

Proyecto Final de Carrera para obtener la titulación de:

Ingeniero Superior en Informática

**Especificación de procesos en la herramienta
CAME MOSKitt4ME y construcción de un marco
de evaluación para herramientas CAME y CASE**

Autor: Alicia Jáen Bielsa

29 septiembre de 2011

Director: Victoria Torres Bosch
Co-director: Manuela Albert Albiol

índice

1. Introducción.....	1
2. Objetivos.....	2
3. Introducción a la Ingeniería de Métodos.....	2
4. Aplicación de MOSKIT4ME para la construcción de soporte a procesos de gvMétrica.....	3
4.1. Contexto.....	4
4.2. Metamodelos para la definición de procesos.....	4
4.2.1. SPEM 2.0.....	4
4.2.2. BPMN 2.0.....	7
4.3. Definición de Procedimientos con MOSKitt4ME.....	8
4.3.1. Características de los Procedimientos de la CITMA.....	9
4.3.2. Definición de los procedimientos en EPF Composer.....	13
4.3.3. Generación de diagramas BPMN desde el EPFC.....	16
4.4. Construcción de herramientas CASE con MOSKitt4ME.....	20
5. Validación de herramientas CAME y CASE.....	23
6. Conclusiones generales.....	24
7. Agradecimientos.....	24
8. Trabajo futuros.....	24
9. Anexos.....	25
9.1. ANEXO I - Introducción a la calidad del Software según el Modelo de calidad ISO/IEC 9126.....	25
9.2- ANEXO 2- Cuestionario EPFC	29
9.3- ANEXO 3- Cuestionario Gestor Proyecto	34
9.4- ANEXO 4: Estructura creada en EPFC con SPEM.....	40
10. Referencias.....	49
11. Bibliografía.....	49

1. Introducción

Este documento presenta el Proyecto Final de Carrera (PFC) desarrollado en el periodo de prácticas de la autora en la Conselleria de Infraestructuras, Territorio y Medio Ambiente de Valencia (CITMA). Durante este periodo, la proyectanda ha participado en el Proyecto MOSKitt (Modeling Software Kitt) que se está desarrollando dentro de la CITMA para la construcción de la herramienta CASE MOSKitt. Esta herramienta, es una herramienta Open Source basada en Eclipse cuyo objetivo es proporcionar soporte a la metodología gvMétrica (metodología utilizada en la CITMA para el desarrollo de Software). MOSKitt constituye a su vez una Infraestructura para el Desarrollo de Herramientas de Soporte al Desarrollo de Software dirigido por Modelos (DSDM).

En el contexto de MOSKitt, se ha desarrollado un prototipo de herramienta CAME (Computer Aided Method Engineering) para dar soporte a la construcción de métodos de producción de software (MPS). Esta herramienta es la denominada MOSKitt4ME. Este PFC se centra fundamentalmente en esta herramienta.

MOSKIITT4ME permite especificar métodos de producción de Software y mediante transformaciones entre modelos, obtener reconfiguraciones de MOSKitt que den soporte software (en forma de herramientas CASE) al método especificado. Esta herramienta CAME utiliza para la definición de MPS el estándar de la OMG SPEM. Este lenguaje permite la definición completa de MPS cubriendo todos los aspectos necesarios para su especificación, permitiendo definir diferentes roles (responsabilidades), diferentes tipos de tareas (por ej. repetitivas o cíclicas), puntos de decisión y bifurcaciones.

La definición de MPS realizada en SPEM está más orientada al proceso, donde las tareas se distribuyen entre los diferentes roles que participan en el método. A partir de la especificación SPEM de un MPS, MOSKitt4ME genera una reconfiguración de MOSKitt que da soporte a la ejecución de dicho MPS. Esto se consigue integrando un plugin adicional en la reconfiguración de MOSKitt. A este plugin que constituye la herramienta CASE de soporte al MPS, se le ha llamado Gestor de Proyectos.

La herramienta MOSKitt4ME está en fase de desarrollo y es necesario llevar a cabo una validación de la herramienta que permita completar su desarrollo. El presente PFC se centrará en esta validación donde es necesario establecer inicialmente un marco de evaluación. Este marco de evaluación se ha definido utilizando como base un conjunto de estándares ISO. Este marco contempla las dos herramientas involucradas en este entorno:

- La herramienta CAME que constituye propiamente MOSKitt4ME y
- La herramienta CASE que genera la CAME.

En base al marco de evaluación se definirán plantillas que van a servir para evaluar con usuarios reales de la CITMA ambas herramientas.

2. Objetivos

Los objetivos del proyecto son los siguientes:

- Especificar utilizando MOSKitt4ME un conjunto de procedimientos de gvMétrica utilizados en la CITMA.
- Construir un Marco de Evaluación para la validación de MOSKitt4ME:
 - Construir cuestionarios para validar la herramienta CAME.
 - Validar las herramientas CASE que genera MOSKitt4ME, haciendo especial énfasis en el Gestor de Proyectos incluido en ésta:
 - Desarrollar una serie de proyectos SW utilizando el Gestor de Proyectos como soporte a uno de los procedimientos de la CITMA definidos en el objetivo anterior.
 - Construir cuestionarios para validar la herramienta CASE.

3. Introducción a la Ingeniería de Métodos

A continuación se va a realizar una breve introducción a la Ingeniería de Métodos. Este punto va a ser tratado en base las definiciones que se encuentran en [1] .

Los términos *análisis de operaciones*, *simplificación del trabajo* e *ingeniería de métodos* se utilizan con frecuencia como sinónimos. En la mayor parte de los casos se refieren a una técnica para aumentar la producción por unidad de tiempo y, en consecuencia, reducir el costo por unidad.

En 1932, el termino "Ingeniería de Métodos" fue desarrollado y utilizado por H.B.Maynard y sus asociados, quedando definido como sigue:

"Es la técnica que somete cada operación de una determinada parte del trabajo a un delicado análisis en orden a eliminar toda operación innecesaria y en orden a encontrar el método más rápido para realizar toda operación necesaria; abarca la normalización del equipo, métodos y condiciones de trabajo; entrena al usuario a seguir el método normalizado; realizado todo lo precedente (y no antes)."

En ese momento, la Ingeniería de Métodos surgió como una forma prometedora de abordar la adaptación de los métodos de producción de software y las herramientas para especificar las necesidades del proyecto.

Tal y como se explica en [1], los proyectos de desarrollo de software pueden estar enmarcados en numerosos contextos y utilizar una amplia variedad de elementos circunstanciales para poder ser llevados a cabo. Estos elementos, especialmente los que proporcionan las personas o los factores organizacionales, tienen un gran impacto en el desarrollo del proceso. Por lo tanto, con el fin de maximizar la productividad y mejorar la calidad del desarrollo de software, el proceso de desarrollo debe de estar gobernado por un método de producción de software que sea adaptable a estas necesidades circunstanciales de las que se habla.

Con el fin de abarcar la necesidad de adaptación de los métodos, es necesario encontrar alternativas que no solo permitan la definición interna (y adaptación) de los métodos sinó también la

construcción de las correspondientes herramientas de soporte. Hasta ahora, la disciplina de la Ingeniería de Métodos Situacional parece ser la solución más prometedora para cubrir esta necesidad. Esta ingeniería estudia y propone técnicas y métodos para llevar a cabo la construcción de los métodos que están en sintonía con las situaciones específicas de los proyectos de desarrollo.

Estas técnicas permiten la creación de métodos de desarrollo adecuado para cualquier situación de desarrollo. El desarrollo de cada sistema se inicia entonces, con una fase de definición del método, donde se construye el método de desarrollo en el lugar.

La disciplina de Ingeniería de Métodos Situacional, abarca todos los aspectos en cuanto a la creación de métodos¹ para situaciones específicas y constituye una sub-área de un amplio campo llamado Ingeniería de Métodos.

La Ingeniería de Métodos se puede definir según [2] como:

“La disciplina de la ingeniería que diseña, construye y adapta métodos, técnicas y herramientas para el desarrollo de Sistemas de Información”.

Durante los últimos veinte años muchas propuestas teóricas han contribuido a establecer una base teórica amplia y sólida en este campo. Sin embargo, las herramientas existentes no están a la altura de las expectativas principalmente debido a la complejidad que conlleva poner esta base teórica en práctica.

Con el fin de mejorar esta situación, en [1] se propone el uso del paradigma de Desarrollo de Software Dirigido por Modelos como solución que permite manejar esta complejidad de forma eficiente. De este modo se debe definir un marco metodológico que hace uso de técnicas de meta-modelado y transformaciones de modelos para abordar el diseño e implantación de métodos de producción de software.

Por lo tanto, con el fin de contribuir a la mejora de la situación, se están haciendo grandes esfuerzos de estandarización (Ej: las iniciativas de SPEM [3] y la ISO/IEC 24744 [4]). Estos estándares tienen el objetivo de proveer lenguajes dedicados a la especificación del método que no presenten las deficiencias encontradas en las teorías iniciales.

4. Aplicación de MOSKIT4ME para la construcción de soporte a procesos de gvMétrica

En esta sección se describe cómo se han obtenido herramientas CASE de soporte a un conjunto de procedimientos utilizados en la CITMA utilizando MOSKitt4ME. El primer apartado introduce el contexto en el que se ha realizado el trabajo. El siguiente apartado describe como se han definido los procedimientos en MOSKitt4ME. En esta sección se introducen de forma general los lenguajes y las notaciones gráficas que incluye MOSKitt4ME (SPEM y BPMN) y que han sido utilizados para la especificación de los procedimientos de la CITMA. Por último se presenta cómo MOSKitt4ME obtiene las herramientas CASE a partir de la definición de los procedimientos.

¹ En este documento, los términos “método” y “metodología” son usados como sinónimos de método de producción de software.

4.1. Contexto

El presente PFC se ha realizado en el contexto de una beca realizada en la CITMA. La CITMA gestiona diversos proyectos, entre los que se encuentra MOSKitt. MOSKitt permite definir los procesos de desarrollo de software que se utilizan en la Conselleria. Estos procesos siguen una metodología definida también internamente como gvMétrica y es la encargada de realizar el seguimiento interno de los proyectos de desarrollo. El uso de esta metodología hace necesario la implantación de una herramienta CASE que facilite su aplicación. En el proceso de adaptación de la metodología gvMétrica a la herramienta MOSKitt surgen problemas de usabilidad, integración y en la evolución de la metodología.

Es posible abordar las limitaciones tecnológicas que se comentan en este mismo punto en cuanto a la adaptabilidad de la metodología a la herramienta, de tal forma que MOSKitt sea más que una herramienta CASE de soporte a gvMétrica. Esto se realiza dando soporte desde MOSKitt a las fases de:

- **Desarrollo de MPS:** incorporando soporte de herramienta para la especificación de MPS.
- **Ejecución de MPS:** incorporando un entorno de ejecución de proyectos (basado en roles, productos, guías, etc...).

MOSKitt4ME propone un marco metodológico para la construcción de MPSs que consta de las siguientes fases:

1. **Construcción del modelo del MPS.** En esta fase se especifica el método utilizando el lenguaje SPEM. Este modelo se puede construir desde cero o reutilizando fragmentos de otros métodos. El modelo no contiene detalles sobre tecnologías o notaciones.
2. **Configuración del MPS.** En esta fase se configura el modelo construido en la fase anterior incluyendo detalles sobre las notaciones o tecnologías que se utilizarán durante la ejecución del método.
3. **Implementación del MPS** En esta fase se obtiene una herramienta CASE que soporta el método modelado. La obtención de esta herramienta es semi-automática. La herramienta que se obtiene ofrece soporte a:
 1. Proceso, proporcionando un entorno de guía para la ejecución de proyectos.
 2. Producto, proporcionando las herramientas necesarias para la manipulación (creación, modificación, ...) de los productos involucrados en el proyecto.

4.2. Metamodelos para la definición de procesos

A continuación se describen los metamodelos de los que MOSKitt4ME hace uso para la especificación de procesos. En primer lugar se introduce SPEM y en segundo lugar BPMN.

4.2.1. SPEM 2.0

Como se ha comentado anteriormente, en MOSKitt4ME se utiliza (en el diseño del método), el estándar SPEM para la especificación del modelo del método, por ello a continuación se realiza una breve introducción al lenguaje.

SPEM (Software Process Engineering Metamodel) es un estándar de la OMG cuyo objetivo principal es proporcionar un marco formal para la definición de procesos de desarrollo de sistemas y de software así como para la definición y descripción de todos los elementos que los componen. SPEM ofrece un marco de trabajo para el modelado, documentación, presentación, gestión e intercambio de los procesos de desarrollo Software y sus componentes, dando una sintaxis y una estructura común para cada aspecto del proceso de desarrollo, incluyendo: roles, tareas, work products y guías.

La idea central de SPEM para representar procesos está basada en tres elementos básicos: **rol**, **producto de trabajo** y **tarea**. Las tareas representan el esfuerzo a hacer, los roles representan quien lo hace y los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen. La idea central subyacente es que un modelo de proceso consiste, básicamente, en decir quien (rol) realiza qué (tarea) para, a partir de unas entradas (productos de trabajo) obtener unas salidas (productos de trabajo).

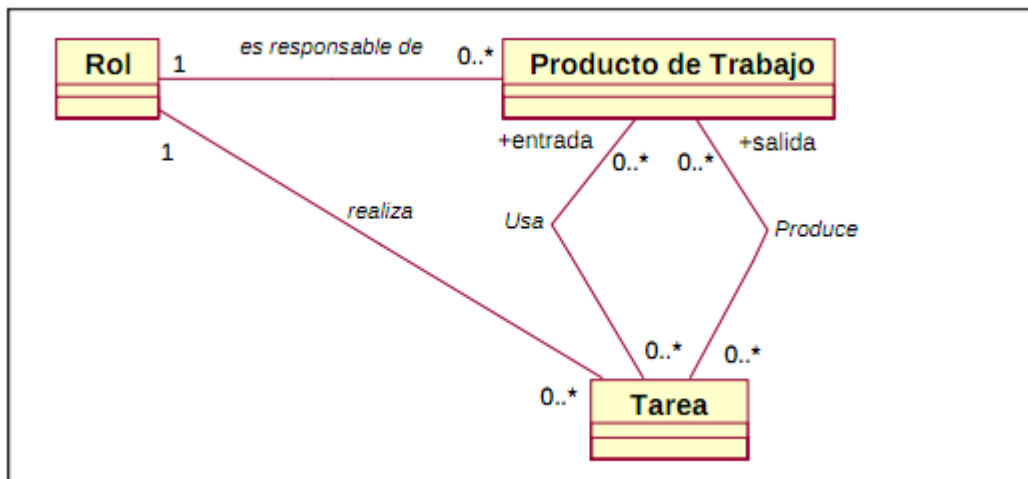


Figura 1: Idea básica de proceso en SPEM 2

Además de un metamodelo para ingeniería de procesos, SPEM también es un marco de trabajo conceptual que provee los conceptos necesarios para modelar, documentar, presentar, publicar, gestionar, intercambiar y realizar métodos y procesos software. Por ello, está destinado a ingenieros de procesos, jefes de proyectos, gestores de proyectos y programas; que son responsables de mantener e implementar procesos para sus organizaciones o para proyectos concretos.

La siguiente figura muestra un resumen del marco de trabajo general de SPEM, es decir, de los escenarios más habituales de su uso.



Figura 2: Marco de trabajo conceptual de SPEM 2.0

Tal como se deduce del marco de trabajo (ver Figura 2), en SPEM 2 se distinguen dos grupos de conceptos a la hora de implementar una metodología :

1. Primero, se puebla el **Method Content** (contenido del método) con Content Elements (elementos de contenido), es decir, los elementos primarios o constructores básicos.
2. Después, se combinan y reutilizan dichos elementos para obtener **Processes** (procesos).

La siguiente imagen muestra esta estructura:

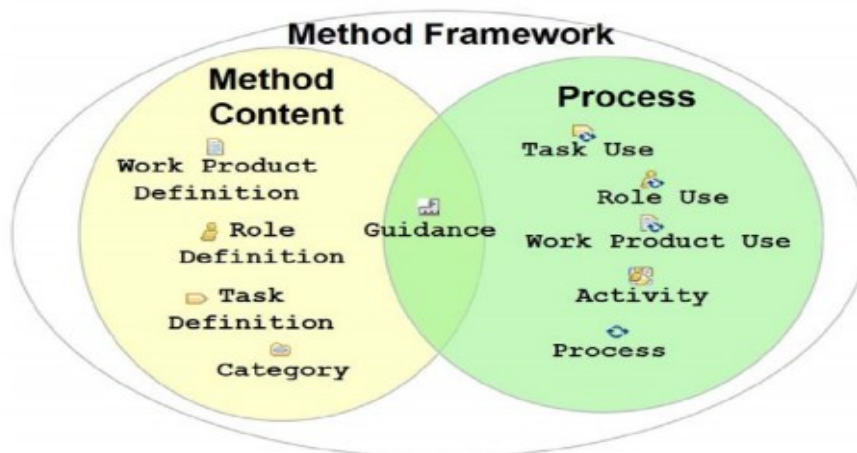


Figura 3: Aspectos principales para modelar con SPEM

- En un lado se observa el paquete **Method Content**, donde se especifican los **Métodos**. El objetivo principal de este paquete es el de definir tareas (Task Definition), organizarlas en distintos pasos (Steps), definir cuáles son los productos de entrada-salida de cada una de ellas (Work Product Definition) y especificar quién ha sido el que ha realizado dicha tarea (Role Definition).
- En la intersección de los métodos y los procesos se encuentran las guías (**Guidance**) son plantillas, ejemplos, roadmaps, etc, y dan soporte tanto a Métodos como a procesos.
- En el otro lado se encuentran los **Procesos**. El elemento principal para representar los procesos en SPEM 2.0, es la **Actividad**, que está formada por un grupo de elementos (otras actividades, task uses, etc) que están definidos dentro de un espacio de nombres y para los que se han descrito una serie de relaciones específicas para un método o proyecto.

4.2.2. BPMN 2.0

En el presente proyecto, las descripciones de Proceso realizadas en SPEM, se utilizan para construir diagramas BPMN que muestran las tareas, dependencias entre estas, productos y fases de forma gráfica, por ello, vamos a hacer una pequeña introducción a esta notación.

BPMN (Business Process Modelling Notation) es un estándar de la BPMI (Business Process Management Initiative), organismo que ha sido absorbido recientemente por la OMG² cuyo principal objetivo es “proporcionar una notación fácilmente comprensible por todos los usuarios del negocio, desde los analistas, los desarrolladores técnicos hasta aquellos que monitorizarán y gestionarán los procesos“. Se trata de un estandar internacional de modelado de procesos aceptado por la comunidad.

BPMN , es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio.

Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades.

Es importante modelar con BPMN ya que cumple las siguientes características:

- BPMN es un estándar internacional de modelado de procesos aceptado por la comunidad.
- BPMN es independiente de cualquier metodología de modelado de procesos.
- BPMN crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.
- BPMN permite modelar los procesos de una manera unificada y estandarizada
- permitiendo un entendimiento a todas las personas de una organización.

La siguiente imagen muestra un ejemplo de un proceso modelado con BPMN:

2 Object Management Group, <http://www.omg.org/>

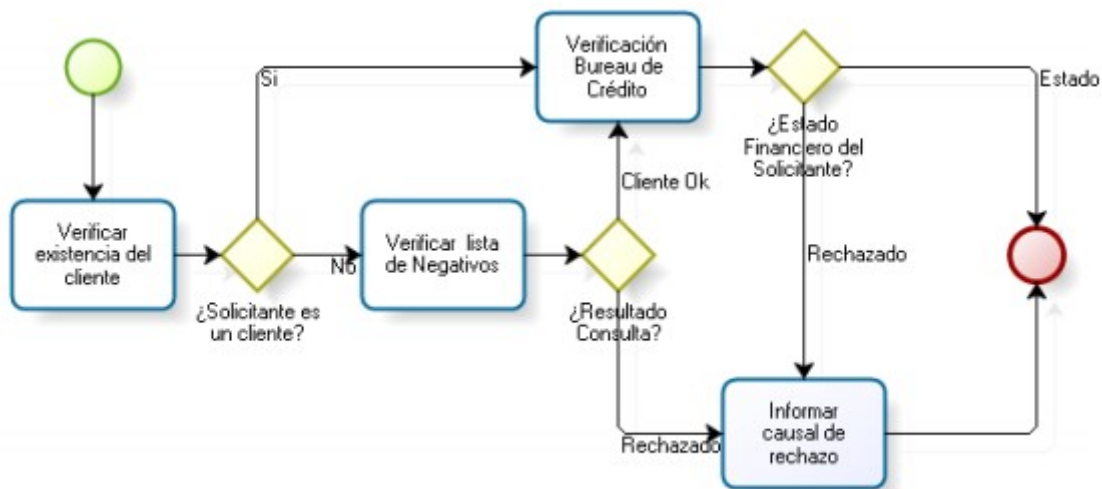


Figura 4: Ejemplo de diagrama BPMN

Los modelos BPMN se expresan gráficamente mediante diagramas BPMN. Estos diagramas constan de una serie de elementos que nos van a permitir diferenciar claramente las tres secciones (o submodelos) básicos que existen en un modelo BPMN. Estas secciones son:

- Procesos de negocio privados (internos).
- Procesos abstractos (públicos).
- Procesos de colaboración (globales).

Las características más relevantes de BPMN son las siguientes:

- BPMN es independiente de cualquier metodología de modelado de procesos.
- BPMN crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.
- BPMN permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización.

En la actualidad, es un hecho que cada día están teniendo más importancia los procesos de negocio y por extensión las herramientas que sirven para modelar, simular, supervisar y gestionar ese tipo de procesos.

4.3. Definición de Procedimientos con MOSKitt4ME

Esta sección se divide en cuatro subsecciones que contienen información acerca de los Procedimientos de la CITMA, del proceso de definición y de la generación de diagramas BPMN. En el primer subapartado se comentan las características de los procedimientos que se han escogido para el proyecto, se explican los tipos, sus características y la información mediante la cual se han

obtenido los requisitos para la definición. En el segundo, se explica cómo se ha realizado la definición de los procedimientos usando la herramienta EPFC y *Activity Designer*, ambas integradas en MOSKIt4ME. En el último subapartado, se comenta cómo se ha realizado la generación de los diagramas BPMN's asociados a cada procedimiento.

4.3.1. Características de los Procedimientos de la CITMA

En este punto se realiza una breve descripción de los procedimientos definidos. Los procesos de gvMétrica que se han seleccionado para la realización del proyecto pueden ser de dos tipos según las características del framework en el que se vayan a desarrollar, herramientas utilizadas o código final generado. En base a sus características, los procedimientos pueden ser los siguientes:

- Nuevas Aplicaciones (NAPL)
 - Nuevas Aplicaciones en gvHidra
 - Nuevas Aplicaciones en gvNix
- Mantenimiento de aplicaciones (MANT)
- Procedimiento de Organización

Las **Nuevas Aplicaciones (NAPL)** de la CITMA, son aplicaciones nuevas que parten desde cero. Se trata de proyectos nuevos generales, en los que no se conoce inicialmente el framework en el que serán desarrolladas en su fase de construcción. Para estos tipos de proyectos, existen unas plantillas definidas en el entorno de la Conselleria y con la que se pretende definir las tareas, subtareas e hitos que se deben de realizar durante el desarrollo de un proyecto de estas características. Dichas plantillas son internas de la CITMA y son accesibles únicamente por los Jefes de Proyecto, Jefes de Sección y Analistas de los Proyectos.

Esta plantilla se muestra en la siguiente imagen:

Estado de planificación	Código	Nombre
	ORDER00096-00001	Proyecto X
	ORDER00096-00002	(H) CGPI-APRALC: Aceptación del Alcance por el Responsable de Seguimiento
	ORDER00096-00003	Versión 0.1.0
	ORDER00096-00004	CASI-Análisis del Sistema de Información
	ORDER00096-00021	CDSI: Diseño del Sistema de Información
	ORDER00096-00039	CCSI: Construcción del Sistema de Información
	ORDER00096-00065	CIAS: Implantación del Sistema de Información

Figura 5: Plantilla de la CITMA para las Nuevas aplicaciones

Este tipo de proyectos consta de dos fases:

- Fase de Gestión de Proyecto (GP)
- Fase de Desarrollo del Proyecto (Desarrollo)

La siguiente imagen muestra las fases de un proyecto con estas características:

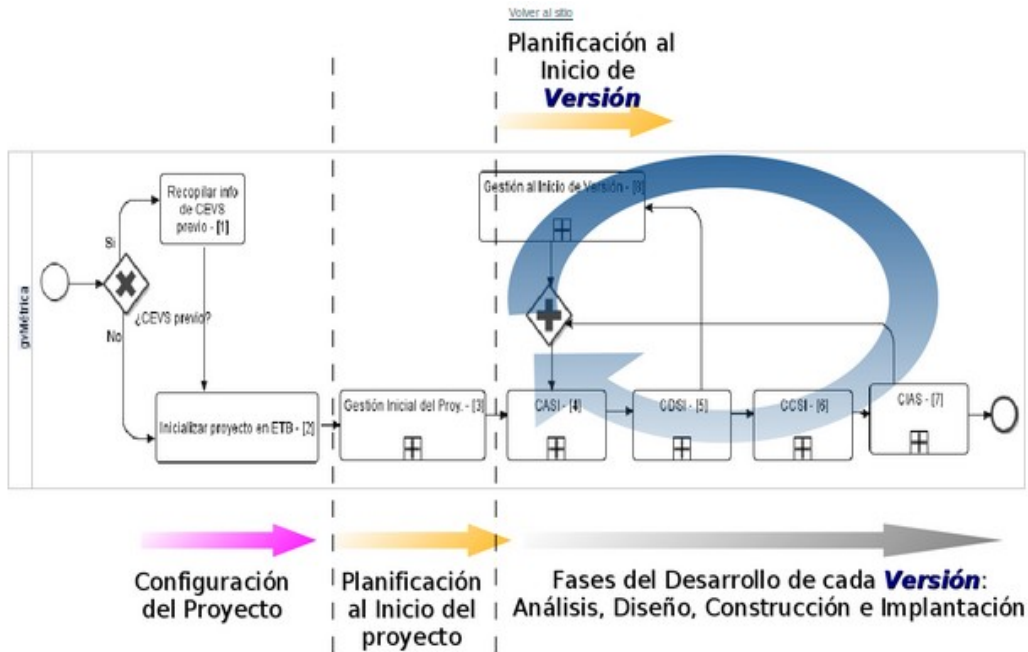


Figura 6: Fases de un Proyecto NAPL en CITMA

Dentro del grupo de las Nuevas Aplicaciones se pueden dar varios casos según el framework en el que se vaya a desarrollar la aplicación:

- *Nuevas aplicaciones plataforma gvHidra.* Se trata de nuevas aplicaciones que desde su comienzo se sabe que van a ser desarrolladas en el framework de la CITMA gvHidra³. Este tipo de aplicaciones son muy similares a las NAPL's generales pero en las fases de construcción el código resultante será código gvHidra.

A continuación se muestra el ciclo de vida de este tipo de aplicaciones:

³ gvHidra, <http://www.gvpontis.gva.es/cast/gvhidra-herramienta/>

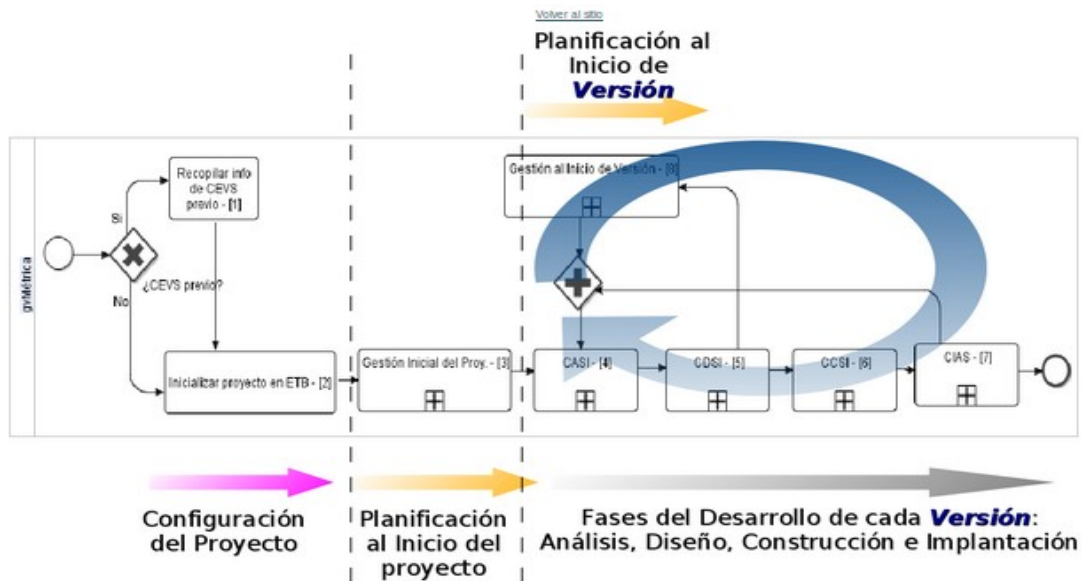


Figura 7: Fases de un Proyecto NAPL-gvHIDRA en CITMA

- **Nuevas aplicaciones plataforma gvNix.** Se trata de Nuevas Aplicaciones que desde su comienzo se sabe que van a ser desarrolladas en el framework gvNix⁴. Es muy similar al NAPL general pero en las fases de construcción el código resultante será código gvNix. La siguiente figura muestra el ciclo de vida de los proyectos de este tipo:

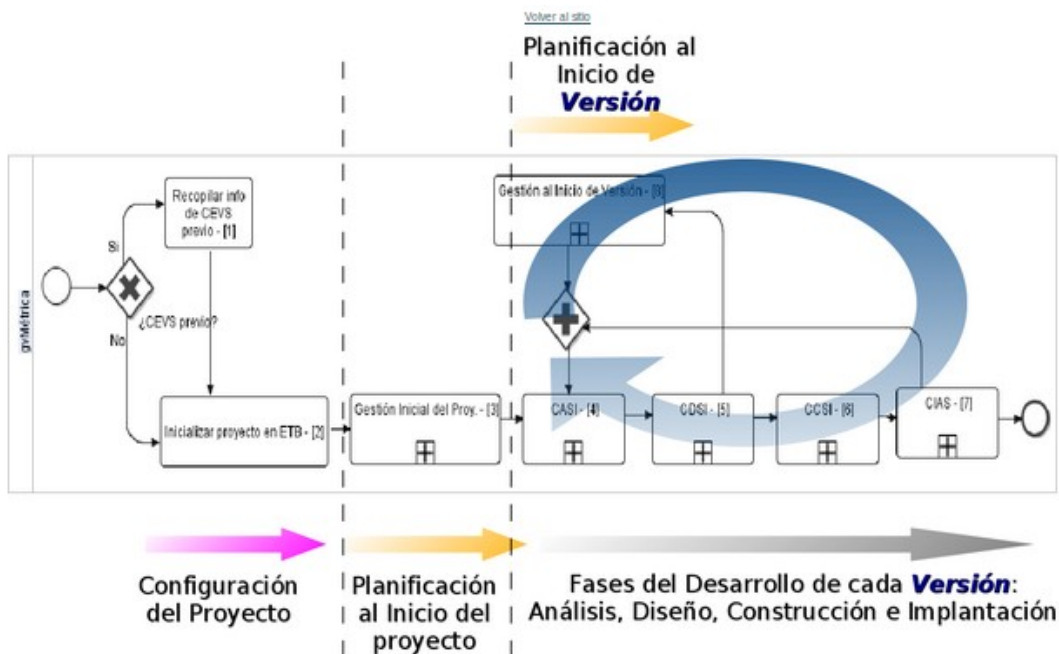


Figura 8: Fases de un Proyecto NAPL-gvNIX en CITMA

⁴ gvNix, <http://www.gvpontis.gva.es/cast/gvnix/>

Para ambos subtipos de procedimientos, todavía no se han especificado plantillas en la CITMA, por lo que se utilizan de forma provisional, las de las Nuevas Aplicaciones generales. (Ver Figura 5).

Mantenimiento de Aplicaciones (MANT). Este tipo de proyectos son mantenimientos de aplicaciones que ya existen en la CITMA. Se trata de proyectos que ya han sido desarrollados con anterioridad y en un momento determinado es necesario realizar ampliaciones, modificar requisitos o reelaborar partes del código o funcionalidades.

Para este tipo de proyectos, también existen plantillas definidas en la CITMA, que contienen las tareas en sus fases de desarrollo. Esta plantilla es la siguiente:

Estado de planificación	Código	Nombre	H
[-] C	ORDER00102-00001	Proyecto X	0
[-] C	ORDER00102-00003	Versión 0.1.0	0
[+] C	ORDER00102-00004	CASI-Análisis del Sistema de Información	0
[+] C	ORDER00102-00021	CDSI: Diseño del Sistema de Información	0
[+] C	ORDER00102-00039	CCSI: Construcción del Sistema de Información	0
[+] C	ORDER00102-00065	CIAS: Implantación del Sistema de Información	0
[+] C	Código nuevo	RESOLINCID	0

Figura 9: Plantillas de la CITMA para los MANT

Existe para este tipo de proyectos un diagrama BPMN, que define las principales tareas, dependencias y documentos que componen el procedimiento. La siguiente imagen muestra un BPMN del procedimiento:

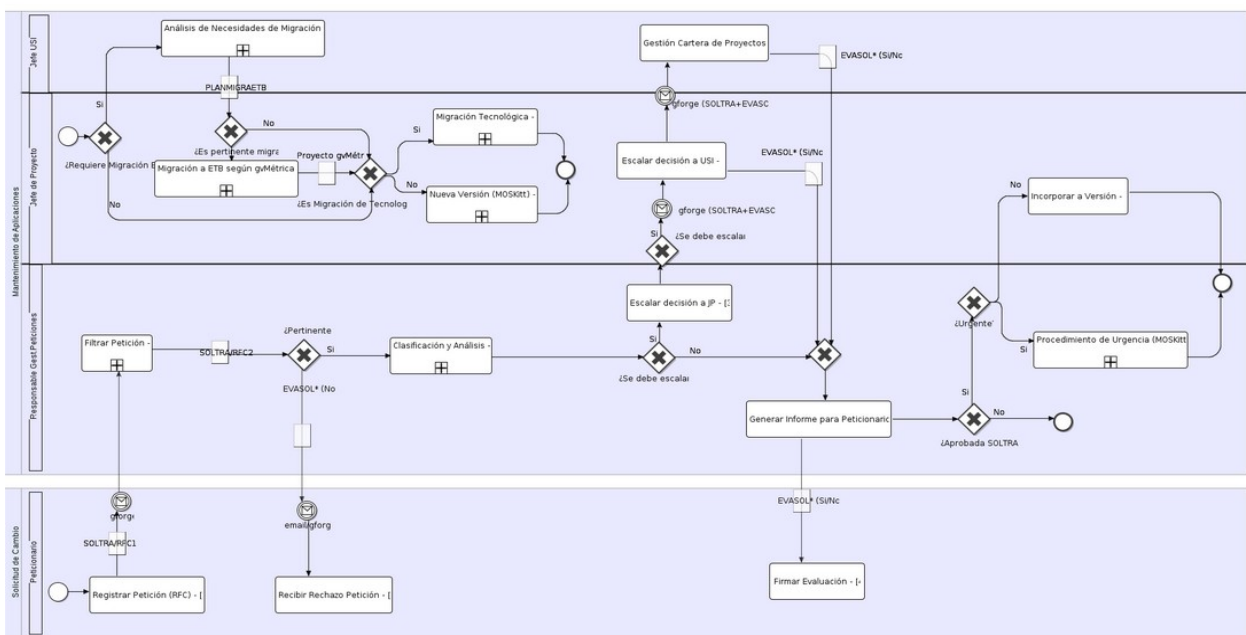


Figura 10: Procedimiento MANT en la CITMA

Procedimientos de Organización. Se trata de un procedimiento algo distinto, ya que no lo ejecutan los Analistas Informáticos, sino los Analistas de Organización de la CITMA. Es necesario definirlo porque forma parte de las NAPL's pero no es un procedimiento en sí si no que se ejecuta dentro de una de las tareas de las Nuevas Aplicaciones y no en todos los casos. Por tanto, este procedimiento no va a ser explicado en esta memoria.

4.3.2. Definición de los procedimientos en EPF Composer

MOSKitt4ME tiene integradas dos herramientas que permiten la especificación completa de un método o proceso. Por un lado posee la herramienta *Eclipse Process Framework Composer*, (EPFC) para la descripción de los fragmentos de métodos o procesos y por otro lado la herramienta *Activity Designer* para la especificación del proceso.

El 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 XMI y, al estar basados en el estándar SPEM 2, pueden ser reutilizados por cada vez más herramientas CASE. En suma, EPFC es un editor de procesos SPEM 2, que incluye opciones adicionales para publicar de forma automáticamente sitios web.

Para la creación de los procedimientos en EPFC, se han utilizado las plantillas descritas en el apartado anterior, tomando como base las tareas y dependencias que allí aparecen.

A continuación se describe brevemente la estructura de los modelos en SPEM.

Los elementos de contenido en SPEM se pueden organizar y jerarquizar mediante la creación de **paquetes de contenido**. En cada paquete de contenido se pueden definir los elementos de contenido de método básicos: **roles, tareas, productos de trabajo, ejecutables o guías** (instrucciones).

Se ha creído necesaria la creación de tres paquetes de contenido (Figura 11). Estos paquetes hacen referencia a las dos fases que forman cualquier proyecto de desarrollo en la CITMA (Gestión de Proyecto y Desarrollo) y otro que contenga la información de los roles definidos en esta.

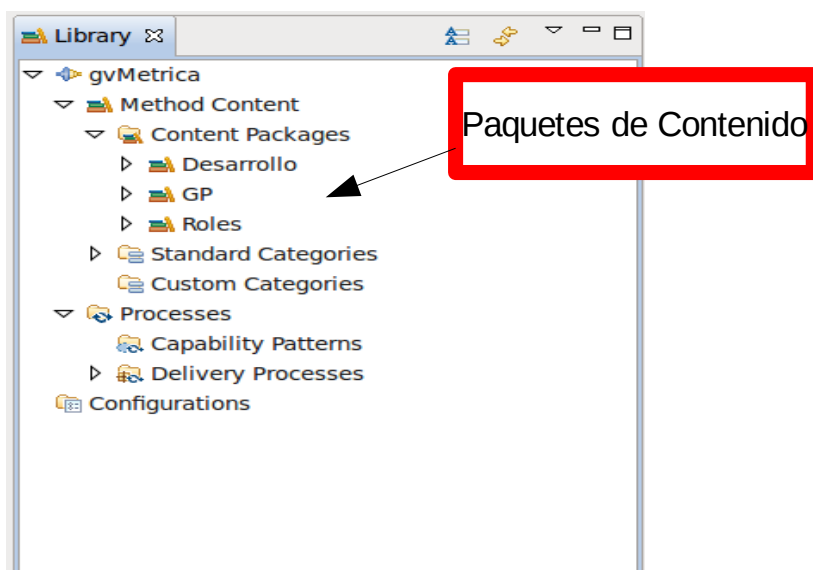


Figura 11: Paquetes de contenido creados en EPFC

Cada uno de los paquetes de contenido tiene la siguiente estructura:

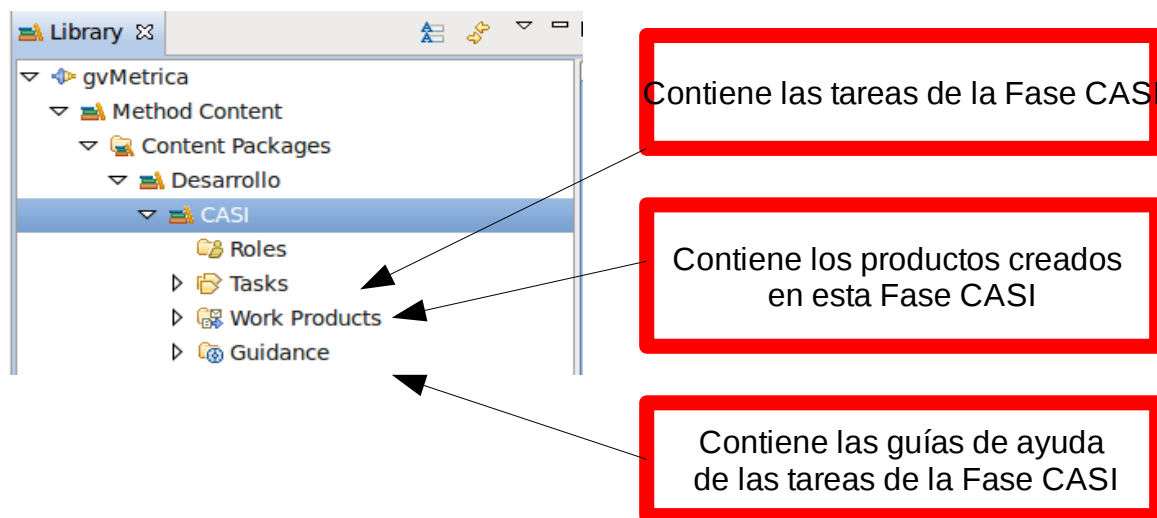


Figura 12: Ejemplo de la estructura de los modelos en SPEM (FASE CASI)

Es importante resaltar que el orden de edición en cada tarea ha sido el siguiente:

1. Crear los productos que se van a necesitar tanto si son de entrada o de salida. Esto se realiza en la carpeta Work Products del elemento el paquete.
2. Crear las guías que se van a necesitar para la tarea. Esto se realiza en la carpeta Guidance del elemento el paquete.
3. Crear la Tarea en la carpeta Tasks.

Cabe comentar que en el ANEXO 4 existe una descripción más detallada del proceso que se ha seguido en la creación del contenido de cada uno de los paquetes creados.

Una vez se han definido los elementos de contenido de método (tal y como se explica en el ANEXO 4) éstos se pueden utilizar para crear procesos, en este caso los cuatro procesos de la CITMA escogidos. Cabe señalar que no es necesario haber creado elementos de contenido para utilizarlos en la creación de procesos, pues se pueden definir directamente en un proceso, pero esta opción dificulta mucho la reutilización y actualización posterior de dichos elementos y de sus procesos.

Se han definido cuatro procesos, en base a los dos tipos de procedimientos que se debían definir como se muestra en la siguiente imagen:

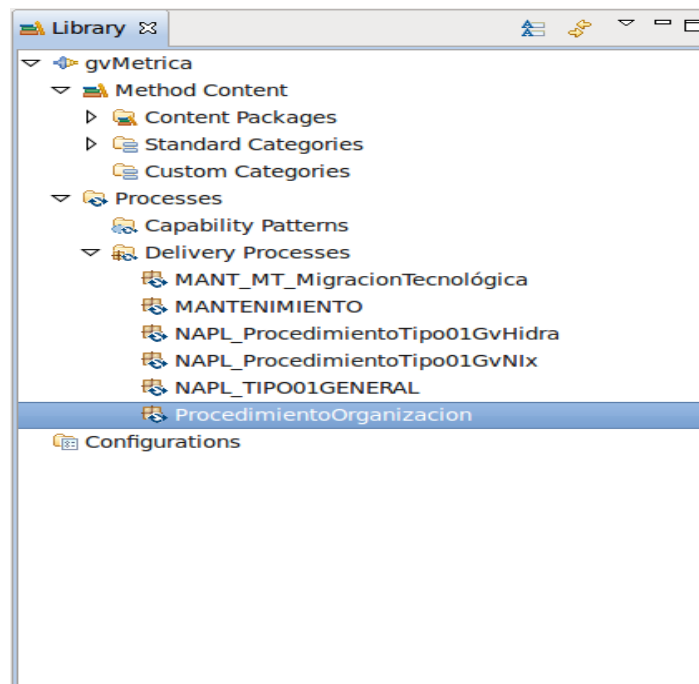


Figura 13: Contenido de la carpeta Processes

Se va a tomar como ejemplo para mostrar el resultado de la definición de un proceso en SPEM, el procedimiento de Nuevas Aplicaciones general (NAPL_TIPO01GENERAL, Ver Figura 14).

Los pasos que se han seguido en la creación de cada proceso ha sido:

1. Crear un elemento **Activity** por cada Fase (contiene subtareas).
2. Crear elementos **Task Descriptor** para cada tarea ejecutable (son subtareas que se ejecutan).
3. Crear dependencias (Columna **Predecessors**) entre las tareas de cada fase (deben ser tareas “hoja”).
4. Especificar las características (si es necesario) en las tareas. En este paso, se seleccionan las casillas de las columnas que se refieran a repetibilidad de una tarea (**Repeatable**), Opcionalidad (**Optional**), etc...de forma que queden reflejados algunos requisitos en los procedimientos.

Presentation Name	Inde:	Prede:	Type	Plannec	Repeatat	Multiple	Ong
NAPL: Procedimiento de tipo 01	0		Delivery Pr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CEVS- Recopilar Información sobre el Estudio de Viabilidad del SI	1		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Inicialización del Proyecto	5		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI01: Inicio del Proyecto	9	5	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0201: Planificación del Proyecto	13	9	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación del Alcance del Proyecto	15	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	20	15	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC03: Configuración del Entorno de Trabajo	23	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0103: Inicio de Versión	25	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0202: Planificación de la nueva Versión	29	25	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación al Inicio de la Versión	31	29	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Soporte a la Nueva Versión	35	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	37	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CASI: Análisis del Sistema de Información	40	23,31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CDSI: Diseño del Sistema de Información	75	40	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CCSI: Construcción del Sistema de Información	100	75	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CIAS: Implantación del Sistema de Información	121	100	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Actualizar el estado del Proyecto - Diario	139	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Seguimiento y Control del Proyecto - Semanal	141		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 14: Tareas en el procedimiento NAPL general

A modo de ejemplo, la siguiente imagen muestra la estructura a nivel más interno de la Fase CASI:

Presentation Name	Index	Predecessors	Type	Planned
CASI: Análisis del Sistema de Información	40	23,31	Activity	<input checked="" type="checkbox"/>
CASI01-REQ-NF: Establecimiento de Requisitos No Funcionales	41		Activity	<input checked="" type="checkbox"/>
CASI0101-REQ-SOLPROC: Solicitud de definición del Procedimiento	42		Task Descri	<input type="checkbox"/>
(H) APRPRO: Aceptación de la Definición del Procedimiento	43	44	Task Descri	<input type="checkbox"/>
CASI0102-REQ-DTECHENV: Definición Entorno Tecnológico	44		Activity	<input checked="" type="checkbox"/>
Revisar el ET Base para el Proyecto	45		Task Descri	<input type="checkbox"/>
Completar Entorno Tecnológico	46	45	Task Descri	<input type="checkbox"/>
Definición del Catálogo de Requisitos No Funcionales	47		Task Descri	<input type="checkbox"/>
Realizar el Catálogo de Normas	48		Task Descri	<input type="checkbox"/>
Identificar Catálogo Inicial de Requisitos Funcionales	49		Task Descri	<input type="checkbox"/>
CASI02-REQ-PREV-FUNC: Análisis y Tratamiento de Información	50		Activity	<input checked="" type="checkbox"/>
Generar Modelo UML2	51		Task Descri	<input type="checkbox"/>
CASI03-REQ-FUNC: Análisis Funcional	52		Activity	<input checked="" type="checkbox"/>
CASI0301-INTUSU: Análisis de la Interfaz de Usuario	53		Activity	<input checked="" type="checkbox"/>
CASI0302-LNEG: Análisis de la lógica del negocio	59		Activity	<input checked="" type="checkbox"/>
CASI0303-PDAT: Análisis de la Persistencia de los Datos	66		Activity	<input checked="" type="checkbox"/>
CASI0304-PLANPRU: Definición del Plan de Pruebas	70		Activity	<input checked="" type="checkbox"/>
CASI0305-VAL: Validación del Análisis	72	70,53,59,66	Activity	<input checked="" type="checkbox"/>
(H) APREQ: Aprobación de la definición de Requisitos y Funciones	74		Task Descri	<input type="checkbox"/>
CDSI: Diseño del Sistema de Información	75	40	Activity	<input checked="" type="checkbox"/>
CCSI: Construcción del Sistema de Información	100	75	Activity	<input checked="" type="checkbox"/>

Figura 15: Estructura detallada de la fase CASI

4.3.3. Generación de diagramas BPMN desde el EPFC

En este proyecto, los diagramas BPMN cumplen una labor primordial, ya que serán el resultado de la definición en base a los procedimientos definidos en SPEM con el EPF Composer. Es importante indicar que ha sido necesario la extensión en MOSKitt para poder generar diagramas BPMN's a partir de los modelos SPEM .

La forma de generar diagramas a partir de cada procedimiento es la siguiente:

1. Seleccionamos uno de los procedimientos definidos en SPEM.
2. Pulsamos sobre el elemento raíz (este elemento será el primero que se ejecute)
3. De este modo, comienza la generación de diagramas BPMN's en base a las fases, tareas, roles, recursos y dependencias definidas en los procedimientos.
4. Como resultado, se crea una carpeta nueva llamada **Activiti** que contiene todos los diagramas generados junto con las imagenes. En la siguiente imagen podemos ver una de las carpetas generadas en el directorio del procedimiento NAPL_TIPO01GENERAL:

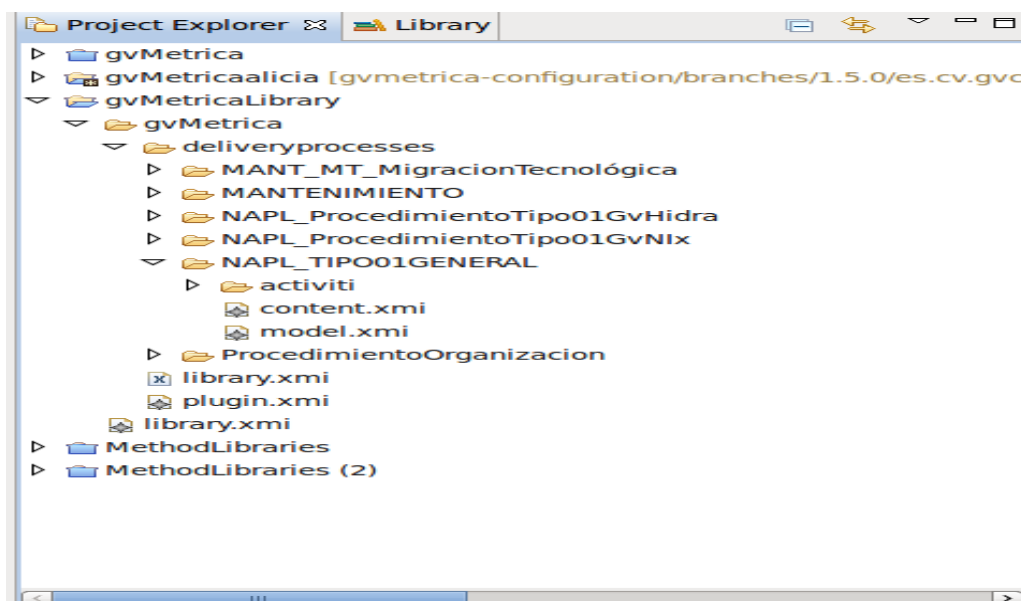


Figura 16: Carpeta activii creada para el proceso NAPL_TIPO01GENERAL

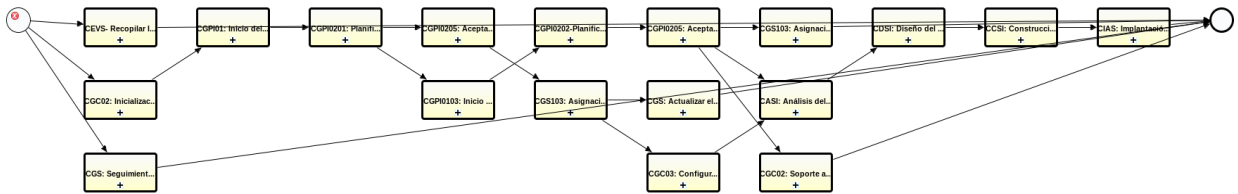
Es importante señalar que la creación de esta carpeta es visible desde al vista *Project Explorer* de MOSKitt.

A continuación se muestra cada uno de los procedimientos en SPEM y el BPMN generado a partir de el:

• **Procedimiento NAPL_TIPO01GENERAL en SPEM:**

Presentation Name	Inde:	Prede	Type	Plannc	Repeat	Multiple	Ong
NAPL: Procedimiento de tipo 01	0		Delivery Pr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CEVS- Recopilar Información sobre el Estudio de Viabilidad del S	1		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Inicialización del Proyecto	5		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI01: Inicio del Proyecto	9	5	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0201: Planificación del Proyecto	13	9	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación del Alcance del Proyecto	15	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	20	15	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC03: Configuración del Entorno de Trabajo	23	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0103: Inicio de Versión	25	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0202-Planificación de la nueva Versión	29	25	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación al Inicio de la Versión	31	29	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Soporte a la Nueva Versión	35	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	37	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CASI: Análisis del Sistema de Información	40	23,31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CDSI: Diseño del Sistema de Información	75	40	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CCSI: Construcción del Sistema de Información	100	75	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CIAS: Implantación del Sistema de Información	121	100	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Actualizar el estado del Proyecto - Diario	139	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Seguimiento y Control del Proyecto - Semanal	141		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

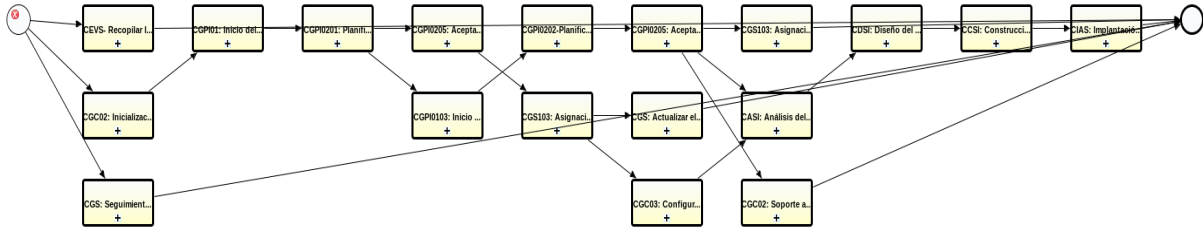
• **Procedimiento NAPL_TIPO01GENERAL en BPMN**



• **Procedimiento NAPL_ProcedimientoTipo01GvHidra en SPEM**

Presentation Name	Index	Predeceso	Mod	Type	Plannc	Repeatable	Multiple	Ongoing	Event-Driven	Option
NAPL: Procedimiento de tipo 01 GvHidra	0			Delivery Pr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CEVS- Recopilar Información sobre el Estudio de Viabilidad del Sistema	1			Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Inicialización del Proyecto	5			Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI01: Inicio del Proyecto	9	5		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0201: Planificación del Proyecto	13	9		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación del Alcance del Proyecto	15	13		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	20	15		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC03: Configuración del Entorno de Trabajo	23	20		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0103: Inicio de Versión	25	13		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0202-Planificación de la nueva Versión	29	25		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación al Inicio de la Versión	31	29		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Soporte a la Nueva Versión	35	31		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	37	31		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CASI: Análisis del Sistema de Información	40	23,31		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CDSI: Diseño del Sistema de Información	75	40		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CCSI: Construcción del Sistema de Información	100	75		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CIAS: Implantación del Sistema de Información	121	100		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Actualizar el estado del Proyecto - Diario	139	20		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Seguimiento y Control del Proyecto - Semanal	141			Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

• **Procedimiento NAPL_ProcedimientoTipo01GvHidra en BPMN**

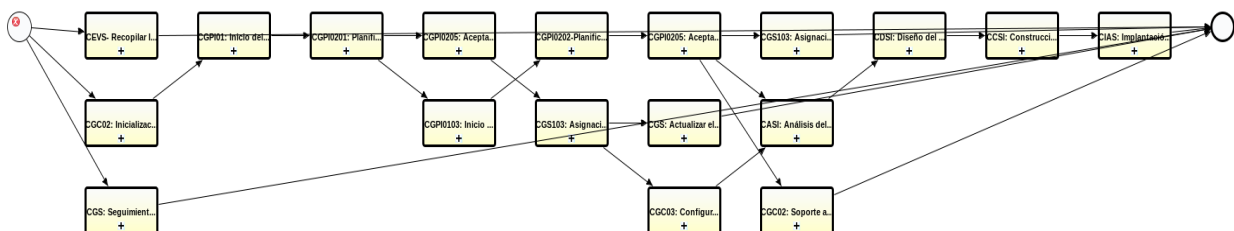


• **Procedimiento NAPL_ProcedimientoTipo01GvNix en SPEM**

Presentation Name	Inde:	Predecessors	Type	Plannec	Repeat:	Multiple	Ongoing	Event-D	Optional
NAPL_ProcedimientoTipo01GvNix	0		Delivery Pr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CEVS: Recopilar Información sobre el Estudio de Viabilidad del Sistema	1		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Inicialización del Proyecto	5		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI01: Inicio del Proyecto	9	5	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0201: Planificación del Proyecto	13	9	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación del Alcance del Proyecto	15	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	20	15	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC03: Configuración del Entorno de Trabajo	23	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0103: Inicio de Versión	25	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0202-Planificación de la nueva Versión	29	25	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGPI0205: Aceptación al Inicio de la Versión	31	29	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGC02: Soporte a la Nueva Versión	35	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS103: Asignación de Tareas y Comunicación al Equipo	37	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CASI: Análisis del Sistema de Información	40	23,31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CDSI: Diseño del Sistema de Información	74	40	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CCSI: Construcción del Sistema de Información	96	74	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CIAS: Implantación del Sistema de Información	116	96	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Actualizar el estado del Proyecto - Diario	135	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CGS: Seguimiento y Control del Proyecto - Semanal	137		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

• **Procedimiento NAPL_ProcedimientoTipo01GvNix en BPMN**

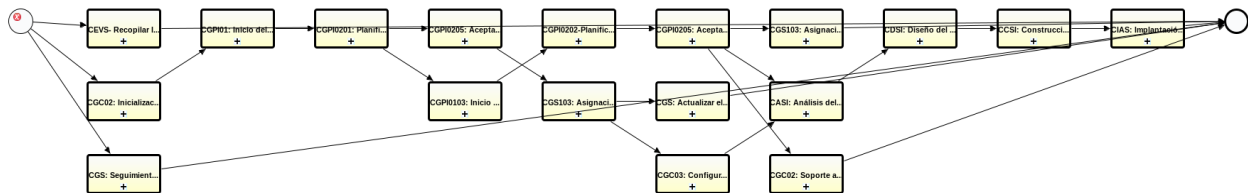


- **Procedimiento Mantenimiento en SPEM**

Presentation Name	Inde:	Predecessors	Type	Plannec	Repeatz	Multiple	Ongoing	Event-D	Optional
▼ NAPL_ProcedimientoTipo01GvNix	0		Delivery Pr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CEVS- Recopilar Información sobre el Estudio de Viabilidad del Siste	1		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGC02: Inicialización del Proyecto	5		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI01: Inicio del Proyecto	9	5	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI0201: Planificación del Proyecto	13	9	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI0205: Aceptación del Alcance del Proyecto	15	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGS103: Asignación de Tareas y Comunicación al Equipo	20	15	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGC03: Configuración del Entorno de Trabajo	23	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI0103: Inicio de Versión	25	13	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI0202-Planificación de la nueva Versión	29	25	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGPI0205: Aceptación al Inicio de la Versión	31	29	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGC02: Soporte a la Nueva Versión	35	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGS103: Asignación de Tareas y Comunicación al Equipo	37	31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CASI: Análisis del Sistema de Información	40	23,31	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CDSI: Diseño del Sistema de Información	74	40	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CCSI: Construcción del Sistema de Información	96	74	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CIAS: Implantación del Sistema de Información	116	96	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGS: Actualizar el estado del Proyecto - Diario	135	20	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ CGS: Seguimiento y Control del Proyecto - Semanal	137		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

- **Procedimiento Mantenimiento en BPMN**



4.4. Construcción de herramientas CASE con MOSKitt4ME

Una vez especificados los modelos de los procedimientos con el EPFC y el *Activity Designer*, se han generado automáticamente con MOSKitt4ME las herramientas CASE de soporte a estos procedimientos. Al entorno de la herramienta CASE generada con MOSKitt4ME para la ejecución de instancias de estos procedimientos se le denomina Gestor de Proyectos. En la figura 17 se muestra el resultado obtenido en la ejecución con el Gestor de Proyectos en la plataforma MOSKitt.

Como podemos observar, el asistente de creación de un proyecto de tipo gvMétrica en MOSKitt, nos permite seleccionar la creación del Proyecto (Ver Figura 17) en base a la antigua metodología (Dashboard) o haciendo uso del Gestor de Proyectos incorporado (SPEM Process) para cada tipo de Proyecto.

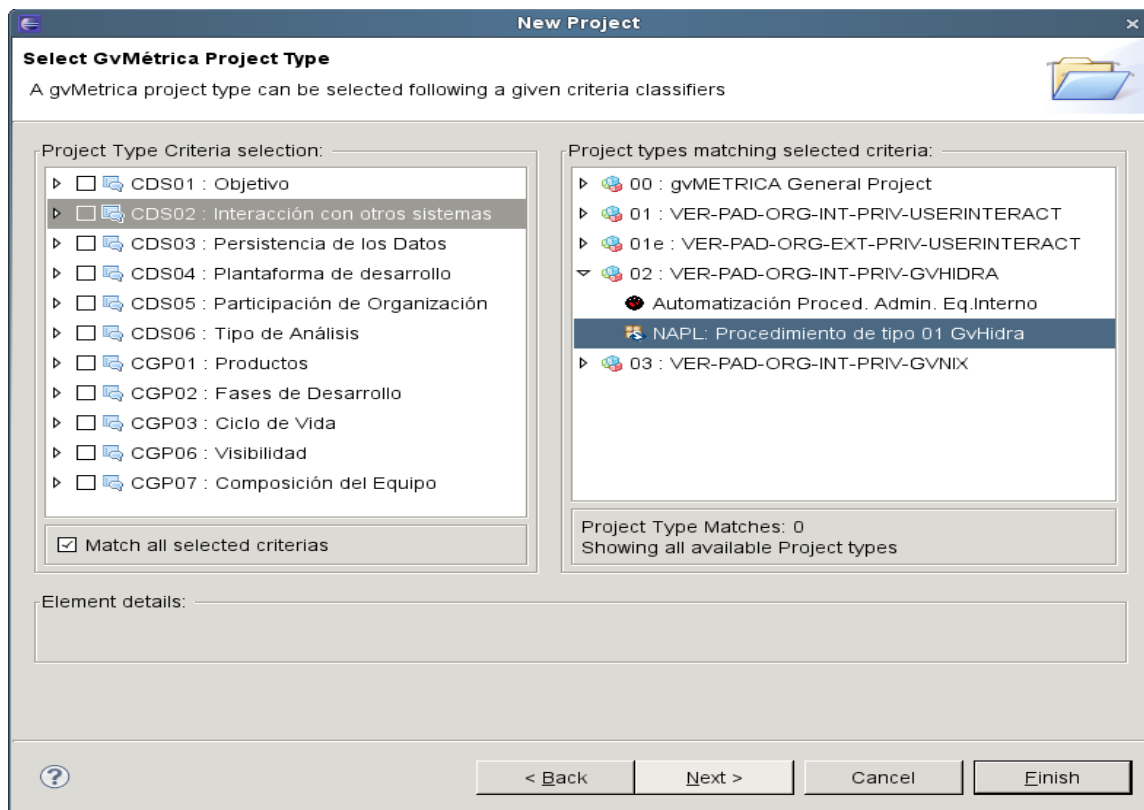


Figura 17: Asistente en la creación de un Proyecto tipo gvMétrica en MOSKitt, en concreto NAPL-gvHidra.

A modo de ejemplo, vamos a tomar como referencia para observar la ejecución del Gestor en MOSKitt, un proyecto de tipo NAPL-gvHidra. La siguiente imagen muestra el Gestor en la vista *Method Execution* de MOSKitt:

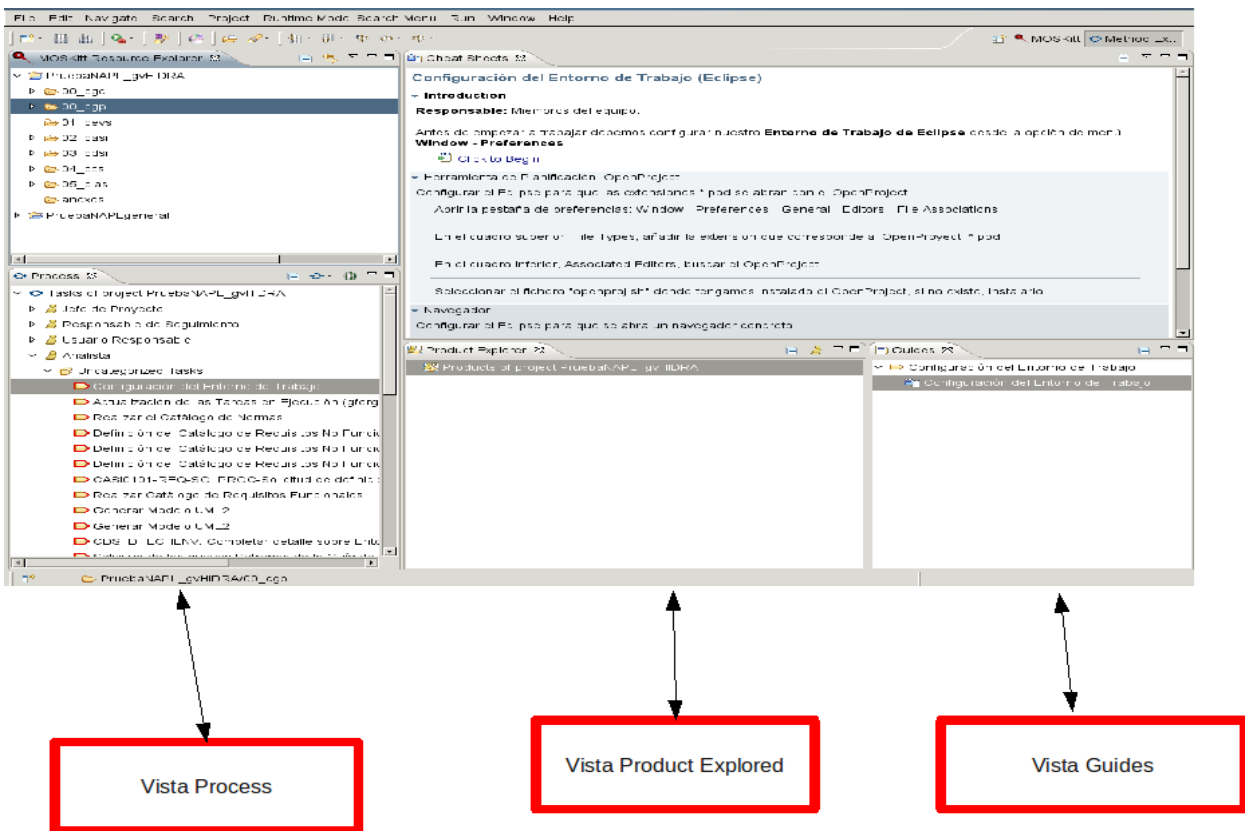


Figura 18: Vistas presentes en el Gestor de Proyectos sobre MOSKitt

Cada una de las vistas del Gestor se comentan a continuación:

El Gestor de Proyectos dota a la herramienta CASE generada de una interfaz gráfica compuesta por una serie de vistas en Eclipse. Cada una de estas vistas proporcionan funcionalidad específica pero su objetivo principal es facilitar la participación del usuario en un proyecto específico.

Los detalles de estas vistas son:

- **Product Explorer:** Esta vista muestra el conjunto de productos que son usados (consumidos, finalizados y/o producidos) en el transcurso y en las tareas finalizadas del proceso. Esta vista puede ser filtrada por roles para que los usuarios que sigan un rol específico tengan solo acceso a los productos que gestionan ellos.
- **Process:** Esta vista muestra que tareas pueden ser ejecutadas dentro del estado actual del Proyecto. La ejecución de las tareas se puede llevar a cabo de forma automática (lanzando la transformación asociada a la tarea) o manualmente por el ingeniero de Software (por medio de una herramienta software asociada al producto de salida de la tarea). De forma similar a la vista Product Explorer, esta vista también puede ser filtrada por rol, viendo solo las tareas en las que el rol está involucrado.

- **Guides:** Esta vista muestra la lista de guías asociadas a la tarea seleccionada en la vista Process. El objetivo de estas guías es asistir al usuario durante la ejecución de dicha tarea, proporcionando algunos detalles sobre cómo deberían manipularse los productos que fueron asociados en la fase de configuración del método.

5. Validación de herramientas CAME y CASE

Para completar el proceso de desarrollo de todo software es necesario llevar a cabo una etapa de validación que evalúe la calidad del software desarrollado. Una de las técnicas que se puede utilizar para esta validación consiste en definir cuestionarios que permitan capturar las bondades o defectos del software. Estos cuestionarios son completados por los usuarios finales de la herramienta. Para construir estos cuestionarios se ha definido un marco de evaluación atendiendo a los siguientes estándares:

- **ISO 9001:** estándar que describe el sistema de calidad utilizado para mantener el desarrollo de un producto que implique diseño.
- **ISO 9000-3:** documento específico que interpreta el ISO 9001 para el desarrollador de software.
- **ISO 9004-2:** documento que proporciona las directrices para el servicio de facilidades del software como soporte de usuarios.
- **ISO/IEC 9126:** documento desarrollado en un intento de identificar los atributos clave de calidad para el software. La norma ISO/IEC 9126 está enfocada a la calidad de Producto. El modelo estructura los atributos de calidad de software en seis características (funcionalidad, fiabilidad, utilidad, eficacia, capacidad de mantenimiento y portabilidad) que se subdividen en subcaracterísticas.

Por el tipo de características que se desean evaluar en las herramientas CAME y CASE, se ha decidido adoptar el estándar ISO/IEC para la construcción del marco de evaluación de las herramientas CAME y CASE.

Se han realizado dos cuestionarios con el fin de evaluar la herramienta CAME y CASE. Los cuestionarios se han realizado en base al estándar ISO/IEC 9126 [Ver ANEXO I] con el fin de obtener información sobre los aspectos más relevantes del modelado con SPEM usando el EPFC y del uso de los procedimientos modelados utilizando el Gestor de Proyectos integrado en MOSKitt.

En los anexos 2 y 3 [VER ANEXO 2 y 3] se encuentran los cuestionarios creados. En dichos cuestionarios se añaden también las instrucciones para completarlos, de forma que facilite el trabajo.

Inicialmente, el objetivo de la creación de los cuestionarios era la de evaluar la herramienta EPFC y obtener en base a los resultados obtenidos, una valoración de los principales funcionalidades, requisitos, carencias, debilidades y mejoras fundada en las respuestas de los cuestionados pero finalmente el experimento se realizará más adelante con usuarios tanto expertos como no en el

Alicia Jaén Bielsa

modelado de procesos.

Por lo tanto, no se han podido obtener valoraciones en cuanto a la herramienta EPFC ni por lo tanto del Gestor de Proyectos.

6. Conclusiones generales

El objetivo principal del proyecto ha sido evaluar: (1) la herramienta EPFC integrada en MOSKitt para modelar un conjunto de procesos internos de la CITMA y (2) la herramienta CASE generada con MOSKitt4ME para el soporte a estos procesos, el Gestor de Proyectos. A lo largo de todo el proceso de desarrollo, la autora ha necesitado documentarse acerca de las fases, subfases, dependencias y demás información relevante para la definición de los procedimientos. Esta labor se ha llevado a cabo utilizando plantillas existentes en la CITMA además de diagramas BPMN's que contienen información más concreta de los procedimientos.

En cuanto al uso de MOSKitt4ME, al tratarse de un prototipo vertical y desarrollado en la plataforma MOSKitt, no ha resultado complicado el uso de la herramienta pues la autora es usuario habitual del Software. MOSKitt4ME tiene un triple beneficio: (1) ayuda a evaluar la arquitectura propuesta, (2) marca las bases para el desarrollo del entorno de una CAME completa y (3) permite a los usuarios de la comunidad MOSKitt usar el prototipo para proveer un "feedback" que puede ser usado para el refinado de la arquitectura y el marco metodológico.

Por otro lado, la autora ha necesitado conocer a fondo el funcionamiento del Gestor de Proyectos y para ello se ha basado en [1] que introduce las nuevas funcionalidades que aporta el Gestor de Proyectos a MOSKitt.

La autora ha realizado la definición de los procedimientos, ha generado los diagramas BPMN y ha utilizado el Gestor de Proyectos integrado en MOSKitt para la ejecución de proyectos reales. En opinión de la proyectanda, este Gestor aporta a MOSKitt una nueva forma de seguir la metodología gvmétrica, de forma que permite visualizar los recursos, tareas y guías que se necesitan en cada momento. Por otro lado, permite a los usuarios (Analistas, Jefes de Proyecto, etc...) poder escoger la forma de trabajo que más les interese, haciendo uso de Dashboard como herramienta de trabajo o del Gestor, aportando una visión más enfocada al proceso.

7. Agradecimientos

El presente proyecto se ha realizado gracias a la ayuda de las dos tutoras Victoria Torres y Manoli Albert, a Begoña Bonet, Jefa del proyecto MOSKitt y a Mario Cervera, quienes han sido de gran ayuda en todo momento.

8. Trabajo futuros

El trabajo que en esta memoria se presenta tiene como objetivo la validación de ambas herramientas (CAME y CASE), por ello se han realizado los cuestionarios de los anexos. Con el fin de poder visualizar, estudiar y valorar los resultados que en estos se deben plasmar, se ha propuesto la realización de dos experimentos que permitan esto. En primer lugar, se desea realizar una prueba a usuarios reales, analistas con o sin experiencia en el modelado de procesos, que prueben la

herramienta EPFC. Esta prueba constará de dos fases: 1) Modelado de un pequeño proceso usando la herramienta EPFC y 2) realización del cuestionario del anexo 2 con el fin de obtener resultados reales

En segundo lugar, se desea valorar el Gestor de Proyectos integrado en MOSKitt, para ello, se plantea la idea de realizar otra prueba con usuarios, esta vez expertos en el modelado y uso de MOSKitt. Concretamente estos usuarios serán Analistas Informáticos y Analistas de Organización pertenecientes a la CITMA. Estos usuarios serán quienes utilicen el Gestor de Proyectos durante el trabajo con MOSKitt, por ello será de gran ayuda este segundo experimento. La prueba consistirá en dos partes también: 1) Prueba del proceso de desarrollo de un proyecto usando el Gestor de Proyectos y 2) realización del cuestionario del anexo 3 con el fin de obtener y valorar resultados.

9. Anexos

9.1. ANEXO I - Introducción a la calidad del Software según el Modelo de calidad ISO/IEC 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cuál sigue los mismos conceptos.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

- **Funcionalidad** - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.
 - Idoneidad.
 - Exactitud.
 - Interoperabilidad.
 - Seguridad.
 - Cumplimiento de normas.

- **Fiabilidad** - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
 - Madurez
 - Recuperabilidad
 - Tolerancia a fallos

- **Usabilidad**- Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y

en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- Aprendizaje
 - Comprensión
 - Operatividad
 - Atractividad
-
- **Eficiencia**- Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas.
 - Comportamiento en el tiempo
 - Comportamiento de recursos
 - **Mantenibilidad**- Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
 - Estabilidad
 - Facilidad de análisis
 - Facilidad de cambio
 - Facilidad de pruebas
 - **Portabilidad**- Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.
 - Capacidad de instalación
 - Capacidad de reemplazamiento
 - Adaptabilidad
 - Co-Existencia

La subcaracterística Conformidad no está listada arriba ya que se aplica a todas las características.

Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en

condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus socios, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

La Organización Internacional para la Estandarización (ISO) dispone de dos definiciones de usabilidad:

“La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso”

Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su usabilidad, funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por ello un producto no es en ningún caso intrínsecamente usable, sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada (Bevan, 1994).”

La norma ISO/IEC 9126 está enfocada a la calidad de Producto y consta de las siguientes partes:

- Parte 1: Modelo de Calidad.
- Parte 2: Métricas externas.
- Parte 3: Métricas internas.
- Parte 4: Calidad en el uso de métricas

La especificación y la evaluación de la calidad de producto de software se puede conseguir definiendo características de calidad apropiadas, tomando en cuenta el objetivo de uso del producto de software.

Las métricas utilizadas se van a presentar según la norma ISO/IEC 9126–1 (Modelo de Calidad) y su evaluación se realizará aplicando la norma ISO 14598.

El modelo estructura los atributos de calidad de software en seis características (funcionalidad, fiabilidad, utilidad, eficacia, capacidad de mantenimiento y portabilidad), que se subdividen en subcaracterísticas. Las subcaracterísticas pueden ser medidas por métricas internas o externas.

CAPACIDAD DE ANÁLISIS

Deberían ser capaces de medir atributos tales como los recursos o esfuerzo de mantenimiento o del usuario en el diagnóstico de incidencias o causas de fallo del software, o identificar las partes a ser modificadas.

Las métricas presentadas en ISO/IEC TR 9126–2, son:

Alicia Jaén Bielsa

- Soporte a la función de diagnosis
- Datos registrados durante la operación.
- Tiempo de análisis del fallo
- Éxitos al encontrar causas de fallo
- Monitorización del estado durante la operación

CAPACIDAD DE CAMBIO

Deberían ser capaces de medir atributos tales como el esfuerzo del personal de mantenimiento o del usuario midiendo el comportamiento del personal de mantenimiento, usuario o sistemas, incluyendo el software cuando tratan de implementar una modificación especificada.

- Registrabilidad de cambios.
- Facilidad de parametrización.
- Disposición para el cambio.
- Tiempo empleado en implementar el cambio para satisfacción del usuario.
- Tiempo empleado en implementar un cambio por el personal de mantenimiento.

A continuación se presentan una serie de conclusiones generales sobre la aplicación de métricas para las subcaracterísticas “Capacidad de Análisis” y “Capacidad de Cambio”, utilizando la norma ISO 9126. La aplicación del modelo obliga a disponer de:

- Una normativa metodológica de gestión de las actividades de mantenimiento y desarrollo.
- Herramienta de gestión integradas en la metodología.
- Definición de responsabilidades en la ejecución de actividades.
- Necesidad de informar correctamente en las herramientas.
- Mecanismos de comunicación entre diferentes tipos de usuarios.
- La aplicación del modelo ha planteado los siguientes problemas:
- Asignación a las incidencias de su origen.
- Recogida de información asociada a pruebas.
- Filtrado de información por tipo de tecnología.
- La asignación de intervalos de referencia para realizar una evaluación, puede exigir un análisis riguroso en función de uno o varios parámetros.

La aplicación del modelo ha permitido:

- Evaluar calidad de producto.
- Evaluar calidad del equipo de mantenimiento.
- Posibilidad de incorporar las subcaracterísticas en la relación proveedor-cliente.
- Posibilidad de incorporar las subcaracterísticas en un ciclo de mejora.
- Posibilidad de incorporar las subcaracterísticas en un modelo de procesos.
- Posibilidad de incorporar requisitos de calidad asociados a la explotación del sistema.
- Posibilidad de establecer requisitos de aceptación asociados a productos y capacidad organizativa del equipo de mantenimiento.

9.2- ANEXO 2- Cuestionario EPFC

CUESTIONARIO HERRAMIENTA EPF Composer

El presente cuestionario se ha elaborado con el fin de evaluar la herramienta EPF Composer (Eclipse Process Framework Project). Está dividido en dos partes:

- Preguntas relacionadas con el perfil del usuario que va a realizar la prueba.
- Preguntas relacionadas con el uso de la herramienta.

En el cuestionario existen tres tipos de preguntas:

- **Tipo 1: Preguntas directas, en las que únicamente se indicará Si o No**
- **Tipo 2: Preguntas en las que es necesario puntuar un aspecto de la herramienta en la cual se usarán los valores de 1 a 5, siendo 1 la nota más baja y 5 la más alta.**
- **Tipo 3: Preguntas breves de desarrollo.**

Responda a las siguientes preguntas:

Fecha:

Nº	PERFIL DEL USUARIO	RESPUESTA
1	-Especifique sus estudios :	
2	-¿Hombre o mujer?:	
3	-¿Es usuario habitual en el modelado de sistemas?	
4	-Indique el número de sistemas modelados a lo largo de su vida:	
5	-¿Primera vez que hace uso del editor EPFC?	
6	-Indique número de veces que ha modelado procesos con EPFC	

7	-¿Había modelado con anterioridad usando SPEM?	
---	--	--

Responda al siguiente cuestionario teniendo especial atención al tipo de preguntas que se plantean (los tipos de preguntas se explican al comienzo de este documento)

1.	FUNCIONALIDAD	VALORACIÓN
1.1	<p>Adecuación</p> <p>-¿Es posible aplicar el editor al ámbito de trabajo (académico) requerido?</p>	[]
1.2	<p>Idoneidad</p> <p>-¿Ofrece una forma de trabajo apropiada?</p> <p>-¿Se ajusta a los requisitos necesarios para definir los procedimientos?</p> <p>-Puntúe según convenga:</p> <p style="padding-left: 40px;">-¿Ofrece un entorno adecuado para definir el procedimiento?</p>	<p>[]</p> <p>[]</p> <p>[]</p>
1.3	<p>Precisión</p> <p>-¿Permite definir todos los requisitos funcionales de los procedimientos?</p> <p>-¿Es posible definir la información necesaria para especificar el funcionamiento completo del procedimiento?</p> <p>-¿Se han encontrado limitaciones en la definición?</p> <p>-Describa las limitaciones encontradas brevemente:</p> <p style="padding-left: 40px;">-¿Se han podido subsanar utilizando mecanismos derivados internos?</p> <p>-¿Qué número de requisitos se han visto mermados? (porcentaje %):</p>	<p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p>

1.4	-Facilidades de uso:	
	<ul style="list-style-type: none"> -¿Dispone de ayuda interactiva?: -¿La herramienta está en Castellano?: -¿Existe la posibilidad de deshacer el último cambio?: -¿Se trata de una herramienta monousuario? : -¿Se trata de una herramienta multiusuario? : -Si es así, indique el número máximo de usuarios: -¿Permite trabajo en grupo?: 	<ul style="list-style-type: none"> [] [] [] [] [] [] []
2	SEGURIDAD	VALORACIÓN
	<ul style="list-style-type: none"> -¿Existen mecanismos que permitan guardar copias de seguridad?: -Si es así, ¿es posible acceder a las copias de seguridad con facilidad?: -¿Es posible deshacer cambios con facilidad?: 	<ul style="list-style-type: none"> [] [] []
4	PERSONALIZACIÓN DE LA HERRAMIENTA	VALORACIÓN
	<ul style="list-style-type: none"> -Elementos de la herramienta que se pueden configurar: -Metodología : -Documentación: -Impresos: -Pantallas: -Iconos: -Símbolos: -Menús : -Funciones: -Nomenclatura: 	<ul style="list-style-type: none"> [] [] [] [] [] [] [] []
4	USABILIDAD	VAORACIÓN
4.1	Operatividad	
	<ul style="list-style-type: none"> -¿La herramienta facilita la definición de procedimientos? -¿La herramienta permite definir fácilmente el siguiente tipo de información? -Tareas -Fases 	<ul style="list-style-type: none"> [] [] []

	-Recursos -Roles -Dependencia	[] [] []
4.2	Atractivo	
	-¿Presenta una estructura ordenada?: -¿Presenta una estructura intuitiva?: -¿Utiliza interfaces amigables?:	[] [] []
5	EFICIENCIA	VALORACIÓN
5.1	Utilización de los recursos	
	-Indique la rapidez de la herramienta según su experiencia: -¿La relación calidad/consumo de recursos es equilibrada?:	[] []
6	EFFECTIVIDAD	
	-¿El editor es útil en el trabajo que necesita el usuario?: -Si es así, ¿lo realiza en el tiempo razonable?:	[] []
7	FIABILIDAD	VALORACIÓN
7.1	Tolerancia de Fallos	
	-¿Se ha producido pérdida de datos?: -¿Se han roto modelos por acciones permitidas?:	[] []
8	PRUEBAS	VALORACIÓN
	-¿Dispone de Trazadores?: -¿Dispone de Simuladores?:	[] []

8.1	Control de calidad	VALORACIÓN
	<p>-¿Dispone de comprobadores de requisitos? :</p> <p>-¿Dispone de generadores de prueba basados en las especificaciones del diseño? :</p> <p>-¿Dispone de trazadores de requisitos a probar? :</p>	<p>[]</p> <p>[]</p> <p>[]</p>
8	UTILIZACIÓN DE RECURSOS	VALORACIÓN
	<p>Indique la cantidad de memoria que requiere la herramienta en sus PC:</p> <p>-Lógica:</p> <p>-Sistemas operativos/Versión (especificar) :</p> <p>-Otros requisitos(especificar) :</p> <p>-Física:</p> <p>-Espacio requerido en disco: (MB)</p> <p>-¿Necesita pantalla gráfica?:</p> <p>-¿Necesita ratón? :</p>	<p>[]</p> <p>[]</p> <p>[]</p>
9	SATISFACCIÓN	VALORACIÓN
	<p>-Puntúe la herramienta en términos generales y en base a su criterio y experiencia al realizar las tareas para las que fueron planeadas:</p> <p>-¿Ha alcanzado sus expectativas iniciales?</p> <p>-¿Escogería de nuevo esta herramienta para realizar este tipo de tareas?</p>	<p>[]</p> <p>[]</p> <p>[]</p>

Comentarios acerca de la herramienta:

9.3- ANEXO 3- Cuestionario Gestor Proyecto

CUESTIONARIO Gestor de Proyectos

El presente cuestionario se ha elaborado con el fin de evaluar la herramienta generada en base a procedimientos definidos en SPEM, l Gestor de Proyectos. El G.P se ha implantado en MOSKitt con el fin de dar una nueva opción en el proceso de desarrollo de aplicaciones con MOSKitt. El G.P dota a la herramienta CASE generada de una interfaz gráfica compuesta por una serie de vistas en Eclipse que no existían en la herramienta Dashboard.

El cuestionario consta dos apartados:

- Preguntas relacionadas con el perfil del usuario que va a realizar la prueba.
- Preguntas relacionadas con el uso de la herramienta.

En el cuestionario existen tres tipos de preguntas:

- **Tipo 1: Preguntas directas, en las que únicamente se indicará Si o No**
- Tipo 2: Preguntas en las que es necesario puntuar un aspecto de la herramienta en la cual se usarán los valores de 1 a 5, siendo 1 la nota más baja y 5 la más alta.
- Tipo 3: Preguntas breves de desarrollo.

Responda a las siguientes preguntas:

Nº	PERFIL DEL USUARIO	RESPUESTA
1	-Especifique sus estudios :	
2	-¿Hombre o mujer?:	
3	-¿Conoce el Proyecto MOSKitt?	
4	-¿Es usuario habitual de MOSKitt?	
5	-¿Ha utilizado alguna vez MOSKitt como herramienta de trabajo?	
6	-¿Conoce el concepto MOSKitt4ME?	

7	-¿Ha usado alguna vez el Dashboard?	
8	-Puntúe su valoración en cuanto al Dashboard como ayuda metodológica:	
9	-En su trabajo diario, ¿utiliza software que contenga funcionalidades de ayuda como guías o Gestores de Proyecto?	

Responda al siguiente cuestionario teniendo especial atención al tipo de preguntas que se plantean (los tipos de preguntas se explican al comienzo de este documento).

NOTA: Si no ha utilizado MOSKitt el asistente Dashboard, omita las preguntas que hablan de el.

1.	FUNCIONALIDAD	VALORACIÓN
1.1	<p>Idoneidad</p> <p>-¿El G.P ofrece una forma de trabajo apropiada?:</p> <p>-Puntúe según convenga:</p> <p>-¿Ofrece un entorno adecuado como guía durante el trabajo?:</p>	<p>[]</p> <p>[]</p>
1.3	<p>Precisión</p> <p>-¿Permite visualizar la información relevante en todo momento?</p> <p>-¿Es necesario cambiar de vistas de forma rápida?</p> <p>-¿Se han encontrado limitaciones en el manejo?</p> <p>-Describa las limitaciones encontradas brevemente:</p>	<p>[]</p> <p>[]</p> <p>[]</p>

	-¿Se han podido subsanar utilizando mecanismos derivados internos?	[]
1.4	-Facilidades de uso: -¿El G.P está en castellano?: -¿Se trata de una herramienta monousuario? : -¿Se trata de una herramienta multiusuario? : -Si es así, indique el número máximo de usuarios: -¿Permite trabajo en grupo?:	[] [] [] [] []
2	SEGURIDAD	VALORACIÓN
	-¿Existen mecanismos que permitan guardar copias de seguridad durante el proceso?: -Si es así, ¿es posible acceder a las copias de seguridad con facilidad?: -¿Es posible deshacer cambios con facilidad?:	[] [] []
4	PERSONALIZACIÓN DE LA HERRAMIENTA	VALORACIÓN
	-Elementos de la herramienta que se pueden configurar: -Metodología : -Documentación: -Impresos: -Pantallas: -Iconos: -Símbolos: -Menús : -Funciones: -Nomenclatura:	[] [] [] [] [] [] [] []
4	USABILIDAD	VAORACIÓN
4.1	Operatividad	
	-¿La herramienta facilita el seguimiento del proceso de los proyectos?	[]

	<p>-¿Permite abrir editores?:</p> <p>-¿Permite visualizar plantillas?:</p> <p>-¿Permite crear productos nuevos?:</p> <p>-¿Permite visualizar las guías de ayuda asociadas (Cheatsheets)?:</p> <p>-¿Permite ejecutar transformaciones?:</p> <p>-¿El G.P ha creado la estructura de plantillas correcta en base al tipo de proyecto seleccionado?:</p> <p>-Comente si ha tenido algún problema a la hora de ejecutar transformaciones desde el Gestor de Proyectos:</p> <p>-¿La herramienta permite diferenciar claramente los siguientes tipos de información ?:</p> <p style="padding-left: 40px;">-Tareas:</p> <p style="padding-left: 40px;">-Fases:</p> <p style="padding-left: 40px;">-Recursos:</p> <p style="padding-left: 40px;">-Roles:</p> <p style="padding-left: 40px;">-Dependencias entre tareas:</p>	<p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p>
<p>4.2</p>	<p>Atractivo</p>	
	<p>-¿El G.P permite visualizar la información más relevante de cada proceso?:</p> <p>-¿El G.P permite visualizar la información más relevante de forma ordenada?:</p> <p>-¿Las distintas vistas del G.P son intuitivas?</p> <p>-¿El G.P utiliza interfaces amigables?:</p>	<p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p>
	<p>EFICIENCIA</p>	<p>VALORACIÓN</p>

5		
5.1	Utilización de los recursos	
	-Indique la rapidez de la herramienta según su experiencia:	[]
	-¿La relación calidad/consumo de recursos es equilibrada?:	[]
6	EFFECTIVIDAD	
	-¿Ejecuta las tareas correctas en base a sus órdenes?:	[]
	-Si es así, ¿lo realiza en el tiempo razonable?:	[]
	-Indique el número de tareas que ha ejecutado de forma errónea:	
7	FIABILIDAD	VALORACIÓN
7.1	Tolerancia de Fallos	
	-¿Se ha producido pérdida de datos?:	[]
	-¿Se han roto modelos, plantillas o información por acciones permitidas?:	[]
8	PRUEBAS	VALORACIÓN
8.1	Control de calidad	VALORACIÓN
	-¿Dispone de comprobadores de requisitos? :	[]
	-¿Dispone de trazadores de requisitos a probar? :	[]
9	SATISFACCIÓN	VALORACIÓN
	-Puntúe la herramienta en términos generales y en base a su criterio y	

<p>experiencia al realizar las tareas para las que fueron planeadas:</p> <p>-¿Prefiere trabajar con el Gestor de Proyectos o con Dashboard?. Indique razones de la elección:</p> <p>-¿El G.P ha alcanzado sus expectativas iniciales?:</p> <p>-¿Escogería de nuevo la opción del G.P para realizar este tipo de tareas?:</p> <p>-¿Cree que el G.P aporta mejoras a MOSKitt?:</p> <p>-Indique qué mejoras aporta el G.P:</p> <p>-Indique qué carencias ha encontrado en el G.P :</p> <p>-Comentarios acerca de la herramienta en relación con el Dashboard (si no se dispone de criterios suficientes unicamente valore la herramienta según su experiencia):</p>	<p>[]</p> <p>[]</p> <p>[]</p> <p>[]</p>
--	---

9.4- ANEXO 4: Estructura creada en EPFC con SPEM

En este anexo se realiza una breve descripción de cada uno de los paquetes que han sido necesarios crear para la definición de los procedimientos:

1. Paquete GP (Gestión de Proyecto)

En todo proyecto existe una parte de Gestión de proyecto. Esta fase se lleva a cabo durante todo el ciclo de vida del proyecto y hace referencia al control y seguimiento tanto de las tareas como de los plazos definidos. Esta fase es vital para el buen funcionamiento y transcurso de todo proyecto por pequeño que sea.

Normalmente, esta tarea es labor del Jefe de Proyecto, pues es el quien debe de realizar un seguimiento periódico de la consecución de los elementos (pantallas, diagramas, etc...), plantillas y/o hitos a cumplir.

- **Contenido del paquete GP**

Dentro del paquete GP, se han definido las fases (Ver Figura 19) que hacen referencia a las tareas propias de la gestión del proyecto así como de los estudios previos realizados para la obtención de requisitos iniciales en un Proyecto.

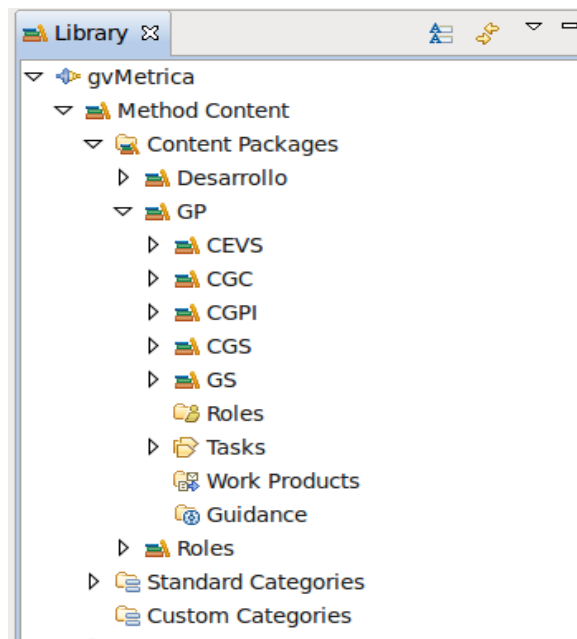


Figura 19: Contenido del paquete GP

Como se muestra en la imagen superior, el paquete contiene las siguientes fases:

- CEVS: Fase de estudio de viabilidad.
- CGC. Fase de Gestión de la configuración.
- CGPI: Fase de Gestión inicial del Proyecto.
- CGS: *Fase de Gestión de seguimiento.*

2. Paquete Desarrollo

Además de la Gestión, todo Proyecto de desarrollo de Software tiene una fase en la que se realizan las tareas propias de los proyectos de desarrollo. En esta fase, cada uno de los componentes del Proyecto, ejecuta sus tareas de forma secuencial o en paralelo con el resto del equipo de desarrollo. Es imprescindible que las tareas estén bien definidas, en cuanto a roles, dependencias y recursos para poder conseguir los objetivos del Proyecto.

• Contenido del Paquete Desarrollo

Dentro de este paquete, están definidas las tareas que hacen referencia a la parte de desarrollo (Ver figura 20), es decir, las tareas propias del ciclo de vida del software. Estas tareas son:

- CASI: Análisis del Sistema de Información
- CDSI: Diseño del Sistema de Información
- CCSI: Construcción del Sistema de Información
- CIAS: Implantación del Sistema de Información

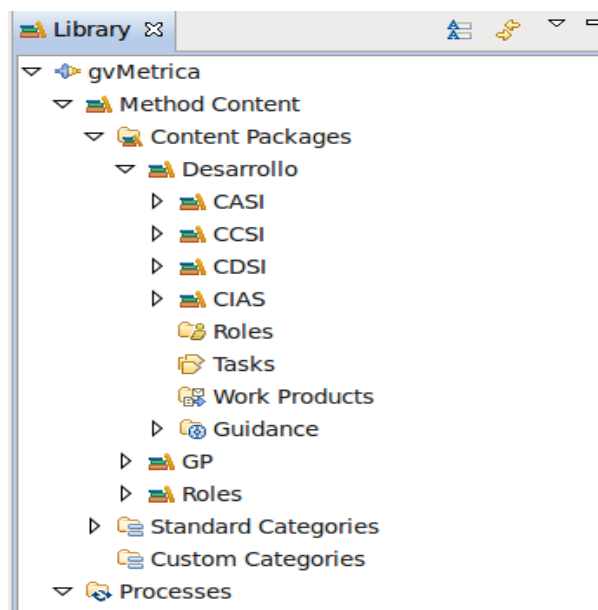


Figura 20: Contenido del paquete Desarrollo

3. Paquete Roles

Este paquete se ha creado con el único fin de definir los roles que existen en los proyectos de la CITMA y contiene únicamente cada uno de estos en la carpeta Roles del paquete. Esto se ha tenido

Alicia Jaén Bielsa

que realizar de esta forma, ya que no era posible crear una carpeta individual para definir los roles existentes. Por tanto, la solución adoptada ha sido crear un paquete específicamente para los roles.

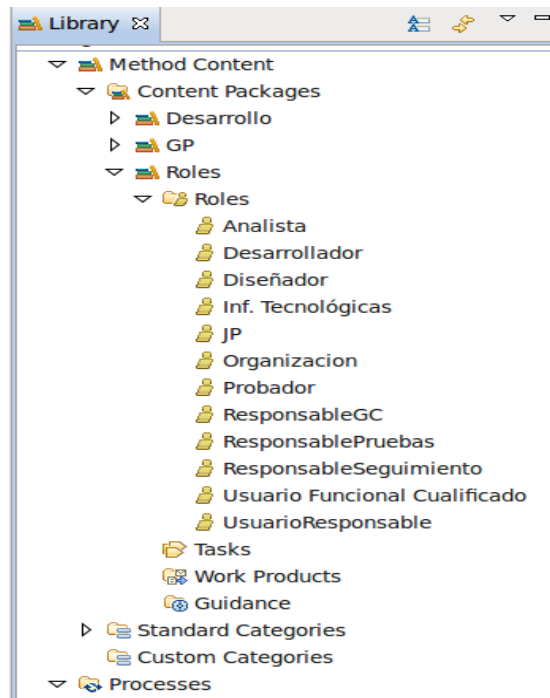


Figura 21: Contenido del paquete Roles

El siguiente paso ha sido crear en cada uno de los paquetes, el conjunto de tareas propias de la fase en la que se encuentre.

Se va a tomar el contenido del paquete Desarrollo para la explicación de los pasos seguidos a partir de este punto.

El siguiente paso es crear cada una de las tareas que forman cada fase y a su vez conforman el paquete. La estructura que presenta un paquete EPFC para la definición de las tareas, roles, productos y guías es la siguiente:

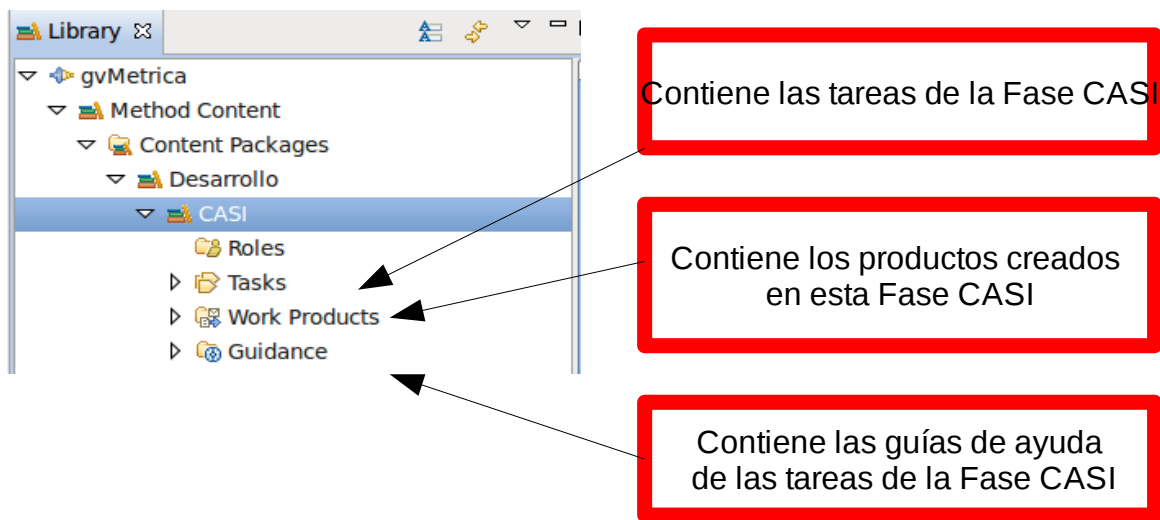


Figura 22: Estructura del contenido del elemento CASI del paquete Desarrollo.

Una vez creado el elemento dentro del paquete, es necesario poblar cada una de sus carpetas (Tasks, Work Products y Guidance) .

Contenido de carpeta Work Products

En esta carpeta es necesario crear todos los productos (Ver Figura 23) que se crearán en la fase CASI. Es necesario enlazar estas guías con los plugis de MOSKitt para que exista correspondencia en la ejecución del Gestor de Proyectos. Estos productos son del tipo *Artifact* definido en SPEM.

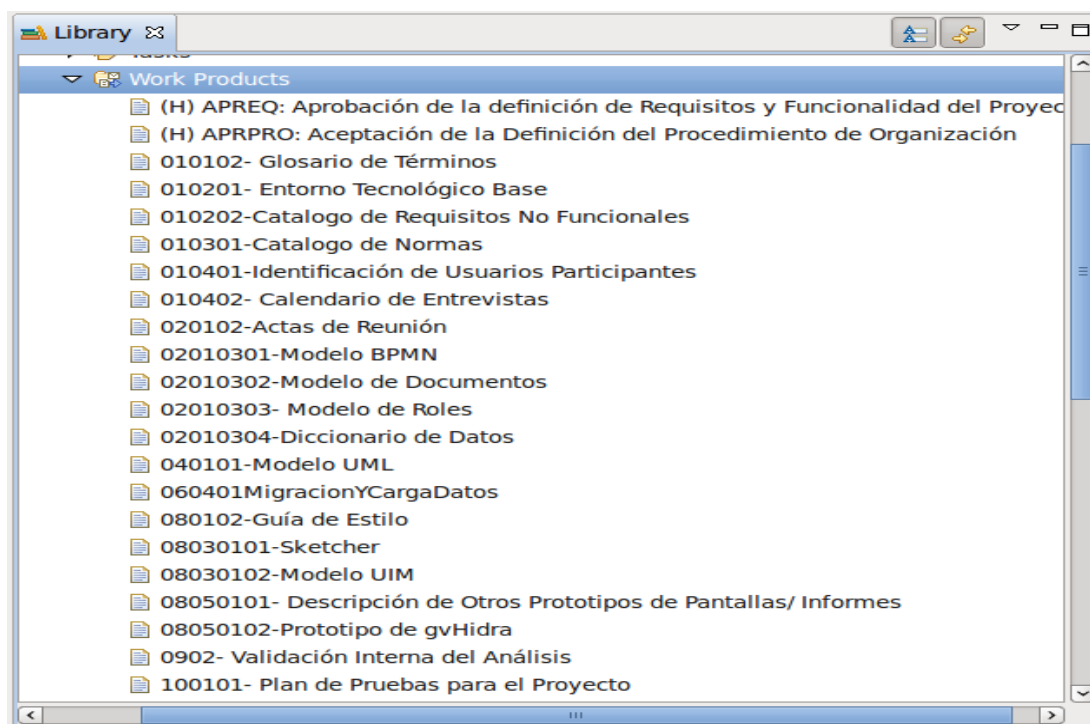


Figura 23: Ejemplo Algunos de productos creados en CASI

Contenido de carpeta Guidance

En esta carpeta es necesario crear todas las guías (Ver Figura 24) que se usarán en las tareas de la fase del ejemplo CASI. Es necesario enlazar estas guías con los plugis de MOSKitt para que exista correspondencia en la ejecución del Gestor de Proyectos.

En la creación de este tipo de objetos se ha tenido que tener en cuenta dos aspectos:

1. En SPEM, la forma de indicar el elemento es guía es escogiendo en la creación el tipo Guideline.
2. En SPEM, la forma de indicar que el elemento será una tarea ejecutable es escogiendo el tipo Tool Mentor. Este tipo de objetos se usa para las tareas que necesitan de un editor o de un asistente de configuración. De esta forma, cuando se seleccione una tarea de este tipo en el Gestor de Proyectos, se deberá de abrir o un editor de modelos (UML, UIM, BPMN...) o un asistente para lanzar transformaciones. En la creación de elementos de este tipo, habrá que especificar la dirección del plugin donde se encuentra el editor o del asistente de transformación necesario.

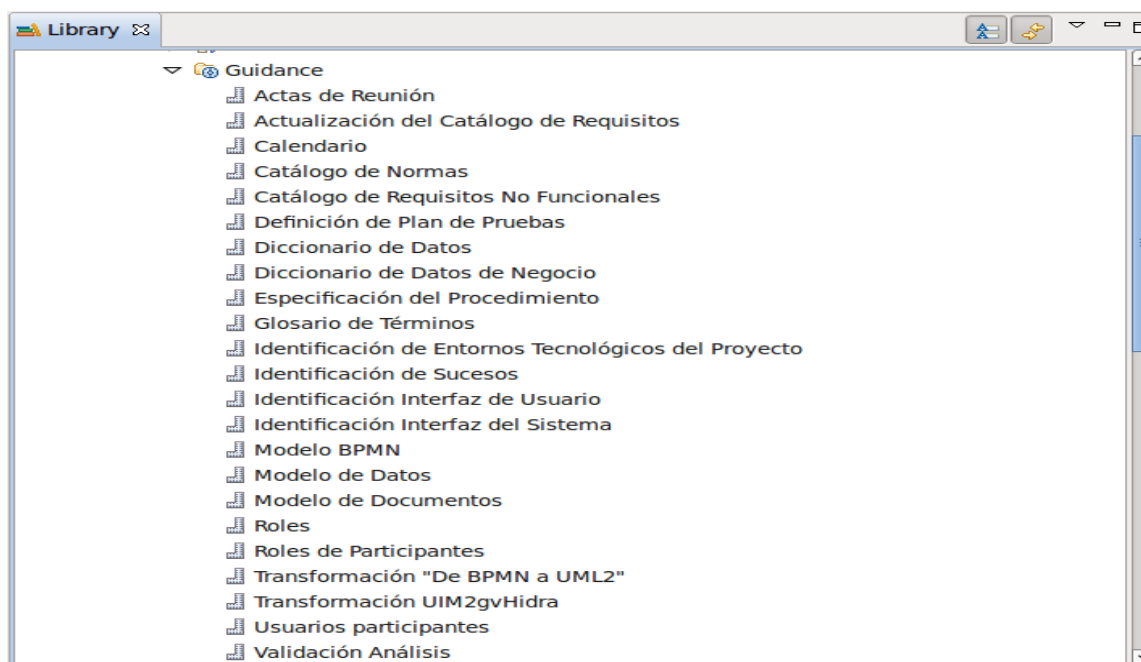


Figura 24: Algunos de las guías creadas en CASI

Contenido de carpeta Tasks

En la tarea tomada como ejemplo, se han definido las tareas (Ver figura 25) de la fase CASI, esta fase es una de las más costosas e importantes del ciclo de vida de todo proyecto por lo que es una de las que más contenido tiene.

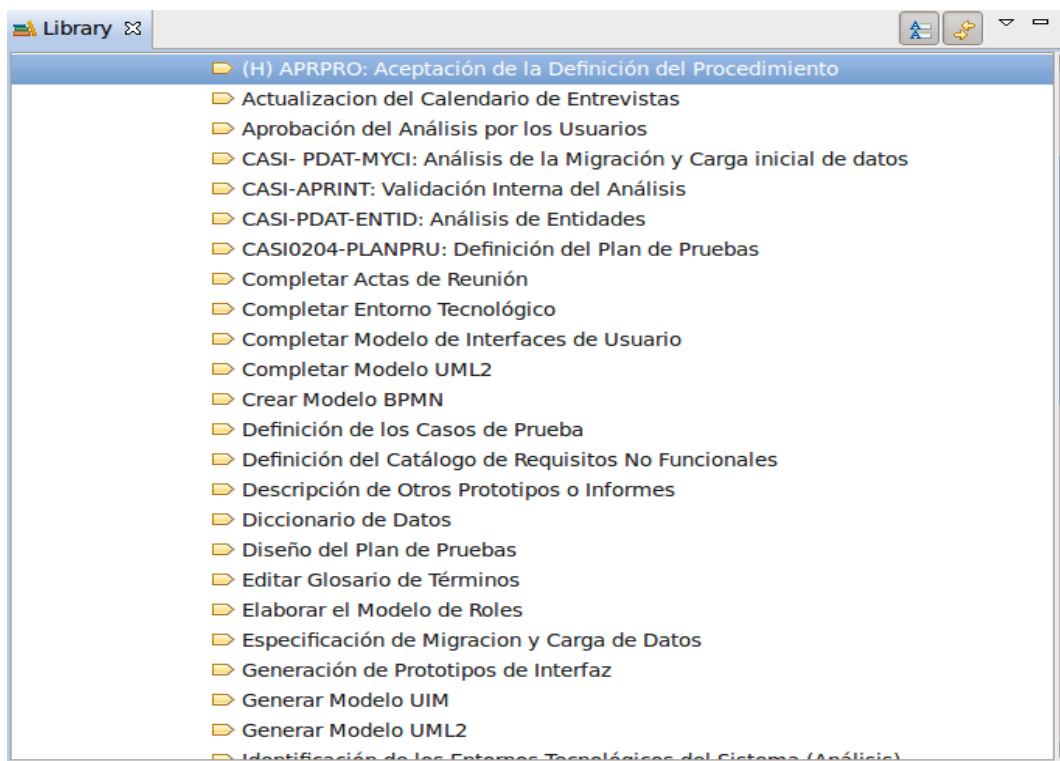


Figura 25: Parte del contenido de la carpeta Tasks de la fase CASI

Si pinchamos en cada una de las tareas definidas, podemos observar su estructura. Cada tarea tiene 8 pestañas (Ver figura 26) pero únicamente hemos utilizado 4 que son:

- **Description.** Contiene el nombre, un identificador de la tarea y una breve descripción, además de información complementaria.
- **Roles:** aquí se definen los roles que participan en la tarea. Estos pueden ser primarios o secundarios.
- **Work Products:** contiene información a cerca de las entradas y salidas de la tarea. Estas pueden ser documentos, modelos o plantillas.
- **Guidance:** contiene información de los plugins de cada guía definidos en MOSKitt como ayuda a la realización de cada tarea.

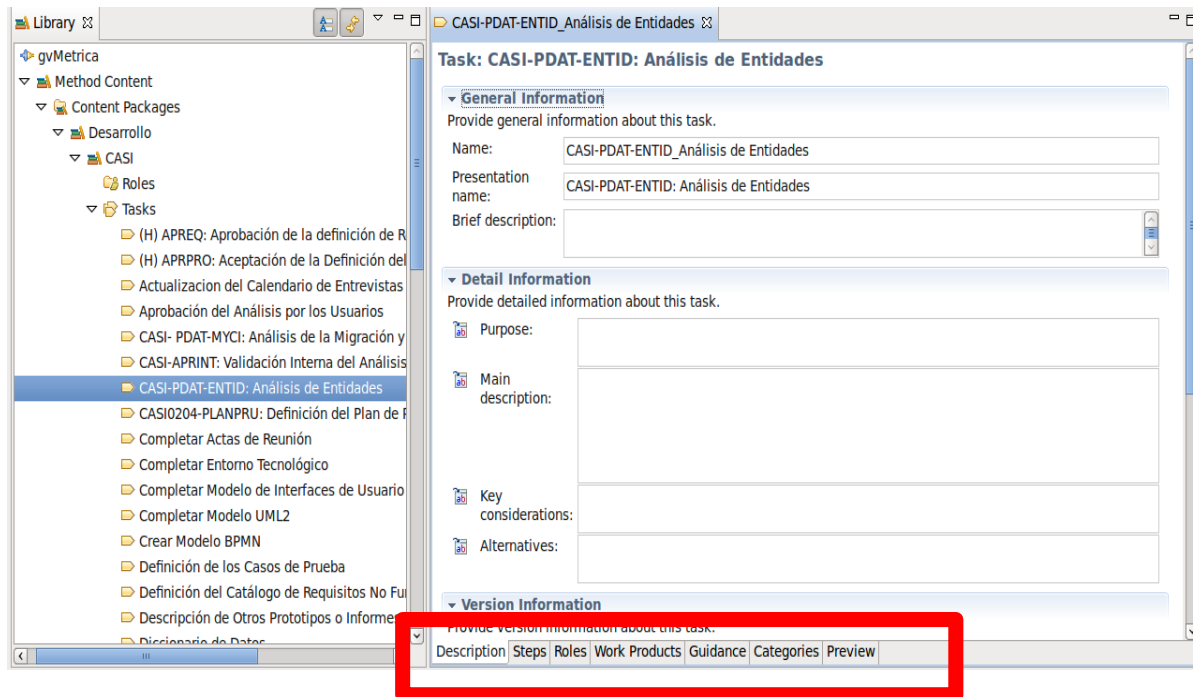


Figura 26: Pestañas de la tarea CASI-PDAT-ENTID_Análisis de Entidades

Una vez se crea una tarea, es necesario especificar información a cerca de quien la realiza (Pestaña Rol) qué productos tiene como entrada y/o salida (Pestaña Work Products) y qué guías tiene de ayuda (Pestaña Guidance). Por esta razón, es recomendable por comodidad crear primero los productos y guías y como último paso, crear las tareas y enlazar con la información pertinente.

Por lo tanto, el ultimo paso es añadir en cada una de las pestañas la información necesaria, enlazando mediante el asistente la *Descripción* necesaria (Ver Figura 27), los *Roles* (Ver Figura 28), *Productos* (Ver Figura 29) y *Guías* (Ver Figura 30).

10. Referencias

[1] Cervera, M. *Model Driven Method Engineering. A Supporting Infrastructure. Tesina del Máster. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. Diciembre 2010.*

[2] Brinkkemper, S.: *Method engineering: engineering of information systems development methods and tools. Information and Software Technology*, 38, 275-280, (1996).

[3] *Software & Systems Process Engineering Meta-model (SPEM) OMG Available Specification version 2.0. OMG Document Number: formal/2008-04-01.*

[4] *ISO/IEC 24744: Software Engineering: Metamodel for Development Methodologies. International Standards Organization/ International Electrotechnical Commission, Geneva (2007).*

11. Bibliografía

•RUIZ, Francisco, VERDUGO, Javier. *Guía de Uso de SPEM 2 con EPF Composer* [en línea]. Versión 3.0. Abril 2008. Disponible sur Internet: http://alarcos.inf-cr.uclm.es/doc/psgc/doc/lec/parte2b/guia-spem2&epf_v30.pdf.

•ALBERT Manolil, TORRES Victoria. *MOSKitt4ME: Ingeniería de Métodos al alcance de su organización*. [Presentación gráfica]. Valencia, 24 diapositivas. Universidad Politécnica de Valencia.

•BONET, Begoña. *MOSKitt: Soporte a una Metodología de Desarrollo Dirigido por Modelos, Basada en Eclipse*. [Presentación gráfica]. Valencia, 43 diapositivas. Conselleria de Infraestructuras, Territorio y Medio Ambiente.

•NICKNAFS Ali, RAMSIN Raman. *Computer-aided Method Engineering : An Analysis of existing environments*. Department of Computing Engineering, Sharif University of Technology, Tehran, Iran.

•Object Management Group. *Software Process Engineering Metamodel (SPEM) 2.0 (final adopted specification)*. <http://www.omg.org/docs/ptc/07-03-03.pdf>

•*Method Workbench User's Guide*, <http://www.metacase.com/support/40/manuals/mwb40sr2a4.pdf>

•ALIAGA CAVERO, Felipe. *Calidad de Software*. [Presentación gráfica]. 2ª ed, Instituto Continental, [2010], 46 diapositivas.

•Object Management Group. *Meta Object Facility (MOF) Core Specification. Version 2.0. formal/06-01-01*, <http://www.omg.org/docs/formal/06-01-01.pdf>, último acceso Septiembre 2011.