>
- (NYCE, 2009) NYCE (2009, Septiembre, 29) , “Acreditación de la norma MoProSoft”, [En línea], Disponible en: <http://www.nyce.org.mx/uvti.htm>
- (SEI, 2009) Software Engineering Institute (2009, Septiembre, 29) , “Capability Maturity Model Integration (CMMI)”, [En línea], Disponible en: <http://www.sei.cmu.edu/cmmi/>
- (Ventura, 2006) - Ventura Miranda Ma. Teresa, Peñaloza Báez Marcela (2006, Marzo) , “MoProSoft: modelo de procesos de software hecho en México”, [En línea], Disponible en: <http://www.enterate.unam.mx/Articulos/2006/marzo/moprosoft.htm>
- (SPEM, 2007) - Object Management Group, “Software & Systems Process Engineering Metamodel Specification, v2.0”, Document Number: ptc/2007-08-07, Beta 2 pp. 234.2007.
- (Contacto PyME, 2009) - Contacto PyMe, SE (2009, Septiembre, 29) , “Información de Micros, Pequeñas y Medianas Empresas”, [En línea], Disponible en: <http://www.economia.gob.mx/?P=7700>
- (Rosas, 2006) - Rosas, Ana María (2006, Mayo, 12) , “México a la vanguardia de las TI: Economía”, [En línea], Disponible en: <http://fox.presidencia.gob.mx/buenasnoticias/ciencia/?contenido=24906&pagina=9>
- (Kroll, 2009) - Kroll per (2009, Mayo, 20) , “Eclipse Process Framework Project (EPF)”, [En línea], Disponible en: <http://www.eclipse.org/epf/>
- (Marchal, 2009) Marchal David (2009, Septiembre, 29) , “Oferta y demanda de profesionales de TI, el equilibrio imperfecto”, [En línea], Disponible en: <http://www.equiposytalento.com/contenido/download/17/equilibrio.pdf>
- (Oktaba, 2005) - Oktaba Hanna, Alquicira Esquivel Claudia y Bastida Martínez Brenda, “Modelo de Procesos de Software (MoProSoft) por niveles de capacidad de procesos”, Versión 1.3, Agosto de 2005.

- (OMG, 2009) - Object Management Group (2009, Mayo, 10), "*Object Management Group (OMG)*", [En línea], Disponible en: <http://www.omg.org/>.
- (UML, 2009) - OMG (2009, Mayo, 10), "*Unified Modeling Language*", [En línea], Disponible en: <http://www.uml.org/>
- (Ruiz, 2008) - Ruiz Francisco, Verdugi Javier, "*Guía de uso de SPEM 2.0 con EPF Composer*", Versión 3.0, pp. 11, 1 de Abril de 2008.
- (Piattini, 2008) - Oktaba Hanna, Mario Piattini, Francisco J. Pino, Maria Julia Orozco y Claudia Alquicira, "*COMPETISOFT: Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos*", editorial Ra Ma, España, noviembre de 2008, ISBN: 978-84-7897-901-1.
- (Ramirez, 2007) – Ramirez Ramos, Alejandro Alberto, "*Utilización de redes de Petri coloreadas para la validación de la consistencia del Modelo de Procesos para la industria de software (MoProSoft)*", Tesis UNAM, MCC, México, 2007.

## 8 APÉNDICE A.

### 8.1 MANUAL DE ECLIPSE PROCESS FRAMEWORK COMPOSER (EPFC).

Como ya se menciona EPFC es un editor de procesos SPEM 2.0, desarrollado dentro del entorno ECLIPSE, que sirve para editar fragmentos de método, procesos o metodologías, y generar automáticamente la documentación adecuada en formato para la web y que además es gratuito. A continuación se describen los pasos sugeridos de uso de la herramienta para construir un nuevo modelo.

### 8.2 CONTENIDO DE MÉTODO

Para poder generar un nuevo modelo es necesario que primero se definan los elementos de método (rol, productos de trabajo, tareas y guías), pero para definirlos, hay que definir los siguientes elementos que los albergaran.

Primero hay que crear una biblioteca de métodos, esta contendrá todos los plugins y configuraciones de la metodología que va a definirse, para crearla hay que seleccionar en la herramienta Archivo->Nuevo->Biblioteca de métodos (fig 33).

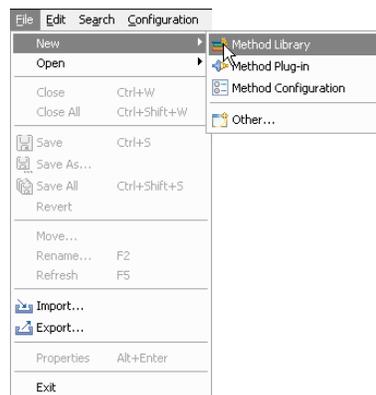


Figura 33.- Creación de una Biblioteca de métodos.

Ya que se creó la biblioteca de métodos, hay que agregarle un plugin de método, el cual contendrá los elementos necesarios para la definición de un proceso. Para crearlo hay que seleccionar en la herramienta Archivo->Nuevo->Plugin de método, o clic derecho en la vista biblioteca->Nuevo plugin de método (ver fig 33 y 34).

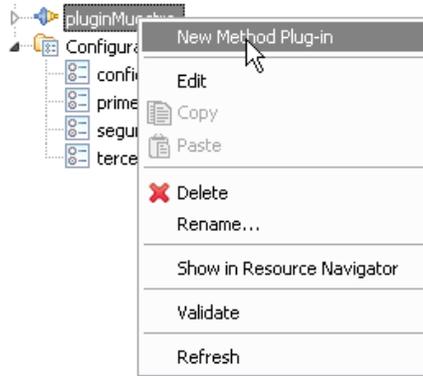


Figura 34.- Creación de un plugin.

Para crear los elementos de método (guías, productos de trabajo, roles, tareas y sus sub-apartados), primero se tiene que crear al menos un paquete de contenido ( clic derecho sobre Paquetes de contenido->Nuevo->Paquete de contenido ) ( opcionalmente, dependiendo de como se quiera estructurar la información se pueden crear paquetes de contenido dentro de otros) (fig 35).

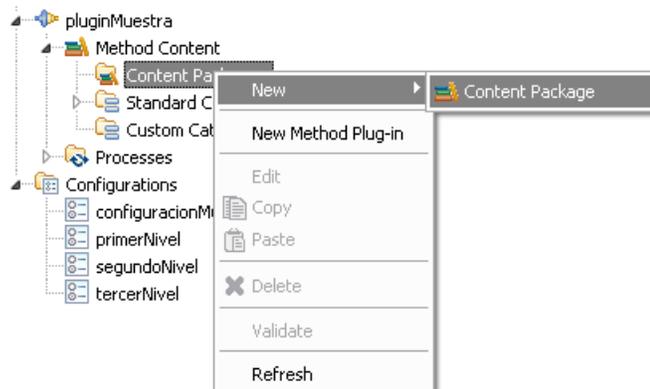
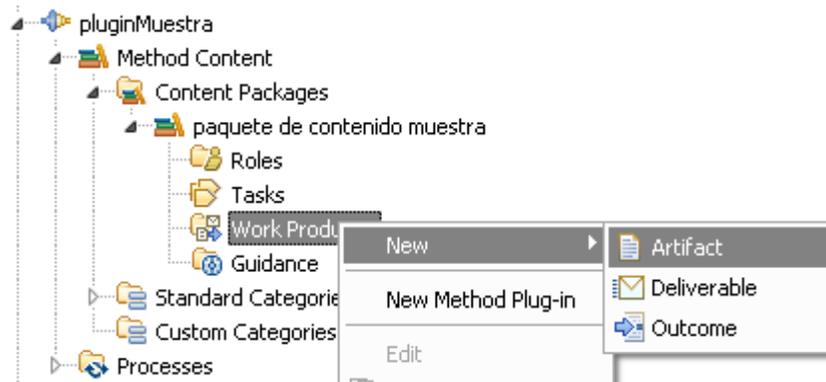


Figura 35.- Creación de Paquete de contenido.

Se crean de forma automática cuatro carpetas para almacenar las definiciones de cada tipo de elemento. Para crear un elemento de método se hace: clic derecho sobre una de dichas carpetas (la del tipo que nos interese)->Nuevo->Tipo\_del\_elemento.



*Figura 36.- Creación de un elemento de contenido.*

Donde Tipo\_del\_elemento puede ser: Rol, Tarea, Guía, Producto de trabajo.

### **8.2.1 CATEGORIZACIÓN DE ELEMENTOS DE MÉTODO.**

En la herramienta para crear una categoría estándar se utiliza el método habitual: clic derecho sobre el tipo de categoría que se quiere crear->Nuevo->Tipo\_de\_categoria (para acceder a las categorías estándar en la vista Biblioteca: nombre\_del\_plug-in->Contenido de Método->Categorías estándar) (fig 37).

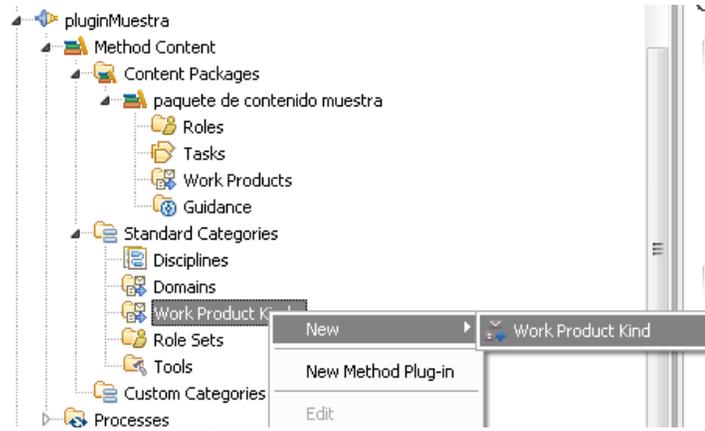


Figura 37.- Creación de una categoría estándar.

### 8.2.2 CREANDO PROCESOS.

En la herramienta una vez que se han creado los elementos de método y las categorías estándar, y antes de poder crear un proceso que hace uso de dichos elementos (sea patrón de proceso o proceso para despliegue), se debe crear una **configuración**. Esto es necesario porque cada elemento de proceso debe especificar su configuración (o configuraciones, ya que un proceso puede estar asociado con varias configuraciones).

Para crear una configuración: Archivo->Nuevo->Configuración de método, o desde la vista biblioteca: clic derecho sobre Configuraciones->Nuevo->Configuración de método (ver fig. 33 y fig. 38).

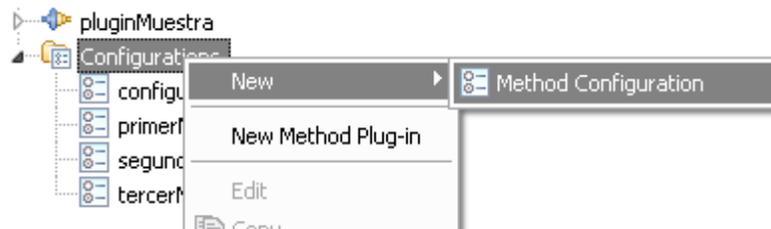


Figura 38.- Creación de una configuración.

En la herramienta una vez que se ha creado al menos una configuración, se pueden empezar a crear los procesos, se

recomienda primero crear los patrones de procesos de forma ascendente, para que se puedan reutilizar.

Lo anterior se puede conseguir si primero se crean los patrones de proceso básicos en los que se reutilizan las tareas definidas en el contenido de método. Para crear un patrón de proceso desde la vista de biblioteca: hacer clic derecho sobre Patrones de proceso->Nuevo->Patrón de Proceso (se accede a Patrones de proceso desde la vista Biblioteca: nombre\_del\_plug-in->Contenido de método->Procesos->Paquetes de proceso) (ver fig 39).

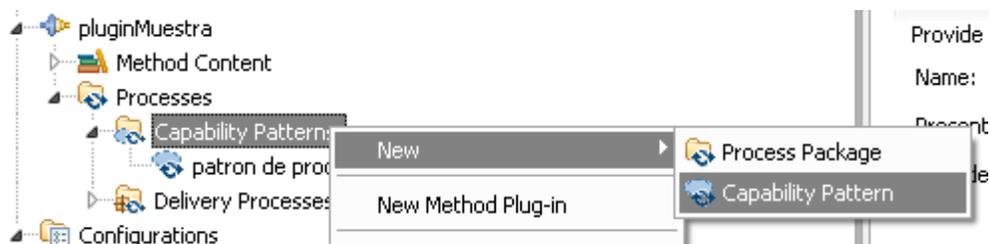


Figura 39.- Creación de patrón de proceso.

Una vez que se ha definido la estructura del proceso se pueden definir los diagramas de actividad, para especificar los detalles de cada actividad, así como las dependencias de productos de trabajo para cada patrón de proceso.

A partir de los patrones procesos básicos se pueden crear patrones de procesos más complejos. De esta forma se pueden crear jerarquías de patrones de proceso que reflejen la estructura de fases, iteraciones, actividades, etc. de un proceso.

Para usar un patrón en otro existen dos opciones: copiarlo o ampliarlo. Si el patrón que se va a aplicar no se piensa personalizar dentro del patrón aplicado, se recomienda utilizar la opción ampliar, ya que resulta mucho más fácil de mantener. Las modificaciones posteriores sólo hace falta hacerlas en el patrón original, y los cambios se reflejarán en todos los patrones donde se aplicó dicho

patrón mediante ampliación. Sin embargo, si el patrón debe personalizarse dentro del proceso en que se aplica, la aplicación mediante ampliación no permite hacerlo (permite añadir nuevos elementos, pero no eliminar elementos existentes que no sean útiles en el contexto en el que se aplica el patrón), por lo que habría que aplicarlo mediante copia. Un patrón copiado sí puede personalizarse para adaptarse al contexto en el que se aplica, sin embargo, su mantenibilidad disminuye, ya que los cambios posteriores deben hacerse sobre todas las copias manualmente.

Una vez creados los patrones de procesos, podemos crear los procesos de despliegue (ver fig 40) reutilizando dichos patrones de proceso.

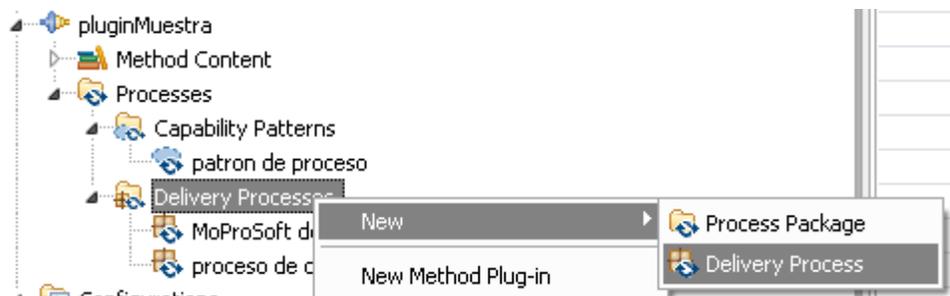


Figura 40.- Creación de un proceso de despliegue.

La reutilización de patrones de proceso para hacer procesos de despliegue es de la misma forma que el punto anterior.

En general con esto podemos implementar una jerarquía tarea-actividad-subproceso-proceso y con ello definir alguna metodología en específico.

## **9 GLOSARIO.**

### **Plugin**

Se trata de un pequeño programa que proporciona alguna funcionalidad específica a otra aplicación mayor o más compleja