



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“Herramienta para establecer y controlar iniciativas de mejora al proceso software basadas en el modelo MoProSoft”

TESIS

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

PRESENTA

ING. DAGOBERTO CRUZ SANDOVAL

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

HUAJUAPAN DE LEÓN, OAX.; JUNIO DE 2014

**Tesis presentada el
ante los siguientes sinodales:**

Dr. José Aníbal Arias Aguilar
Dr. Santiago Omar Caballero Morales
Dra. Carla Lenín Pacheco Agüero
M. C. Everth Haidee Rocha Trejo

Director de Tesis:

Dr. Iván A. García Pacheco

Dedicatoria

Agradecimientos

Índice

Índice	ix
Lista de tablas	xiii
Lista de figuras	xv
Resumen	xix
1. Introducción.....	1
1.1. Importancia del problema y necesidad de la solución.....	5
1.2. Delimitaciones de la tesis	9
1.3. Limitaciones de la tesis	9
1.4. Objetivos de la tesis.....	9
1.4.1. Objetivo general	10
1.4.2. Objetivos específicos.....	10
1.5. Solución propuesta	10
1.6. Estructura de la tesis.....	13
1.7. Publicaciones generadas	14
2. Marco Conceptual.....	15
2.1. Antecedentes.....	15
2.2. Mejora al Proceso de Software.....	17
2.2.1. IDEAL: Un modelo para mejorar los procesos software	20
2.3. Modelos de procesos y evaluación para el desarrollo y mantenimiento de software para la industria de software en México.....	21
2.3.1. Modelo de Procesos para la Industria Software (MoProSoft).....	23
2.3.1.1. Estructura.....	24
2.3.1.1.1. Categorías de procesos	24
2.3.1.1.2. Procesos	25
2.3.1.1.3. Patrón de procesos y uso del modelo.....	26
2.3.2. Método de Evaluación de Procesos para la Industria Software (EvalProSoft)	28
2.3.2.1. Usos del método de evaluación	29
2.3.2.2. Modelo de capacidades de procesos	30
2.3.2.3. Proceso de evaluación	31
2.4. Impacto de MoProSoft en la industria de software	32
2.4.1. Proyecto CompetiSoft	34
2.4.2. Estándar ISO/IEC 29110	35
2.5. Herramientas auxiliares para la implantación de iniciativas de SPI.....	37
2.5.1. processMax®.....	38
2.5.2. iGrafx®.....	40
2.5.3. Stages®.....	42
2.5.4. KWE 2.0.....	45

2.6. Análisis empírico sobre las herramientas de soporte a SPI	47
2.6.1. Definición de criterios de comparación	48
2.6.2. Comparación de las herramientas SPI analizadas	49
3. Desarrollo de una herramienta colaborativa para establecer y controlar iniciativas de SPI basadas en el modelo MoProSoft	51
3.1. Metodología de desarrollo	52
3.2. Especificación de requerimientos	57
3.2.1. Ámbito del sistema	58
3.2.2. Descripción General	59
3.2.3. Requerimientos de Kaizen	60
3.2.3.1. Definición de casos de uso	60
3.2.3.2. Diagrama de flujo de datos	63
3.3. Análisis y diseño del sistema	66
3.3.1. Método para el diseño y creación de modelos para aplicaciones RIA	66
3.3.1.1. Patrones RIA	66
3.3.1.2. Ingeniería Web basada en UML (UWE)	67
3.3.1.3. Los patrones RIA en el diseño con UWE	68
3.3.1.4. Modelo conceptual	70
3.3.1.5. Modelo de navegación	72
3.3.1.6. Modelo presentación	75
3.4. Construcción del sistema	77
3.4.1. Tecnología de desarrollo	77
3.4.2. Arquitectura del sistema	80
3.4.3. Método de desarrollo	82
3.4.4. Incremento compromiso	83
3.4.5. Incremento evaluación	84
3.4.5.1. Evaluación por cuestionario	84
3.4.5.2. Evaluación por diagrama de procesos	88
3.4.5.3. Combinación de resultados y obtención de resultado final	91
3.4.6. Incremento para generación de planes de mejora	92
3.4.6.1. Codificación	95
3.4.6.2. Creación del individuo (cromosoma)	95
3.4.6.3. Crear la población inicial	95
3.4.6.4. Fortaleza (fitness)	96
3.4.6.5. Selección natural	96
3.4.6.6. Operador de cruce	96
3.4.6.7. Mutación	97
3.4.6.8. Criterio de parada	97
3.4.6.9. Creación de la calendarización	98
3.4.6.10. Algoritmo para generar el plan de actividades	99
3.4.7. Incremento para implantación de planes de mejora	100
4. Resultados experimentales	103
4.1. Definición del caso de estudio	104
4.1.1. Establecimiento de la línea base inicial	105
4.1.2. Implantación de la mejora por medio de Kaizen en un proyecto piloto	109
4.1.3. Comparación evaluación final con la línea base inicial	112
4.1.4. Verificación de los objetivos particulares	115
5. Conclusiones	117

6. Anexo A.- Acrónimos	121
7. Anexo B.- Cuestionario para la evaluación del modelo MoProSoft	123
7.1. Gestión de Negocio	123
7.2. Gestión de Procesos	125
7.3. Gestión de Proyectos	126
7.4. Gestión de Recursos	128
7.5. Administración de Proyectos Específicos	129
7.6. Desarrollo y Mantenimiento de Software.....	131
8. Anexo C.- Especificación de Casos de Uso	135
8.1. Casos de uso Administrador	135
8.1.1. Iniciar sesión.....	136
8.1.2. Registrar empresa	136
8.1.3. Registrar usuario.....	137
8.1.4. Actualizar empresa	138
8.1.5. Consultar empresa	139
8.1.6. Generar reporte de estadísticas	139
8.2. Casos de uso Líder de Proyecto.....	140
8.2.1. Registrar miembro de equipo	141
8.2.2. Alta proyecto de mejora	141
8.2.3. Elegir procesos a mejorar	142
8.2.4. Asignar procesos a roles	142
8.2.5. Establecer compromiso.....	143
8.2.6. Consultar plan de actividades	144
8.2.7. Asignar actividades	145
8.2.8. Reporte de actividades.....	145
8.2.9. Iniciar proyecto de mejora.....	146
8.2.10. Consultar estado del proyecto de mejora.....	146
8.2.11. Consultar resultados de evaluación	147
8.2.12. Consultar progreso de actividades	148
8.2.13. Generar reporte de proyecto de mejora	148
8.3. Casos de uso para Miembro de Equipo	149
8.3.1. Realizar evaluación.....	150
8.3.2. Responder cuestionario.....	150
8.3.3. Modelar diagrama de procesos	151
8.3.4. Agregar elemento	152
8.3.5. Eliminar elemento.....	153
8.3.6. Crear conexión.....	153
8.3.7. Eliminar conexión	154
8.3.8. Consultar plan de actividades	155
8.3.9. Finalizar actividad	155
8.3.10. Generar mensaje/anuncio	156
9. Anexo D.- Experiencia de uso: proyecto de mejora para implantar un proceso DMS-MoProSoft nivel 1	157
9.1. Configuración de Empresa y Responsable de Mejora.....	157
9.2. Configuración de la Iniciativa y Establecimiento del Compromiso.....	158
9.3. Evaluación del proceso actual y presentación de resultados	161
9.4. Definición de infraestructura y planificación para la mejora	163
9.5. Implantación de la mejora	165

10. Anexo E. – Actas de Publicaciones	167
11. Referencias Bibliográficas.....	175
11.1. Sitios de Internet	180

Lista de tablas

Tabla 1. Participación por segmento de software y sectores de demanda [Mochi, 2007].....	6
Tabla 2. Estructura de la norma NMX-I-059-NYCE-2005 [NYCE, 2005].	23
Tabla 3. Usos del método de evaluación [NYCE, 2005a].....	29
Tabla 4. Niveles de capacidad y atributos de proceso de MoProSoft [Oktaba, 2005b].	30
Tabla 5. Grado de cumplimiento de un atributo [NYCE, 2005a].	31
Tabla 6. Calificación del nivel de capacidad del proceso [NYCE, 2005a].	32
Tabla 7. Datos de esfuerzo y mejora por empresa [Oktaba, 2006].	33
Tabla 8. Descripción de las partes del estándar ISO/IEC 29110.....	36
Tabla 9. <i>Benchmarking</i> sobre las herramientas analizadas.	49
Tabla 10. Relación de productos.	52
Tabla 11. Roles involucrados y descripción.....	53
Tabla 12. Actividades a desarrollar durante el desarrollo de Kaizen.	54
Tabla 13. Características básicas de la herramienta Kaizen.....	59
Tabla 14. Descripción del actor: Administrador.	62
Tabla 15. Descripción del actor: Líder de Proyecto.....	62
Tabla 16. Descripción del actor: Miembro de Equipo.....	62
Tabla 17. Descripción de patrones RIA especificados en UWE [Scott, 2007].	68
Tabla 18. Descripciones de estereotipos de navegación en UWE.....	72
Tabla 19. Comparativa sobre las tecnologías para desarrollo de aplicaciones RIA.....	78
Tabla 20. Descripción de incrementos.	82
Tabla 21. Elección de respuestas para preguntas.	84
Tabla 22. Cuestionario de DMS para nivel 1.	85
Tabla 23. Símbolos para la elaboración de diagramas de procesos [Fowler et al., 1999].....	88
Tabla 24. Perfil de las empresas participantes en el caso de estudio.....	104
Tabla 25. Principales diferencias entre los procesos de las MPyME y los procesos redefinidos por Kaizen.	111
Tabla 26. Valores objetivo del proyecto piloto para cada MPyME.	112
Tabla 27. Incremento en los niveles de cobertura por proceso.	114
Tabla 28. Análisis de resultados para el cumplimiento de objetivos por MPyME.	116

Lista de figuras

Figura 1.1. Modelo clásico de la gestión.	1
Figura 1.2. Descripción general del modelo AFIM [Cuevas et al., 2005].....	4
Figura 1.3. Empresas certificadas en MoProSoft por NYCE [URL-4].....	8
Figura 1.4. Contexto de trabajo de la solución propuesta.	12
Figura 1.5 Descripción general de los objetivos específicos a alcanzar.....	13
Figura 2.1. Principales causas de no cumplir con los requerimientos del cliente en proyectos software [Standish, 2009].....	17
Figura 2.2. Ciclo de Deming para la mejora continua de procesos [Deming, 1982].	18
Figura 2.3. Modelo IDEAL [McFeeley, 1996].....	20
Figura 2.4. Estrategias de PROSOFT [SE, 2006].....	22
Figura 2.5. Estructura de MoProSoft [NYCE, 2005].	25
Figura 2.6. Diagrama de relación de procesos [NYCE, 2005].....	27
Figura 2.7. Relación entre los elementos del método de evaluación.....	29
Figura 2.8. Relación de elementos del proceso de evaluación [NYCE, 2005a].....	30
Figura 2.9. Diagrama de proceso de evaluación [NYCE, 2005].	33
Figura 2.10. Distribución de certificaciones por entidad federativa [URL-4].....	34
Figura 2.11. Posicionamiento de las partes dentro del marco de trabajo de ISO/IEC 29110 [ISO, 2011a].	37
Figura 2.12. Generador de reportes de processMax® [URL-16].	39
Figura 2.13. Pantalla para modelar procesos en iGrafx® [URL-17].....	41
Figura 2.14. Flujo de procesos generado con Stages® [URL-11].....	44
Figura 2.15. Repositorio de actividades y productos de trabajo en KWE 2.0 [URL-4].....	46
Figura 2.16. Pantalla de designación de actividades en KWE 2.0 [URL-4].	47
Figura 3.1. Diagrama de flujo de trabajo para la metodología de desarrollo.	56
Figura 3.2. Diagrama detallado de la fase de construcción.	57
Figura 3.3. Plataforma colaborativa para la implantación de iniciativas de SPI en MPyMEs.	61
Figura 3.4. Relación entre actores.	63
Figura 3.5. Diagrama de casos de uso Administrador.	63
Figura 3.6. Diagrama de contexto de Kaizen.	64
Figura 3.7. Diagrama 0 para el sistema Kaizen.	65
Figura 3.8. Relación de modelos en el método UWE [Koch et al., 2008].	67
Figura 3.9. Metamodelo para la capa de presentación de UWE [Koch et al., 2009].	68
Figura 3.10. Modelo conceptual de Kaizen.....	71
Figura 3.11. Modelo de navegación de Kaizen.	74
Figura 3.12. Modelo de presentación para Kaizen.	76
Figura 3.13. Arquitectura de Kaizen.	81
Figura 3.14. Conversión diagrama de procesos a grafo dirigido.....	90

Figura 3.15. Algoritmo genético general.....	94
Figura 3.16. Algoritmo para crear cromosomas.....	95
Figura 3.17. Algoritmo para la creación de la población inicial.....	96
Figura 3.18. Algoritmos de selección natural.....	96
Figura 3.19. Representación algorítmica del operador de cruce.....	97
Figura 3.20. Algoritmo de mutación.....	98
Figura 3.21. Algoritmo para obtener calendario de mejor individuo.....	99
Figura 3.22. Algoritmo para generar plan de actividades.....	99
Figura 3.23. Control de implantación de actividades de mejora.....	101
Figura 4.1. Fase de evaluación presentada en Kaizen: (1) basada en cuestionarios, (2) a través de modelado.....	106
Figura 4.2. Pantalla de resultados de Kaizen: (1) para jefes de proyecto, (2) selección de actividades.....	107
Figura 4.3. Nivel de cobertura de las MPyME para los procesos de la categoría Operación.....	108
Figura 4.4. Pantalla de planificación de actividades para el nuevo proceso.....	110
Figura 4.5. Información de actividades del nuevo proceso.....	111
Figura 4.6. Gráfica de comparación de coberturas antes y después de la implantación del nuevo proceso.....	113
Figura 8.1. Diagrama de casos de uso Administrador.....	135
Figura 8.2. Diagrama actividades Iniciar Sesión.....	136
Figura 8.3. Diagrama de actividades Registrar Empresa.....	137
Figura 8.4. Diagrama de actividades Registrar Usuario.....	138
Figura 8.5. Diagrama de actividades Actualizar Empresa.....	138
Figura 8.6. Diagrama de actividades Consultar Empresa.....	139
Figura 8.7. Diagrama de actividades Generar Reporte de Estadísticas.....	140
Figura 8.8. Casos de Uso Líder de Proyecto.....	140
Figura 8.9. Diagrama de Actividades Registrar Miembro de Equipo.....	141
Figura 8.10. Diagrama de actividades de Alta Proyecto Mejora.....	142
Figura 8.11. Diagrama de Actividades Elegir Procesos a Mejorar.....	143
Figura 8.12. Diagrama de Actividades Asignar Proceso a Roles.....	143
Figura 8.13. Diagrama de Actividades Establecer Compromiso.....	144
Figura 8.14. Diagrama de Actividades Consultar Plan de Actividades.....	144
Figura 8.15. Diagrama de Actividades Asignar Actividades.....	145
Figura 8.16. Diagrama de Actividades Generar Reporte de Actividades.....	146
Figura 8.17. Diagrama de Actividades Iniciar Proyecto de Mejora.....	146
Figura 8.18. Diagrama de Actividades Consultar Estado del Proyecto de Mejora.....	147
Figura 8.19. Diagrama de Actividades Consultar Resultados Evaluación.....	148
Figura 8.20. Diagrama de Actividades Consultar Progreso de Actividades.....	148
Figura 8.21. Diagrama de Actividades Reporte de Proyecto de Mejora.....	149
Figura 8.22. Casos de uso para Miembro de Equipo.....	150
Figura 8.23. Diagrama de actividades Realizar Evaluación.....	150
Figura 8.24. Diagrama Actividades Responder Cuestionario.....	151
Figura 8.25. Diagrama de actividades Modelar Diagrama de Procesos.....	152
Figura 8.26. Diagrama de actividades Agregar Elemento.....	153
Figura 8.27. Diagrama de actividades Eliminar Elemento.....	153
Figura 8.28. Diagrama de actividades Crear Conexión.....	154
Figura 8.29. Diagrama de actividades Eliminar Conexión.....	155
Figura 8.30. Diagrama de Actividades Consultar Plan de Actividades.....	155

Figura 8.31. Diagrama de actividades Finalizar Actividad.	156
Figura 8.32. Diagrama de actividades Generar Mensaje/Anuncio.	156
Figura 9.1. Alta empresa (1) y registro de responsable de iniciativa de mejora (2).	157
Figura 9.2. Ingreso a Kaizen.	158
Figura 9.3. Registro de equipo de trabajo y programa de mejora.	159
Figura 9.4. Registro de nueva iniciativa de mejora.	159
Figura 9.5. Configuración de inicio y evaluación.	159
Figura 9.6. Elección de los procesos a trabajar durante la iniciativa.	160
Figura 9.7. Configuración del equipo de trabajo.	160
Figura 9.8. Notificación de nuevas evaluaciones.	161
Figura 9.9. Evaluación por cuestionario.	162
Figura 9.10. Evaluación por diagrama de procesos.	162
Figura 9.11. Resumen de los resultados de evaluación.	163
Figura 9.12. Análisis de resultados y configuración de la planificación.	163
Figura 9.13. Planificación de la iniciativa de mejora.	164
Figura 9.14. Información detallada para cada tipo de actividad.	164
Figura 9.15. Asignación de actividades para los miembros de equipo.	165
Figura 9.16. Notificación de caducidad de fecha de entrega.	166
Figura 9.17. Envío de mensajes.	166

Resumen

En los últimos años la mejora al proceso software se ha posicionado como el enfoque correcto para incrementar la calidad de los productos software, en base a su principal premisa: un proceso de desarrollo capaz y maduro tiene como consecuencia un producto de mayor calidad. Las fases más importantes de un programa de mejora incluyen el compromiso con la mejora, evaluación del proceso actual, generación de planes de mejora e implantación de la mejora. En México la mayoría de las iniciativas de mejora se encuentran basadas en el modelo MoProSoft, enfocado en micro, pequeñas y medianas empresas desarrolladoras de software. Sin embargo, las pequeñas organizaciones en México carecen de conocimiento y experiencia en este tipo de proyectos, por lo que la ejecución de un programa de mejora se torna como una desgastante y difícil tarea. Este trabajo de tesis, se enfoca en el desarrollo de un marco de trabajo para establecer y controlar mejoras de procesos basadas en el modelo MoProSoft, que permita a las empresas ejecutar un proyecto de mejora mediante las fases compromiso, evaluación, planeación y ejecución, tomando como referencia al modelo MoProSoft. El marco de trabajo viene acompañado del desarrollo de un sistema computarizado llamado Kaizen, para automatizar las actividades y fases de un programa de mejora, con el objetivo de que las empresas encuentren en Kaizen el soporte perfecto para la implantación de este tipo de iniciativas. La reducción de costos y tiempo en la implantación de la mejora es la principal meta de este trabajo, así como proporcionar a las empresas mexicanas un sistema que les brinde apoyo en las fases necesarias para llevar a cabo una iniciativa de mejora sobre su proceso software de forma efectiva.

Palabras clave

Mejora al proceso software, MoProSoft, calidad de software, iniciativa de mejora, pequeñas empresas, evaluación y mejora de procesos, planificación y control de la mejora de procesos.

1. Introducción

La gestión puede ser definida como el conjunto de actividades y tareas llevadas a cabo por una o más personas con el fin de planificar y controlar el trabajo que otras realizan para alcanzar objetivos que de otro modo no se pudieran lograr [Christensen y Thayer, 2001]. En este sentido, la Figura 1.1 muestra el modelo clásico de gestión, el cual se basa en cinco funciones principales (la planeación, la organización, la dotación de personal, la conducción, y el control). De acuerdo con Christensen y Thayer [2011], la manera en que estas funciones son implementadas varía en base a la naturaleza de la actividad que se pretenda gestionar, las tecnologías disponibles para realizar y gestionar el trabajo, la industria dentro de la cual se está realizando el trabajo, el contenido específico de la actividad, los factores reguladores y el clima social.

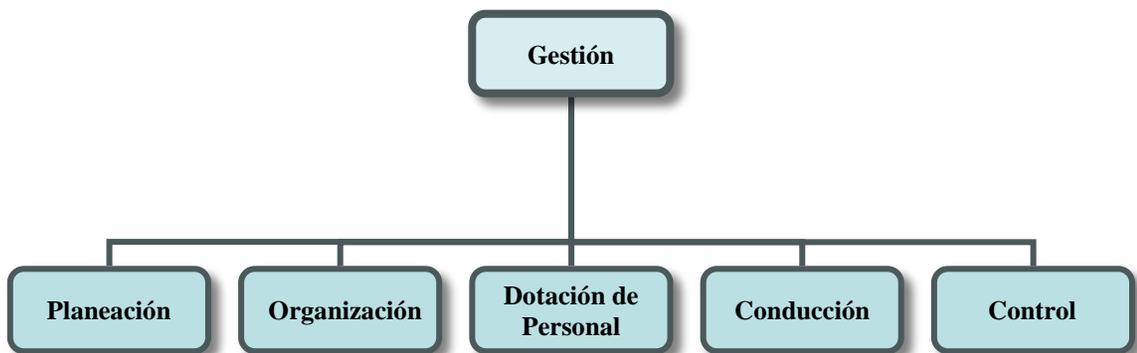


Figura 1.1. Modelo clásico de la gestión.

De acuerdo a este planteamiento, se dice que la gestión de proyectos es “*un conjunto de procedimientos, prácticas, tecnologías, habilidades y experiencias necesarias para implementar las cinco funciones principales del modelo clásico de gestión*”. En este sentido, la gestión de los proyectos de software es un sub-disciplina de la gestión de proyectos que asegura que éstos sean planificados, monitoreados y controlados [Huang y Han, 2008].

Por lo tanto, el objetivo de la gestión de los proyectos de software es el generar productos de calidad que satisfagan las necesidades del cliente en materia de tiempo y con costos aceptables [Wang et al., 2008]. Diversos estudios, como el reporte “*Chaos*” [Standish, 2009], avalado por el Standish Group, o autores como Johnson [2006], revelan que en la actualidad una gran cantidad de proyectos de software fracasan, especialmente en las pequeñas empresas, en las cuales muchos equipos de desarrollo no son capaces de generar productos de calidad en tiempo y costo.

Así, la gestión de los proyectos de software tiene por objetivo resolver los problemas del desarrollo de software desde la perspectiva de gestión, a través de la aplicación de experiencias maduras para el desarrollo de software y garantizar la implementación exitosa de los proyectos de manera más razonable, económica y eficaz [Huang y Han, 2008]. Así, la gestión de proyectos de software

incluye dos fases importantes: *la monitorización y el control*. A través de la monitorización se puede comprobar si un proyecto está siendo desarrollado conforme a la planificación y el calendario previsto. Por otro lado, mediante el control el jefe de proyecto implementa las medidas necesarias para resolver cualquier contingencia o desviación en cuanto a la calendarización y planificación de actividades [Hashim y Keshlaf, 2009].

En este contexto, de acuerdo con Humphrey [2005] los principios básicos de la gestión de proyectos son los siguientes:

- Cada proyecto cuenta con un plan basado en la jerarquía y características del mismo.
- Existe un sistema de gestión para resolver los conflictos naturales entre los proyectos y entre el personal de las organizaciones.
- Existe un sistema de revisión y supervisión que audite y de seguimiento al progreso en base al plan generado para el proyecto.

De manera similar, en [Fairley, 2009] se contemplan cuatro actividades fundamentales que se deben de llevar a cabo para una gestión exitosa de los proyectos:

- Planificación y estimación.
- Medición y control.
- Comunicación, coordinación y dirección.
- Gestión de riesgos.

Así, es importante mencionar que Humphrey [2005] y Fairley [2009] presentan las actividades que, desde su punto de vista, son necesarias para llevar a cabo una gestión exitosa de los proyectos de software, convergiendo en las actividades principales:

1. Planificar inicialmente el trabajo de una manera estructurada, organizada y lógica, como una serie de tareas bien definidas e interrelacionadas.
2. Monitorizar el estado de cómo se realizan las tareas, asegurando que los recursos estén disponibles cuando sea necesario.
3. Controlar mediante la aplicación de medidas correctivas cuando las tareas no se realizan de acuerdo a como están planificadas.

Por lo tanto, la planificación de los proyectos abarca la conversión por parte del equipo de trabajo de los requerimientos del proyecto en las tareas que debe realizar el equipo, incluidos los plazos de entrega y los recursos necesarios para llevar a cabo dichas tareas [Forsberg et al., 2005]. El plan de proyecto define el trabajo y el cómo hacerlo. Este plan proporciona una definición de las tareas más importantes del proyecto, una estimación del tiempo y recursos requeridos para cada tarea, y un marco de trabajo para la evaluación de la gestión y control del proyecto [Humphrey, 2005].

Sin embargo, llevar a cabo un plan de proyecto no es tarea fácil, ya que muchos de los proyectos software son muy complejos y requieren de un alto nivel de interrelación entre las actividades. Además de que en esta clase de proyectos existe, a menudo, la dependencia de los datos y eventos externos. Es importante mencionar también que algunos proyectos en particular cuentan con tareas que dependen o influyen en otras tareas de otros proyectos. Muchas de estas

circunstancias específicas hacen que la gestión de un proyecto de software se vuelva una actividad compleja y difícil de realizar.

Por lo tanto sin un plan de proyecto no está basado en una comprensión clara de qué es necesario, por qué es necesario, y cuándo es necesario, éste será completado en el mejor de los casos de una forma accidentada.

De acuerdo a Christensen y Thayer [2001], las actividades principales para realizar un plan de proyecto son las siguientes:

- Crear una estimación inicial sobre los costos y calendarización, necesarios para producir el producto software en base a la especificación de requerimientos.
- Construir y estructurar una lista de todas las tareas necesarias para completar el proyecto.
- Identificar las interrelaciones entre las tareas.
- Crear una red de las tareas ordenadas por tiempo, es decir la calendarización del proyecto.
- Desarrollar las estimaciones para los requerimientos de la calendarización y los recursos para cada una de las tareas.
- Asignar recursos, especialmente personas, a las tareas.
- Crear o aplicar un sistema para supervisar el desempeño de los proyectos y cómo se llevan a cabo las tareas.
- Supervisar la ejecución de las tareas.
- Aplicar las acciones correctivas cuando las cosas no se estén desarrollando de acuerdo al plan, incluyendo los posibles cambios en las tareas individuales que lleven consigo la reestructuración de toda la red de tareas.

En la actualidad, investigaciones como [Forsberg et al., 2005; Hashim y Keshlaf, 2009; Pan et al., 2007; Wang et al, 2008] han mostrado que para la obtención de resultados más favorables en el desarrollo de software, el trabajo conjunto de la gestión de proyectos y la Mejora al Proceso de Software (SPI, acrónimo en inglés para *Software Process Improvement*) es la clave del éxito. Es cierto que la gestión de proyectos marca la pauta para llevar por buen camino el desarrollo de software; sin embargo las actividades propuestas deben de someterse a una medición y a una posible mejora, ya que de esto se desprenderá la obtención de procesos de software más sólidos y maduros dentro de la empresa [Humphrey, 2005]. Es así como la gestión de los proyectos de software y SPI, mediante un trabajo conjunto, pueden brindar un marco de trabajo redituable para la empresa. El primero mediante la planificación, y monitorización y control de las tareas que se llevan a cabo dentro del ciclo de desarrollo; y el segundo mediante la medición, enriquecimiento y mejora de las actividades que se llevan a cabo durante todo el proceso de software.

En este contexto, cabe mencionar que existen herramientas que proporcionan soporte a las fases necesarias para llevar a cabo un ciclo de SPI tales como: *Stages for CMMI* [URL-11], que brinda soporte a todas las fases del ciclo SPI y reúne las funcionalidades necesarias para llevar a cabo la gestión de proyectos en una empresa que siga el modelo CMMI-DEV v1.3 [CMMI, 2010]. Sin embargo, a pesar de este soporte las empresas aún deben tener el conocimiento sobre los modelos de mejora puesto que la herramienta asume que el entorno está listo para su uso. En principio, la empresa debe comprometerse a aportar las personas, tiempo y recursos que sean necesarios para tener éxito en la ejecución del proyecto de mejora. Posteriormente se determina cuál es el estado actual del proceso software, es decir, se realiza una evaluación de la situación actual de los procesos de la empresa. Posteriormente se crea la infraestructura necesaria para la mejora del

proceso. Y por último se implementan los planes de acción, experimentando los procesos en proyectos piloto [Cuevas et al, 2005]. Un ejemplo de estos modelos es el modelo de mejora AFIM (o *Action Focused Improvement Model*), el cual brinda la pauta para propiciar la mejora en las empresas, incluidas las Micro, Pequeñas y Medianas Empresas (MPyMEs) desarrolladoras de software (véase Figura 1.2).

Por lo tanto, las fases de un ciclo de SPI comprenden dos grupos: el primero dedicado a la evaluación de la situación actual de la organización para determinar la madurez y la capacidad de los procesos actuales y establecer una iniciativa de mejora en base a las debilidades encontradas; y el segundo grupo que comprende la planificación de las actividades de mejora, la monitorización y control de éstas, y la implantación de la mejora. Estas fases deben de estar interrelacionadas, ya que cada una depende de la otra.

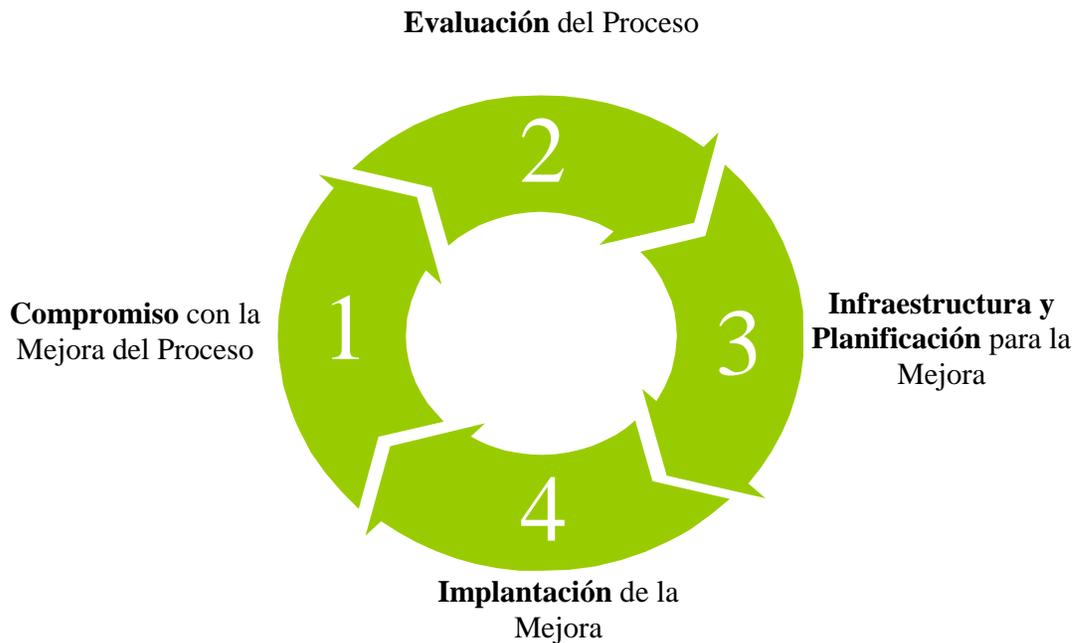


Figura 1.2. Descripción general del modelo AFIM [Cuevas et al., 2005].

En la actualidad es posible encontrar algunas herramientas computacionales que brindan el soporte para realizar estas dos fases. Por ejemplo, existen herramientas de evaluación como *CMM-Quest* [URL-12] y *Appraisal Wizard* [URL-13] que se basan en el modelo CMMI-DEV v1.3, y otras dedicadas a la gestión y planificación de proyectos como *Rommana* [URL-14] y *Microsoft Project* [URL-15]. Sin embargo, el inconveniente es que estas aplicaciones realizan estas fases de forma independiente, por lo que no se encuentran relaciones entre tareas, fases, productos y demás. Esta relación se establece mediante el trabajo humano, lo que facilita que se cometan errores. Aunado a esta problemática, se presenta el fenómeno de que son muy pocas las herramientas de este tipo que hayan sido desarrolladas bajo las características del modelo MoProSoft [NYCE, 2005]. Pero más importante aún es que en la actualidad son escasas también las herramientas comerciales que conjuntan todas las fases de un ciclo de SPI, y las pocas que existen están basadas en modelos internacionales como el CMMI-DEV v1.2; además de que tienen un costo elevado lo que las hace poco redituables para las MPyMEs desarrolladoras de software que dominan el mercado de nuestro país.

De lo anterior surge la necesidad de desarrollar una herramienta gratuita y completamente basada en MoProSoft, apegada a todas las características propias del modelo (categorías, procesos,

roles, productos y actividades) y que brinde a las MPyMEs desarrolladoras de software el soporte necesario para establecer e implantar iniciativas de SPI. Además, se pretende que esta herramienta posea no solamente las características propias de una herramienta de gestión de proyectos, sino que también controle en su totalidad un ciclo continuo de mejora dentro de la organización. Por lo tanto, el principal objetivo de este trabajo es el desarrollo de un sistema de Planificación de Recursos Empresariales (ERP, acrónimo en inglés para *Enterprise Resource Planning*) enfocado a brindar a las empresas desarrolladoras de software el soporte automatizado sobre un ciclo de SPI y la gestión de los proyectos de mejora. El sistema comprende las fases de evaluación, mejora e implantación de un ciclo de SPI y planificación, monitoreo y control de un sistema de gestión de proyectos, de esta manera se pretenda auxiliar a las MPyMEs desarrolladoras de software en la gestión de sus proyectos y reducir el costo y tiempo de adopción de un modelo de procesos, en este caso MoProSoft.

1.1. Importancia del problema y necesidad de la solución

Las pequeñas y medianas empresas que desarrollan software hecho a la medida han sido parte importante en la consolidación de la industria de software en países como India, Irlanda e Israel [Bonanomi, 2006]. En México particularmente, las MPyME representan al 87% de las empresas de acuerdo al estudio realizado por la Asociación Mexicana de la Industria de las Tecnologías de la Información (AMITI) en el 2005 [AMITI, 2005], y aunque existen empresas que desarrollan o comercializan software empaquetado, el cual representa el 80% del total de las ventas en el país (aunque casi todo es importado), el segmento de software a la medida solamente representa un 20%.

Si bien la estructura de la industria mexicana de software se encuentra en una etapa relativamente joven, para las empresas ubicadas en el segmento de “software hecho a la medida” existe la oportunidad de desarrollo económico y tecnológico, ya que como han demostrado diversos estudios [Chandler y Coartada, 2003; Hippel, 2005; Mowery, 1996], este segmento ha definido una parte importante de la evolución de la industria a nivel internacional y ha impactado a diversas actividades industriales y de servicio. Al mismo tiempo, esta estructura expone a las empresas mexicanas de software a diferentes tipos de interacción con las empresas usuarias en el diseño y desarrollo de nuevos productos.

Por lo tanto, en México la industria de software aún se encuentra en un proceso de desarrollo y crecimiento, dado que presenta un nivel de gasto en Tecnologías de la Información y Comunicación (TIC) del 1.4% respecto al Producto Interno Bruto (PIB), ubicándose en el lugar 19 a nivel mundial; un gasto pobre comparado con el 4.3% del promedio mundial de los países de la Organización para la Cooperación y el Desarrollo Económicos (OCDE), y el 5.5% de los Estados Unidos [SE, 2008]. Este rezago es aún mayor en términos de gasto en software, ya que al respecto del PIB, la industria mexicana de software representa apenas el 0.1%, esta participación es 6 veces inferior al promedio mundial y 9 veces menor que la de Estados Unidos de América. En América Latina, México ocupa el segundo lugar en todos los sectores TI en cifras absolutas, después de Brasil, aunque en el periodo 2001-2006 creció menos que este país y otros países latinoamericanos.

Durante el periodo 1992-2009, respecto al total de la industria de TI (Tecnologías de la Información) en México, la industria software tuvo una participación promedio anual de 8.0%, una participación pequeña comparada con los mayores niveles de participación de la industria de servicios de TI (23% promedio anual) y la industria de hardware (43.8% promedio anual). Si bien la participación de la industria manufacturera de software en las TIC es baja, es necesario considerar dos aspectos. Por un lado, la industria manufacturera de software solamente considera al software empaquetado y no al hecho a la medida, éste se registra en el segmento de servicios de TI. La razón

de registrarlo en este segmento se debe, en parte, a que es considerado una actividad específica a las necesidades del usuario y no se produce en grandes cantidades como el software empaquetado. En el año 2006, el software hecho a la medida alcanzó los 221 MDD [Mochi, 2007]. Por otro lado, los valores del segmento de servicios de TI son mayores al segmento de software porque registra, además del hecho a la medida, diferentes servicios de consultoría y capacitación. Así, es necesario considerar que una gran parte del segmento de servicios de TI (alrededor del 90%) se realiza en México, mientras que una gran parte del software de manufactura es importado, sobre todo las aplicaciones empaquetadas cuya importación se encuentra alrededor del 90%, principalmente a Estados Unidos [González, 2006].

La Tabla 1 muestra que del total de producción de software en México, el software empaquetado representa el 29.4% del total, el segmento del software a la medida representa tan sólo el 8.0%, mientras que el de producción y autoconsumo representa el 62.6%. Sin embargo, si la producción de autoconsumo del gobierno y las industrias se contratara a empresas independientes, la demanda de software hecho a la medida representaría alrededor del 70.6% del total de la industria. Esta demanda potencial puede ser aprovechada por las empresas de software si cuentan con las capacidades tecnológicas para ofrecer soluciones a problemas específicos del sector gobierno y de las industrias.

Tabla 1. Participación por segmento de software y sectores de demanda [Mochi, 2007].

Segmento de la Industria	Producción/Desarrollo (MDD)	Porcentaje de Participación	Sectores de demanda (por orden de importancia)
(A) Software empaquetado	817	29.4%	Servicios, Gobierno, Finanzas, Comercio y Manufactura.
(B) Software hecho a la medida	221	8.0%	Servicios (finanzas, seguros, educación, transporte, salud, cultura), Gobierno, Industria (construcción, electricidad, agua, gas, manufactura y minería), y Comercio.
(C) Software de producción y consumo interno	1,738	62.6%	Gobierno y Manufactura
Total	2,776	100%	

En este sentido, el número de empresas de software en México no es exacto, se especula que existen entre 1200 y 1800. Tampoco existen datos reales de la proporción de empresas desarrolladoras de software, ya que muchas de ellas ofrecen servicios de consultoría o comercializan productos empaquetados. Sin embargo, datos de la Secretaría de Economía en 2007, sugieren que la industria de software y TI en México se encuentra constituida por alrededor de 2,134 empresas. De acuerdo a estudios realizados, presentados en [AMITI, 2005] y [González, 2006], se determinó que la industria presenta una estructura atomística: alrededor del 87% de las empresas son micro y pequeña, 7% son mediana empresa, 5% son gran empresa, y solamente una empresa que cuenta con alrededor de 1500 empleados se considera corporativa. A su vez, los estudios mostraron que el promedio de empleados en las microempresas era de 6, mientras que en las pequeñas llegaba a 23 y en las medianas a 76. Para el caso de las empresas grandes, se estimó un valor medio de 229 empleados. En estos mismos estudios se presenta que del total de las empresas que conforman la industria, el Distrito Federal es la entidad que agrupa la mayor cantidad de ellas (453 organizaciones), y junto con los estados de Nuevo León, Jalisco y Puebla, concentran cerca del 50.8% del total de empresas. Otros Estados con importante presencia de empresas son Baja California, Veracruz y Querétaro.

Sin embargo, si las empresas nacionales quisieran incursionar en un mercado mundial cada vez más globalizado, el escenario se complica. El tamaño mínimo de entrada a un mercado de exportación es de 250 empleados y la empresa debe registrarse por los estándares internacionales de calidad como algún modelo de la familia CMMI, ISO 15504, u otro. En este sentido, la certificación en estos modelos se convierte en una necesidad para poder destacar en un mercado nacional e internacional cada vez más competitivo. Tales certificaciones deben realizarse a través de una iniciativa de SPI para generar ventajas en las tres direcciones siguientes:

- a) La iniciativa de SPI incrementa el nivel de productividad y permite la estandarización y optimización de los procesos y recursos.
- b) Los modelos de procesos son la norma para determinar la madurez de los procesos de desarrollo de software y sirven de estándar para evaluar la capacidad de las empresas.
- c) La calidad del software proporciona una mejor y más sólida posición competitiva, tanto en el mercado mundial como en el local.

No obstante, los problemas más importantes de la industria mexicana de software acentúan la limitada formación de recursos humanos especializados, la infraestructura inadecuada, la ausencia de modelos de evaluación y certificación de los procesos, entre otros. Ante ello, diversos empresarios, instituciones educativas y el Gobierno Federal plantearon en conjunto iniciativas que dieron origen al Programa para el Desarrollo de la Industria Software (PROSOFT). Paralelo a estas iniciativas, una de las estrategias del Gobierno Federal en materia de TIC es posicionar a México como un país competitivo a nivel mundial en el sector de software, estrategia que ha sistematizado a través de ese programa [Sampedro, 2011]. PROSOFT plantea los lineamientos de política industrial que deben de imperar para que la industria de software alcance los niveles de competitividad local en base al mercado interno y de competitividad internacional en base a las exportaciones. De hecho, los apoyos públicos a la industria de software se enfocan en esa dirección, principalmente a través del programa antes mencionado, en el cual se plantean dos estrategias generales: la primera que se orienta a consolidar el mercado interno mediante la formación de recursos humanos calificados, crear un marco regulatorio y promotor de la industria, mejorar la infraestructura en materia de TIC y fomentar la conversión digital de procesos en diferentes industrias, así como incentivar la demanda de software a partir de las compras del sector público; y la segunda que consiste en aumentar la capacidad de exportación [SE, 2008].

En el marco de la primera estrategia, una de las soluciones fue el plantear un modelo de procesos de software que se adaptara a las características y estructura de las empresas que conforman la industria en México. Como consecuencia, se propuso un nuevo modelo para la industria mexicana, MoProSoft (Modelo de Procesos para la Industria Software) [Oktaba, 2006], que fue desarrollado considerando las mejoras prácticas de modelos como CMMI-SW, ISO 9000:2000, PMBoK, entre otros. De acuerdo a sus creadores, el modelo MoProSoft proporciona una nueva estructura de procesos, nuevos elementos para documentar el proceso, una relación más precisa entre procesos y un mecanismo explícito para SPI (para más información sobre MoProSoft y sus características consulte la sección 2.3.1 de esta tesis). En el año 2005, MoProSoft fue adoptado como una norma mexicana bajo el nombre NMX-I-059-NYCE-2005: *Tecnología de la Información – Modelos de procesos y evaluación para desarrollo y mantenimiento de software* [NYCE, 2005], y fue avalada por el Organismo Nacional de Normalización en México conocido como Agencia Mexicana de Normalización y Certificación (NYCE). Por medio del fondo PROSOFT, el Gobierno mexicano ha proporcionado el recurso económico para la certificación de profesionales, empresas de software (organizadas por clústeres) o académicos de alguna universidad previamente registrada.

Para el año 2013, el sitio oficial del NYCE [URL-4] reportó un total de 265 certificaciones de MoProSoft (véase Figura 1.3), 6 de ellas en Nivel 0, 147 en Nivel 1, 108 en Nivel 2, y por último

4 empresas se certificaron en Nivel 3. En base a estos datos y a los estudios que establecen que el número de empresas desarrolladoras de software que componen a la industria mexicana, oscila en una media de 2,000 organizaciones; el porcentaje de empresas certificadas en el modelo MoProSoft alcanza solamente un 13.2%, lo cual demuestra que aunque los esfuerzos destinados a brindar un marco de trabajo basado en prácticas efectivas para el desarrollo de software bajo el modelo MoProSoft han avanzado y han despertado el interés de la industria, el número de empresas certificadas en dicho modelo sigue siendo muy bajo.

Empresas certificadas en MoProSoft

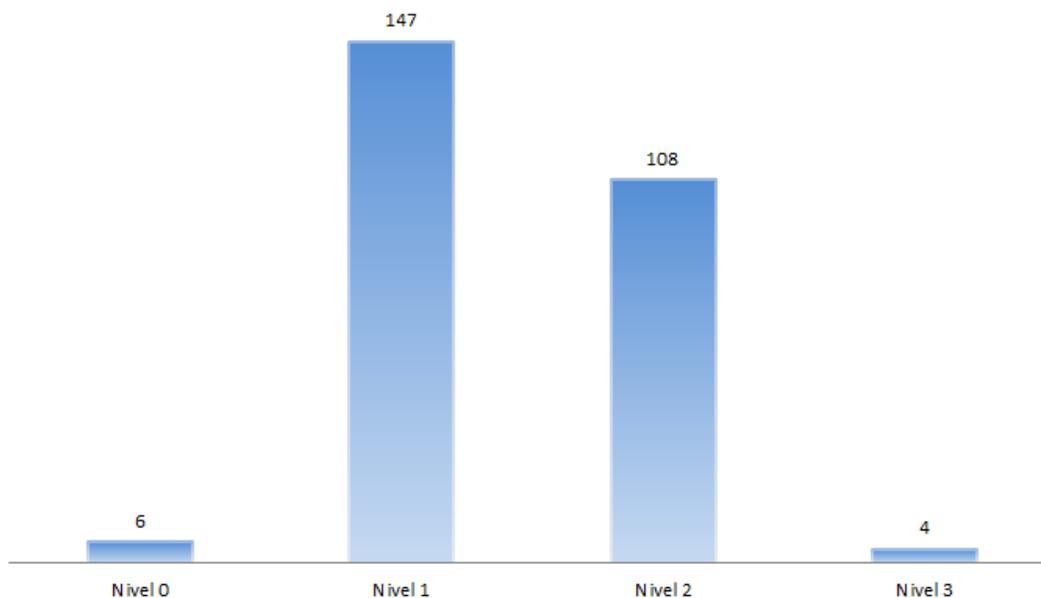


Figura 1.3. Empresas certificadas en MoProSoft por NYCE [URL-4].

Lo anterior muestra que a pesar de las iniciativas y apoyos generados, el establecimiento de un modelo de procesos formal dentro de las empresas mexicanas es aún una tarea complicada y poco explorada. La causa principal de la problemática radica en el hecho de que las empresas carecen de experiencia en la conducción de programas de mejora de procesos, aún y cuando cuenten con un modelo de referencia (MoProSoft) adaptado a las características propias de la industria mexicana. De lo anterior, se puede decir que no es posible implantar un modelo de procesos dentro de una organización, si los ingenieros y desarrolladores de software no saben cómo realizar las prácticas recomendadas en él, y además no cuentan con el suficiente conocimiento de lo que implica un proyecto de este tipo o sobre la secuencia de fases y actividades a realizar para lograr los objetivos del mismo. Si bien la solución principal radica en el hecho de brindar una mejor formación universitaria para que en un futuro próximo los ingenieros de software sean capaces de contar con el conocimiento para afrontar este tipo de retos, dicha solución se plantea a mediano/largo plazo con futuras generaciones de profesionales de TI, lo cual representa una problemática para las empresas en operación actualmente, y que buscan dentro de sus objetivos obtener un proceso de desarrollo maduro y capaz, pero no cuentan con la experiencia necesaria para implantar mejoras basadas en el modelo MoProSoft.

De la problemática anterior se desprende la necesidad de brindar un marco de trabajo que sirva como apoyo para aquellas MPyMEs desarrolladoras de software que busquen la implantación a corto/mediano plazo de un modelo de procesos (MoProSoft), con el objetivo de obtener productos de mejor calidad, así como una ventaja competitiva en un mercado cada vez más globalizado y difícil. De igual forma, es latente la necesidad de un soporte para aquellas empresas que ya han

iniciado el proceso para lograr un proceso maduro y capaz, pero debido a la misma inexperiencia carecen del conocimiento para avanzar y conseguir un nuevo nivel de madurez, como lo muestra el gran porcentaje de empresas que se concentran en el Nivel 1 de MoProSoft (véase Figura 1.3).

1.2. Delimitaciones de la tesis

Este trabajo de tesis se enfoca en desarrollar un sistema que proporcione el apoyo necesario durante el establecimiento de un ciclo de SPI, incluyendo la gestión del proyecto de mejora (planificación, monitorización y control) de forma automatizada. Esta herramienta ayuda a las MPyME desarrolladoras de software relacionadas o interesadas en la implantación del modelo MoProSoft. Lo anterior con el propósito de implantar un proceso de mejora continua en base a las mejores prácticas propuestas por el modelo de referencia, con el objetivo de reducir el costo y tiempo de la implantación de la mejora.

El sistema es desarrollado bajo un enfoque integral considerando las siguientes restricciones:

- Las empresas evaluadas son MPyMEs desarrolladoras de software.
- La herramienta se centra en evaluar empresas, generar planes de acción, planificar las actividades para la mejora, controlar y vigilar la realización de estas actividades.
- La herramienta proporciona un marco de trabajo que auxiliará a las MPyMEs desarrolladoras de software para llevar a cabo la gestión de sus proyectos de mejora.
- La solución propuesta se delimita a proporcionar el soporte para un ciclo de SPI basado en las prácticas propuestas por el modelo MoProSoft.
- No se realizan pruebas de usabilidad a usuario, puesto que se considera más importante la aprobación del contenido sistemático.

1.3. Limitaciones de la tesis

De manera similar, este trabajo está limitado de la siguiente forma:

- La investigación de esta tesis se limita a diseñar y construir un sistema que sustenta la implantación de un ciclo de SPI en base al modelo MoProSoft, así como al análisis de los resultados obtenidos.
- La herramienta desarrollada fue probada en MPyME desarrolladoras de software de los Estados de Oaxaca y Tlaxcala, las cuales cuentan con la implementación del modelo MoProSoft y conocen su estructura.
- Por otra parte se evalúa a una empresa desarrolladora de software del Estado de Oaxaca la cual no tiene implantado ningún modelo de procesos y desconoce la estructura del modelo MoProSoft.

1.4. Objetivos de la tesis

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

1.4.1. Objetivo general

Diseñar y construir una herramienta de software que brinde soporte a las MPyMEs desarrolladoras de software que desean iniciar o gestionar un ciclo de SPI de forma automatizada y considerando al modelo MoProSoft.

1.4.2. Objetivos específicos

Para alcanzar el objetivo general es necesario conseguir ciertos objetivos secundarios. Estos establecen las aportaciones esperadas al final de la tesis y se resumen como:

1. Desarrollar un estudio exploratorio acerca de la estructura del modelo de proceso MoProSoft y el modelo de evaluación de procesos EvalProSoft.
2. Desarrollar un estudio de herramientas de SPI similares orientadas al modelo MoProSoft, si es que existiesen. En caso de no existir, realizar un análisis sobre las herramientas similares que estén desarrolladas en base a otro modelo de procesos.
3. Desarrollar un mecanismo sobre los modelos de planificación, estimación, monitoreo y control necesarios para construir un sistema que cumpla con los componentes necesarios para brindar auxilio a las MPyME en el proceso de gestión de proyectos.
4. Desarrollar un mecanismo de planificación que le proporcione a la organización una guía sobre la calendarización de actividades a realizar para la implantación de la mejora, así como también delimitar qué actividades son críticas para llevar a cabo una mejora efectiva dentro de las MPyME desarrolladoras de software.
5. Desarrollar un mecanismo de control para vigilar el desarrollo de las actividades generadas en el plan de acción para la mejora del proceso. Este mecanismo debe de dar seguimiento al desempeño de cada una de las actividades a mejorar.

1.5. Solución propuesta

La propuesta de esta tesis pretende auxiliar a las MPyME desarrolladoras de software a identificar sus fortalezas y debilidades por medio de un sistema de evaluación interno, confiable y seguro que se apoya en el modelo MoProSoft y que permite identificar la situación actual del proceso de desarrollo de software en la organización. El mecanismo también permite identificar la madurez o capacidad de los procesos actuales, reconociendo actividades débiles, y mediante el análisis de estos datos generar un plan de acción. Posteriormente a la fase de la evaluación, el mecanismo debe ser capaz de planificar en base a las fortalezas y debilidades detectadas en la fase anterior. Esta fase de planificación calendariza las actividades a realizar para brindar un control sobre el desarrollo de las mismas. Dicha calendarización se basa en las propuestas del plan de acción que prioriza aquellas actividades que deben de realizarse en un lapso a corto, mediano y largo plazo. Además, esta planificación es capaz de detectar aquellas actividades de orden crítico, ya sea para alcanzar un nivel más alto (de madurez o capacidad) en la escala propuesta por MoProSoft, o bien para establecer una mejora del proceso de software de forma más efectiva. Por último, el sistema proporciona un mecanismo de control encargado de vigilar el desarrollo de las actividades propuestas para la mejora, este mecanismo es el encargado de realizar un censo de todas las actividades para verificar que se estén realizando de acuerdo a lo planeado. En caso de hallarse resultados negativos el mecanismo debe ejercer un tipo de presión hacia los roles encargados de

dichas actividades para que las lleven a cabo a la brevedad. En este sentido, el mecanismo emite alertas por medio de correos electrónicos o alertas del sistema.

La automatización del análisis hace uso de factores como:

- Áreas débiles y fuertes.
- Actividades del modelo del proceso utilizado como referencia (MoProSoft).
- Características del estado actual de la empresa, así como el personal que labora en ella.
- Necesidades y prioridades de la organización.
- Tamaño y envergadura del proyecto a realizar.

Así, se pretende que este mecanismo sirva de apoyo para que las MPyME mejoren la calidad de sus procesos y como consecuencia la calidad de sus productos a través de la recomendación de prácticas propuestas por un modelo de procesos enfocado principalmente a la estructura de este tipo de empresas. El sistema propuesto busca ser un fragmento de una de las tres partes esenciales que propone el Instituto de Ingeniería de Software (SEI, acrónimo en inglés para *Software Engineering Institute*) de la Universidad de Carnegie Mellon en [CMMI, 2006] para el establecimiento de un proceso mejorado de software:

- Personas con habilidades, entrenamiento y motivación.
- Procesos, procedimientos y métodos para definir las relaciones de cada tarea.
- Herramientas y equipo de soporte.

En base a estas partes esenciales dentro del proceso de mejora, *el personal de la empresa* (capacitado y habilitado para el desarrollo de software) es proporcionado por la misma empresa, *los procesos* son determinados por las prácticas establecidas en el modelo MoProSoft, y por último *las herramientas* de soporte que es donde se contempla al sistema propuesto en esta tesis (véase Figura 1.4).

En este sentido, la Figura 1.5 muestra que la solución propuesta en esta tesis se enfoca en administrar las actividades propuestas por el modelo MoProSoft, las cuales deben ser realizadas por el personal de la empresa. El sistema también es responsable de vigilar el desarrollo de las actividades planificadas y notificar problemas en el progreso de éstas, así como de registrar la conclusión de cada una de ellas. El modelo MoProSoft fue seleccionado puesto que se ha convertido en una norma y estándar nacional, y además está enfocado a la estructura de MPyME desarrolladoras de software mexicanas, lo que ha originado un incremento en su demanda durante los últimos años dentro de este tipo de organizaciones. Además, la falta de herramientas de este tipo (es decir, que permitan que las MPyME desarrolladoras de software lleven a cabo un ciclo de SPI completo) hace pertinente el desarrollo de este sistema. Así, el sistema propuesto establece una *fase de evaluación* que proporciona la información para generar el *plan de mejora*. Este plan debe ser calendarizado en base a las actividades sugeridas para llevar a cabo la mejora del proceso dentro de la organización; para esto, en cada actividad se sugiere el tiempo y orden con el que se deben de llevar a cabo las actividades. En esta misma fase se delegan las actividades de acuerdo a los roles de trabajo de la empresa y en base a las recomendaciones del modelo MoProSoft.

Para vigilar el desarrollo de las actividades dentro del tiempo y en la secuencia propuesta, el *mecanismo de control* asegura que las tareas se realicen en el lapso de tiempo estimado en la planificación.

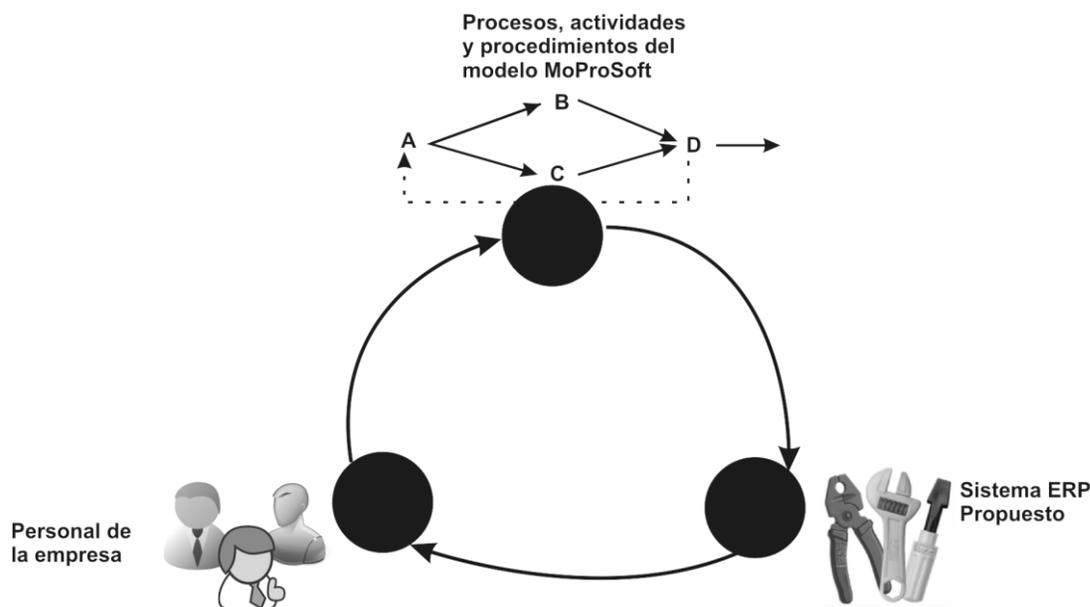


Figura 1.4. Contexto de trabajo de la solución propuesta.

Este mecanismo trabaja conjuntamente con la *fase de planificación* para vigilar que los usuarios cumplan con el desarrollo de las prácticas que le fueron asignadas. Por cada problema detectado, el sistema emite alertas que pueden ser visualizadas por todos los usuarios, ya que el retardo en una actividad puede afectar el flujo de trabajo de todo el conjunto de actividades. Aunado a esto, la herramienta envía en forma de correo electrónico alarmas al rol o roles encargados de llevar a cabo la actividad, además de especificar en dónde se hallaron las inconsistencias (véase Figura 1.5).

La *fase de evaluación* está destinada a los mandos intermedios y operativos, ya que al encontrarse inmiscuidos dentro del proceso de desarrollo de software tienen una visión más clara sobre las fortalezas y padecimientos de la empresa.

La *fase de mejora y planificación* está ligada a los altos mandos ya que ellos deben ser los encargados de implantar la mejora dentro de la empresa. Los altos mandos también son evaluados para determinar el desempeño de la dirección dentro de la empresa de acuerdo a la norma. Tal y como se ha mencionado, las actividades y procedimientos a realizar están basados en las prácticas propuestas por el modelo de procesos MoProSoft, de esta manera además de la mejora que se lleva a cabo dentro de la organización, de forma implícita se están adoptando las prácticas del modelo, por lo que la certificación en un nivel de este modelo se puede llevar a cabo de forma más fácil y eficiente.

De acuerdo a este análisis, el desarrollo de esta propuesta es altamente factible ya que en la actualidad no existen herramientas de este tipo (comerciales o académicas); dado a que el desarrollo de este tipo de herramientas ha sido poco explorado y los pocos intentos que existen han sido desarrollados en base a modelos internacionales que presentan un alto grado de dificultad para su adopción debido a las características de las MPyME mexicanas. Por último, la herramienta está disponible en formato web para que las empresas puedan acceder a ella sin ningún problema de instalación, esto facilita el acceso a las organizaciones para visualizar las recomendaciones proporcionadas por el sistema.

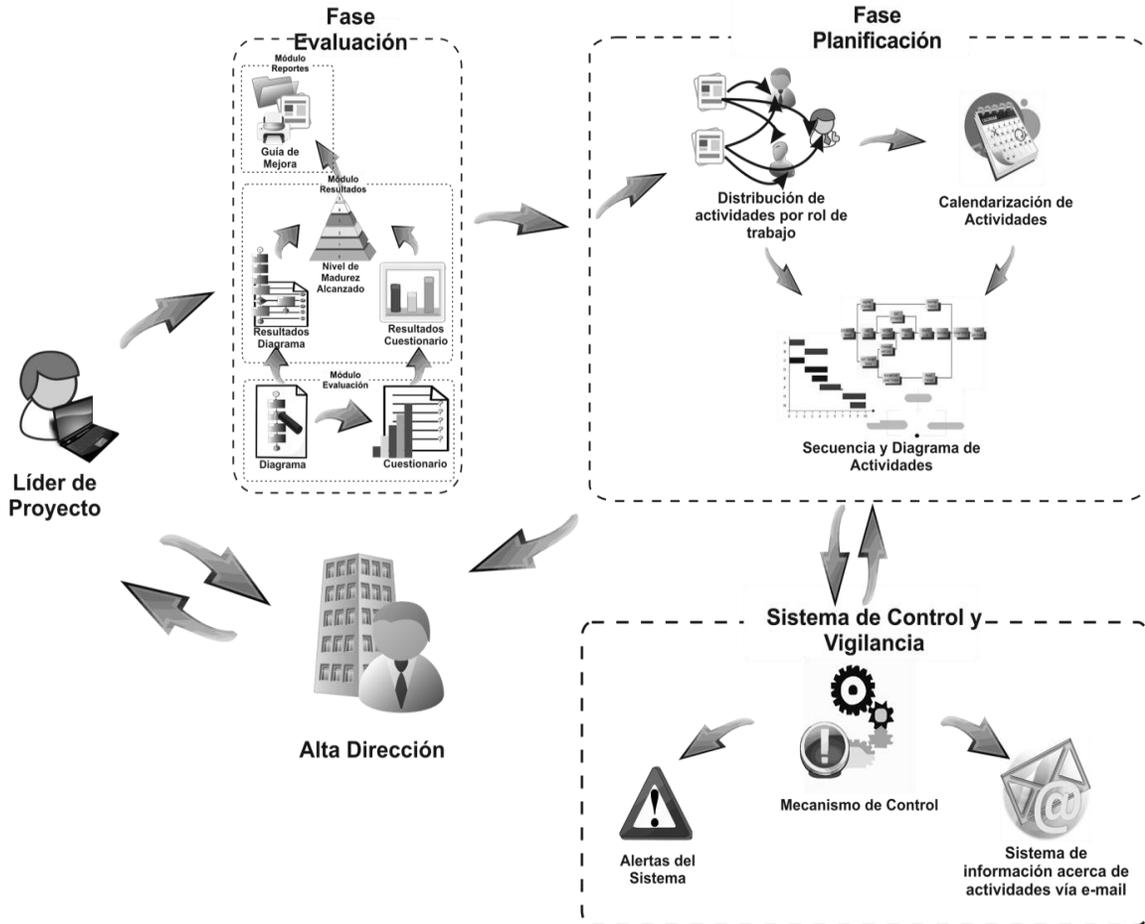


Figura 1.5 Descripción general de los objetivos específicos a alcanzar.

1.6. Estructura de la tesis

La estructura del documento de tesis se detalla a continuación:

El Capítulo 2 presenta el marco teórico sobre el modelo de procesos MoProSoft y el método de evaluación EvalProSoft, el cual sirve de base para desarrollar el mecanismo de evaluación y definición de los planes de acción. Se presenta también la revisión de literatura acerca de herramientas similares y basadas en el modelo de procesos a utilizar, y mediante un estudio comparativo se presenta por qué la solución propuesta en esta tesis es factible.

El Capítulo 3 expone los requisitos, diseño y construcción de la herramienta Kaizen (solución propuesta), utilizando los lenguajes y plataformas que se adapten a los requerimientos del sistema.

El Capítulo 4 presenta los resultados obtenidos al experimentar la herramienta Kaizen en MPyMEs desarrolladoras de software.

El Capítulo 5 presenta un resumen de las conclusiones finales obtenidas durante el desarrollo del presente trabajo y sobre el trabajo futuro que corresponde a mejorar el uso y funcionalidad de la herramienta.

El Anexo A resume una lista de acrónimos.

El Anexo B proporciona el cuestionario utilizado para las evaluaciones en base a los distintos procesos del modelo MoProSoft.

El Anexo C, muestra la especificación de los casos de uso utilizados para obtener los requisitos del sistema Kaizen.

El Anexo D, presenta una experimentación para visualizar y entender el contexto de trabajo de Kaizen.

El Anexo E ofrece una copia de las actas de las publicaciones generadas con este trabajo de tesis.

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

1.7. Publicaciones generadas

A continuación se enlistan algunas de las publicaciones que se generaron durante el desarrollo del presente trabajo.

Autores:	García, I., Pacheco, C., Cruz, D. & Calvo-Manzano, J.
Título:	Implementing the Modeling-based Approach for Supporting the Software Process Assessment in SPI initiatives inside a Small Software Company
Publicación:	Studies in Computational Intelligence Springer-Verlag. Vol.253. ISSN: 1860949X
Año:	2011
Autores:	García, I., Pacheco, C. & Cruz, D.
Título:	Implementation of a RIA Tool for Supporting a Collaborative Initiative of Software Process Improvement
Congreso:	8 th International Conference on Cooperative Design, Visualization and Engineering
Publicación:	Lecture Notes in Computer Science 5738 Springer-Verlag. ISSN: 0302-9743
Lugar:	Mallorca, España.
Año:	2011
Autores:	García, I., Pacheco, C. & Cruz, D.
Título:	Adopting a RIA-based tool for Supporting Assessment, Implementation and Learning in Software Process Improvement under the NMX-I-059/02-NYCE-2005 Standard in Small Software Enterprises
Congreso:	Eight ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2010
Publicación:	2010 IEEE Computer Society Conference Proceedings ISBN: 978-0-7695-4075-7
Lugar:	Montreal, Canadá.
Año:	2010

2. Marco Conceptual

2.1. Antecedentes

En la actualidad el uso de procesos definidos y maduros dentro de una organización para generar o brindar productos y/o servicios de software respectivamente, marca la diferencia entre ofrecer calidad o no. Considerando que cualquier actividad, o conjunto de actividades, que utiliza recursos para transformar elementos de entrada en resultados puede considerarse un proceso, se entiende el por qué de la tendencia que se enfoca a mejorar los procesos de las organizaciones. Por ejemplo, la definición del estándar ISO 9001:2000 [ISO, 2000] para proceso indica que *“es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados”*; mientras que a un producto lo define como el *“resultado de un proceso”*. Por lo tanto, sustituyendo las definiciones se puede enunciar que un producto es el *“resultado de un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman entradas en salidas”*. Sin embargo, para que las organizaciones operen de manera eficaz tienen que identificar y gestionar numerosos procesos interrelacionados y que interactúan entre ellos. A menudo el resultado de un proceso constituye directamente el elemento de entrada del siguiente proceso. La identificación y gestión sistemática de los procesos empleados en la organización y en particular las interacciones entre tales procesos se conocen como *enfoque basado en los procesos*.

Así, la idea de “nosotros no necesitamos un proceso” ha ido en decadencia dentro de las organizaciones. En los últimos años, ha sido cada vez más recurrente que la industria considere ideas como la responsabilidad y la disciplina. Es por eso, que cada vez es mayor la atención que ponen las empresas en el enfoque basado en los procesos. Una organización que utiliza procesos débiles o mal dirigidos, obtiene por consecuencia productos desarrollados de manera débil e impredecible [Persse, 2006].

La mejora del proceso se enfoca en perfeccionar la manera en la que la organización trabaja. Un proceso no puede ser mejorado a través de la adopción de nuevas cosas a ciegas, teniendo la esperanza de que éstas sean cosas que se deberían de realizar. Tampoco se refiere a introducir una nueva serie de prácticas que representan lo que la industria dice que se debe de realizar (de hecho esto es la antítesis de la mejora de los procesos). El éxito reside en tomar un enfoque de trabajo y formalizarlo dentro de un marco de trabajo que los demás puedan seguir.

Así, la mejora de los procesos se conforma de cuatro actividades básicas:

1. Centrarse en lo que se hace.
2. Enfocarse sobre las cosas que se hacen bien (o que se quieren hacer bien).
3. Un marco de actividades y herramientas para ayudar a todos a hacerlo bien, y
4. Vigilar que el enfoque mejore con el tiempo.

Cualquier entorno de producción requiere de un cierto grado de orden y secuencia, los departamentos asociados con las TI no son diferentes. El establecimiento de un proceso es una manera de ayudar en el desarrollo de la tecnología [Persse, 2006]. Existen beneficios reales en la implementación de procesos, algunos de éstos son la estabilidad operativa, la identidad organizacional, la reducción de costos y el aseguramiento de la calidad.

Así, en las últimas décadas el software ha pasado de ser una solución a problemas específicos y una herramienta para el manejo de información, a ser una industria autónoma. La industria de software, al formar parte de las TI, no se encuentra exenta del enfoque basado en procesos. De hecho, para la creación de un producto de software se emplean distintos procesos. Por lo tanto, un proceso software es un “conjunto de actividades y resultados asociados que producen un producto software; estas actividades genéricas pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de software” [Sommerville, 2010]. Además, Humphrey define al proceso software como “el conjunto de actividades, métodos y prácticas que se utilizan en la producción y evolución del software” [Humphrey, 1989]. De lo anterior se puede entender que los procesos marcan la pauta para realizar el trabajo en una empresa de software, debido a que definen el enfoque que se adopta mientras el software está en desarrollo; sin embargo, son necesarias las personas y la tecnología para producir las salidas o resultados de calidad deseados. Aunque existen muchos procesos diferentes para el desarrollo software, éste cuenta con fases genéricas que son comunes para todos ellos [Pressman, 2009]:

1. *Especificación*: Se debe definir la funcionalidad del software y las restricciones en su operación.
2. *Diseño e implementación*: Se debe producir software que cumpla con su especificación.
3. *Validación*: Se debe validar el software para asegurar que hace lo que el cliente desea.
4. *Mantenimiento*: Se centra en el cambio que va asociado con la corrección de errores debido a las mejoras producidas por los requerimientos cambiantes del cliente.

En la actualidad la industria de software ha intentado incrementar su productividad y calidad con la aplicación de nuevas metodologías y tecnologías para el desarrollo de sistemas, pero se ha reconocido que el problema fundamental es *la incapacidad de gestionar el proceso de software*. Prueba de esto se refleja en los estudios realizados por el Standish Group [Standish, 2009], específicamente el informe *Chaos Extreme*¹ que refleja como los principales problemas de los proyectos de software, que evitan el no cumplir con los requerimientos establecidos por el cliente, son los siguientes: en primer lugar el 45% de los proyectos software superan los costos establecidos para su desarrollo, el 63% los supera en materia de tiempo, y en promedio solamente se entrega un 67% de la funcionalidad total del sistema (véase Figura 2.1). De esta manera, el interés de las organizaciones ha cambiado de soluciones basadas en tecnología a soluciones basadas en el proceso de software.

¹ Desde 1994 el reporte del Standish Group, el famoso *Chaos Report*, se ha vuelto el patrón dorado sobre la calidad de los proyectos de software. Estos estudios demuestran la situación de Europa y Estados Unidos en relación al éxito y/o fracaso de los proyectos informáticos. Para más información consultar www.standishgroup.com.

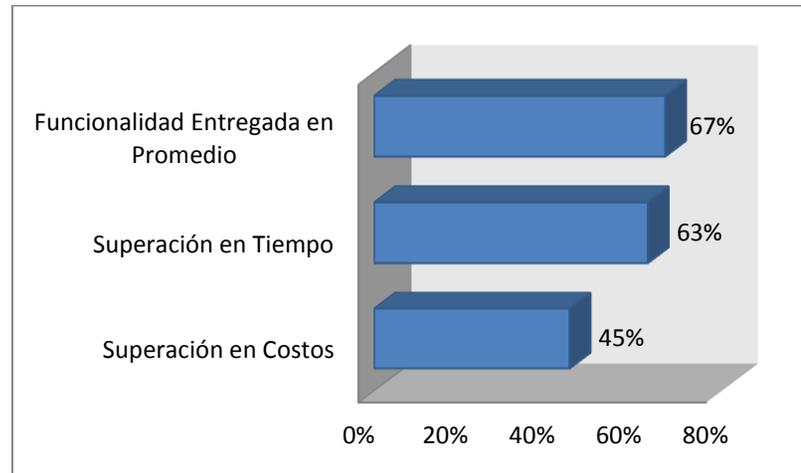


Figura 2.1. Principales causas de no cumplir con los requerimientos del cliente en proyectos software [Standish, 2009].

Sin embargo, los problemas más graves en las organizaciones desarrolladoras de software generalmente no son causados por un administrador individual o por los equipos de desarrollo; por lo general éstos se relacionan con los procedimientos de la organización y con el comportamiento cultural. Esta problemática está lejos de ser resuelta por los administradores individuales en general, por lo que se requiere un enfoque integral y de largo plazo en el proceso de software de la organización [Humphrey, 1992]. Por lo tanto, el impulso de la mejora de la calidad del software está motivado por la tecnología, las necesidades del cliente, la regulación y la competencia en el mercado.

Debido al cada vez mayor interés, que sugiere enfocarse a la mejora y gestión del proceso de software para conseguir productos de calidad dentro de las empresas, se ha observado una particular atención en SPI dentro de la Ingeniería de Software, debido a la búsqueda de competitividad por parte de las empresas desarrolladoras de software en un mercado cada vez más difícil. De acuerdo a Humphrey, este particular interés se ve motivado por la siguiente premisa “*la calidad del producto de software depende de la calidad del proceso utilizado para desarrollarlo*” [Humphrey, 1989].

2.2. Mejora al Proceso de Software

Tal y como se mencionó, en los últimos años SPI se ha convertido en un tópico importante para la Ingeniería de Software sirviendo como guía para que las compañías desarrolladoras de software afronten el reto que supone el creciente y globalizado mercado relacionado con el desarrollo y mantenimiento de software. El principal objetivo de SPI es producir software de calidad en tiempo, por debajo del presupuesto y con la funcionalidad deseada. Así, la calidad en SPI está basada en la premisa de que procesos maduros y capaces generan productos de calidad [Serrano, 2006]. La capacidad del proceso es la habilidad inherente de un proceso para producir los resultados previstos. En teoría, un proceso de software capaz se caracteriza por ser maduro. El objetivo principal de un proceso de software maduro es producir productos de calidad para satisfacer las necesidades de los clientes.

El ciclo de Deming, o también llamado PDCA (acrónimo en inglés para *Plan, Do, Check, Act*), propuesto por William Edwards Deming en 1982 [Deming, 1982], proporciona las bases para establecer una iniciativa de mejora de procesos. En este ciclo se proporciona un marco de trabajo basado en una espiral de mejora continua que se forma de cuatro componentes básicos (véase Figura 2.2):

1. *Planificar*. Incluye actividades tales como identificar el proceso que se requiere mejorar, recopilar los datos necesarios para profundizar en el conocimiento del proceso, analizar e interpretar los datos, establecer los objetivos de la mejora, detallar las especificaciones de los resultados esperados, y definir los procesos necesarios para conseguir estos objetivos, verificando las especificaciones.
2. *Hacer*. Concerniente a tareas como ejecutar los procesos definidos en el paso anterior y documentar las acciones realizadas.
3. *Verificar*. Pasado un periodo de tiempo, recopilar los datos de control y analizarlos comparándolos con los objetivos y especificaciones iniciales para evaluar si se ha producido la mejora esperada. También es el paso encargado de documentar las conclusiones.
4. *Actuar*. Es el componente encargado de modificar los procesos según las conclusiones del paso anterior para alcanzar los objetivos de las especificaciones iniciales, y si fuese necesario aplicar nuevas mejoras (y si éstas han sido detectadas en el paso anterior). Por último, se documenta el proceso.

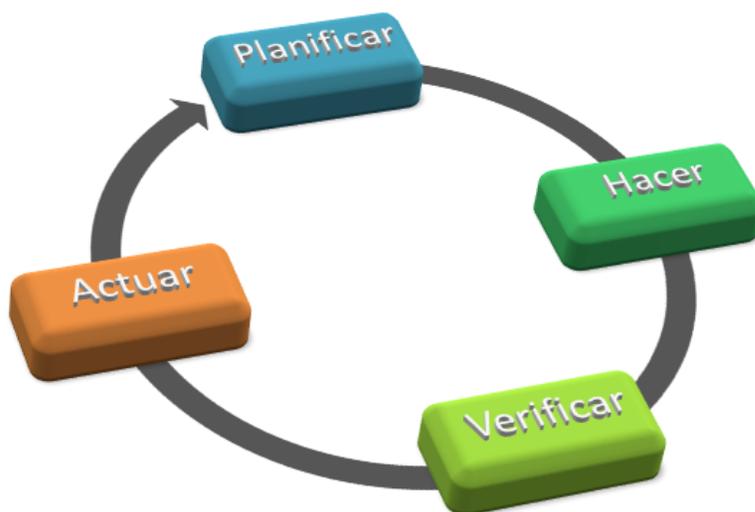


Figura 2.2. Ciclo de Deming para la mejora continua de procesos [Deming, 1982].

De acuerdo a la Figura 2.2, el ciclo comienza definiendo un plan para mejorar una determinada actividad. Una vez que el plan de mejora se ha completado, los resultados se comprueban, y se adoptan las medidas necesarias para corregir las desviaciones. El ciclo entonces se repite. Si la implementación produce los resultados deseados, las acciones se adoptan para hacer el cambio permanente. De acuerdo a la estrategia de procesos del SEI, este plan de mejora es el objetivo principal de una evaluación del proceso de software.

Con la adopción de las mejores prácticas del software se amplía la gama de ventajas y se hace mejor uso de los métodos y tecnologías actualmente disponibles en el desarrollo de software. Los modelos de mejora buscan reconocer las oportunidades de mejora y establecer un programa de mejora mediante la comprensión y el conocimiento de los costos y beneficios que conlleva su implementación. Debido a esto, en los últimos años se han desarrollado diferentes esfuerzos, iniciativas, metodologías, modelos, y normas, para apoyar e implantar proyectos de SPI dentro de las organizaciones. Ejemplo de estas iniciativas internacionales son ISO/IEC 15504 [ISO, 2004], MoProSoft [NYCE, 2005], CMMI-DEV v1.3 [CMMI, 2010], e ISO/IEC 29110 [ISO, 2011]. Además, la norma ISO 9001:2000 [ISO, 2000] ha sido utilizada para ser aplicada en el campo del

desarrollo de software. El propósito de contar con un modelo de procesos es apoyar a la industria de software en su evolución desde un estado actual (donde la calidad de los productos de software dependen principalmente de las habilidades de los individuos) a un estado deseado (donde la calidad de los productos de software es la consecuencia de la madurez de los procesos de las organizaciones). Muestra de la importancia que ha adquirido SPI dentro de la investigación de la Ingeniería de Software, es el número de publicaciones generadas que se han presentado desde [Hall, 2002] y que están relacionadas con esta línea de investigación.

De acuerdo con Pino [2006], un programa SPI involucra diferentes tipos de modelos y métodos:

- El modelo que conduce a la mejora, que describe la infraestructura, las actividades, el ciclo de vida y las consideraciones prácticas para la evolución de los procesos.
- El método de evaluación de procesos, que especifica la ejecución de la evaluación para producir un resultado cuantitativo que caracterice la capacidad del proceso o la madurez de la organización.
- El modelo de procesos de referencia, que describe las *mejores prácticas*² que una organización debe implementar para el desarrollo de software.

El *modelo de mejora* es el encargado de marcar la pauta de trabajo para llevar a cabo una iniciativa de mejora dentro de una organización. Este modelo proporciona las fases que deben desarrollarse dentro de un proyecto de SPI, dichas fases brindan una trayectoria previsible para establecer un programa de mejora; con esto se pretende dar forma al programa en las mejores condiciones posibles, de una manera ordenada y manejable. Muchos de estos modelos están basados en el ciclo PDCA, de hecho en éste se encuentra fases como Diagnosticar y Actuar, en las cuales son utilizados los otros dos componentes: los métodos de evaluación y los modelos de procesos.

Los *métodos de evaluación* se encargan de brindar un marco de referencia para realizar un diagnóstico del proceso de desarrollo actual de la empresa, obteniendo un panorama actual de los procesos utilizados. Un buen diagnóstico, ofrece una mejor oportunidad de formular una solución más efectiva para los problemas o debilidades detectados durante la evaluación. Esta fase también es conocida como Evaluación de Procesos Software (SPA, acrónimo en inglés para *Software Process Assessment*), y utiliza un método de evaluación para determinar la calidad y capacidad de los procesos actuales de la organización, por medio del hallazgo de debilidades y fortalezas en su desempeño diario. Lo anterior con el objetivo de determinar a qué áreas o procesos se deben enfocar los esfuerzos de mejora.

Descubiertas las áreas o procesos a mejorar, debe existir una guía o referencia para implantar las prácticas necesarias para solventar los problemas, y por consecuencia abatir las inconsistencias detectadas. Dicho marco de referencia lo brinda el *modelo de procesos*, el cual es un compendio de las prácticas que una organización debe de llevar a cabo para definir un proceso de desarrollo de software capaz y maduro.

En base a estos tres componentes se sustenta una parte del éxito en un proyecto de mejora, la parte restante la brindan los recursos humanos y técnicos con los que cuenta la empresa para llevar a cabo y gestionar el proyecto.

² De acuerdo a Walker [2003] “*las mejores prácticas son todas aquellas metodologías y herramientas que mejoran consistentemente la productividad y calidad de los productos de software cuando son implementadas, y al aplicarlas se tiene un beneficio directo en el proyecto*”. Por lo tanto, las mejores prácticas de software son aquellas prácticas que las personas con experiencia reconocida en el área específica tienen identificadas a través de la experiencia y esa contribución se hace significativa para el buen término del proyecto de software [Adams, 2004].

2.2.1. IDEAL: Un modelo para mejorar los procesos software

Es importante mencionar que el establecimiento de una iniciativa de SPI es una tarea complicada para cualquier organización. Por esta razón el SEI desarrolló un modelo para auxiliar a quienes realizan este tipo de iniciativas por primera vez y a aquellos que continúan con una iniciativa ya existente: el modelo IDEAL [McFeeley, 1996]. IDEAL puede ser utilizado para guiar el desarrollo de un plan integral de largo alcance e iniciar y gestionar los programas de SPI. La Figura 2.3 muestra que IDEAL establece cinco fases para una iniciativa de SPI, estas fases proporcionan un ciclo continuo a través de los pasos necesarios para alcanzar la mejora. El tiempo necesario para completar un ciclo a través del modelo IDEAL puede variar de organización a organización. De hecho, este tiempo dependerá de los recursos que la organización comprometa con la iniciativa de SPI, esto debido a que muchas actividades pueden llevarse a cabo de forma paralela, dependiendo si se cuenta con los recursos o no.

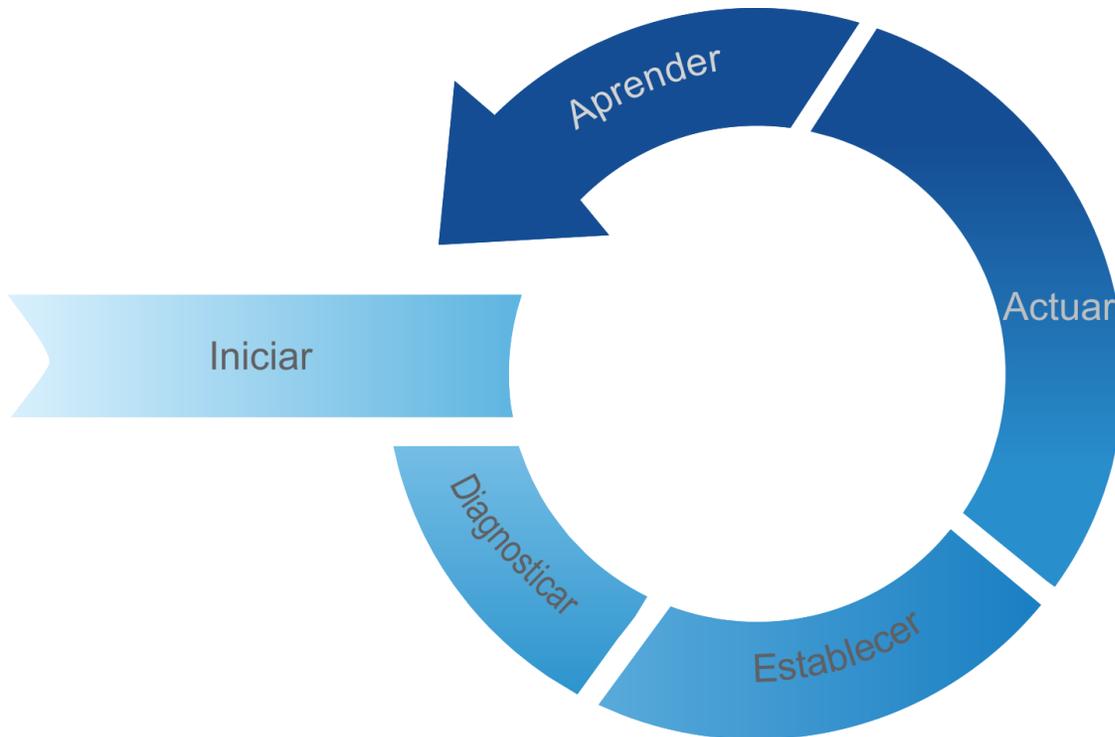


Figura 2.3. Modelo IDEAL [McFeeley, 1996].

De acuerdo a McFeeley [1996], las fases que componen al modelo IDEAL son cinco:

- *Inicio:* Esta fase es el punto de partida del modelo. Su propósito es establecer los fundamentos básicos para garantizar que la iniciativa de SPI sea exitosa. Los objetivos de mejora de la empresa u organización son definidos junto con la alta dirección. Es decir, el apoyo de la alta dirección y de la gerencia en general es básico para el éxito de la iniciativa de mejora. El compromiso de la organización con el programa de mejora garantizará la disponibilidad de recursos, la infraestructura y la priorización de la iniciativa. Las actividades de esta fase marcan una pauta importante entre el éxito o fracaso del programa.
- *Diagnóstico:* Esta fase se enfoca a evaluar mediante un método formal las fortalezas y debilidades del proceso seguido en el desarrollo de los proyectos de software. Estas actividades de evaluación se realizan para establecer una línea base del estado actual de la organización. Los objetivos del programa se relacionan con las prácticas existentes y se

determinan aquellas que no están suficientemente desarrolladas. Los resultados y recomendaciones de las evaluaciones y cualquier otra actividad planeada para los esfuerzos de mejora, son incluidos dentro de un plan de acción.

- *Establecimiento:* El propósito de esta fase se orienta a realizar la planificación específica de las mejoras que se desean alcanzar. En base a los resultados y a los objetivos que la organización se ha propuesto alcanzar se genera un plan de acción para la mejora. En base a las necesidades se establecen prioridades, objetivos y equipos de acción para los procesos. En esta fase se desarrolla un plan detallado que sirve como plan de proyecto de mejora. Una de las partes importantes de esta fase es establecer la estrategia y prioridades; debido al alto costo que implica dar solución a todas las debilidades en un solo ciclo de mejora, se determina dónde es más productivo enfocarse. Dichas acciones son elegibles a razón de los recursos, las necesidades urgentes, el impacto, entre otros aspectos.
- *Ejecución:* En esta fase se implementan las soluciones diseñadas previamente, por lo que su principal objetivo es implementar la mejora de los procesos llevando a cabo el plan de acción. Es normal encontrar actividades como la introducción o mejora de los procesos, la capacitación a los respectivos niveles de personal, la medición de los avances y de los beneficios logrados, la realización de proyectos piloto, la implantación de los procesos mejorados en los nuevos proyectos o en los ya existentes, la realización de mini-evaluaciones para constatar la evolución del plan.
- *Adopción:* En esta fase se analizan los resultados de la implementación, se aprende del ciclo de mejora recién realizado y se aumenta la habilidad de la empresa u organización para mejorar los procesos de forma continua. Además de que se determinan los logros, el esfuerzo invertido, la manera en que las metas fueron satisfechas y la forma más adecuada de implementar cambios en el futuro. También se utilizan las mediciones y registros acumulados durante la aplicación de las etapas anteriores al ciclo.

Las fases que componen al modelo IDEAL brindan un marco de referencia para las nuevas iniciativas de mejora emprendidas en una empresa u organización, o enriquecen a aquellas que ya han sido puestas en marcha. Debido a lo anterior, tomar como base las fases y actividades que proporcionan un modelo de mejora como IDEAL, facilita el trazar el camino para la puesta en marcha de una iniciativa de mejora en los procesos de una organización desarrolladora de software.

2.3. Modelos de procesos y evaluación para el desarrollo y mantenimiento de software para la industria de software en México

A través de los años el gobierno de México ha comprendido la importancia del uso de la tecnología y la informática para el desarrollo del país, así lo expresa el Plan Nacional de Desarrollo (PND) 2006-2012 [SE, 2006]: “...cada día convergen nuevas tecnologías, servicios y contenidos, que ofrecen oportunidades hasta hace poco inimaginables. Éste es el cuarto motor de la globalización, este nuevo entorno en el que convergen tecnologías de gran capacidad y cobertura de diversos servicios sirve como punto de referencia para lograr el salto cualitativo y cuantitativo como nación, también permitirá aprovechar las oportunidades del avance tecnológico y la convergencia para superar los problemas como país”. Uno de los principales objetivos que plantea el PND es elevar y extender la competitividad del país mediante la estrategia de promover el uso y aprovechamiento de la tecnología y de la información. Dicho objetivo se basa en la promoción de las acciones para utilizar y aprovechar las tecnologías como recursos estratégicos que contribuyan a la satisfacción de las necesidades de la sociedad mexicana y adoptar los mejores estándares

tecnológicos [URL-1]. El apoyo para el desarrollo de la industria de software mexicana se ve reflejado en uno de los puntos del plan, denominado “Impulso al desarrollo de la industria de TI”.

De lo anterior, se desprende la cooperación entre la Secretaría de Economía (SE) y empresas del sector y organismos empresariales para diseñar PROSOFT como uno de los medios para conseguir los objetivos planteados. Así, el objetivo de PROSOFT es impulsar a la industria de software y extender el mercado de TI en México. Para alcanzar el objetivo principal del programa, la SE en conjunto con la industria y las organizaciones gubernamentales relacionadas con el sector, acordaron desarrollar siete estrategias [URL-2] (véase Figura 2.4).

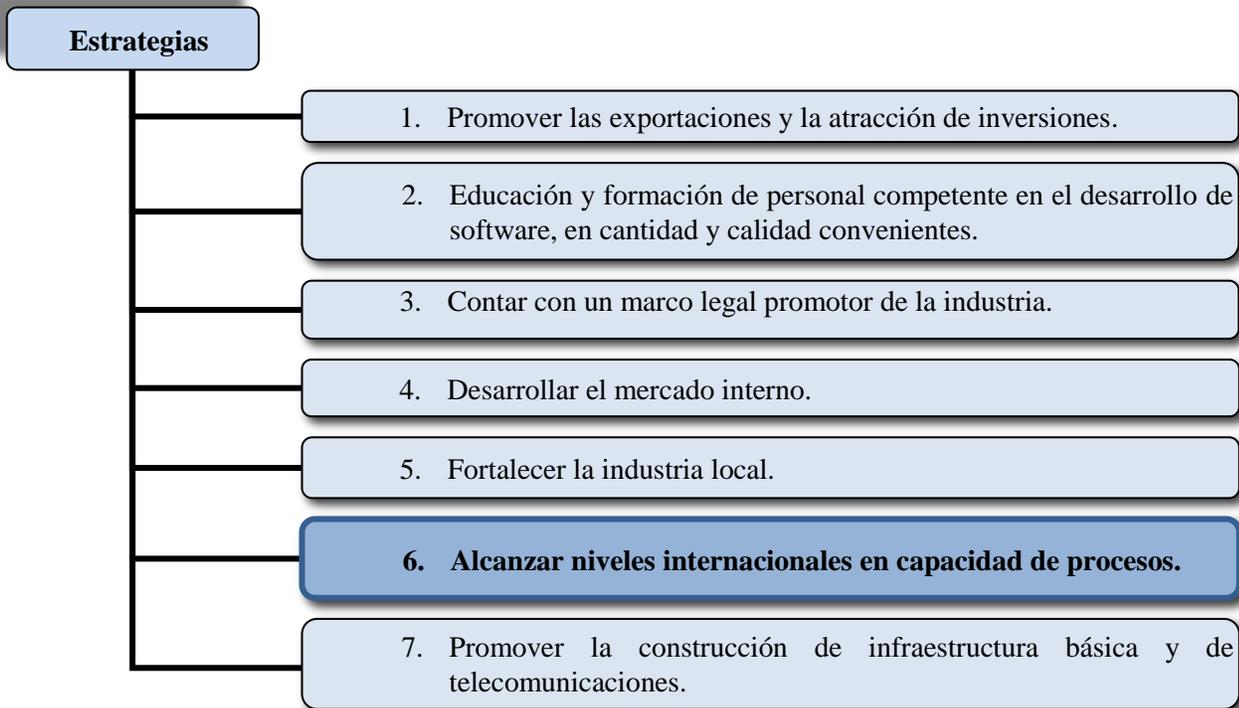


Figura 2.4. Estrategias de PROSOFT [SE, 2006].

De acuerdo a PROSOFT, a partir de la estrategia número seis se desglosan siete puntos para lograr su cometido:

1. Definición de un modelo de procesos y de evaluación apropiado para industria de software mexicana.
2. Formación de instituciones de capacitación y asesoría en mejora de procesos.
3. Apoyo financiero para la capacitación y la evaluación de la capacidad de los procesos.
4. Premio nacional de calidad TI.
5. Estímulos fiscales al desarrollo tecnológico en las empresas.
6. Formación de un cajón de financiamiento para actividades de investigación y desarrollo.
7. Otros apoyos para actividades de investigación y desarrollo.

Es fácil observar que de estos siete puntos para llevar a cabo la estrategia, el primer punto está íntimamente ligado con los procesos de mejora: “Definición de un modelo de procesos y de evaluación apropiado para la industria de software mexicana”.

Sin embargo, el 92% de las empresas dedicadas al desarrollo de software en México se caracteriza porque su tamaño es pequeño y mediano, es decir, con una plantilla menor a cien empleados [URL-3]. A través de un estudio realizado por Hanna Oktaba [Oktaba, 2006], sobre el nivel de las empresas mexicanas de desarrollo de software, se determinó que el nivel de capacidad de los procesos promediaba un 0.9 en base a la escala del 0 a 5 del ISO/IEC 15504. Por lo que la selección de un modelo de procesos de referencia y un método de evaluación accesibles para la industria local, fue el primer problema a resolver. Así, el gobierno y la industria de software definieron criterios de selección, además de considerar el consenso con algunas empresas sobre las necesidades al respecto de un modelo de procesos y su evaluación (lo que se tradujo en “algo fácil de entender, práctico y barato”). Asimismo, la SE requería que se tratara de un modelo que se pudiera adoptar como norma mexicana [Oktaba, 2005]. A partir de estas premisas, se evaluó la “conveniencia de adopción” de los estándares más populares y modelos de referencia, tales como: SW-CMM [Paulk, 1995], CMMI [SEI, 2002], ISO/IEC 12207 [ISO, 1995], ISO 9000:2000 e ISO/IEC 15504.

El resultado de esta evaluación determinó que de acuerdo a Oktaba: “*Ninguno de los modelos propuestos cumplía con todos los criterios de selección. Los altos costos de SW-CMM y CMMI y la necesidad de una norma nacional, fueron las razones principales para el desarrollo de un nuevo modelo de procesos software*”. El nombre que se le dio a este nuevo modelo fue MoProSoft (Modelo de Procesos para la Industria Software), el cual fue desarrollado a partir de las mejores prácticas del SW-CMM, ISO 9000:2000, PMBok [PMI, 2004], y otros.

MoProSoft ofrece una nueva estructura de procesos, algunos nuevos elementos sobre la documentación de los procesos, una relación más precisa entre los procesos, y un mecanismo explícito para la mejora de los mismos [Oktaba, 2006; Oktaba, 2007]. El nuevo modelo de procesos fue complementado por el método de Evaluación de Procesos para la Industria Software (EvalProSoft), el cual fue creado en base a las recomendaciones del ISO/IEC 15504. A mediados del año 2005, MoProSoft y EvalProSoft fueron aprobados como parte de una norma nacional mexicana, el nombre oficial de la norma es NMX-I-059-NYCE-2005: *Tecnología de la Información – Modelos de procesos y evaluación para desarrollo y mantenimiento de software* [NYCE, 2005]. La norma consiste de cuatro fascículos (véase Tabla 2) [Flores, 2008]:

Tabla 2. Estructura de la norma NMX-I-059-NYCE-2005 [NYCE, 2005].

Fascículo	Descripción
NMX-I-059/01-NYCE-2005	Definición de conceptos y productos
NMX-I-059/02-NYCE-2005	Requerimientos de procesos (MoProSoft)
NMX-I-059/03-NYCE-2005	Guía de implementación de procesos
NMX-I-059/04-NYCE-2005	Directrices para la evaluación de procesos (EvalProSoft)

Los fascículos dos y cuatro hacen referencia explícita a MoProSoft y EvalProSoft respectivamente, los dos restantes contienen conceptos y guías para entender y aplicar los modelos en las organizaciones.

2.3.1. Modelo de Procesos para la Industria Software (MoProSoft)

El principal objetivo de MoProSoft en México es fomentar la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e Ingeniería de Software. La adopción del modelo permitiría elevar la capacidad de las organizaciones para ofrecer servicios de calidad y alcanzar niveles internacionales de competitividad [Oktaba, 2005a].

Debido a la estructura de las empresas mexicanas, que en su gran mayoría son MPyMEs, MoProSoft se basa en las mejores prácticas internacionales que cumplen con las siguientes características:

- Fácil de entender.
- Fácil de aplicar.
- No costoso en su adopción.
- Ser la base para alcanzar evaluaciones exitosas con otros modelos o normas, tales como ISO 9000:2000 o CMMI V1.2.

El modelo está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software [Oktaba, 2006]. De acuerdo con [Oktaba, 2005a], el modelo está enfocado a las organizaciones que no cuentan con procesos establecidos, ya que pueden usar el modelo ajustándolo de acuerdo a sus necesidades; mientras que las organizaciones que ya tienen procesos establecidos, pueden usarlo como punto de referencia para identificar los puntos que les hace falta cubrir. Así, MoProSoft, a diferencia de CMM-SW y CMMI, está dirigido a la micro y pequeña industria de software dado que sintetiza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades de la alta dirección, gestión y operación [IIE, 2003].

2.3.1.1. *Estructura*

Con el objetivo de definir la estructura del modelo de procesos, en primera instancia se analizó la estructura de las empresas locales de desarrollo de software. En la mayoría de las empresas, incluso las llamadas microempresas (que tienen menos de 10 empleados), se cuenta con una *alta dirección* que es la encargada de tomar las decisiones que marcan el rumbo del negocio, los *mandos intermedios o de gestión* que son los responsables de los proyectos y la obtención de los recursos y el control de los mismos, y por último un *grupo operativo* encargado del desarrollo de los proyectos utilizando los recursos asignados. Los miembros de estos grupos conocen sus responsabilidades a través de la asignación de roles, las líneas de autoridad (regularmente de tipo vertical) y las relaciones de colaboración (regularmente de tipo horizontal) [Oktaba, 2007]. Por lo tanto, MoProSoft se enfoca en procesos y considera los tres niveles básicos de la estructura de una organización que son: Alta Dirección, Gestión y Operación (véase Figura 2.5). Así, el modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua [NYCE, 2005].

2.3.1.1.1. Categorías de procesos

La Categoría de Alta Dirección (DIR) contiene el proceso de *Gestión de Negocios*. Esta categoría de procesos 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 éstos.

La Categoría de Gerencia (GES) está integrada por los procesos de *Gestión de Procesos*, *Gestión de Proyectos* y *Gestión de Recursos*. Éste último está constituido por los subprocesos de *Recursos Humanos* y *Ambiente de Trabajo*, *Bienes e Infraestructura*, y *Conocimiento de la Organización*. Esta categoría es la encargada de abordar las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Además, 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.

La Categoría de Operación (OPE) está constituida por los procesos de *Administración de Proyectos Específicos* y *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.

Cabe mencionar que para cada proceso se definen los roles responsables para la ejecución de las prácticas que éstos incluyen. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y capacitación para desempeñarlos.

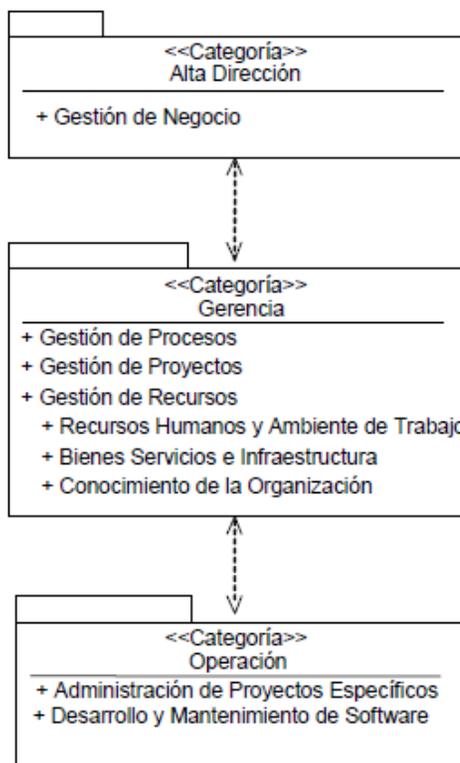


Figura 2.5. Estructura de MoProSoft [NYCE, 2005].

2.3.1.1.2. Procesos

Los procesos son definidos como un “conjunto de prácticas relacionadas entre sí, llevadas a cabo a través de roles y por elementos automatizados, que utilizando recursos y a partir de insumos producen un satisfactor de negocio para el cliente”. Así, MoProSoft está constituido por un total de seis procesos y tres subprocesos que se pueden resumir de la siguiente forma:

DIR.1 Gestión de Negocio

El propósito de la *Gestión de Negocio* es establecer la razón de ser de la organización, sus objetivos y las condiciones para alcanzarlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados y proponer cambios que permitan la mejora continua.

GES.1 Gestión de Procesos

El propósito de la *Gestión de Procesos* es establecer los procesos de la organización en función de los procesos identificados en el plan estratégico. Así como definir, planificar, e implementar las actividades de mejora de los mismos.

GES.2 Gestión de Proyectos	El propósito de la <i>Gestión de Proyectos</i> es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.
GES.3 Gestión de Recursos	El propósito de la <i>Gestión de Recursos</i> es conseguir y dotar a la organización de los recursos humanos, la infraestructura, el ambiente de trabajo y los proveedores, así como crear y mantener la base de conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del plan estratégico de la organización.
GES.3.1 Recursos Humanos y Ambiente de Trabajo	El propósito de <i>Recursos Humanos y Ambiente de Trabajo</i> es proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.
GES.3.2 Bienes Servicios e Infraestructura	El propósito de <i>Bienes, Servicios e Infraestructura</i> es proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requerimientos de adquisición de los procesos y proyectos.
GES.3.3 Conocimiento de la Organización	El propósito de <i>Conocimiento de la Organización</i> es mantener disponible y administrar la base de conocimiento que contiene la información y los productos generados por la organización.
OPE.1 Administración de Proyectos Específicos	El propósito de la <i>Administración de Proyectos Específicos</i> es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costos esperados.
OPE.2 Desarrollo y Mantenimiento de Software	El propósito de <i>Desarrollo y Mantenimiento de Software</i> es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software (nuevos o modificados) cumpliendo con los requerimientos especificados.

La relación entre procesos está basada en el intercambio de productos y el nivel de participación en cada uno de los procesos (véase Figura 2.6). Cada uno de los productos de salida generados por algún proceso, es identificado explícitamente como la entrada de uno o más procesos. Los productos internos son utilizados por el mismo proceso que lo ha generado. La relación de procesos en base a la participación de roles significa que alguno de los roles de un proceso participa en las actividades de otros procesos.

2.3.1.1.3. Patrón de procesos y uso del modelo

El patrón de procesos es un esquema de elementos que sirve para la documentación de los procesos. Este patrón está constituido por tres partes: Definición general del proceso, Prácticas, y Guías de Ajuste.

En la *Definición general del proceso* se identifica su nombre, categoría a la que pertenece, propósito, descripción general de sus actividades, objetivos, indicadores, metas cuantitativas,

responsabilidad y autoridad, subprocessos (en caso de tenerlos), procesos relacionados, entradas, salidas, productos internos y referencias bibliográficas.

Las *Prácticas* identifican los roles involucrados en el proceso y la capacitación requerida, se describen las actividades a detalle, asociándolas a los objetivos del proceso; además 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.

En las *Guías de ajuste* se sugieren modificaciones al proceso que no deben de afectar los objetivos del mismo.

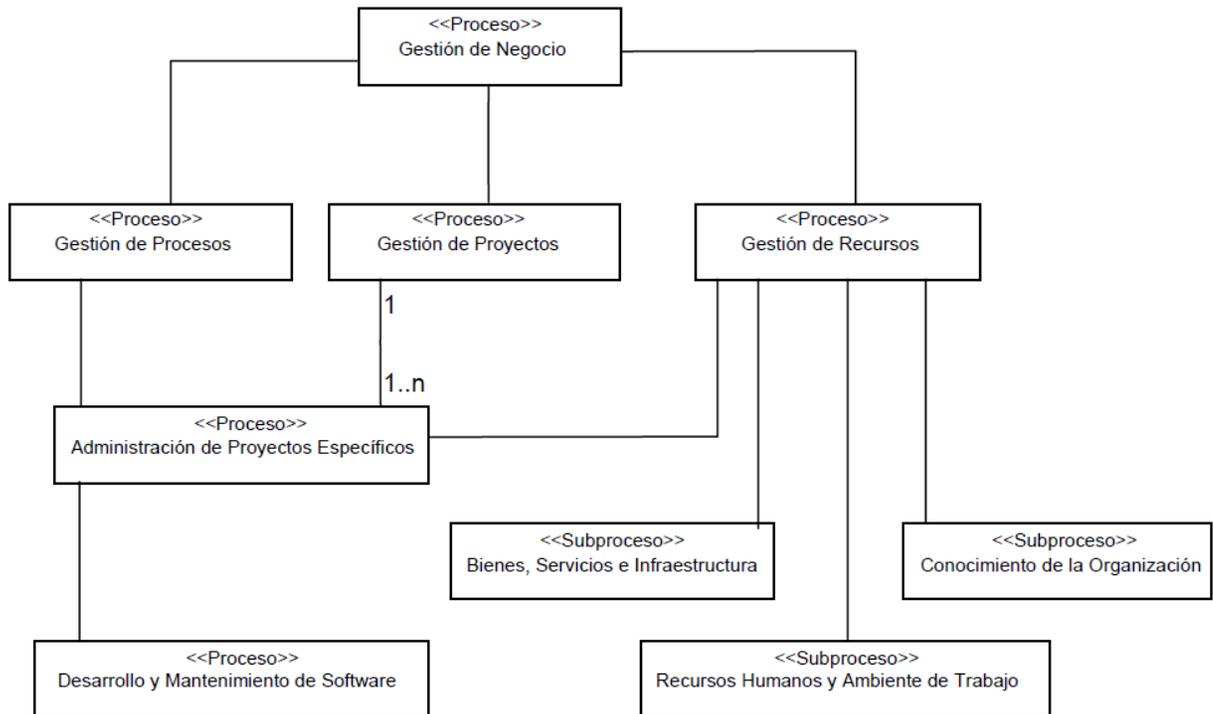


Figura 2.6. Diagrama de relación de procesos [NYCE, 2005].

Así, el patrón de proceso puede ser utilizado para documentar e integrar otros procesos que no fueron contemplados en el modelo. Para usar MoProSoft en una organización que no cuenta con procesos establecidos ni documentados es necesario generar una instancia de cada uno de los procesos, tomando en cuenta las siguientes consideraciones:

- Definir las metas cuantitativas de acuerdo a las estrategias de la organización.
- Revisar los nombres de los roles y productos (entradas, salidas o internos) y en su caso sustituirlos por los que se acostumbran en la organización.
- Para cada producto definir el estándar de documentación cumpliendo con las características mencionadas en la descripción del producto.
- Definir los recursos de infraestructura de cada proceso.
- Analizar si las mediciones de cada proceso son aplicables dentro del contexto de organización o, en su caso, modificarlas.

- Usar las guías de ajuste para adecuar el proceso en función de las estrategias de la organización.
- Posteriormente sustituir las guías de ajuste del modelo por las guías que apliquen en la organización.

Adicionalmente, para el proceso de Desarrollo y Mantenimiento de Software, se requiere:

- Definir métodos, técnicas o procedimientos específicos para las actividades, tareas, verificaciones y validaciones.

Por último, para usar este modelo en una organización con procesos establecidos o documentados, se debe establecer la correspondencia entre estos procesos y el modelo MoProSoft, para identificar las coincidencias y discrepancias. La organización debe de analizar las discrepancias y planificar las actividades de ajuste de los procesos para lograr la cobertura completa de MoProSoft. La organización debe establecer la estrategia de implantación de los procesos definidos y puede decidir probarlos en proyectos piloto o implantarlos al mismo tiempo en toda la organización. Con el transcurso del tiempo, los procesos deben evolucionar con base a las sugerencias de mejora e ir alcanzando los objetivos del plan estratégico de la organización con metas cuantitativas cada vez más ambiciosas. De esta manera la organización puede ir logrando la madurez a través de la mejora continua de procesos [Oktaba, 2006].

2.3.2. Método de Evaluación de Procesos para la Industria Software (EvalProSoft)

EvalProSoft se enfoca en las organizaciones dedicadas al desarrollo y/o mantenimiento de software. En particular a aquellas que han utilizado como modelo de procesos de referencia a MoProSoft para la implantación de sus procesos [Oktaba, 2005b]. EvalProSoft se utiliza para proporcionar a la organización, un perfil del nivel de capacidad sobre sus procesos implementados y, de forma general, el nivel de madurez de la organización [Flores, 2008]. El método de evaluación se encuentra incorporado a la norma como *NMX-I-059-NYCE-2005: Tecnología de la Información – Modelos de procesos y evaluación para desarrollo y mantenimiento de software*.

El método es una guía para establecer los requerimientos tanto para modelos de procesos de referencia como para los métodos de evaluación, sin establecer alguno en particular. EvalProSoft es parte de la norma NMX-I-059-NYCE-2005, específicamente del cuarto fascículo que es donde se presenta el método de evaluación [NYCE, 2005a]. Así, EvalProSoft es parte medular de la norma ya que proporciona el método de evaluación para determinar el nivel de capacidad de los 9 procesos implementados en cualquier organización. El proceso de evaluación considera las condiciones para iniciar una evaluación, además de las actividades de planeación, de ejecución, de generación y entrega de resultados, y de cierre. En este proceso se involucran roles con responsabilidades específicas, de esta forma el rol que dirige la evaluación es el evaluador certificado, que cumple con un perfil definido y cuenta con una acreditación de su competencia. La Figura 2.7 muestra la forma en que el método de evaluación involucra al organismo rector y a la organización a evaluar de forma que se establezca un vínculo sólido de trabajo.

De acuerdo a la Figura 2.7 la organización selecciona un evaluador certificado reconocido por el organismo rector. El evaluador certificado dirige el proceso de evaluación en función de los datos de la organización, apoyándose en el equipo de evaluación y en el paquete de evaluación. Como consecuencia, del proceso de evaluación se obtiene un reporte de resultados para la organización y un reporte estadístico para el organismo rector [Flores, 2008]. En *el reporte de resultados* se documenta el perfil del nivel de capacidad de los procesos y un nivel de madurez de la organización, así como el resumen de los hallazgos detectados. En *el reporte estadístico* se

proporciona la información general de la organización evaluada, los resultados de la evaluación y las lecciones aprendidas sobre el método de evaluación y su modelo de procesos de referencia, MoProSoft.

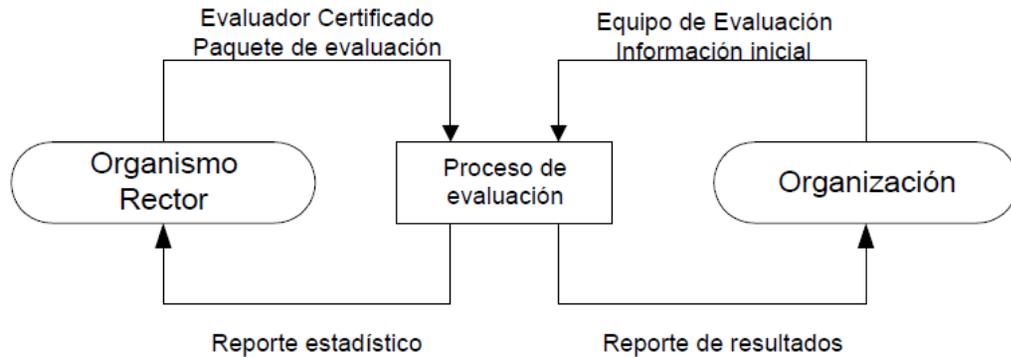


Figura 2.7. Relación entre los elementos del método de evaluación.

2.3.2.1. Usos del método de evaluación

La Tabla 3 resume los posibles usos del método de evaluación:

Tabla 3. Usos del método de evaluación [NYCE, 2005a].

Nombre	Descripción
Evaluación para la acreditación de capacidades	Cuando una organización solicita a un Evaluador Certificado la realización de la evaluación, ya sea para obtener un perfil de nivel de capacidad de los procesos implantados o determinar un nivel de madurez general.
Evaluación de capacidades del proveedor	Cuando un cliente solicita a un Evaluador Certificado la realización de una evaluación para obtener un perfil de nivel de capacidad sobre los procesos implantados por el proveedor de desarrollo y mantenimiento de software. El cliente elige los procesos a evaluar dependiendo del servicio a contratar.
Auto-evaluación de capacidades de proceso	Cuando una organización realiza una evaluación por personal interno o externo que no necesariamente sea un Evaluador Certificado. En este caso no interviene el Organismo Rector.

La evaluación para la acreditación de capacidades es útil cuando la organización pretende obtener un estado certificado del perfil de nivel de capacidad por cada proceso implantado, el cual puede usarse como base para la elaboración del plan de mejora. De acuerdo con Oktaba [Oktaba, 2005b], es necesario hacer énfasis en que uno de los posibles usos del método de evaluación se enfoca a determinar las oportunidades de mejora con respecto al modelo MoProSoft, a fin de optimizar los resultados de la organización de software, procurando mejorar la productividad y la calidad de sus servicios y/o productos para aumentar la competitividad de la industria nacional, de otra forma se mantendría únicamente el interés en la certificación sin orientación a la mejora de los resultados. El nivel de madurez de la organización puede usarse como comparativo con respecto a otras organizaciones en el mercado. El reporte estadístico de la evaluación para la acreditación de capacidades permite que el organismo rector elabore un diagnóstico de las capacidades de la industria de software en México.

La evaluación de capacidades también es útil para que un cliente seleccione a un proveedor de servicios y/o productos. Mientras que la auto-evaluación de capacidades, le permite a la organización obtener un perfil sobre el nivel de capacidad por cada proceso. Esto también puede ser la base para elaborar el plan de mejora de la organización.

Elementos como EvalProSoft, MoProSoft y el modelo de capacidades, son elementos fundamentales para llevar a cabo el proceso de evaluación en una organización. La relación de estos elementos se basa en la evaluación sobre el modelo de procesos de referencia MoProSoft, usando el método de evaluación (EvalProSoft), y tomando como referencia el modelo de capacidades (véase Figura 2.8) [Oktaba, 2005b].

2.3.2.2. Modelo de capacidades de procesos

La capacidad de los procesos es evaluada en una escala de 0 a 5. El valor cero se asocia al nivel de capacidad más bajo y significa que no se alcanza el propósito del proceso. El valor 5 se asocia al nivel de capacidad más alto y significa que se logran las metas de negocios actuales, y proyectadas a través de la optimización y mejora continua del proceso.

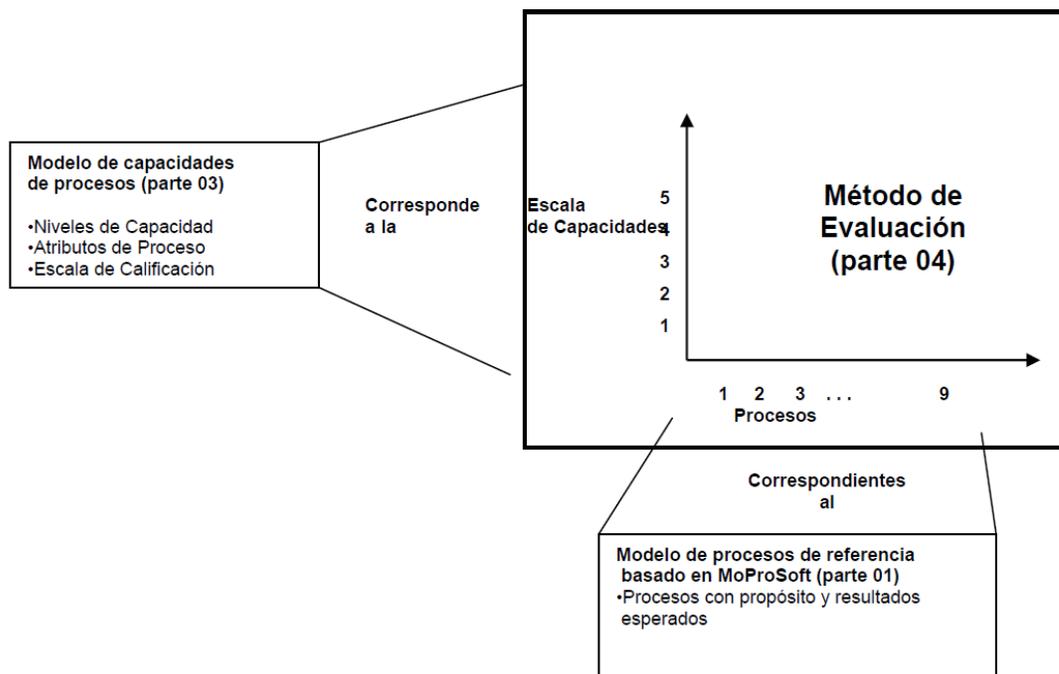


Figura 2.8. Relación de elementos del proceso de evaluación [NYCE, 2005a].

La medición de la capacidad se obtiene a través de un conjunto de Atributos de Procesos (AP), los cuales se usan para determinar cuándo un proceso ha alcanzado una capacidad. Cada atributo mide un aspecto particular de un proceso. La Tabla 4 muestra los niveles que se utilizan para determinar si un proceso ha alcanzado cierto nivel de capacidad [García, 2010; Oktaba, 2005b].

Tabla 4. Niveles de capacidad y atributos de proceso de MoProSoft [Oktaba, 2005b].

Nivel	Descripción	Atributos de Procesos (AP)
Nivel 0 Proceso Incompleto	El proceso no está implantado o falla al alcanzar el propósito del proceso.	--
Nivel 1 Proceso Realizado	El proceso implantado logra su propósito y obtiene los resultados definidos.	1. Atributo de realización del proceso.
Nivel 2 Proceso Administrado	El proceso Realizado se implanta de manera administrada y sus productos de trabajo están apropiadamente establecidos, controlados y	1. Atributo de administración de realización. 2. Atributo de administración del

Nivel	Descripción	Atributos de Procesos (AP)
	mantenidos.	producto de trabajo.
Nivel 3 Proceso Establecido	El proceso Administrado es implantado mediante el proceso definido, el cual es capaz de lograr los resultados del proceso.	1. Atributo de definición del proceso. 2. Atributo de implantación del proceso.
Nivel 4 Proceso Predecible	El proceso Establecido opera dentro de los límites para lograr sus resultados.	1. Atributo de medición del proceso. 2. Atributo de control del proceso.
Nivel 5 Optimizando el Proceso	El proceso Predecible es continuamente mejorado para lograr las metas de negocios actuales y futuras relevantes.	1. Atributo de innovación del proceso. 2. Atributo de optimización del proceso.

En la Tabla 5 se muestra el grado de cumplimiento del atributo del proceso, el cual se califica usando una escala ordinal.

Tabla 5. Grado de cumplimiento de un atributo [NYCE, 2005a].

Calificación	Grado de Cumplimiento	Valor
N	No alcanzado	0 hasta 15% del alcance
P	Parcialmente alcanzado	> 15% hasta 50% del alcance
A	Ampliamente alcanzado	>50% hasta 85% del alcance
C	Completamente alcanzado	>85% hasta 100% del alcance

El conjunto de calificaciones de los atributos de un proceso forman su perfil [Flores, 2008]. El resultado de una evaluación incluye un conjunto de perfiles del proceso para los procesos evaluados. El nivel de capacidad alcanzado por proceso se deriva de la calificación de los atributos correspondientes, tomando como referencia la Tabla 6. El nivel de capacidad del proceso es el nivel cuyo cumplimiento de los atributos es, al menos, ampliamente alcanzado (A) y el cumplimiento de los atributos de los niveles inferiores es completamente alcanzado (C) [NYCE, 2005a]. El conjunto de niveles de capacidad de procesos implantados, que están dentro del alcance de la evaluación, constituye un perfil de nivel de capacidad de los procesos.

2.3.2.3. *Proceso de evaluación*

El proceso definido para la aplicación del método de evaluación considera la preparación (actividad previa a la evaluación), y las actividades propias de la evaluación tales como planeación, ejecución, generación, entrega de resultados y cierre (véase Figura 2.9).

En la planeación, el evaluador certificado confirma el compromiso con el promotor para realizar la evaluación y conoce a los miembros del equipo de evaluación. Posteriormente, identifica los proyectos a evaluar y a los participantes en la evaluación, elabora el plan de evaluación, lo valida con el promotor y prepara al equipo de evaluación y a los participantes. En la ejecución, para cada proyecto a evaluar, el equipo de evaluación realiza una revisión de la documentación solicitada, prepara y realiza la entrevista con el Responsable de Administración del Proyecto Específico y con su equipo de trabajo. Adicionalmente, por cada responsable de los procesos de Alta Dirección y Gestión, se realiza la revisión de la documentación, y se prepara y realiza una entrevista con el responsable. La información recogida se registra como evidencia documental y los cuestionarios de la evaluación proporcionan evidencia oral. Finalmente, se consolida y se corrobora la información para obtener así la tabla de perfiles de calificaciones sobre los atributos [Oktaba, 2005b]. En la *generación de resultados*, el equipo de evaluación general el perfil del nivel de capacidad de los

procesos implantados y el nivel de madurez de la organización. En base a éstos, se elabora el reporte de resultados. En la entrega de resultados, el evaluador certificado presenta a la organización los resultados obtenidos y entrega el reporte de resultados al promotor. En el cierre de la evaluación, el evaluador certificado genera y envía el reporte estadístico al organismo rector y realiza las actividades de cierre con el equipo de evaluación [Oktaba, 2005b].

Tabla 6. Calificación del nivel de capacidad del proceso [NYCE, 2005a].

Atributo	Nivel / Calificación mínima	1	2	3	4	5
Realización del proceso		A	C	C	C	C
Administración de la realización		-	A	C	C	C
Administración del producto de trabajo		-	A	C	C	C
Definición del proceso		-	-	A	C	C
Implantación del proceso		-	-	A	C	C
Medición del proceso		-	-	-	A	C
Control del proceso		-	-	-	A	C
Innovación del proceso		-	-	-	-	A
Optimización del proceso		-	-	-	-	A

Al igual que MoProSoft, EvalProSoft hace uso de un patrón de procesos que sirve para realizar la documentación del proceso utilizado por el método de evaluación. Este patrón consta de las mismas partes: Definición general del proceso, Prácticas, y Guías de Ajuste.

2.4. Impacto de MoProSoft en la industria de software

De acuerdo con Oktaba [Oktaba, 2007], en el 2004 se realizaron pruebas sobre cuatro pequeñas empresas de software mexicanas para evaluar la complejidad de la implantación y utilidad de MoProSoft como modelo de procesos de software, así como también para determinar los costos de utilizar el método de evaluación EvalProSoft.

Las evaluaciones iniciales para establecer la capacidad de la línea base de procesos de las empresas, mostraron que todas las organizaciones evaluadas se encontraban entre un rango de 0 y 1. Durante los siguientes seis meses, diversos consultores dirigieron a las empresas en la adaptación y adopción de MoProSoft. Cuando las empresas fueron evaluadas por segunda ocasión, todas las empresas alcanzaron un aumento promedio del 1.08 en el nivel de capacidad de todos sus procesos [Oktaba, 2006; Oktaba, 2007].

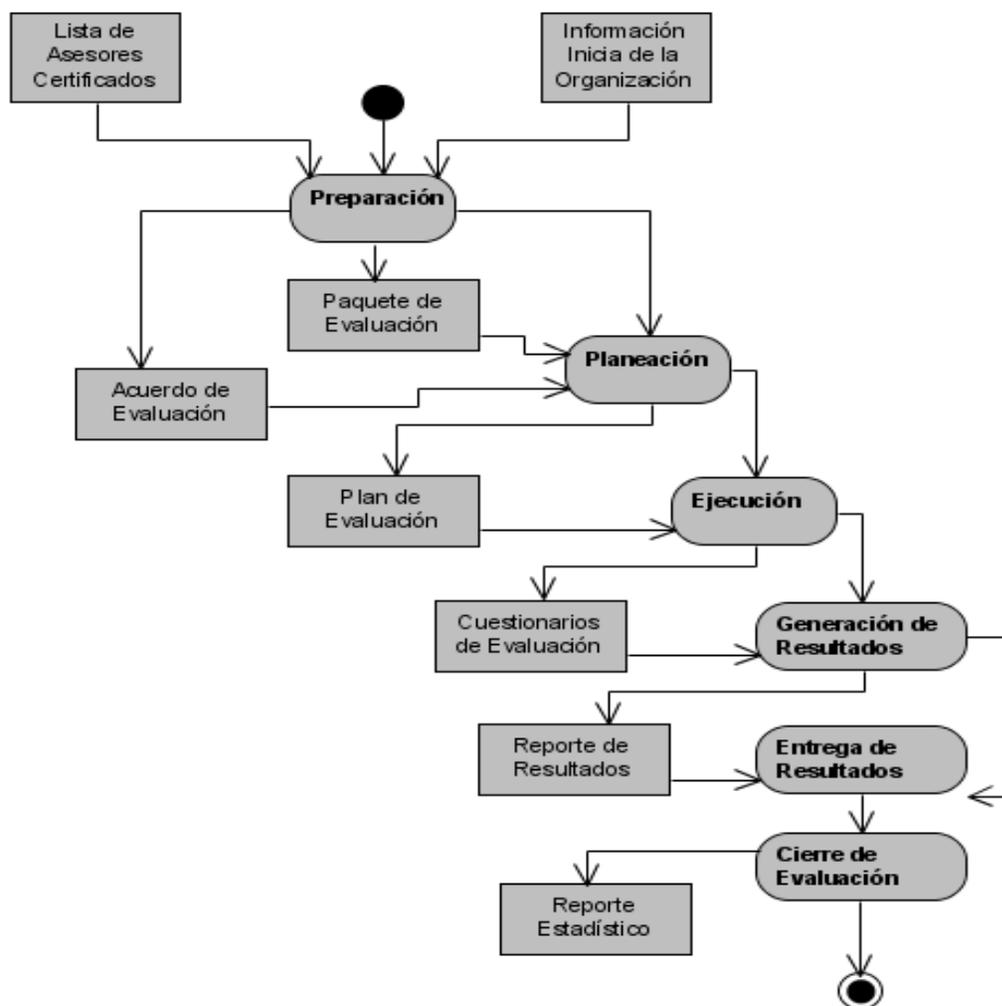


Figura 2.9. Diagrama de proceso de evaluación [NYCE, 2005].

La Tabla 7 muestra el perfil, el total de esfuerzo de mejora en horas, y el esfuerzo por persona por cada una de las empresas participantes. La última columna indica el promedio obtenido en la mejora en la capacidad de los procesos. Las pruebas realizadas a MoProSoft y EvalProSoft confirmaron los hallazgos encontrados en el proceso a través de cinco criterios, presentados en [Oktaba, 2006]. Debido a los resultados de este experimento, la Secretaria de Economía decidió formalizar a MoProSoft y EvalProSoft como una norma mexicana. Esta norma es conocida con el nombre de *NMX-I-059-NYCE-2005: Tecnología de la Información – Modelos de procesos y evaluación para desarrollo y mantenimiento de software*.

Tabla 7. Datos de esfuerzo y mejora por empresa [Oktaba, 2006].

Empresa	Número de Empleados	Esfuerzo total en horas	Esfuerzo por persona	Promedio de mejora
A	17	479	28.18	1.00
B	8	199	24.88	1.00
C	17	628	36.94	1.56
D	29	221	7.62	0.78
Promedio	18	383	21.28	1.08

Una vez que la norma fue establecida, el Gobierno Mexicano brindó apoyos a las empresas interesadas en la adopción y certificación del modelo a través del programa PROSOFT. De esta manera la adopción del modelo tuvo una gran demanda y aceptación dentro de la industria mexicana de software. De acuerdo con los datos de NYCE, hasta Diciembre del 2010 existían un total de 218 empresas que contaban con la certificación en MoProSoft, de las cuales 181 contaban con Nivel 1, 36 con Nivel 2, y solamente 1 con Nivel 3 [URL-4]. Para el año 2013, NYCE reportó un total de 265 certificaciones, 6 de ellas en Nivel 0, 147 en Nivel 1 (34 se perdieron), 108 en Nivel 2, y por último 4 para Nivel 3. El Distrito Federal y Nuevo León son los Estados que cuentan con mayor número de certificaciones, con 69 y 59 respectivamente. La Figura 2.10 muestra la gráfica de distribución de las certificaciones en un total de veinte Entidades Federativas.

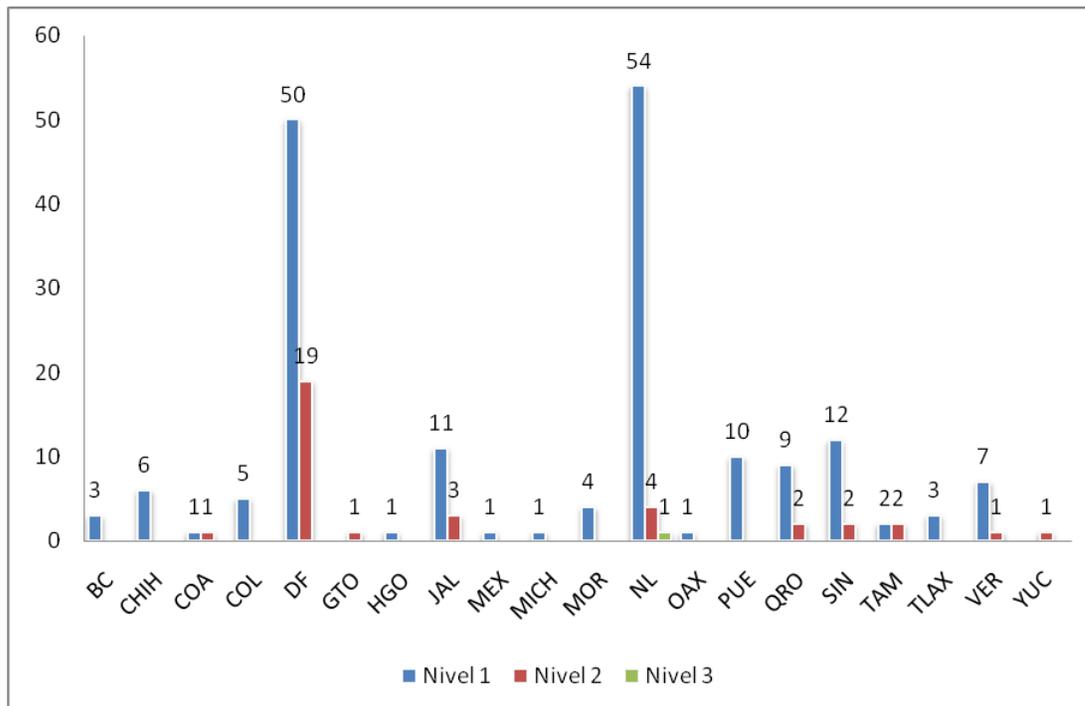


Figura 2.10. Distribución de certificaciones por entidad federativa [URL-4].

De acuerdo a la Secretaría de Economía, hasta el año 2008 existían en México alrededor de 2,130 empresas en la industria de servicios relacionados con las TI, de las cuales un aproximado del 85% se dedicaba al rubro de desarrollo y mantenimiento de software [SE, 2008]. De esta manera, y tomando en cuenta el posible crecimiento del número de empresas de software a dos años, aproximadamente una décima parte de las empresas en México habrían adoptado el modelo MoProSoft.

En el aspecto Internacional, la aparición de MoProSoft ha tenido un impacto positivo en la comunidad dedicada a la investigación de modelos de procesos orientados a las pequeñas entidades desarrolladoras de software. Siendo la principal muestra de lo anterior, el *Proyecto CompetiSoft* [Oktaba et al., 2007] desarrollado a nivel Latinoamérica, además del desarrollo del estándar ISO/IEC 29110 [ISO, 2011] en el ámbito Internacional.

2.4.1. Proyecto CompetiSoft

En 2005, un número importante de investigadores y profesionales enfocados al desarrollo de software en Latinoamérica reconocieron la importancia de contar con un marco de mejora y certificación enfocado exclusivamente a las pequeñas organizaciones. La solución propuesta fue el

desarrollo del proyecto CompetiSoft que se presentó ante el Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED), del cual forman parte 21 países de Latinoamérica, más España y Portugal, y que tiene como objetivo principal la cooperación multilateral científica y tecnológica [URL-5].

Así, los participantes en CompetiSoft fueron divididos en dos categorías principales:

- Investigadores de Universidades en Argentina, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador, México, Perú, Portugal, España, Uruguay y Venezuela.
- El grupo crítico de referencia, integrado por el Instituto de Estandarización y Certificación Argentino, el Gobierno de la Provincia Neuquén, Argentina; y pequeñas compañías de software de diferentes países, incluyendo cinco de Colombia, cuatro de Perú, tres de España, y una de Argentina, Chile, Ecuador, México y Uruguay.

Para desarrollar el proyecto CompetiSoft, se tomaron en cuenta diferentes iniciativas Latinoamericanas, tales como: MoProSoft, el Modelo Brasileño para la Mejora de Procesos [Santos et al., 2007], así como la Mejora Ágil de Procesos Software (Agile SPI) [Hurtado et al., 2008]. La metodología del Ministerio de Administración Pública del Gobierno de España, Métrica v3 [MAP, 2001], también fue considerada, ya que está destinada a mejorar los procesos y/o productos de software.

De acuerdo a [Oktaba et al., 2007], CompetiSoft fue desarrollado utilizando los modelos implantados en las pequeñas empresas de Latinoamérica, especialmente MoProSoft. En propias palabras de los desarrolladores del proyecto: *“Se puede ver a CompetiSoft como una evolución de MoProSoft, fortalecido con la experiencia de investigadores y profesionales en la implementación y mejora de procesos software, que conducen a un nuevo modelo de procesos de referencia y evaluación, mejor que MoProSoft y EvalProSoft, además de un nuevo modelo de mejora de procesos basado en Agile SPI”*. El proyecto ha sido introducido a la industria de software de América Latina para implantar una cultura de mejora de procesos, y se han establecido en las organizaciones los principios metodológicos para la normalización y certificación.

2.4.2. Estándar ISO/IEC 29110

La industria de software a nivel mundial está formada, en gran parte, por MPyMEs que suponen cerca del 90% de las empresas formales, y que generan entre el 40% y 50% del empleo total [Piattini y Garzás, 2007]. Por lo anterior, y debido al problema que supone establecer modelos internacionales (como CMMI-DEV, ISO/IEC 12207, ISO/IEC 15504) dada la importante inversión de dinero, tiempo y recursos, y principalmente por su orientación hacia las grandes organizaciones surge la necesidad de desarrollar un estándar enfocado a MPyMEs.

Así, en el año 2005, ISO reconoció las necesidades y problemas de las microempresas y estableció un grupo de trabajo llamado SC7-WG24³, cuyo objetivo es que sus estándares enfocados a los procesos de software puedan ser utilizados en pequeñas empresas desarrolladoras de software. Este grupo ha establecido un marco de trabajo común para describir perfiles evaluables del ciclo de vida software en Empresas muy Pequeñas (VSE, acrónimo en inglés para *Very Small Enterprises*) u organizaciones con menos de 25 empleados [Calvo-Manzano et al., 2008].

³ Debido a que los estándares de la ISO/IEC son difíciles de implantar en las pequeñas organizaciones, el grupo de trabajo SC7-W24 ha sido creado para desarrollar perfiles de estándares y reportes técnicos para auxiliar a las microempresas a cumplir con los estándares ISO enfocados en la Ingeniería de Software. Australia, Bélgica, Canadá, República Checa, Finlandia, India, Irlanda, Italia, Japón, Corea, Luxemburgo, México, Sudáfrica, Tailandia, EUA y el Reino Unido participan en este grupo, junto con la IEEE y el *International Council on Systems Engineering* (INCOSE).

En Mayo de 2006, durante una reunión en Tailandia, los miembros del SC7-WG24 deciden que MoProSoft fuera la base para el desarrollo del nuevo estándar. Sin embargo, se plantea la consideración de que MoProSoft está dirigido a empresas de mayor tamaño que las VSE. Por lo anterior, y en base a las categorías de procesos de MoProSoft, los perfiles para el nuevo estándar son definidos de la siguiente forma: Categoría de Operación como Perfil Básico, Categoría de Gerencia como Perfil Intermedio y la Categoría de Alta Dirección como Perfil Avanzado; para de esta forma brindar mayores oportunidades a las entidades desarrolladoras de software, de acuerdo a su tamaño [Oktaba, 2010]. Estos perfiles se publicarían en el año 2010 con el nombre de ISO/IEC 29110, pero fue hasta el año 2011, que fueron publicados en el sitio oficial de ISO [URL-6].

El nuevo estándar llamado *ISO/IEC 29110 Software engineering –Lifecycle profiles for Very Small Entities*, fue desarrollado para auxiliar a las pequeñas organizaciones dedicadas al desarrollo de software a alcanzar el reconocimiento internacional en la capacidad de sus procesos. El estándar ISO/IEC 29110 está dirigido a este tipo de entidades, y fue desarrollado para mejorar los productos y/o los servicios de software, y el desempeño de los procesos. Cabe mencionar que ISO/IEC 29110 no tiene como objetivo impedir el uso de ciclos de vida diferentes, tales como: cascada, iterativo, evolutivo, incremental o ágil [Mendoza et al., 2009; Ribaud et al., 2010; Varkoi, 2010].

La Tabla 8 muestra las partes en las que se divide el estándar ISO/IEC29110 [ISO, 2011a].

Tabla 8. Descripción de las partes del estándar ISO/IEC 29110.

Serie	Nombre	Descripción
Parte 1	Visión General	Ofrece los conceptos principales necesarios para comprender y utilizar los documentos del estándar.
Parte 2	Marco de trabajo y taxonomía	Este documento establece la lógica detrás de la definición y aplicación de los perfiles.
Parte 3	Guía de Evaluación	Describe el proceso que se ha de seguir para realizar una evaluación que determine las capacidades del proceso y la madurez organizativa.
Parte 4	Especificaciones de Perfil	Su propósito es proporcionar la composición definitiva de un perfil.
Parte 5	Guía de Ingeniería y Gestión	Proporciona orientación sobre su implementación y uso, o bien sobre un perfil.

La Figura 2.11 muestra las partes del ISO/IEC 29110 y su posición dentro del marco de trabajo de referencia. La Visión General y las Guías son publicadas en los Reportes Técnicos (TR, acrónimo en inglés para *Technical Report*), y los perfiles son publicados como Estándares Internacionales (IS, acrónimo en inglés para *International Standard*) [Calvo-Manzano et al., 2008; ISO, 2011a].

Hasta el momento solamente se ha publicado un grupo de perfiles, llamado Grupo de Perfil Genérico (IS 29110-4-1). Este grupo es aplicable para las VSE que no desarrollan productos críticos de software. Dentro de las Guías de Ingeniería y Gestión, solamente se ha publicado el Perfil Básico (TR 29110-5-1-2), el cual forma parte del Grupo de Perfil Genérico especificado en IS 29110-4-1. El Perfil Básico describe el desarrollo de software de una aplicación simple con un equipo de proyecto único, sin riesgo especial o factores circunstanciales [ISO, 2011a].

La guía TR 29110-5-1-2, cuenta con dos procesos (Gestión de Proyecto e Implementación de Software) que están basados en los procesos MoProSoft: Gestión de Proyectos Específicos, y Desarrollo y Mantenimiento de Software, respectivamente. Por esta razón, se puede deducir que para aquellas entidades que cuenten con la implantación de MoProSoft en sus procesos de desarrollo, se les facilitaría la implantación del nuevo estándar ISO/IEC 29110, ya que, debido a la similitud en estructura, definiciones, procesos y roles, la transición de uno a otro es de menor complejidad.

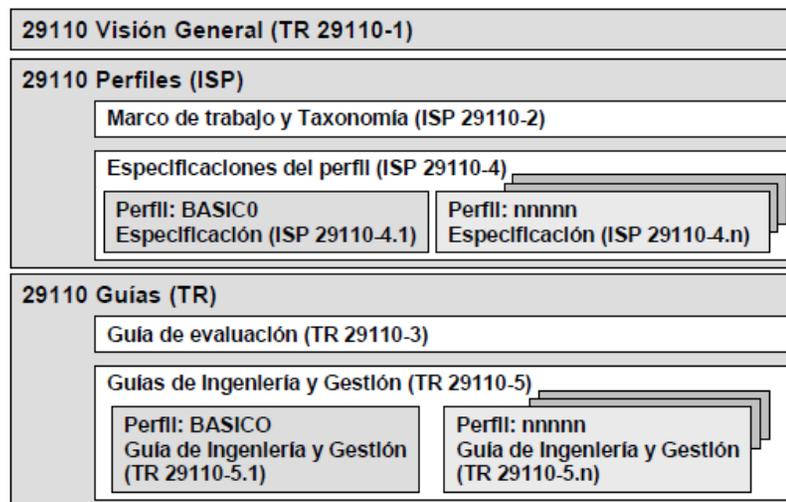


Figura 2.11. Posicionamiento de las partes dentro del marco de trabajo de ISO/IEC 29110 [ISO, 2011a].

2.5. Herramientas auxiliares para la implantación de iniciativas de SPI

La complejidad y la gran cantidad de tareas que implica el establecimiento de iniciativas de mejora debido a los factores que involucran, tales como: el manejo de personal, los flujos de actividades, la delegación de tareas, el manejo de productos, y la administración de la base de conocimientos, por mencionar algunas, demandan que éstas sean cubiertas con el auxilio de herramientas computarizadas, lo cual ya es una opción utilizada actualmente por las empresas. Las opciones modernas son herramientas que ayudan a la realización de tareas específicas, de esta manera el grupo de trabajo cuenta con un conjunto de programas informáticos que ayudan a solventar cada una de las actividades relacionadas con un proyecto de SPI. Sin embargo, al tratarse de herramientas que no han sido desarrolladas específicamente para un proyecto de este tipo, siguen siendo herramientas genéricas auxiliares, y no un marco de trabajo integral que pueda fungir como guía para apoyar a la empresa en un ciclo de mejora en los procesos de desarrollo; aunado a lo anterior, la incompatibilidad para trabajar conjuntamente hace de esta opción, aunque muy utilizada, que sea poco recomendable. No obstante, han surgido iniciativas tanto de empresas como de organismos rectores en materia de calidad de procesos, para crear herramientas integrales que ayuden a las organizaciones en la adopción de un modelo de mejora de procesos, y de esta manera mejorar la calidad de sus productos de software. A nivel internacional organismos como el SEI y NYCE, junto con empresas particulares, han desarrollado herramientas computarizadas para la adopción de los modelos CMMI y MoProSoft respectivamente. A continuación, se describen herramientas de software que pueden ser utilizadas por las organizaciones para conducir un proyecto de SPI. Las primeras de éstas están apegadas a modelos de proceso internacionales como CMMI y cubren todas las fases de un modelo de mejora, en última instancia se presenta la herramienta KWE 2.0 que si bien no cubre todas las fases, es una herramienta desarrollada para ser un soporte en la adopción del modelo mexicano de procesos MoProSoft y es una opción interesante de apoyo para las empresas de nuestro país, ya que es avalada por el organismo rector NYCE. Al final de la sección se presenta un análisis de los entornos de soporte, para detectar las características que ofrecen; con esto se intenta determinar la facilidad de adaptación al pequeño entorno de desarrollo y, posteriormente, su uso en las MPyMEs desarrolladoras de software en México.

2.5.1. processMax®

processMax® es una plataforma desarrollada por *Pragma Systems Corporation* [URL-16]. processMax® es un producto Web intranet que proporciona una solución total para la gestión de proyectos enfocados en el desarrollo y despliegue de los procesos de negocio. De acuerdo a sus creadores, esta herramienta es una solución completa y probada para los retos de implantación de procesos complejos como CMMI, ya que los productos processMax® incluyen procedimientos detallados y orientación para la integración de procesos de alta calidad al trabajo diario. La plataforma proporciona a la organización todo lo necesario en materia de políticas, procedimientos, directrices, flujos de trabajo y la automatización de evaluaciones y reportes, para los Niveles 2 y 3 de CMMI-DEV. Pragma Systems Corporation garantiza que las organizaciones que utilicen processMax® no fracasen en una evaluación oficial del SEI.

Para alcanzar la implantación del modelo CMMI-DEV, la plataforma toma en cuenta los siguientes enfoques:

1. **Centralizar la Gestión de Proyectos.** En la actualidad los entornos de desarrollo de software y sistemas son descentralizados. processMax® mediante la tecnología Web intranet proporciona la visibilidad efectiva y eficiente en los proyectos. Todo lo referente al proyecto, documentos, informes, memorandos, solicitudes de cambio, entre otros, se mantienen centralizados, junto con el estado e historia del proceso, facilitando así la visibilidad de la gestión del proyecto y del personal.
2. **Rápida implantación.** Las organizaciones que utilizan processMax® pueden lograr el cumplimiento de CMMI en meses, no en años. Las estadísticas recopiladas por el SEI muestran que las organizaciones han requerido, en promedio, casi dos años para alcanzar el nivel 2 y dos más para alcanzar el nivel 3. Con processMax® es posible cumplir de forma inmediata, y por lo general puede generar suficientes pruebas de cumplimiento en cuestión de meses.
3. **Integrar los procesos en el trabajo día a día.** processMax® proporciona procedimientos paso a paso combinados con la gestión de documentos y flujos de trabajo automatizados. Por lo tanto, el proceso es totalmente integrado con el trabajo día a día de los líderes de proyecto y desarrolladores, a diferencia de los procedimientos tradicionales a menudo rudimentarios.
4. **Obtener los beneficios de procesos probados.** Con processMax® la organización opera en cumplimiento con CMMI, que tiene como resultado, obtener mejores estimaciones de costos y tiempo, aumentar la productividad, mejorar la visibilidad de la gestión, reducir los defectos y reducir los riesgos.

Las características principales de la plataforma se listan a continuación:

- La plataforma se instala en un servidor Web de una intranet, para ser accesible desde cualquier navegador Web, simplificando el mantenimiento y centralizando la gestión del proceso y proyecto.
- processMax proporciona un repositorio de documentos para cada proyecto, con una gestión total incluyendo control de versiones, control de cambios e historias de procesos.
- La plataforma incluye un flujo automatizado de trabajo para notificar a los miembros del proyecto cualquier tipo de eventos relacionados con el proceso, incluyendo: cuando los documentos están disponibles para revisión y aprobación, cuando nuevas actividades son asignadas, cuando se agendan reuniones y demás.
- Incluye detallados procedimientos paso a paso, escritos desde el punto de vista de un jefe de proyecto o desarrollador. Para cada paso, processMax proporciona instrucciones claras con acceso directo a los documentos relevantes del proyecto.

- Las valoraciones de procesos son más rápidas y eficientes usando processMax®, lo que reduce el costo y la subjetividad.
- La plataforma registra automáticamente el progreso, estatus y calidad de la implantación. processMax® genera informes, tanto en forma gráfica y tabular, proporcionando a los responsables de mejora una visión en tiempo real de los proyectos y el desempeño organizacional (véase Figura 2.12).

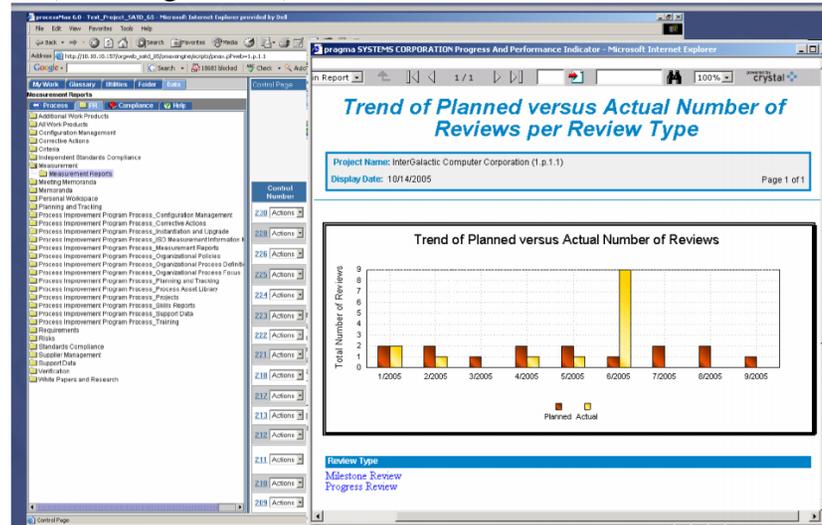


Figura 2.12. Generador de reportes de processMax® [URL-16].

De acuerdo a lo propuesto por Pragma Systems, processMax® es una herramienta que cumple con todas las funcionalidades para cubrir las etapas necesarias para un programa de mejora en organizaciones desarrolladoras de software. Es tal la confianza de los creadores en que su plataforma puede establecer procesos CMMI en Nivel 2 y 3, que garantizan en su página oficial [URL-16] la consecución de una certificación SEI para todas aquellas organizaciones que utilicen su herramienta. Sin embargo, sin poner en tela de juicio tal afirmación, tomando en cuenta las características de la industria de software en México en donde la mayoría de las empresas que la conforman son micro o pequeñas, en un estado emergente y con nula experiencia en materia de implantación de procesos de mejora; processMax® parece ser una herramienta fuera del alcance de las MPyME de nuestro país, en base a las siguientes razones:

- El hecho que se encuentre desarrollada para la implantación del modelo CMMI en sus Niveles 2 y 3, hace que la plataforma se enfoque en grandes empresas con estructuras complejas; por lo que la naturaleza MPyME de la industria mexicana es poco adaptable.
- El hecho de solo enfocarse en los Niveles 2 y 3 de CMMI hace que sea necesario que la empresa cuente con un conocimiento previo o introductorio acerca de este modelo internacional para tener una noción acerca de la terminología y procesos, de no contar con este conocimiento el resultado puede generar una curva de aprendizaje mayor para lograr explotar todas las características y posibilidades que presenta la herramienta.
- Debido a las características que ofrece la plataforma, tiene un precio de licencia bastante considerable, por lo que para las pequeñas empresas mexicanas puede resultar difícil y hasta imposible el costear este tipo de inversión.
- El uso de la plataforma está enfocado en organizaciones con experiencia en la mejora de procesos, sin embargo, debido a la poca o nula experiencia de las MPyME mexicanas, el uso de herramientas de tal complejidad puede resultar una tarea demasiado difícil y confusa, arrojando resultados negativos y distintos a los esperados con la mejora.

2.5.2. iGrafx®

iGrafx® es una suite integrada de software dedicada a la unificación de equipos de mejora de procesos a través de una interfaz amigable para el usuario, un entorno colaborativo para el modelado, análisis y mejora de los procesos de negocios [URL-17]. De acuerdo a los creadores de esta suite, esta herramienta es capaz de adaptarse tanto a pequeñas empresas como a organizaciones multinacionales que buscan crear los diagramas de procesos, alinearse de acuerdo a un estándar internacional y crecer y asegurar el éxito en materia de las TI.

Existen distintas versiones de la suite que se acoplan a distintas áreas de las TI, sin embargo, existe una en particular que se enfoca a la implantación de una mejora de procesos dentro de una empresa desarrolladora de software, esta versión lleva por nombre *iGrafx® Process for Six Sigma*, que como su nombre lo indica se basa en la metodología para la mejora de procesos Six Sigma. Six Sigma es un modelo de mejora de procesos, centrado en la reducción de la variabilidad de los mismos y enfocado en la medición orientada a la mejora continua que se inicia con los objetivos de negocio y el valor directo a los consumidores. El modelo se centra en la eliminación de los defectos del proceso, con el objetivo de que las compañías alcancen resultados significativos respecto a la satisfacción del cliente, ingresos y control de costos. Debido a las cinco fases que definen el ciclo de mejora de Six Sigma: *Definir, Medir, Analizar, Mejorar, y Controlar* el proceso, éste ha sido ampliamente utilizado para conducir proyectos de SPI [Janiszewski, 2004].

iGrafx® proporciona a las compañías soluciones poderosas y exclusivas para que Six Sigma las ayude a mejorar sus procesos, eliminar gastos innecesarios y mejorar el servicio al cliente. También ofrece un marco visual para relacionar los datos derivados de la realización de sus respectivos procesos [URL-17]. El fabricante menciona también que mediante su enfoque centrado en el proceso, los profesionales de Six Sigma pueden aumentar la media de proyectos acabados al año, así como reducir los costos de ese mismo periodo de tiempo.

El uso de iGrafx® permite:

- Alcanzar una mejora en los resultados gracias a una mayor comprensión de los procesos.
- Tener mayor confianza en sus soluciones a través de análisis y test comprensivos.
- Alcanzar mayor calidad organizacional, operativa, de ejecución y en el producto, a través de la identificación y reducción de los riesgos asociados.
- Disminuir el costo y el tiempo de las pruebas a través de la simulación y automatización de las interfaces.
- Conseguir beneficios más rápidos por la reducción del tiempo del proyecto.

Entre sus principales características se encuentran las siguientes:

- Mapeo de procesos.
- Diagramas de causa y efecto.
- Modelado de procesos (véase Figura 2.13).
- Métodos de Six Sigma.
- Diseño de experimentos o proyectos piloto.
- Análisis y Reportes basados en Six Sigma.

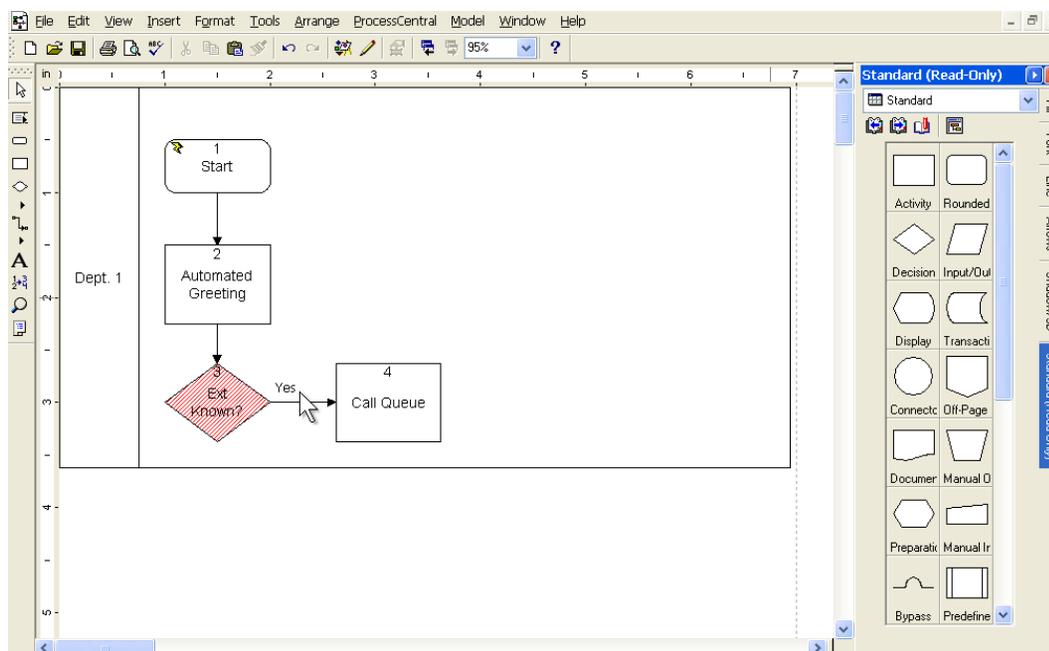


Figura 2.13. Pantalla para modelar procesos en iGrafx® [URL-17].

La suite iGrafx® presenta una variedad de ventajas que pueden ser explotadas por las organizaciones que buscan establecer iniciativas de SPI, la principal de ellas es que ésta se basa en un modelo de mejora reconocido internacionalmente, como lo es Six Sigma, el cual es ampliamente utilizado en la industria de las TI para la mejora de procesos. El hecho de trabajar con un modelo de mejora ya establecido dota a la herramienta de mayor confiabilidad, debido a que se rige por los marcos de trabajo, fases, métricas, medidas y actividades de éste. Dentro de este contexto, esto es muy importante, ya que la misma suite marca el flujo de trabajo que deben seguir los integrantes del grupo de trabajo para llevar por buen camino la iniciativa de mejora. En el aspecto técnico, la herramienta es bastante robusta, ya que integra aspectos que facilitan las actividades y tareas que se realizan en cada una de las fases del modelo. Por ejemplo, para la fase de *Definir el proceso* la herramienta proporciona un módulo para crear el diagrama de procesos de la empresa, para de esta manera poder plasmar y obtener un proceso más apegado a lo que realmente se realiza dentro de la empresa (véase Figura 2.13).

A continuación se enlistan las ventajas más importantes de iGrafx®:

- Se basa en el modelo de mejora internacional conocido como Six Sigma.
- Sirve de guía para la implantación de una iniciativa de SPI.
- Permite el mapeo de procesos de la organización con los procesos recomendados por el modelo de mejora.
- Brinda el seguimiento y control a cada una de las fases de mejora.
- Proporciona módulos especializados que brindan datos cuantitativos sobre el avance del proyecto de mejora y muestran las deficiencias que existen en los procesos de las organizaciones.
- Permite la creación de platillas personalizadas que ayudan a documentar los procesos existentes y el nuevo proceso a implantar.
- Proporciona informes gráficos y tabulares acerca del proceso de la organización.

- Facilita el diseño de experimentos y la simulación de eventos en proyectos piloto.
- Incluye la exportación a formatos especializados para el análisis de datos (Minitab y JMP).

Sin embargo, a pesar de ser una herramienta de gran utilidad, iGrafx® presenta algunas brechas significativas para la implantación de iniciativas de SPI en organizaciones desarrolladoras de software de nuestro país, debido a que la gran mayoría son MPyMEs. Uno de los principales factores es el costo de la licencia para operar la suite, puesto que oscila en los miles de dólares; precio que es muy elevado para pequeñas empresas, que si bien es cierto desean mejorar sus procesos de desarrollo carecen del capital para obtener una herramienta de este tipo para guiar sus proyectos de mejora. Así, las desventajas de iGrafx® en relación a una iniciativa de SPI en MPyMEs se pueden resumir de la siguiente manera:

- Si bien está diseñada para la mejora en todo tipo de proyectos de TI, no está específicamente desarrollada para el rubro de desarrollo de software.
- El modelo Six Sigma está desarrollado para la mejora en empresas establecidas y de gran tamaño, por lo cual, para las pequeñas organizaciones la tarea de establecer una mejora basada en este modelo puede ser confusa y compleja.
- Six Sigma, al tratarse de un modelo de mejora, marca las fases que una organización tiene que llevar a cabo para mejorar sus procesos, sin embargo, iGrafx® no hace uso de prácticas de un modelo de procesos (e.g., CMMI o MoProSoft) para establecer las actividades necesarias para que la empresa logre un proceso de software capaz y maduro.
- Debido a la inexperiencia de las MPyMEs de nuestro país, en materia de mejora de procesos, una herramienta tan robusta y con una gran cantidad de módulos puede ser compleja, confusa y hasta dañina, puesto que podría convertirse en una gran inversión y no explotar por completo todas sus características. Otro riesgo puede ser que no se adapte a las características ni a las necesidades de la organización.
- Se requiere de una licencia comercial para utilizar la suite, la cual oscila entre los 20 mil dólares, lo cual representa una problemática para las MPyMEs que difícilmente cuentan con dicho capital.
- La suite está completamente en inglés, lo que puede influir significativamente su tiempo de adopción haciéndolo aún más prolongado.

2.5.3. Stages®

Stages® es una suite para la gestión de procesos optimizada para la ingeniería de sistemas y otros procesos técnicos de negocio, desarrollada por la empresa alemana Method Park [URL-11]. De acuerdo a sus desarrolladores, Stages® permite a las organizaciones definir, gestionar, aprobar, controlar y mejorar sus procesos, y al mismo tiempo asegurar el cumplimiento de modelos de procesos y estándares internacionales como CMMI, SPICE, COBIT o los estándares de ISO.

El concepto clave detrás del sistema de gestión de procesos de Stages® es realizar de forma conjunta la teoría de procesos y las prácticas de los proyectos reales. Stages® ha sido creada y optimizada para procesos complejos y está integrada por un importante número de módulos que son utilizados comúnmente en entornos de ingeniería de desarrollo y que facilitan su utilización. La suite se centra en el usuario final de los procesos y facilita el acceso a las descripciones del proceso, lo que permite que el usuario final entienda de forma clara los procesos de principio a fin y se centre en los detalles de funcionamiento de cada uno de ellos.

En base a una aplicación Web, los miembros del equipo de trabajo tienen acceso directo y fácil a todos los documentos del proyecto, plantillas de documentos, las mejores prácticas o conjuntos de documentos de “cómo hacerlo”, en lugar de una gran cantidad de teoría sobre los procesos y complejos diagramas de flujo de trabajo.

Para llevar a cabo una mejora dentro de los procesos de la empresa, la suite de gestión de Stages® proporciona diferentes funcionalidades, entre las principales se encuentran las siguientes:

1. Definir y gestionar un conjunto comparable de procesos y productos que se utilizan en la arquitectura de procesos dentro de la organización. Lo anterior con el fin de obtener un marco de procesos comparable con los procesos establecidos por un estándar, y así redefinir y estructurar el proceso actual, reutilizando las mejores prácticas y productos de los procesos actuales (véase Figura 2.14).
2. Comparar el proceso actual de la empresa, con el modelo de referencia y/o estándares de la industria, mediante el mapeo de ambos procesos. Mediante este mapeo se presenta un ahorro considerable de tiempo en la identificación de discrepancias entre ambos procesos.
3. Obtener automáticamente los flujos de procesos, para posteriormente implantarlos en proyectos reales o piloto, y de esta manera obtener un marco de procesos más eficiente, utilizando procesos y productos optimizados.
4. Integrar los procesos y los productos a los proyectos de trabajo, mediante el diagrama que describe el flujo del nuevo proceso, y la asignación de roles para cada actividad. Gestionar los productos de trabajo y ver el estado actual de desarrollo del proyecto mediante las diferentes herramientas del entorno de trabajo de la suite.
5. Mejorar la elaboración de evaluaciones. Llevar a cabo evaluaciones, reduciendo así gastos en la generación de pruebas objetivas, basadas en los procesos y productos de modelo de referencias.
6. Medir y controlar el desempeño e implantación de la mejora dentro de la organización. Para esto, Stages® establece metas globales, alineadas con los objetivos de la organización, lo que permite realizar una medición del desempeño de la mejora para optimizar el rendimiento de la organización.

Al igual que la herramienta anterior, Stages® presenta inconvenientes para el tipo de empresas desarrolladoras de software de México, debido a su estructura organizacional y el tamaño. Al estar basada en modelos de procesos internacionales como CMMI, Stages® se enfoca en la mejora de procesos de grandes organizaciones, dejando de lado las necesidades que por su naturaleza presenta una MPyME mexicana desarrolladora de software. Por la misma razón, la complejidad en el uso de la plataforma es alta, debido a la cantidad de actividades, tareas, roles y productos, que se consideran para llevar a cabo la mejora y con las cuales una empresa mexicana no está familiarizada.

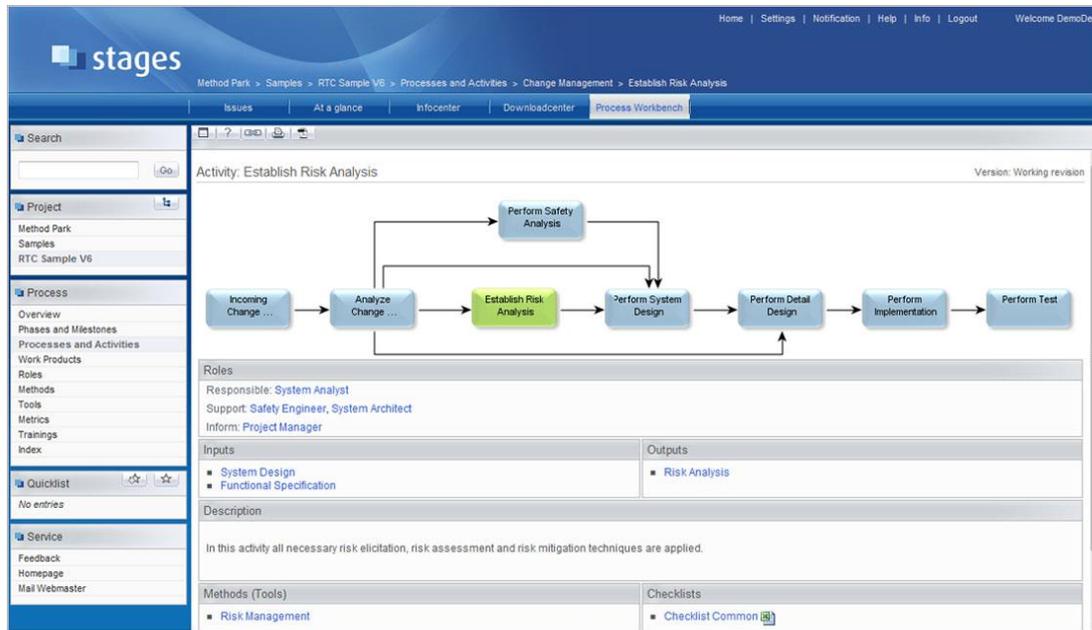


Figura 2.14. Flujo de procesos generado con Stages® [URL-11].

Sin embargo, Stages® es una opción viable para aquellas empresas que cuentan con la estructura, experiencia y recursos para utilizar la plataforma. Así, entre las principales ventajas que presenta Stages® se enlistan las siguientes:

- Está basada principalmente en el modelo internacional de procesos CMMI (en sus versiones DEV v1.2 y v1.3), lo que brinda un marco de trabajo de confianza muy amplio.
- La plataforma cuenta con características y funcionalidades que funcionan como auxiliares en la implantación de iniciativas de mejora en los procesos de software de una organización.
- Cubre las fases más importantes de una iniciativa de mejora, concretamente: Definir, Diagnosticar, Planificar y Establecer.
- En base al proceso actual de la empresa y por medio de un mapeo con el proceso definido por el modelo de referencia CMMI, se define un nuevo proceso de software que se adapte a la organización.
- El nuevo proceso cuenta con la información necesaria para su establecimiento, es decir asignación de tareas para cada rol, flujo de actividades, productos de salida y entrada, fechas de control, dependencia de tareas, y demás información importante para establecer el proceso.

A pesar de las posibilidades ofrecidas por Stages®, la plataforma presenta desventajas notables para la mayoría de las empresas desarrolladoras de software de México; esto debido a la estructura y tamaño de estas organizaciones y a la inmadurez en materia de mejora de procesos. En este sentido, entre sus principales desventajas se encuentran:

- Dado que se basa en un modelo de procesos internacional como lo es CMMI, la plataforma se enfoca en empresas de gran tamaño y con estructuras organizacionales complejas, muy diferentes a las empresas en México que en su gran mayoría son MPyME.

- Terminología y procesos complejos, que tienen como resultado una curva de aprendizaje mayor para lograr explotar todas las características y posibilidades que presenta Stages®.
- Debido a lo potente y robusta que es la plataforma, resulta poco costeable para la mayoría de las empresas mexicanas desarrolladoras de software.
- El uso de la plataforma está enfocado a organizaciones con experiencia en la mejora de procesos, sin embargo, debido a la poca o nula experiencia de las MPyME mexicanas, el uso de herramientas de tal complejidad puede resultar una tarea demasiado difícil y confusa, arrojando resultados negativos y distintos a los esperados con la mejora.

2.5.4. KWE 2.0

A pesar de no contar con muchas de las características de las anteriores herramientas, KWE 2.0 presenta una ventaja para las empresas mexicanas desarrolladoras de software, dado que está basada en el modelo MoProSoft, a través de la norma NMX-I-059-NYCE-2011. KWE 2.0 es la última versión de la herramienta proporcionada por el organismo rector mexicano NYCE, para facilitar la adopción del modelo MoProSoft en una empresa de software.

La herramienta funciona como una base de conocimientos adaptada 100% a los requerimientos de la norma NMX-I-059-NYCE-2011 y permite realizar la gestión, el control y la supervisión de las actividades y los productos de trabajo, así como realizar los seguimientos enunciados en la norma.

De acuerdo a sus creadores, KWE 2.0 fue desarrollada como resultado directo de la realimentación de los clientes, y de la investigación de diferentes prácticas para compartir la información dentro de las organizaciones. La investigación demostró claramente que ninguna solución podía enfrentar estas necesidades para toda una organización, para los equipos de trabajo pequeños o para quienes, con fines específicos, comparten información de diferentes maneras a como lo hacen los equipos grandes de trabajo. En este sentido, cuando las organizaciones implantan KWE 2.0 pueden enfrentar los retos de compartir, gestionar, controlar y emitir información de todo tipo con una infraestructura mediana y de una forma amigable dejando registro de toda la información (conocimiento) que se gesta dentro de las organizaciones [URL-4].

La página oficial de NYCE argumenta que los principales beneficios ofrecidos por KWE 2.0 son los siguientes:

- La herramienta cuenta con tres características principales: personalización, integración y colaboración entre usuarios.
- Permite un mejor seguimiento de las actividades, gracias a la característica de gestión de un número indefinido de seguimientos por cada tarea.
- Permite la creación de repositorios que a su vez son personalizables con su propio conjunto de actividades, lo que ayuda a que la herramienta se adapte de una forma natural a las necesidades de la empresa.
- Permite el envío o la asignación de una tarea específica a diferentes usuarios a la vez, con lo cual se gana eficiencia y practicidad. Permite adjuntar múltiples archivos por tarea o seguimiento.
- Permite el envío de un correo electrónico de notificación cuando es asignada o delegada una actividad.
- Permite llevar una adecuada administración de los proyectos (planeación, seguimiento y control), sin importar las características del mismo.

- Permite crear varios repositorios para personalizar los productos de trabajo y las actividades.

Estos beneficios ofrecen a las empresas una diversidad de funcionalidades para auxiliar a las empresas en la implantación de MoProSoft. Sin embargo, la naturaleza de KWE 2.0 es la de una base de conocimiento automatizada, es decir, una herramienta encargada de gestionar los recursos con el fin de minimizar el tiempo de adopción de MoProSoft considerando que existe el conocimiento previo sobre el modelo y dando por hecho que la empresa tiene experiencia en materia de mejora de procesos. En este sentido, la gestión de procesos, actividades y productos de trabajo es parte de la fase de establecimiento de la mejora y KWE 2.0 la cubre proporcionando módulos para la creación de repositorios que contendrán los productos de trabajo y referencias de las actividades realizadas en cada proceso del modelo (véase Figura 2.15).

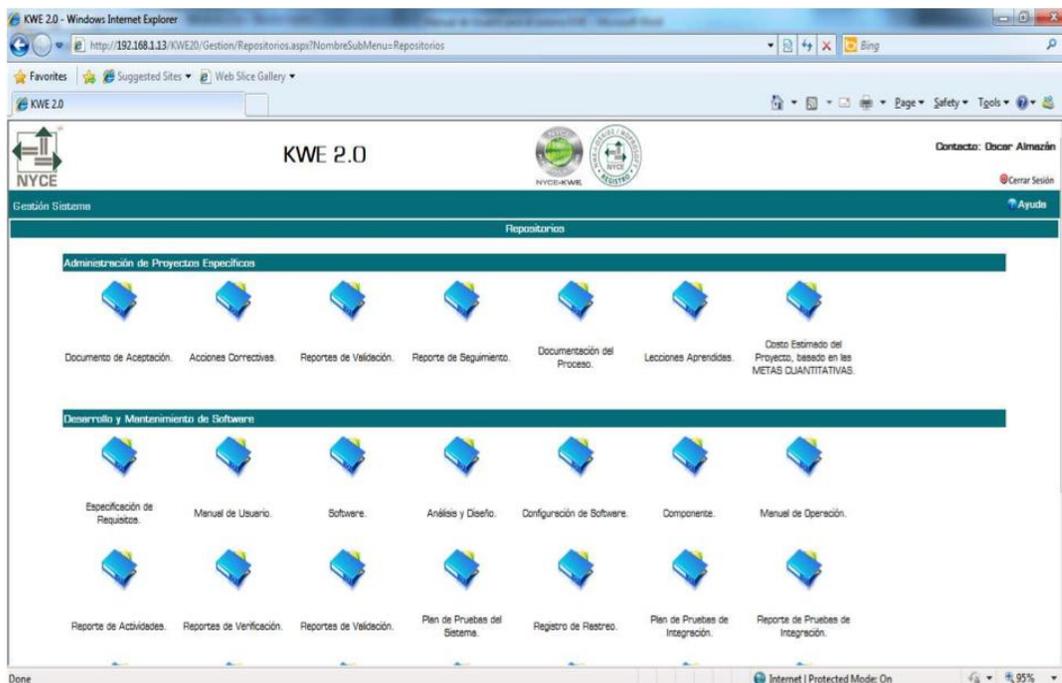


Figura 2.15. Repositorio de actividades y productos de trabajo en KWE 2.0 [URL-4].

La asignación de roles para los participantes en el proyecto (véase Figura 2.16), permite que KWE 2.0 facilite la asignación implícita de actividades del proceso con las cuales está ligada algún rol, y con lo cual posteriormente será posible monitorear el estado de cada una de las actividades designadas a cada uno de los roles. KWE 2.0 ofrece también funcionalidades para gestionar la base de conocimiento generada para cada uno de los procesos de MoProSoft, de igual manera cuenta con características para monitorear el estado de las actividades, y por consecuencia obtener un panorama acerca del avance del proyecto de mejora.

Sin embargo, todas las funcionalidades aquí resumidas se enfocan en cubrir la fase de establecimiento del modelo en la estructura organizacional de la empresa, lo cual genera una brecha bastante amplia entre el inicio y el establecimiento del proyecto, dejando de lado fases de gran importancia como lo son el diagnóstico y la planeación del proyecto. Además, el uso de la herramienta requiere del conocimiento previo sobre el modelo y de la experiencia en la implantación de iniciativas de mejora.

Sin duda, KWE 2.0 es una opción viable para la implantación de manera correcta del modelo MoProSoft, además de que ofrece plena confianza a las empresas por el hecho de estar desarrollada y respaldada por el organismo rector encargado de la evaluación de MoProSoft, que es NYCE. No

obstante, la herramienta no brinda las funcionalidades necesarias para cumplir con un ciclo completo de SPI, quedando al margen el resto de fases que son pilares fundamentales para llevar por buen camino el proceso de mejora.

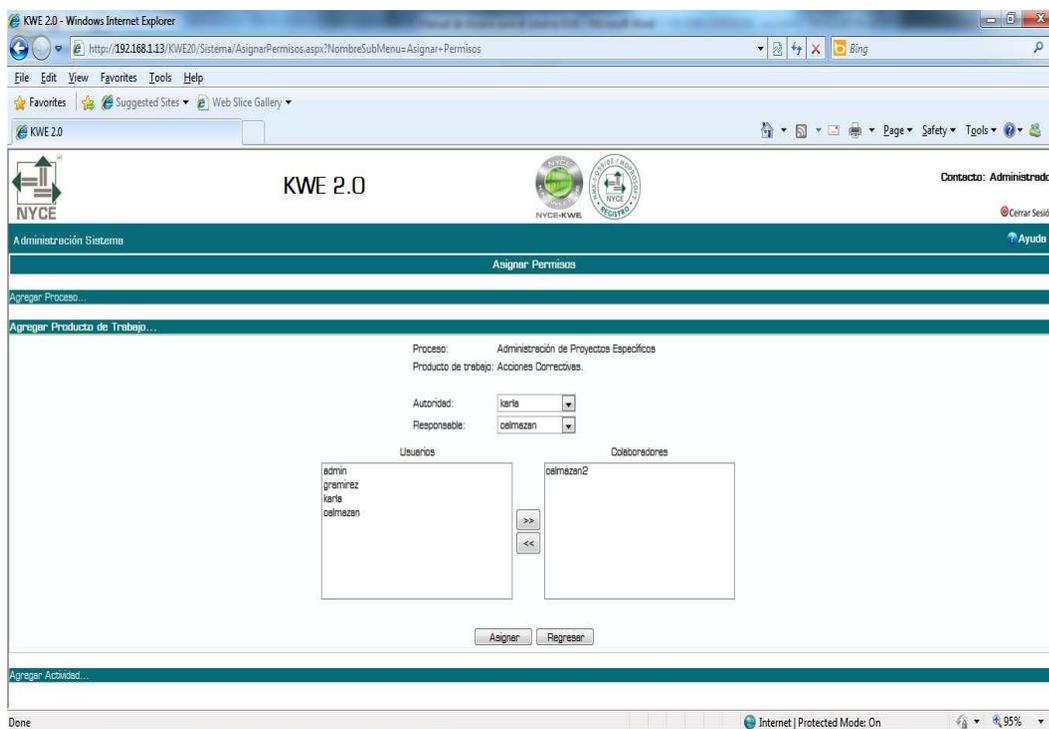


Figura 2.16. Pantalla de designación de actividades en KWE 2.0 [URL-4].

2.6. Análisis empírico sobre las herramientas de soporte a SPI

Como se presentó en los anteriores apartados, existen herramientas que ayudan en la implantación de iniciativas de mejora dentro de las organizaciones, las cuales presentan características diferentes y enfocadas a modelos diferentes.

De lo anterior se considera que una herramienta que brinde apoyo automatizado a todas las fases de un programa de mejora, es una opción viable para todas aquellas organizaciones mexicanas que intenten establecer iniciativas de SPI dentro de su estructura organizacional. Además, el hecho de que la herramienta se base en un modelo de procesos enfocado a las características de la industria de software en México proporciona un buen panorama para el desarrollo de herramientas de este tipo.

Con el objetivo de proponer una herramienta que cumpla con las características antes citadas, a continuación se presenta un *benchmarking*⁴ sobre las herramientas analizadas. En este sentido, el primer paso es definir las características o criterios bajo los cuales las herramientas analizadas han de ser comparadas.

⁴ El *benchmarking* es un anglicismo que, en las Ciencias de la Administración de Empresas, puede definirse como un proceso sistemático y continuo para evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones. Consiste en tomar “comparadores” o benchmarks a aquellos productos, servicios y procesos de trabajo que pertenezcan a organizaciones y que evidencien las mejores prácticas sobre el área de interés, con el propósito de transferir el conocimiento de las mejores prácticas y su aplicación; es decir “copiar al mejor”.

2.6.1. Definición de criterios de comparación

La definición de los criterios de comparación está basada en las características que una herramienta computarizada debe tener para auxiliar a las empresas en llevar a cabo un programa de mejora, estas características deben de dar soporte a las cuatro fases genéricas de mejora de procesos: Compromiso, Evaluación, Infraestructura y Planificación, e Implantación. En este sentido, existen investigaciones que son útiles para establecer criterios de comparación en el contexto de esta tesis. Por ejemplo, la investigación de [García et al., 2010a] ha propuesto una serie de criterios que permiten evaluar las funcionalidades de herramientas de SPI orientadas para pequeñas empresas; de manera similar, el trabajo de [Muñoz et al., 2012] ha establecido un conjunto de requerimientos que debe cubrir cualquier herramienta de software enfocada a proporcionar el soporte en MPyME durante la implementación de iniciativas de SPI.

Ambas investigaciones definen criterios para establecer el nivel de adaptación que pueden tener las empresas que conforman la industria mexicana, basados en el tamaño y necesidades de las mismas. Por último, se establecen comparadores tomando en cuenta características que una herramienta de este tipo debe tener para obtener un mejor desempeño durante la implantación de la mejora.

A continuación se listan los criterios considerados de ambas investigaciones, los cuales describen los aspectos más importantes para una herramienta de soporte a la mejora de procesos software:

- **Configuración de un equipo de trabajo:** Establecer quién o quiénes serán los involucrados en un programa de mejora es fundamental en este tipo de proyectos, ya que es necesario tener la certidumbre de quiénes y cuántos forman parte del equipo de trabajo. En primera instancia para que cada uno de los miembros del equipo adquieran el compromiso necesario para la mejora, y posteriormente delegar actividades dentro del proyecto de acuerdo a una asignación de roles.
- **Evaluación del proceso actual de la empresa:** El punto de referencia para llevar a cabo una mejora es determinar la situación actual de la organización. Con la evaluación se obtiene la situación actual de la empresa, tomando en cuenta los aspectos fuertes y débiles dentro del proceso de desarrollo software, en base a los resultados y análisis de la evaluación se generan planes de acción para mejorar el proceso actual. Es muy importante que una herramienta de soporte para iniciativas de mejora cuente con una funcionalidad de evaluación.
- **Planificación para la implantación de la mejora.** En base a los resultados de la evaluación, la fase siguiente de un modelo de mejora consiste en generar una planificación para llevar a cabo el proyecto, definiendo las actividades a realizar, los responsables de cada actividad, dependencias entre las tareas, estimación del tiempo de ejecución y calendarización de las actividades. Esta fase es muchas veces la más problemática en un proyecto de mejora, por lo que una herramienta de soporte debe de proporcionar funcionalidades para solventar las necesidades de la fase.
- **Control y monitoreo durante la implantación.** Una vez obtenidos los planes de mejora, lo siguiente es implantar el nuevo proceso dentro de la organización. Proporcionar un sistema de control y monitoreo acerca del estado de las actividades del proyecto es recomendable, ya que, mediante el sistema se puede obtener información acerca de las actividades como el estado, retrasos, problemas con actividades o miembros del equipo de trabajo; esto con el fin de realizar las acciones preventivas o correctivas para evitar desfases importantes en los tiempos planeados para el proyecto.
- **Gestión de una base de conocimiento.** Durante la ejecución del proyecto se obtienen productos y lecciones aprendidas necesarias para documentar las mejores prácticas del nuevo proceso. Por lo anterior es deseable que entre las funcionalidades del sistema se puedan

gestionar los productos generados durante el proyecto, con el fin de que los miembros del equipo de trabajo puedan acceder a la información necesaria para realizar las actividades que les han sido delegadas.

- **Enfoque Web.** Para generar un marco de colaboración entre todas las partes involucradas en el programa de mejora es necesario que la herramienta un alto grado de disponibilidad para los miembros del equipo de trabajo. Por lo anterior, un enfoque web para la herramienta es deseable, ya que, independientemente de la plataforma o lugar, las partes puedan consultar información acerca del programa de mejora, a diferencia de las restricciones propias de una aplicación de escritorio.
- **Enfocado a pequeñas organizaciones desarrolladoras de Software.** La industria mexicana de software se encuentra compuesta en su mayoría por MPyME, por lo que si se quiere brindar apoyo para el desarrollo de esta industria las soluciones deben de enfocarse en pequeñas organizaciones. Lo anterior puede satisfacerse si los métodos de evaluación e implantación de la herramienta se encuentran basados en modelos de procesos enfocados a las MPyME, como pueden ser MoProSoft o ISO/IEC 29110.
- **Tipo de licencia.** Existen muchas herramientas que pueden cumplir con todos los requisitos necesarios para brindar un marco de apoyo en iniciativas de mejora a las empresas de la industria mexicana; sin embargo, muchas de las soluciones son comerciales con un costo bastante considerable, y para las MPyME que se encuentran en un estado emergente y solidificando un capital, la adquisición de una herramienta con tales características es incosteable. Por lo cual, una herramienta de uso libre es una buena opción para aquellas organizaciones que se encuentre en etapas tempranas de incursión en la mejora de procesos software dentro de su estructura.

2.6.2. Comparación de las herramientas SPI analizadas

En base a los criterios establecidos en la sección anterior, se realizó una comparación entre las herramientas para dar soporte a iniciativas SPI analizadas en este trabajo, con el fin de obtener la información necesaria acerca de las características necesarias de una nueva herramienta para proporcionar soporte a programas de mejora en la industria software mexicana. La solución que se plantea en las siguientes secciones deberá cubrir las deficiencias encontradas en el estudio de comparación, además de incluir las ventajas más importantes que ofrecen las herramientas analizadas.

La Tabla 9 muestra la comparación de las distintas herramientas analizadas. El símbolo ‘•’ denota que la herramienta cumple con el criterio propuesto, mientras que la ausencia del mismo significa que la herramienta no cumple con el criterio, y por último el símbolo ‘?’ representa que de acuerdo al análisis de la literatura no se encontró suficiente información que permita determinar si se cumple o no el criterio.

Tabla 9. Benchmarking sobre las herramientas analizadas.

Criterios	Herramientas			
	processMax	iGrafx®	Stages for CMMI	KWE 2.0
Configuración de un equipo de mejora	•	?	?	•
Evaluación del proceso actual de la empresa	•	•	•	
Planificación para la implantación de la mejora	•	•	•	
Control y monitoreo durante la	•	•	•	

implantación				
Gestión de una base de conocimiento	•			•
Enfoque Web	•	?	•	•
Enfocado a pequeñas organizaciones desarrolladoras de Software				•
Tipo de Licencia	Comercial	Comercial	Comercial	Libre

3. Desarrollo de una herramienta colaborativa para establecer y controlar iniciativas de SPI basadas en el modelo MoProSoft

Dado que la naturaleza de Kaizen (mejora continua en japonés) es un enfoque RIA, es necesario adoptar una metodología de desarrollo que se adapte a las características de este tipo de aplicaciones. Aunque existe una cantidad importante de metodologías y herramientas que han sido propuestas para diseñar y desarrollar aplicaciones Web, la mayoría de éstas están incompletas o son inadecuadas para solventar las nuevas funcionalidades que los usuarios demandan [Preciado et al., 2005]. Debido a las nuevas funcionalidades y características que las RIA requieren, y la diferencia que representa esto con respecto a las aplicaciones Web tradicionales, se debe tener cuidado con las fases de diseño y desarrollo. De hecho, muchas de las metodologías que se acoplaban correctamente a las aplicaciones Web tradicionales, no son útiles para las RIA, así lo demuestra el estudio de [Preciado et al., 2005] que proporciona un análisis sobre por qué las metodologías existentes para el desarrollo de aplicaciones Web no son adaptables al desarrollo de aplicaciones RIA. En este sentido, las principales características de las RIA que presentan dificultades para un desarrollo mediante una metodología para Web tradicional son: la interacción, el soporte multimedia, la visibilidad continua, el método de sincronización, la colaboración interactiva, las solicitudes paralelas de diferentes fuentes y la recuperación dinámica de los datos. Debido a esta problemática, un número considerable de investigadores han desarrollado nuevas metodologías para solventar esta problemática, muchas de ellas parten de las metodologías basadas en ingeniería web⁵, las cuales han sido mejoradas o ajustadas de acuerdo a las características de las RIA para brindar un marco de desarrollo, en lugar de llevar a cabo prácticas *ad-hoc* durante el proceso de construcción de los sistemas. Un método alternativo que utiliza espacios de interacción, modelos de trabajo y máquinas de estado para diseñar una RIA, es propuesto en [Dolog y Stage, 2007]. Sin embargo, también existe una considerable cantidad de nuevos métodos que tomaron como base a los modelos existentes. En [Toffeti et al., 2007], por ejemplo, se presenta un método que se enfoca al cliente o a las acciones del servidor en aplicaciones RIA que requieren el uso intensivo de datos y de índole colaborativo, describiendo los eventos de manera explícita con WebML.

UWE-R es otro método propuesto en [Mendoza et al., 2009], que es una pequeña extensión de UWE (acrónimo en inglés para *UML-based Web Engineering*) [Koch et al., 2008] para el desarrollo de RIA, incluyendo los aspectos de navegación, procesos y presentación. Este método utiliza los estereotipos para el modelado de las extensiones en lugar de meta-atributos.

OOHRIA [Meliá et al., 2008] es una extensión del método OOH que introduce nuevos elementos de modelado y aplica nuevas transformaciones. Otro tipo de enfoque combina el

⁵ La ingeniería web se define como el proceso de creación, implantación y mantenimiento de sistemas Web de alta calidad [Murugesan et al., 1999]. Esta disciplina surgió en la década de los 90's, con el objetivo de manejar todos los aspectos del desarrollo de aplicaciones Web complejas [Dart, 2000].

modelado de aplicaciones Web con algún método de diseño para la capa de presentación en RIA. Por ejemplo, en [Preciado et al., 2008] el método UWE es completado mediante el método RUX para el diseño de la Interfaz de Usuario (IU). UWE es utilizado para especificar el contenido, la navegación y los procesos de negocio, mientras que el método RUX es utilizado en el modelo de presentación para agregar las capacidades típicas de una IU enriquecida. Sin embargo, el método de transformación resulta confuso y muchas veces no cubre algunas características de las RIA, como puede ser el marco colaborativo.

Muchos de estos enfoques proporcionan las características necesarias para modelar este tipo de aplicaciones, pero todavía siguen dejando una brecha muy marcada entre el diseño y la eficacia del desarrollo. En este sentido, en [Koch et al., 2009] se presenta un nuevo enfoque que busca plasmar las características RIA en el diseño, de hecho cabe resaltar que el grupo de investigadores que proponen este enfoque son los creadores de UWE; por esta razón toman en cuenta aspectos bastante importantes para el diseño de RIAs fundamentándose en aspectos relevantes de UWE y agregando las características necesarias para llevar por buen camino el diseño de aplicaciones Web enriquecidas. Este modelo se enfoca en integrar los patrones o características de las RIA con los métodos de modelado Web utilizados por UWE.

De acuerdo a sus creadores, este nuevo enfoque: 1) reduce los esfuerzos en diseño, manteniendo la expresividad de los modelos, y 2) contribuye al desarrollo basado en modelos RIA. En las siguientes secciones se describirán de forma más detallada las características y el uso de este enfoque.

3.1. Metodología de desarrollo

En base al objetivo general de este trabajo, y considerando la arquitectura y las características presentadas en las secciones anteriores, se define la metodología de desarrollo a seguir para la construcción de la herramienta propuesta como solución en este proyecto de tesis. El proceso de software a implementar está basado en las fases del proceso Desarrollo y Mantenimiento de Software de MoProSoft, que consta de las siguientes etapas: Inicio, Requerimientos, Análisis y Diseño, Construcción, Integración y Pruebas, y Cierre. Un enfoque ágil es adherido al proceso, debido a que, la realimentación en algunas fases del proceso es necesaria para obtener mejores resultados en el producto final y por la pequeña cantidad de personas involucradas en el proceso de desarrollo. Mediante la conjunción del proceso de desarrollo establecido por MoProSoft y un enfoque ágil, se pretende introducir dinámica y simplicidad al proceso, sin dejar de lado la eficiencia y eficacia. La metodología se divide en fases, las cuales están compuestas por un determinado número de actividades y a través de las cuales se utilizará u obtendrá un conjunto de productos claramente identificables (véase Tabla 10).

Tabla 10. Relación de productos.

Nombre	Descripción
IDEAL	Documento del modelo de mejora IDEAL.
MoProSoft	Documento del modelo de procesos MoProSoft.
EvalProSoft	Documento del método de evaluación EvalProSoft.
Reporte de SPI para MPyME	Documento que contiene las características con las que debe de contar un sistema computarizado para apoyar iniciativas de SPI en MPyME basadas en el modelo MoProSoft.
Especificación de Requerimientos	Descripción de requerimientos del software.

Nombre	Descripción
Plan de Pruebas del Sistema	Identificación de pruebas requeridas para el cumplimiento de los requerimientos especificados.
Análisis y Diseño	Contiene la descripción textual y gráfica de la estructura de los componentes software.
Plan de Incrementos	Identificación de entregas incrementales del software de acuerdo a la prioridad de cada incremento considerando su tamaño, complejidad y los recursos a utilizar.
Componente	Conjunto de unidades de código relacionadas.
Incremento	Conjunto de componentes que forman un subsistema software.
Software	Sistema de software, destinado a un usuario, constituido por componentes agrupados en incrementos, posiblemente anidados.
Reporte de Pruebas del Sistema	Registro de defectos encontrados.
Manual de Usuario	Documento electrónico o impreso que describe la forma de uso del software a través de la interfaz de usuario. Éste deberá ser redactado en términos comprensibles para los usuarios.
Manual de Mantenimiento	Documento electrónico o impreso que describe la <i>Configuración del Software</i> y el entorno utilizado para el desarrollo y las pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas).
Configuración del Software	Conjunto consistente de productos software, que incluye: <ul style="list-style-type: none"> • Especificación de Requerimientos. • Análisis y Diseño. • Software. • Manual de Usuario. • Manual de Mantenimiento. • Plan de Pruebas del Sistema. • Plan de Incrementos.

Dichas actividades serán realizadas por los roles involucrados en el desarrollo de la herramienta (véase Tabla 11). De esta forma, cada actividad identifica cómo se involucra cada rol (Participa, Autoriza, Notificado). La Tabla 12 define las actividades de cada fase, así como la función que desempeña cada rol en cada una de las acciones. La Figura 3.1 muestra en forma de diagrama de actividades la metodología de desarrollo empleada en esta tesis, identificando la interacción entre las fases en base a los productos de entrada y salida.

Tabla 11. Roles involucrados y descripción.

Rol	Abreviatura	Descripción
Revisor del Proyecto	RPY	Es el rol encargado de revisar los avances del software a desarrollar, además de brindar apuntes que ayuden a mejorar y corregir la herramienta. Durante el proyecto este rol es delegado al Director de Tesis.
Responsable de Desarrollo de Software	RDS	Encargado del proceso de desarrollo del software que realiza las actividades de Análisis, Diseño, Implementación y Entrega de la herramienta. Durante el transcurso del proyecto el Tesista cumplirá con este rol.
Usuarios Finales	UF	Es el grupo de usuarios a quien está dirigida la herramienta Kaizen. En el caso particular de esta tesis serán los jefes de proyecto de las MPyME desarrolladoras de software que participen en la experimentación.

Tabla 12. Actividades a desarrollar durante el desarrollo de Kaizen.

Descripción	Roles		
	RPY	RDS	UF
A1. Fase Inicio			
A1.1. Revisar el modelo de mejora <i>IDEAL</i> para lograr un entendimiento sobre la forma de implementar iniciativas de mejora en MPyME.		P	
A1.2. Revisar la estructura del modelo <i>MoProSoft</i> para lograr el entendimiento acerca de las características requeridas para su implantación en MPyME.		P	
A1.3. Revisar el método de evaluación <i>EvalProSoft</i> para lograr un entendimiento sobre las características de un marco de evaluación de MPyME.		P	
A1.4. Elaborar un <i>Reporte SPI para MPyME</i> registrando el análisis relacionado con los modelos <i>IDEAL</i> y <i>MoProSoft</i> , y el método <i>EvalProSoft</i> , para la implantación de iniciativas SPI en MPyMEs.	A	P	
A2. Fase de Requerimientos			
A2.1. Documentar la <i>Especificación de Requerimientos</i> .			
• Identificar nuevos requerimientos en base al <i>Reporte SPI para MPyME</i> y otras fuentes de información (p.ej., usuarios, sistemas previos, documentación, etc.).	P	P	P
• Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del proyecto de Tesis.	A	P	
• Elaborar un prototipo de interfaz.	A	P	N
• Generar la <i>Especificación de Requerimientos</i> .		P	
A2.2. Revisar la <i>Especificación de Requerimientos</i> .	A	P	
A2.3. Corregir los defectos encontrados en la <i>Especificación de Requerimientos</i> y obtener la aprobación de las correcciones.	A	P	
A2.4. Elaborar la primera parte del <i>Plan de Pruebas del Sistema</i> .		P	
A2.5. Corregir los defectos encontrados en el <i>Plan de Pruebas del Sistema</i> y obtener la aprobación sobre las correcciones realizadas.	A	P	
A2.6. Incorporar la <i>Especificación de Requerimientos</i> y el <i>Plan de Pruebas del Sistema</i> a la <i>Configuración del Software</i> .	N	P	
A3. Fase de Análisis y Diseño			
A3.1. Documentar el <i>Análisis y Diseño</i> .			
• Analizar la <i>Especificación de Requerimientos</i> para generar la arquitectura del sistema y su descomposición en subsistemas, y éstos a su vez en componentes.	N	P	
• Describir las características y necesidades tecnológicas de forma que se puedan prever los recursos para la implementación en base a la <i>Especificación de Requerimientos</i> .	N	P	
• Describir el detalle y diseño de los componentes que permita su construcción de manera evidente.	N	P	
• Generar el <i>Análisis y Diseño</i> utilizando la metodología de diseño UWE.		P	
A3.2. Revisar el <i>Análisis y Diseño</i> .	A	P	
A3.3. Corregir los defectos encontrados en el <i>Análisis y Diseño</i> y obtener la aprobación sobre las correcciones realizadas.	A	P	
A3.4. Elaborar un <i>Plan de Incrementos</i> .		P	
A3.5. Corregir los defectos encontrados en el <i>Plan de Incrementos</i> y obtener la aprobación de correcciones.	A	P	
A3.6. Incorporar el <i>Análisis y Diseño</i> y el <i>Plan de Incrementos</i> a la <i>Configuración del Software</i> .	N	P	
A4. Fase de Construcción			
A4.1. Construir cada <i>Incremento</i> de acuerdo al <i>Plan de Incrementos</i> .			
• Implementar <i>Componentes</i> con base en el <i>Análisis y Diseño</i> .	N	P	
• Definir pruebas unitarias para cada componente.		P	
• Aplicar pruebas unitarias para verificar el funcionamiento de cada componente.		P	
• Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos).		P	
• Integrar los <i>Componentes</i> en el <i>Incremento</i> .		P	

Descripción	Roles		
	RPY	RDS	UF
A4.2. Definir pruebas de aceptación para cada incremento.		P	
A4.3. Aplicar pruebas de aceptación para verificar el funcionamiento de cada incremento.	A	P	
A4.4. Corregir los defectos encontrados en los <i>Incrementos</i> hasta lograr pruebas de aceptación exitosas (sin defectos).	A	P	
A4.5. Integrar los <i>Incrementos</i> en el sistema del <i>Software</i> .	N	P	
A4.6. Incorporar <i>Componentes, Incrementos y Software</i> a la <i>Configuración del Software</i> .	N	P	
A5. Fase de Pruebas			
A5.1. Realizar pruebas del sistema siguiendo el <i>Plan de Pruebas del Sistema</i> , documentando los resultados en un <i>Reporte de Pruebas del Sistema</i> .		P	P
A5.2. Corregir los defectos encontrados en las pruebas del sistema con base en el <i>Reporte de Pruebas del Sistema</i> y obtener la aprobación sobre las correcciones realizadas.	A	P	N
A5.3. Documentar el <i>Manual de Usuario</i> .		P	
A5.4. Corregir los defectos encontrados en el <i>Manual de Usuario</i> y obtener la aprobación de correcciones.	A	P	
A5.5. Incorporar <i>Reporte de Pruebas del Sistema</i> y <i>Manual de Usuario</i> a la <i>Configuración del Software</i> .	N	P	
A6. Fase de Cierre			
A6.1. Documentar el <i>Manual de Mantenimiento</i> .		P	
A6.2. Corregir los defectos encontrados en el <i>Manual de Mantenimiento</i> y obtener la aprobación sobre las correcciones realizadas.	A	P	
A6.3. Incorporar el <i>Manuel de Mantenimiento</i> a la <i>Configuración del Software</i> y liberarlo.	N	P	

El enfoque ágil implementado dentro de la metodología tiene mayor énfasis en la fase de *Construcción*, que es en donde se realiza un desarrollo incremental. El *Plan de Incrementos* establece el orden de los incrementos y la fase de *Construcción* inicia con el primero de éstos, el cual se lleva a cabo con la construcción de los componentes que lo componen. Cada uno de estos componentes es sometido a pruebas unitarias para verificar que no presenten defectos, una vez asegurado esto, son integrados para constituir al incremento. Al igual que los componentes, los incrementos son sometidos a una serie de pruebas de aceptación para corroborar que no existen defectos y puedan ser integrados al *Software*. Al integrarse el nuevo incremento se verifica si el *Software* está completo, de ser así se termina la fase y se procede a la integración de todos los productos generados a la *Configuración del Software*, de lo contrario se realiza una nueva iteración con un nuevo incremento (véase Figura 3.2).

La implementación de un método ágil para el desarrollo del sistema, asegura la creación de módulos que cumplan con las funciones planteadas con defectos mínimos, lo que deriva en pruebas del sistema menos exhaustivas y con menor probabilidad en la detección de problemas. No obstante un plan de pruebas para el sistema es necesario con el fin de buscar posibles fallos en la ejecución global del sistema.

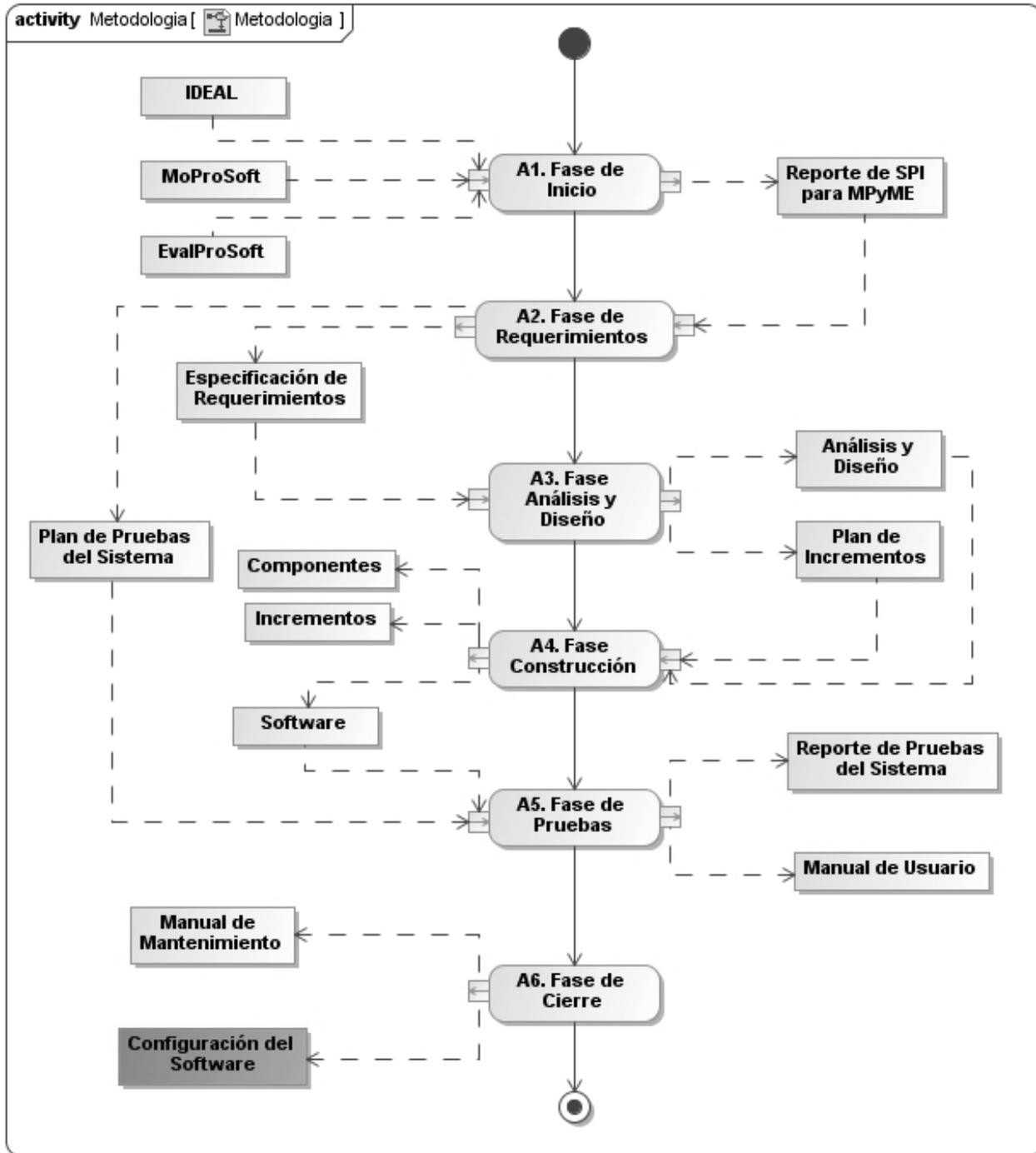


Figura 3.1. Diagrama de flujo de trabajo para la metodología de desarrollo.

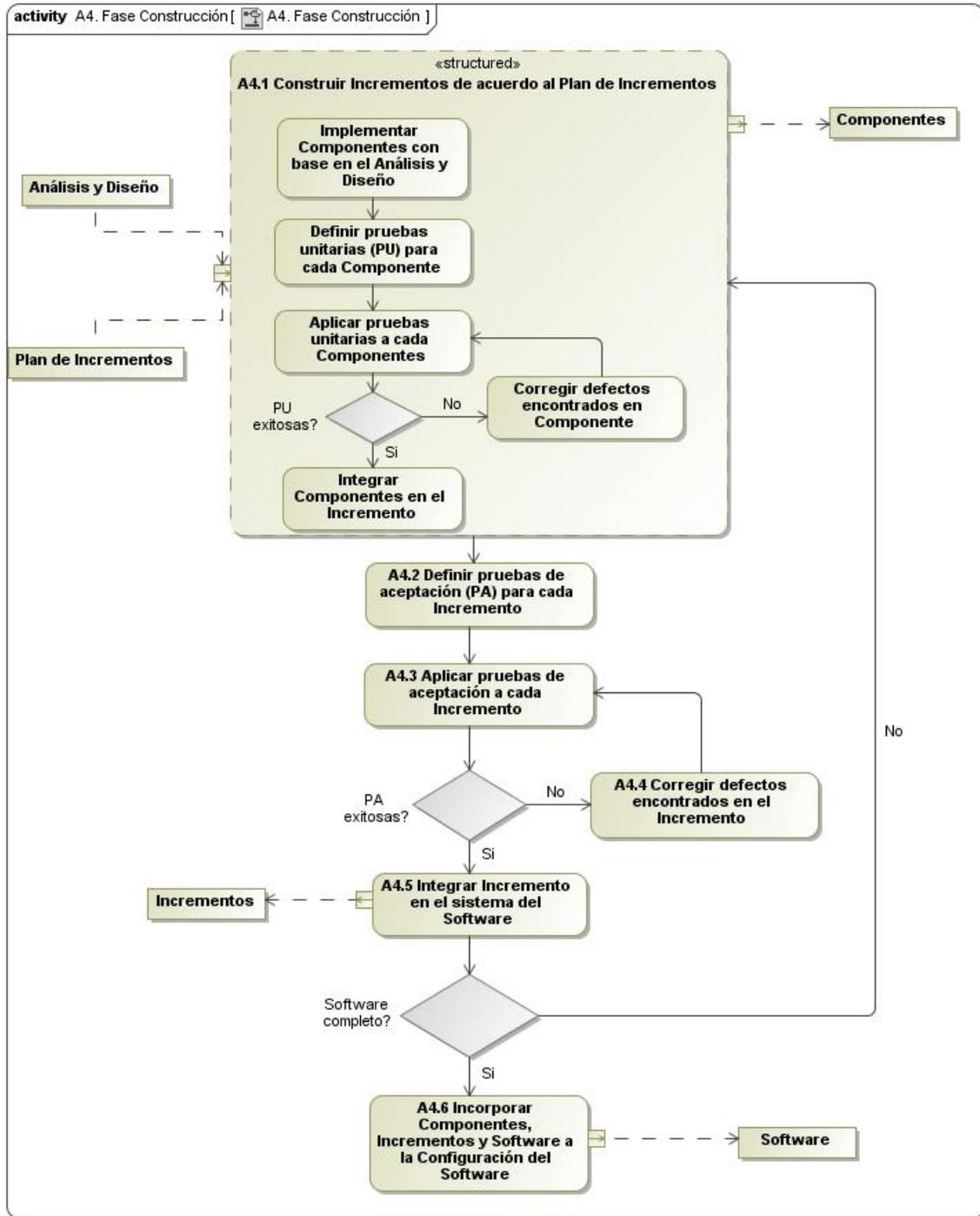


Figura 3.2. Diagrama detallado de la fase de construcción.

3.2. Especificación de requerimientos

La especificación formal de requerimientos para la herramienta se basa en las sugerencias presentadas en el estándar IEEE 830-1998 [IEEE, 1998]. Entre las ventajas de contar con una buena especificación de requerimientos destacan la reducción de esfuerzo para el desarrollo del software, una definición clara y precisa de las funcionalidades del software, y un punto de referencia sólido

para el comienzo del proceso de desarrollo. Los principales propósitos del estándar son los siguientes:

- Ayudar a describir claramente lo que se desea obtener mediante un determinado software. El cliente debe participar activamente en la especificación de requerimientos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo.
- Ayudar a los desarrolladores a entender qué se necesita exactamente. Una especificación formal de requerimientos permite definir todas las necesidades que se desea cubrir y al mismo tiempo los desarrolladores cuentan con una base fija para trabajar.
- Reducir el esfuerzo de desarrollo. Una revisión cuidadosa en la especificación de requerimientos puede revelar omisiones, malentendidos e incoherencias al principio del ciclo de desarrollo, cuando este tipo de problemas son más fáciles de corregir.

3.2.1. **Ámbito del sistema**

El sistema propuesto lleva por nombre Kaizen, “mejora continua” en japonés. El origen del nombre se debe a que la herramienta tiene como principal objetivo, implantar iniciativas de SPI y se pretende que las mejoras sean continuas para la obtención de resultados satisfactorios. Kaizen proporciona apoyo a las MPyME desarrolladoras de software interesadas en mejorar sus procesos de desarrollo y producción de sistemas. A través de un entorno basado en el modelo MoProSoft, el sistema brinda las directrices, actividades y tareas a realizar para cumplir con las sugerencias que el modelo propone, de esta manera se pretende que la empresa mejore sus procesos, convirtiéndolos en procesos capaces y maduros, que tengan como consecuencia la creación de productos software de mejor calidad.

A continuación se enlistan las metas que se pretenden alcanzar mediante el uso de la herramienta Kaizen:

- Auxiliar a las MPyME desarrolladoras de software sin experiencia en la mejora de procesos, en la implantación de iniciativas de SPI dentro de su organización.
- Proporcionar un método automatizado y computarizado para guiar iniciativas de SPI dentro de las estructuras de pequeñas organizaciones desarrolladoras de software.
- Controlar y gestionar iniciativas de SPI dentro de la organización, para que el personal pertinente sea informado acerca de los avances o rezagos que se presenten en la iniciativa.
- Brindar un marco de trabajo para el aprendizaje e institucionalización de la cultura de mejora de procesos de software.
- Fomentar dentro de las empresas una cultura de mejora continua que les ayude a establecer procesos maduros de desarrollo que les permitan producir software de calidad.

Kaizen busca brindar los siguientes beneficios a las MPyME software que la utilicen:

- Proporcionar un mecanismo de evaluación para el proceso de software de la empresa, con el fin de conocer el estado actual de los procesos de desarrollo, indicando las fases, actividades o tareas en las que se debe de enfocar la mejora.
- En base a los resultados obtenidos en la evaluación, Kaizen puede generar un plan de acción de mejora, en donde se especifica la secuencia, orden y encargados de las actividades a realizar para poner en marcha el proyecto de mejora dentro de la estructura de la empresa.

- Controlar y guiar a las estructuras de la empresa para llevar por buen camino la iniciativa de mejora. Con Kaizen los encargados de la iniciativa de SPI, pueden monitorear las actividades realizadas como parte de la iniciativa de mejora, controlar los tiempos de ejecución, así como detectar retrasos en el desarrollo de la iniciativa, y de esta manera realizar acciones correctivas para superar ese tipo de inconvenientes.

3.2.2. Descripción General

En primera instancia, debido a las características necesarias para el desarrollo de una herramienta que auxilie a la implantación de iniciativas de SPI dentro de pequeñas organizaciones, es necesario un estudio referente a las tecnologías viables para satisfacer las necesidades para el diseño y construcción de la aplicación. En primera instancia, la definición de características básicas brinda la pauta para la elección de las tecnologías a utilizar durante el ciclo de desarrollo, en base a esta elección, el siguiente paso es la adopción de una metodología acorde a las características de la tecnología, que marque las directrices para llevar por buen camino el desarrollo del sistema.

La principal característica a destacar es el uso de un entorno Web para hacer uso de la herramienta, esto con la finalidad de que esté disponible desde cualquier lugar que cuente con una computadora y acceso a Internet, ya que, esto brindará un marco de trabajo independiente de terceros entre la MPyME desarrolladora de software y la herramienta. La Tabla 13 muestra las características básicas con las que debe de contar la herramienta Kaizen, y que por consiguiente debe cumplir la tecnología (o grupo de tecnologías) de desarrollo.

Tabla 13. Características básicas de la herramienta Kaizen.

Característica	Descripción
C1. Trabajar sobre un entorno Web	Al tratarse de una herramienta que busca ser independiente de terceros, para no entorpecer el trabajo de una organización que pretende implantar iniciativas de SPI dentro de su estructura organizacional, la herramienta debe de estar disponible en cualquier lugar y a cualquier hora, para que de esta forma los interesados puedan acceder a ella contando con una conexión a Internet, además de buscar con esto, una mayor propagación del uso de Kaizen, debido a la masiva utilización de este medio.
C2. Independiente del Sistema Operativo (SO) de trabajo	Tomando en cuenta que las empresas de software utilizan diversos SO's en sus equipos de trabajo, es necesario contar con una plataforma en donde se pueda acceder a la herramienta independientemente del SO que la empresa maneje.
C3. Capacidad para el desarrollo de aplicaciones robustas	La cantidad de funciones que conformarán a Kaizen, sugiere el desarrollo de una cantidad considerable de módulos y componentes, por lo que, la tecnología de desarrollo debe soportar el desarrollo de sistemas robustos y de gran complejidad.
C4. Soporte para el desarrollo de aplicaciones altamente interactivas	Debido al enfoque que se pretende para la herramienta, la interacción con los usuarios, marca una característica importante para la tecnología de desarrollo, por lo cual ésta debe ser capaz de generar nuevos componentes y permitir una alta interacción interfaz-usuario.
C5. Metodologías de desarrollo	Aunado a las características técnicas expuestas con anterioridad, se debe contar con una serie de directrices que lleven por buen camino el desarrollo de la herramienta, para esto se debe de echar mano de una metodología de desarrollo específica para la tecnología elegida, o en su defecto adaptar una ya existente que se acople a las características de dicha tecnología.
C6. Soporte para el desarrollo colaborativo	El enfoque colaborativo pretende la interacción y colaboración de usuarios y servicios para conseguir objetivos y tareas compartidas, sobre la misma plataforma computacional en tiempo real. Por lo anterior, es primordial que la tecnología utilizada para el desarrollo, brinde las bases y técnicas para la

Característica	Descripción
	construcción de este tipo de aplicaciones.
C7. Almacenamiento y acceso a los datos generados	Debido al uso de Kaizen por parte de diferentes organizaciones, los datos e información generados tendrán que ser almacenados para su posterior consulta. Por lo anterior, la herramienta debe contar con las características que permitan el almacenamiento y acceso a la información.
C8. Herramientas de desarrollo a bajo costo	El uso libre de Kaizen por parte de cualquier empresa que desee utilizarla, hace que los costos de desarrollo de la misma sean nulos o muy bajos, es por esto, que la tecnología de desarrollo debe cumplir con dicha particularidad.

3.2.3. Requerimientos de Kaizen

En este apartado se describen las funciones principales que el software realiza. Para el mejor entendimiento de dichas funciones, se presentan descripciones textuales y gráficas. Para modelar cada una de las unidades funcionales de Kaizen se utiliza la técnica de casos de uso; en primera instancia realizando una descripción detallada de cada uno de los casos de uso, aunado a la descripción se presenta el diagrama correspondiente.

Para presentar el procesamiento de datos del sistema, se utilizan los Diagramas de Flujo de Datos (DFD). Los DFD representan gráficamente la interacción entre entidades externas y la herramienta Kaizen, el flujo de datos entre los procesos que conforman el sistema y el almacenamiento de los datos procesados y generados. La notación de los casos de uso proporciona la noción para crear los DFD. Cada caso de uso define una actividad y su detonador, entrada y salida; lo que permite comprender la naturaleza de los procesos y actividades para crear un DFD. Con la creación de la descripción y modelado de los casos de uso, y la representación del flujo de datos por medio de los DFD, se pretende que la presentación de los requerimientos de la herramienta Kaizen sea clara y fácil de comprender. La Figura 3.3 presenta de manera general las funcionalidades del sistema Kaizen, las cuales buscan cubrir la mayoría de las actividades de una iniciativa de mejora de procesos. El gráfico también muestra la manera en que los diferentes involucrados en la mejora interactúan entre sí, y la secuencia de actividades por las que se guía a los implicados por medio de Kaizen para lograr completar un ciclo de mejora.

3.2.3.1. Definición de casos de uso

Los casos de uso modelan un sistema desde el punto de vista del usuario, es decir son una técnica basada en escenarios para la obtención de requerimientos. Los casos de uso deben de cumplir con los siguientes objetivos [Pressman, 2009]:

- Definir los requerimientos funcionales y operativos del sistema, proporcionando una descripción clara y sin ambigüedades de cómo el usuario final interactúa con el sistema y viceversa.
- Proporcionar una base para la validación de las pruebas.

El conjunto de casos de uso modela todas las posibles interacciones a representar en los requerimientos del sistema. Los escenarios y los casos de uso son técnicas eficaces para obtener requerimientos desde los puntos de vista de aquellos que interactúan con el sistema, donde cada tipo de interacción se puede representar como un caso de uso. Por lo anterior, dado que el sistema se desarrolla bajo un paradigma orientado a objetos, los casos de uso son la técnica adecuada para la obtención de requerimientos, permitiendo así identificar información para el posterior diseño conceptual de la herramienta Kaizen.

Así, los casos de uso para el sistema Kaizen son documentados de manera detallada, cada uno vinculado con un modelo, el cual sigue la notación del Lenguaje Unificado de Modelado (UML) para su representación.

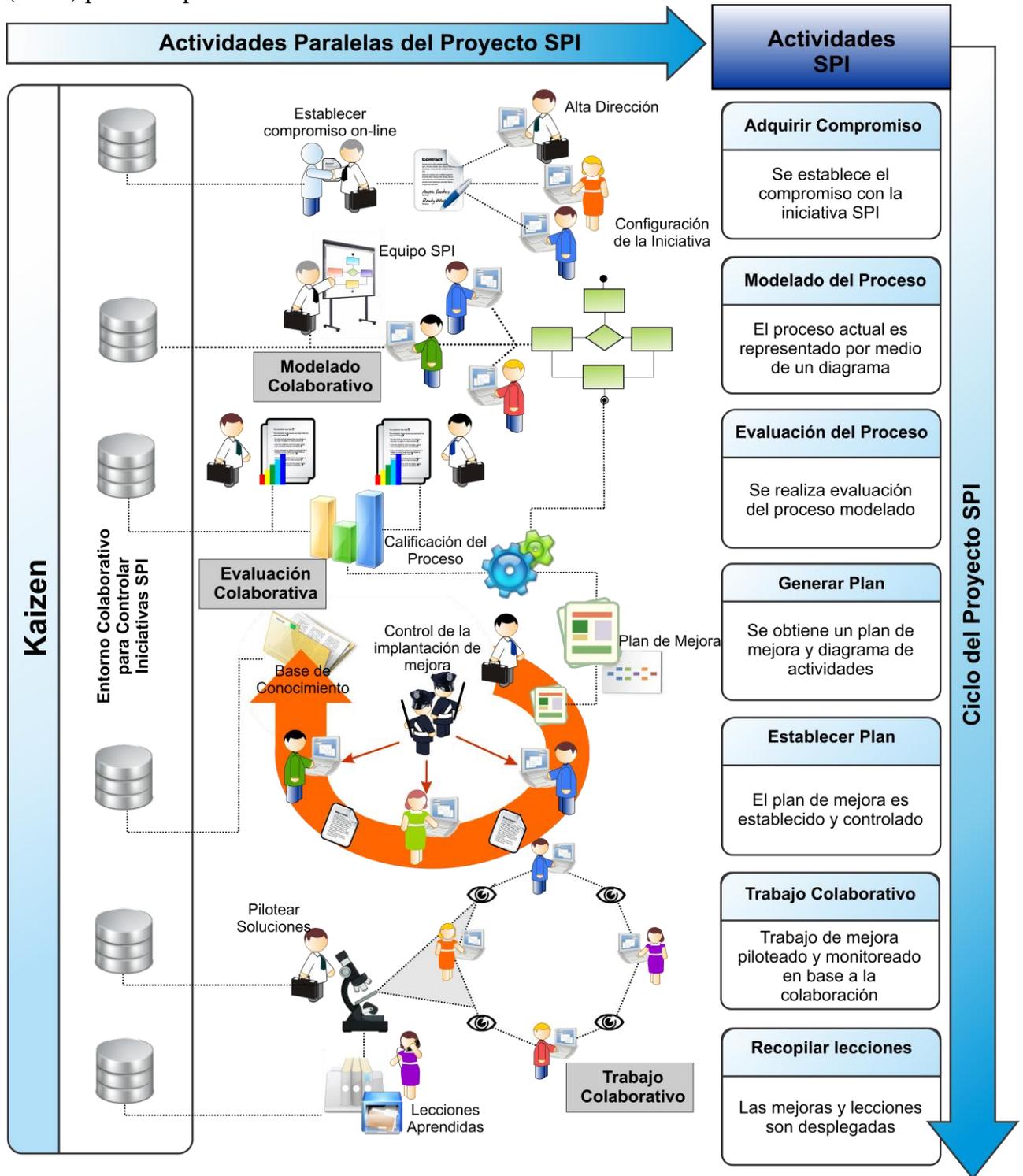


Figura 3.3. Plataforma colaborativa para la implantación de iniciativas de SPI en MPyMEs.

En primera instancia se definen los actores que tienen interacción con el sistema Kaizen. Un actor es una agrupación de uniforme de personas, sistemas o máquinas que interactúan con el sistema, los actores son externos al sistema. Los actores que interactúan con Kaizen son tres:

Administrador, Líder de Proyecto y Miembro del Equipo. La elección de actores toma en cuenta el hecho de que un actor es un rol dentro del sistema, y un usuario es una persona que, cuando usa el sistema, asume un rol. De esta forma, un usuario puede acceder al sistema como distintos actores. Todos los actores mencionados deben ser personas, debido a que Kaizen no interactúa con algún sistema externo. De la Tabla 14 a la número 16, se describe de forma más detallada a cada uno de los actores.

Tabla 14. Descripción del actor: Administrador.

Nombre:	Administrador
Descripción:	Este tipo de actor cuenta con todos los permisos del sistema Kaizen, por consecuencia puede acceder a la información guardada y generada por el sistema. Entre las funciones principales se encuentran el registro de empresas que deseen utilizar el sistema y de un líder de proyecto que sea el administrador del proyecto de una empresa previamente registrada en Kaizen. Además, este rol tiene la posibilidad de acceder a las estadísticas del sistema: número de empresas registradas, usuarios registrados, proyectos en desarrollo, proyectos culminados, entre otras.

Tabla 15. Descripción del actor: Líder de Proyecto.

Nombre:	Líder de Proyecto
Descripción:	Es el encargado de poner en marcha un proyecto de mejora de la empresa por medio de Kaizen, además se desenvuelve como administrador en el ámbito de la empresa a la que pertenece. Entre las actividades que realiza con el sistema se encuentran: el registro de los miembros del equipo, establecer las propiedades del proyecto de mejora próximo a desarrollarse, monitorear las actividades de los miembros del equipo, consulta del avance del proyecto, entre otras.

Tabla 16. Descripción del actor: Miembro de Equipo.

Nombre:	Miembro de Equipo
Descripción:	Cuenta con los permisos mínimos para utilizar el sistema, debido a que solamente puede acceder a las funcionalidades propias de las tareas que le sean delegadas por el jefe de proyecto. Este rol participa en la fase de evaluación e implantación en el proyecto de mejora. Sus actividades dentro de Kaizen se enfocan principalmente a la evaluación del proceso de desarrollo de la empresa, implantación de tareas específicas del proyecto de mejora, registro de documentación, entre otros.

La Figura 3.4 muestra la relación que existe entre cada uno de los actores antes descritos. En ella se puede observar que Jefe de Proyecto es un tipo específico de Miembro de Equipo. Por último el Administrador del sistema no mantiene ningún tipo de relación de generalización con ninguno de los otros actores.

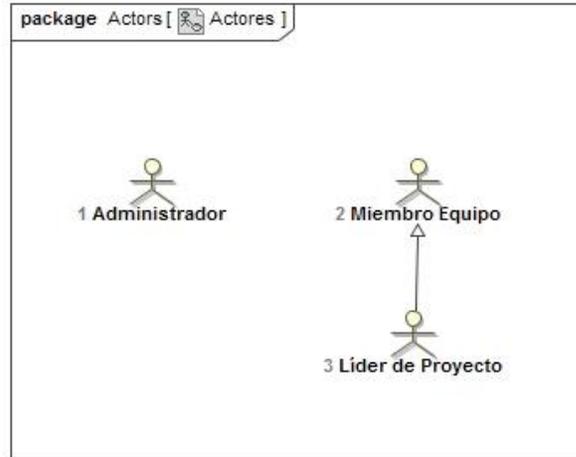


Figura 3.4. Relación entre actores.

En base a los actores descritos se realiza la clasificación de los casos de uso del sistema Kaizen obtenidos al considerar las características y funcionalidades del sistema descritas en los apartados anteriores. La Figura 3.5 presenta el diagrama de casos de uso para el Administrador. En el Anexo C del presente documento se presenta a detalle la especificación de casos de uso obtenidos para el sistema Kaizen.

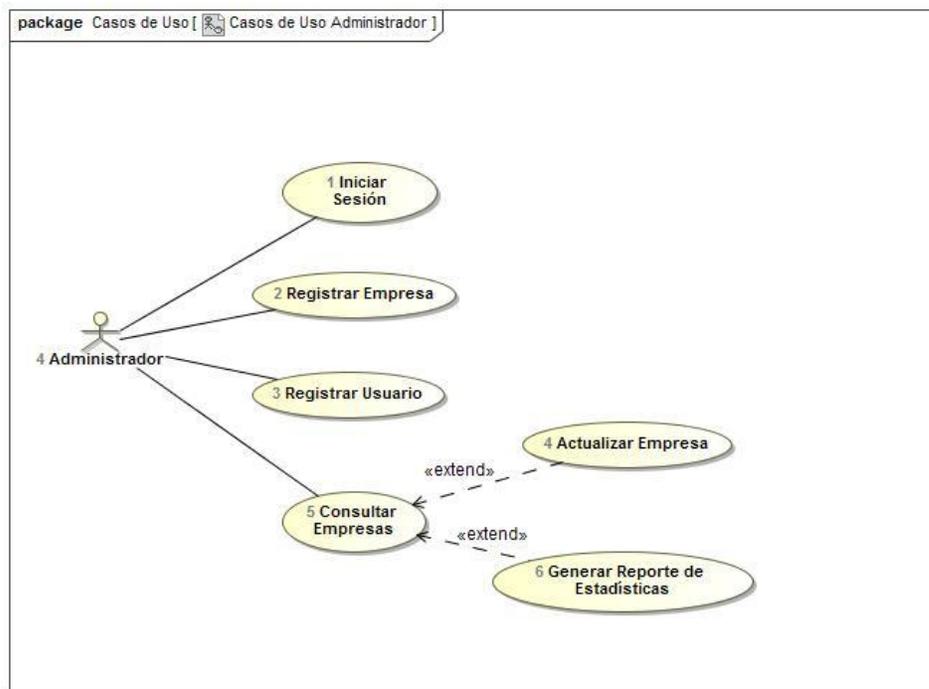


Figura 3.5. Diagrama de casos de uso Administrador.

3.2.3.2. Diagrama de flujo de datos

Los Diagramas de Flujo de Datos (DFD) son una técnica de análisis estructural, la cual proporciona una representación gráfica de los procesos y su relación con los datos en el sistema. El enfoque del flujo de datos posee cuatro ventajas principales sobre las explicaciones descriptivas en la fase de requisitos [Kendall y Kendall, 2011]:

1. Libertad para emprender la implementación técnica del sistema en las etapas tempranas.
2. Una comprensión más profunda de la interrelación entre sistemas y subsistemas.
3. Comunicar a los usuarios el conocimiento sobre el sistema actual mediante DFD.
4. Análisis de un sistema propuesto para determinar si se han definido los datos y procesos necesarios.

Por lo anterior, se ha optado por elegir la técnica DFD para enriquecer la obtención de los requisitos de Kaizen. Así, durante la primera parte de la fase se obtienen los casos de uso y sus respectivos diagramas de actividades, en la segunda parte y en base a los casos de uso obtenidos, se desarrollan los DFD para entender y obtener de manera más clara la relación de los procesos del sistema, la interacción de éstos con diferentes entidades externas y el flujo, transformación, consulta y almacenamiento de los datos. Mediante la unión de los casos de uso y los DFD, se pretende obtener de forma más clara y precisa los requisitos del sistema.

El diagrama de contexto de la Figura 3.6 representa a todo el sistema. El diagrama de contexto es el nivel más alto en un DFD y contiene un solo proceso, que representa a todo el sistema. En el diagrama se muestran todas las entidades externas, así como también los flujos de datos principales que van desde y hacia dichas entidades.

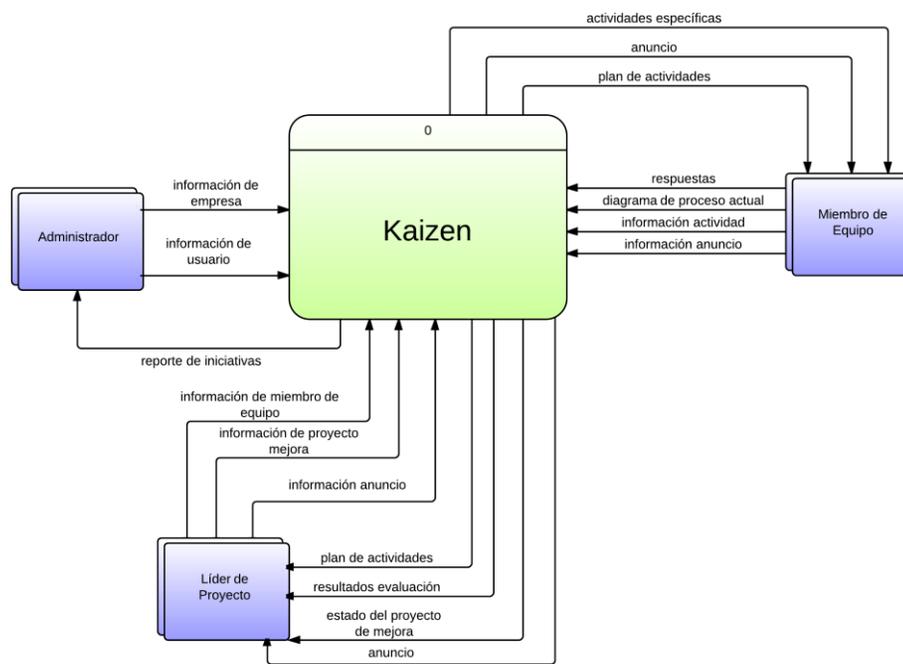


Figura 3.6. Diagrama de contexto de Kaizen.

Cabe destacar que como se ha mencionado en la definición de actores del sistema, el Líder de Proyecto al ser un tipo específico de Miembro de Equipo hereda las actividades que puede realizar el mismo. En el diagrama figuran tres entidades externas que se refieren a los tipos de usuarios que pueden utilizar el sistema, debido a que Kaizen no interactúa con sistemas o subsistemas externos, no se definen entidades de este tipo. El diagrama muestra los flujos de datos principales de las entidades hacia Kaizen y viceversa, los flujos de datos secundarios o específicos se definen en los siguientes niveles del DFD. Al ampliar los diagramas se puede lograr un mayor detalle que con los diagramas de contexto. El Diagrama 0, por ejemplo, es la ampliación del diagrama de contexto, en donde se representan los procesos que componen al sistema, se muestran los almacenes de datos y nuevos flujos de datos de menor nivel. El efecto es similar al de tomar una

lupa para el diagrama de flujo de datos de menor nivel [Kendall y Kendall, 2011]. El Diagrama 0 para Kaizen se muestra en la Figura 3.7.

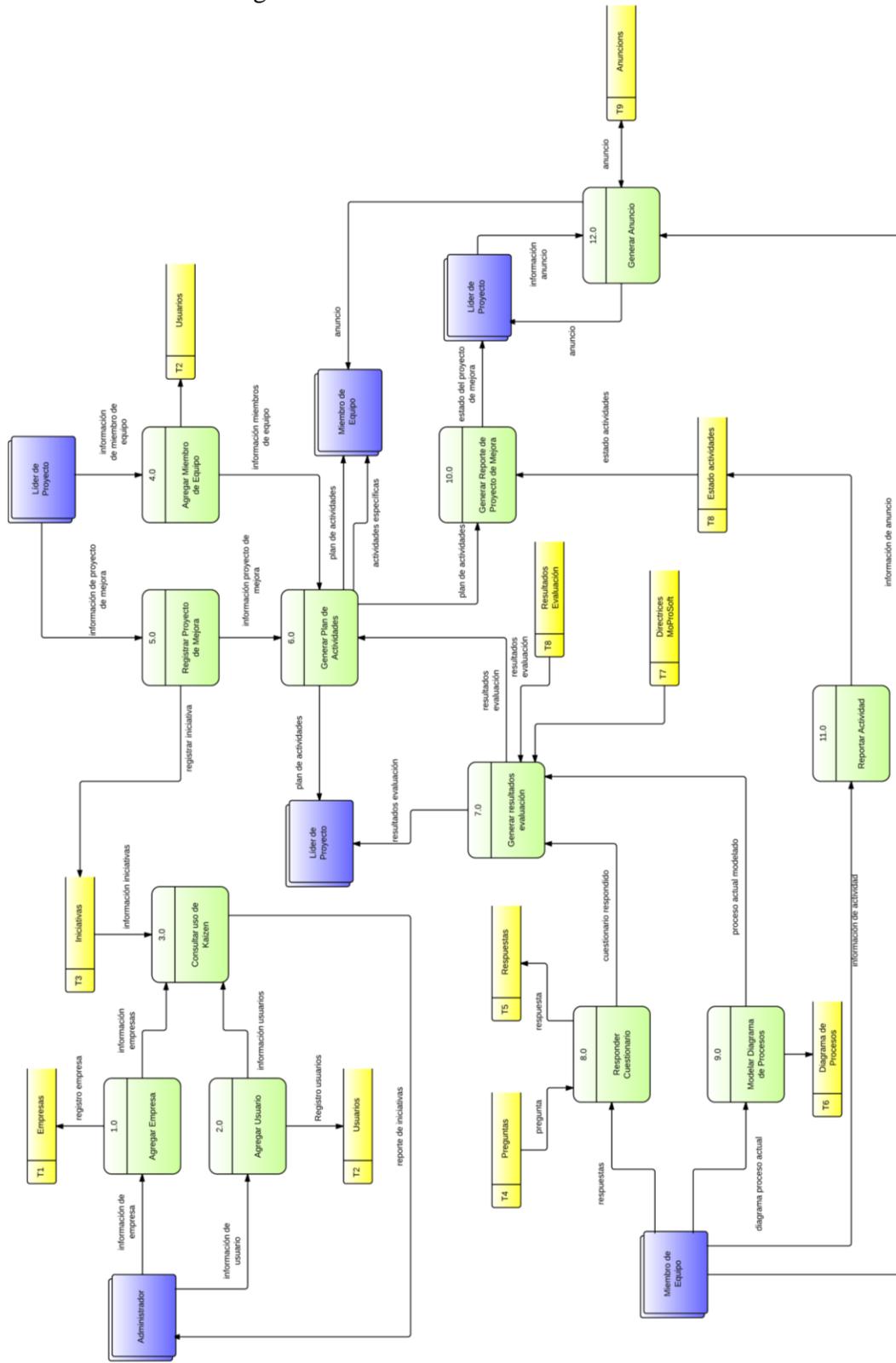


Figura 3.7. Diagrama 0 para el sistema Kaizen.

3.3. Análisis y diseño del sistema

En base a la *Especificación de Requisitos del Software* obtenida en la sección anterior, el paso siguiente de este proyecto es el análisis de los requerimientos para generar un diseño que sirva como base en el desarrollo de Kaizen. Dado que una de sus principales características es el ámbito Web, el análisis y diseño de Kaizen requiere una metodología enfocada a los sistemas Web, y en particular para los sistemas RIA (*Rich Internet Applications*) debido a sus características. Las RIA permiten construir aplicaciones ricas en datos y contenido multimedia, con alta interactividad, lo que aumenta las capacidades que ofrecen las aplicaciones Web tradicionales. El modelado de las RIA requiere técnicas de modelado Web enriquecidas por diferentes elementos, que modelen de forma correcta la interacción con el usuario y la comunicación cliente-servidor, principales características de un sistema RIA. Debido a estas y otras características, los enfoques tradicionales no se adaptan completamente al modelado de las RIA, y a pesar de que existen metodologías de análisis y diseño para sistemas Web que se pueden adaptar, éstas derivan en amplias brechas y vacíos en el diseño del sistema RIA, ya que muchas de las características propias de las aplicaciones enriquecidas de Internet no se toman en cuenta en la parte de diseño.

En este sentido, para el sistema Kaizen se utiliza la metodología propuesta en [Koch et al., 2009]. La metodología se basa en patrones propios de un sistema RIA, combinados con un método existente de modelado Web, en este caso, se utiliza UWE [Koch et al., 2008]. En las siguientes secciones se presenta una descripción más detallada del método y a continuación el diseño obtenido para Kaizen en base al método anteriormente descrito.

3.3.1. Método para el diseño y creación de modelos para aplicaciones RIA

Las RIA son aplicaciones Web aumentadas con las características de una aplicación de escritorio y mecanismos avanzados de comunicación [Goralski y Leon, 2008]. Las RIA mejoran la interacción con el usuario, añadiendo características como arrastrar y soltar (*drag & drop*), presentaciones multimedia y evitar recarga de páginas innecesarias mediante el manejo de datos y operaciones del lado del cliente que minimizan las solicitudes al servidor. Así, existe una cantidad importante de *frameworks*⁶, tales como AJAX, Flex, Ruby on Rails, entre otros, que ayudan al desarrollo de sistemas RIA; sin embargo, el desarrollo de software requiere también del soporte para otras fases como el diseño o las pruebas. En este sentido, diferentes métodos de modelado para aplicaciones Web tradicionales han sido utilizados para llevar a cabo el diseño de las RIA, sin embargo, sigue existiendo una brecha importante entre el diseño y características importantes, conocidas como patrones, que se pueden utilizar en el momento de la construcción del sistema. Los patrones han demostrado ser muy valiosos para una eficiente programación de RIA. En la propuesta presentada en [Koch et al., 2009] se toman en cuenta los patrones de una RIA y se trabaja con un método de modelado para aplicaciones Web, como lo es UWE. El método se enfoca en el uso de máquinas de estado para la representación de los patrones. El uso de estos patrones RIA solamente requiere la definición de puntos de extensión en la metodología y la posterior integración de tales patrones en el método UWE.

3.3.1.1. Patrones RIA

Un patrón RIA es una solución general reutilizable para una situación que ocurre con frecuencia en una aplicación de este tipo. En él se describe la interacción, el funcionamiento y la

⁶ En el desarrollo de software, un *framework* es una estructura conceptual y tecnológica de soporte, definido normalmente por artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser fácilmente organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

presentación de un *widget*⁷ RIA [Scott, 2007]. La interacción es el detonador de un patrón RIA, todos los patrones RIA comienzan con un evento de usuario o del sistema, por ejemplo un evento de tiempo, deslizamiento del ratón por un componente, oprimir una tecla, cambiar el foco de atención entre componentes. Una gran cantidad de operaciones pueden ser inicializadas por medio de estas interacciones, como una validación, una búsqueda o una actualización. Finalmente, el resultado de la operación implica una actualización de la interfaz de usuario. La anatomía de un patrón RIA está dada por:

- **Interacción:** Todos los patrones comienzan con una interacción como un evento de usuario o un evento del sistema. Algunos patrones representan alguna forma de interacción.
- **Operación:** Es comenzada por una interacción. Las operaciones se pueden clasificar en operaciones de búsqueda, para obtener información cuando se necesite, operaciones de acción para llevar a cabo una tarea en el contexto del sistema, operaciones de validación para la prevención de errores de forma temprana y operaciones de mensaje para comunicar algo instantáneamente.
- **Presentación:** La interfaz de usuario muestra el resultado, por ejemplo, agregar datos, cambios en la estructura de la página o una animación.

3.3.1.2. Ingeniería Web basada en UML (UWE)

UWE (acrónimo en inglés para *UML-based Web Engineering*) es un método sistemático para guiar el desarrollo de aplicaciones Web mediante el modelado. El principio de UWE es la separación de los componentes de la aplicación mediante el modelado del contenido, la estructura de la navegación, los procesos de negocio, y la presentación de la aplicación Web [Koch et al., 2008]. El modelo conceptual de la aplicación es usado como la pauta para modelar el modelo de navegación. El modelo de navegación es el siguiente en realizarse, el cual muestra cómo navegar por el espacio de navegación de la herramienta utilizando elementos como acceso a índices, consultas y menús. Por último, el modelo de presentación es construido basado en el modelo de navegación.

El método UWE proporciona una orientación sistemática para la construcción de sistemas Web. La Figura 3.8 muestra los modelos representados por paquetes UML y la relación que existe entre ellos. El objetivo del modelo de casos de uso es realizar el análisis de requisitos, para encontrar los requerimientos funcionales de la aplicación Web y representar estos requisitos mediante diagramas de casos de uso (véase Anexo C del presente documento).

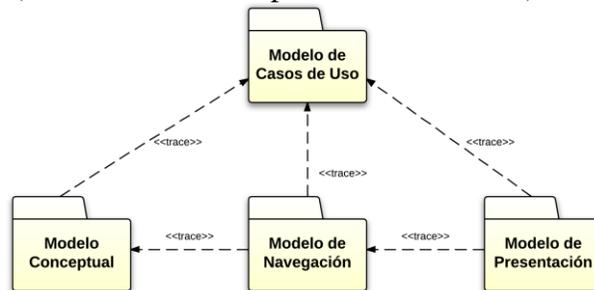


Figura 3.8. Relación de modelos en el método UWE [Koch et al., 2008].

⁷ En informática, un *widget* es una pequeña aplicación o programa que son incluidos en sistemas de mayor complejidad. Entre sus objetivos están dar fácil acceso a funciones frecuentemente utilizadas y proporcionar información visual. Los programadores suelen utilizarlos para construir interfaces gráficas enriquecidas de usuario.

El diseño conceptual se enfoca en la construcción de un modelo conceptual del dominio de la aplicación, tomando en cuenta los requerimientos obtenidos por el modelo de casos de uso. La creación de clases, asociaciones y la definición de estructuras de herencia, son técnicas utilizadas para la creación de este modelo. El modelo es representado mediante un diagrama de clases UML. Tomando como base el modelo conceptual, el modelo de navegación propone un conjunto de directrices para conseguir un marco de navegación para representar el espacio y estructura de navegación del sistema Web. El método incluye un conjunto de elementos propios de UWE, los cuales se pueden presentar como elementos UML estereotipados para el diseño de la navegación, como índices, visitas guiadas, consultas a la base de datos y menús. Estos estereotipos se utilizan para representar el modelo de navegación de una forma clara y sencilla. Por último el modelo de presentación presenta el diseño de interfaces de usuario abstractas y el diseño de la interacción del usuario con la aplicación Web [Koch et al., 2008].

3.3.1.3. *Los patrones RIA en el diseño con UWE*

Los patrones RIA son ideales para el diseño y posterior construcción de sistemas basados en este tipo de tecnología, debido a que con ellos se consigue una mejor especificación del comportamiento de los sistemas. La especificación de estos comportamientos, propios de un sistema RIA, se denota por medio de etiquetas en los elementos UWE de cada modelo. Un comportamiento como autocompletado, validación en tiempo real y actualización en un determinado periodo de tiempo son denotados por medio de una etiqueta en elementos del método UWE [Koch et al., 2009]. La Figura 3.9 muestra el metamodelo para proporcionar una visión general de todos los patrones y atributos definidos en UWE para los patrones RIA (marcados con óvalos). También se muestra la incorporación de los patrones RIA en UWE. La Tabla 17 muestra las etiquetas para los patrones definidos y una descripción de cada uno [Scott, 2007].

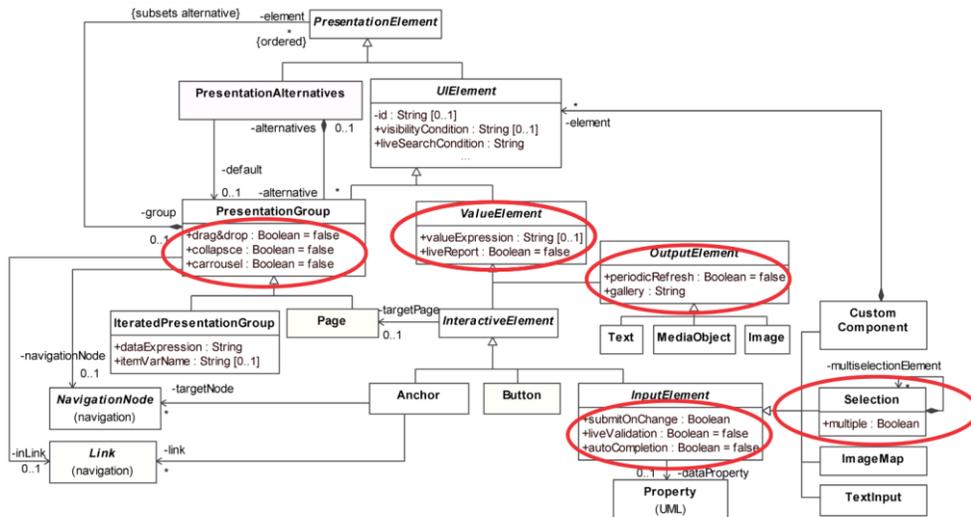


Figura 3.9. Metamodelo para la capa de presentación de UWE [Koch et al., 2009].

Tabla 17. Descripción de patrones RIA especificados en UWE [Scott, 2007].

Patrón	Etiqueta UWE	Uso
Autocompletado	<i>autoCompletion</i>	Cuando el usuario necesita ingresar un dato dentro de una caja de texto que puede ser ambiguo o difícil de recordar y por lo tanto puede estar mal, si éste fuera el caso el autocompletado puede ser utilizado.

Patrón	Etiqueta UWE	Uso
Sugerencias Automáticas	<i>autoSuggestions</i>	Cuando es necesario especificarle al usuario algo que sucederá si hace clic en un determinado objeto, mostrar información acerca del formato de un dato a ingresar o invitar al usuario para interactuar con una característica del sistema, si éste fuera el caso las sugerencias automáticas pueden ser utilizadas.
Colapsar elementos	<i>Collapse</i>	Cuando se quieran desplegar o mostrar elementos de la pantalla, el patrón colapsar ayuda a mostrar u ocultar elementos, según sea el caso, en el contexto de una pantalla mediante transiciones agradables al usuario.
Arrastrar y soltar	<i>dragDrop</i>	Si el usuario necesita una manera de realizar operaciones sobre uno o más objetos mediante el desplazamiento de objetos de un lugar a otro directamente con el ratón.
Mostrado Dinámica	<i>dynamicDisplay</i>	El patrón de mostrado dinámico actúa cuando se quiere cambiar la estructura de la pantalla de interfaz gráfica de forma dinámica, es decir, cuando en base a acciones del usuario la estructura de la interfaz gráfica debe cambiar para mostrar o eliminar elementos.
Filtros	<i>filter</i>	En caso de existir la necesidad de mostrar un bloque de información en base a un criterio, y sin la necesidad de hacer una consulta a la base de datos, el patrón filtro se ocupa sobre una tabla de información, previamente añadida, para realizar búsquedas en tiempo real y rápidas.
Galería	<i>gallery</i>	Un patrón RIA común son las galerías de imágenes, debido a que no existe la necesidad de agregar código complejo y extenso, solamente es necesario utilizar este patrón a los sistemas.
Cuadro de diálogo con transparencia	<i>lightbox</i>	Si el sistema necesita mostrar un mensaje, formulario o un determinado diálogo en base a una acción ejecutada en el sistema, la aparición de un cuadro de diálogo que oculte la pantalla principal es la mejor opción.
Retroalimentación en tiempo real	<i>liveFeedback</i>	La necesidad de conocer el estado de una tarea, el progreso en una cierta actividad o estado de un formulario, sin necesidad de realizar algún tipo de petición al servidor, puede ser abatida con el uso de la retroalimentación en tiempo real.
Reporte en tiempo real	<i>liveReport</i>	En base a la generación de información por parte del usuario o sistema, la generación de reportes en tiempo real es de gran ayuda, debido a que sin la necesidad de realizar peticiones al servidor, se pueden generar reportes con ayuda de gráficas y otros elementos RIA.
Validación en tiempo real	<i>liveValidation</i>	Se utiliza para prevenir errores de entrada de forma temprana. En caso de presentarse errores en la entrada, el usuario puede reaccionar y corregir el error de forma fácil y en el formulario de entrada actual.
Control múltiple	<i>multiple</i>	En diversas tareas es necesario contar con controles múltiples, por ejemplo el clic del ratón, arrastrar y soltar, presión de una tecla y otros eventos, con el patrón control múltiple de las RIA, se denota que elemento contará con múltiples acciones de control.

Patrón	Etiqueta UWE	Uso
Actualización de contenido periódica	<i>periodicRefresh</i>	En caso de ser necesario mostrar información actualizada en base a periodos de tiempo o resultados del sistema, se utiliza el patrón de actualización de contenido periódica.
Editor de texto enriquecido	<i>richEditor</i>	En el desarrollo de sistemas Web, en ocasiones surge la necesidad de editar texto en formato enriquecido (aplicar color, tamaño de fuente, alineación, entre otros). Este patrón RIA permite realizar la tarea anterior de forma rápida y sencilla.
Cambio al enviar	<i>submitchange</i>	Después de aceptar una tarea o enviar un formulario, pantalla de interfaz de usuario puede cambiar su estructura para presentar nuevos elementos y funcionalidades del sistema, sin la necesidad de cambiar de una página Web a otra.

Los patrones descritos en la tabla anterior son utilizados en los modelos de UWE para especificar qué elementos y procesos van a utilizar los patrones RIA, lo anterior con miras a la etapa de desarrollo. La forma de denotar el uso de los patrones se realiza mediante la agregación de etiquetas en los elementos del modelo.

3.3.1.4. *Modelo conceptual*

El modelo conceptual se basa en el análisis de requisitos. El objetivo del modelo conceptual es proporcionar una especificación visual del dominio de la información relevante para el sistema Web, definiendo el diseño conceptual del sistema [Koch et al., 2008]. El diseño conceptual se basa en el modelo de clases de UML, ignorando las rutas de navegación, capas de presentación y aspectos de la interacción entre el usuario y el sistema. Dichos aspectos se abordan en los modelos de navegación y presentación respectivamente [Koch y Kraus, 2002].

La forma de representar el modelo conceptual de UWE es con un diagrama de clases UML, proporcionando una visión estática que demuestre la colección estática del dominio. La creación del modelo conceptual puede ser guiado en base a las directrices fundamentales del modelado orientado a objetos para construir un diagrama de clases, como:

1. Búsqueda de clases del sistema.
2. Especificación de los atributos más importantes.
3. Determinación de las asociaciones entre clases.
4. Identificar agregación y composición de clases.
5. Definir herencia entre clases.
6. Definir condiciones.

La Figura 3.10 presenta el modelo conceptual del sistema Kaizen. El diagrama muestra los elementos principales que componen la estructura conceptual del sistema, un total de veinte elementos forman parte del modelo. Debido a que el sistema Kaizen tiene como principal objetivo el brindar un marco de trabajo para la implantación de iniciativas de mejora basadas en el modelo MoProSoft, el principal elemento conceptual del sistema es la iniciativa o proyecto de mejora que en particular una empresa quiera llevar a cabo mediante la guía del sistema.

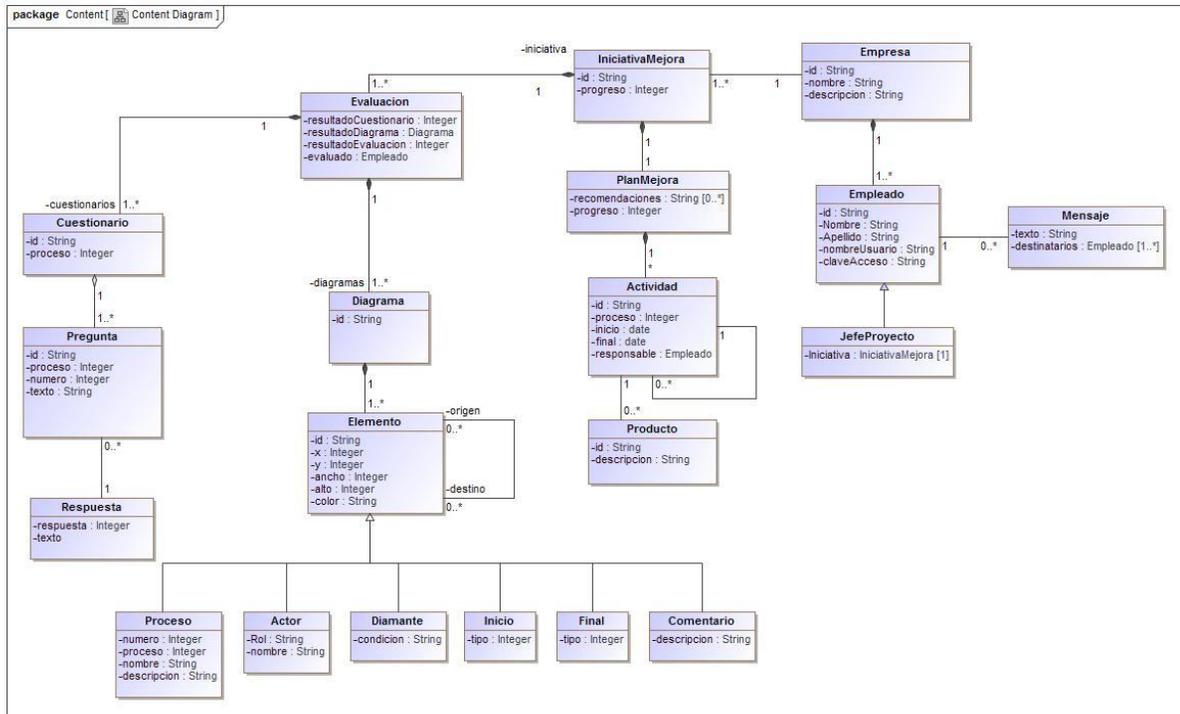


Figura 3.10. Modelo conceptual de Kaizen.

En primera instancia la iniciativa de mejora está compuesta por una o varias evaluaciones, dependiendo del número de empleados o miembros que se agreguen al proyecto. Como se menciona en la parte de funcionalidades, la evaluación se compone de un cuestionario y un modelado del proceso actual de desarrollo software de la empresa. Uno o más cuestionarios componen la evaluación en base a los procesos elegidos por el responsable de la iniciativa de mejora, éstos a su vez contienen una serie de preguntas ligadas a las recomendaciones del modelo MoProSoft y que están vinculadas a las respuestas de los empleados que hayan sido sometidos a la primera parte de la evaluación. Los diagramas actuales de procesos de la empresa son la siguiente parte de la evaluación, como se ha mencionado. Estos diagramas tienen como objetivo adquirir de manera gráfica el estado actual de los procesos de la empresa, en ellos los miembros de la iniciativa plasman mediante diferentes elementos el proceso que se lleva a cabo actualmente en la empresa para el desarrollo de sus productos de software. Los elementos que componen el modelo conceptual de la evaluación se muestra en la parte izquierda de la Figura 3.10.

En la parte central de la figura se muestran los elementos relacionados con la implantación de la iniciativa de mejora, el componente principal es el plan de mejora el cual contiene un compendio de actividades a realizar por parte de la empresa, en base a las recomendaciones y resultados obtenidos en la etapa de evaluación. Las actividades que componen al plan de mejora pueden o no estar relacionadas con los productos, es decir, las actividades requieren de un cierto producto obtenido en el desarrollo o tienen como resultado un producto que es ocupado por otra actividad del plan. Las fechas de inicio y final estimadas, forman parte de la estructura de la actividad, debido a que el plan generado mostrará esta información en forma gráfica a los miembros involucrados en el proyecto de mejora.

El modelo conceptual de los elementos encargados de la iniciativa de mejora (parte derecha de la Figura 3.10) giran en torno a la empresa, debido a que es la entidad a la cual está relacionada la iniciativa de mejora. La empresa está compuesta por varios empleados, que serán los encargados de

implantar la iniciativa de mejora en la estructura de la empresa, en particular uno de ellos será el responsable de dicha iniciativa, el llamado jefe del proyecto de mejora.

3.3.1.5. Modelo de navegación

En base al análisis de requisitos y el modelo de contenido, el modelo de navegación se construye para especificar la estructura del hipertexto del sistema, compuesta por nodos y enlaces [Preciado et al., 2008]. El modelo de navegación es parte fundamental en el proceso UWE, debido a que es el antecedente principal del modelo de presentación, y ambos son esenciales en el modelado de una aplicación Web. Por una parte el modelo de navegación comprende la especificación de qué objetos pueden ser visitados mediante la navegación a través de la aplicación Web y las asociaciones entre ellos, mientras que el modelo de presentación proporciona una idea de cómo van a presentarse esos objetos al usuario. En el marco de UWE el modelo de presentación se destaca como el más importante, pues con él se pueden representar elementos estáticos, a la vez que se pueden incorporar lineamientos semánticos de referencia para las funcionalidades dinámicas de una aplicación Web [Ocaña y Rossainz, 2004].

UWE proporciona una serie de estereotipos de clase para crear el modelo de navegación (véase Tabla 18) para mostrar los objetos que pueden ser visitados y la asociación que existe con otros objetos de la aplicación Web mediante la navegación [Kroiß y Koch, 2008].

Tabla 18. Descripciones de estereotipos de navegación en UWE.

Nombre	Símbolo	Descripción
Navegación (<i>navigationClass</i>)		Una clase de navegación representa un nodo navegable en la estructura del hipertexto, además de establecer la conexión entre los modelos de navegación y contenido. Una clase de navegación que está asociada con una clase de contenido se utiliza para representar una instancia de esa clase.
Proceso (<i>processClass</i>)		Las clases de procesos son utilizadas para integrar los procesos de negocio del sistema dentro del modelo de navegación y para definir los datos que son intercambiados con el usuario durante el proceso.
Nodo (<i>node</i>)		Puede representar cualquier nodo en un grafo de navegación. Esto significa que cuando el nodo es alcanzado en el marco de navegación, al usuario se le proporciona una determinada información y opcionalmente se ofrece la posibilidad de llevar a cabo una o más acciones. Un nodo no representa necesariamente una página de la aplicación Web, aunque puede serlo.
Menú (<i>menu</i>)		Un menú se utiliza para controlar las rutas alternativas de navegación.
Índice (<i>index</i>)		Un índice permite seleccionar una instancia de un conjunto de instancias que han sido obtenidas durante la navegación anterior. Esto significa que las instancias que puede seleccionar el usuario se toman del predecesor en el recorrido de navegación.
Consulta (<i>query</i>)		Una consulta se utiliza para recuperar contenido de un determinado origen de datos. A diferencia del índice, no obtiene el contenido o instancias a partir de la página precursora de la ruta de navegación, sino a partir de una base de datos o de cualquier otra fuente de datos que admite búsquedas.
Visita guiada (<i>guidedTour</i>)		Una visita guiada proporciona un proceso secuencial para el marco de navegación. El usuario puede navegar hacia atrás y adelante a través de una colección de enlaces.

El modelo de navegación para Kaizen (véase Figura 3.11) es una representación del marco de navegación de la herramienta que automatiza las fases necesarias para establecer iniciativas de mejora. El modelo facilita el entendimiento de los flujos de ejecución, y el cómo debe comportarse Kaizen a determinados eventos realizados por el usuario. En base a los estereotipos presentados en la Tabla 18, se plasma el marco de navegación en la herramienta, con el fin de proporcionar a la fase

de desarrollo un esquema fácil y claro que permita comprender la implementación de los diferentes módulos que forman parte de la herramienta.

El marco de navegación de Kaizen indica que éste comienza en el momento en que el usuario inicia sesión para poder ingresar a las funcionalidades de la herramienta (**Inicio**). Después de ingresar los datos de usuario, se realiza una consulta a la base de datos, obteniendo respuesta a la solicitud de ingreso (parte superior izquierda de la Figura 3.11). Ya dentro de la aplicación se presenta el menú principal de la herramienta, el menú desplegado dependerá del tipo de usuario, el cual despliega las distintas opciones para realizar dentro de la herramienta (**MenuPrincipal**). Desde este punto, los responsables de la administración del sistema pueden dar de alta nuevas empresas, nuevos responsables de iniciativas de mejora de procesos, consultar las iniciativas y empresas (**RegistroEmpresa, RegistroUsuario, InformacionEmpresa**). En el caso de los responsables de las iniciativas, en este punto de la navegación pueden acceder a las funcionalidades para dar de alta un nuevo miembro del equipo de trabajo y registrar nuevas iniciativas dentro de la empresa (**RegistroIniciativa, RegistroMiembro**). Al momento de registrar una nueva iniciativa, los miembros del proyecto de mejora pueden acceder al menú de la nueva iniciativa (**MenuIniciativa**); en primera instancia las funciones activadas están relacionadas con la fase de evaluación para los responsables de proceso (**Cuestionario, ModelarDiagrama**). Como se observa en la Figura 3.11 (parte inferior izquierda), se realizan consultas y procesos para realizar cada una de las funcionalidades. Cada vez que un responsable de proceso culmina la fase de evaluación, para el encargado de iniciativa se activa la opción de resultados (**ConsultarResultados**), en donde el encargado puede obtener y consultar los resultados de los miembros de la iniciativa (**ResultadosMiembro**), una vez que todos los responsables de proceso culminen su evaluación es posible consultar los resultados de cada proceso evaluado (**ResultadosProcesos**). De igual manera, una vez que todos los responsables de procesos han sido evaluados, se puede dar comienzo a la implantación de la mejora. En primera instancia se muestran los resultados de la evaluación para después configurar la implantación con la elección de las fases y actividades a trabajar en el ciclo de mejora (**ConfiguracionImplantacion**). Una vez configurada la iniciativa en su fase de implantación, se genera el plan de actividades a seguir (**GenerarPlanActiviades**) para posteriormente dar inicio a la fase de implantación (**ComenzarImplantacion**).

Al dar comienzo con la implantación, los miembros del equipo de trabajo pueden acceder a un menú relacionado con el plan de actividades generado por Kaizen (**MenuPlanActividades**) donde se puede acceder a funcionalidades del plan de trabajo para visualizar la planeación, visualizar el estado de la implantación, generar anuncios para los miembros del equipo de trabajo, reportar actividades y finalizar una actividad, entre otras (**PlanActividades, GenerarReporteActividades, GenerarAnuncio, ProgresoActividad, ReportarActividad, FinalizarActividad**). Esta parte de la herramienta es la encargada de controlar la implantación de la mejora dentro de la organización por lo que muchos de los estados de navegación para esta fase son generados automáticamente por Kaizen. Como función adicional Kaizen ofrece en esta fase la opción de envío de mensajes entre el equipo de trabajo (**GenerarMensaje**) para mantener una comunicación especial a lo largo el proyecto de mejora. Para mantener informada a la alta dirección acerca de los avances del proyecto de mejora, la herramienta genera reportes sobre el estado de la iniciativa (**InformacionIniciativa, GenerarReporteIniciativa**).

Así, el modelo de navegación de Kaizen sirve como guía en la fase de construcción de la herramienta, ya que en base a los elementos presentados en el modelo se generan los marcos y flujos de navegación necesarios para acceder a cada una de las funcionalidades especificadas en los requerimientos de Kaizen. De igual forma, ayudan a diseñar e implementar las consultas a la base de datos y los *scripts* para llevar a cabo los procesos y consultas necesarias para mostrar, almacenar y modificar información durante la ejecución de la herramienta.

3.3.1.6. *Modelo presentación*

El modelo de presentación de UWE pretende proporcionar una representación abstracta de la interfaz de usuario final y definir la interacción de las clases navegables con el usuario. Este modelo se encuentra basado en el modelo de navegación. El objetivo principal de esta capa es conseguir un modelo abstracto de la interfaz de usuario, que permita mostrar la forma en que la estructura de navegación es presentado al usuario [Preciado et al., 2008]. El diseño del modelo de presentación brinda un panorama acerca del aspecto final de la interfaz de usuario. Las decisiones sobre los colores, tamaños, tipo de fuente u otros elementos específicos en la elaboración de interfaces no se consideran por ser parte de la implementación. El objetivo principal del modelo según [Koch y Kraus, 2002] es visualizar la organización de la estructura de la aplicación Web de una manera más intuitiva que como lo hace el modelo de navegación.

En primera instancia el modelo de presentación representa elementos básicos para la esquematización de la interfaz de usuario, como lo son texto, imágenes, formas, audio y video. Los cuales son presentados al usuario en una vista de interfaz usuario (UI View) o en una página de presentación (*presentationPage*). Sin embargo, estos elementos son insuficientes para presentar algunas de las características necesarias para implementar muchas de las funcionalidades de Kaizen. Debido al alto índice de interacción que presentan estas tareas, es por eso que además de los elementos tradicionales para crear una interfaz de usuario Web 1.0, se hace uso de los patrones RIA (presentados en la sección 3.3.1.1) para dotar al modelo con características propias de este tipo de aplicaciones, y de esta forma brindar al modelo de navegación un mejor acercamiento de las funciones que debe de cumplir la interfaz gráfica de Kaizen. A cada elemento de la interfaz se agrega una etiqueta, la cual denota la característica RIA con la que cuenta el componente, de esta forma se busca que sea tomada en cuenta dicha característica en el proceso de construcción. El modelo de presentación toma como referencia al modelo de navegación, por lo cual debe de cumplir con el marco de navegación que presenta dicho modelo.

La Figura 3.12 muestra el modelo de presentación de la herramienta Kaizen. Como se ha mencionado anteriormente, el modelo de presentación está basado en el modelo de navegación, por lo que el diagrama mostrado se desarrolla en base al modelo de la Figura 3.11. El modelo enriquecido con los patrones RIA de UWE, brinda un marco de referencia y una guía idónea para el comienzo del desarrollo de la herramienta, debido a que además de presentar el marco de navegación que se debe de seguir durante la ejecución de la aplicación, proporciona una descripción de los elementos y el comportamiento que éstos deben de tener durante la interacción con el usuario. Los elementos como menús, hipervínculos, textos, formularios, imágenes y otros, son presentados en cada una de las páginas de presentación que compondrán la interfaz del sistema. El modelo proporciona un amplia idea de cómo debe de estructurarse la interfaz de usuario final de la herramienta, tomando en cuenta que durante la implementación pueden presentarse cambios debido a las características de la tecnología utilizada para su desarrollo. Las características RIA que presentan algunos de los elementos son vitales para la consecución de los objetivos de Kaizen. Por ejemplo, en el caso de la edición del diagrama de procesos, la característica de “arrastrar y soltar” (*drag & drop*) es imprescindible para la realización de dicha actividad. Otras características como vistas dinámicas, son necesarias para mostrar alertas o contenido en base a operaciones en tiempo real de la aplicación, un refrescado periódico es también necesario debido a que el estado de la iniciativa de mejora puede cambiar debido a las alertas generadas por el sistema. Es por eso que la descripción que los patrones RIA UWE proporcionan al modelo de presentación de Kaizen, juegan un papel muy importante en el momento de la construcción de la herramienta.

3.4. Construcción del sistema

La fase de implementación es el compendio de actividades donde se convierte el resultado del diseño en componentes, código fuente y en una aplicación funcional. Una vez obtenido el desarrollo. El primer paso en la fase de construcción es la definición de una tecnología de desarrollo que se adapte y sea capaz de dar solución a los requerimientos que la herramienta Kaizen establece. Tomando en cuenta que una aplicación de tipo RIA es el enfoque idóneo para solventar las características de alta interacción que la herramienta solicita, ésta se toma como punto de partida para el análisis de la tecnología de desarrollo. La propuesta de una arquitectura base para el proceso de desarrollo es un punto importante, ya que en base a ella, se puede tomar en un cuenta el patrón de implementación a seguir para la elaboración de los componentes del sistema. Tomando en cuenta la arquitectura propuesta, se presentan las fases de construcción del sistema Kaizen dividiendo su construcción en incrementos basados en las fases que debe de cubrir una herramienta para la implantación de iniciativas de mejora: Compromiso, Evaluación, Generación de planes de mejora e Implantación de los planes.

En las siguientes secciones se presentan cada uno de los puntos anteriores, detallando puntos y actividades importantes para el desarrollo y consecución de cada uno de los objetivos.

3.4.1. Tecnología de desarrollo

Tomando en cuenta la característica primordial de la herramienta, la cual establece el entorno Web como plataforma de despliegue de la herramienta, se descartan las tecnologías que son utilizadas para la generación de aplicaciones de escritorio o ajenas a ejecutarse sobre el ambiente del ciberespacio. El segundo punto a considerar es la independencia de la plataforma; en general, las tecnologías Web son independientes de la plataforma que el usuario final utilice, debido a que los servidores Web son los encargados de interpretar los códigos, y retornarlos en formato de Lenguaje de marcado de Hipertexto (HTML), el cual es descifrado por los navegadores Web. En algunos casos es necesario el uso de algún tipo de complemento adicional para visualizar la aplicación, pero al tratarse de una pequeña aplicación para el navegador, no afecta de ninguna manera el funcionamiento del SO que el usuario maneje.

Debido a las características relacionadas con el desarrollo de aplicaciones robustas y altamente interactivas, es necesario abordar un nuevo tipo de aplicaciones Web, llamadas Aplicaciones Enriquecidas de Internet (RIA, acrónimo en inglés para *Rich Internet Applications*). La aparición de estas aplicaciones se debe principalmente a que en los últimos años, la demanda de aplicaciones basadas en Internet continúa creciendo y, en general es bastante diferente a la demanda que se presentaba en la década de los noventa. Los usuarios finales ahora exigen más de sus inversiones en tecnología Internet [Tapper et al., 2009].

La capacidad para distribuir valores reales a los usuarios ha obligado a muchas empresas a buscar modelos de aplicaciones de Internet más ricas, modelos que combinen la potencia de las aplicaciones de escritorio, ricas en dispositivos, con la utilización y naturaleza rica en contenido de las aplicaciones Web [Davis y Phillips, 2008].

Debido a su nuevo enfoque las RIA ofrecen grandes ventajas respecto a las aplicaciones Web tradicionales. Una de las principales ideas de este tipo de aplicaciones es el mejorar el método en que se envía y recibe el flujo de información entre el cliente y servidor, las aplicaciones cargan todo el procesamiento de datos en el servidor, recargando toda la página con cualquier acción que realice el usuario, aún si el cambio realizado es mínimo; mientras que bajo un enfoque RIA, solamente es necesario cargar la interfaz una sola vez, y para cada acción del usuario se realiza una petición específica de la información solicitada al servidor sin la necesidad de recargar toda la página. [Pérez

y Messeguer, 2008; Tapper et al., 2009]. De esta manera, se consigue reducir el tiempo de espera del usuario, el ancho de banda consumido y la carga de trabajo para el servidor [Davis y Phillips, 2008].

Las RIA ofrecen a los desarrolladores un modo económico y de probada eficiencia para proporcionar modernas aplicaciones con las ventajas de los negocios reales [Tapper et al., 2009]. Algunas de las ventajas que los autores presentan sobre éstas, son las siguientes:

- Se alejan de la arquitectura basada en la página, lo que reduce la carga de los servidores Web y reduce el tráfico total de la red, lo que deriva en una navegación Web mucho más ágil [Pérez & Messeguer, 2008; Tapper et al., 2009].
- Se pueden desarrollar aplicaciones más potentes. Las tecnologías que implementan el concepto RIA suelen facilitar la generación de módulos y componentes complejos, además de la generación de efectos que hacen la navegación mucho más cómoda [Farré y Messeguer, 2005].
- Permiten desarrollar aplicaciones más interactivas que no se pueden obtener utilizando solo HTML [Goralski y Leon, 2008].
- Permiten crear aplicaciones más atractivas mediante el uso de audio, video y gráficos avanzados [Brown, 2008].
- Una RIA bien diseñada reduce el nivel de frustración del usuario, porque ya no es necesario navegar por varias páginas para encontrar lo que necesita o esperar a que una página nueva se cargue antes de continuar con la actividad [Tapper et al., 2009].
- El tiempo que el usuario necesita para aprender cómo utilizar la aplicación puede reducirse mucho, por lo cual el usuario tiende a estar más capacitado en el uso de la aplicación [Tapper et al., 2009].

En base a las ventajas que presentan las RIA, y dado que se acoplan a las características que deben tener las tecnologías de desarrollo para la aplicación Kaizen, se adopta un enfoque RIA para el desarrollo. En este sentido, existen diversas tecnologías para construir aplicaciones de este tipo. Las opciones más populares son las iniciativas basadas en HTML como Ext JS [URL-7], y las opciones basadas en complementos o *plug-in* como Adobe Flex [URL-8]. También existe la tecnología de Microsoft, Silverlight [URL-9]. Estas tecnologías convergen en muchas características, pero existen pequeñas diferencias que deben de ser tomadas en cuenta, ya que algunas de ellas marcan la pauta para la elección de una u otra. La Tabla 19 muestra una comparación de las tecnologías mencionadas, ésta se basa en las características C6, C7 y C8 de la Tabla 12. Cabe mencionar que en el estudio se dejaron fuera tecnologías como Adobe Flash o similares, ya que tienden a enfocarse a aplicaciones de tipo animación en lugar de aplicaciones complejas.

Tabla 19. Comparativa sobre las tecnologías para desarrollo de aplicaciones RIA.

Criterios	Tecnologías		
	Ext JS	Adobe Flex	Silverlight
Distribución de las herramientas de desarrollo	Comercial y Libre	Comercial y Libre	Comercial y Libre
Distribución de aplicaciones desarrolladas	Libre	Libre	Libre

Criterios	Tecnologías		
	Ext JS	Adobe Flex	Silverlight
Uso de Base de Datos (BD)	Conexión mediante un script interpretado del lado del servidor	Conexión mediante un script interpretado del lado del servidor	Conexión mediante un script interpretado del lado del servidor
Lenguaje de Desarrollo	JavaScript, DHTML y DOM	MXML y ActionScript	XAML y C#
Bibliotecas para el desarrollo de aplicaciones colaborativas	-	LCCS (LiveCycleCollaborationService.)	-

Como puede observarse en la Tabla 19, las características de las tecnologías de desarrollo son muy parecidas, debido a que al encontrarse bajo un enfoque RIA comparten muchas de ellas. Una gran parte de este tipo de tecnologías hace uso de scripts en el servidor para conectarse a un manejador de BD, como por ejemplo un script en PHP (*PHP Hypertext Pre-processor*), para realizar la conexión y posterior consulta a una BD construida en MySQL. Una diferencia marcada se encuentra en los lenguajes de desarrollo, en donde cada tecnología hace uso de lenguajes de programación y etiquetado. La diferencia radica en que los lenguajes de Ext JS son interpretados dentro de un navegador Web, los que componen a Flex y Silverlight necesitan de un complemento adicional para poder ser interpretados. Sin embargo, dichos complementos no tienen ningún costo y son soportados por la mayoría de los navegadores Web, el más claro ejemplo es Flash Player, el cual está disponible para los navegadores más populares y es usado habitualmente por la gran mayoría de usuarios de Internet.

El factor que marca la tendencia de la comparación, es la característica para desarrollar aplicaciones colaborativas. Si bien es cierto que en las tres plataformas se pueden desarrollar aplicaciones colaborativas, solamente Adobe Flex brinda una serie de bibliotecas mejor conocidas como Interfaz de Programación de Aplicaciones (API, *Application Programming Interface*) que brinda un conjunto de funciones, procedimientos y métodos para el desarrollo de aplicaciones colaborativas, además de una serie de servicios y herramientas para cumplir dichos objetivos. Servicio de colaboración LiveCycle (LCCS, *LiveCycleCollaborationService*) [URL-10] es el nombre que maneja Adobe para el API, este servicio permite a los desarrolladores agregar fácilmente capacidades de colaboración de tiempo real y social en sus aplicaciones. Con el uso de LCCS se puede añadir funciones de colaboración a aplicaciones nuevas o ya existentes. LCCS ofrece las siguientes características:

- Mensajería en tiempo real: Permite crear un flujo de comunicación entre componentes y aplicaciones.
- Componentes pre-construidos: Permiten acelerar el desarrollo mediante el uso de pizarras, ventanas de chat, webcam y otros componentes pre-diseñados.
- Transmisión multimedia: Permite el uso de video y audio compartido con el apoyo de la transmisión P2P.
- Gestión de usuarios: Permite detectar, modificar, estado y funciones de los usuarios.

Dadas las ventajas que representan estos servicios, la tecnología elegida para el desarrollo de la herramienta Kaizen es Adobe Flex en su versión 4.0, la más reciente hasta el momento del

desarrollo de este proyecto. Adicionalmente, se usa LCCS para el desarrollo de los módulos colaborativos, y PHP y MySQL para la gestión de la BD.

3.4.2. Arquitectura del sistema

La arquitectura es la forma en que se organizan los componentes de un sistema, cómo interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad. En base a las características del sistema y la tecnología seleccionada para su desarrollo, y tomando en cuenta en primera instancia un esquema Cliente-Servidor debido a la naturaleza Web de la aplicación, el patrón de arquitectura de software elegido para desarrollar Kaizen es el Modelo-Vista-Controlador (MVC), el cual tiene como principal característica separar los datos y la lógica de negocios de una aplicación de la interfaz de usuario. Un punto importante para elegir MVC como patrón de arquitectura, es que la metodología de diseño UWE, se adapta de forma natural a este tipo de arquitecturas, siendo el modelo conceptual UWE el punto de partida para el desarrollo del componente Modelo, el modelo de navegación UWE para la construcción del controlador y el modelo de presentación para el desarrollo de la vista.

MVC es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz de usuario y es el más utilizado en aplicaciones Web, ya que, facilita la funcionalidad, escalabilidad y mantenimiento de un sistema, de forma simple y sencilla. MVC divide las aplicaciones en tres niveles de abstracción:

- **Modelo:** Representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como intermediario con la base de datos. Lo que en un ejemplo de programación orientada a objetos serían las clases abstractas.
- **Vista:** Es la encargada de mostrar la información al usuario de forma gráfica y humanamente legible.
- **Controlador:** Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que se presenten al usuario.

Para el buen funcionamiento del patrón, se deben tener en cuenta tres premisas: sencillez y simplicidad, facilidad de mantenimiento, y practicidad para evolucionar.

La Figura 3.13 muestra la arquitectura para el desarrollo de Kaizen. La arquitectura se basa en un patrón de diseño MVC, a través del cual el usuario accede a la herramienta por medio de un navegador Web, el cual cuenta con el complemento Flash Player (necesario para ejecutar aplicaciones Flex). Como característica propia de una aplicación RIA, solamente se realiza una sola petición al servidor para el despliegue de la interfaz y funcionalidades que no necesiten ser albergadas en él. De esta forma se pone de manifiesto una de las principales ventajas de las RIA: desarrollar gran parte de la lógica de negocios del lado del cliente en lugar de realizar todas estas funciones en el servidor.

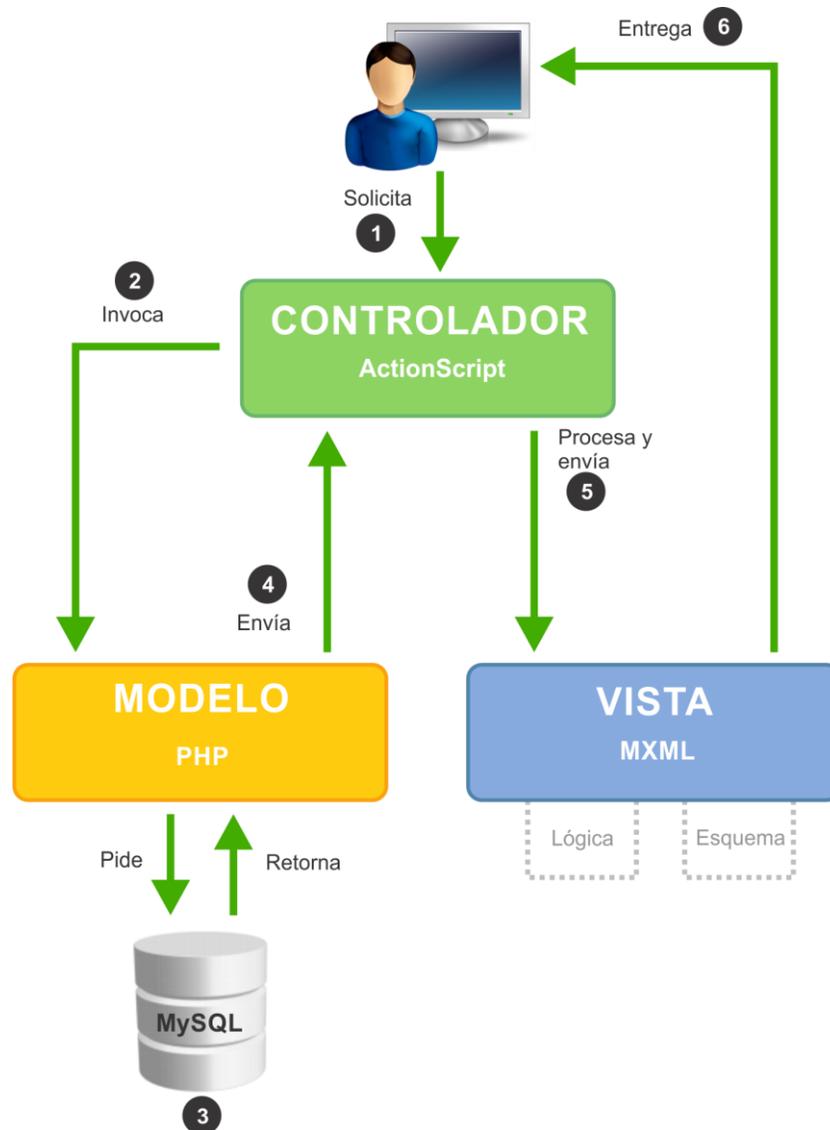


Figura 3.13. Arquitectura de Kaizen.

El funcionamiento de la arquitectura de Kaizen, puede explicarse de la siguiente forma:

1. El usuario realiza una petición.
2. El controlador construido en base a ActionScript, captura el evento. El controlador hace la llamada al modelo/modelos correspondientes, efectuando las modificaciones pertinentes sobre el modelo.
3. El modelo, implementado mediante códigos PHP, es el encargado de interactuar con la base de datos implementada bajo el manejador MySQL, para solicitar y obtener información de ella.
4. Una vez obtenida la información de la base de datos, el modelo regresa la información al controlador mediante elementos XML (acrónimo en inglés para *eXtensible Markup Language*).
5. El controlador recibe la información y la envía a la vista.

6. La vista creada en MXML, procesa la información. La lógica de la vista, una vez procesados los datos, los acomodará en base al diseño del esquema de la interfaz de usuario y los mostrará al usuario de forma legible.

Existen elementos albergados del lado del servidor como el servicio LCSS, dado que desde su posición en el servidor realiza las conexiones entre aplicaciones cliente para llevar a cabo el paso de mensajes, y de esta forma dar pie al trabajo colaborativo entre las aplicaciones clientes que se encuentren conectadas. Los mensajes son recibidos por cada una de las aplicaciones clientes, las cuales hacen uso de la información para realizar distintas tareas. Otros elementos ubicados en el servidor, son aquellas funcionalidades Web que permiten la construcción de módulos relacionados con el almacenamiento de archivos desde la aplicación cliente al servidor, emitir alertas del sistema mediante correos electrónicos, acceso a archivos, además de otras funcionalidades menores, pero necesarias para el buen funcionamiento del sistema.

3.4.3. Método de desarrollo

Para la construcción del sistema Kaizen, se implementa un método ágil basado en incrementos. La definición de incrementos se encuentra fundada en las fases que un proyecto de mejora de procesos debe de cumplir, para conseguir la implantación de una mejora dentro de cierta organización: Compromiso, Evaluación, Generación de Planes de Mejora, e Implantación de los Planes de Mejora. La metodología de construcción (véase Figura 3.2) propone un enfoque en componentes para generar un incremento, que posteriormente se integra al software total. La Tabla 20 muestra la descripción de los incrementos, en ella se muestran las características y partes del diseño más importantes que serán implementadas en dicho avance, igualmente se presenta la prioridad que se ha establecido para cada incremento. La prioridad ha quedado establecida en base al orden que se debe de llevar a cabo en un proyecto de mejora, ya que cada fase previa es altamente necesaria para llevar a cabo una fase actual.

Tabla 20. Descripción de incrementos.

Nombre	Orden	Descripción	Componentes Relacionados
Compromiso	1	En éste incremento se construyen e integran todos los componentes relacionados con la configuración inicial de un proyecto de mejora, para de esta manera dejar de manifiesto el compromiso de mejora, y poder llevar a cabo la fase de evaluación.	IniciarSesion RegistroIniciativa RegistroMiembro EstablecerCompromiso ConfiguracionIniciativa
Evaluación	2	El avance comprende el desarrollo de los métodos de evaluación de procesos propuestos para la herramienta: Cuestionario y Modelado de Procesos.	Cuestionario ConsultarPreguntas GuardarRespuesta ModelarDiagrama AgregarElemento CrearConexion BorrarElemento EliminarConexion
Generación Planes	3	En base a las evaluaciones relacionadas, éste incremento genera los resultados por miembro evaluado y por proceso evaluado. En base a los resultados obtenidos se hace la propuesta de los actividades a mejorar dentro de un proceso, donde el encargado de la iniciativa elige que actividades tratar, y en base a ellas se genera un plan de mejora.	ConsultaResultados ResultadosEvaluacion ResultadosMiembro ResultadosProceso ObtenerResultados ConfiguracionImplantacion GenerarPlanActividades

Nombre	Orden	Descripción	Componentes Relacionados
Implantación Planes	4	En el avance se desarrollan los componentes relacionados con el control y monitoreo de la realización de actividades que forman parte del plan de mejora.	ComenzarImplantacion PlanActividades AsignarActividad GenerarAnuncio ProgresoActividad ReportarActividad FinalizarActividad AgregarProducto

Cabe resaltar que en el apartado de Componentes Relacionados solamente se presentan los componentes más importantes del incremento, otros componentes son complementarios y auxiliares para la realización de los incrementos dentro de las funcionalidades del sistema.

En las secciones siguientes se presentan los mecanismos y detalles que toman en cuenta para la construcción de cada incremento.

3.4.4. Incremento compromiso

El objetivo de una etapa de compromiso en un proyecto de mejora es conseguir el compromiso de la alta dirección para que la empresa se involucre en el proyecto. El equipo directivo debe de entender y comprender las implicaciones de un programa de mejora, y comprometerse a aportar tantas personas, tiempo y otros recursos como sean necesarios para tener éxito en la ejecución del proyecto mejora. También se seleccionará al equipo evaluador, los participantes en la evaluación, la logística necesaria e, incluso, los procesos a evaluar en función, normalmente, de los objetivos de negocio de la empresa [Cuevas-Agustín et al., 2002].

En base a lo anterior, se plantean las siguientes funcionalidades para cumplir con los objetivos del incremento:

1. Establecer las bases de la mejora. La funcionalidad busca mostrar a la alta dirección y encargado de la mejora en la empresa aspectos relativos a qué es la mejora de procesos y sus beneficios. En base a expertos de la materia, se elaboran textos introductorios, para inmiscuir al personal que estará involucrado en el proyecto de mejora. Kaizen muestra de forma obligatoria los textos a los participantes de la mejora, con el fin de que cada miembro del proyecto tenga conocimiento sobre el objetivo de un proyecto de este tipo, qué beneficios se obtienen con su ejecución, y a qué se comprometen durante la ejecución del mismo. De igual forma la organización establece el origen y objetivos de la iniciativa. La funcionalidad tiene como objetivo brindar un marco introductorio antes de pasar a las siguientes etapas del proyecto.
2. Determinar el alcance de la evaluación. Se decide el alcance de la evaluación y en qué forma se van a utilizar los resultados, se escoge si se trata de una evaluación por madurez o bien por capacidad del proceso. La organización elige los procesos a evaluar, así como el nivel en qué se evaluarán.
3. Seleccionar a los participantes. Una vez que se configuró el programa de mejora, se selecciona quién formará parte del equipo. En primera instancia se registran las personas que formarán parte del proyecto de mejora, para después ser asignados a un rol dentro del proceso(s) elegido(s). En base a la asignación, el miembro será evaluado y tomado en cuenta para la asignación de actividades en la fase de implantación.

El desarrollo del incremento presenta una complejidad promedio, ya que, la mayoría de las operaciones se encuentran relacionadas con la inserción, actualización y eliminación de la

información en la base de datos. La información generada en la base de datos por medio de las funcionalidades de este incremento es indispensable para el inicio de la mejora, debido a que fases posteriores como la generación de las evaluaciones se realizan en base la información obtenida con estas funcionalidades.

3.4.5. Incremento evaluación

A través de la etapa de evaluación del proceso se obtiene el nivel de madurez de la empresa o de capacidad de los procesos que se van a evaluar, según sea el caso, además de una buena referencia a los puntos fuertes y débiles de la organización en los procesos evaluados. Asimismo, también se obtienen y eligen los procesos y actividades que van a mejorarse en las siguientes etapas del ciclo de mejora [Calvo-Agustín et al., 2002].

La propuesta de Kaizen, es realizar la evaluación del proceso en base a la combinación de los resultados de dos técnicas, la primera en forma de cuestionario generado en base a la información obtenida a través de la configuración del alcance del proyecto y actividades propuestas por MoProSoft, y la segunda obteniendo la descripción del proceso actual por medio de la edición de un diagrama, buscando así obtener una representación fidedigna y muy cercana al proceso que se lleva actualmente en la organización. La combinación de los métodos tiene como objetivo principal presentar hallazgos en el proceso, que no se obtendrían llevando a cabo por separado ambos métodos.

3.4.5.1. Evaluación por cuestionario

En base a la información obtenida en la configuración del alcance de la iniciativa se obtienen los cuestionarios para evaluar el proceso. Los cuestionarios se generan por proceso y nivel establecidos en el alcance de la mejora, además, se segmentan en base a las fases en las que se divide el proceso según MoProSoft. Las respuestas del cuestionario corresponden a la forma de trabajar de la empresa en lo general y corresponden con la forma de trabajo de la propia organización. En este sentido, las respuestas no corresponden a la forma de trabajar de un único jefe de proyecto evaluado, pues no se evalúa a ellos, sino a la empresa. Por lo tanto, el cuestionario es el mecanismo inicial para la obtención de datos sobre el proceso de desarrollo de software de la organización.

Para cada pregunta hay cinco posibles respuestas: *Siempre*, *Usualmente*, *Algunas veces*, *Rara vez*, y *Nunca*. Cada pregunta del cuestionario debe de ser respondida en base al conocimiento de los proyectos llevados a cabo por cada jefe de proyecto y de aquellos a los que se tiene referencia. La forma de seleccionar el tipo de respuesta se basa en dos variables:

- Documentación o no de la actividad.
- Porcentaje de veces que se realiza la actividad.

La elección del tipo de respuesta varía en función de las dos variables anteriores, como se muestra en la Tabla 21.

Tabla 21. Elección de respuestas para preguntas.

	% Ocurrencia			
	100-75	74-25	24-1	0
Documentada	Siempre	Usualmente	Rara Vez	Nunca
No Documentada	Algunas Veces	Algunas Veces	Rara Vez	Nunca

- Siempre (S), cuando la actividad está Documentada y se realice entre el 100% - 75% de las veces.
- Usualmente (U), cuando la actividad está Documentada y se aplica en la organización entre el 74% y 25% de las veces.
- Algunas Veces (A), cuando la actividad no está Documentada y se aplica entre el 100% y 25% de las veces.
- Rara Vez (R), cuando la actividad se aplica sólo en proyectos especiales, pero la actividad no es del todo común en la organización. Dicha práctica podrá estar Documentada o no, y se aplica en un rango menor al 25% de las veces pero mayor a 0.
- Nunca (N), cuando la práctica no se utiliza nunca en la organización, según el conocimiento del evaluado. Se utiliza también “Nunca” si no existe la actividad, rol o producto al que se refiere la pregunta.

La Tabla 22 muestra un ejemplo del cuestionario obtenido para la evaluación del proceso Desarrollo y Mantenimiento de Software en el Nivel 1, para consultar los cuestionarios de todos los procesos y diferentes niveles consúltense el Anexo B de esta tesis.

Tabla 22. Cuestionario de DMS para nivel 1.

N°	Fase	Pregunta
1	Inicio	¿Revisan con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un entendimiento común y obtener un compromiso con el proyecto?
2	Requerimientos	¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?
3	Requerimientos	¿Identifican y consultan fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos?
4	Requerimientos	¿Analizan los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto?
5	Requerimientos	¿Elaboran o modifican el prototipo de la interfaz con el usuario?
6	Requerimientos	¿Generan o actualizan la Especificación de Requerimientos?
7	Requerimientos	¿Documentan una versión preliminar del Manual de Usuario o modifican un manual existente?
8	Requerimientos	¿Incorporan la Especificación de Requerimientos y Manual de Usuario a la Configuración de Software?
9	Análisis y Diseño	¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?
10	Análisis y Diseño	¿Analizan la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos?
11	Análisis y Diseño	¿Describen a detalle la apariencia y el comportamiento de la interfaz con base en la Especificación de Requerimientos de forma que se puedan prever los recursos para su implementación?
12	Análisis y Diseño	¿Describen a detalle los componentes que permitan su construcción de manera evidente?

N°	Fase	Pregunta
13	Análisis y Diseño	¿Generan o actualizan el Análisis y Diseño?
14	Análisis y Diseño	¿Incorporan el Análisis y Diseño a la Configuración de Software?
15	Construcción	¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?
16	Construcción	¿Implementan o modifican Componente(s) de software con base a la parte detallada del Análisis y Diseño?
17	Construcción	¿Incorporan los Componentes a la Configuración de Software?
18	Integración y Pruebas	¿Distribuyen tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?
19	Integración y Pruebas	¿Integran los componentes en subsistemas o en el sistema del Software?
20	Integración y Pruebas	¿Documentan el Manual de Operación o modifican el manual existente?
21	Integración y Pruebas	¿Documentan el Manual de Usuario o modifican el existente?
22	Integración y Pruebas	¿Incorporan el Software, Manual de Operación y Manual de Usuario a la Configuración de Software?

Una vez que los encargados de los procesos y jefes de proyecto han contestado los cuestionarios, se determina el nivel de cumplimiento o cobertura de los diferentes procesos evaluados, así como se obtiene una visión global de los puntos fuertes y débiles de cada proceso.

Para ello, se considera el nivel de cobertura de cada pregunta del cuestionario. Este nivel de cobertura de cada pregunta se calcula a partir de la suma ponderada de los porcentajes de cobertura de cada uno de los tipos de respuesta. La forma de calcular la cobertura de una pregunta es la siguiente:

1. Se calcula el porcentaje de cobertura de cada tipo de respuesta, teniendo en cuenta las respuestas de todos los responsables de proceso, mediante la siguiente ecuación:

$$Cr_{ij} = \frac{R_{ij}}{P} \quad (3.1)$$

Siendo Cr_{ij} es el porcentaje de cobertura para la pregunta i con el tipo de respuesta j , R_{ij} es el número de respuestas j para la pregunta i , P es el número de responsables de procesos evaluados. Donde, i varía desde uno hasta el número de preguntas que generadas en base al área de proceso y nivel elegidos para la evaluación, y j varía desde siempre hasta nunca.

2. El porcentaje de cobertura de cada pregunta se obtiene sumando ponderadamente los porcentajes de cobertura de sus tipos de respuesta, en base a:

$$Cp_i = Cr_{iS} + 0.75Cr_{iU} + 0.5Cr_{iA} + 0.25Cr_{iR} \quad (3.2)$$

Siendo Cp_i , el porcentaje total de cobertura para la pregunta i . La razón de la ponderación es debido a que se considera que las aseveraciones más fuertes, deben de tener mayor importancia que aquellas que son débiles, dado, que por ejemplo, no puede influir lo mismo algo que se realiza siempre que algo que se hace rara vez. Si no fuese así, todas las preguntas tendrían siempre una cobertura de 100%, lo que no aclararía nada.

3. También por cada pregunta, se calcula su media:

$$\bar{X}_i = \frac{4R_{iS} + 3R_{iU} + 2R_{iA} + R_{iR}}{P} \quad (3.3)$$

Siendo \bar{X}_i la media para la pregunta i . Se establecen pesos de ponderación mayores (4, 3, 2, 1 y 0), como factor, de forma que las diferencias existentes sean más visibles. Aun así, existe una correspondencia biunívoca o el mismo significado entre el porcentaje de cobertura de la pregunta y su media (la única diferencia es que los factores de escala son diferentes), con lo que un porcentaje de cobertura elevado, también reflejará una media elevada y en la misma proporción que dicho porcentaje.

En base a la media y la cobertura de preguntas se calcula la desviación estándar:

$$\sigma_i = \sqrt{\frac{P(4^2R_{iS} + 3^2R_{iU} + 2^2R_{iA} + R_{iR}) - P^2\bar{X}_i^2}{P^2}} \quad (3.4)$$

Donde σ_i es la desviación estándar de la pregunta i .

Por último, se calcula la cobertura total del proceso como media de las diferentes coberturas de preguntas:

$$C(x) = \frac{\sum_{i=1}^n Cp_i}{n} \quad (3.5)$$

Donde $C(x)$ es la cobertura total del proceso, x es el proceso y n es el número de preguntas del cuestionario correspondientes a ese proceso.

Una vez realizados todos los cálculos para los procesos evaluados, se analizan sus puntos fuertes y débiles. La forma de analizarlos es a través de las respuestas dadas por los responsables de proceso a las prácticas del cuestionario. Para ello se realizan gráficas de barra para la cobertura por pregunta y desviación estándar.

Ahora bien, en aquellas preguntas con cobertura menor a 85%, la práctica correspondiente a esa pregunta no está suficientemente implantada en la organización y tendrá que considerarse un aspecto a mejorar (punto débil) del proceso en cuestión. En aquellas con preguntas con cobertura mayor o igual al 85% es necesario analizar el valor de su desviación estándar.

Si su desviación estándar es:

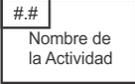
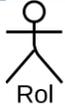
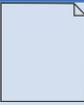
- Menor a 0.8. La práctica correspondiente a esa pregunta constituye un punto fuerte del proceso en cuestión.
- Mayor o igual a 0.8. La práctica correspondiente a esa pregunta debería de explorarse con mayor profundidad, por ejemplo a través de entrevistas posteriores o verificación de los productos generados por la actividad, para poder dilucidar si constituye un punto fuerte o un aspecto a mejorar. Significa que han existido fuertes discrepancias en las respuestas obtenidas de los responsables de proceso que han sido evaluados, es decir, los responsables de proceso responden de forma divergente a esa pregunta y, por tanto, el encargado de la mejora debería de explorar con mayor profundidad.

3.4.5.2. Evaluación por diagrama de procesos

Los diagramas de procesos son una técnica utilizada para representar de manera gráfica un determinado proceso dentro de una organización, debido a que facilitan la comprensión de un proceso gracias a una representación gráfica desarrollada con una serie de símbolos estandarizados. Al estar basado en procesos y actividades, el desarrollo de software se puede representar mediante este tipo de diagramas. Fuera de las recomendaciones de cualquier modelo de procesos, esta técnica puede obtener una representación veraz y fidedigna del proceso actual de desarrollo de una organización, ya que mediante una serie de símbolos permite que los líderes de proyecto puedan plasmar una representación real de los procesos que se llevan a cabo dentro de la organización.

En este sentido, la primera parte de la evaluación por diagrama de procesos consiste en el desarrollo de un editor de diagramas de procesos. Para desarrollar los diagramas de procesos se utiliza la notación gráfica propuesta por SEI y el SEPG (acrónimo en inglés para *OrgName's Software Process Engineering Groups*), la cual ayuda a construir diagramas de procesos mediante una serie de símbolos estandarizados (véase Tabla 23). Así, con un número limitado de símbolos es posible plasmar una representación completa de los procesos de una forma sencilla y rápida, debido a estas características esta metodología fue la elegida para la construcción del método de evaluación.

Tabla 23. Símbolos para la elaboración de diagramas de procesos [Fowler et al., 1999].

Nombre	Símbolo	Descripción
Flujo		Indican la secuencia de actividades. En otros casos representa la asignación de una actividad para un determinado miembro del equipo de trabajo.
Actividad		Se utiliza para representar una actividad dentro del ciclo de vida del proceso Software. Cada una contiene el nombre y número de cada actividad dentro del proceso.
Decisión		Representa una decisión binaria (Si/No).
Inicio		Representa el inicio del proceso de desarrollo.
Final		El símbolo final representa la culminación del proceso de desarrollo.
Persona		Este símbolo representa un determinado rol dentro de la empresa el cual es responsable de cierta actividad.
Comentario		Se utiliza para añadir algún tipo de nota o comentario para la mejor comprensión de los procesos.

En base a la notación anterior los líderes de proyecto plasman el proceso que se lleva actualmente dentro de la organización para desarrollar software. Esto con la finalidad de que a través

de cualquier indicación del equipo o sugerencia de un modelo de procesos se obtenga el proceso real de la empresa, con sus fortalezas y debilidades. Debido a la alta interacción que representa la edición de un diagrama por medio de una herramienta computarizada, el enfoque RIA adoptado para el desarrollo de Kaizen es de gran ayuda para crear este tipo de soluciones.

Las restricciones que se deben de tomar en cuenta para la edición de los diagramas de procesos y para el desarrollo del editor de diagramas son:

1. Todos los diagramas deben de contar con un principio y final.
2. Las actividades deben ser secuenciales, es decir, el número de actividad deben ser consecutivos.
3. No pueden existir elementos independientes.
4. Las actividades deben de estar relacionadas con una fase de desarrollo, de acuerdo al proceso al que se refiere.

Una vez obtenido el diagrama de procesos actual de la empresa, se procede a realizar el análisis para encontrar las diferencias y anomalías con respecto al diagrama de procesos sugerido por MoProSoft. El análisis está basado en la comparación de ambos diagramas y encontrar las diferencias importantes entre uno y otro.

El primer paso para llevar a cabo la comparación es crear un Grafo Dirigido (GD), ya que de esta manera es más sencillo llevar a cabo este tipo de comparación. De acuerdo a la definición, un grafo dirigido es un conjunto de vértices V y un conjunto de aristas A , tal que cada arista perteneciente al conjunto aristas A se asocia con dos vértices de forma ordenada (dirigida). La definición anterior hace factible el hecho de representar los diagramas obtenidos por medio del editor con grafos dirigidos teniendo en cuenta los siguientes puntos:

- El conjunto de vértices está compuesto por los símbolos del diagrama (inicio, final, actividad, decisión y persona), el símbolo de comentario es despreciado para crear el GD, ya que no es parte estructural del diagrama, sino más como una anotación.
- El conjunto de aristas se obtiene a partir de los flujos (flechas) que contiene el diagrama, estableciendo el vértice de origen y destino.
- A cada vértice se le proporciona información de acuerdo a la naturaleza del símbolo del diagrama que representa (nombre, número, fase relacionada y demás).

La representación del diagrama de procesos por medio de un GD (véase Figura 3.14) hace posible la ejecución del mecanismo para el mapeo de ambos diagramas para obtener coincidencias y diferencias, y tomar en cuenta los resultados para obtener los resultados finales.

Los pasos para obtener el GD que representa al diagrama de procesos es el siguiente:

1. Tomar cada elemento del diagrama de procesos y crear un vértice del GD, excepto los elementos persona y comentario, los primeros forman parte de la información del vértice y los segundos se desprecian debido a que son sólo de carácter informativo.
2. Se establece la información del vértice dependiendo del tipo de elemento: Nombre, Número, Descripción, Persona Responsable, Decisión, y demás información.
3. Para cada flujo del diagrama se debe obtener su origen y su destino para posteriormente crear una arista del GD, el peso de cada arista es nulo, excepto en el caso de las aristas que tengan

como origen un símbolo de decisión, en donde el costo será una representación del tipo de decisión (Si/No).

4. Los GD se representan mediante una matriz de adyacencia.

El mecanismo para la comparación de los dos procesos no tiene como objetivo encontrar dos grafos isomorfos, sino encontrar diferencias marcadas entre ambas representaciones (proceso de la empresa y proceso propuesto por MoProSoft) y de esta forma obtener un proceso más robusto de evaluación, que viene a enriquecer a la evaluación por cuestionario presentada en secciones anteriores. Los puntos a tomar en cuenta para este tipo de evaluación son los siguientes:

1. Orden de las actividades.
2. Roles inmiscuidos en el proceso.
3. Los roles responsables de las actividades.
4. El orden de las fases relacionadas con las actividades.
5. Que no exista traslape entre las fases.
6. Que existan actividades relacionadas con las fases de un cierto proceso y nivel de capacidad.
7. Que las fases de un determinado proceso de acuerdo al nivel de evaluación, se encuentren representadas al menos por una actividad.

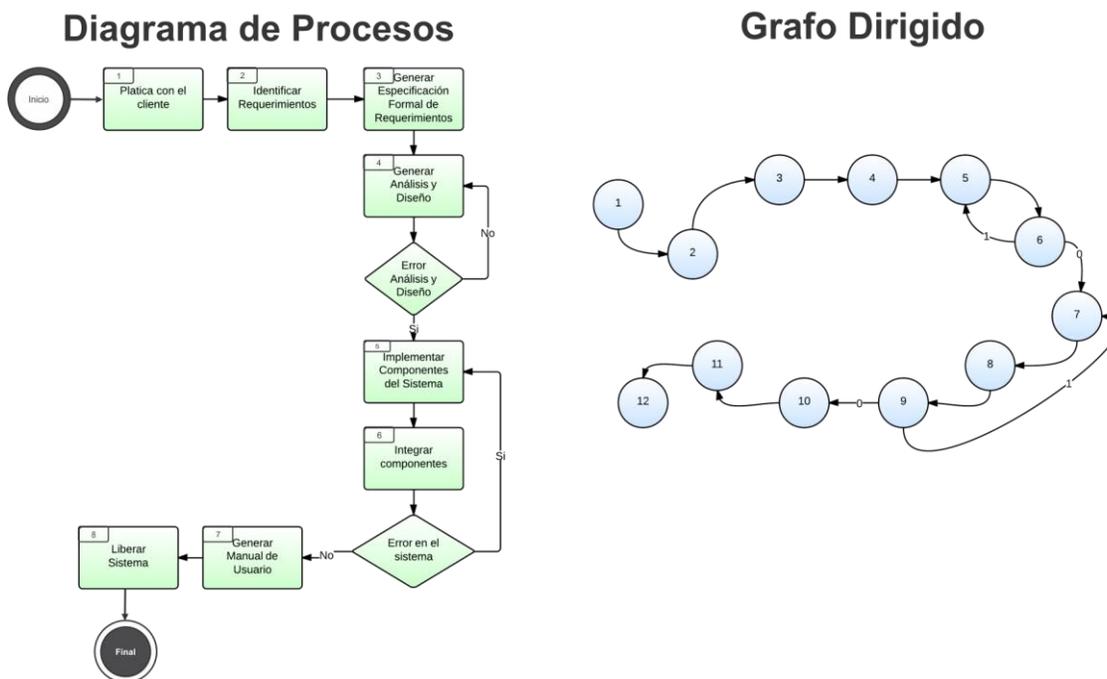


Figura 3.14. Conversión diagrama de procesos a grafo dirigido.

Si se encuentran inconsistencias significativas en una determinada fase o actividades (por ejemplo, que en el diagrama de procesos no se encuentren contempladas algunas actividades relacionadas con la fase de Requerimientos pertenecientes al proceso DMS de MoProSoft), Kaizen establecerá automáticamente la desviación estándar a todas las respuestas del cuestionario relacionadas con esa fase un valor de 1.0, con eso se asegura que esa práctica se deberá de explorar

con mayor profundidad, debido a que no se puede tomar con total veracidad una respuesta en el cuestionario, si en el diagrama de procesos no se consideran las actividades relacionadas con la misma.

3.4.5.3. *Combinación de resultados y obtención de resultado final*

La combinación de ambos métodos de evaluación robustece la valoración de los procesos de la empresa, ya que anomalías no detectadas en una de las técnicas, pueden ser detectadas o complementadas por la otra. El objetivo del método híbrido de evaluación sin embargo no es un detector de falta de veracidad en las respuestas de los evaluados, ya que debe de quedar muy en claro que la única posibilidad de llevar a cabo una iniciativa de mejora es el obtener y mantener un compromiso sólido de cada una de las partes involucradas. Sin embargo, si se pueden detectar anomalías o hallazgos importantes que puedan ayudar a establecer un mejor proyecto de mejora dentro de la organización.

La obtención de los resultados de los cuestionarios en relación a las actividades propuestas por el modelo de procesos utilizado como referencia, se enriquece y complementa por los obtenidos por la evaluación del diagrama del proceso actual de la empresa en comparación por el propuesto por MoProSoft. El porcentaje de cobertura de actividades no se ve afectado por la evaluación por diagrama, el resultado afecta a la desviación estándar para cada actividad. En caso de encontrarse discrepancias considerables en el diagrama de procesos, como pueden ser la omisión de actividades relacionadas con una fase de un proceso a un nivel establecido, que los roles responsables se asignen a actividades donde no corresponden, o que el orden de las fases del proceso no sea el correcto, significa que existen fuertes discrepancias entre las respuestas del cuestionario y el diagrama de procesos actual, por lo cual el valor de la desviación estándar se establece en 1.0 y automáticamente la actividad o fase debe de explorarse con mayor profundidad con el cotejo de productos realizados en dichas actividades o entrevistas cara a cara con los responsables de dichas actividades.

Finalmente, se representa por medio de un diagrama de Kiviat el nivel de cobertura de cada una de las fases de los procesos evaluados, de igual forma se presentan el nivel de cobertura de cada uno de los procesos evaluados según el nivel obtenido.

En este sentido, de acuerdo con [Calvo-Manzano et al., 2002] se ha demostrado que en programas o proyectos de mejora a lo más se pueden realizar tres mejoras de proceso en paralelo. Por lo anterior, Kaizen considera la elección de los procesos a incluir en el proyecto de mejora. Los criterios para seleccionar dichos procesos son los siguientes:

- Nivel de cobertura del proceso. Con este criterio serán procesos candidatos a la selección aquellos cuyo:
 - Nivel de cobertura sea menor o igual que 85%.
 - Nivel de cobertura sea mayor a 85% y tenga prácticas con desviación estándar mayor a 0.8, es decir con fuertes discrepancias en las evaluaciones de los responsables de proceso.
- Objetivos de negocio de la empresa. En base a sus propios objetivos, los responsables de la mejora, definen qué procesos incluir en la iniciativa considerando las prioridades y necesidades de la empresa.

Por último, queda en manos del responsable del proyecto de mejora elegir los procesos que Kaizen le sugiera; la elección de procesos y actividades son importantes, ya que, son los que serán tomados en cuenta para la generación de los planes de mejora.

3.4.6. Incremento para generación de planes de mejora

Para poder implantar la mejora dentro de la organización es necesario obtener una secuencia bien definida de eventos con un principio y un final identificados. Lo anterior muestra que la implantación de la mejora debe ser tratada como un proyecto.

Todo proyecto puede ser dividido en una serie de tareas o actividades bien definidas. Cada actividad tiene un cierto tiempo para ser completada. Algunas tareas pueden ser realizadas simultáneamente, mientras que otras necesitan ser ejecutadas en secuencia. También puede ser necesario definir algunos hitos, u objetivos intermedios, que puedan ser utilizados para controlar el progreso del proyecto antes de que éste finalice. Además, cada tarea requiere de la disponibilidad de los recursos adecuados, tales como personas, herramientas e infraestructura [Gutiérrez, 2008].

Los recursos son los factores productivos necesarios para realizar una actividad, no se consumen con su uso ni se incorporan al producto resultante de la actividad. Por lo tanto, al término de la tarea quedan disponibles para ser utilizados por otras tareas o por otros proyectos. Algunos ejemplos de recursos son la mano de obra, los equipos y las herramientas.

Una vez que los procesos y las actividades han sido evaluados, y se han elegido a aquellos que representan debilidades o tienen mayor prioridad para la empresa, la siguiente fase es generar una planificación para implantar estas mejoras en la infraestructura de la organización. En particular los aspectos a considerar en la planificación del proyecto son los siguientes:

- Orden: Proporcionar el orden de ejecución de las actividades y marcar las dependencias que existan entre ellas.
- Tiempo: Presentar un tiempo estimado acerca del tiempo de ejecución de las diferentes actividades, dando como resultado una calendarización de las mismas, tomando en cuenta fecha de inicio, fecha de finalización y tiempo de tolerancia para finalizar las actividades.
- Recursos: Proporcionar los recursos que utilizan y generan las actividades contempladas en el proyecto de mejora. Los recursos a tomar en cuenta son los miembros del equipo de trabajo involucrados en cada una de las actividades, los productos de entrada y productos generados por la actividad.

Tomando en cuenta lo anterior se debe generar una secuencia la cual muestre el orden de ejecución de las diferentes actividades obtenidas en la fase de evaluación de los procesos, respetando la limitante de recursos, de tal manera que el tiempo de finalización del proyecto sea el menor posible. Este tipo de problema suele ser conocido como Problema de Calendarización de Proyectos con Recursos Limitados (RCPSP, acrónimo en inglés para *Resource Constrained Project Scheduling Problem*).

En su forma más general el RCPSP está definido en [Wall, 1996] de la siguiente manera. Datos:

- Un conjunto de actividades a ejecutar,
- Un conjunto de recursos para llevar a cabo las actividades,
- Un conjunto de restricciones que deben ser satisfechas,
- Un conjunto de objetivos con los cuales juzgar el desempeño de un calendario.

¿Cuál es la mejor manera de asignar recursos a las actividades en momentos específicos, tal que todas las restricciones sean satisfechas?

La forma general incluye las siguientes características:

- Cada tarea se puede ejecutar en más de una forma, dependiendo de qué recursos están asignados a la misma.
- Es necesario considerar las relaciones de precedencia de actividades.
- Cada tarea puede requerir más de un recurso de diversos tipos.
- Los recursos pueden ser renovables (mano de obra, máquinas) o no renovables (materia prima).
- La disponibilidad de los recursos puede variar a lo largo de la duración del proyecto.

Tomando en cuenta la definición y características que presenta un RCPSp, la generación de un plan de mejora, basado en las evaluaciones al proceso de desarrollo software de una empresa, se encuentra dentro de la definición de este tipo de problemas. Las consideraciones a tomar en cuenta para generar la solución a dicha problemática particular son las siguientes:

1. El conjunto de actividades a ejecutar, se encuentra definido por las actividades elegidas en la fase de evaluación.
2. La precedencia de actividades se encuentra definida en el orden de ejecución de las actividades de un determinado proceso que plantea el modelo MoProSoft para las actividades de un determinado proceso.
3. El conjunto de objetivos es definido por los productos que se tienen que conseguir después de la ejecución de ciertas actividades.
4. Los recursos, en este caso renovables, son los miembros del equipo inmiscuidos en el proyecto de mejora, y que juegan un determinado rol dentro del o los procesos a mejorar.
5. La necesidad de recursos humanos para cada actividad se encuentra definida por la delegación de responsabilidades a los diversos roles establecidos por MoProSoft.

En base a las características anteriores, es necesario diseñar un algoritmo para resolver la problemática. En la resolución de este tipo de problemas se han utilizado muchas técnicas como lo muestran estudios publicados en [Wall, 1996] y [Alcaraz y Maroto, 2001], algunos de los métodos utilizados son métodos exactos (e.g. método de la ruta crítica, programación lineal y enumeración acotada) y los métodos heurísticos (p. ej. heurísticas para calendarización, heurísticas para secuencialización, enfoques de inteligencia artificial y algoritmos evolutivos).

La razón del estudio de una gran diversidad de métodos de solución, es que el RCPSp se encuentra categorizado dentro de la clase de complejidad NP-duros (*NP-hard*), por lo que métodos exactos generan soluciones óptimas en pequeños proyectos, generalmente con menos de 50 actividades, y que no están altamente limitados en recursos [Alcaraz y Maroto, 2001].

Al tratarse de un problema de optimización, una técnica que puede utilizarse en este tipo de problemas son los denominados Algoritmos Genéticos (AG), que se han utilizado para optimizar una gran cantidad de problemas. El poder de los AG proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas originados en distintas áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el AG encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable dentro de un tiempo competitivo comparado con el resto de algoritmos de optimización combinatoria [Wall, 1996].

Estudios realizados, como el presentado en [Leon y Barlakrishnan, 1995] demuestran que al comparar el rendimiento de soluciones basadas en métodos heurísticos, el AG supera por mucho los resultados obtenidos por los otros métodos. Diversas investigaciones como [Wall, 1996], [Lee y Kim, 1996], [Mori y Tseng, 1998], [Özsamar, 2000] y [Alcaraz y Maroto, 2001], acerca de la solución del RCPSP, posiciona a los AG como un método de solución eficaz, independientemente del tamaño del proyecto y las restricciones de recursos del mismo.

Para la implementación del algoritmo para generar el plan de mejora se tomaron como base los trabajos de [Wall, 1996] y [Alcaraz y Maroto, 2001]. Debido a que el objetivo del trabajo no se enfoca en la generación de un plan de proyecto de mejora altamente optimizado, sino en generar un plan de mejora que distribuya de manera correcta los recursos de la organización y muestre una aproximación bastante fiable de la distribución de los tiempos y los recursos humanos, la implementación del AG para resolver dicha problemática se encuentra basado en gran medida en las propuestas de los trabajos mencionados anteriormente. Si bien es cierto que las propuestas sufrieron ligeras modificaciones debido a las características propias del problema, no son modificaciones sustanciales con respecto a las propuestas originales.

La Figura 3.15 muestra el pseudocódigo del mecanismo general para un AG. En primera instancia, la población inicial es generada tomando en cuenta la representación empleada para las soluciones. Entonces, cada individuo es evaluado, por ejemplo tomando en cuenta el tiempo de desarrollo del proyecto. Después de eso, los siguientes pasos son repetidos hasta que se cumpla la condición de paro (tiempo de ejecución, número de calendarios o número de generaciones). En el proceso de selección, cada individuo es evaluado según su fortaleza (*fitness*). Entonces, los individuos se cruzan al azar por lo que cada pareja se somete a una cruce para generar descendencia. Finalmente algunos individuos de la población se someten a un proceso de mutación y la población es evaluada de nueva cuenta.

Algoritmo 1 Algoritmo Genético General

- 1: Generar Población Inicial
 - 2: Evaluar Población
 - 3: **Mientras** No se cumpla condición de parada **Hacer**
 - 4: Seleccionar Población
 - 5: Cruzar Población
 - 6: Mutar Población
 - 7: Evaluar Población
 - 8: **Fin Mientras**
-

Figura 3.15. Algoritmo genético general.

En base a lo anterior, se puede establecer un esquema general para un AG considerando lo siguiente [Özsamar, 2000]:

- Existe una población inicial de individuos (cromosomas) con determinadas características.
- Los individuos se cruzan entre sí dando origen a los habitantes de la siguiente generación.
- Estos habitantes heredarán las características de sus padres y si éstas son favorables se irán propagando a través de generaciones futuras, en caso contrario desaparecerán por el proceso de selección natural que indica que sólo los individuos más fuertes sobreviven.
- Durante el proceso de cruce, puede presentarse una mutación en el individuo, si la mutación es mala seguramente el individuo desaparecerá, pero si acarrea beneficios se transmitirá a las generaciones futuras. El proceso de mutación se realiza mediante la alteración de forma

significativa del código genético del individuo, otorgándole nuevas características que nadie posee en la población.

El diseño del AG para resolver el RCPSP en la implantación de la mejora se basa en la estructura general de este tipo de algoritmos (presentada en la Figura 3.15), aplicando las características propias del problema. En las siguientes secciones se presentan las partes que componen el diseño del algoritmo implementado en Kaizen.

3.4.6.1. *Codificación*

La codificación de los individuos de la población se representa mediante un vector $(a_0, a_1, \dots, a_{n+1})$ que indica el orden en el cual se van a ubicar las tareas dentro de la calendarización. Dicho orden se establece en base al orden de ejecución presentado en MoProSoft, con lo cual se simplifica la complejidad del algoritmo. Donde n es el número de actividades, además a_0 y a_{n+1} son actividades de duración nula para denotar el inicio y final del proyecto.

3.4.6.2. *Creación del individuo (cromosoma)*

Para elaborar cada individuo se utiliza una secuencia en serie en base a diferentes aspectos. Las tareas se eligen en base a una serie de etapas (iteraciones), en cada etapa se agrega una nueva tarea al conjunto de tareas secuenciadas que representan al individuo. Existe otro conjunto llamado tareas elegibles que posee todas las posibles tareas a ser secuenciadas. Una tarea es elegible si todos sus predecesores ya han sido secuenciados.

El criterio de selección de una actividad elegible se realiza de forma aleatoria, para poder crear la diversidad de individuos. Al finalizar el procedimiento se obtiene un individuo que garantiza el cumplimiento de todas las restricciones de precedencia. El algoritmo para crear un cromosoma se presenta en la Figura 3.16.

Algoritmo 2 Algoritmo CrearCromosoma

Entrada: $A \leftarrow \{a_0, a_1, \dots, a_{n+1}\}$: conjunto de actividades.

Salida: $CROM$: cromosoma generado.

1: $CROM \leftarrow \{a_0\}$

2: **Mientras** $CROM.length < n+1$ **Hacer**

3: $ELEG \leftarrow \{a_i \in A/P_i \subseteq CROM\}$

4: Seleccionar aleatoriamente una actividad $a \in ELEG$

5: $CROM \leftarrow CROM \cup \{a\}$

6: **Fin Mientras**

Figura 3.16. Algoritmo para crear cromosomas.

3.4.6.3. *Crear la población inicial*

Para crear la población inicial se parte de un conjunto vacío, al cual se van a ir agregando nuevos individuos generados por el proceso anterior. En base a esta población inicial se forma el conjunto de cromosomas que conformarán los pobladores de la primera población de búsqueda a explorar. Además del procedimiento para la creación de individuos, se toma en cuenta una matriz de precedencias que establece la relación de las diversas actividades. Para crear la población inicial se implementa el algoritmo de la Figura 3.17.

Algoritmo 3 Algoritmo CrearPoblacionInicial

Entrada: *TAM*: tamaño de la población inicial**Salida:** *PoblacionInicial* : población inicial generada

- 1: $PoblacionInicial \leftarrow \emptyset$
 - 2: **Mientras** $PoblacionInicial.length < TAM$ **Hacer**
 - 3: $Individuo \leftarrow CrearCromosoma$
 - 4: $PoblacionInicial \leftarrow PoblacionInicial \cup \{Individuo\}$
 - 5: **Fin Mientras**
-

Figura 3.17. Algoritmo para la creación de la población inicial.**3.4.6.4. Fortaleza (fitness)**

Un aspecto importante en los AG es la determinación de la aptitud o fortaleza de los individuos en la población, esta función evalúa la calidad de las soluciones. Dada la naturaleza del problema, la fortaleza de los individuos es el tiempo total de ejecución del proyecto, con lo cual mientras menor sea el valor más fuerte es el individuo.

3.4.6.5. Selección natural

La selección natural es el proceso que permite al algoritmo elegir los mejores individuos de la población y cruzarlos de tal manera que los hijos hereden las mejores características de cada padre. El criterio de selección a utilizar se basa en ordenar a los individuos de mayor a menor según sus fortalezas, y formar las parejas tomando los dos primeros individuos y luego los dos siguientes, y así sucesivamente. El criterio de selección permite que los individuos con mayor fortaleza se reproduzcan con individuos fuertes y los débiles con los débiles. El algoritmo implementado se muestra en la Figura 3.18.

Algoritmo 4 Algoritmo SeleccionNatural

Entrada: *PoblacionInicial*: población inicial generada**Salida:** *Parejas*: Conjunto de parejas para cruzar

- 1: Calcular el fitness de los individuos en *PoblacionInicial*
 - 2: Ordenar *PoblacionInicial* de mayor a menor según el fitness
 - 3: $Parejas \leftarrow \{(1, 2), (3, 4), \dots, (p - 1, p)\}$
-

Figura 3.18. Algoritmos de selección natural.**3.4.6.6. Operador de cruce**

El operador de cruce es la función mediante la cual se crean los nuevos individuos. Por medio de una pareja de cromosomas se generan los hijos de la cruce de ambos individuos. Los hijos deben garantizar el cumplimiento de las restricciones de precedencia, por lo cual deben seguir los siguientes pasos para crear a los nuevos individuos (véase Figura 3.19):

1. Se genera un número aleatorio u entre 1 y n , el cual representa el punto de cruce entre los cromosomas.
2. El primer hijo hereda los genes del padre de izquierda a derecha hasta u , y completa su secuencia heredando los de la madre, de izquierda a derecha que aún no formen parte del nuevo individuo.
3. De forma similar, el segundo hijo hereda los genes de la madre de izquierda a derecha hasta u , y completa su secuencia genética heredando los genes del padre de izquierda a derecha que aún no formen parte de la misma.

Con lo anterior se garantiza la creación de una pareja de hijos por cada pareja de padres que cumpla con todas las restricciones de precedencia.

Algoritmo 5 Algoritmo Cruzar

Entrada: $Padre$: Individuo Padre, $Madre$: Individuo Madre
Salida: $Hijo_1$: Nuevo individuo 1, $Hijo_2$: Nuevo individuo 2

```

1:  $u \leftarrow Random(1, n)$ 
2: Para  $i \leftarrow 1$  hasta  $u$  Hacer
3:    $Hijo_1[i] = Padre[i]$ 
4:    $Hijo_2[i] = Madre[i]$ 
5: Fin Para
6:  $m \leftarrow u$ 
7: Para  $i \leftarrow 1$  hasta  $n$  Hacer
8:   Si  $Madre[i] \notin Hijo_1$  Entonces
9:      $u \leftarrow u + 1$ 
10:     $Hijo_1[u] \leftarrow Madre[i]$ 
11:   Fin Si
12:   Si  $Padre[i] \notin Hijo_2$  Entonces
13:      $m \leftarrow m + 1$ 
14:     $Hijo_2[m] \leftarrow Padre[i]$ 
15:   Fin Si
16: Fin Para

```

Figura 3.19. Representación algorítmica del operador de cruce.

3.4.6.7. *Mutación*

La mutación es un mecanismo por el cual se busca la posibilidad de salir de un entorno de búsqueda monótono y pasar a otro que probablemente tenga entre sus habitantes al individuo óptimo, debido a esto juega un papel fundamental en cualquier AG. Cuando se aplique algún tipo de mutación a alguno de los nuevos miembros de la población se debe asegurar que el individuo mutado aún pertenezca a la misma especie que los demás habitantes, en el contexto del problema esto significa que aún preserve las restricciones de precedencia que forman parte de la ejecución de las actividades. El algoritmo para realizar la mutación en los individuos (véase Figura 3.20) consiste en intercambiar un par de genes del nuevo individuo, siempre y cuando estos genes no tengan relaciones de dependencia, tomando en cuenta el siguiente procedimiento:

1. Generar un número aleatorio u entre 1 y $n - 1$.
2. Si la actividad ubicada en el puesto u de la secuencia genética no es predecesora de la actividad situada en la posición $u + 1$ de la misma secuencia, entonces intercambiar ambas actividades. En caso contrario regresar al paso 1.

3.4.6.8. *Criterio de parada*

La finalización del AG por lo general puede establecerse por diferentes criterios, por ejemplo:

- Alcanzar un número máximo de generaciones.
- Alcanzar el tiempo máximo de resolución.
- Encontrar una solución que satisfice las condiciones mínimas deseadas.

- Alcanzar una solución de tal manera que las siguientes generaciones no producen una mejora sustancial.
- Combinación de dos o más criterios.

Algoritmo 6 Algoritmo Mutación

Entrada: *Individuo* : Individuo que sufrirá mutación, *P* : Conjunto de precedencias

Salida: *IndividuoMutado*: Individuo mutado

- 1: $u \leftarrow \text{Random}(1, n)$
 - 2: $\text{IndividuoMutado} \leftarrow \text{Individuo}$
 - 3: **Si** $\text{Individuo}[u] \notin P_{\text{Individuo}(u+1)}$ **Entonces**
 - 4: $\text{IndividuoMutado}[u + 1] = \text{Individuo}[u]$
 - 5: $\text{IndividuoMutado}[u] = \text{Individuo}[u + 1]$
 - 6: **Fin Si**
-

Figura 3.20. Algoritmo de mutación.

En el caso de la solución implementada, se ha elegido el primer criterio. La solución generada por el AG, después de alcanzar el criterio de parada, es aquel individuo que durante todo el proceso evolutivo ha sido el más fuerte en concordancia con el problema, esto es minimizar el tiempo total de ejecución del proyecto.

3.4.6.9. Creación de la calendarización

La ejecución del AG no consiste en determinar los tiempos de inicio de cada actividad, sino únicamente elaborar la secuencia de ejecución de las actividades y obtenida esa secuencia, obtener y asignar los tiempos de inicio de cada tarea.

Para ubicar las actividades en la calendarización final se siguen los siguientes pasos:

1. Seleccionar la actividad que corresponda según el orden establecido en la secuencia.
2. Ubicar tiempo de inicio de la actividad seleccionada en el primer día donde se satisfaga la restricción de disponibilidad de recursos y todos los predecesores de la actividad hayan concluido.
3. Si ya se han ubicado todas las actividades en el calendario, finalizar el proceso y la fortaleza de la secuencia de actividades será el tiempo de inicio de la última actividad agregada a_{n+1} , en caso contrario regresar al paso 1.

El proceso obtiene la calendarización del mejor individuo (véase Figura 3.21), necesaria para generar el plan de actividades.

Algoritmo 7 Algoritmo MejorIndividuo

Entrada: $Poblacion$: Poblacion, R : Conjunto de restricciones de recursos, P : Conjunto de precedencias, D : Conjunto de duración de actividades

Salida: $MejorIndividuo$: Mejor Individuo de la población

```

1:  $fitnessPoblacion \leftarrow \infty$ 
2: Para todo  $Individuo \in Poblacion$  Hacer
3:    $fitnessIndividuo \leftarrow 0$ 
4:    $Calendarizacion \leftarrow \emptyset$ 
5:   Para todo  $a \in Individuo$  Hacer
6:      $liberarRecursos(a_i, fitnessIndividuo, R)$ 
7:      $Tarea \leftarrow \{a_i / P_i \subseteq Calendarizacion \wedge r_i = Verdadero\}$ 
8:      $fitnessIndividuo \leftarrow fitnessIndividuo + d_i$ 
9:      $Calendarizacion \leftarrow Calendarizacion \cup \{Tarea\}$ 
10:  Fin Para
11:  Si  $fitnessIndividuo < fitnessPoblacion$  Entonces
12:     $fitnessPoblacion \leftarrow fitnessIndividuo$ 
13:     $MejorIndividuo \leftarrow Calendarizacion$ 
14:  Fin Si
15: Fin Para

```

Figura 3.21. Algoritmo para obtener calendario de mejor individuo.

3.4.6.10. Algoritmo para generar el plan de actividades

Para generar el plan de actividades se utilizan todos los algoritmos presentados en las secciones anteriores, anexando procedimientos auxiliares necesarios para el buen funcionamiento del algoritmo completo. La Figura 3.22 presenta el algoritmo principal para la generación del plan de actividades.

Algoritmo 8 Algoritmo ObtenerPlanActividades

Entrada: A : Lista de actividades a implantar en el proyecto de mejora

Salida: Calendario de Actividades

```

1:  $Precedencia \leftarrow CrearMatrizPrecedencia(A)$ 
2:  $PoblacionInicial \leftarrow CrearPoblacionInicial(TAM)$ 
3:  $PoblacionInicial \leftarrow SeleccionNatural(PoblacionInicial)$ 
4:  $MejorIndividuo \leftarrow MejorIndividuo(PoblacionInicial, Precedencia, R, D)$ 
5:  $Generacion = PoblacionInicial$ 
6: Para  $i \leftarrow 1$  hasta  $\#Generaciones$  Hacer
7:    $Hijos \leftarrow Cruzar(Padre, Madre)$ 
8:    $Generacion \leftarrow Generacion \cup Hijos$ 
9:    $Mutacion(IndividuoAleatorio(Generacion), Precedencia)$ 
10:   $Generacion \leftarrow SeleccionNatural(Generacion)$ 
11:   $MejorIGeneracion = MejorIndividuo(Generacion)$ 
12:  Si  $MejorIGeneracion$  es mejor que  $MejorIndividuo$  Entonces
13:     $MejorIndividuo \leftarrow MejorIGeneracion$ 
14:  Fin Si
15: Fin Para
16:  $Graficar(MejorIndividuo)$ 

```

Figura 3.22. Algoritmo para generar plan de actividades

3.4.7. Incremento para implantación de planes de mejora

Una vez definidos los planes de acción, éstos deben de implementarse. Para ello se tienen que realizar las actividades definidas previamente en el plan, como por ejemplo definir los nuevos procesos. También es necesario experimentar estos procesos en proyectos piloto para asegurarse que los nuevos procesos y actividades que se acaban de definir funcionan en el entorno de la empresa. Por último, es necesario realizar una difusión de estos procesos a toda la empresa en general para que todo el personal los utilice en su trabajo diario.

La implantación consiste en aplicar el plan genérico de mejora obtenido a cada uno de los procesos que desean mejorarse, es decir llevar a cabo cada uno de los planes de acción generados. Cada uno de estos planes de acción corresponde a una de las diferentes mejoras que se van a implantar en la organización [Calvo-Agustín et al., 2002]. La implantación de los planes de acción conlleva principalmente dos actividades: implantación de los procesos definidos en un proyecto piloto y divulgación o institucionalización de estos procesos en la organización.

Como se ha planteado en secciones anteriores Kaizen busca ser una herramienta integral para establecer iniciativas de mejora de procesos dentro de la organización, por lo que es parte fundamental el soporte a la fase de implantación de la mejora. En la sección anterior se describió como se automatiza la fase de generación de planes de acción de mejora, con ello se obtuvieron fechas de inicio y secuencia de las actividades. Cada actividad es responsabilidad de un rol dentro de la organización, el cual previamente fue dado de alta como miembro del equipo de proyecto. Además, algunas de las actividades que forman parte del proceso a mejorar se encuentran relacionadas con un producto de entrada o salida, en este caso los productos pueden fungir como indicadores para comprobar que una actividad ha sido realizada. Las relaciones de actividad y producto se definen en el modelo MoProSoft.

Por lo anterior, el motor de control para la implantación y realización de las actividades del plan de mejora se basa en la detección de hitos, los cuales pueden presentarse por los siguientes motivos:

1. Asignación de actividad. Al comienzo de la ejecución del proyecto de mejora, Kaizen notifica a los miembros del equipo las responsabilidades delegadas y distribuye la información de las actividades de las que son responsables.
2. Inicio de actividad. La fecha del sistema coincide con la fecha de inicio de cierta actividad, por lo cual se generan mensajes a los responsables para informarles que tienen que atender la realización de la tarea calendarizada.
3. Próximo vencimiento. La fecha del sistema es cercana a la fecha de terminación de una actividad, por lo que el sistema alerta al responsable la fecha próxima de vencimiento para que tome las medidas necesarias para evitar un desfase en el proyecto.
4. Vencimiento de actividad. La fecha del sistema coincide con la fecha de terminación de una actividad, y en caso de no haberse finalizado, el sistema envía notificaciones a los responsables tanto de la actividad como del proyecto para informar el incumplimiento de la tarea.
5. Finalizar actividad. En el momento que un responsable notifica al sistema que se ha concluido una actividad, Kaizen emite mensajes al responsable del proyecto para hacer de su conocimiento la finalización de la tarea, de igual forma el sistema emite alertas a los responsables de actividades posteriores según la secuencia del proyecto o las cuales necesiten los productos de la actividad para poder realizar su trabajo.

6. Mensajes y/o anuncios. Los miembros del equipo pueden emitir anuncios en el sistema para enterar a los demás integrantes sobre situaciones o incidencias que se pudiesen presentar durante la ejecución de la mejora. De igual forma el responsable de la mejora puede enviar mensajes directos a miembros del equipo para comunicar aspectos importantes sobre las actividades que debe de realizar.

En base a los hitos anteriores es que se desarrolla el motor de control necesario para auxiliar al responsable del proyecto a implantar la mejora dentro de la organización. Los hitos 1, 5 y 6 son generados por algún miembro del equipo de trabajo, por lo que Kaizen se encarga de enviar la información pertinente a los usuarios deseados. El evento desencadenador es realizado por los usuarios, lo que da pie a la ejecución de distintos procedimientos por parte del sistema para actualizar la base de datos con el fin de emitir alertas, mensajes o notificaciones.

Por otro lado, los acontecimientos 2, 3 y 4 no se realizan en base a una acción del usuario, sino que deben de ejecutarse automáticamente en base a las fechas de control generadas por el sistema en la fase de generación de planes de mejora (véase Figura 3.23). Para implementar la funcionalidad deseada, se hace uso de tareas programadas (*cron jobs*) que se ejecutan en función de un intervalo de tiempo especificado. De esta forma, se pueden ejecutar ciertas tareas de un modo automático en el servidor, donde se aloja el sistema Kaizen. Las tareas se ejecutan en segundo plano en el servidor sin necesidad de un control por parte del usuario. La tarea programada se configura para ejecutarse cada 24 horas, en donde se ejecuta un *script* que realiza una consulta a la BD en base a la fecha del sistema, para verificar si existen actividades por comenzar, próximas a vencer y vencidas. En caso de detectar cualquiera de los casos anteriores se actualiza un registro en la BD, para que en la próxima ejecución de Kaizen se muestren las alertas o notificaciones dependiendo del tipo de acontecimiento detectado.

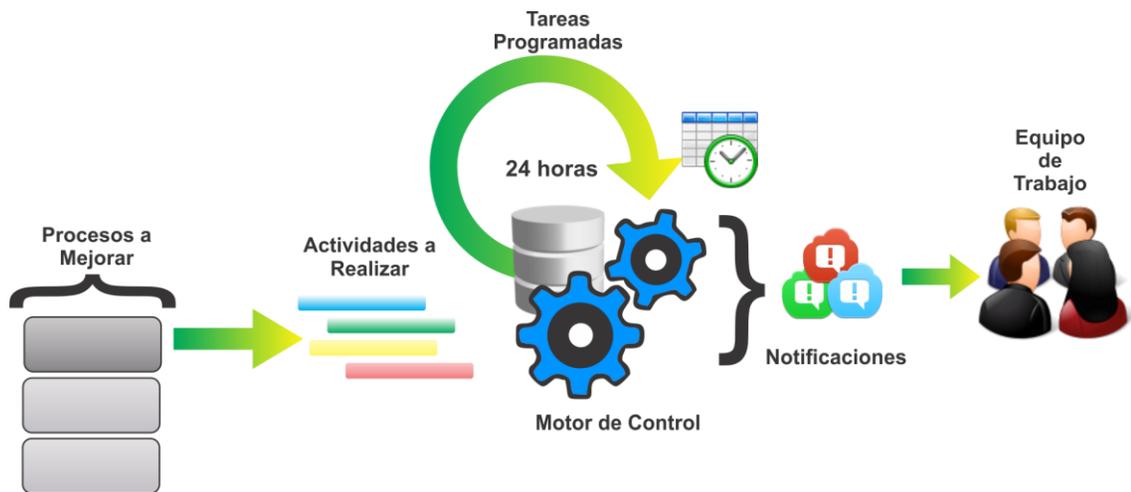


Figura 3.23. Control de implantación de actividades de mejora.

4. Resultados experimentales

En este capítulo se presentan los resultados obtenidos al utilizar la herramienta desarrollada en este trabajo dentro de organizaciones desarrolladoras de software. El uso de Kaizen se limita a las características de cuatro MPyME desarrolladoras de software. Los resultados experimentales se obtienen partir de la ejecución de un caso de estudio que permita establecer una iniciativa de mejora basada en MoProSoft dentro de la estructura organizacional de las empresas. Como parte del caso de estudio se definen proyectos piloto dentro de las organizaciones, para poner en marcha los proyectos de mejora y comparar los resultados antes y después de la implantación de la mejora asistida por Kaizen. Por último, se presentan las mejoras alcanzadas por las MPyMEs desarrolladoras de software y los hallazgos obtenidos por medio del uso de Kaizen.

Un caso de estudio se utiliza habitualmente como una estrategia de investigación empírica en el campo de los sistemas de información, a menudo es utilizado para describir las relaciones dentro de la organización. En particular 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 periodo de tiempo. Si se desean comparar dos métodos, puede ser necesario realizar la experimentación como un caso de estudio o un experimento. La elección depende de la escala de evaluación. En base a lo anterior, Calvo-Manzano et al. [2002] propone el uso de proyectos piloto para evaluar los efectos de un cambio con respecto a alguna línea base. Los casos estudio son adecuados para la evaluación industrial de métodos y herramientas de Ingeniería de Software porque pueden evitar problemas de escalado [Wohlin et al., 2012]. Una ventaja de los casos de estudio es que su planificación resulta más fácil, pero la principal desventaja es que los resultados son difíciles de generalizar e interpretar, es posible mostrar los efectos en una situación típica, pero no pueden generalizarse a cualquier situación [Yin, 2009].

Para obtener los resultados de esta tesis se han escogido a los casos de estudio como estrategia empírica de experimentación, para sustentar así la validez de la hipótesis planteada al inicio de este trabajo. En este sentido, existen tres diferentes estrategias para desarrollar un caso de estudio [Wohlin et al., 2012]:

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

Para la experimentación de este trabajo, se opta por elegir la primera de las estrategias anteriores, para lo cual se consideran como línea base los resultados obtenidos en una primera evaluación de los procesos de la empresa, con lo cual se obtienen las fortalezas y debilidades de éstas antes de utilizar Kaizen para implantar una iniciativa de mejora. Después de la ejecución de los planes de acción generados por la herramienta y de que la mejora haya sido implantada, la fase de

evaluación es repetida para comparar los resultados antes de implantar cualquier tipo de mejora y después de utilizar Kaizen para implantar una mejora de procesos dentro de la organización.

Con la experimentación se busca validar la hipótesis presentada al inicio de este trabajo, en la cual se plantea que Kaizen apoyaría a las MPyME desarrolladoras de software a implantar mejoras de procesos, basadas en las prácticas efectivas presentadas en el modelo MoProSoft. Lo anterior considerando un mecanismo automatizado que cumpla con las fases genéricas de un ciclo de mejora: *Compromiso, Evaluación, Creación de Planes de Acción, e Implantación de la Mejora*.

A través de la validación, los jefes de proyecto de cada empresa participan en una evaluación del proceso usado dentro de su organización, para obtener una línea base que será comparada con la posterior aplicación de la mejora con la ayuda de la herramienta. Una vez que el plan de acción generado es implantado y utilizado, la fase de evaluación es repetida para comparar los resultados antes de realizar la mejora y después de implantarla con la ayuda de Kaizen.

4.1. Definición del caso de estudio

El caso de estudio se enfoca en la aplicación de una mejora de proceso asistida por la herramienta Kaizen en empresas mexicanas desarrolladoras de software. Un aspecto importante para la elección de las empresas participantes, es la disposición a colaborar con las actividades a realizar durante el ciclo de mejora, así como el establecimiento de un compromiso serio, para disponer de recursos en materia de personal, instalaciones y tiempo necesarios para llevar por buen camino la iniciativa de mejora de procesos dentro de la organización. Lo anterior es indispensable, puesto que, la falta de compromiso es el principal factor en el fracaso de iniciativas de SPI dentro de las empresas. Otro factor importante a tomar en cuenta es ubicar a las empresas sobre las expectativas que pueden tener sobre este tipo de proyectos, ya que si los objetivos y resultados esperados por los directores no son realistas, el esfuerzo de mejora del proceso puede ser un fracaso. Por último, se debe poner en claro a la empresa que la consecución de un nivel de MoProSoft no debe convertirse en el principal objetivo. Muchas empresas pueden ver la consecución de un nivel de madurez específico de MoProSoft como el objetivo principal de la mejora de procesos, lo que puede ser perjudicial para el negocio, por lo contrario la mejora de procesos y MoProSoft deben verse como mecanismos para alcanzar los objetivos reales de la empresa, lo que en un futuro tendrá como resultado natural la consecución de un nivel específico del modelo.

Después de establecer las limitaciones mencionadas, cuatro MPyME desarrolladoras de software estuvieron interesadas en formar parte de la experimentación, por motivos de confidencialidad durante el estudio se hace referencia a ellas como ES1, ES2, ES3 y ES4. La Tabla 24 describe las características de cada una de estas empresas.

Tabla 24. Perfil de las empresas participantes en el caso de estudio.

MPyME	Número de Empleados	Actividad Principal	Experiencia
ES1	10	Desarrollo de software a la medida.	Experiencia en modelo de procesos y programas de mejora.
ES2	8	Servicios de hardware & software y soluciones integrales.	Sin experiencia alguna.
ES3	15	Desarrollo de software empaquetado y a la medida.	Conocimiento limitado sobre modelos de procesos y programas de mejora.
ES4	5	Mantenimiento y desarrollo de software para pequeñas empresas.	Conocimiento limitado sobre modelos de procesos y programas de mejora.

Las MPyME comparten la misma estructura administrativa, dado su tamaño, la cual se encuentra compuesta por tres departamentos:

- Departamento administrativo. Se encarga de la dirección de la empresa a nivel técnico-administrativo. Es controlado por un director con conocimiento y/o relacionado con el área de TI.
- Departamento contable. Se encarga de realizar todas las gestiones de índole contable con los clientes y con la planta de trabajadores. Este departamento es controlado por un director relacionado con la gestión empresarial.
- Área de desarrollo. Es el área encargada de desarrollar los proyectos de software solicitados por los clientes. Dentro de este personal se encuentran los jefes de proyecto de cada empresa que participan en la evaluación y posterior ejecución de la mejora de procesos.

En relación a la experiencia de las cuatro entidades, los responsables de las empresas expresaron los siguientes puntos:

- Las cuatro MPyME se han enfocado en el desarrollo de software a la medida con el objetivo de brindar soluciones tecnológicas al mercado regional por más de 4 años. Regularmente los problemas en la gestión les impiden hacerse cargo de un mayor número de proyectos.
- ES1 es la que cuenta con mayor experiencia en la implantación de modelos de procesos, ha realizado esfuerzos en el pasado en la adopción del MoProSoft sin llegar a conseguirlo; ES3 y ES4 tienen un conocimiento limitado sobre los modelos de procesos y programas de mejora, sin embargo nunca los han utilizado; y por último ES2 no tiene experiencia en este tipo de actividades.

Una vez definidas las características de las cuatro empresas, el caso de estudio plantea evaluar la efectividad del método organizacional para la aplicación del sistema Kaizen en la identificación de factores a mejorar y posteriormente su aplicación, evolución, control e implantación dentro de las cuatro empresas pequeñas. En las siguientes secciones del capítulo se muestran los resultados de la línea base inicial, los resultados de la implantación de la mejora, los resultados obtenidos después de implantar las mejoras basadas en MoProSoft mediante la herramienta Kaizen y por último, la comparativa de la línea base y los resultados después de la implantación, y el análisis detallado sobre los resultados de la mejora.

4.1.1. Establecimiento de la línea base inicial

El primer paso del caso de estudio, es obtener una línea inicial que sirva como base en la comparación del “antes y después” de implantar la mejora mediante Kaizen. Para conseguir este objetivo se decide que debido a las características de las MPyME participantes (principalmente la poca o nula experiencia en la conducción de iniciativas de mejora) se incluyen solamente los procesos relacionados con la categoría de Operación de MoProSoft, específicamente Administración de Proyectos Específicos (APE), y Desarrollo y Mantenimiento de Software (DMS). La anterior decisión fue inducida por la razón que el modelo MoProSoft e ISO 29110 recomiendan iniciar por estos procesos para las MPyME que carezcan de experiencia en el uso de modelos de procesos dentro de su estructura [Oktaba, 2010; ISO, 2011a]. Lo anterior, permite identificar las debilidades en los procesos seleccionados y establecer la línea base que servirá de comparativa en una segunda evaluación después de la implantación de la mejora a partir del modelo MoProSoft y con la ayuda de Kaizen. Como parte de la experimentación se ha participado como parte del equipo de consultoría que proporciona las directrices necesarias para los procesos de evaluación, control e implantación de la mejora mediante el uso de la herramienta desarrollada. En este sentido, se formaron equipos de evaluación conformados por el equipo consultor y un jefe de proyecto de mejora de cada MPyME, y

posteriormente se seleccionaron dos jefes de proyectos más de cada empresa con el objetivo de evaluar las áreas de proceso definidas anteriormente.

Así, para comenzar se debe de configurar la iniciativa de mejora, como se mencionó anteriormente se seleccionaron los procesos de la categoría de Operación en su nivel 1, de igual forma se seleccionaron los jefes de proyectos relacionados con los roles de responsables de los procesos mencionados. La configuración es necesaria, ya que más allá de la fase de evaluación ésta es parte importante dentro de la parte de control e implantación. Delegados los roles dentro de Kaizen para cada uno de los miembros del equipo de trabajo, los jefes de proyecto (responsables de procesos) fueron sometidos a la fase de evaluación (presentada en el capítulo anterior) basada en cuestionarios (véase número 1 de la Figura 4.1) y el modelado de los diagramas de procesos (véase número 2 de la Figura 4.1).

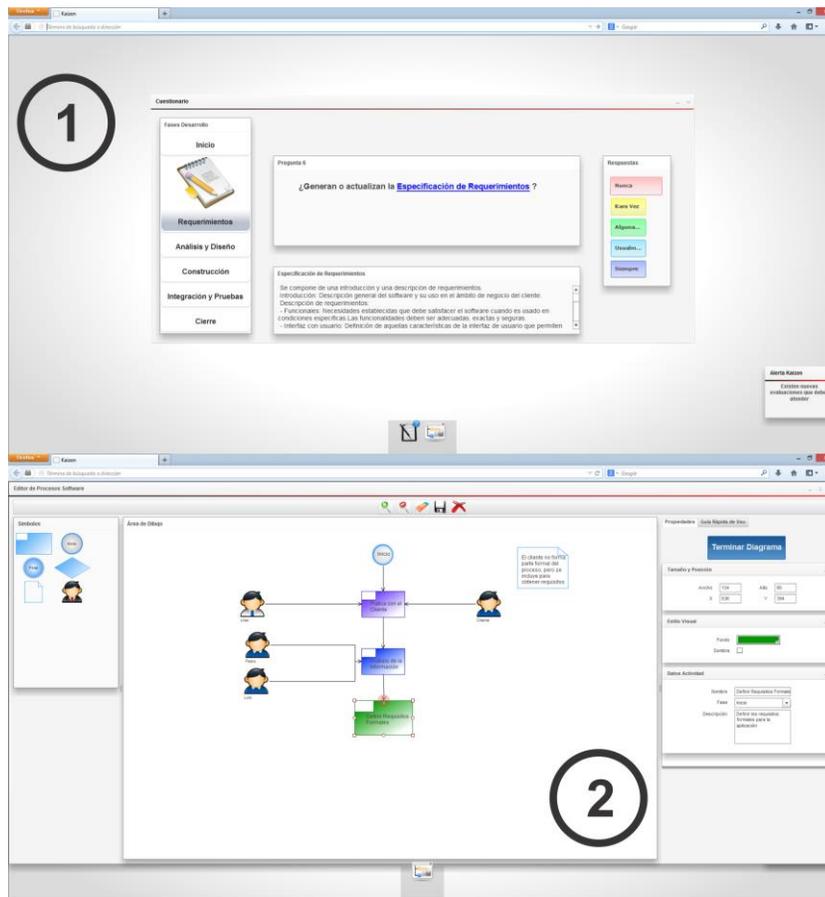


Figura 4.1. Fase de evaluación presentada en Kaizen: (1) basada en cuestionarios, (2) a través de modelado.

De esta forma, el mecanismo de evaluación propuesto en Kaizen proporciona información útil relacionada con el estado actual de los procesos de las MPyME, indicando aquellas prácticas que requieren atención inmediata. Así, por medio del mecanismo de evaluación se determinó el nivel de realización de las prácticas específicas del modelo MoProSoft para los procesos APE y DMS; estos resultados junto con los obtenidos en la tercera fase de la experimentación, permitió determinar qué actividades habían sido cubiertas e implementadas por el equipo de trabajo y cuáles no se han terminado de institucionalizar dentro de la organización.

Con el objetivo de establecer la situación actual o línea base sobre los procesos de las cuatro MPyME, la evaluación se realizó mediante la fase de evaluación presentada en el Capítulo 3 de esta tesis, donde en primera instancia se obtienen los resultados del cuestionario (nivel de cobertura por

fase) y éstos son enriquecidos con el mapeo del diagrama obtenido con el diagrama de proceso presentado por MoProSoft. La Figura 4.1 muestra las pantallas de Kaizen relacionadas con la fase de evaluación, en el número 1 se muestra el formato de cuestionario presentado para el proceso de Desarrollo y Mantenimiento de Software (DMS) y el número 2 muestra el editor de diagramas de proceso. Para finalizar la fase de evaluación, todos los responsables de proceso deben de culminar ambos tipos de evaluaciones, ya que en base a las respuestas y modelos generados por cada jefe de proyecto se obtiene la línea base inicial. Una vez culminada la evaluación los resultados son presentados a los encargados de la iniciativa de mejora dentro de la organización, presentando la cobertura por actividad y por fase, y denotando las actividades que necesitan de una atención urgente dentro del proceso de desarrollo actual. Cada una de las MPyME obtuvo los resultados en base a las respuestas y modelos generados por los jefes de proyecto evaluados (véase número 1 de la Figura 4.2). Una vez concluidas las evaluaciones para todos los jefes de proyecto involucrados en el proyecto de mejora por parte de cada empresa, se presenta el análisis de los resultados tomando en cuenta la cobertura y desviación estándar por actividad, en base a estos resultados se obtienen los puntos débiles (color rojo), puntos fuertes (color verde) y las actividades que deben de explorarse con mayor profundidad para poder dilucidar si constituye punto fuerte o un aspecto a mejorar (color amarillo). Una vez obtenidas las actividades en las cuales se debe de enfocar la mejora, el responsable de mejora dentro de la MPyME debe elegir las actividades con las cuales trabajar durante la implantación (véase número 2 de la Figura 4.2).



Figura 4.2. Pantalla de resultados de Kaizen: (1) para jefes de proyecto, (2) selección de actividades.

Es importante mencionar que las figuras mostradas solamente son ilustrativas y representan una pequeña muestra de la fase de evaluación presentada en Kaizen. Para más detalle ver Anexo C.

La Figura 4.3 muestra el nivel de cobertura obtenido por las MPyME para los procesos de APE y DMS del modelo MoProSoft, incluyendo una línea horizontal que representa el promedio de cobertura por proceso. La gráfica de nivel de cobertura resume el porcentaje de actividades que se realizan por cada proceso, antes de utilizar Kaizen para implantar una mejora dentro de cada MPyME. La gráfica muestra que en la evaluación inicial se obtuvo un 36.6% de cobertura promedio para el proceso de APE y del 50.1% para el proceso de DMS. Lo anterior muestra que el proceso de DMS se encontraba ligeramente mejor implementado dentro de las empresas que APE, dicho resultado es comprensible debido a que el proceso DMS describe las actividades generales y básicas con las que un proceso de desarrollo de software debe de contar, y las MPyME evaluadas aunque con procesos de desarrollo *ad-hoc*, cuentan con algunas actividades o prácticas que son recomendadas por el proceso. Pese a lo anterior, la cobertura inicial (50.1%) se encuentra muy por debajo de la establecida por MoProSoft (85%) para alcanzar el Nivel 1 de cobertura. En el caso del proceso APE, la pobre cobertura es aún más evidente pues solamente alcanza el 36.6%, siendo la ES1 con el más alto porcentaje (42.5%). La pobre implementación de este proceso en la estructura de las MPyME demuestra lo poco que se trabaja en materia de gestión dentro del desarrollo de un proyecto, ya que el proceso APE se enfoca en dicho rubro, realizando recomendaciones acerca de ciclos de desarrollo, estimación, calendarización, capacitación, adquisición y formalización de plazos de entrega; lo que demuestra que las MPyME se enfocan muy poco a este tipo de actividades, lo cual ha quedado demostrado que la mayoría de las veces puede conducir al fracaso en un proyecto.

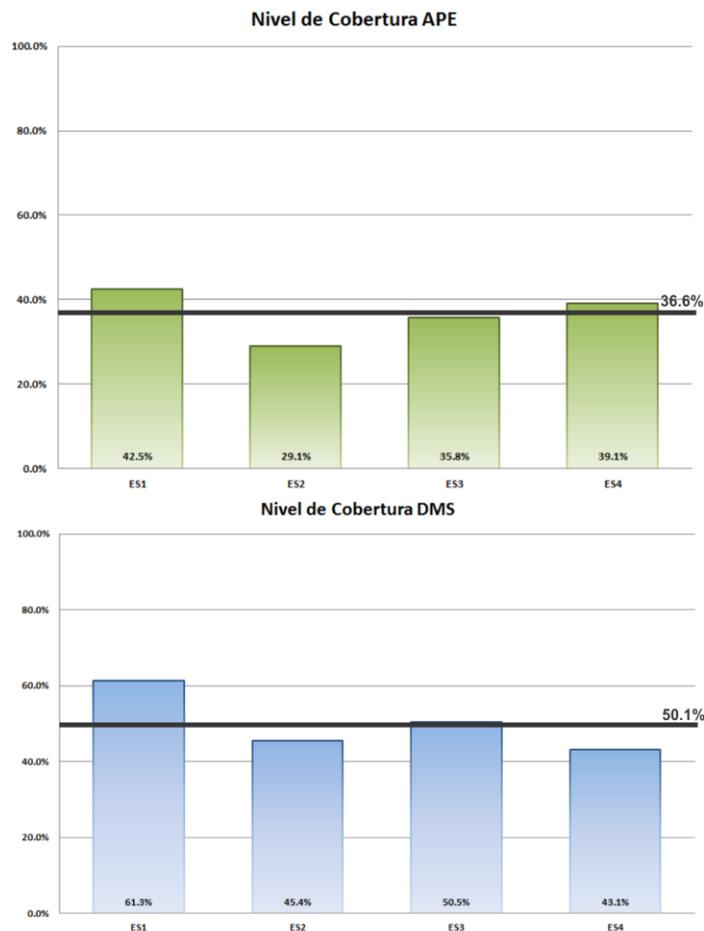


Figura 4.3. Nivel de cobertura de las MPyME para los procesos de la categoría Operación.

Los resultados obtenidos son evidencia clara que los procesos evaluados (en el Nivel 1) se encontraban pobremente implementados en las organizaciones. A partir de dicha evidencia, se enfocaron los esfuerzos de mejora en ambos procesos, puesto que representan las actividades básicas que una empresa desarrolladora de software debe de realizar independientemente de su tamaño.

4.1.2. Implantación de la mejora por medio de Kaizen en un proyecto piloto

Culminada la fase de evaluación, la implantación de la mejora se inicia con la definición del nuevo proceso que será utilizado en un proyecto piloto a desarrollar dentro de las MPyME (el mismo proyecto piloto para las cuatro). Para conseguir tal objetivo se siguieron los pasos descritos a continuación:

1. Definir contexto de la mejora. Con el objeto de familiarizar a los participantes de la mejora con los nuevos conceptos, se realiza un taller de formación al equipo de trabajo constituida por una capacitación inicial en el proceso de mejora, proporcionar una visión global del proceso a mejorar y proporcionar aspectos técnicos generales del proceso. La actividad es apoyada por Kaizen brindando manuales y textos a los involucrados en la mejora.
2. Definir actividades y proceso a mejorar. Para la experimentación de este trabajo, se eligieron dos procesos APE y DMS, mientras que las actividades fueron elegidas por cada responsable de mejora en las empresas, en base a los objetivos y necesidades de cada organización. Como se mostró en la Figura 4.2 de la sección anterior, Kaizen permite a los responsables la elección de tales actividades después del proceso de evaluación. A partir de dicha elección, Kaizen genera un nuevo proceso el cual debe ser establecido en las empresas, además realiza la planificación del nuevo proceso estableciendo el orden de las actividades, dependencias, calendarización y responsables de cada actividad (véase Figura 4.4). El nuevo proceso es generado a partir del mecanismo presentado en la sección 3.4.6.
3. Seleccionar un proyecto piloto. Los proyectos piloto deben tener una duración del orden de 3 a 5 meses para que los resultados de la mejora sean visibles en un corto espacio de tiempo, no deben ser críticos, pero su resultado si debe de ser importante para la empresa con objeto de que ésta sea sensible al proceso de mejora.
4. Realizar el proyecto piloto. Llevar a cabo la implantación del proyecto piloto de acuerdo al calendario y nuevo proceso obtenidos. Kaizen cuenta con el módulo de control de implantación, con lo cual los jefes de proyecto pueden monitorear las actividades que desempeñan los miembros del equipo de trabajo, con el fin de llevar por buen camino la implantación del nuevo proceso durante el desarrollo del proyecto piloto.
5. Evaluar el comportamiento de los nuevos procesos. Se evalúa el comportamiento de los nuevos procesos.

En este sentido, el proyecto piloto consistió en el desarrollo de un software para gestionar un proceso de venta de productos, es decir un punto para venta de menudo y mayoreo. El objetivo principal del software fue optimizar la cobranza y gestión de la información de ventas y clientes de una empresa en particular. La descripción del proceso y las funcionalidades necesarias para desarrollar el sistema fueron entregadas a cada una de las empresas con el fin de que ejecutarán todas las actividades presentadas en el nuevo proceso. La decisión de utilizar el mismo proyecto para las cuatro empresas se basa en que se contaba con un entorno de desarrollo similar: mismo problema, mismo procesos de desarrollo y producto a obtener, similares condiciones de trabajo, el mismo nivel de complejidad, las mismas funcionalidades y limitaciones muy parecidas en las cuatro

MPyME. Esto permitió comparar los resultados al final y determinar el por qué una organización actuó mejor que la otra, y cuáles son las actividades y procesos que se han mejorado.

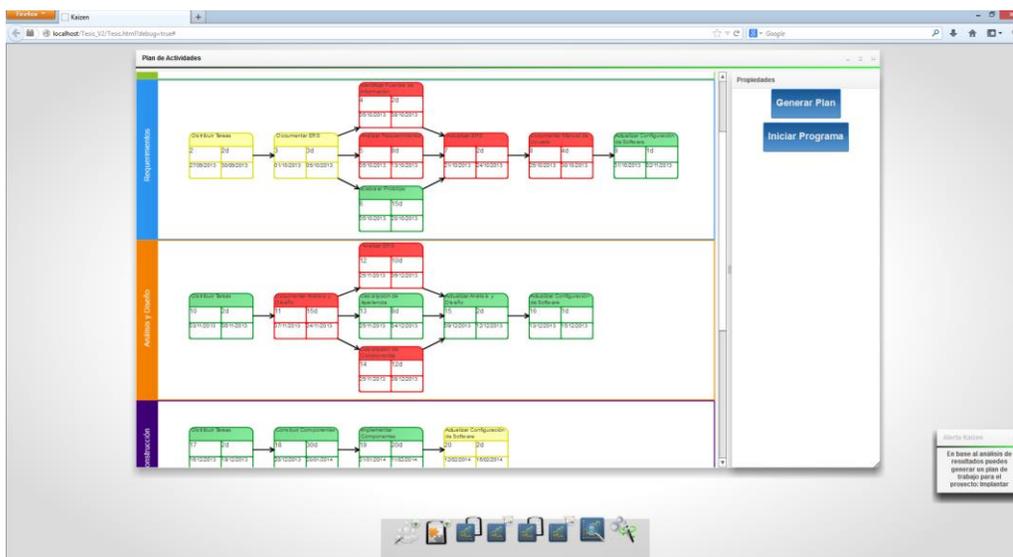


Figura 4.4. Pantalla de planificación de actividades para el nuevo proceso.

Como muestra la Figura 4.4, a partir de la fase de evaluación y de la elección de actividades a mejorar Kaizen genera una planificación del proyecto en base a las sugerencias de MoProSoft, la planeación es presentada en un formato de diagrama PERT, con el fin de mostrar de forma gráfica la planificación de las actividades a los integrantes del equipo de trabajo, incluyendo información importante para su desarrollo como calendarización, uso de recursos, productos de entrada y salida, asignaciones y nivel de atención. El diagrama PERT generado muestra la dependencia entre las actividades, así como la fase del proceso a la que pertenece cada tarea. La información adicional para realizar la actividad que se toma en cuenta es la siguiente: nombre de la actividad, un identificador de la actividad, un tiempo estimado para realizarla, responsable(s), una fecha de inicio y una fecha de final, entradas, descripción y salidas. Kaizen comunica a cada uno de los miembros de trabajo dados de alta las actividades de las cuales son responsables, así como las fechas de inicio y el tiempo para desarrollarlas (véase Figura 4.5). El color de cada nodo, representa el nivel de atención para esa actividad, como se mostró en la sección anterior, al finalizar la evaluación se detectan los puntos fuertes, débiles y en los que se debe de realiza una mayor exploración debido a una inconsistencia en las respuestas de los evaluados; en base a lo anterior los nodos se colorean de la siguiente forma:

- Verde: La actividad representa un punto fuerte de la empresa, lo cual indica que la actividad se realiza dentro de la estructura de la organización por lo que su foco de atención no es urgente, ya que solamente se debe de controlar y asegurar que la actividad se siga realizando en el nuevo ciclo como se viene haciendo comúnmente en la empresa.
- Amarillo: Los resultados de la evaluación indican que si bien la actividad mostraba un porcentaje de cobertura aceptable, las respuestas de los evaluados presentaron una discrepancia alta (lo que se demuestra por una desviación estándar alta), por lo que su foco de atención es medio, ya que se debe de verificar si la actividad realmente se realiza en el nuevo proyecto piloto o las respuestas en la evaluación no fueron congruentes con lo que se realiza en la MPyME.

- Rojo: Representan a los puntos débiles en la empresa, es decir, son las actividades que no están implantadas en el ciclo de desarrollo de la organización, lo que representa un foco de atención alto, ya que es necesario monitorear de manera más rígida su implantación dentro del nuevo ciclo que se pretende establecer en las empresas.

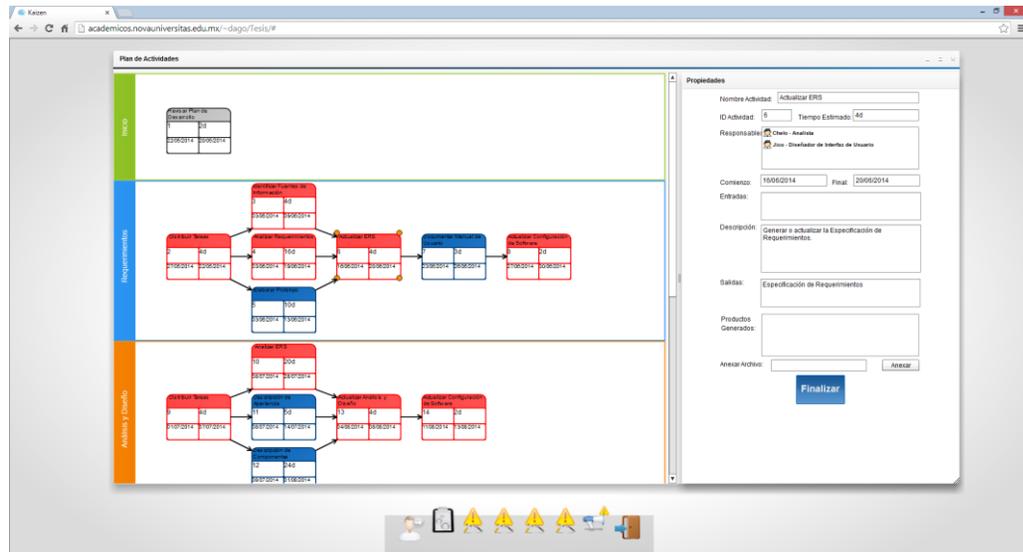


Figura 4.5. Información de actividades del nuevo proceso.

Considerando las deficiencias halladas en los procesos APE y DMS y los nuevos procesos definidos por Kaizen, la Tabla 25 resume de forma general las diferencias entre los procesos actuales y el proceso definido por Kaizen.

Tabla 25. Principales diferencias entre los procesos de las MPyME y los procesos redefinidos por Kaizen.

Proceso MoProSoft	Causa de fracaso	Recomendación Kaizen
APE	El proceso carece de formalidad en los plazos de entrega y ciclo de desarrollo.	Establecer las guías básicas para definir la formalidad de un ciclo de desarrollo.
	Los métodos de estimaciones son casi inexistentes al inicio de los ciclos de desarrollo.	Establecimiento del proceso de estimación y medición de los atributos de los proyectos.
	El proceso para planificar el proyecto es ad-hoc y no hay un método formal establecido.	Establecimiento de procesos formales para la obtención de planes reales
	No existe un mecanismo para monitorizar el progreso de los proyectos.	Implementación de actividades para la monitorización del proyecto.
	Existe traslape y duplicación de actividades	Reducción del re-trabajo.
DMS	El equipo de trabajo no revisa el plan de desarrollo para un proyecto, y no todos tienen conocimiento del mismo.	Adaptación de directrices para crear un entorno de trabajo participativo e incluyente.
	La identificación y consulta de fuentes de información es pobre, por lo que los requisitos de los sistemas no son plenamente obtenidos.	Implementación de las guías básicas para la obtención de una especificación de requisitos formal.
	La descripción de la estructura interna del sistema y su descomposición en subsistemas no coincide plenamente o no se encuentra totalmente alineada a los requisitos obtenidos.	Implementación de actividades relacionadas con análisis de la especificación de los requisitos para generar una descripción de la estructura sistema, subsistemas y componentes, definiendo las interfaces entre ellos.

	No existen personal asignado a roles específicos para el desarrollo, mantenimiento o actualización de manuales de sistema.	Establecer delegación de responsabilidades por medio de roles establecidos por MoProSoft.
--	--	---

Los jefes de proyecto como responsables de la mejora deben de monitorizar y controlar la implantación del nuevo proceso en el desarrollo, para tal actividad, Kaizen ofreció ayuda por medio del control y monitorización automatizado presentado en la sección 3.1.3.3.4. Con esta funcionalidad, los responsables de la mejora fueron alertados sobre el estado de las actividades del proyecto de mejora, término de actividades y productos, información sobre las próximas entregas y desfase en los tiempos de entrega de las actividades, lo anterior con el objetivo de brindar un marco de desarrollo controlado para el equipo de trabajo. Por medio de dicha funcionalidad se implantó la monitorización del proyecto, con lo cual el responsable de mejora puede mantener el control de la realización de las actividades a realizar a lo largo del proyecto.

Antes de comenzar el proyecto y una vez definido el nuevo proceso definido por Kaizen, se realizó una reunión informativa, en donde la alta dirección de las MPyME definió los objetivos a alcanzar por medio de la iniciativa de mejora asistida por Kaizen. Para la definición de objetivos se establecieron tres medidas: calendarización, esfuerzo y tamaño; los cuales representan los porcentajes que se buscan reducir en cada uno de los rubros mencionados. Estos valores representan la media de los datos históricos de las empresas y en etapas posteriores son utilizados para demostrar la efectividad del nuevo proceso implantado por medio de Kaizen. La Tabla 26 resume los objetivos establecidos por las cuatro MPyME.

Tabla 26. Valores objetivo del proyecto piloto para cada MPyME.

Objetivo	Valor objetivo ES1	Valor objetivo ES2	Valor objetivo ES3	Valor objetivo ES4
Desviación de calendario	<14%	<8%	<15%	<12%
Desviación de esfuerzo	<12%	<10%	<11%	<12%
Desviación de tamaño	<13%	<10%	<13%	<12%

En la tabla anterior se puede observar que la ES1 fue la que tuvo mayores expectativas de mejora, lo anterior se debe a que la organización conoce la estructura de MoProSoft por lo que tiene muy buenas posibilidades de poder implantar el modelo mediante la ayuda de Kaizen, ya que sus principales debilidades se encontraban en la gestión y monitorización de las actividades del proyecto. En el caso de la segunda MPyME, los valores objetivo fueron definidos junto con ayuda del grupo experto, ya que, debido al desconocimiento total del modelo la curva de aprendizaje podría ser más lenta y por consecuencia la obtención de los objetivos más difícil. Lo anterior se realizó con todas las empresas involucradas en los programas de mejora, en base a las recomendaciones de [Calvo-Manzano, et al., 2002], donde se define que una de las principales causas de fracaso en un programa de mejora es la creación de expectativas no realistas, es decir, un entusiasmo excesivo por parte de la alta dirección puede también poner en riesgo el programa de mejora, si los resultados no son realistas el esfuerzo de mejora del proceso está abocado al fracaso.

4.1.3. Comparación evaluación final con la línea base inicial

Después de la conclusión del proyecto piloto desarrollado en base al nuevo proceso propuesto, y que fue controlado y monitorizado por Kaizen, la siguiente fase de la experimentación consistió en realizar una evaluación final, el propósito de la evaluación es comprobar la utilidad de

los nuevos procesos definidos para los entornos MPyME involucrados. Como se mostró en la sección 4.1.1, la línea base obtenida fue caracterizada por tener una pobre capacidad de acuerdo a las prácticas sugeridas por MoProSoft en sus procesos APE y DMS.

Al cabo de dos meses de haber culminado los proyectos piloto, se siguió brindando ayuda y asesorías en el uso de los nuevos procesos definidos y se realizó una segunda evaluación a cada una de las MPyME. En la segunda evaluación se obtuvieron nuevas coberturas para los procesos (véase Figura 4.6), y en base a la nueva y antigua cobertura se obtuvo el incremento de la mejora, que es calculado mediante la siguiente fórmula:

$$I = \frac{(C_f - C_i) \times 100}{C_i} \quad (4.1)$$

Donde I representa al incremento obtenido, C_f es la cobertura del proceso después de la implantación del nuevo proceso con la asistencia de Kaizen y C_i es la cobertura inicial obtenida en la línea base.

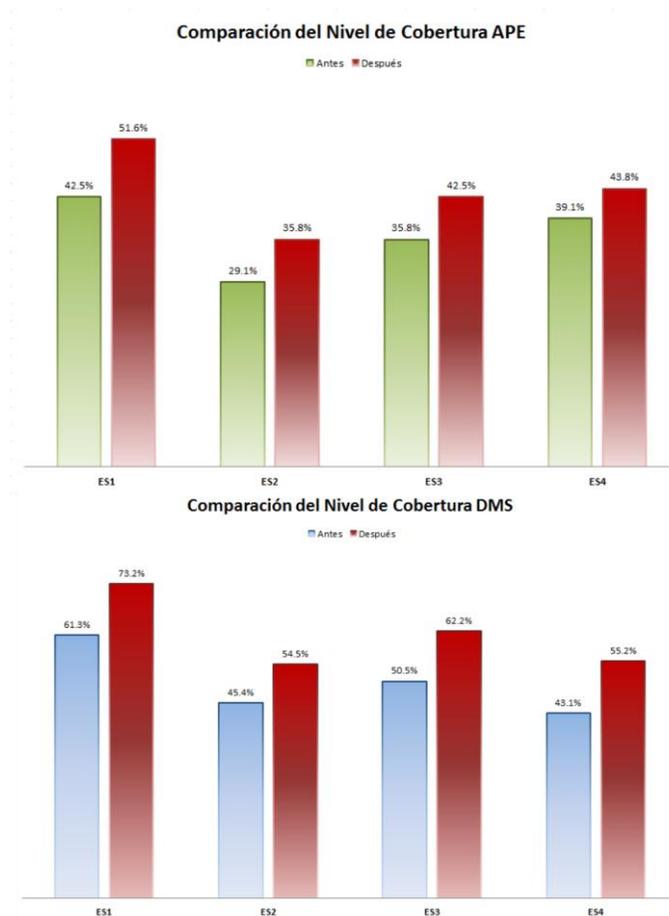


Figura 4.6. Gráfica de comparación de coberturas antes y después de la implantación del nuevo proceso.

Los resultados obtenidos en la evaluación final (mostrados en la Tabla 27) evidencian que todas las empresas incrementaron la cobertura de los procesos APE y DMS alrededor de un 20% en relación a los valores obtenidos en la primera evaluación. Por ejemplo, ES2 presentó una cobertura inicial de 29.1% en el proceso APE antes de la implantación del nuevo proceso basado en MoProSoft con ayuda de Kaizen. Una evaluación posterior a la implantación del nuevo proceso asistido por la herramienta mostró un aumento en el proceso de 23.0% calculado por la ecuación 4.1. Cabe destacar que el mayor incremento promedio en los dos procesos fue en DMS, con un 22.7%, lo

cual demuestra que muchas de las empresas que ya cuentan con un proceso *ad-hoc* tienen menos dificultad para seguir las recomendaciones del modelo MoProSoft, sin embargo, al carecer de experiencia en el manejo de modelo de procesos, les es difícil implantarlo dentro de sus estructuras organizacionales, por lo cual el uso de la herramienta Kaizen es de gran ayuda, ya que proporciona un medio automatizado para presentar y controlar la directrices básicas para implantar las prácticas establecidas por MoProSoft.

Tabla 27. Incremento en los niveles de cobertura por proceso.

MPyME	Coberturas			
	Proceso	Antes	Después	Incremento
ES1	APE	42.5%	51.6%	21.4%
	DMS	61.3%	73.2%	19.4%
ES2	APE	29.1%	35.8%	23.0%
	DMS	45.4%	54.5%	20.0%
ES3	APE	35.8%	42.5%	18.7%
	DMS	50.5%	62.2%	23.2%
ES4	APE	39.1%	43.8%	12.0%
	DMS	43.1%	55.2%	28.1%

Después de obtener toda la información para obtener los resultados experimentales, ha quedado en claro que existen factores importantes para alcanzar las metas en un proyecto de mejora de procesos, que si bien no tiene un fin altamente técnico, si inciden de manera directa e importante en los resultados de la mejora, estos factores son el compromiso, la iniciativa y la responsabilidad de las cuatro MPyME. Sin los componentes antes mencionados, una iniciativa de mejora no tiene ningún tipo de sentido, ya que la organización es la única que sabe en qué estado se encuentra, a dónde quiere llegar y lo importante que es para ella mejorar.

La experimentación también permitió mostrarles a pequeños grupos de trabajo cómo realizar iniciativas de mejora dentro de las empresas con ayuda de Kaizen, el uso de la herramienta es un valioso soporte para este tipo de proyectos, ya que brinda una serie de directrices para llevar a cabo las fases genéricas que componen a una iniciativa de mejora auxiliando a los equipos de trabajo sin experiencia a controlar las mejoras mediante el uso de la herramienta. En base a lo observado en esta experimentación se adquieren nuevas y valiosas experiencias de cómo realizar las cosas en entornos pequeños de trabajo, algo de gran utilidad debido a la estructura general de la industria de software en México; y también el observar el comportamiento de una MPyME durante el proceso de mejora.

De manera adicional a la evaluación sobre el nivel de cobertura actual, las evaluaciones sobre los procesos APE y DMS nos permitieron identificar actividades concretas de la gestión y desarrollo de los proyectos que no se encontraban funcionando correctamente y que fueron mejoradas con la implantación de las actividades propuestas por MoProSoft mediante el uso de Kaizen (véase Tabla 24). En base a lo anterior, se decidió juntos con las organizaciones involucradas en la experimentación definir valores objetivo a través de medidas como calendarización, esfuerzo y tamaño (véase Tabla 25), para validar el resultado de la implantación del nuevo modelo propuesto por Kaizen. En este sentido, se tomaron como base los datos históricos de las empresas en el desarrollo de proyecto anteriores y de características similares al propuesto como proyecto piloto, con el fin de establecer una segunda base comparativa para determinar la eficiencia de los procesos

creados por Kaizen. Los datos recogidos corresponden a las estimaciones y medidas de calendarización, tamaño y esfuerzo, y se calcularon medidas derivadas para realizar la comparación con los resultados obtenidos por medio del proyecto piloto.

4.1.4. Verificación de los objetivos particulares

La verificación de los objetivos de los proyectos piloto se realizó mediante la comparación de los resultados obtenidos con los objetivos establecidos en la Tabla 25.

Para la gestión y desarrollo efectivo (en Nivel 1), el nuevo procesos incorporó actividades para la planificación del proyecto que establecieron procesos de estimación mejor estructurados, incluir el factor formalidad en los ciclos de entrega y desarrollo, actividades de monitorización y control del plan de proyecto por medio de Kaizen. Cabe mencionar que tales actividades fueron tomadas de las propuestas del modelo MoProSoft con el fin de ofrecer a los responsables del desarrollo nuevas formas para detectar problemas a tiempo y poder tomar las medidas pertinentes (preventivas y/o correctivas) para afectar en lo mínimo los plazos en el desarrollo del proyecto.

El uso de Kaizen propició un entorno de trabajo más participativo y colaborativo que es apoyado por la estructura y características de la herramienta, que permite mantenerse comunicados y enterados de las actividades realizadas por los miembros del equipo de trabajo. Para apoyar a las organizaciones en la adopción del nuevo proceso, en primera instancia se programaron reuniones personales en lapsos quincenales con los jefes de proyecto para evaluar los calendarios y objetivos; una vez establecidos y familiarizados con el uso de Kaizen las reuniones fueron realizadas de manera remota y en lapsos de tiempo más amplios.

La Tabla 27 muestra los resultados estimados (objetivos) y reales del proyecto, asociados a las tres medidas mencionadas: calendarización, esfuerzo y tamaño. Esta tabla muestra que tres de las cuatro empresas lograron terminar el proyecto dentro del calendario planificado, el esfuerzo en estas tres empresas es negativo puesto que se terminó antes que el tiempo estimado. Aunque sólo una de las empresas no pudo alcanzar totalmente los objetivos (ES2), la tabla muestra que estuvo muy cerca de conseguirlos, y en el caso particular al no contar con ningún tipo de experiencia en materia de mejora de procesos, se puede decir que los valores alcanzados son positivos ya que sin ningún tipo de experiencia y siguiendo el plan establecido por Kaizen logro alcanzar valores bastante aprobables en base a su situación antes de la mejora.

La Tabla 28 sirve como referencia comparativa para analizar los beneficios de la implantación del nuevo proceso basado en MoProSoft generado por Kaizen mediante la comparación del valor objetivo establecido al inicio por cada MPyME, con el valor obtenido con la aplicación del nuevo proceso mediante Kaizen (VK). La desviación permite analizar la diferencia entre el valor obtenido con el valor objetivo. La desviación es calculada en base al valor obtenido y al umbral establecido en base al valor objetivo planteado de la forma:

$$D = \frac{100 \times VK}{Umbral} - 100 \quad (4.2)$$

En el caso de ES2, el valor de la desviación para el esfuerzo es de -49.6% y se considera que el valor objetivo debe ser menor al 12%, esto quiere decir que el umbral establecido es del 11.9%, por lo que el resultado es obtenido de la forma:

$$D = \frac{100 \times 6}{11.9} - 100 = -49.6\%$$

Lo que quiere decir que existe una diferencia del -49.6% entre el valor objetivo y el valor obtenido.

Tabla 28. Análisis de resultados para el cumplimiento de objetivos por MPyME.

	ES1			ES2			ES3			ES3		
	Valor objetivo	VK	Desviación	Valor objetivo	VK	Desviación	Valor objetivo	VK	Desviación	Valor objetivo	VK	Desviación
Desviación en calendario	<14%	10%	-28.1%	<8%	8%	1.3%	<15%	12%	-19.5%	<12%	10.5%	-11.8%
Desviación en esfuerzo	<12%	6%	-49.6%	<10%	10%	1.0%	<11%	9%	-17.4%	<12%	11%	-7.6%
Desviación en tamaño	<13%	11%	-14.7%	<10%	9%	-9.1%	<13%	10%	-22.5%	<12%	10%	-16.0%

Mediante el análisis de estos resultados se puede concluir que la aplicación del nuevo proceso basado en las prácticas de MoProSoft y el uso de Kaizen para su implantación dentro de las MPyME para la definición de un nuevo proceso, han sido propicias para alcanzar los objetivos de los proyectos y de la organización.

5. Conclusiones

Debido a la demanda de mejor calidad en los productos, y ante la necesidad de incursionar en mercados internacionales, no sin antes satisfacer por completo las necesidades de un mercado local, la industria de software en México se encuentra en una etapa de evolución y adquisición del conocimiento necesario para poder cumplir dichas metas. El cambio de paradigma en las empresas se ha ido presentando gradualmente, un cambio que representa el dejar de enfocarse totalmente en problemas y soluciones basadas en tecnología para dirigir los esfuerzos a la gestión de los procesos de software utilizados para desarrollar y mantener aplicaciones computarizadas. Motivadas por adquirir estándares de calidad que los ayuden a producir productos de mayor calidad que a la postre se pueda reflejar en una ventaja competitiva, las organizaciones mexicanas han incursionado en programas de Mejora del Proceso de Software (SPI) implantando el modelo de procesos nacional MoProSoft para obtener un proceso de desarrollo capaz y maduro. Sin embargo, debido a la estructura de la industria mexicana (MPyME y emergente) la experiencia en iniciativas de SPI en algunos casos es limitada y en los más nula. La inexperiencia tiene como resultado que las organizaciones se muestren cautas y escépticas de las oportunidades y ventajas que puede traer una iniciativa de este tipo, aunado a lo anterior el hecho de que gran parte de los ingenieros y desarrolladores de software que conforman éstas empresas carecen de los conocimientos necesarios para llevar a cabo un programa de mejora de procesos; lo cual genera apatía para el impulso de programas de mejora en las organizaciones.

Este trabajo de investigación, se enfocó en el desarrollo de una herramienta para la implantación de iniciativas de mejora en MPyME mexicanas, brindando un marco de trabajo que proporcione las fases y actividades dentro de un solo entorno de ejecución para que las organizaciones que pretendan implantar un programa de mejora basado en MoProSoft cuenten con las guías y directrices necesarias para llevar por buen camino este tipo de proyectos. La plataforma Kaizen (mejora continua en japonés) se basa en las etapas genéricas de un modelo de mejora: compromiso con la mejora, evaluación actual del proceso de software de la empresa, infraestructura y planes de mejora e implantación de dichos planes. El aspecto más importante a corroborar, era que con el apoyo de Kaizen las pequeñas organizaciones eran capaces de implantar ciclos de mejora, cumpliendo con todas las fases necesarias para este tipo de proyectos, y de esta forma conseguir implantar el nuevo proceso y gradualmente convertirlo en parte fundamental de la cultura de trabajo dentro de la estructura de la organización.

La experimentación de la adopción de Kaizen se llevó a cabo con empresas de distintas características (Capítulo 4 del presente trabajo), en donde los resultados arrojan, que si bien Kaizen en primera instancia no sustituye o elimina la falta de conocimiento en aspectos específicos de programas de mejora o en la estructura de MoProSoft; si proporciona una guía ideal para conducir

este tipo de mejora de procesos en aquellas organizaciones que tienen un escaso o nulo conocimiento sobre los pasos a seguir para conseguir la implantación de un modelo de procesos, lo cual por medio de un número de iteraciones considerable puede servir para generar un arraigo de las actividades de MoProSoft en el proceso de desarrollo de la empresa. Asociado a lo anterior, el hecho de proporcionar muchas de las tareas y fases de un ciclo de mejora de manera automatizada como la generación de resultados de la evaluación, generación automática de los planes de mejora, asignación de actividades a los miembros del equipo, generación de un calendario de actividades y un módulo de control y monitoreo de la realización de las actividades especificadas en el calendario; hacen de Kaizen una opción interesante, viable y recomendable a las MPyME interesadas en la adopción de prácticas de MoProSoft dentro de su infraestructura. Uno de los aspectos a tener en cuenta es que Kaizen ofrece todas sus funcionalidades de forma libre, con el fin de que cualquier empresa pueda acceder a todas las funcionalidades con las que cuenta la plataforma, tomando en cuenta que dicho aspecto fue controlado en la experimentación de este trabajo debido a la necesidad de recopilación de resultados. Algo a tomar en cuenta y claro es que Kaizen es una herramienta de apoyo, que si bien ofrece muchas ventajas respecto a una conducción “manual” de un programa de mejora, las empresas deben preocuparse por la formación y actualización de sus grupos de trabajo para obtener mejores resultados.

Dentro de los hallazgos importantes detectados durante el desarrollo de este trabajo fue el encontrar un punto de convergencia entre diferentes campos de las ciencias computacionales y la Ingeniería de Software. Un claro ejemplo son las técnicas de algoritmia ocupadas en la etapas de desarrollo de Kaizen referentes a la evaluación de procesos y generación de un calendario de actividades, con las cuales se logró dar un funcionamiento idóneo a la automatización de actividades que de manera manual pueden tornarse complicadas y confusas en un ciclo de mejora. La teoría de grafos fue de gran ayuda en la parte de evaluación por diagrama de procesos, ya que por medio de la caracterización de estos diagramas por medio de un grafo dirigido se consiguió la comparación de dos diagramas de procesos distintos, primeramente con la consecución de una representación estándar para todos los diagramas, seguido de una comparación basada en recorridos y comparación de vértices (actividades) en base a la dirección de los flujos de secuencia (aristas). En el caso de la calendarización de actividades propuesta en los planes de mejora generados en base a la fase de evaluación, la cual cumple con las características de un Problema de Calendarización de Proyectos con Recursos Limitados (RCPSP), se utilizó la técnica de Algoritmos Genéticos (AG) la cual cumplió con los requisitos planteados para dicha funcionalidad, las características propias de la problemática planteada, como el hecho de que la secuencia de actividades y la delegación de las mismas ya se encuentra estipulada por MoProSoft, hacen que los AG sean una muy buena opción para la calendarización de actividades SPI basadas en un modelo de referencia, en este caso MoProSoft o cualquier otro de estructura similar.

En materia del estado y evolución de las organizaciones participantes en el caso de estudio llevado a cabo, los descubrimientos corroboraron los datos e hipótesis que plantean diferentes estudios acerca del estado de la industria mexicana de software y en general con las deficiencias más latentes en un proceso de desarrollo. El principal hallazgo es el hecho de que las organizaciones siguen teniendo una noción clara del proceso general que se tiene que seguir para el desarrollo de software, lo que se demuestra mediante los resultados obtenidos en la evaluación del proceso de Desarrollo y Mantenimiento de Software de MoProSoft, sin embargo, carecen de prácticas específicas y fases de desarrollo que hacen de que sus procesos presenten inconsistencias preocupantes durante el lapso de ejecución. Aún más preocupante es el hecho de que las MPyME participantes, tienden a tener esa falsa ideología de que la gestión de los proyectos es una actividad

que no incide directamente en la calidad del producto, es por eso que dentro de sus procesos desprecian actividades como la definición de ciclos formales, estimaciones en materia de tamaño, tiempo y costo, calendarización de actividades, documentación de las fases más importantes y lecciones aprendidas para cada ciclo, lo anterior se demuestra en la baja cobertura presentada en las evaluaciones del proceso de Administración de Proyectos Específicos. Lo anterior más que una actitud de desidia por parte de la organización, refleja un desconocimiento de las ventajas y oportunidades que representa una buena gestión. Otro punto a tomar en cuenta es que muchas veces los integrantes del equipo de desarrollo desconocen el tipo de técnicas a utilizar en tales prácticas, carecen de experiencia en el uso de estos métodos y no saben en qué fase particular del proyecto deben de llevar a cabo tales tareas. Si bien la mejor solución es el hecho de brindar a los actuales y próximos recursos humanos cursos de capacitación especializados tanto en su formación universitaria como en cursos de certificación o actualización dentro de sus entornos laborales, dichas soluciones representan obtener resultados a mediano y largo plazo; de igual forma pueden representar un gasto considerable para las organizaciones. De esta forma, Kaizen brindó apoyo a estas organizaciones, para encaminar y controlar las iniciativas de mejora, en un corto y mediano plazo. El brindar las directrices necesarias para implantar una mejora fue de gran ayuda para las organizaciones, ya que, como se demostró en la sección de experimentación, en un primer ciclo las empresas llevaron a cabo muchas de las actividades que se despreciaban hasta antes del uso de Kaizen, lo cual tuvo como resultado una disminución en los tiempos y esfuerzo de desarrollo, obteniendo un producto que cumpla con las expectativas marcadas al inicio del ciclo de desarrollo. Lo anterior establece que una herramienta con las características de Kaizen es una opción recomendable para aquellas pequeñas organizaciones con poca o nula experiencia en iniciativas de mejora y que buscan la implantación de un modelo de procesos dentro de su proceso actual, debido al marco de trabajo controlado y colaborativo que ofrece.

La necesidad de competir no solo en mercados locales sino mundiales, hace que las pequeñas organizaciones busquen la implantación de modelos de procesos internacionales enfocados a las características de micro y pequeñas empresas. Por lo anterior, se considera como trabajo futuro desarrollar nuevas versiones de Kaizen basadas en modelos internacionales como ISO/IEC 29110 y CMMI en su versión para pequeños entornos, dicho cambio representa la actualización de cuestionarios, diagramas de procesos base y forma de evaluación, manteniendo la estructura y fases presentadas en este trabajo. De igual forma se pretende lanzar un sitio Web, para que todas las organizaciones interesadas en utilizar Kaizen puedan registrarse y posteriormente utilizar de forma libre y gratuita todas las funcionalidades ofrecidas por la herramienta, con el objetivo de fomentar la evolución de los procesos software hacia un estado capaz y maduro, y que tenga como consecuencia la mejora en la calidad de los productos de la empresa, lo que paulatinamente pueda representar el fortalecimiento de la industria de software en México.

6. Anexo A.- Acrónimos

IEEE	Instituto de Ingenieros en Electricidad y Electrónica
DRA	Desarrollo Rápido de Aplicaciones
IS	Ingeniería de Software
XP	Programación Extrema
SPI	Mejora al Proceso Software
CMMI	Modelo de Madurez y Capacidad Integrado
MoProSoft	Modelo de Procesos para la Industria de Software
SPA	Modelo de Evaluación del Proceso Software
Pymes	Pequeñas y medianas empresas
MPyME	Micro, Pequeñas y Medianas Empresas
SEI	Instituto de Ingeniería de Software
ISO	Organización Internacional para la Estandarización
TI	Tecnologías de Información
SE	Secretaría de Economía
PROSOFT	Programa para el Desarrollo de la Industria del Software
SNITI	Sistema Nacional de Indicadores de la Industria de Tecnologías de Información
RIA	Aplicaciones Ricas en Internet
PND	Plan Nacional de Desarrollo
EvalProSoft	Método de Evaluación de procesos para la Industria Software

7. Anexo B.- Cuestionario para la evaluación del modelo MoProSoft

El siguiente cuestionario fue elaborado en base a las prácticas propuestas por los procesos definidos en el modelo MoProSoft. Cada cuestionario se divide en base a las fases propuestas por las áreas de proceso.

7.1. Gestión de Negocio

El propósito de gestión de negocio es establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua. Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.

I. Planificación Estratégica	
Pregunta	Nivel
1. ¿Articulan, documentan o actualizan la Misión, Visión y Valores?	1
2. ¿Identifican las oportunidades y amenazas con base en las necesidades de los clientes, información sobre competidores, tendencias tecnológicas y otros factores?	1
3. ¿Identifican las fortalezas y debilidades con base en: análisis financieros, identificación de recursos, entre otras?	1
4. ¿Definen o actualizan los Objetivos, y las Estrategias que especifiquen el medio para alcanzar estos objetivos?	1
5. ¿Definen o actualizan los Indicadores que permitan medir el logro de Objetivos?	3
6. ¿Determinan el valor actual de los indicadores y estableces Metas Cuantitativas deseadas?	4
7. ¿Definen o actualizan procesos y proyectos, considerando las Propuestas de Mejora?	1
8. ¿Identifican los procesos requeridos?	1
9. ¿Definen una Cartera de Proyectos?	1
10. ¿Definen o actualizan la Estructura de la Organización para la implantación de un plan, considerando las Propuestas de Mejora?	1
11. ¿Definen una Estrategia de Recursos?	1
12. ¿Identifican y distribuyen los recursos necesarios para la implantación del plan?	1
13. ¿Identifican los elementos de la Base de Conocimiento necesarios para el almacenamiento y consulta de la información generada en la organización?	1
14. ¿Calculan el presupuesto requerido (gastos e ingresos esperados) para lograr la implantación del Plan Estratégico, y determinar el periodo para qué aplicará?	1

15. ¿Definen mecanismos de comunicación con el cliente para su atención y documentarlos en el Plan de Comunicación con el Cliente?	1
16. ¿Integran y documentan el Plan Estratégico?	1
17. ¿Verifican el Plan Estratégico?	2
18. ¿Corrigen los defectos encontrados en el Plan Estratégico con base en el Reporte de Verificación para obtener la aprobación de las correcciones?	2
19. ¿Validan el Plan Estratégico?	2
20. ¿Corrigen los defectos encontrados en el Plan Estratégico con base en el Reporte de Validación para obtener la aprobación de las correcciones?	2
21. ¿Elaboran un Plan de Adquisiciones y Capacitación para el proceso Gestión de Negocio?	1
II. Preparación para la Realización	
22. ¿Procuran un ambiente adecuado para la implantación del Plan Estratégico?	1
23. ¿Identifican las líneas y medios de comunicación, que permitan la divulgación efectiva del Plan Estratégico?	2
24. ¿Identifican cómo efectuar los cambios necesarios en la estructura de la organización?	2
25. ¿Definen cómo establecer y distribuir los recursos necesarios y adecuados?	2
26. En base a los puntos anteriores, ¿definen y ejecutan un Plan de Comunicación e Implantación del Plan Estratégico?	2
27. ¿Validan el Plan de Comunicación e Implantación?	2
28. ¿Corrigen los defectos encontrados en el Plan de Comunicación e Implantación con base en el Reporte de Validación para obtener la aprobación de correcciones?	2
III. Valoración y Mejora Continua	
29. ¿Realizan un análisis de los Reportes Cuantitativos y Cualitativos de procesos y proyectos para comparar resultados con las metas planteadas?	3
30. ¿Realizan un análisis del Reporte de Acciones Correctivas o Preventivas Relacionadas con Clientes, en referencia a la satisfacción de las necesidades del cliente?	3
31. ¿Analizan las Propuestas Tecnológicas, para adoptar alguna(s) en beneficio de las actividades de la organización?	3
32. ¿Analizan los Reportes Financieros para determinar la viabilidad de proyectos y ajustes a los mismos, así como determinar ajustes requeridos al presupuesto calculado?	3
33. ¿Analizan los Factores Externos, para hacer algún reajuste correspondiente?	3
34. ¿Evalúan el desempeño alcanzado con la estrategia actual, considerando la evaluación del cumplimiento de los Objetivos?	3
35. ¿Generan un Reporte de Valoración en donde se registran los detalles del análisis previo?	3
36. ¿Generan una Propuesta de Mejora al Plan Estratégico actual?	4
37. ¿Validan la Propuesta de Mejora?	4
38. ¿Corrigen los defectos encontrados en la Propuesta de Mejoras con base en el Reporte de Validación para obtener la aprobación de correcciones?	4
39. ¿Generan un Reporte de Mediciones y Sugerencias de Mejora del proceso, de acuerdo al Plan de Mediciones de Procesos?	3
40. ¿Identifican las Lecciones Aprendidas y las integran a la Base de Conocimiento?	3

7.2. Gestión de Procesos

El propósito de Gestión de Procesos es establecer los procesos de la organización, en función de los Procesos Requeridos identificados en el Plan Estratégico. Así, como definir, planificar, e implantar las actividades de mejora en los mismos.

Pregunta	Nivel
I. Planificación	
1. ¿Revisan los modelos de procesos de referencia para definir y actualizar los elementos y la estructura que conformarán los Procesos Requeridos en el Plan Estratégico?	1
2. ¿Establecen o actualizan la Definición de Elementos de Procesos?	1
3. ¿Establecen un Calendario para mantener y mejorar procesos?	1
4. ¿Establecen o actualizan un Plan de Adquisiciones y Capacitación?	1
5. ¿Consideran la Asignación de Recursos?	1
6. ¿Identifican las necesidades del personal capacitado?	1
7. ¿Identifican las necesidades de infraestructura y herramientas?	1
8. ¿Identifican las necesidades de capacitación de la organización, con respecto a los procesos?	1
9. ¿Incluyen una lista de proveedores en el Plan de Adquisiciones y Capacitación?	1
10. ¿Determinan que tipo de evaluaciones (interna o externa) se realizarán en la organización?	
11. ¿Se determina para cada evaluación el objetivo, el alcance, los métodos y criterios de evaluación, el calendario y los recursos necesarios?	1
12. ¿En base a lo anterior se establece un Plan de Evaluación?	1
13. ¿Establecen un Plan de Mediciones de Proceso?	3
14. ¿El Plan de Mediciones de Procesos, especifica el tipo de mediciones a realizar en los procesos?	3
15. ¿El Plan de Mediciones de Procesos, determina la periodicidad de aplicación de las mediciones?	3
16. ¿Se asigna a cada medición un responsable?	3
17. ¿Establecen un Plan de Manejo de Riesgos para la Gestión de Procesos?	1
18. ¿Identifican y evalúan los riesgos en cada proceso?	1
19. ¿Definen un plan de contingencia de riesgos?	1
20. ¿Definen un plan de contingencia?	1
21. ¿Integran el Plan de Proceso?	1
22. ¿Verifican el Plan de Procesos?	2
23. ¿Corrigen los defectos encontrados en el Plan de Procesos con base en el Reporte de Verificación para obtener la aprobación de las correcciones?	2
24. ¿Validan el Plan de Procesos?	2
25. ¿Corrigen los defectos encontrados en el Plan de Procesos con base en el Reporte de Validación para obtener la aprobación de las correcciones?	2
II. Preparación de la Implantación	
26. ¿Gestionan el Plan de Adquisiciones y Capacitación identificado en el Plan de Procesos?	1
27. ¿Asignan y notifican a los Responsables de Procesos?	1
28. ¿Elaboran la Documentación del Proceso de acuerdo al Plan de Procesos?	1

29. ¿Verificar la Documentación de Procesos?	2
30. ¿Corrigen los defectos encontrados en la Documentación de Procesos con base en el Reporte de Verificación para obtener la aprobación de las correcciones?	2
31. ¿Capacitan a la organización en los procesos?	1
32. ¿Implantan los procesos en un proyecto piloto?	
III. Evaluación y Control	
33. ¿Brindan seguimiento a las actividades de implantación de proceso del Calendario establecido en el Plan de Procesos?	2
34. ¿Generan un Reporte de Mediciones y Sugerencias de Mejora del proceso, de acuerdo al Plan de Mediciones del Proceso?	3
35. ¿Generan el Reporte Cuantitativo y Cualitativo a partir de los Reportes de Mediciones y Sugerencias de Mejora recolectados, el cual se entrega al Responsable de Gestión de Negocio?	3
36. ¿Realizan las evaluaciones establecidas en el Plan de Evaluación?	3
37. ¿En las evaluaciones, identifican y documentan los hallazgos detectados y establecen un Plan de Acciones para resolver lo encontrado?	3
38. ¿Identifican y documentan las oportunidades de mejora de los procesos?	3
39. ¿Elaboran un Reporte de Evaluación?	3
40. ¿Verifican el Plan de Acciones?	3
41. ¿Corrigen los defectos encontrados en el Plan de Acciones con base en el Reporte de Verificación para obtener la aprobación de las correcciones?	3
42. ¿Validan el Plan de Acciones?	3
43. ¿Corrigen los defectos encontrados en el Plan de Acciones con base en el Reporte de Validación para obtener la aprobación de las correcciones?	3
44. ¿Generan un Plan de Mejora a partir del análisis de las sugerencias de mejora y de las oportunidades de mejora detectadas durante la evaluación?	3
45. ¿Verifican el Plan de Mejora?	3
46. ¿Corrigen los defectos encontrados en el Plan de Mejora en el Reporte de Verificación para obtener la aprobación en las correcciones?	3
47. ¿Validan el Plan de Mejora?	3
48. ¿Corrigen los defectos encontrados en el Reporte de Validación para obtener la aprobación de las correcciones?	3
49. ¿Dan seguimiento al Plan de Adquisiciones y al Plan de Mejora?	3
50. ¿Supervisan el control de riesgos de acuerdo al Plan de Manejo de Riesgos de procesos?	2
51. ¿Identifican lecciones aprendidas de procesos para integrarlas a la Base de Conocimiento?	3

7.3. Gestión de Proyectos

El propósito de la Gestión de Proyectos es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización. La Gestión de Proyectos se ocupa de los proyectos externos, internos y de las oportunidades de proyectos de la organización.

Pregunta	Nivel
I. Planificación	
1. ¿Realizan un análisis para generar Alternativas de Realización de Proyectos Internos?	1
2. ¿Seleccionan una alternativa para los proyectos internos?	1
3. ¿Generan un Plan de Gestión de Proyectos en función de la Cartera de Proyectos del Plan Estratégico?	1
4. ¿Elaboran un Plan de Ventas, incluyendo acciones y programa de trabajo para generar y cerrar oportunidades de proyecto?	1
5. ¿Elaboran un Plan de Proyectos para gestionar los proyectos externos e internos, considerando las Alternativas de Realización de Proyectos Internos?	1
6. ¿Elaboran un Plan de Adquisiciones y Capacitación, incluyendo los recursos y la capacitación requerida por los proyectos?	1
7. ¿Establecen Mecanismos de Comunicación con los Clientes de acuerdo al Plan de Comunicación con el Cliente?	2
8. ¿Validan el Plan de Gestión de Proyectos, Plan de Adquisiciones y Capacitación y los Mecanismos de Comunicación con los Clientes?	2
9. ¿Corrigen los defectos encontrados en el Plan de Gestión de Proyectos, Plan de Adquisiciones y Capacitación y los Mecanismos de Comunicación con los Clientes en base al Reporte de Validación para obtener la aprobación de correcciones?	2
II. Realización	
10. ¿Identifican prospectos y necesidades de posibles clientes?	1
11. ¿Estiman tiempos y costos conjuntamente con los representantes del grupo de desarrollo y mantenimiento de Software?	1
12. ¿Generan y presentan propuestas para oportunidades identificadas?	1
13. ¿Elaboran Contrato(s)?	1
14. ¿Generan un Registro de Proyecto para los proyectos contratados o internos?	1
15. ¿Generan una Descripción del Proyecto, y si el proyecto es interno consideran las Alternativas de Realización de Proyectos Internos?	1
16. ¿Generan Metas Cuantitativas para el Proyecto?	4
17. ¿Asignan un Responsable para la Administración de Proyectos Específico?	1
18. ¿Reciben y aprueban un Plan de Proyecto?	1
19. ¿Recolectan los Reportes de Seguimiento?	2
20. ¿Cierran los proyectos internos o contratados, al recibir un Documento de Aceptación?	1
21. ¿Implantan Mecanismos de Comunicación con los Clientes para recabar los Comentarios y Quejas del Cliente?	2
III. Evaluación y Control	
22. ¿Analizan el cumplimiento del Plan de Ventas, para generar y dar seguimiento a las Acciones Correctivas o Preventivas?	2
23. ¿Analizan los Reportes de Seguimiento de los proyectos y Comentarios y Quejas del Cliente con respecto a los proyectos, para generar y dar seguimiento a las Acciones Correctivas o Preventivas?	2
24. ¿Analizan los Comentario y Quejas del Cliente con respecto a los mecanismos de comunicación, para generar y dar seguimiento a las Acciones Correctivas o Preventivas?	2

25. ¿Generan un Reporte Cuantitativo y Cualitativo con base en los reportes de seguimiento de los proyectos y al cumplimiento del Plan de Ventas?	3
26. ¿Generan un Reporte de Acciones Correctivas o Preventivas Relacionadas con Clientes?	2
27. ¿Generan un Reporte de Mediciones y Sugerencias de Mejora de este proceso, de acuerdo al Plan de Mediciones de Procesos?	3
28. ¿Identifican las Lecciones Aprendidas para integrarlas a la Base de Conocimiento?	3

7.4. Gestión de Recursos

El propósito de Gestión de Recursos es conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la Base de Conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del Plan Estratégico de la organización.

Pregunta	Nivel
I. Planificación de Recursos	
1. ¿Generan o actualizan un Plan de Adquisiciones y Capacitación necesario para un proceso actual?	1
2. ¿Generan un Plan Operativo de Recursos Humanos y Ambiente de Trabajo, a partir del Plan Estratégico y los Planes de Adquisiciones y Capacitación?	1
3. ¿Establecen elementos a considerar en la selección, asignación, aceptación, capacitación, evaluación y desempeño de los recursos humanos?	1
4. ¿Verifican el Plan Operativo de Recursos Humanos y Ambiente de Trabajo?	2
5. ¿Corrigen los defectos encontrados en el Plan Operativo de Recursos Humanos y Ambiente de Trabajo en base al Reporte de Verificación para obtener la aprobación de las correcciones?	2
6. ¿Generan un Plan Operativo de Bienes, Servicios e Infraestructura, a partir del Plan Estratégico y los Planes de Adquisiciones y Capacitación?	1
7. ¿Establecen elementos para garantizar la adquisición y asignación de bienes, servicios e infraestructura, necesarios para realizar las actividades de la organización?	1
8. ¿Establecer elementos para evaluar y calificar el servicio de proveedores?	1
9. ¿Verifican el Plan Operativo de Bienes, Servicios e Infraestructura?	2
10. ¿Corregir los defectos encontrados en el Plan Operativo de Bienes, Servicios e Infraestructura con base en el Reporte de Verificación para obtener la aprobación de correcciones?	2
11. ¿Generan un Plan Operativo de Conocimiento de la Organización, a partir del Plan Estratégico?	1
12. ¿Establecen elementos para la definición, operación y mantenimiento, del conocimiento generado en la organización?	1
13. ¿Verifican el Plan Operativo de Conocimiento de la Organización?	2
14. ¿Corrigen los defectos encontrados en el Plan Operativo de Conocimiento de la Organización con base en el Reporte de Verificación para obtener la aprobación de correcciones?	2
II. Seguimiento y Control	

15. ¿Dan seguimiento a la ejecución del Plan Operativo de Recursos Humanos y Ambiente de Trabajo en función del Reporte de Recursos Humanos Disponibles, Capacitación y Ambiente de Trabajo?	2
16. ¿Determinan si la selección, asignación, aceptación, capacitación, evaluación y desempeño de los recursos humanos es adecuada?	2
17. ¿Determinan si el ambiente de trabajo es el adecuado?	2
18. ¿Dan seguimiento a la ejecución del Plan Operativo de Bienes, Servicios e Infraestructura en función del Reporte de Bienes, Servicios e Infraestructura?	2
19. ¿Determinan si la adquisición y asignación de los bienes y servicios es adecuada?	2
20. ¿Determinan si el servicio de los proveedores es adecuado y oportuno?	2
21. ¿Dan seguimiento a la ejecución del Plan Operativo de Conocimiento de la Organización en función del Reporte de Estado de la Base de Conocimiento?	2
22. ¿Determinan si el conocimiento de la organización se almacena y actualice correctamente?	2
23. ¿Determinan si el conocimiento de la organización está disponible para su consulta?	2
24. ¿Analizan periódicamente el uso de recursos y el ambiente de trabajo en la organización para compararlos con el Plan de Comunicación e Implantación?	2
25. ¿Generan un Reporte de Mediciones y Sugerencias de Mejora para el proceso, de acuerdo al Plan de Mediciones de Procesos?	3
26. ¿Identifican las Lecciones Aprendidas e integrarlas a la Base de Conocimiento?	3
III. Investigación de Tendencias Tecnológicas	
27. ¿Realizan un análisis prospectivo y de viabilidad de las tendencias tecnológicas?	3
28. ¿Determinan el beneficio y el impacto de las tendencias tecnológicas al Plan Estratégico?	3
29. ¿Generan una Propuesta Tecnológica?	3

7.5. Administración de Proyectos Específicos

El propósito de la Administración de Proyectos Específicos es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

Pregunta	Nivel
I. Planificación	
1. ¿Revisan con el Responsable de Gestión de Proyectos la Descripción del Proyecto?	1
2. Con base en la Descripción del Proyecto, ¿Definen el Proceso Específico del proyecto a partir del proceso Desarrollo y Mantenimiento de Software de la organización o a partir del acuerdo establecido con el Cliente, considerando el alcance, magnitud y complejidad del proyecto?	3
3. ¿Definen conjuntamente con el Cliente un Protocolo de Entrega de cada uno de los entregables especificados en la Descripción del Proyecto?	1
4. ¿Identifican el número de ciclos y las actividades específicas que deben llevarse a cabo para producir los entregables y sus componentes identificados en la Descripción del Proyecto?	1

5. ¿Identifican las actividades específicas que deben llevarse a cabo para cumplir con los objetivos del proyecto, definiendo las actividades para llevar a cabo revisiones periódicas al producto o servicio que se está creciendo y para efectuar revisiones entre colegas?	2
6. ¿Identifican las actividades para llevar a cabo el Protocolo de Entrega?	1
7. ¿En base a las actividades planteadas, documentan el resultado como Ciclo y Actividades?	1
8. ¿Identifican y documentan la relación y dependencia de cada una de las actividades?	1
9. ¿Establecen un Tiempo Estimado para desarrollar cada actividad?	1
10. ¿Para el Tiempo Estimado se considera la información histórica de la organización?	2
11. ¿Para el Tiempo Estimado se consideran las Metas Cuantitativas para el Proyecto?	4
12. ¿Elaboran un 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?	1
13. ¿Conforman un Equipo de Trabajo, asignando roles y responsabilidades basándose en la Descripción del Proyecto?	1
14. ¿Asignan fechas de inicio y fin a cada una de las actividades para general el Calendario de trabajo tomando en cuenta los recursos asignados, la secuencia y la dependencia de actividades?	1
15. ¿Evalúan y documentan el Costo Estimado del Proyecto?	1
16. ¿Para el Costo Estimado del Proyecto toman en cuenta las Metas Cuantitativas del Proyecto?	4
17. ¿Identifican, describen y evalúan los riesgos que pueden afectar al 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?	1
18. ¿Identifican la probabilidad e impacto de cada riesgo estimando sus implicaciones en los objetivos del proyecto (análisis cuantitativo)?	1
19. ¿Priorizan los efectos de los riesgos sobre los objetivos del proyecto (análisis cualitativo)?	1
20. ¿Desarrollan procedimientos para reducir el impacto de los riesgos, documentándolos en el Plan de Manejo de Riesgos?	1
21. ¿Generan un Plan de Proyecto o lo actualizan antes de iniciar un nuevo ciclo de desarrollo?	1
22. ¿Generan el Plan de Desarrollo en función del Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo?	1
23. ¿Verifican el Plan de Proyecto y el Plan de Desarrollo?	2
24. ¿Corrigen los defectos encontrados en el Plan de Proyecto y en el Plan de Desarrollo con base en el Reporte de Verificación para obtener la aprobación de las correcciones?	2
25. ¿Validan el Plan de Proyecto y el Plan de Desarrollo?	2
26. ¿Corrigen los defectos encontrados en el Plan de Proyecto y Plan de Desarrollo con base en el Reporte de Validación para obtener la aprobación de correcciones?	2
27. ¿Dan inicio formal a un nuevo ciclo de desarrollo una vez que se hayan asegurado el cumplimiento de las condiciones iniciales del ciclo?	3
II. Realización	

28. ¿Acuerdan con el Responsable de Desarrollo y Mantenimiento de del proyecto la asignación de tareas al Equipo de Trabajo incluyendo a los subcontratistas?	1
29. ¿Acuerdan la distribución de la información necesaria para el equipo de trabajo en base al Plan de Comunicación e Implantación?	2
30. ¿Revisan con el Responsable de Desarrollo y Mantenimiento del proyecto la Descripción del Producto, el Equipo de Trabajo y Calendario?	2
31. ¿Dan seguimiento al Plan de Adquisiciones y Capacitación, aceptando o rechazando la Asignación de Recursos Humanos o subcontratistas y distribuyendo los recursos a los miembros del equipo para que puedan llevar a cabo sus actividades?	2
32. ¿Manejan 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?	2
33. ¿Recolectan y analizan los Reportes de Actividades y productos de trabajo?	2
34. ¿Recolecta y analizan los Reportes de Mediciones y Sugerencias de Mejora?	3
35. ¿Registran los costos y recursos reales del ciclo?	2
36. ¿Revisan el Registro de Rastreo de los requerimientos del usuario a través del ciclo de trabajo?	2
37. ¿Revisan los productos generados durante el ciclo, que forman parte de la Configuración de Software?	2
38. ¿Reciben y analizan las Solicitudes de Cambios e incorporan los cambios aprobados en el Plan de Proyecto y en el Plan de Desarrollo?	2
39. ¿Realizan y conducen reuniones de revisión con el equipo de trabajo y con el Cliente, generando Minutas con puntos tratados y acuerdos tomados?	2
III. Evaluación y Control	
40. ¿Evalúan 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?	2
41. ¿Se realiza un seguimiento para controlar el Plan de Manejo de Riesgos, identificando nuevos riesgos y actualizando el plan?	2
42. ¿Generan un Reporte de Seguimiento del proyecto, considerando los Reportes de Actividades?	2
IV. Cierre	
43. ¿Formalizan la terminación del ciclo o proyecto de acuerdo al Protocolo de Entrega establecido en el Plan de Proyecto para obtener un Documento de Aceptación?	1
44. ¿Efectúan un cierre con subcontratistas de acuerdo al contrato establecido?	2
45. ¿Generan un Reporte de Mediciones y Sugerencias de Mejora del proceso, de acuerdo al Plan de Mediciones de Procesos?	3
46. ¿Identifican las Lecciones Aprendidas y las integran a la Base de Conocimiento?	3

7.6. Desarrollo y Mantenimiento de Software

El propósito de Desarrollo y Mantenimiento de Software es la realización sistemática de actividades de análisis, diseño, construcción, integración y pruebas de productos de Software nuevos o modificados cumpliendo con los requerimientos especificados.

Pregunta	Nivel
I. Realización de la fase de Inicio	
1. ¿Revisan con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un entendimiento común y obtener un compromiso con el proyecto?	1
2. ¿Elaboran un Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas?	2
II. Realización de la fase de Requerimientos	
3. ¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?	1
4. ¿Documentan o modifican la Especificación de Requerimientos?	1
5. ¿Identifican y consultan fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos?	1
6. ¿Analizan los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto?	1
7. ¿Elaboran o modifican el prototipo de la interfaz con el usuario?	1
8. ¿Generan o actualizan la Especificación de Requerimientos?	1
9. ¿Verifican la Especificación de Requerimientos?	2
10. ¿Corrigen los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
11. ¿Validan la Especificación de Requerimientos?	2
12. ¿Corrigen los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Validación y obtienen la aprobación de las correcciones?	2
13. ¿Elaboran o modifican un Plan de Pruebas de Sistema?	2
14. ¿Verifican el Plan de Pruebas de Sistema?	2
15. ¿Corrigen los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtiene la aprobación de las correcciones?	2
16. ¿Documentan una versión preliminar del Manual de Usuario o modifican un manual existente?	1
17. ¿Verifican el Manual de Usuario?	2
18. ¿Corrigen los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
19. ¿Incorporan la Especificación de Requerimientos y Manual de Usuario a la Configuración de Software?	1
20. ¿Incorporan la Especificación de Requerimientos, Plan de Pruebas de Sistema y Manual de Usuario como líneas base a la Configuración de Software?	2
21. ¿Elaboran un Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad?	2
22. ¿Elaborar el Reporte de Actividades registrando mediciones requeridas?	4
III. Realización de la fase de Análisis y Diseño	
23. ¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?	1
24. ¿Documentan o modifican el Análisis y Diseño?	1

25. ¿Analizan la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos?	1
26. ¿Describen a detalle la apariencia y el comportamiento de la interfaz con base en la Especificación de Requerimientos de forma que se puedan prever los recursos para su implementación?	1
27. ¿Describen a detalle los componentes que permitan su construcción de manera evidente?	1
28. ¿Generan o actualizan el Análisis y Diseño?	1
29. ¿Generan o modifican el Registro de Rastreo?	2
30. ¿Verifican el Análisis y Diseño y el Registro de Rastreo?	2
31. ¿Corrigen los defectos encontrados en el Análisis y Diseño y en el Registro de Rastreo con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
32. ¿Validan el Análisis y Diseño?	2
33. ¿Corrigen los defectos encontrados en el Análisis y Diseño con base en el Reporte de Validación y obtienen la aprobación de las correcciones?	2
34. ¿Elaboran o modifican un Plan de Pruebas de Integración?	2
35. ¿Verifican el Plan de Pruebas de Integración?	2
36. ¿Corrigen los defectos encontrados en el Plan de Pruebas de Integración con base en el Reporte de Verificación y obtiene la aprobación de las correcciones?	2
37. ¿Incorporan el Análisis y Diseño a la Configuración de Software?	1
38. ¿Incorporan el Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base a la Configuración de Software?	2
39. ¿Elaboran el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad?	2
40. ¿Elaboran el Reporte de Actividades registrando mediciones requeridas?	4
IV. Realización de la fase de Construcción	
41. ¿Distribuyen las tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?	1
42. ¿Construyen o modifican el(los) Componente(s) de software?	1
43. ¿Implementan o modifican Componente(s) de software con base a la parte detallada del Análisis y Diseño?	1
44. ¿Definen y aplican pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del Análisis y Diseño?	2
45. ¿Corrigen los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos)?	2
46. ¿Actualizan el Registro de Rastreo, incorporando los componentes construidos o modificados?	2
47. ¿Verifican el Registro de Rastreo?	2
48. ¿Corrigen los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
49. ¿Incorporan los Componentes a la Configuración de Software?	1
50. ¿Incorporan los Componentes y Registro de Rastreo como líneas base a la Configuración de Software?	2

51. ¿Elaboran el Reporte de Actividades, registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad?	2
52. ¿Elaboran el Reporte de Actividades, registrando las mediciones requeridas?	4
V. Realización de la fase de Integración y Pruebas	
53. ¿Distribuyen tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual?	1
54. ¿Integran los componentes en subsistemas o en el sistema del Software?	1
55. ¿Aplican las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración?	2
56. ¿Corrigen los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (sin defectos)?	2
57. ¿Actualizan el Registro de Rastreo?	2
58. ¿Documentan el Manual de Operación o modifican el manual existente?	1
59. ¿Verifican el Manual de Operación?	2
60. ¿Corrigen los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
61. ¿Realizan las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de Sistema?	2
62. ¿Corrigen los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtienen la aprobación de las correcciones?	2
63. ¿Documentan el Manual de Usuario o modifican el existente?	1
64. ¿Verifican el Manual de Usuario?	2
65. ¿Corrigen los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
66. ¿Incorporan el Software, Manual de Operación y Manual de Usuario a la Configuración de Software?	1
67. ¿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?	2
68. ¿Elaboran el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad?	2
69. ¿Elaboran el Reporte de Actividades registrando las mediciones requeridas?	4
VI. Realización de la fase de Cierre	
70. ¿Documentan el Manual de Mantenimiento o modifican el existente?	2
71. ¿Verifican el Manual de Mantenimiento?	2
72. ¿Corrigen los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtienen la aprobación de las correcciones?	2
73. ¿Incorporan el Manual de Mantenimiento como línea base a la Configuración de Software?	2
74. ¿Identifican las Lecciones Aprendidas y las integran a la Base de Conocimiento? Como ejemplo, se pueden considerar mejores prácticas, experiencias exitosas de manejo de riesgos, problemas recurrentes, entre otras.	3
75. ¿Generan el Reporte de Mediciones y Sugerencias de Mejora?	3
76. ¿Elaboran el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad?	2
77. ¿Elaborar el Reporte de Actividades registrando las mediciones requeridas?	4

8. Anexo C.- Especificación de Casos de Uso

El objetivo del anexo es detallar la arquitectura del sistema Kaizen. Se describen uno a uno los principales casos de uso implementados que cubren las funcionalidades del sistema. Los casos de uso son divididos en base a los actores definidos para el sistema.

8.1. Casos de uso Administrador

La Figura 8.1 presenta el diagrama de casos de uso para el Administrador.

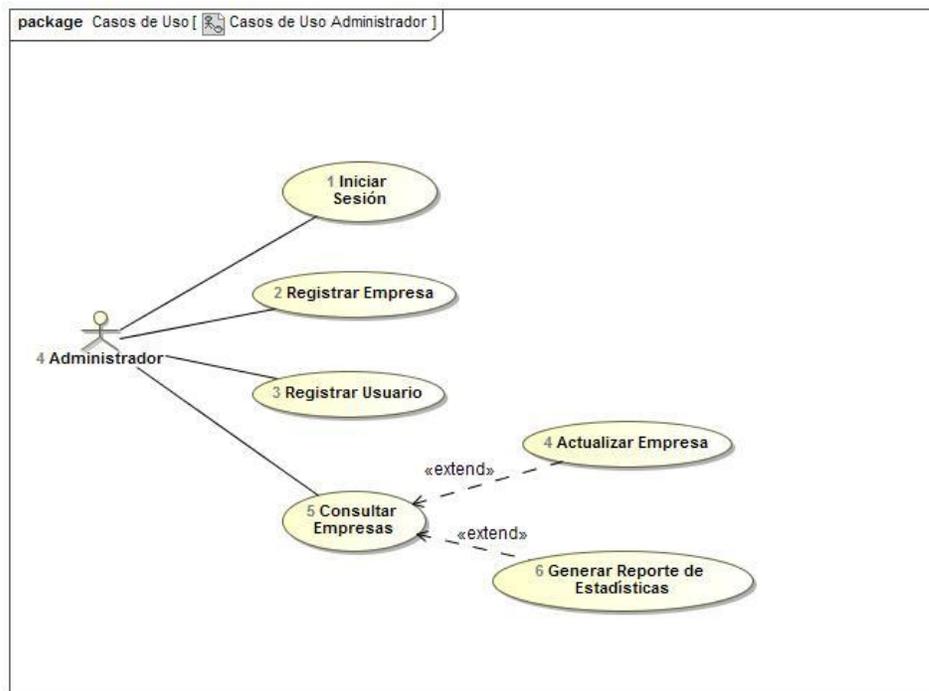


Figura 8.1. Diagrama de casos de uso Administrador.

A continuación se presenta la descripción detallada de cada uno de los casos de uso, en estas secciones también se presenta el diagrama de actividades para brindar un panorama más amplio acerca de la funcionalidad.

8.1.1. Iniciar sesión

La Figura 8.2 muestra que dependiendo del tipo de usuario, éste accede a las funcionalidades del sistema Kaizen en base a un nombre de usuario y contraseña de su registro.

Precondición:

El usuario debe de estar registrado en el sistema Kaizen.

Poscondición:

Administrador con sesión iniciada.

Flujo normal:

1. Ingresar nombre de usuario y contraseña.
2. Enviar solicitud para iniciar sesión (A1).
3. Acceder al sistema.

Flujo alternativo:

- 2.1. Datos Incorrectos.
 - 2.1.1. Mostrar mensaje de datos incorrectos.
 - 2.1.2. Volver a ingresar nombre de usuario y contraseña.

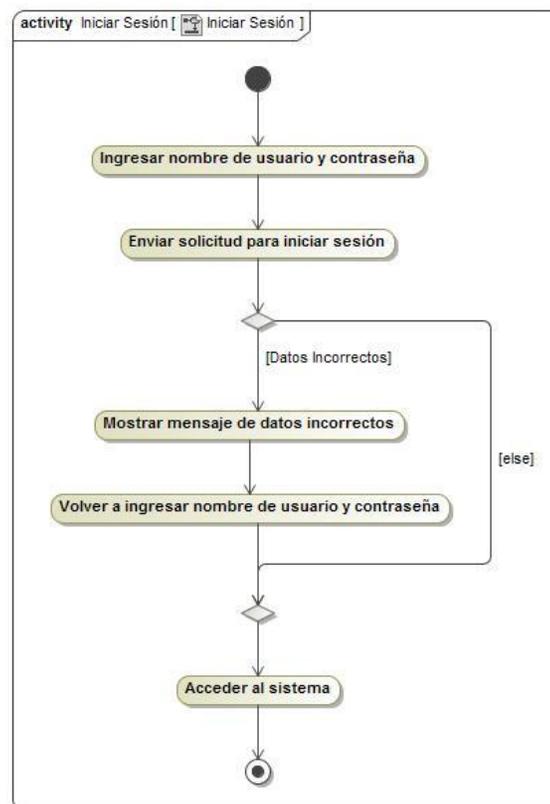


Figura 8.2. Diagrama actividades Iniciar Sesión.

8.1.2. Registrar empresa

La Figura 8.3 muestra que el administrador registra una empresa desarrolladora de software en el sistema Kaizen.

Precondición:

Administrador con sesión iniciada.

Poscondición:

Empresa registrada.

Flujo normal:

1. Ingresar datos de la empresa.
2. Enviar solicitud de registro (A1).
3. Registrar empresa.

Flujo Alternativo:

- 2.1. Empresa ya registrada.
 - 2.1.1. Mostrar mensaje de empresa registrada anteriormente.
 - 2.1.2. Ingresar nuevamente datos de la empresa.

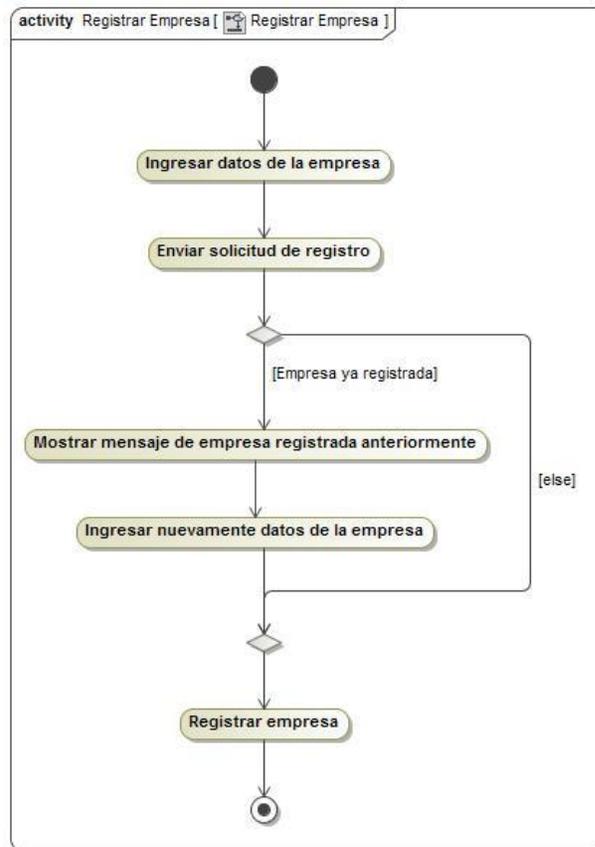


Figura 8.3. Diagrama de actividades Registrar Empresa.

8.1.3. Registrar usuario

La Figura 8.4 muestra que el administrador registra un nuevo usuario asociado con una empresa, definiendo el tipo de usuario (Líder de proyecto o Administrador).

Precondición:

Administrador con sesión iniciada.

Poscondición:

Usuario registrado.

Flujo normal:

1. Ingresar datos del nuevo usuario.
2. Enviar solicitud de registro de usuario (A1).
3. Registrar usuario.

Flujo alternativo:

- 2.1. Usuario ya registrado.

- 2.1.1. Mostrar mensaje de usuario registrado anteriormente.
- 2.1.2. Ingresar nuevamente datos de nuevo usuario.

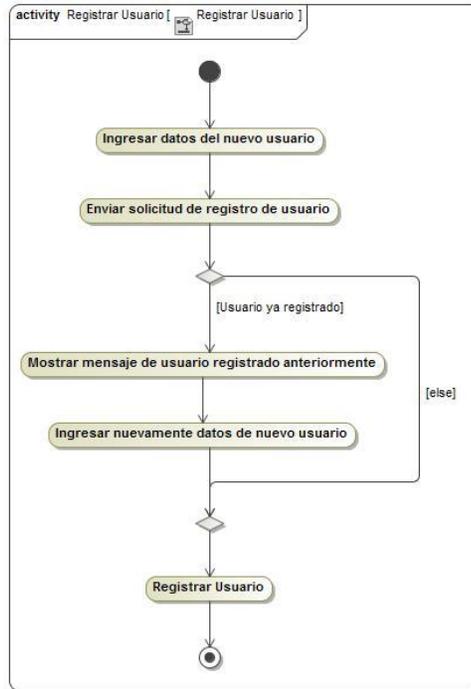


Figura 8.4. Diagrama de actividades Registrar Usuario.

8.1.4. Actualizar empresa

La Figura 8.5 muestra que es necesario actualizar los datos de la empresa en el sistema.

Precondición:

Empresa seleccionada.

Poscondición:

Datos de la empresa actualizados.

Flujo normal:

1. Editar datos de la empresa.
2. Actualizar información de la empresa.

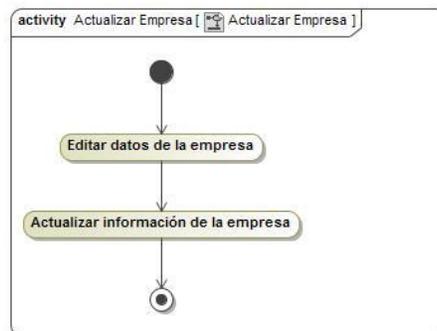


Figura 8.5. Diagrama de actividades Actualizar Empresa.

8.1.5. Consultar empresa

La Figura 8.6 muestra que el administrador consulta las actividades que realiza una empresa, los proyectos de mejora en proceso o terminados y aspectos de los proyectos de mejor como duración, número de miembros, porcentaje de avance e información de las evaluaciones.

Precondición:

Administrador con sesión iniciada.

Poscondición:

Administrador consulta las actividades de las empresas.

Flujo normal:

1. Elegir empresa.
2. Solicitar información.
3. Mostrar información.



Figura 8.6. Diagrama de actividades Consultar Empresa.

8.1.6. Generar reporte de estadísticas

La Figura 8.7 muestra que el administrador genera reportes en base a la información de los proyectos realizados por una empresa, como porcentaje de avance, miembros involucrados, evaluaciones y actividades realizadas.

Precondición:

Empresa seleccionada.

Poscondición:

Reporte con información de las actividades de la empresa generado.

Flujo normal:

1. Seleccionar información para reporte.
2. Generar reporte.



Figura 8.7. Diagrama de actividades Generar Reporte de Estadísticas.

8.2. Casos de uso Líder de Proyecto

La Figura 8.8 muestra los casos de uso para el Líder de proyecto.

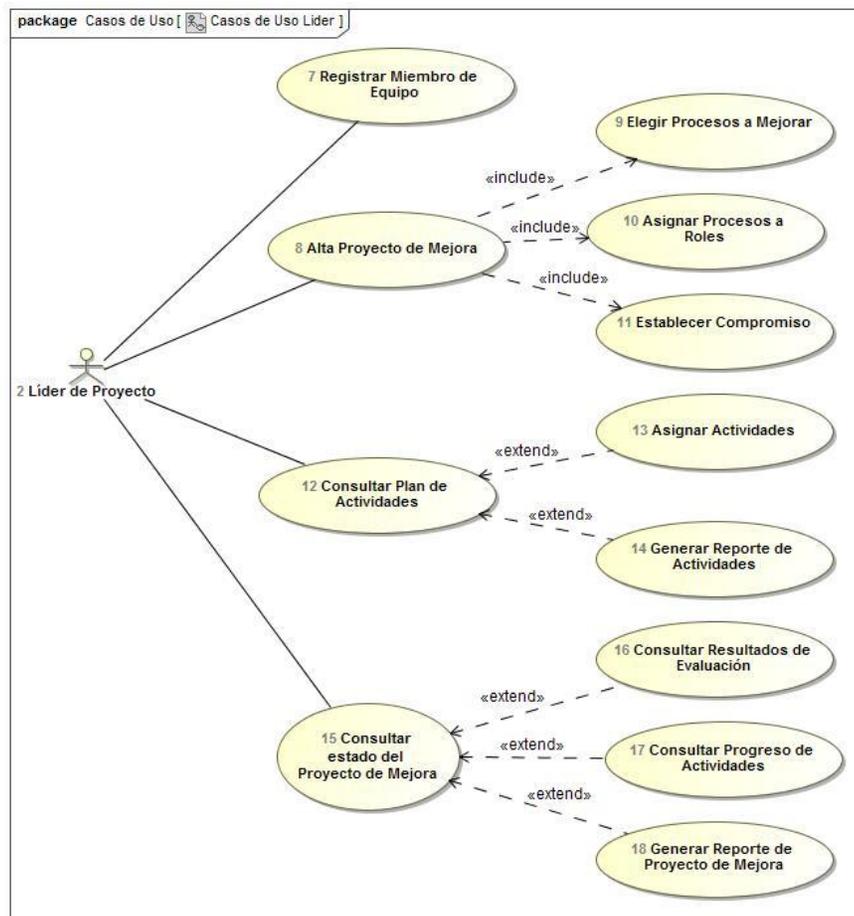


Figura 8.8. Casos de Uso Líder de Proyecto.

Los siguientes apartados presentan una descripción detallada de cada uno de los casos de uso obtenidos en el análisis.

8.2.1. Registrar miembro de equipo

La Figura 8.9 muestra que el líder de proyecto registra a los miembros del equipo que participan en el proyecto de mejora.

Precondición:

Líder de Proyecto con sesión iniciada.

Poscondición:

Miembro del equipo registrado en la base de datos.

Flujo normal:

1. Ingresar datos del miembro.
2. Enviar solicitud de registro (A1).
3. Registrar miembro.

Flujo alternativo:

- 2.1. Miembro ya registrado.
 - 2.1.1. Mostrar mensaje de miembro ya registrado.
 - 2.1.2. Ingresar nuevamente datos del miembro.

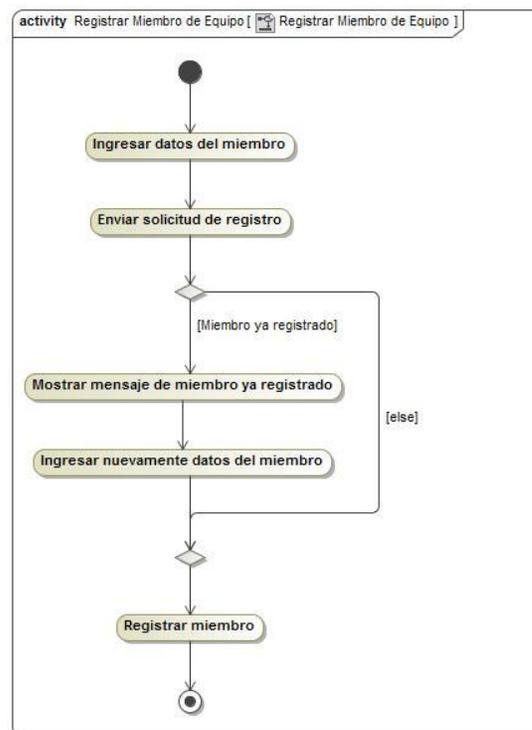


Figura 8.9. Diagrama de Actividades Registrar Miembro de Equipo.

8.2.2. Alta proyecto de mejora

La Figura 8.10 muestra que el líder registra un proyecto de mejora asociado con la empresa a la que pertenece, estableciendo las características y miembros del equipo que participaran.

Precondición:

Líder de Proyecto con sesión iniciada.

No tener otro proyecto de mejora en curso.

Poscondición:

Proyecto de mejora registrado.

Flujo normal:

1. Nombrar al proyecto de mejora.
2. Caso de Uso incluido: Elegir Procesos a Mejorar.
3. Caso de Uso incluido: Asignar Procesos a Roles.
4. Caso de Uso incluido: Establecer Compromiso.
5. Registrar proyecto.

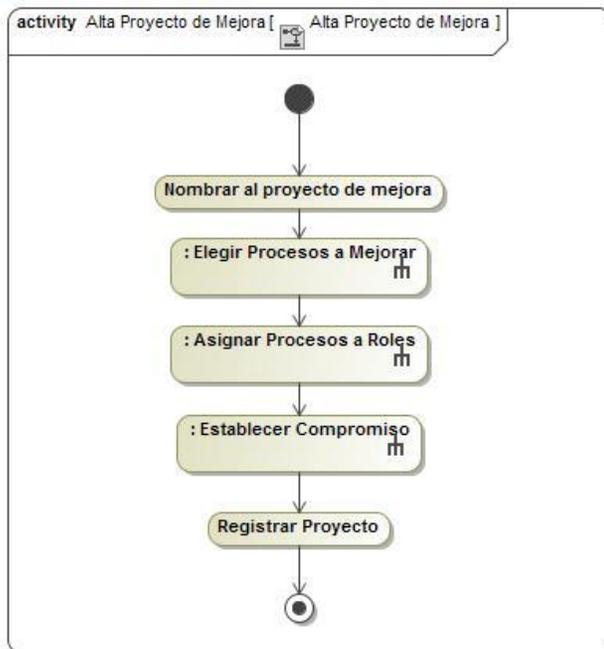


Figura 8.10. Diagrama de actividades de Alta Proyecto Mejora.

8.2.3. Elegir procesos a mejorar

La Figura 8.11 muestra que el líder elige los procesos que se incluyen en el proyecto de mejora. El caso de uso está incluido en Alta Proyecto Mejora.

Precondición:

Registro de proyecto iniciado.

Poscondición:

Procesos elegidos para la iniciativa de mejora.

Flujo normal:

1. Mostrar los procesos disponibles.
2. Elegir el nivel de madurez de los procesos.
3. Elegir los procesos a mejorar.

8.2.4. Asignar procesos a roles

La Figura 8.12 muestra que es necesario asignar a los miembros del equipo de trabajo uno o más procesos, de los cuales serán responsables durante la duración del proyecto. El caso de uso está incluido en Alta Proyecto Mejora.

Precondición:

Registro de proyecto iniciado.

Poscondición:

Procesos elegidos para la iniciativa de mejora.

Flujo normal:

1. Consultar los miembros del equipo registrados.
2. Consultar los procesos disponibles.
3. Asignar procesos a los miembros del equipo.

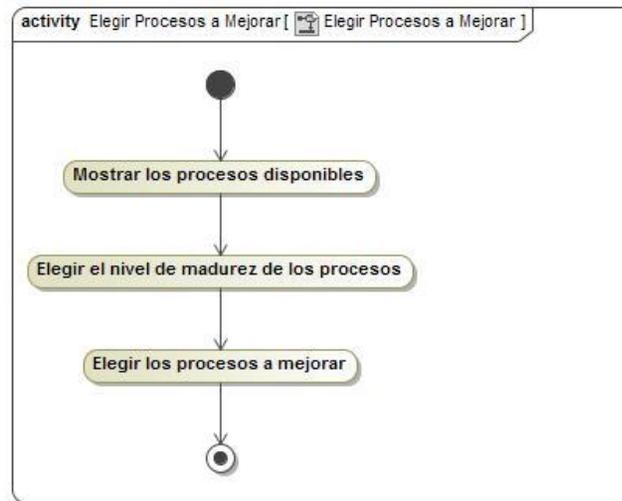


Figura 8.11. Diagrama de Actividades Elegir Procesos a Mejorar.

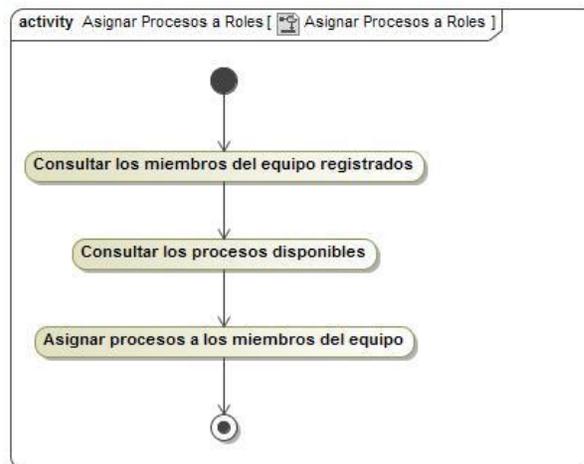


Figura 8.12. Diagrama de Actividades Asignar Proceso a Roles.

8.2.5. Establecer compromiso

La Figura 8.13 muestra que el líder debe comprometerse con el proyecto de mejora a través del establecimiento de responsabilidades que todas las partes involucradas en la iniciativa de mejora deben aceptar. Este caso de uso está incluido en el caso de uso Alta Proyecto Mejora.

Precondición:

Propiedades del proyecto establecidas.

Poscondición:

Compromiso de mejora establecido.

Flujo normal:

1. Mostrar documento de compromiso.
2. Aceptar compromiso.

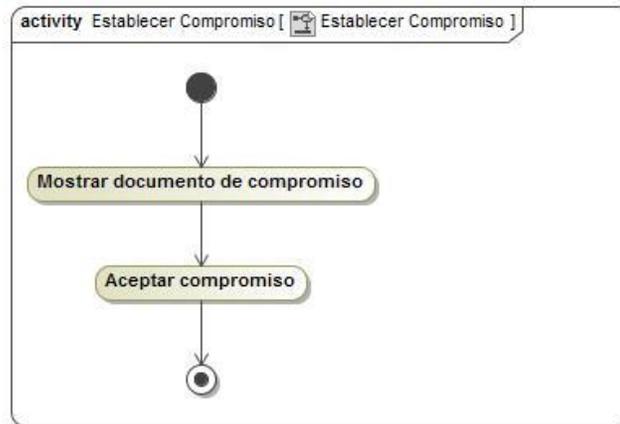


Figura 8.13. Diagrama de Actividades Establecer Compromiso.

8.2.6. Consultar plan de actividades

La Figura 8.14 muestra que el líder de proyecto puede consultar el plan de actividades que debe realizar, generado por Kaizen en base a los resultados obtenidos en la fase de evaluación, consultando las estimaciones de tiempo para realizar una actividad y los miembros del equipo que deben realizarla.

Precondición:

Evaluación realizada.

Poscondición:

Consulta del plan de actividades a realizar por los miembros del equipo dentro del proyecto de mejora.

Flujo normal:

1. Acceder a consulta de plan de actividades.
2. Visualizar plan de actividades.
3. Caso de Uso Extendido: Asignar Actividades.
4. Caso de Uso Extendido: Generar Reporte de Actividades.

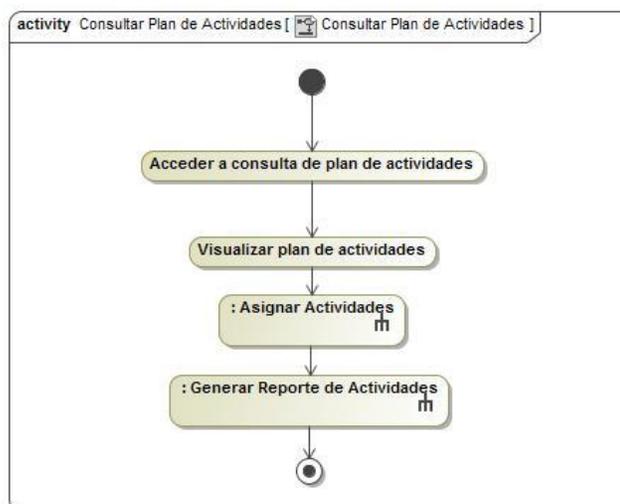


Figura 8.14. Diagrama de Actividades Consultar Plan de Actividades.

8.2.7. Asignar actividades

La Figura 8.15 muestra que el líder de proyecto realiza una asignación, no contemplada en el plan de actividades original. Este es un caso de uso extendido de Consultar Plan de Actividades.

Precondición:

Plan de actividades consultado.

Poscondición:

Asignación de actividades a un miembro del equipo.

Flujo Normal:

1. Seleccionar actividad.
2. Seleccionar miembro de equipo.
3. Asignar actividad a miembro.

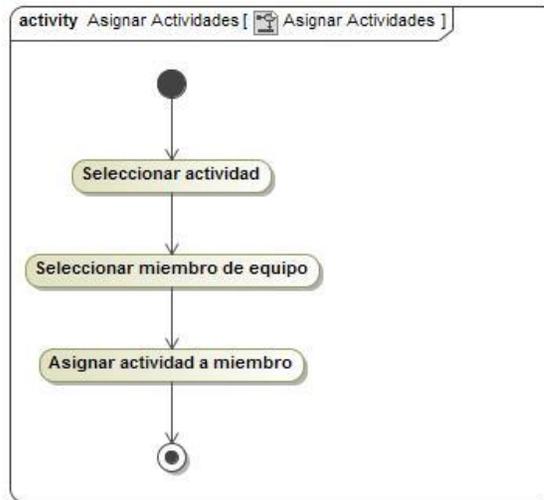


Figura 8.15. Diagrama de Actividades Asignar Actividades.

8.2.8. Reporte de actividades

La Figura 8.16 muestra que el líder de proyecto genera un reporte donde se detallan las actividades que se deberán de llevar a cabo durante el proyecto de mejora. Reporte de Actividades extiende el caso de uso Consultar Plan de Actividades.

Precondición:

Plan de actividades consultado.

Poscondición:

Reporte de plan de actividades generado.

Flujo normal:

1. Elegir elementos del reporte.
2. Generar reporte.



Figura 8.16. Diagrama de Actividades Generar Reporte de Actividades.

8.2.9. Iniciar proyecto de mejora

La Figura 8.17 muestra que el líder de proyecto da comienzo a la implantación de la iniciativa por medio de la consecución de las actividades generadas por medio de Kaizen. De igual forma se realiza una notificación a los miembros del equipo que el proyecto de mejora está en marcha.

Precondición:

- Evaluación realizada.
- Plan de actividades generado.

Poscondición:

- Proyecto de mejora iniciado.

Flujo normal:

1. Iniciar el proyecto de mejora.
2. Notificar a los miembros el inicio del proyecto.

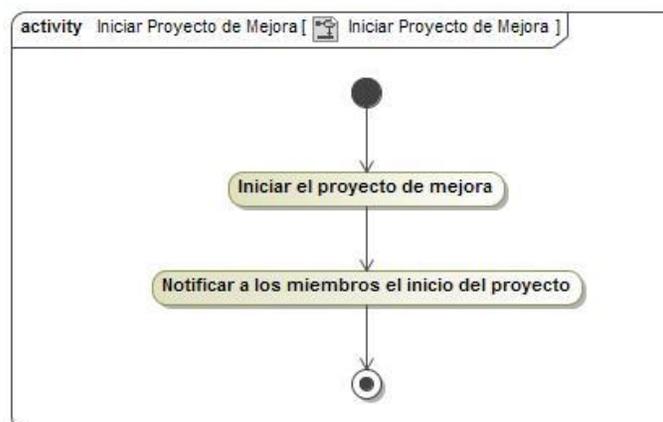


Figura 8.17. Diagrama de Actividades Iniciar Proyecto de Mejora.

8.2.10. Consultar estado del proyecto de mejora

La Figura 8.18 muestra que el líder de proyecto puede consultar el estado del proyecto de mejora en base al porcentaje de avance del plan de actividades generado por Kaizen.

Precondición:

Evaluación realizada.
Plan de actividades generado.

Poscondición:

Proyecto de mejora iniciado.

Flujo normal:

1. Acceder a consulta del estado del proyecto.
2. Visualizar estado del avance del proyecto.
3. Caso de Uso Extendido: Consultar Resultados de Evaluación.
4. Caso de Uso Extendido: Consultar Progreso de Actividades.
5. Caso de Uso Extendido: Generar Reporte de Proyecto de Mejora.

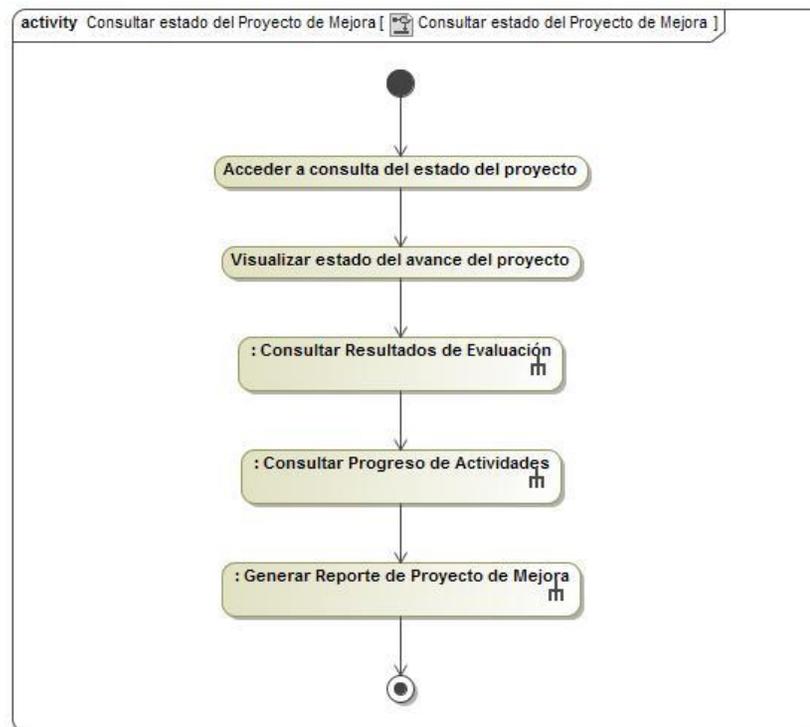


Figura 8.18. Diagrama de Actividades Consultar Estado del Proyecto de Mejora.

8.2.11. Consultar resultados de evaluación

La Figura 8.19 muestra que el líder de proyecto puede consultar los resultados obtenidos en la fase de evaluación. Los resultados se muestran por proceso, por miembro o por nivel de madurez obtenido. Este es un caso de uso extendido de Consultar Estado del Proyecto de Mejora.

Precondición:

Evaluación realizada.

Poscondición:

Consulta de los resultados de la evaluación.

Flujo normal:

1. Acceder a consulta de resultados.
2. Consultar resultados de la evaluación.

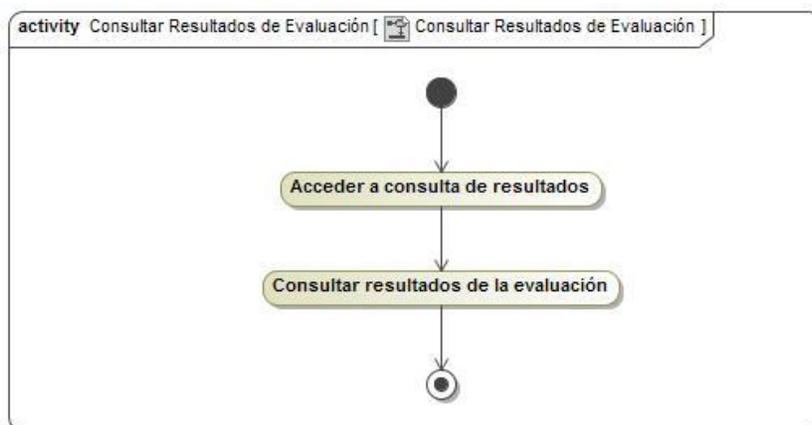


Figura 8.19. Diagrama de Actividades Consultar Resultados Evaluación.

8.2.12. Consultar progreso de actividades

La Figura 8.20 muestra que el líder de proyecto puede consultar el estado en que se encuentran las actividades como pendientes, terminadas o el porcentaje de avance. Este es un caso de uso extendido de Consultar Estado del Proyecto de Mejora.

Precondición:

Actividad elegida.

Poscondición:

Muestra el porcentaje de avance por actividades.

Flujo normal:

1. Elegir proceso.
2. Consultar progreso de las actividades.



Figura 8.20. Diagrama de Actividades Consultar Progreso de Actividades.

8.2.13. Generar reporte de proyecto de mejora

La Figura 8.21 muestra que el líder de proyecto genera un reporte acerca del estado del proyecto de mejora. El reporte puede incluir los resultados de la evaluación, el porcentaje de avance del proyecto, porcentaje de avance de las actividades o avance por miembro. Este es un caso de uso extendido de Consultar Estado del Proyecto de Mejora.

Poscondición:

Reporte de proyecto de mejora generado.

Flujo normal:

1. Elegir la información del reporte.
2. Generar el reporte.

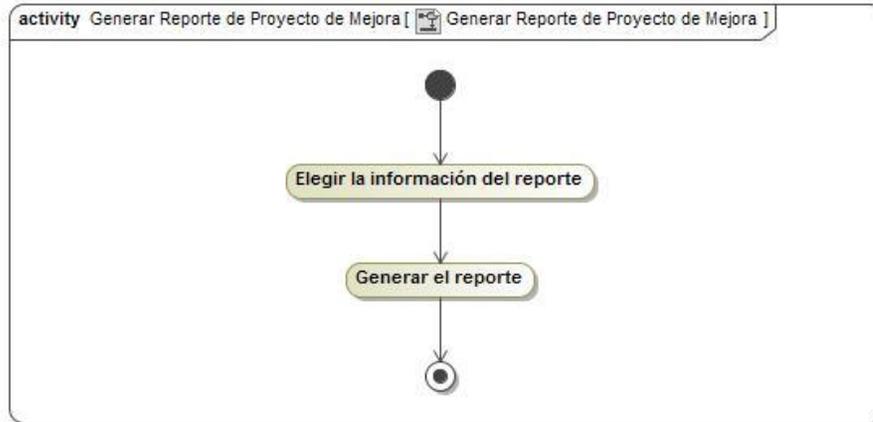


Figura 8.21. Diagrama de Actividades Reporte de Proyecto de Mejora.

8.3. Casos de uso para Miembro de Equipo

La Figura 8.22 presenta los casos de uso obtenidos para el actor Miembro de Equipo. En este caso, al ser el Líder de Proyecto un tipo específico de Miembro de Equipo, algunos de los casos de uso también pueden estar asociados a un Líder de Proyecto. A continuación se presenta la descripción detallada de cada uno de los casos de uso.

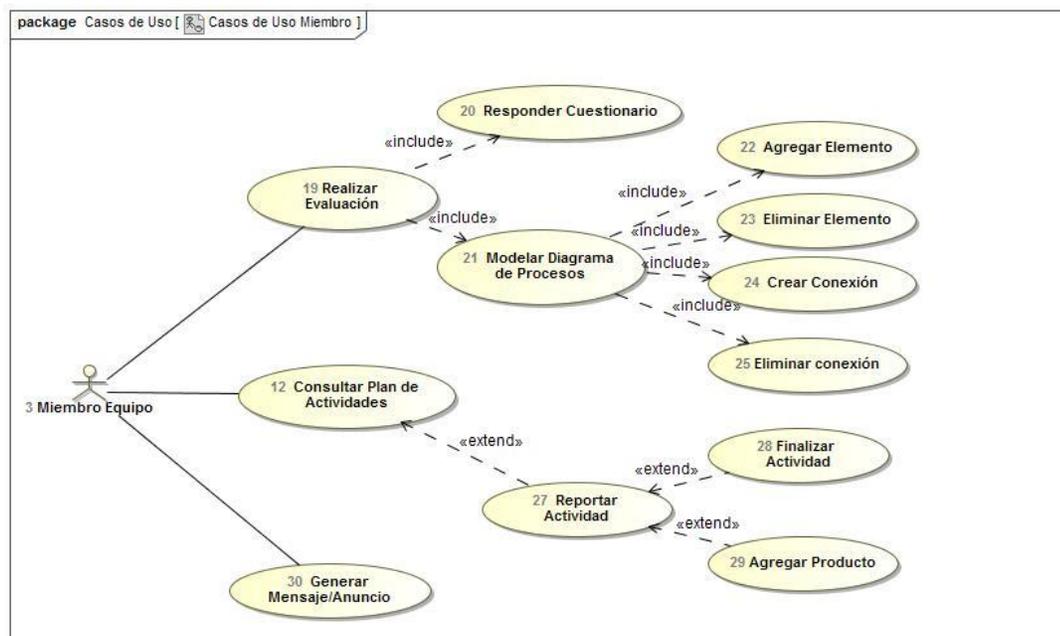


Figura 8.22. Casos de uso para Miembro de Equipo.

8.3.1. Realizar evaluación

La Figura 8.23 muestra que el Miembro de Equipo debe completar la evaluación del proceso que se le ha asignado. La evaluación consta de dos partes, en primera instancia se somete a un cuestionario relacionado con las actividades que debe de realizar un proceso de desarrollo, seguido del modelado del proceso actual para encontrar posibles deficiencias contra el modelo de referencia utilizado.

Precondición:

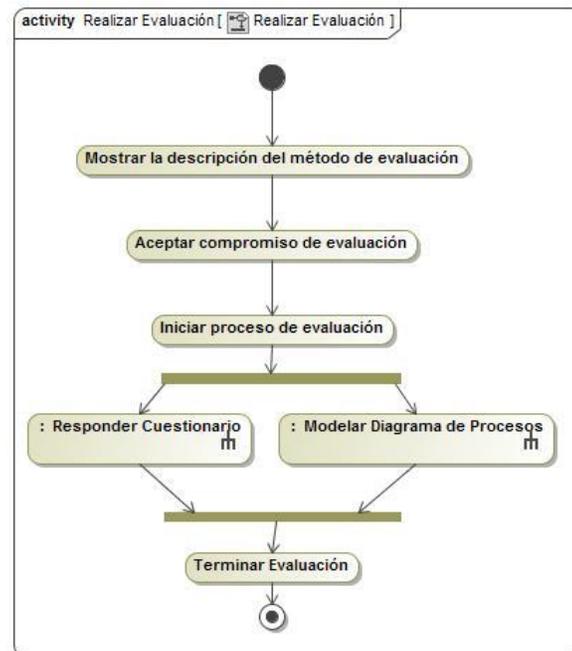
Miembro de equipo con sesión iniciada.
Proyecto de mejora iniciado.

Poscondición:

Evaluación de proceso realizada.

Flujo normal:

1. Mostrar la descripción del método de evaluación.
2. Aceptar compromiso de evaluación.
3. Iniciar proceso de evaluación.
4. Caso de Uso incluido: Responder Cuestionario.
5. Caso de Uso incluido: Modelar Diagrama de Procesos.
6. Terminar Evaluación.

**Figura 8.23.** Diagrama de actividades Realizar Evaluación.

8.3.2. Responder cuestionario

La Figura 8.24 muestra que el Miembro de Equipo debe responder a las distintas preguntas que componen al cuestionario.

Precondición:

Evaluación iniciada.

Poscondición:

Cuestionario contestado.

Respuestas almacenadas en la base de datos.

Flujo normal:

1. Mostrar pregunta.
2. Elegir respuesta.
3. Enviar respuesta (A1).
4. Terminar Cuestionario.

Flujo alternativo:

- 3.1. Faltan preguntas.
 - 3.1.1. Mostrar nueva pregunta.

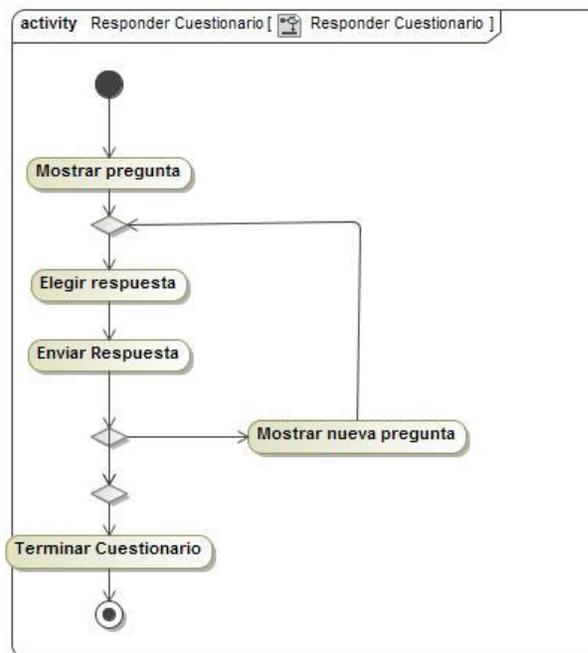


Figura 8.24. Diagrama Actividades Responder Cuestionario.

8.3.3. Modelar diagrama de procesos

La Figura 8.25 muestra que el Miembro de Equipo debe modelar el proceso actual de desarrollo para que Kaizen realice un mapeo con el diagrama recomendado por el modelo de referencia, para obtener así las inconsistencias que se presentan en el proceso actual.

Precondición:

Evaluación iniciada.

Poscondición:

Diagrama de procesos actual modelado.

Flujo normal:

1. Mostrar información acerca del modelado
2. Caso de Uso incluido: Agregar Elemento
3. Caso de Uso incluido: Eliminar Elemento
4. Caso de Uso incluido: Crear Conexión

5. Caso de Uso incluido: Eliminar Conexión

6. Guardar diagrama de procesos

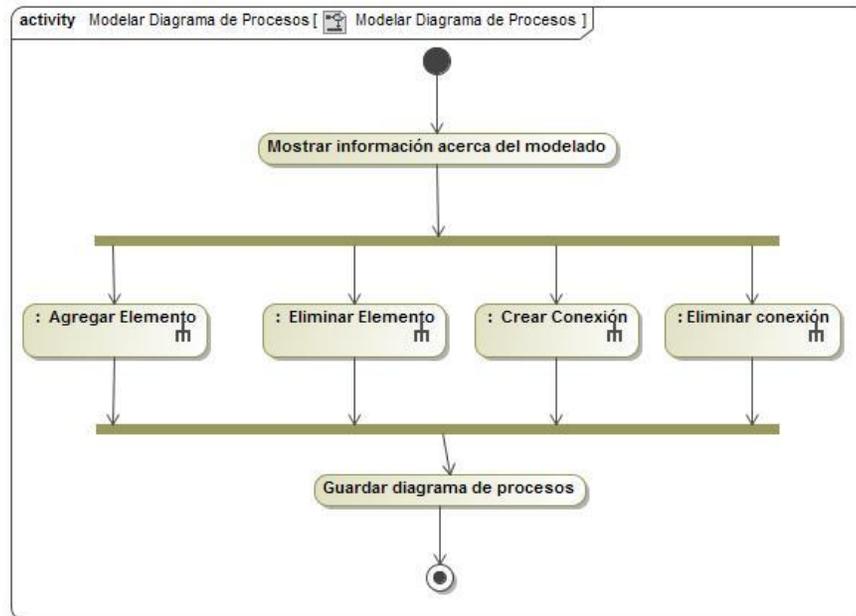


Figura 8.25. Diagrama de actividades Modelar Diagrama de Procesos.

8.3.4. Agregar elemento

La Figura 8.26 muestra que puede ser necesario agregar un nuevo elemento al diagrama de procesos en una posición determinada. Este caso de uso está incluido dentro del caso de uso Modelar Diagrama de Procesos.

Precondición:

Editor de diagramas iniciado.

Poscondición:

Elemento agregado al diagrama de procesos.

Flujo normal:

1. Elegir elemento.
2. Establecer posición del elemento.
3. Agregar elemento.

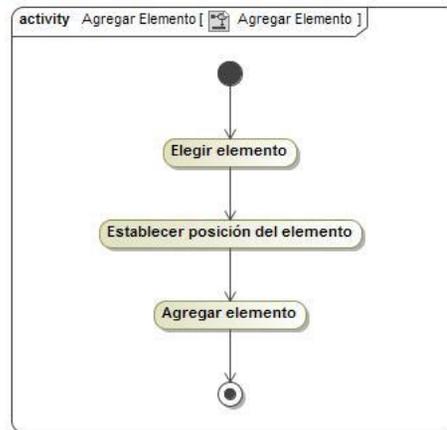


Figura 8.26. Diagrama de actividades Agregar Elemento.

8.3.5. Eliminar elemento

La Figura 8.27 muestra que es posible eliminar un determinado elemento del diagrama de procesos. El caso de uso Eliminar Elemento se encuentra incluido dentro de Modelar Diagrama de Procesos.

Precondición:

Elemento agregado al diagrama de procesos.

Poscondición:

Elemento eliminado del diagrama de procesos.

Flujo normal:

1. Seleccionar elemento.
2. Eliminar elemento.

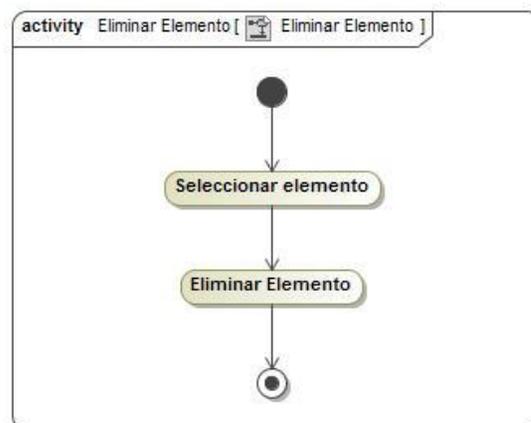


Figura 8.27. Diagrama de actividades Eliminar Elemento.

8.3.6. Crear conexión

La Figura 8.28 muestra que es posible crear una conexión entre dos elementos previamente agregados al diagrama de procesos. Crear Conexión es un caso de uso incluido en Modelar Diagrama de Procesos.

Precondición:

Dos o más elementos agregados al diagrama de procesos.

Poscondición:

Conexión entre dos elementos del diagrama de procesos.

Flujo normal:

1. Elegir elemento origen.
2. Elegir elemento destino.
3. Crear conexión.

8.3.7. Eliminar conexión

La Figura 8.29 muestra que es posible eliminar una conexión entre elementos del diagrama de procesos. Eliminar Conexión es un caso de uso incluido en Modelar Diagrama de Procesos.

Precondición:

Conexión creada en el diagrama de procesos.

Poscondición:

Conexión eliminada en el diagrama de procesos.

Flujo normal:

1. Elegir conexión.
2. Eliminar conexión.

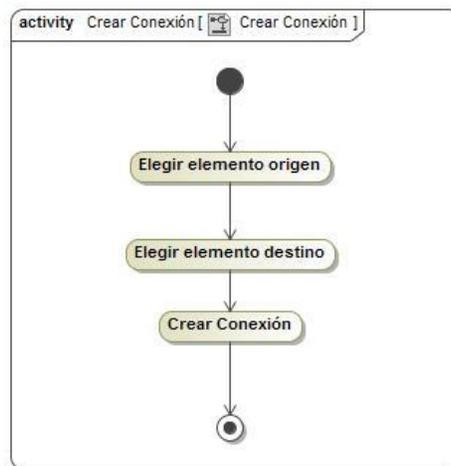


Figura 8.28. Diagrama de actividades Crear Conexión.

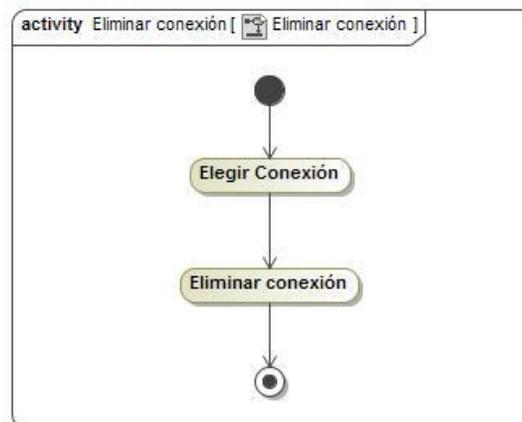


Figura 8.29. Diagrama de actividades Eliminar Conexión.

8.3.8. Consultar plan de actividades

La Figura 8.30 muestra que el Miembro de Equipo puede consultar el plan de actividades generado por Kaizen a partir de la fase de evaluación. Dicho miembro consulta que actividades tiene que realizar a lo largo del proyecto.

Precondición:

Evaluación realizada.

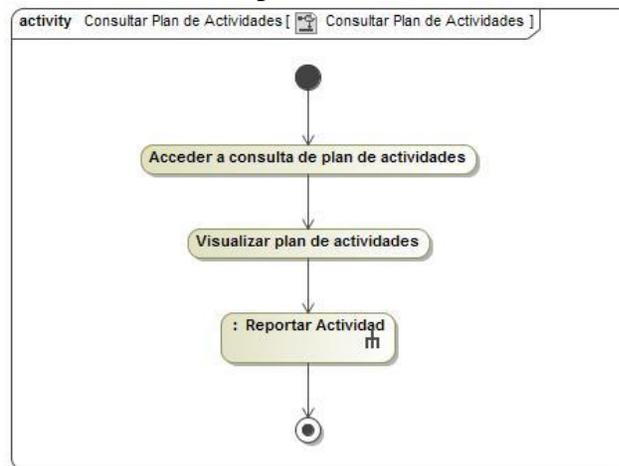
Plan de actividades generado por el Líder de Proyecto.

Poscondición:

Consulta del plan de actividades a realizar por los miembros del equipo dentro del proyecto de mejora.

Flujo normal:

1. Acceder a consulta de plan de actividades.
2. Visualizar plan de actividades.
3. Caso de Uso extendido: Reportar Actividad.

**Figura 8.30.** Diagrama de Actividades Consultar Plan de Actividades.

8.3.9. Finalizar actividad

La Figura 8.31 muestra que el Miembro de Equipo debe reportar el final de una actividad que se le ha asignado previamente en el plan de actividades. El caso de uso Finalizar Actividad es un extendido de Reportar Actividad.

Precondición:

Actividad seleccionada.

Poscondición:

Reporte de final de actividad.

Flujo normal:

1. Realizar anotaciones de la actividad.
2. Finalizar la actividad.
3. Notificar el final de la actividad.



Figura 8.31. Diagrama de actividades Finalizar Actividad.

8.3.10. Generar mensaje/anuncio

La Figura 8.32 muestra que el Miembro de Equipo puede publicar un mensaje o anuncio en el sistema para que uno o más miembros del proyecto lo visualicen.

Precondición:

Miembro de Equipo con sesión iniciada.

Poscondición:

Mensaje publicado en Kaizen.

Flujo normal:

1. Editar mensaje.
2. Elegir destinatarios.
3. Publicar mensaje.

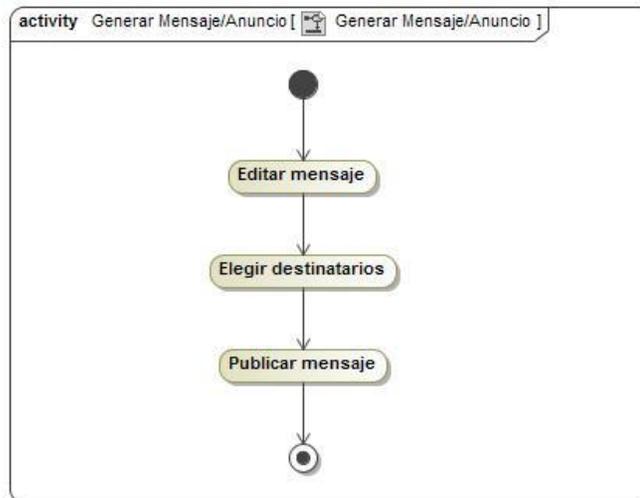


Figura 8.32. Diagrama de actividades Generar Mensaje/Anuncio.

9. Anexo D.- Experiencia de uso: proyecto de mejora para implantar un proceso DMS-MoProSoft nivel 1

El propósito de este anexo es presentar paso a paso cómo se utilizó la herramienta Kaizen para llevar a cabo la experimentación presentada en el Capítulo 4 de esta tesis. En particular se presenta la experimentación para la introducción de las prácticas recomendadas por el modelo MoProSoft en su proceso Desarrollo y Mantenimiento de Software en Nivel 1 (Realizado).

9.1. Configuración de Empresa y Responsable de Mejora

Para utilizar las funcionalidades que ofrece Kaizen, el primer paso a realizar es registrar a la empresa participante y posteriormente a la persona que fungirá como responsable del proyecto de mejora (véase Figura 9.1), esta tarea es realizada por el administrador, con quien las organizaciones interesadas deben ponerse en contacto previamente.

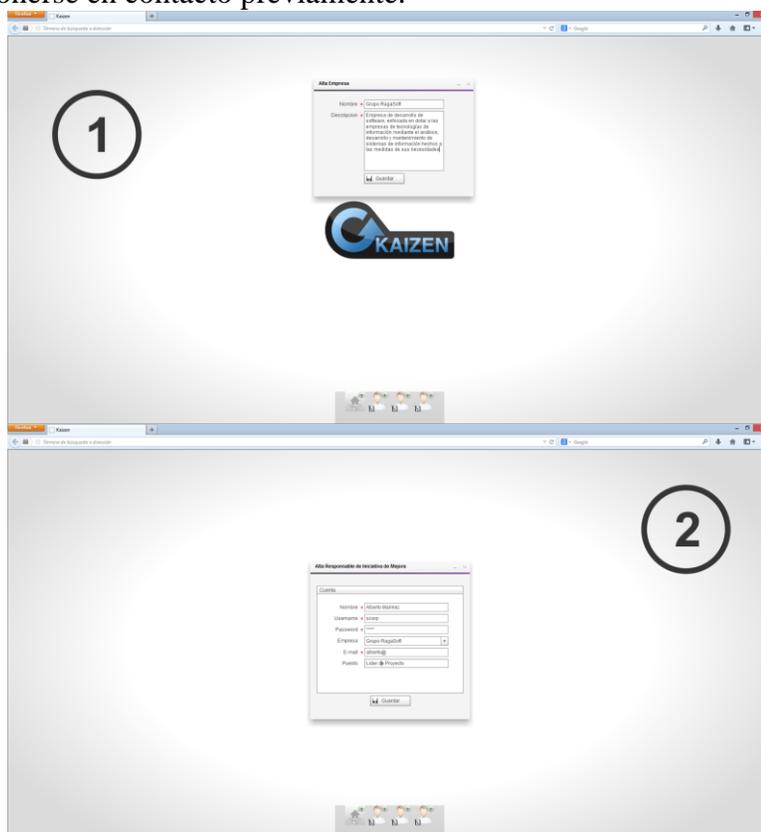


Figura 9.1. Alta empresa (1) y registro de responsable de iniciativa de mejora (2).

Una vez registradas tanto la empresa como el responsable de la mejora, los responsables pueden acceder a las funcionalidades de la herramienta para llevar a cabo la iniciativa de mejora.

9.2. Configuración de la Iniciativa y Establecimiento del Compromiso

Cuando el responsable de la mejora se ha registrado, puede acceder a las funcionalidades de Kaizen por medio del nombre de usuario y contraseña del registro previo, como se muestra en la Figura 9.2.

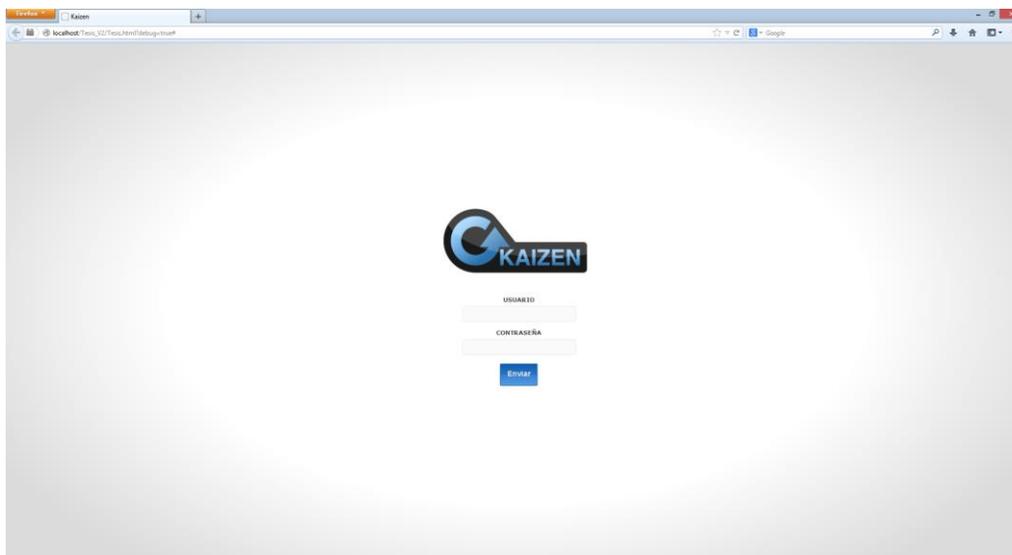


Figura 9.2. Ingreso a Kaizen.

En primera instancia Kaizen solo ofrece dos funcionalidades: registrar al equipo de trabajo y dar de alta y configurar la iniciativa de mejora (véase Figura 9.3). La primera funcionalidad se ocupa para dar de alta a todo el personal que formará parte del equipo de trabajo, dentro de la información solicitada destaca la categoría (Alta Dirección, Gerencia y Operación) del empleado, ya que de ella depende el tipo de responsabilidad que se les puede asignar dentro del programa de mejora. Para el caso del registro y configuración de un nuevo programa de mejora se tienen que cubrir distintas etapas.

Para poder registrar un programa de mejora, el líder de proyecto de mejora debe asegurarse de haber dado de alta a todo el personal considerado para el programa de mejora, ya que es necesario para la asignación de roles dentro de la configuración. Como primer paso el líder del proyecto debe establecer un nombre para el proyecto, así como el origen, el objetivo general y los objetivos específicos de la iniciativa (véase Figura 9.4). Lo anterior es necesario para establecer las expectativas de la organización en relación al programa de mejora.

Los siguientes pasos se enfocan en la configuración de la iniciativa de mejora. Como lo muestra la Figura 9.5, es necesario establecer la fecha de inicio de la iniciativa. Después se eligió el tipo de evaluación: Evaluación por nivel de madurez y por nivel de capacidad. En la primera se realiza una evaluación de todos los procesos establecidos en MoProSoft, hasta el nivel indicado; en el segundo tipo se realiza la evaluación únicamente con los procesos elegidos, hasta el nivel indicado. Para este caso se eligió la segunda opción en un Nivel 1 o Realizado.

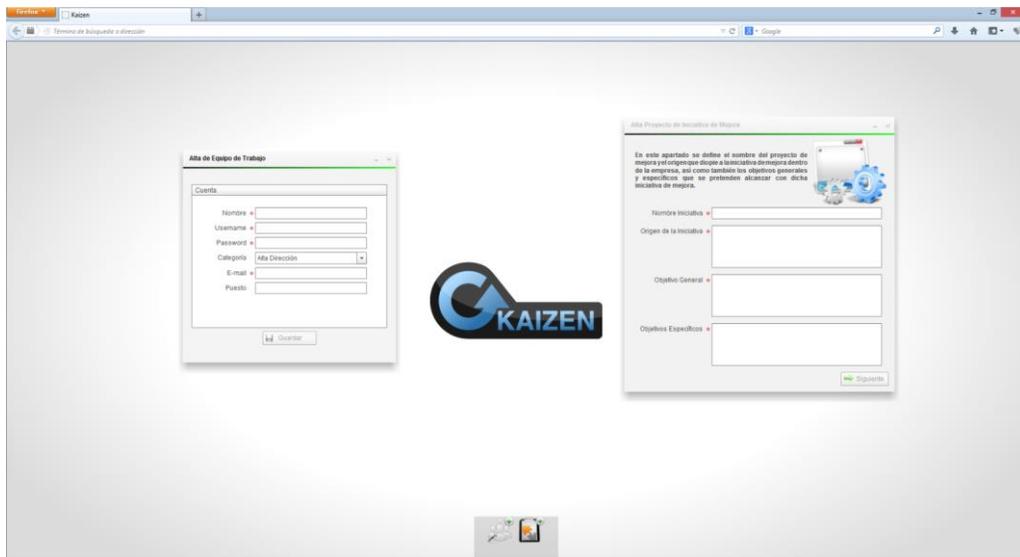


Figura 9.3. Registro de equipo de trabajo y programa de mejora.



Figura 9.4. Registro de nueva iniciativa de mejora.

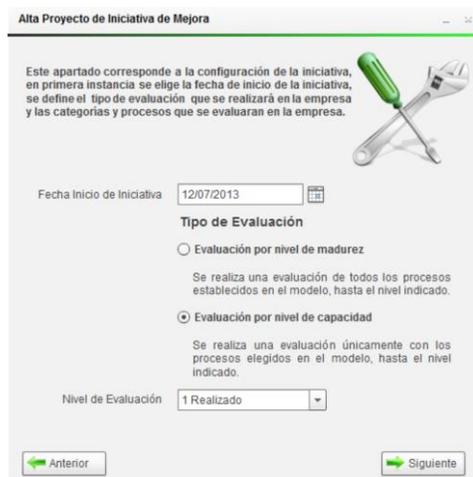


Figura 9.5. Configuración de inicio y evaluación.

Una vez elegido el tipo y nivel de la evaluación, se procedió a elegir los procesos a trabajar en el programa de mejora, para este caso se eligió el proceso de Desarrollo y Mantenimiento de Software (véase Figura 9.6).

Alta Proyecto de Iniciativa de Mejora

En este apartado se configuran las categorías y procesos que se incluirán en la iniciativa de mejora, en las etapas de Inicio, Diagnóstico, Establecimiento, Actuación y Aprendizaje de la iniciativa de Mejora.

Elección de Procesos a Mejorar

Para elegir los procesos a incluir en la iniciativa de mejora arrastrar los procesos elegidos a la lista de mejora

Proceso	Categoría
Gestión de Negocios	Alta Dirección
Gestión de Procesos	Gerencia
Gestión de Proyectos	Gerencia
Gestión de Recursos	Gerencia
Administración de Proyectos Específicos	Operación

Proceso	Categoría
Desarrollo y Mantenimiento de Software	Operación

Anterior Siguiente

Figura 9.6. Elección de los procesos a trabajar durante la iniciativa.

Por último, el encargado del programa de mejora debe asignar los roles establecidos en MoProSoft a cada uno de los integrantes del grupo de trabajo. De esta forma, en la parte de planeación se asignarán aquellas actividades al rol que correspondan. De igual forma, el encargado establece a los líderes de proyecto que fungirán como los responsables del proceso DMS, los cuales cumplirán con la etapa de evaluación. En la configuración, como lo muestra la Figura 9.7 se listan todos los miembros del equipo de trabajo, a los cuales se les asigna uno o más roles, lo anterior debido a que en pequeñas organizaciones un empleado puede asumir distintos roles, o distintos empleados pueden compartir un rol.

Alta Proyecto de Iniciativa de Mejora

En este apartado se configura el grupo de trabajo que llevará a cabo la iniciativa de mejora dentro de la empresa. Por medio de la asignación de roles establecidos por el modelo MoProSoft, a cada uno de los integrantes del equipo, con lo cual, las actividades y tareas de responsabilidad del rol quedará a cargo del elemento del grupo de trabajo elegido.

Para asignar responsables a cada uno de los procesos a mejorar, se deberán arrastrar los miembros del equipo de trabajo a cada uno de los procesos.

Equipo de Trabajo:

- Alberto Martínez
- Uriel Vasquez
- Enrique Lopez
- Jose Miguel Martinez
- Luis Velasco
- Pedro MuñozCano

Datos

Nombre: Uriel Vasquez
Puesto: Líder de Proyecto
Categoría: Gerencia

Roles MoProSoft

- RDM
- AN
- DU
- DI
- PR
- RM

Roles Asignados

- RDM
- AN

Analista

Conocimiento y experiencia en la obtención, especificación y análisis de los requerimientos.

Anterior Guardar

Figura 9.7. Configuración del equipo de trabajo.

Una vez hecha la configuración se registra el nuevo programa de mejora y en base a la fecha de inicio configurada se da inicio al programa de mejora. Con lo anterior queda establecido el

compromiso de la organización, ya que a partir de ese momento Kaizen empieza a generar las alertas referentes a la etapa de evaluación basados en los roles establecidos en esta primera etapa.

9.3. Evaluación del proceso actual y presentación de resultados

Ya configurada el tipo de evaluación y procesos a evaluar, Kaizen emite alertas a los responsables de tales procesos para realizar las evaluaciones pertinentes. Una vez que el responsable del proceso DMS inicie sesión el sistema lo alertará de que existen nuevas evaluaciones que tiene que atender y aparecerán en la interfaz nuevos iconos haciendo referencia a la evaluación por cuestionario y evaluación por diagrama de procesos actual, presentado en el Capítulo 3 de esta tesis (véase Figura 9.8). Por cada proceso del que sea responsable se añaden evaluaciones por cuestionario y diagrama.

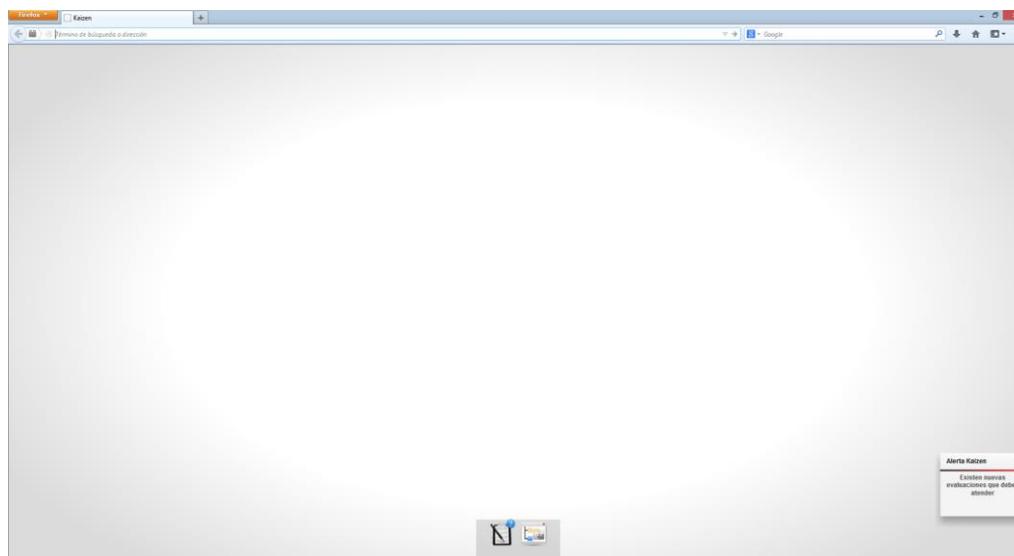


Figura 9.8. Notificación de nuevas evaluaciones.

El evaluado puede elegir con que evaluación comenzar, pero es requisito imprescindible para continuar, el culminar las dos evaluaciones, así como que todos los responsables del proceso evaluado terminen las evaluaciones. Para este ejemplo, existen dos responsables para el proceso DMS. Con el diseño del cuestionario se pretende que el evaluado se enfoque solamente en la pregunta actual, en lugar de mostrar el conjunto completo de preguntas, para poder elegir la respuesta correcta o que se adapte más a la situación de la empresa. De igual forma, si el evaluado tiene dudas acerca de los productos que se le mencionan en los cuestionamientos, Kaizen ofrece una descripción del producto en cuestión (véase Figura 9.9).



Figura 9.9. Evaluación por cuestionario.

Las preguntas se seleccionan en base a las fases que sugiere el modelo, para que el evaluado esté enterado de la fase de desarrollo a la que se refiere la pregunta. En la evaluación por diagrama de proceso, el responsable del proceso crea un diagrama que plasma la situación actual de la empresa, las actividades que se realizan, los responsables de las actividades, la dependencia de actividades, el flujo de ejecución y las decisiones tomadas durante el proceso. Como se puede observar en la Figura 9.10, Kaizen ofrece ciertas funcionalidades para la edición visual del diagrama de procesos, con lo que se busca que el evaluado plasme el diagrama más cercano al proceso actual, dotado de características visuales para la mejor comprensión del mismo.

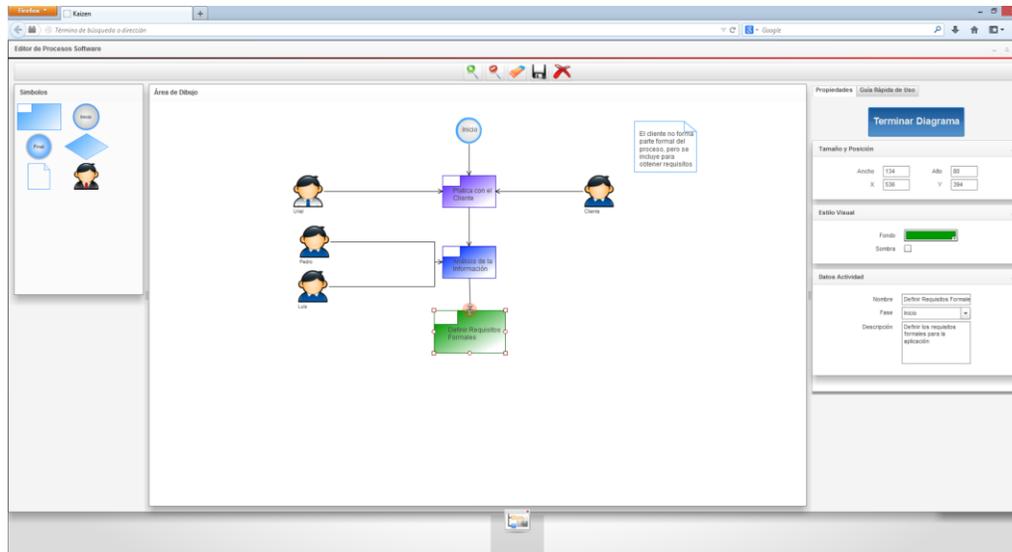


Figura 9.10. Evaluación por diagrama de procesos.

Cada vez que alguno de los responsables de proceso termine una evaluación, Kaizen notificará al encargado de la mejora la culminación de la evaluación, cuando un responsable cumpla con la evaluación tanto de cuestionario como de diagrama de procesos, el encargado de la mejora puede revisar los resultados obtenidos. La Figura 9.11 muestra cuando uno de los dos responsables ha cumplido con el ciclo de evaluación. La pantalla de resultados ofrece un resumen acerca de los hallazgos encontrados en la evaluación.

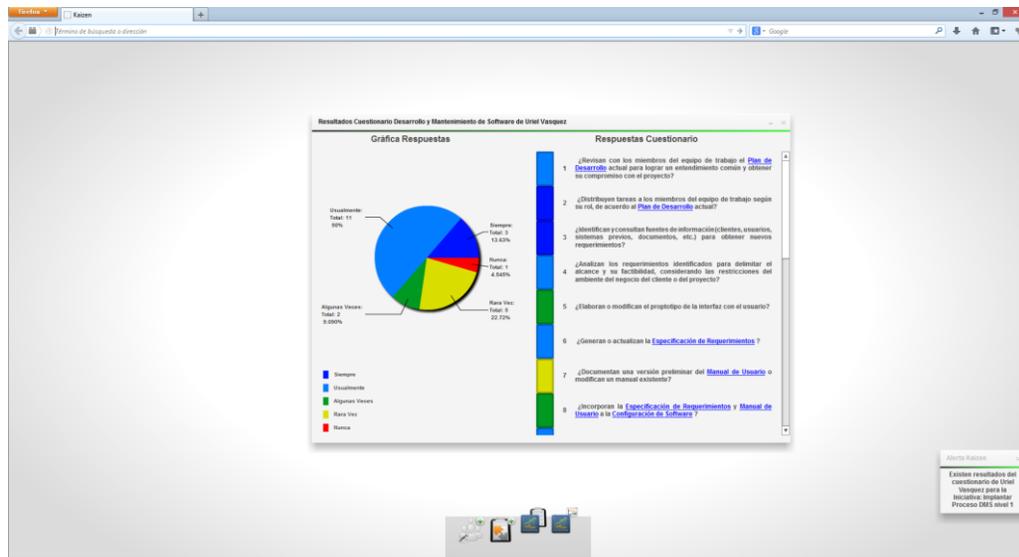


Figura 9.11. Resumen de los resultados de evaluación.

Una vez que todos los responsables del proceso han culminado sus respectivas evaluaciones, Kaizen presenta la funcionalidad del análisis de resultados y configuración para la agregación de actividades. En esta funcionalidad se realiza el análisis de la cobertura de actividades presentado en el Capítulo 3, en base a la cobertura de las actividades y la desviación estándar entre las respuestas y evaluaciones de cada líder de proyecto. De la misma forma se clasifican las actividades en puntos fuertes (color verde), puntos a explorar con mayor profundidad (color amarillo) y puntos débiles (color rojo). De las cuales el encargado de la mejora selecciona aquellas actividades a agregar en la planificación del proyecto de mejora (véase Figura 9.12).

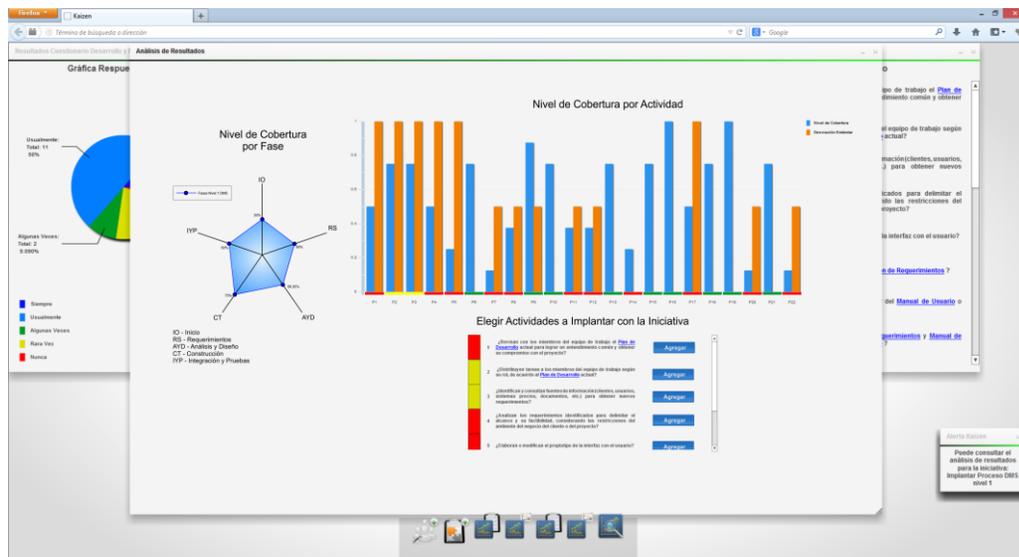


Figura 9.12. Análisis de resultados y configuración de la planificación.

9.4. Definición de infraestructura y planificación para la mejora

Cuando han sido seleccionadas las actividades que se tomarán en cuenta en la planificación de la iniciativa, Kaizen ofrece la funcionalidad de generar el plan de actividades en base a los resultados obtenidos. La planificación se divide en cada una de las fases que recomienda MoProSoft, cada nodo representa una actividad y presenta información como el nombre de la

actividad, número de actividad, tiempo estimado para desarrollar la actividad y fecha de inicio y fin (véase Figura 9.13).

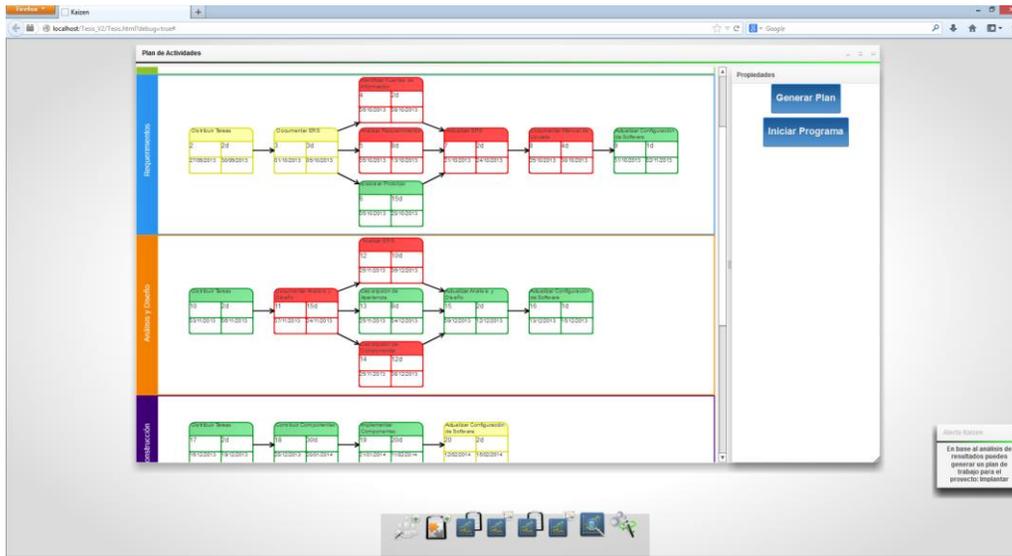


Figura 9.13. Planificación de la iniciativa de mejora.

Como lo muestra la figura, cada actividad se identifica por un color, el color verde indica que son actividades que ya se encuentran implantadas en el proceso actual de la empresa, sin embargo aparecen en la planeación para que la organización tenga presente que no debe de dejar de desarrollarlas; el amarillo corresponde a las actividades que presentaron inconsistencias en la etapa de evaluación, en el caso de ejemplo representa a aquellas actividades en las cuales los responsables discreparon claramente en las evaluaciones; por último, el rojo representa aquellas actividades de las cuales no hay ningún tipo de evidencia que se realicen en la empresa. La herramienta ofrece la información detallada de cada actividad como los responsables de la actividad, descripción y productos de entrada y de salida, para consultarla solamente basta un clic en cada elemento de la planificación para consultar dicha información (véase Figura 9.14).

Formulario de Planificación de Actividades	Formulario de Planificación de Actividades	Formulario de Planificación de Actividades
Nombre Actividad: Análisis ERS	Nombre Actividad: Documentar ERS	Nombre Actividad: Construir Componentes
ID Actividad: 12 Tiempo Estimado: 10d	ID Actividad: 3 Tiempo Estimado: 8d	ID Actividad: 18 Tiempo Estimado: 30d
Responsable: Uriel Vasquez, Enrique Lopez, Pedro Muñozcano	Responsable: Uriel Vasquez, Enrique Lopez	Responsable: Luis Velasco, Jose Miguel Martinez
Comienzo: 25/11/2013 Final: 06/12/2013	Comienzo: 01/10/2013 Final: 05/10/2013	Comienzo: 20/12/2013 Final: 20/01/2014
Entradas: Especificación de Requerimientos	Entradas: Fuentes de Información	Entradas: Análisis y Diseño
Descripción: Analizar la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos.	Descripción: Documentar o modificar la Especificación de Requerimientos.	Descripción: Construir o modificar el(los) Componente(s) de software
Salidas: Análisis y Diseño	Salidas: Especificación de Requerimientos	Salidas: Componentes Software

Figura 9.14. Información detallada para cada tipo de actividad.

El responsable es el encargado de dar inicio a las actividades especificadas en el plan. Una vez iniciadas, Kaizen notificará a los participantes sobre las actividades a su cargo, así como la información necesaria para realizar las actividades.

9.5. Implantación de la mejora

Cuando un miembro del equipo de trabajo inicia sesión en Kaizen, el sistema automáticamente envía una notificación en donde entera al empleado que se le han asignado actividades dentro del proyecto de mejora. La herramienta muestra la planificación de actividades para el miembro del equipo, tomando en cuenta solamente las actividades que se le han asignado o en las que colabora. Una vez que el miembro del equipo haya culminado la actividad, registra el evento en Kaizen, con lo cual se envían notificaciones a los interesados en la culminación de dicha práctica para continuar con la implantación (véase Figura 9.15). Cuando una actividad es marcada como terminada, en la planificación cambia de color para denotar que la práctica ha sido terminada.

Otra característica importante dentro de la parte de implantación son las notificaciones, generadas automáticamente por Kaizen en caso de que la finalización de una actividad se encuentre próxima a caducar, haya caducado o no se pueda iniciar porque una actividad de la que depende no se ha terminado (véase Figura 9.16).

Por último, Kaizen ofrece una funcionalidad para mandar mensajes a los miembros del equipo de trabajo con el fin de intercambiar información acerca del estado de las actividades, comentar cuestiones particulares en el desarrollo de las actividades o hacer del conocimiento general cuestiones problemáticas con el desarrollo de las prácticas. Los mensajes pueden enviarse a todo el grupo de trabajo o miembros específicos como se muestra en la Figura 9.17. La funcionalidad tiene como fin brindar un marco de comunicación que fomente el trabajo colaborativo entre los diferentes roles inmiscuidos en el programa de mejora.

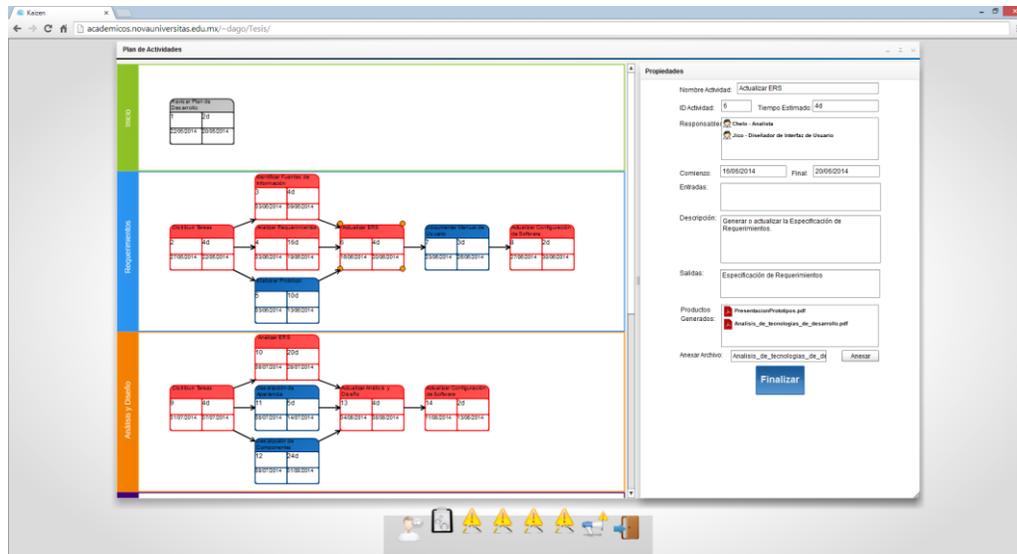


Figura 9.15. Asignación de actividades para los miembros de equipo.

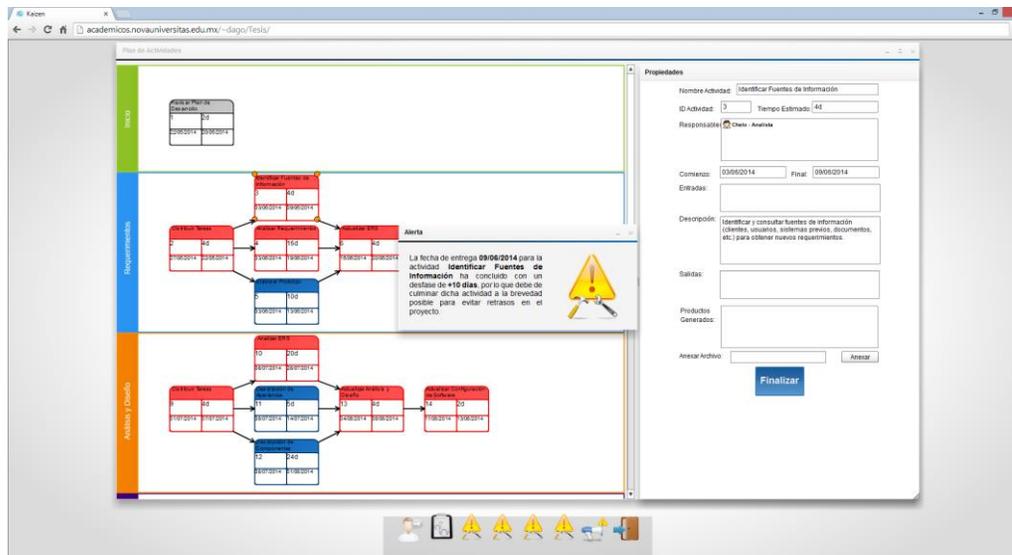


Figura 9.16. Notificación de caducidad de fecha de entrega.

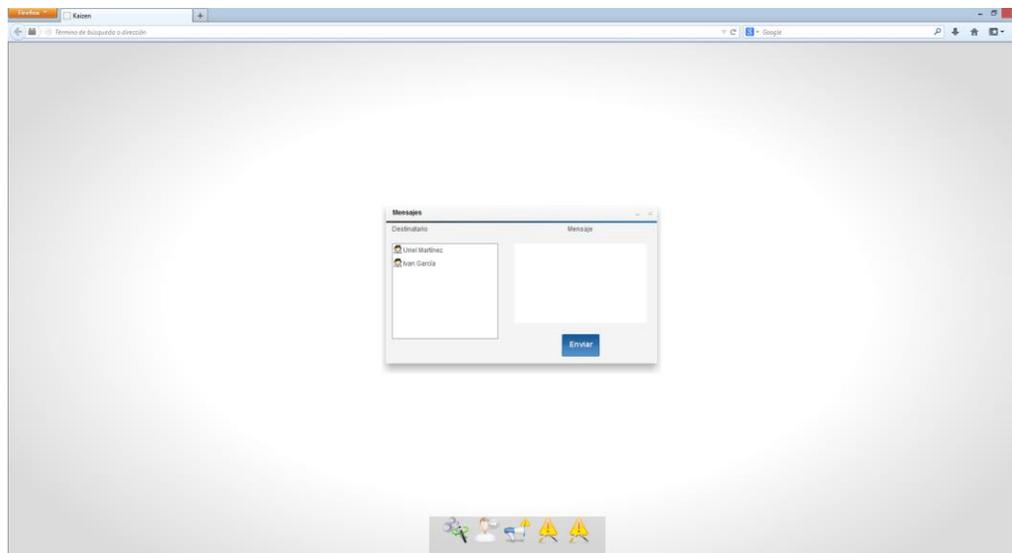


Figura 9.17. Envío de mensajes

10. Anexo E. – Actas de Publicaciones

Implementing the Modeling-Based Approach for Supporting the Software Process Assessment in SPI Initiatives Inside a Small Software Company

I. Garcia, C. Pacheco, D. Cruz, and J.A. Calvo-Manzano

Abstract. Software Process Improvement (SPI) has become more and more important during the past ten years, since competition is increasingly determined by the proportion of software products and services. Over the years, many different SPI approaches have been developed; most of them are either best-practice-oriented approaches or continuous improvement approaches, which require an accurate assessment process as a basis from which to begin the improvement. Usually, almost all research is focused on a questionnaire-based approach for process' assessment. However, without the organizational commitment it is too difficult obtain realistic and accurate results. The main achievement of this paper, lies in the development of an integrated mechanism for assessing software processes, using a hybrid mechanism that incorporates modeling-based assessment. This mechanism was evaluated using the EvalProSoft framework and descriptive concepts, to facilitate establishing SPI initiatives in a small Mexican software company.

1 Introduction

Software has increasingly penetrated society during the last decades. Software is not only used as an independent product by itself, but also as part of everyday systems such as kitchen appliances, washing machines, or mobile phones. However, the increasing significance of software in our daily lives leads to an increasing

I. Garcia · C. Pacheco
Technological University of the Mixtec Region
e-mail: [>{ivan,leninca}@mixteco.utm.mx}](mailto:{ivan,leninca}@mixteco.utm.mx)

D. Cruz
Cañada University
e-mail: dago@naxoloxa.unca.edu.mx

J.A. Calvo-Manzano
Technical University of Madrid
e-mail: jacalvo@fi.upm.es

Implementation of a RIA Tool for Supporting a Collaborative Initiative of Software Process Improvement

I. Garcia¹, C. Pacheco¹, and D. Cruz²

¹Postgraduate Department, Faculty of Computer Science,
Technological University of the Mixtec Region, Mexico

²Faculty of Computer Science,
Cañada University, Mexico

{Ivan, leninca}@mixteco.utm.mx, dago@naxoloxa.unca.edu.mx

Abstract. Under a Software Process Improvement (SPI) environment, all phases of a process improvement initiative involving establishing commitment, assessment or diagnosing, improvement plans generation, pilot implementation and improvements deployment, may be accomplished collaboratively by different groups inside an enterprise. Organizational, technical and process-based circumstances have an impact on process assessment and modeling practices. Based on a Model-based Collaborative Design, a strategy for collaborative process assessment and modeling is proposed. This collaborative support helps project managers in overcoming complexities and to create a common understanding of the process and products of a SPI initiative. Finally, a Rich Internet Application (RIA) is developed and applied to provide strong support for distributed project managers, to collaboratively assess and model their software process within a SPI project, the CEFoSPI (Collaborative Environment For Software Process Improvement) prototype. This tool represents a collaborative strategy to support SPI teams in handling the different phases of a typical SPI lifecycle.

Keywords: Collaborative model, software process improvement, process modeling, process assessment, rich-internet applications.

1 Introduction

Over the last two decades, the Software Engineering (SE) community has expressed special interest in SPI in an effort to increase software product quality, as well as the productivity of software development. This means that in contrast to traditional production, the focus of quality assurance has to be set during the development process and not during the production process. Also worthy of note is the number of process assessment methods that have been developed (i.e., CBA-IPI [1], SCAMPI [2], ISO/IEC 15504:2004 [3]) and a large number of SPI approaches including the technologies. Nowadays, assessment methods are used to evaluate the maturity of the software development process of a software-producing organization; building a roadmap for process improvement. In order to ensure and facilitate the implementation of the assessment findings, an accepted process description reflecting

2010 Eighth ACIS International Conference on Software Engineering Research, Management and Applications

Adopting an RIA-based tool for Supporting Assessment, Implementation and Learning in Software Process Improvement under the NMX-I-059/02-NYCE-2005 Standard in Small Software Enterprises

Garcia, I., Pacheco, C. & Cruz, D.

Postgraduate Department,
Technological University of the Mixtec Region
Huajuapán de León, Oaxaca (Mexico) <http://www.utm.mx>
{ivan, leninca, dago}@mixteco.utm.mx

Abstract—Nowadays there are models and standards which attempt to introduce quality in the enterprises' software development process with the objective to introduce high quality levels in the produced software. The NMX-I-059/02-NYCE-2005 standard (also known as MoProSoft) is focused on small and medium software enterprises, or small groups of software development within a larger organization, with the aim of promoting the standardization of an effective process in the software industry. Mexican enterprises now have a software standard that enables them to achieve a high level of quality in the software that they produce. However, the adoption of any standard is not an easy task. This paper aims to show that the development and implementation of a RIA-based tool that could support improvement initiatives, therefore strengthening the standard adoption.

Keywords—Software process improvement, MoProSoft, EvalProSoft, small and medium software enterprises

I. INTRODUCTION

Software has arisen as a fundamental pillar in the evolution of computational products and services. In the last two decades, software has changed from a "specific problem solution" to an autonomous industry [11]. However, this change keeps the old same problems since the "software crisis" in 1969. Nowadays, the quantity and quality demands dominate the market. According to the last report of Standish Group Inc. [38]:

- Software is (almost) always delivered out of the initial planning,
- Software is more expensive than the original cost,
- Software has a different functionality.

Pressman says that: "*the majority of software crisis causes have their origins in myths and theories that arose in the early years of Software Engineering*" This origin makes the myths more dangerous; but the truth is that they do not look like myths any more. A recent study [39] establishes that software does not accomplish the original requirements because: 45% of software exceeds the cost, 63% of software exceeds the planned schedule, and software fulfills the 67% of the required functionality. Software industry has tried to increase productivity and quality adopting alternative methodologies and technologies, but it has been recognized that the main problem is related to the incapacity to manage the software process [27] [17]. In this way, enterprises have

been changed from technological-based solutions to process-based solutions. From the beginning of the 90's, industry and researchers interested in Software Engineering have been expressing special interest in Software Process Improvement (SPI) [21]. An indicator of this is the increasing number of international initiatives related to SPI, such as CMMI-DEV v1.2 [35], ISO/IEC 15504:2004 [14], SPICE [13], and ISO/IEC 12207:2004 [15].

In addition, many methods for evaluating improvements in organizations, such as SCAMPI [19], ISO/IEC 15504:2004 and CBA-IPi [4], and improvement models such as IDEAL [18] have been developed. This interest in software improvement in large enterprises is now being extended to small enterprises. However, the problem is the high implementation cost, independent of the size of the company [5]. Because models have been developed for large enterprises, only a few Small Software Enterprises (SE) are aware of them. In Mexico, in April 2007 there were almost 1,500 SE that accounted for 99.87 % of all companies in this category. Due to this, the Software Industry Process Model (MoProSoft) [23] and its process-assessment method (EvalProSoft) were defined [24]. Thus, Mexican SE have gained impulse to improve their software processes as a strategy to assure the quality of their software products [6]. However, besides the certification desire, the SE has the problem to adopt quickly and efficiently the standard NMX-I-059-NYCE-2005 without loss resources and time to develop new projects.

This paper presents the experiences on adopting a RIA-based tool as support to facilitate the certification process by four small software companies located in the states of Tlaxcala, Puebla and Oaxaca. The document is structured as follows: Section 2 is an introduction to the MoProSoft model and Section 3 discusses related work. Section 4 describes the architecture and development of RIA-based tool. Section 5 describes the methodology used and presents the characteristics of the case study. Section 6 describes the initial scenario prior to the tool support and presents results and lessons learned. Finally, conclusions are shown in Section 7.

II. THE MOPROSOFT MODEL

In 2006 the Mexican government, through the National Plan for Development, established the objectives to increase and extend the country's competitiveness using the

11. Referencias Bibliográficas

- [Adams et al., 2004] Adams, R., Eslinger, S., Owens, K. & Rich, M. A. “Software acquisition best practices – 2004 edition” *Proc. of the 3rd OSD Conference on the Acquisition of Software-Intensive Systems*, The Aerospace Corporation, pp. 45-89, 2004.
- [Alcaraz y Maroto, 2001] Alcaraz, J. & Maroto, C. “A robust genetic algorithm for resource allocation in project scheduling” *Annals of Operations Research*, 102(1-4): 83-109, 2001.
- [AMITI, 2005] AMITI. “*Esquema de apoyo gubernamental a la Industria Software*” Asociación Mexicana de la Industria de las Tecnologías de la Información, México. 2005.
- [Wall, 1996] Wall, M. “*A genetic algorithm for resource-constrained scheduling*” PhD Thesis. Department of Mechanical Engineering, Massachusetts Institute Technology, 1996.
- [Brown, 2008] Brown, C. *The essential guide to Flex 3*. New York, NY: Apress. 2008.
- [Christensen y Thayer, 2001] Christensen, M. & Thayer, R. *The project manager’s guide to software engineering’s best practice*. Los Alamitos, CA: IEEE Computer Society, 2001.
- [Calvo-Manzano et al., 2008] Calvo-Manzano, J. A., Garzás, J., Piattini, M., Pino, F. J., Salillas, J., Sánchez, J. L. “Perfiles del ciclo de vida del software para pequeñas empresas: los informes técnicos ISO/IEC 29110” *REICIS Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(2): 96-108, 2008.
- [CMMI, 2006] CMMI Product Team. “*CMMI® for Development, Version 1.2. Improving processes for developing better products and services*” CMU/SEI-2006-TR-008, ESC-TR-2006-008. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, August 2006.
- [CMMI, 2010] CMMI Product Team. “*CMMI® for Development, Version 1.3. Improving processes for developing better products and services*” CMU/SEI-2010-TR-033, ESC-TR-2010-033. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, November 2010.
- [Chandler y Coartada, 2003] Chandler, A. & Coartada, J. W. *Una nación transformada por la información*. México, DF: Oxford University Press. 2003.
- [Cuevas et al., 2002] Cuevas, G., De Amescua, A., San Feliu, T., Calvo-Manzano, J., Arcilla, M., García, M. & Cerrada, J. *Gestión del proceso software*. Madrid, España: Editorial Universitaria Ramón Areces, 2002.
- [Cuevas et al., 2005] Cuevas, G., Calvo-Manzano, J. A., García, I., Serrano, A. and San Feliu, T. *AFIM: Modelo de mejora enfocado a la acción*. Madrid, España: Cátedra para la Mejora al Proceso Software en el Espacio Iberoamericano – Universidad Politécnica de Madrid. 2005.
- [Dart, 2000] Dart, S. *Configuration management: the missing link in web engineering*. Boston, MA: Artech House Publishers. 2000.
- [Davis y Phillips, 2008] Davis, M. & Phillips, J. *Flex 3: A beginner’s guide*. Seattle, WA: McGraw-Hill Companies. 2008.
- [Deming, 1982] Deming, W. E. *Out of the crisis*. Cambridge, MA: MIT Center for Advanced Engineering Study. 1982.

- [**Bonanomi, 2012**] Bonanomi, E. “Análisis comparativo de la industria de software y servicios informáticos de la Argentina, Brasil y México” Reporte técnico de ESEADE (Escuela Superior de Economía y Administración de Empresas), 2012.
- [**Dolog y Stage, 2007**] Dolog, P. & Stage, J. “Designing interaction spaces for rich internet applications with UML” *Proc. of the 7th International Conference on Web Engineering (ICWE’07)*, Springer-Verlag, pp. 358-363, 2007.
- [**Fairley, 2009**] Fairley, R. *Managing and leading software projects*. Hoboken, NJ: John Wiley & Sons, Inc. 2009.
- [**Farré y Messeguer, 2005**] Farré, X. & Messeguer, R. *Rich Internet Applications*. Universidad Politécnica de Cataluña. Julio, 2005.
- [**Flores et al., 2008**] Flores, B., Astorga, M., Olguín, J. & Andrade, M. “Experiences on the implementation of MoProSoft and assessment processes under the NMX-I-059/02-NYCE-2005 standard in a small software development enterprise” *Proc. of the 2008 Mexican International Conference on Computer Science*, IEEE Computer Society, pp. 323-328, 2008.
- [**Forsberg et al., 2005**] Forsberg, K., Mooz, H. & Cotterman, H. *Visualizing project management: Models and frameworks for mastering complex systems*. Hoboken, NJ: John Wiley & Sons, Inc. 2005.
- [**Fowler et al., 1999**] Fowler, P., Middlecoat, B. & Yo, S. “Lessons learned collaborating on a process for SPI at Xerox” CMU/SEI-99-TR-006, ESC-TR-99-06. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, December 1999.
- [**Garcia et al., 2010**] Garcia, I., Pacheco, C., and Cruz, D. “Adopting an RIA-based tool for supporting assessment, implementation and learning in software process improvement under the NMX-I-059/02-NYCE-2005 standard in small software enterprises” *Proc. of the Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010)*, IEEE Computer Society, pp.29-35, 2010.
- [**Garcia et al., 2010a**] Garcia, I., Pacheco, C. & Calvo-Manzano, J. “Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting” *IET Software*, 4(4): 237-251, 2010.
- [**González, 2006**] González, D. “Estudio exploratorio de los factores críticos de éxito de la industria mexicana del software y su relación con la orientación estratégica de negocio” Informe de trabajo de investigación, Doctorado en Integración de las Tecnologías de la Información en las Organizaciones. Universidad Politécnica de Valencia, España. 2006.
- [**Goralski y Leon, 2008**] Goralski, G. & Leon, L. *Foundation Flex for designers*. New York, NY: Apress. 2008.
- [**Gutiérrez, 2008**] Gutiérrez de Mesa, J. A. *Planificación y gestión de proyectos informáticos*. Madrid, España: Universidad de Alcalá de Henares. 2008.
- [**Hall et al., 2002**] Hall, T., Rainer, A. & Badoo, N. “Implementing software process improvement: An empirical study” *Software Process: Improvement and Practice*, 7(1): 3-15, 2002.
- [**Hashim y Keshlaf, 2009**] Hashim, K. & Keshlaf, A. A. “An approach to sharing solutions to software project management problems” *Proc. of the International Conference on Information Management and Engineering (ICIME 2009)*, IEEE Computer Society, pp. 694-697, 2009.
- [**Huang y Han, 2008**] Huang, S. J. & Han, W. M. “Exploring the relationship between software project duration and risk exposure: A cluster analysis” *Information & Management*, 45(3): 175-182, 2008.
- [**Humphrey, 1989**] Humphrey, W. *Managing the software process*. Reading, MA: Addison-Wesley Publishing Company. 1989.

- [Humphrey, 1992] Humphrey, W. "Introduction to software process improvement" CMU/SEI-92-TR-007, ESC-TR-92-007. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 1992.
- [Humphrey, 2005] Humphrey, W. "The Software Process: Global Goals" In: Li, M., Boehm, B. & Osterweil, L. J. (Eds.), *Unifying the Software Process Spectrum*. Springer-Verlag, pp. 35-42, 2005.
- [Hurtado et al., 2008] Hurtado, J. A., Pino, F. J., Vidal, J. C., Pardo, C. & Fernández, L. E. "Agile SPI: Software process agile improvement— A colombian approach to software process improvement in small software organizations" In: Oktaba, H., & Piattini, M. (Eds.), *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, IGI Global, pp. 177-192, 2008.
- [IEEE, 1998] Institute of Electrical and Electronic Engineers. *IEEE Std. 830-1998: Recommended practice for software requirements specifications*. New York, NY: IEEE Computer Society, 1998.
- [IIE, 2003] Instituto de Investigaciones Eléctricas. "Moprosoft: el nuevo modelo que impondrá una norma mexicana para la calidad de la industria del software" Boletín IIE, pp. 81-83, Julio-Septiembre del 2003.
- [ISO, 2000] International Organization for Standardization. "ISO 9001:2000: Quality management systems - Requirements" Geneva, 2000.
- [ISO, 2002] International Organization for Standardization. "ISO/IEC 12207:1995. Information Technology Software Life Cycle Processes" Geneva, 2002.
- [ISO, 2004] International Organization for Standardization. "ISO/IEC 15504-2:2003/Cor.1: 2004 (E): Information Technology – Process Assessment – Part 2: Performing an Assessment" Geneva 2004.
- [ISO, 2011] International Organization for Standardization. "ISO/IEC 29110:2011- Software engineering - Lifecycle profiles for Very Small Entities (VSEs)" Geneva, 2011.
- [ISO, 2011a] International Organization for Standardization. "Technical Report: ISO/IEC TR 29110-5-1-2" Geneva, 2011.
- [Janiszewski, 2004] Janiszewski, S. "Six sigma and software process improvement" *Proc. of the 2003 SPI Symposium*, Software Engineering Institute, pp. 125-137, 2004.
- [Johnson, 2006] Johnson, J. *My life is failure: 100 things you should know to be a successful project leader*. West Yarmouth, MA: Standish Group International. 2006.
- [Kendall y Kendall, 2011] Kendall, K. & Kendall, J. *Análisis y diseño de sistemas*. México, DF: Pearson Educación. 2011.
- [Koch y Kraus, 2002] Koch, N. & Kraus, A. "The expressive power of UML-based web engineering" *Proc. of the 2nd International Workshop on Web-Oriented Software Technology*, CYTED, pp. 79-86, 2002.
- [Koch et al., 2008] Koch, N., Knapp, A., Zhang, G. & Baumeister, H. "UML-based web engineering: An approach based on standards" In: Rossi, G., Pastor, O., Schwabe, D. & Olsina, L (Eds.), *Web Engineering: Modeling and Implementing Web Applications*, Springer-Verlag, pp. 57-191, 2008.
- [Koch et al., 2009] Koch, N., Pigerl, M., Zhang, G. & Morozova, T. "Patterns for the model-based development of RIAs" In: Gaedke, M., Grossniklaus, M. & Díaz, O. (Eds.), *Web Engineering*, Springer-Verlag, pp. 283-291, 2009.
- [Kroiß y Koch, 2008] Kroiß, C. & Koch, N. "UWE metamodel and profile: User guide and reference" Technical Report 0802 v1.0, Institute for Informatics Ludwig-Maximilians-Universität, München, Germany. February, 2008.
- [Lee y Kim, 1996] Lee, J. & Kim, D. "Search heuristics for resource constrained project scheduling" *Journal of the Operation Research Society*, 47(5): 678-689, 1996.

- [Leon y Balakrishnan, 1995] Leon, V. J. & Balakrishnan, R. "Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling" *OR Spektrum*, 17(2-3): 370-379, 1995.
- [MAP, 2001] Ministerio de Administraciones Públicas. "Métrica versión 3: Metodología de planificación, desarrollo y mantenimiento de sistemas de información" Madrid, España: Ministerio de Administraciones Públicas del Gobierno de España. 2001.
- [Machado et al., 2009] Machado, L. Filho, O. & Ribeiro, J. "UWE-R: An extension to a web engineering methodology for rich internet applications" *WSEAS Transactions on Information Science and Applications*, 6(4): 601-610, 2009.
- [McFeeley, 1996] McFeeley, R. *IDEAL: A User's Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1996.
- [Mendoza et al., 2009] Cruz, R., Morales, M., Morgado, M., Oktaba, H., Ibarguengoitia, G., Pino, F. J. & Piattini, M. "Supporting the software process improvement in very small entities through e-learning: The HEPALE! project" *Proc. of the 2009 Mexican International Conference on Computer Science*, pp.221-231, 2009.
- [Meliá et al., 2008] Meliá, S., Gómez, J., Pérez, S. & Díaz, O. "A model-driven development for GWT-based rich internet applications with OOH4RIA" *Proc. of the Eighth International Conference on Web Engineering (ICWE'08)*, IEEE Computer Society, pp. 13-23, 2008.
- [Mochi, 2007] Mochi, P. "Oportunidades y desafíos de la industria software en México" In: Bastos, P. & Silveira, F. (Eds.), *Oportunidades y Desafíos de la Industria Software en América Latina*, CEPAL, pp. 112-125, 2007.
- [Mori y Tseng, 1998] Mori, M. & Tseng, C. "A genetic algorithm for multi-mode resource constrained project scheduling problem" *European Journal of Operation Research*, 100(1): 134-141, 1998.
- [Mowery, 1996] Mowery, D. *International computer software industry*. New York, NY: Oxford University Press. 1996.
- [Muñoz et al., 2012] Muñoz, M., Mejia, J., Calvo-Manzano, J. A., Cuevas, G., San Feliu, T. and De Amescua, A. "Expected Requirements in Support Tools for Software Process Improvement in SMEs" *IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, pp. 135-140, 2012.
- [Murugesan et al., 1999] Murugesan, S., Deshpande, Y., Hansen, S. & Ginlge, A. "Web engineering: A new discipline for development of web-based systems" *Proc. of the 1st ICSE Workshop on Web Engineering (ICSE'99)*, IEEE Computer Society, pp. 175-186, 1999.
- [NYCE, 2005] Normalización y Certificación Electrónica, A. C. "Tecnología de la Información – Software – Modelos de Proceso y Evaluación para Desarrollo y Mantenimiento de Software – Parte 02: Requisitos de Procesos (MoProSoft)" NMX-I-059/02-NYCE-2005. México, DF: NYCE. 2005.
- [NYCE, 2005a] Normalización y Certificación Electrónica, A. C. "Tecnología de la Información – Software – Modelos de Proceso y Evaluación para Desarrollo y Mantenimiento de Software – Parte 04: Directrices para la evaluación de procesos (EvalProSoft)" NMX-I-059/04-NYCE-2005. México, DF: NYCE. 2005.
- [Ocaña y Rossainz, 2004] Ocaña, J. & Rossainz, M. "Introducción a la Ingeniería Web basada en UML" Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación. 2004.
- [Oktaba, 2005] Oktaba, H. "Historia de una norma: MoProSoft y sus primeros pasos" *Revista Software Gurú*, 1(8): 6-11, 2005.
- [Oktaba, 2005a] Oktaba, H. "Modelo de procesos para la industria software: MoProSoft". Versión 1.3. Agosto, 2005.

- [Oktaba, 2005b] Oktaba, H. “Método de evaluación de procesos para la industria software: *EvalProSoft*” Versión 1.3. Agosto, 2005.
- [Oktaba, 2006] 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, pp. 93-101, 2006.
- [Oktaba et al., 2007] Oktaba, H., Garcia, F., Piattini, M., Ruiz, F., Pino, F. J. & Alquicira, C. “Software process improvement: The Competisoft project” *IEEE Computer*, 40(10): 21-28, 2007.
- [Oktaba, 2010] Oktaba, H. “Pasado, presente y futuro de MoProSoft” *Revista Software Gurú*, 4(1): 25-32, 2010.
- [Özdamar, 2000] Özdamar, L. “A genetic algorithm approach to a general category project scheduling problem” *IEEE Transactions on Systems, Man and Cybernetics: Applications to Review*, 29(1): 193-208, 2000.
- [Pan et al., 2007] Pan, Z., Park, H., Baik, J. & Choi, H. “A six sigma framework for software process improvements and its implementation” *Proc. of the 14th Asia-Pacific Software Engineering Conference (APSEC 2007)*, IEEE Computer Society, pp. 446-453, 2007.
- [Paulk, 1995] Paulk, M. C. *The capability maturity model guidelines for improving the software process*. Boston, MA: Addison-Wesley. 1995.
- [Pérez y Messeguer, 2008] Pérez, M. & Messeguer, R. “Evaluación y prueba de aplicaciones RIA con AJAX”. Universidad Politécnica de Cataluña. Abril, 2008.
- [Persse, 2006] Persse, J. R. *Process improvement essentials: CMMI, Six Sigma, and ISO 9001*. Sebastopol, CA: O’Reilly Media Inc. 2006.
- [Pino, 2006] Pino, F. J., García, F. & Piattini, M. “Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas” *Revista Española de Innovación, Calidad e Ingeniería del Software*, 2(1): 6-23, 2006.
- [PMI, 2004] Project Management Institute. *A guide to the Project Management Body of Knowledge (PMBok Guide)*. Los Alamitos, CA: Project Management Institute. 2004.
- [Piattini y Garzás, 2007] Piattini, M. & Garzás, J. *Fábricas de software: Experiencias, tecnologías y organización*. Madrid, España: Editorial Ra-Ma. 2007.
- [Preciado et al., 2005] Preciado, J. C., Linaje, M., Sanchez, F. & Comai, S. “Necessity of methodologies to model rich internet applications” *Proc. of the Seventh IEEE International Symposium on Web Site Evolution (WSE 2005)*, IEEE Computer Society, pp. 7-13, 2005.
- [Preciado et al., 2008] Preciado, J. C., Linaje, M., Morales, R., Sánchez-Figueroa, F., Zhang, G., Kroiss, C. & Koch, N. “Designing rich internet applications combining UWE and RUX-Method” *Proc. of the Eighth International Conference on Web Engineering (ICWE 2008)*, IEEE Computer Society, pp. 148-154, 2008.
- [Pressman, 2009] Pressman, R. S. *Software engineering: A practitioner’s approach*. 7th Edition. New York, NY: McGraw-Hill. 2009.
- [Ribaud et al., 2010] Ribaud, V., Saliou, P. & Laporte, C. Y. “Experience management for very small entities: Improving the copy-paste model” *Proc. of the Fifth International Conference on Software Engineering Advances (ICSEA 2010)*, IEEE Computer Society, pp. 311-318, 2010.
- [Sampedro, 2011] Sampedro, J. L. *Conocimiento y empresa: La industria del software en México*. México, DF: Editorial Plaza y Valdés - UAM Cuajimalpa, 2011.
- [Santos et al., 2007] Santos, G., Montoni, M., Vasconcellos, J., Figueiredo, S., Cabral, R., Cerdeiral, C., Katsurayama, A., Lupo, P., Zanetti, D. & Rocha, A. “Implementing Software Process Improvement Initiatives in Small and Medium-Size Enterprises in Brazil” *Proc. of the 6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, IEEE Computer Society, pp. 187-198, 2007.

- [Scott, 2007] Scott, B. “*RIA patterns: Best practices for common patterns of rich interaction*” UIE Web App Summit, 2007.
- [SE, 2001] Secretaría de Economía del Gobierno de México. “*Plan de desarrollo nacional 2001-2006*”. México, DF: Secretaría de Economía, 2001.
- [SE, 2008] Secretaría de Economía del Gobierno de México. “*PROSOFT 2.0: Programa de desarrollo del sector de servicios de tecnologías de información*” México, DF: Secretaría de Economía, 2008.
- [Serrano et al., 2006] Serrano, M., Montes de Oca, C. & Cedillo, K. “An experience on Implementing the CMMI in a Small Organization Using the Team Software Process” *Proc. of the First International Research Workshop for Process Improvement in Small Settings*, Software Engineering Institute, Carnegie Mellon University, pp. 81-92, 2006.
- [Sommerville, 2010] Sommerville, I. *Software engineering*. 9th Edition. Seattle, WA: Addison Wesley, 2010.
- [Standish, 2004] The Standish Group International. “*Chaos Extreme*”. 2004.
- [Standish, 2009] The Standish Group International. “*Chaos Report*”. 2009.
- [Tapper et al., 2009] Tapper, J., Labriola, M., Boles, M. & Talbot, J. *Adobe Flex 3: Training from the source*. Berkeley, CA: Peachpit. 2009.
- [Toffetti et al., 2007] Toffetti, G., Comai, S., Bozzon, A. & Fraternali, P. “Modeling distributed events in data-intensive rich internet applications” *Proc. of the 8th International Conference on Web Information Systems Engineering (WISE'07)*, Springer-Verlag, pp. 593-602, 2007.
- [Varkoi, 2010] Varkoi, T. “Process assessment in very small entities - An ISO/IEC 29110 based method” *Proc. of the Seventh International Conference on the Quality of Information and Communications Technology (QUATIC)*, IEEE Computer Society, pp. 436-440, 2010.
- [Walker, 2003] Walker, E. “Implementing Best Practices in the Joint Battlespace Infosphere (JBI) program at AFRL” *Proc. of the Conference on the Acquisition of Software-Intensive System*, Software Engineering Institute, Carnegie Mellon University, pp. 28-30, 2003.
- [Wang et al., 2008] Wang, E. T. G., Ju, P. H., Jiang, J. J. & Klein, G. “The effects of change control and management review on software flexibility and project performance” *Information & Management*, 45(7): 438-443, 2008.
- [Wohlin et al., 2012] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslén, A. *Experimentation in software engineering*. Boston, MA: Kluwer Academic Publishers. 2012.
- [Yin, 2009] Yin, R. K. *Case study research. Design and methods*. 4th Edition. Thousand Oaks: CA, Sage Publications. 2009.

11.1. Sitios de Internet

[URL-1] <http://pnd.fox.presidencia.gob.mx/>

Plan Nacional de Desarrollo 2001-2006 (Último acceso: Diciembre, 2010).

[URL-2] <http://www.prosoft.economia.gob.mx/>

Programa para el Desarrollo de la Industria Software (Último acceso: Diciembre 2010).

[URL-3] <http://www.software.net.mx/>

Estudio sobre la industria de Software (Último acceso: Diciembre 2010).

[URL-4] <http://www.nyce.org.mx/>

Asociación de Normalización y Certificación Electrónica A.C. (Último acceso: Febrero 2013).

[URL-5] <http://www.cytcd.org/>

Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (Último acceso: Febrero 2011).

[URL-6] <http://www.iso.org/>

- International Organization for Standardization (Último acceso: Marzo 2011).
[URL-7] <http://www.sencha.com/products/extjs/>
- JavaScript Framework for Rich Apps in Every Browser (Último acceso: Marzo 2011).
[URL-8] <http://www.adobe.com/es/products/flex/>
- Adobe Flex (Último acceso: Marzo 2011).
[URL-9] <http://silverlight.net/>
- Microsoft Silverlight (Último acceso: Marzo 2011).
[URL-10] <http://www.adobe.com/devnet/flashplatform/services/collaboration.html>
- LiveCycle Collaboration Service (Último acceso: Abril 2011).
[URL-11] <http://www.methodpark.com/>
- Stages Process Management Suite (Último acceso: Mayo 2012).
[URL-12] <http://www.cmm-quest.com/>
- CMM- Quest: Appraisal Tool for CMMI-DEV 1.3 (Último acceso: Mayo 2012).
[URL-13] <http://www.isd-inc.com/tools.appraisalWizard/>
- Appraisal Wizard (Último acceso: Mayo 2012).
[URL-14] <http://www.rommanasoftware.com/>
- Rommana Tool–The Premier in Integrated Application Lifecycle Management (Último acceso: Mayo 2012).
[URL-15] <http://www.microsoft.com/project/es/es/>
- Microsoft Project Professional 2010 (Último acceso: Mayo 2012).
[URL-16] <http://www.pragmasystems.com/products>
- processMax – The process solution (Último acceso: Junio 2012).
[URL-17] <http://www.igrafx.com/es>