

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

# **MODELADO Y MEJORA DE PROCESOS DE SOFTWARE**

**CRISTIÁN ANDRÉS RIOSECO REINOSO**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO  
PROFESIONAL DE  
INGENIERO CIVIL EN INFORMÁTICA

JULIO 2012



PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

# **MODELADO Y MEJORA DE PROCESOS DE SOFTWARE**

**CRISTIÁN ANDRÉS RIOSECO REINOSO**

Profesor Guía: **Rodolfo Villarroel Acevedo**

Profesor Co-referente: **Silvana Roncagliolo de la Horra**

JULIO 2012

Doy mis más sinceros agradecimientos a mi familia, polola, amigos y seres queridos, quienes han hecho de mí la persona que hoy soy, me han permitido vivir plenamente y llenan mis días de alegrías y buenos momentos. A todos ustedes, muchas gracias.

# Índice

---

---

<b>Lista de figuras.....</b>	<b>ix</b>
<b>Lista de tablas .....</b>	<b>xii</b>
<b>Resumen .....</b>	<b>xiv</b>
<b>Abstract .....</b>	<b>xv</b>
<b>1. Introducción .....</b>	<b>1</b>
1.1. Descripción del problema y alcance.....	1
1.2. Objetivos.....	10
1.2.1. Objetivo general .....	10
1.2.2. Objetivos específicos .....	10
1.3. Estructuración de contenidos .....	10
<b>2. Marco Metodológico.....</b>	<b>12</b>
2.1. Procesos de software .....	12
2.1.1. Orígenes de los procesos de software .....	12
2.1.2. Definiciones y conceptos de procesos de software .....	13
2.2. Modelado de procesos de software .....	14
2.2.1. Elementos básicos de los modelos de procesos de software.....	15
2.2.2. Tipos de modelos de procesos de software .....	16
2.3. Ingeniería de procesos de software .....	18
2.3.1. Implementación de procesos y cambio .....	18
2.3.2. Definición de procesos.....	19
2.3.3. Evaluación de procesos.....	19
2.3.4. Medición de procesos y productos .....	20
2.4. Mejoramiento de procesos de software .....	20
2.4.1. Capability maturity model integration – CMMI .....	20
2.4.1.1. Elementos principales .....	21
2.4.1.2. Estructura de representación continua y por etapas.....	22
2.4.1.3. Niveles de capacidad .....	24
2.4.1.4. Niveles de madurez.....	24
2.4.1.5. Representación continua .....	27
2.4.1.6. Representación por etapas.....	28
2.4.2. ISO/IEC15504 .....	29
2.4.2.1. Modelo de referencia .....	29

2.4.2.2. Evaluación.....	30
2.4.3. Competisoft.....	31
2.4.3.1. Modelo de procesos.....	32
2.4.3.2. Modelo de evaluación .....	34
2.4.3.3. Modelo de mejora .....	35
2.5. Notaciones para el modelado de procesos de software. ....	38
2.5.1. Características de las notaciones para el modelado de procesos de software ...	38
2.5.2. Otras notaciones para el modelado de procesos de software .....	39
2.5.3. System and Software Process Engineering Metamodel – SPEM2.....	40
2.5.3.1. Aspectos principales de SPEM2.....	40
2.5.3.2. Estructura de paquetes de SPEM2.....	44
2.5.4. Comparación entre notaciones de modelado.....	46
2.6. Herramientas de modelado de procesos de software .....	47
2.6.1. IBM Rational Method Composer .....	47
2.6.2. Osellus IRIS .....	48
2.6.3. Eclipse Process Framework Composer.....	50
<b>3. Modelado del proceso de mantenimiento Agil_Mantema.....</b>	<b>53</b>
3.1. Proceso de mantenimiento de software .....	53
3.1.1. Tipos de mantenimiento.....	54
3.1.2. Roles dentro del proceso de mantenimiento .....	55
3.1.3. Proceso de mantenimiento .....	55
3.1.4. Detalles de los procesos .....	57
3.1.4.1. I0: Planificación del proceso .....	57
3.1.4.2. I1: Atención de la petición .....	58
3.1.4.3. SNP: SprintM no planificable .....	59
3.1.4.4. SP: SprintM planificable .....	60
3.1.4.5. SSM: Seguimiento del SprintM .....	62
3.1.4.6. F1: Finalización de la intervención .....	63
3.1.4.7. F2: Retiro .....	64
3.1.4.8. F3: Finalización del servicio .....	65
3.2. Proceso de modelado en EPF Composer.....	65
3.3. Modelado de Agil_Mantema en EPF Composer .....	73
3.3.1. Elementos del modelado .....	73
3.3.1.1. I0: Planificación del proceso .....	75

3.3.1.2. I1: Atención de la petición .....	77
3.3.1.3. SNP: SprintM no planificable .....	79
3.3.1.4. SP: SprintM planificable .....	82
3.3.1.5. SSM: Seguimiento del SprintM .....	86
3.3.1.6. F1: Finalización de la intervención .....	88
3.3.1.7. F2: Retiro .....	90
3.3.1.8. F3: Finalización del servicio .....	92
<b>4. Aplicación de la mejora de procesos de software .....</b>	<b>94</b>
4.1. Empresa: TyC Software.....	94
4.1.1. Informaciones generales .....	94
4.1.2. Análisis estratégico de TyC Software.....	95
4.1.2.1. Áreas de negocio .....	96
4.1.2.2. Nivel de soporte funcional de sistemas.....	96
4.1.2.3. Problemas y necesidades.....	97
4.1.2.4. Desarrollo de la organización.....	97
4.2. Instalación del ciclo.....	98
4.3. Diagnóstico de procesos .....	99
4.3.1. Entrevistas realizadas.....	107
4.3.2. Proceso de mantenimiento actual .....	108
4.3.2.1. Tipos de mantenimiento.....	108
4.3.2.2. Roles .....	108
4.3.2.3. Actividades.....	109
4.4. Formulación de la mejora .....	112
4.4.1. Planificación de la mejora .....	112
4.4.2. Análisis de la mejora.....	112
4.4.3. Diseño de la mejora .....	114
4.4.3.1. Tipos de mantenimiento.....	114
4.4.3.2. Roles .....	114
4.4.3.3. Actividades.....	115
4.4.4. Definición nuevo proceso .....	115
4.4.4.1. TyC-1: Ingreso y recepción de incidentes.....	116
4.4.4.2. TyC-2: Asignación de incidentes .....	119
4.4.4.3. TyC-3: Resolución de incidente .....	123
4.4.4.4. TyC-4: Seguimiento de incidentes .....	126

4.4.4.5. TyC-5: Finalización de la intervención .....	129
4.4.4.6. TyC-6: Retiro .....	132
4.4.4.7. TyC-7: Finalización del servicio .....	137
4.5. Ejecución de la mejora .....	138
4.5.1. Capacitaciones .....	139
4.5.2. Diagnóstico final.....	140
4.6. Revisión de la mejora .....	148
4.6.1. Retroalimentación.....	149
4.6.2. Análisis del impacto de la mejora.....	149
4.6.3. Análisis del compromiso.....	150
<b>5. Conclusiones .....</b>	<b>152</b>
<b>Referencias .....</b>	<b>156</b>
<b>Anexos .....</b>	<b>159</b>
Cuestionario OPE3:.....	159



## Lista de figuras

---

---

Figura 1.1 Principales prioridades dentro de las empresas (GECHS) .....	6
Figura 1.2 Beneficios de implementar procesos formales (GECHS) .....	7
Figura 1.3 Razones que motivarían a implementar procesos formales (GECHS) .....	7
Figura 1.4 Nivel de uso de estándares en gestión de TI (ENTI 2010).....	8
Figura 1.5 Exigencia de certificación TI a proveedores (ENTI 2007) .....	8
Figura 2.1 Estructura de proceso ISO/IEC 12207 .....	14
Figura 2.2 Relación entre componentes básicos.....	16
Figura 2.3 Representación continua.....	23
Figura 2.4 Representación por etapas .....	23
Figura 2.5 Nivel de capacidad de áreas de proceso elegidas.....	28
Figura 2.6 Etapas de la representación por etapas .....	28
Figura 2.7 Arquitectura modelo de referencia.....	29
Figura 2.8 Modelo de procesos.....	32
Figura 2.9 Relación entre los elementos de EvalProSoft .....	34
Figura 2.10 Relación entre los elementos de PmCompetisofit.....	36
Figura 2.11 Idea central de SPEM2 .....	41
Figura 2.12 Niveles de modelado de MOF .....	42
Figura 2.13 Marco teórico de SPEM2 .....	43
Figura 2.14 Ortogonalidad en el modelo de procesos RUP .....	43
Figura 2.15 Jerarquía de desglose del trabajo .....	44
Figura 2.16 Estructura de paquetes de SPEM2 .....	45
Figura 2.17 Arquitectura EPF Composer .....	52
Figura 3.1 Estructura general Agil_Mantema .....	53
Figura 3.2 Proceso de Agil_Mantema.....	57
Figura 3.3 Elementos de un plug-in.....	66
Figura 3.4 Paquetes de contenido jerarquizados .....	67
Figura 3.5 Creación de roles, tareas, productos de trabajos y guías .....	68
Figura 3.6 Pestañas ventana de una tarea .....	69
Figura 3.7 Creación de categorías.....	69
Figura 3.8 Creación de configuración.....	70
Figura 3.9 Creación de capability patterns .....	71
Figura 3.10 Agregación de tareas a un proceso.....	72
Figura 3.11 Diagrama de actividad.....	72
Figura 3.12 Elementos Plug-in Agil_Mantema .....	74
Figura 3.13 DESC I0: Planificación del proceso.....	75
Figura 3.14 WBS I0: Planificación del proceso .....	75
Figura 3.15 TA I0: Planificación del proceso .....	76
Figura 3.16 WPU I0: Planificación del proceso .....	76
Figura 3.17 DESC I1: Atención de la petición.....	77
Figura 3.18 WBS I1: Atención de la petición .....	77
Figura 3.19 TA I1: Atención de la petición.....	78
Figura 3.20 WPU I1: Atención de la petición .....	78
Figura 3.21 DESC SNP1: Análisis del error .....	79
Figura 3.22 WBS SNP1: Análisis del error.....	79

Figura 3.23 TA SNP1: Análisis del error .....	80
Figura 3.24 WPU SNP1: Análisis del error .....	80
Figura 3.25 DESC I0: Planificación del proceso .....	80
Figura 3.26 WBS I0: Planificación del proceso .....	81
Figura 3.27 TA I0: Planificación del proceso .....	81
Figura 3.28 WPU I0: Planificación del proceso .....	82
Figura 3.29 DESC SP1: Análisis de la petición .....	82
Figura 3.30 WBS SP1: Análisis de la petición .....	83
Figura 3.31 TA SP1: Análisis de la petición .....	83
Figura 3.32 WPU SP1: Análisis de la petición.....	83
Figura 3.33 DESC I0: Planificación del proceso.....	84
Figura 3.34 WBS I0: Planificación del proceso .....	84
Figura 3.35 TA I0: Planificación del proceso .....	85
Figura 3.36 WPU I0: Planificación del proceso .....	85
Figura 3.37 DESC SSM: Seguimiento del SprintM .....	86
Figura 3.38 WBS SSM: Seguimiento del SprintM.....	86
Figura 3.39 TA SSM: Seguimiento del SprintM.....	87
Figura 3.40 WPU SSM: Seguimiento del SprintM .....	87
Figura 3.41 DESC F1: Finalización de la intervención .....	88
Figura 3.42 WBS F1: Finalización de la intervención.....	88
Figura 3.43 TA F1: Finalización de la intervención.....	89
Figura 3.44 WPU F1: Finalización de la intervención .....	89
Figura 3.45 DESC F2: Retiro .....	90
Figura 3.46 WBS F2: Retiro.....	90
Figura 3.47 TA F2: Retiro .....	91
Figura 3.48 WPU F2: Retiro .....	91
Figura 3.49 DESC F3: Finalización del servicio .....	92
Figura 3.50 WBS F3: Finalización del servicio .....	92
Figura 3.51 TA F3: Finalización del servicio.....	93
Figura 3.52 WPU F3: Finalización del servicio .....	93
Figura 4.1 Organigrama y áreas de TyC Software .....	96
Figura 4.2 Actividad: Ingreso y recepción de incidentes .....	110
Figura 4.3 Actividad: Asignación de incidentes.....	111
Figura 4.4 EyS TyC-1Coordinador de incidentes.....	117
Figura 4.5 EyS TyC-1Usuario .....	118
Figura 4.6 Diagrama actividad TyC-1 .....	118
Figura 4.7 EyS TyC-2Coordinador de incidentes.....	120
Figura 4.8 EyS TyC-2Equipo de mantenimiento .....	121
Figura 4.9 EyS TyC-2Usuario .....	121
Figura 4.10 Diagrama actividad TyC-2 .....	122
Figura 4.11 EyS TyC-3Coordinador de incidentes.....	124
Figura 4.12 EyS TyC-3Equipo de mantenimiento .....	124
Figura 4.13 EyS TyC-3Usuario .....	125
Figura 4.14 Diagrama actividad TyC-3 .....	125
Figura 4.15 EyS TyC-4Coordinador de incidentes.....	127
Figura 4.16 EyS TyC-4Equipo de mantenimiento .....	128
Figura 4.17 Diagrama actividad TyC-4 .....	128

Figura 4.18 EyS TyC-5Cliente .....	130
Figura 4.19 EyS TyC-5Coordinador de incidentes.....	130
Figura 4.20 EyS TyC-5Equipo de mantenimiento .....	131
Figura 4.21 Diagrama actividad TyC-5 .....	132
Figura 4.22 EyS TyC-6 Usuario .....	134
Figura 4.23 EyS TyC-6 Equipo de mantenimiento.....	135
Figura 4.24 EyS TyC-6 Cliente .....	136
Figura 4.25 Diagrama actividad TyC-6 .....	136
Figura 4.26 EyS TyC-7 Cliente .....	137
Figura 4.27 EyS TyC-7 Equipo de mantenimiento.....	138
Figura 4.28 Diagrama actividad TyC-7 .....	138

## Lista de tablas

---

Tabla 1.1 Cifras Standish Report 1994 - 2009.....	2
Tabla 1.2 Tamaño de las empresas encuestadas (estudio ACTI).....	2
Tabla 1.3 Rubro de las empresas encuestadas (estudio ACTI).....	3
Tabla 1.4 Certificaciones de las empresas encuestadas (estudio ACTI).....	3
Tabla 1.5 Mercados que atienden las empresas encuestadas (estudio ACTI) .....	4
Tabla 1.6 Interés de exportar por parte de las empresas encuestadas (estudio ACTI).....	4
Tabla 1.7 Aspectos a desarrollar para realizar exportaciones (estudio ACTI) .....	5
Tabla 2.1 Comparación niveles de capacidad y madurez.....	24
Tabla 2.2 Niveles de madurez y áreas de proceso asociadas.....	27
Tabla 2.3 Niveles de capacidad ISO/IEC 15504 .....	30
Tabla 2.4 Esquema de agregación de atributos de proceso .....	31
Tabla 2.5 Esquema de agregación de atributos de proceso .....	47
Tabla 3.1 Resumen niveles de servicio y capacidad Agil_Mantema .....	54
Tabla 3.2 Entradas y salidas tareas actividad de planificación del proceso .....	58
Tabla 3.3 Entradas y salidas tareas actividad de atención de la petición .....	59
Tabla 3.4 Entradas y salidas tarea subactividad de análisis de error .....	60
Tabla 3.5 Entradas y salidas de la tarea de la subactividad de intervención correctiva urgente.	60
Tabla 3.6 Entradas y salidas tarea subactividad de análisis de la petición.....	61
Tabla 3.7 Entradas y salidas tarea subactividad de intervención y pruebas .....	62
Tabla 3.8 Entradas y salidas tareas actividad de seguimiento del SprintM.....	63
Tabla 3.9 Entradas y salidas tareas actividad de finalización de la intervención .....	64
Tabla 3.10 Entradas y salidas tareas actividad de retiro.....	65
Tabla 3.11 Entradas y salidas tareas actividad de finalización del servicio .....	65
Tabla 4.1 Sistemas y funcionalidades utilizados .....	97
Tabla 4.2 (A.1.) Planificación del proceso .....	100
Tabla 4.3 (A.2.) Atención petición de modificación.....	101
Tabla 4.4 (A.3.) Tipo de mantenimiento no planificable .....	102
Tabla 4.5 (A.4.) Intervención correctiva urgente.....	102
Tabla 4.6 (A.5.) Tipo de mantenimiento planificable .....	103
Tabla 4.7 (A.6.) Intervención y pruebas.....	103
Tabla 4.8 (A.7.) Seguimiento SprintM.....	104
Tabla 4.9 (A.8.) Finalización de la intervención .....	105
Tabla 4.10 (A.9.) Pasar a producción.....	106
Tabla 4.11 (A.10.) Retirada .....	106
Tabla 4.12 (A.11.) Finalización del servicio .....	107
Tabla 4.13 Resultados diagnóstico inicial .....	107
Tabla 4.14 Entrevistas realizadas.....	107
Tabla 4.15 (A.1.)Planificación del proceso .....	140
Tabla 4.16 (A.2.) Atención petición de modificación.....	142
Tabla 4.17 (A.3.) Tipo de mantenimiento no planificable .....	142
Tabla 4.18 (A.4.) Intervención correctiva urgente.....	143
Tabla 4.19 (A.5.) Tipo de mantenimiento planificable .....	144
Tabla 4.20 (A.6.) Intervención y pruebas .....	144
Tabla 4.21 (A.7.) Seguimiento SprintM.....	145

Tabla 4.22 (A.8.) Finalización de la intervención..... 146  
Tabla 4.23 (A.9.) Pasar a producción..... 147  
Tabla 4.24 (A.10.) Retirada ..... 147  
Tabla 4.25 (A.11.) Finalización del servicio ..... 148  
Tabla 4.26 Resultados diagnóstico final..... 148

# Resumen

---

El realizar de la mejor manera posible las actividades del desarrollo de software siempre ha sido difícil para cualquier tipo de organización. Esto dada la falta de un enfoque ingenieril que guíe dichas actividades. Por ello, a lo largo de la historia del desarrollo y la ingeniería de software se han investigado, creado y aplicado una serie de posibles soluciones, unas con más éxito que otras, que intentan mejorar el desarrollo de software. Entre estas soluciones se encuentra la mejora de procesos de software. Lo que se propone es definir, controlar y mejorar los procesos y actividades que utiliza una organización para el desarrollo de software para aumentar los niveles de calidad en los productos y servicios ofrecidos.

Una serie de estudios revela que las organizaciones, de todos los tamaños, dedicadas al desarrollo de software en Chile tienen en consideración la mejora de procesos de software como una herramienta que les permite asegurar y certificar la calidad de sus productos y servicios a nivel nacional e internacional.

Además, la industria del desarrollo de software chilena comienza a mirar el mercado internacional para ofrecer sus productos y servicios, por lo que las certificaciones y evaluaciones internacionales relacionadas a la mejora de procesos de software son un aspecto primordial para poder competir con la industria desarrolladora de otros países. Aún así, muchas empresas no han aplicado o no cuentan con certificaciones para sus procesos. Esto se ve en mayor medida en las pequeñas y medianas empresas – PyMEs dedicadas a este rubro, pues no cuentan con los recursos ni los conocimientos para implementar este tipo de mejoras y certificaciones.

El presente documento hace uso del marco metodológico de Competisoft, particularmente el proceso de mejoras, aplicándolo como caso práctico en una PyME. La definición de procesos será apoyada por el modelado de procesos de software con la notación estándar SPEM2 y la herramienta de modelado Eclipse Process Framework Composer.

Así se expondrá la ejecución del proceso de mejoras al proceso de soporte y mantenimiento de una PyME, donde se tomó como referencia el proceso de mantenimiento Agil\_Mantema, definido por Competisoft, así también se tomaron elementos del proceso original que poseía la empresa y que mostraron alto grado de cumplimiento según el diagnóstico inicial realizado. Finalmente, se darán a conocer los resultados y lecciones obtenidas tras la aplicación de las mejoras.

*Palabras clave: desarrollo de software, calidad, mejoramiento de proceso, modelado de procesos.*

## Abstract

---

To achieve, in the best possible way, the activities of software development has been always difficult for any organization. Because of the un-engineering view which guides these activities. This is way, through the history of development and history and software engineering it has been investigated, created and applied a set of possible solutions, some more successful than others, seeking to improve software development. Among these solutions it's found software process improvement. What is proposed is define, control and improve the process and activities used by and organization for the software development to increase the quality levels in products and services offered.

A number of studies show that organizations, of all sizes, dedicated to software development in Chile have in mind the software process improvement like a tool that allows them to ensure and certified the quality of their products and services in national and international levels.

Besides, the Chilean software development industry is beginning to look the international market to offer their products and services, so that the international certifications and assessments related to the software process improvement are a key aspect to compete with the development industry in others countries. Still, many organizations have not applied or do not have certifications for their processes. This is seen more in small and medium enterprises – PyMEs dedicated to this area, they do not have the resources or knowledge to implement these kinds of improvements and certifications.

This paper uses the methodological framework of Competisoft, process improvement particularly, applying it as a practical case in a PyME. The definition of processes will be supported by the software process modeling with the standard notation SPEM2 and the modeling tool Eclipse Process Framework Composer.

This will expose the implementation of process improvement of the support and maintenance process of a PyME, which references the maintenance process Agil\_Mantema, defined by Competisoft, also took elements from the original process of the enterprise, which had high compliance according to the initial diagnosis made. Finally, we will announce the results and lessons learned followed by the implementation of the improvements.

*Key words: software development, quality, process improvement, process modeling.*

# 1. Introducción

---

En esta sección se presentará la descripción detallada del problema que se tratará a lo largo de todo este trabajo, el alcance que tendrá y la solución que se buscará aplicar. Posteriormente se darán a conocer los objetivos, general y específicos. Terminando esta sección se presentará la estructura general que posee este documento para guiar al lector.

## 1.1. Descripción del problema y alcance

Tal como se presenta en Boehm [2006] el desarrollo de software ha sido una actividad que se lleva a cabo desde, aproximadamente, la década de los '50. Desde entonces han ido surgiendo distintas problemáticas a causa de la mayor complejidad de los sistemas a desarrollar.

En un principio se construía software siguiendo el proceso utilizado para la creación del hardware de la época. Con el tiempo y el crecimiento de los sistemas este enfoque de construcción de software cambió por uno más ad-hoc, un primer esbozo de modelo cascada con sus etapas secuenciales relativamente definidas. Aun así habían problemas como el “código spaghetti” o el “codificar y arreglar”, la solución común para estos tipos de problemas era el llamado “programador héroe”, se trataba de programadores que se esforzaban al máximo para cumplir de la mejor manera su trabajo, lo que significaba un sacrificio personal para salvar el proyecto.

Con el pasar de los años se desarrollaron diversos métodos y técnicas para resolver los distintos problemas con que se enfrentaba el desarrollo de software. En la década de los '70 surgió un nuevo paradigma, los métodos estructurados (programación, diseño, etc.). Esto fue una solución hasta la conocida crisis del software (década de los '80), donde estos métodos llevaron a sistemas que no podían terminar de ser desarrollados por exceder los recursos presupuestados (costo y tiempo) o, los que eran terminados con una gran cantidad de personal, no cumplían con los requerimientos del usuario, es decir, eran de baja calidad y confiabilidad. Durante esta época una gran parte de los proyectos de desarrollo eran cancelados o el producto entregado era deficiente, por lo que requerían cambios.

Para ofrecer una solución a este problema, se miró a la industria manufacturera donde el proceso utilizado para crear un producto tiene directa relación en este último, es decir, si se mejora el proceso que se utiliza, o se reemplaza por uno más eficiente, se obtendrán mejoras en la calidad del producto final. Considerando esto, se buscó aplicarlo a la industria del desarrollo de software a través de la creación de variados estándares (CMM, CMMI, ISO/IEC 15504, etc.) que buscan certificar y/o mejorar los procesos de las organizaciones desarrolladoras de software. Así se puede tener claro cómo son los procesos utilizados en relación a las buenas prácticas establecidas por estos estándares.

Aún con la existencia de estos estándares el problema no está resuelto, pues éstos simplemente indican cuáles son las mejores prácticas y evalúan los procesos, pero no dicen cómo se debe llevar a cabo un proceso de mejora. Hasta el día de hoy existe una gran cantidad de proyectos de desarrollo que son cancelados o deben someterse a cambios por no cumplir



con los requerimientos del usuario. Esto se puede ver en las cifras publicadas en el período 1994 - 2009 por el Standish Group en su *Standish Report* que se pueden observar en la Tabla 1.1.

**Tabla 1.1** Cifras Standish Report 1994 - 2009

Año	Exitoso	Cambiado	Fracasado
1994	16%	53%	31%
1996	27%	33%	40%
1998	26%	46%	28%
2000	28%	49%	23%
2004	29%	53%	18%
2006	35%	46%	19%
2009	32%	44%	24%

Actualmente las empresas desarrolladoras de software han visto la necesidad de adoptar una forma de trabajo que les permita: entender, controlar, comunicar, mejorar, predecir y certificar los procesos que utilizan para llevar a cabo las actividades alrededor del desarrollo de software. Además, se busca que los procesos utilizados puedan ser flexibles para poder modificarlos o reemplazarlos ante la constante dinámica de cambio existente en el mundo y, por supuesto, en la industria del desarrollo de software. Así el producto que se obtendrá será de mayor calidad y cumplirá de mejor manera las necesidades del cliente, con lo que se generará una mejor imagen de la organización y de los productos que en ella se crean.

La realidad chilena no escapa de los problemas recién planteados, pues son pocas las organizaciones desarrolladoras que tienen sus procesos bien establecidos y documentados. Además, la cantidad de empresas certificadas por algún estándar internacional para este ámbito es una mínima parte del total de la industria de software en Chile. Así se puede apreciar en el estudio “Centro Servicios Globales TICs 2010” presentado por la Asociación Chilena de Empresas de Tecnologías de Información A.G. – ACTI [ACTI, 2010]. La muestra de este estudio fue de 260 empresas (extrapolando el universo total a unas 2500 organizaciones, donde unas 2000 serían pequeñas y medianas empresas - PyMEs). Algunos detalles a considerar sobre este estudio se presentan en las Tablas 1.2, 1.3 y 1.4.

**Tabla 1.2** Tamaño de las empresas encuestadas (estudio ACTI)

Tamaño empresas	Total encuestas
Empresas grandes	78
Empresas medianas	80
Empresas pequeñas	102
<b>Total</b>	<b>260</b>

**Tabla 1.3** Rubro de las empresas encuestadas (estudio ACTI)

Rubro empresas (*)	Total
Servicios	90%
Software	62%
Hardware	30%

(\*): Las empresas pueden tener más de un rubro.

De este estudio se conocieron las certificaciones con que cuentan las empresas o su personal (Tabla 1.4). Se observa que existe una preocupación por la mejora de procesos, pues la certificación ISO 9001-2000 es la que domina ampliamente (con un 64,0%) por sobre cualquiera de las otras certificaciones internacionales y nacionales. Esto es insuficiente en la industria del desarrollo de software, pues es necesario contar con certificaciones apropiadas al mercado, como la ISO/IEC 15504 o evaluaciones CMMI. Este último tipo de evaluación cuenta con muy pocas empresas acreditadas, sólo un 3,8% cuenta con evaluación nivel 2 y un 2,9% con nivel 3.

**Tabla 1.4** Certificaciones de las empresas encuestadas (estudio ACTI)

Certificaciones con las que cuenta	Total
ISO 9001 – 2000	64,0%
Certificaciones persona	12,1%
ISO 9000	8,4%
En proceso	6,7%
NCH 2728	4,2%
CMMI Nivel 2	3,8%
CMMI Nivel 3	2,9%
NCH 2909	2,5%
ISO – Todas	2,9%
ISO 27001	2,1%
ISO 14000	2,5%
ISO 9001 - 2008	2,1%

Por otra parte, este estudio reconoce cuáles son mercados a los que se dirige la venta de productos software. Un 54% (51% a mercado nacional e internacional más un 3% sólo a mercado internacional) de las empresas exportan sus productos, Tabla 1.5, y un 81% presenta interés en realizar exportaciones, Tabla 1.6.

**Tabla 1.5** Mercados que atienden las empresas encuestadas (estudio ACTI)

Mercados que atiende	Total
Mercado nacional e internacional	51%
Mercado sólo nacional	46%
Mercado sólo internacional	3%

**Tabla 1.6** Interés de exportar por parte de las empresas encuestadas (estudio ACTI)

Interés en exportar	Total
Interesado en exportar	81%
No interesado en exportar	19%

Dado el creciente interés en exportar sus productos, en el mismo estudio se consultó a las empresas que evaluarán cuáles son los aspectos que deben desarrollar para poder realizar exportaciones, Tabla 1.7.

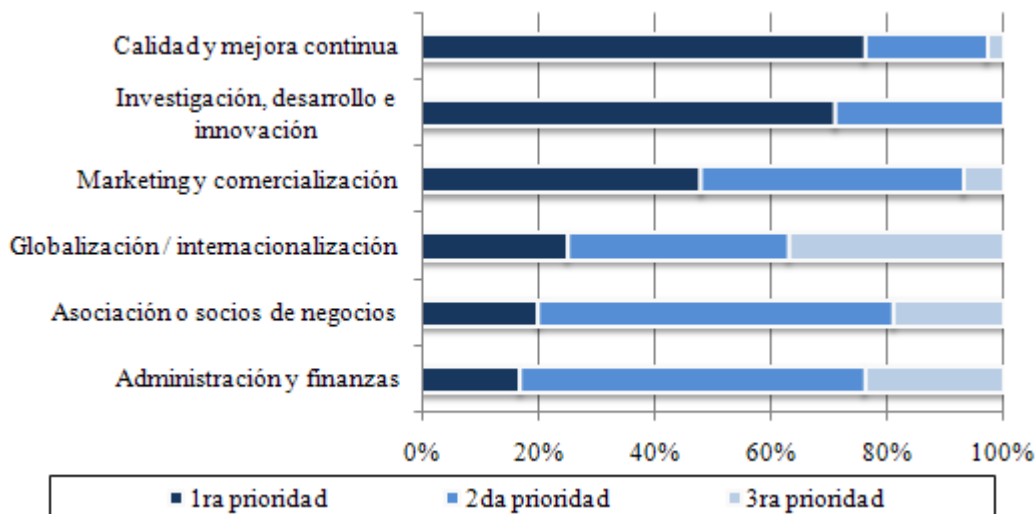
Las empresas evaluaron los siguientes aspectos como los de más alta importancia: generación de redes internacionales (52%), aumentar la fluidez del idioma inglés (49%), certificación internacional (49%), identificar productos y servicios (47%) y aumentar la capacidad de *management* de la empresa (42%). De estos aspectos es claro que se necesitan generar vínculos y crear redes internacionales para promover los productos en el extranjero. También es conocido el bajo dominio del idioma inglés en Chile, por lo tanto no es una sorpresa que aparezca como uno de los aspectos a desarrollar. En tercer lugar se encuentra la necesidad de certificación internacional, para poder competir de igual a igual con organizaciones desarrolladoras de otros países. Además, las certificaciones internacionales permiten representar el buen nivel y la estandarización de los procesos utilizados, permitiendo la comparación con otras organizaciones de la industria del software.

Si se considera la realidad de las PyMEs (alrededor de 2000 según datos de la misma ACTI) que no cuentan con los recursos necesarios para adaptarse a los estándares internacionales, es necesario tomar medidas para que este tipo de organizaciones puedan certificarse de una forma económica y que las mejoras aplicadas evidencien de manera temprana un aumento en la calidad del producto.

**Tabla 1.7** Aspectos a desarrollar para realizar exportaciones (estudio ACTI)

Servicios para el desarrollo de exportación	% importancia		
	Alta	Media	Baja
Plataforma de negocios para la innovación en la Costa Oeste de EE.UU.	23%	25%	52%
Apoyo en la generación de fusiones con otras empresas TIC	19%	28%	53%
Certificación internacional	49%	31%	20%
Aumentar la capacidad de <i>management</i> de la empresa	42%	30%	28%
Apoyo en el desarrollo estratégico del negocio para exportar	29%	32%	39%
Aumento de la capacidad de comercialización internacional	39%	37%	25%
Implementación de buenas prácticas	33%	37%	29%
Desarrollo del capital humano de la empresa	32%	37%	31%
Desarrollar competencias en el personal de la empresa	31%	35%	34%
Aumentar la fluidez en el idioma inglés	49%	26%	25%
Generación de redes internacionales	52%	35%	13%
Establecer redes de asociatividad	36%	38%	26%
Identificar principales tendencias globales en TIC	29%	35%	36%
Identificar productos y servicios	47%	28%	24%
Participar en proyectos de innovación tecnológica	24%	38%	37%
Obtención de financiamiento internacional para proyectos globales de TIC	38%	34%	28%
Proporcionar indicadores de resultados	26%	37%	37%

En el Sexto Diagnóstico de la Industria Nacional de Software y Servicios [GECHS, 2008] se demuestra que, para las empresas pertenecientes al grupo Asociación Gremial de Empresas Chilenas Desarrolladoras de Software – GECHS, los temas de primera prioridad para las organizaciones encuestadas son (Figura 1.1): calidad y mejora continua; investigación, desarrollo e innovación; y marketing y comercialización. En primer lugar está calidad y mejora continua, la que se puede lograr a través de la definición de procesos para las actividades desarrolladas por la empresa.



**Figura 1.1** Principales prioridades dentro de las empresas (GECHS)

La Figura 1.2 muestra los principales beneficios que ha entregado la definición de procesos formales a las empresas del grupo GECHS que han implementado estas prácticas. Dentro de los mayores beneficios que se obtienen están: mejora de imagen, mayor satisfacción al cliente y mejor administración. Dos de estos tres beneficios (mejora de la imagen y mayor satisfacción al cliente) están relacionados con aspectos diferenciadores, que permiten aumentar el valor agregado del producto. Por otra parte, la mejor administración puede traer beneficios económicos directos, pero aún así otros aspectos de este tipo son de menor importancia, como la mejor capacidad de venta y la reducción de costos.

También se consultó a las empresas por las motivaciones más importantes al momento de decidir implementar procesos formales con miras a la certificación. Como resultado se obtuvo que las motivaciones más importantes son: mejorar la competitividad, mejorar la confiabilidad del producto y mejorar la calidad del producto. Nuevamente, se dejan de lado aspectos de tipo económico, como el aumentar las ventas o disminuir las pérdidas. Lo que corrobora que este tipo de prácticas no son aplicadas para obtener beneficios económicos directos, sino para agregar valor al producto y a la propia organización. La Figura 1.3 muestra los aspectos considerados en este informe de diagnóstico.

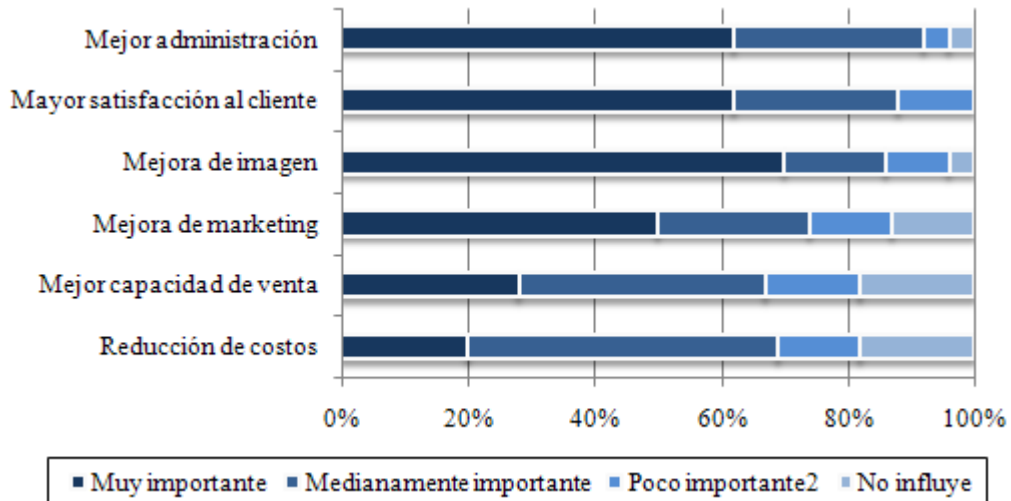


Figura 1.2 Beneficios de implementar procesos formales (GECHS)

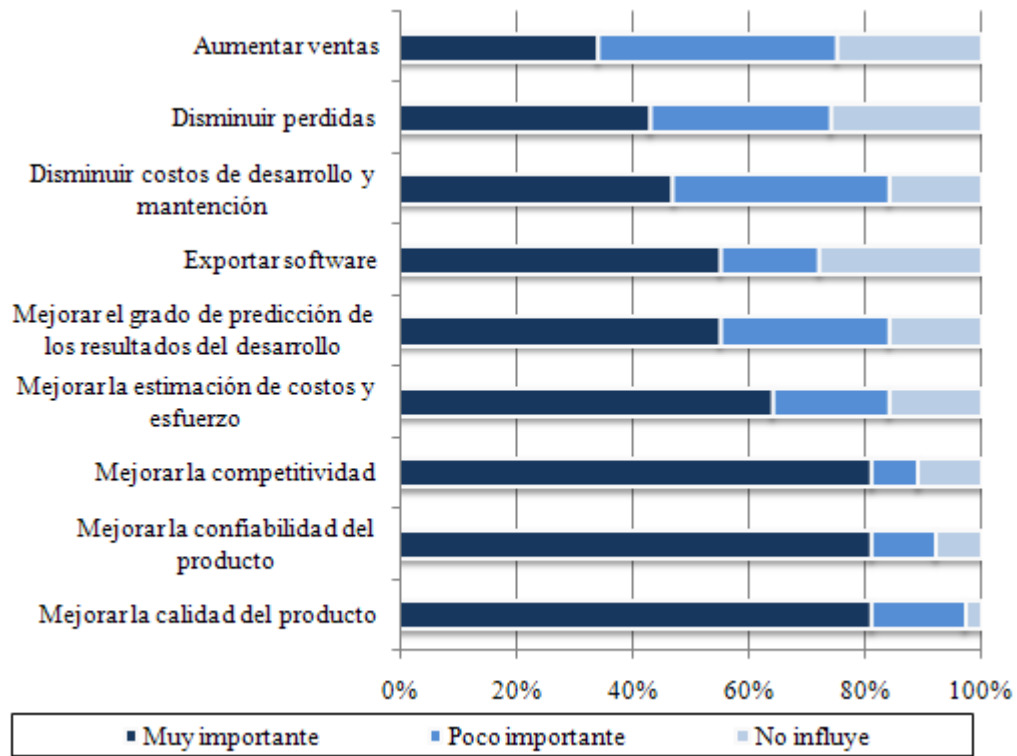
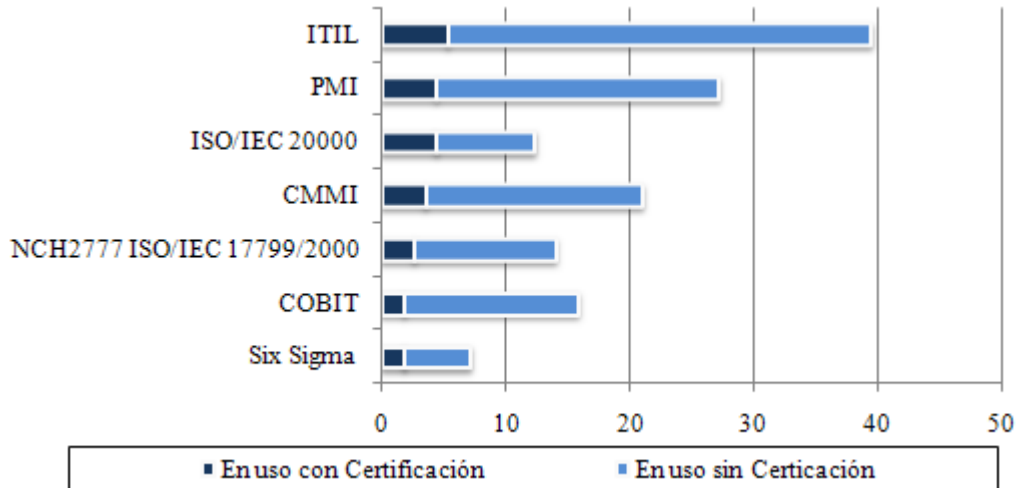
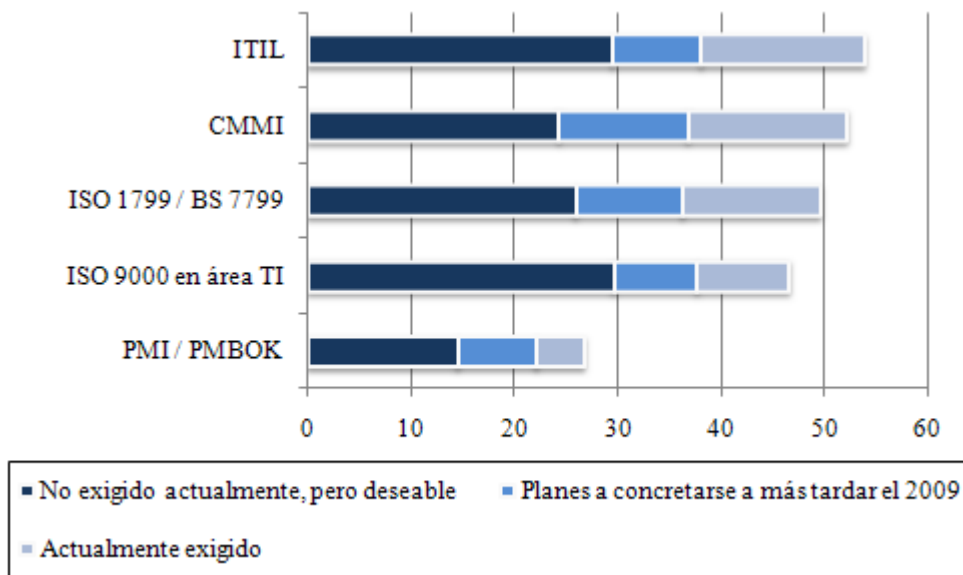


Figura 1.3 Razones que motivarían a implementar procesos formales (GECHS)



**Figura 1.4** Nivel de uso de estándares en gestión de TI (ENTI 2010)

En el Estudio Nacional sobre Tecnologías de la Información – ENTI realizado el 2010 [CETIUC, 2010] se aprecia que la adopción de buenas prácticas por parte de las organizaciones encuestadas es cada vez mayor, pero la certificación aún es baja, tal como se puede observar en la Figura 1.4. Lo que debería fomentarse para poder responder a las exigencias en este mismo aspecto, certificación, por parte de las empresas clientes a sus proveedores. Este tipo de exigencias existe hace un tiempo, lo que se demuestra en el ENTI 2007[CETIUC, 2007], donde se obtuvieron datos sobre las exigencias de certificación (Figura 1.5).



**Figura 1.5** Exigencia de certificación TI a proveedores (ENTI 2007)

Como se puede apreciar en los distintos estudios recién expuestos, existe una preocupación por la implementación de mejoras de procesos en las organizaciones desarrolladoras de software. Esto se ve como una ventaja competitiva que trae consigo una

variedad de beneficios, principalmente de diferenciación. Por otra parte, la mejora de procesos y su certificación es completamente necesaria para poder competir abiertamente en el mercado internacional, más aún si la industria de software en Chile busca expandirse más allá del mercado nacional. Además, las certificaciones se han transformado en una exigencia común de los clientes, esto les asegurará que su proveedor y el producto que compran cumplan con las normas, buenas prácticas y niveles de calidad esperados.

La industria del software es una oportunidad para los países en vías de desarrollo, debido a la alta actividad económica que presenta a nivel mundial [Aspray *et al.*, 2006]. Chile ya ha visto esta industria como un foco que permita el desarrollo económico. Por otra parte la ACTI y sus miembros también auguran que las exportaciones de esta área aumentarán. Pero como ya se ha visto existen trabas para las organizaciones en general, una de ellas es la falta de certificación. Para ello es necesario que se puedan definir los procesos con que se trabaja para establecer mejoras en ellos, es decir, se requiere que la mejora continua sea un pilar dentro de las organizaciones desarrolladoras de software para contar con productos de mayor calidad y, además, poder diferenciarse en la industria.

La implementación de mejoras de procesos y la propia certificación de estándares internacionales es algo muy complejo y difícil de lograr para todo tipo de organizaciones, dado que es un proceso caro y de largo aliento para desarrollarlo de la manera adecuada. Esto se restringe aún más cuando se trata de PyMEs, pues no cuentan con los recursos, ni con los conocimientos necesarios que les permitan implementar estas prácticas. A nivel iberoamericano han surgido diversas propuestas para mejorar los procesos de las organizaciones, como el Modelo de Referencia de Procesos para la Industria Software - MoProSoft en México, el Productivity and Quality Software Program – PBQP-Software en Brasil, SIMEP-SW en Colombia, etc. Así surgió el proyecto Competisoft [Oktaba *et al.*, 2008], el que busca integrar las propuestas nacidas en Iberoamérica que están relacionadas con la mejora de proceso de software para las PyMEs desarrolladoras de software y entregar un marco metodológico, estableciendo modelos que faciliten la adopción de estándares internacionales y que puedan ser utilizados para establecer procesos de mejora.

En este trabajo se buscará brindar apoyo a una PyME (caso práctico) que requiera aplicar mejoras a sus procesos de software. Para ello se utilizará parte del marco metodológico, el modelo de referencia de procesos y el modelo de mejoras, del proyecto Competisoft. Como fase inicial del trabajo se modelará un proceso de la categoría operaciones de Competisoft, en una segunda parte, una vez realizado el diagnóstico respectivo, se comparará este proceso ideal de Competisoft con el utilizado por la organización y, luego, se comenzará con una iteración del ciclo de mejora propuesto en el modelo de mejora de Competisoft. El modelado de procesos se elaborará con el estándar de la Object Management Group – OMG destinado a esta área, el metamodelo de procesos de software SPEM2 (System and Software Process Engineering Metamodel versión 2). La herramienta de modelado de procesos de software Eclipse Process Framework Composer será la utilizada para realizar esta actividad, pues soporta la notación de SPEM2 y, además, posibilitará establecer una base de conocimiento mantenible para la organización, con esto se podrá sacar provecho de todas las ventajas que ofrece la notación y conceptos tras SPEM2.



## **1.2. Objetivos**

El planteamiento de los objetivos está dividido en dos partes. La primera, el objetivo general que engloba todo el trabajo que se desea cumplir. Y la segunda parte corresponde a los objetivos específicos, objetivos de menor alcance, pero que en conjunto permitirán dar por cumplido el objetivo general.

### **1.2.1. Objetivo general**

Realizar mejoras al proceso de mantenimiento de una PyME, como caso de estudio real, apoyándose en el modelado de procesos de software con la notación estándar de modelado de procesos de software SPEM2 y la herramienta de modelado Eclipse Process Framework Composer. Esto bajo la guía del marco metodológico de Competisoft.

### **1.2.2. Objetivos específicos**

- Investigar los conceptos relacionados con la mejora de procesos de software y la notación de modelado de procesos de software SPEM2.
- Describir el modelo de referencia de Competisoft y modelar el proceso de mantenimiento con la herramienta de modelado Eclipse Process Framework Composer.
- Diagnosticar la situación actual del proceso de mantenimiento de una PyME, como caso de estudio.
- Llevar a cabo una iteración del ciclo de mejoras para el proceso de mantenimiento, midiendo los resultados obtenidos.

## **1.3. Estructuración de contenidos**

Este documento presenta cuatro capítulos.

El primero es el actual capítulo introductorio, donde se ha presentado la problemática, los objetivos del trabajo y su motivación.

El marco teórico se expone en la segunda sección. Donde se expondrán los detalles sobre los conceptos que se han investigado y utilizado para poder llevar a cabo la presente investigación.

En el capítulo 3 se presentará el proceso de Mantenimiento Agil\_Mantema, utilizado en Competisoft como uno de sus procesos en la categoría Operación. Así también, se expondrá el modelado detallado de este proceso, el que ha sido realizado con la herramienta de modelado de procesos de software EPF Composer.

La ejecución del proceso de Mejoras aplicado y llevado a cabo en la PyME se dará a conocer en el capítulo 4. Se conocerá: el estado actual de la empresa, gracias a un breve análisis estratégico; los representantes de los roles para el proceso de mejoras; las mejoras que espera la empresa; y la situación inicial del proceso de Mantenimiento, junto a su modelado. Luego se expondrán las mejoras realizadas, el nuevo proceso y el modelado realizado con la

herramienta EPF Composer, junto a las impresiones finales recogidas del interior de la empresa.

Para finalizar este documento, la sección 5 presentará las conclusiones del trabajo realizado, destacando los aspectos positivos de éste, reconociendo y reflexionando sobre los aspectos negativos agregando posibles soluciones a éstos y se referirá al cumplimiento de los objetivos planteados junto con manifestar los desafíos que se ven próximos para futuros trabajos.

## **2. Marco Metodológico**

---

En esta sección se presentarán los conceptos que forman la base para este trabajo y que han sido utilizados como marco teórico y de referencia. Estos conceptos son: procesos de software; ingeniería de procesos de software; modelado de procesos de software; mejoramiento de procesos de software, haciendo especial hincapié en Competisoft; notaciones para el modelado de procesos de software, centrándose en SPEM2; y una vista a las herramientas de modelado de procesos de software, en particular a Eclipse Process Framework Composer– EPFC.

### **2.1. Procesos de software**

Un proceso es un conjunto de actividades sucesivas y coordinadas que se realizan con un fin determinado. Los procesos son utilizados en distintos ámbitos y con diversos objetivos, y generalmente están vinculados al ámbito tecnológico, o a la informática, para automatizar su gestión.

En la informática los procesos de software se han transformado en un factor clave para el desarrollo de aplicaciones, por la creciente complejidad que esta actividad presenta, porque se deben coordinar esfuerzos creativos que, principalmente, dependen de las personas, organización y procedimientos utilizados.

#### **2.1.1. Orígenes de los procesos de software**

Durante las décadas de los '60 y '70 los estudios y prácticas tenían como objetivo definir ciclos de vida de software para especificar las diferentes etapas en la línea de vida de un software. Las fases típicas son: análisis, diseño, implementación, pruebas, implantación, mantenimiento y retiro. Aun así, los distintos modelos de ciclo de vida sólo definen un esqueleto y filosofía a seguir, pero no establecen cursos de acción precisos, ni algún tipo de organización sobre los equipos de trabajo, ni procedimientos de operación o políticas de desarrollo, tampoco consideran las herramientas que se utilizan o como deben utilizarse. Por lo que adoptar un ciclo de vida como guía para llevar un control sobre la gestión de un proyecto de software no es suficiente.

La definición de ciclos de vida significó la primera aproximación hacia la noción de procesos de software, un marco teórico y práctico que permite organizar diferentes aspectos que giran alrededor del desarrollo de software, como lo son: las personas, los métodos y procedimientos, y las herramientas y equipamiento.

A fines de los años '80 el Departamento de Defensa (Department of Defense – DoD) de Estados Unidos tenía problemas con las empresas contratistas a las que encargaba el desarrollo de software, pues las estimaciones de costo y tiempo que éstas realizaban no eran precisas. Para mejorar esta situación el DoD encargó al Software Engineering Institute (SEI), perteneciente a Carnegie Mellon University, la creación de un modelo que permitiera evaluar el nivel de madurez de los procesos utilizados por las empresas en el desarrollo de software. El resultado fue denominado Capability Maturity Model – CMM el que establece un conjunto de procesos con sus debidas buenas prácticas, las que deben ser definidas, tener los recursos

necesarios para llevarse a cabo, ejecutadas, medidas y verificadas. Con CMM se dio un paso importante hacia el uso de los procesos de software para la implementación de mejoras en la calidad de los procesos de producción de software.

La noción de proceso de software se construye sobre el concepto de ciclo de vida del software y la idea de establecer un marco conceptual que permita organizar los distintos factores y actividades alrededor del desarrollo para mejorarlos.

### **2.1.2. Definiciones y conceptos de procesos de software**

Existen diversas definiciones de procesos de software, las más importantes que se pueden encontrar en la literatura son:

Lonchamp [1993] establece un marco conceptual y terminológico para la ingeniería de procesos de software. Allí un proceso es definido como: un conjunto de pasos del proceso parcialmente ordenado, relacionado con un grupo de artefactos, recursos humanos y computarizados, estructuras organizacionales y limitaciones, destinados a producir y mantener las prestaciones del software solicitado.

Por otra parte, Conradi *et al.* [1994] define un proceso de software como: todos los elementos del mundo real envueltos en el desarrollo y mantención de un producto (artefactos, herramientas de soporte para la producción, actividades, actores y procesos de soporte). Está compuesto por un proceso de producción, un meta proceso y un proceso de soporte que sirve a estos últimos dos (con notaciones y métodos de modelado de procesos, modelos de proceso y herramientas de proceso).

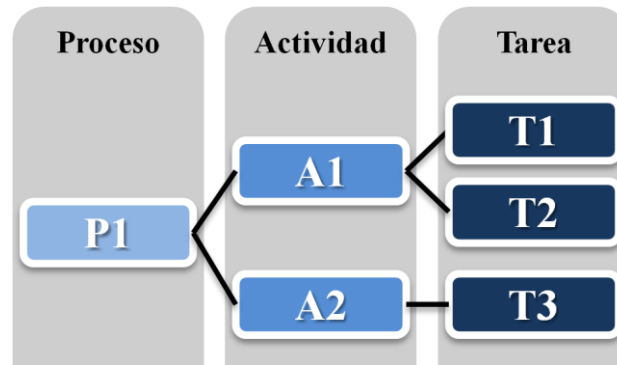
La norma ISO/IEC 12207 [1995], define un proceso como: un conjunto de actividades interrelacionadas, que transforma entradas en salidas. Además, un proceso está compuesto de actividades, éstas son un conjunto detallado de tareas y las tareas son acciones con entradas y salidas (Figura 2.1). Esta definición es bastante general, pues es implementada por otras normas ISO.

En cambio Fuggetta [2000] hace una definición mucho más precisa y acorde de lo que es un proceso de software: un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software.

De esta última definición se puede entender que los procesos de software explotan diversos aspectos y conceptos:

- *Tecnología de desarrollo de software:* contar con la tecnología apropiada para que la actividad de desarrollo de software sea fiable económicamente y se lleve a cabo de la mejor manera, es decir, se necesitan herramientas, infraestructura y ambientes acordes para esto.
- *Métodos y técnicas para el desarrollo de software:* utilizar metodologías que guíen el desarrollo para utilizar los recursos tecnológicos con que se dispone efectivamente.
- *Comportamiento organizacional:* el desarrollo de software es efectuado por equipos de personas, los que deben ser gestionados efectivamente por una estructura organizacional.

- *Marketing y economía:* un software es un producto que debe ser vendido, por ello es que debe estar ajustado a la medida de los clientes o mercado objetivo al que pretende ser vendido, atendiendo a sus necesidades.



**Figura 2.1** Estructura de proceso ISO/IEC 12207

Con esto en vista, se puede decir que los procesos de software son complejos, pues:

- *No son como los procesos productivos industriales:* los procesos de software son muy distintos a los conocidos en el ambiente industrial, dado que: están dirigidos por excepciones al ser realizados, en su mayoría, por personas; pueden estar determinados por situaciones impredecibles; y cada uno cuenta con sus particularidades.
- *No son proceso de ingeniería “pura”:* los elementos ingenieriles no están plasmados en este tipo de procesos, ya que no se conocen las abstracciones adecuadas para ellos, dependen en gran medida de la gente que los aplica o crea y los presupuestos, tiempos y calidad no pueden ser estimados de manera fiable.
- *Están basados en los descubrimientos que dependen de la comunicación, coordinación y cooperación existente en el marco de trabajo establecido:* hay aspectos que se descubren a medida que se lleva a cabo el desarrollo y que deben ser informados para ser gestionados, como nuevos requerimientos que pueden surgir por parte del usuario, si está involucrado, o de entregables intermedios.

Estos puntos están directamente relacionados con la calidad del producto que se desarrolla, además los sistemas que se crean son cada vez más grandes y complejos. Por ello es que muchas empresas expresan un creciente interés en la mejora de procesos de software, para que esto se vea reflejado en productos de mejor calidad. Para ello es necesario que los procesos de software cumplan con requisitos de calidad como lo son: correcta definición, estar alineados e ir mejorando en función de los objetivos del negocio y producir los resultados esperados. Esto se puede conseguir a través de la Gestión de Procesos de Software. Para poder llevar a cabo una buena gestión es necesario disminuir la complejidad que presentan los procesos, esto es posible gracias al modelado de procesos de software.

## 2.2. Modelado de procesos de software

El modelado de procesos de software utiliza los beneficios de los modelos para generar múltiples representaciones de la realidad, es decir, abstracciones mucho más simples, que

evitan cualquier daño o irreversibilidad que se pueda causar trabajando directamente sobre el mundo real [Rothenberg, 1989]. Esto permite centrar la atención en los aspectos de los procesos que se quieren estudiar con la opción de agregar algún soporte automatizado para facilitar dicha tarea.

La utilización de modelos de procesos de software implica una serie de beneficios:

- *Fácil entendimiento*: un modelo debe representar de forma precisa la estructura y organización de los procesos, incluyendo la información necesaria.
- *Entrenamiento y educación*: la formalización precisa de los procesos permite su utilización como una herramienta más para el entrenamiento de personal recién contratado.
- *Soporte y control de la gestión de procesos*: los procesos pueden ser usados en diferentes niveles y proyectos según sea necesario, éstos pueden ser monitoreados, coordinados y gestionados.
- *Soporte a la mejora de procesos*: permite conocer y evaluar problemas, así también se pueden evaluar los procesos que sirven como alternativas, aquellos procesos efectivos que pueden ser reutilizados o desarrollar nuevos procesos.
- *Soporte a la automatización*: permite la ejecución de partes de procesos y guía de éstos automáticamente. Además, es posible incorporar y almacenar métricas de los procesos para realizar evaluaciones.

### 2.2.1. Elementos básicos de los modelos de procesos de software

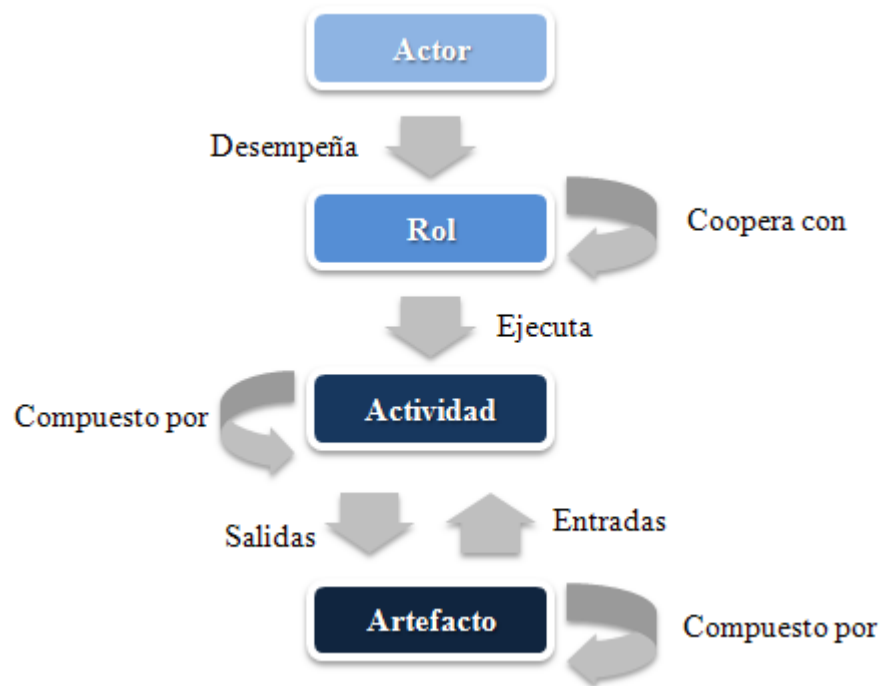
Existen muchos aspectos de los procesos de software que podrían ser modelados, los más comunes son: actividades, roles, pasos de procesos, estructuras, artefactos, herramientas, agentes, proyectos, soporte de producción, etc. [Lonchamp, 1993], [Conradi *et al.*, 1994]. De estos elementos los más comunes y relevantes son definidos a continuación, según lo expuesto por Acuña y Ferré [2001]:

- *Actor*: es la entidad que ejecuta el proceso, está caracterizado por las propiedades del rol que desempeña y sus habilidades, puede tener varios roles que realizan una actividad. Se agrupan en dos categorías: a) actores humanos, son las personas que están involucradas con el proceso de software y pueden ser organizados en equipos; b) actores sistemas, son el hardware o software que compone los procesos, pertenecen al dominio de soporte tecnológico.
- *Rol*: describe un grupo de responsabilidades, derechos y habilidades necesarias para cumplir con una actividad de proceso de software. Puede haber cooperación entre distintos roles. Es la asociación entre los actores y las actividades en términos del conjunto de responsabilidades que deben ejecutar los actores.
- *Actividad*: es la etapa del proceso, realizada por un actor, que produce cambios visibles externamente en el estado del producto software. Incluye e implementa procedimientos, reglas, políticas y metas para generar y modificar un conjunto de artefactos. Tiene entradas, salidas y resultados intermedios, generalmente productos terminados como documentos de especificación de requerimientos, esquemas de base de datos, etc. Las actividades pueden estar organizadas en redes, donde la dimensión horizontal es comprendida como el encadenamiento de actividades y la dimensión vertical se entiende

como la descomposición de éstas. Las actividades están asociadas con roles, otras actividades y artefactos.

- *Artefacto*: es el (sub) producto y la materia prima de un proceso. Un artefacto producido por un proceso luego puede ser utilizado por el mismo u otro proceso como materia prima para producir otro artefacto. Un artefacto es desarrollado y mantenido en un proceso. Un artefacto puede tener un largo tiempo de vida útil, por lo tanto habrán diferentes versiones de cada artefacto como procesos de software. El (sub)producto puede ser creado, utilizado o modificado durante una actividad. La agregación de artefactos de software es llamada producto software.

En la Figura 2.2 se muestran las relaciones existentes entre los elementos básicos del modelado de proceso de software.



**Figura 2.2** Relación entre componentes básicos

Un actor puede desempeñar uno o más roles según sea conveniente. Un rol ejecuta una actividad en término de las responsabilidades que esta conlleva, además puede haber cooperación entre distintos roles relacionados por la actividad que se quiere ejecutar. Una actividad está compuesta por un encadenamiento de éstas y por su descomposición (dimensión horizontal y vertical, respectivamente, de la organización en red), por otra parte los resultados (salidas) que se obtienen de una actividad, artefactos, pueden ser las entradas de otra actividad. Por esto último, un artefacto puede estar compuesto por uno o más artefactos intermedios, que han sido resultado de otras actividades.

### 2.2.2. Tipos de modelos de procesos de software

En la literatura se pueden encontrar diversos tipos y clasificaciones para los modelos de procesos de software dependiendo de los objetivos o la perspectiva que se está modelando.

Curtis *et al.* [1992] presenta la siguiente clasificación para las perspectivas de modelos de procesos de software.

- *Funcional*: representa cuáles actividades deben ser ejecutadas y cuáles entidades de los flujos de información son importantes para éstas.
- *Comportamiento*: representa cuándo y bajo qué condiciones las actividades deben ser implementadas.
- *Organizacional*: representa cuándo y por quién de la organización las actividades deben ser ejecutadas.
- *Informacional*: representa las entidades de información que da como salida o las que son manipuladas por un proceso, incluyendo su estructura y relaciones.

Por otra parte, Lonchamp [1993] presenta una taxonomía para los modelos de procesos de software. Según su disposición un modelo puede ser:

- *Descriptivo*: estudia procesos existentes con el objetivo de conocer como se está desarrollando software actualmente en la empresa u organización. Responde a la pregunta “¿Cómo es o ha sido desarrollado el software actualmente?”.
- *Preceptivo*: define los procesos para llevar a cabo el desarrollo de software de la manera que se desea. Responde a la pregunta “¿Cómo debería ser desarrollado el software?”.

A su vez estas disposiciones apuntan a:

- *Expresar*: describir el proceso actual o deseado (según sea el caso) formalmente para su entendimiento, comunicación, educación, reutilización o estandarización.
- *Analizar*: estudiar la descripción del proceso actual o deseado (según sea el caso) a través de alguna técnica formal para obtener un mayor entendimiento, comparación, mejoramiento, análisis de impacto o previsión.

Además, los modelos de disposición preceptiva también pueden:

- *Guiar*: permite entregar apoyo indirecto a quienes ejecutan los procesos de software, a través de información de ayuda, como el estatus del proceso, la siguiente etapa que debe ser ejecutada, etc. Esto es provisto por la interpretación manual o automática de la descripción de los procesos de software.
- *Forzar*: entrega apoyo directo a quienes ejecutan los procesos de software. Las actas de descripción de procesos de software son interpretadas por motores de procesos pertenecientes a un ambiente centrado en la ingeniería de procesos que permite orquestar el funcionamiento del desarrollo y automatizarlo lo más posible. Lo que no puede ser programado para la interpretación automática se deja en manos de personas.

Los tipos de modelos que se construyen dependen de la notación, abstracción y formalismo utilizado para representar la información relativa a las características presentadas. Aun así, no se pueden cubrir todas estas características, sin embargo las notaciones para el modelado están orientadas hacia alguna de ellas.



## 2.3. Ingeniería de procesos de software

Para conseguir los resultados esperados al mejorar los modelos de proceso de software es necesario utilizar una perspectiva ingenieril. Para ello ha surgido la ingeniería de procesos de software como un área de la ingeniería de software especialmente enfocada a solucionar los problemas que suscitan los procesos utilizados en el desarrollo de software.

La ingeniería de procesos de software puede ser vista desde dos perspectivas [SWEBOK, 2004]. La primera abarca las actividades técnicas y de gestión de los procesos del ciclo de vida del software, como la adquisición de software, desarrollo, mantenimiento y retiro. La segunda está interesada en la definición, implementación, valoración, medición, gestión, cambio y mejora de los procesos del ciclo de vida de software. Este segundo es el que se ocupa normalmente como definición de ingeniería de proceso de software.

En SWEBOK [2004] se establecen las cuatro sub-áreas de las que se preocupa la ingeniería de procesos de software: implementación de procesos y cambio, definición de procesos, evaluación de procesos y medición de procesos y productos, las que son detalladas a continuación.

### 2.3.1. Implementación de procesos y cambio

Esta sub-área está enfocada hacia la descripción de actividades, modelos, infraestructura y consideraciones prácticas para la implementación y cambio, principalmente ante la situación de un primer desarrollo o cambio de procesos.

Para poder establecer los procesos de software de la manera correcta es necesario contar con el personal competente, herramientas para ello y el financiamiento, además se deben definir claramente y asignar los deberes y responsabilidades para asegurar el éxito de los cambios, es decir, debe haber la infraestructura adecuada para llevar a cabo la actividad de mejoras.

McFeeley [1996] establece como foco central de la infraestructura al Grupo de Ingeniería de Procesos de Software (Software Engineering Process Group - SEPG) [Fowler y Rifkin, 1990]. El SEPG es quien asigna las responsabilidades para facilitar las tareas y coordinación de las mejoras de proceso de software. No es un ente ejecutor de tareas, sino un facilitador de éstas y es el guía del proceso de mejora.

Una vez que se cuenta con la infraestructura se deben llevar a cabo las actividades de mejora, para ello existe un ciclo de gestión, el que está compuesto por cuatro actividades secuenciales e iterativas.

- *Establecer la infraestructura:* buscar el compromiso y recursos para la implementación y cambios de procesos, además de poner a disposición la infraestructura necesaria para llevar esto a cabo.
- *Mejorar los procesos:* conocer y entender los procesos existentes como también las necesidades y objetivos de la organización, junto con sus fortalezas y debilidades para realizar un plan de acción adecuado.
- *Ejecutar cambios:* llevar a cabo el plan de implementación de las mejoras de procesos de software.

- *Controlar*: evaluar los resultados de la implementación de las mejoras, lo que permitirá usar esto como entrada para un siguiente ciclo de mejora.

### **2.3.2. Definición de procesos**

Son los procedimientos, políticas o estándares que permiten: mejorar la calidad del producto, facilitar el entendimiento y la comunicación de los procesos, soportar la gestión y mejora de procesos, proveer guías y ejecutar automáticamente algunos procesos.

La forma en que se definen los procesos puede variar según el objetivo que se busca y de la naturaleza del trabajo, como la aplicación del dominio, el modelo de ciclo de vida que se utiliza y el nivel de madurez de los procesos que emplea la organización.

El modelo de ciclo de vida de software es una definición de alto nivel para las fases que ocurren durante el desarrollo, es decir, brinda una guía general sobre las actividades principales y sus relaciones, no una definición detallada de los procesos que se utilizan. Complementario a este modelo, los procesos del ciclo de vida del software entregan un detalle mucho mayor, definiendo roles, actividades, tareas y responsabilidades. Para describir y detallar los elementos de los procesos de software se debe elegir la notación adecuada que se utilizará según los objetivos, la información y nivel abstracción que se requiere.

Además de los procesos de software que se quieren implementar, pueden existir procesos o estándares que se utilizan en la organización que pueden ser reutilizados, pero que no cumplen con el detalle necesario. Por ello deben ser adaptados para tener la estructura deseada, cumplir con las necesidades y estar dentro del contexto organizacional.

La definición y modificación de herramientas existentes que soporten la automatización de procesos es una actividad importante que puede contribuir a la definición de procesos y a ayudar, como guía, con los procesos ya definidos.

### **2.3.3. Evaluación de procesos**

Esta área de la ingeniería de procesos de software se lleva a cabo usando un modelo de evaluación y un método de evaluación.

Los modelos de evaluación de procesos permiten a la empresa evaluar su capacidad de desarrollo contra un conjunto reconocido de buenas prácticas. Así el número de prácticas, la maestría con que se manejan y el nivel de integración organizacional en el desarrollo determina la valoración que obtiene la empresa. Hay dos arquitecturas que hacen distintas suposiciones sobre el orden en que los procesos deben ser evaluados: evaluación continua o evaluación escalonada. La elección de cuál aplicar depende de los objetivos que la empresa desea alcanzar.

Existen diversos modelos de evaluación, entre los más reconocidos se puede encontrar: CMMI, ISO 9001, ISO/IEC 15504, entre otros.

Para poder realizar una evaluación es necesario seguir un método específico de evaluación, el que producirá una puntuación cuantitativa que caracterizará las capacidades o

nivel de madurez de los procesos utilizados en la empresa. Estas guías para la evaluación son diversas y están directamente relacionadas con los modelos de evaluación y mejoras.

### **2.3.4. Medición de procesos y productos**

Para lograr una mejora en los procesos y producto de software es necesario establecer una línea base de mediciones, pues la medición provee el soporte para la iniciación o evaluación del procesos de implementación y cambios, ayudando a identificar los procesos anómalos que generan cuellos de botella o un mayor número de fallas. Además, si se lleva el registro de las mediciones hechas y los datos que éstas arrojaron se pueden reunir los datos necesarios que permitan realizar predicciones a las características estudiadas para determinar su comportamiento futuro.

La realización de la medición puede ser utilizada para la gestión de un proyecto de desarrollo o, lo que interesa en este caso, detectar las fortalezas y debilidades de los procesos, para luego evaluar esos procesos que han sido implementados o cambiados.

Para esta medición se toman en cuenta tres perspectivas:

- *Tamaño:* se mide el tamaño del producto de software a través de su largo (en líneas de código) o funcionalidades (punto función).
- *Estructura:* busca medir los flujos de control, flujos de datos, anidamientos, estructuras de control, estructuras modulares e interacción.
- *Calidad:* es necesario medir los aspectos multidimensionales de la calidad, éstos dependen del modelo de calidad que se está utilizando.

Finalmente, se debe conocer la calidad de las mediciones obtenidas, para conocer su precisión, reproducibilidad, repetitividad, convertibilidad y errores aleatorios. Esto es esencial para proveer de resultados exactos y acotados.

## **2.4. Mejoramiento de procesos de software**

Tal como se ha mencionado anteriormente, existe un estrecho vínculo entre la calidad del proceso de desarrollo y la calidad del producto de software. Si el objetivo de la empresa es aumentar la calidad de sus productos software, es necesario fijar la mirada y realizar mejoras a los procesos existentes o implementar nuevos procesos para ser utilizados en el desarrollo. El mejoramiento de procesos es una tarea sumamente compleja pues requiere de la experiencia ganada en ensayos del tipo prueba y error, cosa que consume muchos recursos, como tiempo y costos. Esta necesidad de experiencia puede ser, de cierta forma, proporcionada por algún modelo de mejoramiento de procesos de software, ya que han sido desarrollados en base a las experiencias de diversas empresas que han llevado a cabo dicho mejoramiento. Algunos de estos modelos serán brevemente presentados a continuación y, luego, se detallará en más profundidad el marco metodológico del proyecto Competisoft.

### **2.4.1. Capability maturity model integration – CMMI**

CMMI es un marco de trabajo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas software que surge como la evolución del modelo CMM, que sólo abordaba los procesos de software, y la integración con otros

modelos. Las áreas de interés de CMMI son: desarrollo de productos y servicios; establecimiento, manejo y entrega de servicios; y adquisición de productos y servicios. Las prácticas incluidas para estas áreas deben ser adaptadas a los objetivos de negocio de la empresa para que las mejoras tengan éxito.

#### **2.4.1.1. Elementos principales**

El modelo para implementar mejoras de procesos en el desarrollo de software contiene 22 áreas de procesos que describen los aspectos de desarrollo de producto que deben ser cubiertos por los procesos organizacionales, estas áreas son:

- Análisis causal y resolución (CAR)
- Administración de configuración (CM)
- Análisis de decisiones y resolución (DAR)
- Administración integrada de proyectos (IPM)
- Medición y análisis (MA)
- Definición organizacional de procesos (OPD)
- Enfoque organizacional de procesos (OPF)
- Administración organizacional de rendimiento (OPM)
- Rendimiento organizacional de procesos (OPP)
- Entrenamiento organizacional (OT)
- Integración de productos (PI)
- Monitoreo de proyectos y control (PMC)
- Planeación de proyectos (PP)
- Aseguramiento de calidad de procesos y productos (PPQA)
- Administración cuantitativa de proyectos (QPM)
- Desarrollo de requerimientos (RD)
- Administración de requerimientos (REQM)
- Administración del riesgo (RSKM)
- Administración de acuerdos de proveedores (SAM)
- Solución técnica (TS)
- Validación (VAL)
- Verificación (VER)

Cada área de proceso posee una serie de elementos, los más importantes son:

- *Declaración de propósitos:* describe el propósito de un área de proceso y actúa como componente informativo.
- *Notas introductorias:* describe los principales conceptos cubiertos por un área de proceso y actúa como componente informativo
- *Áreas de procesos relacionadas:* es una lista que referencia a las áreas de procesos relacionadas y, desde un alto nivel, refleja sus relaciones.
- *Metas específicas:* describe las características únicas que deben estar presentes para satisfacer un área de proceso. Una meta específica es un componente requerido del modelo y es utilizado en evaluaciones para ayudar a determinar cuando un área de proceso es satisfecha.
- *Metas genéricas:* son llamadas “genéricas” porque la misma meta es aplicada a múltiples áreas de proceso. Una meta genérica describe las características que deben

estar presentes para institucionalizar procesos que implementa un área de proceso. Una meta genérica es un componente requerido del modelo y es utilizado en evaluaciones para ayudar a determinar cuando un área de proceso es satisfecha.

- *Prácticas específicas:* es una descripción de una actividad que es considerada importante para el logro de una meta específica asociada. Las prácticas específicas describen las actividades esperadas como resultado del cumplimiento de metas específicas de un área de proceso.
- *Prácticas generales:* son llamadas “genéricas” porque la misma práctica es aplicada a múltiples áreas de proceso. Las prácticas genéricas asociadas con una meta genérica describen las actividades que son consideradas importantes para el logro de una meta genérica y contribuye a la institucionalización de procesos asociados con el área de proceso.

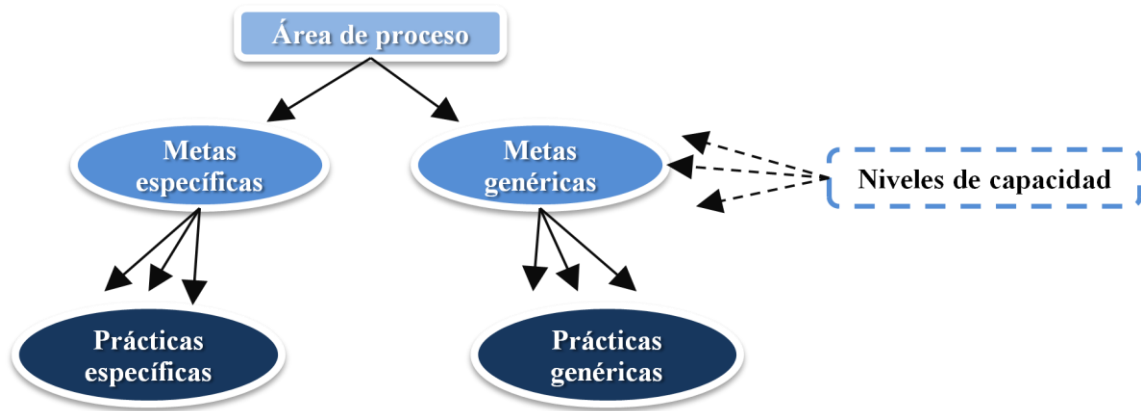
#### **2.4.1.2. Estructura de representación continua y por etapas**

CMMI soporta dos caminos para la mejora utilizando niveles, estos caminos son la representación continua y por etapas. La representación continua permite a la organización mejorar incrementalmente procesos correspondientes a un área de proceso en particular (o un grupo de áreas de proceso) seleccionada por la organización, lo que se permite cumplir con los niveles de capacidad. Por su parte, la representación por etapas mejora incremental y sucesivamente un conjunto de procesos relacionados de un grupo de áreas de proceso, lo que permite lograr los niveles de madurez.

Para lograr un nivel en particular, la organización debe satisfacer todas las metas del área de proceso o del conjunto de áreas de proceso a las que apunta la mejora, sin importar si es un nivel de capacidad o de madurez.

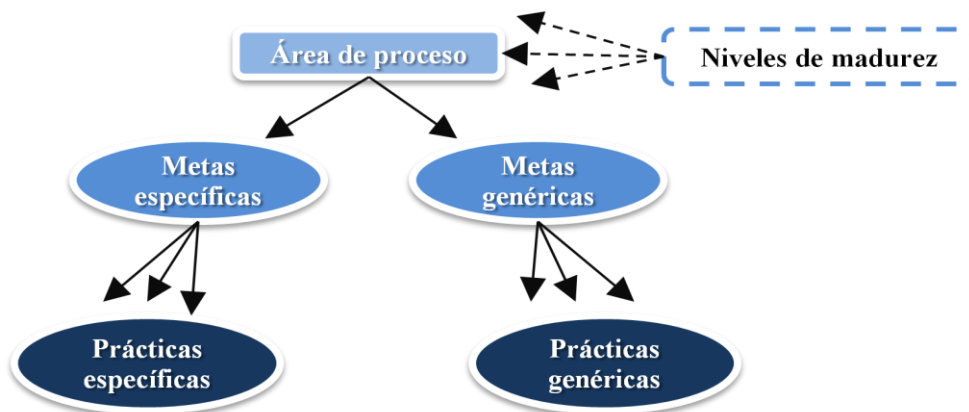
Ambas representaciones proveen caminos para mejorar los procesos para lograr objetivos de negocio y ambos proveen el mismo contenido esencial y usan los mismos componentes del modelo.

Los niveles de capacidad aplican la representación continua para lograr mejoras de procesos de una organización en un área de proceso individual. Estos niveles son el medio para la mejora incremental de los procesos de un área de proceso dada. Cuenta con cuatro niveles, numerados desde 0 hasta 3.



**Figura 2.3** Representación continua

Los niveles de madurez aplican la representación por etapas para lograr mejoras de procesos de una organización a través de múltiples áreas de proceso. Estos niveles son el medio para mejorar incrementalmente los procesos correspondientes a un conjunto de áreas de proceso. Cuenta con cinco niveles, numerados desde 1 hasta 5.



**Figura 2.4** Representación por etapas

En la Tabla 2.1 se encuentra la comparación entre los niveles de capacidad y de madurez.

**Tabla 2.1** Comparación niveles de capacidad y madurez

Nivel	Niveles de capacidad (representación continua)	Niveles de madurez (representación por etapas)
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		Optimizado

### 2.4.1.3. Niveles de capacidad

Los niveles de capacidad son utilizados para valorar los procesos de una organización que optó por la representación de mejora continua. Para cumplir con un nivel de capacidad es necesario que todas las metas genéricas sean satisfechas. En la medida que las meta y las prácticas asociadas se vayan cumpliendo se obtendrán beneficios en la mejora de procesos para esa área.

Los niveles de capacidad se miden en una escala de 0 a 3. Cada nivel se entiende como:

0. *Incompleto*: es un proceso ejecutado parcialmente o, simplemente y llanamente, no es ejecutado. Una o más metas específicas del área de procesos no son satisfechas y no existen metas genéricas para este nivel, por lo que no hay razón para institucionalizar un proceso parcialmente realizado.
1. *Realizado*: es un proceso que cumple con las necesidades del trabajo para producir productos de trabajo; las metas específicas del área de proceso son satisfechas.
2. *Gestionado*: es un proceso planeado y ejecutado en concordancia con las políticas de la organización; los empleados son personas hábiles y adecuadas para producir salidas controladas; se involucra a los stakeholders relevantes; es monitoreado, controlado y revisado; y el apego a la descripción del proceso es evaluada.
3. *Definido*: es un proceso adecuado desde el conjunto de procesos estándar definidos por las directrices de la organización; su descripción es mantenida; y contribuye con la experiencia relacionada a los procesos organizacionales.

### 2.4.1.4. Niveles de madurez

Un nivel de madurez ayuda a valorar los procesos en la representación de mejoras por etapas. Consiste en prácticas específicas y genéricas relacionadas a un conjunto predefinido de áreas de proceso para mejorar el desempeño de toda la organización.

Los niveles de madurez de una organización proveen un camino para caracterizar su desempeño. La experiencia ha demostrado que las organizaciones lo hacen mejor cuando centran sus esfuerzos de mejora en un número manejable de áreas de proceso al mismo tiempo y esas áreas requieren aumentar su sofisticación para las mejoras de la organización.

Define una evolución escalada para la mejora de procesos organizacional. Cada nivel de madurez es un subconjunto importante de procesos de la organización, preparado para avanzar hacia el próximo nivel de madurez. Los niveles de madurez son medidos según el cumplimiento de las metas específicas y genéricas asociadas con cada conjunto de áreas de procesos predefinido.

La escala de los niveles de madurez va desde 1 a 5, los que se describen como:

1. *Inicial*: usualmente son procesos ad hoc y caóticos. Usualmente, la organización no provee del ambiente adecuado para el soporte de los procesos. El éxito en esta organización depende de las competencias y el heroísmo de la gente y no de los procesos provistos. A pesar del caos, las organizaciones de nivel 1 a menudo producen productos y servicios que funcionan, pero que frecuentemente exceden los presupuestos y la planificación propuesta. Estas organizaciones se caracterizan por tender a sobre comprometerse, abandonar sus procesos en tiempo de crisis y no poder repetir sus éxitos.
2. *Gestionado*: los proyectos aseguran que sus procesos son planeados y ejecutados en concordancia con las políticas; los proyectos emplean personal calificado que cuentan con los recursos para producir salidas controladas; involucra a los stakeholders relevantes; son monitoreados, controlados y revisados; y se evalúa el apego a la descripción de los procesos. La disciplina de los procesos reflejada en el nivel de madurez 2 ayuda a asegurar la existencia de prácticas que son retenidas aún en tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se desempeñan y manejan de acuerdo a su planificación propuesta. Además, en este nivel el estado del trabajo es visible para los administradores como puntos definidos. Se ha establecido el compromiso a través de los stakeholders relevantes y son revisados cuando sea necesario. Los productos de trabajo son controlados apropiadamente. Los productos y servicios cumplen con la especificación de sus procesos, estándares y procedimientos.
3. *Definido*: son procesos bien caracterizados y entendidos, además son descritos a través de estándares, procedimientos, herramientas y métodos. El conjunto estándar de procesos organizacionales ha sido establecido y es mejorado a través del tiempo. Estos procesos estándar son utilizados para establecer consistencias a través de la organización. Los proyectos establecidos definen procesos confeccionados desde el conjunto estándar de procesos organizacionales de acuerdo a las guías de confección. A diferencia del nivel 2; donde los estándares, descripciones de procesos y procedimientos pueden ser distintos en cada instancia específica del proceso; en este nivel los estándares, descripciones de procesos y procedimientos para un proyecto son confeccionados desde el conjunto estándar de procesos organizacionales para adaptarse a un proyecto en particular o una unidad organizacional y, además, ser más consistente, excepto por las diferencias permitidas según las guías de confección. Otra diferencia crítica entre ambos niveles, es que en el nivel de madurez 3 los procesos son descritos con mayor rigurosidad que en el nivel 2. Un proceso definido establece claramente su propósito, entradas, criterios de entrada, actividades, roles, medidas, pasos de verificación, salidas y criterios de salida. En este nivel, los procesos son administrados proactivamente, comprendiendo las interrelaciones entre las actividades del proceso y las medidas detalladas del proceso, sus productos de trabajo y sus servicios.



4. *Gestionado cuantitativamente:* la organización y los proyectos establecen objetivos cuantitativos para la calidad y el desempeño de procesos, y los utilizan como criterio para la administración de proyectos. Los objetivos cuantitativos están basados en las necesidades del cliente, el usuario final, la organización y los procesos implementados. La calidad y el desempeño de procesos son entendidos como términos estadísticos y son administrados a través del ciclo de vida del proyecto. Una diferencia crítica entre los niveles de madurez 3 y 4 es la previsibilidad del desempeño de procesos. En el nivel de madurez 4, el desempeño de los proyectos y de los subprocesos seleccionados es controlado a través de técnicas estadísticas y otras técnicas cuantitativas, y las predicciones están basadas, en parte, en el análisis estadístico de datos refinados de los procesos.
5. *Optimizado:* la organización mejora continuamente sus procesos basados en el entendimiento cuantitativo de los objetivos de negocio y las necesidades de desempeño. La organización utiliza un enfoque cuantitativo para comprender las variaciones inherentes del proceso y las causas de sus resultados. Se enfoca en la mejora continua del desempeño de procesos a través de mejoras incrementales e innovaciones en los procesos y la tecnología. Los objetivos de calidad y de desempeño de los procesos de la organización están establecidos y son continuamente revisados y utilizados como criterio en la administración de las mejoras. Los efectos de las mejoras de procesos realizadas son medidos utilizando técnicas estadísticas y otras técnicas cuantitativas y comparadas con los objetivos de calidad y desempeño de procesos. Los procesos definidos de un proyecto, el conjunto estándar de procesos organizacionales y la tecnología de apoyo son objetivo de actividades de mejoras medibles. Una diferencia crítica entre los niveles de madurez 4 y 5 es el foco en la administración y mejora del desempeño organizacional. En el nivel 4, la organización y los proyectos se enfocan en la comprensión y control del desempeño de los subprocesos del nivel y usan ese resultado para administrar proyectos. En el nivel 5, la organización está comprometida con todo el desempeño organizacional utilizando los datos recolectados de los múltiples proyectos. El análisis de los datos identifica las carencias o deficiencias en el rendimiento. Estas deficiencias son utilizadas para guiar mejoras de procesos organizacionales que generan mejoras medibles en el rendimiento.

Cada nivel de madurez está asociado a una serie de áreas de procesos, éstas se detallan en la Tabla 2.2.

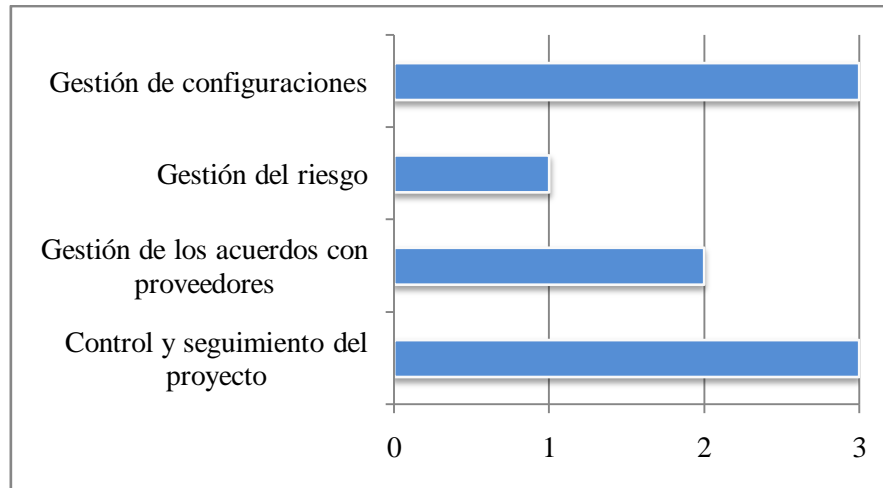
**Tabla 2.2** Niveles de madurez y áreas de proceso asociadas

Nivel	Área de procesos
5.- Optimizado	Análisis causal y resolución (CAR) Administración organizacional de rendimiento (OPM)
4.- Gestionado cuantitativamente	Rendimiento organizacional de procesos (OPP) Administración cuantitativa de proyectos (QPM)
3.- Definido	Análisis de decisiones y resolución (DAR) Administración integrada de proyectos (IPM) Definición organizacional de procesos (OPD) Enfoque organizacional de procesos (OPF) Entrenamiento organizacional (OT) Integración de productos (PI) Desarrollo de requerimientos (RD) Administración del riesgo (RSKM) Solución técnica (TS) Validación (VAL) Verificación (VER)
2.- Gestionado	Administración de configuración (CM) Medición y análisis (MA) Monitoreo de proyectos y control (PMC) Planeación de proyectos (PP) Aseguramiento de calidad de procesos y productos (PPQA) Administración de requerimientos (REQM) Administración de acuerdos de proveedores (SAM)
1.- Inicial	

#### 2.4.1.5. Representación continua

El modelo de representación continua de mejoras evalúa cada área de proceso, individualmente, con su respectivo nivel de capacidad por lo que se tiene un conjunto de valores que demuestran la madurez de la empresa en el grupo de procesos.

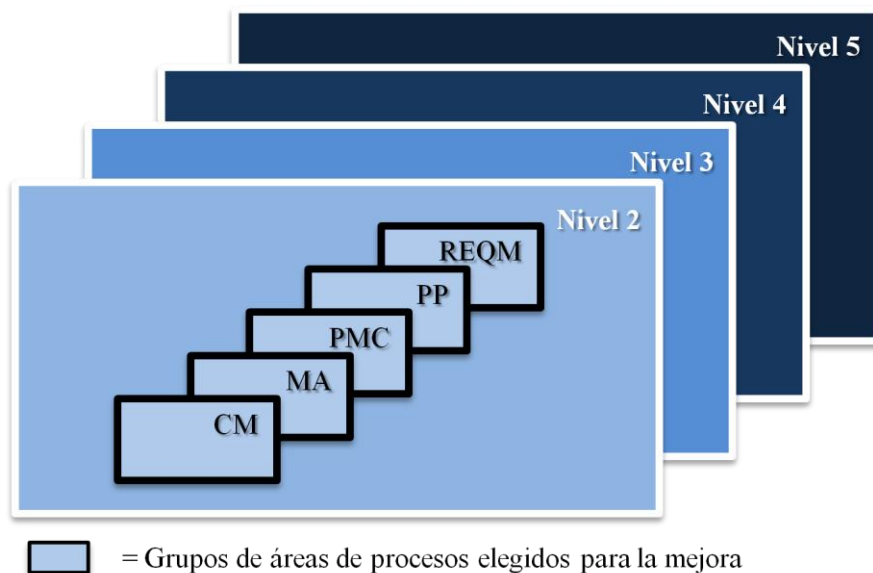
Con este modelo se ofrece una mayor flexibilidad a las empresas para que puedan centrar sus esfuerzos de mejora de acuerdo a sus propios requerimientos y necesidades. Pudiendo elegir mejorar el rendimiento de un punto problemático o varios dominios paralelamente que están fuertemente relacionados con el objetivo del negocio, siendo de un mismo o distinto nivel. Además, este modelo permite visibilidad de la capacidad alcanzada en cada área de proceso.



**Figura 2.5** Nivel de capacidad de áreas de proceso elegidas

#### 2.4.1.6. Representación por etapas

El modelo de representación por etapas es una manera sistemática y estructurada de ir mejorando los procesos utilizados por la empresa, donde cada etapa superada asegura una infraestructura adecuada para actuar como base de la etapa siguiente. En esta representación las áreas de proceso están organizadas según nivel madurez, éstas son las que determinan el nivel de madurez de la organización (Figura 2.6).



**Figura 2.6** Etapas de la representación por etapas

Con la organización de las áreas de proceso se establece el camino a seguir para implementar las mejoras y pasar de un nivel a otro. Por otra parte, entrega una trayectoria predefinida y probada de mejora a las empresas que no tienen experiencia en este aspecto.

## 2.4.2. ISO/IEC15504

Es un marco de trabajo para la evaluación y mejora de procesos que fue creado por la International Organization for Standardization – ISO y la International Electrotechnical Commission – IEC, este estándar también es conocido bajo el nombre de SPICE (Software Process Improvement and Capability dEtermination). Esta norma busca la mejora de procesos a través de su modelo de referencia y provee una guía para llevar a cabo la evaluación.

### 2.4.2.1. Modelo de referencia

Este modelo posee una arquitectura de dos dimensiones (Figura 2.7). La dimensión de proceso, donde se define un conjunto de procesos derivados de la ISO/IEC 12207 que están agrupados en distintas categorías. Cada proceso cuenta con indicadores de realización (prácticas básicas y productos resultantes). Ortogonal a la dimensión de proceso está la dimensión de la capacidad, la que tiene seis niveles de capacidad de 0 hasta 5. Cada nivel está sustentado por un conjunto de atributos de procesos, los que representan características medibles necesarias para manejar un proceso y mejorar su capacidad de desempeño. Todos los atributos de calidad contienen indicadores de capacidad (prácticas genéricas y recursos genéricos). La descripción de los niveles de capacidad y sus respectivos atributos de procesos se encuentran en la Tabla 2.3.

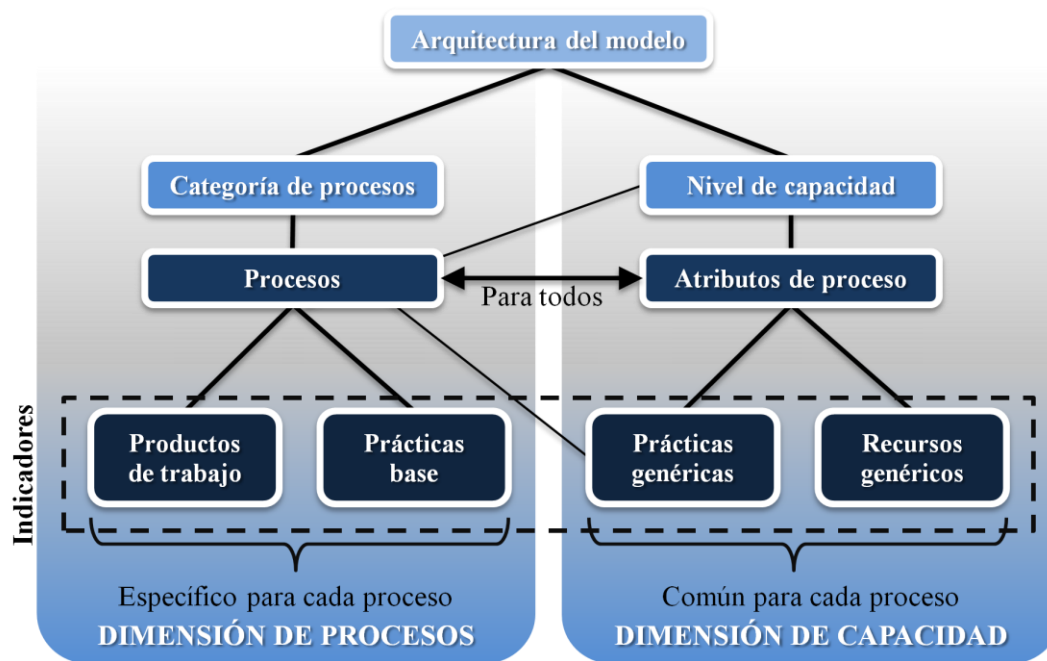


Figura 2.7 Arquitectura modelo de referencia

**Tabla 2.3** Niveles de capacidad ISO/IEC 15504

Nivel	Definición	Atributos de proceso
0. Incompleto	Generalmente fracasan los intentos por alcanzar los fines del proceso. Hay poca o ninguna facilidad para identificar los productos o salidas del proceso	N/A
1. Realizado	Generalmente se alcanzan los propósitos del proceso. Los logros pueden no ser rigurosamente planeados o seguidos. Hay productos identificables por los procesos y esto testifica el logro del propósito	<ul style="list-style-type: none"> <li>▪ Realización del proceso</li> </ul>
2. Gestionado	La entrega de productos es de acuerdo a los procedimientos especificados, es planeada y seguida. Los productos son conforme a los estándares y requerimientos especificados	<ul style="list-style-type: none"> <li>▪ Gestión de la realización</li> <li>▪ Gestión de productos resultantes</li> </ul>
3. Establecido	El proceso se efectúa y gestiona mediante un proceso basado en las buenas practicas de la ingeniería de software. Uso probado de implementaciones individuales del proceso, versiones adaptadas de estándares, procesos documentados para lograr las salidas del proceso	<ul style="list-style-type: none"> <li>▪ Definición del proceso</li> <li>▪ Recursos del proceso</li> </ul>
4. Predecible	El proceso definido es realizado consistentemente en la práctica con limites de control definidos, para lograr sus metas	<ul style="list-style-type: none"> <li>▪ Medida del proceso</li> <li>▪ Control del proceso</li> </ul>
5. Optimizado	El rendimiento del proceso es optimizado para satisfacer necesidades del negocio actuales y futuras y el proceso consigue repetidamente cumplir con las metas del negocio definidas	<ul style="list-style-type: none"> <li>▪ Innovación del proceso</li> <li>▪ Optimización del proceso</li> </ul>

#### 2.4.2.2. Evaluación

Dada la bidimensionalidad del modelo cada proceso evaluado es calificado según el nivel de cumplimiento de los atributos de procesos asociados. El nivel de cumplimiento de cada atributo de proceso puede variar entre cuatro puntos, éstos son:

1. No cumplido (N): no hay evidencia que el atributo definido ha sido logrado (entre el 0 y el 15%).
2. Parcialmente cumplido (P): hay alguna evidencia que el atributo definido ha sido logrado (entre el 16% y el 50%).
3. Ampliamente cumplido (L): hay evidencia significativa que el atributo definido ha sido logrado (entre el 51% y el 85%).
4. Totalmente cumplido (F): hay plena evidencia que el atributo definido ha sido logrado (entre el 86% y el 100%).

La salida del proceso de evaluación es un conjunto de perfiles de procesos, uno por cada instancia de cada proceso sometido a la evaluación. Estos perfiles consisten en la calificación de uno o más atributos de procesos, estas calificaciones representan la medida en que se logran los atributos según el juicio del evaluador. La agregación de las calificaciones de los procesos

permite conocer el nivel de capacidad de la organización, el esquema para la agregación se muestra en la Tabla 2.4.

**Tabla 2.4** Esquema de agregación de atributos de proceso

Nivel	Atributos de proceso	Calificación
1. Realizado	Realización del proceso	L o F
2. Gestionado	Realización del proceso Gestión de la realización Gestión de productos resultantes	F L o F L o F
3. Establecido	Realización del proceso Gestión de la realización Gestión de productos resultantes Definición del proceso Recursos del proceso	F F F L o F L o F
4. Predecible	Realización del proceso Gestión de la realización Gestión de productos resultantes Definición del proceso Recursos del proceso Medida del proceso Control del proceso	F F F F F L o F L o F
5. Optimizado	Realización del proceso Gestión de la realización Gestión de productos resultantes Definición del proceso Recursos del proceso Medida del proceso Control del proceso Innovación del proceso Optimización del proceso	F F F F F F F L o F L o F

### 2.4.3. Competisoft

La mejora de procesos de software es una práctica que es llevada a cabo por todo tipo de empresas que buscan la mejora de calidad de sus productos y la certificación de modelos reconocidos en la industria del software. Esto es algo difícil para las pequeñas y medianas empresas – PyMEs, pues la adopción de estos modelos es compleja y trae consigo un riesgo muy alto, dado que no están hechos para este tamaño de empresas.

En vista de esto, es que se desarrolló el proyecto Competisoft – Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software en Iberoamérica [Oktaba *et al.*, 2008] que ofrece un marco metodológico para la mejora de procesos de software en la pequeña y mediana empresa al integrar diferentes propuestas que son aplicables a éstas.

El marco metodológico de Competisoft ha sido elaborado a partir de diferentes iniciativas desarrolladas en Iberoamérica para la mejora de procesos de software, como:

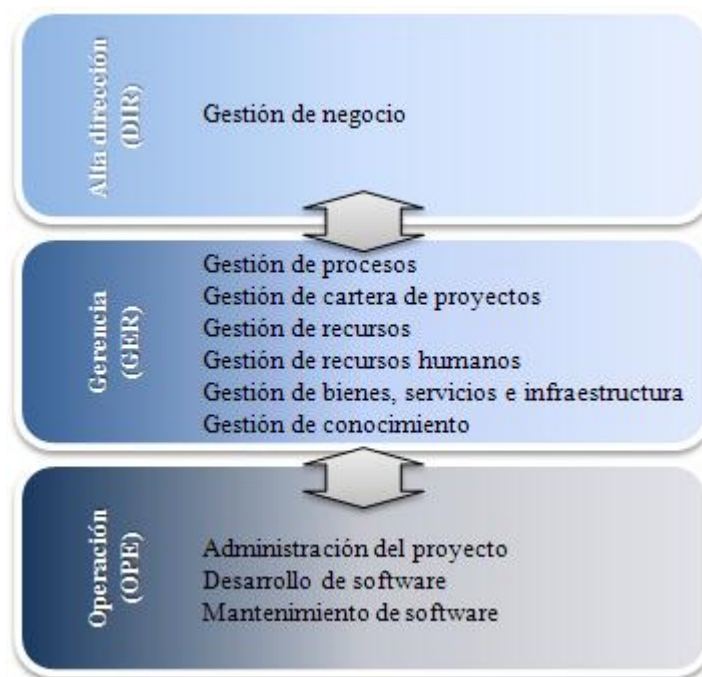
MoProSoft, desarrollado en México; Agile SPI, desarrollado en Colombia; MPS.BR, desarrollado en Brasil; y MANTEMA Y METRICA V.3, metodologías desarrolladas en España. A estas iniciativas se le agregan aspectos de las normas internacionales ISO 15504, ISO 12207 y CMMI, junto a la experiencia de las unidades gubernamentales, de los investigadores y de las propias PyMEs.

Los modelos propuestos por Competisoft son detallados a continuación.

### 2.4.3.1. Modelo de procesos

El modelo de referencia de procesos utilizado en Competisoft es una versión mejorada, luego de ser revisado y probado, del modelo MoProSoft 1.3 [NYCE, 2005]. Este tiene como alcance empresas o áreas internas de desarrollo y/o mantenimiento de software, así puede servir como un primer modelo para empresas que realicen mejoras por primera vez o, para aquellas que ya tienen sus procesos establecidos, puede ser utilizado como punto de referencia para identificar los elementos faltantes.

El modelo agrupa los procesos en tres categorías según la estructura de una organización: Alta Dirección, Gerencia y Operación, las que se pueden ver en la Figura 2.8.



**Figura 2.8** Modelo de procesos

La Categoría Alta Dirección es la encargada de establecer lo que se desea alcanzar en la empresa y las estrategias que se utilizarán para conseguir esta meta. Debe tener una visión integradora de lo que se busca para poder planear, dirigir y controlar las acciones que se emprendan a futuro en pos del objetivo.

Aquí se puede encontrar el proceso de Gestión de negocio, cuya principal actividad es la planeación estratégica, es decir, este proceso establece la razón de ser de la empresa,

identificando metas para el corto, mediano y largo plazo; define, implementa, comunica y mejora las estrategias; y mide el cumplimiento de las metas.

Para llevar esto a cabo es necesario realizar un diagnóstico del medio externo, para conocer cómo se comporta el entorno, e interno a la empresa, para reconocer las competencias de la empresa. Este diagnóstico se puede realizar a través de la técnica FODA (Fortalezas, Oportunidades, Debilidades y Amenazas). Con esto se pueden fijar los objetivos, sus metas cuantitativas y las estrategias para llevarlos a cabo, además de formas de control para ajustar las estrategias en caso de ser necesario.

Una vez definidas las estrategias a seguir es necesario identificar los procesos que permitirán llevarlas a cabo. La categoría de Gerencia es quien asegura el aprovisionamiento de recursos, establecimiento de recursos y realización de proyectos para alcanzar los objetivos de la organización, es decir, proporcionar los elementos necesarios para el funcionamiento de los procesos de la categoría de Operación. Por otro lado, recibe y evalúa información de la categoría de Operación y esta información es comunicada a la categoría de Alta dirección. En esta categoría se pueden encontrar los siguientes procesos:

- *Gestión de procesos:* establece los procesos de la organización en función de los requeridos e identificados en el plan estratégico. Además, define, planifica e implementa las actividades de mejora de estos procesos.
- *Gestión de cartera de proyectos:* asegura que los proyectos internos y externos contribuyan al cumplimiento de los objetivos y estrategias de la organización.
- *Gestión de recursos:* consigue y dota a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crea y mantiene la base de conocimiento de la organización, con la finalidad de apoyar el cumplimiento de los objetivos del plan estratégico. Tiene como complemento a los procesos de gestión de recursos humanos; gestión de bienes, servicios e infraestructura; y gestión de conocimiento.
- *Gestión de recursos humanos:* provee de los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la capacitación, evaluación de su desempeño y del ambiente de trabajo.
- *Gestión de bienes, servicios e infraestructura:* proporciona proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.
- *Gestión de conocimiento:* mantiene disponible y administra la base de conocimiento que contiene la información y los productos generados por la organización.

La Categoría Operación se encarga de abordar las prácticas de los proyectos de desarrollo y mantenimiento de software. Además, busca cumplir con los compromisos de tiempo, costo y alcance establecidos, generando productos consistentes que sirvan para el fin que fueron solicitados. La categoría de operación realiza todas sus actividades de acuerdo a los elementos entregados por la categoría de gerencia y entrega información a ésta sobre los productos generados. En esta categoría se distinguen tres procesos:

- *Administración de proyecto:* establece y lleva a cabo sistemáticamente las actividades que permiten cumplir con los objetivos de los proyectos en tiempo, costo y alcance.



- *Desarrollo de software*: realiza sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software para que cumplan con los requisitos especificados.
- *Mantenimiento de software*: realiza sistemáticamente las actividades necesarias para modificar adaptar los productos de software a los nuevos requisitos y necesidades de éste.

### 2.4.3.2. Modelo de evaluación

Competisoft no define ningún modelo de evaluación en particular, sino sugiere que se utilice un modelo de evaluación según las necesidades de la organización y que este modelo sea conforme a la norma internacional ISO/IEC 15504. Esto permite el reconocimiento de las evaluaciones independiente de donde sea aplicado el modelo de referencia de procesos de Competisoft.

El proyecto Competisoft, en su libro, utiliza el método de evaluación Light MECPDS, el cual realiza la evaluación según la norma ISO/IEC 15504, pero sólo toma en cuenta los dos primeros niveles de capacidad (realizado y gestionado) junto a los atributos de procesos relacionados a estos niveles (realización del proceso perteneciente al nivel realizado y gestión de la realización, y gestión de productos resultantes pertenecientes al nivel gestionado). Este método es utilizado por ser liviano y sencillo de llevar a cabo, además que los dos primeros niveles de capacidad son suficientes para las PyMEs. Si se requiere un tipo de evaluación más pesada y completa se recomienda MA-MPS [Weber *et al.*, 2005], MARES [Anacleto *et al.*, 2004] o EvalProSoft [Oktaba, 2004], este último será detallado a continuación.

EvalProSoft es un método de evaluación de procesos de organizaciones dedicadas al desarrollo y/o mantenimiento de software, en especial para aquellas que han utilizado el modelo de referencia MoProSoft para implementar estos procesos.

Este método involucra al Organismo Rector y a la Organización que se evaluará. La organización selecciona a un Evaluador certificado reconocido por el Organismo rector, este evaluador es quien dirige el proceso de evaluación en función de los datos de la organización de la organización, apoyándose en el Equipo de Evaluación y en el Paquete de Evaluación. Del proceso de evaluación se obtiene un reporte de resultados para la organización y un reporte estadístico para el Organismo rector. Estas relaciones pueden verse en la Figura 2.9.

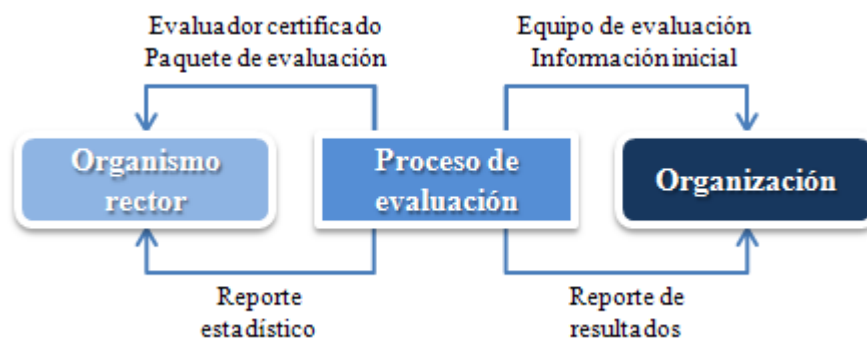


Figura 2.9 Relación entre los elementos de EvalProSoft

### 2.4.3.3. Modelo de mejora

El modelo de mejora que se propone en Competisoft, llamado PmCompetisoft, está basado en algunos componentes de Agile SPI [Hurtado *et al.*, 2008]. Con este modelo se busca establecer los elementos necesarios para guiar y gestionar la mejora de procesos de software en una pequeña organización, lo que permitirá institucionalizar la cultura de mejora continua en la organización. Además, el modelo facilita su aplicación para pequeñas organizaciones, las que no cuentan con los recursos necesarios, y ofrece resultados visibles de mejoras en el corto plazo.

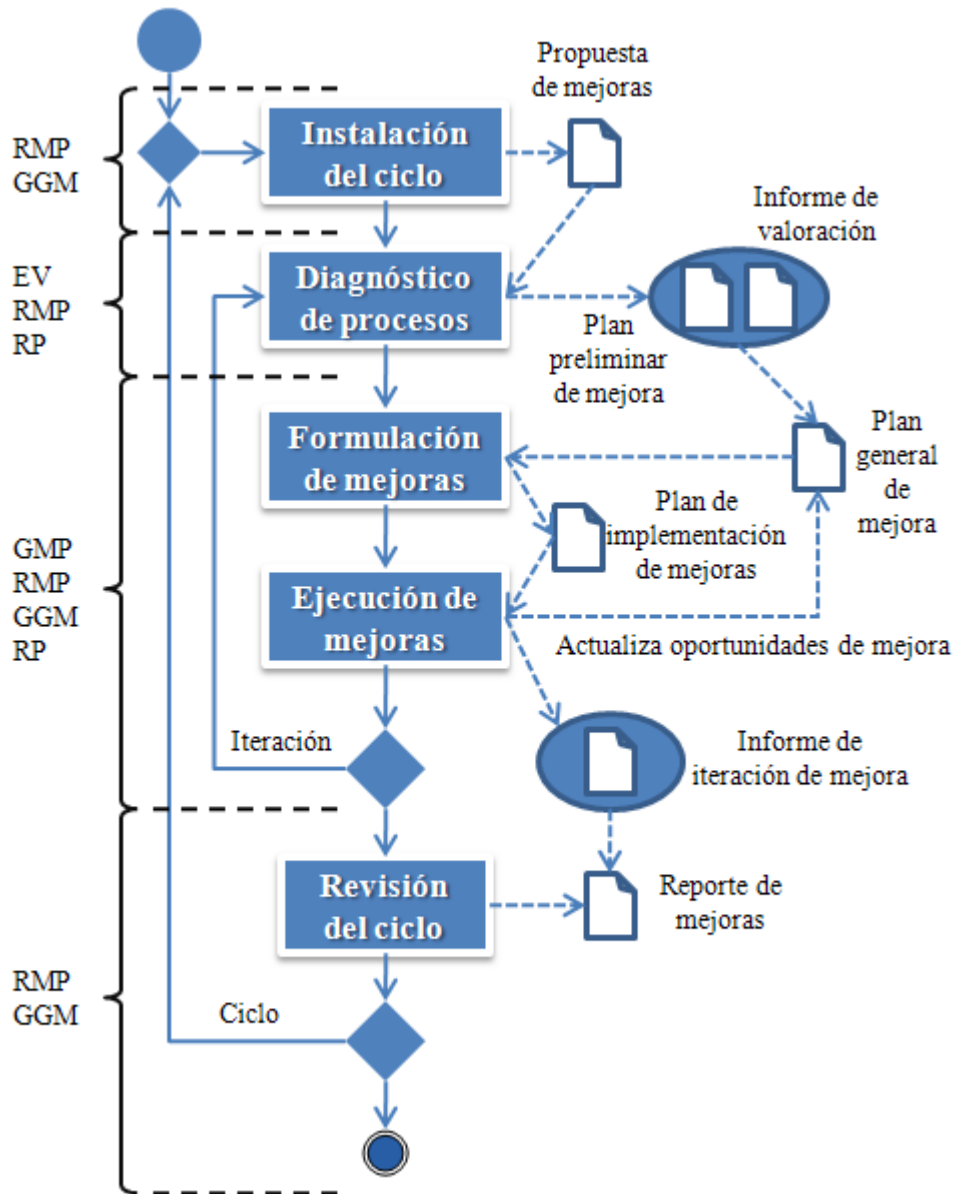
PmCompetisoft tiene como propósitos los siguientes aspectos:

- Satisfacer las necesidades de mejora de procesos de la organización, priorizando la entrega temprana (corto plazo) y continua (ciclo de vida iterativo e incremental) de mejoras visibles en los procesos de desarrollo de software.
- Diagnosticar continuamente los procesos de la organización, analizando, priorizando y ejecutando requisitos de mejora según la factibilidad de éstos.
- Establecer una colaboración entre todas las personas involucradas y relacionadas con el ciclo de mejoras.
- Construir proyectos individuales, grupales y organizacionales en torno a individuos motivados hacia la mejora de proceso. Brindar la oportunidad, respaldo y confianza necesaria para estas tareas autónomas.

Por otra parte, PmCompetisoft promueve:

- El desarrollo sostenido del ciclo de mejoras, a través del trabajo continuo.
- Una infraestructura organizacional dinámica, basada en objetivos, que permita soportar las mejoras de procesos.
- El aprendizaje continuo como una disciplina clave.

A continuación se presentan las etapas del ciclo de mejora, los roles y los productos de trabajo que PmCompetisoft ha definido. La relación entre estos elementos puede verse en la Figura 2.10.



**Figura 2.10** Relación entre los elementos de PmCompetisoft

PmCompetisoft es un proceso iterativo e incremental donde los ciclos de mejora engloban a un conjunto de iteraciones de mejora. Las que buscan implementar casos de mejora, los que han sido definidos según los objetivos de la organización. Por su parte, los casos de mejora agrupan oportunidades de mejoras concretas, las que se pueden realizar en el corto plazo entregando resultados visibles de mejoras.

Cada ciclo de mejora consta de 5 actividades, éstas son:

1. *Instalación del ciclo*: se crea o actualiza la Propuesta de mejora según el plan estratégico definido. Esta propuesta guiará a la organización en cada una de las fases del ciclo de mejora.

2. *Diagnóstico de procesos*: se evalúan los procesos de la organización (apoyado por METvalCompetisoft) y se analizan para establecer las oportunidades de mejora, esto queda consignado en el Informe de valoración. Luego se realiza una planificación preliminar del ciclo de mejora y de sus iteraciones, dando lugar al Plan preliminar de mejora. Con estos dos informes se crea el Plan general de mejora.
3. *Formulación de mejoras*: esta actividad se realiza una o varias veces en el ciclo. En ella se planifica la iteración actual y se define la estrategia a seguir para mejorar el proceso seleccionado. Si es la primera iteración del ciclo se obtiene la medida del esfuerzo (costo, tiempo, recursos, etc.) que significará realizar las mejoras en las siguientes iteraciones. Esta información se plasma en el Plan de implementación de mejora.
4. *Ejecución de mejoras*: se gestionan y ejecutan los casos de mejora de la iteración actual según lo establecido en la planificación. Si esto se realiza satisfactoriamente se aceptan e institucionalizan los nuevos procesos en la organización, para ello se debe llevar a cabo una evaluación que permitirá analizar si se han cumplido los objetivos de mejora establecidos. La información de esta actividad queda registrada en el Plan de implementación de mejora.
5. *Revisión del ciclo*: se corrigen y ajustan los elementos relacionados con la ejecución de cada iteración del ciclo de mejora. Nuevamente se realiza una evaluación para comprobar el cumplimiento de los objetivos de mejora. Al finalizar la última iteración del ciclo se analiza el trabajo realizado que servirá de retroalimentación antes de la instalación de un nuevo ciclo. Los análisis realizados, junto a las medidas desarrolladas para medir el nivel de cumplimiento de los objetivos y las lecciones aprendidas en el ciclo de mejora deben ser registradas en el Reporte de mejora.

Los roles definidos en PmCompetisoft son los siguientes:

- *Grupo directivo (GD)*: es quien tiene el conocimiento del esfuerzo requerido para llevar a cabo la planeación estratégica y, sobre todo, está comprometido con ésta y el proyecto de mejora.
- *Responsable de mejora de procesos (RMP)*: es una persona con capacidad de liderazgo y capaz de gestionar un proyecto de mejora. Además posee conocimientos de mejora de procesos de software, puede definir y diseñar procesos, guiar su implementación y garantizar la calidad del proyecto de mejora.
- *Grupo de mejora de procesos (GMP)*: un grupo de personas que poseen el conocimiento para planear y diseñar las mejoras a los procesos utilizados por la organización, lo que permitirá coordinar aspectos técnicos en las iteraciones de mejora.
- *Grupo de gestión de mejora (GGM)*: un grupo de personas (un representante del GD, uno del GMP y el RMP) que posee el conocimiento para administrar y gestionar el ciclo de mejoras.
- *Responsable de proceso (RP)*: puede ser una o un grupo de personas relacionadas directamente con un proceso. Conocen cómo se realiza o están relacionados con este proceso. Son los encargados de aportar la información necesaria para valorar un proceso e incluir las mejoras diseñadas para éste.
- *Evaluador (EV)*: persona que conoce la metodología y aplicación de evaluación de procesos.

Los productos de trabajo del modelo de mejora son descritos a continuación:

- *Propuesta de mejora (PM)*: proporciona una introducción e iniciación a un ciclo de mejora. Contiene objetivos de mejora generales, el proceso de mejora continua, el alcance y metas globales del ciclo, además de los recursos asignados.
- *Plan general de mejora (PGM)*: está compuesto por el informe de valoración y el plan preliminar de mejora. El informe de valoración recopila el estado, análisis y priorización de los procesos. Mientras el plan preliminar de mejora define el número de iteraciones, la planeación general, el plan de mediciones, el plan de capacitación, el plan de manejo de riesgos y el cronograma.
- *Plan de implementación de mejora (PIM)*: define las prácticas y actividades a seguir para crear, ejecutar e institucionalizar los casos de mejora. Contiene la planeación específica de la iteración, el registro de la ejecución de la mejora, las evaluaciones de los casos de mejora, la documentación de los procesos y el plan de aceptación e institucionalización.
- *Reporte de mejora (RM)*: cierra una iteración o ciclo de mejora. Contiene los procesos mejorados, la evaluación inicial, la evaluación final, el esfuerzo involucrado, logros alcanzados, lecciones aprendidas, revisión post mortem y recomendaciones de ajustes al proceso de mejora.

Para facilitar la construcción de los productos de trabajo, PmCompetisoft ha desarrollado una plantilla para cada uno. Además, cada uno de los productos de trabajo registran las métricas del esfuerzo de realizar las tareas asociadas con cada actividad con que se relacionan.

## **2.5. Notaciones para el modelado de procesos de software.**

En este apartado se presentará: el concepto de notación de modelado de procesos de software, las más reconocidas notaciones para procesos de software, se expondrá el metamodelo SPEM2 (con el que se modelará un proceso de software de Competisoft) y, finalmente, una comparación entre las distintas notaciones.

### **2.5.1. Características de las notaciones para el modelado de procesos de software**

Debido a la complejidad que presentan los procesos de software es necesario contar con técnicas y métodos que permitan facilitar y, en parte, automatizar la tarea de trabajar con ellos. Así, los investigadores han creado o adaptado notaciones para llevar a cabo el modelado de procesos de software y poder formalizar aspectos que permitan representar de manera precisa y completa algunas de sus características.

Los aspectos que se consideran importantes para las notaciones de modelado de procesos de software según lo descrito por Fuggetta [2000] son:

- Actividades que deben ser completadas para lograr los objetivos del proceso.
- Roles de las personas en los procesos.
- Estructura y naturaleza de los artefactos que son creados y mantenidos.
- Herramientas utilizadas.

Las notaciones para el modelado de proceso de software tienen múltiples propósitos que, según la notación, permiten aprovechar los beneficios que implica la aplicación de estas técnicas. Los propósitos son:

- *Entendimiento de los procesos*: representar en forma precisa la estructura y organización de los procesos, permitiendo eliminar inconsistencias en su definición.
- *Diseño de procesos*: plantear nuevos procesos describiendo su estructura y organización.
- *Entrenamiento y educación*: utilizar la descripción de los procesos para enseñar los procedimientos y operaciones al nuevo personal.
- *Simulación y optimización de procesos*: simular los procesos descritos para evaluar cuáles provocan problemas, cuellos de botella u oportunidades de mejora.
- *Soporte de procesos*: usar e interpretar los procesos para proveer diferentes niveles de soporte al personal que los utilizan.

Estos propósitos permiten apoyar el modelado de procesos de software, favoreciendo a la obtención de los beneficios que ambos elementos, notaciones para el modelado y el mismo modelado de procesos de software, ofrecen.

## **2.5.2. Otras notaciones para el modelado de procesos de software**

Otras notaciones para el modelado de procesos o elementos de otras notaciones que son utilizados para el modelado de procesos de software son: diagramas de actividad UML, Bussines Process Managment Notation – BPMN, XML Process Definition Language – XPDL, jBoss Business Process Management – jBPM, Architecture of Integrated Information Systems – ARIS, y ICAM Definition Languages – IDEF. Estas notaciones serán brevemente descritas.

### *Diagrama de actividad UML*

El lenguaje unificado de modelado, o UML, entrega una serie de diagramas para modelar y describir distintos aspectos de un sistema software. Entre los aspectos que UML da soporte se encuentra el comportamiento del sistema, donde se pueden encontrar los diagramas de estado, secuencia y actividad. Estos últimos, tienen como objetivo describir la lógica procedural, flujos de trabajo y procesos de negocio. Los diagramas de actividad pueden ser utilizados para modelar procesos de software [Wohed *et al.*, 2004].

### *Bussines Process Managment Notation – BPMN*

Es un estándar que busca proporcionar una notación sencilla y comprensible para poder diseñar e implementar procesos de negocio y, además, prestar una notación gráfica para éstos. BPMN fue concebido tomando la experiencia de otros estándares para enfocarse solamente en los procesos de negocio, pero debido a sus patrones y expresividad a la hora de especificar los procesos se utiliza como notación para los procesos de software [Wohed *et al.*, 2005].

### *XML Process Definition Language – XPDL*

Es una de las cinco interfaces funcionales, definidas por la Workflow Management Coallition – WfMC, en un proceso o servicio workflow [Hollingsworth, 1995]. XPDL es parte de la documentación que da soporte a la definición y a la importación/exportación de modelado de procesos, con el objetivo de, aunque el modelado sea realizado en un programa,

pueda ser usado por otros programas de modelado y/o por otras aplicaciones que trabajen en el entorno de ejecución.

#### *jBoss Business Process Management – jBPM*

jBPM es un sistema flexible y extensible de administración de workflow que cuenta con un lenguaje de proceso intuitivo para expresar gráficamente procesos de negocio en términos de tareas, estados de espera para comunicación asíncrona, temporizadores, acciones automatizadas, etc. [jBoss Company, 2006].

#### *Architecture of Integrated Information Systems – ARIS*

ARIS es un framework para describir estructuras organizativas, procesos y aplicaciones de negocio de toda una empresa. Es una metodología compleja, de gran extensión cuya notación cubre todos los aspectos relacionados con el desarrollo, optimización, integración e implementación de un sistema de información. Sus diagramas son sencillos y comprensibles, hasta para personas que no son especialistas. Además, permite vistas multinivel de los procesos y se integra con UML [IDS Scheer, 2005].

#### *ICAM Definition Languages – IDEF*

En un proyecto iniciado en los años 70, por la *United States Air Force*, que ha desarrollado una serie de especificaciones y diversas metodologías, compatibles entre sí, para los distintos aspectos relacionados a la creación de sistemas de información. Para lenguajes de procesos se cuenta con las especificaciones: IDEF0 para el modelado de procesos dentro de una organización e IDEF3 para la captura de descripciones de procesos [Martínez San Germán, 2003].

### **2.5.3. System and Software Process Engineering Metamodel – SPEM2**

Es un metamodelo, basado en MOF (Meta Object Facility, [OMG, 2006] ) y algunas librerías de infraestructura de UML2 [OMG, 2010], para el modelado de procesos de software. SPEM2 posee los elementos mínimos necesarios para poder definir procesos asociados a las actividades del desarrollo de software sin añadir particularidades sobre el dominio utilizado, pero permite la utilización de distintos procesos, en distintos niveles, proyectos, culturas organizacionales, etc.

#### **2.5.3.1. Aspectos principales de SPEM2**

La idea central de SPEM2 se puede resumir con la frase: “quién realiza qué para, a partir de unas entradas, obtener unas salidas”; donde el quién es un rol, el qué es una tarea y las entradas y salidas son productos de trabajo. Las relaciones básicas se presentan en la Figura 2.11.

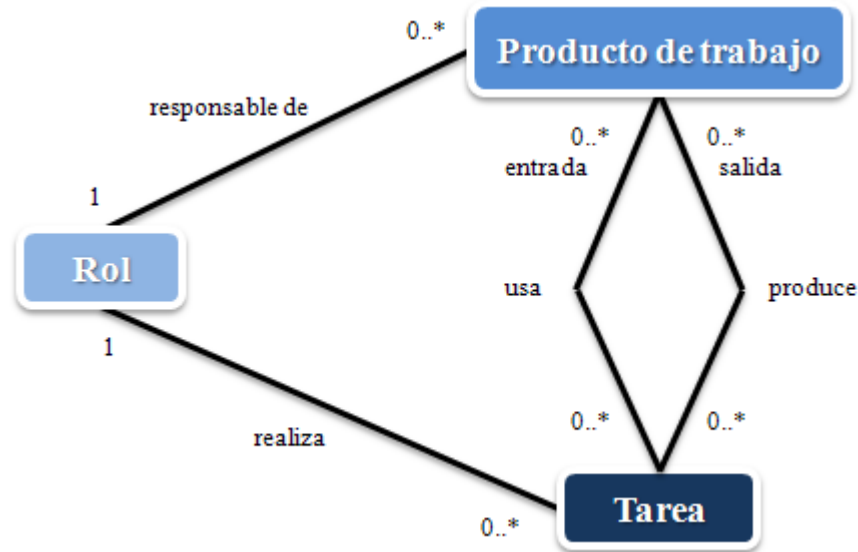


Figura 2.11 Idea central de SPEM2

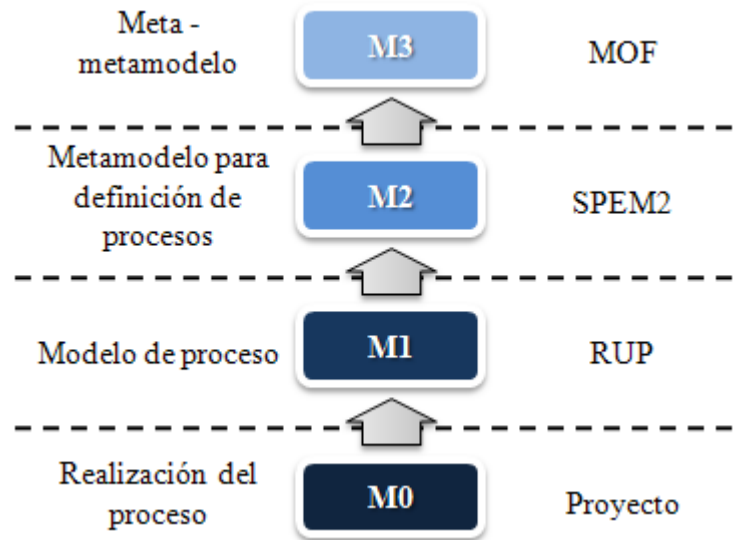
Los principales usos de SPEM2 son:

- *Proveer de una representación estandarizada y bibliotecas gestionadas de contenidos de métodos reutilizables:* poder crear una base de conocimiento que permita crear, cambiar y reutilizar un conjunto de roles, tareas, productos de trabajo, guías, procesos, etc. de forma estandarizada para toda la organización. Esto proporciona al personal el conocimiento necesario sobre: las habilidades requeridas para cada rol, el desarrollo y mantenimiento de software; los productos de trabajo que se necesitan u obtienen en cada tarea; directrices para la ejecución de las tareas en cada proceso, etc.
- *Dar soporte al desarrollo, gestión y crecimiento de procesos software:* definir y seleccionar procesos acorde a las necesidades de cada proyecto, dado que los métodos pueden ser utilizados en contextos y proyectos distintos. Así cada jefe de proyecto puede aplicar distintos procesos o el mismo de manera distinta (reutilizándolos) en cualquier nivel de un proyecto, es decir, permite el ensamblado de “trozos” de procesos predefinidos desde catálogos para adaptarlos a proyectos concretos y específicos.
- *Dar soporte al despliegue de contenido de método y proceso que justo se necesita:* configurar un conjunto de procesos de la manera adecuada y particularizarlo para cada proyecto específico, pues un proceso nunca se ejecuta dos veces de la misma forma. Para esto SPEM2 ofrece reutilización, variabilidad y particularización de procesos.
- *Dar soporte a la realización de un proceso para un proyecto software:* desplegar los procesos en formatos que permitan su realización automática por sistemas de gestión de proyectos, motores de flujos de trabajo, etc. SPEM2 incluye estructuras que permiten definir la forma en que estos procesos deben ser expresados para ser procesados automáticamente.

SPEM2, por estar basado en MOF, define una arquitectura de modelado que incluye cuatro niveles conceptuales. Como se puede ver en la Figura 2.12 el nivel más alto pertenece



al meta-metamodelo<sup>1</sup>, el propio MOF. En el nivel inmediatamente inferior se puede encontrar SPEM2, en caso que se esté buscando un metamodelo para definir procesos de software (en el caso de modelado de sistemas de software, en este nivel podría encontrarse UML). En el siguiente nivel ya se pueden encontrar elementos o modelos genéricos que utilizan los conceptos definidos por el metamodelo, como podría ser el modelado de la metodología RUP – Rational Unified Process. En el nivel más bajo se encuentra la instanciación pura de los elementos de un proyecto con datos concretos.



**Figura 2.12** Niveles de modelado de MOF

El marco teórico de SPEM2 define dos conjuntos de conceptos para implementar cualquier tipo de metodología. Por una parte se encuentra el Contenido de métodos donde se encuentran todos los elementos primarios o constructores básicos, por ejemplo la definición de roles o tareas. El otro conjunto es el de procesos, donde se combinan y reutilizan los componentes definidos en el contenido de métodos para obtener como resultado los procesos, como el uso de un rol o tarea. En la intersección de estos dos conjuntos se encuentran las guías que proporcionan información relacionada a otros elementos, por ejemplo cómo usar una herramienta. El marco teórico se grafica en la Figura 2.13, además la relación de estos conceptos con un modelo de procesos, en este caso RUP, se observa en la Figura 2.14.

<sup>1</sup>Metamodelo: descripción de un conjunto de conceptos genéricos y sus interrelaciones que sirven de base para la definición de modelos de un cierto dominio.

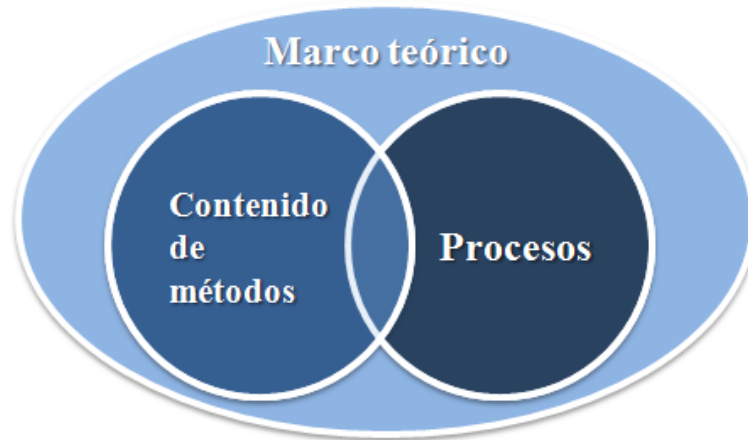


Figura 2.13 Marco teórico de SPEM2

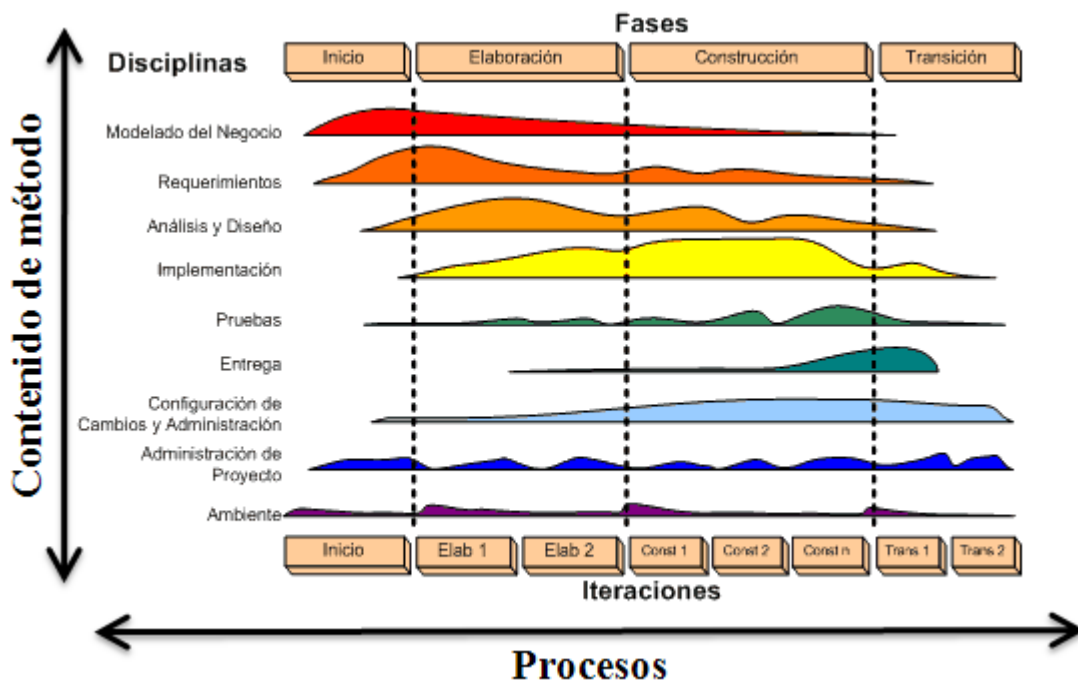


Figura 2.14 Ortogonalidad en el modelo de procesos RUP

La jerarquía de desglose del trabajo que utiliza SPEM2 para representar el esfuerzo que se debe realizar en los distintos niveles de trabajo está compuesta por cuatro niveles. Desde el más general al más particular son:

1. *Delivery process*: representa un proceso tan complejo como sea necesario para que se utilice como base para realizar cierto tipo de proyecto.
2. *Capability pattern*: representa una porción de un proceso que puede ser reutilizado en más de una ocasión por un delivery process.
3. *Activity*: es el elemento central para definir procesos, pues permite organizar sus elementos básicos.

4. *Task descriptor / Workproduct descriptor*: el primero, es la unidad básica de trabajo en un modelo de proceso de SPEM2. El segundo son las entradas o salidas de una actividad.

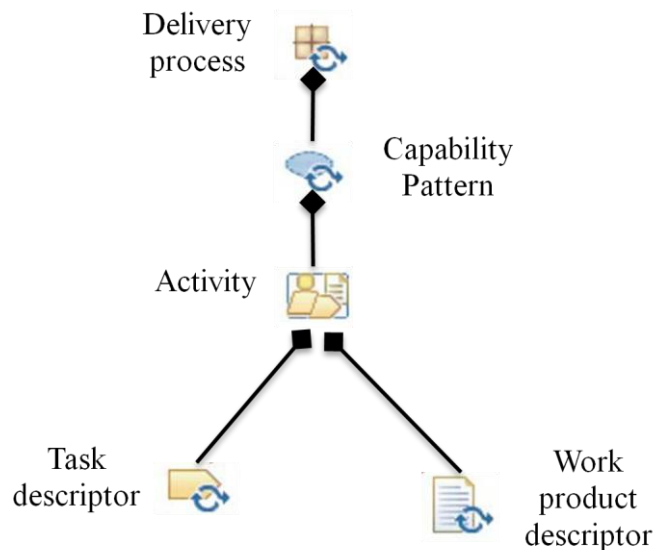


Figura 2.15 Jerarquía de desglose del trabajo

### 2.5.3.2. Estructura de paquetes de SPEM2

SPEM2 está compuesto por siete paquetes, al igual que UML por razones de organización y reutilización, cada uno es una unidad lógica que entrega estructuras y capacidades adicionales a los paquetes con que se relaciona. Para relacionar los paquetes se utiliza el mecanismo de paquetes “merge” de UML2, el que importa los contenidos desde el paquete destino al paquete fuente, donde se actualizan los contenidos existentes o se agregan los no existentes. Las relaciones de cada uno de los siete paquetes se pueden apreciar en la Figura 2.16 y su descripción es:

1. *Core*: contiene aquellas clases y abstracciones del metamodelo que construyen la base para las clases de otros paquetes, es decir, las clases comunes para los paquetes de niveles superiores están aquí. Principalmente provee de dos capacidades básicas de SPEM2: (1) crear cualificaciones definidas por el usuario para definir distintos tipos de instancias de una clase. (2) un conjunto de clases abstractas que permiten definir el trabajo como procesos de SPEM2.
2. *Process Structure*: define la base para la creación de todos los modelos de procesos sencillos y flexibles, definiendo estructuras de desglose para actividades anidadas que mantienen una lista que referencia a los roles que las realizan, así como también a los productos de trabajo de entradas y/o salidas para cada actividad. Además, provee mecanismos para la reutilización de procesos a través de la unión dinámica de patrones, permitiendo a los usuarios el ensamblado de procesos con conjuntos de actividades vinculadas dinámicamente.
3. *Process Behavior*: amplía al paquete *Process Structure* agregando modelos de comportamiento. Sin embargo, SPEM2 no define modelos de comportamiento propios,

sino que ofrece vínculos a modelos definidos externamente, como por ejemplo los diagramas de actividad de UML.

4. *Managed Content*: permite gestionar descripciones en lenguaje natural o agregar documentos, lo que permite una mejor comprensión. Muchos detalles o aspectos de los procesos no pueden ser simplemente incorporados en los modelos, por lo que la documentación y guías en lenguaje natural son el complemento ideal, ya que se puede o no agregar este tipo de información textual extra.
5. *Method Content*: incluye los conceptos para desarrollar una base de conocimiento independiente de los procesos específicos y los proyectos de desarrollo, es decir, incluye los elementos principales que sirven de base para el ensamblado de procesos, metodologías, ciclos de vida, etc. permitiendo describir qué objetivos deben ser logrados, por cuáles roles, con qué recursos y qué resultados deben obtenerse. Los procesos pueden utilizar estos elementos y relacionarlos en secuencias parcialmente ordenadas para determinados tipos de proyectos.
6. *Process with Methods*: define nuevas estructuras y redefine otras existentes para integrar los procesos definidos en el paquete *Process Structure* con instancias de conceptos del paquete *Method Content*. Al asociar elementos que constituyen los métodos, como tareas, roles y productos de trabajo, a partes específicas del proceso que referencian a clases creadas para almacenar cambios individuales de éstas, que expresan cómo y qué parte del método debe ser aplicado en un punto particular del proceso.
7. *Method Plugin*: introduce conceptos para diseñar y gestionar librerías o repositorios de contenido de métodos y de procesos mantenibles de gran escala, reusables y configurables. Con conceptos como *Method Plugin*, *Process Component* y *Variability* se pueden definir distintas vistas de un mismo proceso, a través de las Configuraciones de métodos, según sea el tipo de usuario final al que están destinadas.

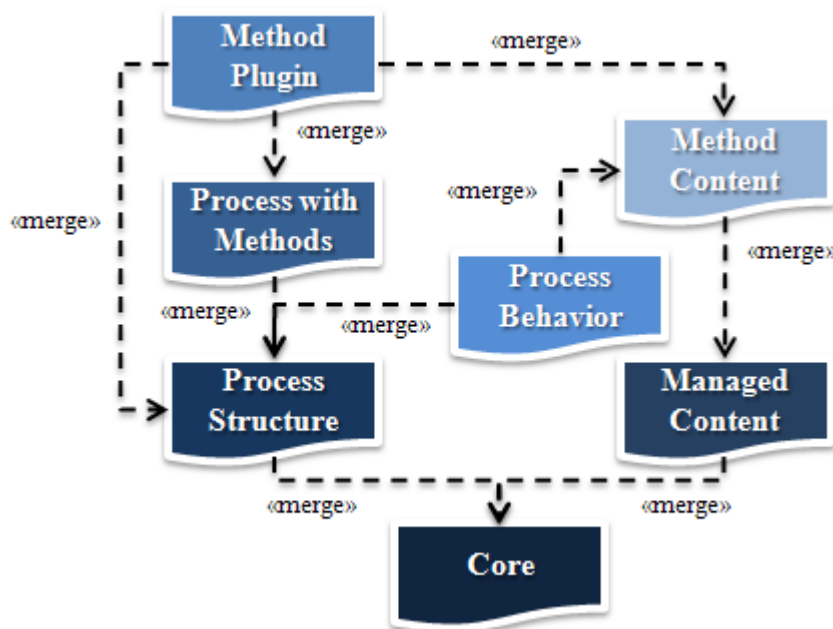


Figura 2.16 Estructura de paquetes de SPEM2

#### 2.5.4. Comparación entre notaciones de modelado

Pérez [2007] establece criterios para realizar la comparación de las distintas notaciones para el modelado de proceso de software. Utiliza diez características que considera importantes y que deberían ser cubiertas por las notaciones. Las características para la comparativa son:

1. *Expresividad*: modelar la complejidad de los procesos de negocio.
2. *Roles*: representar roles y su asignación a diferentes tareas.
3. *Calidad*: especificar las características de calidad de los procesos de negocio.
4. *Re-uso*: especificar repositorios de procesos que permitan la reutilización de éstos mediante la utilización de conceptos como la variabilidad y la extensibilidad.
5. *Gestión*: especificar atributos que permitan gestionar los procesos (monitorear, controlar o planificar los mismos).
6. *Multinivel*: permitir una vista procesos a partir de descripciones de alto nivel, más generales y comprensibles, como también tener la posibilidad de alcanzar niveles inferiores de descripción con gran cantidad de detalles.
7. *Comprensible*: ser fácil de entender para aquellos que no son especialistas en modelado. Esta característica es especialmente útil si con posterioridad se pretende utilizarlos modelos para la fase de requerimientos.
8. *Integración y soporte*: permitir la integración y soporte para otro tipo de notaciones facilitando la interacción entre las herramientas que den soporte a ambas notaciones.
9. *Código*: posibilitar la vinculación de manera directa entre una actividad y un fragmento de código en un lenguaje de programación.
10. *Herramientas*: existencia de herramientas para trabajar con la notación de modelado.

Con estas características definidas Pérez [2007] realizó la comparativa entre las notaciones obteniendo los resultados expuestos en la Tabla 2.5.

De la tabla se puede observar que SPEM es una de las notaciones para el modelado de procesos de software más potentes, cumpliendo con la gran mayoría de las características establecidas para la comparación, donde deja abierta la utilización de terceras notaciones en varios aspectos (expresividad, multinivel y comprensible), además solamente no cumple con dos características, el modelado de atributos de calidad y con el vínculo directo que debe existir entre las actividades y el código. Estas características que no fueron cumplidas por SPEM, tampoco fueron cumplidas por la mayor parte de las notaciones. Sólo ARIS posee elementos para modelar los atributos de calidad y, en cuanto a código, jBPM tiene ese vínculo directo, siendo la única notación que lo permite. Por otra parte, SPEM destaca por sobre las demás notaciones al ser el único que presenta la característica de reuso y, junto con ARIS, es uno de las dos notaciones que entrega atributos que permiten la gestión de los procesos.

**Tabla 2.5** Esquema de agregación de atributos de proceso

		Notación						
		SPEM	DA	BPMN	XPDL	jBPM-jPDL	ARIS	IDEF
Características	Expresividad	(1)	Sí	Sí	Sí	Sí	(1)	Sí
	Roles	Sí	Sí	Sí	Sí	Sí	Sí	No
	Calidad	No	No	No	No	No	Sí	No
	Re-uso	Sí	No	No	No	No	No	No
	Gestión	Sí	No	No	No	No	Sí	No
	Multinivel	(1)	No	Sí	No	No	Sí	No
	Comprensible	(1)	Sí	Sí	Sí	Sí	Sí	Sí
	Integración y soporte	BPMN, DA, XPDL	No	XPDL	BPMN	No	UML, SAPR/3	No
	Código	No	No	No	No	Sí	No	No
	Herramientas	Sí	Sí	Sí	Sí	Sí	Sí	Sí

(1): Se deja a terceras notaciones.

## 2.6. Herramientas de modelado de procesos de software

En el mercado existen variadas herramientas que se pueden utilizar para el modelado de procesos de software. A continuación se revisarán las de mayor importancia, revisando con mayor detalle la herramienta Eclipse Process Framework Composer - EPF Composer.

### 2.6.1. IBM Rational Method Composer

Es una plataforma para la creación y personalización de métodos y procesos de software, pertenece a la familia de software Rational de la empresa IBM, por lo tanto requiere licencia pagada. Al ser de IBM utiliza la metodología RUP como base.

Las principales características de IBM Rational Method Composer son:

#### **Configuración de procesos.**

*Configuración:* ayuda a seleccionar una configuración base de RUP, aplicar plug-ins y otros componentes del proceso, configurar las vistas del proceso y publicar la personalización de RUP.

*Plug-ins:* todos los procesos son creados como plug-ins. Muchos de éstos son extensiones de RUP, pero pueden haber plug-ins que extienden contenidos de procesos detrás de la aplicación desarrollada.

### **Creación de procesos.**

*Creación:* fácil integración con los procesos guías y contenido de configuraciones RUP de la organización. Permite la edición y cambio de procesos, agregando, eliminando, modificando roles, productos de trabajo, tareas, etc.

### **Particularización de procesos.**

*Constructor de procesos:* una configuración de método es una selección de contenido de la librería de métodos que puede ser publicada en una página web o en un documento. El objetivo de utilizar un constructor de procesos es crear configuraciones de métodos fáciles de navegar, revisar y seleccionar sin conocer detalles sobre el diseño y estructura de la librería y su contenido.

*Particularización de perspectivas:* describe como existen los contenidos de métodos y procesos personalizados según las necesidades de un proyecto del mundo real. Típicamente esto es realizado al comienzo de un proyecto por el administrador del proyecto.

### **Despliegue de procesos.**

RSS: usar RSS para generar información específica para los subscriptores.

*Guía de herramienta:* entrega guías por el uso de las herramientas de realización de tareas de Rational.

## **2.6.2. Osellus IRIS**

Es una herramienta de pago perteneciente a la empresa Osellus para la administración visual de procesos, permitiendo la colaboración para la creación y particularización de los procesos para asegurar que la organización conozca sus objetivos. La administración de procesos estructurados es posible gracias a una arquitectura que facilita la personalización eficiente y la reutilización de procesos a través de toda la organización. IRIS es independiente de la metodología, permitiendo a la organización combinar el mejor contenido que se puede encontrar en el mercado con sus propias prácticas establecidas que han demostrado ser útiles. Además, la herramienta utiliza tecnología web 2.0 para proveer un portal que facilita el desarrollo de los procesos online.

Las principales características de IRIS son:

### **Experiencia de usuario:**

Arrastrar y soltar modelos de workflows, estructuras de desglose de trabajo, máquinas de estado, etc.

Edición de texto enriquecido para descripciones y otros campos.

Herramientas y paneles sensibles al contexto.

### **Arquitectura de proceso:**

Empaquetado enriquecido de procesos y herencia de modelos.

Modelado de roles por usuario para librerías de procesos.

Clasificación de procesos personalizada para la fácil identificación.

**Definición de procesos:**

- Compatible con SPEM – elementos para ciclo de vida, fases, iteraciones, definición de trabajo, actividades, productos de trabajo, guías, realizador de procesos y roles de proceso.
- Terminología personalizable para los elementos de proceso (por ejemplo: “actividad” puede referirse a “tarea” en algunas metodologías).
- Definición multinivel de workflows.
- Relaciones avanzadas de modelado, como dependencias de precedencia, actividades de precondition y objetivos base en estados de productos de trabajo, impacto de productos de trabajo, etc.
- Máquinas de estado para productos de trabajo.
- Campos para detallar completamente los elementos del proceso, incluyendo campos definidos y personalizables por la organización.
- Vínculos a guías externas.
- Validación de procesos para asegurar la integridad fin-a-fin.

**Importación, exportación y reporte:**

- Importación XML basada en metodologías de la industria.
- Exportación configurable con opciones predefinidas para MS Project, VSTS, CMMI Reports, etc.

**Publicidad y retroalimentación:**

- Publicaciones online, incluyendo diagramas con alcance de salidas definidas
- Publicar como un Wiki para facilitar la particularización de procesos y la contribución en el equipo.
- Publicar en un Proceso Central para habilitar los procesos transversalmente en la organización.
- Las salidas pueden ser editadas en Word y Visio.
- Contenido, estilo y capas configurables en las plantillas de salida.

**Proceso Central:**

- Repositorio de procesos online para promover el conocimiento y adopción de procesos.
- Facilitar la retroalimentación permitiendo a los miembros de la comunidad explorar, discutir y contribuir a los procesos.
- Búsqueda de texto completo y metadatos habilitada para encontrar el proceso adecuado para su utilización.
- Modificación de definición de procesos utilizando la tecnología Wiki.



### 2.6.3. Eclipse Process Framework Composer

Eclipse Process Framework Composer – EPF Composer es parte del entorno Eclipse, desarrollado por la organización del mismo nombre, siendo una herramienta open-source completamente gratuita diseñada para que ingenieros de procesos y administradores de proyectos puedan crear, combinar y publicar métodos y procesos para proyectos y desarrollo de software de una organización.

El proyecto tras EPF Composer persigue dos objetivos:

1. Proporcionar un marco de trabajo extensible y herramientas ejemplares a la ingeniería de procesos de software – creación de métodos y procesos, administración de bibliotecas, configuración y publicación de un proceso.
2. Proporcionar contenido de proceso ejemplar y extensible para una gama de desarrollo y administración de procesos de software que soporte desarrollo iterativo, ágil e incremental, y sea aplicable a un amplio conjunto de plataformas y aplicaciones de desarrollo.

La arquitectura de esta herramienta se observa en la Figura 2.17 y cada una de sus partes es descrita a continuación.

#### Componentes dependientes:

*Eclipse Rich Client Platform – RCP:* es una plataforma para construir aplicaciones Java que luzcan y se comporten como una aplicación nativa de escritorio. EPF Composer fue diseñado para correr independiente de la plataforma, sea Windows o Linux.

*Eclipse Modeling Framework – EMF:* es un framework para el modelado y generación de código para aplicaciones basadas en modelos de datos estructurados.

*Graphical Editing Framework – GEF:* es un framework para desarrollar editores gráficos y diagramas.

*International Components for Unicode for Java – ICU4J:* es una librería Java que soporta Unicode.

*JTidy:* es un verificador de sintaxis y publicador para HTML.

#### Estándares soportados:

*XML Meta Interchange – XMI:* es una especificación de la OMG para el almacenamiento e intercambio de metadatos en formato XML.

*Unified Modeling Language – UML 2.0 Diagram Interchange Specification:* es una especificación de la OMG para el intercambio de diagramas de UML2 entre diferentes herramientas de modelado.

*Extensible Stylesheet Language Transformation – XSLT:* es una recomendación de la World Wide Web Consortium (W3C) para la transformación de documentos XML en otras formas de documentos XML.

*Dynamic HTML – DHTML:* comprende especificaciones como HTML, JavaScript, Cascading Style Sheets (CSS) y Document Object Model (DOM).

*Componentes de EPF Composer:*

*Common:* provee infraestructura común para todos los componentes de servicios de EPF Composer, como manejo de errores, manipulación de strings, etc.

*Unified Method Architecture – UMA:* provee acceso básico y soporte a la edición de métodos y elementos de procesos almacenados en la librería de métodos.

*Library Management:* provee la interfaz y los servicios para administrar la librería de métodos.

*Rich Text:* provee la capacidad de editar texto en los elementos de cualquier método.

*Authoring:* permite ver y editar la perspectiva de creación de procesos de EPF Composer para poder tener a la vista todos los elementos necesarios para la creación de procesos de manera ordenada.

*Browsing:* provee la vista que permite la navegación a través de los elementos de EPF Composer.

*Publishing:* provee la interfaz y los servicios para publicar una configuración o proceso en un sitio web estático.

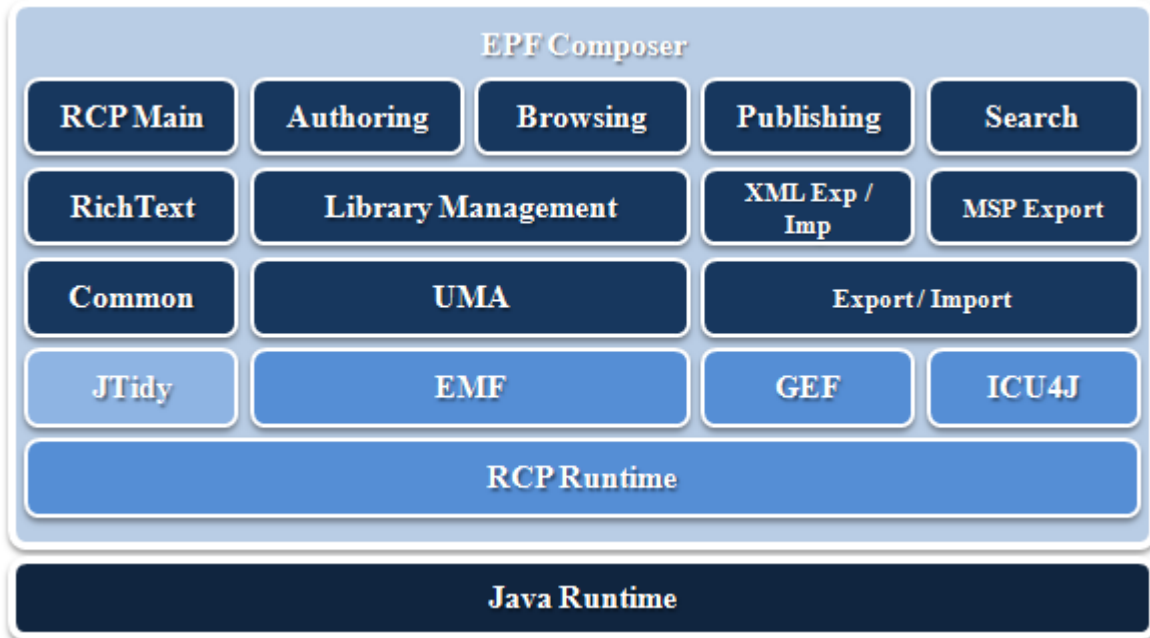
*Search:* provee la interfaz y los servicios para la búsqueda de un elemento de método específico en la librería de métodos.

*Export / Import:* provee la interfaz y los servicios para exportar e importar plug-ins de métodos y configuraciones empaquetadas en archivos XML.

*XML Export/Import:* provee la interfaz y los servicios para exportar e importar contenido de la librería de métodos a XML o desde XML.

*Microsoft Project - MSP Export:* provee la interfaz y los servicios para exportar un proceso a archivo Microsoft Project 2003 XML.

*RCP Main:* provee especificaciones para la personalización de la vista, empaquetado y actualizaciones del sitio de EPF Composer.



**Figura 2.17** Arquitectura EPF Composer

### 3. Modelado del proceso de mantenimiento Agil\_Mantema

---

A lo largo de esta sección se presentará y detallará el proceso de mantenimiento de software, planteado dentro del modelo de referencia de procesos de Competisoft, junto a sus respectivos modelos y diagramas realizados con la herramienta Eclipse Process Framework Composer.

#### 3.1. Proceso de mantenimiento de software

En el proyecto Competisoft se ha definido como proceso de mantenimiento de software a Agil\_Mantema, el cual fue creado al aplicar la metodología ágil Scrum [Takeuchi y Nonaka, 1986] a la metodología para mantenimiento de software Mantema [Polo *et al.*, 1999]. Con esto se obtiene una propuesta metodológica enfocada a las PyMEs, donde el proceso de mantenimiento es liviano y sigue una metodología ágil, al mismo tiempo detalla y guía paso a paso a través de todo el proceso indicando qué debe realizarse, cuándo, cómo y por quién.

En la Figura 3.1 se presenta la estructura general de Agil\_Mantema, la cual se basa en la estructura de Mantema y las indicaciones de la ISO/IEC 12207. Esto se complementa con:

- *Niveles de servicio*: definen el tipo de servicio que se presta, junto con especificar las interfaces para cada proceso, con los que se puede aumentar su nivel de capacidad.
- *Niveles de capacidad*: la implementación de las actividades descritas por las interfaces permiten llevar a cabo prácticas base y de gestión que incrementan el nivel de capacidad, los que están basados en la ISO/IEC 15504.

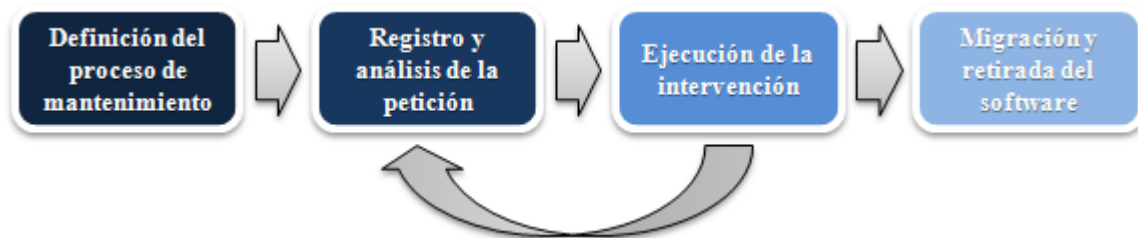


Figura 3.1 Estructura general Agil\_Mantema

La Tabla 3.1 muestra las actividades respectivas para cada nivel de servicio y de capacidad. Las actividades de los niveles de capacidad, definidas por las interfaces fundamentales, se evalúan según su nivel de cumplimiento determinado por la ISO/IEC 15504.

**Tabla 3.1** Resumen niveles de servicio y capacidad Agil\_Mantema

			Nivel de capacidad		
			Uno	Dos	Tres
Nivel de servicio	Básico	Tipo mantenimiento	<ul style="list-style-type: none"> <li>▪ Correctivo urgente</li> </ul>		
		Interfaz fundamental	<ul style="list-style-type: none"> <li>▪ Soporte al cliente (L o F)</li> <li>▪ Gestión de resolución de problemas (L o F)</li> </ul>		
	Intermedio	Tipo mantenimiento	<ul style="list-style-type: none"> <li>▪ Correctivo urgente</li> </ul>	<ul style="list-style-type: none"> <li>▪ Correctivo no urgente</li> <li>▪ Perfectivo</li> </ul>	
		Interfaz fundamental	<ul style="list-style-type: none"> <li>▪ Soporte al cliente (F)</li> <li>▪ Gestión de resolución de problemas (F)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Gestión de la configuración (L o F)</li> <li>▪ Aseguramiento de la calidad (L o F)</li> </ul>	
	Avanzado	Tipo mantenimiento	<ul style="list-style-type: none"> <li>▪ Correctivo urgente</li> </ul>	<ul style="list-style-type: none"> <li>▪ Correctivo no urgente</li> <li>▪ Perfectivo</li> </ul>	<ul style="list-style-type: none"> <li>▪ Adaptativo</li> <li>▪ Preventivo</li> </ul>
		Interfaz fundamental	<ul style="list-style-type: none"> <li>▪ Soporte al cliente (F)</li> <li>▪ Gestión de resolución de problemas (F)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Gestión de la configuración (F)</li> <li>▪ Aseguramiento de la calidad (F)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Gestión de cambios de requisitos (L o F)</li> <li>▪ Gestión de proyectos (L o F)</li> </ul>

### 3.1.1. Tipos de mantenimiento

Se definen cinco tipos de mantenimiento, los mismos identificados en Mantema. Éstos se clasifican en planificables y no planificables para lograr una mejor gestión y optimización del Registro de peticiones, ofreciendo un criterio al Gestor de Peticiones (rol que se encarga de ordenar las peticiones de modificación) para clasificar y priorizar las peticiones de mantenimiento.

Los tipos de mantenimiento son:

#### **No planificables:**

- *Correctivo urgente (nivel básico):* es aquel que se da en situaciones en que existe un error que bloquea el producto software en explotación, deben ser atendidas y resueltas en el menor tiempo posible.

#### **Planificables:**

- *Correctivo no urgente (nivel intermedio):* se produce cuando existe un error en el producto software en explotación que no es crítico, pero que impide el normal funcionamiento de ésta durante un período de tiempo relativamente corto.
- *Perfectivo (nivel intermedio):* añade al producto software en explotación nuevas funcionalidades requeridas por el cliente.
- *Adaptativo (nivel avanzado):* se aplica ante situaciones en que el producto software en explotación se ajusta para continuar funcionando correctamente en un entorno distinto al concebido originalmente.

- *Preventivo (nivel avanzado)*: se lleva a cabo cuando se desea mejorar las características internas del producto software en explotación para disminuir el esfuerzo de mantenimiento en el futuro.

### 3.1.2. Roles dentro del proceso de mantenimiento

Agil\_Mantema define a los participantes del proceso de mantenimiento según roles, para describir habilidades y responsabilidades específicas, sin relacionar directamente estas características con un integrante particular de la organización. Por lo tanto, los roles de Agil\_Mantema son:

- *Cliente*: es la organización propietaria del producto, por lo tanto es quien recibe el servicio de mantenimiento. El propietario del producto representa a todos los interesados en el producto final. Sus áreas de responsabilidad son: el financiamiento del proyecto, retorno de la inversión del proyecto y el lanzamiento del proyecto. El propietario del producto por lo general formula peticiones de modificación del tipo perfectivo o adaptativo.
- *Usuario*: es quien utiliza el software. Propone las peticiones de modificaciones correctivas y perfectivas.
- *Mantenedor*: es quien realiza las modificaciones del software. Se pueden distinguir tres responsabilidades diferentes:
  - *Gestor de Peticiones*: es quien acepta o rechaza las peticiones de modificación y decide el tipo de mantenimiento que corresponde. En caso de ser perfectivo pone al tanto al Cliente para conocer la viabilidad del mantenimiento. Si es cualquier otro tipo de mantenimiento se sitúa la petición a la Lista de Espera de Peticiones asignándole su prioridad.
  - *Responsable de Mantenimiento*: es el responsable de preparar el proceso y establece las normas y procedimientos necesarios para aplicar la metodología. Interactúa con el cliente y el equipo de mantenimiento, siendo miembro de éste último, trabajando a la par con sus miembros, coordinando encuentros del equipo y, además, se encarga de los eventuales problemas que puedan ocurrir. También, es el responsable que las prácticas, valores y reglas de Scrum se lleven a cabo.
  - *Equipo de Mantenimiento*: es un grupo de personas que implementa las peticiones de mantenimiento. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos. También puede proponer peticiones de mantenimiento preventivo.

### 3.1.3. Proceso de mantenimiento

Agil\_Mantema comienza al momento en que el Cliente solicita el servicio de mantenimiento, con ello la organización mantenedora (quién ofrece el servicio de mantenimiento) se reúne con el Cliente para fijar los roles que desempeñará el persona.

Por su parte, el Equipo de Mantenimiento deberá conocer la aplicación a mantener y preparará los ambientes de prueba. Así también se fijará el protocolo que se deberá seguir para solicitar mantenimiento. Todo esto se ejecutará en el proceso de Planificación del proceso.

Una vez fijados los aspectos básicos para la actividad de mantenimiento, se comienza con la atención de las peticiones de modificación, donde el Gestor de Peticiones las recibirá y corroborará que sean peticiones validas, de ser así las clasificará según el tipo de mantenimiento que necesite y le asignará una prioridad. Finalmente, la petición de modificación será almacenada en el registro de peticiones, donde están todas las peticiones que deben ser atendidas ordenadas según su prioridad (lista de espera). Estas tareas se llevan a cabo en el proceso de Atención de la petición. En este proceso se inicia el ciclo de iteraciones de mantenimiento.

Si la petición de modificación que se debe atender requiere mantenimiento correctivo urgente, ésta pasa al proceso de SprintM No Planificable. De lo contrario, si la petición de modificación requiere mantenimiento: correctivo no urgente, perfectivo, preventivo o adaptativo, pasa al proceso de SprintM Planificable. En ambos procesos se realizan tareas donde se analiza lo requerido en la petición de modificación, se plantean alternativas para corregir o modificar el producto de software que está siendo mantenido, se ejecutan las acciones necesarias para resolver el problema planteado y, finalmente, se llevan a cabo las pruebas requeridas para comprobar el desempeño de los cambios efectuados al producto de software.

Paralelamente a la ejecución de los procesos SprintM No Planificable o SprintM Planificable se lleva a cabo el proceso Seguimiento del SprintM. Donde se establecen reuniones periódicas del Equipo de Mantenimiento para evaluar el estado de las tareas que se están desarrollando. Así también presenta una pequeña guía para llevar estas reuniones, donde cada miembro del equipo presenta: el estado de las tareas que le han sido asignadas, da a conocer los problemas que han tenido durante la ejecución de sus actividades y plantea una serie de objetivos a cumplir antes de la próxima reunión. Con este proceso se puede controlar el estado de cada una de las tareas que están realizando los miembros del Equipo de Mantenimiento.

Una vez terminado el proceso de SprintM No Planificable o SprintM Planificable, para dar término a la iteración de mantenimiento, se pasa al proceso de Finalizar intervención. En este proceso se verifica que los cambios realizados al producto de software eliminan el error o modifican lo requerido por el Usuario. Una vez comprobado esto, se procede a ejecutar el producto de software intervenido en el ambiente de producción de la organización. Además, se actualiza el manual de usuario, agregando la información necesaria referente a la modificación ejecutada en el producto de software. Finalmente, se realiza una tarea de retroalimentación donde se rescatan las lecciones aprendidas de la iteración de mantenimiento realizada y, además, se define la próxima iteración del ciclo. Por otra parte, se realiza el proceso de Retiro, en el cual se planifica y lleva a cabo el retiro del producto de software antiguo.

Al terminar el periodo de prestación del servicio de mantenimiento se pasa a la actividad de Finalización del servicio. Con este proceso se concluye formalmente la relación existente entre la organización mantenedora y el Cliente. La organización mantenedora hace entrega al Cliente de todos los artefactos generados durante la prestación del servicio y legaliza el término de las actividades de mantenimiento. Este proceso puede requerir del proceso de Retiro, para retirar el producto de software antiguo y dejar como una aplicación operativa en el ambiente de producción al producto de software que ha sido mantenido y sometido a modificaciones.

Todo el proceso de mantenimiento se representa en la Figura 3.2.

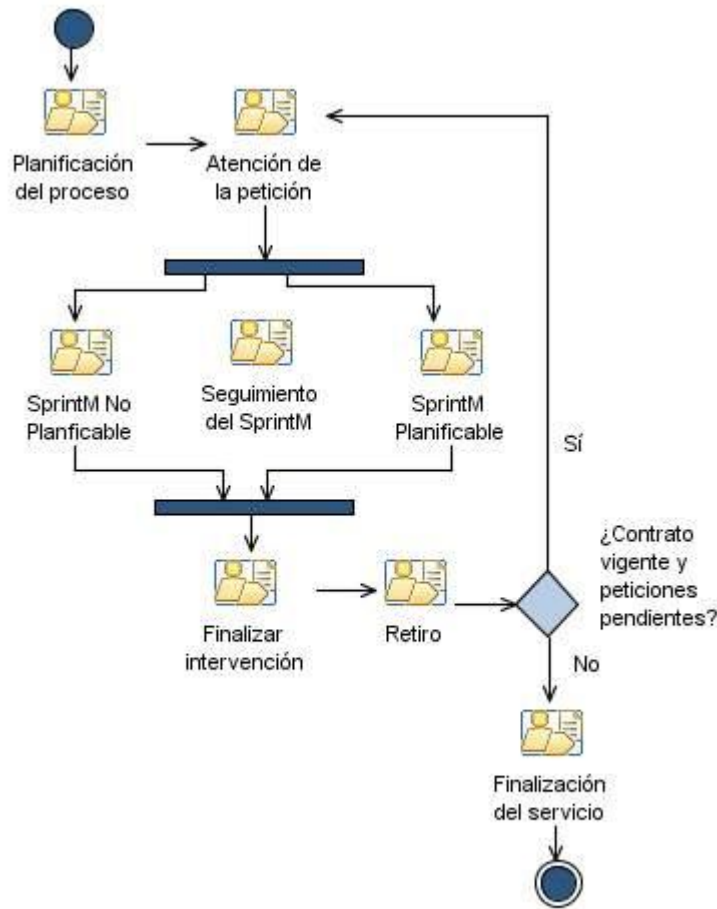


Figura 3.2 Proceso de Agil\_Mantema

### 3.1.4. Detalles de los procesos

A continuación se detallarán todos los procesos de Agil\_Mantema y sus respectivas tareas, definiendo las entradas y salidas de cada uno.

#### 3.1.4.1. I0: Planificación del proceso

El propósito de esta actividad es llevar a cabo la planificación del proceso de mantenimiento. La actividad inicial planificación del proceso sólo se ejecuta cuando el Cliente contacta con la Organización Mantenedora para que realice el proceso de mantenimiento.

##### *Tareas:*

*I0.1: Asignar Responsables:* se asigna el personal adecuado a los roles que actúan del proceso de mantenimiento Al finalizar esta tarea, el Gestor de Peticiones, Responsable de



Mantenimiento y Equipo de Mantenimiento están claramente establecidos e identificados en la organización.

*10.2: Adquirir conocimiento del producto software:* el Equipo de Mantenimiento obtiene la información del Producto de Software en Explotación que se va a mantener. Luego el Equipo de Mantenimiento estudia la documentación, código fuente, referencias cruzadas, se entrevista con los usuarios, observar cómo trabaja éste, con el fin de obtener el conocimiento necesario y suficiente del Producto de Software en Explotación a mantener. Durante el tiempo que dure esta tarea no se realiza ningún tipo de mantenimiento. Al finalizar esta tarea, el Equipo de Mantenimiento entrega al Cliente un informe acerca de los elementos del producto software y el estado de éstos, de manera que el Cliente verifique que el Equipo de Mantenimiento ha adquirido un conocimiento adecuado del software.

*10.3: Preparar entornos de pruebas:* en esta tarea el Equipo de Mantenimiento prepara el entorno de pruebas. Realiza Copias de los Elementos del Producto de Software, preparar las base de datos y archivos (el entorno), que sean semejantes a la realidad y que cubran la totalidad de las funcionalidades del sistema. El objetivo es que el software pueda funcionar en un ambiente aislado, que no afecte la operación normal de los sistemas en uso.

*10.4: Definir procedimientos de petición de modificación:* el Equipo de Mantenimiento genera el documento que el Usuario presentará para solicitar mantenimiento. Dicho documento será llamado Petición de Modificación. En esta tarea también se establecerán, junto con el Cliente, los procedimientos de la Petición de Modificación y a los usuarios se les indicará quién será el receptor de dicha petición llamado Gestor de Peticiones.

**Tabla 3.2** Entradas y salidas tareas actividad de planificación del proceso

Tarea	Entradas	Salidas
I0.1	<ul style="list-style-type: none"> <li>▪ Listado de posibles responsables</li> </ul>	<ul style="list-style-type: none"> <li>▪ Listado de asignación de responsables a los roles</li> </ul>
I0.2	<ul style="list-style-type: none"> <li>▪ Producto de software en explotación</li> </ul>	<ul style="list-style-type: none"> <li>▪ Listado de elementos del producto de software</li> </ul>
I0.3	<ul style="list-style-type: none"> <li>▪ Elementos del producto de software en explotación</li> </ul>	<ul style="list-style-type: none"> <li>▪ Copias de los elementos del producto de software en explotación</li> </ul>
I0.4	<ul style="list-style-type: none"> <li>▪ Plan de mantenimiento</li> </ul>	<ul style="list-style-type: none"> <li>▪ Petición de modificación</li> <li>▪ Procedimientos</li> </ul>

### 3.1.4.2. I1: Atención de la petición

El propósito de esta actividad es recibir una Petición de Modificación, a la que se le asigna un tipo de mantenimiento y su prioridad, además se almacena en el Registro de Peticiones, en el que se encuentra una Lista de Espera ordenada por prioridad. En esta actividad comienzan las iteraciones del ciclo de mantenimiento para atender a cada Petición de Modificación.

### **Tareas:**

*11.1: Recibir petición de modificación:* el Usuario entrega una Petición de Modificación, que es recibida y registrada por el Gestor de Peticiones. Este deberá asignar un identificador único a cada Petición de Modificación.

*11.2: Decidir el tipo de mantenimiento:* a partir de la Petición de Modificación Recibida y Registrada, de la tarea anterior, el Gestor de Peticiones decide aceptar o rechazar la petición. Si la petición es aceptada se decide el tipo de mantenimiento que debe aplicarse y es agregada al Registro de Peticiones asignándole una prioridad. Si la petición es rechazada se debe justificar las razones. Finalmente se analiza la relación entre las peticiones que están en el Registro de Peticiones y se decide cuáles pueden abordarse en forma conjunta (Lista de Espera). También se realiza la estimación del esfuerzo necesario para llevar a cabo la Petición de Modificación.

**Tabla 3.3** Entradas y salidas tareas actividad de atención de la petición

Tarea	Entradas	Salidas
11.1	<ul style="list-style-type: none"><li>▪ Petición de modificación</li></ul>	<ul style="list-style-type: none"><li>▪ Petición de modificación recibida y registrada</li></ul>
11.2	<ul style="list-style-type: none"><li>▪ Petición de modificación recibida y registrada</li></ul>	<ul style="list-style-type: none"><li>▪ Decisión sobre el tipo de mantenimiento</li><li>▪ Lista de espera</li><li>▪ Petición de modificación</li><li>▪ Registro de peticiones</li></ul>

### **3.1.4.3. SNP: SprintM no planificable**

El propósito de esta actividad es brindar atención urgente a las Peticiones de Modificación que bloquean o interrumpen el funcionamiento del Producto de Software en Explotación. El Sprint del Mantenimiento No Planificable se ejecuta cuando se asume un mantenimiento correctivo urgente. Estas actividades se ejecutan cuando el error presentado en la Petición de Modificación paraliza de manera seria el funcionamiento normal del resto del sistema de información y el de la organización, de forma que la corrección del error deba ser inminente. Se recomienda la ejecución de SprintM cortos, entre uno y siete días (dependiendo del tipo de error) con reuniones diarias.

Esta actividad se compone de dos subactividades.

#### **SNP1: Análisis del error**

En esta subactividad el Equipo de Mantenimiento, junto al Usuario, verifica el error informado en la Petición de Modificación y sus causas. Se plantean alternativas de solución y se genera la Lista de Elementos Software por Corregir.

### **Tarea:**

*SNP1.1: Investigar y analizar causas:* el Equipo de Mantenimiento analiza la Petición de Modificación, verifica el problema con la colaboración del Usuario que realizó la petición y reproduce el error. Además se estudian diferentes alternativas para implementar la

modificación que permitirá corregir el error. También se construye una lista de los elementos software a corregir (módulos, rutinas, documentos, etc.).

**Tabla 3.4** Entradas y salidas tarea subactividad de análisis de error

Tarea	Entradas	Salidas
SNP1.1	<ul style="list-style-type: none"> <li>▪ Petición de modificación</li> <li>▪ Producto de software en explotación con error crítico</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lista de elementos de software por corregir</li> </ul>

### **SNP2: Intervención correctiva urgente**

En esta subactividad el Equipo de Mantenimiento ejecuta todas las acciones que sean necesarias para corregir el error lo antes posible, se identifican los elementos del producto de software que fueron afectados por las correcciones y, finalmente, se realizan pruebas unitarias a los elementos que se vieron afectados en este proceso.

#### **Tareas:**

*SNP2.1: Realizar acciones correctivas:* el Equipo de Mantenimiento ejecuta las acciones necesarias para corregir el problema detectado. Se debe identificar todos los elementos del producto de software (rutinas, bases de datos, etc.) afectados por la intervención.

*SNP2.2: Ejecutar pruebas unitarias:* el Equipo de Mantenimiento debe comprobar el correcto funcionamiento de todos los cambios realizados. Las pruebas realizadas se deben registrar en el documento de Pruebas Unitarias Realizadas. Esta tarea sirve para comprobar la correcta operación del módulo al que se le han practicado las acciones correctivas.

**Tabla 3.5** Entradas y salidas de la tarea de la subactividad de intervención correctiva urgente

Tarea	Entradas	Salidas
SNP2.1	<ul style="list-style-type: none"> <li>▪ Lista de elementos de software por corregir</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lista de elementos de software corregidos</li> </ul>
SNP2.2	<ul style="list-style-type: none"> <li>▪ Casos de prueba</li> <li>▪ Lista de elementos de software corregidos</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lista de elementos de software corregidos y aprobados</li> <li>▪ Pruebas unitarias realizadas</li> </ul>

#### **3.1.4.4. SP: SprintM planificable**

El propósito de esta actividad es brindar atención a las Peticiones de Modificación que afectan de alguna manera el funcionamiento del Producto de Software en Explotación. El Sprint de Mantenimiento Planificable se ejecuta cuando se asumen los mantenimientos: correctivo no urgente, perfectivo, preventivo y adaptativo. Al ser un Sprint con menor urgencia que el anterior se recomienda la ejecución de SprintM más largos, de entre ocho y quince días (dependiendo del tipo de mantenimiento) con reuniones cada dos días.

Esta actividad se compone de dos subactividades.

## SP1: Análisis de la petición

En esta subactividad, según el tipo de mantenimiento que se asigna a la Petición de Modificación, se proponen alternativas para las modificaciones que deben realizarse.

### *Tarea:*

*SP1.1: Analizar y elegir solución:* si se trata de Mantenimiento correctivo no urgente o perfectivo (CP), se documenta la causa del error y se indican las posibles alternativas de implementación de la solución en el Diagnóstico Error y Posibles Soluciones. Si se trata de Mantenimiento preventivo o adaptativo (PA), solamente se indican las posibles alternativas de implementación en el Diagnóstico Error y Posibles Soluciones. Luego de analizar cada una de las posibles soluciones de la Petición de Modificación, se elige la alternativa de implementación adecuada, y se rellena el Diagnóstico Error y Posibles Soluciones, indicando la alternativa elegida para la corrección.

**Tabla 3.6** Entradas y salidas tarea subactividad de análisis de la petición

Tarea	Entradas	Salidas
SP1.1	<ul style="list-style-type: none"><li>▪ Alternativas de implementación</li><li>▪ Petición de modificación</li><li>▪ Producto de software en explotación</li></ul>	<ul style="list-style-type: none"><li>▪ Alternativa seleccionada</li></ul>

## SP2: Intervención y pruebas

En esta subactividad el Equipo de Mantenimiento lleva a cabo las modificaciones necesarias, realiza las pruebas unitarias y de sistema, y se ejecuta el nuevo producto de software junto al antiguo para corroborar que no haya ningún problema.

### *Tareas:*

*SP2.1: Ejecutar intervención:* el Equipo de Mantenimiento debe ejecutar las acciones necesarias para ofrecer una solución a la Petición de Modificación conforme con la alternativa seleccionada, la cual está registrada en el documento Diagnóstico Error y Posibles Soluciones. Se deben identificar todos los componentes del producto software (rutinas, bases de datos, etc.) afectadas por la intervención.

*SP2.2: Ejecutar pruebas unitarias y de integración (CP/P/A):* el Equipo de Mantenimiento realiza las pruebas unitarias y de integración sobre el Producto de Software Intervenido. Se debe comprobar que la Petición de Modificación queda atendida y de que los diferentes elementos de software funcionan correctamente en forma conjunta. Una vez finalizadas, se genera el documento de Pruebas Realizadas. El propósito es probar el correcto funcionamiento del software tanto en módulos independientes como en todo el sistema.

*SP2.3: Ejecutar paralelamente el producto de software antiguo y el nuevo:* el Equipo de Mantenimiento ejecuta operaciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso. Pueden aplicarse pruebas de no regresión, de manera que no se repitan errores anterior a la intervención.

**Tabla 3.7** Entradas y salidas tarea subactividad de intervención y pruebas

Tarea	Entradas	Salidas
SP2.1	<ul style="list-style-type: none"> <li>▪ Copia del producto de software (A)</li> <li>▪ Diagnostico del error (CP)</li> <li>▪ Mejora a realizar (P)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Copia adaptada (A)</li> <li>▪ Producto de software intervenido</li> </ul>
SP2.2	<ul style="list-style-type: none"> <li>▪ Casos de prueba</li> <li>▪ Producto de software intervenido</li> </ul>	<ul style="list-style-type: none"> <li>▪ Producto de software intervenido comprobado</li> <li>▪ Pruebas realizadas</li> </ul>
SP2.3	<ul style="list-style-type: none"> <li>▪ Producto de software intervenido comprobado</li> </ul>	<ul style="list-style-type: none"> <li>▪ Producto de software intervenido comprobado y en correcto funcionamiento</li> </ul>

### 3.1.4.5. SSM: Seguimiento del SprintM

El propósito de esta actividad es conducir reuniones de control para hacer un seguimiento del estado de avance y resolver problemas de las intervenciones realizadas mediante el o los Sprint de Mantenimiento (SprintM). En esta actividad se lleva a cabo la revisión del trabajo realizado y se solucionan las dificultades encontradas. Estas actividades son comunes para los dos tipos de SprintM (planificable y no planificable).

**Tareas:**

*SSM1.1: Reuniones habituales:* en estas reuniones, de al menos 15 minutos, se reúne todo el Equipo de Mantenimiento y cada miembro del equipo expone sólo los siguientes temas: ¿Qué es lo que hizo desde la última reunión?; ¿Qué es lo que va hacer hasta la siguiente reunión? Es muy importante que al salir de la reunión todos los involucrados sepan lo que deben hacer y que todos están alineados en la misma dirección: el mantenimiento del software; ¿Cómo se lo llevará a cabo? ¿Algo le hace falta? Todos deben tener claro cómo realizar su trabajo, con esta pregunta deben surgir los problemas que tienen las personas para la realización de la Petición de Modificación. Sólo se tratan estos temas para que la reunión sea rápida y no se pierda tiempo, su finalidad es alinear a las personas en la misma dirección y sacar a la luz los problemas e impedimentos que hay para solucionarlos y conseguir el objetivo.

*SSM1.2: Seguir los cambios:* se realiza el control de los cambios producidos en la ejecución del SprintM. Esto abarca: realizar la traza de los cambios que la Petición de Modificación ha provocado a lo largo de los procesos de desarrollo implicados; verificar que se han realizado satisfactoriamente las pruebas unitarias, de integración y del sistema que se consideraron necesarias para los componentes a modificar; comprobar que sólo se ha modificado lo establecido y, en caso contrario, justificar el motivo; y llevar el control de los distintos desarrollos existentes en paralelo sobre un mismo componente, con el fin de coordinar las modificaciones incluidas en cada uno de ellos, y asegurar que en el paso a producción se implanten correctamente.

**Tabla 3.8** Entradas y salidas tareas actividad de seguimiento del SprintM

Tarea	Entradas	Salidas
SSM1.1	<ul style="list-style-type: none"> <li>▪ Informe ejecutivo de las tareas realizadas</li> <li>▪ Problemas encontrados</li> </ul>	<ul style="list-style-type: none"> <li>▪ Informe ejecutivo de las tareas realizadas actualizado</li> <li>▪ Problemas solucionados</li> </ul>
SSM1.2	<ul style="list-style-type: none"> <li>▪ Producto de software en mantenimiento</li> <li>▪ Productos de software relacionados</li> </ul>	<ul style="list-style-type: none"> <li>▪ Validación y control de los cambios realizados</li> </ul>

### 3.1.4.6. F1: Finalización de la intervención

En esta actividad el equipo de mantenimiento junto al usuario verifican el correcto funcionamiento del nuevo producto de software en el entorno de producción. Además, se realizan las actualizaciones necesarias al Manual de Usuario. Para finalizar, se define cuál será la próxima iteración del ciclo de mantenimiento. Con esto se da por terminado un proceso de intervención relacionado con una Petición de Modificación.

**Tareas:**

*F1.1: Verificar y validar corrección con el cliente:* el Equipo de Mantenimiento y la organización usuaria del sistema se reúnen para comprobar que el producto intervenido funciona correctamente. En esta tarea debe haber interacción con el Cliente para recabar impresiones, sugerencias, mejoras y relevancia que le da al producto que se ha entregado. También se evalúan y registran nuevos posibles cambios en el Registro de Peticiones.

*F1.2: Pasar a producción:* el Equipo de Mantenimiento pasa al entorno de producción el software corregido para su utilización por parte de los Usuarios.

*F1.3: Documentar manual de usuario:* el Equipo de Mantenimiento debe documentar o re-documentar el Manual de Usuario, si es que ha cambiado el modo de operación del software o si se han agregado nuevas funcionalidades.

*F1.4: Registrar la intervención:* la intervención de modificación queda registrada, según los procedimientos de la organización.

*F1.5: Reunión de retrospectiva:* se deben extraer las mejores prácticas de la última intervención. Aquí el Responsable de Mantenimiento pregunta a su Equipo de Mantenimiento: ¿Qué aspectos positivos rescatan del último Sprint? ¿Qué se puede mejorar? Toda esta información es tomada para optimizar el próximo SprintM. Una labor muy importante a realizar en esta tarea es la definición y anuncio del próximo Sprint de Mantenimiento a ejecutar.

**Tabla 3.9** Entradas y salidas tareas actividad de finalización de la intervención

Tarea	Entradas	Salidas
F1.1	<ul style="list-style-type: none"> <li>▪ Elementos de software corregidos y aprobados</li> <li>▪ Pruebas realizadas</li> </ul>	<ul style="list-style-type: none"> <li>▪ Documento de aceptación de la corrección validado por el cliente</li> <li>▪ Producto de software totalmente comprobado</li> </ul>
F1.2	<ul style="list-style-type: none"> <li>▪ Producto de software totalmente comprobado</li> </ul>	<ul style="list-style-type: none"> <li>▪ Producto de software totalmente comprobado en explotación</li> </ul>
F1.3	<ul style="list-style-type: none"> <li>▪ Manual de usuario</li> <li>▪ Producto de software totalmente comprobado</li> </ul>	<ul style="list-style-type: none"> <li>▪ Manual de usuario actualizado</li> </ul>
F1.4	<ul style="list-style-type: none"> <li>▪ Documentación generada en esta etapa</li> </ul>	<ul style="list-style-type: none"> <li>▪ Información registrada de la intervención</li> </ul>
F1.5	<ul style="list-style-type: none"> <li>▪ Lecciones aprendidas del SprintM</li> </ul>	<ul style="list-style-type: none"> <li>▪ Anuncio del próximo SprintM</li> <li>▪ Sugerencias para mejorar el SprintM</li> </ul>

### 3.1.4.7. F2: Retiro

En esta actividad el Equipo de Mantenimiento genera la planificación para el retiro del Producto de Software Antiguo en Explotación. Esto se comunica formalmente a los Usuarios para que no hayan problemas durante este proceso. Se retira la aplicación antigua y se respaldan los datos que ésta pueda contener.

**Tareas:**

*F2.1: Desarrollar plan de retirada:* el Equipo de Mantenimiento redacta un documento en el que describe cuándo y cómo se llevará a cabo el retiro del Producto de Software Antiguo en Explotación.

*F2.2: Notificar futuro retiro:* el Equipo de Mantenimiento notifica al Cliente el momento en el que se ejecutará la retirada del software.

*F2.3: Ejecutar en paralelo:* el Usuario, con el visto bueno del Cliente y bajo la supervisión del Equipo de Mantenimiento, realiza operaciones reales sobre el software que se va a retirar y el software nuevo a implantar (si es que va a ser sustituido).

*F2.4: Notificar y ejecutar retiro:* se notifica la inminencia de la retirada. El Producto de Software Antiguo en Explotación es retirado y el Producto de Software Nuevo es el que queda en funcionamiento para explotación.

*F2.5: Almacenar datos del producto de software antiguo:* los datos del Producto de Software Antiguo Retirado son almacenados a modo de prevención ante cualquier necesidad de éstos.

**Tabla 3.10** Entradas y salidas tareas actividad de retiro

Tarea	Entradas	Salidas
F2.1	<ul style="list-style-type: none"> <li>▪ Producto de software antiguo en explotación</li> <li>▪ Producto de software nuevo</li> </ul>	<ul style="list-style-type: none"> <li>▪ Plan de retiro</li> </ul>
F2.2	<ul style="list-style-type: none"> <li>▪ Plan de retiro</li> </ul>	<ul style="list-style-type: none"> <li>▪ Notificación a los usuarios del retiro</li> </ul>
F2.3	<ul style="list-style-type: none"> <li>▪ Plan de retiro</li> <li>▪ Productos de software nuevo preparados</li> </ul>	<ul style="list-style-type: none"> <li>▪ Coexistencia del producto de software antiguo con el nuevo</li> </ul>
F2.4	<ul style="list-style-type: none"> <li>▪ Plan de retiro</li> </ul>	<ul style="list-style-type: none"> <li>▪ Producto de software antiguo retirado</li> <li>▪ Producto de software nuevo en explotación</li> </ul>
F2.5	<ul style="list-style-type: none"> <li>▪ Datos del entorno antiguo</li> </ul>	<ul style="list-style-type: none"> <li>▪ Datos del entorno antiguo guardados</li> </ul>

### 3.1.4.8. F3: Finalización del servicio

El propósito de esta actividad es dar por finalizado, de manera formal, el servicio de mantenimiento de software al Cliente. Esta actividad se realiza cuando el mantenedor deja de prestar sus servicios a la organización Cliente.

**Tareas:**

*F3.1: Entregar inventario y la documentación:* se entregan los productos de software generados y modificados al Cliente.

*F3.2: Terminar entrega del servicio:* la Organización de mantenimiento deja de prestar sus servicios al Cliente con carácter definitivo

**Tabla 3.11** Entradas y salidas tareas actividad de finalización del servicio

Tarea	Entradas	Salidas
F3.1	<ul style="list-style-type: none"> <li>▪ Ninguna</li> </ul>	<ul style="list-style-type: none"> <li>▪ Documentos y productos generados durante el proceso de mantenimiento</li> </ul>
F3.2	<ul style="list-style-type: none"> <li>▪ Ninguna</li> </ul>	<ul style="list-style-type: none"> <li>▪ Documento de formalización termino de prestación del servicio</li> </ul>

## 3.2. Proceso de modelado en EPF Composer

A continuación se presentarán los pasos que permiten realizar un modelado básico en la herramienta Eclipse Process Framework Composer. Estos pasos fueron utilizados para llevar a cabo el modelado del proceso de mantenimiento Agil.

Como guía para el modelado se utilizó el texto *Guía de Uso de SPEM2 con EPF Composer* [Ruiz y Verdugo, 2008]. Éste presenta información detallada sobre la notación SPEM2 y cómo se aplica en la herramienta EPF Composer, además de explicar cómo se crean los procesos en esta aplicación.



Para la realización de modelado en EPF Composer se siguieron los siguientes pasos:

**Paso 1:**

Se debe crear una biblioteca de métodos, donde se podrán almacenar los plug-ins y configuraciones de la metodología que se está modelando. Para esto, se debe dar un nombre a la librería y seleccionar la ruta donde será guardada.

Ruta de ejecución: File → New → Method Library.

Una vez creada la librería, en la pestaña *Library* aparecerá un ícono llamado *Configurations*, éste se crea automáticamente al crear la librería. Una librería puede estar compuesta por una o más configuraciones y uno o más plug-ins.

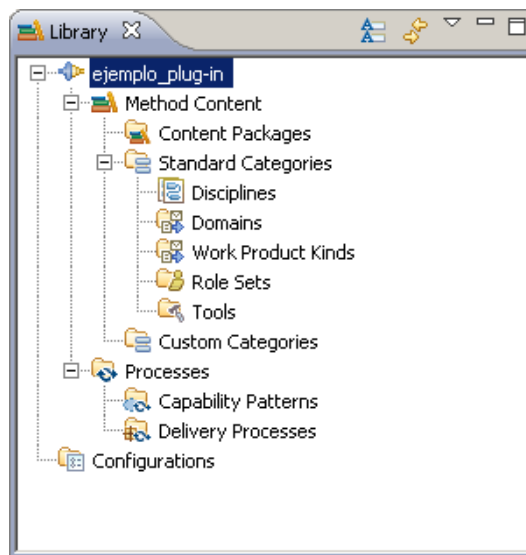
**Paso 2:**

A continuación, se debe dar un nombre y crear un plug-in para almacenar el contenido de método y los procesos por separado, tal como se establece en SPEM2.

Ruta de ejecución: File → New → Method Plug-in.

Junto con la creación del plug-in aparecerán los siguientes elementos (Figura 3.3):

- *Method Content*: el que contendrá: Content Packages, Estándar Categories (Disciplines, Domains, Work Product Kinds, Role Sets y Tools) y Custom Categories.
- *Processes*: el que contendrá: Capability Patterns y Delivery Patterns.



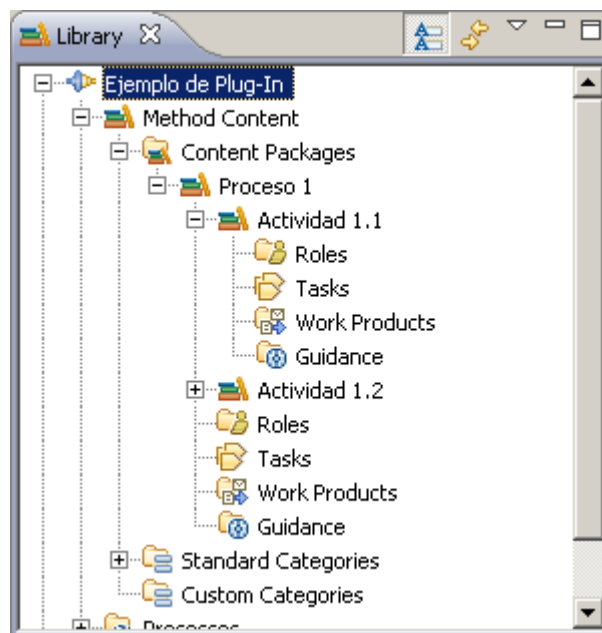
**Figura 3.3** Elementos de un plug-in

**Paso 3:**

El siguiente paso es crear un paquete de contenido para poder crear los elementos básicos (roles, tareas, productos de trabajo y guías).

Ruta de ejecución: Content Packages → New → Content Package.

Los paquetes de contenido pueden permitir realizar una vista jerárquica de sus elementos, así se puede crear un paquete de contenido para un proceso que contenga otros paquetes para representar sus actividades y éstos últimos contienen sus elementos básicos. En el caso de la Figura 3.4, se tiene el Proceso 1 compuesto por dos actividades, Actividad 1.1 y Actividad 1.2. Cada una de estas actividades contendrá las tareas necesarias para desempeñarlas.



**Figura 3.4** Paquetes de contenido jerarquizados

#### ***Paso 4:***

Una vez creados los paquetes de contenido y siguiendo los conceptos de SPEM2 el contenido de método debe ser poblado, es decir, se deben crear los roles, tareas, productos de trabajo y guías involucradas en cada proceso o actividad. Cada uno de estos elementos debe contar con la información que se estime necesaria según los campos que entrega EPF Composer.

Se recomienda seguir el siguiente orden para la creación de estos elementos:

#### **Roles:**

Crear todos los roles desde un principio. Estos deberán ser creados en el paquete de contenido de mayor jerarquía, en el caso del ejemplo será el paquete de contenido llamado Proceso 1. Esto será consistente desde el punto de vista conceptual, pues los roles son comunes para todo el proceso y no particulares para ciertas actividades.

Ruta de ejecución: Content Package de mayor jerarquía → Roles → New → Role.

### Tareas, productos de trabajo y guías:

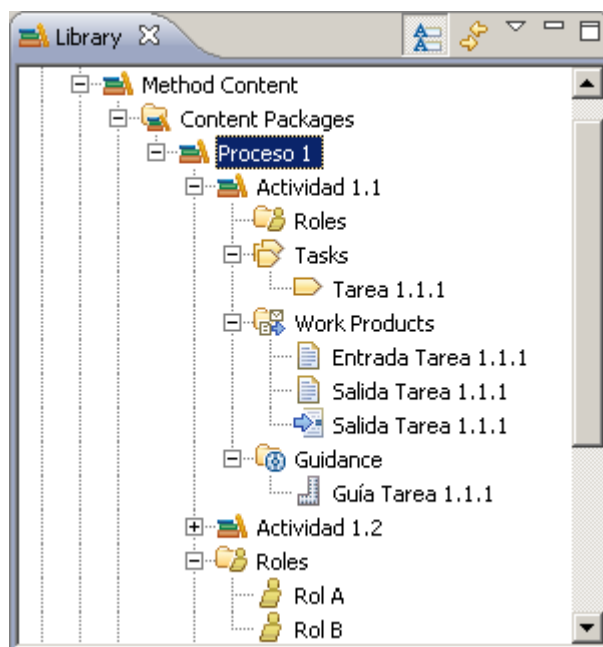
Se deben crear una a una las tareas de las actividades, junto a sus respectivos productos de trabajo y guías necesarias. De esta forma se realizará la creación completa de las tareas (ya que éstas incluyen productos de trabajo y guías) sin perder de vista algún elemento que forma parte de éstas (Figura 3.5).

Ruta de ejecución:

Content Package de la actividad → Task → New → Task.

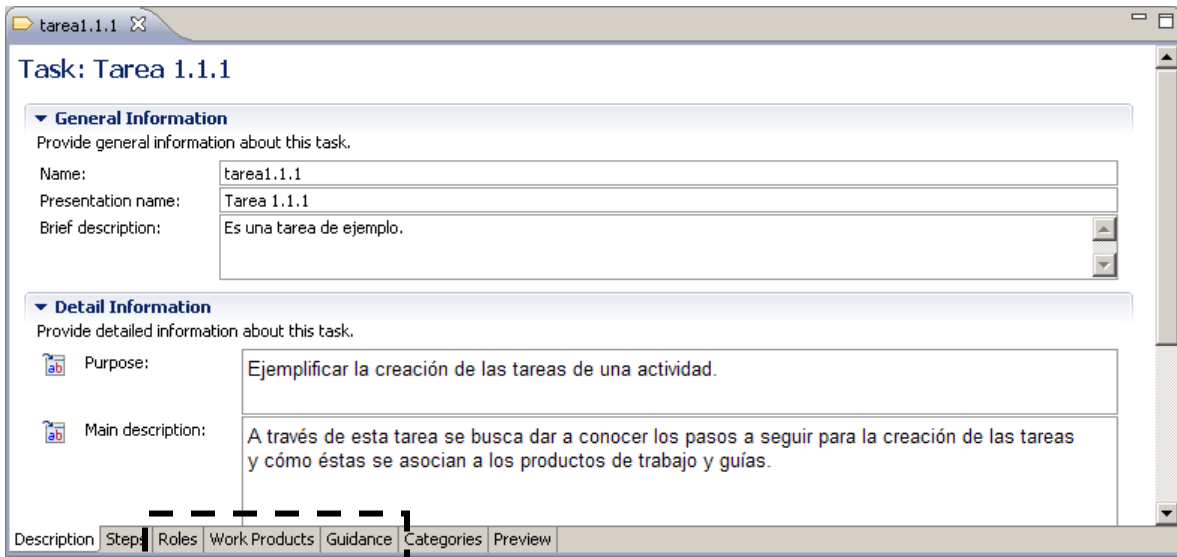
Content Package de la actividad → Work Products → New → Producto necesario.

Content Package de la actividad → Guidance → New → Guía necesaria.



**Figura 3.5** Creación de roles, tareas, productos de trabajos y guías

Una vez creada una tarea, ésta debe asociarse con los roles, los productos de trabajo y sus guías. Esto se realiza a través de las pestañas: Roles, Work Products y Guidances, ubicadas en la parte inferior de la ventana de una tarea (Figura 3.6).



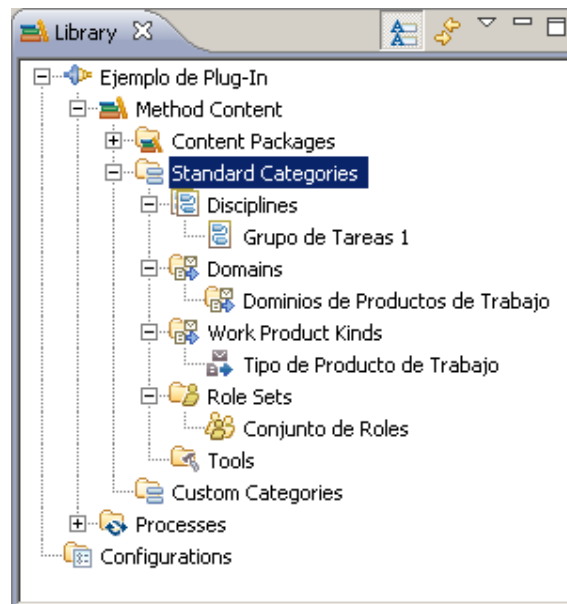
**Figura 3.6** Pestañas ventana de una tarea

Desde estas pestañas se buscan agregan los elementos respectivos asociados a la tarea, como por ejemplo: los roles (primarios y adicionales) que la realizan, los productos de trabajo de entrada y de salida, como también las guías que apoyan a dicha tarea.

***Paso 5 (Opcional):***

Se pueden crear categorías (estándar o personalizadas) para agrupar elementos del mismo tipo (roles, tareas, productos de trabajo y guías) según características comunes (Figura 3.7).

Ruta de ejecución: Content Package de mayor jerarquía → Standard Categories→ Categoría → Categoría elegida o grupo de ésta.



**Figura 3.7** Creación de categorías

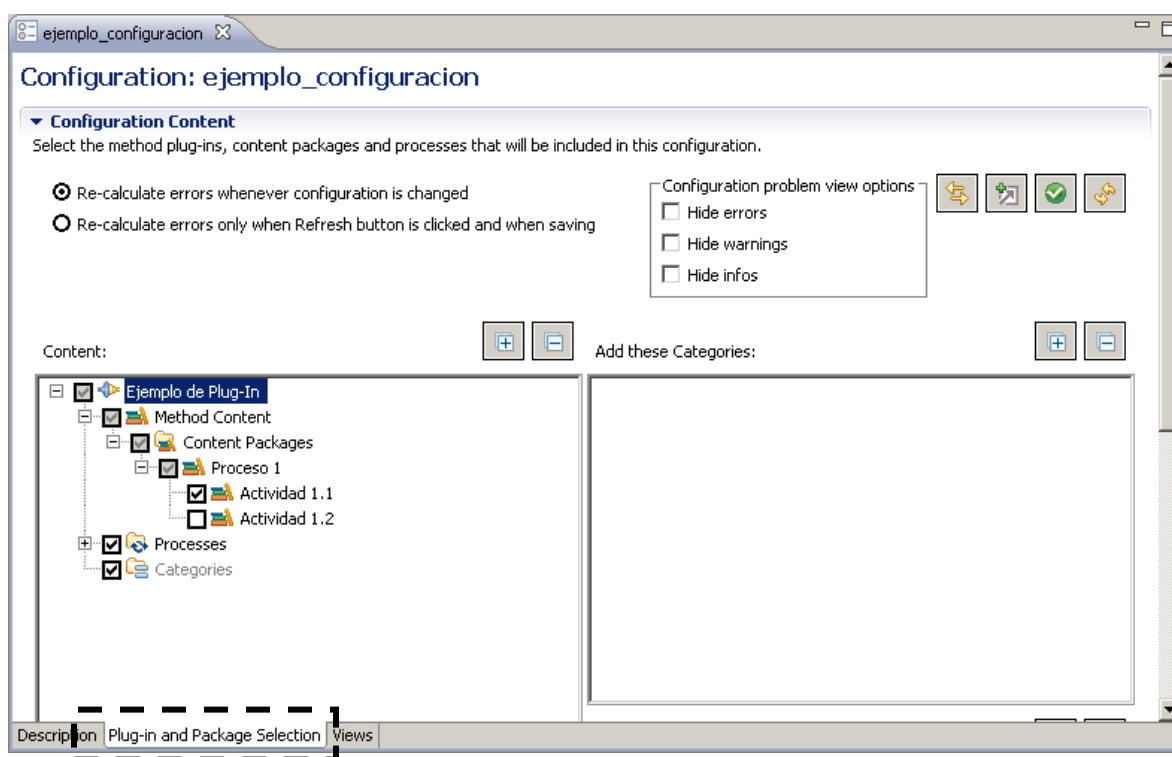
### ***Paso 6:***

Antes de poder utilizar los elementos creados para generar los procesos se debe crear al menos una configuración. Una configuración permite seleccionar que elementos serán visualizados en un proceso que utilice esta configuración, es decir, permite fijar distintas vistas.

Ruta de ejecución: File → New → Method Configuration.

Al crear la configuración se deben seleccionar los elementos que estarán disponibles para los procesos que utilizan dicha configuración, esto se hace en la pestaña Plug-in and Package Selection.

En la Figura 3.8 se muestra una configuración que permite ver todos los elementos del plug-in, salvo los pertenecientes a la Actividad 1.2.



**Figura 3.8** Creación de configuración

### ***Paso 7:***

Ya creada, al menos, una configuración es posible crear procesos. Estos pueden ser de dos tipos, Capability Patterns o Delivery Patterns. El primero es un fragmento de un proceso mayor que describe un grupo de actividades reutilizables, mientras el segundo describe una aproximación completa e integrada de un proceso.

Primeramente, deben crearse los Capability Patterns, que cuentan con paquetes de procesos para mantener un orden y establecer una jerarquía conceptual.

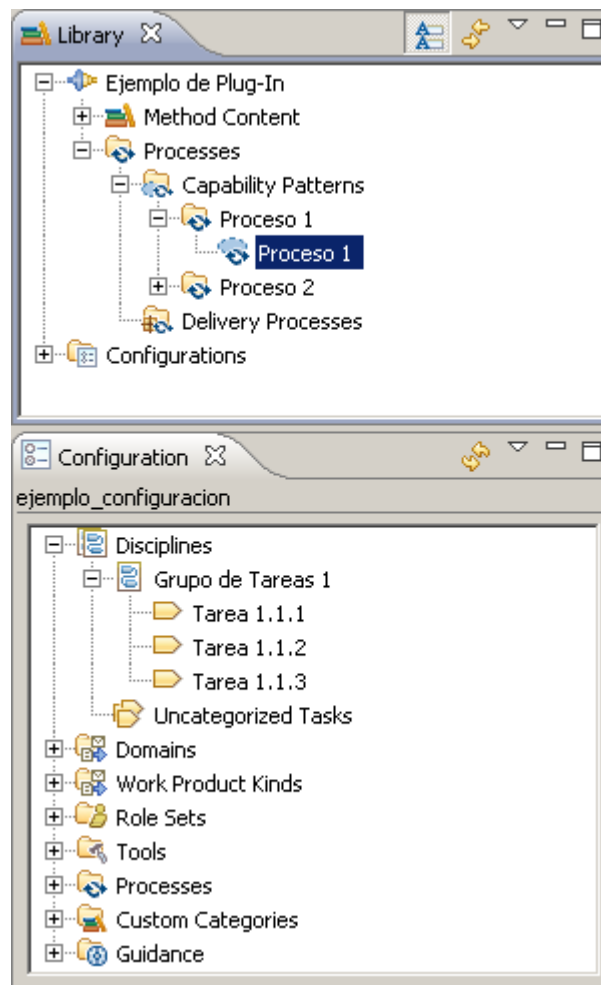
Ruta de ejecución:

Processes → Capability Patterns → Process Package.

Processes → Capability Patterns → Paquete de procesos → Capability Patterns.

Al crear el Capability Pattern aparecerá en la ventana de Configuration los elementos, agrupados por categorías, que se han seleccionado en la configuración que utiliza el proceso, en este caso, todos los elementos salvo los presentes en la Actividad 1.2.

La Figura 3.9 muestra un Capability Pattern creado y los elementos que aparecen en la ventana de Configuration. Las tareas se pueden agregar al Capability Pattern en la pestaña Work Breakdown Structure (Figura 3.10) simplemente arrastrando los elementos desde la ventana de Configuration.



**Figura 3.9** Creación de capability patterns

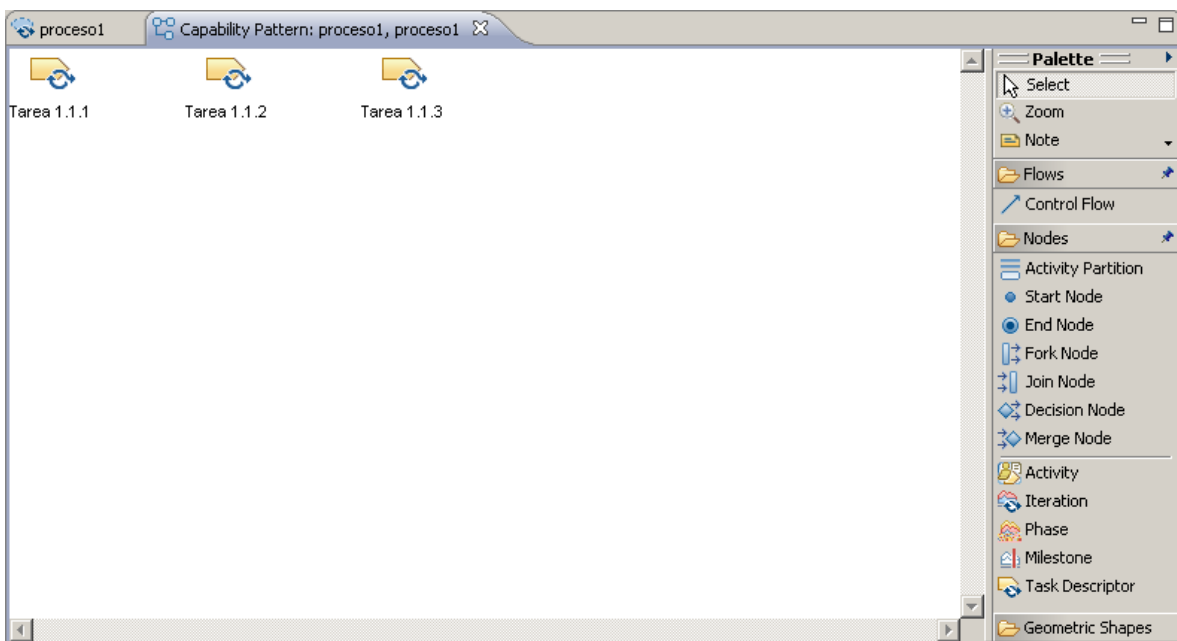
Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeat...	Multiple...	Ongoing
Proceso 1	0			Capability P...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tarea 1.1.1	1			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tarea 1.1.2	2			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tarea 1.1.3	3			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Figura 3.10** Agregación de tareas a un proceso

Desde esta misma pestaña se pueden crear los distintos tipos de diagramas asociados al proceso (diagramas de actividad, diagramas de actividad detallados y diagramas de dependencia de productos de trabajo). A modo de ejemplo, se creará un diagrama de actividad, el que es presentado en la Figura 3.11 sin ninguna modificación, tal como lo entrega la aplicación al generarlo. Se puede apreciar que aparecen menús para modificar el diagrama y completarlo, pues la aplicación solo entrega las tareas a realizar.

Ruta de ejecución:

Elegir proceso → Click derecho → Diagrams → Diagrama elegido.



**Figura 3.11** Diagrama de actividad

**Paso 8:**

Una vez creados todos los procesos (Capability Patterns) se debe crear el proceso de mayor nivel conceptual, el Delivery Pattern. Básicamente se crea a partir de los mismos pasos

llevados a cabo para la creación de un Capability Pattern, la diferencia está en que un Delivery Pattern está compuesto por los fragmentos de procesos, los Capability Pattern.

### **3.3. Modelado de Agil\_Mantema en EPF Composer**

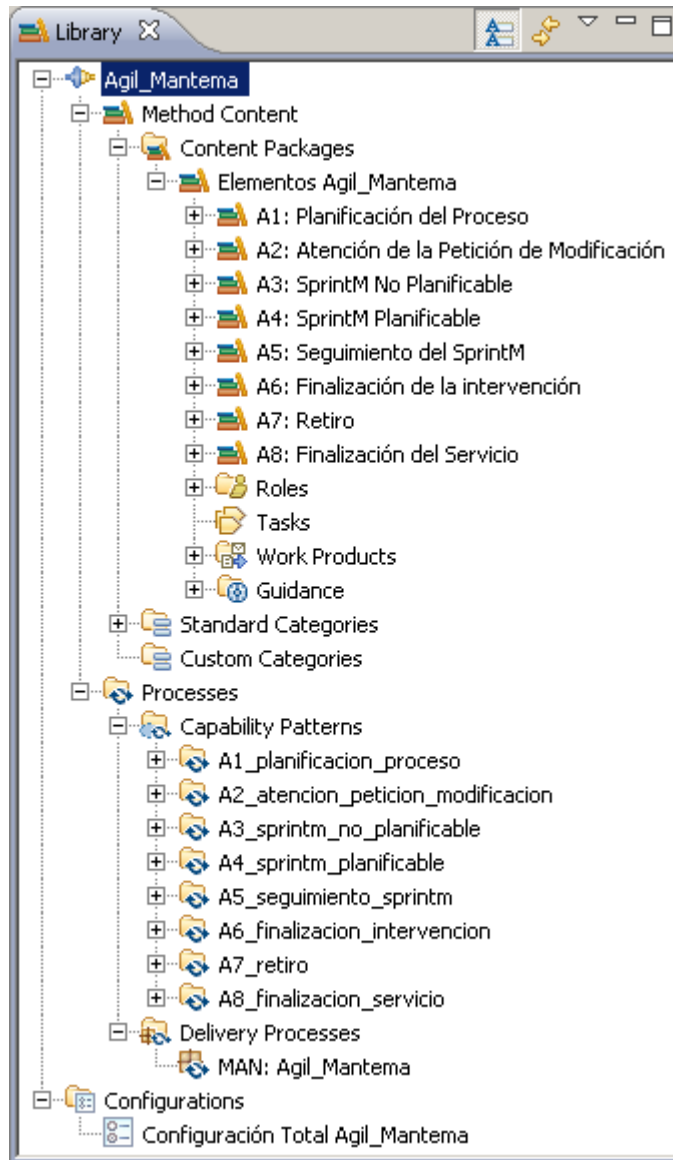
En el siguiente apartado se expondrá el modelado del proceso de mantenimiento Agil\_Mantema realizado con la herramienta EPF Composer. Primeramente se presentarán los elementos que se han modelado, para luego exponer en detalle el resultado del proceso de modelado. Por simplicidad se seguirá el orden planteado en el punto *3.1.4 Detalles del proceso* de este capítulo para presentar los procesos modelados desde la vista ofrecida para la publicación en la web que ofrece la herramienta EPF Composer.

#### **3.3.1. Elementos del modelado**

Los elementos utilizados para modelar el proceso de mantenimiento Agil\_Mantema son los expuestos en la Figura 3.12. En el contenido de método se utilizó un paquete de contenido (Elementos Agil\_Mantema) para agrupar todos los elementos relacionados a Agil\_Mantema, dentro de éste se encuentran ocho Content Packages, uno para cada actividad del proceso, y los elementos comunes para las actividades, como los roles, algunos productos de trabajo y guías, además de las categorías. Cada Content Package tiene el detalle de: las tareas, los productos de trabajo y guías, asociadas a cada actividad. Por otra parte, en los Process se incluyen: un Capability Patterns que agrupa por Process Packages que representan las actividades del proceso de mantenimiento; y un Delivery Processes, el que incluye el conjunto de procesos que conforman Agil\_Mantema. Finalmente, se creó sólo una configuración que fue utilizada para crear y presentar todos los elementos de esta sección.

Desde la Figura 3.12 hasta la Figura 3.52 se expone el modelado realizado en EPFC.






**Figura 3.12** Elementos Plug-in Agil\_Mantema

### 3.3.1.1. I0: Planificación del proceso

**Capability Pattern: I0: Planificación del Proceso**

 Es la primera actividad para comenzar a prestar los servicios de mantenimiento.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) [Collapse All Sections](#)

**Purpose**

Iniciar los servicios de mantenimiento.

[Back to top](#)

**Relationships**

**Description**

El propósito de esta actividad es llevar a cabo la planificación del proceso de mantenimiento. La actividad inicial planificación del proceso solo se ejecuta cuando el **Cliente** contacta con la **Organización Mantenedora** para que realice el proceso de mantenimiento.

**Figura 3.13** DESC I0: Planificación del proceso<sup>2</sup>

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type	Team
<input checked="" type="checkbox"/> I0.1: Asignar Responsables	●●●●	1		Task Descriptor	
<input type="checkbox"/> I0.2: Adquirir Conocimiento del Producto Software	●●●	2		Task Descriptor	
Equipo de Mantenimiento				Role Descriptor	
Usuario				Role Descriptor	
Producto de Software en Explotación				Work Product Slot	
Lista de Elementos del Producto Software				Artifact Descriptor	
<input type="checkbox"/> I0.3: Preparar entornos de pruebas	●●●●	3	2	Task Descriptor	
Equipo de Mantenimiento				Role Descriptor	
Elementos Producto Software en Explotación				Artifact Descriptor	
Copias de los Elementos del Producto de Software				Artifact Descriptor	
<input checked="" type="checkbox"/> I0.4: Definir Procedimientos de Petición de Modificación	●●●	4	3	Task Descriptor	

**Figura 3.14** WBS I0: Planificación del proceso<sup>3</sup>

<sup>2</sup> DESC: Description – Descripción.

<sup>3</sup> WBS: Work Breakdown Structure – Estructura de desglose de trabajo.

Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
+ Expand All Sections					
- Team Breakdown					
+ Expand All Sections					
Breakdown Element	Model Info	Team	Type	Planned	Multi
+ Cliente			Role Descriptor		
- Equipo de Mantenimiento			Role Descriptor		
Copias de los Elementos del Producto de Software	Modifies		Artifact Descriptor		
Lista de Elementos del Producto Software	Modifies		Artifact Descriptor		
Petición de Modificación	Modifies		Work Product Slot		
Procedimientos	Modifies		Artifact Descriptor		
I0.2: Adquirir Conocimiento del Producto Software	Performs as Owner		Task Descriptor		
I0.3: Preparar entornos de pruebas	Performs as Owner		Task Descriptor		
I0.4: Definir Procedimientos de Petición de Modificación	Performs as Owner		Task Descriptor		
+ Responsable de Mantenimiento			Role Descriptor		
+ Usuario			Role Descriptor		

Figura 3.15 TA I0: Planificación del proceso<sup>4</sup>

Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
+ Exp					
- Work Product Breakdown					
+ Exp					
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Copias de los Elementos del Producto de Software	Output				Artifact Descriptor
Elementos Producto Software en Explotación	Mandatory Input				Artifact Descriptor
Lista de Elementos del Producto Software	Output				Artifact Descriptor
Listado de Asignación de Responsables a los Roles	Output				Artifact Descriptor
Listado de Posibles Responsables	Mandatory Input				Artifact Descriptor
Petición de Modificación	Output				Work Product Slot
Plan de Mantenimiento	Mandatory Input				Artifact Descriptor
Procedimientos	Output				Artifact Descriptor
Producto de Software en Explotación	Mandatory Input				Work Product Slot


Figura 3.16 WPU I0: Planificación del proceso<sup>5</sup>

<sup>4</sup> TA: Team Allocation – Asignación de trabajo.

<sup>5</sup> WPU: Work Product Usage – Utilización de productos de trabajo.

### 3.3.1.2. I1: Atención de la petición

**Capability Pattern: I1: Atención de la Petición de Modificación**

 Permite empezar la actividad de mantenimiento atendiendo las peticiones entrantes.

**Description** | Work Breakdown Structure | Team Allocation | Work Product Usage

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Atender las Petición de Modificación entrantes.

[Back to top](#)

**Relationships**

**Description**

El propósito de esta actividad es recibir una [Petición de Modificación](#), a la cual se le asigna un tipo de mantenimiento y su prioridad, además se almacena en el [Registro de Peticiones](#), en el que se encuentra una [Lista de Espera](#) ordenada por prioridad. En esta actividad comienzan las iteraciones del ciclo de mantenimiento para atender a cada [Petición de Modificación](#).

Figura 3.17 DESC I1: Atención de la petición

**Description** | Work Breakdown Structure | Team Allocation | Work Product Usage

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type	Team
<input checked="" type="checkbox"/> I1.1: Recibir Petición de Modificación	●●	1		Task Descriptor	
Gestor de Peticiones				Role Descriptor	
Usuario				Role Descriptor	
Petición de Modificación				Work Product Slot	
Petición de Modificación Recibida y Registrada				Outcome Descriptor	
<input checked="" type="checkbox"/> I1.2: Decidir el Tipo de Mantenimiento	●●●●●	2	1	Task Descriptor	
Gestor de Peticiones				Role Descriptor	
Petición de Modificación Recibida y Registrada				Outcome Descriptor	
Lista de Espera				Artifact Descriptor	
Petición de Modificación Recibida y Registrada				Outcome Descriptor	
Registro de Peticiones				Deliverable Descriptor	

Figura 3.18 WBS I1: Atención de la petición

Description	Work Breakdown Structure	Team Allocation	Work Product Usage	
+ Expand All Se				
- Team Breakdown				
+ Expand All Se				
Breakdown Element	Model Info	Team	Type	Planned
- Gestor de Peticiones			Role Descriptor	
Lista de Espera	Modifies		Artifact Descriptor	
Petición de Modificación Recibida y Registrada	Modifies		Outcome Descriptor	
Registro de Peticiones	Modifies		Deliverable Descriptor	
11.1: Recibir Petición de Modificación	Performs as Owner		Task Descriptor	
11.2: Decidir el Tipo de Mantenimiento	Performs as Owner		Task Descriptor	
- Usuario			Role Descriptor	
Petición de Modificación Recibida y Registrada	Modifies		Outcome Descriptor	
11.1: Recibir Petición de Modificación	Performs as Owner		Task Descriptor	

Figura 3.19 TA I1: Atención de la petición

Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
+ Exp					
- Work Product Breakdown					
+ Exp					
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Lista de Espera	Output				Artifact Descriptor
Petición de Modificación	Mandatory Input				Work Product Slot
Petición de Modificación Recibida y Registrada	Mandatory Input, Output				Outcome Descriptor
Registro de Peticiones	Output				Deliverable Descriptor


Figura 3.20 WPU I1: Atención de la petición

### 3.3.1.3. SNP: SprintM no planificable

Esta actividad se compone de dos subactividades.

#### SNP1: Análisis del error

**Capability Pattern: SNP1: Análisis del Error**

 Corroborar el error y propone alternativas para corregirlo.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Analizar y verificar el error para plantear posibles soluciones.

[Back to top](#)

**Relationships**

**Description**

En esta subactividad el **Equipo de Mantenimiento**, junto al **Usuario**, verifica el error informado en la **Petición de Modificación** y sus causas. Se plantean alternativas de solución y se genera la **Lista de Elementos Software por Corregir**.

**Figura 3.21** DESC SNP1: Análisis del error

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type	Team
<b>SNP1.1: Investigar y Analizar Causas</b>	●●●●	1		Task Descriptor	
Equipo de Mantenimiento				Role Descriptor	
Usuario				Role Descriptor	
Petición de Modificación				Work Product Slot	
Producto Software en Explotación con Error Crítico				Artifact Descriptor	
Lista de Elementos Software por Corregir				Artifact Descriptor	

**Figura 3.22** WBS SNP1: Análisis del error

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<a href="#">Expand All Sections</a> <a href="#">Collapse All Sections</a>							
<b>Team Breakdown</b>							
<a href="#">Expand All Sections</a> <a href="#">Collapse All Sections</a>							
Breakdown Element	Model Info	Team	Type	Planned	Multiple Occurrences		
[-] Equipo de Mantenimiento			Role Descriptor				
Lista de Elementos Software por Corregir	Modifies		Artifact Descriptor				
SNP1.1: Investigar y Analizar Causas	Performs as Owner		Task Descriptor				
[-] Usuario			Role Descriptor				
Lista de Elementos Software por Corregir	Modifies		Artifact Descriptor				
SNP1.1: Investigar y Analizar Causas	Performs as Owner		Task Descriptor				

Figura 3.23 TA SNP1: Análisis del error

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<a href="#">Exp</a>							
<b>Work Product Breakdown</b>							
<a href="#">Exp</a>							
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type		
Lista de Elementos Software por Corregir	Output				Artifact Descriptor		
Petición de Modificación	Mandatory Input				Work Product Slot		
Producto Software en Explotación con Error Crítico	Mandatory Input				Artifact Descriptor		

Figura 3.24 WPU SNP1: Análisis del error

## SNP2: Intervención correctiva urgente

### Capability Pattern: SNP2: Intervención Correctiva Urgente



Soluciona el problema modificando el producto de software y realiza pruebas en éste.

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<a href="#">Expand All Sections</a> <a href="#">Collapse All Sections</a>							
<b>Purpose</b>							
Corregir los errores urgentes y aplicar pruebas unitarias a los elementos modificados.							
<a href="#">Back to top</a>							
<b>Relationships</b>							
<b>Description</b>							
En esta subactividad el <b>Equipo de Mantenimiento</b> ejecuta todas las acciones que sean necesarias para corregir el error lo antes posible, se identifican los elementos del producto de software que fueron afectados por las correcciones y, finalmente, se realizan pruebas unitarias a los elementos que se vieron afectados en este proceso.							

Figura 3.25 DESC I0: Planificación del proceso

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
<input type="checkbox"/> Work Breakdown							
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
Breakdown Element	Steps	Index	Predecessors	Type			
<input type="checkbox"/> SNP2.1: Realizar Acciones Correctivas	••	1		Task Descriptor			
Equipo de Mantenimiento				Role Descriptor			
Lista de Elementos Software por Corregir				Artifact Descriptor			
Lista de Elementos Software Corregidos				Artifact Descriptor			
<input type="checkbox"/> SNP2.2: Ejecutar Pruebas Unitarias	••	2		Task Descriptor			
Equipo de Mantenimiento				Role Descriptor			
Casos de Pruebas				Artifact Descriptor			
Elementos de Software Corregidos				Work Product Slot			
Elementos de Software Corregidos y Aprobados				Outcome Descriptor			
Pruebas Unitarias Realizadas				Artifact Descriptor			

Figura 3.26 WBS IO: Planificación del proceso

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections							
<input type="checkbox"/> Team Breakdown							
<input type="checkbox"/> Expand All Sections							
Breakdown Element	Model Info	Team	Type	Planned			
<input type="checkbox"/> Equipo de Mantenimiento			Role Descriptor				
Elementos de Software Corregidos y Aprobados	Modifies		Outcome Descriptor				
Lista de Elementos Software Corregidos	Modifies		Artifact Descriptor				
Pruebas Unitarias Realizadas	Modifies		Artifact Descriptor				
SNP2.1: Realizar Acciones Correctivas	Performs as Owner		Task Descriptor				
SNP2.2: Ejecutar Pruebas Unitarias	Performs as Owner		Task Descriptor				

Figura 3.27 TA IO: Planificación del proceso



Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Casos de Pruebas	Mandatory Input				Artifact Descriptor
Elementos de Software Corregidos	Mandatory Input				Work Product Slot
Elementos de Software Corregidos y Aprobados	Output				Outcome Descriptor
Lista de Elementos Software Corregidos	Output				Artifact Descriptor
Lista de Elementos Software por Corregir	Mandatory Input				Artifact Descriptor
Pruebas Unitarias Realizadas	Output				Artifact Descriptor


Figura 3.28 WPU IO: Planificación del proceso

### 3.3.1.4. SP: SprintM planificable

Esta actividad se compone de dos subactividades.

#### SP1: Análisis de la petición

**Capability Pattern: SP1: Análisis de la Petición**

 Se proponen alternativas para realizar las modificaciones según el tipo de petición.

Description	Work Breakdown Structure	Team Allocation	Work Product Usage
<a href="#">Expand All Sections</a> <a href="#">Collapse All Sections</a>			
<b>Purpose</b>			
Analizar y proponer alternativas para las modificaciones que deben realizarse.			
<a href="#">Back to top</a>			
<b>Relationships</b>			
<b>Description</b>			
En esta subactividad, según el tipo de mantenimiento que se asigna a la <a href="#">Petición de Modificación</a> , se proponen alternativas para las modificaciones que deben realizarse.			

Figura 3.29 DESC SP1: Análisis de la petición

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
<input type="checkbox"/> Work Breakdown							
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
Breakdown Element	Steps	Index	Predecessors	Type			
<input type="checkbox"/> SP1.1: Analizar y Elegir Solución	●●●	1		Task Descriptor			
Equipo de Mantenimiento				Role Descriptor			
Alternativas de Implementación				Work Product Slot			
Petición de Modificación				Work Product Slot			
Producto de Software en Explotación				Work Product Slot			
Alternativa Seleccionada				Outcome Descriptor			

**Figura 3.30** WBS SP1: Análisis de la petición

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
<input type="checkbox"/> Team Breakdown							
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All Sections							
Breakdown Element	Model Info	Team	Type	Planned	Multiple Occurrences		
<input type="checkbox"/> Equipo de Mantenimiento			Role Descriptor				
Alternativa Seleccionada	Modifies		Outcome Descriptor				
SP1.1: Analizar y Elegir Solución	Performs as Owner		Task Descriptor				

**Figura 3.31** TA SP1: Análisis de la petición

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections							
<input type="checkbox"/> Work Product Breakdown							
<input type="checkbox"/> Expand All Sections							
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type	Planned	
Alternativas de Implementación	Mandatory Input				Work Product Slot		
Alternativa Seleccionada	Output				Outcome Descriptor		
Petición de Modificación	Mandatory Input				Work Product Slot		
Producto de Software en Explotación	Mandatory Input				Work Product Slot		

**Figura 3.32** WPU SP1: Análisis de la petición

## SP2: Intervención y Pruebas

### Capability Pattern: SP2: Intervención y Pruebas



Se modifica y prueba el producto de software.

#### Description

#### Work Breakdown Structure

#### Team Allocation

#### Work Product Usage

[Expand All Sections](#)

[Collapse All Sections](#)

#### [-] Purpose

Modificar el producto de software y probar que los cambios han sido realizados correctamente.

[Back to top](#)

#### [+] Relationships

#### [-] Description

En esta subactividad el **Equipo de Mantenimiento** lleva a cabo las modificaciones necesarias, realiza las pruebas unitarias y de sistema, y se ejecuta el nuevo producto de software junto al antiguo para corroborar que no haya ningún problema.

Figura 3.33 DESC IO: Planificación del proceso

#### Description

#### Work Breakdown Structure

#### Team Allocation

#### Work Product Usage

[Expand All Sections](#)

[Collapse All Sections](#)

#### [-] Work Breakdown

[Expand All Sections](#)

[Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type
<input type="checkbox"/> SP2.1: Ejecutar Intervención	●●	1		Task Descriptor
Equipo de Mantenimiento				Role Descriptor
Copia del Producto de Software (A)				Artifact Descriptor
Diagnostico del Error (CP)				Artifact Descriptor
Mejora a Realizar (P)				Artifact Descriptor
Copia Adaptada (A)				Artifact Descriptor
Producto de Software Intervenido				Work Product Slot
<input type="checkbox"/> SP2.2: Ejecutar Pruebas Unitarias y de Integración (CP/P/A).	●●●	2		Task Descriptor
<input type="checkbox"/> SP2.3: Ejecutar Paralelamente el Producto de Software Antigo y el Nuevo	●●	3	2	Task Descriptor

Figura 3.34 WBS IO: Planificación del proceso

Description	Work Breakdown Structure	Team Allocation	Work Product Usage	
Expand All Se				
Team Breakdown				
Expand All Se				
Breakdown Element	Model Info	Team	Type	Planned
Equipo de Mantenimiento			Role Descriptor	
Usuario			Role Descriptor	
SP2.3: Ejecutar Paralelamente el Producto de Software Antiguo y el Nuevo	Performs as Additional		Task Descriptor	


Figura 3.35 TA I0: Planificación del proceso

Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
Exp					
Work Product Breakdown					
Exp					
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Copia Adaptada (A)	Output				Artifact Descriptor
Copia del Producto de Software (A)	Mandatory Input				Artifact Descriptor
Diagnostico del Error (CP)	Mandatory Input				Artifact Descriptor
Mejora a Realizar (P)	Mandatory Input				Artifact Descriptor
Producto de Software Intervenido	Mandatory Input, Output				Work Product Slot
Producto Intervenido Comprobado	Mandatory Input, Output				Outcome Descriptor
Producto Intervenido Comprobado y en Correcto Funcionamiento	Output				Outcome Descriptor
Pruebas Unitarias Realizadas	Output				Artifact Descriptor

Figura 3.36 WPU I0: Planificación del proceso

### 3.3.1.5. SSM: Seguimiento del SprintM

**Capability Pattern: SSM1: Seguimiento del SprintM**

 Conducir reuniones de control para realizar el seguimiento de las tareas.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Realizar el seguimiento a las tareas asignadas. Encontrar dificultades para la realización de las tareas.

[Back to top](#)

**Relationships**

**Description**

El propósito de esta actividad es *conducir reuniones de control* para hacer un seguimiento del estado de avance y resolver problemas de las intervenciones realizadas mediante el ó los Sprint de Mantenimiento (SprintM). En esta actividad se lleva a cabo *la revisión del trabajo realizado y se solucionan las dificultades* encontradas. En estas actividades son comunes para los dos tipos de SprintM (planificable y no planificable).

Figura 3.37 DESC SSM: Seguimiento del SprintM

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type
<input type="checkbox"/> SSM1.1: Reuniones Habituales	●●●	1		Task Descriptor
Equipo de Mantenimiento				Role Descriptor
Responsable de Mantenimiento				Role Descriptor
Informe Ejecutivo de las Tareas Realizadas				Artifact Descriptor
Problemas Encontrados				Outcome Descriptor
Informe Ejecutivo de las Tareas Realizadas Actualizado				Artifact Descriptor
Problemas Solucionados				Outcome Descriptor
<input checked="" type="checkbox"/> SSM1.2: Seguimientos de los Cambios	●●●●	2	1	Task Descriptor

Figura 3.38 WBS SSM: Seguimiento del SprintM

Description	Work Breakdown Structure	Team Allocation	Work Product Usage	
+ Expand All Se				
Team Breakdown				
+ Expand All Se				
Breakdown Element	Model Info	Team	Type	Planned
+ Equipo de Mantenimiento			Role Descriptor	
- Responsable de Mantenimiento			Role Descriptor	
Informe Ejecutivo de las Tareas Realizadas Actualizado	Modifies		Artifact Descriptor	
Problemas Solucionados	Modifies		Outcome Descriptor	
Validación y Control de los Cambios Realizados	Modifies		Artifact Descriptor	
SSM1.1: Reuniones Habituales	Performs as Owner		Task Descriptor	
SSM1.2: Seguimientos de los Cambios	Performs as Owner		Task Descriptor	


Figura 3.39 TA SSM: Seguimiento del SprintM

Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
+ Exp					
Work Product Breakdown					
+ Exp					
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Informe Ejecutivo de las Tareas Realizadas	Mandatory Input				Artifact Descriptor
Informe Ejecutivo de las Tareas Realizadas Actualizado	Output				Artifact Descriptor
Problemas Encontrados	Mandatory Input				Outcome Descriptor
Problemas Solucionados	Output				Outcome Descriptor
Producto de Software en Mantenimiento	Mandatory Input				Artifact Descriptor
Productos de Software Relacionados	Mandatory Input				Artifact Descriptor
Validación y Control de los Cambios Realizados	Output				Artifact Descriptor

Figura 3.40 WPU SSM: Seguimiento del SprintM

### 3.3.1.6. F1: Finalización de la intervención

**Capability Pattern: F1: Finalización de la Intervención**

 Da termino a la intervención relacionada a una petición de modificación.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Revisar el producto de software modificado con el usuario, actualizar la documentación, dejar operando el producto de software nuevo y definir el próximo SprintM.

[Back to top](#)

**Relationships**

**Description**

En esta actividad el equipo de mantenimiento junto al usuario verifican el correcto funcionamiento del producto de software nuevo en el entorno de producción. Además, se realizan las actualizaciones necesarias al [Manual de Usuario](#). Finalmente, se define cuál será la próxima iteración del ciclo de mantenimiento. Con esto se da por terminado un proceso de intervención relacionado con una [Petición de Modificación](#).

**Figura 3.41** DESC F1: Finalización de la intervención

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type
<input type="checkbox"/> F1.1: Verificar y Validar Corrección con el Cliente	●●●	1		Task Descriptor
Cliente				Role Descriptor
Equipo de Mantenimiento				Role Descriptor
Usuario				Role Descriptor
Elementos de Software Corregidos y Aprobados				Outcome Descriptor
Pruebas Unitarias Realizadas				Artifact Descriptor
Documento de Aceptación de la Corrección Validado por el Cliente				Artifact Descriptor
Producto de Software Totalmente Comprobado				Outcome Descriptor
<input type="checkbox"/> F1.2: Pasar a Producción	●	2	1	Task Descriptor
<input type="checkbox"/> F1.3: Documentar Manual de Usuario	●●	3	2	Task Descriptor
<input type="checkbox"/> F1.4: Registro de la Intervención	●	4	3	Task Descriptor
<input type="checkbox"/> F1.5: Reunión de Retrospección	●●●	5	4	Task Descriptor

**Figura 3.42** WBS F1: Finalización de la intervención

Description	Work Breakdown Structure	Team Allocation	Work Product Usage	
Expand All Se				
Team Breakdown				
Expand All Se				
Breakdown Element	Model Info	Team	Type	Planned
Cliente			Role Descriptor	
Documento de Aceptación de la Corrección Validado por el Cliente	Modifies		Artifact Descriptor	
Producto de Software Totalmente Comprobado	Modifies		Outcome Descriptor	
F1.1: Verificar y Validar Corrección con el Cliente	Performs as Owner		Task Descriptor	
Equipo de Mantenimiento			Role Descriptor	
Responsable de Mantenimiento			Role Descriptor	
Usuario			Role Descriptor	

**Figura 3.43** TA F1: Finalización de la intervención


Description	Work Breakdown Structure	Team Allocation	Work Product Usage		
Exp					
Work Product Breakdown					
Exp					
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type
Anuncio del Próximo SprintM	Output				Artifact Descriptor
Documentación Generada en esta Etapa	Mandatory Input				Artifact Descriptor
Documento de Aceptación de la Corrección Validado por el Cliente	Output				Artifact Descriptor
Elementos de Software Corregidos y Aprobados	Mandatory Input				Outcome Descriptor
Información Registrada de la Intervención	Output				Artifact Descriptor
Lecciones Aprendidas del SprintM	Mandatory Input				Artifact Descriptor
Manual de Usuario	Mandatory Input				Artifact Descriptor
Manual de Usuario Actualizado	Output				Artifact Descriptor
Producto de Software Totalmente Comprobado	Mandatory Input, Output				Outcome Descriptor

**Figura 3.44** WPU F1: Finalización de la intervención



### 3.3.1.7. F2: Retiro

**Capability Pattern: F2: Retiro**

 Planifica el retiro del producto de software antiguo.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Planear la retirada del producto de software antiguo y ejecutar el retiro de éste.

[Back to top](#)

**Relationships**

**Description**

En esta actividad el **Equipo de Mantenimiento** genera la planificación para el retiro del **Producto de Software Antiguo en Explotación**. Esto se comunica formalmente a los **Usuarios** para que no hayan problemas durante este proceso. Se retira la aplicación antigua y se respaldan los datos que ésta pueda contener.

Figura 3.45 DESC F2: Retiro

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type
<input type="checkbox"/> F2.1: Desarrollar Plan de Retirada	●●●	1		Task Descriptor
Equipo de Mantenimiento				Role Descriptor
Producto de Software Antiguo en Explotación				Work Product Slot
Producto de Software Nuevo				Work Product Slot
Plan de Retiro				Artifact Descriptor
<input type="checkbox"/> F2.2: Notificar Futuro Retiro	●	2	1	Task Descriptor
<input type="checkbox"/> F2.3: Ejecutar en Paralelo	●●●	3	2	Task Descriptor
<input type="checkbox"/> F2.4: Notificar Retiro	●	4	3	Task Descriptor
<input type="checkbox"/> F2.5: Almacenar Datos del Producto de Software Antiguo	●	5	4	Task Descriptor

Figura 3.46 WBS F2: Retiro

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All							
<input type="checkbox"/> Team Breakdown							
<input type="checkbox"/> Expand All Sections <input type="checkbox"/> Collapse All							
Breakdown Element	Model Info	Team	Type	Planned	Multiple Occurrences		
<input type="checkbox"/> Cliente			Role Descriptor				
Notificación a los Usuarios del Retiro	Modifies		Artifact Descriptor				
F2.2: Notificar Futuro Retiro	Performs as Owner		Task Descriptor				
F2.3: Ejecutar en Paralelo	Performs as Additional		Task Descriptor				
<input type="checkbox"/> Equipo de Mantenimiento			Role Descriptor				
<input type="checkbox"/> Usuario			Role Descriptor				


Figura 3.47 TA F2: Retiro

Description		Work Breakdown Structure		Team Allocation		Work Product Usage	
<input type="checkbox"/> Exp							
<input type="checkbox"/> Work Product Breakdown							
<input type="checkbox"/> Exp							
Breakdown Element	Model Info	Entry State	Exit State	Deliverable	Type		
Coexistencia del Producto de Software Antigo con el Nuevo	Output				Outcome Descriptor		
Datos del Entorno Antigo	Mandatory Input				Work Product Slot		
Datos del Entorno Antigo Guardados	Output				Outcome Descriptor		
Notificación a los Usuarios del Retiro	Output				Artifact Descriptor		
Plan de Retiro	Mandatory Input, Output				Artifact Descriptor		
Producto de Software Antigo en Explotación	Mandatory Input				Work Product Slot		
Producto de Software Antigo Retirado	Output				Outcome Descriptor		
Producto de Software Nuevo	Mandatory Input				Work Product Slot		
Producto de Software Nuevo en Explotación	Output				Outcome Descriptor		

Figura 3.48 WPU F2: Retiro

### 3.3.1.8. F3: Finalización del servicio

**Capability Pattern: F3: Finalización del Servicio**

 Terminar formalmente con la prestación del servicio de mantenimiento.

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Purpose**

Dar por terminada la prestación de servicios de mantenimiento con el Cliente.

[Back to top](#)

**Relationships**

**Description**

El propósito de esta actividad es dar por finalizado de manera formal el servicio de mantenimiento de software al **Cliente**. Esta actividad se realiza cuando el mantenedor deja de prestar sus servicios a la organización **Cliente**.

**Figura 3.49** DESC F3: Finalización del servicio

**Description** | **Work Breakdown Structure** | **Team Allocation** | **Work Product Usage**

[Expand All Sections](#) | [Collapse All Sections](#)

**Work Breakdown**

[Expand All Sections](#) | [Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Type
<input type="checkbox"/> F3.1: Entrega del Inventario y de la Documentación		1		Task Descriptor
Cliente				Role Descriptor
Equipo de Mantenimiento				Role Descriptor
Documentos y Productos Generados Durante el Proceso de Mantenimiento				Deliverable Descriptor
<input type="checkbox"/> F3.2: Termino definitivo del Servicio		2	1	Task Descriptor
Equipo de Mantenimiento				Role Descriptor
Usuario				Role Descriptor
Documento de Formalización Termino de Prestación del Servicio				Artifact Descriptor

**Figura 3.50** WBS F3: Finalización del servicio

Description				Work Breakdown Structure				Team Allocation				Work Product Usage			
+ Exp															
Team Breakdown															
+ Exp															
Breakdown Element								Model Info		Team		Type			
+ Cliente												Role Descriptor			
- Equipo de Mantenimiento												Role Descriptor			
Documento de Formalización Termino de Prestación del Servicio								Modifies				Artifact Descriptor			
Documentos y Productos Generados Durante el Proceso de Mantenimiento								Modifies				Deliverable Descriptor			
F3.1: Entrega del Inventario y de la Documentación								Performs as Owner				Task Descriptor			
F3.2: Termino definitivo del Servicio								Performs as Owner				Task Descriptor			
+ Usuario												Role Descriptor			

**Figura 3.51** TA F3: Finalización del servicio

Description				Work Breakdown Structure				Team Allocation				Work Product Usage					
+ Exp																	
Work Product Breakdown																	
+ Exp																	
Breakdown Element								Model Info		Entry State		Exit State		Deliverable		Type	
Documento de Formalización Termino de Prestación del Servicio								Output								Artifact Descriptor	
Documentos y Productos Generados Durante el Proceso de Mantenimiento								Output								Deliverable Descriptor	

**Figura 3.52** WPU F3: Finalización del servicio

## **4. Aplicación de la mejora de procesos de software**

---

A lo largo de este apartado se presentará la empresa donde será aplicado el ciclo de mejoras, además del análisis estratégico que se le realizó. Luego, se detallará las actividades que se han llevado a cabo asociadas a la aplicación del ciclo de mejora de procesos de software.

### **4.1. Empresa: TyC Software**

A continuación se describirá la empresa, TyC Software, donde se llevará a cabo el proceso de mejora. Además, se expondrá parte del análisis estratégico que fue ejecutado a la organización para identificar y conocer en detalle, desde el punto de vista ejecutivo, el negocio de la empresa, los cargos de la empresa, los sistemas de información que utilizan, y sus problemas y necesidades. En vista de obtener las bases para poder planificar el proceso de mejora de procesos.

#### **4.1.1. Informaciones generales**

TyC Software nace en Julio del año 2004 instalando su casa matriz en Viña del Mar, con el propósito de dar soluciones eficientes a sus clientes, a través de asesorías, gestión, representación y desarrollo de proyectos, utilizando tecnologías que explotan al máximo la conectividad global, comunicando a las empresas con sus procesos, clientes, proveedores, accionistas y empleados.

En Mayo de 2006, entra en operación la sucursal Santiago de la empresa, abarcando un importante mercado, donde sus clientes reciben una rápida y eficiente atención por parte de los profesionales a cargo de los respectivos proyectos.

Actualmente, TyC cuenta con una moderna infraestructura y un equipo de profesionales conformado por 27 personas, entre las cuales se encuentran Ingenieros Civiles Informáticos, Ingenieros en Ejecución Informática, Técnicos en Computación y personal administrativo.

Cabe destacar que el principal producto que ofrece TyC Software es el Sistema de Control de Bodega Inteligente – SCBi. Esta aplicación permite apoyar la operación y realizar el control de la logística y distribución de los productos terminados y materias primas de la empresa en donde es implementado. Una parte importante de los clientes que posee TyC Software son empresas que utilizan el sistema SCBi, a las que se les brinda un importante número de horas de mantenimiento.

TyC Software cuenta con una importante y creciente lista de proyectos en carpeta que constituyen un importante desafío para el futuro próximo. Esta gran cantidad de proyectos conducirá al crecimiento de empresa, incorporando nuevos recursos humanos y modificando la manera en que se realizan las actividades en la actualidad. Por ello y ante la posibilidad que se le fue presentada para la incorporación de buenas prácticas y la formalización de procesos de software, es que TyC Software accedió a ser parte de este proyecto, transformándose en el caso de estudio, donde se aplicará el marco metodológico de Competisoft.

### **4.1.2. Análisis estratégico de TyC Software**

Como paso previo al inicio del ciclo de mejoras, se ha llevado a cabo un análisis estratégico de TyC Software, en busca de alinear los objetivos estratégicos de la empresa con los que se deberán plantear para el proceso de mejora. Además, este análisis permitirá conocer en detalle a la empresa, pudiendo identificar su negocio, problemas y necesidades

El detalle del negocio de la empresa se obtuvo tras la aplicación de una entrevista, apoyada en un cuestionario, con la plana directiva de la empresa. A continuación se presentan los resultados obtenidos:

#### Industria:

TyC Software se posiciona en la industria de la ingeniería de software, fundamentalmente apunta al desarrollo de productos de software, así también ofrece servicios de mantenimiento (soporte), consultoría y capacitaciones.

#### Mercado:

El mercado en el que se enfoca TyC Software es el área de la producción, logística y distribución, sin dejar de lado otras áreas, como la comercial, pero no es su foco central.

#### Productos/Servicios:

Se ofrecen diversos productos orientados al mercado de la producción, logística y distribución. Entre estos productos se encuentran: Sistema de Control de Bodega Inteligente – SCBi, Sistema de Packing – SISTPACK y Sistema de Producción – SP. Cabe destacar que estos productos son completamente nacionales, desarrollados por ingenieros chilenos, por lo tanto tienen incorporado el contexto país en sus funciones. Además, se ofrecen servicios en de consultoría en tecnologías de la información, también enfocada hacia el área de la producción, logística y distribución, además de consultorías sobre administración.

#### Posición relativa en el mercado:

La posición de TyC Software ocupa una pequeña porción del mercado, pero con clientes de mediano y gran tamaño.

#### Clientes:

Los principales clientes de la empresa son: LAN, Sopraval, Edelpa, Automotores Gildemeister, Agunsa, Propal, Rhona, Fasa, Imahe, Caimi, Imperial, Metlife, entre otros.

#### Competencia Relevante:

Dependiendo del área de negocio que se considere, la empresa tiene claramente identificada su competencia. Dentro del área de desarrollo existe una larga lista de competidores dentro del mercado. En el área de los productos de logística compiten principalmente con productos importados por empresas como Binaria.

Empleados:

La empresa cuenta con un equipo de 27 personas, 7 personas al área administrativas/contable y 20 dedicadas al área de las tecnologías de la información.

**4.1.2.1. Áreas de negocio**

A continuación se presenta el organigrama (Figura 4.1) de TyC Software que muestra las gerencias y áreas de negocio la empresa.

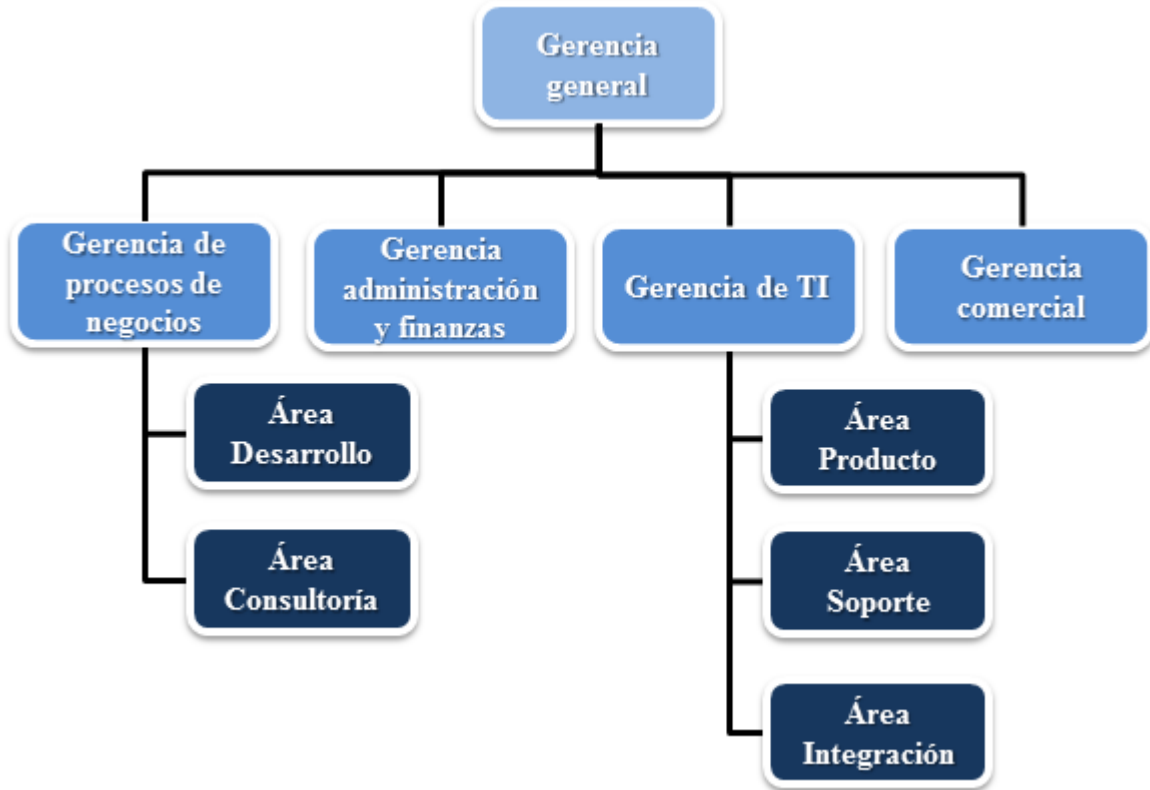


Figura 4.1 Organigrama y áreas de TyC Software

**4.1.2.2. Nivel de soporte funcional de sistemas**

Como toda empresa asociada al rubro de las tecnologías de la información, TyC Software debe manejar diversos sistemas para llevar a cabo las aplicaciones que desarrolla. Así, los sistemas que utilizan la empresa y su respectivo uso son:

**Tabla 4.1** Sistemas y funcionalidades utilizados

Sistema	Funcionalidad
Servicios Gmail	Agendas de jefes de proyecto
Software propio para administración de proyectos	Administración de proyectos, cartas Gantt, riesgos, curvas S, etc.
Transtecnia	Sistema financiero contable
Suite Visual Studio	Desarrollo de software
Sistema manual de planillas de soporte	Controlar y seguir los recursos asignados al área de soporte
Oracle, SQL Server, entre otras BD	Desarrollo de software

#### 4.1.2.3. Problemas y necesidades

Dentro del ciclo de entrevistas realizado se han podido identificar aquellas alteraciones al orden natural en que se desarrollan las actividades y que limitan o impiden la consecución de un objetivo de la empresa, es decir, se han identificado los problemas que ésta tiene. Los problemas identificados son:

- No poder administrar de manera adecuada los recursos asignadas al área de soporte.
- No poder manejar de mejor manera la administración de proyectos.
- Problemas propios del desarrollo de software.

Así también, se han descubierto y formalizado todas aquellas deficiencias que inciden directamente en el desarrollo de una actividad, retrasándola, agregándole trabajo operativo o impidiéndola, es decir, las necesidades de la empresa. Estas son:

- Se necesita analizar el proceso de soporte y mantención para poder administrar esta área de mejor manera, especialmente que permita el seguimiento de los recursos asignados.
- Afrontar de manera adecuada la creciente cantidad de clientes a los que se les brinda el servicio de soporte y mantenimiento.
- Se necesita invertir en una herramienta o hacer un desarrollo interno para mejorar el proceso de administración de proyectos.

#### 4.1.2.4. Desarrollo de la organización

Se consultó a la empresa por el estado actual y el que se espera en el futuro de diversos puntos importantes para el negocio. Los resultados obtenidos son:

- El producto principal (SCBi) de la empresa es utilizado por grandes clientes y pretende seguir en crecimiento, además se buscará agregar más funcionalidades y utilizar la tecnología de punta para hacer crecer esta aplicación, además de ofrecer múltiples opciones de personalización e integración.
- Abarcar el mercado transversalmente de la producción, logística y distribución a nivel nacional en un plazo de tres años, para luego exportar el producto.
- Geográficamente la empresa tiene clientes que operan en Santiago y Valparaíso, próximamente se agregarán clientes en el Biobío, Puerto Montt y Calama. Además, está en mira el mercado de Colombia y Ecuador.
- El día de hoy TyC Software busca seguir especializándose en el rubro de la producción, logística y distribución, más que apuntar a la especialización en la tecnología.



- En la actualidad la empresa utiliza en sus productos tecnologías móviles, GPS, etc. con tal de brindar las mejores funcionalidades a sus clientes. En el futuro buscarán las tecnologías ad-hoc a sus productos para que puedan brindar más y mejores funcionalidades.

## 4.2. Instalación del ciclo

Tal como se estableció en el Marco Teórico el primer paso que se debe llevar a cabo para comenzar con el proceso de mejoras es la instalación del ciclo. Por lo tanto se han seguido paso a paso las actividades definidas para esta etapa.

### Empezar la instalación:

En esta actividad el Grupo Directivo nombra al alumno tesista como Responsable de Mejora de Procesos (RMP), para que sea el impulsor principal del ciclo de mejora dado el conocimiento que posee sobre éste.

### Identificar las necesidades del negocio y los requisitos que conducen la mejora teniendo en cuenta el Plan Estratégico de la organización:

Gracias al análisis estratégico realizado previamente a TyC Software, se pueden plantear objetivos claros y alineados con los objetivos estratégicos para el proceso de mejora, buscando satisfacer las necesidades de mejora y atacar los problemas identificados. De esta forma, se han definido como necesidades:

- Adecuar los procesos de las áreas de la empresa para el creciente número de clientes y proyectos en carpeta.
- Mejorar la administración del área de soporte y mantenimiento y realizar su integración con alguna herramienta.
- Mejorar la administración de proyectos y definir si la aplicación de una herramienta es necesaria.

Igualmente, se ha propuesto como objetivo general de las mejoras:

Realizar mejoras en los procesos de las áreas con mayores problemas (soporte y administración de proyectos) para afrontar de mejor manera el creciente número de clientes.

Finalmente, el Grupo Directivo ha designado el personal que desempeñará los roles requeridos por el modelo. Los roles han sido conformados de la siguiente manera:

*Grupo de Mejora de Procesos (GMP):* para este rol han sido designados los jefes y gerentes de proyectos, dado su conocimiento sobre las capacidades con que cuenta cada uno de los integrantes de su equipo de trabajo.

*Grupo de Gestión de Mejora (GGM):* dentro de este grupo se encuentra un integrante del Grupo Directivo que posee experiencia en el levantamiento de procesos, un gerente de proyectos que está a cargo del proyecto piloto donde se realizará la implementación de las mejoras de procesos y el alumno tesista en su rol de Responsable de Mejora de Procesos.

*Responsable de Proceso (RP)*: para este rol se ha designado a la jefa del proyecto en que será realizará la implementación de las mejoras de procesos.

*Evaluador (EV)*: no se contará con un evaluador, dado que no se cuentan con los recursos económicos para financiar una evaluación certificada.

#### Construir una Propuesta de Mejora del proceso de software:

La propuesta de mejora que se construyó para el primer ciclo de mejoras que se realizaría en la empresa y, además, el primero realizado por el RMP (alumno tesista) considera los siguientes aspectos:

- Dar prioridad al proceso de Mantenimiento de Software (llevar a cabo su mejora en la primera iteración), dado que el proceso Agil\_Mantema ya ha sido modelado con la herramienta EPF Composer, lo que permitirá entregar resultados con mayor rapidez para TyC Software.
- Realizar el modelado del proceso de mantenimiento de software, que se utiliza actualmente en la organización, con la herramienta EPF Composer para ser comparado con el proceso Agil\_Mantema.
- Para la primera iteración se estiman aproximadamente dos meses de ejecución.
- Considerar en iteraciones posteriores el proceso de Desarrollo de Software y Administración de Proyectos, cuando se tenga mayor experiencia y madurez en la aplicación del modelo.

#### Lanzamiento del ciclo:

Para finalizar con esta actividad se ha sociabilizado con el personal de la empresa el inicio del ciclo de mejora de procesos, presentando los pasos que se llevarán a cabo y lo que se busca obtener en la primera iteración del ciclo. Además, se explicó cómo las mejoras podrían afectar el trabajo que ellos realizan en la actualidad.

### **4.3. Diagnóstico de procesos**

Para llevar a cabo esta actividad se utilizará el Cuestionarios de Mantenimiento Agil\_Mantema [Gajardo *et al.*, 2008] (Anexo) para poder obtener información sobre el conocimiento que tienen los encargados de mantenimiento sobre este proceso y la cercanía que existe entre lo que se realiza actualmente en TyC Software con lo que se propone en Agil\_Mantema.

El cuestionario fue entregado a 6 personas y los resultados obtenidos, agrupados por actividades del proceso de mantenimiento, se presentan desde la tabla 4.2 hasta la tabla 4.12 (donde D significa digital y M significa manual):

Tabla 4.2 (A.1.) Planificación del proceso

	Sí	No	D ó M	Respuesta
<b>A.1 PLANIFICACIÓN DEL PROCESO</b>				
<b>A.1.1 Asignar responsables</b>				
	A.1.1.1	6	0	
	A.1.1.2			Jefe soporte, encargado soporte
	A.1.1.3	6	0	
	A.1.1.4			Jefe soporte, encargado soporte
	A.1.1.5	0	6	
	A.1.1.6	6	0	
	A.1.1.7	6	0	
	A.1.1.8		D	
<b>A.1.2 Adquirir conocimiento de la aplicación</b>				
	A.1.2.1			Documentación, código fuente, referencias, entrevistas, observación
	A.1.2.2	3	3	
	A.1.2.3			Jefe soporte, encargado soporte
	A.1.2.4	1	5	
	A.1.2.5			Formulario asignación de incidentes
	A.1.2.6	1	5	
	A.1.2.7	0	6	
	A.1.2.8		D	
<b>A.1.3 Preparar entornos de pruebas</b>				
	A.1.3.1	6	0	
	A.1.3.2	6	0	
	A.1.3.3			Copias de software, preparación BD y archivos
	A.1.3.4	6	0	
	A.1.3.5			Depende del ambiente de control en el soporte
<b>A.1.4 Definir procedimientos de petición de modificación</b>				
	A.1.4.1	6	0	
	A.1.4.2			Jefe soporte, encargado soporte
	A.1.4.3	6	0	
	A.1.4.4	6	0	
	A.1.4.5	6	0	
	A.1.4.6		D	
<b>A.1.5 Verificar la planificación del proceso</b>				
	A.1.5.1	3	3	
	A.1.5.2			Jefe soporte, encargado soporte
	A.1.5.3	1	5	
	A.1.5.4	1	5	
	A.1.5.5			D
	A.1.5.6	3	3	
	A.1.5.7			Desarrollador
	A.1.5.8	2	4	
	A.1.5.9	2	4	
	A.1.5.10			D

	A.1.5.11	6	0	
	A.1.5.12			Jefe soporte, encargado soporte
	A.1.5.13	6	0	
	A.1.5.14	6	0	
	A.1.5.15			D
<b>A.1.6</b>	<b>Validar petición de modificación</b>			
	A.1.6.1	6	0	
	A.1.6.2			Jefe soporte, encargado soporte
	A.1.6.3	6	0	
	A.1.6.4	6	0	
	A.1.6.5			D

Tabla 4.3 (A.2.) Atención petición de modificación

	Sí	No	D ó M	Respuesta
<b>A.2 ATENCIÓN PETICIÓN DE MODIFICACIÓN</b>				
<b>A.2.1 Recibir petición de modificación</b>				
	A.2.1.1	6	0	
	A.2.1.2			Jefe soporte, encargado soporte
	A.2.1.3	6	0	
	A.2.1.4	6	0	
	A.2.1.5	6	0	
	A.2.1.6			D
<b>A.2.2 Decidir tipo de mantenimiento</b>				
	A.2.2.1	6	0	
	A.2.2.2			Documento, e-mail, oral
	A.2.2.3	6	0	
	A.2.2.4	6	0	
	A.2.2.5			D
	A.2.2.6			En base al análisis de la petición, en reunión con el jefe de proyecto
	A.2.2.7	6	0	
<b>A.2.3 Verificar la administración de la petición de modificación</b>				
	A.2.3.1	5	1	
	A.2.3.2			Encargado soporte
	A.2.3.3	4	2	
	A.2.3.4	4	2	
	A.2.3.5			D

Tabla 4.4 (A.3.) Tipo de mantenimiento no planificable

	Sí	No	D ó M	Respuesta
<b>A.3 TIPO DE MANTENIMIENTO NO PLANIFICABLE</b>				
<b>A.3.1 Análisis del error</b>				
	A.3.1.1	6	0	
	A.3.1.2	6	0	
	A.3.1.3	5	1	
	A.3.1.4	4	2	
	A.3.1.5		D	

Tabla 4.5 (A.4.) Intervención correctiva urgente

	Sí	No	D ó M	Respuesta
<b>A.4 INTERVENCIÓN CORRECTIVA URGENTE</b>				
<b>A.4.1 Realizar acciones correctivas</b>				
	A.4.1.1	6	0	
	A.4.1.2	6	0	
	A.4.1.3			Desarrollador
	A.4.1.4	6	0	
	A.4.1.5	6	0	
	A.4.1.6	6	0	
	A.4.1.7		D	
<b>A.4.2 Ejecutar pruebas unitarias</b>				
	A.4.2.1	6	0	
	A.4.2.2			Jefe soporte, encargado soporte
	A.4.2.3	6	0	
	A.4.2.4	6	0	
	A.4.2.5	6	0	
	A.4.2.6	5	1	
	A.4.2.7	4	2	
	A.4.2.8		D	
<b>A.4.3 Verificar la nueva configuración de software</b>				
	A.4.3.1	6	0	
	A.4.3.2			Jefe soporte, encargado soporte
	A.4.3.3	3	3	
	A.4.3.4	3	3	
	A.4.3.5		D	
<b>A.4.4 Validar la nueva configuración de software</b>				
	A.4.4.1	6	0	
	A.4.4.2			Jefe soporte, encargado soporte
	A.4.4.3	3	3	
	A.4.4.4	1	5	
	A.4.4.5		D	

Tabla 4.6 (A.5.) Tipo de mantenimiento planificable

	Sí	No	D ó M	Respuesta
<b>A.5 TIPO DE MANTENIMIENTO PLANIFICABLE</b>				
<b>A.5.1 Selección de peticiones</b>				
	A.5.1.1	6	0	
	A.5.1.2			Cliente, tipo de problema, modulos afectados, fecha de recepción del incidente
	A.5.1.3	6	0	
	A.5.1.4			Jefe soporte, encargado soporte
<b>A.5.2 Analizar peticiones y elegir solución</b>				
	A.5.2.1	6	0	
	A.5.2.2			Correctivo no urgente, perfectivo, preventivo, adaptativo
	A.5.2.3	0	6	
	A.5.2.4	0	6	
	A.5.2.5		D	
	A.5.2.6	0	6	
	A.5.2.7	0	6	

Tabla 4.7 (A.6.) Intervención y pruebas

	Sí	No	D ó M	Respuesta
<b>A.6 INTERVENCIÓN Y PRUEBAS</b>				
<b>A.6.1 Ejecutar intervención (Correctivo no urgente, perfectivo, preventivo adaptativo)</b>				
	A.6.1.1	3	3	
	A.6.1.2	6	0	
	A.6.1.3	5	1	
	A.6.1.4	4	2	
	A.6.1.5		D	
<b>A.6.2 Ejecutar pruebas unitaras y de integración (Correctivo no urgente, perfectivo, preventivo adaptativo)</b>				
	A.6.2.1	6	0	
	A.6.2.2	6	0	
	A.6.2.3	5	1	
	A.6.2.4	4	2	
	A.6.2.5		D	
	A.6.2.6	6	0	
	A.6.2.7	6	0	
	A.6.2.8	6	0	
	A.6.2.9			Cliente, jefe soporte, encargado soporte
	A.6.2.10	5	1	
	A.6.2.11	3	3	
	A.6.2.12		D	
	A.6.2.13	5	1	
	A.6.2.14	3	3	

	A.6.2.15			D
<b>A.6.3 Ejecutar paralelamente el software antiguo y el nuevo</b>				
	A.6.3.1	6	0	
	A.6.3.2	6	0	
	A.6.3.3			Desarrollador, jefe soporte, encargado soporte
	A.6.3.4	1	5	
	A.6.3.5	1	5	
	A.6.3.6	1	5	
	A.6.3.7			D
<b>A.6.4 Verificar nueva configuración de software</b>				
	A.6.4.1	6	0	
	A.6.4.2			Cliente, jefe soporte, encargado soporte
	A.6.4.3	4	2	
	A.6.4.4	4	2	
	A.6.4.5			D
<b>A.6.5 Validar nueva configuración de software</b>				
	A.6.5.1	6	0	
	A.6.5.2			Cliente
	A.6.5.3	4	2	
	A.6.5.4	4	2	
	A.6.5.5			D
<b>A.6.6 Verificar la administración de la petición de modificación</b>				
	A.6.6.1	6	0	
	A.6.6.2			Jefe soporte, encargado soporte
	A.6.6.3	4	2	
	A.6.6.4	3	3	
	A.6.6.5			D

Tabla 4.8 (A.7.) Seguimiento SprintM

	Sí	No	D ó M	Respuesta
<b>A.7 SEGUIMIENTO SPRINTM</b>				
<b>A.7.1 Reuniones habituales</b>				
	A.7.1.1	1	5	
	A.7.1.2			El avance en la solución, se evalúan nuevos problemas detectados y se analizan
	A.7.1.3	1	5	
	A.7.1.4	1	5	
	A.7.1.5			D
	A.7.1.6	2	4	
	A.7.1.7			Jefe soporte, encargado soporte
<b>A.7.2 Seguimiento de cambios</b>				
	A.7.2.1	6	0	
	A.7.2.2	3	3	
	A.7.2.3	6	0	
	A.7.2.4	6	0	

	A.7.2.5	1	5	
	A.7.2.6	6	0	
	A.7.2.7	6	0	
	A.7.2.8	4	2	
	A.7.2.9	6	0	
	A.7.2.10	6	0	
	A.7.2.11	6	0	
	A.7.2.12			Jefe soporte, encargado soporte
<b>A.7.3</b>	<b>Verificar seguimiento del SprintM</b>			
	A.7.3.1	0	6	
	A.7.3.2			
	A.7.3.3	0	6	
	A.7.3.4	0	6	
	A.7.3.5			D

Tabla 4.9 (A.8.) Finalización de la intervención

	Sí	No	D ó M	Respuesta
<b>A.8 FINALIZACIÓN DE LA INTERVENCIÓN</b>				
<b>A.8.1</b>	<b>Verificar y validar corrección con el cliente</b>			
	A.8.1.1	5	1	
	A.8.1.2	2	4	
	A.8.1.3	2	4	
	A.8.1.4	2	4	
	A.8.1.5	2	4	
	A.8.1.6			D
<b>A.8.2</b>	<b>Documentar manual de usuario</b>			
	A.8.2.1	2	4	
	A.8.2.2	0	6	
	A.8.2.3	1	5	
	A.8.2.4			D
	A.8.2.5	2	4	
	A.8.2.6			Desarrollador
<b>A.8.3</b>	<b>Registro de la intervención</b>			
	A.8.3.1	6	0	
	A.8.3.2	6	0	
	A.8.3.3			D
<b>A.8.4</b>	<b>Reunión de retrospectión</b>			
	A.8.4.1	0	6	
	A.8.4.2	0	6	
	A.8.4.3	0	6	
	A.8.4.4			D
<b>A.8.5</b>	<b>Verficar la finalización de la intervención</b>			
	A.8.5.1	3	3	
	A.8.5.2			Jefe soporte, encargado soporte



A.8.5.3	3	3	
A.8.5.4	3	3	
A.8.5.5			D

Tabla 4.10 (A.9.) Pasar a producción

	Sí	No	D ó M	Respuesta
<b>A.9 PASAR A PRODUCCIÓN</b>				
<b>A.9.1 Pasar a producción</b>				
A.9.1.1	0	6		
A.9.1.2	1	5		

Tabla 4.11 (A.10.) Retirada

	Sí	No	D ó M	Respuesta
<b>A.10 RETIRADA</b>				
<b>A.10.1 Desarrollar plan de retirada</b>				
A.10.1.1	1	5		
A.10.1.2	1	5		
A.10.1.3			D	
<b>A.10.2 Notificar futura retirada</b>				
A.10.2.1	6	0		
A.10.2.2				Documento, e-mail, oral
A.10.2.3	6	0		
A.10.2.4				Jefe de proyecto
<b>A.10.3 Ejecutar en paralelo</b>				
A.10.3.1	6	0		
A.10.3.2	6	0		
<b>A.10.4 Notificar la retirada</b>				
A.10.4.1	6	0		
A.10.4.2				Documento, e-mail
<b>A.10.5 Almacenar datos del software antiguo</b>				
A.10.5.1	6	0		
A.10.5.2				Repositorio de respaldos
A.10.5.3	5	1		
A.10.5.4				Desarrollador
<b>A.10.6 Verificar plan de retirada del software</b>				
A.10.6.1	2	4		
A.10.6.2				Jefe de proyecto
A.10.6.3	0	6		
A.10.6.4	0	6		
A.10.6.5			D	

**Tabla 4.12** (A.11.) Finalización del servicio

	Sí	No	D ó M	Respuesta
<b>A.11 FINALIZACIÓN DEL SERVICIO</b>				
<b>A.11.1 Entrega del inventario y de la documentación</b>				
	A.11.1.1	6	0	
	A.11.1.2			Producto corregido, documentación, base de conocimiento, fuentes
	A.11.1.3	4	2	
	A.11.1.4	6	0	
	A.11.1.5			Jefe de proyecto, jefe de mantención y soporte
<b>A.11.2 Cese definitivo del servicio</b>				
	A.11.2.1	5	1	
	A.11.2.2	5	1	
	A.11.2.3			D
<b>A.11.3 Verificar finalización del servicio</b>				
	A.11.3.1	6	0	
	A.11.3.2			Jefe soporte, encargado soporte
	A.11.3.3	3	3	
	A.11.3.4	3	3	
	A.11.3.5			D

El resultado obtenido de la aplicación de la encuesta, siguiendo la ISO/IEC 15504 es:

**Tabla 4.13** Resultados diagnóstico inicial

Color	Nivel	Resultado	Nivel de cumplimiento
	Realizado	74%	Ampliamente cumplido
	Gestionado	52%	Ampliamente cumplido
	Establecido	54%	Ampliamente cumplido

### 4.3.1. Entrevistas realizadas

Se han realizado una serie de entrevistas al personal relacionado con el proceso de mantenimiento de software en busca de información que permita conocer la forma en que se realiza actualmente esta actividad. A continuación se detallan las entrevistas realizadas:

**Tabla 4.14** Entrevistas realizadas

Entrevistado	Fecha	Duración de la entrevista
Gerente TyC Software	06/04/2011	1 hora aprox.
Gerente área soporte	14/04/2011	40 minutos aprox.
Encargado soporte	18/04/2011	1 hora y 30 minutos aprox.

De estas entrevistas, principalmente la llevada a cabo con el Gerente de TyC Software, se ha obtenido la información pertinente para el análisis estratégico realizado a la empresa. Por

otra parte, de las entrevistas realizadas al Gerente de área de soporte y al principal encargado de soporte se ha podido visualizar la forma en que opera esta área.

Cabe destacar que la persona Encargada de soporte tiene amplia experiencia dentro de esta área, dado que previamente había trabajado en el área de soporte y mantenimiento de una empresa multinacional, por lo tanto tiene el conocimiento necesario sobre los procesos que se deben llevar a cabo para realizar. Relacionado a esto, es importante mencionar que esta persona ha definido el protocolo de la empresa en el tema de la mantención de sistemas, además ha definido las actividades y planillas necesarias para este proceso según su experiencia. Dada la experiencia y el proceso que conocía de la empresa multinacional, el modus operandi del proceso definido es demasiado extenso para TyC Software, por lo que ha sido acotado en actividades y roles que son asignados.

### **4.3.2. Proceso de mantenimiento actual**

Del diagnóstico realizado se ha obtenido información sobre el proceso de mantenimiento actual de TyC Software, el que es aplicado de manera recortada dado que está basado en el proceso de mantenimiento de una empresa multinacional. A continuación se describen los principales elementos del proceso actual.

#### **4.3.2.1. Tipos de mantenimiento**

Se definen tres tipos de mantenimiento según la prioridad que el usuario le asigna a la petición de mantenimiento. Esta prioridad establece la importancia del incidente respecto de otros incidentes, dada su naturaleza y la importancia estratégica desde el punto de vista del negocio del cliente.

- *Alta:* afecta no permitiendo el desarrollo del negocio, ahora.
- *Media:* afecta el desarrollo parcial del negocio, hoy.
- *Baja:* afecta el desarrollo del negocio, pero es manejable por el usuario, mañana.

#### **4.3.2.2. Roles**

En el proceso de mantenimiento definido por TyC Software se distinguen cinco roles, uno asociado a la empresa Cliente y cuatro asociados al área de soporte.

- *Usuario:* persona de la empresa a la que se le presta el servicio de mantenimiento, que canaliza las peticiones de mantención hacia el área de soporte. El usuario hace llegar las peticiones de mantenimiento a través del formulario de incidentes.
- *Coordinador de incidentes:* encargado del área de soporte de TyC Software que actúa de nexo con la empresa cliente. Esta persona recibe los tickets de soporte y se contacta con el Usuario que ha reportado el problema. Además, asigna a los especialistas de negocio y lenguaje y al desarrollador a cargo de resolver el problema. Por otro lado, puede llegar a realizar actividades de análisis del problema detectado y, al finalizar un ticket de soporte, actúa como validador (por parte de TyC Software) de la solución dada por el desarrollador.
- *Especialista en el negocio:* persona encargada de apoyar las consultas del desarrollador respecto del problema desde el punto de vista del negocio de la empresa cliente a la que

se le brinda el servicio de mantenimiento. Si el sistema que está bajo mantención es un sistema desarrollado por TyC Software, este especialista es una persona que ha estado en el proyecto original de desarrollo y en contacto con la empresa cliente. Si es un sistema desarrollado por terceros, este especialista ha recibido capacitación sobre el negocio de la empresa cliente.

- *Especialista en el lenguaje:* persona que apoya el análisis y al desarrollador en las consultas técnicas que éste tenga para la implementación de la solución del incidente identificado por el Usuario. Tiene conocimientos específicos sobre el lenguaje en que ha sido desarrollado el sistema que está en mantención.
- *Desarrollador:* persona encargada de realizar la programación necesaria para dar solución al problema identificado por el usuario.

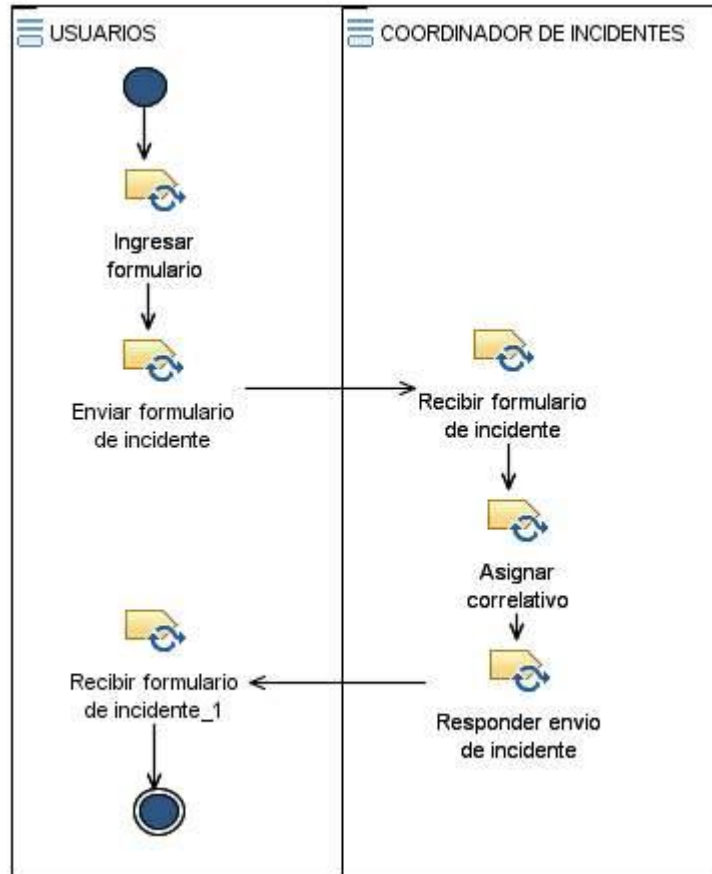
### 4.3.2.3. Actividades

Además de las entrevistas realizadas, se pudo obtener documentación sobre el procedimiento que actualmente es utilizado por TyC Software para el mantenimiento de software, además de las planillas usadas en dicho proceso. Esta documentación es bastante pobre y poco detallada, pero se pueden identificar dos grandes actividades dentro del proceso de mantenimiento: Ingreso y recepción de incidentes, y Asignación de incidentes.

Cabe destacar que se ha realizado el modelado de dichas actividades con la herramienta EPF Composer. Una pequeña parte de lo modelado se utilizará como apoyo para explicar cada una de estas actividades (Figura 4.2 y Figura 4.3).

#### Ingreso y recepción de incidentes:

En esta actividad participan: el usuario y el coordinador de incidentes. Comienza cuando el usuario detecta algún error al estar utilizando el sistema. El usuario ingresa los datos necesarios en el formulario de incidentes, adjunta los antecedentes pertinentes y luego envía todo esto por correo electrónico. Este correo es recibido por el coordinador de incidentes, quien le asigna un identificador (correlativo) y, además, registra la fecha de recepción del incidente. Acto seguido, cambia el estado del formulario a recibido y responde el correo enviado por el usuario. Así el usuario recibe el correo con el nuevo estado del formulario, el identificador del incidente y la fecha de su reporte.



**Figura 4.2** Actividad: Ingreso y recepción de incidentes

Asignación de incidentes:

En esta actividad se realizan todas las tareas asociadas a la asignación de recursos, identificación, replicación y solución del incidente reportado. Así es como en una primera instancia el coordinador de incidentes revisa el problema, las posibilidades de resolverlo y si esta probable solución está dentro de los márgenes del contrato vigente con la empresa cliente.

Una vez superada la tarea anterior, el coordinador de incidentes asigna los recursos que se necesitarán para poder dar una solución, así se hace entrega del problema a un desarrollador, quien es apoyado por un Especialista en el negocio y un Especialista en el lenguaje de programación. Luego, se informa vía correo electrónico a las personas que desempeñarán los roles y se realiza la actualización del listado de incidentes.

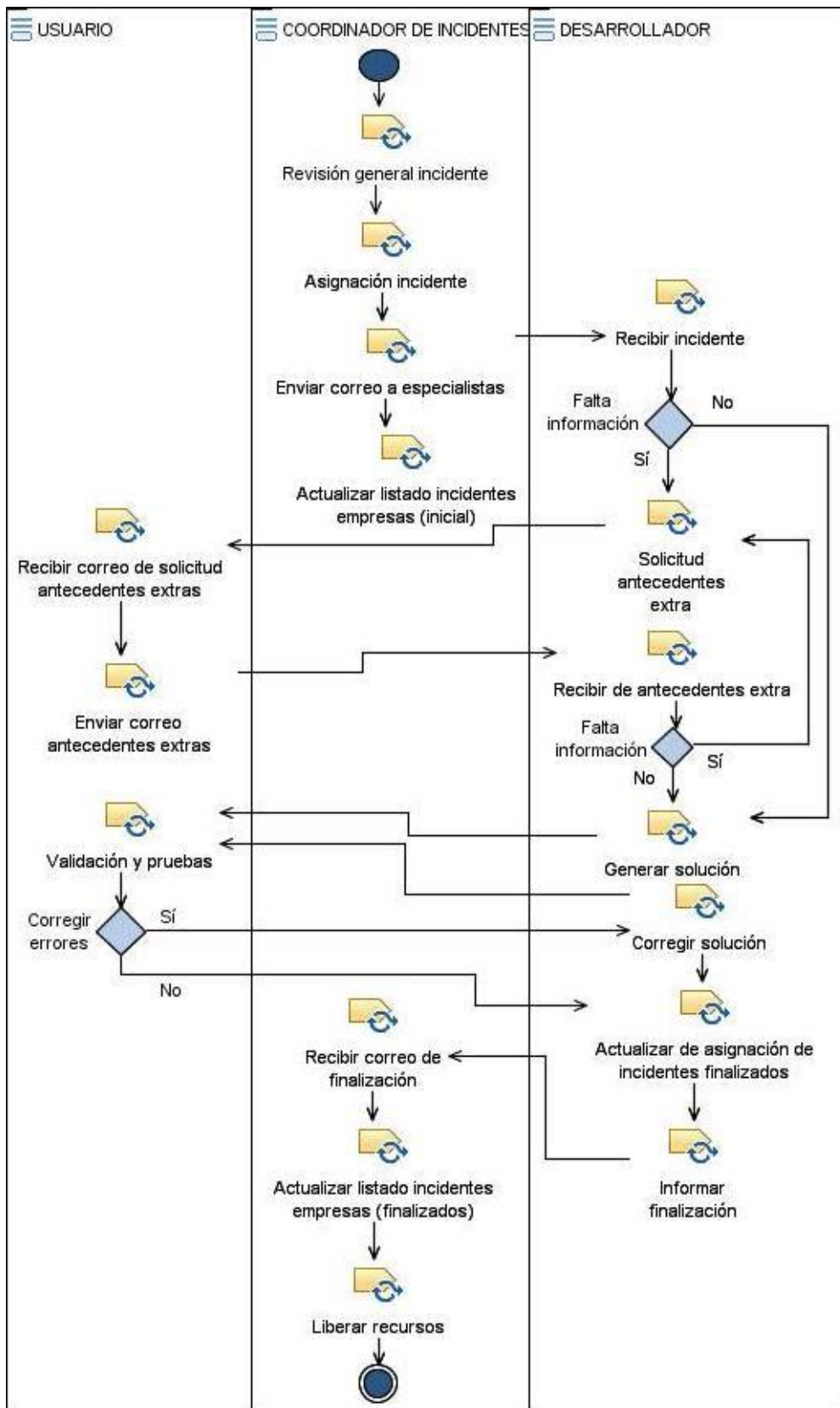


Figura 4.3 Actividad: Asignación de incidentes

El desarrollador verifica que los antecedentes entregados sean suficientes para poder entender y replicar el incidente, de no ser así, se solicitan antecedentes extras al usuario. Con los antecedentes necesarios, el desarrollador genera una solución al problema, la que es entregada al usuario para realizar las validaciones y pruebas pertinentes. Una vez aceptada la solución, se da por finalizada la participación del desarrollador en la actividad.

Para finalizar esta actividad, el coordinador de incidentes recibe los datos de finalización entregados por el desarrollador. Así se actualiza el listado de incidentes, cambiando el estado del incidente actual a finalizado.

## **4.4. Formulación de la mejora**

Siguiendo el modelo de mejora de procesos y luego de haber obtenido los resultados del diagnóstico, éstos se utilizarán como base para el análisis de la mejora y su posterior diseño. Previamente se debe realizar la planificación de las acciones a seguir en conjunto con la empresa.

### **4.4.1. Planificación de la mejora**

La planificación se llevó a cabo en conjunto con los principales involucrados dentro del área de soporte y mantenimiento de TyC Software, tomando como inicio la fecha de finalización del diagnóstico realizado y asignando fechas específicas (considerando cierta holgura en éstas) a las próximas actividades a realizar, Análisis, Diseño y Ejecución de la mejora.

Según lo acordado, la actividad Ejecución de la mejora para el proceso de soporte y mantenimiento eventualmente podría extenderse por más tiempo considerando las peticiones de mantenimiento recibidas que permitan aplicar las mejoras del proceso. Por lo tanto, la fecha de finalización será adecuada para poder realizar la revisión de la mejora.

### **4.4.2. Análisis de la mejora**

De los resultados obtenidos en la evaluación del actual proceso de soporte y mantenimiento de TyC Software y las entrevistas realizadas se puede concluir que:

- Se distinguen 3 tipos de incidencias (baja, media y alta), es decir, existe una categorización de los mantenimientos a realizar de acuerdo a su gravedad y la urgencia que presenta para el cliente.
- No se hace una distinción explícita entre las incidencias planificadas y no planificadas, sólo se utiliza la categorización asociada a la gravedad del problema.
- Hay 5 roles definidos (usuario, coordinador de incidentes, especialista en el negocio, especialista en el lenguaje y desarrollador), pero no todos los roles son utilizados al momento de realizar una mantención. En la práctica sólo se asignan los roles de: usuario, coordinador de incidentes (siempre asignado a la encargada de soporte) y desarrollador.
- La asignación de los roles es conocida por todas las personas involucradas en el proceso.
- La gran mayoría de los documentos utilizados son planillas definidas en formato digital.

- No existe conocimiento transversal del proceso de mantenimiento, es decir, no se conoce el proceso en su totalidad, la tendencia es conocer sólo el segmento del proceso que compete a la persona. La única persona que conoce el proceso en su totalidad es la encargada de soporte y mantenimiento de la empresa.
- Se distinguen dos actividades claras en el proceso de mantenimiento, ingreso y recepción de incidentes, y asignación de incidentes.
- En el proceso definido se mantiene un registro con los incidentes recibidos, el cual se actualiza a medida que el incidente recorre las distintas actividades.
- Existe una plantilla definida donde se especifica el detalle del incidente (número identificador del incidente, fecha de recepción, cliente, roles asignados, etc.) junto a la lista de elementos que fueron modificados y/o agregados para construir la solución.
- Se consideran tareas de validación y verificación para la solución desarrollada.
- No existe una actividad definida que reúna al equipo de mantenimiento para poder obtener retroalimentación entre el propio grupo.
- Existe un registro de todas las mantenciones realizadas a un cliente específico.
- El seguimiento de cada mantenimiento se limita al estado (campo en la plantilla de incidente) del incidente y conversación directa con el desarrollador a cargo.
- No se realizan cambios a la documentación del producto que ha presentado algún error.
- La finalización de una intervención al software sólo queda documentada en la plantilla del incidente, donde el desarrollador da por finalizado el desarrollo de la solución.
- Los procesos de la empresa son independientes, no existe conexión alguna entre ellos.
- La retirada del software antiguo se hace a través de la contra parte de la empresa cliente, apoyado por un integrante de TyC Software.
- Las distintas versiones de los software sometidos a mantención quedan respaldadas en un servidor especialmente habilitado para esto.
- No existe una actividad formal que dé por finalizado un contrato de mantención.

De estos puntos se puede concluir que es necesario:

1. Definir una actividad formal para el seguimiento del proceso y que facilite la administración de los recursos asociados.
2. Definir una actividad formal para la finalización de una mantención, donde se puedan obtener retroalimentación sobre lo realizado, que también apoye la administración de recursos liberándolos oportunamente y definiendo explícitamente un proceso para la integración de los cambios realizados en el software que se encuentra en el ambiente de producción.
3. Definir una actividad formal para la finalización de un contrato de mantención con un cliente, para poder entregar la documentación actualizada y dar por finalizado el proceso adecuadamente.
4. Definir nuevos roles que realmente se asignen durante la realización del proceso.
5. Dar a conocer el proceso completo a todos los involucrados, para que tomen conciencia de éste en su totalidad y pueda cooperar de mejor manera.



### 4.4.3. Diseño de la mejora

Para el diseño de la mejora se tomó como base el proceso actual, manteniendo la gran mayoría de los elementos existentes, y se le agregaron elementos del proceso de mantenimiento Agil\_Mantema para poder cumplir con las necesidades que fueron identificadas en el análisis.

Los elementos que se incorporarán de Agil\_Mantema deben ser adaptados para que calcen con los elementos del proceso de TyC que serán utilizados. Así se deberán unificar los productos de trabajo de entrada y salida de cada actividad para su congruencia.

#### 4.4.3.1. Tipos de mantenimiento

Se mantendrán los tipos de mantenimiento existentes en el proceso actual, agregando como subcategorías las definidas en Agil\_Mantema. De esta forma se podrán medir los niveles de calidad entregados, a partir de lo definido en Agil\_Mantema.

De esta forma las categorías se definen:

- *Alta*: afecta no permitiendo el desarrollo del negocio, ahora.
  - *Correctivo urgente (nivel básico)*: es aquel que se da en situaciones en que existe un error que bloquea el producto software en explotación, deben ser atendidas y resueltas en el menor tiempo posible.
- *Media*: afecta el desarrollo parcial del negocio, hoy.
  - *Correctivo no urgente (nivel intermedio)*: se produce cuando existe un error en el producto software en explotación que no es crítico, pero que impide el normal funcionamiento de éste durante un período de tiempo relativamente corto.
  - *Perfectivo (nivel intermedio)*: añade a al producto software en explotación nuevas funcionalidades requeridas por el cliente.
- *Baja*: afecta el desarrollo del negocio pero es manejable por el usuario, mañana.
  - *Adaptativo (nivel avanzado)*: se aplica ante situaciones en que el producto software en explotación se ajusta para continuar funcionando correctamente en un entorno distinto al concebido originalmente.
  - *Preventivo (nivel avanzado)*: se lleva a cabo cuando se desea mejorar las características internas del producto software en explotación para disminuir el esfuerzo de mantenimiento en el futuro.

#### 4.4.3.2. Roles

Para el nuevo proceso propuesto se eliminarán los roles que actualmente no se utilizan en la práctica (Especialista de negocio y Especialista de lenguaje), en su reemplazo se utilizará el rol Equipo de mantenimiento, propuesto en Agil\_Mantema. De esta forma se da una cierta libertad para asignar personal a este rol, que por definición es mucho más amplio y abarca a los roles del proceso actual de Desarrollador, Especialista de negocio y Especialista de lenguaje.

Por otra parte, del proceso actual se mantendrá el rol de Coordinador de incidentes, el que se asimila al rol Gestor de peticiones definido en Agil\_Mantema, pero que además tiene atribuciones de Responsable de mantenimiento. Además, se mantendrá de Usuario definido en el proceso original.

Finalmente, se agregará el rol de Cliente, quien estará asociado a las actividades de finalización del servicio de mantenimiento.

Así, los roles que conformarán el proceso propuesto son:

- *Cliente*: es la organización propietaria del producto, por lo tanto es quien recibe el servicio de mantenimiento. El propietario del producto representa a todos los interesados en el producto final. Sus áreas de responsabilidad son: la financiación del proyecto, retorno de la inversión del proyecto y el lanzamiento del proyecto. El propietario del producto por lo general formula peticiones de modificación del tipo perfectivo o adaptativo.
- *Usuario*: persona, de la empresa a la que se le presta el servicio de mantenimiento, que canaliza las peticiones de mantención hacia el área de soporte. El usuario hace llegar las peticiones de mantenimiento a través del formulario de incidentes.
- *Coordinador de incidentes*: encargado del área que actúa de nexo con la empresa cliente. Esta persona recibe los tickets de soporte y se contacta con el Usuario que ha reportado el problema. Realiza la asignación del Equipo de mantenimiento a cargo de resolver el problema. Por otro lado, puede llegar a realizar actividades de análisis del problema detectado y, al finalizar un ticket de soporte, actúa como validador.
- *Equipo de Mantenimiento*: es un grupo de personas que implementa las peticiones de mantenimiento. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos. También puede proponer peticiones de mantenimiento preventivo.

#### **4.4.3.3. Actividades**

Tal como se detectó en el diagnóstico, al proceso actual se deben agregar nuevas actividades. Las asociadas a la administración de los recursos, a la finalización de una iteración de la mantención y para la finalización de un ciclo de mantención. Para ello, se adaptarán los procesos de Agil\_Mantema que permitan cumplir con dichas actividades.

Por otro lado, se mantendrán las actividades actuales del proceso: Ingreso y recepción de incidentes, y Asignación de incidentes. La primera se mantendrá ya que los usuarios conocen el protocolo a seguir y la actividad no presenta mayores deficiencias. La actividad de Asignación de incidentes será dividida en dos, una actividad de Asignación de incidentes y otra de Resolución de incidentes. Ya que la actividad original tomaba en consideración ambos aspectos y su nombre no era representativo.

#### **4.4.4. Definición nuevo proceso**

A continuación se expondrá la definición formal del nuevo proceso, donde se explicitarán las actividades que lo componen, sus tareas, entradas y salidas, y los roles involucrados.

#### **4.4.4.1. TyC-1: Ingreso y recepción de incidentes**

Comienza cuando el usuario detecta algún error al estar utilizando el sistema. El usuario ingresa los datos necesarios en el formulario de incidentes, adjunta los antecedentes pertinentes y luego envía todo esto por correo electrónico. Este correo es recibido por el coordinador de incidentes, quién le asigna un identificador (correlativo) y, además, registra la fecha de recepción del incidente. Acto seguido, cambia el estado del formulario a recibido y responde el correo enviado por el usuario. Así el usuario recibe el correo con el nuevo estado del formulario, el identificador del incidente y la fecha de su reporte.

##### *Roles involucrados:*

- Coordinador de incidentes
- Usuario

##### *Tareas:*

*TyC-1.1. Ingreso de formulario de incidente:* el usuario ingresa los datos requeridos en el formulario de ingreso de incidentes para reportar una anomalía en el sistema, adjunta los antecedentes necesarios y envía toda esta información.

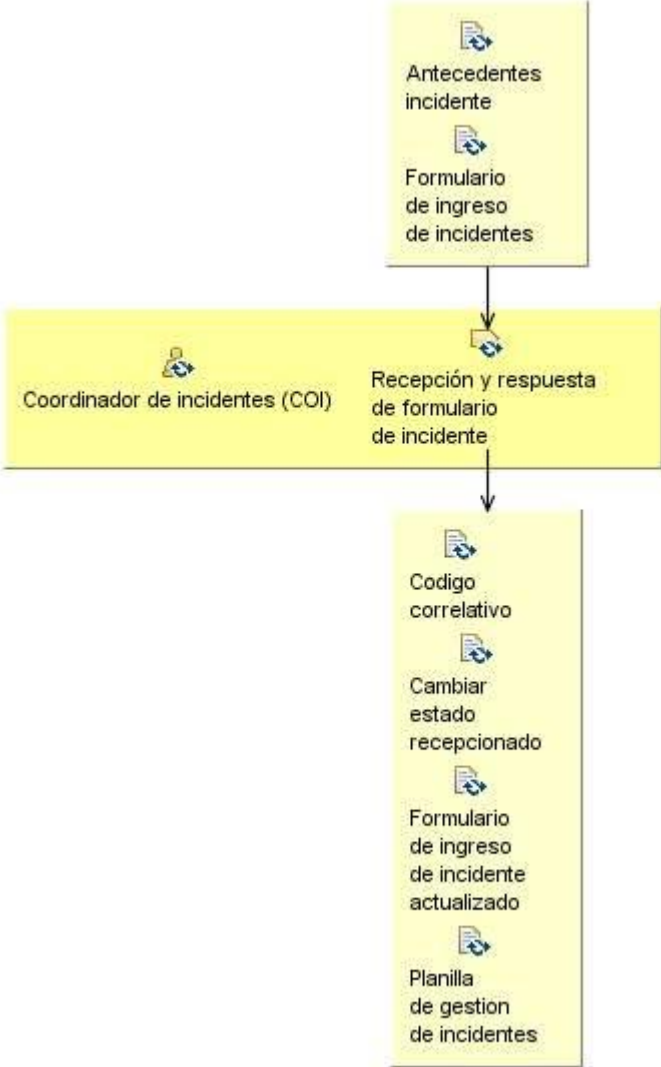
*TyC-1.2. Recepción y respuesta de formulario de incidente:* se recibe el formulario de ingreso de incidente en el área de soporte y mantenimiento de TyC Software. Se registra en el formulario la fecha de recepción, se le asigna el código correlativo correspondiente y se actualiza su estado a recibido. El formulario actualizado es ingresado a la planilla de gestión de incidentes. Se informa al usuario la recepción del incidente a través de un mail.

*TyC-1.3. Recepción respuesta de formulario de incidente:* el usuario recibe el formulario de ingreso de incidente actualizado.

##### *Elementos modelados:*

En la Figura 4.4 y Figura 4.5 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.6 se expone el diagrama de actividad.

Entradas y salidas - Eys:



**Figura 4.4** Eys TyC-1Coordinador de incidentes

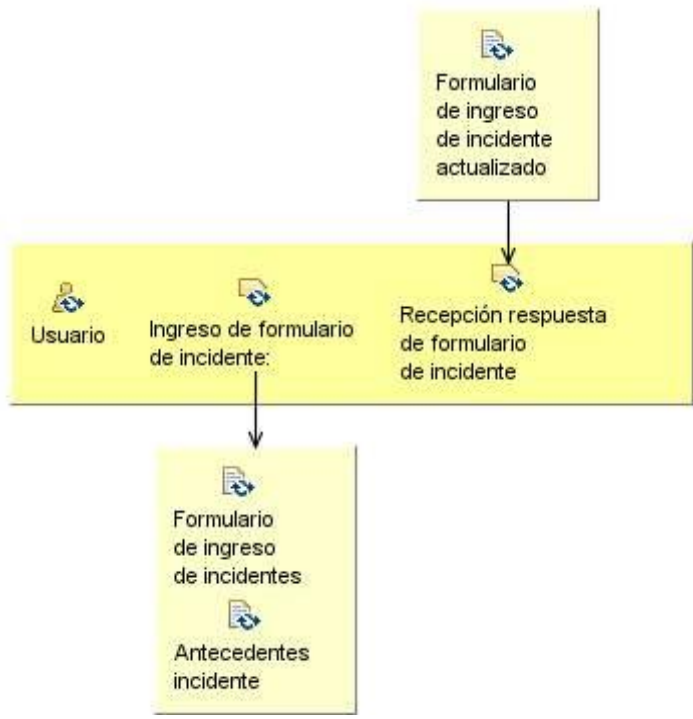


Figura 4.5 EyS TyC-1Usuario

Modelado:

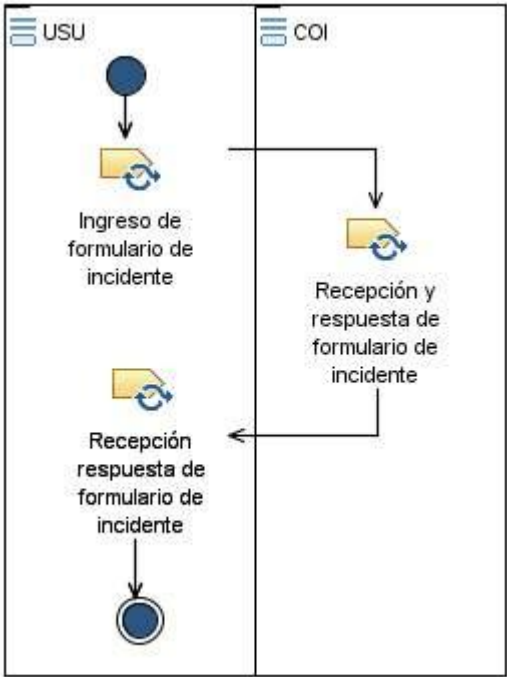


Figura 4.6 Diagrama actividad TyC-1

#### 4.4.4.2. TyC-2: Asignación de incidentes

El coordinador de incidentes revisa el problema, las posibilidades de resolverlo y si esta probable solución está dentro de los márgenes del contrato vigente con la empresa cliente. Posteriormente se asignan los recursos que se necesitarán para poder dar una solución. Así se entrega el problema al equipo de mantenimiento asignado. El equipo de mantenimiento debe revisar los antecedentes para corroborar que sean suficientes, si no son suficientes se solicitan antecedentes extras al usuario. Con esto cumplido se podrá comenzar con el desarrollo de la solución.

##### Roles involucrados:

- Coordinador de incidentes
- Equipo de mantenimiento
- Usuario

##### Tareas:

*TyC-2.1. Revisión contrato:* se verifica que el incidente reportado esté dentro de los márgenes estipulados en el contrato con el cliente (estado de aprobación de mantenimiento: Verdadero). De no ser así, la iteración de mantenimiento se termina hasta resolver la situación contractual (estado de aprobación de mantenimiento: Falso).

*TyC-2.2. Revisión general del incidente:* si el estado de aprobación de mantenimiento es verdadero: se revisa de manera general el incidente reportado para poder estimar los recursos que se le deben asignar y el tipo de mantenimiento que recibirá el incidente.

*TyC-2.3. Asignación del incidente:* se asignan las personas que tomarán el rol de Equipo de mantenimiento y que estarán a cargo de darle solución al incidente ingresado. Se les envía un correo a todos los involucrados y se realiza la asignación de los recursos en el formulario de asignación de incidentes.

*TyC-2.4. Revisión del incidente y antecedentes:* el equipo de mantenimiento asignado revisa el incidente ingresado junto a los antecedentes adjuntados. Si los antecedentes no son suficientes para aclarar el incidente, se solicitarán antecedentes extras al usuario (Estado de suficiencia de antecedentes: Falso). De lo contrario, se procederá con la próxima actividad (Estado de suficiencia de antecedentes: Verdadero).

*TyC-2.4.1. Solicitud antecedentes extras:* se le solicitan al usuario antecedentes extras sobre el incidente ingresado, para poder comprenderlo de la mejor manera, poder reproducirlo en el ambiente de mantenimiento y darle una solución adecuada.

*TyC-2.4.2. Envío antecedentes extras:* el usuario reúne antecedentes extras sobre el incidente ingresado para poder enviarlos al equipo de mantenimiento.

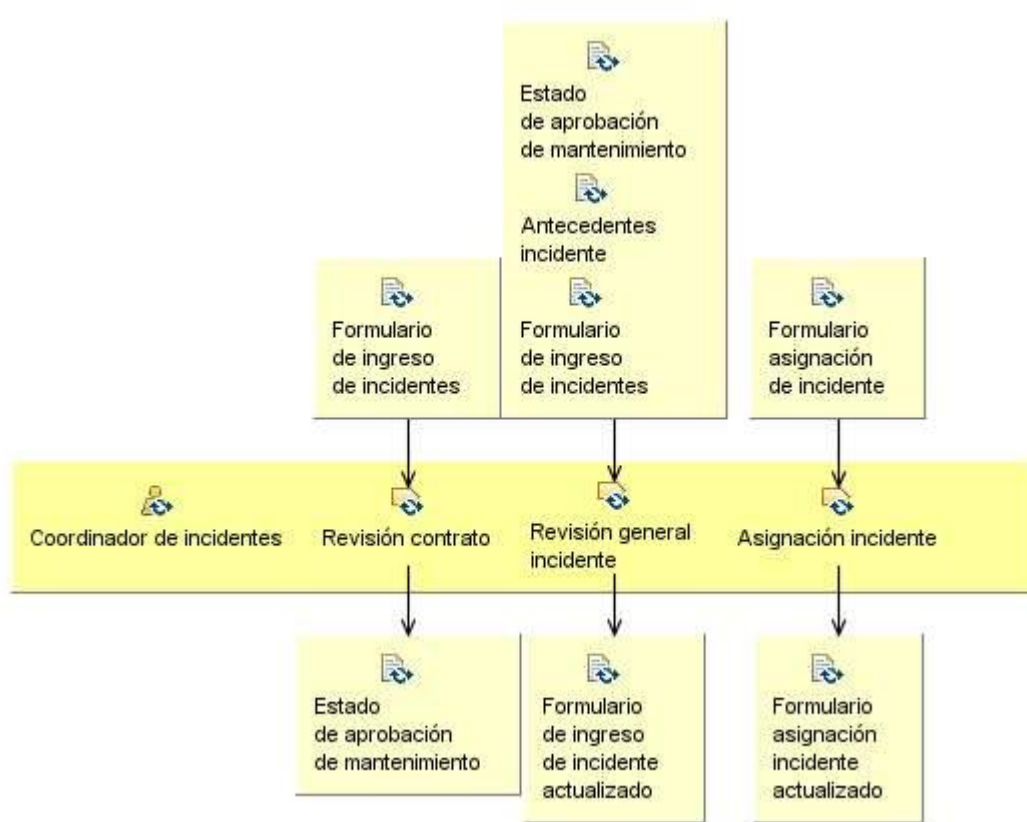
*TyC-2.4.3. Recepción y revisión de antecedentes extras:* el equipo de mantenimiento recibe y revisa los antecedentes extras enviados por el usuario. Si estos son suficientes se

continúa con la próxima actividad (Estado de suficiencia de antecedentes: Verdadero). Si no, se vuelven a solicitar antecedentes extra (Estado de suficiencia de antecedentes: Falso).

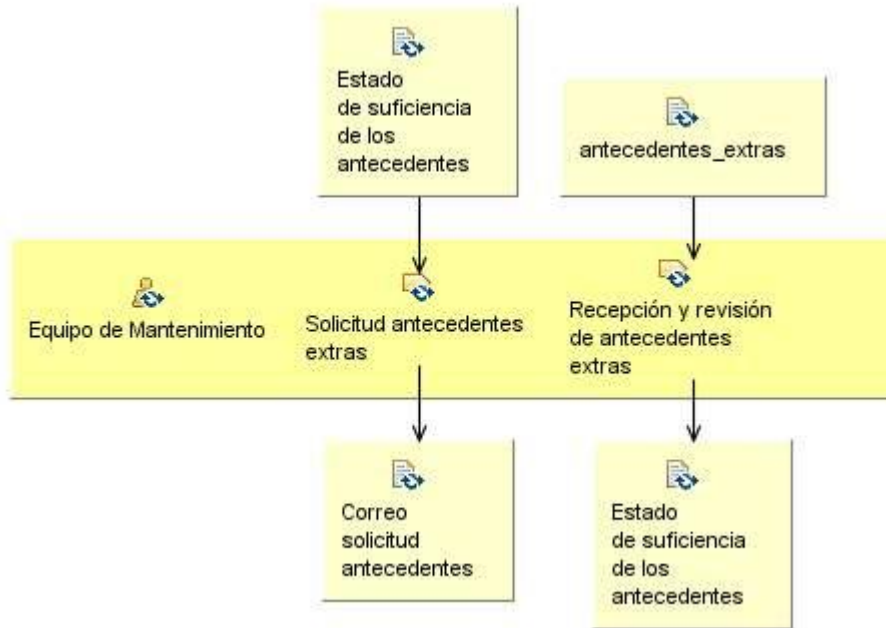
Elementos modelados:

En la Figura 4.7 hasta Figura 4.9 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.10 se expone el diagrama de actividad.

Entradas y salidas - EyS:



**Figura 4.7** EyS TyC-2Coordinador de incidentes



**Figura 4.8** EyS TyC-2Equipo de mantenimiento



**Figura 4.9** EyS TyC-2Usuario



Modelado:

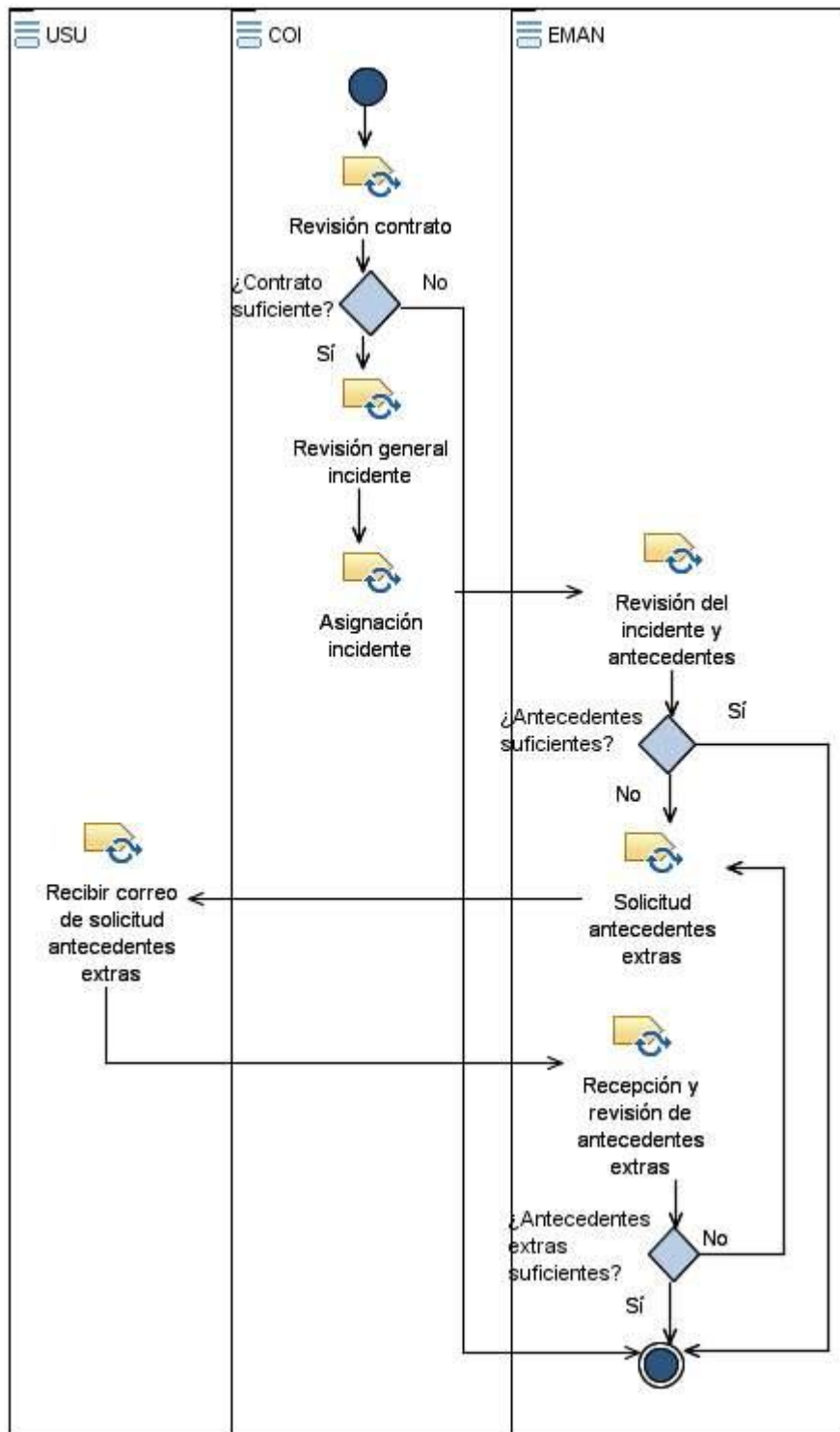


Figura 4.10 Diagrama actividad TyC-2

#### 4.4.4.3. TyC-3: Resolución de incidente

Con los antecedentes necesarios, el Equipo de mantenimiento genera una solución al incidente ingresado. Esta solución es verificada y validada por el Coordinador de incidentes y el Usuario. Si la solución planteada no satisface al Usuario, se irá corrigiendo hasta que sea aprobada. Una vez aceptada la solución, se da por finalizada la actividad y se debe informar al Coordinador de incidentes.

##### Roles involucrados:

- Coordinador de incidentes
- Equipo de mantenimiento
- Usuario

##### Tareas:

*TyC-3.1. Generación de la solución:* el Equipo mantenedor desarrolla una solución para el incidente ingresado por el usuario.

*TyC-3.2. Verificación y validación ambiente mantenimiento:* el Coordinador de incidentes realiza verificaciones y validaciones la solución generada por el Equipo de mantenimiento. Si no cumple con los requerimientos para solucionar el incidente (Estado de suficiencia de la solución: Falso) se devuelve la solución al Equipo de mantenimiento con las correcciones a desarrollar. Si no, se hace entrega de la solución al usuario para que realice sus propias pruebas (Estado de suficiencia de la solución: Verdadero).

*TyC-3.3. Pruebas ambiente cliente:* se entrega la solución generada al usuario para que realice las pruebas pertinentes. Si está no cumple con los requerimientos para solucionar el incidente (Estado de suficiencia de la solución: Falso) se devuelve la solución al Equipo de mantenimiento con las correcciones a desarrollar. Si no, se finaliza con esta etapa (Estado de suficiencia de la solución: Verdadero).

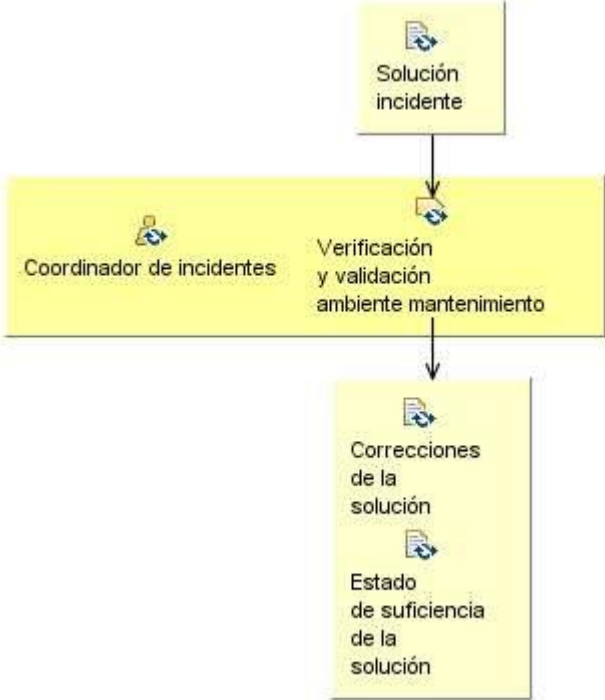
*TyC-3.3.1. Corrección de la solución:* se reciben las correcciones asociadas a la solución desarrollada para poder implementarlas en una nueva versión de la solución.

*TyC.3.4. Informar finalización:* se actualiza el estado del formulario de asignación de incidentes a finalizado y se le informa al coordinador de incidentes sobre la finalización de la generación de la solución.

##### Elementos modelados:

En la Figura 4.12 y Figura 4.13 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.14 se expone el diagrama de actividad.

Entradas y salidas - EYS:



**Figura 4.11** Eys TyC-3Coordinador de incidentes



**Figura 4.12** Eys TyC-3Equipo de mantenimiento

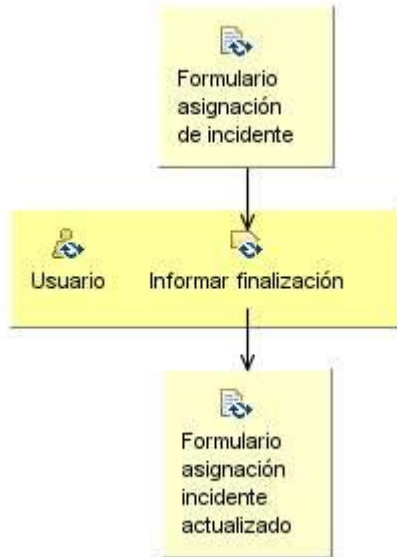


Figura 4.13 Eys TyC-3Usuario

*Modelado:*

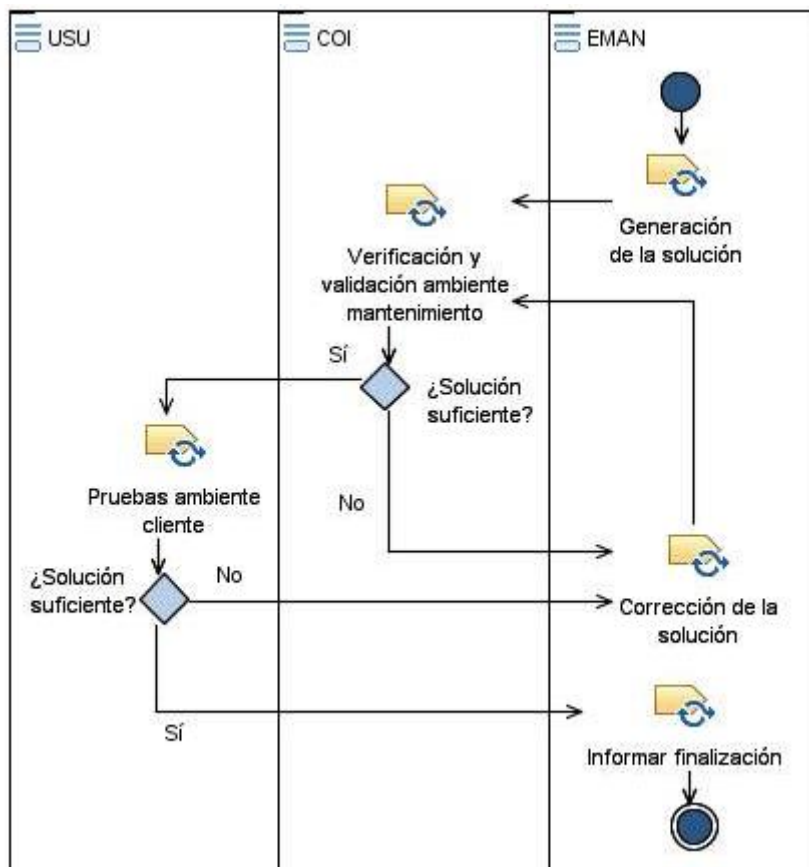


Figura 4.14 Diagrama actividad TyC-3

#### 4.4.4.4. TyC-4: Seguimiento de incidentes

El propósito de esta actividad es conducir reuniones de control para hacer un seguimiento del estado de avance y resolver problemas de las intervenciones realizadas en la Resolución de incidentes. En esta actividad se lleva a cabo la revisión del trabajo realizado y se solucionan las dificultades encontradas.

##### Roles involucrados:

- Coordinador de incidentes
- Equipo de mantenimiento

##### Tareas:

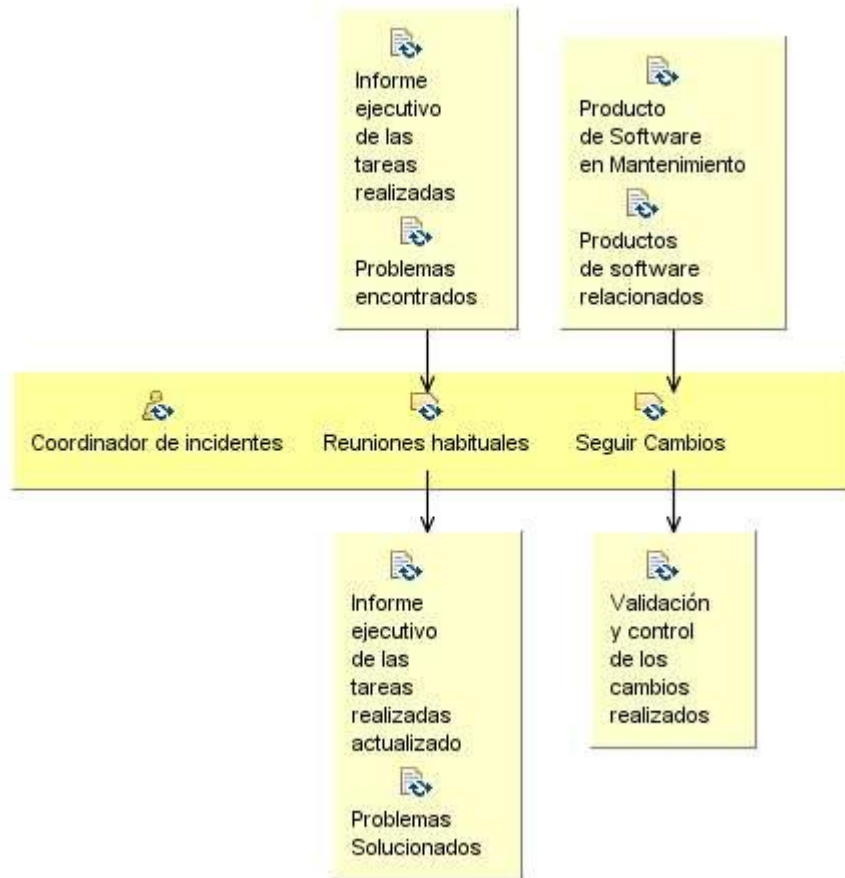
*TyC-4.1: Reuniones habituales:* en estas reuniones, de al menos 15 minutos, se reúne todo el Equipo de Mantenimiento y cada miembro del equipo expone sólo los siguientes temas: ¿Qué es lo que hizo desde la última reunión?; ¿Qué es lo que va hacer hasta la siguiente reunión? Es muy importante que al salir de la reunión todos los involucrados sepan lo que deben hacer y que todos están alineados en la misma dirección: el mantenimiento del software; ¿Cómo se lo llevará a cabo? ¿Algo le hace falta? Todos deben tener claro cómo realizar su trabajo, con esta pregunta deben surgir los problemas que tienen las personas para la realización de la Petición de Modificación. Sólo se tratan estos temas para que la reunión sea rápida y no se pierda tiempo, su finalidad es alinear a las personas en la misma dirección y sacar a la luz los problemas e impedimentos que hay para solucionarlos y conseguir el objetivo.

*TyC-4.2: Seguir los cambios:* se realiza el control de los cambios producidos en la ejecución de la Resolución de incidentes, este control abarca los siguientes aspectos: realizar la traza de los cambios que la Petición de Modificación ha provocado a lo largo de los procesos de desarrollo implicados; verificar que se han realizado satisfactoriamente las pruebas unitarias, de integración y del sistema que se consideraron necesarias para los componentes a modificar; comprobar que sólo se ha modificado lo establecido y, en caso contrario, justificar el motivo; y llevar el control de los distintos desarrollos existentes en paralelo sobre un mismo componente, con el fin de coordinar las modificaciones incluidas en cada uno de ellos, y asegurar que en el paso a producción se implanten correctamente.

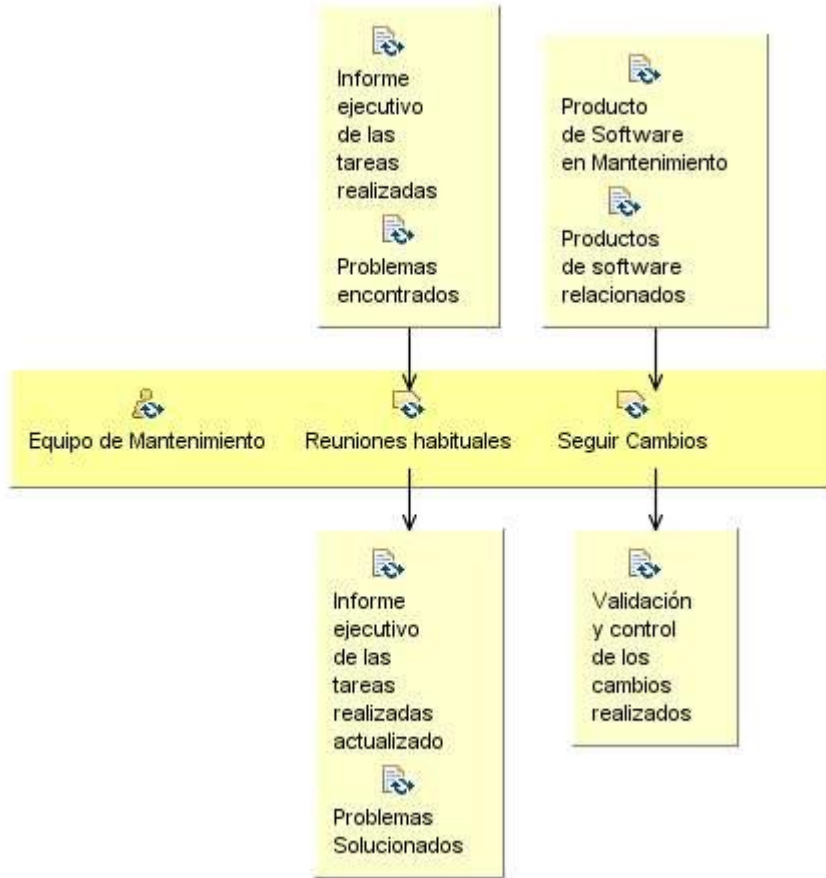
##### Elementos modelados:

En la Figura 4.15 y Figura 4.16 se exponen las entradas y salidas de la tarea ejecutada por cada rol. En la Figura 4.17 se expone el diagrama de actividad.

Entradas y salidas - EyS:

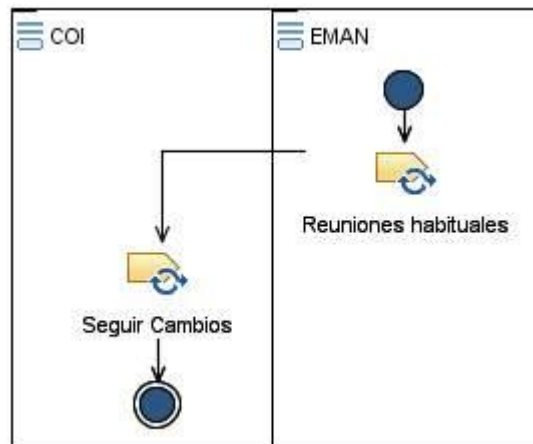


**Figura 4.15** EyS TyC-4Coordinador de incidentes



**Figura 4.16** Eys TyC-4Equipo de mantenimiento

*Modelado:*



**Figura 4.17** Diagrama actividad TyC-4

#### 4.4.4.5. TyC-5: Finalización de la intervención

En esta actividad el Equipo de Mantenimiento junto al Usuario verifica el correcto funcionamiento del producto de software nuevo en el entorno de producción. Además, se realizan las actualizaciones necesarias al Manual de Usuario. Finalmente, se define cuál será la próxima iteración del ciclo de mantenimiento. Con esto se da por terminado un proceso de intervención relacionado con un Formulario de asignación de incidentes.

##### Roles involucrados:

- Cliente
- Coordinador de incidentes
- Equipo de mantenimiento

##### Tareas:

*TyC-5.1: Verificar y validar corrección con el cliente:* el Equipo de Mantenimiento y la organización usuaria del sistema se reúnen para comprobar que el producto intervenido funciona correctamente. En esta tarea debe haber interacción con el Cliente para recabar impresiones, sugerencias, mejoras y su relevancia sobre el producto que se ha entregado. También se evalúan y registran nuevos posibles cambios en Planilla de gestión de incidentes.

*TyC-5.2: Pasar a producción:* el Equipo de Mantenimiento pasa al entorno de producción el software corregido para su utilización por parte de los Usuarios.

*TyC-5.3: Documentar manual de usuario:* el Equipo de Mantenimiento debe documentar o re-documentar el Manual de Usuario, si es que ha cambiado el modo de operación del software o si ha agregado nuevas funcionalidades.

*TyC-5.4: Registrar la intervención:* la intervención de modificación queda registrada, según los procedimientos de la organización.

*TyC-5.5: Reunión de retrospectiva:* se deben extraer las mejores prácticas de la última intervención. Aquí el Coordinador de incidentes pregunta a su Equipo de Mantenimiento: ¿Qué aspectos positivos rescatan de la última intervención? ¿Qué se puede mejorar? Toda esta información es tomada para optimizar la próxima intervención.

##### Elementos modelados:

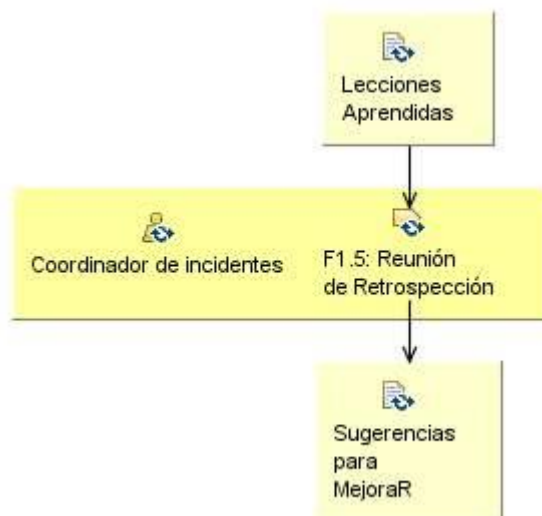
En la Figura 4.18 hasta la Figura 4.20 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.21 se expone el diagrama de actividad.



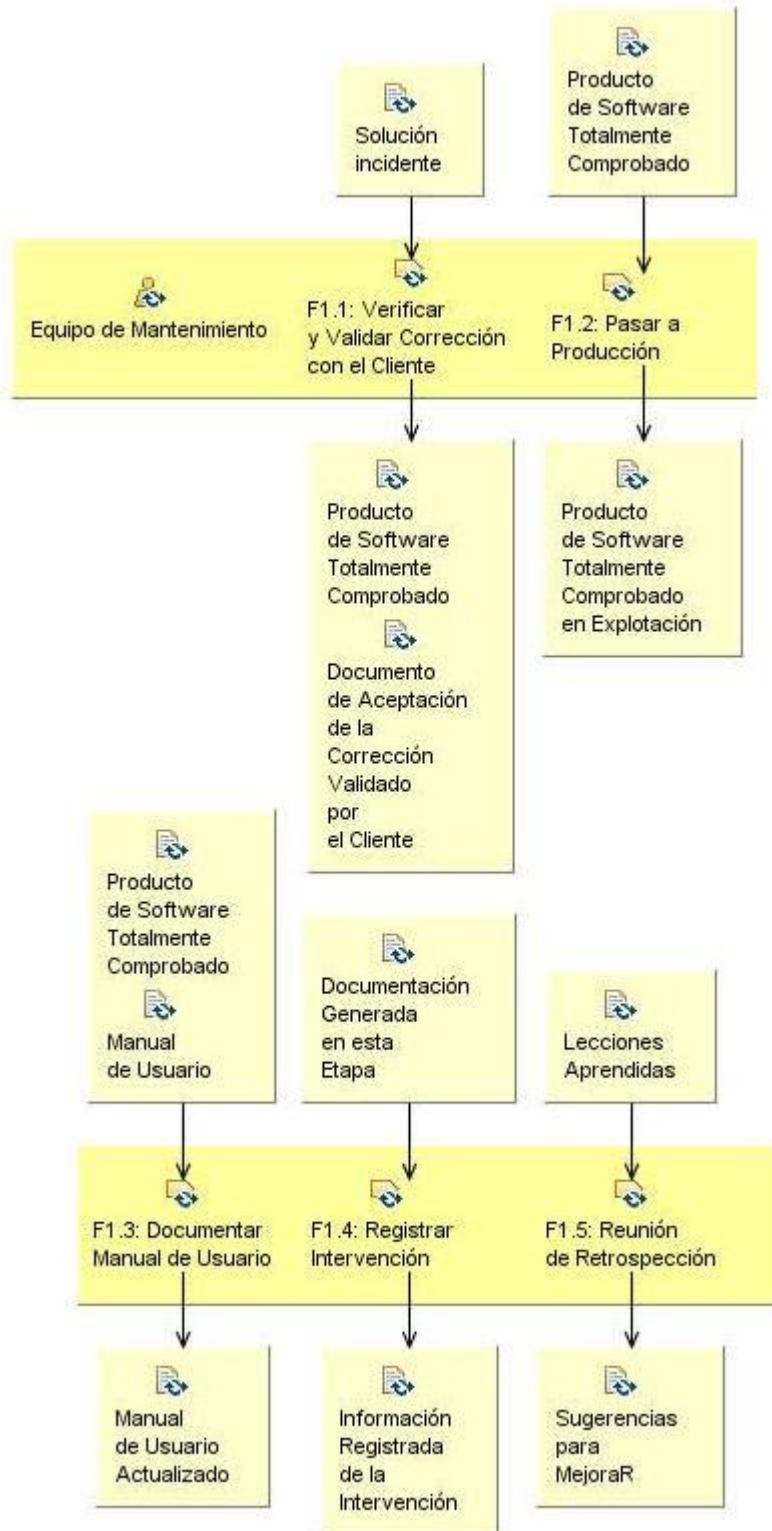
Entradas y salidas - EyS:



**Figura 4.18** EyS TyC-5Cliente

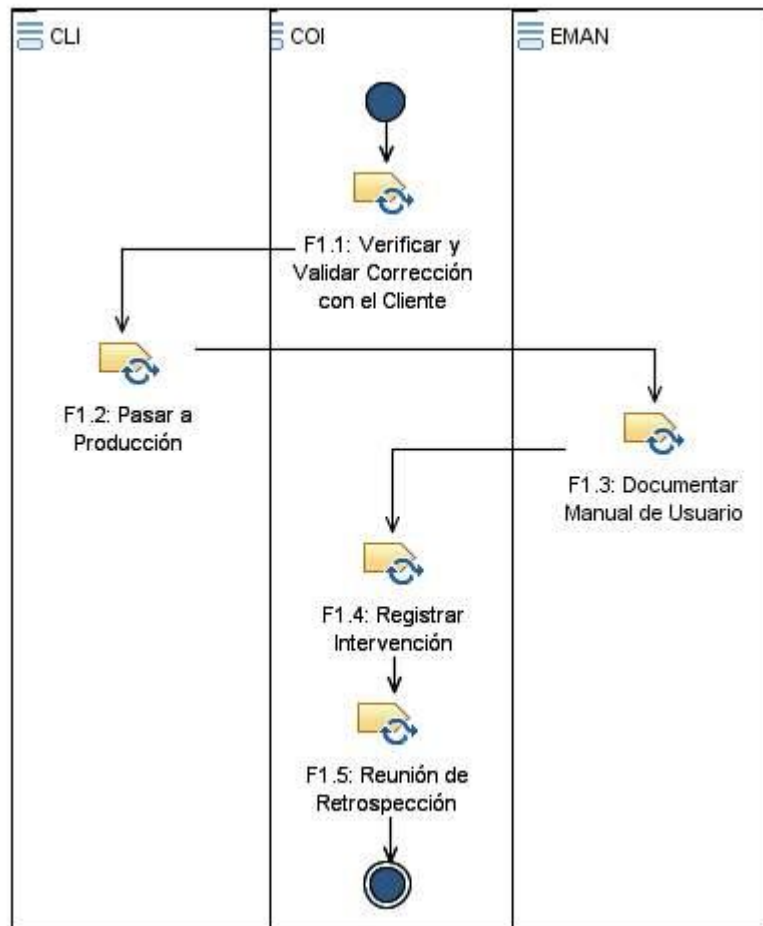


**Figura 4.19** EyS TyC-5Coordinador de incidentes



**Figura 4.20** Eys TyC-5Equipo de mantenimiento

Modelado:



**Figura 4.21** Diagrama actividad TyC-5

#### 4.4.4.6. TyC-6: Retiro

En esta actividad el Equipo de Mantenimiento genera la planificación para el retiro del Producto de Software Antiguo en Explotación. Esto se comunica formalmente a los Usuarios para que no hayan problemas durante este proceso. Se retira la aplicación antigua y se respaldan los datos que ésta pueda contener.

Roles involucrados:

- Cliente
- Equipo de mantenimiento
- Usuario

### Tareas:

*TyC-6.1: Desarrollar plan de retirada:* el Equipo de Mantenimiento redacta un documento en el que describe cuándo y cómo se llevará a cabo el retiro del Producto de Software Antiguo en Explotación.

*TyC-6.2: Notificar futuro retiro:* el Equipo de Mantenimiento notifica al cliente el momento en el que se ejecutará la retirada del software.

*TyC-6.3: Ejecutar en paralelo:* el Usuario, con el visto bueno del cliente y bajo la supervisión del Equipo de Mantenimiento, realiza operaciones reales sobre el software que se va a retirar y el software nuevo a implantar (si es que aquél va a ser sustituido).

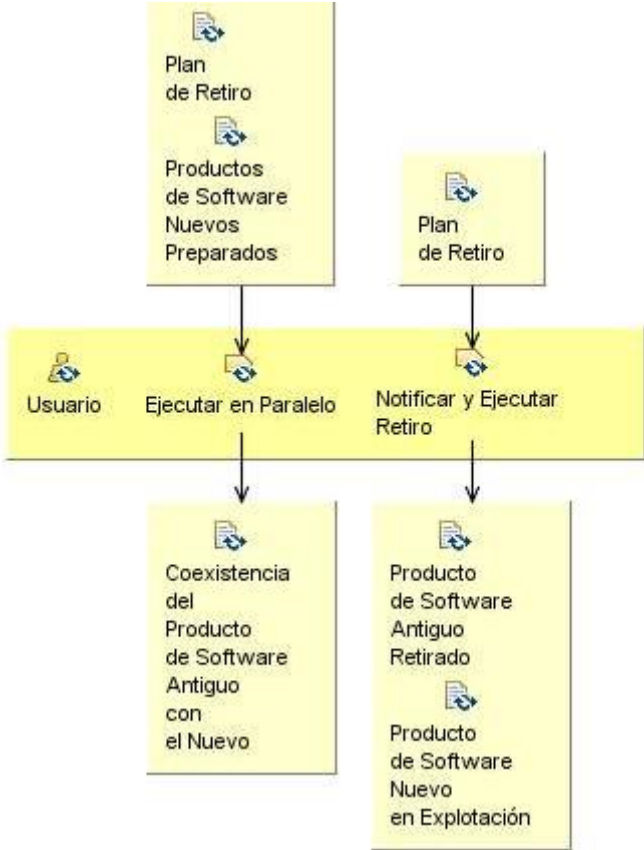
*TyC-6.4: Notificar y ejecutar retiro:* se notifica la inminencia de la retirada. El Producto de Software Antiguo en Explotación es retirado y el Producto de Software Nuevo es el que queda en funcionamiento para explotación.

*TyC-6.5: Almacenar datos del producto de software antiguo:* los datos del Producto de Software Antiguo Retirado son almacenados a modo de prevención ante cualquier necesidad de éstos.

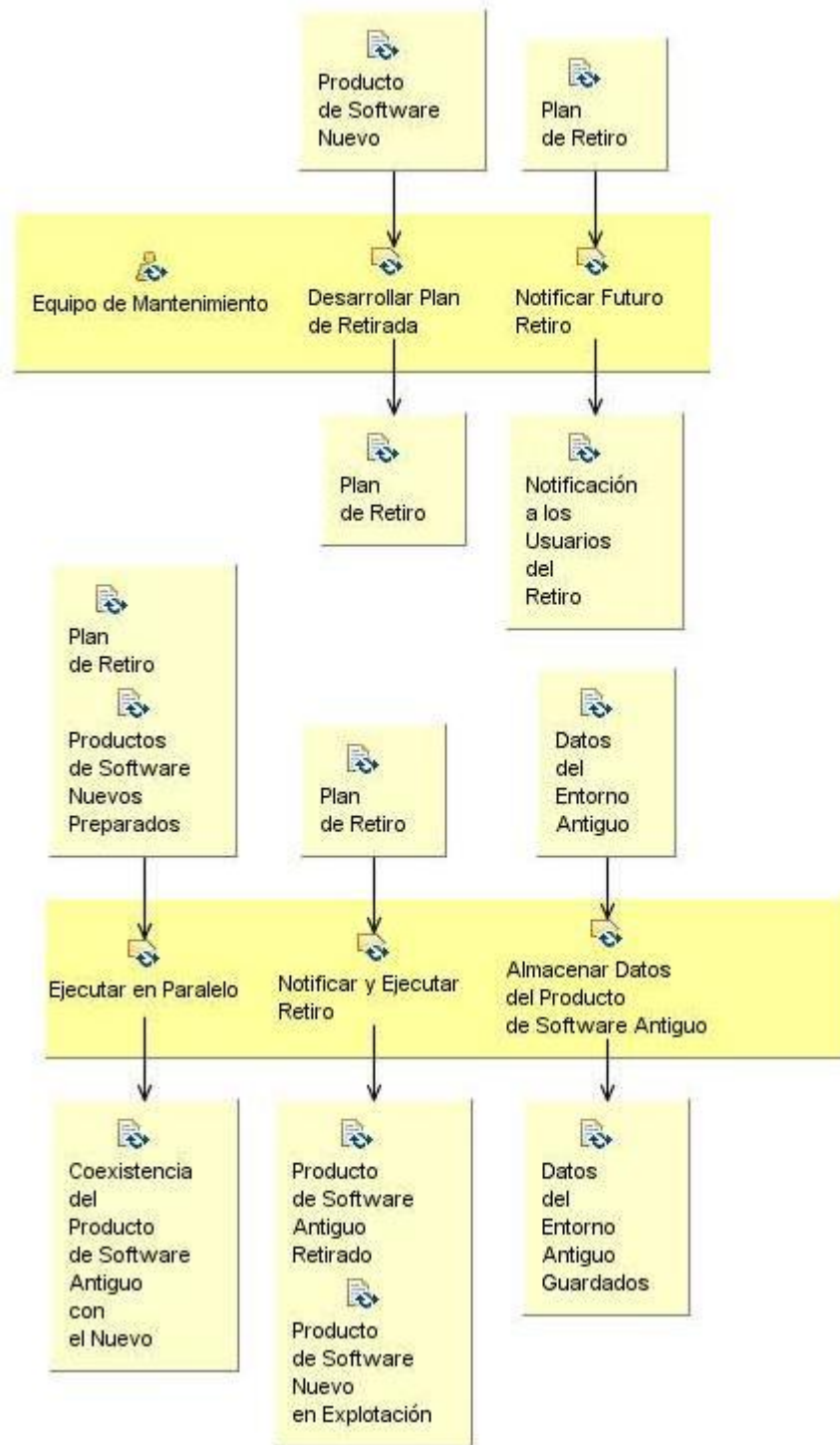
### Elementos modelados:

En la Figura 4.22 hasta la Figura 4.24 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.25 se expone el diagrama de actividad.

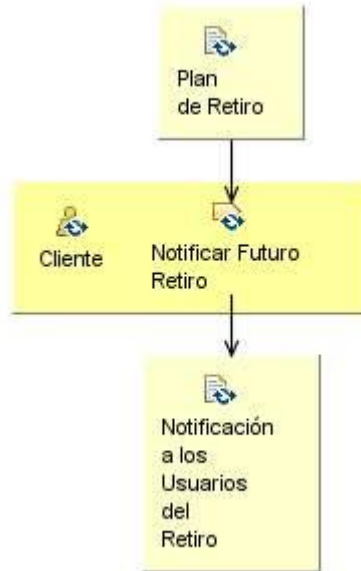
Entradas y salidas - Eys:



**Figura 4.22** Eys TyC-6 Usuario

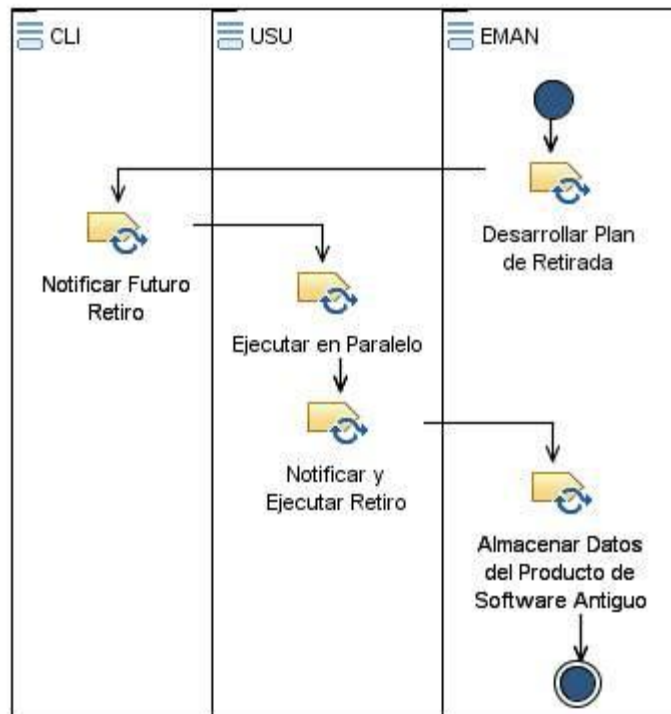


**Figura 4.23** EyS TyC-6 Equipo de mantenimiento



**Figura 4.24** Eys TyC-6 Cliente

*Modelado:*



**Figura 4.25** Diagrama actividad TyC-6

#### 4.4.4.7. TyC-7: Finalización del servicio

El propósito de esta actividad es dar por finalizado de manera formal el servicio de mantenimiento de software al Cliente. Esta actividad se realiza cuando el mantenedor deja de prestar sus servicios a la organización Cliente.

##### Roles involucrados:

- Cliente
- Equipo de mantenimiento

##### Tareas:

*TyC-7.1: Entregar inventario y la documentación:* se entregan los productos de software generados y modificados al Cliente.

*TyC-7.2: Terminar entrega del servicio:* la Organización de mantenimiento deja de prestar sus servicios al cliente con carácter definitivo.

##### Elementos modelados:

En la Figura 4.26 y Figura 4.27 se exponen las entradas y salidas de las tarea ejecutada por cada rol. En la Figura 4.28 se expone el diagrama de actividad.

##### Entradas y salidas - EyS:



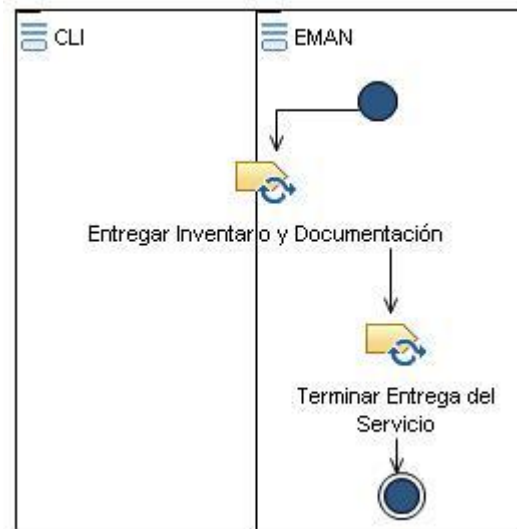
**Figura 4.26** EyS TyC-7 Cliente





**Figura 4.27** Eys TyC-7 Equipo de mantenimiento

Modelado



**Figura 4.28** Diagrama actividad TyC-7

## 4.5. Ejecución de la mejora

Esta etapa de la aplicación del proceso de mejoras contempló dos actividades. La primera, un conjunto de sesiones de capacitación para dar a conocer el nuevo proceso de mantenimiento a los involucrados en éste. En la segunda actividad se aplicará nuevamente el Cuestionarios de Mantenimiento Agil\_Mantema [Gajardo *et al.*, 2008] para poder calcular cuantitativamente el cambio que significó la aplicación del nuevo proceso de mantenimiento.

### 4.5.1. Capacitaciones

En conjunto con la empresa se programó un itinerario de capacitaciones durante una semana para poder dar a conocer el nuevo proceso de mantenimiento a los involucrados en éste. La duración y contenido de cada capacitación realizado acorde a las personas que podían estar presentes, pues hubo que adaptar las sesiones a los horarios del personal de la empresa.

#### Sesión 1:

Día: Lunes 8 de Agosto.

Duración: 2 horas.

Contenido presentado:

- Introducción
- Ingreso y recepción de incidentes
- Asignación de incidentes
- Resolución de incidentes

#### Sesión 2:

Día: Martes 9 de Agosto.

Duración: 1 hora y media.

Contenido presentado:

- Introducción
- Ingreso y recepción de incidentes
- Asignación de incidentes

#### Sesión 3:

Día: Miércoles 10 de Agosto.

Duración: 1 hora.

Contenido presentado:

- Seguimiento de incidentes
- Finalización de la intervención

#### Sesión 4:

Día: Jueves 11 de Agosto.

Duración: 1 hora y media.

Contenido presentado:

- Resolución de incidentes
- Seguimiento de incidentes
- Finalización de la intervención

Sesión 5

Día: Viernes 12 de Agosto.

Duración: 1 hora.

Contenido presentado:

- Retiro
- Finalización del servicio

### 4.5.2. Diagnóstico final

El cuestionario fue entregado a las mismas 6 personas que se les entregó en el diagnóstico inicial y los resultados obtenidos, agrupados por actividades del proceso de mantenimiento, se presentan desde la tabla 4.15 hasta la tabla 4.25 (donde D significa digital y M significa manual):

Tabla 4.15 (A.1.) Planificación del proceso

	Sí	No	D ó M	Respuesta
<b>A.1 PLANIFICACIÓN DEL PROCESO</b>				
<b>A.1.1 Asignar responsables</b>				
A.1.1.1	6	0		
A.1.1.2				Jefe soporte, encargado soporte
A.1.1.3	6	0		
A.1.1.4				Jefe soporte, encargado soporte
A.1.1.5	0	6		
A.1.1.6	6	0		
A.1.1.7	6	0		
A.1.1.8			D	
<b>A.1.2 Adquirir conocimiento de la aplicación</b>				
A.1.2.1				Documentación, código fuente, referencias, entrevistas, observación
A.1.2.2	3	3		
A.1.2.3				Jefe soporte, encargado soporte
A.1.2.4	1	5		

	A.1.2.5			Formulario asignación de incidentes
	A.1.2.6	5	1	
	A.1.2.7	4	1	
	A.1.2.8			D
<b>A.1.3 Preparar entornos de pruebas</b>				
	A.1.3.1	6	0	
	A.1.3.2	6	0	
	A.1.3.3			Copias de software, preparación BD y archivos
	A.1.3.4	6	0	
	A.1.3.5			Depende del ambiente de control en el soporte
<b>A.1.4 Definir procedimientos de petición de modificación</b>				
	A.1.4.1	6	0	
	A.1.4.2			Jefe soporte, encargado soporte
	A.1.4.3	6	0	
	A.1.4.4	6	0	
	A.1.4.5	6	0	
	A.1.4.6			D
<b>A.1.5 Verificar la planificación del proceso</b>				
	A.1.5.1	5	1	
	A.1.5.2			Jefe soporte, encargado soporte
	A.1.5.3	4	2	
	A.1.5.4	4	0	
	A.1.5.5			D
	A.1.5.6	5	1	
	A.1.5.7			Desarrollador
	A.1.5.8	4	2	
	A.1.5.9	3	3	
	A.1.5.10			D
	A.1.5.11	6	0	
	A.1.5.12			Jefe soporte, encargado soporte
	A.1.5.13	6	0	
	A.1.5.14	6	0	
	A.1.5.15			D
<b>A.1.6 Validar petición de modificación</b>				
	A.1.6.1	6	0	
	A.1.6.2			Jefe soporte, encargado soporte
	A.1.6.3	6	0	
	A.1.6.4	6	0	
	A.1.6.5			D

Tabla 4.16 (A.2.) Atención petición de modificación

	Sí	No	D ó M	Respuesta
<b>A.2 ATENCIÓN PETICIÓN DE MODIFICACIÓN</b>				
<b>A.2.1 Recibir petición de modificación</b>				
	A.2.1.1	6	0	
	A.2.1.2			Jefe soporte, encargado soporte
	A.2.1.3	6	0	
	A.2.1.4	6	0	
	A.2.1.5	6	0	
	A.2.1.6		D	
<b>A.2.2 Decidir tipo de mantenimiento</b>				
	A.2.2.1	6	0	
	A.2.2.2			Documento, e-mail, oral
	A.2.2.3	6	0	
	A.2.2.4	6	0	
	A.2.2.5		D	
	A.2.2.6			En base al análisis de la petición, en reunión con el jefe de proyecto
	A.2.2.7	6	0	
<b>A.2.3 Verificar la administración de la petición de modificación</b>				
	A.2.3.1	5	1	
	A.2.3.2			Encargado soporte
	A.2.3.3	5	1	
	A.2.3.4	4	2	
	A.2.3.5		D	

Tabla 4.17 (A.3.) Tipo de mantenimiento no planificable

	Sí	No	D ó M	Respuesta
<b>A.3 TIPO DE MANTENIMIENTO NO PLANIFICABLE</b>				
<b>A.3.1 Análisis del error</b>				
	A.3.1.1	6	0	
	A.3.1.2	6	0	
	A.3.1.3	6	0	
	A.3.1.4	6	0	
	A.3.1.5		D	

Tabla 4.18 (A.4.) Intervención correctiva urgente

	Sí	No	D ó M	Respuesta
<b>A.4 INTERVENCIÓN CORRECTIVA URGENTE</b>				
<b>A.4.1 Realizar acciones correctivas</b>				
	A.4.1.1	6	0	
	A.4.1.2	6	0	
	A.4.1.3			Desarrollador
	A.4.1.4	6	0	
	A.4.1.5	6	0	
	A.4.1.6	6	0	
	A.4.1.7		D	
<b>A.4.2 Ejecutar pruebas unitarias</b>				
	A.4.2.1	6	0	
	A.4.2.2			Jefe soporte, encargado soporte
	A.4.2.3	6	0	
	A.4.2.4	6	0	
	A.4.2.5	6	0	
	A.4.2.6	5	1	
	A.4.2.7	5	1	
	A.4.2.8		D	
<b>A.4.3 Verificar la nueva configuración de software</b>				
	A.4.3.1	6	0	
	A.4.3.2			Jefe soporte, encargado soporte
	A.4.3.3	6	0	
	A.4.3.4	6	0	
	A.4.3.5		D	
<b>A.4.4 Validar la nueva configuración de software</b>				
	A.4.4.1	6	0	
	A.4.4.2			Jefe soporte, encargado soporte
	A.4.4.3	5	1	
	A.4.4.4	5	1	
	A.4.4.5		D	

Tabla 4.19 (A.5.) Tipo de mantenimiento planificable

	Sí	No	D ó M	Respuesta
<b>A.5 TIPO DE MANTENIMIENTO PLANIFICABLE</b>				
<b>A.5.1 Selección de peticiones</b>				
	A.5.1.1	6	0	
	A.5.1.2	Cliente, tipo de problema, modulos afectados, fecha de recepción del incidente		
	A.5.1.3	6	0	
	A.5.1.4	Jefe soporte, encargado soporte		
<b>A.5.2 Analizar peticiones y elegir solución</b>				
	A.5.2.1	6	0	
	A.5.2.2	Correctivo no urgente, perfectivo, preventivo, adaptativo		
	A.5.2.3	5	1	
	A.5.2.4	5	1	
	A.5.2.5		D	
	A.5.2.6	5	1	
	A.5.2.7	4	2	

Tabla 4.20 (A.6.) Intervención y pruebas

	Sí	No	D ó M	Respuesta
<b>A.6 INTERVENCIÓN Y PRUEBAS</b>				
<b>A.6.1 Ejecutar intervención (Correctivo no urgente, perfectivo, preventivo adaptativo)</b>				
	A.6.1.1	5	1	
	A.6.1.2	6	0	
	A.6.1.3	5	1	
	A.6.1.4	5	1	
	A.6.1.5		D	
<b>A.6.2 Ejecutar pruebas unitaras y de integración (Correctivo no urgente, perfectivo, preventivo adaptativo)</b>				
	A.6.2.1	6	0	
	A.6.2.2	6	0	
	A.6.2.3	5	1	
	A.6.2.4	5	1	
	A.6.2.5		D	
	A.6.2.6	6	0	
	A.6.2.7	6	0	
	A.6.2.8	6	0	
	A.6.2.9	Cliente, jefe soporte, encargado soporte		
	A.6.2.10	5	1	
	A.6.2.11	5	1	
	A.6.2.12		D	

	A.6.2.13	5	1	
	A.6.2.14	5	1	
	A.6.2.15			D
<b>A.6.3 Ejecutar paralelamente el software antiguo y el nuevo</b>				
	A.6.3.1	6	0	
	A.6.3.2	6	0	
	A.6.3.3			Desarrollador, jefe soporte, encargado soporte
	A.6.3.4	0	6	
	A.6.3.5	0	6	
	A.6.3.6	0	6	
	A.6.3.7			D
<b>A.6.4 Verificar nueva configuración de software</b>				
	A.6.4.1	6	0	
	A.6.4.2			Cliente, efe soporte, encargado soporte
	A.6.4.3	4	2	
	A.6.4.4	4	2	
	A.6.4.5			D
<b>A.6.5 Validar nueva configuración de software</b>				
	A.6.5.1	6	0	
	A.6.5.2			Cliente
	A.6.5.3	4	2	
	A.6.5.4	4	2	
	A.6.5.5			D
<b>A.6.6 Verificar la administración de la petición de modificación</b>				
	A.6.6.1	6	0	
	A.6.6.2			Jefe soporte, encargado soporte
	A.6.6.3	5	1	
	A.6.6.4	5	1	
	A.6.6.5			D

Tabla 4.21 (A.7.) Seguimiento SprintM

	Sí	No	D ó M	Respuesta
<b>A.7 SEGUIMIENTO SPRINTM</b>				
<b>A.7.1 Reuniones habituales</b>				
	A.7.1.1	6	0	
	A.7.1.2			El avance en la solución, se evalúan nuevos problemas detectados y se analizan
	A.7.1.3	6	0	
	A.7.1.4	5	1	
	A.7.1.5			D
	A.7.1.6	5	1	
	A.7.1.7			Jefe soporte, encargado soporte
<b>A.7.2 Seguimiento de cambios</b>				



A.7.2.1	6	0	
A.7.2.2	5	1	
A.7.2.3	6	0	
A.7.2.4	6	0	
A.7.2.5	0	6	
A.7.2.6	6	0	
A.7.2.7	6	0	
A.7.2.8	6	0	
A.7.2.9	6	0	
A.7.2.10	6	0	
A.7.2.11	6	0	
A.7.2.12			Jefe soporte, encargado soporte
<b>A.7.3 Verificar seguimiento del SprintM</b>			
A.7.3.1	5	1	
A.7.3.2			
A.7.3.3	3	3	
A.7.3.4	2	4	
A.7.3.5			D

Tabla 4.22 (A.8.) Finalización de la intervención

	Sí	No	D ó M	Respuesta
<b>A.8 FINALIZACIÓN DE LA INTERVENCIÓN</b>				
<b>A.8.1 Verificar y validar corrección con el cliente</b>				
A.8.1.1	5	1		
A.8.1.2	5	1		
A.8.1.3	4	2		
A.8.1.4	5	1		
A.8.1.5	5	1		
A.8.1.6			D	
<b>A.8.2 Documentar manual de usuario</b>				
A.8.2.1	5	1		
A.8.2.2	4	2		
A.8.2.3	2	4		
A.8.2.4			D	
A.8.2.5	3	3		
A.8.2.6				Desarrollador
<b>A.8.3 Registro de la intervención</b>				
A.8.3.1	6	0		
A.8.3.2	6	0		
A.8.3.3			D	
<b>A.8.4 Reunión de retrospcción</b>				
A.8.4.1	4	2		
A.8.4.2	2	4		
A.8.4.3	0	6		

	A.8.4.4		D	
<b>A.8.5</b>	<b>Verificar la finalización de la intervención</b>			
	A.8.5.1	5	1	
	A.8.5.2			Jefe soporte, encargado soporte
	A.8.5.3	4	2	
	A.8.5.4	3	3	
	A.8.5.5		D	

Tabla 4.23 (A.9.) Pasar a producción

	Sí	No	D ó M	Respuesta
<b>A.9</b>	<b>PASAR A PRODUCCIÓN</b>			
<b>A.9.1</b>	<b>Pasar a producción</b>			
	A.9.1.1	0	6	
	A.9.1.2	0	6	

Tabla 4.24 (A.10.) Retirada

	Sí	No	D ó M	Respuesta
<b>A.10</b>	<b>RETIRADA</b>			
<b>A.10.1</b>	<b>Desarrollar plan de retirada</b>			
	A.10.1.1	5	1	
	A.10.1.2	3	3	
	A.10.1.3		D	
<b>A.10.2</b>	<b>Notificar futura retirada</b>			
	A.10.2.1	6	0	
	A.10.2.2			Documento, e-mail, oral
	A.10.2.3	6	0	
	A.10.2.4			Jefe de proyecto
<b>A.10.3</b>	<b>Ejecutar en paralelo</b>			
	A.10.3.1	6	0	
	A.10.3.2	6	0	
<b>A.10.4</b>	<b>Notificar la retirada</b>			
	A.10.4.1	6	0	
	A.10.4.2			Documento, e-mail
<b>A.10.5</b>	<b>Almacenar datos del software antiguo</b>			
	A.10.5.1	6	0	
	A.10.5.2			Repositorio de respaldos
	A.10.5.3	5	1	
	A.10.5.4			Desarrollador
<b>A.10.6</b>	<b>Verificar plan de retirada del software</b>			
	A.10.6.1	5	1	

A.10.6.2			Jefe de proyecto
A.10.6.3	4	2	
A.10.6.4	0	6	
A.10.6.5			D

Tabla 4.25 (A.11.) Finalización del servicio

	Sí	No	D ó M	Respuesta
<b>A.11 FINALIZACIÓN DEL SERVICIO</b>				
<b>A.11.1 Entrega del inventario y de la documentación</b>				
	A.11.1.1	6	0	
	A.11.1.2			Producto corregido, documentación, base de conocimiento, fuentes
	A.11.1.3	4	2	
	A.11.1.4	6	0	
	A.11.1.5			Jefe de proyecto, jefe de mantención y soporte
<b>A.11.2 Cese definitivo del servicio</b>				
	A.11.2.1	5	1	
	A.11.2.2	5	1	
	A.11.2.3			D
<b>A.11.3 Verificar finalización del servicio</b>				
	A.11.3.1	6	0	
	A.11.3.2			Jefe soporte, encargado soporte
	A.11.3.3	5	1	
	A.11.3.4	4	2	
	A.11.3.5			D

El resultado obtenido de la aplicación de la encuesta se ve en la tabla 4.26:

Tabla 4.26 Resultados diagnóstico final

Color	Nivel	Resultado	Nivel de cumplimiento
	Realizado	85%	Completamente cumplido
	Gestionado	73%	Ampliamente cumplido
	Establecido	85%	Completamente cumplido

## 4.6. Revisión de la mejora

Tal como se planteó desde un comienzo, se hará una revisión de la mejora de forma extraordinaria, pues el modelo de mejora dispone de esta actividad para el cierre de un ciclo de

mejoras y no para el cierre de una iteración como ocurre con el presente caso, pero en esta ocasión se hará una excepción y se considerará esta actividad en la finalización de la iteración.

A continuación se revisarán las opiniones, experiencias y lecciones aprendidas por el personal que se vio involucrado en la mejora. Así también se analizará el impacto que tuvo el trabajo realizado. Y, además, se evaluará el nivel de compromiso de la alta gerencia con el proceso de mejora.

#### **4.6.1. Retroalimentación**

Para obtener la retroalimentación por parte de los involucrados en la iteración de mejoras se llevaron a cabo reuniones informales, de manera tal que se pudiesen expresar lo más libremente posible sus opiniones y poder conocer su punto de vista del trabajo realizado.

De las reuniones se puede destacar:

- Fue una buena experiencia, a pesar de ser algo nuevo para la gran mayoría de los involucrados.
- Hubieron mejoras claras en el proceso de soporte y mantención. Al menos el proceso es claramente conocido por los involucrados en éste.
- Aún quedan detalles por afinar en la administración de recursos para el proceso, pero eso va de la mano a la implementación de un software que facilite esta tarea.
- El nuevo proceso está abarcando más actividades que el proceso original, pero también es más claro, acotado y específico en las tareas que se deben realizar.
- Se notó la poca experiencia de la persona que llevó a cabo el proceso de mejoras, pues normalmente recurría a material de apoyo para las distintas actividades que realizó.
- Fue una experiencia provechosa que permitió a la organización ver su proceso de soporte y mantenimiento con mirada crítica, para poder reconocer sus falencias y, también, sus aciertos.
- Esta primera experiencia en la aplicación de mejoras de procesos abre nuevas perspectivas para poder mejorar las demás actividades que desarrolla la empresa, pues tener procesos definidos no sólo ayuda a llevar el control sobre las actividades que se ejecutan, sino que también se pueden aprovechar desde el punto de vista comercial, al vender y promocionar un producto software que ha sido desarrollado o es mantenido siguiendo un proceso formal.

#### **4.6.2. Análisis del impacto de la mejora**

De los resultados obtenidos a través de la aplicación del Cuestionario de Mantenimiento Agil\_Mantema [Gajardo *et al.*, 2008] se puede decir que:

- El nuevo proceso incorporó actividades de Agil\_Mantema y que son necesarias para el completo desarrollo del soporte y mantenimiento para la empresa. De esta forma, se pudo cubrir de inicio a fin cada actividad y tarea requerida en este proceso.
- Al mantener la actividad de Ingreso y asignación de incidentes, se obtuvo total transparencia para el cliente y usuario del área de soporte de la empresa. Pues para ellos

el protocolo a seguir para reportar algún incidente es absolutamente el mismo, siendo que las actividades que le siguen al ingreso han sido modificadas.

- La separación de la actividad Asignación de incidentes en otras nuevas actividades más acotadas resultó algo natural para la organización y, en particular, para la encargada del área de soporte y mantenimiento, pues era una actividad saturada de tareas (abarcaba mucho) y la incorporación de las nuevas actividades hacía obvio la su descomposición y acotamiento.
- Se reconoce un aumento en el conocimiento completo del nuevo proceso de soporte y mantenimiento. Esto debido a la capacitación que han recibido los involucrados en el proceso.
- Se ha podido aumentar un 11% (desde un 74% inicial a un 85% final) el cumplimiento de las tareas asociadas al nivel Realizado. Este nivel ha sido el que menor aumento presentó, debido a que ya poseía un alto nivel de cumplimiento y hay tareas que realmente no se realizan en el antiguo ni en el nuevo proceso, por ejemplo: pruebas de regresión. Por lo tanto, las preguntas del cuestionario a esos tópicos no obtuvieron respuestas positivas.
- Se ha podido aumentar en un 21% (desde un 52% inicial a un 73% final) en el nivel de cumplimiento de las tareas asociadas al nivel Gestionado. Esta considerable alza se explica por la mejora e incorporación de los documentos donde se registraban los resultados de las tareas realizadas.
- Se ha podido aumentar en un 31% (desde un 54% inicial a un 85% final) en el nivel de cumplimiento de las tareas asociadas al nivel Establecido. En este nivel se registró la mayor alza del porcentaje de cumplimiento, esto debido a la incorporación de tareas que permiten verificar y validar el trabajo hecho, además de documentar dichas validaciones.

### **4.6.3. Análisis del compromiso**

Este análisis se ha llevado a cabo a través del punto de vista del Responsable de mejoras, es decir, el alumno tesista. Del compromiso de los participantes en el proceso se puede decir:

- De la alta dirección, hubo entusiasmo y compromiso inicial, pero con el pasar del tiempo éste se fue diluyendo durante el tiempo en que no se veían resultados en las mejoras. Así también, fue muy difícil poder coincidir en los horarios con los involucrados de esta área de la empresa, pues se le daba mayor prioridad a otras actividades. A pesar de todo, los resultados obtenidos en el proceso de mejoras dejó conforme a la alta dirección, lo que ha reforzado el compromiso con el proceso de mejoras.
- De los niveles gerenciales, en un comienzo el compromiso fue bajo, pues no tenían confianza en que se pudiera llegar a una mejora real en el proceso estudiado. A medida que el proceso de mejora era llevado a cabo y se interactuaba con ellos el nivel de compromiso fue subiendo, hasta alcanzar un buen nivel de confianza para poder implementar de buena manera las mejoras realizadas.
- Finalmente, del nivel de operaciones, fue donde hubo menor compromiso. En la ejecución de esta iteración de mejoras se trató sólo con una persona de éste nivel (el desarrollador que normalmente se encargaba de resolver los incidentes que se ingresaban). Desde un comienzo hubo poco compromiso, pues veía las mejoras como un aumento en su carga de trabajo. Esta persona fue considerada dentro del proceso de

mejora y fue capacitado, pero al poco tiempo de finalizado el proceso, dejó la empresa. Actualmente el desarrollo de soluciones para los incidentes ingresados recae en cualquiera de los demás desarrolladores, quienes no han recibido la capacitación formal, pero poseen un cierto grado de conocimiento sobre el nuevo proceso.

## 5. Conclusiones

---

A continuación se presentarán las conclusiones del trabajo expuesto en este documento. En primera instancia, se revisará el cumplimiento de los objetivos específicos que permiten dar por cumplido el objetivo general planteado, luego algunos comentarios sobre los elementos utilizados para llevar a cabo la mejora de procesos y finalmente se proponen temas para ser tratados a futuro.

*Objetivo específico:* Investigar los conceptos relacionados con la mejora de procesos de software y la notación de modelado de procesos de software SPEM2.

Este objetivo ha sido altamente cumplido a través de la sección Marco teórico, requerido para el desarrollo de este trabajo y donde además se pudo obtener el conocimiento básico que permitió al alumno tesista desenvolverse y manejar con cierta facilidad los conceptos claves del modelado y mejora de procesos de software, los que permitieron dar a conocer de manera adecuada el trabajo propuesto a la empresa donde finalmente fue llevado a cabo.

*Objetivo específico:* Describir el modelo de referencia de Competisoft y modelar el proceso de mantenimiento con la herramienta de modelado Eclipse Process Framework Composer.

Parte de este objetivo específico fue cumplido al investigar el proyecto Competisoft para incluirlo en el marco teórico. Por otra parte, completando el objetivo planteado, la descripción detallada de Agil\_Mantema se puede encontrar en la sección número 3, junto con el modelado de dicho proceso, el que fue realizado en su totalidad con la herramienta EPFC.

Es importante recalcar que, al describir en detalle el modelo Agil\_Mantema, se encontraron inconsistencias dentro de éste. Especialmente en la congruencia de los nombres de los productos de trabajos utilizados por las actividades y en las entradas y salidas asociadas a sus tareas. Estos detalles fueron evidentes al realizar el modelado del proceso en la herramienta EPFC y, por supuesto, se solucionaron estableciendo una consistencia total del modelo, lo que facilita aún más su comprensión.

Cabe destacar que al recopilar información y guías para realizar el modelado de procesos de software con la herramienta EPFC, se encontraron divergencias en los pasos y la forma en que se utilizan las funcionalidades que permiten generar el modelado. Por esta razón, se optó por tomar la guía más completa y reconocida, la Guía de Uso de SPEM 2 con EPF Composer [Ruiz y Verdugo, 2008]. Y, para utilizar características de la herramienta que no estaban explicadas claramente en dicho documento, se definió un modo de uso estándar, que luego fue utilizado en las siguientes iteraciones de modelado de procesos con la herramienta.

Esto último se dio particularmente al modelar los Capability Process y Delivery Process, en la sección Process. En donde la mayor parte de los ejemplos encontrados no utilizaban los Capability Process y las tareas eran agrupadas en Activities. Por otro lado, la guía de Ruiz y Verdugo no explícita de manera clara la utilización de estos elementos, dejando vacíos conceptuales si se consideran los fundamentos de SPEM2.

De esta manera y superando las dificultades recién planteadas, el segundo objetivo específico formulado para este trabajo fue cumplido.

*Objetivo específico:* Diagnosticar la situación actual del proceso de mantenimiento de una PyME, como caso de estudio.

Primero que todo y fundamental para el desarrollo de gran parte de este trabajo, se encontró una PyME, como caso de estudio, que permitiera realizar el proceso de mejoras.

El diagnóstico del proceso de mantenimiento fue la actividad más dificultosa de las que se realizaron en este trabajo. Para esto se requería tener un conocimiento acabado de Agil\_Mantema (conseguido al llevar a cabo el modelado del proceso) y del modelo de evaluación elegido. Se eligió utilizar la norma ISO/IEC15504 para evaluar el proceso, ya que Agil\_Mantema cuenta con material disponible para seguir esta especificación. Los problemas surgieron al tener poca información sobre la aplicación práctica de la ISO/IEC 15504 dado que no es un modelo libre (se debe pagar para obtener información sobre ella). Por lo que se dedicó más tiempo del planificado para poder encontrar guías que facilitaran su aplicación sin contar con un evaluador certificado.

De esta forma, para llevar a cabo el diagnóstico fue necesario entrevistar a los diversos involucrados dentro del proceso de mantenimiento de la empresa. Lo que demoró bastante dado que no se encontraba ninguna ventana de tiempo libre que coincidiera entre estas personas y el ejecutor del diagnóstico.

Así también se aplicó el Cuestionarios de mantenimiento Agil\_Mantema [Gajardo *et al.*, 2008] para medir el proceso actual según el marco de la ISO/IEC 15504. Este material fue descargado desde la página oficial de Competisoft y, en una primera instancia, aplicado a la empresa. Al realizar el análisis de los resultados obtenidos se pudo observar que las preguntas del cuestionario no eran congruentes con Agil\_Mantema, es decir, el documento descargado estaba desactualizado. Por lo que fue necesario contactar a personas asociadas al proyecto Competisoft para obtener el documento actualizado y aplicarlo nuevamente.

Pasados todos los obstáculos presentados, se pudo cumplir con el tercer objetivo específico de manera exitosa, obteniendo información cualitativa (a través de las entrevistas) y cuantitativa (a través de los resultados del cuestionario) del proceso de mantenimiento de la PyME.

*Objetivo específico:* Llevar a cabo una iteración del ciclo de mejoras para el proceso de mantenimiento, midiendo los resultados obtenidos.

La realización de este objetivo fue la más extensa del trabajo realizado y la que necesitó de mayor esfuerzo y dedicación, dado que el alumno tesista no poseía mayor experiencia en la aplicación de mejoras de procesos y, fundamentalmente, porque la realización de esta actividad es la parte principal de todo este trabajo.

Para el cumplimiento de este objetivo fue fundamental la ayuda y apoyo de las personas de la empresa involucradas en el proceso de soporte y mantenimiento, pues las mejoras, en cierto grado, cambiaron su forma de trabajo, agregando trabajo adicional y eliminando otras



tareas que no tenían mayor sentido. Es decir, si las personas no hubiesen tenido la disposición a mejorar el trabajo que realizaban, el cumplimiento de este objetivo no hubiese sido posible.

También es importante resaltar la guía que entrega Competisoft a través del Proceso de mejoras, lo que facilita enormemente la planeación y aplicación de las mejoras a cualquier proceso de una organización. Al ser una metodología liviana y pensada en su ejecución en pequeñas empresas, entrega resultados velozmente, aún en esta aplicación donde, como ya se mencionó, el ejecutor de las mejoras no tenía mayor experiencia que la investigada en otros casos de estudio encontrados en la literatura.

Al aplicar el Proceso de mejoras de Competisoft al proceso de mantenimiento de TyC Software se pudo observar, al comparar el diagnóstico inicial y final, un aumento considerable en el cumplimiento de los niveles medidos. Un 11% en el nivel 1 Realizado, un 21% en el nivel 2 Gestionado y un 31% en el nivel 3 Establecido.

Dado lo anterior, el último objetivo específico fue cumplido exitosamente y, más aún, pudiendo reportar resultados positivos en la aplicación de la mejora de procesos.

Ya cumplidos los cuatro objetivos específicos formulados, en consecuencia, se puede dar por cumplido el objetivo general.

Con el desarrollo de esta investigación y ejecución de la mejora de procesos a través del caso práctico expuesto, se puede concluir que en toda organización y, en particular, PyMEs que no tienen sus procesos definidos formalmente, o su definición es muy vaga, es posible llevar a cabo mejoras en sus procesos obteniendo resultados positivos. Para ello es totalmente necesario e irrenunciable contar con el apoyo de la alta dirección de la organización y de las personas que verán modificado su actuar diario luego de implementados los cambios que trae consigo la aplicación de las mejoras. Así también es fundamental contar con una guía que permita llevar a cabo este proceso.

Respecto al marco metodológico de Competisoft, fue fundamental al momento de llevar a cabo la mejora de procesos, tanto por la guía entregada para formular, planear y ejecutar las mejoras, como por los modelos de procesos utilizados (en este caso Agil\_Mantema). Aún como primera experiencia en la propuesta y ejecución de mejoras, se pudo obtener un buen resultado al mejorar el área de soporte y mantenimiento de TyC Software. Las mejoras propuestas surgieron naturalmente al poder comparar los elementos del proceso original de la empresa con el propuesto por Competisoft. Esto lleva a pensar que la aplicación completa de Competisoft a una organización puede ser un primer paso para poder institucionalizar la mejora de procesos en vista de la calidad de los productos y servicios prestados.

También se debe reconocer la utilidad que presentó el modelado de procesos de software con la notación SPEM2 y la herramienta Eclipse Process Framework Composer. Ya que sin el modelado, hubiese sido inmensamente más difícil y casi imposible poder comparar el proceso original de la empresa y el propuesto por Competisoft para el mantenimiento de software. Esto facilitó enormemente el reconocimiento de falencias y elementos útiles de los procesos modelados. El modelado de procesos además fue sumamente provechoso al momento de usarlo para poder exponer el nuevo proceso al personal involucrado. Los elementos que presta SPEM2 y EPFC para esto son sumamente eficaces.

Por otra parte, el modelado de procesos resultó ser bastante engorroso y trabajoso, a pesar de contar con una guía detallada para ello. Así también, se debe mencionar que la herramienta de modelado EPFC dio buenos resultados, aunque la versión con que se trabajó (1.5.1.3) no tenía soporte para el sistema operativo Windows 7, por lo que se debió trabajar obligadamente en un equipo antiguo con sistema operativo Windows XP. Además, se debe reconocer que existe muy poca información y experiencias sobre el modelado con esta herramienta, lo que dificulta su utilización y propagación de su uso. Aunque en la investigación de las demás herramientas, tampoco se pudo encontrar mayor información y documentación sobre éstas, por lo que se cree que el uso de estas herramientas particularizadas para el modelado con SPEM2 no son reconocidas, ni ampliamente utilizadas por la comunidad.

Como trabajos futuros relacionados a los temas tratados se pueden tratar temas como:

- Utilizar modelos de mejoras como CMMI o la ISO/IEC 15504, para adaptarlos al contexto de las pequeñas y medianas empresas.
- Comparar resultados obtenidos en la aplicación de mejoras de procesos entre Competisoft y CMMI o ISO/IEC 15504.
- Comparar los modelos de Competisoft con los de CMMI.
- Llevar a cabo otras mejoras a otros procesos de una empresa. Pudiendo llegar a aplicar todo el marco metodológico de Competisoft para medir su impacto en la organización al ejecutarlo en plenitud.
- Aplicar la mejora de procesos de Competisoft medir sus resultados y, desde ese punto de partida, poder diagnosticar el estado de los procesos de la organización con CMMI. Para tener claridad, a través de la experiencia, en la cercanía o cuánto se aproxima Competisoft con los niveles de CMMI.
- Investigar y realizar el modelado de procesos con otras herramientas del mercado.
- Investigar y realizar el modelado de procesos de software con otras notaciones existentes.

## Referencias

---

[ACTI, 2010] Asociación Chilena de Empresas de Tecnologías de Información A.G. (2010). Presentación Estudio Centro Servicios Globales TICs 2010.

[Acuña y Ferré, 2001] Acuña, S., y Ferré, X. (2001). Software Process Modelling . In Proceedings of the 5th. World Multiconference on Systemics, Cybernetics and Informatics.

[Anacleto *et al.*, 2004] Anacleto, A., Wangenheim, C., Salviano, C., y Savi, R. (2004). A Method for Process Assessment in Small Software Companies. 4th International SPICE Conference on Process Assessment and Improvement, (págs. 69-76). Portugal.

[Aspray *et al.*, 2006] Aspray, W., Mayadas, F., y Vardi, M. Y. (2006). Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force. Association for Computing Machinery.

[Boehm, 2006] Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. In Proc. ICSE'06 , 12-29.

[CETIUC, 2007] Centro de Estudios de Tecnologías de Información, Pontificia Universidad Católica de Chile. (2007). Estudio Nacional sobre Tecnologías de Información. Pontificia Universidad Católica de Chile, Centro de Estudios de Tecnologías de Información.

[CETIUC, 2010] Centro de Estudios de Tecnologías de Información, Pontificia Universidad Católica de Chile. (2010). Estudio Nacional sobre Tecnologías de Información. Pontificia Universidad Católica de Chile, Centro de Estudios de Tecnologías de Información.

[Conradi *et al.*, 1994] Conradi, R., Fernström, C., y Fuggetta, A. (1994). Concepts for Evolving Software Processes. Research Studies Press 9-31.

[Curtis *et al.*, 1992] Curtis, B., Kellner, M., y Over, J. (1992). Process Modelling. Communications of the ACM 35.

[Fowler y Rifkin, 1990] Fowler, P., y Rifkin, S. (1990). Software Engineering Process Group Guide. Software Engineering Institute.

[Fuggetta, 2000] Fuggetta, A. (2000). Software Process: A Roadmap. Proceedings of the Conference on The Future of Software Engineering.

[Gajardo *et al.*, 2008] Fajardo, R., Rodríguez, O., Pino, F., Ruiz, F. y Piattini, M. (2008). Cuestionarios de mantenimiento Agil\_Mantema. Documentos Competisoft.

[GECHS, 2008] Asociación Gremial de Empresas Chilenas Desarrolladoras de Software. (2008). Sexto Diagnóstico de la Industria Nacional de Software y Servicios.

[Hollingsworth, 1995] Hollingsworth, D. (1995). The Workflow Reference Model. Workflow Management Coalition.

[Hurtado *et al.*, 2008] Hurtado, J., Pino, F., Pardo, C., y Fernández, L. (2008). Agile SPI: Software Process Agile Improvement, A Colombia Approach to Software Process Improvement in Small Software Organizations. Software Process Improvement for Small and Medium Enterprises: Techniques and Cases Studies. USA, Idea Group Inc. , 177-192.

[IDS Scheer, 2005] IDS Scheer. (2005). Methods Aris 7.0.

[jBoss Company, 2006] jBoss Company. (2006). JBoss jBPM 3.1 Guía Práctica de Workflow y BPM. jBoss division of Red Hat Company.

[Lonchamp, 1993] Lonchamp, J. (1993). A Structured Conceptual and Terminological Framework for Software Process Engineering. Proceedings of the Second International Conference on Software Process.

[Martinez San Germán, 2003] Martinez San Germán, J. (2003). Métodos de Modelado IDEF0 e IDEF3 y Uso Básico. Comisión Federal de Electricidad.

[McFeeley, 1996] McFeeley, B. (1996). IDEAL: A User's Guide for Software Process Improvement. Software Engineering Institute, Carnegie Mellon University.

[NYCE, 2005] Normalización y Certificación Electronica A.C. (2005). Modelo de Procesos para la Industria de Software - MoproSoft - Versión 1.3. Organismo Nacional de Normalización y Evaluación de la Conformidad.

[Oktaba, 2004] Oktaba, H. (2004). Método de Evaluación de procesos para la industria del software - EvalProSoft - Versión 1.1, Marzo 2004. Ciudad de México: Organismo Nacional de Normalización y Evaluación de Conformidad - NYCE.

[Oktaba *et al.*, 2008] Oktaba, H., Piattini, M., Pino, F., Orozco, M. J., y Alquicira, C. (2008). COMPETISOFT Mejora de Procesos Software para Pequeñas y Medianas Empresas y Proyectos. RA-MA.

[OMG, 2006] Object Management Group. (2006). Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0. Object Management Group.

[OMG, 2008] Object Management Group. (2008). Software & Systems Process Engineering Meta-Model Specification version 2.0. Object Management Group.

[OMG, 2010] Object Management Group. (2010). OMG Unified Modeling Language (OMG UML), Infrastructure Version 2.3. Object Management Group.

[Pérez, 2007] Pérez, J. D. (2007). Notaciones y lenguajes de procesos. Una visión global. Sevilla.

[Pino *et al.*, 2008] Pino, F., Ruiz, F., y Salas, S. (2008). *Agil\_Mantema: una metodología de mantenimiento de software para pequeñas empresas*. Ciudad Real, España: CYTED.

[Polo *et al.*, 1999] Polo, M., Piattini, M., Ruiz, F., y Calero, C. (1999). MANTEMA: A Software Maintenance Methodology Based on the ISO/IEC 12207 Standard. Proceedings of the 4th IEEE International Symposium and Forum on Software Engineering Standards (págs. 76-81). Curitiba, Brazil: IEEE Computer Society.

[Rothenberg, 1989] Rothenberg, J. (1989). The nature of modeling.

[Ruiz y Verdugo, 2008] Ruiz, F. y Verdugo, J. (2008). Guía de Uso de SPEM 2 con EPF Composer. Universidad de Castilla - La Mancha, Escuela Superior de Informática, Departamento de Tecnologías y Sistemas de Información, Grupo Alarcos.

[SWEBOK, 2004] Software Engineering Body of Knowledge. (2004). Guide to the Software Engineering Body of Knowledge. IEEE Computer Society.

[Takeuchi y Nonaka, 1986] Takeuchi, H., y Nonaka, I. (1986). The New New Product Development Game. Harvard Business Review.

[Weber *et al.*, 2005] Weber, K., Araújo, E., Rocha, A., Machado, C., Scalet, D., & Salviano, C. (2005). Brazilian Software Process Reference Model and Assessment Method. Computer and Information Sciences (págs. 402-411). Heidelberg: Springer Berlin.

[Wohed *et al.*, 2004] Wohed, P., van der Aalst, W. M., Dumas, M., ter Hofstede, A. H., y Russell, N. (2004). Pattern-based Analysis of UML Activity Diagrams. Eindhoven University of Technology, Department of Technology Management.

[Wohed *et al.*, 2005] Wohed, P., van der Aalst, W. M., Dumas, M., ter Hofstede, A. H., y Russell, N. (2005). Pattern-based analysis of bpmn. University of Technology, Department of Technology Management Eindhoven.

## Anexos

### Cuestionario OPE3:

**Proceso:** Mantenimiento

**Categoría:** Operación (OPE)

**Propósito:** El proceso de mantenimiento tiene como propósito definir una guía explícita para realizar las modificaciones solicitadas en un producto software detallando qué debe realizarse, cuándo, cómo y por quién. Es decir, busca guiar paso a paso el proceso de mantenimiento del software para pequeñas organizaciones.

		Posibles respuestas	¿A dónde ir?
<b>A.1.</b>	<b>Planificación del Proceso</b>		
<b>A.1.1.</b>	<b>Asignar Responsables. Se asignan los roles del proceso de mantenimiento a las personas involucradas en el mismo. El Propietario del Producto, el Gestor de Peticiones, el Responsable de Mantenimiento, y el Equipo de mantenimiento están claramente establecidos e identificados en la organización.</b>		
A.1.1.1.	¿Existe una persona que acepta o rechaza las peticiones de modificación y decide el tipo de mantenimiento que debe aplicarse?	Si __ No __	A.1.1.2. A.1.1.3.
A.1.1.2.	¿Quién ocupa ese rol dentro del área?	Roles	A.1.1.3.
A.1.1.3.	¿Existe una persona responsable de preparar la etapa de mantenimiento y que interactúe con el cliente y el equipo de trabajo?	Si __ No __	A.1.1.4. A.1.1.5.
A.1.1.4.	¿Quién ocupa ese rol dentro del área?	Roles	A.1.1.5.
A.1.1.5.	¿Existen personas que se dediquen exclusivamente al mantenimiento del software?	Si __ No __	A.1.1.6.
A.1.1.6.	¿Existe un plan o estrategia de distribución del equipo de trabajo al llevar a cabo una de las actividades de mantenimiento?	Si __ No __	A.1.1.7. A.1.2.1.
A.1.1.7.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.1.8.
A.1.1.8.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.1.2.1.
<b>A.1.2.</b>	<b>Adquirir Conocimiento de la Aplicación. El Equipo de mantenimiento obtiene la información del software que se va a mantener.</b>		
A.1.2.1.	¿De qué forma el equipo de trabajo obtiene información del software a mantener?	Documentación/ Código Fuente/ Referencias/ Entrevistas/ Observación/	A.1.2.2.
A.1.2.2.	¿Existe un responsable encargado de administrar la documentación del software a mantener?	Si __ No __	A.1.2.3. A.1.2.4.
A.1.2.3.	¿Quién ocupa ese rol dentro del área?	Roles	A.1.2.4.
A.1.2.4.	¿Existe un procedimiento que permita analizar de acuerdo a la documentación el estado del software a mantener?	Si __ No __	A.1.2.5. A.1.3.1.
A.1.2.5.	¿Dónde queda registrado?	Texto	A.1.2.6.

A.1.2.6.	¿Esta documentación es facilitada al Cliente para que determine si se ha adquirido un conocimiento adecuado del software?	Si __ No __	A.1.2.7.
A.1.2.7.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.2.8.
A.1.2.8.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.1.3.1.
<b>A.1.3.</b>	<b>Preparar entornos de Pruebas.</b>		
A.1.3.1.	¿Existe un procedimiento definido para preparar el entorno de pruebas?	Si __ No __	A.1.3.2. A.1.3.3.
A.1.3.2.	¿Está definido formalmente en un documento (manual/electrónico)?	Si __ No __	A.1.3.3.
A.1.3.3.	¿De qué manera se prepara el entorno de pruebas?	Copias de Software/ Preparación de BD y archivos/	A.1.3.4.
A.1.3.4.	¿Existe un responsable de preparar el ambiente de pruebas?	Si __ No __	A.1.3.5. A.1.4.1.
A.1.3.5.	¿Quién ocupa ese rol dentro del área?	Roles	
<b>A.1.4.</b>	<b>Definir procedimientos de petición de modificación.</b>		
A.1.4.1.	¿Existe un responsable de entregar/recibir peticiones de modificación?	Si __ No __	A.1.4.2. A.1.4.3.
A.1.4.2.	¿Quién ocupa ese rol dentro del área?	Roles	A.1.4.3.
A.1.4.3.	¿El Cliente tiene conocimiento de quién es el responsable de las peticiones de modificación?	Si __ No __	A.1.4.4.
A.1.4.4.	¿Existe un documento de Petición de Modificación?	Si __ No __	A.1.4.5. A.1.5.1.
A.1.4.5.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.4.6
A.1.4.6.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.1.5.1.
<b>A.1.5.</b>	<b>Verificar la Planificación del proceso, Copia del producto Software, Petición de Modificación.</b>		
A.1.5.1.	¿Se verifica la Planificación del proceso?	Si __ No __	A.1.5.2. A.1.5.4.
A.1.5.2.	¿Quién la verifica?	Roles	A.1.5.3.
A.1.5.3.	¿Queda documentado?	Si __ No __	A.1.5.4. A.1.5.6.
A.1.5.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.5.5.
A.1.5.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.1.5.6.
A.1.5.6.	¿Se verifica la Copia del producto Software?	Si __ No __	A.1.5.7. A.1.5.11.
A.1.5.7.	¿Quién la verifica?	Roles	A.1.5.8.
A.1.5.8.	¿Queda documentado?	Si __ No __	A.1.5.9. A.1.5.11.
A.1.5.9.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.5.10.
A.1.5.10	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.1.5.11.
A.1.5.11	¿Se verifica la Petición de Modificación?	Si __ No __	A.1.5.12. A.1.6.1.
A.1.5.12	¿Quién la verifica?	Roles	A.1.5.13.
A.1.5.13	¿Queda documentado?	Si __ No __	A.1.5.14. A.1.6.1.
A.1.5.14	¿Responde a una plantilla predefinida?	Si __ No __	A.1.5.15.
A.1.5.15	¿En qué medio queda documentado manual/digital)?	Manual/	A.1.6.1.

		Digital/	
<b>A.1.6</b>	<b>Validar la Petición de Modificación.</b>		
A.1.6.1.	¿Se valida la Petición de Modificación?	Si __ No __	A.1.6.2. A.2.1.1.
A.1.6.2.	¿Quién la valida?	Roles	A.1.6.3.
A.1.6.3.	¿Queda documentado?	Si __ No __	A.1.6.4. A.2.1.1.
A.1.6.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.1.6.5.
A.1.6.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.2.1.1.
<b>A.2.</b>	<b>Atención de la Petición de Modificación</b>		
<b>A.2.1</b>	<b>Recibir Petición de Modificación.</b>		
A.2.1.1.	¿Existe un responsable de recibir peticiones de modificación?	Si __ No __	A.2.1.2. A.2.1.3.
A.2.1.2.	¿Quién ocupa ese rol dentro del área?	Roles	A.2.1.3.
A.2.1.3.	¿Existe un procedimiento para recibir e informar al equipo de mantenimiento sobre la petición de modificación?	Si __ No __	A.2.1.4.
A.2.1.4.	¿Se lleva a cabo un registro y control de peticiones (identificador/ priorización)?	Si __ No __ Texto	A.2.1.5. A.2.2.1.
A.2.1.5.	¿Responde a una plantilla predefinida?	Si __ No __	A.2.1.6.
A.2.1.6.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.2.2.1.
<b>A.2.2</b>	<b>Decidir el tipo de mantenimiento.</b>		
A.2.2.1.	En caso de rechazar una petición, ¿se crea un documento donde se justifique la razón del rechazo?	Si __ No __	A.2.2.2.
A.2.2.2.	¿De qué forma se da a conocer al cliente el rechazo de la petición?	Documento/ Correo electrónico/ Memorándum/ Oral/ Otra/	A.2.2.3.
A.2.2.3.	¿Se cuenta con un Product Backlog (Registro de Peticiones) donde se vayan guardando las peticiones en forma ordenada?	Si __ No __	A.2.2.4.
A.2.2.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.2.2.5.
A.2.2.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.2.2.6.
A.2.2.6.	¿Cómo se decide el tipo de mantenimiento a aplicar en base a la petición de modificación correspondiente?	Texto	A.2.2.7.
A.2.2.7.	¿Existe una priorización de peticiones de modificación?	Si __ No __	A.3.1.1.
<b>A.2.3.</b>	<b>Verificar la Administración de la petición de modificación.</b>		
A.2.3.1.	¿Se verifica la Administración de la Petición de Modificación?	Si __ No __	A.2.3.2. A.3.1.1.
A.2.3.2.	¿Quién la verifica?	Roles	A.2.3.3.
A.2.3.3.	¿Queda documentado?	Si __ No __	A.2.3.4. A.3.1.1.
A.2.3.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.2.3.5.
A.2.3.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.3.1.1.
	<b>SI EL TIPO DE MANTENIMIENTO ES NO PLANIFICABLE</b>		



<b>A.3.</b>	<b>Análisis del Error</b>		
<b>A.3.1.</b>	<b>Investigar y Analizar causas.</b>		
A.3.1.1.	¿El equipo de mantenimiento realiza el análisis de la petición de modificación y verificación del problema en colaboración con el usuario que realizó la petición?	Si __ No __	A.3.1.2.
A.3.1.2.	¿El equipo de mantenimiento realiza un estudio sobre las diferentes alternativas con que cuenta para implementar la modificación?	Si __ No __	A.3.1.3.
A.3.1.3.	¿Se construye una lista de elementos de software a corregir (módulos, rutinas, documentos, etc.)?	Si __ No __	A.3.1.4. A.4.1.1.
A.3.1.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.3.1.5.
A.3.1.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.4.1.1.
<b>A.4.</b>	<b>Intervención correctiva urgente</b>		
<b>A.4.1.</b>	<b>Realizar acciones correctivas.</b>		
A.4.1.1.	¿Existe un plan para llevar a cabo acciones correctivas?	Si __ No __	A.4.1.1. A.4.1.4.
A.4.1.2.	¿Existe un responsable que realiza estas acciones?	Si __ No __	A.4.1.3. A.4.1.4.
A.4.1.3.	¿Quién ocupa ese rol dentro del área?	Roles	A.4.1.4.
A.4.1.4.	¿Se identifican las rutinas y bases de datos afectadas por la intervención?	Si __ No __	A.4.1.5. A.4.2.1.
A.4.1.5.	¿Esto último queda registrado en un documento?	Si __ No __	A.4.1.6. A.4.2.1.
A.4.1.6.	¿Responde a una plantilla predefinida?	Si __ No __	A.4.1.7.
A.4.1.7.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.4.2.1.
<b>A.4.2.</b>	<b>Ejecutar pruebas unitarias.</b>		
A.4.2.1.	¿Existe un responsable que realiza estas acciones?	Si __ No __	A.4.2.2. A.4.2.3.
A.4.2.2.	¿Quién ocupa ese rol dentro del área?	Roles	A.4.2.3.
A.4.2.3.	¿Existe un procedimiento para comprobar la corrección de todos los cambios realizados?	Si __ No __	A.4.2.4. A.4.2.5.
A.4.2.4.	¿Se encuentra definido formalmente en un documento?	Si __ No __	A.4.2.5.
A.4.2.5.	¿Existe un plan de pruebas unitarias?	Si __ No __	A.4.2.6. A.4.3.1.
A.4.2.6.	¿Se documentan las pruebas unitarias?	Si __ No __	A.4.2.7.
A.4.2.7.	¿Responde a una plantilla predefinida?	Si __ No __	A.4.2.8.
A.4.2.8.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.4.3.1.
<b>A.4.3.</b>	<b>Verificar la Nueva configuración de software.</b>		
A.4.3.1.	¿Se verifica la Nueva configuración del software?	Si __ No __	A.4.3.2. A.4.4.1.
A.4.3.2.	¿Quién la verifica?	Roles	A.4.3.3.
A.4.3.3.	¿Queda documentada?	Si __ No __	A.4.3.4. A.4.4.1.
A.4.3.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.4.3.5.
A.4.3.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.4.4.1.
<b>A.4.4.</b>	<b>Validar la Nueva configuración de software.</b>		
A.4.4.1.	¿Se valida la Nueva configuración del software?	Si __ No __	A.4.4.2. A.5.1.1.

A.4.4.2.	¿Quién la valida?	Roles	A.4.4.3.
A.4.4.3.	¿Queda documentada?	Si __ No __	A.4.4.4. A.5.1.1.
A.4.4.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.4.4.5.
A.4.4.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.5.1.1.
<b>SI EL TIPO DE MANTENIMIENTO ES PLANIFICABLE</b>			
<b>A.5. Selección y Análisis de las peticiones.</b>			
<b>A.5.1 Selección de las peticiones.</b>			
A.5.1.1.	¿Se selecciona un conjunto de peticiones a abordar en la intervención?	Si __ No __	A.5.1.2. A.5.2.1.
A.5.1.2.	¿Cómo se selecciona y en base a qué criterio?	Texto	A.5.1.3.
A.5.1.3.	¿Existe un responsable que realiza esta acción?	Si __ No __	A.5.1.4. A.5.2.1.
A.5.1.4.	¿Quién ocupa ese rol dentro del área?	Roles	A.5.2.1.
<b>A.5.2. Analizar peticiones y elegir Solución.</b>			
A.5.2.1.	¿Se identifica previamente el tipo de mantenimiento a realizar?	Si __ No __	A.5.2.2.
A.5.2.2.	¿Cuáles son los tipos de mantenimiento que realiza el equipo de mantenimiento?	Correctivo no urgente/ Perfectivo/ Preventivo/ Adaptativo/	A.5.2.3.
A.5.2.3.	¿Existe el documento de Diagnóstico del Error y Posibles Soluciones para cada tipo de mantenimiento?	Si __ No __	A.5.2.4. A.6.1.1.
A.5.2.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.5.2.5.
A.5.2.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.5.2.6.
A.5.2.6.	¿Se analiza en equipo cada una de las posibles soluciones y se elige una de ellas para su implementación?	Si __ No __	A.5.2.7.
A.5.2.7.	¿Se registra la solución escogida en el documento de Diagnóstico del Error y Posibles Soluciones para cada tipo de mantenimiento?	Si __ No __	A.6.1.1.
<b>A.6. Intervención y pruebas</b>			
<b>A.6.1. Ejecutar intervención (Correctivo no Urgente/Perfectivo/Preventivo/Adaptativo)</b>			
A.6.1.1.	Para ejecutar las acciones necesarias para servir la petición de modificación conforme a la solución acordada ¿Se guían por el documento de Diagnóstico del Error y Posibles Soluciones para cada tipo de mantenimiento?	Si __ No __	A.6.1.2.
A.6.1.2.	¿Se identifican las rutinas y bases de datos afectadas por la intervención?	Si __ No __	A.6.1.3. A.6.2.1.
A.6.1.3.	¿Esto queda registrado en un documento?	Si __ No __	A.6.1.4. A.6.2.1.
A.6.1.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.1.5.
A.6.1.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.2.1.
<b>A.6.2. Ejecutar Pruebas unitarias y de integración (Correctivo no Urgente/Perfectivo/Preventivo/Adaptativo)</b>			
A.6.2.1.	¿Se realizan pruebas unitarias sobre el producto de software intervenido?	Si __ No __	A.6.2.2.
A.6.2.2.	¿Se realizan pruebas de integración sobre el producto de software intervenido?	Si __ No __	A.6.2.3.
A.6.2.3.	¿Existe un procedimiento establecido para llevar a cabo las pruebas unitarias y de integración?	Si __ No __	A.6.2.4. A.6.2.6.
A.6.2.4.	¿Responde a una plantilla predefinida?	Si __	A.6.2.5.

		No __	
A.6.2.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.2.6.
A.6.2.6.	¿Se comprueba que la petición queda servida completamente luego de llevar a cabo las pruebas?	Si __ No __	A.6.2.7.
A.6.2.7.	¿Se comprueba que los elementos del software luego de terminadas las pruebas funcionan correctamente en forma conjunta?	Si __ No __	A.6.2.8.
A.6.2.8.	¿Existe un responsable que realiza estas acciones?	Si __ No __	A.6.2.9. A.6.2.10.
A.6.2.9.	¿Quién ocupa ese rol dentro del área?	Roles	A.6.2.10.
A.6.2.10.	¿Los resultados de las pruebas unitarias quedan registrados en un documento?	Si __ No __	A.6.2.11. A.6.2.13.
A.6.2.11.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.2.12.
A.6.2.12.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.2.13.
A.6.2.13.	¿Los resultados de las pruebas de integración quedan registrados en un documento?	Si __ No __	A.6.2.14. A.6.3.1.
A.6.2.14.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.2.15.
A.6.2.15.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.3.1.
<b>A.6.3.</b>	<b>Ejecutar paralelamente el software antiguo y nuevo</b>		
A.6.3.1.	¿Se ejecutan acciones reales en el software antiguo y en el nuevo para detectar y prevenir errores de proceso?	Si __ No __	A.6.3.2. A.6.3.4.
A.6.3.2.	¿Existe un responsable que realiza estas acciones?	Si __ No __	A.6.3.3. A.6.3.4.
A.6.3.3.	¿Quién ocupa ese rol dentro del área?	Roles	A.6.3.4.
A.6.3.4.	¿Realizan pruebas de regresión?	Si __ No __	A.6.3.5. A.6.4.1.
A.6.3.5.	¿Existe un plan de pruebas de regresión?	Si __ No __	A.6.3.6. A.6.4.1.
A.6.3.6.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.3.7.
A.6.3.7.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.4.1.
<b>A.6.4.</b>	<b>Verificar la Nueva configuración de software.</b>		
A.6.4.1.	¿Se verifica la Nueva configuración del software?	Si __ No __	A.6.4.2. A.6.5.1.
A.6.4.2.	¿Quién la verifica?	Roles	A.6.4.3.
A.6.4.3.	¿Queda documentado?	Si __ No __	A.6.4.4. A.6.5.1.
A.6.4.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.4.5.
A.6.4.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.5.1.
<b>A.6.5.</b>	<b>Validar la Nueva configuración de software.</b>		
A.6.5.1.	¿Se valida la Nueva configuración del software?	Si __ No __	A.6.5.2. A.6.6.1.
A.6.5.2.	¿Quién la valida?	Roles	A.6.5.3.
A.6.5.3.	¿Queda documentado?	Si __ No __	A.6.5.4. A.6.6.1.
A.6.5.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.5.5.
A.6.5.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.6.6.1.

<b>A.6.6.</b>	<b>Verificar la Administración de la petición de modificación.</b>		
A.6.6.1.	¿Se verifica la Administración de la petición de modificación?	Si __ No __	A.6.6.2. A.7.1.1.
A.6.6.2.	¿Quién la verifica?	Roles	A.6.6.3.
A.6.6.3.	¿Queda documentado?	Si __ No __	A.6.6.4. A.7.1.1.
A.6.6.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.6.6.5.
A.6.6.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.7.1.1.
<b>A.7.</b>	<b>Seguimiento del SprintM</b>		
<b>A.7.1.</b>	<b>Reuniones Habituales</b>		
A.7.1.1.	¿Se realizan reuniones periódicas en donde se reúna el equipo de mantenimiento?	Si __ No __	A.7.1.2. A.7.2.1.
A.7.1.2.	¿Qué temas se discuten en las reuniones?	Texto	A.7.1.3.
A.7.1.3.	¿Quedan registrados en minutas los objetivos y acuerdos para las próximas reuniones?	Si __ No __	A.7.1.4. A.7.1.6.
A.7.1.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.7.1.5.
A.7.1.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.7.1.6.
A.7.1.6.	¿Existe un responsable de fijar y coordinar las reuniones?	Si __ No __	A.7.1.7. A.7.2.1.
A.7.1.7.	¿Quién ocupa ese rol?	Roles	A.7.2.1.
<b>A.7.2.</b>	<b>Seguimiento de los cambios</b>		
A.7.2.1.	¿Se realiza un control de los cambios producidos en la ejecución del mantenimiento?	Si __ No __	A.7.2.2.
A.7.2.2.	¿Se realiza una traza de los cambios que la petición ha provocado a lo largo de los procesos de desarrollo implicados?	Si __ No __	A.7.2.3.
A.7.2.3.	¿Se verifica que se han realizado exitosamente las pruebas unitarias de cada uno de los componentes a modificar?	Si __ No __	A.7.2.4.
A.7.2.4.	¿Se verifica que se han realizado exitosamente las pruebas integración de los componentes a modificar?	Si __ No __	A.7.2.5.
A.7.2.5.	¿Se verifica que se han realizado exitosamente las pruebas de regresión (si es que se hacen)?	Si __ No __	A.7.2.6.
A.7.2.6.	¿Se comprueba que los cambios corresponden al documento de petición de modificación?	Si __ No __	A.7.2.7. A.7.2.8.
A.7.2.7.	En el caso que los cambios no corresponden al documento de petición ¿Esto queda registrado en un documento?	Si __ No __	A.7.2.8. A.7.2.9.
A.7.2.8.	¿Se hace llegar una copia del documento al cliente?	Si __ No __	A.7.2.9.
A.7.2.9.	¿Se comunica oportunamente al cliente?	Si __ No __	A.7.2.10.
A.7.2.10.	¿Se lleva a cabo un control de los distintos desarrollos en paralelo sobre un mismo componente?	Si __ No __	A.7.2.11. A.8.1.1.
A.7.2.11.	¿Existe un responsable que realiza estas acciones?	Si __ No __	A.7.2.12.
A.7.2.12.	¿Quién ocupa ese rol dentro del área?	Roles	A.7.3.1.
<b>A.7.3.</b>	<b>Verificar el Seguimiento del SprintM.</b>		
A.7.3.1.	¿Se verifica el Seguimiento del SprintM?	Si __ No __	A.7.3.2. A.8.1.1.
A.7.3.2.	¿Quién lo verifica?	Roles	A.7.3.3.

A.7.3.3.	¿Queda documentado?	Si __ No __	A.7.3.4. A.8.1.1.
A.7.3.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.7.3.5.
A.7.3.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.8.1.1.
<b>A.8.</b>	<b>Finalización de la Intervención</b>		
<b>A.8.1.</b>	<b>Verificar y validar corrección con el Cliente</b>		
A.8.1.1.	¿Se realiza una reunión entre el equipo de mantenimiento y el cliente para comprobar que el producto intervenido funciona correctamente?	Si __ No __	A.8.1.2. A.8.2.1.
A.8.1.2.	¿Se recogen impresiones, sugerencias, mejoras sobre el producto de parte del cliente?	Si __ No __	A.8.1.3.
A.8.1.3.	¿Se evalúan posibles cambios en el Product Backlog (Registro de Peticiones)?	Si __ No __	A.8.1.4.
A.8.1.4.	¿La información recogida de esta reunión, queda registrada en un documento?	Si __ No __	A.8.2.1.
A.8.1.5.	¿Responde a una plantilla predefinida?	Si __ No __	A.8.1.6.
A.8.1.6.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.8.2.1.
<b>A.8.2.</b>	<b>Documentar manual de Usuario</b>		
A.8.2.1.	¿Se documenta un nuevo manual de usuario luego de efectuar el mantenimiento sobre el software?	Si __ No __	A.8.2.3. A.8.2.2.
A.8.2.2.	¿Se re-documenta el manual de usuario existente de acuerdo a las modificaciones realizadas durante el mantenimiento?	Si __ No __	A.8.2.3. A.8.3.1.
A.8.2.3.	¿Responde a una plantilla predefinida?	Si __ No __	A.8.2.4. A.8.2.5.
A.8.2.4.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.8.2.5.
A.8.2.5.	¿Existe un responsable de realizar estas acciones?	Si __ No __	A.8.2.6. A.8.3.1.
A.8.2.6.	¿Quién ocupa ese rol?	Roles	A.8.3.1.
<b>A.8.3.</b>	<b>Registro de la intervención</b>		
A.8.3.1.	¿La intervención queda registrada según los procedimientos de la organización?	Si __ No __	A.8.3.2. A.8.4.1.
A.8.3.2.	¿Responde a una plantilla predefinida?	Si __ No __	A.8.3.3.
A.8.3.3.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.8.4.1.
<b>A.8.4.</b>	<b>Reunión de retrospectiva</b>		
A.8.4.1.	¿Se realiza una reunión de retrospectiva donde se extraen las mejores prácticas de la última intervención?	Si __ No __	A.8.4.2. A.8.5.1.
A.8.4.2.	¿Existe un documento donde se registren las mejores prácticas?	Si __ No __	A.8.4.3. A.8.5.1.
A.8.4.3.	¿Responde a una plantilla predefinida?	Si __ No __	A.8.4.4.
A.8.4.4.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.8.5.1.
<b>A.8.5.</b>	<b>Verificar la Finalización de la intervención</b>		
A.8.5.1.	¿Se verifica la Finalización de la intervención?	Si __ No __	A.8.5.2. A.9.1.1.
A.8.5.2.	¿Quién la verifica?	Roles	A.8.5.3.
A.8.5.3.	¿Queda documentado?	Si __ No __	A.8.5.4. A.9.1.1.
A.8.5.4.	¿Responde a una plantilla predefinida?	Si __ No __	A.8.5.5.

A.8.5.5.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.9.1.1.
<b>A.9</b>	<b>Pasar a Producción.</b>		
<b>A.9.1.</b>	<b>Pasar a Producción.</b>		
A.9.1.1.	¿Se establece una interfaz (comunicación) con el Proceso de Gestión de Configuración para garantizar la corrección de la tarea?	Si __ No __	A.9.1.2.
A.9.1.2.	¿Se establece una interfaz (comunicación) con el Proceso de Infraestructura por si es necesario cambiar o agregar máquinas, software necesario, herramientas, etc.?	Si __ No __	A.10.1.1.
<b>A.10.</b>	<b>Retirada</b>		
<b>A.10.1.</b>	<b>Desarrollar plan de retirada</b>		
A.10.1.1.	¿Se redacta un documento en el que se describa la retirada del software antiguo?	Si __ No __	A.10.1.2. A.10.2.1.
A.10.1.2.	¿Responde a una plantilla predefinida?	Si __ No __	A.10.1.3.
A.10.1.3.	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.10.2.1.
<b>A.10.2.</b>	<b>Notificar Futura retirada</b>		
A.10.2.1.	¿El equipo de mantenimiento notifica al cliente el momento en que se ejecutará la retirada del software antiguo?	Si __ No __	A.10.2.2.
A.10.2.2.	¿Cómo se comunica la retirada al Cliente?	Documento/ Correo electrónico/ Memorándum/ Oral/ Otra	A.10.2.3.
A.10.2.3.	¿Existe un responsable de realizar estas acciones?	Si __ No __	A.10.2.4. A.10.3.1.
A.10.2.4.	¿Quién ocupa ese rol?	Roles	A.10.3.1.
<b>A.10.3.</b>	<b>Ejecutar Paralelo</b>		
A.10.3.1.	¿Se coordina una sesión en que participe el usuario y el equipo de mantenimiento para realizar operaciones sobre el software que se va a retirar y el software nuevo?	Si __ No __	A.10.3.2. A.10.4.1.
A.10.3.2.	¿El equipo de mantenimiento supervisa las operaciones que realiza el usuario?	Si __ No __	A.10.4.1.
<b>A.10.4.</b>	<b>Notificar la retirada</b>		
A.10.4.1.	¿Se notifica la inminencia de la retirada del software antiguo al cliente y usuarios?	Si __ No __	A.10.4.2. A.10.5.1.
A.10.4.2.	¿Cómo se comunica la retirada al Cliente?	Documento/ Correo electrónico/ Memorándum/ Oral/ Otra	A.10.5.1.
<b>A.10.5.</b>	<b>Almacenar datos del software antiguo</b>		
A.10.5.1.	¿Los datos del software antiguo son almacenados?	Si __ No __	A.10.5.2. A.10.6.1.
A.10.5.2.	¿En qué medio queda almacenado?	Texto	A.10.5.3.
A.10.5.3.	¿Existe un responsable de realizar estas acciones?	Si __ No __	A.10.5.4. A.10.6.1.
A.10.5.4.	¿Quién ocupa ese rol?	Roles	A.10.6.1.

.			
<b>A.10.6.</b>	<b>Verificar el Plan de retirada del Software.</b>		
A.10.6.1	¿Se verifica el Plan de retirada del Software?	Si __ No __	A.10.6.2. A.11.1.1.
A.10.6.2	¿Quién lo verifica?	Roles	A.10.6.3.
A.10.6.3	¿Queda documentado?	Si __ No __	A.10.6.4. A.11.1.1.
A.10.6.4	¿Responde a una plantilla predefinida?	Si __ No __	A.10.6.5.
A.10.6.5	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.11.1.1.
<b>A.11.</b>	<b>Finalización del Servicio.</b>		
<b>A.11.1.</b>	<b>Entrega del Inventario y de la documentación</b>		
A.11.1.1	¿Se entregan los productos de software generados y modificados al cliente?	Si __ No __	A.11.1.2. A.11.2.1.
A.11.1.2	¿Qué productos se entregan?	Texto	A.11.1.3.
A.11.1.3	¿Queda registrado en un documento los entregables para el cliente?	Si __ No __	A.11.1.4.
A.11.1.4	¿Existe un responsable de realizar estas acciones?	Si __ No __	A.11.1.5. A.11.2.1.
A.11.1.5	¿Quién ocupa ese rol?	Roles	A.11.2.1.
<b>A.11.2.</b>	<b>Cesión definitiva del servicio</b>		
A.11.2.1	¿Existe un documento de aceptación en donde se registre la conformidad del cliente y por ende el cese del servicio al producto?	Si __ No __	A.11.2.2. A.11.3.1.
A.11.2.2	¿Responde a una plantilla predefinida?	Si __ No __	A.11.2.3.
A.11.2.3	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	A.11.3.1.
<b>A.11.3.</b>	<b>Verificar la Finalización del servicio.</b>		
A.11.3.1	¿Se verifica la Finalización del servicio?	Si __ No __	A.11.3.2. FIN
A.11.3.2	¿Quién la verifica?	Roles	A.11.3.3.
A.11.3.3	¿Queda documentado?	Si __ No __	A.11.3.4. FIN
A.11.3.4	¿Responde a una plantilla predefinida?	Si __ No __	A.11.3.5.
A.11.3.5	¿En qué medio queda documentado (manual/digital)?	Manual/ Digital/	FIN

