



**UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

**“HERRAMIENTA PARA LA INTEGRACIÓN DE LA METODOLOGÍA  
SCRUM CON MOPROSOFT PARA INCREMENTAR LA  
PRODUCTIVIDAD DEL PERSONAL Y LA CALIDAD DEL  
PRODUCTO EN LAS MIPYMES DE DESARROLLO SOFTWARE”**

**TESIS**

**PARA OBTENER EL GRADO DE  
MAESTRO EN ELECTRÓNICA Y COMPUTACIÓN**

**PRESENTA**

**ING. JOSÉ ANTONIO GARCÍA HERNÁNDEZ**

**DIRECTOR DE TESIS  
DR. IVÁN ANTONIO GARCÍA PACHECO**

**HUAJUAPAN DE LEÓN, OAX., JUNIO DE 2014**



**Tesis presentada el 6 de junio de 2014,  
ante los siguientes sinodales:**

M.C. Everth Hayde Rocha Trejo

M.C. Ricardo Ruiz Ruiz

Dra. Carla Leninca Pacheco Agüero

Dr. Moisés Homero Sánchez López

**Director de Tesis:**

Dr. Iván Antonio García Pacheco



## **Dedicatoria**

A mi esposa Alejandra, porque desde que inicie este camino estuvo a mi lado apoyándome y motivándome a continuar durante el trayecto.

A mi hijo primogénito Joseph, porque entendió el sacrificio de no estar en algún momento con él, por ser el motor de la familia.

A Ustedes, todo mi amor y reconocimiento por darme su tiempo y aún más. Gracias.



## **Agradecimientos**

Primeramente agradezco a Dios, de quien viene todo lo bueno, y puso a las personas adecuadas en mi camino.

Gracias a mis padres, Nicolás Antonio Martínez y Andrea Hernández Hernández, por su disposición, confianza y apoyo desde que inicié este proyecto y culminó ahora con este logro en mi carrera profesional.

Gracias al Dr. Iván Antonio García Pacheco, por su paciencia y dedicación para dirigir mi trabajo y enfocarlo adecuadamente.

Gracias a mis hermanos y hermanas que me animaban a terminar. Nicolás, Alan, Andrea y Cesia.

Por último, pero sin ser menos importante, quiero agradecer a mi suegra la Sra. Luz Ma. Aguilar Solano y cuñadas, Sandra, Mariela y Raquel por su apoyo y ánimo constante.





## Índice

Dedicatoria .....	v
Agradecimientos .....	xi
Índice .....	xi
Lista de tablas .....	xv
Lista de figuras .....	xix
Resumen .....	xxi
Abstract .....	xxiii
1. Introducción .....	1
1.1. Introducción y motivaciones .....	1
1.2. Justificación e importancia del problema .....	2
1.3. Hipótesis .....	4
1.4. Limitaciones y delimitaciones del trabajo .....	4
1.5. Objetivo del trabajo .....	4
1.5.1. Objetivo general .....	5
1.5.2. Objetivos específicos .....	5
1.6. Aproximación a la solución .....	5
1.7. Estructura de la tesis .....	6
2. Marco conceptual .....	7
2.1. Introducción a los modelos de procesos de software .....	7
2.2. Modelos de referencia .....	10
2.2.1. CMMI-DEV V1.3 .....	10
2.2.1.1. Estructura .....	11
2.2.1.2. Niveles .....	12
2.2.1.3. Categorías de procesos .....	15
2.2.1.3.1. Gestión de procesos .....	15
2.2.1.3.2. Gestión de proyectos .....	16
2.2.1.3.3. Ingeniería .....	16
2.2.1.3.4. Soporte .....	16
2.2.1.4. Procesos involucrados en el desarrollo y mantenimiento de Software .....	17

2.2.2. MoProSoft .....	19
2.2.2.1. Estructura.....	19
2.2.2.2. Niveles.....	21
2.2.2.3. Categorías de procesos.....	21
2.2.2.3.1. Alta Dirección.....	22
2.2.2.3.2. Gerencia.....	22
2.2.2.3.3. Categoría de Operación.....	22
2.2.2.4. Procesos involucrados en el desarrollo y mantenimiento de Software.....	22
2.2.3. ISO/IEC 12207 .....	23
2.2.3.1. Estructura.....	24
2.2.3.2. Niveles.....	25
2.2.3.3. Categorías de procesos.....	25
2.2.3.3.1. Procesos de acuerdo.....	25
2.2.3.3.2. Procesos de habilitación de proyectos organizacionales.....	25
2.2.3.3.3. Procesos de proyecto.....	26
2.2.3.3.4. Procesos técnicos.....	26
2.2.3.3.5. Procesos de implementación de software.....	27
2.2.3.3.6. Procesos de soporte de software.....	27
2.2.3.3.7. Procesos de reúso de software.....	28
2.2.3.4. Procesos involucrados en el desarrollo y mantenimiento de software.....	28
2.2.4. Comparativa de los modelos de referencia.....	29
2.3. Métodos ágiles.....	31
2.3.1. Programación extrema.....	31
2.3.1.1. Valores y principios.....	31
2.3.1.2. Contenido.....	32
2.3.2. Scrum.....	34
2.3.2.1. Valores y principios.....	34
2.3.2.2. Contenido.....	34
2.3.3. Comparativa entre metodologías ágiles.....	37
2.4. Trabajo relacionado.....	37
2.5. Conclusiones del marco teórico.....	39
3. Planteamiento de la solución.....	41
3.1. Modelado de procesos.....	41
3.1.1. MoProSoft.....	42
3.1.1.1. Procesos.....	42
3.1.1.2. Roles.....	45
3.1.1.2.1. Administración de Proyectos Específicos.....	45
3.1.1.2.2. Desarrollo y Mantenimiento de Software.....	45
3.1.1.3. Productos.....	46

---

3.1.1.3.1. Administración de Proyectos Específicos.....	46
3.1.1.3.2. Desarrollo y Mantenimiento de Software.....	48
3.1.2. Scrum.....	49
3.1.2.1. Procesos.....	49
3.1.2.2. Roles.....	49
3.1.2.3. Productos.....	51
3.2. Análisis de MoProSoft vs Scrum.....	51
3.2.1. Procesos.....	51
3.2.2. Roles.....	55
3.2.3. Productos.....	56
3.2.4. Resultados del análisis.....	57
3.2.4.1. Administración de Proyecto Específico.....	57
3.2.4.1.1. Actividades.....	58
3.2.4.1.2. Roles.....	60
3.2.4.1.3. Productos.....	61
3.2.4.2. Desarrollo y Mantenimiento de Software.....	61
3.3. Diseño del patrón relacionado con la mezcla “MoProSoft – Scrum”.....	61
3.3.1. Administración de Proyectos Específicos.....	62
3.3.1.1. Productos.....	62
3.3.1.2. Roles involucrados y capacitación.....	64
3.3.1.3. Actividades.....	64
3.3.2. Desarrollo y Mantenimiento de Software.....	67
3.3.2.1. Productos.....	67
3.3.2.2. Roles involucrados y capacitación.....	70
3.3.2.3. Actividades.....	70
4. Soporte computacional para la integración de MoProSoft y Scrum.....	75
4.1. Requisitos funcionales.....	75
4.1.1. Gestión de acceso.....	75
4.1.2. Gestión de usuarios.....	76
4.1.3. Gestión de proyectos.....	78
4.1.4. Gestión de equipos de proyecto.....	80
4.1.5. Gestión de registros de pendientes de producto.....	81
4.1.6. Gestión de sprints.....	84
4.1.7. Gestión de tareas.....	85
4.2. Casos de uso.....	86
4.2.1. Casos de uso detallados.....	88
4.2.1.1. Gestión de acceso.....	88
4.2.1.2. Gestión de usuarios.....	90
4.2.1.3. Gestión de proyectos.....	92

---

4.2.1.4. Gestión de equipos de proyecto .....	95
4.2.1.5. Gestión de registros de pendientes de producto .....	98
4.2.1.6. Gestión de sprints.....	101
4.2.1.7. Gestión de tareas.....	103
4.3. Diseño de alto nivel: modelo entidad-relación para la gestión de proyectos.....	105
4.4. Diseño completo de la base de datos .....	106
4.4.1. Diseño de tablas y atributos.....	106
4.4.1.1. Tabla ítem.....	106
4.4.1.2. Tabla project.....	106
4.4.1.3. Tabla project_has_user.....	107
4.4.1.4. Tabla sprint.....	107
4.4.1.5. Tabla task.....	107
4.4.1.6. Tabla user .....	108
5. Experimentación sobre la mezcla de MoProSoft y Scrum .....	109
5.1. Contexto de la experimentación .....	109
5.2. Definición del caso de estudio.....	110
5.3. Implantación de la mezcla de MoProSoft y Scrum en el proyecto definido.....	112
5.3.1. Planificación.....	112
5.3.2. Realización .....	120
5.3.3. Evaluación y control.....	121
5.3.4. Cierre.....	121
5.4. Resultados .....	122
5.4.1. Análisis detallado sobre los resultados de la experimentación .....	122
5.4.1.1. Verificación de los objetivos en los proyectos hermanos .....	122
5.4.1.1.1. Objetivo 1: Cumplir con el calendario, esfuerzo y tamaño previsto.....	122
5.4.1.1.2. Objetivo 2: Reducir los tiempos en la identificación de defectos.....	123
5.5. Validación de la hipótesis de la tesis .....	124
6. Conclusiones y líneas futuras.....	125
7. Anexo A.- Acrónimos.....	129
8. Anexo B.- Carta de colaboración .....	133
9. Bibliografía .....	135
9.1. Sitios de Internet.....	137

## Lista de tablas

Tabla 1. Niveles de capacidad del CMMI-DEV v1.3 [CMMI10].	13
Tabla 2. Niveles de madurez del CMMI-DEV v1.3 [CMMI10].	14
Tabla 3. Áreas de proceso, categorías y niveles de madurez asociados del CMMI-DEV v1.3 [CMMI10].	15
Tabla 4. Niveles de capacidad de MoProSoft [NYCE05].	21
Tabla 5. Estructura de MoProSoft [NYCE05].	21
Tabla 6. Características evaluadas en los modelos de referencia analizados.	31
Tabla 7. Características evaluadas en los modelos ágiles analizados.	37
Tabla 8. Resumen de herramientas para la adopción de Scrum.	38
Tabla 9. Características evaluadas en las mezclas MoProSoft/XP y MoProSoft/Scrum.	40
Tabla 10. Participación de los roles de MoProSoft en OPE.1 APE.	45
Tabla 11. Participación de los roles de MoProSoft en OPE.1 APE.	46
Tabla 12. Insumos de las actividades del proceso de OPE.1 APE.	47
Tabla 13. Productos de las actividades del proceso de OPE.1 APE.	48
Tabla 14. Insumos de las actividades del proceso de OPE.2 DMS.	48
Tabla 15. Productos de las actividades del proceso de OPE.2 DMS.	49
Tabla 16. Roles participantes en el proceso de Scrum.	50
Tabla 17. Insumos de las actividades del proceso de Scrum.	51
Tabla 18. Productos de las actividades del proceso de Scrum.	51
Tabla 19. Análisis de MoProSoft vs Scrum.	52
Tabla 20. Intersecciones de las actividades de MoProSoft con las actividades de Scrum.	55
Tabla 21. Equivalencia de roles de MoProSoft con roles de Scrum.	56
Tabla 22. Relación entre productos de MoProSoft con productos de Scrum.	57
Tabla 23. Modificaciones a la tabla <i>Actividades del proceso</i> OPE.1 APE de MoProSoft.	59
Tabla 24. Modificaciones a la tabla <i>Roles involucrados y capacitación del proceso</i> OPE.1 APE de MoProSoft.	60
Tabla 25. Modificaciones a la tabla <i>Salidas del proceso</i> OPE.1 APE de MoProSoft.	61
Tabla 26. Tabla de entradas del proceso OPE.1 APE de MoProSoft – Scrum.	62
Tabla 27. Tabla de salidas del proceso OPE.1 APE de MoProSoft – Scrum.	63
Tabla 28. Tabla de productos internos del proceso OPE.1 APE de MoProSoft – Scrum.	64

Tabla 29. Tabla de roles involucrados y capacitación del proceso OPE.1 APE de MoProSoft – Scrum.....	64
Tabla 30. Tabla de actividades del proceso OPE.1 APE de MoProSoft – Scrum. ....	64
Tabla 31. Tabla de entradas del proceso OPE.2 DMS de MoProSoft – Scrum. ....	67
Tabla 32. Tabla de salidas del proceso OPE.2 DMS de MoProSoft – Scrum. ....	67
Tabla 33. Tabla de productos internos del proceso OPE.2 DMS de MoProSoft – Scrum. ....	70
Tabla 34. Tabla de roles involucrados y capacitación del proceso OPE.2 DMS de MoProSoft – Scrum.....	70
Tabla 35. Tabla de actividades del proceso OPE.2 DMS de MoProSoft – Scrum. ....	70
Tabla 36. Requisito funcional 1.1: Iniciar sesión. ....	75
Tabla 37. Requisito funcional 1.2: Editar información de acceso.....	76
Tabla 38. Requisito funcional 1.3: Cerrar sesión. ....	76
Tabla 39. Requisito funcional 2.1: Listar usuarios.....	76
Tabla 40. Requisito funcional 2.2: Crear usuario.....	77
Tabla 41. Requisito funcional 2.3: Editar usuario.....	77
Tabla 42. Requisito funcional 2.4: Borrar usuario.....	77
Tabla 43. Requisito funcional 3.1: Listar proyectos.....	78
Tabla 44. Requisito funcional 3.2: Cargar proyecto.....	78
Tabla 45. Requisito funcional 3.3: Cerrar proyecto.....	78
Tabla 46. Requisito funcional 3.4: Crear proyecto.....	79
Tabla 47. Requisito funcional 3.5: Editar datos de proyecto.....	79
Tabla 48. Requisito funcional 3.6: Borrar proyecto.....	79
Tabla 49. Requisito funcional 4.1: Listar miembros.....	80
Tabla 50. Requisito funcional 4.2: Agregar miembro.....	80
Tabla 51. Requisito funcional 4.3: Editar rol de miembro.....	81
Tabla 52. Requisito funcional 4.4: Remover miembro.....	81
Tabla 53. Requisito funcional 5.1: Listar pendientes de producto.....	81
Tabla 54. Requisito funcional 5.2: Crear pendiente de producto.....	82
Tabla 55. Requisito funcional 5.3: Editar pendiente de producto.....	82
Tabla 56. Requisito funcional 5.4: Borrar pendiente de producto.....	82
Tabla 57. Requisito funcional 5.5: Agregar pendiente a sprint.....	83
Tabla 58. Requisito funcional 5.6: Remover pendiente de sprint.....	83
Tabla 59. Requisito funcional 6.1: Listar sprints.....	84
Tabla 60. Requisito funcional 6.2: Crear sprint.....	84
Tabla 61. Requisito funcional 6.3: Editar sprint.....	84
Tabla 62. Requisito funcional 6.4: Borrar sprint.....	85
Tabla 63. Requisito funcional 7.1: Agregar tarea.....	85
Tabla 64. Requisito funcional 7.2: Editar tarea.....	85
Tabla 65. Requisito funcional 7.3: Borrar tarea.....	86

---

Tabla 66. Descripción de campos de la tabla ítem.....	106
Tabla 67. Descripción de campos de la tabla project.....	106
Tabla 68. Descripción de campos de la tabla project_has_user. ....	107
Tabla 69. Descripción de campos de la tabla sprint.....	107
Tabla 70. Descripción de campos de la tabla task. ....	107
Tabla 71. Descripción de campos de la tabla user. ....	108
Tabla 72. Número de empleados y actividad principal de la empresa incluida en el caso de estudio. .....	110
Tabla 73. Infraestructura de la empresa incluida en el caso de estudio. ....	110
Tabla 74. Resumen de funcionalidades de la herramienta desarrollada con los proyectos hermanos. .....	111
Tabla 75. Valores objetivos de los proyectos hermanos. ....	112
Tabla 76. Análisis de resultados para el objetivo relacionado con la productividad del equipo...	123
Tabla 77. Análisis de resultados para el objetivo relacionado con la calidad del producto.....	123





## Lista de figuras

Figura 1.1. Arquitectura de MeSoFT .....	3
Figura 1.2. Diagrama de procesos de MoProSoft [NYCE05] y los procesos donde es complementado con Scrum.....	5
Figura 2.1. Modelo de desarrollo en cascada .....	8
Figura 2.2. Modelo de desarrollo iterativo e incremental .....	8
Figura 2.3. Componentes del CMMI-DEV [CMMI10].....	11
Figura 2.4. Áreas de proceso de ingeniería [CMMI10] .....	17
Figura 2.5. Áreas de proceso de ingeniería [NYCE05] .....	23
Figura 2.6. Grupos de proceso del ciclo de vida [ISO08] .....	24
Figura 2.7. Reglas de la programación extrema [URL-6].....	32
Figura 2.8. Flujo de trabajo de la programación extrema [URL-6].....	33
Figura 2.9. El proceso de Scrum [Sutherland10; Schwaber10].....	35
Figura 3.1. Modelo descriptivo de OPE.1 Administración de Proyectos Específicos.....	42
Figura 3.2. Modelo descriptivo de OPE.2 Desarrollo y Mantenimiento de Software .....	43
Figura 3.3. Modelo descriptivo de la categoría de operación (OPE).....	44
Figura 3.4. Modelo descriptivo del proceso de Scrum .....	50
Figura 3.5. Análisis sobre el modelo descriptivo de la Categoría de Operación (OPE) versus Scrum .....	54
Figura 3.6. Relaciones de roles de OPE.2 DMS con el rol Equipo de Desarrollo de Scrum.....	56
Figura 3.7. Análisis de la Categoría de Operación (OPE) versus Scrum.....	58
Figura 4.1. Casos de uso para la herramienta para la integración MoProSoft - Scrum.....	87
Figura 4.2. Casos de uso de Gestión de acceso .....	88
Figura 4.3. Casos de uso de Gestión de usuarios.....	90
Figura 4.4. Casos de uso de Gestión de proyectos.....	92
Figura 4.5. Casos de uso de Gestión de equipos de proyecto.....	95
Figura 4.6. Casos de uso de Gestión de registros de pendientes de producto .....	98
Figura 4.7. Casos de uso de Gestión de sprints .....	101
Figura 4.8. Modelo de la base de datos para la herramienta de soporte a la mezcla MoProSoft – Scrum.....	105
Figura 5.1. Configuración de participantes del grupo experimental.....	113

Figura 5.2. Creación del proyecto por el Maestro de Scrum .....	113
Figura 5.3. Formación del grupo experimental .....	114
Figura 5.4. Agregación de pendientes .....	114
Figura 5.5. Creación del registro de pendientes .....	115
Figura 5.6. Opciones del equipo de trabajo .....	115
Figura 5.7. Análisis de tareas y adición de otras necesarias .....	116
Figura 5.8. Estimación de tiempo por parte del equipo de trabajo .....	116
Figura 5.9. Estimación de tiempo terminada .....	117
Figura 5.10. Estimación de esfuerzo .....	117
Figura 5.11. Creación del primer sprint para el proyecto .....	118
Figura 5.12. Asignación de pendientes a sprint .....	118
Figura 5.13. Estimación de sprints .....	119
Figura 5.14. Estimación de fechas de inicio y fin del proyecto .....	119
Figura 5.15. Reuniones diarias sobre el trabajo realizado .....	120
Figura 5.16. Sprints con actualización de tiempos .....	120
Figura 5.17. Actualización del estado de los pendientes .....	121

## **Resumen**

La Secretaría de Economía busca impulsar la calidad de los procesos de desarrollo de software de las empresas mexicanas mediante la implementación y certificación del Modelo de Procesos de Software (MoProSoft). De acuerdo al estudio sobre la Implementación de los Modelos de Calidad en la Construcción de Software en México, la tendencia para el año 2007 apuntaba al uso de metodologías ágiles para desarrollo de software, y colocaba a MoProSoft como el principal modelo nacional para el aseguramiento de la calidad.

En este contexto, la integración de Scrum con MoProSoft no parece ser una idea errada, si se considera que se ha logrado implementar con éxito dentro de un entorno donde ya funcionaba un modelo de mejora de procesos, puntualmente CMMI, y donde esta combinación ha logrado resultados impresionantes.

Así, esta propuesta de tesis plantea el desarrollo de un método sistemático para implementar, documentar y aprender sobre cómo se puede acoplar Scrum dentro de MoProSoft, específicamente para la categoría de procesos de Operación que aborda las prácticas de los procesos de desarrollo y mantenimiento de software. Con esto, se pretende obtener una guía y una herramienta que ayude a las empresas a implementar Scrum en entornos donde ya se esté utilizando, o donde se pretenda utilizar, MoProSoft. Esta herramienta permitirá incrementar la productividad del equipo de desarrollo y mejorar la calidad del producto final.



## **Abstract**

The Mexican Secretariat of Economics seeks to promote the quality of software development processes in Mexican companies through the implementation and certification of the Mexican Software Process Model (MoProSoft). According with the study on the Implementation of Quality Models in Software Development in Mexico, the trend for 2007 pointed to the use of agile methodologies for software development, and stood to MoProSoft as the main national model for quality assurance.

In this context, the integration of Scrum and MoProSoft is not a mistaken idea, considering that it has successfully implemented within an environment where a model for software process improvement had been used, specifically CMMI, and where this combination has achieved impressive results.

Thus, this thesis proposes the development of a systematic method to implement, document and learn about how MoProSoft and Scrum can be mixed, specifically for the category of Operation that addresses the processes for software development and maintenance. Thus, this research provides a guideline and a software tool that helps software companies to implement Scrum in environments where MoProSoft is already used, or in companies where intend to use it. The achieved results show that this software tool increases the productivity of the software team and improves the quality of the final product.



# 1. Introducción

## 1.1. Introducción y motivaciones

A nivel mundial la mayoría del software es producido por pequeñas empresas, con menos de 20 empleados, y medianas empresas, con menos de 50 empleados [Mishra08]. En este sentido, de acuerdo con Herrera y Martínez [Herrera03] aquellas empresas que fueran capaces de implementar una metodología de trabajo tendrían un mejor nivel competitivo que las empresas que no lo hicieran. Sin embargo, la implementación exitosa de las metodologías existentes en el entorno de las pequeñas y medianas empresas generalmente no es posible, debido a que estas no pueden absorber el costo de implementar algún programa de mejora de procesos de software [Mishra08].

Es bien sabido que las empresas de todos los tamaños tienen que gestionar y mejorar sus procesos de software, lidiar con rápidos cambios tecnológicos, dar mantenimiento a sus productos, operar en un entorno de software global, y sostener la organización mientras crecen. Sin embargo, las pequeñas empresas no son una versión a menor escala de las grandes, y requieren aproximaciones diferentes para modelos y metas de negocios específicas, nichos de mercado, disponibilidad de recursos (tanto financieros como humanos), capacidad de gestión y procesos, estructura organizacional, entre otras. Además las pequeñas empresas son extremadamente entusiastas y flexibles, que es lo que les brinda una ventaja competitiva [Richardson07].

En este sentido, un estudio realizado en el 2007 y publicado en el 2008 [Gutiérrez08], muestra que los modelos de calidad para las empresas mexicanas de desarrollo de software deben considerar las siguientes características de la industria mexicana de software:

- Equipos pequeños de desarrollo (36% entre 3 a 5 personas y el 28% de 6 a 10 personas).
- Entregas de producto en plazos de 3 a 5 meses.
- Desarrollo con base en la especificación de los Requisitos del cliente.
- Uso de desarrollos iterativos y entregas de prototipos funcionales al cliente.
- La baja calidad de la documentación generada en el proceso de desarrollo de software.

En este contexto, de acuerdo a la Secretaría de Economía el 99.8% de las empresas en México son Micro, Pequeñas y Medianas Empresas (MiPyMEs), y éstas generan el 52% del PIB y contribuyen con un 72% de los empleos formales [URL-1]. Así, el software en México es desarrollado principalmente por pequeños equipos dentro de empresas pequeñas que se dedican en

su mayoría (55%) al desarrollo de software a la medida. Pero en este escenario las pequeñas empresas son las más afectadas por la, todavía agobiante, crisis del software.

De entre tantas justificaciones, Zavala define a la crisis del software como una crisis de gestión, dado que la gestión de los proyectos ha sido uno de los problemas principales de fracaso tanto en las grandes empresas de software como en las pequeñas y medianas [Zavala08].

Como respuesta a esto, en el 2004 el Instituto de Ingeniería de Software (SEI, *Software Engineering Institute*) promovió un proyecto para implementar el Modelo de Madurez y Capacidad Integrado (CMMI, *Capability Maturity Model Integration*) [CMMI02] en “small settings” o pequeños entornos; proveyendo aproximaciones, herramientas, técnicas y guías [Richardson07].

En México, en el 2005 la Secretaría de Economía publicó la norma mexicana NMX-059-NYCE-2005 [NYCE05] basada en el estándar ISO/IEC 12207 [ISO95] e incluyendo prácticas de ISO 9000:2000 [ISO00], CMMI [CMMI02], PMBoK [PMI04], SWEBoK [IEEE04], e ISO/IEC 15504 [ISO04]. De manera similar, en el 2006 en Brasil, la Asociación para la promoción de la Excelencia del Software Brasileño (*Associação para promoção de Excelência do Software Brasileiro*) desarrolló el modelo MPS.BR (*Melhoria de Processo de Software Brasileiro*) con base en el estándar ISO/IEC 15504 [ISO04], alineándolo al marco de trabajo de CMMI enfocado en pequeñas y medianas organizaciones de software [Montoni09].

Así, el estudio de Gutiérrez et al., [Gutiérrez08] muestra que un 44.11% de las empresas ha considerado implementar MoProSoft como modelo de aseguramiento de la calidad de software, contra un 6% de las empresas que si lo han implementado. Esta diferencia abismal se debe en gran medida a que en los primeros niveles de madurez de cualquier modelo de referencia, el personal no requiere de capacitación específica para “realizar” las actividades que ahí son establecidas. Sin embargo, los niveles superiores deben ser soportados por prácticas que no compliquen aún más las tareas de las MiPyMEs y les permitan “administrar” sus proyectos, posiblemente una forma ágil de trabajo.

## 1.2. Justificación e importancia del problema

De acuerdo con Deemer et al., [Deemer08], se puede destacar que la metodología Scrum puede ayudar a cubrir cuatro de las características que, de acuerdo con [Gutiérrez08], debe mostrar un modelo de calidad enfocado a las empresas mexicanas:

1. Scrum en sí mismo es iterativo, un marco de trabajo incremental para proyectos y desarrollo de productos o aplicaciones.
2. Scrum fortalece el trabajo con equipos de 5 a 9 personas.
3. Scrum establece una estructura de desarrollo en ciclos de duración establecida y que no se amplían (se haya terminado el trabajo o no).
4. En Scrum el desarrollo se rige por una lista de características ordenadas por prioridad. Estas características son identificadas por el dueño del producto (o cliente).

Sin embargo, cabe mencionar que la baja calidad de la documentación es un problema que no puede ser resuelto con Scrum, pues éste no la considera importante bajo el supuesto de que un código funcional da más confianza que una documentación extensa de un sistema que no funciona.

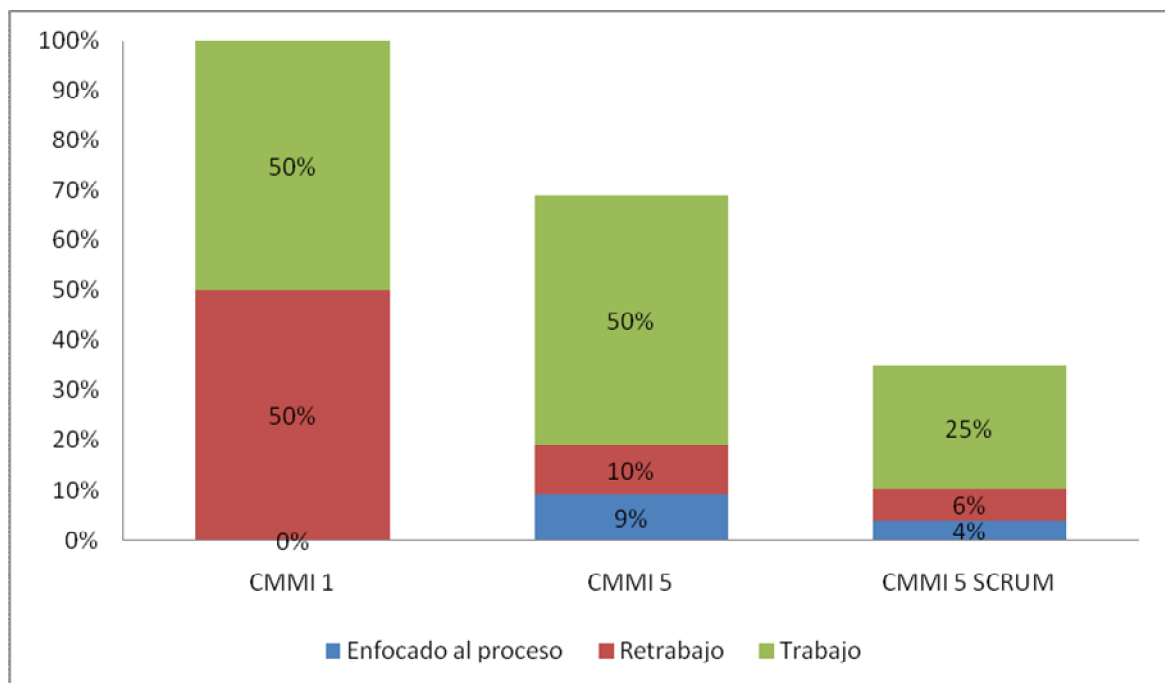
En este sentido, las ventajas de las metodologías ágiles de desarrollo de software, de acuerdo con [Mahanti06], se pueden resumir como:



- Aumento en el ROI (*Return of Investment*).
- Detección y cancelación temprana de proyectos fallidos.
- Calidad alta del software.
- Mejora del control sobre el proyecto.
- Reduce la dependencia en individuos e incrementa la flexibilidad en el equipo.

En este sentido, en [Sutherland07; Jakobsen09] se han presentado antecedentes de éxito relacionados con la conjunción de un modelo de mejora de procesos, CMMI, y una metodología ágil, Scrum, que reportan grandes beneficios. La implementación de esta “poción mágica” realizada en *Systematic Software Engineering* logró reducir el re-trabajo en un 80%, y también elevar la productividad en un 31%.

Así, la empresa *Systematic Software Engineering* [URL-2] obtuvo el Nivel 5 de CMMI en noviembre de 2005, y durante el 2006 adoptó Scrum [Jakobsen09]. La Figura 1.1 muestra que en un reporte temprano presentado por Jeff Sutherland en la Conferencia Agile 2007 se mostraron buenos resultados de esta combinación. Mejor aún, la empresa siguió implementando Scrum en más proyectos, mejorando todavía más los resultados obtenidos [Jakobsen09].



**Figura 1.1.** Arquitectura de MeSoFT

Por lo tanto, al usar Scrum con CMMI se mejoró el rendimiento del proyecto sin dejar de cumplir todas las exigencias del modelo CMMI. En la Figura 1.1 se observa que Scrum redujo cada categoría del trabajo (defectos, re-trabajo, trabajo total requerido, y gastos generales del proceso) en casi un 50%. También se hace notar que con equipos pequeños la reducción del trabajo llega a ser de hasta del 75%. La integración de CMMI con Scrum obedece a la combinación de CMMI, como

modelo de mejora que dice qué hacer, y Scrum, como una metodología de desarrollo que dice cómo hacer las cosas.

Ahora bien, en nuestro contexto México cuenta con MoProSoft [NYCE05] (el qué de la empresa), el cual está siendo apoyado por el modelo TSP (*Team Software Process*) [Humphrey02] (el cómo de la empresa) para fortalecer el área de gestión del desarrollo software. Sin embargo TSP es un modelo que requiere conocimiento sobre su antecesor el PSP (*Personal Software Process*) [Humphrey97] y que a efectos de la práctica, puede complicar la labor de un entorno pequeño de trabajo que no está acostumbrado a trabajar con plantillas de trabajo.

Dado lo anterior, por el tipo de empresas que constituyen la industria mexicana de software y tomando en consideración las características de los proyectos que desarrollan, se plantea combinar un modelo de mejora de procesos con una metodología ágil, específicamente MoProSoft con Scrum en la categoría de procesos de Operación. Esta combinación pretende mejorar la calidad de los procesos y productos en empresas mexicanas que ya implementan MoProSoft como modelo o en aquellas que estén planeando realizar una iniciativa de mejora.

### 1.3. Hipótesis

La hipótesis general para guiar el desarrollo de la presente tesis es:

*“Es posible desarrollar una herramienta para gestionar la integración de Scrum dentro de la categoría de procesos de Operación del modelo MoProSoft en las empresas mexicanas, con el objetivo de incrementar la productividad de la plantilla de trabajo y la calidad del producto.”*

### 1.4. Limitaciones y delimitaciones del trabajo

El presente trabajo de tesis busca proveer, a las empresas que estén relacionadas o interesadas en trabajar con el modelo MoProSoft, un método práctico para integrar el modelo de procesos y la metodología ágil de desarrollo software Scrum; proveyendo así una herramienta que cumpla con las especificaciones de MoProSoft, sin perder los beneficios de usar una metodología ágil.

El alcance de la presente tesis estará limitado por los siguientes factores:

1. Se trabajará con, al menos, una empresa que desee integrar Scrum en la categoría de procesos de Operación, como lo define el modelo MoProSoft, por lo menos hasta su Nivel 2.
2. Se desarrollará una herramienta en base a un análisis propio sobre la integración del modelo MoProSoft y la metodología Scrum, en la categoría de procesos de Operación, como lo define el modelo MoProSoft, por lo menos hasta su Nivel 2.
3. La herramienta desarrollada será probada en, al menos, una empresa desarrolladora de software del estado de Oaxaca que intente desplegar Scrum dentro de sus procesos de desarrollo y mantenimiento de software.
4. La solución será planteada únicamente para empresas del entorno MiPyME.

### 1.5. Objetivo del trabajo

Los objetivos del trabajo están compuestos de un objetivo general y varios objetivos secundarios, los cuales son descritos a continuación:

### 1.5.1. Objetivo general

*Obtener una herramienta para gestionar la integración de la metodología Scrum dentro de los procesos correspondientes a la categoría de Operación especificada por MoProSoft, que permita incrementar la productividad de la plantilla de trabajo de la organización y la calidad del producto.*

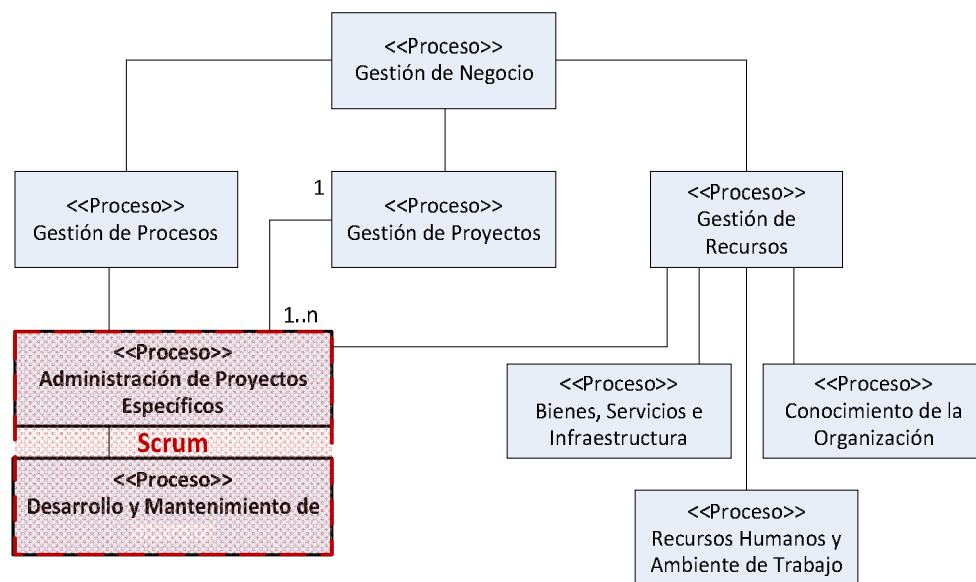
Para alcanzar el objetivo general será necesario conseguir ciertos objetivos secundarios. Estos establecerán las aportaciones esperadas al final de la tesis:

### 1.5.2. Objetivos específicos

- Estudiar la mezcla entre la norma mexicana MoProSoft y la metodología Scrum.
- Obtener datos porcentuales sobre los beneficios obtenidos cuando se implementa la mezcla MoProSoft - Scrum.
- Documentar la forma en que se implementó la mezcla MoProSoft - Scrum.
- Documentar las buenas prácticas en la implementación de la mezcla MoProSoft - Scrum.
- Identificar desaciertos en la implementación de la mezcla MoProSoft - Scrum.
- Obtener una herramienta computacional que ayude a obtener una mejor implementación de la mezcla MoProSoft - Scrum.

## 1.6. Aproximación a la solución

Para resolver el problema sobre la integración del modelo MoProSoft y la metodología Scrum, primeramente se pretende modelar los procesos de Administración de Proyectos Específicos (APE) y Desarrollo y Mantenimiento de Software (DMS), pertenecientes a la categoría de procesos de Operación, tal como lo define MoProSoft (ver Figura 1.2).



**Figura 1.2.** Diagrama de procesos de MoProSoft [NYCE05] y los procesos donde es complementado con Scrum

Este modelado usará la notación BPMN 2.0 (*Business Process Modeling Notation v2.0*), la cual es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo. También se analizarán los procesos de Scrum mediante el modelado de los procesos implicados en la metodología usando BPMN 2.0, para contar con una notación estándar y poder identificar de una mejor manera los procesos comunes y los lugares donde Scrum encaja dentro de la categoría de procesos de Operación de MoProSoft. Una vez obtenidos los resultados del análisis se obtendrá una lista de características que debe cumplir una herramienta para la integración de Scrum en la categoría de procesos de Operación de MoProSoft.

Por último se construirá la herramienta computacional y se desplegará en una empresa de desarrollo de software para poder validar el análisis, las características y la herramienta obtenida, mediante la documentación de la implementación de la herramienta en la empresa. Tal y como se indicó, la herramienta considerará la integración de Scrum con MoProSoft, específicamente para los procesos de APE y DMS, y las relaciones entre estos (ver Figura 1.2).

Los usuarios de esta herramienta serán los correspondientes roles de los procesos de APE y DMS, permitiéndoles generar los productos de entrada, salida e internos especificados por el modelo.

## **1.7. Estructura de la tesis**

La estructura del documento de tesis se detalla a continuación.

El Capítulo 2 presenta el marco conceptual sobre los modelos de referencia y metodologías ágiles utilizados por la industria de desarrollo de software en México, se expone un análisis comparativo sobre los modelos y metodologías revisadas.

En el Capítulo 3 se describe la mezcla que combina las prácticas del modelo MoProSoft con las técnicas de gestión incorporadas en Scrum.

El Capítulo 4 documenta la solución propuesta definiendo cada una de las partes que conforman la herramienta computacional que da soporte a la mezcla desarrollada en el capítulo anterior.

El Capítulo 5 narra la experimentación realizada durante esta investigación. Además de que se documentan los diferentes resultados parciales obtenidos, los objetivos alcanzados y los problemas encontrados en el desarrollo de esta tesis.

El Capítulo 6 presenta las conclusiones y líneas futuras sobre este trabajo.

El anexo A presenta una lista de acrónimos que se mencionan durante el desarrollo del presente trabajo.

El anexo B presenta la carta de colaboración con una empresa local para probar la solución planteada en esta tesis.

Por último se presentan las referencias bibliográficas utilizadas en el desarrollo de esta tesis.

## 2. Marco conceptual

### 2.1. Introducción a los modelos de procesos de software

De acuerdo con Sommerville [Sommerville09], el proceso de software es un conjunto de actividades que conducen a la creación de un producto software, y que además identifica cuatro actividades fundamentales:

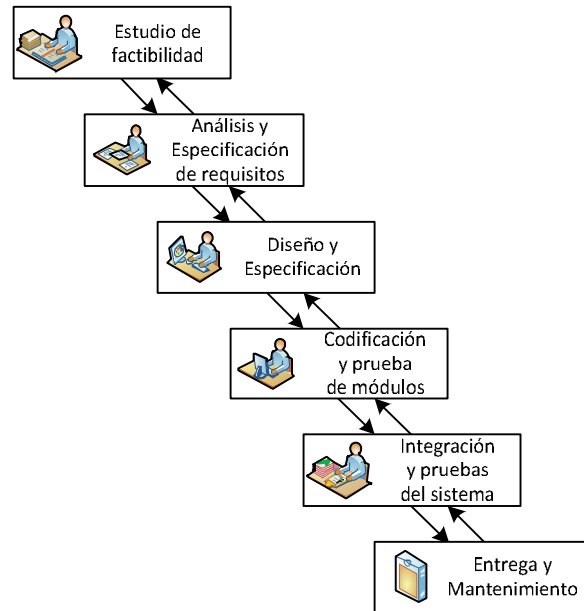
- Especificación. Actividad que define la funcionalidad del software y las restricciones de su operación.
- Diseño e implementación. Actividad donde se produce el software, cumpliendo con su especificación.
- Validación. Actividad que valida el software para asegurar que hace lo que el cliente requiere.
- Evolución. Actividad donde se modifica el software para adaptarlo a los cambios requeridos por el cliente.

Somerville también define a un modelo de proceso de software como una descripción simplificada de un proceso de software que presenta una visión de ese proceso. Por otro lado, Buede [Buede11] identifica cuatro modelos de proceso de software convencionales:

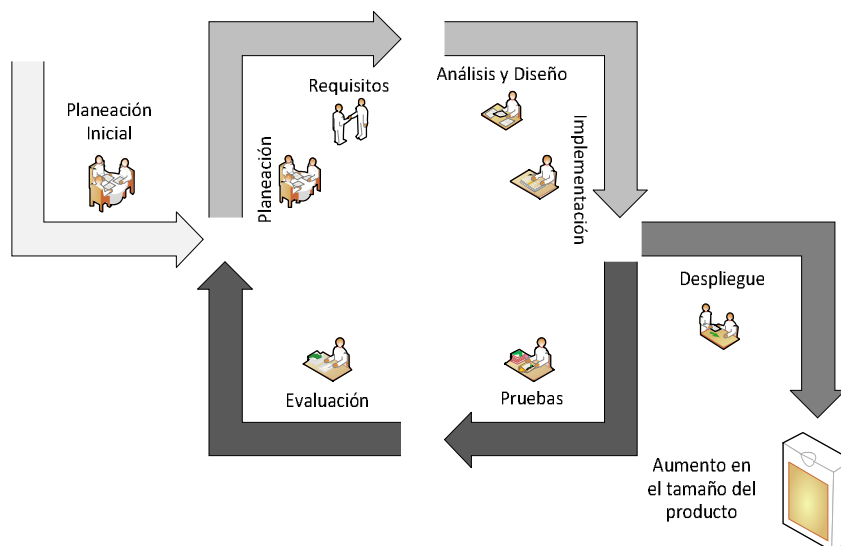
- Desarrollo en cascada. Es un modelo de fases ordenadas en forma de cascada simple en el cual el proceso de software evoluciona a través de una secuencia ordenada desde una fase a la siguiente según el orden. En la Figura 2.1 se muestra una visión general del modelo de cascada para el desarrollo de software.
- Refinamiento paso a paso. En esta aproximación los sistemas de software son desarrollados a través de un refinamiento y crecimiento progresivo de las especificaciones del sistema de alto nivel en componentes de código fuente. Tiene dos enfoques conocidos: *top-down* y *bottom-up*.
- Desarrollo y entregas incrementales. Esta aproximación se enfoca a desarrollar sistemas a través de entregas incrementales que proporcionen primeramente funciones operativas esenciales, y después proporcionen a los usuarios del sistema versiones mejoradas y más capaces del sistema en intervalos regulares. Este enfoque ha sido extendido a través del uso de herramientas y técnicas de prototipado de software, las cuales proporciona de forma más directa el soporte a desarrollos y entregas incrementales e iterativas para retroalimentación y

evaluación temprana y sobre la marcha. En la Figura 2.2 se muestra un ejemplo de un modelo de desarrollo iterativo e incremental.

- Estándares industriales y militares, y modelos de capacidad. Las firmas industriales con frecuencia adoptan alguna variación del modelo de cascada como la base para la estandarización de sus prácticas de desarrollo de software. Debido a que la estandarización con frecuencia está motivada por la necesidad de simplificar o eliminar complicaciones que surgen durante la gestión de desarrollos software o proyectos grandes.



**Figura 2.1.** Modelo de desarrollo en cascada



**Figura 2.2.** Modelo de desarrollo iterativo e incremental

Desde 1970 a la fecha, muchos contratistas del gobierno organizaron sus actividades de desarrollo software de acuerdo a los estándares de software militar. Estos estándares crecieron de manera diferente a las actividades del ciclo de vida clásicos, al igual que los documentos requeridos por los clientes quienes requieren o bien sistemas de software o plataformas complejas acopladas en sistemas de software incrustado.

En los entornos industriales, los modelos estándar de desarrollo de software a menudo proporcionan guías explícitas de cómo implementar, instalar, personalizar y optimizar una nueva entrega del software en su entorno de aplicación. Además, estas normas están destinadas a ser compatibles con el Aseguramiento de la Calidad del software, la Gestión de la Configuración, y los procesos de Verificación y Validación independientes en un proyecto de desarrollo de varios contratistas. Esta aproximación considera esfuerzos en el seguimiento y la medición del rendimiento de los procesos de software. Estos esfuerzos, a su vez contribuirán a preparar el camino para lo que muchas organizaciones de desarrollo de software practican, o se han certificado para practicar, evaluando la capacidad del proceso de software, siguiendo por ejemplo el modelo CMMI-DEV v1.2 (*Capability Maturity Model Integration for Development v1.2*) [CMMI06], perteneciente a la constelación de modelos CMMI desarrollados por el SEI [URL-3] y la Universidad de Carnegie Mellon.

Para Deemer et al., [Deemer08] los modelos de proceso convencionales tienen la gran fortaleza de pensar de manera muy lógica antes de empezar a construir, escribiendo todo, siguiendo un plan, y manteniendo todas las cosas tan organizadas como sea posible. Deemer et al., también afirman que estos modelos convencionales solo tienen una gran debilidad, y es el factor humano involucrado. En una aproximación convencional todas las buenas ideas deben aparecer en el inicio del lanzamiento del ciclo. Y si una buena idea ocurre en etapas avanzadas y el proceso no permite cambios, entonces la innovación se perderá. Dentro de las alternativas de los modelos de desarrollo convencionales, se encuentra la familia de modelos ágiles de desarrollo, los cuales han evolucionado a partir de las aproximaciones del modelo de desarrollo iterativo e incremental.

Estos modelos ágiles se basan en los 12 principios del Manifiesto Ágil [URL-4], los cuales se definen de la siguiente manera:

1. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptar que los requisitos cambian, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar una ventaja competitiva al cliente.
3. Entregar software funcional de manera frecuente, entre dos semanas y dos meses, dando preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Es necesario brindar el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo, y entre sus miembros, es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.

9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para que, a consecuencia, su comportamiento sea ajustado y perfeccionado.

Así, existen dos aproximaciones para el uso de los modelos de procesos de software, la aproximación convencional que usa modelos de referencia, como CMMI-DEV v1.2 [CMMI06] y MoProSoft [NYCE05], y la aproximación ágil que usa metodologías ágiles, como la Programación Extrema (XP, *Extreme Programming*) [Beck04] y Scrum [Schwaber10]. De acuerdo con [Mahanti06], [Sutherland10], y [Buede11] ambas aproximaciones son ampliamente usadas en las empresas y pueden ser aplicadas en pequeños entornos.

En este sentido, Gutiérrez et al., [Gutierrez08] afirman que entre los modelos de referencia más considerados en México se encuentran los siguientes:

- CMMI-DEV v1.2 de la familia de modelos de CMMI, desarrollado por el SEI [CMMI06].
- MoProSoft, que es la norma mexicana (NMX-059-NYCE-2005) para modelos de procesos de software [NYCE05].
- El estándar ISO/IEC 15504 [ISO04], el cual se puede ajustar a pequeñas y medianas organizaciones de software.

Se menciona también que las empresas mexicanas utilizan metodologías ágiles en un 41.25%. Además, existen otras empresas que utilizan estas metodologías, entre las cuales destacan:

- Scrum, utilizado por Easel Corporation, VMARK, IDX Systems, Patient Keeper, Systematic, Microsoft y otros.
- Extreme Programming (XP), implementado en proyectos pilotos por ABB, Daimler-Chrysler, Motorola, y Nokia.

Con el objetivo de analizar las fortalezas y debilidades de cada modelo en relación con el tema de tesis, en las siguientes secciones se hará un análisis detallado de los tres modelos de referencia y las dos metodologías ágiles mencionadas anteriormente.

## 2.2. Modelos de referencia

### 2.2.1. CMMI-DEV V1.3

La suite de productos CMMI fue desarrollada por el Equipo del Producto CMMI, el cual está conformado por expertos de mejora de procesos del Gobierno de los Estados Unidos de Norte América, de la industria, y del SEI. El objetivo del equipo es mejorar el Modelo de Madurez y Capacidad de Software existente (SW-CMM), lanzado en 1991 [URL-5].

El modelo CMMI-DEV, o CMMI para Desarrollo (*Capability Maturity Model Integration for Development*), fue introducido en su versión 1.2 en la suite de productos CMMI y la última versión es la 1.3 [CMMI10]. CMMI-DEV es un conjunto integrado de guías para desarrollar productos y

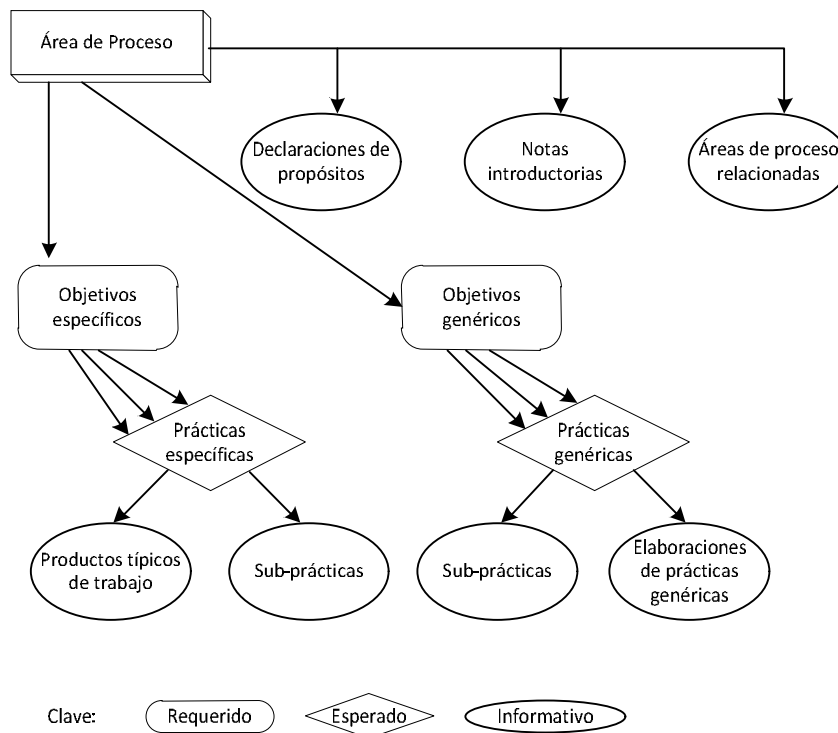


servicios software que fue diseñado como una guía en la aplicación de las mejores prácticas del modelo CMMI en una organización de desarrollo. Estas mejores prácticas del modelo están enfocadas en las actividades para desarrollar productos y servicios de calidad, y además para satisfacer las necesidades de los clientes y usuarios finales.

### 2.2.1.1. Estructura

El modelo agrupa tres tipos de componentes (ver Figura 2.3):

- Componentes requeridos. Describen lo que una organización debe realizar para satisfacer un área de proceso. Los componentes requeridos en un modelo CMMI son las metas específicas y los objetivos genéricos.
- Componentes esperados. Describen lo que una organización puede implementar para lograr un componente requerido. Los componentes esperados incluyen a las prácticas específicas y las prácticas genéricas.
- Componentes informativos. Proporcionan detalles que ayudan a la organización a comenzar la aproximación a los componentes requeridos y esperados. Las subprácticas, los productos de trabajo típicos, las ampliaciones, las elaboraciones de las prácticas genéricas, los títulos de metas y prácticas, las notas de las metas y las prácticas, y las referencias son ejemplos de componentes informativos dentro del modelo.



**Figura 2.3.** Componentes del CMMI-DEV [CMMI10]

A continuación se describen los componentes del modelo:

- Declaración de propósitos. La declaración de propósitos describe la finalidad del área de proceso y es un componente informativo.

- **Notas introductorias.** La sección de notas introductorias del área de proceso describe los conceptos principales cubiertos por el área de proceso y es un componente informativo.
- **Áreas de proceso relacionadas.** La sección de áreas de proceso relacionadas lista las referencias a áreas de proceso que están en relación y refleja las relaciones de alto nivel entre las áreas de proceso. Es un componente informativo.
- **Metas específicas.** Una meta específica describe las características únicas que deben estar presentes para satisfacer el área de proceso. Una meta específica es un componente requerido del modelo que se utiliza en las evaluaciones para ayudar a determinar si se satisface un área de proceso.
- **Metas genéricas.** Las metas genéricas se denominan genéricas porque la misma declaración de la meta se aplica a múltiples áreas de proceso. Una meta genérica describe las características que deben estar presentes para institucionalizar los procesos que implementan un área de proceso. Una meta genérica es un componente requerido del modelo y se utiliza en las evaluaciones para determinar si se satisface un área de proceso.
- **Prácticas específicas.** Una práctica específica es la descripción de una actividad que se considera importante para alcanzar la meta específica asociada. Las prácticas específicas describen las actividades que se espera que produzcan la consecución de las metas específicas de un área de proceso. Una práctica específica es un componente esperado del modelo.
- **Productos de trabajo típicos.** La sección de productos de trabajo típicos lista muestras de resultados de una práctica específica. Estos ejemplos se denominan productos de trabajo típicos porque a menudo hay otros productos de trabajo que son igual de eficaces pero no están en la lista. Un producto de trabajo típico es un componente informativo del modelo.
- **Subprácticas.** Una subpráctica es una descripción detallada que proporciona una guía para interpretar e implantar una práctica específica o genérica. Las subprácticas pueden tomar un carácter prescriptivo, pero realmente son un componente informativo indicado sólo para proporcionar ideas que puedan ser útiles para la mejora de proceso.
- **Prácticas genéricas.** Las prácticas genéricas se denominan genéricas porque la misma práctica se aplica a múltiples áreas de proceso. Una práctica genérica es la descripción de una actividad que se considera importante para el logro de la meta genérica asociada. Una práctica genérica es un componente esperado del modelo.
- **Elaboración de prácticas genéricas.** Una elaboración de práctica genérica aparece después de una práctica genérica en un área de proceso, para proporcionar una guía sobre cómo la práctica genérica debería aplicarse de forma exclusiva al área de proceso. Una elaboración de práctica genérica es un componente informativo del modelo.

### **2.2.1.2. Niveles**

Los niveles se utilizan en CMMI para describir un camino evolutivo recomendado para una organización que quiera mejorar los procesos que utiliza para desarrollar y mantener sus productos y servicios de software. Los niveles pueden también ser el resultado de las evaluaciones realizadas para organizaciones, incluyendo compañías completas o grupos más pequeños, tal como un grupo de proyectos o una sección dentro de una compañía.

CMMI soporta dos caminos de mejora:

- Un camino permite a las organizaciones mejorar de forma incremental los procesos que corresponden a un área de proceso individual seleccionada por la organización.
- El otro camino permite a las organizaciones mejorar un conjunto de procesos relacionados, tratando de forma incremental conjuntos sucesivos de áreas de proceso.

Estos dos caminos de mejora están asociados con los dos tipos de niveles que corresponden a las dos representaciones:

- La representación continua, que utiliza el término de “nivel de capacidad” (ver Tabla 1).
- La representación por etapas, que utiliza el término de “nivel de madurez” (ver Tabla 2).

**Tabla 1.** Niveles de capacidad del CMMI-DEV v1.3 [CMMI10].

<b>Nivel</b>	<b>Nombre</b>	<b>Descripción</b>
0	Incompleto	Un proceso incompleto es un proceso que, o bien no se ejecuta, o se ejecuta parcialmente. Al menos una de las metas específicas del área de proceso no se satisface y no existen metas genéricas para ese nivel, ya que no hay ninguna razón para institucionalizar un proceso ejecutado parcialmente.
1	Realizado	Un proceso de nivel de capacidad 1 se caracteriza como un proceso realizado. Un proceso realizado es un proceso que satisface las metas específicas del área de proceso. Soporta y permite el trabajo necesario para producir los productos del trabajo.
2	Gestionado	Un proceso de nivel de capacidad 2 se caracteriza como un proceso gestionado. Un proceso gestionado es un proceso realizado (nivel de capacidad 1) que tiene la infraestructura básica dispuesta para soportar el proceso. Se planifica y ejecuta de acuerdo a políticas; emplea personal con habilidades; tiene los recursos adecuados para producir resultados controlados; involucra a las partes interesadas relevantes; se monitoriza, controla y revisa; y se evalúa la adherencia a su descripción de proceso. La disciplina de proceso reflejada por el nivel de capacidad 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés.
3	Definido	Un proceso de nivel de capacidad 3 se caracteriza como un proceso definido. Un proceso definido es un proceso gestionado (nivel de capacidad 2) que se adapta a partir del conjunto de procesos estándar de la organización, de acuerdo a las guías de adaptación de la organización, y contribuye a los activos de proceso de la organización con productos del trabajo, métricas e información adicional de mejora de procesos.

**Tabla 2.** Niveles de madurez del CMMI-DEV v1.3 [CMMI10].

<b>Nivel</b>	<b>Nombre</b>	<b>Descripción</b>
1	Inicial	En el nivel de madurez 1, los procesos son generalmente <i>ad-hoc</i> y caóticos. La organización generalmente no proporciona un entorno estable para dar soporte a los procesos. El éxito en estas organizaciones depende de la competencia y heroicidad del personal de la organización y no del uso de procesos probados. A pesar de este caos, las organizaciones de nivel de madurez 1 a menudo producen productos y servicios que funcionan; sin embargo, frecuentemente exceden sus presupuestos y no cumplen sus calendarios.
2	Gestionado	En el nivel de madurez 2, los proyectos de la organización se asegura que los procesos se planifican y realizan de acuerdo a políticas; los proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados; involucran a las partes interesadas relevantes; se monitorizan, controlan y revisan; y se evalúan en cuanto a su adherencia a sus descripciones de proceso. La disciplina de proceso reflejada por el nivel de madurez 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados.
3	Definido	En el nivel de madurez 3, los procesos son bien caracterizados y comprendidos, y se describen en estándares, procedimientos, herramientas y métodos. El conjunto de procesos estándar de la organización, que es la base del nivel de madurez 3, se establece y mejora a lo largo del tiempo. Estos procesos estándar se usan para establecer la consistencia en toda la organización. Los proyectos establecen sus procesos definidos adaptando el conjunto de procesos estándar de la organización de acuerdo a las guías de adaptación.
4	Gestionado cuantitativamente	En el nivel de madurez 4, la organización y los proyectos establecen objetivos cuantitativos en cuanto al rendimiento de calidad y del proceso, y los utilizan como criterios en la gestión de los procesos. Los objetivos cuantitativos se basan en las necesidades del cliente, usuarios finales, organización e implementadores del proceso.
5	Optimizando	El nivel de madurez 5 se centra en mejorar continuamente el rendimiento de procesos mediante mejoras incrementales e innovadoras de proceso y tecnológicas. Los objetivos cuantitativos de mejora de procesos para una organización se establecen, se revisan continuamente para reflejar el cambio a los objetivos del negocio, y se utilizan como criterios para gestionar la mejora de procesos. Los efectos de las mejoras de procesos desplegadas se miden y evalúan frente a los objetivos cuantitativos de mejora de procesos. Tanto los procesos definidos como el conjunto de procesos estándar de la organización son objeto de las actividades de mejora cuantitativa.

### 2.2.1.3. Categorías de procesos

El modelo cuenta con 22 áreas de proceso divididas en cuatro categorías (Gestión de procesos, Gestión de proyectos, Ingeniería, y Soporte). La Tabla 3 resume las áreas de proceso con su categoría y nivel correspondientes:

**Tabla 3.** Áreas de proceso, categorías y niveles de madurez asociados del CMMI-DEV v1.3 [CMMI10].

Abreviatura	Nombre	Categoría	Nivel de madurez
OPD	Definición de procesos de la organización	Gestión de procesos	3
OPF	Enfoque en procesos de la organización	Gestión de procesos	3
OPM	Gestión del rendimiento de la organización	Gestión de procesos	5
OT	Formación en la organización	Gestión de procesos	3
OPP	Rendimiento de procesos de la organización	Gestión de procesos	4
PMC	Monitorización y control del proyecto	Gestión de proyectos	2
PP	Planificación del proyecto	Gestión de proyectos	2
SAM	Gestión de acuerdos con proveedores	Gestión de proyectos	2
IPM	Gestión integrada del proyecto	Gestión de proyectos	3
RSKM	Gestión de riesgos	Gestión de proyectos	3
QPM	Gestión cuantitativa del proyecto	Gestión de proyectos	4
REQM	Gestión de requisitos	Ingeniería	2
PI	Integración del producto	Ingeniería	3
RD	Desarrollo de requisitos	Ingeniería	3
TS	Solución técnica	Ingeniería	3
VAL	Validación	Ingeniería	3
VER	Verificación	Ingeniería	3
CM	Gestión de configuración	Soporte	2
MA	Medición y análisis	Soporte	2
PPQA	Aseguramiento de la calidad del proceso y del producto	Soporte	2
DAR	Análisis de decisiones y resolución	Soporte	3
CAR	Análisis causal y resolución	Soporte	5

#### 2.2.1.3.1. Gestión de procesos

Las áreas de proceso de la *Gestión de procesos* contienen las actividades transversales a los proyectos relacionadas con la definición, planificación, despliegue, implementación, monitorización, control, evaluación, medición y mejora de los procesos.

Las áreas de proceso de Gestión de procesos de CMMI-DEV son las siguientes:

- Enfoque en Procesos de la Organización (OPF).

- Definición de Procesos de la Organización (OPD).
- Formación en la Organización (OT).
- Rendimiento de Procesos de la Organización (OPP).
- Gestión del Rendimiento de la Organización (OPM).

#### 2.2.1.3.2. Gestión de proyectos

Las áreas de proceso de la *Gestión de proyectos* cubren las actividades de gestión de proyectos relacionadas con la planificación, monitorización y control de proyectos.

Las áreas de proceso de Gestión de proyectos del CMMI son las siguientes:

- Planificación del Proyecto (PP).
- Monitorización y Control del Proyecto (PMC).
- Gestión de Acuerdos con Proveedores (SAM).
- Gestión Integrada del Proyecto (IPM).
- Gestión de Riesgos (RSKM).
- Gestión Cuantitativa del Proyecto (QPM).

#### 2.2.1.3.3. Ingeniería

Las áreas de proceso de *Ingeniería* cubren las actividades de desarrollo y de mantenimiento que se comparten entre las disciplinas de ingeniería. Las áreas de proceso de *Ingeniería* también integran los procesos asociados con diferentes disciplinas de ingeniería en un único proceso de desarrollo de producto, dando soporte a una estrategia de mejora de procesos orientada a producto. Las áreas de proceso de *Ingeniería* se aplican al desarrollo de cualquier producto o servicio dentro del dominio de desarrollo.

Las áreas de proceso de Ingeniería de CMMI son las siguientes:

- Desarrollo de Requisitos (RD).
- Gestión de Requisitos (REQM).
- Solución Técnica (TS).
- Integración del Producto (PI).
- Verificación (VER).
- Validación (VAL).

#### 2.2.1.3.4. Soporte

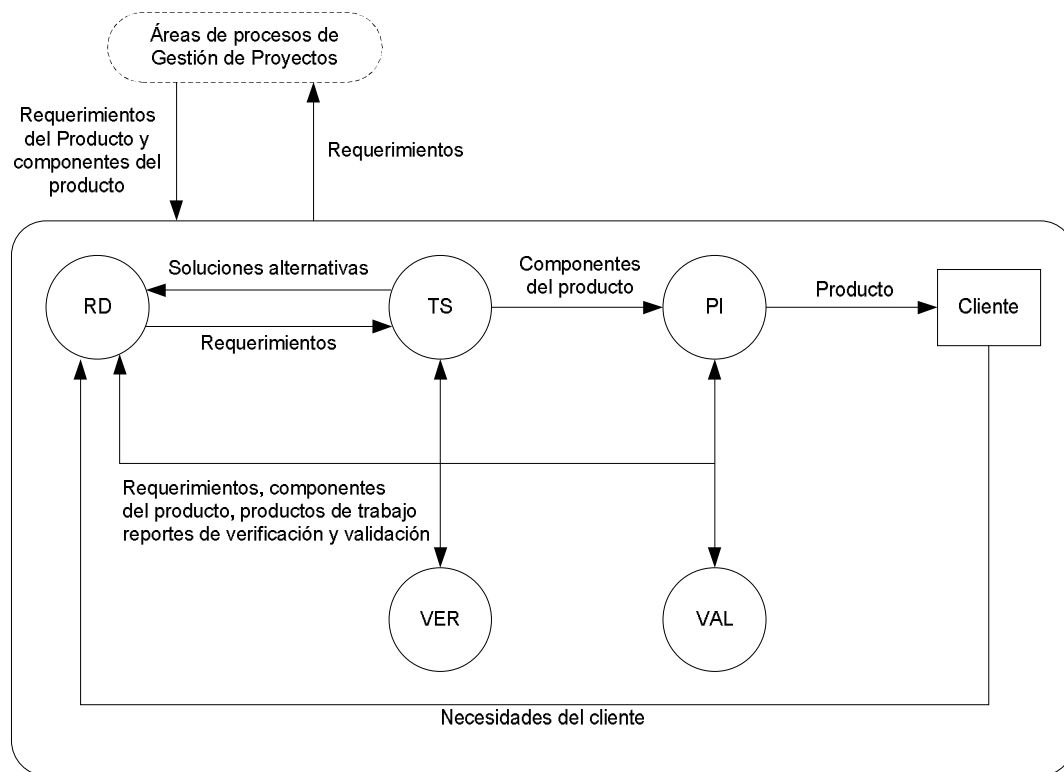
Las áreas de proceso de *Soporte* cubren las actividades que dan soporte al desarrollo y al mantenimiento del producto. Las áreas de proceso de *Soporte* tratan los procesos que se usan en el contexto de la ejecución de otros procesos. En general, las áreas de proceso de *Soporte* tratan los procesos que están orientados al proyecto y pueden tratar procesos que se aplican de manera más general a la organización.

Las áreas de proceso de soporte de CMMI-DEV son las siguientes:

- Gestión de Configuración (CM).
- Aseguramiento de la Calidad del Proceso y del Producto (PPQA).
- Medición y Análisis (MA).
- Análisis de Decisiones y Resolución (DAR).
- Análisis Causal y Resolución (CAR).

#### 2.2.1.4. Procesos involucrados en el desarrollo y mantenimiento de Software

Los procesos del CMMI-DEV involucrados en el desarrollo y mantenimiento de software se encuentran dentro de la categoría de Ingeniería. La Figura 2.4 muestra las interacciones entre las áreas de proceso que componen esta categoría.



PI = Integración del producto  
 RD = Desarrollo de requerimientos  
 TS = Solución técnica  
 VAL = Validación  
 VER = Verificación

**Figura 2.4.** Áreas de proceso de ingeniería [CMMI10]

El área de proceso de RD identifica las necesidades del cliente y traduce dichas necesidades en Requisitos del producto. El conjunto de Requisitos del producto se analiza para producir una solución conceptual de alto nivel. Este conjunto de Requisitos se asigna entonces para establecer un conjunto inicial de Requisitos de componentes del producto. Otros Requisitos que ayudan a definir el producto se derivan y asignan a componentes del producto. Este conjunto de Requisitos del producto y de componentes del producto describe de manera clara la ejecución del producto, las características del diseño, los Requisitos de verificación, etcétera, de tal modo que el desarrollador

pueda comprender y usar. Así, el área de proceso de RD suministra los Requisitos al área de proceso TS, donde los Requisitos se convierten en la arquitectura del producto, el diseño de los componentes del producto, y los propios componentes del producto. Los Requisitos se suministran también al área de proceso de PI, donde se combinan los componentes del producto y se verifican las interfaces para asegurar que cumplen con los Requisitos de interfaz suministrados por el área de proceso RD.

El área de proceso de REQM mantiene los Requisitos. Describe las actividades para obtener y controlar los cambios a los Requisitos, y asegurar que otros planes y datos relevantes se mantengan actualizados. Proporciona la trazabilidad de los Requisitos desde el cliente al producto y de éste a sus componentes. REQM asegura que los cambios a los Requisitos se reflejen en los planes, actividades y productos de trabajo del proyecto. Este ciclo de cambios puede afectar a todas las otras áreas de proceso de Ingeniería; así, REQM es una secuencia de eventos dinámica y a menudo recursiva. El área de proceso de REQM es fundamental para un proceso de diseño de ingeniería controlado y disciplinado.

El área de proceso de TS desarrolla los paquetes de datos técnicos relativos a los componentes del producto que serán usados por el área de proceso de PI o SAM. Se examinan soluciones alternativas con la intención de seleccionar el diseño óptimo basado en criterios establecidos. Estos criterios pueden ser significativamente diferentes entre los productos, dependiendo del tipo de producto, entorno operativo, Requisitos de ejecución, Requisitos de soporte y costes o calendarios de entrega. La tarea de seleccionar la solución final hace uso de las prácticas específicas del área DAR. TS se basa en las prácticas específicas del área de proceso de VER para realizar la verificación del diseño y las revisiones entre pares durante el diseño y antes del ensamblaje final.

El área de proceso de VER asegura que los productos de trabajo seleccionados cumplan los Requisitos especificados. El área de proceso de VER selecciona productos de trabajo y métodos de verificación que se usarán para verificar los productos de trabajo frente a los Requisitos especificados. La verificación es generalmente un proceso incremental, que comienza con la verificación de componentes del producto y normalmente concluye con la verificación de los productos ensamblados totalmente. La verificación también trata las revisiones entre pares. Las revisiones entre pares son un método probado para eliminar defectos de manera temprana y proporcionar una visión valiosa sobre los productos de trabajo y los componentes de producto que están siendo desarrollados y mantenidos.

El área de proceso de VAL valida de manera incremental los productos frente a las necesidades del cliente. La validación puede realizarse en el entorno operacional o en un entorno operacional simulado. La coordinación con el cliente sobre los Requisitos de validación es un elemento importante de esta área de proceso. El alcance del área de proceso de VAL incluye la validación de productos, de componentes de producto, de productos de trabajo intermedios seleccionados y de procesos. Estos elementos validados pueden requerir con frecuencia volver a ser verificados y validados. Los problemas descubiertos durante la validación se resuelven normalmente en las áreas de proceso de RD o de TS.

El área de proceso de PI contiene las prácticas específicas asociadas con la generación de la mejor secuencia de integración posible, integrando los componentes del producto, y entregando el producto al cliente. PI usa las prácticas específicas de ambas áreas de proceso, VER y VAL, al implementar el proceso de integración de producto. Las prácticas de verificación verifican las interfaces y los Requisitos de interfaz de los componentes del producto antes de la integración del producto. Esto es un evento esencial en el proceso de integración. Durante la integración del producto en el entorno operacional, se usan las prácticas específicas del área de proceso de VAL.



### 2.2.2. MoProSoft

MoProSoft es el Modelo de Procesos para la Industria de Software en México [NYCE05], y fue desarrollado con el propósito de fomentar la estandarización de las operaciones de la industria, a través de la incorporación de las mejores prácticas en gestión e Ingeniería de Software. También se pretende, mediante la adopción del modelo, elevar la capacidad de las organizaciones para ofrecer servicios de calidad y alcanzar niveles internacionales de competitividad. MoProSoft va dirigido a las empresas o áreas internas dedicadas al desarrollo y mantenimiento de software [Oktaba08].

#### 2.2.2.1. Estructura

La estructura de MoProSoft se basa en un patrón de procesos que esquematiza los elementos que servirán para la documentación de los procesos. Está constituido por tres partes:

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

Los componentes de la Definición general del proceso son:

- Proceso. Nombre de proceso, precedido por el acrónimo establecido en la definición de los elementos de la estructura del modelo de procesos.
- Categoría. Nombre de la categoría a la que pertenece el proceso y el acrónimo entre paréntesis.
- Propósito. Objetivos generales medibles y resultados esperados de la implantación efectiva del proceso.
- Descripción. Descripción general de las actividades y productos que componen el flujo de trabajo del proceso.
- Objetivos. Objetivos específicos cuya finalidad es asegurar el cumplimiento del propósito del proceso.
- Indicadores. Definición de los indicadores para evaluar la efectividad del cumplimiento de los objetivos del proceso.
- Metas cuantitativas. Valor numérico o rango de satisfacción por indicador.

- Responsabilidad y autoridad. Responsabilidad es el rol principal responsable por la ejecución del proceso. Autoridad es el rol responsable por validar la ejecución del proceso y el cumplimiento de su propósito.
- Subprocesos. De forma opcional se listan los procesos de los cuales se compone el proceso en cuestión.
- Procesos relacionados. Nombres de los procesos relacionados.
- Entradas. Nombre del producto o recurso, y referencia al origen del producto o recurso.
- Salidas. Nombre, descripción y características del producto o recurso, y referencia al destinatario del producto o recurso.
- Productos internos. Nombre, descripción y características de los productos generados y utilizados en el propio proceso.
- Referencias bibliográficas. Bibliografía que sustenta el proceso: normas, modelos de referencia, libros y otras fuentes.

Los componentes de las Prácticas de los procesos son:

- Roles involucrados y capacitación. Identificación de roles involucrados y capacitación requerida.
- Actividades. Se asocian a los objetivos y describen las tareas y roles responsables.
- Diagrama de flujo de trabajo. Diagrama de actividades de UML, donde se especifican las actividades del flujo de trabajo y los productos.
- Verificaciones y validaciones. Se definen las verificaciones y validaciones asociadas a los productos generados en las actividades que se mencionan.
- Incorporación a la base de conocimiento. Incluye nombre del producto y además la forma de aprobación.
- Recursos de infraestructura. Los recursos requeridos para actividades o tareas.
- Mediciones. Mediciones que se establecen para evaluar los indicadores del proceso.
- Capacitación. Definición de las reglas para proporcionar la capacitación necesaria a los roles involucrados en el proceso.
- Situaciones excepcionales. Definición de los mecanismos para el manejo de las situaciones excepcionales durante la ejecución del proceso.
- Lecciones aprendidas. Definición de los mecanismos para aprovechar las lecciones aprendidas durante la ejecución del proceso.

El componente de las Guías de ajuste es:

- Guía de ajuste. Descripción de posibles modificaciones al proceso que no deben afectar los objetivos del mismo.

### 2.2.2.2. Niveles

Los niveles de capacidad de procesos considerados por el modelo MoProSoft se muestran en la Tabla 4.

**Tabla 4.** Niveles de capacidad de MoProSoft [NYCE05].

Nivel	Nombre	Descripción
0	Incompleto	El proceso no está implantado o falla en el alcanzar el propósito del proceso.
1	Realizado	El proceso implantado logra su propósito y obtiene los resultados definidos.
2	Administrado	El proceso Realizado se implanta de manera administrada y sus productos de trabajo están apropiadamente establecidos, controlados y mantenidos.
3	Establecido	El proceso Administrado es implantado mediante el proceso definido, el cual es capaz de lograr los resultados del proceso.
4	Predecible	El proceso Establecido opera dentro de los límites para lograr sus resultados.
5	Optimizado	El proceso Predecible es continuamente mejorado para lograr las metas de negocios actuales y futuras relevantes.

En MoProSoft se utilizan los niveles para describir el nivel de capacidad de los procesos de la organización. Para realizar la evaluación y definir el nivel de capacidad de los procesos de una empresa existe el método de Evaluación de Procesos para la Industria Software (EvalProSoft), el cual fue creado basado en las recomendaciones de ISO/IEC15504 (Parte 2).

### 2.2.2.3. Categorías de procesos

MoProSoft contiene procesos divididos en tres categorías: Alta Dirección, Gerencia y Operación. Estas tres categorías buscan reflejar las estructuras de cualquier organización. Este modelo cuenta con 9 procesos dentro de las 3 categorías arriba mencionadas, y estas repartidas como se muestra en la Tabla 5.

**Tabla 5.** Estructura de MoProSoft [NYCE05].

Categoría	Procesos
Alta Dirección (DIR)	DIR. 1 Gestión de Negocio
Gerencia (GER)	GES. 1 Gestión de Procesos
	GES. 2 Gestión de Proyectos
	GES. 3 Gestión de Recursos
	GES. 3.1 Recursos Humanos y Ambiente de Trabajo
	GES. 3.2 Bienes, Servicios e Infraestructura
	GES. 3.3 Conocimiento de la Organización
Operación (OPE)	OPE.1 Administración de Proyectos Específicos
	OPE. 2 Desarrollo y Mantenimiento de Software

En cada proceso están definidos los roles responsables por la ejecución de las prácticas. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y formación para desempeñarlos.

#### 2.2.2.3.1. Alta Dirección

Categoría de procesos que aborda las prácticas de la Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de *Gerencia* y se retroalimenta con la información generada por ellos.

La categoría de *Alta Dirección* la compone un único proceso:

- Gestión de Negocio.

#### 2.2.2.3.2. Gerencia

Categoría de procesos que aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de *Operación*, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de *Alta Dirección*. Esta categoría está compuesta por los siguientes procesos:

- Gestión de Procesos.
- Gestión de Proyectos.
- Gestión de recursos.

Esta última categoría se divide está compuesta por tres procesos más:

- Recursos Humanos y Ambientes de Trabajo.
- Bienes, Servicios e Infraestructura.
- Conocimiento de la Organización.

#### 2.2.2.3.3. Categoría de Operación

Categoría de procesos que aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de *Gerencia* y entrega a ésta la información y productos generados.

- Administración de Proyectos Específicos.
- Desarrollo y Mantenimiento de Software.

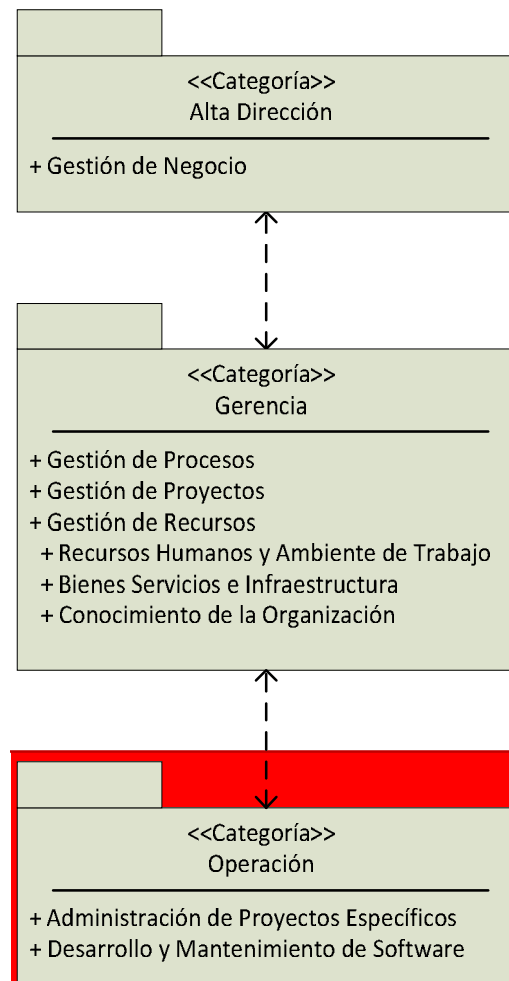
#### 2.2.2.4. *Procesos involucrados en el desarrollo y mantenimiento de Software*

Los procesos involucrados para el desarrollo y mantenimiento de software en MoProSoft se encuentran dentro de la categoría de Operación (OPE) (ver Figura 2.5), la cual está compuesta por los procesos de Administración de Proyectos Específicos (APE) y Desarrollo y Mantenimiento de Software (DMS).

El propósito de APE es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costos esperados, mediante la coordinación y el manejo de los recursos del mismo. APE también busca mantener informado al cliente mediante la realización de reuniones de avance del proyecto, además de que permite atender las Solicitudes de Cambio del cliente mediante la recepción y análisis de las mismas. APE aplica los conocimientos, las habilidades, las técnicas y herramientas, a las actividades del proyecto de Planificación, Realización, Evaluación y Control, y Cierre.

El propósito de DMS es la realización sistemática de las actividades de Inicio, Requisitos, Análisis y Diseño, Construcción, Integración y Pruebas, y Cierre para productos nuevos o

modificados de software, cumpliendo con los Requisitos especificados. DMS busca lograr que los productos de salida sean consistentes con los productos de entrada en cada fase de un ciclo de desarrollo mediante las actividades de verificación, validación o prueba. Así, DMS debe poder sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros mediante la integración de la Configuración de Software del ciclo actual.



**Figura 2.5.** Áreas de proceso de ingeniería [NYCE05]

En este sentido, es en DMS donde se llevan a cabo las actividades de las fases de un ciclo mediante el cumplimiento del Plan de Desarrollo del proyecto.

### 2.2.3. ISO/IEC 12207

La norma ISO/IEC 12207 fue publicada por primera vez por la ISO (Organización Internacional para la Estandarización, por sus siglas en inglés) y por la IEC (Comisión Electrotécnica Internacional, por sus siglas en inglés) en el año 1995, y actualmente en su versión del 2008 tiene 43 procesos para el desarrollo y mantenimiento de software (ver Figura 2.6).

ISO/IEC 12207 busca ser el estándar que defina todas las tareas y resultados requeridos por las actividades y procesos para el desarrollo y mantenimiento de software.

### 2.2.3.1. Estructura

Los procesos del estándar ISO/IEC 12207 son descritos buscando facilitar el uso conjunto con el estándar ISO/IEC 15288 en una sola organización o proyecto, y es por eso que es redactado de manera similar a éste.

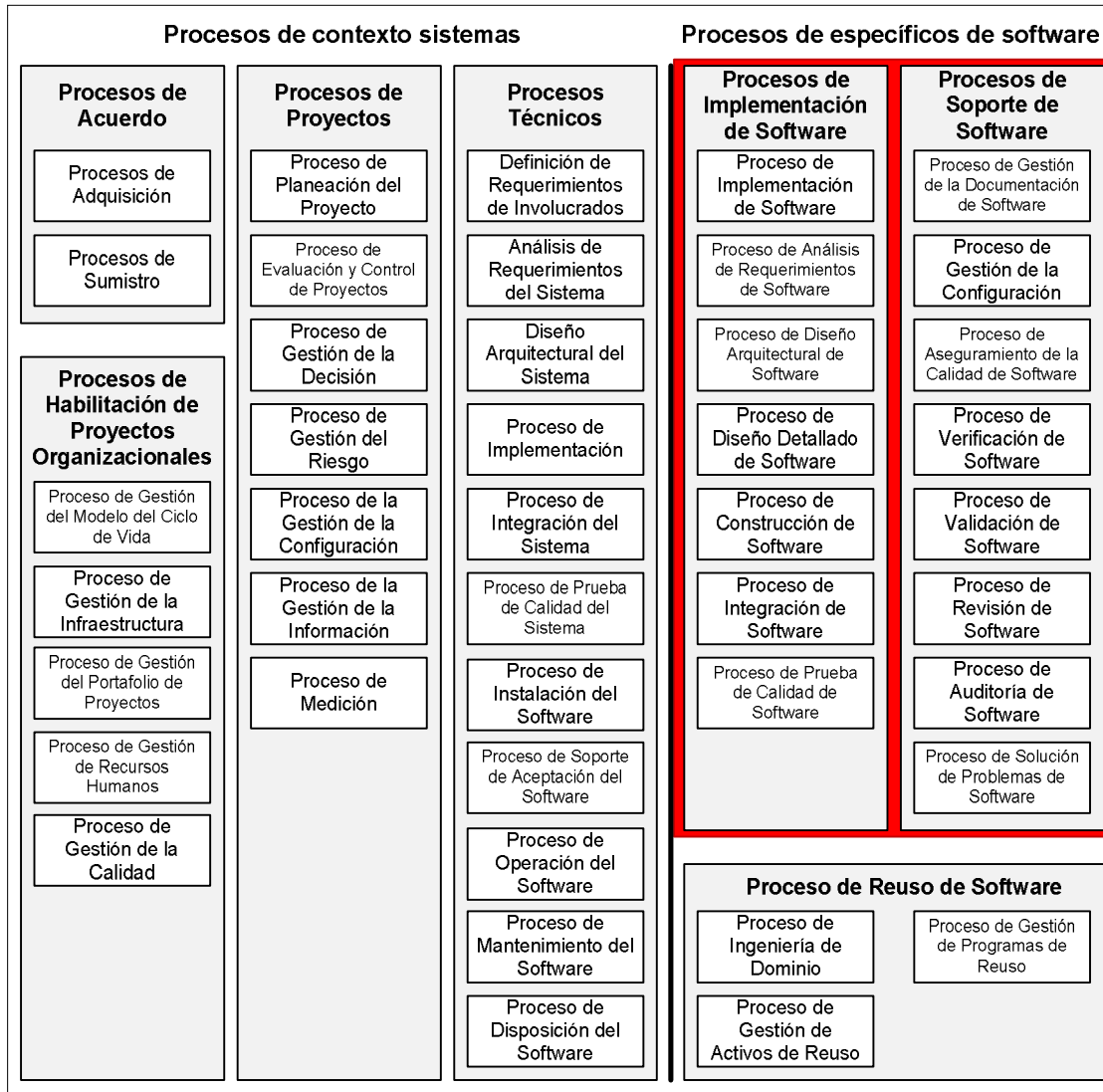


Figura 2.6. Grupos de proceso del ciclo de vida [ISO08]

Cada proceso está descrito en términos de los siguientes atributos:

- Título. Expresar el alcance del proceso en su conjunto.
- Propósito. Describe las metas de llevar a cabo el proceso.
- Salidas. Expresan los resultados observables esperados en la realización exitosa de del proceso.
- Actividades. Lista de acciones que son usadas para conseguir las salidas.

- Tareas. Lista de tareas correspondientes a una actividad, y pueden ser Requisitos, recomendaciones, o acciones permitidas que intentan auxiliar en la consecución de las salidas.

### 2.2.3.2. Niveles

Para evaluar el nivel de capacidad de las organizaciones que utilizan el modelo ISO/IEC 12207, se puede mapear dicho modelo al modelo internacional ISO/IEC 15504 [ISO04], y así poder emplear sus 6 niveles de capacidad de procesos, los cuales son descritos en la Tabla 4.

### 2.2.3.3. Categorías de procesos

ISO/IEC 12207 agrupa las actividades que pueden ser realizadas durante el ciclo de vida de un sistema de software dentro de 7 grupos de procesos: Procesos de acuerdo, Procesos de habilitación de proyectos organizacionales, Procesos de proyecto, Procesos técnicos, Procesos de implementación de software, Procesos de soporte de software, Procesos de reúso de software.

La forma en que se agrupan los 43 procesos del ciclo de vida en los 7 grupos de procesos se pueden ver en la Figura 2.6.

#### 2.2.3.3.1. Procesos de acuerdo

Estos procesos definen las actividades necesarias para establecer un acuerdo entre dos partes de la organización. Si el proceso de adquisición es invocado, éste proporciona los medios para dirigir los negocios con un proveedor externo de productos (los cuales forman parte de un sistema operacional), o servicios que apoyan a un sistema operacional ya existente, o elementos de un sistema que está siendo desarrollado por un proyecto. Si el proceso proveído es invocado, éste provee los medios para conducir un proyecto en el cual el resultado es un producto o servicio que es entregado al comprador. Los *procesos de acuerdo* consisten en:

- Procesos de Adquisición.
- Procesos de Suministro.

#### 2.2.3.3.2. Procesos de habilitación de proyectos organizacionales

Los *procesos de habilitación de proyectos organizacionales* administran la capacidad de la organización para adquirir y proveer productos o servicios a través de la iniciación, soporte y control de proyectos. Estos procesos proveen recursos e infraestructura necesaria para dar soporte a proyectos y asegurar la satisfacción de los objetivos organizacionales y acuerdos establecidos. Los *procesos de habilitación de proyectos organizacionales* consisten en:

- Proceso de Gestión del Modelo del Ciclo de Vida.
- Proceso de Gestión de Infraestructura.
- Proceso de Gestión de Portafolio de Proyecto.
- Proceso de Gestión de Recursos Humanos.
- Proceso de Gestión de Calidad.

#### 2.2.3.3.3. Procesos de proyecto

En ISO/IEC 12207, los proyectos han sido escogidos como el contexto para describir procesos concernientes con la planeación, la evaluación y el control. Los principios relativos a estos procesos pueden ser aplicados en cualquier área de una administración de la organización.

Existen dos categorías de *procesos de proyectos*: *los procesos de administración de proyectos* son usados para planear, ejecutar, evaluar y controlar el progreso de un proyecto; mientras que los *procesos de soporte de proyectos* dan soporte especializado a los objetivos de la administración. Los *procesos de administración de proyectos* son usados para establecer y evolucionar los planes de proyectos, para evaluar logros y progresos actuales contra lo planeado, y para controlar la ejecución de los proyectos a través de su cumplimiento. Los *procesos de administración de proyectos* pueden ser invocados individualmente en cualquier momento del ciclo de vida y en cualquier nivel en una jerarquía de proyectos, de acuerdo a como sean requeridos por los planes de proyectos o por eventos imprevistos. Los *procesos de administración de proyectos* son aplicados con un nivel de rigor y formalidad que depende de los riesgos y complejidad del proyecto. Estos procesos son:

- Proceso de Planeación de Proyectos.
- Proceso de Evaluación y Control de Proyectos.

Los *procesos de soporte de proyectos* proveen un conjunto específico enfocado de tareas para desempeñar un objetivo de administración especializado. Todos estos son evidentes en la administración de cualquier empresa, que van desde una organización completa hasta un proceso del ciclo de vida y sus tareas. Los *procesos de soporte de proyectos* son:

- Proceso de Gestión de Decisión.
- Proceso de Gestión de Riesgos.
- Proceso de Gestión de la Configuración.
- Proceso de Gestión de la Información.
- Proceso de Medición.

#### 2.2.3.3.4. Procesos técnicos

Los procesos técnicos son usados para definir los Requisitos de un sistema, para transformar los Requisitos en un producto efectivo, para permitir la reproducción consistente de un producto donde sea necesario, para proveer servicios requeridos, para sustentar la provisión de estos servicios, y para disponer de los productos cuando estos sean retirados del servicio. Los procesos técnicos definen las actividades que habilitan funciones organizacionales y de proyectos para optimizar los beneficios y reducir los riesgos que surgen de las decisiones y acciones técnicas. Estas actividades habilitan a los productos y servicios para poseer la oportunidad y disponibilidad, la efectividad, y la funcionalidad, fiabilidad, mantenibilidad, producibilidad, usabilidad y otras cualidades requeridas para la adquisición y provisión de organizaciones. También habilitan productos y servicios para ajustarse a las expectativas o Requisitos establecidos por la sociedad, incluyendo salud, cuidado, seguridad y factores del entorno.

Los procesos técnicos consisten de los procesos siguientes:

- Definición de Requisitos de Involucrados.



- Análisis de Requisitos del Sistema.
- Diseño Arquitectural del Sistema.
- Proceso de Implementación.
- Proceso de Integración del Sistema.
- Proceso de Prueba de Calidad del Sistema.
- Proceso de Instalación del Software.
- Proceso de Soporte de Aceptación del Software.
- Proceso de Operación del Software.
- Proceso de Mantenimiento del Software.
- Proceso de Disposición del Software.

#### 2.2.3.3.5. Procesos de implementación de software

Los *procesos de implementación de software* son usados para producir un elemento de sistema especificado (ítem de software) implementado en software. Estos procesos transforman comportamientos especificados, interfaces e implementaciones de restricciones dentro de acciones de implementación que resultan en un elemento de sistema que satisface los Requisitos derivados de los Requisitos del sistema.

Los *procesos de implementación de software* tienen varios procesos de bajo nivel específicos de software:

- Proceso de Análisis de Requisitos de Software.
- Proceso de Diseño Arquitectural de Software.
- Proceso de Diseño Detallado de Software.
- Proceso de Construcción de Software.
- Proceso de Integración de Software.
- Proceso de Prueba de Calidad de Software.

#### 2.2.3.3.6. Procesos de soporte de software

Los *procesos de soporte de software* proveen un conjunto específico enfocado a actividades para desempeñar procesos de software especializado. Un proceso de soporte asiste al proceso de implementación de software como una parte integral con un propósito distinto, contribuyendo al éxito y calidad del proyecto de software. Estos procesos son ocho y se listan a continuación:

- Proceso de Gestión de Documentación de Software.
- Proceso de Gestión de Configuración de Software.
- Proceso de Aseguramiento de la Calidad de Software.
- Proceso de Validación de Software.
- Proceso de Verificación de Software.

- Proceso de Revisión de Software.
- Proceso de Auditoría de Software.
- Proceso de Solución de Problemas de Software.

#### 2.2.3.3.7. Procesos de reúso de software

El grupo de *procesos de reúso de software* consiste en tres procesos que dan soporte a una capacidad de la organización para reusar ítems de software más allá de los límites del proyecto. Estos procesos son únicos porque, por su naturaleza, son operados fuera de los límites de un proyecto particular.

Los *procesos de reúso de software* son:

- Proceso de Ingeniería del Dominio.
- Proceso de Gestión de Activos de Reúso.
- Proceso de Gestión de Programas de Reúso.

#### 2.2.3.4. *Procesos involucrados en el desarrollo y mantenimiento de software*

Los procesos involucrados para el desarrollo y mantenimiento de software en el estándar ISO/IEC 12207 se encuentran dentro de las categorías de *procesos de implementación de software* y *procesos de soporte de software* las cuales están compuestas por los procesos mostrados en la Figura 2.5. El principal, el proceso de implementación de software, tiene como objetivo producir un sistema específico implementado como un producto o servicio de software. También transforma comportamientos, interfaces y restricciones de implementación específicas, en acciones para crear un elemento del sistema implementado como un producto o servicio de software, también conocidos como ítems de software. Estos procesos resultan en un ítem de software que satisface los Requisitos del diseño arquitectural a través de la verificación, y los Requisitos de los involucrados a través de la validación. Los otros seis procesos dentro de los *procesos de implementación de software* son considerados subprocesos; específicamente son los siguientes:

- El *proceso de análisis de Requisitos* que busca establecer los Requisitos de los elementos de software de sistema.
- El *proceso de diseño arquitectural de software* que pretende proveer un diseño para el software para que pueda ser implementado y verificado contra los Requisitos.
- El *proceso de diseño detallado de software* que proporciona un diseño para el software que pueda implementarse y verificarse contra los Requisitos y la arquitectura del software, y que además sea suficientemente detallada para permitir codificar y probar.
- El *proceso de construcción de software* que permite producir unidades de software ejecutables que reflejen apropiadamente el diseño de software.
- La combinación de unidades y componentes de software se lleva a cabo en el *proceso de integración de software*, también produce ítems de software integrados, consistentes con el diseño de software, que demuestra que los Requisitos funcionales y no funcionales son satisfechos sobre una plataforma operacional completa o equivalente.
- El último proceso de los procesos de bajo nivel correspondientes a los *procesos de implementación de software* es el *proceso de prueba de calidad de software*, y tiene como

propósito confirmar que el producto de software integrado coincide con los Requisitos definidos.

Por otro lado, los *procesos de soporte de software* tienen el rol integral de asistir a los *procesos de implementación de software*. Concretamente, estos procesos son los siguientes:

- El *proceso de gestión de la documentación del software* es una especialización del *proceso de gestión de información*, y tiene como propósito desarrollar y mantener los registros de la información de software producida por un proceso.
- El *proceso de gestión de la configuración de software* es una especialización del *proceso de gestión de la configuración*, y tiene el propósito de establecer y mantener la integridad de los ítems de software de un proceso o un proyecto, y hacer que estos estén disponibles para los grupos adecuados.
- El propósito del *proceso de aseguramiento de la calidad de software* es asegurar que los productos funcionen y los procesos cumplan con las provisiones y planes predefinidos.
- El *proceso de verificación de software* busca confirmar que cada producto y/o servicio de trabajo de software de un proceso o proyecto refleje apropiadamente los Requisitos especificados.
- El *proceso de validación de software* tiene como propósito confirmar que los Requisitos para un uso específico previsto de un producto de trabajo de software sean cumplidos.
- El *proceso de revisión de software* mantiene un entendimiento común con los involucrados del progreso contra los objetivos del acuerdo y que podrían ser hechos para ayudar a asegurar el desarrollo de un producto que satisfaga a los involucrados. Las revisiones de software son a nivel de gestión del proyecto y técnico, y se lleva a cabo durante todo el ciclo de vida del proyecto.
- El propósito del *proceso de auditoría de software* es determinar de forma independiente el cumplimiento de determinados productos y procesos con los Requisitos, los planes y el acuerdo, según sea el caso.
- El último proceso, de los procesos de soporte de software, es el *proceso de solución de problemas de software*, y tiene el propósito de asegurar que todos los problemas descubiertos sean identificados, analizados, gestionados y controlados para lograr una solución.

#### 2.2.4. Comparativa de los modelos de referencia

Tomando en consideración los resultados obtenidos en el estudio realizado por [Gutiérrez08] sobre las características de las empresas mexicanas y su relación con los modelos de referencia, las características a evaluar en los modelos presentados son:

- Adaptable al desarrollo de software a la medida. Una de las características de la industria mexicana de software es que desarrollan mucho software a la medida, es por eso que el modelo de referencia adecuado para la industria mexicana debe permitir la realización de este tipo de software.
- Facilita el desarrollo iterativo e incremental. Otra de las características de la industria mexicana es que se hacen desarrollos iterativos y entregas de prototipos funcionales a los clientes, por esta razón el modelo debe considerar un desarrollo iterativo e incremental para que se ajuste al entorno nacional.

- Incluye la definición de plantillas/formatos. Un problema que presenta la industria mexicana es la baja calidad de la documentación generada durante el proceso desarrollo. Es por eso que el modelo debería tener definidos los documentos que generará su proceso de desarrollo, para servir de guía a los interesados.
- Tolerante a cambios continuos en los requisitos del producto. Al considerar un desarrollo a la medida es muy probable que el cliente cambie los Requisitos en algún punto del proceso de desarrollo. Por ello el modelo de referencia debe considerar cambios en los Requisitos del producto.
- Promueve la participación del cliente en el desarrollo. El modelo de referencia debe considerar la participación del cliente para lograr una mejor satisfacción del cliente sobre el producto software terminado.
- Establece un tiempo fijo para los ciclos de desarrollo. El tiempo de duración de los proyectos software en la industria mexicana son periodos cortos de tiempo (de 3 a 5 meses), por lo tanto el modelo de referencia debe considerar tiempos de desarrollo fijos, y enfocarse a la entrega de un producto funcional con valor de negocio, haya cumplido o no con todos los Requisitos del cliente.
- Orientado a empresas pequeñas. El modelo debe estar diseñado para trabajar con empresas pequeñas, debido a que en la industria mexicana de software la mayoría son empresas de este tipo.
- Útil para equipos pequeños. En México el software es desarrollado, en su mayoría, por equipos de desarrollo pequeños de entre 5 a 10 personas, por esta razón el modelo de referencia debe considerar este aspecto.

En la Tabla 6 se muestra el resultado de evaluar los modelos de referencia CMMI DEV v1.3, MoProSoft e ISO/IEC 12207, basándonos en las características arriba mencionadas.

Si se considera que cada una de las características arriba mencionadas tienen la misma importancia, y en base a la tabla comparativa de características que cumplen los modelos de referencia, se obtiene que:

- CMMI-DEV v1.3 cumple en un 37.5%.
- MoProSoft cumple en un 81.25%, y
- ISO/IEC 12207 cumple en un 50%

Es decir, MoProSoft cumple con un porcentaje mayor de las características que un modelo de referencia requiere para cubrir las necesidades de la industria mexicana de software.

**Tabla 6.** Características evaluadas en los modelos de referencia analizados.

Características	CMMI-DEV v1.3	MoProSoft	ISO/IEC 12207
Desarrollo de software a la medida	Si	Si	Si
Desarrollo iterativo e incremental	No	Iterativo únicamente	No
Documentos definidos	No	Si	Si
Procesos de cambios de Requisitos del producto	Si	Si	Si
Participación del cliente en el desarrollo	Si	Si	Si
Tiempo fijo para los ciclos de desarrollo	No	No	No
Empresas pequeñas	No	Si	No
Equipos pequeños	No	Si	No

## 2.3. Métodos ágiles

El término ágil se formalizó en febrero de 2001 [URL-4] cuando representantes de la Programación Extrema, de Scrum, de DSDM, del Desarrollo de Software Adaptativo, de Crystal, del Desarrollo Dirigido por Características, de la Programación Pragmática, y otros especialistas firmaron el manifiesto ágil. El motivo de esta reunión fue la necesidad de identificar una alternativa para los procesos de desarrollo de software tradicionales, que además eran dirigidos por la documentación.

Las metodologías ágiles buscan minimizar los riesgos de desarrollo mediante iteraciones de duraciones cortas, y de la misma forma que las otras metodologías formales incluyen planificación, análisis de Requisitos, diseño, codificación, revisión y documentación, dentro del ciclo de vida de una iteración. A continuación se presentan las dos metodologías ágiles más conocidas por empresas mexicanas, de acuerdo con [Gutiérrez08].

### 2.3.1. Programación extrema

Este enfoque de desarrollo de software fue propuesto por Beck y Andres [Beck04] como un conjunto de mejores prácticas que se aplican de manera dinámica durante el ciclo de vida del software.

#### 2.3.1.1. Valores y principios

La Programación Extrema (XP, *Extreme Programming*) está basada en valores. Las reglas son la extensión y consecuencia natural de maximizar estos valores. XP más que un conjunto de reglas, es una forma de trabajar en armonía con el personal y los valores corporativos. Los valores a los cuales obedece esta metodología son:

- **Simplicidad:** Hacer solo lo necesario y que fue solicitado, y no hacer de más. Esto maximizará el valor creado por la inversión hecha hasta la fecha. Tomar pasos simples y pequeños para lograr la meta y mitigar las fallas que podrían ocurrir. Crear algo de lo que estar orgullosos y mantenerlo a largo plazo con un costo razonable.
- **Comunicación:** Todos son parte de un equipo y se comunican cara a cara diariamente. Trabajan juntos en todas las cosas, desde Requisitos hasta codificación. Juntos deben crear la mejor solución para un problema.

- **Realimentación:** Se debe tomar seriamente el compromiso de la iteración para entregar software funcional. Se debe mostrar el software temprana y frecuentemente, y escuchar cuidadosamente y hacer los cambios necesarios. También se debe obtener retroalimentación sobre el proyecto para poder adaptar el proceso al proyecto.
- **Respeto:** Todos deben dar y sentir el respeto que merecen como un miembro valioso del equipo. Todos aportan valor aún si es solamente entusiasmo. Los desarrolladores deben respetar la experiencia de los clientes y viceversa. Los administradores deben respetar el derecho del equipo de aceptar responsabilidad y recibir autoridad sobre su propio trabajo.
- **Coraje:** Siempre se debe decir la verdad acerca del progreso y las estimaciones. No se documentarán excusas sobre razones de los fallos debido a que se planeó tener éxito. No se debe temer nada porque ninguno trabajará solo. Se debe adoptar al cambio cada vez que éste ocurra.

XP mejora un proyecto de software en 5 formas esenciales: comunicación, simplicidad, retroalimentación, respeto, y coraje. Los programadores extremos se comunican con sus clientes y colegas programadores. El diseño se mantiene simple y limpio. Se obtiene retroalimentación al probar el software desde el primer día. Se entrega el sistema a los clientes tan pronto como sea posible y se implementan los cambios que se sugieran. Cada éxito pequeño profundiza el respeto por las contribuciones únicas de cada miembro del equipo. Con estos fundamentos los programadores extremos son capaces de responder con coraje a los cambios en los Requisitos y la tecnología.

### 2.3.1.2. Contenido

El aspecto más sorprendente de XP son sus reglas sencillas (ver Figura 2.7), las cuales pueden parecer pesadas y hasta ingenuas, incluso al principio, pero se basan en valores y principios sólidos.

<b>Planeación</b> <ul style="list-style-type: none"> <li>• Las historias de usuario se escriben</li> <li>• La planeación de lanzamiento crea la programación de lanzamiento</li> <li>• Frecuentes lanzamientos pequeños</li> <li>• El proyecto se divide en iteraciones</li> <li>• La planeación de la iteración inicia cada iteración</li> </ul>	<b>Administración</b> <ul style="list-style-type: none"> <li>• Dar al equipo un espacio de trabajo abierto dedicado</li> <li>• Establecer un ritmo sostenible</li> <li>• Cada día inicia con una reunión de pie</li> <li>• La Velocidad del Proyecto es medible</li> <li>• La gente se mueve alrededor</li> <li>• Arreglar XP cuando se rompa</li> </ul>	<b>Diseño</b> <ul style="list-style-type: none"> <li>• Simplicidad</li> <li>• Escoger una metáfora para el sistema</li> <li>• Usar cartas Clase-Responsabilidades-Colaboración</li> <li>• Crear soluciones de punta para reducir el riesgo</li> <li>• Ninguna funcionalidad es agregada tempranamente</li> <li>• Refactorizar cuando y donde sea posible</li> </ul>
<b>Codificación</b> <ul style="list-style-type: none"> <li>• El cliente esta disponible siempre</li> <li>• El código debe estar escrito de acuerdo a los estándares</li> <li>• Codificar las pruebas unitarias primero</li> <li>• Todo el código producido debe estar programado en pares</li> <li>• Solo un par integra código por vez</li> <li>• Integración frecuente</li> <li>• Establecer una computadora para integración dedicada</li> </ul>	<b>Pruebas</b> <ul style="list-style-type: none"> <li>• Todo código debe tener pruebas unitarias</li> <li>• Todo código debe pasar todas las pruebas unitarias antes que poder ser lanzado</li> <li>• Cuando un error es encontrado se crean pruebas</li> <li>• Pruebas de aceptación son ejecutadas frecuentemente y su resultado es publicado</li> </ul>	

Figura 2.7. Reglas de la programación extrema [URL-6]

Las reglas establecen expectativas entre los miembros del equipo pero no con el objetivo final de cumplirlas. Estas reglas definen un entorno que promueve la colaboración en equipo y la capacitación, ese es su objetivo. Una vez conseguido el trabajo en equipo productivo este continuará aun cuando las reglas sean cambiadas para adaptarse a las necesidades específicas de cada empresa.

Las reglas de XP trabajan conjuntamente (ver Figura 2.8) durante un proyecto de programación extrema, en cada iteración del proyecto, en cada proceso de desarrollo de una Iteración, y durante la etapa de propiedad de código colectiva de un proceso de desarrollo.

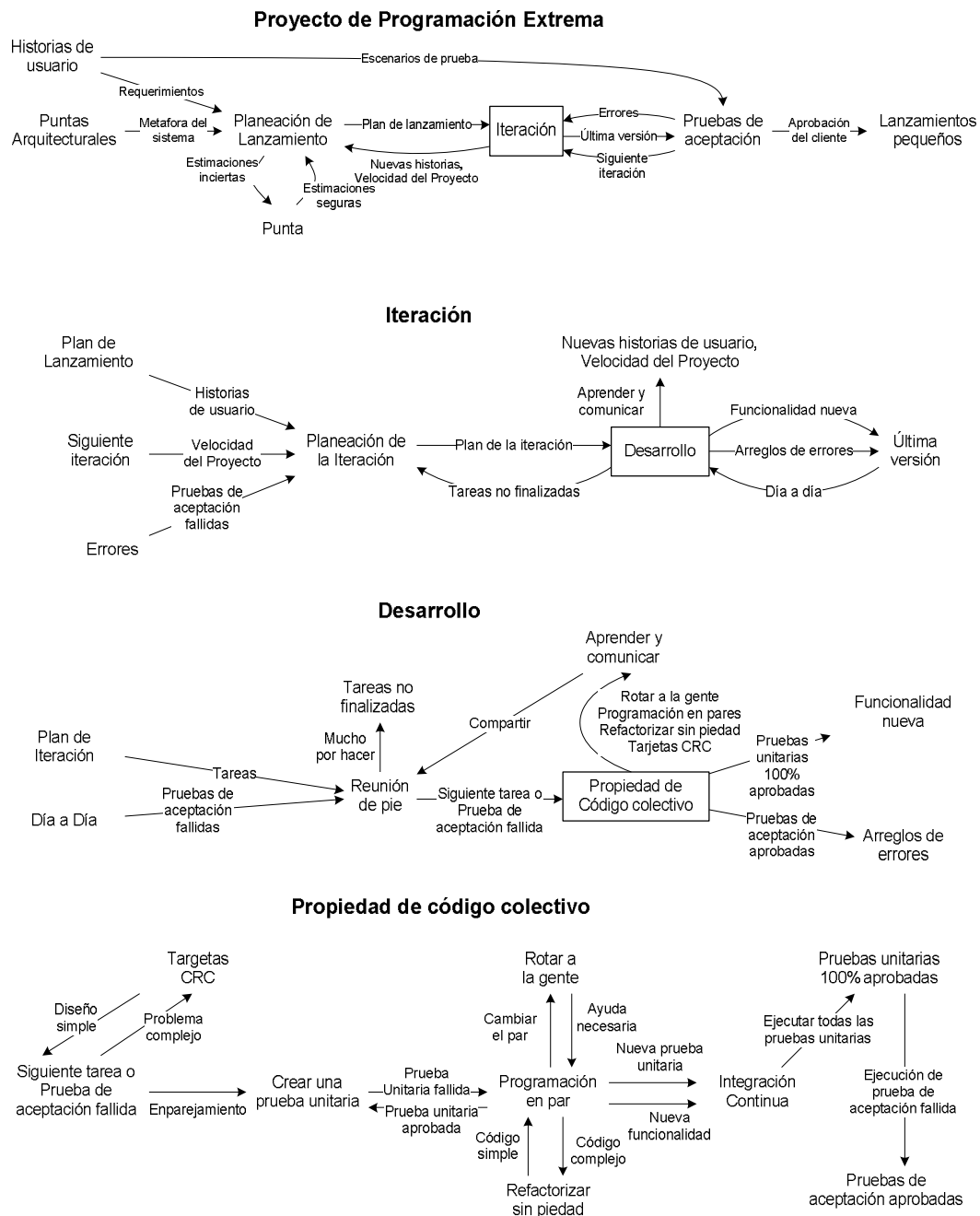


Figura 2.8. Flujo de trabajo de la programación extrema [URL-6]

### 2.3.2. Scrum

Scrum es un marco de trabajo con el cual las personas pueden tratar problemas adaptativos complejos, mientras que de forma productiva y creativa entregan productos del más alto valor posible. Scrum es ligero, sencillo de entender y extremadamente difícil de dominar [Schwaber10].

Así, Scrum es un marco de trabajo de procesos que han sido usados para gestionar el desarrollo de productos complejos desde el inicio de los 90s. Sin embargo, Scrum no es un proceso o técnica para construir productos; más bien es un método para un grupo de trabajo dentro del cual las personas pueden emplear varios procesos y técnicas. Scrum hace evidente la eficacia relativa de las prácticas de gestión y desarrollo de productos usados para que éstos puedan ser mejorados.

#### 2.3.2.1. Valores y principios

Scrum está basado en la teoría de control empírico de procesos, o empirismo. El empirismo asegura que el conocimiento viene de la experiencia y la toma de decisiones basadas en lo que ya es conocido. Scrum emplea una aproximación iterativa e incremental para optimizar la predictibilidad y el control de riesgos.

Cada implementación del control de procesos empírico está sostenida por tres pilares: transparencia, inspección, y adaptación. Estos pilares se pueden definir como:

- Transparencia. Aspectos significativos de un proceso que deben ser visibles por aquellos responsables de los resultados.
- Inspección. Los usuarios de Scrum deben inspeccionar con frecuencia los artefactos de Scrum y avanzar hacia la meta para detectar variaciones indeseables.
- Adaptación. Si un inspector determina que uno o más aspectos de un proceso se desvía de los límites aceptables, y que el producto resultante será inaceptable, el proceso o el material a ser procesado debe ser ajustado. Un ajuste debe hacerse lo más pronto posible para evitar desviaciones adicionales.

Scrum prescribe cuatro oportunidades formales para la inspección y adaptación:

- Planeación del *sprint*.
- Reunión diaria.
- Revisión del *sprint*.
- Retrospectiva del *sprint*.

#### 2.3.2.2. Contenido

El marco de trabajo de Scrum consiste de equipos y roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito y uso de Scrum (ver Figura 2.9).

Un equipo Scrum consiste de un dueño de producto, el equipo de desarrollo, y un maestro de Scrum. Los equipos de Scrum son auto-organizados y tienen funcionalidades compartidas. Los equipos auto-organizados eligen la mejor manera de realizar su trabajo. Los equipos con funcionalidades compartidas tienen todas las capacidades necesarias para realizar su trabajo sin depender de otros que no forman parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, creatividad, y productividad.



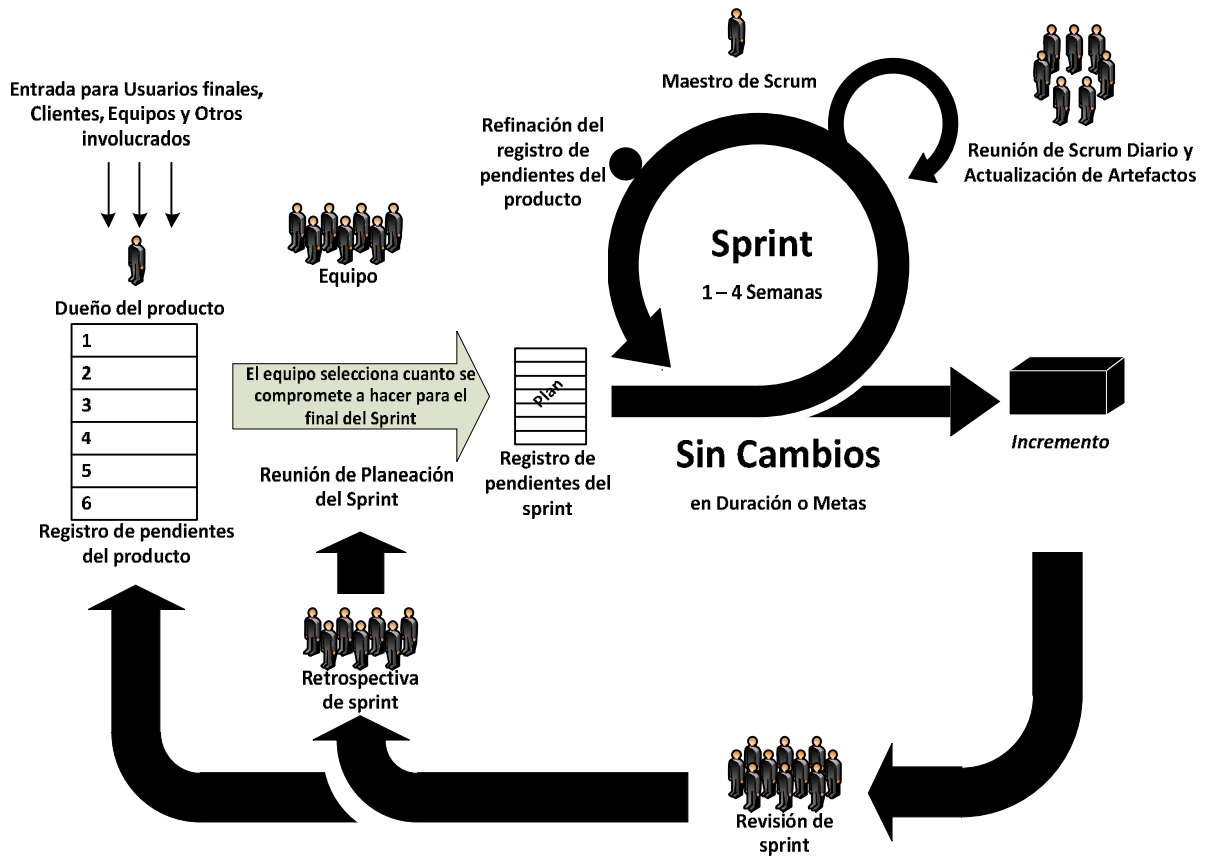


Figura 2.9. El proceso de Scrum [Sutherland10; Schwaber10]

Los equipos Scrum entregan productos iterativa e incrementalmente, maximizando las oportunidades para la realimentación. Las entregas incrementales del producto terminado aseguran siempre la disponibilidad de una versión potencialmente usable de producto funcional. Cada equipo de Scrum tiene tres roles: Equipo de Desarrollo, Dueño del Producto y Maestro de Scrum. Scrum usa eventos prescritos para crear regularidad y minimizar la necesidad de reuniones no definidas. Scrum usa eventos de tiempo fijo, de tal forma que cada evento tenga un tiempo de duración máximo; esto asegura que la cantidad apropiada de tiempo sea gastada planeando sin permitir pérdidas en el proceso de planeación.

Además del mismo *sprint*, el cual es un contenedor de todos los demás eventos, cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar algo. Estos eventos están específicamente diseñados para permitir la transparencia e inspección críticas. La falla en incluir cualquiera de estos eventos da como resultado una transparencia reducida y la pérdida de la oportunidad de inspeccionar y adaptar.

Por lo tanto, el corazón de Scrum es un *sprint*, un evento de tiempo fijo de un mes o menos durante el cual se crea un producto terminado, usable y potencialmente entregable. Los *sprints* tienen duraciones consistentes a lo largo del esfuerzo de desarrollo. Un nuevo *sprint* inicia inmediatamente después de la conclusión del *sprint* previo.

Los *sprints* contienen y consisten de:

- Planeación. El trabajo a realizarse en el *sprint* es planeado en esta reunión. El plan es creado por el trabajo colaborativo de un equipo de Scrum. Generalmente, la planeación consiste en responder a las siguientes dos preguntas:
  - ¿Qué deberá terminarse en este *sprint*?
  - ¿Cómo se lograra terminar el trabajo escogido?
- Reuniones diarias. Este evento tiene una duración máxima de 15 minutos y tiene como objetivo sincronizar las actividades y crear un plan para las siguientes 24 horas. Esto se logra mediante la inspección del trabajo realizado desde la última reunión diaria y pronosticando el trabajo que podría ser terminado antes de la siguiente.
- Revisión del *sprint*. Este evento se lleva a cabo al final del *sprint* para inspeccionar el incremento y adaptar el registro de pendientes del producto si es necesario. Durante la revisión del *sprint*, el equipo de Scrum y los interesados colaboran acerca de lo que fue hecho en el *sprint*. Esta es una reunión informal y la presentación del incremento tiene la intención de elicitar la realimentación y fomentar la colaboración.
- Retrospectiva del *sprint*. Este evento ocurre después de la revisión del *sprint* y antes de la planeación del *sprint* siguiente. El propósito de este evento es:
  - Inspeccionar cómo fue el último *sprint* en el sentido de personas, relaciones, procesos y herramientas.
  - Identificar y ordenar los elementos importantes que fueron bien hechos y que son potencialmente mejorables.
  - Crear un plan para implementar mejoras a la forma en que el equipo de Scrum hace su trabajo.

Los artefactos de Scrum representan el trabajo o valor de varias maneras que son útiles para proveer transparencia y oportunidades de inspección y adaptación. Los artefactos definidos por Scrum están específicamente diseñados para maximizar la transparencia de la información importante, necesaria para asegurar que los equipos de Scrum sean exitosos en la entrega del incremento terminado. En este sentido, los artefactos de Scrum son los siguientes:

- Registro de pendientes del producto. Este artefacto es una lista ordenada de todo lo que puede ser necesario en el producto y es una fuente sencilla de Requisitos para cualquier cambio que sea hecho al producto. Este artefacto es dinámico, constantemente cambiado para identificar que necesita el producto para ser apropiado, competitivo y usable. Así, este registro es una lista de todas las características, funciones, Requisitos, mejoras y arreglos que constituyen los cambios que se harán al producto en futuras entregas.
- Registro de pendientes del *sprint*. Este artefacto es el conjunto de ítems seleccionados del registro de pendientes del producto para el *sprint*, además de un plan para la entrega del incremento del producto y para la realización de la meta del *sprint*. Por lo tanto, es una predicción para el equipo de desarrollo acerca de cuáles funcionalidades estarán en el siguiente incremento y el trabajo necesario para entregarlas.
- Incremento. El incremento es la suma de todos los ítems del registro de pendientes del producto completados durante un *sprint* y todos los anteriores. Al final de un *sprint*, el nuevo incremento debe estar terminado, lo que significa que debe estar en condición de usarse y

cumplir con la definición de ‘terminado’ del equipo de Scrum. El producto debe estar en condición de usarse sin tener en cuenta si el dueño del producto decide liberarlo o no.

Cuando un ítem del registro de pendientes del producto o un incremento es descrito como terminado, todos deben entender qué significa terminado. Aunque esto varía significativamente en cada equipo de Scrum, los miembros deben tener un entendimiento compartido de lo que esto significa para que el trabajo sea completado, y asegurar la transparencia. Así, ésta es la definición de terminado para el equipo de Scrum y es usada para evaluar cuando el trabajo es completado en cada incremento del producto.

Esta misma definición guía al equipo de desarrollo para saber cuántos ítems del registro de pendientes del producto pueden ser seleccionados durante una planeación de *sprint*. El propósito de cada *sprint* es liberar incrementos de funcionalidades potencialmente entregables que cumplan con la definición actual de terminado del equipo de Scrum.

Los roles, artefactos, eventos y reglas de Scrum son inmutables y aunque es posible implementar partes de Scrum, el resultado no es Scrum como tal. Tal y como se mencionó, Scrum solo funciona como un contenedor de otras técnicas, metodologías y prácticas. En la Figura 2.9 se observa el proceso Scrum en su totalidad, con sus eventos y artefactos descritos en esta sección.

### 2.3.3. Comparativa entre metodologías ágiles

Para comparar las metodologías ágiles, y ver cual se ajusta mejor a las características de la industria mexicana de software, se usan las mismas 8 características mostradas en la sección 2.2.4. Los resultados de esta comparativa empírica se muestran en la Tabla 7.

**Tabla 7.** Características evaluadas en los modelos ágiles analizados.

Características	XP	Scrum
Desarrollo de software a la medida	Si	Si
Desarrollo iterativo e incremental	Iterativo únicamente	Si
Documentos definidos	No	No
Procesos de cambios de Requisitos del producto	Si	Si
Participación del cliente en el desarrollo	Si	Si
Tiempo fijo para los ciclos de desarrollo	No	Si
Empresas pequeñas	Si	Si
Equipos pequeños	Si	Si

Observando los resultados de la Tabla 7, se puede afirmar que:

- XP cumple con un 68.75% de las características.
- Scrum cumple con las características en un 87.5%.

## 2.4. Trabajo relacionado

Por último, y de acuerdo a la búsqueda exploratoria realizada en este capítulo, se determinó que se han desarrollado varias herramientas para auxiliar en la implantación del modelo de

referencia MoProSoft y la metodología Scrum dentro de las empresas pero de forma individual; es decir, que aunque existen estas herramientas de apoyo, durante la investigación realizada no se encontró una herramienta que permitiera combinar el modelo MoProSoft y la metodología Scrum (ver Tabla 8).

**Tabla 8.** Resumen de herramientas para la adopción de Scrum.

<b>Nombre</b>	<b>Roles incluidos</b>	<b>Artefactos incluidos</b>	<b>Procesos incluidos</b>	<b>Otras características</b>
Banana Scrum	Dueño del producto. Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Gestión del sprint.	Lista de impedimentos. Burnup chart. Workload chart. Calendario del sprint. Gráfica burndown.
Ice Scrum	Dueño del producto. Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.		Gestión de impedimentos. Burnup chart. Gráfica burndown.
Mingle	Dueño del producto. Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Reunión diaria. Revisión del sprint.	Gráfica burndown.
Rally	Maestro Scrum. Equipo de desarrollo.	Pendientes del producto.	Planeación del sprint. Sprint.	Seguimiento de defectos.
Scrum Ninja	Dueño del producto. Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Gestión del sprint.	Gráfica burndown.
Scrum Works	Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Sprint. Reunión diaria.	Seguimiento de impedimentos. Gráfica burndown.
Sprintometer	Equipo de desarrollo.		Sprint.	Predicción de las desviaciones del sprint. Gráfica burndown.
Tackle	Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Gestión del sprint. Reunión diaria.	Reporte y rastreo del trabajo por día. Gráfica burndown.
Team Trick	Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Sprint.	Gráfica burndown.
Version One	Maestro Scrum. Equipo de desarrollo.	Pendientes del producto. Pendientes del sprint.	Planeación del sprint. Retrospectiva del sprint.	Reportes de carga de trabajo por miembro. Burnup chart. Reportes sobre la precisión de la estimación. Reportes de tendencias del esfuerzo. Gráfica burndown.

Dentro de las herramientas desarrolladas para auxiliar en la adopción de MoProSoft se encuentran Quali [URL-7], que proporciona mecanismos para gestionar procesos y controlar la documentación generada, fue presentada en AceleraProsoft en el 2005, pero actualmente no se encuentra disponible. Otra herramienta que se ha desarrollado para MoProSoft es fTIMoN [Valenzuela07], que permite coordinar la secuencia y dependencia de tareas y productos internos de entrada y salida, así mismo, dar a conocer entre los roles, los estados de las actividades. También existe asistenteHIM [Oktaba06], que permite proveer a los usuarios con información de sus responsabilidades, tareas a realizar y responsables de ellas, sugerencias de la manera de llevarlas a cabo, recordatorios de tareas pendientes y coordinación del trabajo con otros usuarios, todo de acuerdo con la norma MoProSoft. De manera similar, existen muchas herramientas desarrolladas para auxiliar la adopción de Scrum en las empresas. Estas herramientas permiten manejar los artefactos, roles y procesos tal como son definidos en Scrum, algunas de estas herramientas son: Banana-Scrum [URL-8], IceScrum [URL-9], Mingle [URL-10], Rally [URL-11], ScrumNinja [URL-12], ScrumWorks [URL-13], Sprintometer [URL-14], Tackle [URL-15], TeamTrick [URL-16], VersionOne [URL-17], y otros más.

Por otro lado, de acuerdo con Palacio y Ruata [Palacio11] en los últimos años se observa una tendencia a enriquecer los modelos basados en procesos con los beneficios de los métodos ágiles para lograr un conocimiento más completo. Tal es el trabajo expuesto por Pasini et al., [Pasini13] que presenta una adaptación de Scrum para satisfacer el estándar ISO/IEC 29110.

En relación a los trabajos referentes a MoProSoft y Scrum, se cuenta con dos referencias básicas: la primera es la investigación presentada por Rueda [Rueda10], la cual propone una guía para mapear las correspondencias de los procesos de la Categoría de Operación con las prácticas de las metodologías ágiles; el segundo esfuerzo propone la adopción de las prácticas de Scrum dentro del marco de trabajo definido por MoProSoft [Davila08], contribuyendo a ampliar los procesos de la organización, obteniendo así una actualización de MoProSoft.

Sin embargo, a diferencia del trabajo realizado por Dávila y Oktaba [Davila08] la presente tesis busca cumplir con la norma mexicana “Tecnología de la Información-Software-Modelos de procesos y de evaluación para desarrollo y mantenimiento de software” (NMX-I-059-NYCE-2005) sin modificarla.

## 2.5. Conclusiones del marco teórico

De los tres modelos de referencia expuestos en la sección anterior, MoProSoft tiene un porcentaje mayor (81.25%) de cobertura sobre las características comunes en la industria mexicana de software. Esto es lógico ya que fue diseñado para ello, sin embargo no cumple en un 100% con las características que hubiera sido lo ideal. Por el lado de las metodologías ágiles, Scrum cumple con un 87.5% de las características y es mayor al 68.75% que cumple XP. De manera individual Scrum cubre todas las características que la industria mexicana del software necesita.

La propuesta de mezclar el modelo de referencia MoProSoft con la metodología ágil Scrum pretende cubrir las carencias del modelo con las fortalezas de la metodología, y viceversa. Esta mezcla será una nueva implementación de modelo/metodología que podría satisfacer las necesidades de la industria mexicana de software.

Tomando en cuenta las características evaluadas en las comparativas anteriores, se compara la mezcla MoProSoft/Scrum contra la mezcla MoProSoft/XP, para conocer cuál de las dos podría tener más posibilidades de cubrir las características de la industria mexicana de software (ver Tabla 9).

**Tabla 9.** Características evaluadas en las mezclas MoProSoft/XP y MoProSoft/Scrum.

<b>Características</b>	<b>MoProSoft/XP</b>	<b>MoProSoft/Scrum</b>
Desarrollo de software a la medida	Si	Si
Desarrollo iterativo e incremental	Si	Si
Documentos definidos	Si	Si
Procesos de cambios de Requisitos del producto	Si	Si
Participación del cliente en el desarrollo	Si	Si
Tiempo fijo para los ciclos de desarrollo	No	Si
Empresas pequeñas	Si	Si
Equipos pequeños	Si	Si

Observando los resultados de la Tabla 9, se puede afirmar que:

- MoProSoft – XP podría cumplir con un 87.5% de las características necesarias
- MoProSoft – Scrum podría cumplir con las características en un 100%

Estos porcentajes muestran que la mezcla MoProSoft/Scrum podría cumplir idealmente con las características necesarias para la industria mexicana de software en su totalidad. En este sentido, el siguiente capítulo de la tesis establecerá una estrategia que permita combinar ambos modelos.

### **3. Planteamiento de la solución**

Como se pudo demostrar al final del Capítulo 2, la mezcla del modelo MoProSoft con la metodología Scrum permitirá cumplir con las 8 características de las empresas mexicanas de desarrollo de software expuestas en la Sección 2.2.4. En este sentido, a continuación se explica la elaboración del Diseño del Modelo MoProSoft – Scrum.

Primero, con el objetivo de facilitar la comprensión y el estudio de MoProSoft y Scrum, la sección 3.1 presenta la modelación de los procesos de MoProSoft de acuerdo con [NYCE05], y de Scrum de acuerdo con [Schwaber10].

En la sección 3.2 se presenta el análisis de procesos, actividades, roles y documentos de MoProSoft y Scrum, para identificar cómo y dónde Scrum encaja con MoProSoft.

Por último, la sección 3.3 presenta el resultado de cómo debe ser un proceso híbrido MoProSoft/Scrum, es decir, el diseño del nuevo patrón de procesos: “MoProSoft – Scrum”.

#### **3.1. Modelado de procesos**

Un proceso de negocio es un conjunto de actividades y/o tareas estructuradas para producir un servicio o producto específico para un cliente en particular. Un proceso de desarrollo de software puede definirse como un proceso de negocios, es decir “...es un conjunto de actividades y/o tareas estructuradas para producir un producto de software para un cliente en particular”.

La Notación de Modelado de Procesos de Negocios (BPMN, por sus siglas en inglés), es una notación estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo. Para realizar el análisis de los negocios de MoProSoft y de Scrum, primero se obtienen los modelos BPMN de MoProSoft y de Scrum, específicamente en la notación de la versión 2.0. La idea de usar una notación estándar para modelar los procesos de MoProSoft y Scrum es facilitar el análisis y diseño posteriores que emplearán los modelos presentados en esta sección.

En este sentido, [Freund12] explica un marco metodológico llamado BPMN-Framework, el cual permite seleccionar el tipo de objetos y patrones a utilizar o bien proporciona recomendaciones para no utilizar los incorrectos de acuerdo al nivel del marco de trabajo con el que se desee trabajar. Para el modelado, análisis y diseño del planteamiento de la solución se utilizará el “nivel 1 – Procesos Descriptivos”, el cual tiene como audiencia a los dueños de procesos, administradores de procesos, participantes de procesos y analistas de procesos de las primeras fases de un proyecto. En este nivel 1 se define el contexto de los procesos que se deben levantar, modelar, documentar y eventualmente rediseñar. El objetivo de este nivel es además validar el alcance y la funcionalidad principal de los procesos que deben levantarse. En este nivel se describe el flujo normal del proceso, cómo debe ocurrir, sin considerar casos de excepción o errores. El nivel descriptivo permite validar

en forma rápida el alcance del proyecto con los responsables del negocio e introducir al resto de los participantes en él.

En esta primera parte del planteamiento de la solución solamente se presentan los modelos de MoProSoft y Scrum; específicamente se proporciona un análisis sobre la forma en que estos modelos deben ser modelados usando el nivel 1 del BPMN-Framework [Freund12].

### 3.1.1. MoProSoft

#### 3.1.1.1. Procesos

En el caso particular de MoProSoft solamente se modela la Categoría de Operación (OPE), por razones que se explicaron en la sección 1.6, la cual incluye los procesos de Administración de Proyectos Específicos (OPE.1 APE) y Desarrollo y Mantenimiento de Software (OPE.2 DMS). Tomando como base a [NYCE05] es posible modelar el proceso de OPE.1 APE como se muestra en la Figura 2.9.

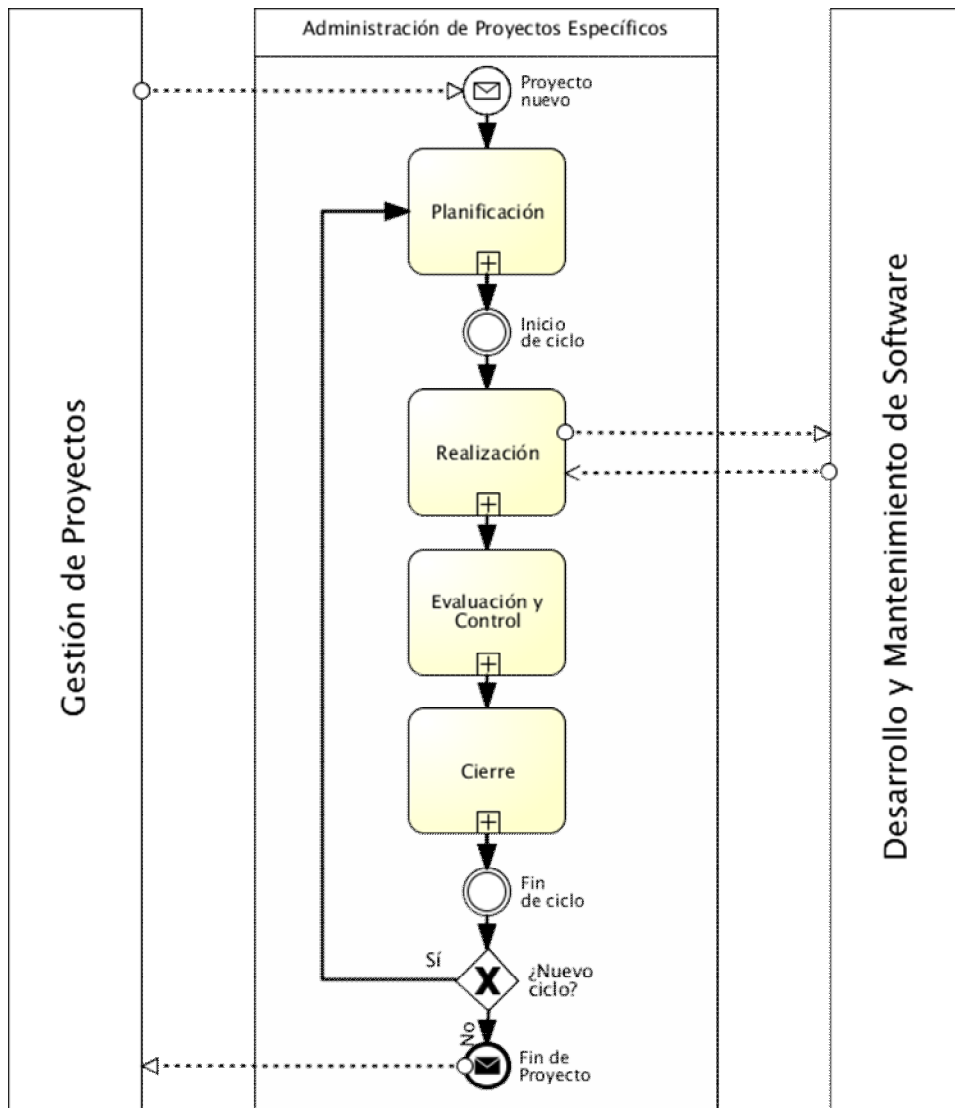


Figura 3.1. Modelo descriptivo de OPE.1 Administración de Proyectos Específicos



En la Figura 2.9 se puede observar que el proceso GES.2 GPY es el encargado de iniciar el proceso de OPE.1 APE. A su vez el proceso OPE.1 APE es el encargado de iniciar el proceso de OPE.2 DMS. Para OPE.1 APE los procesos de OPE.2 DMS y GES.2 GPY son procesos de caja negra, porque lo que ocurra dentro de ellos no modifica el comportamiento de OPE.1 APE, siempre que se cumpla con las entradas requeridas por OPE.1 APE.

Las actividades de OPE.2 DMS detalladas por [NYCE05] son descritas por el modelo correspondiente a la Figura 3.2. De acuerdo con [NYCE05], existen ciclos en OPE.2 DMS, sin embargo, dentro de este proceso no existen mecanismos que permitan la gestión de dichos ciclos, esto es porque los ciclos son administrados desde OPE.1 APE. Además la forma en que OPE.1 APE funciona carece de importancia para OPE.2 DMS, siempre que OPE.1 APE le entregue las entradas requeridas.

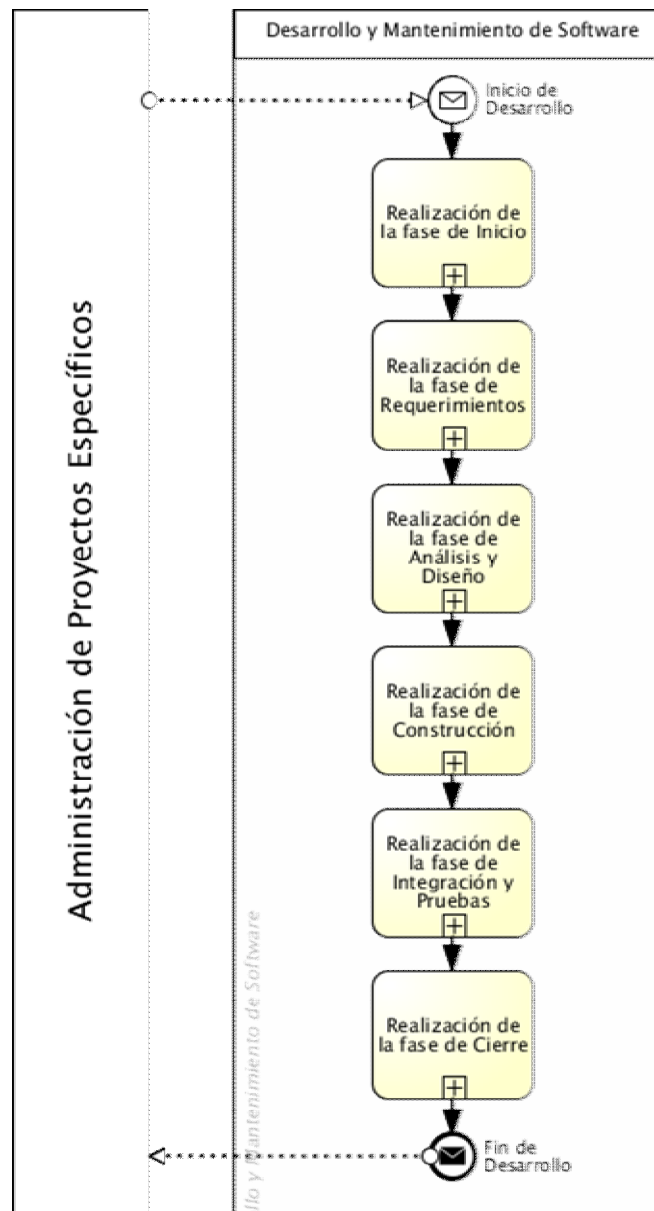
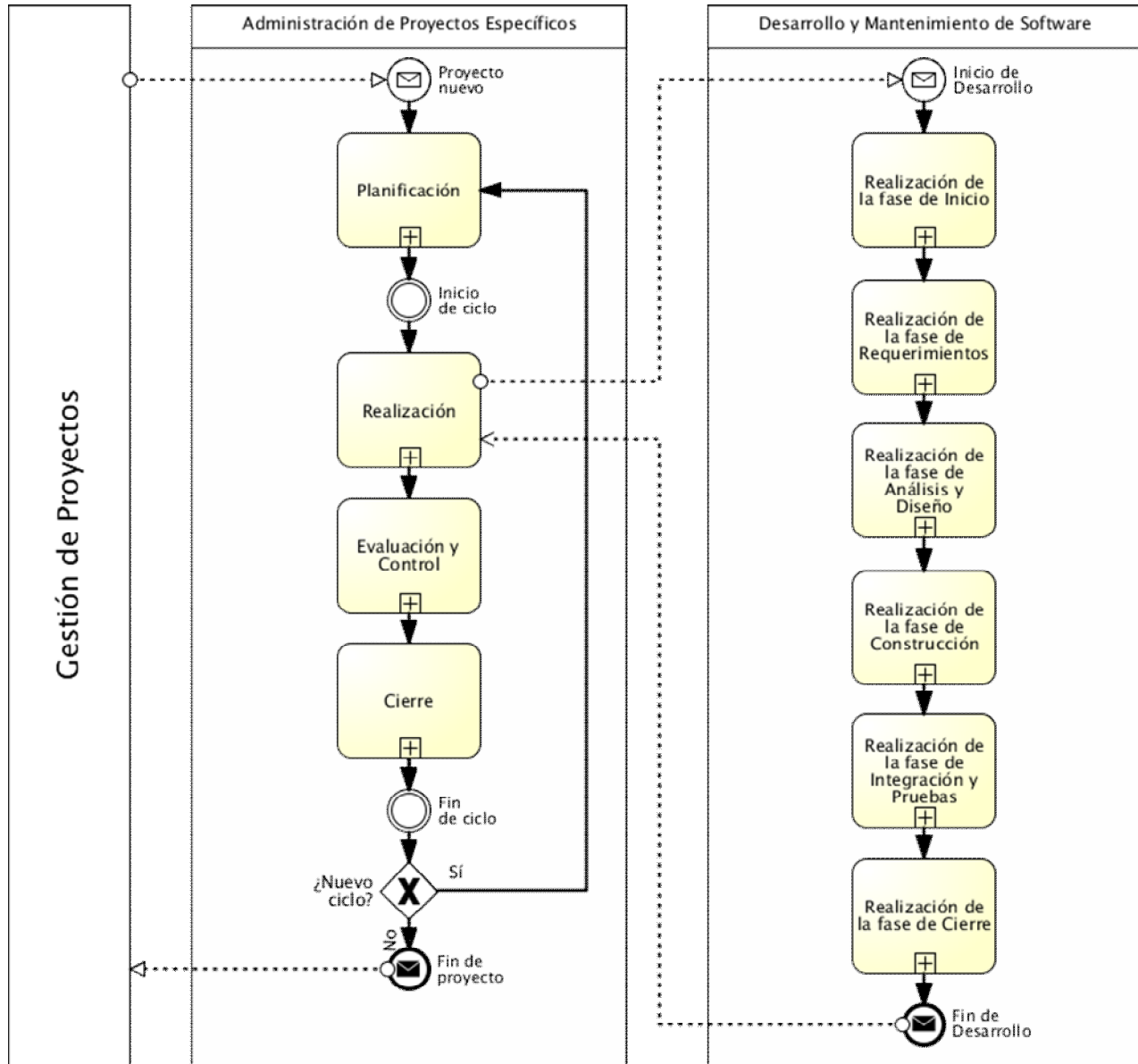


Figura 3.2. Modelo descriptivo de OPE.2 Desarrollo y Mantenimiento de Software

Por lo tanto, OPE está compuesto de dos procesos, OPE.1 APE y OPE.2 DMS, por lo que es necesario modelar el proceso de OPE para ver cómo interactúan dichos procesos. Así, se obtiene el modelo descriptivo de procesos correspondiente a OPE como se muestra en la Figura 3.3.



**Figura 3.3.** Modelo descriptivo de la categoría de operación (OPE)

En este sentido, en [NYCE05] se describe la forma en que MoProSoft debe realizar cada actividad de los procesos de OPE (ver Figura 3.3). Cada organización puede escoger entre hacer todo lo estipulado por la actividad o bien lo correspondiente a su nivel de capacidad.

Aunque en [NYCE05] las tareas de cada actividad vienen numeradas, no se explica el orden o la dependencia entre las actividades, más allá del que se pueda deducir por los insumos necesarios para su realización. Por esta razón no se modelan las actividades en un modelo BPMN 2.0 y se sigue utilizando la norma al realizar el análisis necesario.

### 3.1.1.2. Roles

La modelación de los roles se hace mediante una tabla que indique su nivel de participación en las actividades de los procesos. Para mayor información sobre los roles se recomienda consultar [NYCE05].

#### 3.1.1.2.1. Administración de Proyectos Específicos

Las actividades del proceso OPE.1 APE son llevadas a cabo por los roles siguientes:

- Responsable de Gestión de Proyectos (RGPY).
- Responsable de la Administración del Proyecto Específico (RAPE).
- Cliente (CL).
- Responsable del Subcontrato (RSC).
- Responsable de Desarrollo y Mantenimiento de Software (RDM).
- Equipo de Trabajo (ET).

La participación de los roles, arriba mencionados, en las tareas de las actividades de OPE.1 APE se indica en la Tabla 10.

**Tabla 10.** Participación de los roles de MoProSoft en OPE.1 APE.

<b>Rol</b>	<b>OPE.1 A1</b>	<b>OPE.1 A2</b>	<b>OPE.1 A3</b>	<b>OPE.1 A4</b>
RGPY	Si		Si	
RAPE	Si	Si	Si	Si
CL	Si	Si		Si
RSC		Si		Si
RDM	Si	Si		
ET		Si		

#### 3.1.1.2.2. Desarrollo y Mantenimiento de Software

OPE.2 DMS es llevado a cabo a través de las actividades definidas por el proceso a través de los roles siguientes:

- Responsable de Desarrollo y Mantenimiento de Software (RDM).
- Analista (AN).
- Diseñador de la interfaz de usuario (DU).
- Diseñador (DI).
- Programador (PR).
- Responsable de Pruebas (RPU).
- Revisor (RE).

- Responsable de Manuales (RM).
- Equipo de Trabajo (ET).
- Cliente (CL).
- Usuario (US).

La Tabla 11 describe en que actividades participa cada rol durante el proceso de OPE.2 DMS. Esta información será empleada más adelante durante el análisis.

**Tabla 11.** Participación de los roles de MoProSoft en OPE.1 APE.

Rol	OPE.2 A1	OPE.2 A2	OPE.2 A3	OPE.2 A4	OPE.2 A5	OPE.2 A6
RDM	Si	Si	Si	Si	Si	Si
AN		Si	Si			
DU		Si	Si			
DI			Si			
PR				Si	Si	
RPU		Si	Si		Si	
RE		Si	Si	Si	Si	Si
RM		Si			Si	Si
ET	Si					Si
CL		Si	Si			
US		Si				

### 3.1.1.3. Productos

Como última parte del modelado de MoProSoft se muestran los productos de entrada, salida e intermedios de cada actividad de los procesos de OPE.

#### 3.1.1.3.1. Administración de Proyectos Específicos

El proceso OPE.1 APE necesita como entradas los siguientes insumos:

- Plan de Comunicación e Implantación (DIR.1: PCI).
- Plan de Mediciones de Procesos (GES.1: PMP).
- Desarrollo y Mantenimiento de Software (GES.1: DMS).
- Descripción del Proyecto (GES.2: DP).
- Responsable de Administración del Proyecto Específico (GES.2: RAPE).
- Metas Cuantitativas para el Proyecto (GES.2: MCP).
- Acciones Correctivas o Preventivas (GES.2: ACP).
- Asignación de Recursos (GES.3.1: AR).
- Reporte de Actividades (OPE.2: RA).

- Reporte de Mediciones y Sugerencias de Mejora (OPE.2: RMSM).
- Configuración de Software (OPE.2: CS).
- Solicitud de Cambios (CL: SC).

En la Tabla 12 se muestra en qué actividades son utilizados los insumos de este proceso.

**Tabla 12.** Insumos de las actividades del proceso de OPE.1 APE.

Insumo	OPE.1 A1	OPE.1 A2	OPE.1 A3	OPE.1 A4
DIR.1: PCI		Entrada		
GES.1: PMP				Entrada
GES.1: DMS	Entrada			
GES.2: DP	Entrada			
GES.2: RAPE	Entrada			
GES.2: MCP	Entrada			
GES.2: ACP	Entrada			
GES.3.1: AR		Entrada		
OPE.2: RA		Entrada		
OPE.2: RMSM		Entrada		
OPE.2: CS		Entrada		
CL: SC	Entrada	Entrada		

Las salidas generadas durante el proceso OPE.1 APE son las siguientes:

- Reporte de Mediciones y Sugerencias de Mejora (OPE.1: RMSM).
- Plan del Proyecto (OPE.1: PP).
- Reporte de Seguimiento (OPE.1: RS).
- Documentación de Aceptación (OPE.1: DA).
- Plan de Adquisiciones y Capacitación (OPE.1: PAC).
- Lecciones Aprendidas (OPE.1: LA).
- Plan de Desarrollo (OPE.1: PD).

En la Tabla 13 se muestra en qué actividades son generadas las salidas de este proceso.

**Tabla 13.** Productos de las actividades del proceso de OPE.1 APE.

Salida	OPE.1 A1	OPE.1 A2	OPE.1 A3	OPE.1 A4
OPE.1: RMSM				Salida
OPE.1: PP	Salida	Salida		
OPE.1: RS			Salida	
OPE.1:DA				Salida
OPE.1: PAC	Salida			
OPE.1: LA				Salida
OPE.1: PD	Salida	Salida		

### 3.1.1.3.2. Desarrollo y Mantenimiento de Software

El proceso OPE.2 DMS necesita como entradas los siguientes insumos:

- Plan de Desarrollo (OPE.1: PD).

En la Tabla 14 se muestra en qué actividades es utilizado el insumo de este proceso.

**Tabla 14.** Insumos de las actividades del proceso de OPE.2 DMS.

Insumo	OPE.2 A1	OPE.2 A2	OPE.2 A3	OPE.2 A4	OPE.2 A5	OPE.2 A6
OPE.2: PD	Entrada	Entrada	Entrada	Entrada	Entrada	

Las salidas generadas durante el proceso OPE.2 DMS son las siguientes:

- Especificación de Requisitos (OPE.2: ER).
- Análisis y Diseño (OPE.2: AD).
- Componente (OPE.2: C).
- Software (OPE.2: S).
- Configuración de Software (OPE.2: CS).
- Manual de Usuario (OPE.2: MU).
- Manual de Operación (OPE.2: MO).
- Manual de Mantenimiento (OPE.2: MM).
- Reporte de Actividades (OPE.2: RA).
- Lecciones Aprendidas (OPE.2: LA).
- Reporte de Mediciones y Sugerencias de Mejora (OPE.2: RMSM).
- Registro de Rastreo (OPE.2: RR).
- Plan de Pruebas de Sistema (OPE.2: PPS).
- Reporte de Pruebas de Sistema (OPE.2: RPS).

- Plan de Pruebas de Integración (OPE.2: PPI).
- Reporte de Pruebas de Integración (OPE.2: RPI).

En la Tabla 15 se muestra en qué actividades son generadas las salidas de este proceso.

**Tabla 15.** Productos de las actividades del proceso de OPE.2 DMS.

Insumo	OPE.2 A1	OPE.2 A2	OPE.2 A3	OPE.2 A4	OPE.2 A5	OPE.2 A6
OPE.2: ER		Salida				
OPE.2: AD			Salida			
OPE.2: C				Salida		
OPE.2: S					Salida	
OPE.2: CS		Salida	Salida	Salida	Salida	Salida
OPE.2: MU		Salida			Salida	
OPE.2: MO					Salida	
OPE.2: MM						Salida
OPE.2: RA	Salida	Salida	Salida	Salida	Salida	
OPE.2: LA						Salida
OPE.2: RMSM						Salida
OPE.2: RR			Salida	Salida	Salida	
OPE.2: PPS		Salida				
OPE.2: RPS					Salida	
OPE.2: PPI			Salida			
OPE.2: RPI					Salida	

### 3.1.2. Scrum

#### 3.1.2.1. Procesos

Scrum tiene un único proceso que será modelado en su totalidad. El modelo del proceso de Scrum, de acuerdo con [Schwaber10], es el mostrado en la Figura 3.4. La descripción de cada actividad de Scrum está basada también en este documento.

#### 3.1.2.2. Roles

Las actividades del proceso de Scrum son llevadas a cabo por tres roles:

- Dueño del Producto (DP).
- Equipo de Desarrollo (ED).
- Maestro de Scrum (MS).

La participación de los roles, arriba mencionados, en las tareas de las actividades de Scrum se indica en la Tabla 16.

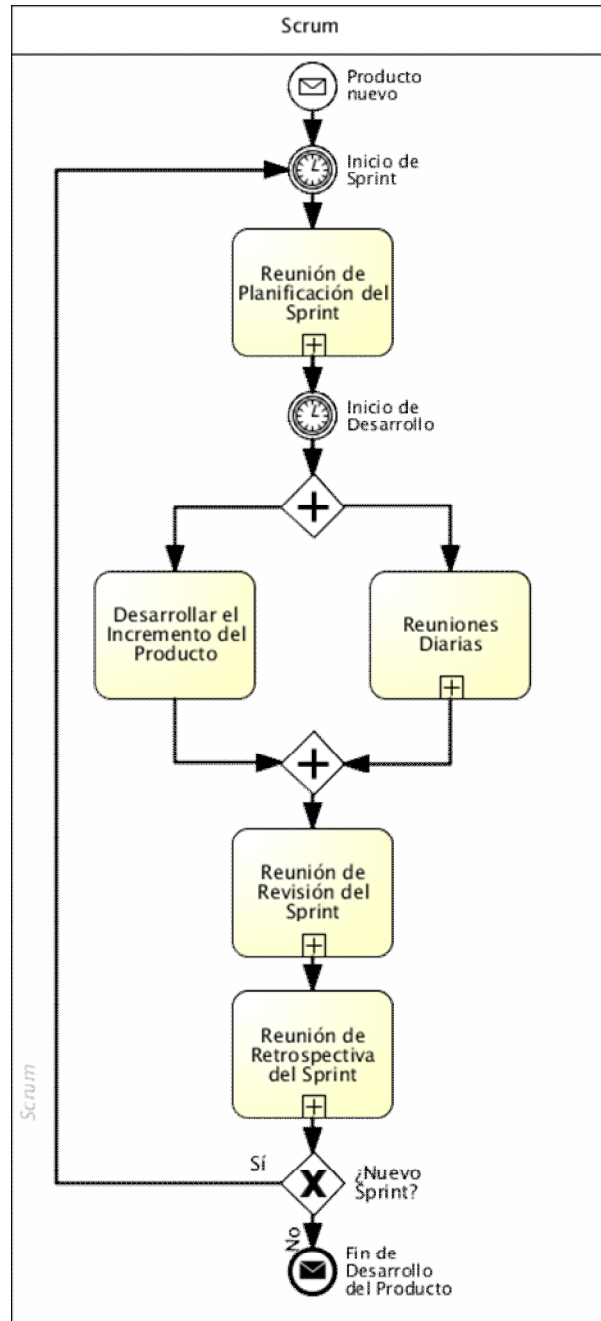


Figura 3.4. Modelo descriptivo del proceso de Scrum

Tabla 16. Roles participantes en el proceso de Scrum.

Rol	Planificación	Reuniones diarias	Desarrollo	Revisión	Retrospectiva
DP	Si			Si	Si
ED	Si	Si	Si	Si	Si
MS	Si	Si		Si	Si



### 3.1.2.3. Productos

El proceso de Scrum necesita como entrada los siguientes insumos:

- Registro de Pendientes del Producto (DP: RPP).
- Registro de Pendientes del Sprint (ED: RPS).
- Incremento (ED: I).

En la Tabla 17 se muestra en qué actividades son utilizados los insumos de este proceso.

**Tabla 17.** Insumos de las actividades del proceso de Scrum.

Rol	Planificación	Reuniones diarias	Desarrollo	Revisión	Retrospectiva
DP: RPP	Entrada			Entrada	Entrada
ED: RPS			Entrada		Entrada
ED: I				Entrada	

Las salidas generadas durante el proceso de Scrum son las siguientes:

- Registro de Pendientes del Producto (DP: RPP).
- Registro de Pendientes del Sprint (ED: RPS).
- Incremento (ED: I).

En la Tabla 18 se muestra en qué actividades son generadas las salidas de este proceso.

**Tabla 18.** Productos de las actividades del proceso de Scrum.

Rol	Planificación	Reuniones diarias	Desarrollo	Revisión	Retrospectiva
DP: RPP				Salida	
ED: RPS	Salida	Salida			
ED: I			Salida		

## 3.2. Análisis de MoProSoft vs Scrum

En esta sección se comienza por mostrar las diferencias y coincidencias existentes entre el proceso de MoProSoft y Scrum. Los puntos a analizar son los siguientes:

- Proceso.
- Roles.
- Artefactos.

### 3.2.1. Procesos

Al analizar los procesos de MoProSoft (ver Figura 3.3) y de Scrum (ver Figura 3.4) se observa que ambos procesos consideran ciclos desarrollo del producto de software en su estructura, es decir

ambos son iterativos. MoProSoft durante OPE.2 DMS hace la integración de los componentes al sistema existente y alcanza al final un incremento; mientras que en Scrum el desarrollo de los componentes se hace considerando cada elemento del registro de pendientes del producto como un incremento al producto. Lo anteriormente expresado quiere decir que MoProSoft y Scrum son iterativos e incrementales, lo cual se resume en la Tabla 19.

**Tabla 19.** Análisis de MoProSoft vs Scrum.

<b>Tema</b>	<b>MoProSoft</b>	<b>Scrum</b>
Iterativo incremental	Si	Si
Proceso de gestión de tiempo y costo	Si	Si
Proceso de gestión del trabajo de desarrollo	Si	No
Inicio y fin del proyecto	Si	Si
Inicio y fin del ciclo	Si	Si
Inicio y fin de trabajo de desarrollo	Si	Si

Por otro lado, MoProSoft separa en dos procesos diferenciados la gestión de tiempo y el costo del proyecto, y la gestión del trabajo de desarrollo del proyecto. Mientras que Scrum se enfoca más a la gestión de tiempo y costo del proyecto, dejando el trabajo de desarrollo del proyecto a una actividad auto-gestionada por el equipo de desarrollo, lo anterior también se muestra en el resumen de la Tabla 19.

Tanto MoProSoft como Scrum cuentan con hitos para el inicio y fin del proyecto, que permiten delimitar cada proyecto, ciclo y trabajo de desarrollo, esto también es agregado al resumen de la Tabla 19. Tomando en cuenta el análisis anterior, las consideraciones sobre el proceso de MoProSoft para que éste soporte a Scrum son las siguientes:

1. MoProSoft ya es iterativo e incremental.
2. Las actividades de gestión de tiempo y costo de Scrum, deben ser realizadas en MoProSoft durante el proceso correspondiente, es decir, en OPE.1 APE.
3. La actividad del trabajo de desarrollo de Scrum está completamente cubierta por MoProSoft en el proceso OPE.2 DMS.
4. El inicio y fin del proyecto, de cada ciclo, y de cada trabajo de desarrollo seguirá según lo planteado por OPE (ver Figura 3.3).

Los puntos 1 y 4 arriba mencionados sólo son recordatorios de lo que no se debe modificar en MoProSoft cuando se obtenga el diseño del nuevo patrón de procesos. Para concluir el análisis de los procesos se indica en qué actividades de MoProSoft deberán ser colocadas las actividades de Scrum de la siguiente forma:

- Reunión de Planeación del Sprint.
- Reuniones Diarias.
- Desarrollar el Incremento del Producto.
- Reunión de Revisión del Sprint.

- Reunión de Retrospectiva del Sprint.

La actividad Reunión de Planeación del Sprint debe realizarse según el punto 2 de las consideraciones en OPE.1 APE. Al revisar las tareas de las actividades de OPE.1 APE se encuentra mayor coincidencia con la parte final de la actividad OPE.1 A1 Planificación. Las tareas correspondientes son:

- Generar el Plan de Desarrollo en función del Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo. Además, el Plan de Desarrollo se puede actualizar a razón de una Solicitud de Cambios por parte del Cliente, Acciones Correctivas o Preventivas provenientes de Gestión de Proyectos o Acciones Correctivas de este proceso.
- Verificar el Plan del Proyecto y el Plan de Desarrollo.
- Corregir los defectos encontrados en el Plan del Proyecto y en el Plan de Desarrollo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
- Validar el Plan del Proyecto y el Plan de Desarrollo.
- Corregir los defectos encontrados en el Plan del Proyecto y Plan de Desarrollo con base en el Reporte de Validación y obtener la aprobación de las correcciones.

La actividad de Reuniones Diarias debe realizarse también en el proceso OPE.1 APE, pero en alguna actividad en paralelo al trabajo realizado durante el proceso OPE.2 DMS. La actividad que más se ajusta es OPE.1 A2 Realización y esta actividad cuenta con una tarea compatible con el objetivo de la actividad de Reuniones Diarias, dicha tarea es:

- Acordar con el Responsable de Desarrollo y Mantenimiento del proyecto la asignación de tareas al Equipo de Trabajo incluyendo a los subcontratistas.

La actividad de Desarrollar el Incremento del Producto, tal y como se explicó en el punto 3 de las consideraciones, es realizada completamente por el proceso OPE.2 DMS.

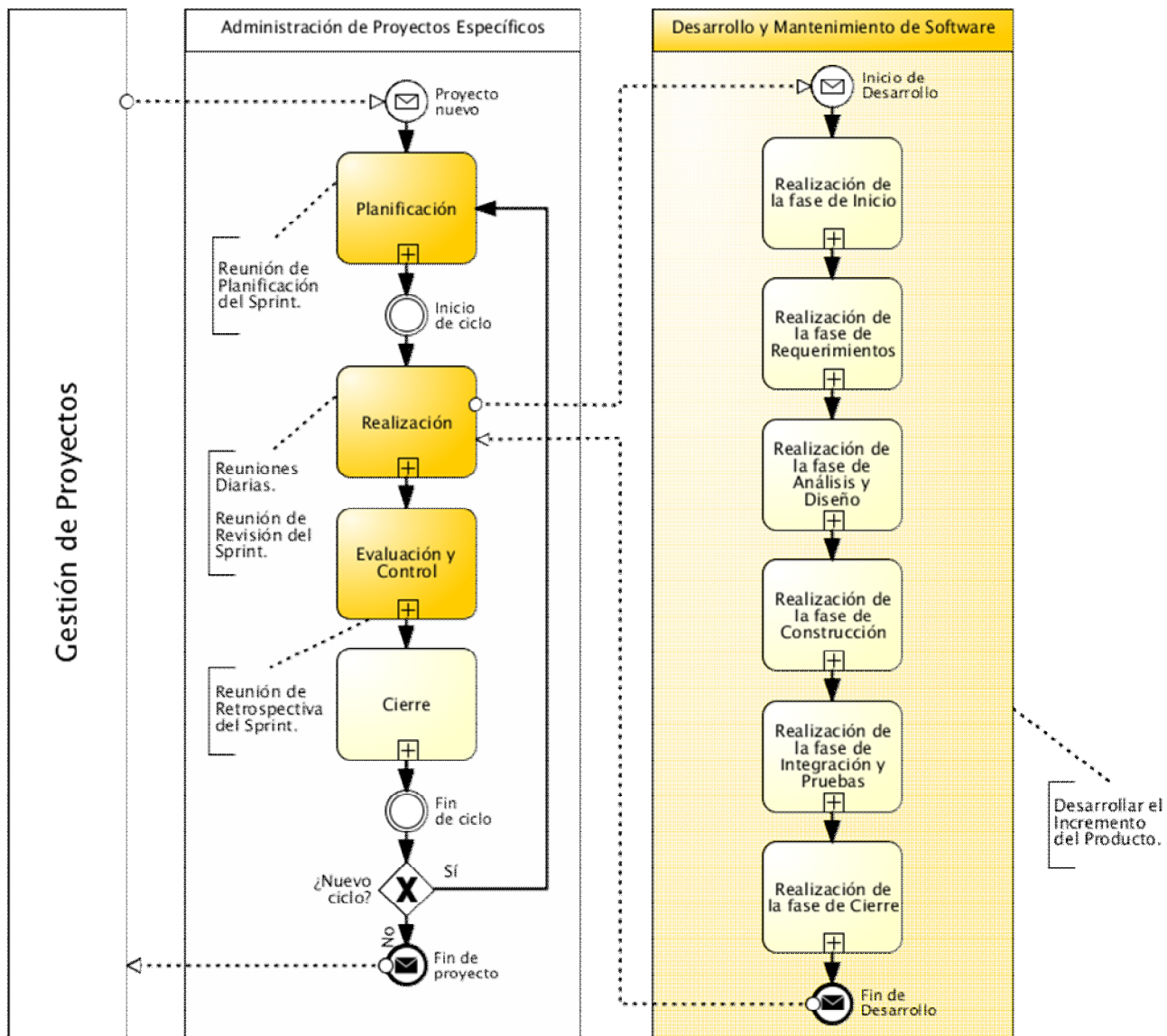
La actividad Reunión del Sprint se realiza una vez terminado el trabajo de desarrollo del producto, estando todavía dentro del ciclo, y de acuerdo el punto 2 de las consideraciones debe ser realizado dentro del proceso OPE.1 APE. Al revisar las tareas de las actividades de OPE.1 APE que se realizan dentro del ciclo (OPE.1 A2, OPE.1 A3 y OPE.1 A4) se encuentra mayor coincidencia con la actividad OPE.1 A2 Realización. Dichas tareas son:

- Recibir y analizar las Solicitudes de Cambios e incorporar los cambios aprobados en el Plan del Proyecto y en el Plan de Desarrollo. En caso de cambios a Requisitos éstos son incorporados al inicio de un nuevo ciclo.
- Conducir reuniones de revisión con el equipo de trabajo y con el Cliente, generando Minutas con puntos tratados y acuerdos tomados.

La actividad Reunión de Retrospectiva del Sprint debe realizarse después de la reunión de Revisión del Sprint, y según el punto 2 de las consideraciones en el proceso OPE.1 APE. La actividad que más se ajusta es OPE.1 A3 Evaluación y Control, esta actividad cuenta con una tarea que coincide con el objetivo de la actividad de Reunión de Retrospectiva del Sprint, la cual es:

- Evaluar el cumplimiento del Plan del Proyecto y el Plan de Desarrollo, con respecto al alcance, costo, calendario, equipo de trabajo, proceso y establecer Acciones Correctivas.

En la Figura 3.5 se muestra un resumen del resultado del análisis resaltando aquellas actividades o procesos que serán modificados, tomando como base el Modelo de Procesos Descriptivo de la Categoría de Operación (OPE) mostrado en la Figura 3.3.



**Figura 3.5.** Análisis sobre el modelo descriptivo de la Categoría de Operación (OPE) versus Scrum

De manera complementaria a la Figura 3.5, la Tabla 20 muestra en color gris las intersecciones de las actividades de OPE.1 APE y OPE.2 DMS, con las actividades de Scrum. A través de esta tabla se obtiene del análisis correspondiente a esta sección 3.2.1 que tiene como objetivo facilitar el análisis de roles y de productos, ya que el proceso resultante solo modificará las actividades de MoProSoft intersectadas por Scrum.

**Tabla 20.** Intersecciones de las actividades de MoProSoft con las actividades de Scrum.

		Scrum				
		Reunión de planificación del sprint	Reuniones diarias	Desarrollar el incremento del producto	Reunión de revisión del sprint	Reunión de retrospectiva del sprint
MoProSoft	OPE.1 APF	A1 Planificación	■			
		A2 Realización		■		■
		A3 Evaluación y control				■
		A4 Cierre				
	OPE.2 DMS	A1 Inicio			■	
		A2 Requisitos			■	
		A3 Análisis y Diseño			■	
		A4 Construcción			■	
		A5 Integración y Pruebas			■	
		A6 Cierre			■	

### 3.2.2. Roles

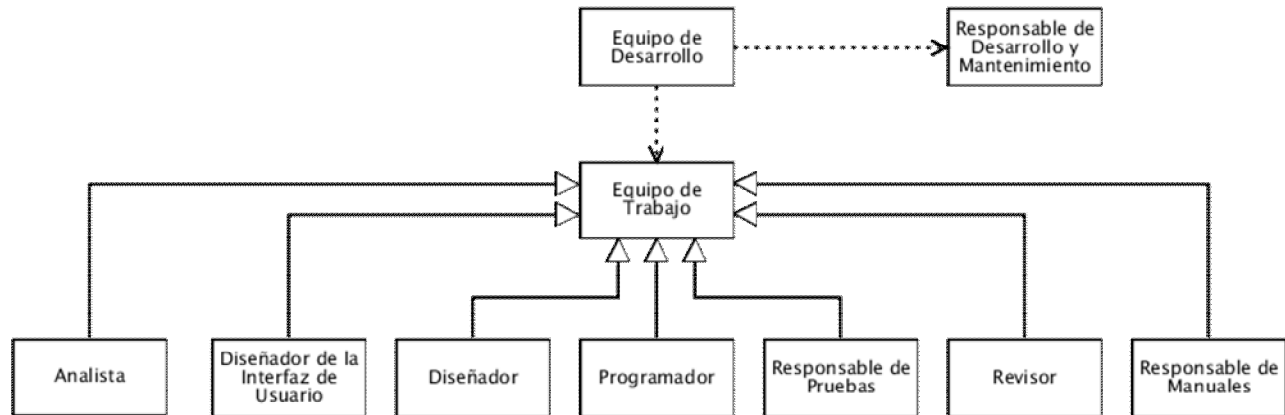
En esta sección se analizan las correspondencias entre los roles de Scrum y los roles de MoProSoft. Para esto se consideran la modelación de los roles de MoProSoft de la sección 3.1.1.2 y de Scrum de la sección 3.1.2.2, también se tomará en cuenta el resultado del análisis de los procesos de la sección 3.2.1. Los roles de Scrum a analizar son:

- Dueño de Producto.
- Equipo de Desarrollo.
- Maestro de Scrum.

Tanto Scrum como MoProSoft definen roles responsables del cumplimiento del proyecto de desarrollo del producto dentro del tiempo, costo y alcance. De acuerdo con [Schwaber10], esta responsabilidad recae en el rol del Dueño de Producto y le corresponde al rol Responsable de la Administración de Proyecto Específico [NYCE05].

Así, el equipo de desarrollo está formado por profesionales que hacen el trabajo para entregar un Incremento potencialmente liberable del producto, terminado al final de cada *sprint*, esto quiere decir que son todas las personas que participan en la actividad de Desarrollar el Incremento del Producto, y de acuerdo con la Figura 3.5 toda la actividad de Scrum para Desarrollar el Incremento del Producto es realizada a través de OPE.2 DMS.

Por lo anterior se concluye que todos los involucrados en la construcción dentro del proceso de OPE.2 DMS corresponderían al equipo de desarrollo (ver Figura 3.6).



**Figura 3.6.** Relaciones de roles de OPE.2 DMS con el rol Equipo de Desarrollo de Scrum

Por último, en cuanto al Maestro de Scrum, obviamente no existe una correspondencia, y es necesario agregar el rol de Maestro de Scrum en el proceso OPE.1 pues es aquí donde se realizan las actividades de administración del proyecto y para las cuales el Maestro de Scrum debe proveer orientación sobre cómo se deben hacer las actividades de Scrum y asegurar que el Dueño de Producto y el Equipo de Desarrollo se adhieran a la teoría, prácticas y reglas de Scrum.

En la Tabla 21 se muestran las equivalencias obtenidas en este análisis de esta sección.

**Tabla 21.** Equivalencia de roles de MoProSoft con roles de Scrum.

Proceso de MoProSoft	Roles de Scrum			
	Roles de MoProSoft	Dueño del Producto	Equipo de Desarrollo	Maestro de Scrum
OPE. 1 APE	RAPE	Correspondiente		
	RDM		Es parte de	
	ET		Correspondiente	
OPE.2 DMS	RDM		Es parte de	
	AN		Es parte de	
	DU		Es parte de	
	DI		Es parte de	
	PR		Es parte de	
	RPU		Es parte de	
	RE		Es parte de	
	RM		Es parte de	
	ET		Correspondiente	

### 3.2.3. Productos

Tal y como se explicó en la sección 3.1.2.3, Scrum define 3 artefactos:

- Registro de Pendientes del Producto.
- Registro de Pendientes del Sprint.

- Incremento del Producto.

En MoProSoft la descripción del producto se encuentra en el insumo llamado Descripción del Producto, el cual entra a OPE.1 APE dentro de Descripción del Proyecto y en Scrum la descripción de lo que se requiere del producto se encuentra en el Registro de Pendientes del Producto, pero priorizado por el responsable del producto. Entonces, si MoProSoft desea usar Scrum, el Registro de Pendientes del Producto debe ser generado a partir de la Descripción del Producto. Es necesario entonces que durante OPE.1 APE se genere el Registro de Pendientes del Producto tomando como base la Descripción del Producto y que éste sea agregado al Plan de Proyecto.

El Plan de Desarrollo generado en OPE.1 APE que ingresa a OPE.2 DMS es la guía para la ejecución del desarrollo o mantenimiento de software; de forma similar, en Scrum el Registro de Pendientes del Sprint dirige el trabajo de desarrollo, pero además incluye la lista de funcionalidades que estarán en la siguiente entrega del Incremento.

Durante la fase de Integración y Pruebas de OPE.2 DMS se genera el software, que es el resultado de integrar los componentes a los subsistemas o al sistema existente, esto es equivalente al Incremento de Scrum.

En la Tabla 22 se muestra el resultado del análisis sobre productos realizado en esta sección.

**Tabla 22.** Relación entre productos de MoProSoft con productos de Scrum.

Proceso de MoProSoft	Productos de MoProSoft	Productos de Scrum		
		Registro de pendientes del producto	Registro de pendientes del sprint	Incremento
OPE. 1 APE	Plan del proyecto	Incluye a		
	Descripción del producto	Usado por		
	Plan de desarrollo		Correspondiente	
OPE.2 DMS	Plan de desarrollo		Correspondiente	

Al resultado del análisis mostrado en la Figura 3.5 se le debe sumar la generación del Registro de Pendientes del Producto y del Registro de Pendientes del Sprint, como se muestra en la Figura 3.7.

### 3.2.4. Resultados del análisis

Como resultado del análisis presentado, a continuación se muestra el resumen de cambios a MoProSoft para que pueda implementar Scrum.

#### 3.2.4.1. Administración de Proyecto Específico

El proceso OPE.1 APE debe ser modificado de acuerdo con la norma MoProSoft [NYCE05]. A continuación se presentan los cambios necesarios que deben hacerse en las actividades, roles y productos del proceso OPE.1 APE de MoProSoft.

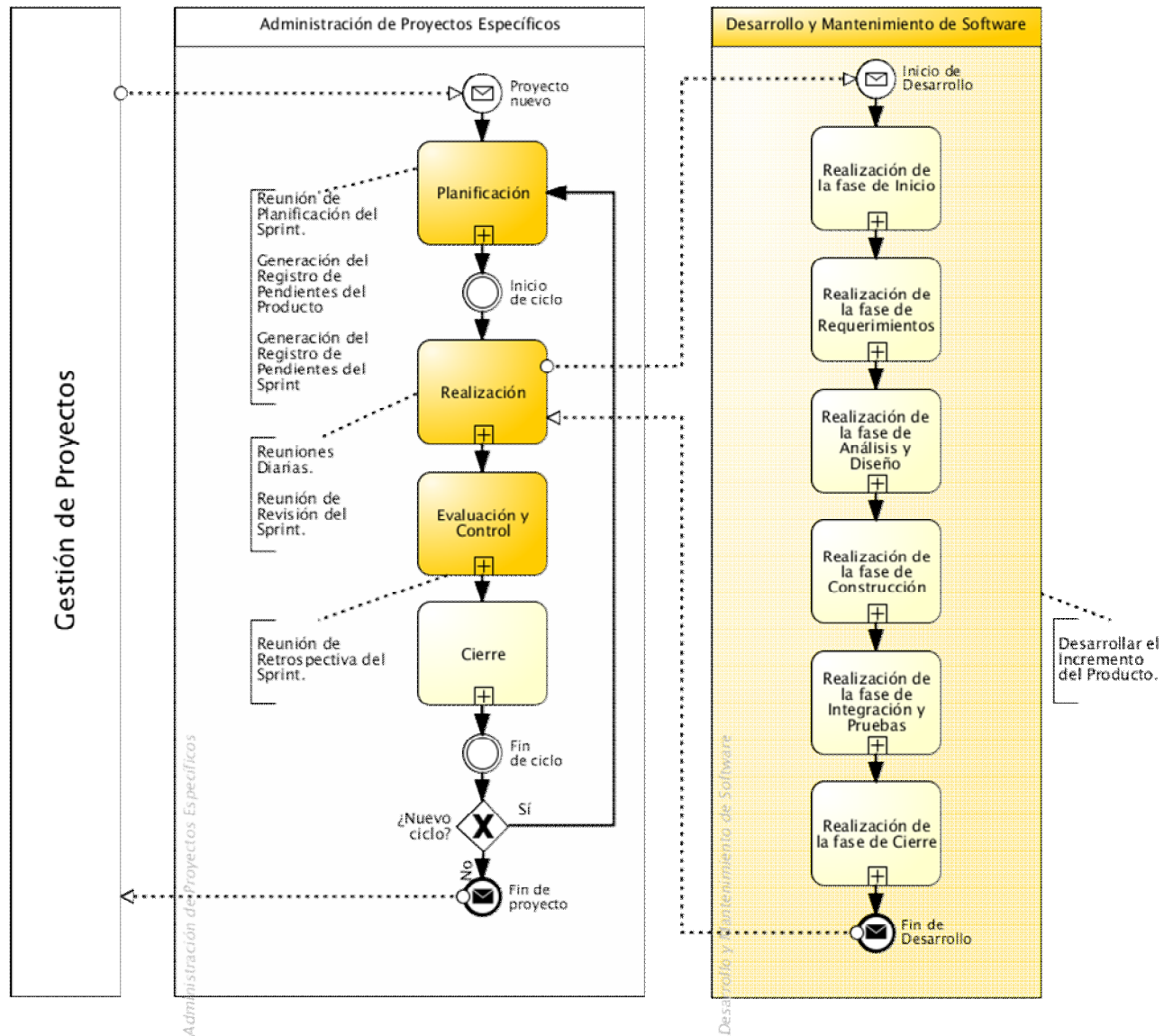


Figura 3.7. Análisis de la Categoría de Operación (OPE) versus Scrum

### 3.2.4.1.1. Actividades

Tal y como se indicó en la Figura 3.7, para “MoProSoft – Scrum” se modifican las siguientes actividades de los procesos de OPE.1 APE de MoProSoft:

- OPE.1 A1 Planificación.
- OPE.1 A2 Realización.
- OPE.1 A3 Evaluación y Control.

Durante la actividad OPE.1 A1 Planificación, como última parte de la tarea A1.1, el RAPE debe generar y priorizar el Registro de Pendientes del Producto, debido a que ahora el rol de RAPE tiene las responsabilidades del rol de Dueño de Producto.



En la misma actividad OPE.1 A1 Planificación, la generación, validación, verificación y correcciones al Plan de Desarrollo descritas en las tareas A1.13, A1.14, A1.15, A1.16 y A1.17 deben ser realizadas durante la Reunión de Planificación del Sprint. Además es necesario que el rol de Equipo de Trabajo participe en la generación y corrección del Plan de Desarrollo, ya que este rol debe ser el equivalente al rol de Equipo de Desarrollo de Scrum.

Durante cada día de trabajo, mientras el proceso de OPE.2 DMS esté funcionando, la actividad OPE.1 A2 Realización, debe realizar la tarea A2.1 durante las Reuniones Diarias. Esta tarea complementa lo que la Reunión Diaria debe responder cada día. En la reunión debe participar todo el Equipo de Trabajo, no sólo el RDM.

Las tareas de A2.9, A2.10 y A2.11 de la actividad OPE.1 A2 Realización, deben realizarse durante la Reunión de Revisión y después de que el proceso OPE.2 DMS haya concluido, y además el Equipo de Trabajo debe estar presente durante la revisión. Esto es porque Scrum ordena que ningún cambio, que afecte las metas del ciclo, sea hecho durante el desarrollo dentro del ciclo. Las razones por las cuales no se debe afectar las metas son explicadas por [Schwaber10].

Por último, durante la actividad OPE.1 A3 Evaluación y control, la actividad A3.1 debe realizarse durante la Reunión de Retrospectiva. Pues con esto se logrará crear la oportunidad para que el RAPE y el Equipo de Trabajo se inspeccionen asimismo y creen un plan para mejorar y vigilar en el siguiente ciclo. En la Tabla 23 se subrayan los cambios realizados a la tabla *Actividades* descrita en [NYCE05], además se agrega el rol de Maestro de Scrum en las tareas que son realizadas durante las actividades que provinieron de Scrum, para indicar que este rol debe proveer guía sobre el marco de trabajo de Scrum.

**Tabla 23.** Modificaciones a la tabla *Actividades del proceso* OPE.1 APE de MoProSoft.

Rol	Descripción
<i>A1. Planificación (O1)</i>	
RGPY RAPE RDM	A1.1. Revisar con el Responsable de Gestión de Proyectos la Descripción del Proyecto. Generar y priorizar el Registro de Pendientes del Producto.
RAPE ET MS	A1.13. Realizar la Reunión de Planificación, y el Responsable de Administración del Proyecto Específico Explicará el Equipo de Trabajo el Plan del Proyecto. Generar el Plan de Desarrollo en función del Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo. Además el Plan de Desarrollo se puede actualizar a causa de Solicitud de Cambios por parte del Cliente, Acciones Correctivas o Preventivas provenientes de Gestión de Proyectos o Acciones Correctivas de este proceso.
RAPE RDM	A1.14 Durante la Reunión de Planificación: Verificar el Plan del Proyecto y el Plan de Desarrollo (Ver1).
RAPE ET MS	A1.15 Durante la Reunión de Planificación: Corregir los defectos encontrados en el Plan del Proyecto y en el Plan de Desarrollo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RGPY	A1.16 Durante la Reunión de Planificación: Validar el Plan del Proyecto y el Plan de Desarrollo (Val1).
RAPE ET MS	A1.17 Durante la Reunión de Planificación: Corregir los defectos encontrados en el Plan del Proyecto y Plan de Desarrollo con base en el Reporte de Validación y obtener la aprobación de las correcciones.

<i>A2. Realización (O1, O2, O3)</i>	
RAPE RDM ET	A2.1 Realizar Reuniones Diarias para informar al RDM sobre el trabajo realizado el día anterior, el trabajo a realizar el día actual y los impedimentos encontrados para el logro de las tareas asignadas.  Acordar con el Responsable de Desarrollo y Mantenimiento del proyecto la asignación de tareas al Equipo de Trabajo incluyendo a los subcontratistas.
RAPE RDM ET MS	A2.10 Recibir y analizar las Solicitudes de Cambios e incorporar los cambios aprobados en el Plan del Proyecto y en el Plan de Desarrollo. En caso de cambios a Requisitos se incorporan al inicio de un nuevo ciclo.  Estos cambios pueden ser recibidos antes, durante y después de la revisión del producto realizada en la Reunión de Revisión.
RAPE ET CL MS	A2.11 Conduce reuniones de revisión con el equipo de trabajo y con el Cliente, generando Minutas con puntos tratados y acuerdos tomados.  En la Reunión de Revisión identificar los acuerdos para el ciclo actual que se terminaron y los acuerdos que no se terminaron pasarlos a un ciclo siguiente.
<i>A3. Evaluación y Control (O1)</i>	
RAPE ET MS	A3.1 Realizar la Reunión de Retrospectiva para identificar lo que se hizo bien y lo que se hizo mal durante el ciclo, identificar las mejoras al proceso que deben incorporarse en el siguiente ciclo, y agregarlas a las Acciones Correctivas que se establezcan.  Evaluar el cumplimiento del Plan del Proyecto y el Plan de Desarrollo, con respecto al alcance, costo, calendario, equipo de trabajo, proceso y se establecen Acciones Correctivas.

### 3.2.4.1.2. Roles

Para el nuevo patrón de procesos “MoProSoft – Scrum” el rol de RAPE es el equivalente al Dueño del Producto, además de las responsabilidades descritas por la norma, éste adquiere la responsabilidad de priorizar el Registro de Pendientes del Producto.

En cuanto al rol de Equipo de Trabajo, además de las responsabilidades descritas por la Norma, éste debe participar en la generación del Plan de Desarrollo ya que se estará comprometiendo a cumplirla en tiempo y alcance durante el siguiente ciclo. Con esto se logra que el rol Equipo de Trabajo de MoProSoft sea equivalente al rol de Equipo de Desarrollo de Scrum.

El rol Maestro de Scrum debe ser agregado a los roles de “MoProSoft – Scrum”, y tiene las mismas responsabilidades que en Scrum, esto es, asegurar el cumplimiento de las reglas, prácticas y actividades de Scrum por parte del Dueño de Producto, ahora RAPE, y del Equipo de Desarrollo, ahora Equipo de Trabajo. En la Tabla 24 se muestran los cambios realizados en la tabla Roles involucrados y capacitación descrita en [NYCE05].

**Tabla 24.** Modificaciones a la tabla *Roles involucrados y capacitación del proceso OPE.1 APE de MoProSoft.*

<b>Rol</b>	<b>Abreviatura</b>	<b>Capacitación</b>
Responsable de Administración de Proyectos Específicos	RAPE	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal, delegación y supervisión, finanzas, desarrollo de software, priorización del producto y comunicación con <i>stakeholders</i> .
Maestro de Scrum	MS	Capacidad para liderar y enseñar el marco de trabajo de Scrum a la organización y todos los participantes, servicio y facilitación para la realización de los eventos de Scrum.

### 3.2.4.1.3. Productos

Dadas las modificaciones anteriores, en OPE.2 se debe agregar al Plan del Proyecto el Registro de Pendientes del Producto. En cuanto al Registro de Pendientes del Sprint éste tiene su equivalente con el Plan de Desarrollo, y por lo tanto no es necesario agregarlo. En la Tabla 25 se muestra la modificación de la tabla de Salidas que se describe en [NYCE05].

**Tabla 25.** Modificaciones a la tabla *Salidas del proceso* OPE.1 APE de MoProSoft.

Nombre	Descripción	Destino
Plan del proyecto	<p>Documento formal usado como guía para la ejecución y control del proyecto. Está conformado por:</p> <ul style="list-style-type: none"> <li>• Registro de Pendientes del Producto</li> <li>• Ciclos y Actividades</li> <li>• Tiempo Estimado</li> <li>• Plan de Adquisiciones y Capacitación.</li> <li>• Equipo de Trabajo</li> <li>• Costo Estimado</li> <li>• Calendario</li> <li>• Plan de Manejo de Riesgos</li> <li>• Protocolo de Entrega</li> </ul>	Gestión de Proyectos

### 3.2.4.2. Desarrollo y Mantenimiento de Software

Tal y como se explicó en el análisis, el proceso OPE.2 DMS permanecerá intacto, además que ya cumple con el rol y los productos que le corresponden. En cuanto a roles, el rol de Equipo de Trabajo de MoProSoft es el equivalente al Equipo de Desarrollo de Scrum.

Para el producto de Plan de Desarrollo de MoProSoft es el equivalente al Registro de Pendientes del Sprint. Y para el software obtenido al final de los procesos de MoProSoft su equivalente es el producto de Scrum llamado Incremento.

Por lo tanto, en el nuevo patrón de proceso “MoProSoft – Scrum” el proceso OPE.2 Desarrollo y Mantenimiento de Software se utiliza tal como se describe en la norma MoProSoft [NYCE05].

## 3.3. Diseño del patrón relacionado con la mezcla “MoProSoft – Scrum”

El nuevo patrón de procesos “MoProSoft – Scrum” debe tener las siguientes características:

- Cumplir con los procesos de la categoría de Operación de MoProSoft.
- Cumplir con los roles marcados por los procesos de la categoría de Operación de MoProSoft.
- Cumplir con los productos de entrada, salida e internos de los procesos de la categoría de Operación de MoProSoft.
- Agregar las actividades de Scrum a las actividades de los procesos de la categoría de Operación de MoProSoft, si no existen sus equivalentes.
- Agregar los roles de Scrum a los roles de los procesos de la categoría de Operación de MoProSoft, si no existen sus equivalentes. O aumentar las responsabilidades de los roles equivalentes, según sea necesario.

- Agregar a los productos de los procesos de la categoría de Operación de MoProSoft los artefactos sugeridos por Scrum, sino existen sus equivalentes.

A continuación se presentan las actividades, roles y productos del nuevo patrón de procesos, siguiendo el orden en que son descritos en [NYCE05].

### 3.3.1. Administración de Proyectos Específicos

#### 3.3.1.1. Productos

Los productos del proceso en MoProSoft – Scrum están divididos en 3 tablas, la Tabla 26 muestra las entradas necesarias, la Tabla 27 muestra las salidas producidas y la Tabla 28 muestra los productos internos entre actividades.

**Tabla 26.** Tabla de entradas del proceso OPE.1 APE de MoProSoft – Scrum.

Nombre	Fuente
Plan de Comunicación e Implantación	Gestión de Negocio
Plan de Procesos	Gestión de Procesos
Plan de Mediciones de Proceso	
Documentación de Procesos	Gestión de Procesos
Desarrollo y Mantenimiento de Software	
Descripción del Proyecto	Gestión de Proyectos
Descripción del Producto	
Alcance	
Objetivos	
Entregables	
Responsable de Administración del Proyecto Específico	Gestión de Proyectos
Metas Cuantitativas del Proyecto	Gestión de Proyectos
Acciones Correctivas o Preventivas	Gestión de Proyectos
Asignación de Recursos	Recursos Humanos y Ambiente de Trabajo
Reporte de Actividades	Desarrollo y Mantenimiento de Software
Reporte de Mediciones y Sugerencias de Mejor	Desarrollo y Mantenimiento de Software
Configuración de Software	Desarrollo y Mantenimiento de Software
Solicitud de Cambios	Cliente

**Tabla 27.** Tabla de salidas del proceso OPE.1 APE de MoProSoft – Scrum.

Nombre	Descripción	Destino
Reporte de Mediciones y Sugerencias de Mejora	Registro que contiene: <ul style="list-style-type: none"> <li>• Mediciones de los indicadores del proceso de Administración de Proyectos Específicos.</li> <li>• Sugerencias de mejora al proceso de Administración de Proyectos Específicos (métodos, herramientas, formatos, estándares, entre otros).</li> </ul>	Gestión de Proyectos
Plan del Proyecto	Documento formal usado como guía para la ejecución y control del proyecto. Está conformado por: <ul style="list-style-type: none"> <li>• Registro de Pendientes del Producto.</li> <li>• Ciclos y Actividades.</li> <li>• Tiempo Estimado.</li> <li>• Plan de Adquisiciones y Capacitación.</li> <li>• Equipo de Trabajo.</li> <li>• Costo Estimado.</li> <li>• Calendario.</li> <li>• Plan de Manejo de Riesgos.</li> <li>• Protocolo de Entrega.</li> </ul>	Gestión de Proyectos
Reporte de Seguimiento	Contiene el registro del avance de las actividades realizadas incluyendo las llevadas a cabo en el Plan de Manejo de Riesgos. El avance se registra por ciclo, incluyendo fecha de inicio y fin, costo real del proyecto, esfuerzo realizado, cambios implementados y clasificados por tipo, tiempo real invertido, defectos encontrados, tamaño de los productos y trabajo duplicado.	Gestión de Proyectos
Documento de Aceptación	Documento que establece la aceptación del Cliente sobre los entregables establecidos en el proyecto.	Gestión de Proyectos
Plan de Proyecto <ul style="list-style-type: none"> <li>• Plan de Adquisiciones y Capacitación</li> </ul>	Contiene la relación de recursos humanos, capacitación, materiales, equipo y herramientas necesarios para la ejecución del proyecto.	Gestión de Recursos
Lecciones Aprendidas	Registro de mejores prácticas, problemas recurrentes y experiencias exitosas en la solución de problemas encontrados durante el desarrollo del proyecto.	Conocimiento de la Organización
Plan de Desarrollo	Documento usado como guía para la ejecución del desarrollo o mantenimiento de software. Contiene: <ul style="list-style-type: none"> <li>• Descripción del Producto y Entregables: Comprende la descripción del producto que se va a construir o del cambio que se va a efectuar y la descripción de los entregables.</li> <li>• Proceso Específico: Incluye el proceso ajustado al proyecto que se debe aplicar (proceso definido a partir del proceso de la organización) o del acuerdo con el Cliente. Indica el número de ciclos y las fases de cada ciclo.</li> <li>• Equipo de Trabajo: Recursos humanos asignados al proyecto.</li> <li>• Calendario: Contiene las actividades que se deben llevar a cabo con las fechas de inicio y fin.</li> </ul>	Desarrollo y Mantenimiento de Software

**Tabla 28.** Tabla de productos internos del proceso OPE.1 APE de MoProSoft – Scrum.

Nombre	Descripción
Acciones Correctivas	Acciones establecidas para corregir una desviación o problema con respecto al cumplimiento del Plan de Proyecto y Plan de Desarrollo.
Minuta(s)	Documento que describe el objetivo de las reuniones realizadas, los puntos tratados y los acuerdos.
Reporte de Verificación	Registro de participantes, fecha, lugar, duración y defectos encontrados.
Reporte de Validación	Registro de participantes, fecha, lugar, duración y defectos encontrados.

### 3.3.1.2. Roles involucrados y capacitación

Los roles del proceso en “MoProSoft – Scrum” se encuentran condensados en la Tabla 29, la cual muestra los roles involucrados en el proceso de OPE.1 APE, así como la capacitación requerida para cubrir el puesto.

**Tabla 29.** Tabla de roles involucrados y capacitación del proceso OPE.1 APE de MoProSoft – Scrum.

Rol	Abreviatura	Capacitación
Responsable de Gestión de Proyectos	RGPY	Conocimiento sobre las actividades necesarias para llevar a cabo la gestión de proyectos.
Responsable de la Administración del Proyecto Específico	RAPE	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal, delegación y supervisión, finanzas, desarrollo de software, priorización del producto y comunicación con otros interesados.
Cliente	CL	Conocimiento en la expedición de Solicitudes de Cambios
Responsable del Subcontrato	RSC	Conocimiento en la administración de proyectos.
Responsable de Desarrollo y Mantenimiento de Software	RDM	Conocimiento y experiencia en el desarrollo y mantenimiento de software.
Equipo de Trabajo	ET	Conocimiento y experiencia de acuerdo a su rol.
Maestro de Scrum	MS	Capacidad para liderar y enseñar el marco de trabajo de Scrum a la organización y todos los participantes, servicio y facilitación para la realización de los eventos de Scrum.

### 3.3.1.3. Actividades

Las actividades del proceso en “MoProSoft – Scrum” se encuentran condensadas en la Tabla 30, la cual muestra, además de las actividades, las tareas y los roles necesarios para llevar a cabo cada actividad del proceso de OPE.1 APE.

**Tabla 30.** Tabla de actividades del proceso OPE.1 APE de MoProSoft – Scrum.

Rol	Descripción
	<i>A1. Planificación (O1)</i>
RGPY A1.1.	Revisar con el Responsable de Gestión de Proyectos la Descripción del Proyecto.

RAPE RDM		Generar y priorizar el Registro de Pendientes del Producto.
RAPE	A1.2	Con base en la Descripción, definir el Proceso Específico del proyecto a partir del proceso de Desarrollo y Mantenimiento de Software de la organización o a partir del acuerdo establecido con el Cliente. Se considera el alcance, la magnitud y complejidad del proyecto.
RAPE CL	A1.3.	Definir conjuntamente con el Cliente el Protocolo de Entrega de cada uno de los entregables especificados en la Descripción del Proyecto.
RAPE	A1.4.	Identificar el número de ciclos y las actividades especificadas que deben llevarse a cabo para producir los entregables y sus componentes identificados en la Descripción del Proyecto. Identificar las actividades específicas que deben llevarse a cabo para cumplir con los objetivos del proyecto, definir las actividades para llevar a cabo revisiones periódicas al producto o servicio que se está ofreciendo y para revisiones entre colegas. Identificar las actividades para llevar a cabo el Protocolo de Entrega. Documentar el resultado como Ciclos y Actividades.
RAPE	A1.5.	Identificar y documentar la relación y dependencia de cada una de las actividades.
RAPE RDM	A1.6	Establecer el Tiempo Estimado para desarrollar cada actividad considerando la información histórica y las Metas Cuantitativas para el Proyecto.
RAPE	A1.7.	Elaborar el Plan de Adquisiciones y Capacitación, definiendo las características y el calendario en cuanto a recursos humanos, materiales, equipo y herramientas, incluyendo la capacitación requerida para que el equipo de trabajo pueda desempeñar el proyecto.
RGPY RAPE	A1.8.	Conformar el Equipo de Trabajo, asignando roles y responsabilidades basándose en la Descripción del Proyecto.
RAPE	A1.9.	Asignar fechas de inicio y fin a cada una de las actividades para generar el Calendario de trabajo tomando en cuenta los recursos asignados, la secuencia y dependencia de las actividades.
RAPE	A1.10.	Evaluar y documentar el Costo Estimado del proyecto, tomando en cuenta las Metas Cuantitativas para el Proyecto.
RGPY RAPE RDM	A1.11.	Identificar, describir y evaluar los riesgos que pueden afectar el proyecto que contemple riesgos relacionados con el equipo de trabajo incluyendo al Cliente y a los usuarios, riesgos con la tecnología o la metodología, riesgos con la organización del proyecto (costo, tiempo, alcance y recursos) o riesgos externos al proyecto. Identificar la probabilidad e impacto de cada riesgo estimando sus implicaciones en los objetivos del proyecto (análisis cuantitativo). Priorizar los efectos de los riesgos sobre los objetivos del proyecto (análisis cualitativo). Desarrollar procedimientos para reducir el impacto de los riesgos. Documentar en el Plan de Manejo de Riesgos o actualizarlo.
RAPE	A1.12.	Generar el Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo. Además el Plan del Proyecto se puede actualizar a causa de Solicitud de Cambios por parte del Cliente, Acciones Correctivas o Preventivas provenientes de Gestión de Proyectos o Acciones Correctivas de este proceso.
RAPE ET MS	A1.13.	Realizar la Reunión de Planificación, y el Responsable de Administración del Proyecto Específico Explicará el Equipo de Trabajo el Plan del Proyecto. Generar el Plan de Desarrollo en función del Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo. Además el Plan de Desarrollo se puede actualizar a causa de Solicitud de Cambios por parte del Cliente, Acciones Correctivas o Preventivas provenientes de Gestión de Proyectos o Acciones Correctivas de este proceso.
RAPE RDM	A1.14.	Durante la Reunión de Planificación: Verificar el Plan del Proyecto y el Plan de Desarrollo (Ver1).

RAPE ET MS	A1.15. Durante la Reunión de Planificación: Corregir los defectos encontrados en el Plan del Proyecto y en el Plan de Desarrollo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RGPY	A1.16. Durante la Reunión de Planificación: Validar el Plan del Proyecto y el Plan de Desarrollo (Val1).
RAPE ET MS	A1.17. Durante la Reunión de Planificación: Corregir los defectos encontrados en el Plan del Proyecto y en el Plan de Desarrollo con base en el Reporte de Validación y obtener la aprobación de las correcciones.
RAPE RDM	A1.18. Dar inicio formal a un nuevo ciclo una vez que se haya asegurado el cumplimiento de las condiciones iniciales del ciclo.
<i>A2. Realización (O1, O2, O3)</i>	
RAPE RDM ET	A2.1. Realizar Reuniones Diarias para informar al RDM sobre el trabajo realizado el día anterior, el trabajo a realizar el día actual y los impedimentos encontrados para el logro de las tareas asignadas. Acordar con el Responsable de Desarrollo y Mantenimiento del proyecto la asignación de tareas al Equipo de Trabajo incluyendo a los subcontratistas.
RAPE RDM	A2.2. Acordar la distribución de la información necesaria al equipo de trabajo con base en el Plan de Comunicación e Implantación.
RAPE RDM	A2.3. Revisar con el Responsable de Desarrollo y Mantenimiento del proyecto la Descripción del Producto, el Equipo de Trabajo y Calendario.
RAPE RDM RSC	A2.4. Dar seguimiento al Plan de Adquisiciones y Capacitación. Aceptar o rechazar la Asignación de Recursos humanos, o subcontratistas. Distribuir los recursos a los miembros del equipo para que puedan llevar a cabo las actividades.
RAPE RSC	A2.5. Manejar la relación con subcontratistas que implica planificar, revisar y auditar las actividades, asegurando la calidad de los productos o servicios contratados y el cumplimiento con los estándares y especificaciones acordadas.
RAPE	A2.6. Recolectar y analizar los Reportes de Actividades, Reportes de Mediciones y Sugerencias de Mejora y productos de trabajo.
RAPE	A2.7. Registrar los costos y recursos reales del ciclo.
RAPE	A2.8. Revisar el Registro de Rastreo de los Requisitos del usuario a través del ciclo.
RAPE RDM	A2.9. Revisar los productos generados durante el ciclo, que forma parte de la Configuración de Software.
RAPE RDM ET MS	A2.10. Recibir y analizar las Solicitudes de Cambios e incorporar los cambios aprobados en el Plan del Proyecto y en el Plan de Desarrollo. En caso de cambios a Requisitos se incorporan al inicio de un nuevo ciclo. Estos cambios pueden ser recibidos antes, durante y después de la revisión del producto realizada en la Reunión de Revisión.
RAPE ET CL MS	A2.11 Conduce reuniones de revisión con el equipo de trabajo y con el Cliente, generando Minutas con puntos tratados y acuerdos tomados. En la Reunión de Revisión identificar los acuerdos para el ciclo actual que se terminaron y los acuerdos que no se terminaron pasarlos a un ciclo siguiente.
<i>A3. Evaluación y Control (O1)</i>	
RAPE ET	A3.1. Realizar la Reunión de Retrospectiva para identificar lo que se hizo bien y lo que se hizo mal durante el ciclo, identificar las mejoras al proceso que deben incorporarse en el siguiente ciclo, y agregarlas a las Acciones Correctivas que se establezcan.



MS		Evaluar el cumplimiento del Plan de Proyecto y el Plan de Desarrollo, con respecto al alcance, costo, calendario, equipo de trabajo, proceso y se establecen Acciones Correctivas.
RAPE RGPY	A3.2.	Dar Seguimiento y controlar el Plan de Manejo de Riesgos. Identificar nuevos riesgos y actualizar el plan.
RAPE	A3.3.	Generar el Reporte de Seguimiento del proyecto, considerando los Reportes de Actividades.
<i>A4. Cierre (O1)</i>		
RAPE CL	A4.1.	Formalizar la terminación del ciclo o del proyecto de acuerdo al Protocolo de Entrega establecido en el Plan del Proyecto y obtener el Documento de Aceptación.
RAPE RSC	A4.2.	Efectuar el cierre con subcontratistas de acuerdo al contrato establecido.
RAPE	A4.3.	Generar el Reporte de Mediciones y Sugerencias de Mejora de este proceso, de acuerdo al Plan de Mediciones de Procesos.
RAPE	A4.4.	Identificar las Lecciones Aprendidas e integrarlas a la Base de Conocimiento. Como ejemplo se pueden considerar mejores prácticas, experiencias exitosas de manejo de riesgos problemas recurrentes, entre otras.

### 3.3.2. Desarrollo y Mantenimiento de Software

#### 3.3.2.1. Productos

Los productos del proceso en “MoProSoft – Scrum” para el proceso de OPE.2 DMS, están divididos en 3 tablas, las Tablas 31, 32 y 33 muestran las entradas necesarias, las salidas producidas y productos internos entre actividades, respectivamente.

**Tabla 31.** Tabla de entradas del proceso OPE.2 DMS de MoProSoft – Scrum.

Nombre	Fuente
Plan de Desarrollo, que incluye:	
<ul style="list-style-type: none"> <li>• Descripción del Producto.</li> <li>• Entregables.</li> <li>• Proceso Específico.</li> <li>• Equipo de Trabajo.</li> <li>• Calendario.</li> </ul>	Administración de Proyectos Específicos

**Tabla 32.** Tabla de salidas del proceso OPE.2 DMS de MoProSoft – Scrum.

Nombre	Descripción	Destino
Especificación de Requisitos	<p>Se compone de una introducción y una descripción de Requisitos.</p> <ul style="list-style-type: none"> <li>• Introducción: Descripción general del software y su uso en el ámbito de negocio del cliente.</li> <li>• Descripción de Requisitos: <ul style="list-style-type: none"> <li>○ Funcionales: Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y seguras.</li> <li>○ Interfaz con usuario: Definición de aquellas características de la interfaz de usuario que permiten</li> </ul> </li> </ul>	Administración de Proyectos Específicos

	<p>que el software sea fácil de entender, aprender, que genere satisfacción con el cual el usuario pueda desempeñar su tarea eficientemente. Incluyendo la descripción del prototipo de la interfaz.</p> <ul style="list-style-type: none"> <li>○ Interfaces externas: Definición de las interfaces con otro software o con hardware.</li> <li>○ Confiabilidad: Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación.</li> <li>○ Eficiencia: Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos.</li> <li>○ Mantenimiento: Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software.</li> <li>○ Portabilidad: Descripción de las características del software que permitan su transferencia de un ambiente a otro.</li> <li>○ Restricciones de diseño y construcción: Necesidades impuestas por el cliente.</li> <li>○ Legales y reglamentarios: Necesidades impuestas por leyes, reglamentos, entre otros.</li> </ul>	
Análisis y Diseño	<p>Este documento contiene la descripción textual y gráfica de la estructura de los componentes de software, consta de las siguientes partes:</p> <ul style="list-style-type: none"> <li>• Arquitectónica: Contiene la estructura interna del sistema, es decir la descomposición del sistema en subsistemas. Así como la identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.</li> <li>• Detallada: Contiene el detalle de los componentes que permita de manera evidente su construcción y prueba en el ambiente de programación.</li> </ul>	Administración de Proyectos Específicos
Componente	Conjunto de unidades de código relacionadas	Administración de Proyectos Específicos
Software	Sistema de software, destinado a un cliente o usuario, constituido por componentes agrupados en subsistemas, posiblemente anidados.	Administración de Proyectos Específicos
Configuración de Software	<p>Conjunto consistente de productos de software que incluye:</p> <ul style="list-style-type: none"> <li>• Especificación de Requisitos</li> <li>• Análisis y Diseño</li> <li>• Software</li> <li>• Registro de Rastreo</li> <li>• Plan de Pruebas del Sistema</li> <li>• Reporte de Pruebas de Sistema</li> <li>• Plan de Pruebas de Integración</li> <li>• Reporte de Pruebas de Integración</li> <li>• Manual de Usuario</li> </ul>	Administración de Proyectos Específicos

	<ul style="list-style-type: none"> <li>• Manual de Operación</li> <li>• Manual de Mantenimiento</li> </ul>	
Manual de Usuario	Documento electrónico o impreso que describe la forma de uso del software con base a la interfaz del usuario. Éste deberá ser redactado en términos comprensibles a los usuarios.	Administración de Proyectos Específicos
Manual de Operación	Documento electrónico o impreso que contenga la información indispensable para la instalación y administración del software, así como el ambiente de operación (sistema operativo, base de datos, servidores, etc.). Este deberá ser redactado en términos comprensibles al personal responsable de la operación.	Administración de Proyectos Específicos
Manual de Mantenimiento	Documento electrónico o impreso que describe la Configuración de Software y el ambiente usado para el desarrollo y pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este deberá ser redactado en términos comprensibles al personal de mantenimiento.	Administración de Proyectos Específicos
Reporte de Actividades	<p>Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como:</p> <ul style="list-style-type: none"> <li>• tiempo de producción, de corrección, de verificación y de validación,</li> <li>• defectos encontrados en verificación, validación o prueba,</li> <li>• tamaño de productos</li> </ul>	Administración de Proyectos Específicos
Lecciones Aprendidas	Registro de mejores prácticas, problemas recurrentes y experiencias exitosas en la solución de problemas, encontrados en un ciclo de desarrollo y mantenimiento.	Administración de Proyectos Específicos
Reporte de Mediciones y Sugerencias de Mejora	<p>Registro que contiene:</p> <ul style="list-style-type: none"> <li>• Mediciones de los indicadores del proceso de Desarrollo y Mantenimiento de Software (Ver Mediciones en la norma).</li> </ul>	Administración de Proyectos Específicos
Registro de Rastreo	Relación entre Requisitos, elementos de análisis y diseño, componentes y planes de pruebas.	Administración de Proyectos Específicos
Plan de Pruebas de Sistema	Identificación de pruebas requeridas para el cumplimiento de los Requisitos especificados.	Administración de Proyectos Específicos
Reporte de Pruebas de Sistema	Registro de participantes, fechas, lugar, duración y de defectos encontrados.	Administración de Proyectos Específicos
Plan de Pruebas de Integración	<p>Descripción que contiene:</p> <ul style="list-style-type: none"> <li>• El orden de integración de los componentes o subsistemas, guiado por la parte arquitectónica del Análisis y Diseño.</li> <li>• Pruebas que se aplicarán para verificar la interacción entre los componentes.</li> </ul>	Administración de Proyectos Específicos
Reporte de Pruebas de Integración	Registro de participantes, fechas, lugar, duración y de defectos encontrados.	Administración de Proyectos Específicos

**Tabla 33.** Tabla de productos internos del proceso OPE.2 DMS de MoProSoft – Scrum.

Nombre	Descripción
Reporte(s) de Verificación	Registro de participantes, fecha, lugar, duración y defectos encontrados.
Reporte(s) de Validación	Registro de participantes, fecha, lugar, duración y defectos encontrados.

### 3.3.2.2. Roles involucrados y capacitación

Para el proceso de OPE.2 DMS de “MoProSoft – Scrum” se requieren los roles explicados en la Tabla 34, dicha tabla incluye también la capacitación requerida para el puesto.

**Tabla 34.** Tabla de roles involucrados y capacitación del proceso OPE.2 DMS de MoProSoft – Scrum.

Rol	Abreviatura	Capacitación
Responsable de la Administración del Proyecto Específico	RAPE	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software.
Responsable de Desarrollo y Mantenimiento de Software	RDM	Conocimiento y experiencia en el desarrollo y mantenimiento de software.
Analista	AN	Conocimiento y experiencia en la obtención, especificación y análisis de los Requisitos.
Diseñador de Interfaz de Usuario	DU	Conocimiento en diseño de interfaces de usuario y criterios ergonómicos.
Diseñador	DI	Conocimiento y experiencia en el diseño de la estructura de los componentes de software.
Programador	PR	Conocimiento y/o experiencia en la programación, integración y pruebas unitarias.
Responsable de Pruebas	RPU	Conocimiento y experiencia en la planificación y realización de integración y pruebas de sistema.
Revisor	RE	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software.
Responsable de Manuales	RM	Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de software.
Equipo de Trabajo	ET	Conocimiento y experiencia de acuerdo a su rol.
Cliente	CL	Interpretación del estándar de la especificación de Requisitos.
Usuario	US	Ninguna.

### 3.3.2.3. Actividades

Las actividades del proceso OPE.2 DMS de “MoProSoft – Scrum” se encuentran condensadas en la Tabla 35, la cual muestra, además de las actividades, las tareas y roles que sean necesarios para llevar a cabo cada actividad del proceso.

**Tabla 35.** Tabla de actividades del proceso OPE.2 DMS de MoProSoft – Scrum.

Rol	Descripción
<i>A1. Realización de la fase de Inicio (O3)</i>	
ET	A1.1. Revisar con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un

		entendimiento común y obtener su compromiso con el proyecto.
RDM	A1.2.	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<i>A2. Realización de la fase de Requisitos (O1, O3)</i>		
RDM AN	A2.1.	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
AN CL US DU	A2.2.	Documentar o modificar la Especificación de Requisitos. <ul style="list-style-type: none"> <li>Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos Requisitos.</li> <li>Analizar los Requisitos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.</li> <li>Elaborar o modificar el prototipo de la interfaz con el usuario.</li> <li>Generar o actualizar la Especificación de Requisitos.</li> </ul>
RE	A2.3	Verificar la Especificación de Requisitos (Ver1).
AN DU	A2.4.	Corregir los defectos encontrados en la Especificación de Requisitos con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
CL US RPU	A2.5.	Validar la Especificación de Requisitos (Val1).
AN DU	A2.6.	Corregir los defectos encontrados en la Especificación de Requisitos con base en el Reporte de Validación y obtener la aprobación de las correcciones.
RPU AN	A2.7.	Elaborar o modificar Plan de Pruebas de Sistema.
RE	A2.8.	Verificar el Plan de Pruebas de Sistema (Ver2).
RPU	A2.9.	Corregir los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RM	A2.10.	Documentar la versión preliminar del Manual de Usuario o modificar el manual existente.
RE	A2.11.	Verificar el Manual de Usuario. (Ver3).
RM	A2.12.	Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A2.13.	Incorporar Especificación de Requisitos, Plan de Pruebas de Sistema y Manual de Usuario como líneas base a la Configuración de Software.
RDM	A2.14.	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<i>A3. Realización de la fase de Análisis y Diseño (O1, O3)</i>		
RDM AN DI	A3.1.	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
AN DI DU	A3.2.	Documentar o modificar el Análisis y Diseño: <ul style="list-style-type: none"> <li>Analizar la Especificación de Requisitos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y estos a su vez en componentes, definiendo las interfaces entre ellos.</li> </ul>

	<ul style="list-style-type: none"> <li>• Describir el detalle de la apariencia y el comportamiento de la interfaz con base en la Especificación de Requisitos de forma que se puedan prever los recursos para su implementación.</li> <li>• Descripción de los componentes que permita su construcción de manera evidente.</li> <li>• Generar o actualizar el Análisis y Diseño.</li> <li>• Generar o actualizar el Registro de Rastreo.</li> </ul>
RE	A3.3. Verificar el Análisis y Diseño y el Registro de Rastreo (Ver4).
AN DI DU	A3.4. Corregir los defectos encontrados en el Análisis y Diseño y en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
CL	A3.5. Verificar el Análisis y Diseño (Val2).
AN DI DU	A3.6. Corregir los defectos encontrados en el Análisis y Diseño con base en el Reporte de Validación y obtener la aprobación de las correcciones.
RPU	A3.7. Elaborar o modificar Plan de Pruebas de Integración.
RE	A3.8. Verificar el Plan de Pruebas de Integración (Ver5).
RPU	A3.9. Corregir los defectos encontrados en el Plan de Pruebas de Integración con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A3.10. Incorporar Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base a la Configuración de Software.
RDM	A3.11. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<i>A4. Realización de la fase de Construcción (O1, O3)</i>	
RDM	A4.1. Distribuir a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
PR	<p>A4.2. Construir o modificar el(los) Componente(s) de software:</p> <ul style="list-style-type: none"> <li>• Implementar o modificar Componente(s) con base a la parte detallada del Análisis y Diseño.</li> <li>• Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del Análisis y Diseño.</li> <li>• Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos).</li> <li>• Actualizar el Registro de Rastreo, incorporando los componentes construidos o modificados.</li> </ul>
RE	A4.3. Verificar el Registro de Rastreo (Ver6).
PR	A4.4. Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A4.5. Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software.
RDM	A4.6. Elaborar el Reporte de Actividades, registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<i>A5. Realización de la fase de Integración y Pruebas (O1, O3)</i>	
RDM	A5.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
PR RPU	<p>A5.2. Realizar integración y pruebas.</p> <ul style="list-style-type: none"> <li>• Integrar los componentes en subsistemas o en el sistema de Software y aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración.</li> <li>• Corregir los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una</li> </ul>

	prueba de integración exitosa (sin defectos).
	<ul style="list-style-type: none"> <li>• Actualizar el Registro de Rastreo.</li> </ul>
RM	A5.3. Documentar el Manual de Operación o modificar el manual existente.
RE	A5.4. Verificar el Manual de Operación (Ver7).
RM	A5.5. Corregir los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RPU	A5.6. Realizar las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de Sistema.
PR	A5.7. Corregir los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtener la aprobación de las correcciones.
RM	A5.8. Documentar el Manual de Usuario o modificar el existente.
RE	A5.9. Verificar el Manual de usuario (Ver8).
RM	A5.10. Corregir los defectos encontrados en el manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A5.11. Incorporar Software, Reporte de Pruebas de Integración, Registro de Rastreo, Manual de Operación y Manual de Usuario como líneas base a la Configuración de Software.
RDM	A5.12. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<i>A6. Realización de la fase de Cierre (O2)</i>	
RM	A6.1. Documenta el Manual de Mantenimiento o modificar el existente.
RE	A6.2. Verificar el Manual de Mantenimiento (Ver9).
RM	A6.3. Corregir los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A6.4. Incorporar Manual de Mantenimiento como línea base a la Configuración de Software.
RDM ET	A6.5. Identificar las Lecciones Aprendidas e integrarlas al a Base de Conocimiento. Como ejemplo, se pueden considerar mejores prácticas, experiencias exitosas de manejo de riesgos, problemas recurrentes, entre otras.
RDM ET	A6.6. Generar el Reporte de Mediciones y Sugerencias de Mejora.
RDM	A6.7. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.





## 4. Soporte computacional para la integración de MoProSoft y Scrum

En este capítulo se describe el diseño de la herramienta computacional que permite implementar la mezcla de MoProSoft y Scrum, descrito en el Capítulo 3. Este capítulo está organizado de la siguiente forma: las secciones 4.1 y 4.2 contienen los requisitos funcionales de la herramienta, así como una descripción de los casos de uso a nivel general, la sección 4.3 describe un diseño de alto nivel que incluirá el modelo entidad/relación de los datos y una descripción más exhaustiva de los casos de uso, ampliando la descripción de la sección 4.2; la sección 4.4 contiene el diseño de los diagramas de clases para los servicios de la aplicación, del lado del servidor; y por último, la sección 4.5 describe los componentes visuales utilizados por la herramienta.

### 4.1. Requisitos funcionales

Los requisitos funcionales son agrupados por áreas funcionales, para cada requisito funcional se dará una breve explicación del mismo así como unos valores de entrada y salida (si los hubiera). Se hace referencia también a qué caso de uso pertenece cada requisito, los cuales son descritos a detalle en la sección 4.2 de este capítulo.

#### 4.1.1. Gestión de acceso

Tabla 36. Requisito funcional 1.1: Iniciar sesión.

<b>Nombre</b>	Iniciar sesión.
<b>Descripción</b>	Solicitar los datos de acceso que permitirán al usuario iniciar sesión en el sistema.
<b>Entrada</b>	El usuario registrado introducirá los siguientes datos: Nombre de usuario, contraseña.
<b>Salida</b>	La sesión del usuario registrado estará registrada en el sistema.
<b>Proceso</b>	<ol style="list-style-type: none"><li>1. El usuario ingresa al sistema.</li><li>2. El sistema solicita al usuario que introduzca sus datos de acceso.</li><li>3. El usuario introduce y envía sus datos de acceso al sistema.</li><li>4. El sistema valida los datos de acceso del usuario con los datos de su base de datos.</li><li>5. El sistema muestra las opciones correspondientes a los roles del usuario.</li></ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de acceso.

**Tabla 37.** Requisito funcional 1.2: Editar información de acceso.

<b>Nombre</b>	Editar información de acceso.
<b>Descripción</b>	Edición de los datos del usuario que ha iniciado sesión.
<b>Entrada</b>	Se introducirán los siguientes datos: Nombre de usuario, nombre completo, contraseña.
<b>Salida</b>	Los datos de usuario se encontrarán registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Editar información de acceso.</li> <li>2. El sistema muestra un formulario con los datos del usuario.</li> <li>3. El usuario introduce y envía los datos del usuario al sistema.</li> <li>4. El sistema actualiza los datos del usuario en su base de datos.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de acceso.

**Tabla 38.** Requisito funcional 1.3: Cerrar sesión.

<b>Nombre</b>	Cerrar sesión.
<b>Descripción</b>	Finalizar la sesión del usuario que ha iniciado sesión, invalidándola y requiriendo nuevamente sus datos de acceso al sistema.
<b>Entrada</b>	La sesión del usuario que ha iniciado sesión.
<b>Salida</b>	La sesión del usuario no será válida en el sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Cerrar sesión.</li> <li>2. El sistema invalida la sesión del usuario en el sistema.</li> <li>3. El sistema mueve al usuario fuera del sistema.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de acceso.

#### 4.1.2. Gestión de usuarios

**Tabla 39.** Requisito funcional 2.1: Listar usuarios.

<b>Nombre</b>	Listar usuarios.
<b>Descripción</b>	Entregar la lista de usuarios registrados en el sistema.
<b>Entrada</b>	Ninguna.
<b>Salida</b>	El listado de usuarios registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador selecciona la opción Listar usuarios.</li> <li>2. El sistema obtiene la lista de usuarios de su base de datos.</li> <li>3. El sistema muestra la lista de usuarios del sistema.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de usuario.

**Tabla 40.** Requisito funcional 2.2: Crear usuario.

<b>Nombre</b>	Crear usuario.
<b>Descripción</b>	Crear usuario y definición de sus datos de acceso.
<b>Entrada</b>	Se introducirán los siguientes datos: Nombre de usuario, nombre completo, contraseña.
<b>Salida</b>	Los datos de usuario se encontrarán registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar usuarios.</li> <li>2. El usuario selecciona la opción Crear usuario.</li> <li>3. El sistema muestra un formulario para el ingreso de los datos del usuario nuevo.</li> <li>4. El usuario introduce y envía los datos del usuario al sistema.</li> <li>5. El sistema registra los datos del nuevo usuario en su base de datos.</li> <li>6. El sistema actualiza la lista de usuarios del sistema.</li> </ol>
<b>Referencias</b>	Listar usuarios.
<b>Casos de uso</b>	Gestión de usuarios.

**Tabla 41.** Requisito funcional 2.3: Editar usuario.

<b>Nombre</b>	Editar usuario.
<b>Descripción</b>	Edición de los datos de un usuario registrado en el sistema.
<b>Entrada</b>	Se modificarán los siguientes datos: Nombre completo, contraseña.
<b>Salida</b>	Los datos de usuario se encuentran actualizados en la base de datos del sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar usuarios.</li> <li>2. El usuario selecciona el usuario a ser dado modificado.</li> <li>3. El usuario selecciona la opción Editar usuario.</li> <li>4. El sistema muestra un formulario con los datos del usuario.</li> <li>5. El usuario actualiza y envía los datos del usuario al sistema.</li> <li>6. El sistema actualiza los datos del usuario en su base de datos.</li> <li>7. El sistema actualiza la lista de usuarios del sistema.</li> </ol>
<b>Referencias</b>	Listar usuarios.
<b>Casos de uso</b>	Gestión de usuarios.

**Tabla 42.** Requisito funcional 2.4: Borrar usuario.

<b>Nombre</b>	Borrar usuario.
<b>Descripción</b>	Baja permanente de un usuario registrado en el sistema.
<b>Entrada</b>	El usuario a ser dado de baja.
<b>Salida</b>	Los datos del usuario dado de baja ya no se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar usuarios.</li> <li>2. El usuario selecciona el usuario a ser dado de baja.</li> <li>3. El usuario selecciona la opción Borrar usuario.</li> </ol>

4. El sistema solicita confirmación para la eliminación del usuario.
5. El usuario confirma la eliminación del usuario.
6. El sistema elimina los datos del usuario de la base de datos.
7. El sistema actualiza la lista de usuarios del sistema.

**Referencias** Listar usuarios.

**Casos de uso** Gestión de usuarios.

### 4.1.3. Gestión de proyectos

**Tabla 43.** Requisito funcional 3.1: Listar proyectos.

<b>Nombre</b>	Listado de proyectos.
<b>Descripción</b>	Entregar la lista de proyectos del usuario que ha iniciado sesión.
<b>Entrada</b>	Usuario que ha iniciado sesión.
<b>Salida</b>	El listado de proyectos en los cuales el usuario tiene un rol.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El sistema obtiene los proyectos del usuario desde su base de datos.</li> <li>2. El sistema muestra la lista de proyectos del usuario.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de proyectos.

**Tabla 44.** Requisito funcional 3.2: Cargar proyecto.

<b>Nombre</b>	Cargar proyecto.
<b>Descripción</b>	Cargar la información de un proyecto del usuario que ha iniciado sesión.
<b>Entrada</b>	El proyecto a ser cargado.
<b>Salida</b>	Se carga el Registro de pendientes del producto y el Registro de pendientes de los Sprints, correspondientes al proyecto.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Cargar proyecto.</li> <li>2. El sistema ejecuta la opción Listar proyectos.</li> <li>3. El usuario selecciona el proyecto a ser accedido.</li> <li>4. El sistema obtiene los datos relacionados al proyecto desde su base de datos.</li> <li>5. El sistema ejecuta la opción Listar pendientes de producto para el proyecto cargado.</li> <li>6. El sistema ejecuta la opción Listar pendientes de sprints para el proyecto cargado.</li> </ol>
<b>Referencias</b>	<p>Listar proyectos.</p> <p>Listar pendientes de producto.</p> <p>Listar pendientes de sprints.</p>
<b>Casos de uso</b>	Gestión de proyectos.

**Tabla 45.** Requisito funcional 3.3: Cerrar proyecto.

<b>Nombre</b>	Cerrar proyecto.
<b>Descripción</b>	Deja de mostrar la información de un proyecto del usuario que ha iniciado sesión.
<b>Entrada</b>	El proyecto a ser cerrado.

<b>Salida</b>	Se deja de mostrar el Registro de pendientes del producto y el Registro de pendientes de los Sprints, correspondientes al proyecto.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Cerrar proyecto.</li> <li>2. El sistema deja de mostrar en pantalla el Registro de pendientes del producto y los Registros de pendientes del Sprint.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de proyectos.

**Tabla 46.** Requisito funcional 3.4: Crear proyecto.

<b>Nombre</b>	Crear proyecto.
<b>Descripción</b>	Crear un proyecto, con el usuario como Maestro de Scrum del proyecto.
<b>Entrada</b>	Se introducirán los siguientes datos para crear un proyecto: Nombre, descripción, alcance y nivel de capacidades.
<b>Salida</b>	Los datos del nuevo proyecto se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Crear proyecto.</li> <li>2. El sistema muestra un formulario para el ingreso de los datos del proyecto nuevo.</li> <li>3. El usuario introduce y envía los datos del proyecto nuevo al sistema.</li> <li>4. El sistema registra los datos del nuevo proyecto en su base de datos.</li> <li>5. El sistema ejecuta la opción Cargar proyecto de manera automática.</li> </ol>
<b>Referencias</b>	Cargar proyecto.
<b>Casos de uso</b>	Gestión de proyectos.

**Tabla 47.** Requisito funcional 3.5: Editar datos de proyecto.

<b>Nombre</b>	Editar datos de proyecto.
<b>Descripción</b>	Editar los datos de un proyecto cargado.
<b>Entrada</b>	Se editarán los siguientes datos de un proyecto: Nombre, descripción, alcance, nivel de capacidades.
<b>Salida</b>	Los datos actualizados del proyecto se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Editar datos de proyecto.</li> <li>2. El sistema muestra un formulario con los datos del proyecto.</li> <li>3. El usuario introduce y envía los datos actualizados del proyecto al sistema.</li> <li>4. El sistema registra los datos actualizados del proyecto en su base de datos.</li> <li>5. El sistema actualiza los datos del proyecto que se despliegan en pantalla.</li> </ol>
<b>Referencias</b>	Cerrar proyecto.
<b>Casos de uso</b>	Gestión de proyectos.

**Tabla 48.** Requisito funcional 3.6: Borrar proyecto.

<b>Nombre</b>	Borrar proyecto
<b>Descripción</b>	Baja permanente de un proyecto registrado en el sistema.

<b>Entrada</b>	El proyecto a ser dado de baja.
<b>Salida</b>	Los datos del proyecto ya no se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Editar datos de proyecto.</li> <li>2. El usuario selecciona la opción Borrar proyecto.</li> <li>3. El usuario confirma la eliminación del proyecto.</li> <li>4. El sistema cierra el proyecto cargado.</li> <li>5. El sistema elimina los datos del proyecto de la base de datos.</li> </ol>
<b>Referencias</b>	Editar datos de proyecto.
<b>Casos de uso</b>	Gestión de proyecto.

#### 4.1.4. Gestión de equipos de proyecto

**Tabla 49.** Requisito funcional 4.1: Listar miembros.

<b>Nombre</b>	Listar miembros.
<b>Descripción</b>	Entrega la lista de miembros que han sido dados de alta en alguno de los roles del equipo de trabajo de un proyecto cargado.
<b>Entrada</b>	El proyecto del que se desea ver el equipo de trabajo.
<b>Salida</b>	Entregar la lista de miembros del equipo del proyecto y sus roles correspondientes.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar miembros.</li> <li>2. El sistema obtiene los miembros del equipo de proyecto desde su base de datos.</li> <li>3. El sistema muestra en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de equipos de proyecto.

**Tabla 50.** Requisito funcional 4.2: Agregar miembro.

<b>Nombre</b>	Agregar miembro.
<b>Descripción</b>	Agrega un usuario registrado en el sistema como miembro de un equipo de trabajo de un proyecto cargado.
<b>Entrada</b>	Se introducirán los siguientes datos para dar de alta un miembro en el equipo de trabajo: Nombre de usuario y rol en el proyecto.
<b>Salida</b>	El usuario y rol relacionados con el equipo de trabajo de proyecto registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar miembros.</li> <li>2. El usuario selecciona la opción Agregar miembro.</li> <li>3. El sistema muestra un formulario para el ingreso de los datos del miembro nuevo.</li> <li>4. El usuario introduce y envía los datos al sistema.</li> <li>5. El sistema registra los datos actualizados en su base de datos.</li> <li>6. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.</li> </ol>
<b>Referencias</b>	Listar miembros.
<b>Casos de uso</b>	Gestión de equipos de proyecto.

**Tabla 51.** Requisito funcional 4.3: Editar rol de miembro.

<b>Nombre</b>	Editar rol de miembro.
<b>Descripción</b>	Editar el rol de un miembro del equipo de trabajo.
<b>Entrada</b>	Se editarán los siguientes datos de un miembro del equipo de trabajo: Rol en el proyecto.
<b>Salida</b>	El nuevo rol del miembro del equipo de trabajo ha sido actualizado en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar miembros.</li> <li>2. El usuario selecciona al miembro al que se le cambiará el rol.</li> <li>3. El usuario selecciona la opción Editar rol de miembro.</li> <li>4. El sistema muestra un formulario con los datos del miembro del equipo.</li> <li>5. El usuario modifica y envía los datos al sistema.</li> <li>6. El sistema actualiza los datos del equipo en su base datos.</li> <li>7. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.</li> </ol>
<b>Referencias</b>	Listar miembros.
<b>Casos de uso</b>	Gestión de equipos de proyecto.

**Tabla 52.** Requisito funcional 4.4: Remover miembro.

<b>Nombre</b>	Remover miembro.
<b>Descripción</b>	Remover un miembro del equipo de trabajo de un proyecto.
<b>Entrada</b>	El miembro a ser borrado.
<b>Salida</b>	El miembro no existe más en el equipo de trabajo y ha sido actualizado el cambio en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Listar miembros.</li> <li>2. El usuario selecciona el miembro a ser removido.</li> <li>3. El usuario selecciona la opción Remover miembro.</li> <li>4. El usuario confirma la remoción del miembro del equipo de trabajo.</li> <li>5. El sistema actualiza los datos del equipo en su base de datos.</li> <li>6. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.</li> </ol>
<b>Referencias</b>	Listar miembros.
<b>Casos de uso</b>	Gestión de equipos de proyecto.

#### 4.1.5. Gestión de registros de pendientes de producto

**Tabla 53.** Requisito funcional 5.1: Listar pendientes de producto.

<b>Nombre</b>	Listar pendientes de producto.
<b>Descripción</b>	Muestra en pantalla el registro de pendientes de producto de un proyecto cargado, incluidas las tareas de cada pendiente.
<b>Entrada</b>	El proyecto del cual se mostrará el registro de pendientes del producto.
<b>Salida</b>	El registro de pendientes de producto para el proyecto cargado se encuentra visualizado en pantalla.

<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El sistema obtiene el registro de pendientes del producto del proyecto desde su base de datos.</li> <li>2. El sistema muestra en pantalla el registro de pendientes del producto del proyecto cargado.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

**Tabla 54.** Requisito funcional 5.2: Crear pendiente de producto.

<b>Nombre</b>	Crear pendiente de producto.
<b>Descripción</b>	Crear un pendiente al registro de pendientes de producto de un proyecto cargado.
<b>Entrada</b>	Se introducirán los siguientes datos del pendiente de producto: Nombre, descripción, beneficio, penalidad, esfuerzo y estado del pendiente.
<b>Salida</b>	Los datos del pendiente del producto se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Crear pendiente de producto.</li> <li>2. El sistema muestra el formulario para introducir los datos del pendiente de producto nuevo.</li> <li>3. El usuario introduce y envía los datos al sistema.</li> <li>4. El sistema registra el nuevo pendiente de producto en su base de datos.</li> <li>5. El sistema agrega las tareas para las actividades de DMS correspondientes al nivel de capacidad del proyecto.</li> <li>6. El sistema actualiza en pantalla el registro de pendientes de producto.</li> </ol>
<b>Referencias</b>	Agregar tarea.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

**Tabla 55.** Requisito funcional 5.3: Editar pendiente de producto.

<b>Nombre</b>	Editar pendiente de producto
<b>Descripción</b>	Edición de un pendiente del registro de pendientes de producto de un proyecto cargado.
<b>Entrada</b>	Se editarán los siguientes datos del pendiente de producto : Nombre, descripción, beneficio, penalidad, esfuerzo y estado del pendiente.
<b>Salida</b>	Los datos del pendiente de producto estarán actualizados en la base de datos del sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el pendiente de producto a ser editado.</li> <li>2. El usuario selecciona la opción Editar pendiente de producto.</li> <li>3. El usuario modifica y envía los datos al sistema.</li> <li>4. El sistema actualiza los datos del pendiente de producto en su base de datos.</li> <li>5. El sistema actualiza en pantalla el registro de pendientes de producto.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

**Tabla 56.** Requisito funcional 5.4: Borrar pendiente de producto.

<b>Nombre</b>	Borrar pendiente de producto.
<b>Descripción</b>	Borrar un pendiente del registro de pendientes de producto de un proyecto cargado.
<b>Entrada</b>	El pendiente de producto que se desea borrar del registro de pendientes del producto.



<b>Salida</b>	Los datos del pendiente de producto ya no se encuentran registrados en la base de datos del sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el pendiente de producto a ser borrado.</li> <li>2. El usuario selecciona la opción Borrar pendiente de producto.</li> <li>3. El usuario confirma el borrado del pendiente de producto.</li> <li>4. El sistema borra el pendiente de producto de su base de datos.</li> <li>5. El sistema actualiza en pantalla el registro de pendientes de producto.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

**Tabla 57.** Requisito funcional 5.5: Agregar pendiente a sprint.

<b>Nombre</b>	Agregar pendiente a sprint.
<b>Descripción</b>	Agregar un pendiente al registro de pendientes de un sprint, desde el registro de pendientes de producto.
<b>Entrada</b>	El pendiente de producto a ser agregado.
<b>Salida</b>	El pendiente de producto se encuentra relacionado en la base de datos a un registro de pendientes de un sprint.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el pendiente de producto a ser agregado.</li> <li>2. El usuario selecciona la opción Agregar pendiente a sprint.</li> <li>3. El usuario selecciona el sprint donde será agregado el pendiente.</li> <li>4. El sistema relaciona el pendiente de producto al sprint en su base de datos.</li> <li>5. El sistema actualiza en pantalla el registro de pendientes de producto.</li> <li>6. El sistema actualiza en pantalla el registro de pendientes de sprints.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

**Tabla 58.** Requisito funcional 5.6: Remover pendiente de sprint.

<b>Nombre</b>	Remover pendiente de sprint.
<b>Descripción</b>	Remover un pendiente del registro de pendientes de un sprint.
<b>Entrada</b>	El pendiente de sprint a ser removido.
<b>Salida</b>	El pendiente de producto ya no se encuentra relacionado en la base de datos al registro de pendientes de un sprint.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el pendiente de sprint a ser removido.</li> <li>2. El usuario selecciona la opción Remover pendiente de sprint.</li> <li>3. El sistema remueve la relación del pendiente con el sprint en su base de datos.</li> <li>4. El sistema actualiza en pantalla el registro de pendientes de producto.</li> <li>5. El sistema actualiza en pantalla el registro de pendientes de sprints.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de registros de pendientes de producto.

#### 4.1.6. Gestión de sprints

**Tabla 59.** Requisito funcional 6.1: Listar sprints.

<b>Nombre</b>	Listar sprints.
<b>Descripción</b>	Muestra en pantalla los sprints de un proyecto cargado.
<b>Entrada</b>	El proyecto del cual se mostrará los sprints.
<b>Salida</b>	Los sprints del proyecto cargado se encuentran visualizados en pantalla, incluidos los registros de pendientes de los sprints y sus tareas.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El sistema obtiene los sprints y sus registros de pendientes de cada sprint del proyecto cargado desde su base de datos.</li> <li>2. El sistema muestra en pantalla los sprints del proyecto cargado.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de sprints.

**Tabla 60.** Requisito funcional 6.2: Crear sprint.

<b>Nombre</b>	Crear sprint.
<b>Descripción</b>	Crear un sprint del proyecto cargado.
<b>Entrada</b>	Se introducirán los siguientes datos para crear un sprint: Nombre, objetivos, fecha de inicio y fecha de fin.
<b>Salida</b>	Los datos del nuevo sprint se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Crear sprint.</li> <li>2. El sistema muestra un formulario para el ingreso de los datos del sprint nuevo.</li> <li>3. El usuario introduce y envía los datos al sistema.</li> <li>4. El sistema registra los datos del nuevo sprint en su base de datos.</li> <li>5. El sistema actualiza en pantalla la lista de sprints del proyecto cargado.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de sprints.

**Tabla 61.** Requisito funcional 6.3: Editar sprint.

<b>Nombre</b>	Editar sprint.
<b>Descripción</b>	Editar los datos de un sprint de un proyecto cargado.
<b>Entrada</b>	Se podrán editar los siguientes datos para crear un sprint: Nombre, objetivos, fecha de inicio y fecha de fin.
<b>Salida</b>	Los datos actualizados del sprint se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el sprint que desea editar.</li> <li>2. El usuario selecciona la opción Editar sprint.</li> <li>3. El sistema muestra un formulario con los datos del sprint.</li> <li>4. El usuario actualiza y envía los datos del sprint al sistema.</li> <li>5. El sistema registra los datos actualizados del sprint en su base de datos.</li> <li>6. El sistema actualiza en pantalla la lista de sprints del proyecto cargado.</li> </ol>

<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de sprints.

**Tabla 62.** Requisito funcional 6.4: Borrar sprint.

<b>Nombre</b>	Borrar sprint
<b>Descripción</b>	Borrar permanente un sprint de un proyecto cargado.
<b>Entrada</b>	El sprint a ser borrado.
<b>Salida</b>	Los datos del sprint ya no se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona el sprint que desea borrar.</li> <li>2. El usuario selecciona la opción Borrar sprint.</li> <li>3. El usuario confirma la eliminación del sprint.</li> <li>4. El sistema elimina los datos del sprint de la base de datos.</li> <li>5. El sistema actualiza en pantalla la lista de sprints del proyecto cargado.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de sprints.

#### 4.1.7. Gestión de tareas

**Tabla 63.** Requisito funcional 7.1: Agregar tarea.

<b>Nombre</b>	Agregar tarea.
<b>Descripción</b>	Agregar una tarea a un pendiente de producto dentro o fuera de un sprint.
<b>Entrada</b>	El pendiente de producto al que se le agregará la tarea y los siguientes datos para crear una tarea: Nombre, descripción, esfuerzo estimado, esfuerzo actual, estado, miembro asignado, fecha de inicio y fecha de fin.
<b>Salida</b>	Los datos de la tarea del pendiente de producto se encuentran registrados en la base de datos de sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Agregar tarea.</li> <li>2. El sistema muestra el formulario para introducir los datos de la tarea.</li> <li>3. El usuario introduce y envía los datos al sistema.</li> <li>4. El sistema registra la nueva tarea en su base de datos.</li> <li>5. El sistema actualiza en pantalla el pendiente de producto donde se agregó la tarea.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de tareas.

**Tabla 64.** Requisito funcional 7.2: Editar tarea.

<b>Nombre</b>	Editar tarea.
<b>Descripción</b>	Edición de una tarea de un pendiente del registro de pendientes de producto asignado, o no, a un sprint.
<b>Entrada</b>	Se editarán los siguientes datos del pendiente de producto : Nombre, descripción, esfuerzo estimado, esfuerzo actual, estado, miembro asignado, fecha de inicio y fecha de fin.

<b>Salida</b>	Los datos de la tarea del pendiente de producto estarán actualizados en la base de datos del sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la tarea a ser editada.</li> <li>2. El usuario selecciona la opción Editar tarea.</li> <li>3. El usuario modifica y envía los datos al sistema.</li> <li>4. El sistema actualiza los datos de la tarea en su base de datos.</li> <li>5. El sistema actualiza en pantalla el pendiente de producto de la tarea editada.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de tareas.

**Tabla 65.** Requisito funcional 7.3: Borrar tarea.

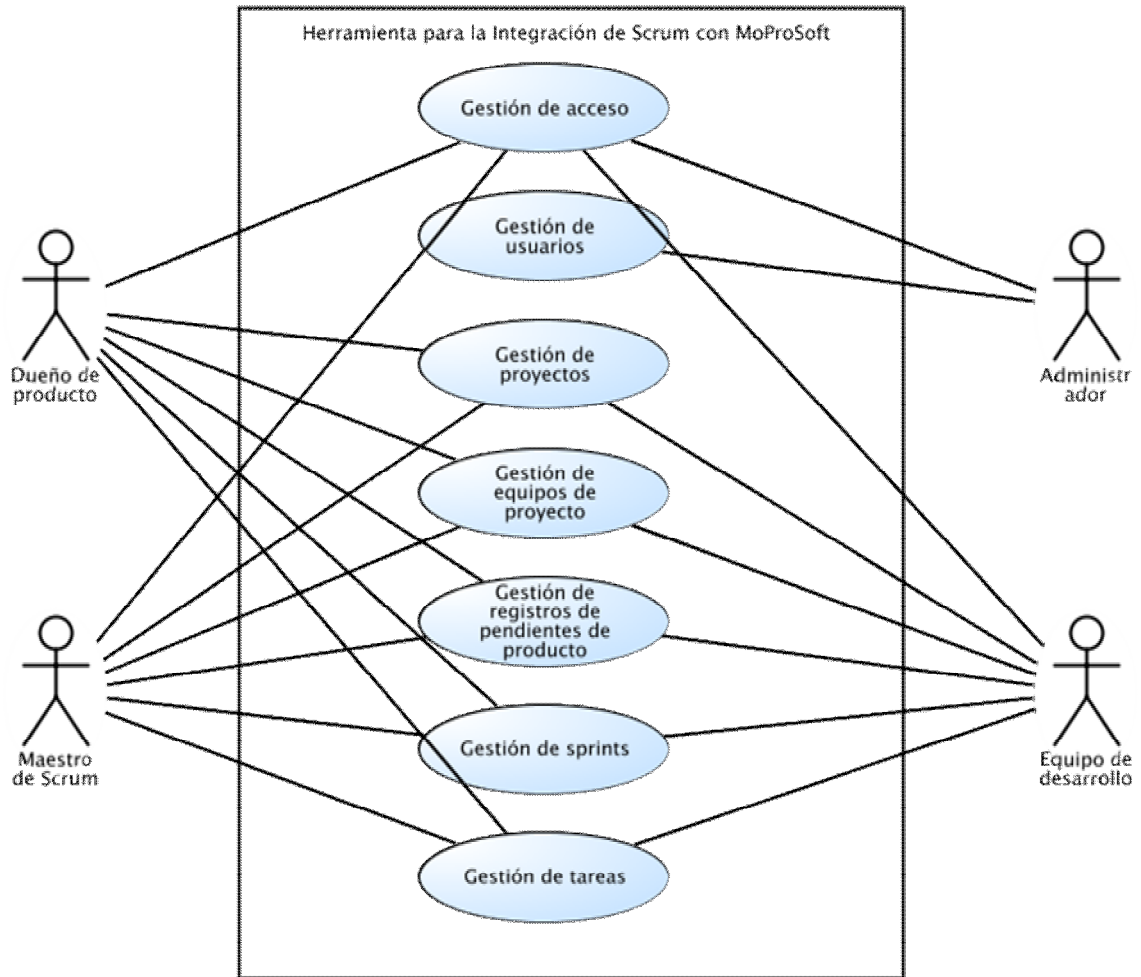
<b>Nombre</b>	Borrar tarea.
<b>Descripción</b>	Borrar una tarea de algún pendiente de producto, asignado o no a un sprint.
<b>Entrada</b>	La tarea que se desea borrar del pendiente de producto.
<b>Salida</b>	Los datos de la tarea ya no se encuentran registrados en la base de datos del sistema.
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la tarea a ser borrada.</li> <li>2. El usuario selecciona la opción Borrar tarea.</li> <li>3. El usuario confirma el borrado de la tarea.</li> <li>4. El sistema borra los datos de la tarea de su base de datos.</li> <li>5. El sistema actualiza en pantalla el pendiente de producto de la tarea borrada.</li> </ol>
<b>Referencias</b>	Ninguna.
<b>Casos de uso</b>	Gestión de tareas.

## 4.2. Casos de uso

A continuación se presentan los casos de uso de los requisitos funcionales listados en la Sección 4.1, los casos de uso estarán relacionados a algunos de los roles de los usuarios, según se indique más adelante (ver Figura 4.1).

Los roles de usuarios son los siguientes:

- **Administrador:** El usuario encargado de administrar el sistema. No es parte del equipo de trabajo de un proyecto.
- **Maestro de Scrum:** Dentro de un proyecto, el usuario asignado al rol de maestro de Scrum, según el marco de trabajo de Scrum.
- **Dueño de producto:** Dentro de un proyecto, el usuario asignado al rol de dueño de producto, según el marco de trabajo de Scrum.
- **Equipo de desarrollo:** Dentro de un proyecto, el usuario asignado al rol de equipo de desarrollo, según el marco de trabajo de Scrum.



**Figura 4.1.** Casos de uso para la herramienta para la integración MoProSoft - Scrum

La Figura 4.1 muestra los casos de uso de la herramienta de soporte a la mezcla propuesta, éstos pueden ser descritos en términos generales de la siguiente forma:

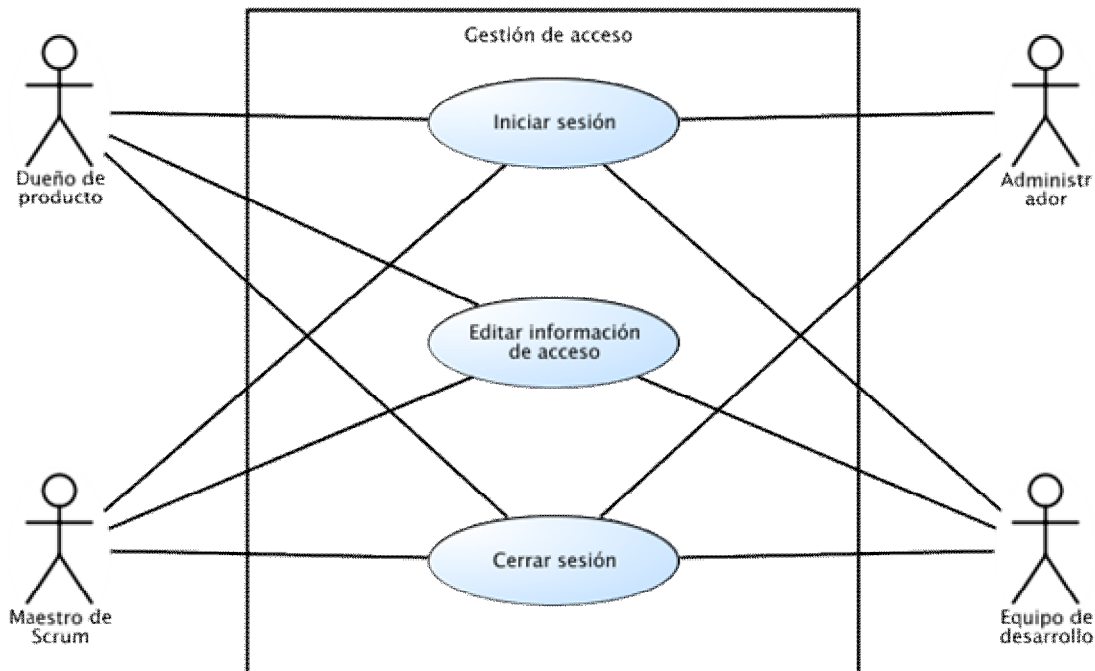
- **Gestión de acceso:** Permite a los usuarios registrados acceder al sistema, prohibiendo el acceso a usuarios no registrados.
- **Gestión de usuarios:** Permite listar, crear y eliminar usuarios, así como modificar la información de los usuarios.
- **Gestión de proyectos:** Permite listar, crear y eliminar proyectos, así como modificar los datos de estos.
- **Gestión de equipos de proyecto:** Permite listar, agregar y remover usuarios de los equipos de trabajo de los proyectos, así como cambiar los roles de los miembros del equipo de trabajo.
- **Gestión de registros de pendientes de producto:** Permite listar, crear y eliminar pendientes de los registros de pendientes de producto de los proyectos, también permite agregar o remover pendientes a los *sprints* de los proyectos.
- **Gestión de sprints:** Permite listar, crear y eliminar *sprints* de los proyectos, así como también editar los datos de los *sprints*.

- **Gestión de tareas:** Permite crear y eliminar tareas de los pendientes de los registros de pendientes de los proyectos, ya sea que se encuentren en un *sprint* o no, también permite editar la información de las tareas.

#### 4.2.1. Casos de uso detallados

##### 4.2.1.1. Gestión de acceso

Este caso de uso se divide en tres casos de uso, como se muestra en la Figura 4.2, con las funcionalidades que permitirán iniciar y cerrar sesión para usuarios registrados, y editar los datos del usuario por los mismos usuarios.



**Figura 4.2.** Casos de uso de Gestión de acceso

- **Caso de uso:** Iniciar sesión.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de producto, Equipo de desarrollo.
  - **Propósito:** Acceder al sistema mediante el inicio de sesión.
  - **Visión general:** El usuario accede al sistema mediante el ingreso de sus datos de acceso.
  - **Referencias:** Requisito funcional 1.1: Iniciar sesión.
  - **Curso típico de eventos:**
    1. El usuario ingresa al sistema.
    2. El sistema solicita al usuario que introduzca sus datos de acceso.
    3. El usuario introduce y envía sus datos de acceso al sistema.

4. El sistema válida los datos de acceso del usuario con los datos de su base de datos.
  5. El sistema muestra las opciones correspondientes a los roles del usuario.
- **Cursos alternativos:**
    - 4a. El sistema detecta invalidez en los datos enviados y despliega un mensaje de error. Regresa al paso 1.
- **Caso de uso:** Editar información de acceso.
    - **Actores:** Maestro de Scrum, Dueño de producto, Equipo de desarrollo.
    - **Propósito:** El usuario pueda editar su información de acceso propia.
    - **Visión general:** El usuario edita su información de acceso.
    - **Referencias:** Requisito funcional 1.2: Editar información de acceso.
    - **Curso típico de eventos:**
      1. El usuario selecciona la opción Editar información de acceso.
      2. El sistema muestra un formulario con los datos del usuario.
      3. El usuario introduce y envía los datos del usuario al sistema.
      4. El sistema actualiza los datos del usuario en su base de datos.
    - **Cursos alternativos:**
      - 4a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 4.
- **Caso de uso:** Cerrar sesión.
    - **Actores:** Administrador, Maestro de Scrum, Dueño de producto, Equipo de desarrollo.
    - **Propósito:** Evitar el acceso al sistema mediante el cierre de sesión.
    - **Visión general:** El usuario cierra su sesión en el sistema, invalidando su sesión previamente iniciada.
    - **Referencias:** Requisito funcional 1.3: Cerrar sesión.
    - **Curso típico de eventos:**
      1. El usuario selecciona la opción Cerrar sesión.
      2. El sistema invalida la sesión del usuario en el sistema.
      3. El sistema mueve al usuario fuera del sistema.
    - **Cursos alternativos:** Ninguno.

#### 4.2.1.2. Gestión de usuarios

Este caso de uso entrega las funcionalidades de listar, crear y eliminar usuarios, y editar los datos de los usuarios registrados en el sistema, en la Figura 4.3 se muestran los 4 casos de uso en los cuales se divide la gestión de usuarios.

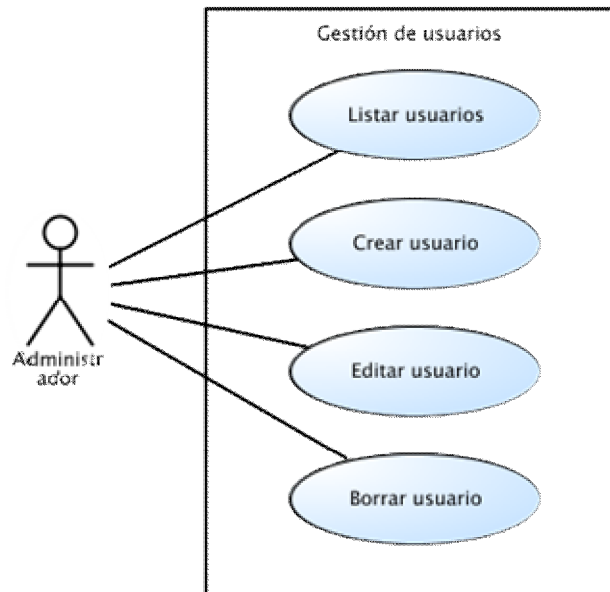


Figura 4.3. Casos de uso de Gestión de usuarios

- **Caso de uso:** Listar usuarios.
  - **Actores:** Administrador.
  - **Propósito:** Listar los usuarios registrados en el sistema.
  - **Visión general:** El usuario ve la lista de todos los usuarios existentes en el sistema.
  - **Referencias:** Requisito funcional 2.1: Listar usuarios.
  - **Curso típico de eventos:**
    1. El usuario administrador selecciona la opción Listar usuarios.
    2. El sistema obtiene la lista de usuarios de su base de datos.
    3. El sistema muestra la lista de usuarios del sistema.
  - **Cursos alternativos:** Ninguno.
  
- **Caso de uso:** Crear usuario.
  - **Actores:** Administrador.
  - **Propósito:** Crear un nuevo usuario en el sistema.
  - **Visión general:** El usuario crea un usuario para un nuevo miembro de equipos de proyectos.



- **Referencias:** Requisito funcional 2.2: Crear usuario, Requisito funcional 2.1 Listar usuarios.
- **Curso típico de eventos:**
  1. El usuario selecciona la opción Listar usuarios.
  2. El usuario selecciona la opción Crear usuario.
  3. El sistema muestra un formulario para el ingreso de los datos del usuario nuevo.
  4. El usuario introduce y envía los datos del usuario al sistema.
  5. El sistema registra los datos del nuevo usuario en su base de datos.
  6. El sistema actualiza la lista de usuarios del sistema.
- **Cursos alternativos:**
  - 4a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 4.
- **Caso de uso:** Editar usuario.
  - **Actores:** Administrador.
  - **Propósito:** Editar datos de un usuario del sistema.
  - **Visión general:** El usuario modifica los datos de un usuario ya registrado en el sistema.
  - **Referencias:** Requisito funcional 2.3: Editar usuario, Requisito funcional 2.1 Listar usuarios.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Listar usuarios.
    2. El usuario selecciona el usuario a ser dado modificado.
    3. El usuario selecciona la opción Editar usuario.
    4. El sistema muestra un formulario con los datos del usuario.
    5. El usuario actualiza y envía los datos del usuario al sistema.
    6. El sistema actualiza los datos del usuario en su base de datos.
    7. El sistema actualiza la lista de usuarios del sistema.
  - **Cursos alternativos:**
    - 5a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 5.
- **Caso de uso:** Borrar usuario.
  - **Actores:** Administrador.

- **Propósito:** Borrar un usuario registrado en el sistema.
- **Visión general:** El usuario borra un usuario registrado en el sistema.
- **Referencias:** Requisito funcional 2.4: Borrar usuario, Requisito funcional 2.1: Listar usuarios.
- **Curso típico de eventos:**
  1. El usuario selecciona la opción Listar usuarios.
  2. El usuario selecciona el usuario a ser dado de baja.
  3. El usuario selecciona la opción Borrar usuario.
  4. El sistema solicita confirmación para la eliminación del usuario.
  5. El usuario confirma la eliminación del usuario.
  6. El sistema elimina los datos del usuario de la base de datos.
  7. El sistema actualiza la lista de usuarios del sistema.
- **Cursos alternativos:**
  - 5a. El usuario cancela la eliminación del usuario. Regresa al paso 2.

#### 4.2.1.3. Gestión de proyectos

Como se muestra en la Figura 4.4, el caso de uso gestión de proyectos se divide en 6 casos de uso, con las funcionalidades de listar, crear, editar y eliminar proyectos, además de poder cargar y cerrar los proyectos en el espacio de trabajo.

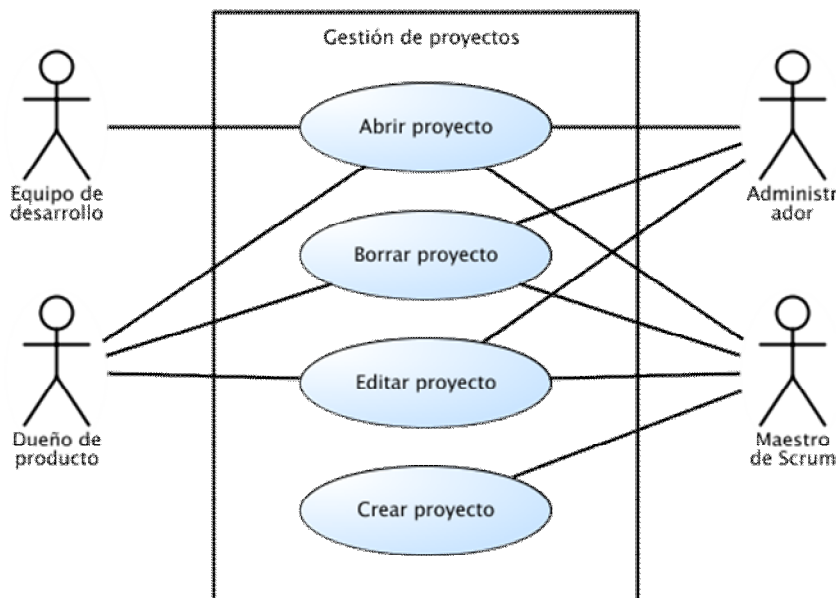


Figura 4.4. Casos de uso de Gestión de proyectos

- **Caso de uso:** Listar proyectos.
  - **Actores:** Administrador, Scrum Master, Dueño de proyecto, Equipo de Desarrollo.

- **Propósito:** Listar los proyectos de un usuario.
- **Visión general:** El usuario ve la lista de a los cuales tiene permiso de acceder, ya sea por ser administrador o por ser miembro del equipo de proyecto.
- **Referencias:** Requisito funcional 3.2: Listar proyecto.
- **Curso típico de eventos:**
  1. El sistema obtiene los proyectos del usuario desde su base de datos.
  2. El sistema muestra la lista de proyectos del usuario.
- **Cursos alternativos:** Ninguno.
  
- **Caso de uso:** Cargar proyecto.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.
  - **Propósito:** Cargar un proyecto en el espacio de trabajo.
  - **Visión general:** El usuario carga, en el espacio de trabajo, uno de los proyectos a los cuales tiene acceso.
  - **Referencias:** Requisito funcional 3.1: Listar proyectos, Requisito funcional 3.2: Cargar proyectos.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Cargar proyecto.
    2. El sistema ejecuta la opción Listar proyectos.
    3. El usuario selecciona el proyecto a ser accedido.
    4. El sistema obtiene los datos relacionados al proyecto desde su base de datos.
    5. El sistema ejecuta la opción Listar pendientes de producto para el proyecto cargado.
    6. El sistema ejecuta la opción Listar pendientes de sprints para el proyecto cargado.
  - **Cursos alternativos:**
    - 3a. El usuario cancela la carga de proyecto. Regresa al paso 1.
  
- **Caso de uso:** Cerrar proyecto.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.
  - **Propósito:** Cerrar un proyecto abierto en el espacio de trabajo.
  - **Visión general:** El usuario limpia el espacio de trabajo cerrando el proyecto.
  - **Referencias:** Requisito funcional 3.3: Cerrar proyecto.

- **Curso típico de eventos:**
  1. El usuario selecciona la opción Cerrar proyecto.
  2. El sistema deja de mostrar en pantalla el Registro de pendientes del producto y los Registros de pendientes del Sprint.
- **Cursos Alternativos:** Ninguno.
  
- **Caso de uso: Crear proyecto.**
  - **Actores:** Maestro de Scrum.
  - **Propósito:** Crear un nuevo proyecto en el sistema.
  - **Visión general:** El usuario crea un proyecto nuevo, y es asignado como Maestro de Scrum.
  - **Referencias:** Requisito funcional 3.1 Listar proyectos, Requisito funcional 3.4: Crear proyecto.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Crear proyecto.
    2. El sistema muestra un formulario para el ingreso de los datos del proyecto nuevo.
    3. El usuario introduce y envía los datos del proyecto nuevo al sistema.
    4. El sistema registra los datos del nuevo proyecto en su base de datos.
    5. El sistema ejecuta la opción Cargar proyecto de manera automática.
  - **Cursos alternativos:**
    - 4a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.
  
- **Caso de uso: Editar proyecto.**
  - **Actores:** Maestro de Scrum, Dueño de producto.
  - **Propósito:** Editar las propiedades de un proyecto.
  - **Visión general:** El usuario modifica las propiedades del producto, del cual son miembros de equipo.
  - **Referencias:** Requisito funcional 3.5: Editar proyectos.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Editar datos de proyecto.
    2. El sistema muestra un formulario con los datos del proyecto.
    3. El usuario introduce y envía los datos actualizados del proyecto al sistema.
    4. El sistema registra los datos actualizados del proyecto en su base de datos.

5. El sistema actualiza los datos del proyecto que se despliegan en pantalla.
- **Cursos alternativos:**
    - 3a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.
- **Caso de uso: Borrar proyecto.**
    - **Actores:** Maestro de Scrum, Dueño de producto.
    - **Propósito:** Borrar un proyecto registrado en el sistema.
    - **Visión general:** El usuario borra un proyecto del cual es miembro de equipo.
    - **Referencias:** Requisito funcional 3.1: Listar usuarios, Requisito funcional 3.6: Borrar usuario.
    - **Curso típico de eventos:**
      1. El usuario selecciona la opción Editar datos de proyecto.
      2. El usuario selecciona la opción Borrar proyecto.
      3. El usuario confirma la eliminación del proyecto.
      4. El sistema cierra el proyecto cargado.
      5. El sistema elimina los datos del proyecto de la base de datos.
    - **Cursos alternativos:**
      - 3a. El usuario cancela la eliminación del proyecto. Regresa al paso 2.

#### 4.2.1.4. Gestión de equipos de proyecto

Este caso de uso se divide en 4 funcionalidades, como se observa en la Figura 4.5, permitiendo gestionar los equipos de los proyectos mediante la agregación y remoción de usuarios a los equipos, además de ver el listado y poder modificar los roles de los miembros.

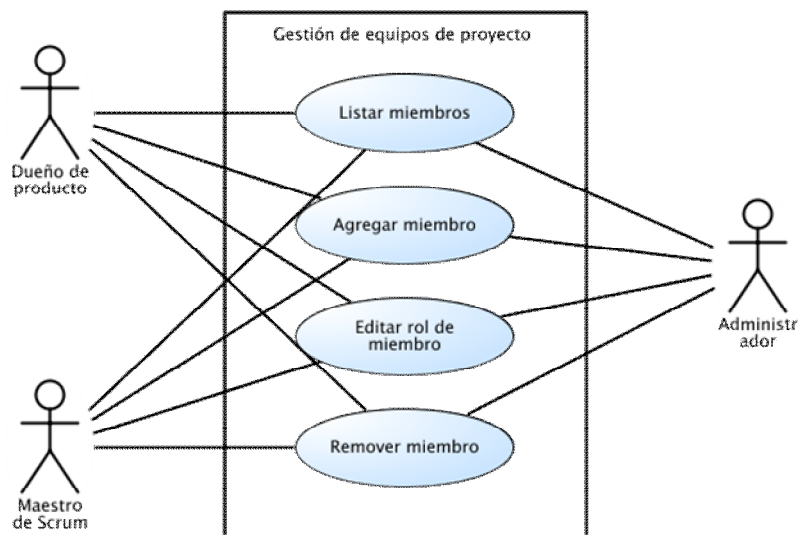


Figura 4.5. Casos de uso de Gestión de equipos de proyecto

- **Caso de uso:** Listar miembros.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Listar los miembros de equipo de un proyecto.
  - **Visión general:** El usuario ve la lista de los miembros de equipo del proyecto actualmente cargado.
  - **Referencias:** Requisito funcional 4.1: Listar miembros.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Listar miembros.
    2. El sistema obtiene los miembros del equipo de proyecto desde su base de datos.
    3. El sistema muestra en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.
  - **Cursos alternativos:** Ninguno.
  
- **Caso de uso:** Agregar miembro.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Agrega un miembro al equipo de proyecto.
  - **Visión general:** El usuario agrega un nuevo miembro de equipo del proyecto, de los usuarios actualmente registrados.
  - **Referencias:** Requisito funcional 4.1 Listar miembros, Requisito funcional 4.2: Agregar miembro.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Listar miembros.
    2. El usuario selecciona la opción Agregar miembro.
    3. El sistema muestra un formulario para el ingreso de los datos del miembro nuevo.
    4. El usuario introduce y envía los datos al sistema al sistema.
    5. El sistema registra los datos actualizados en su base de datos.
    6. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.
  - **Cursos alternativos:**
    - 4a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 4.
  
- **Caso de uso:** Editar rol de miembro.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto.

- **Propósito:** Editar el rol de un miembro de equipo.
- **Visión general:** El usuario modifica el rol asignado a un miembro de equipo del proyecto.
- **Referencias:** Requisito funcional 4.1: Listar miembros, Requisito funcional 4.3: Editar rol de miembro.
- **Curso típico de eventos:**
  1. El usuario selecciona la opción Listar miembros.
  2. El usuario selecciona al miembro al que se le cambiará el rol.
  3. El usuario selecciona la opción Editar rol de miembro.
  4. El sistema muestra un formulario con los datos del miembro del equipo.
  5. El usuario modifica y envía los datos al sistema.
  6. El sistema actualiza los datos del equipo en su base datos.
  7. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.
- **Cursos alternativos:**
  - 5a. El sistema detecta datos inválidos siendo ingresado y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 5.
- **Caso de uso:** Remover miembro.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Remover un miembro de equipo de un proyecto.
  - **Visión general:** El usuario remueve uno de los miembros del equipo del proyecto cargado actualmente.
  - **Referencias:** Requisito funcional 4.1: Listar miembros, Requisito funcional 4.4: Remover miembro.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Listar miembros.
    2. El usuario selecciona el miembro a ser removido.
    3. El usuario selecciona la opción Remover miembro.
    4. El usuario confirma la remoción del miembro del equipo de trabajo.
    5. El sistema actualiza los datos del equipo en su base de datos.
    6. El sistema actualiza en pantalla la lista de miembros del equipo de trabajo, con sus respectivos roles.
  - **Cursos alternativos:**
    - 4a. El usuario cancela la remoción del miembro. Regresa al paso 2.

#### 4.2.1.5. Gestión de registros de pendientes de producto

El caso de uso de registro de pendientes comprende las funcionalidades de listar, agregar, editar y borrar pendientes de un producto, también incluyen agregar los pendientes en *sprints* o removerlos de los *sprints*. En la Figura 4.6 se muestran los 6 casos de uso en los cuales se divide la gestión de pendientes de producto.

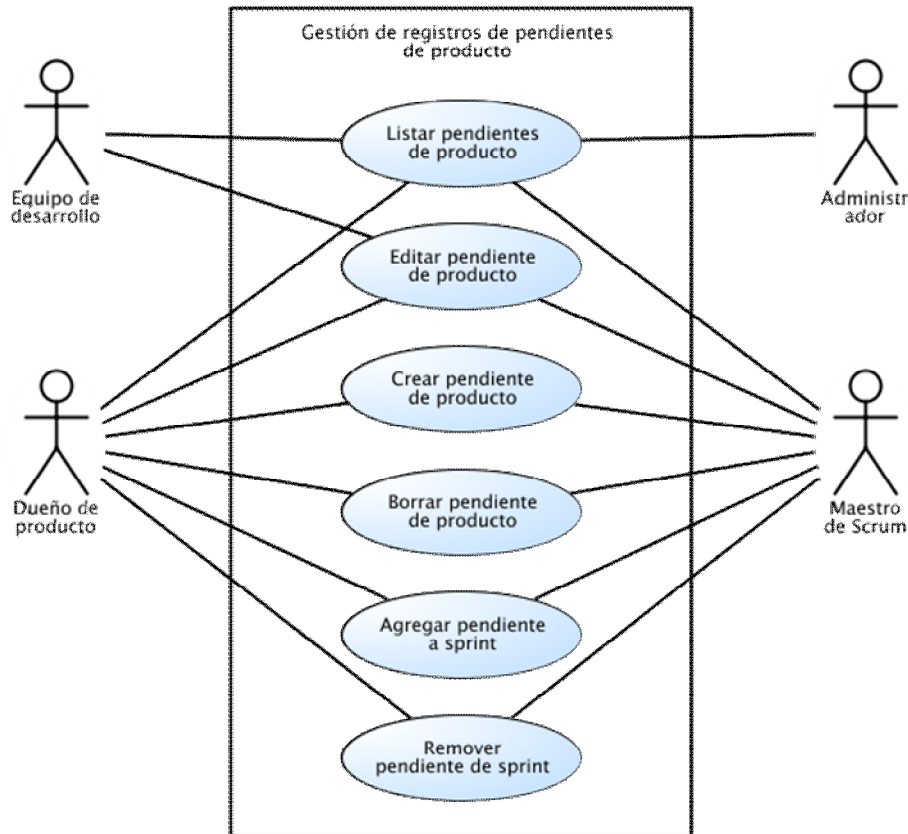


Figura 4.6. Casos de uso de Gestión de registros de pendientes de producto

- **Caso de uso:** Listar pendientes de producto.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.
  - **Propósito:** Listar los pendientes de producto de un proyecto.
  - **Visión general:** El usuario ve la lista de los pendientes de producto del proyecto actualmente cargado, incluidas las tareas asignadas a ellos.
  - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto.
  - **Curso típico de eventos:**
    1. El sistema obtiene el registro de pendientes del producto del proyecto desde su base de datos.
    2. El sistema muestra en pantalla el registro de pendientes del producto del proyecto cargado.



- **Cursos Alternativos:** Ninguno.
  
- **Caso de uso:** Crear pendiente de producto.
  - **Actores:** Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Crear un nuevo pendiente de producto.
  - **Visión general:** El usuario crea un pendiente de producto en el registro de pendientes de un proyecto actualmente cargado.
  - **Referencias:** Requisito funcional 5.1 Listar pendientes de producto, Requisito funcional 5.2: Crear pendiente de producto, Requisito funcional 7.1: Agregar tarea.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Crear pendiente de producto.
    2. El sistema muestra el formulario para introducir los datos del pendiente de producto nuevo.
    3. El usuario introduce y envía los datos al sistema.
    4. El sistema registra el nuevo pendiente de producto en su base de datos.
    5. El sistema agrega las tareas para las actividades de DMS correspondientes al nivel de capacidad del proyecto.
    6. El sistema actualiza en pantalla el registro de pendientes de producto.
  - **Cursos alternativos:**
    - 3a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.
  
- **Caso de uso:** Editar pendiente de producto.
  - **Actores:** Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.
  - **Propósito:** Editar las propiedades de un pendiente de producto.
  - **Visión general:** El usuario modifican las propiedades de un pendiente de producto, según su rol.
  - **Referencias:** Requisito funcional 5.3: Editar pendiente de producto.
  - **Curso típico de eventos:**
    1. El usuario selecciona el pendiente de producto a ser editado.
    2. El usuario selecciona la opción Editar pendiente de producto.
    3. El usuario modifica y envía los datos al sistema.
    4. El sistema actualiza los datos del pendiente de producto en su base de datos.
    5. El sistema actualiza en pantalla el registro de pendientes de producto.
  - **Cursos alternativos:**

3a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.

- **Caso de uso:** Borrar pendiente de producto.
  - **Actores:** Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Borrar un pendiente de producto.
  - **Visión general:** El usuario borra un pendiente de producto del registro de pendientes de producto del proyecto actualmente cargado.
  - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto, Requisito funcional 5.4: Borrar pendiente de producto.
  - **Curso típico de eventos:**
    1. El usuario selecciona el pendiente de producto a ser borrado.
    2. El usuario selecciona la opción Borrar pendiente de producto.
    3. El usuario confirma el borrado del pendiente de producto.
    4. El sistema borra el pendiente de producto de su base de datos.
    5. El sistema actualiza en pantalla el registro de pendientes de producto.
  - **Cursos alternativos:**
    - 3a. El usuario cancela la eliminación del pendiente de producto. Regresa al paso 2.
- **Caso de uso:** Agregar pendiente a *sprint*.
  - **Actores:** Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Agrega un pendiente de producto a un *sprint* del proyecto.
  - **Visión general:** El usuario agrega a un *sprint* del proyecto un pendiente de producto.
  - **Referencias:** Requisito funcional 5.1 Listar pendientes de producto, Requisito funcional 5.5: Agregar pendiente a *sprint*.
  - **Curso típico de eventos:**
    1. El usuario selecciona el pendiente de producto a ser agregado.
    2. El usuario selecciona la opción Agregar pendiente a *sprint*.
    3. El usuario selecciona el *sprint* donde será agregado el pendiente.
    4. El sistema relaciona el pendiente de producto al *sprint* en su base de datos.
    5. El sistema actualiza en pantalla el registro de pendientes de producto.
    6. El sistema actualiza en pantalla el registro de pendientes de *sprints*.
  - **Cursos alternativos:**
    - 3a. El usuario cancela la agregación del pendiente al *sprint*. Regresa al paso 1.

- **Caso de uso:** Remover pendiente de *sprint*.
  - **Actores:** Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Remover un pendiente de producto de un *sprint*.
  - **Visión general:** El usuario remueve un pendiente de producto del registro de pendientes de un *sprint*.
  - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto, Requisito funcional 5.6: Remover pendiente de *sprint*.
  - **Curso típico de eventos:**
    1. El usuario selecciona el pendiente de *sprint* a ser removido.
    2. El usuario selecciona la opción Remover pendiente de *sprint*.
    3. El sistema remueve la relación del pendiente con el *sprint* en su base de datos.
    4. El sistema actualiza en pantalla el registro de pendientes de producto.
    5. El sistema actualiza en pantalla el registro de pendientes de *sprints*.
  - **Cursos alternativos:**
    - 3a. El usuario cancela la remoción del pendiente del *sprint*. Regresa al paso 1.

#### 4.2.1.6. Gestión de sprints

Este caso de uso se va a dividir en cuatro casos de uso, como se muestra en la Figura 4.7, con las funcionalidades que permitirán listar, crear, editar y eliminar *sprints* en un proyecto.

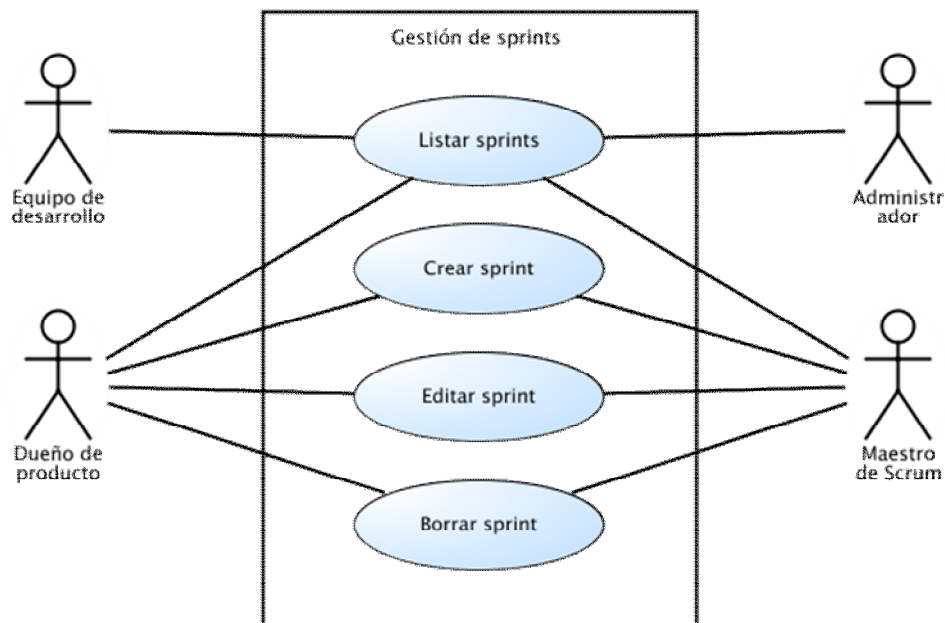


Figura 4.7. Casos de uso de Gestión de sprints

- **Caso de uso:** Listar *sprints*.
  - **Actores:** Administrador, Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.

- **Propósito:** Listar los *sprints* de un proyecto.
- **Visión general:** El usuario ve la lista de los *sprints* del proyecto actualmente cargado, incluidos los pendientes asignados a cada *sprint* y las tareas correspondientes.
- **Referencias:** Requisito funcional 6.1: Listar *sprints*.
- **Curso típico de eventos:**
  1. El sistema obtiene los *sprints* y sus registros de pendientes de cada *sprint* del proyecto cargado desde su base de datos.
  2. El sistema muestra en pantalla los *sprints* del proyecto cargado.
- **Cursos Alternativos:** Ninguno.
  
- **Caso de uso:** Crear *sprint*.
  - **Actores:** Maestro de Scrum, Dueño de proyecto.
  - **Propósito:** Crear un *sprint*.
  - **Visión general:** El usuario crea un *sprint* para un proyecto actualmente cargado.
  - **Referencias:** Requisito funcional 6.1 Listar *sprints*, Requisito funcional 6.2: Crear *sprint*.
  - **Curso típico de eventos:**
    1. El usuario selecciona la opción Crear *sprint*.
    2. El sistema muestra un formulario para el ingreso de los datos del *sprint* nuevo.
    3. El usuario introduce y envía los datos al sistema.
    4. El sistema registra los datos del nuevo *sprint* en su base de datos.
    5. El sistema actualiza en pantalla la lista de *sprints* del proyecto cargado.
  - **Cursos alternativos:**
    - 3a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.
  
- **Caso de uso:** Editar *sprint*.
  - **Actores:** Maestro de Scrum, Dueño de proyecto, Equipo de Desarrollo.
  - **Propósito:** Editar las propiedades de un *sprint*.
  - **Visión general:** El usuario modifican las propiedades de un *sprint*, según su rol.
  - **Referencias:** Requisito funcional 6.3: Editar *sprint*.
  - **Curso típico de eventos:**
    1. El usuario selecciona el *sprint* que desea editar.
    2. El usuario selecciona la opción Editar *sprint*.
    3. El sistema muestra un formulario con los datos del *sprint*.

4. El usuario actualiza y envía los datos del *sprint* al sistema.
  5. El sistema registra los datos actualizados del *sprint* en su base de datos.
  6. El sistema actualiza en pantalla la lista de *sprints* del proyecto cargado.
- **Cursos alternativos:**
    - 4a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 4.
  - **Caso de uso:** Borrar *sprint*.
    - **Actores:** Maestro de Scrum, Dueño de proyecto.
    - **Propósito:** Borrar un *sprint*.
    - **Visión general:** El usuario borra un *sprint* del proyecto actualmente cargado.
    - **Referencias:** Requisito funcional 6.1: Listar *sprints*, Requisito funcional 6.4: Borrar *sprint*.
    - **Curso típico de eventos:**
      1. El usuario selecciona el *sprint* que desea borrar.
      2. El usuario selecciona la opción Borrar *sprint*.
      3. El usuario confirma la eliminación del *sprint*.
      4. El sistema elimina los datos del *sprint* de la base de datos.
      5. El sistema actualiza en pantalla la lista de *sprints* del proyecto cargado.
    - **Cursos alternativos:**
      - 3a. El usuario cancela la eliminación del pendiente de producto. Regresa al paso 1.

#### 4.2.1.7. Gestión de tareas

Este caso de uso entrega las funcionalidades de crear y eliminar tareas, y editar los datos de las tareas de un pendiente de producto, ya sea que esté o no en un *sprint*, en la Figura 4.8 se muestran los tres casos de uso en los cuales se divide la gestión de tareas.

- **Caso de uso:** Crear tarea.
  - **Actores:** Maestro de Scrum, Equipo de desarrollo.
  - **Propósito:** Crear una tarea de un pendiente de producto.
  - **Visión general:** El usuario crea una tarea en un pendiente de producto, asignado o no a un *sprint* del proyecto cargado actualmente.
  - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto, Requisito funcional 6.1: Listar *sprints*, Requisito funcional 7.1: Crear tarea.
    1. El usuario selecciona la opción Agregar tarea.
    2. El sistema muestra el formulario para introducir los datos de la tarea.
    3. El usuario introduce y envía los datos al sistema.

4. El sistema registra la nueva tarea en su base de datos.
  5. El sistema actualiza en pantalla el pendiente de producto donde se agregó la tarea.
- **Cursos alternativos:**
    - 3a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 3.
- **Caso de uso:** Editar tarea.
    - **Actores:** Maestro de Scrum, Equipo de desarrollo.
    - **Propósito:** Editar las propiedades de una tarea.
    - **Visión general:** El usuario modifican las propiedades de una tarea.
    - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto, Requisito funcional 6.1: Listar *sprints*, Requisito funcional 7.2: Editar tarea.
    - **Curso típico de eventos:**
      1. El usuario selecciona la tarea a ser editada.
      2. El usuario selecciona la opción Editar tarea.
      3. El usuario modifica y envía los datos al sistema.
      4. El sistema actualiza los datos de la tarea en su base de datos.
      5. El sistema actualiza en pantalla el pendiente de producto de la tarea editada.
    - **Cursos alternativos:**
      - 3a. El sistema detecta datos inválidos siendo ingresados y bloquea el envío de los datos, hasta que el usuario introduce datos válidos. Regresa al paso 4.
  - **Caso de uso:** Borrar tarea.
    - **Actores:** Maestro de Scrum, Equipo de desarrollo.
    - **Propósito:** Borrar una tarea.
    - **Visión general:** El usuario borra una tarea de un pendiente de producto del proyecto actualmente cargado.
    - **Referencias:** Requisito funcional 5.1: Listar pendientes de producto, Requisito funcional 6.1: Listar *sprints*, Requisito funcional 7.3: Borrar tarea.
    - **Curso típico de eventos:**
      1. El usuario selecciona la tarea a ser borrada.
      2. El usuario selecciona la opción Borrar tarea.
      3. El usuario confirma el borrado de la tarea.
      4. El sistema borra los datos de la tarea de su base de datos.

5. El sistema actualiza en pantalla el pendiente de producto de la tarea borrada.
- **Cursos alternativos:**
    - 3a. El usuario cancela la eliminación del pendiente de producto. Regresa al paso 1.

### 4.3. Diseño de alto nivel: modelo entidad-relación para la gestión de proyectos

El diseño de la base de datos representa la información referente a proyectos, *sprints*, ítems del registro de pendientes, tareas, usuarios y las relaciones entre estos, como se muestra en la Figura 4.8.

El diseño se basa en las tablas *project*, *sprint*, *item*, *task* y *project\_has\_user*, las cuales serán explicadas a detalle en la sección 4.4.

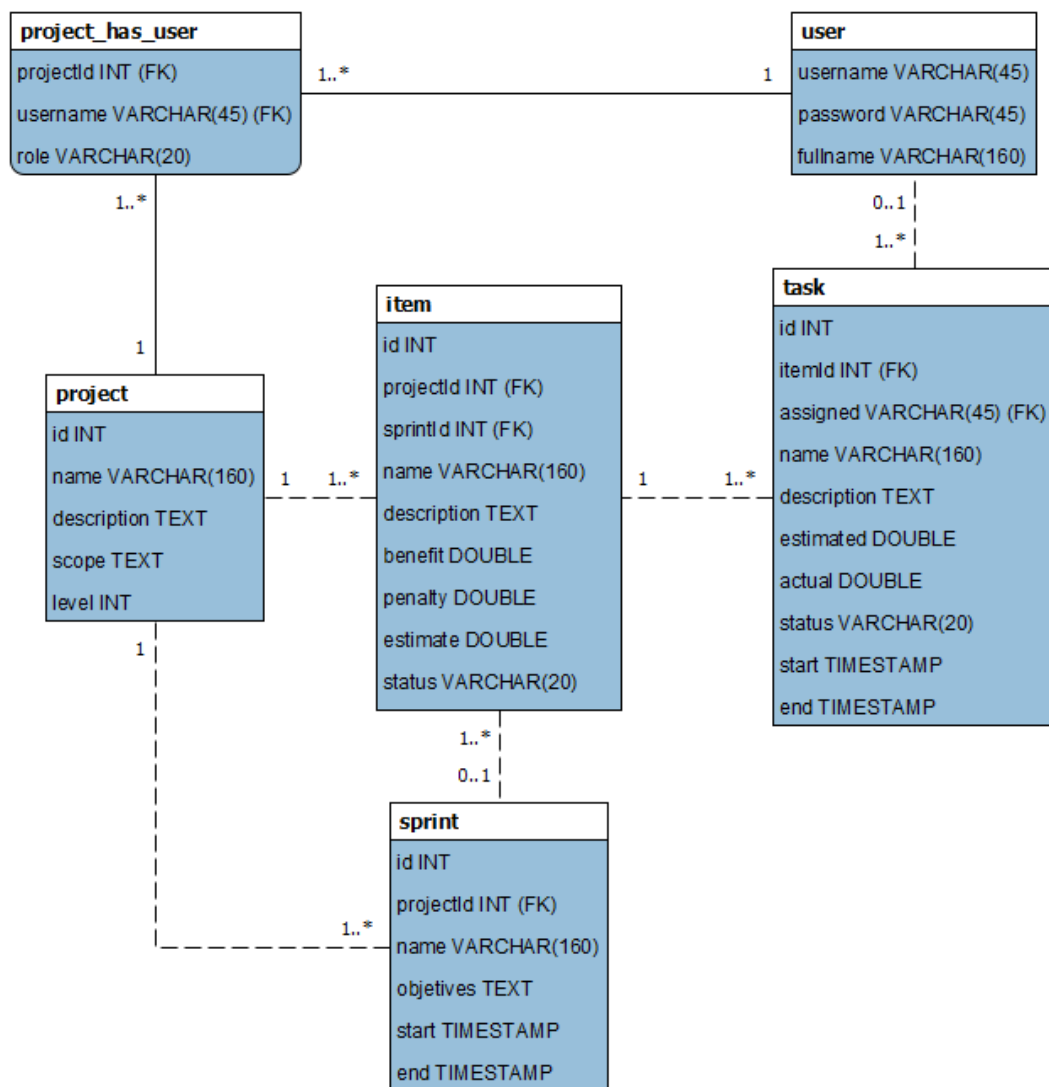


Figura 4.8. Modelo de la base de datos para la herramienta de soporte a la mezcla MoProSoft – Scrum

## 4.4. Diseño completo de la base de datos

En esta sección se explica cada una de las distintas tablas que componen el diseño de la base de datos, así como todas sus relaciones y atributos.

### 4.4.1. Diseño de tablas y atributos

A continuación se presenta, en orden alfabético, las diversas tablas que componen a la herramienta. Para cada una de ellas se hace una descripción de sus campos, así como las claves primarias y foráneas que la componen, además del código SQL para crearla en la base de datos.

La descripción de las tablas será acompañada de una tabla que describirá el nombre de los campos, si está permitidos los valores nulos, si es una clave y si lo es si es clave primaria o foránea, y la descripción del campo.

#### 4.4.1.1. Tabla *ítem*

Tabla en la que se insertan cada ítem del registro de pendientes del producto, correspondiente a un proyecto, sus estimaciones de valor de negocio y de esfuerzo, además de llevar el control del status del pendiente de producto (ver Tabla 66).

**Tabla 66.** Descripción de campos de la tabla ítem.

Campo	Tipo	Nulo	Clave	Descripción
benefit	Double	Si	No	Valor de beneficio si el ítem se realiza
description	Text	Si	No	Descripción del ítem
estimate	Double	Si	No	Esfuerzo estimado del ítem
id	Integer	No	Primaria	Identificador del ítem del registro de pendientes.
name	Varchar	No	No	Nombre del ítem
penalty	Double	Si	No	Valor de penalidad si el ítem no se realiza
projectId	Integer	No	Foránea	Identificador del proyecto al que pertenece.
sprintId	Integer	Si	Foránea	Identificador del sprint al que pertenece.
status	Varchar	No	No	Estado del ítem

#### 4.4.1.2. Tabla *project*

Tabla en la que se insertan los proyectos a ser gestionados por la herramienta, así como su alcance y nivel capacidad (ver Tabla 67).

**Tabla 67.** Descripción de campos de la tabla project.

Campo	Tipo	Nulo	Clave	Descripción
description	Text	Si	No	Descripción del proyecto
id	Integer	No	Primaria	Identificador del proyecto
level	Integer	No	No	Nivel de capacidad del proyecto
name	Varchar	No	No	Nombre del proyecto
scope	Text	Si	No	Explicación de los alcances del proyecto



#### 4.4.1.3. Tabla *project\_has\_user*

Tabla en la que se insertan las relaciones de proyectos con sus usuarios y los roles que estos tienen dentro del proyecto (ver Tabla 68).

**Tabla 68.** Descripción de campos de la tabla *project\_has\_user*.

Campo	Tipo	Nulo	Clave	Descripción
projectId	Integer	No	Foránea	Identificador del proyecto
role	Varchar	No	No	Rol del usuario en el proyecto
username	Varchar	No	Foránea	Nombre de usuario

#### 4.4.1.4. Tabla *sprint*

Tabla en la que se insertan los sprints del proyecto junto con sus objetivos y las fechas de inicio de sprint y fin de sprint (ver Tabla 69).

**Tabla 69.** Descripción de campos de la tabla *sprint*.

Campo	Tipo	Nulo	Clave	Descripción
end	Timestamp	No	No	Fecha de fin del sprint
id	Integer	No	Primaria	Identificador del sprint
name	Varchar	No	No	Nombre del sprint
objetives	Text	Si	No	Descripción de los objetivos del sprint
projectId	Integer	No	Foránea	Identificador del proyecto al que pertenece
start	Timestamp	No	No	Fecha de inicio del sprint

#### 4.4.1.5. Tabla *task*

Tabla en la que se insertan las tareas de los ítems del registro de pendientes de producto, el usuario al que es asignado, el esfuerzo estimado en horas, el esfuerzo actual en horas, el estado de la tarea, y la fecha de inicio y fin (ver Tabla 70).

**Tabla 70.** Descripción de campos de la tabla *task*.

Campo	Tipo	Nulo	Clave	Descripción
actual	Double	Si	No	Esfuerzo actual de la tarea
assigned	Varchar	Si	Foránea	Usuario asignado a la tarea
description	Text	Si	No	Descripción de la tarea
end	Timestamp	Si	No	Fecha de fin de la tarea
estimated	Double	Si	No	Esfuerzo estimado de la tarea
id	Integer	No	Primaria	Identificador de la tarea
itemId	Integer	No	Foránea	Identificador del ítem al que pertenece
name	Varchar	No	No	Nombre de la tarea
start	Timestamp	Si	No	Fecha de inicio de la tarea
status	Varchar	No	No	Estado de la tarea

#### 4.4.1.6. Tabla user

Tabla en la que se insertan los usuarios registrados en el sistema, guardando su nombre completo, que será usado para ser mostrado, además de su nombre de usuario y su contraseña, utilizados para acceder al sistema (ver Tabla 71).

**Tabla 71.** Descripción de campos de la tabla user.

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Clave</b>	<b>Descripción</b>
fullname	Varchar	No	No	Nombre de usuario para desplegar
password	Varchar	No	No	Contraseña de acceso al sistema
username	Varchar	No	Primaria	Nombre de usuario de acceso al sistema

## **5. Experimentación sobre la mezcla de MoProSoft y Scrum**

### **5.1. Contexto de la experimentación**

De acuerdo con Basili et al. [Basili86], la experimentación en la Ingeniería de Software debe promover el avance en el área a través de un proceso iterativo de aprendizaje. Con este objetivo en mente, desde hace más de dos décadas el proceso experimental ha sido aplicado en una variedad de entornos para estudiar una diversidad de áreas relacionadas con la tecnología de software.

De acuerdo con Juristo y Moreno [Juristo10], la experimentación en Ingeniería de Software se refiere a demostrar con hechos las suposiciones, hipótesis, especulaciones y creencias que abundan alrededor de la construcción y uso del software. Por lo tanto, es necesario analizar, estudiar y utilizar los métodos de investigación cuando se trata de validar científicamente alguna propuesta en el campo de la Ingeniería de Software. Por lo tanto, la experimentación en la Ingeniería de Software traslada a la Ingeniería de Software el paradigma experimental. En este sentido, Wohlin et al. [Wohlin03] indican que las estrategias empíricas para el desarrollo de experimentos en Ingeniería de Software incluyen a las encuestas, casos de estudio y experimentos.

Particularmente un caso de estudio se utiliza para investigar una única entidad o fenómeno dentro de un espacio de tiempo específico. El investigador recopila información detallada sobre, por ejemplo, un único proyecto durante un período de tiempo. Sin embargo, si se desea comparar dos métodos, puede ser necesario organizar el estudio como un caso de estudio o un experimento. De acuerdo con Kitchenham et al. [Kitchenham95], un ejemplo puede ser utilizar un proyecto piloto para evaluar los efectos de un cambio con respecto a alguna línea base. Una ventaja de los casos de estudio es que son más fáciles de planificar, pero la principal desventaja es que los resultados son difíciles de generalizar y más difíciles de interpretar, es decir, es posible mostrar los efectos en una situación típica, pero no puede generalizarse a cualquier situación [Yin09].

Es evidente así, que se ha escogido un caso de estudio como la estrategia empírica experimental a seguir en la validación de la hipótesis establecida en el Capítulo 1 de esta tesis. Por lo tanto, de acuerdo con las recomendaciones de Wohlin et al. [Wohlin12], existen tres diferentes estrategias para desarrollar un caso de estudio:

1. Comparar los resultados obtenidos de una nueva propuesta y una línea base.
2. Desarrollar dos proyectos en paralelo (“proyectos hermanos”), eligiendo uno de ellos como base.
3. Aplicar la nueva propuesta sobre algunos componentes seleccionados y comparar los resultados que se obtengan con los de los componentes en los cuales no se aplicó.

Para este capítulo de experimentación se presenta la aplicación de la segunda estrategia, para la cual se considerará un proyecto que debe desarrollarse de forma paralela con un grupo de control (que trabaja con la forma tradicional) y un grupo experimental (que trabaja con el patrón propuesto en esta tesis). En este sentido, los jefes de proyecto de la empresa recibirán la capacitación necesaria para utilizar el patrón MoProSoft – Scrum y su herramienta asociada. Una vez que el patrón sea implementado en el proyecto, los resultados serán comparados con los obtenidos en el proyecto que no utilizó el patrón.

## 5.2. Definición del caso de estudio

El caso de estudio se enfoca en la implementación de la mezcla de MoProSoft y Scrum en una pequeña empresa mexicana desarrolladora de software llamada SOCIEPRO S.A.P.I. de C.V. Esta empresa fue constituida en 2009, en la Ciudad de Huajuapán de León, Oaxaca. La Tabla 72 resume el número de empleados y la actividad principal de esta empresa.

**Tabla 72.** Número de empleados y actividad principal de la empresa incluida en el caso de estudio.

Empresa	Número de empleados	Actividad principal
SOCIEPRO S.A.P.I. de C.V.	20	SaS (Software as Service)

La empresa se compone de dos departamentos:

- Departamento administrativo. Se encarga de la dirección de la empresa a nivel técnico y gerencial. Este departamento es controlado por un director relacionado con el área de TI.
- Área de desarrollo. Se encarga de desarrollar los proyectos software solicitados por los clientes. Dentro de este personal se encuentran cuatro jefes de proyectos que gestionarán el uso del patrón MoProSoft – Scrum, y 12 miembros de la plantilla de software.

La Tabla 73 resume la infraestructura con la que cuenta la empresa.

**Tabla 73.** Infraestructura de la empresa incluida en el caso de estudio.

Infraestructura	Cantidad
Servidor de gama media dedicado a datos y aplicaciones	1
Computadoras personales	3
Computadoras portátiles	18
Impresoras de inyección de tinta	1
Página web de la empresa	1
Escritorios de trabajo	18
Proyectores de video	1
Licencias de software (lenguajes de programación)	3
Licencias de software (ofimática)	5
Licencia de software (gestión)	1
Licencia de software (soporte)	1
Software desarrollado por la empresa (para gestión de documentos y personal)	2

En relación a su experiencia previa, la empresa mostró evidencia de lo siguiente:

- SOCIEPRO S.A.P.I. de C.V es una empresa del grupo VEUREKA que se constituyó con la misión de mejorar la calidad de vida de la sociedad innovando su modo de proceder con las prácticas de gestión de Proyectos, Procesos y Productividad Ejecutiva. En este sentido, la empresa se ha enfocado por más de 5 años al desarrollo de software a la medida y SaS con el objetivo de proporcionar soluciones tecnológicas al mercado local y nacional. Regularmente los problemas en la gestión les impiden tomar mayor cantidad de proyectos.
- SOCIEPRO S.A.P.I. de C.V implementó Scrum por tres años como forma ágil de trabajo a través de la certificación oficial de ScrumAlliance.org y Scrum.org de tres de sus empleados como Scrum Master (uno de ellos es el tesista).
- En relación a sus problemas de gestión (relacionadas con la Planificación del Proyecto, y Monitorización y Control del Proyecto), SOCIEPRO S.A.P.I. de C.V pretende implantar las actividades del PSP (*Personal Software Process*) para la gestión a través de la reciente certificación oficial del SEI de seis de sus empleados (uno de ellos es el tesista).
- Por último, SOCIEPRO S.A.P.I. de C.V tiene algún conocimiento sobre los modelos de referencia, particularmente MoProSoft, pero nunca lo ha utilizado.

Considerando estas condiciones, se decidió establecer las características de un software de gestión de actividades (llamado POC People Light) a desarrollar a través de proyectos hermanos, éstas son resumidas en la Tabla 74.

**Tabla 74.** Resumen de funcionalidades de la herramienta desarrollada con los proyectos hermanos.

<b>Funcionalidades del gestor de actividades</b>		
<b>Sesión</b>	<ul style="list-style-type: none"> <li>• Login de administrador</li> <li>• Logout de administrador</li> </ul>	<ul style="list-style-type: none"> <li>• Login de usuario</li> <li>• Logout de usuario</li> </ul>
<b>Gestión de usuario</b>	<ul style="list-style-type: none"> <li>• Alta de usuario</li> <li>• Baja de usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Edición de usuario</li> <li>• Ver usuarios</li> </ul>
<b>Gestión de grupo</b>	<ul style="list-style-type: none"> <li>• Alta de grupo</li> <li>• Baja de grupo</li> </ul>	<ul style="list-style-type: none"> <li>• Edición de grupo</li> <li>• Ver grupos</li> </ul>
<b>Gestión de actividades</b>	<ul style="list-style-type: none"> <li>• Preparar recordatorio</li> <li>• Aceptar recordatorio</li> <li>• Preparar orden</li> <li>• Enviar orden</li> </ul>	<ul style="list-style-type: none"> <li>• Acusar orden</li> <li>• Preparar solicitud</li> <li>• Enviar solicitud</li> <li>• Revisar solicitud</li> </ul>
<b>Seguimiento de actividades</b>	<ul style="list-style-type: none"> <li>• Ver productos que debe un usuario asimismo</li> <li>• Ver productos que debe un usuario a otros</li> <li>• Ver productos que le deben al usuario</li> <li>• Ver productos que otros usuarios deben a otros</li> </ul>	<ul style="list-style-type: none"> <li>• Suspender solicitud</li> <li>• Reactivar solicitud</li> <li>• Cancelar solicitud</li> <li>• Ver solicitud</li> </ul>

Una vez definidas las funcionalidades del software a construir, se dio una breve charla informativa y se iniciaron los proyectos hermanos con una reunión de lanzamiento. Para esto, los jefes de proyecto de la empresa definieron los objetivos de negocio resumidos en la Tabla 75. Estos valores objetivo representan la media de los datos históricos de SOCIEPRO S.A.P.I de C.V y serán utilizados más adelante para demostrar la efectividad de la mezcla propuesta. Así, las siguientes secciones muestran la implantación del patrón MoProSoft - Scrum y la comparativa de resultados entre los proyectos hermanos.

**Tabla 75.** Valores objetivos de los proyectos hermanos.

Objetivo	Valor objetivo
Desviación de calendario	<12%
Desviación de esfuerzo	<10%
Desviación de tamaño	<13%
Desviación de número de defectos encontrados por fase	<10%

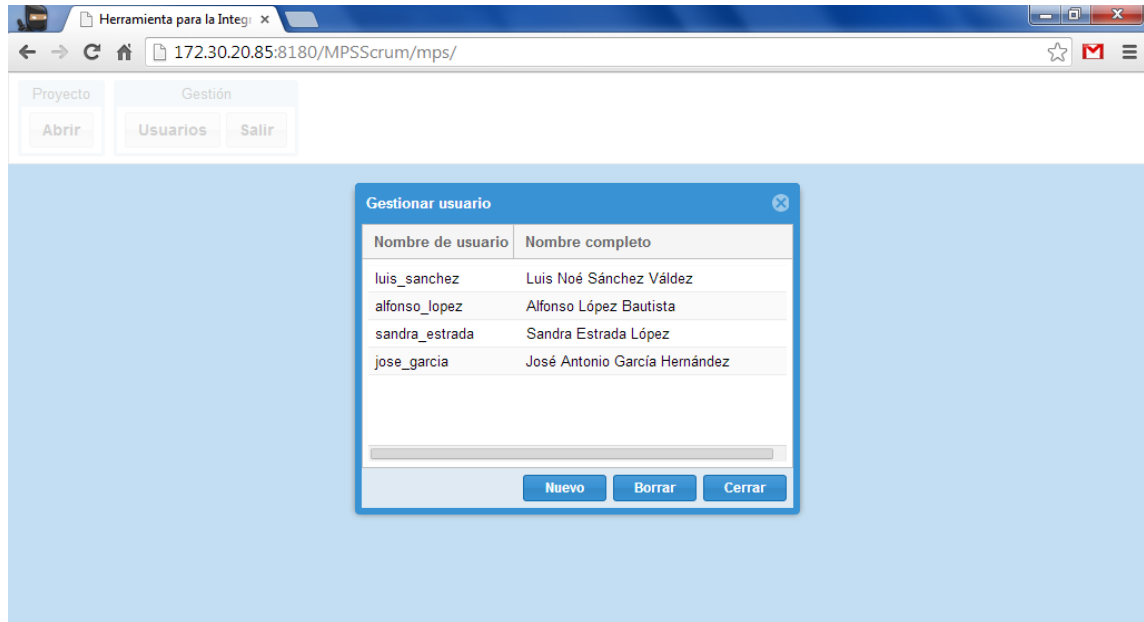
### 5.3. Implantación de la mezcla de MoProSoft y Scrum en el proyecto definido

Dado que se pretende demostrar la efectividad del patrón creado, a continuación se muestra la implementación de las actividades definidas en las Tablas 30 y 35 de esta tesis a través de la herramienta creada para introducir la gestión de los proyectos en SOCIEPRO S.A.P.I de C.V. Cabe mencionar que la información mostrada únicamente corresponde a la forma en que el patrón y su herramienta asociada son utilizados en un ciclo del proyecto POC Beople Light; en este sentido, al final únicamente se compararán los resultados obtenidos con el enfoque ágil carente de gestión.

#### 5.3.1. Planificación

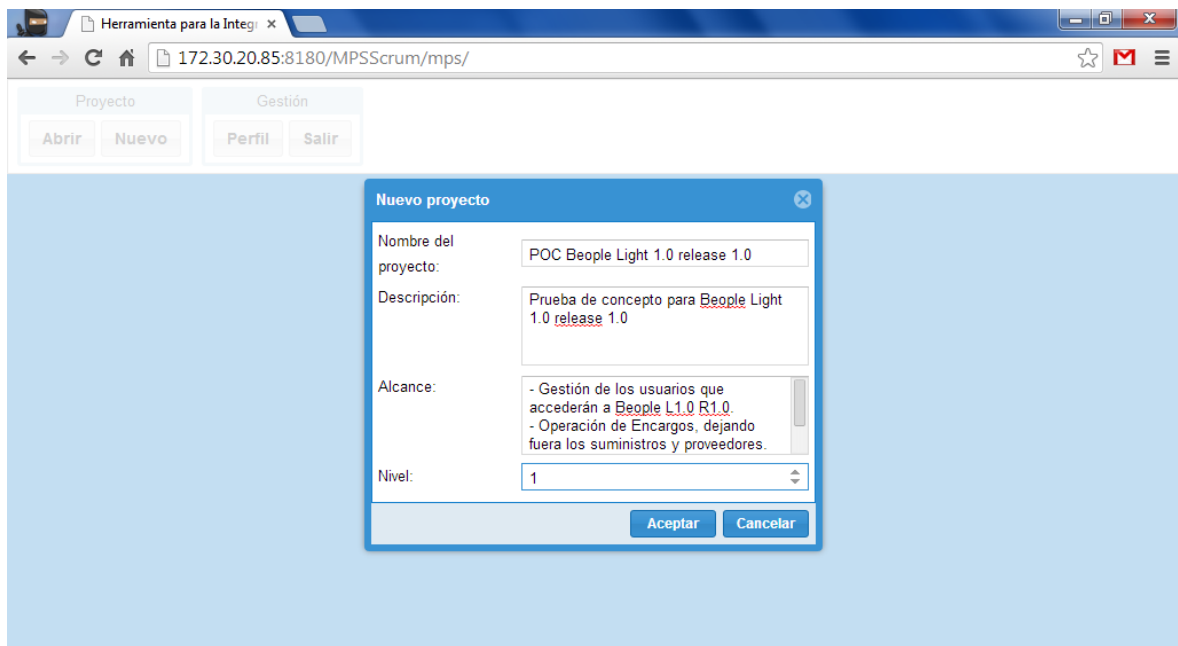
La Figura 5.1 muestra la forma en que la actividad APE A.1.1 del patrón es realizada. Cabe mencionar que los equipos formados para desarrollar los proyectos hermanos cuentan con el mismo nivel de experiencia y conocimientos. En este sentido, el grupo experimental está formado por los siguientes roles:

- CL: Anónimo.
- RGPY: Carlos Benavides Martínez, Iván A. García Pacheco.
- RAPE: Sandra Estrada López.
- RDM: Sandra Estrada López.
- MS: José Antonio García Hernández.
- ET: Luis Noé Sánchez Valdez, Alfonso López Bautista.
- Medidor externo: Iván A. García Pacheco.

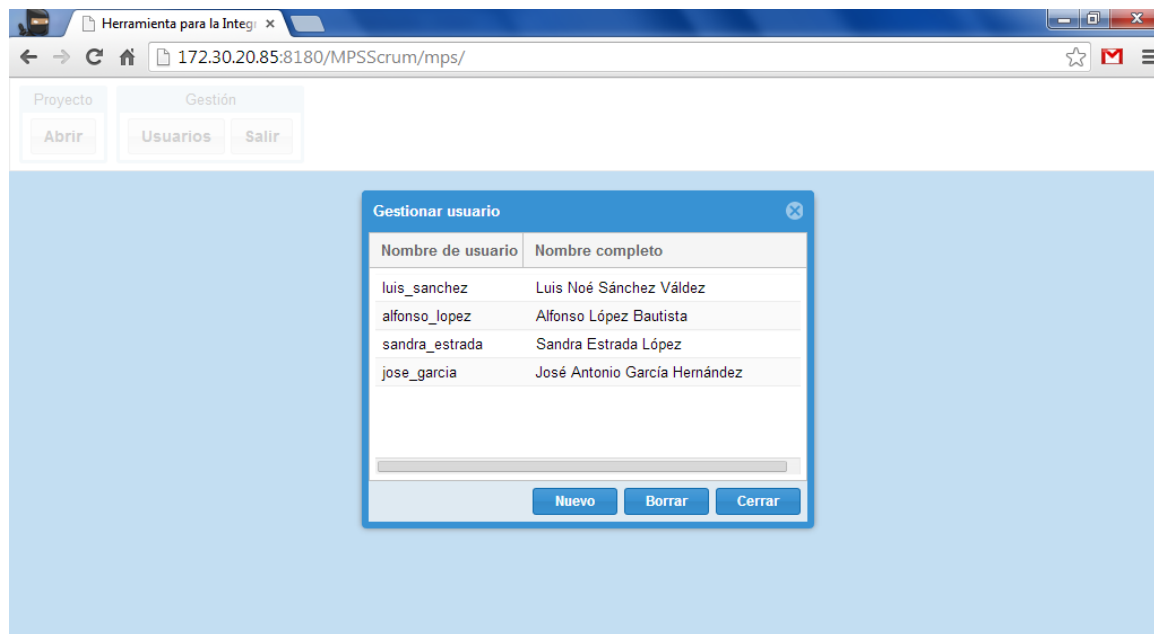


**Figura 5.1.** Configuración de participantes del grupo experimental

Una vez que la actividad es terminada, el MS es el responsable de crear el proyecto objeto de esta experimentación (ver Figura 5.2) y dar de alta al grupo experimental (ver Figura 5.3).

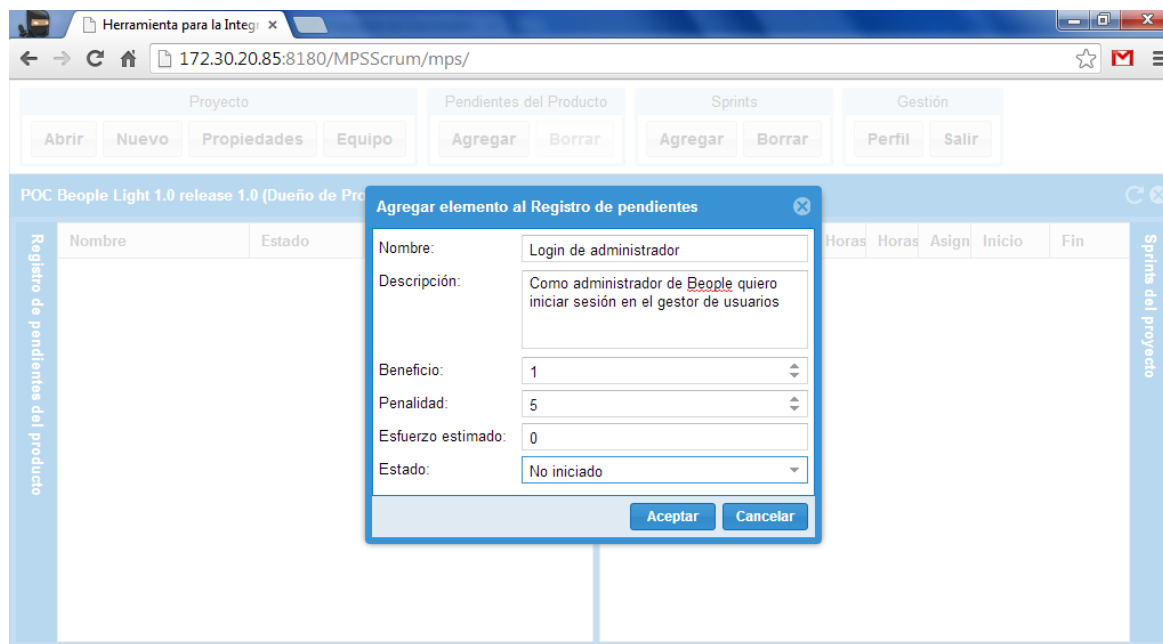


**Figura 5.2.** Creación del proyecto por el Maestro de Scrum



**Figura 5.3.** Formación del grupo experimental

Posteriormente, el RGPY y el RAPE revisan la descripción del proyecto para que éste último genere (ver Figura 5.4) y priorice el Registro de Pendientes del Producto de acuerdo a la actividad APE A1.1 del patrón (ver Figura 5.5). La Figura 5.5 muestra que el RGPY ha definido una escala de estimación de beneficio y penalidad que corresponde a los siguientes valores: Muy poco – 1, Poco – 2, Medio – 3, Alto – 5, Muy alto – 8.



**Figura 5.4.** Agregación de pendientes



Nombre	Estado	Beneficio	Penalida	Peso de	Valor (%)	Costo (%)	Prioridad
Enviar solicitud	No iniciado	8	8	16	7.73	0.00	0.00
Ver encargos	No iniciado	8	8	16	7.73	0.00	0.00
Ver productos que deb	No iniciado	8	8	16	7.73	0.00	0.00
Ver productos que me	No iniciado	8	8	16	7.73	0.00	0.00
Ver productos que otro	No iniciado	8	8	16	7.73	0.00	0.00
Ver productos que me	No iniciado	8	8	16	7.73	0.00	0.00
Revisar solicitud	No iniciado	8	8	16	7.73	0.00	0.00
Cancelar encargo	No iniciado	5	5	10	4.83	0.00	0.00
Ver usuario	No iniciado	3	5	8	3.86	0.00	0.00
Aceptar recordatorio	No iniciado	5	2	7	3.38	0.00	0.00
Acusar orden	No iniciado	5	2	7	3.38	0.00	0.00
Alta de usuario	No iniciado	2	5	7	3.38	0.00	0.00
Enviar orden	No iniciado	5	2	7	3.38	0.00	0.00

Figura 5.5. Creación del registro de pendientes

Con esta información, el RAPE define conjuntamente con el CL el Protocolo de Entrega de acuerdo a la actividad APE A1.3 del patrón. En este punto, el RAPE ya ha conformado al ET y debe definir el Plan de Adquisiciones y Capacitación como lo indican las actividades APE 1.7 y 1.8 del patrón. Durante la Reunión de Planificación, el RAPE explica los elementos del Registro de Pendientes del Producto a cada integrante del ET para cumplir con la primera parte de la actividad APE A1.13 del patrón (ver Figura 5.6).

Nombre	Estado	Valor (%)	Costo (%)	Prioridad
Ver usuario	No iniciado	3.86	0.00	0.00
Logout de administrac	No iniciado	2.42	0.00	0.00
Login de usuario	No iniciado	2.90	0.00	0.00
Logout de usuario	No iniciado	2.90	0.00	0.00
Alta de usuario	No iniciado	3.38	0.00	0.00
Baja de usuario	No iniciado	2.90	0.00	0.00
Acusar orden	No iniciado	3.38	0.00	0.00
Preparar solicitud	No iniciado	1.93	0.00	0.00
Enviar solicitud	No iniciado	7.73	0.00	0.00
Revisar solicitud	No iniciado	7.73	0.00	0.00
Preparar recordatorio	No iniciado	1.93	0.00	0.00
Aceptar recordatorio	No iniciado	3.38	0.00	0.00
Preparar orden	No iniciado	1.93	0.00	0.00

Figura 5.6. Opciones del equipo de trabajo

Posteriormente, el ET debe identificar las actividades que deben realizarse para que cada pendiente pueda ser considerado como terminado (cumpliendo así con la primera parte de las actividades APE A1.4 y APE A1.5 del patrón) (ver Figura 5.7).

Nombre	Valor (%)	Horas estimadas
Login de administrador	2.90	0.00
A2.2. Documentar o modificar la Especificación de Requerimientos		0.00
A2.10. Documentar o modificar Manual de Usuario		0.00
A2.13. Incorporar Especificación de Requerimientos y Manual de Usuario a Configuración de Software		0.00
A3.2. Documentar o modificar Análisis y Diseño		0.00
A3.10. Incorporar Análisis y Diseño a Configuración de Software		0.00
A4.2. Construir o modificar Componente(s) de software		0.00
A4.5. Incorporar Componentes a Configuración de Software		0.00
A5.2 Realizar integración		0.00
A5.3. Documentar o modificar Manual de Operación		0.00
A5.8. Documentar o modificar Manual de Usuario		0.00
A5.11. Incorporar Software, Manual de Operación y Manual de Usuario a Configuración de Software		0.00
Alta de usuario	3.38	0.00

**Figura 5.7.** Análisis de tareas y adición de otras necesarias

Cabe mencionar que las tareas mostradas en la Figura 5.7 corresponden con las actividades marcadas por DMS en el Nivel de Madurez 1 (o realizado). La Figura 5.8 muestra que el ET debe establecer el tiempo de cada actividad de acuerdo a lo marcado por la actividad APE A1.6 del patrón para generar una estimación completa (ver Figura 5.9), y estimar el esfuerzo de cada pendiente agregado al registro (ver Figura 5.10). En este sentido, se definió la siguiente escala de estimación de esfuerzo: Muy poco – 1, Poco – 2, Medio – 3, Alto – 5, Muy Alto – 8.

**Editar tarea**

Nombre: A2.2. Documentar o modificar la Especificación de Requerimientos

Descripción: A2.2. Documentar o modificar la Especificación de Requerimientos. - Identificar y consultar fuentes de información (clientes, usuarios, ...)

Horas estimadas: 3

Horas actuales: 0

Estado: No iniciado

Asignado a: [Seleccionar]

Inicio: [Calendario]

Fin: [Calendario]

Aceptar Cancelar

**Figura 5.8.** Estimación de tiempo por parte del equipo de trabajo

Nombre	Valor (%)	Costo (%)	Prioridad	Horas estimadas
Login de administrador	2.90	0.00	0.00	16.00
Alta de usuario	3.38	0.00	0.00	12.50
Baja de usuario	2.90	0.00	0.00	13.00
Ver usuario	3.86	0.00	0.00	13.00
Logout de administrador	2.42	0.00	0.00	13.00
Login de usuario	2.90	0.00	0.00	13.00
Logout de usuario	2.90	0.00	0.00	13.00
Preparar recordatorio	1.93	0.00	0.00	13.00
Aceptar recordatorio	3.38	0.00	0.00	13.00
Preparar orden	1.93	0.00	0.00	13.00
Enviar orden	3.38	0.00	0.00	13.00
Acusar orden	3.38	0.00	0.00	11.00
Preparar solicitud	1.93	0.00	0.00	13.00

Figura 5.9. Estimación de tiempo terminada

Nombre	Valor (%)	Esfuerzo	Costo (%)	Prioridad	Horas estimadas
Login de administrador	2.90	2	1.33	2.17	11.00
Alta de usuario	3.38	5	3.33	1.01	12.50
Baja de usuario	2.90	8	5.33	0.54	13.00
Ver usuario	3.86	8	5.33	0.72	13.00
Logout de administrador	2.42	8	5.33	0.45	13.00
Login de usuario	2.90	8	5.33	0.54	13.00
Logout de usuario	2.90	8	5.33	0.54	13.00
Preparar recordatorio	1.93	8	5.33	0.36	13.00
Aceptar recordatorio	3.38	8	5.33	0.63	13.00
Preparar orden	1.93	8	5.33	0.36	13.00
Enviar orden	3.38	8	5.33	0.63	13.00
Acusar orden	3.38	2	1.33	2.54	11.00
Preparar solicitud	1.93	8	5.33	0.36	13.00

Figura 5.10. Estimación de esfuerzo

Con las estimaciones terminadas, los sprints son creados por el RAPE (ver Figura 5.11) y a través de la prioridad (Valor/Costo) las actividades son asignadas a éstos (ver Figura 5.12), mientras estos quepan en horas, para obtener así el compromiso del ET y cumplir con lo marcado en la actividad APE A1.4 del patrón (ver Figura 5.13).

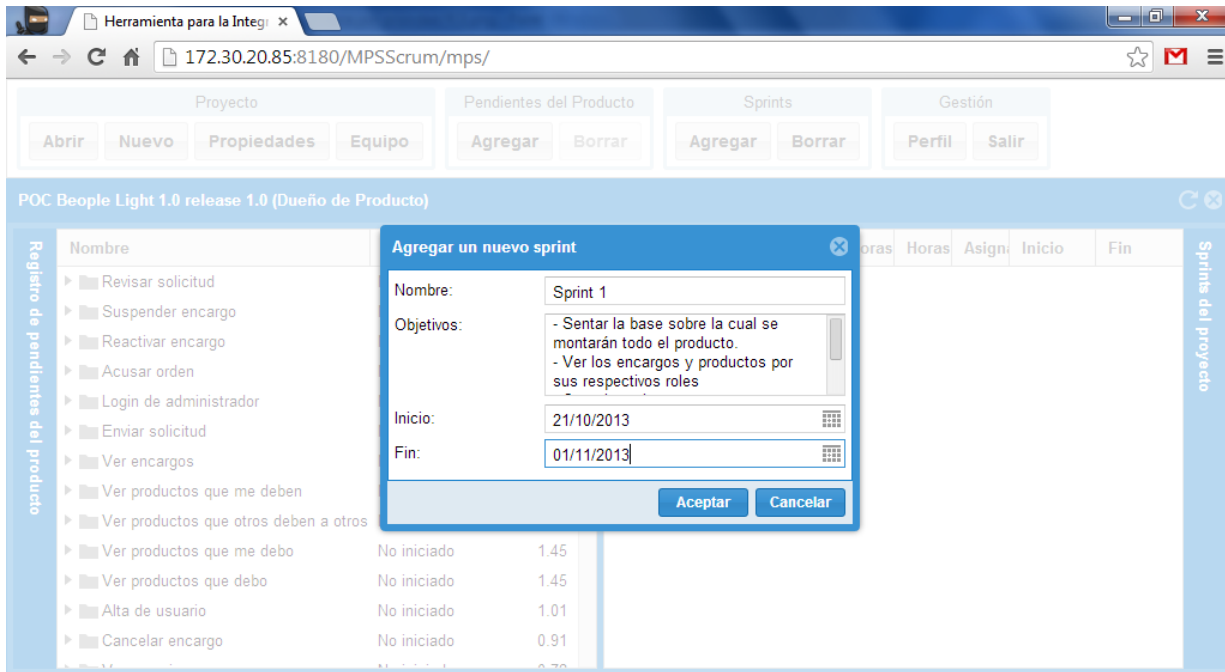


Figura 5.11. Creación del primer sprint para el proyecto

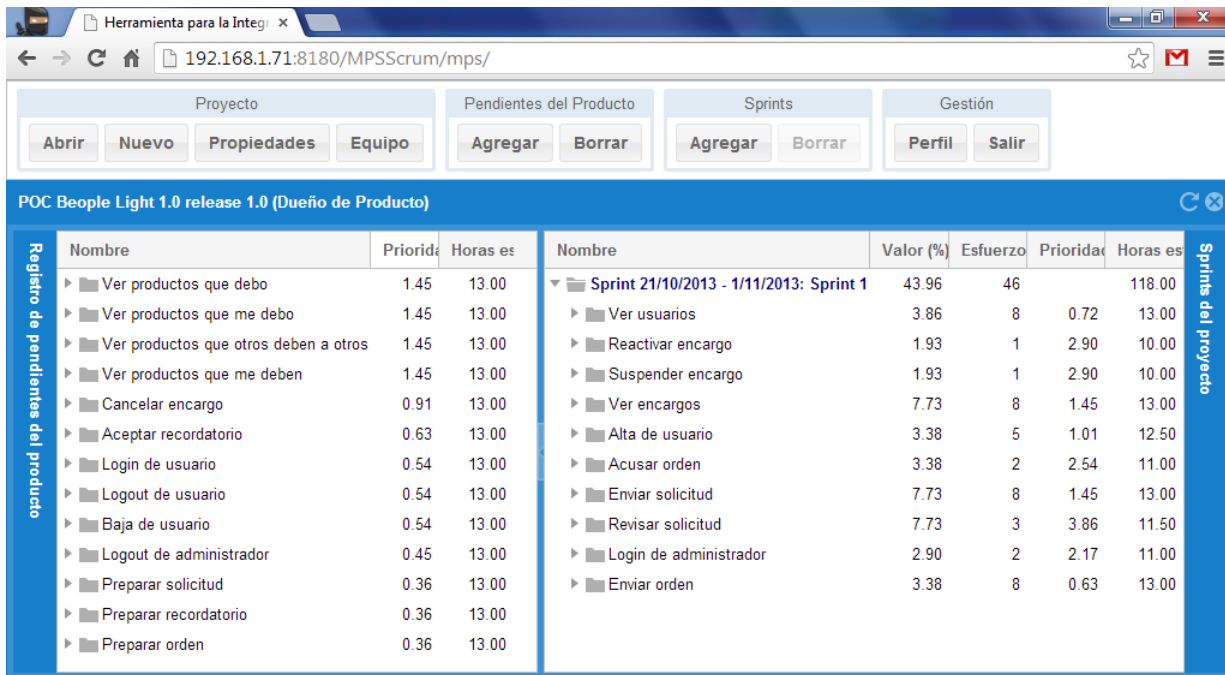


Figura 5.12. Asignación de pendientes a sprint

Nombre	Valor (%)	Esfuerzo	Costo (%)	Prioridad ↓	Horas estimadas
<b>Sprint 21/10/2013 - 1/11/2013: Sprint 1</b>	43.96	46	30.67		118.00
Revisar solicitud	7.73	3	2.00	3.86	11.50
Reactivar encargo	1.93	1	0.67	2.90	10.00
Suspender encargo	1.93	1	0.67	2.90	10.00
Acusar orden	3.38	2	1.33	2.54	11.00
Login de administrador	2.90	2	1.33	2.17	11.00
Ver encargos	7.73	8	5.33	1.45	13.00
Enviar solicitud	7.73	8	5.33	1.45	13.00
Alta de usuario	3.38	5	3.33	1.01	12.50
Ver usuarios	3.86	8	5.33	0.72	13.00
Enviar orden	3.38	8	5.33	0.63	13.00
<b>Sprint 4/11/2013 - 15/11/2013: Sprint 2</b>	47.83	72	48.00		117.00
<b>Sprint 18/11/2013 - 29/11/2013: Sprint 3</b>	8.21	32	21.33		52.00

Figura 5.13. Estimación de sprints

Continuando con la planificación, con los *sprints* creados y los pendientes asignados, el ET realiza las estimaciones de los tiempos necesarios para realizar las actividades de acuerdo a la actividad APE A1.9 del patrón (ver Figura 5.14).

**Editar tarea**

Nombre: A2.2. Documentar o modificar la Especificación de Requerimientos

Descripción: A2.2. Documentar o modificar la Especificación de Requerimientos. - Identificar y consultar fuentes de información (clientes, usuarios, ...)

Horas estimadas: 1

Horas actuales: 0

Estado: No iniciado

Asignado a: Luis Noé Sánchez Váldez

Inicio: 21/10/2013

Fin: 21/10/2013

Aceptar Cancelar

Figura 5.14. Estimación de fechas de inicio y fin del proyecto

Considerando el tiempo estimado de los tres *sprints* creados en la Figura 5.13, el costo estimado del proyecto es calculado de acuerdo a la actividad APE A1.10 del patrón. Por último, se genera el Plan de Manejo de Riesgos de acuerdo a la actividad APE A1.11 del patrón y se junta la información en el Plan de Desarrollo (segunda parte de APE A1.13). De acuerdo con la guía de

ajuste de APE sobre Plan de Desarrollo se asume que el Plan de Proyecto y el Plan de Desarrollo son iguales (APE A1.12).

### 5.3.2. Realización

Una vez iniciado el *sprint*, además de realizar las actividades de DMS Nivel 1, se efectúa diariamente la reunión diaria de acuerdo a la actividad APE A2.1 del patrón (ver Figura 5.15).

Nombre	Estado	Horas estima	Horas actual	Asignado a	Inicio	Fin
▼ Sprint 21/10/2013 - 1/11/2013: S						
Reactivar encargo	No iniciado	10.00	0.00			
Ver usuarios	No iniciado	13.00	6.50			
A2.2. Documentar o modific	Hecho	1.00	0.75	Luis Noé S...	21/10/2013	21/10/2013
A2.10. Documentar o modifi	Hecho	1.00	0.75	Luis Noé S...	21/10/2013	21/10/2013
A2.13. Incorporar Especifica	Hecho	0.50	0.25	Luis Noé S...	21/10/2013	21/10/2013
A5.11. Incorporar Software,	No iniciado	0.50	0.00	Luis Noé S...	23/10/2013	23/10/2013
A5.8. Documentar o modific	No iniciado	1.00	0.00	Luis Noé S...	22/10/2013	23/10/2013
A5.3. Documentar o modific	No iniciado	1.00	0.00	Luis Noé S...	22/10/2013	22/10/2013
A5.2. Realizar integración	No iniciado	1.00	0.00	Luis Noé S...	22/10/2013	22/10/2013
A4.5. Incorporar Component	No iniciado	0.50	0.00	Luis Noé S...	22/10/2013	22/10/2013
A4.2. Construir o modificar	En progreso	4.00	3.50	Luis Noé S...	21/10/2013	22/10/2013
A3.10. Incorporar Análisis y	Hecho	0.50	0.25	Luis Noé S...	21/10/2013	21/10/2013

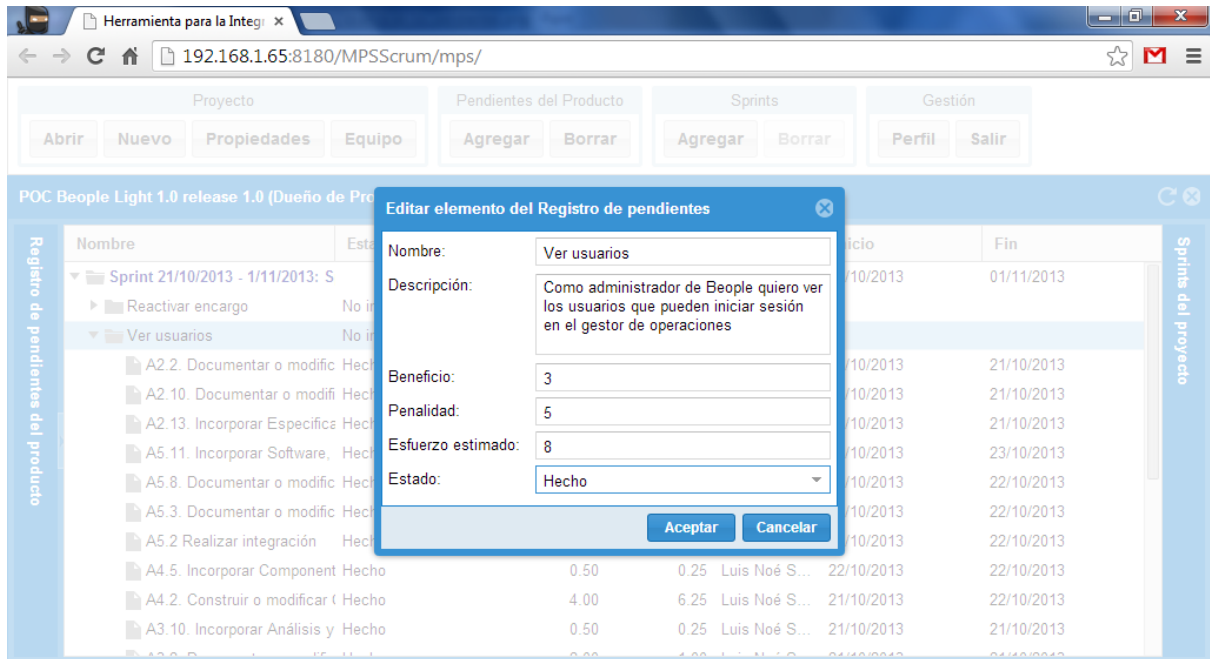
Figura 5.15. Reuniones diarias sobre el trabajo realizado

Esto se debe efectuar mientras dure el *sprint*, de tal forma que al final se tengan el reporte de tiempo de todas las actividades (ver Figura 5.16).

Nombre	Estado	Horas estima	Horas actua	Asignado a	Inicio	Fin
▼ Sprint 21/10/2013 - 1/11/2013: Sprint 1						
Reactivar encargo	No iniciado	10.00	9.50			
Ver usuarios	No iniciado	13.00	12.25			
Suspender encargo	No iniciado	10.00	9.00			
Ver encargos	No iniciado	13.00	12.75			
Alta de usuario	No iniciado	12.50	12.25			
Acusar orden	No iniciado	11.00	7.50			
Enviar solicitud	No iniciado	13.00	13.50			
Revisar solicitud	No iniciado	11.50	11.50			
Login de administrador	No iniciado	11.00	9.25			
Enviar orden	No iniciado	13.00	14.00			
▶ Sprint 4/11/2013 - 15/11/2013: Sprint 2						
▶ Sprint 18/11/2013 - 29/11/2013: Sprint 3						

Figura 5.16. Sprints con actualización de tiempos

Una vez terminado el tiempo del *sprint*, se procede a realizar la Reunión de Revisión de acuerdo a lo establecido por la actividad APE 2.11 del patrón, durante esta revisión el RAPE se asegura que cada pendiente este completo y modifica su estado a “Hecho” (ver Figura 5.17).



**Figura 5.17.** Actualización del estado de los pendientes

A modo de ejemplo, en este proyecto fueron aceptados 9 pendientes y 1 no, el cual fue regresado al *registro de pendientes* del producto, para que se decidiera en la siguiente reunión de planificación, correspondiente al segundo *sprint*, si se agregaba o no.

### 5.3.3. Evaluación y control

Cuando se termina la Reunión de Revisión se debe realizar la Reunión de Retrospectiva de acuerdo a lo establecido por la actividad APE 3.1 del patrón. En este sentido, para el desarrollo del proyecto hermano se recogieron las siguientes propuestas de mejora:

- Descontar del tiempo del *sprint* de 120 horas (60 horas efectivas por cada programador), el tiempo para la Reunión de Planeación (2 horas), el tiempo de la Reunión de Revisión (2 horas), y el tiempo de la Reunión de Retrospectiva (1 hora), quedando 110 horas (55 horas efectivas por cada programador).
- Dividir la actividad DMS 4.2 del patrón en más tareas que indiquen la estrategia para producir el pendiente.
- Agregar pruebas unitarias en la actividad DMS 4.2 para que sea obligatorio que todos los programadores las realicen.

### 5.3.4. Cierre

Con la terminación de la Reunión de Retrospectiva se da por terminado el primer *sprint* de acuerdo a la actividad APE A4.1 del patrón.

## 5.4. Resultados

Durante los cuatro meses que se dedicaron para desarrollar los proyectos hermanos, se recogieron datos en la gestión de proyectos que permitieron establecer una base comparativa. El sesgo en esta información fue controlado mediante el uso de verificaciones sobre documentación y entrevistas con el MS, RAPE y RDM.

Es importante mencionar que después de recoger toda la información relacionada con la experimentación, se determinó que el compromiso, la iniciativa y la responsabilidad de SOCIEPRO S.A.P.I. de C.V fueron factores clave que contribuyeron en gran medida a alcanzar los resultados resumidos en este capítulo. Esta experimentación también permitió obtener una combinación de experiencias valiosas, “más conocimientos sobre cómo hacer las cosas en pequeños entornos” y observar el comportamiento de una PyME durante el proceso de mejora con MoProSoft y Scrum.

### 5.4.1. Análisis detallado sobre los resultados de la experimentación

Con el objetivo de identificar si la mezcla de MoProSoft y Scrum contribuye a resolver la problemática identificada al inicio de esta tesis, se decidió junto con la PyME establecer que los principales objetivos a validar como resultado de la experimentación fueran:

- Mejorar la productividad del equipo,
- Incrementar la calidad del producto.

El proceso inició con la recogida de datos históricos de proyectos anteriores de SOCIEPRO S.A.P.I de C.V con el objetivo de facilitar la estimación de los proyectos hermanos y establecer una segunda base comparativa para determinar la eficiencia del patrón creado. Los datos recogidos corresponden a las estimaciones y medidas del calendario, tamaño y esfuerzo por fase, y número de defectos encontrados por fase y se calcularon medidas derivadas para realizar la comparación con los resultados del proyecto piloto.

#### 5.4.1.1. Verificación de los objetivos en los proyectos hermanos

La verificación de los objetivos en los proyectos hermanos se llevó a cabo mediante la comparación de los resultados obtenidos por el grupo experimental y el grupo de control con los objetivos establecidos en la Tabla 75.

##### 5.4.1.1.1. Objetivo 1: Cumplir con el calendario, esfuerzo y tamaño previsto

La Tabla 76 muestra que los resultados estimados y reales en el proyecto, asociados al calendario, esfuerzo y tamaño, determinan la utilidad del patrón. La Tabla 76 muestra que el grupo experimental logro terminar el proyecto dentro del calendario planificado. El esfuerzo obtenido fue negativo puesto que se terminó antes que el tiempo estimado por la empresa. Esto evidencia que el grupo de control, al carecer de elementos de gestión no puede mantener el ciclo dentro del tiempo o esfuerzo estimados.

Utilizando la misma tabla como referencia comparativa es posible analizar los beneficios de la implantación del patrón mediante la comparación del valor objetivo establecido por la empresa con el valor obtenido por la aplicación del patrón (denotado en adelante por VP). La desviación obtenida permite analizar qué tanto se aleja el valor obtenido del objetivo. Por ejemplo, el valor de -39.4% mostrado como desviación de esfuerzo considera el valor objetivo que debería ser menor del 10%, esto quiere decir que el 100% está representado por el 9.9% (o umbral establecido), por lo que el



cálculo se realiza como  $100 * 6$  (que es lo obtenido con el patrón) / 9.9, obteniendo un 60.6%. Por lo tanto, la diferencia es del  $60.6\% - 100\% = 39.4\%$ .

**Tabla 76.** Análisis de resultados para el objetivo relacionado con la productividad del equipo.

Objetivo	Valor objetivo	Grupo experimental		Grupo de control	
		VP	Desviación	Valor medido	Desviación
Desviación de calendario	<12%	10.7%	-10%	13.7%	15.1%
Desviación de esfuerzo	<10%	6%	-39.4%	11%	11.1%
Desviación de tamaño	<13%	13%	0%	13%	0%

Para la gestión efectiva del proyecto, el patrón incorporó actividades para la planificación del proyecto que establecieron la formulación de planes para mejorar la visión sobre el avance del proyecto hermano; así como actividades para la monitorización y control del plan de proyecto a través del método del Valor Ganado. Cabe mencionar que estas actividades permitieron que el RAPE detectara problemas a tiempo y los corrigieran sin generar algún retraso, a diferencia del grupo de control que utilizó la forma tradicional de trabajo carente de herramientas de gestión.

Para apoyar a la empresa con la adopción del patrón, se programaron reuniones semanales con los jefes de proyecto para evaluar el calendario, los objetivos, y los riesgos. El ET participó desde la formulación del plan inicial y trabajó de acuerdo a los objetivos establecidos por todos los integrantes. El uso de la herramienta de soporte permitió que se estableciera un entorno de trabajo más participativo e integrado que es apoyado por las actividades establecidas por MoProSoft.

#### 5.4.1.1.2. Objetivo 2: Reducir los tiempos en la identificación de defectos

Considerando de nuevo que los valores objetivo fueron establecidos a partir de los datos históricos de la empresa, la Tabla 77 muestra el análisis sobre la reducción de defectos. En este caso, los dos grupos fueron capaces de cubrir los umbrales permitidos puesto que cuentan con herramientas especializadas para realizar las pruebas unitarias y de integración. Los resultados mostraron una leve mejoría en el caso del grupo experimental puesto que a pesar de que el Plan de Pruebas de Integración y el Plan de Pruebas de Sistema no están incorporados a las actividades tomadas de DMS del patrón MoProSoft - Scrum, sí sirvió para que SOCIEPRO fortaleciera su proceso de *testing* que hizo posible definir, gestionar y controlar las pruebas sobre el código.

**Tabla 77.** Análisis de resultados para el objetivo relacionado con la calidad del producto.

Objetivo	Valor objetivo	Grupo experimental		Grupo de control	
		VP	Desviación	Valor medido	Desviación
Desviación de número de defectos encontrados por fase	<10%	6%	-39.3%	7%	-29.2%

Mediante el análisis de los resultados presentados se puede concluir que la aplicación de la mezcla de MoProSoft y Scrum, junto con la incorporación de la herramienta computacional en las actividades de SOCIEPRO S.A.P.I de C.V., han sido propicias para alcanzar los objetivos de los proyectos hermanos y de la organización.

De forma adicional a estos resultados favorables, el siguiente apartado del documento explicará los hallazgos alcanzados en relación a la validación de la hipótesis establecida al inicio de esta tesis.

## **5.5. Validación de la hipótesis de la tesis**

Por último, este apartado pretende reunir la evidencia que permita aceptar o rechazar la hipótesis planteada en el Capítulo 1 de esta tesis.

- *Hipótesis: Es posible desarrollar una herramienta para gestionar la integración de Scrum dentro de la categoría de procesos de Operación del modelo MoProSoft en las empresas mexicanas, con el objetivo de incrementar la productividad de la plantilla de trabajo y la calidad del producto.*

Esta hipótesis es aceptada dado que con la implantación de la mezcla de MoProSoft y Scrum en una herramienta de gestión (Capítulos 3 y 4 de esta tesis), se establece en la PyME un proceso eficiente de gestión de los proyectos software dado que ésta fue provista de prácticas ágiles que le permitieron planificar adecuadamente, efectuar el seguimiento apropiado, cerrar eficientemente y realimentar los datos obtenidos para nuevos proyectos. Concretamente las Tabla 76 y 77 mostraron que las desviaciones en calendario, esfuerzo, tamaño y defectos fueron reducidas considerablemente en el proyecto realizado por el grupo experimental.

Es importante mencionar que esta hipótesis, que ha sido establecida en el contexto de la tesis, es aceptada para el caso de la empresa que participó en el caso de estudio. De ninguna manera es posible generalizar que la solución propuesta en esta tesis proporcionará los mismos resultados en cualquier empresa debido a que esto requiere continuar con la experimentación incluyendo una muestra más significativa. Así pues, en el contexto de la experimentación en Ingeniería de Software se afirma que los resultados positivos sobre una efectividad parcial, mostrados en un trabajo de investigación o reporte, evidencian que es posible establecer un caso de estudio con un mayor número de participantes y proyectos [Juristo10].

## 6. Conclusiones y líneas futuras

Este capítulo presenta las principales conclusiones del presente trabajo de tesis, además de establecer futuras líneas de investigación que pueden dar paso como trabajos futuros en el área de mejora del proceso de software enfocadas en las MiPyMEs mexicanas.

Para lograr la mezcla “MoProSoft – Scrum” primero fue necesario identificar las categorías de procesos de MoProSoft que debían ser modeladas. Después, durante el análisis se identificaron las actividades dentro de APE y DMS en las cuales encajaban las actividades de Scrum. En este sentido, también fue necesario identificar las brechas entre estas actividades, tanto por sus diferencias como por las tareas que no son consideradas mutuamente. Por último, estos hallazgos fueron unidos dentro de MoProSoft, dando paso así a un nuevo patrón de procesos.

El proceso obtenido a partir de la mezcla “MoProSoft – Scrum” permitió obtener propuestas de mejora de parte de los ingenieros en la Reunión de Retrospectiva planteada por Scrum. El proceso además está preparado para recibir los cambios disminuyendo el riesgo de contar con un Software incompleto al finalizar el tiempo estimado.

La mezcla “MoProSoft – Scrum” también permite enfocarse a las 5 características descritas en la Sección 1.1, cumpliendo con 4 de éstas:

- Orientado a equipos pequeños.
- Entregas de producto en plazos cortos.
- Desarrollo en base a la especificación de los requisitos del cliente.
- Desarrollos iterativos y prototipos funcionales.

Además de mejorar la calidad que se tenía con la documentación generada y requerida por MoProSoft, mediante la inclusión de la documentación como requisito para poder revisar cada funcionalidad del producto.

Ya que el software en México es desarrollado principalmente por pequeños equipos, véase Sección 1.1, es posible ayudar a las empresas mexicanas, mediante el uso de la metodología Scrum dentro de MoProSoft, para que éstas cubran cuatro de las características que, de acuerdo con Gutiérrez et al., [Gutiérrez08], debe mostrar un modelo de calidad enfocado a estas empresas.

Al realizar esta tesis, que plantea la mezcla del modelo MoProSoft con la metodología Scrum, no se dejó de cumplir con las actividades de la norma, esto quiere decir que, en una evaluación hipotética sobre el cumplimiento de la norma, sería posible cubrir los niveles de madurez de capacidad en los procesos requeridos para obtener el Dictamen de la Unidad Verificadora de Tecnologías de información (UVTI).

“MoProSoft – Scrum” permite cumplir con el modelo completamente, pues no se le quitó ninguna definición o característica, sino por el contrario fue Scrum el que se adaptó al modelo. Como se observó en la experimentación de la Sección 5, “MoProSoft – Scrum” fue implementado con el nivel de capacidad 1, y se cumplió con lo establecido en la categoría de Operación, tanto en OPE.1 APE y OPE.2 DMS.

En este sentido, este nuevo patrón de procesos coincide con el patrón dialéctico de tesis, antítesis y síntesis defendido en [Boehm03]. Es decir, MoProSoft como modelo basado en procesos ha sido parte de la tesis que inicia el conocimiento para desarrollar sistemas de software. Scrum como metodología ágil representa a la antítesis que cuestiona a estos modelos basados en procesos. Y la mezcla “MoProSoft – Scrum” representa la síntesis, logrando un conocimiento más completo y depurado que fue enriquecido por ambos.

En cuanto a la herramienta de soporte a la integración de MoProSoft y Scrum, esta herramienta permitió llevar la gestión del plan de proyecto como un registro de pendientes de producto, y los ciclos como *sprints*, como es requerido por OPE.1 APE. Además que se le dio un orden a las actividades de OPE.1 APE pues en la definición de las actividades de MoProSoft de APE A1 Planificación no se tiene clara la secuencia que se debe seguir para cumplir con toda la actividad de planificación.

La herramienta también permitió incluir todas las tareas de las actividades de OPE.2 DMS como tareas de cada elemento del registro de pendientes, logrando con esto cumplir con los documentos de salida solicitados por el proceso DMS. Esto fue posible al convertir las tareas de DMS a requisitos para poder revisar cada elemento del registro de pendientes del producto realizados en cada *sprint*, aceptando así los elementos que, además de cumplir con las expectativas del cliente, tienen su documentación completa e integrada en la Configuración de Software, como lo marca el modelo MoProSoft, en su respectivo nivel de capacidad.

En este sentido y tomando en cuenta el establecimiento de posibles líneas de investigación a futuro, se considera que esfuerzos posteriores se pueden enfocar en las siguientes necesidades:

- Primero, se debe incrementar el número de prácticas de la herramienta, mediante la experimentación en un número mayor de proyectos y empresas. Logrando con esto, tener un mayor conocimiento sobre las prácticas que realmente funcionan en entornos de MiPyMEs. Para lograr esto se pone a disposición de otros investigadores y empresas el código fuente de la herramienta en la siguiente dirección: <http://github.com/chepike/MPSScrum>.
- Incorporar en la herramienta mecanismos de estimación, ya que Scrum deja la estimación de esfuerzo y sus unidades a decisión de la organización; con mecanismos de estimación automatizados, basados en variables de fácil visibilidad, se pueden disminuir el tiempo de estimación invertido en las reuniones de planeación ahorrando tiempo para revisar con mayor cuidado el plan generado en cada ciclo.
- Incorporar a la herramienta la gestión de riesgos utilizando el reporte de impedimentos obtenidos en las juntas diarias, ya que esta ventaja de Scrum permite al Maestro de Scrum gestionar los riesgos detectados por el equipo de desarrollo casi al momento en que se detectan.
- Agregar a la herramienta la generación automática de documentos de MoProSoft haciendo uso de la información almacenada en la base de datos, ya sea generando los documentos completamente o de manera parcial para después ser completados por los roles correspondientes.

Por último, se concluye que la investigación resultante sustenta la hipótesis propuesta en la sección 1.3 de esta tesis de maestría: *“Es posible desarrollar una herramienta para gestionar la integración de Scrum dentro de la categoría de procesos de Operación del modelo MoProSoft en las empresas mexicanas, con el objetivo de incrementar la productividad de la plantilla de trabajo y la calidad del producto”*.



## 7. Anexo A.- Acrónimos

MoProSoft	Modelo de Procesos de Software
CMMI	Capability Maturity Model Integration
MiPyMEs	Micro, Pequeñas y Medianas Empresas
PIB	Producto Interno Bruto
SEI	Software Engineering Institute
NYCE	Normalización y Certificación Electrónica S.C.
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
PMBok	Project Management Body of Knowledge
SWEBoK	Software Engineering Body of Knowledge
MPS.BR	Melhoria de Processo de Software Brasileiro
ROI	Return of Investment
TSP	Team Software Process
PSP	Personal Software Process
APE	Administración de Proyectos Específicos
DMS	Desarrollo y Mantenimiento de Software
BPMN	Business Process Modeling Notation
CMMI-DEV	Capability Maturity Model Integration for Development
XP	eXtreme Programming
SW-CMM	Capability Maturity Model for Software
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPM	Organizational Performance Management
OT	Organizational Training
OPP	Organizational Process Performance
PMC	Project Monitoring and Control
PP	Project Planning
SAM	Supplier Agreement Management

IPM	Integrated Project Management
RSKM	Risk Management
QPM	Quantitative Project Management
REQM	Requirements Management
PI	Product Integration
RD	Requirements Development
TS	Technical Solution
VAL	Validation
VER	Verification
CM	Configuration Management
MA	Measurement and Analysis
PPQA	Process and Product Quality Assurance
DAR	Decision Analysis and Resolution
CAR	Causal Analysis and Resolution
CAM	Capacity and Availability Management
UML	Unified Modeling Language
EvalProSoft	Método de Evaluación de procesos para la industria de software
DIR	Alta Dirección
GER	Gerencia
OPE	Operación
GES	Gestión
DSDM	Dynamic Systems Development Method
RGPY	Responsable de Gestión de Proyectos
RAPE	Responsable de la Administración del Proyecto Específico
CL	Cliente
RSC	Responsable del Subcontrato
RDM	Responsable de Desarrollo y Mantenimiento de Software
ET	Equipo de Trabajo
US	Usuario
AN	Analista
DU	Diseñador de la Interfaz de Usuario
DI	Diseñador
PR	Programador
RPU	Responsable de Pruebas
RE	Revisor
RM	Responsable de Manuales
DIR.1: PCI	Plan de Comunicación e Implantación
GES.1: PMP	Plan de Mediciones de Procesos
GES.1: DMS	Desarrollo y Mantenimiento de Software



---

GES.2: DP	Descripción del Proyecto
GES.2: RAPE	Responsable de Administración del Proyecto Específico
GES.2: MCP	Metas Cuantitativas para el Proyecto
GES.2: ACP	Acciones Correctivas o Preventivas
GES.3.1: AR	Asignación de Recursos
OPE.2: RA	Reporte de Actividades
OPE.2: RMSM	Reporte de Mediciones y Sugerencias de Mejora
OPE.2: CS	Configuración de Software
CL: SC	Solicitud de Cambios
OPE.1: RMSM	Reporte de Mediciones y Sugerencias de Mejora
OPE.1: PP	Plan del Proyecto
OPE.1: RS	Reporte de Seguimiento
OPE.1: DA	Documentación de Aceptación
OPE.1: PAC	Plan de Adquisiciones y Capacitación
OPE.1: LA	Lecciones Aprendidas
OPE.1: PD	Plan de Desarrollo
OPE.2: ER	Especificación de Requisitos
OPE.2: AD	Análisis y Diseño
OPE.2: C	Componente
OPE.2: S	Software
OPE.2: CS	Configuración de Software
OPE.2: MU	Manual de Usuario
OPE.2: MO	Manual de Operación
OPE.2: MM	Manual de Mantenimiento
OPE.2: RA	Reporte de Actividades
OPE.2: LA	Lecciones Aprendidas
OPE.2: RMSM	Reporte de Mediciones y Sugerencias de Mejora
OPE.2: RR	Registro de Rastreo
OPE.2: PPS	Plan de Pruebas de Sistema
OPE.2: RPS	Reporte de Pruebas de Sistema
OPE.2: PPI	Plan de Pruebas de Integración
OPE.2: RPI	Reporte de Pruebas de Integración
DP	Dueño del Producto
ED	Equipo de Desarrollo
MS	Maestro de Scrum
DP: RPP	Registro de Pendientes del Producto
ED: RPS	Registro de Pendientes del Sprint
ED: I	Incremento



## 8. Anexo B.- Carta de colaboración



www.sociepro.com

Huajuapán de León, Oaxaca a 4 de Noviembre de 2013

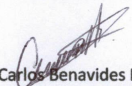
A quien corresponda:

A través de la presente confirmo que el **C. José Antonio García Hernández** ha recogido información relacionada con los proyectos de software que han sido desarrollados dentro de la empresa que dirijo desde el 2009, como parte de su trabajo de tesis denominado "Herramienta para la integración de la metodología Scrum con MoProSoft para incrementar la productividad del personal y la calidad del producto en las MiPyMEs de desarrollo software". Específicamente, reconozco nuestra participación en la adopción del patrón MoProSoft – Scrum utilizado para el desarrollo del software denominado "POC de Beople Light 1.0 release 1.0".

Por último, en SOCIEPRO consideramos que esta tesis no brindará beneficios únicamente en el contexto académico, puesto que puede ser un esfuerzo importante por ampliar los horizontes en cuanto a la formación de recursos humanos de calidad que requiere la industria actualmente.

En caso de requerir más información, por favor no duden en contactarme.

Quedo de Uds.

  
Ing. Carlos Benavides Martínez  
Gerente de la FVP de  
SOCIEPRO S.A.P.I de C.V.



**SociePro S.A.P.I de C.V.**  
Priv. Matamoros #5, Col. Centro, Huajuapán de León  
Oaxaca, C.P. 69000  
Tels: (953) 5300513, (953) 5320279



## 9. Bibliografía

- [**Basili86**] Basili, V. R., Selby, R. W. & Hutchens, D. H. “Experimentation in software engineering” *IEEE Transactions on Software Engineering*, 12(7): 733-743, 1986
- [**Beck04**] Beck, K. & Andres, C. *Extreme programming explained: Embrace change*. Second Edition, NY: Addison-Wesley Professional. 2004.
- [**Boehm03**] Boehm, B. & Turner, R. *Balancing agility and discipline*. Boston: Addison-Wesley. 2003.
- [**Buede11**] Buede, D. M. *The engineering design of systems: models and methods*. Third Edition. Hoboken, NY: John Wiley & Sons. 2011.
- [**CMMI02**] CMMI Product Team. “*CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1). Continuous Representation*” CMU/SEI-2002-TR-011, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. 2002.
- [**CMMI06**] CMMI Product Team. “*CMMI for Development (CMMI-DEV, v1.2)*” CMU/SEI-2006-TR-008, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. 2006.
- [**CMMI10**] CMMI Product Team. “*CMMI for Development (CMMI-DEV, v1.3)*” CMU/SEI-2010-TR-033, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. 2010.
- [**Davila08**] Dávila-Muñoz, M. & Oktaba, H. “Desarrollo de una especialización de MoProSoft basada en el método ágil Scrum” *Tesis para obtener el grado de Maestro en Ciencias de la Computación*. Universidad Autónoma de México, Instituto de Investigación en Matemáticas Aplicadas y Sistemas. 2008.
- [**Deemer08**] Deemer, P., Benefield, G., Larman, C. & Vodde, B. “*The scrum primer*”. Version 1.2. Scrum Training Institute. 2008.
- [**Freund12**] Freund, J. & Rücker, B. *Real-life BPMN: Using BPMN 2.0 to analyze, improve, and automate processes in your company*. First Edition, MA: CreateSpace Independent Publishing Platform. 2012.
- [**Gutiérrez08**] Gutiérrez, A., Gutiérrez, E., Pérez, A. & Márquez, L. “Acerca de la implementación de los modelos de calidad en la construcción de software en México” *Revista Tencua – Tecnología y Cultura Afirmando el Conocimiento*, 13(25): 119-132, 2009.
- [**Herrera03**] Herrera, E. & Ramírez, R. “A methodology for self-diagnosis for software quality assurance in small and medium-sized industries in Latin America” *The Electronic Journal on Information Systems in Developing Countries*, 4(1): 1-13, 2003.
- [**Humphrey97**] Humphrey, W. S. *Introduction to the personal software process*. SEI Series in Software Engineering, Reading, MA: Addison-Wesley. 1997.

- [**Humphrey02**] Humphrey, W. S. *Introduction to the team software process*. SEI Series in Software Engineering, Reading, MA: Addison-Wesley. 2002.
- [**IEEE04**] Institute of Electrical and Electronics Engineers. *Guide to the software engineering body of knowledge (SWEBoK)*. New York: IEEE. 2004.
- [**ISO95**] International Organization for Standardization/International Electrotechnical Commission. “*ISO/IEC 12207:1998(e): Information technology – software life cycle processes*” Geneva, Switzerland: ISO/IEC. 1995.
- [**ISO00**] International Standardization for Organization. “*ISO 9000: quality management systems – Fundamentals and vocabulary*” Third Edition. Stage: 60.60. TC 176/SC 1. 2000.
- [**ISO04**] International Organization for Standardization/International Electrotechnical Commission. “*ISO/IEC 15504:2003/cor.1:2004(e): Information technology – process assessment. Parts 1-5*” Geneva, Switzerland: ISO/IEC. 2004.
- [**ISO08**] International Organization for Standardization/International Electrotechnical Commission. “*ISO/IEC 12207:2008: Information technology – software life cycle processes*” Geneva, Switzerland: ISO/IEC. 2008.
- [**Jakobsen09**] Jakobsen, C. R. & Sutherland, J. “Scrum and CMMI going from good to great: are you ready-ready to be done-done” *Proc. of the Agile Conference (AGILE '09)*, IEEE Computer Society, pp. 333-337, 2009.
- [**Juristo10**] Juristo, N. & Moreno, A. *Basics of software engineering experimentation*. Dordrecht, the Netherlands: Kluwer Academic Press. 2010.
- [**Kitchenham95**] Kitchenham, B., Pickard, L. & Pfleeger, S. L. “Case studies for method and tool evaluation” *IEEE Software*, 12(4): 52-62, 1995.
- [**Mahanti06**] Mahanti, A. “Challenges in enterprise adoption of agile methods – a survey” *Journal of Computing and Information Technology*, 14(3): 197-206, 2006.
- [**Mishra08**] Mishra, D. & Mishra, A. “A study of software process improvement in small and medium organizations” In: *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, Oktaba, H, & Piattini, M. (Eds). Hershey, PA: Information Science Reference Publisher, pp. 141-159, 2008.
- [**Montoni09**] Montoni, M. A., Rocha, R. & Weber, K. C. “MPS.BR: a successful program for software process improvement in Brazil” *Software Process: Improvement and Practice*, 14(5): 289-300, 2009.
- [**NYCE05**] Normalización y Certificación. *NMX-I-059/02-NYCE-2005. Tecnología de la Información -Software- Modelos de procesos y evaluación para desarrollo y mantenimiento de software. Parte 02: Requisitos de procesos (MoProSoft)*. México, DF: NYCE. 2005.
- [**Oktaba06**] Oktaba, H. “MoProSoft: A software process model for small enterprises” *Proc. of the First International Research Workshop for Process Improvement in Small Settings*, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2006-SR-001, pp. 93-101, 2006.
- [**Oktaba08**] Oktaba, H. & Piattini, M. *Software process improvement for small and medium enterprises: Techniques and case studies*. Hershey, PA: Information Science Reference Publisher. 2008.
- [**Palacio11**] Palacio, J. & Ruata, C. *Scrum manager: Gestión de proyectos*. Editorial Autoedición. 2011.

- [Pasini13] Pasini, A., Esponda S., Boracchia, M. & Pesado, P. “Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110”. *Proc. of the Congreso Argentino de Ciencias de la Computación (CACIC 2013)*, Universidad CAECE, pp. 898-909, 2013.
- [PMI04] Project Management Institute. *A guide to the project management body of knowledge (PMBOK guide)*. Newton Square, PA: Project Management Institute. 2004.
- [Richardson07] Richardson, I. & von Wangenheim, C. “Why are small software organizations different” *IEEE Software*, 24(1): 18-22, 2007.
- [Rueda10] Rueda, A. “Propuesta de una guía para interpretar los procesos de MoProSoft de la categoría de operación usando una combinación de métodos ágiles” *Tesis para obtener el grado de Maestro en Ciencias en Informática*. Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas. 2010.
- [Schwaber10] Schwaber, K. & Sutherland, J. *Scrum: Developed and sustained by Ken Schwaber and Jeff Sutherland*. Scrum.org. 2010.
- [Sommerville09] Sommerville, I. *Software engineering*. Ninth Edition. Boston, MA: Pearson Education, Inc. 2009.
- [Sutherland07] Sutherland, J., Jacobson, C. & Johnson, K. “Scrum and CMMI level 5: A magic potion for code warriors!” *Proc. of the Agile Conference (AGILE '07)*, IEEE Computer Society, pp. 134-145, 2007.
- [Sutherland10] Sutherland, J. & Schwaber, K. *The scrum papers: Nut, bolts, and origins of an agile framework*. Scrum Training Institute. 2010.
- [Valenzuela07] Valenzuela, L., Flores, B. L. & Olguín, J. M “Arquitectura para la coordinación de flujos de trabajo de MoProSoft por niveles de capacidad de procesos” *Presented in 2º Congreso Internacional en Ciencias Computacionales*, pp. 153–155, México, 2007.
- [Wohlin03] Wohlin, C., Hoest, M. & Henningsson, K. “Empirical research methods in software engineering” Conradi, R. & Wang, A. (Eds.), *Empirical Methods and Studies in Software Engineering*, Lecture Notes in Computer Science, vol. 2765, Berlin-Heidelberg, Germany: Springer-Verlag, pp. 145-165, 2003.
- [Wohlin12] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslén, A. *Experimentation in software engineering*. Boston, MA: Kluwer Academic Publishers, 2012.
- [Yin09] Yin, R. K. *Case study research: Design and methods*. Fourth Edition, Thousand Oaks, CA: Sage Publications. 2009.
- [Zavala08] Zavala-Ruiz, J. “Organizational analysis of small software organizations: Framework and case study” In: *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, Oktaba, H, & Piattini, M. (Eds). Hershey, PA: Information Science Reference Publisher, pp. 1-41, 2008.

## 9.1. Sitios de Internet

- [URL-1] Secretaría de Economía de México. Disponible en: <http://www.economia.gob.mx>. Último acceso: Enero 2014.
- [URL-2] Systematic – software products. Disponible en: <http://www.systematic.com>. Último acceso: Enero 2014.
- [URL-3] Software Engineering Institute. Disponible en: <http://www.sei.cmu.edu>. Último acceso: Enero 2014.

- [URL-4] Manifiesto for Agile Software Development. Disponible en: <http://agilemanifesto.org>.  
Último acceso: Enero 2014.
- [URL-5] Brief History of CMMI. Software Engineering Institute. Disponible en: <http://www.sei.cmu.edu/library/assets/cmmihistory.pdf>. Último acceso: Enero 2014.
- [URL-6] Extreme Programming: A Gentle Introduction. Disponible en: <http://www.extremeprogramming.org>. Último acceso: Enero 2014.
- [URL-7] Kualí: Herramienta Auxiliar para la implementación de MoProSoft. Disponible en: <http://team.intellekt.ws/media/p/13163.aspx>. Último acceso: Diciembre 2013.
- [URL-8] Banana Scrum, Scrum tool for agile teams. Disponible en: <http://www.bananascrum.com>. Último acceso: Enero 2014.
- [URL-9] iceScrum, Your open source agile tool. Disponible en: <http://www.icescrum.org>.  
Último acceso: Enero 2014.
- [URL-10] Agile Project Management Solution – Mingle. Disponible en: <http://www.thoughtworks-studios.com/mingle-agile-project-management>. Último acceso: Enero 2014.
- [URL-11] Agile Project Management for Agile Development. Disponible en: <http://www.rallydev.com>. Último acceso: Enero 2014.
- [URL-12] ScrumNinja - Scrum Software for Agile Development. Disponible en: <http://scrumninja.com/scrum-software>. Último acceso: Enero 2014.
- [URL-13] Scrum Tools – ScrumWorks Pro & ScrumWorks Basic. Disponible en: <http://www.danube.com/scrumworks>. Último acceso: Enero 2014.
- [URL-14] Sprintometer – Free modern agile tool for Scrum and XP projects management and tracking. Disponible en: <http://sprintometer.com>. Último acceso: Enero 2014.
- [URL-15] Tackle. Disponible en: <http://tackle.codeplex.com>. Último acceso: Enero 2014.
- [URL-16] TeamTrick: Scrum on Rails. Disponible en: <http://team-trick.net>. Último acceso: Octubre 2013.
- [URL-17] VersionOne – Agile Project Management, Agile Development and Scrum Tool. Disponible en: <http://www.versionone.com>. Último acceso: Enero 2014.
- [URL-18] Ralph Stacey's Agreement & Certainty Matrix. Disponible en: [http://www.plexusinstitute.com/edgeware/archive/think/main\\_aides3.html](http://www.plexusinstitute.com/edgeware/archive/think/main_aides3.html). Último acceso: Julio 2013.