



UNIVERSIDAD DEL PAPALOAPAN

Campus Loma Bonita

INGENIERÍA EN COMPUTACIÓN

“MARCO DE TRABAJO PARA ESTABLECER INICIATIVAS DE
MEJORA DEL PROCESO DE SOFTWARE CON MOPROSOFT®:
CASO DE ESTUDIO EN UNA PEQUEÑA EMPRESA
OAXAQUEÑA”

TESIS

PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA

NAHIR CARRILLO LÓPEZ

ASESOR DE TESIS

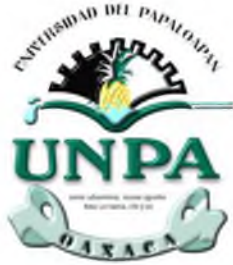
M.C. EDUARDO ORTIZ HERNÁNDEZ

CO-ASESOR DE TESIS

DR. IVÁN ANTONIO GARCÍA PACHECO

LOMA BONITA, OAXACA.

JULIO DE 2016



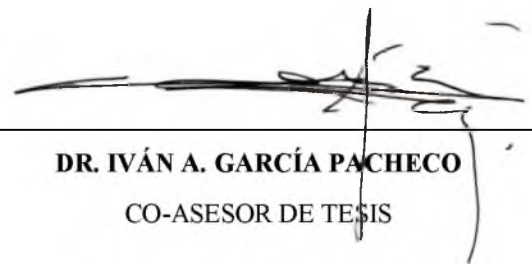
UNIVERSIDAD DEL PAPALOAPAN

INGENIERÍA EN COMPUTACIÓN

LA PRESENTE TESIS TITULADA “**MARCO DE TRABAJO PARA ESTABLECER INICIATIVAS DE MEJORA DEL PROCESO DE SOFTWARE CON MOPROSOFT®: CASO DE ESTUDIO EN UNA PEQUEÑA EMPRESA OAXAQUEÑA**” PRESENTADA POR LA SUSTENTANTE DE LICENCIATURA **C. NAHIR CARRILLO LÓPEZ** BAJO LA DIRECCIÓN DEL **M.C. EDUARDO ORTIZ HERNÁNDEZ** Y DEL **DR. IVÁN A. GARCÍA PACHECO** HA SIDO REVISADA Y ACEPTADA POR EL COMITÉ EXAMINADOR PARA SER DEFENDIDA EN EL EXAMEN PROFESIONAL Y OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN.

M. C. EDUARDO ORTIZ HERNÁNDEZ

ASESOR



DR. IVÁN A. GARCÍA PACHECO

CO-ASESOR DE TESIS

**M.C. ARIEL LÓPEZ
RODRÍGUEZ**

PRESIDENTE

**M.C. JOSÉ ANTONIO ROSALES
BARRALES**

SECRETARIO

M.C. ISAAC MACHORRO CANO

VOCAL

Índice

Índice	xi
Lista de tablas	xv
Lista de figuras	xvii
Resumen	xix
1. Introducción.....	1
1.1. Importancia del problema y necesidad de la solución.....	4
1.2. Hipótesis de la tesis	6
1.3. Objetivos de la tesis.....	6
1.3.1. Objetivo general	6
1.3.2. Objetivos específicos.....	7
1.4. Delimitaciones de la tesis	7
1.5. Limitaciones de la tesis	7
1.6. Solución propuesta	7
1.7. Estructura de la tesis	10
2. Marco Teórico	11
2.1. Fundamentos de la Mejora del Proceso de Software.....	15
2.1.1. Modelos de proceso	15
2.1.2. Modelos de mejora	16
2.1.3. Modelos de evaluación	17
2.2. Modelo de proceso para la industria mexicana de software: MoProSoft®.....	18
2.3. Modelo de evaluación para la industria mexicana de software: EvalProSoft®	23
2.4. Modelo de mejora para implantar MoProSoft®.....	28
2.5. Estado del arte	30

2.5.1. Manejador de documentos de MoProSoft® (MDM).....	30
2.5.1.1. Objetivo	30
2.5.1.2. Descripción de la propuesta.....	30
2.5.1.3. Resultados alcanzados	32
2.5.2. Implementación de MoProSoft® en una empresa dedicada al desarrollo de software	34
2.5.2.1. Objetivo	34
2.5.2.2. Descripción de la propuesta.....	34
2.5.2.3. Resultados alcanzados	36
2.5.3. Caso de estudio de mejora continua de procesos de software en una PYME mexicana	40
2.5.3.1. Descripción de la propuesta.....	40
2.5.3.2. Resultados alcanzados	44
2.5.4. Propuesta de una guía para interpretar los procesos de MoProSoft® con métodos ágiles.....	45
2.5.4.1. Objetivo	45
2.5.4.2. Descripción de la propuesta.....	45
2.5.4.3. Resultados alcanzados	50
2.5.5. Diseño e implementación de la guía de autoevaluación sobre la categoría de operación del Modelo de Procesos para la industria de software (MoProSoft®).....	51
2.5.5.1. Objetivo	51
2.5.5.2. Descripción de la propuesta.....	51
2.5.5.3. Resultados alcanzados	54
2.5.6. Modelo de calidad para la microempresa basado en MoProSoft®	55
2.5.6.1. Objetivo	55
2.5.6.2. Descripción de la propuesta.....	55
2.5.6.3. Resultados alcanzados	58
2.6. Conclusiones finales sobre el marco teórico	59
3. Marco de trabajo para guiar iniciativas de mejora con MoProSoft®.....	61
3.1. Fase Definir	62
3.2. Fase Evaluar.....	65
3.3. Fase Realizar.....	70
3.4. Fase Extender.....	74
4. Caso de estudio.....	77
4.1. Definición del caso de estudio.....	78
4.1.1. Definir el compromiso de mejora	80

4.1.2. Evaluar los procesos.....	84
4.1.3. Realizar las mejoras.....	90
4.1.4. Extender el proceso y continuar con la mejora.....	96
4.1.5. Comparación evaluación final con la línea base inicial.....	99
5. Discusión y conclusiones.....	101
Anexo A. Acrónimos.....	103
Anexo B. Plantillas definidas para las actividades de DERE.....	107
B.1. Plantilla para la Definición del Compromiso.....	109
B.2. Plantilla para el Plan de Evaluación.....	117
B.3. Plantilla para entrega de los Resultados de la Evaluación.....	127
B.4. Plantilla para la Guía del Plan de Acción.....	133
Anexo C. Cuestionarios de Evaluación.....	141
C.1. Evaluación del Proceso de Administración de Proyectos Específicos.....	142
C.2. Evaluación del Proceso de Desarrollo y Mantenimiento de Software.....	158
Anexo D. Formatos diseñados para mejorar los procesos de la empresa.....	171
D.1. Formato para Minuta de Lanzamiento (DMS A.1.1).....	173
D.2. Formato para Reporte de Actividades (DMS A.1.3).....	175
D.3. Formato para Necesidades del Sistema (DMS A.2.3).....	181
D.4. Formato para Reporte de Verificación de Necesidades (DMS A.2.4).....	189
D.5. Formato para Especificación de Requerimientos (DMS A.2.6).....	191
D.6. Formato para Reporte de Verificación de Especificación (DMS A.2.11).....	203
D.7. Formato para Reporte de Validación de Especificación (DMS A.2.13).....	205
D.8. Formato para Plan de Pruebas de Sistema (DMS A.2.15).....	207
D.9. Formato para Reporte de Verificación de Plan de Pruebas (DMS A.2.17).....	213
D.10. Formato para Reporte de Validación de Plan de Pruebas (DMS A.2.19).....	215
D.11. Formato para Manual de Usuario (DMS A.2.20).....	217
D.12. Formato de Reporte de Verificación de Manual de Usuario (DMS A.2.21).....	221
D.13. Formato para Configuración de Software (DMS A.2.23).....	223
D.14. Formato para Registro de Rastreo (DMS A.3.2).....	225
D.15. Formato para Análisis y Diseño (DMS A.3.3).....	227
D.16. Formato para Reporte de Verificación de Análisis & Diseño (DMS A.3.9).....	235
D.17. Formato para Reporte de Verificación de Registro de Rastreo (DMS A.3.9).....	237
D.18. Formato para Reporte de Validación de Análisis y Diseño (DMS A.3.11).....	239
D.19. Formato para Plan de Pruebas de Integración (DMS A.3.13).....	241
6. Bibliografía.....	247
6.1. Sitios de Internet.....	251

Lista de tablas

Tabla 1. Clasificación de las empresas desarrolladoras de software en México. Fuente: Asociación Mexicana de la Industria de Tecnologías de Información.....	1
Tabla 2. Total de producción y nivel de participación de la industria del software en México (Sampedro, 2011).	2
Tabla 3. Definición de los procesos establecidos por MoProSoft®.....	22
Tabla 4. Resultados sobre la utilidad del Manejador de Documentos de MoProSoft® (Caballero, 2005).....	33
Tabla 5. Resultados sobre la evaluación cualitativa del Manejador de Documentos de MoProSoft® (Caballero, 2005).....	33
Tabla 6. Nomenclatura definida para el análisis entre procesos de MoProSoft® y métodos ágiles (Rueda, 2010).	46
Tabla 7. Análisis de APE y Scrum (Rueda, 2010).	46
Tabla 8. Niveles de capacidad de MoProSoft® y cobertura de Scrum (Rueda, 2010).	46
Tabla 9. Análisis de DMS, Scrum y XP (Rueda, 2010).	47
Tabla 10. Niveles de capacidad de MoProSoft® y cobertura de Scrum y XP (Rueda, 2010).	48
Tabla 11. Mapeo entre roles de MoProSoft® y Scrum (Rueda, 2010).	48
Tabla 12. Constructos y variables de estudio (Preciado, 2010).....	52
Tabla 13. Entregables generados para OPE de MoProSoft® (Rueda, 2010).	54
Tabla 14. Comparativa entre MoProSoft y el modelo propuesto (Flores, 2012).	56
Tabla 15. Clasificación de niveles de rendimiento durante la evaluación del proceso (Garcia et al., 2007).....	67
Tabla 16. Perfil de la empresa participante en el caso de estudio.	78
Tabla 17. Escala normativa de calificación definida por EvalProSoft®.....	85
Tabla 18. Porcentajes de cobertura obtenidos con la evaluación de las fases de APE.....	88
Tabla 19. Porcentajes de cobertura obtenidos con la evaluación de las fases de DMS.....	89

Tabla 20. Ejemplo de priorización de los problemas clave detectados en EmpSoft.	92
Tabla 21. Ejemplo de lecciones aprendidas para el proyecto “Sistema de Administración de Bibliotecas”.....	97
Tabla 22. Ejemplo de lecciones aprendidas para el proyecto “Sistema de Control Escolar”.....	98

Lista de figuras

Figura 1.1. Evolución del número de proyectos de software anunciados por región, 2003-2010. Fuente: CEPAL	3
Figura 1.2. Categorías de modelos relacionados con una iniciativa de mejora.....	8
Figura 1.3. Propuesta del marco de trabajo para conducir iniciativas de mejora en MiPyMEs.....	9
Figura 2.1. Tendencia de publicación de estudios realizados sobre la Mejora del Proceso de Software. Modificada de (Pino, García, & Piattini, 2008).	12
Figura 2.2. Modelos involucrados en una iniciativa de mejora.....	15
Figura 2.3. Representación gráfica de las categorías de procesos definidas en MoProSoft®	21
Figura 2.4. Relación entre los elementos de EvalProSoft®	24
Figura 2.5. Modelo de capacidades establecido por EvalProSoft®	25
Figura 2.6. Niveles de capacidad definidos por EvalProSoft®	25
Figura 2.7. Etapas del proceso de evaluación establecido por EvalProSoft®.....	26
Figura 2.8. Diagrama de secuencia de los procesos de Gestión de Negocios y Gestión de Procesos (Caballero, 2005).....	31
Figura 2.9. Diagrama de secuencia del proceso de Administración de Proyectos Específicos (Caballero, 2005).....	32
Figura 2.10. Estructura organizativa de Informática Lógica S. C. (Barradas, 2008)	35
Figura 2.11. Flujograma de productos de trabajo entre procesos (Barradas, 2008).....	36
Figura 2.12. Estructura organizacional resultante con la implementación de MoProSoft® (Barradas, 2008)	37
Figura 2.13. Diagrama definido para APE (Rueda, 2010)	49
Figura 2.14. Diagrama definido para DMS (Rueda, 2010)	50
Figura 2.15. Pantalla inicial de la herramienta (Preciado, 2010)	53
Figura 2.16. Pantalla que muestra un ejemplo de una pregunta de los cuestionarios de evaluación (Preciado, 2010)	53

Figura 2.17. Pantalla que muestra un ejemplo de resultados de la evaluación (Preciado, 2010)	54
Figura 2.18. El modelo de calidad para la microempresa basado en MoProSoft (Flores, 2012)	57
Figura 3.1. Fases del marco de trabajo DERE.....	62
Figura 3.2. Roles involucrados en la fase de definición	63
Figura 3.3. Ejemplo de análisis de respuestas realizado por los jefes de proyectos.....	69
Figura 3.4. Ejemplo de una gráfica de Kiviat para mostrar los resultados de evaluar diferentes procesos	70
Figura 3.5. Ejemplo de formato para mejorar el proceso de DMS y guía de uso.....	73
Figura 4.1. Proceso establecido para la incorporación de DERE al caso de estudio.....	79
Figura 4.2. Ejemplo de diapositivas utilizadas en EmpSoft para concientizar a la alta dirección sobre la importancia del compromiso.....	80
Figura 4.3. Ejemplo de diapositivas utilizadas en EmpSoft para motivar al personal técnico a participar en la mejora	81
Figura 4.4. Ejemplo de diapositivas utilizadas en la capacitación del personal directivo de EmpSoft.....	82
Figura 4.5. Ejemplo de diapositivas utilizadas en la capacitación del equipo de evaluación.....	85
Figura 4.6. Ejemplo de diapositivas utilizadas para la capacitación sobre APE y DMS	86
Figura 4.7. Proceso modelado por el personal técnico de EmpSoft	87
Figura 4.8. Proceso modelado vs proceso establecido por MoProSoft®	88
Figura 4.9. Coberturas totales de APE y DMS al finalizar la evaluación	89
Figura 4.10. Ejemplo de diapositivas utilizadas para la presentación de los hallazgos de la evaluación.....	91
Figura 4.11. Formación del grupo de trabajo para la fase “Realizar”	93
Figura 4.12. Diseño del nuevo proceso para EmpSoft	95
Figura 4.13. Incremento de la cobertura de DMS a través de la iniciativa de mejora.....	99

Resumen

En los últimos años, las iniciativas de mejora han adquirido gran importancia dentro de la industria de software de cualquier país. Estas “mejoras” buscan incrementar la calidad del software (como producto) a través del incremento de la calidad del proceso utilizado para su desarrollo. Esto, a través de modelos concretos que son utilizados como guías precisas que indican cómo hacer las cosas. Sin embargo, estos modelos tienen costos demasiado elevados para poder ser implementados dentro de las Micro, Pequeñas y Medianas Empresas de software (MiPyMEs), además de que son difíciles de entender y generan documentación excesiva. Es por ello que el Gobierno Federal Mexicano, a través de la Secretaría de Economía, lanzó en el 2002 el Programa para el Desarrollo de la Industria del Software (PROSOFT) con la finalidad de mejorar la calidad de los procesos y productos de la industria del país. En este contexto, esta tesis analiza al modelo MoProSoft® con el objetivo de proporcionar una descripción práctica y detallada a MiPyMEs Oaxaqueñas que buscan, a través de su adopción, mejorar y elevar la calidad de sus procesos y productos. Concretamente, se plantea diseñar un marco de trabajo que haga posible que estas empresas alcancen mejoras significativas de manera práctica, sencilla y a través de costos aceptables.

1. Introducción

La industria de software engloba al conjunto de empresas especializadas en los procesos para el desarrollo de software —aplicaciones, servicios e ingeniería— que incorpora además a otras industrias involucradas en el desarrollo de productos aplicados a las necesidades de industrias verticales, como los grandes conglomerados tecnológicos, la industria financiera y la industria electrónica. En este contexto, actualmente la industria de software en México está formada por Micros, Pequeñas y Medianas Empresas (MiPyMEs) que se componen en su mayoría por aproximadamente entre 15 a 200 empleados. En estas empresas es fundamental la calidad del software generado, ya que ésta podría generarles una ventaja competitiva en el mercado local y nacional. A pesar de que no existen estadísticas concretas, de acuerdo al Instituto Nacional de Estadística y Geografía (INEGI) [URL-1] México cuenta con 206 empresas desarrolladoras de software registradas en la Asociación Mexicana de la Industria de Tecnologías de la Información (AMITI), que han manifestado el interés de exportar sus servicios y productos. La Tabla 1 muestra la estructura de la oferta de acuerdo con el tamaño de las empresas.

Tabla 1. Clasificación de las empresas desarrolladoras de software en México. Fuente: Asociación Mexicana de la Industria de Tecnologías de Información.

Empresas desarrolladoras de software en México registradas en AMITI según clasificación por número de empleados			
Tipo de empresa	Número de empleados	Promedio de empleados (estimado)	Cantidad de empresas
Micro	<15	7	63
Pequeña	De 16 a 100	60	117
Mediana	De 101 a 250	175	14
Grande	De 251 a 1000	600	11
Corporativa	Más de 1000	1500	1

En este sentido y de acuerdo con (Mochi, Prudencio & Hualde, 2009), la industria mexicana de software se encuentra dividida en cuatro categorías:

- Industria nacional de software y servicios informáticos.

- Producción interna (consumo propio interno).
- Filiales de las grandes empresas transnacionales de software empaquetado y
- Grandes empresas transnacionales exportadoras de productos electrónicos.

Siguiendo esta clasificación, el total de producción de esta industria en México está dividido de la siguiente manera: el software empaquetado representa el 29.4% del total, el segmento de software hecho a la medida representa el 7.9%, mientras que el de producción y autoconsumo representa el 62.6% (véase Tabla 2) (Sampedro, 2011).

Tabla 2. Total de producción y nivel de participación de la industria del software en México (Sampedro, 2011).

Segmento de la industria	Producción/Desarrollo (MDD)	Porcentaje de participación	Sectores de demanda (por orden de importancia)
Software empaquetado	817	29.4%	Servicio, Gobierno, Finanzas, Comercio y Manufactura.
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.
Software de producción y consumo interno	1,738	62.6%	Gobierno y manufactura.
Total	2,776	100%	

En este contexto, la comunidad de especialistas ha afirmado constantemente que la industria de software en el mundo está en constante evolución y México no es la excepción. A pesar de que se sabe que la industria del país aún está en proceso de crecimiento tanto a nivel nacional como internacional, de acuerdo con datos de la Comisión Económica para América Latina y el Caribe (CEPAL) en el período comprendido entre enero de 2003 y octubre de 2010 la base de datos fDi Markets¹ del *Financial Times* registró 2,749 proyectos de inversión transfronteriza para la industria del software, que se localizaron principalmente en la India (24%), China (10%) y los Estados Unidos (10%). Por su parte, América Latina participa con un 5.7% del total de proyectos registrados, en comparación con el 48% de Asia y el Pacífico, el 21% de Europa occidental y el 9.5% de Europa oriental (véase la Figura 1.1).

¹ fDi Markets es la base de datos en línea más completa sobre inversiones transfronterizas, que abarca todos los países y sectores alrededor del mundo. Esta base de datos proporciona acceso a la monitorización en tiempo real de los proyectos de inversión, inversión de capital y creación de empleos.

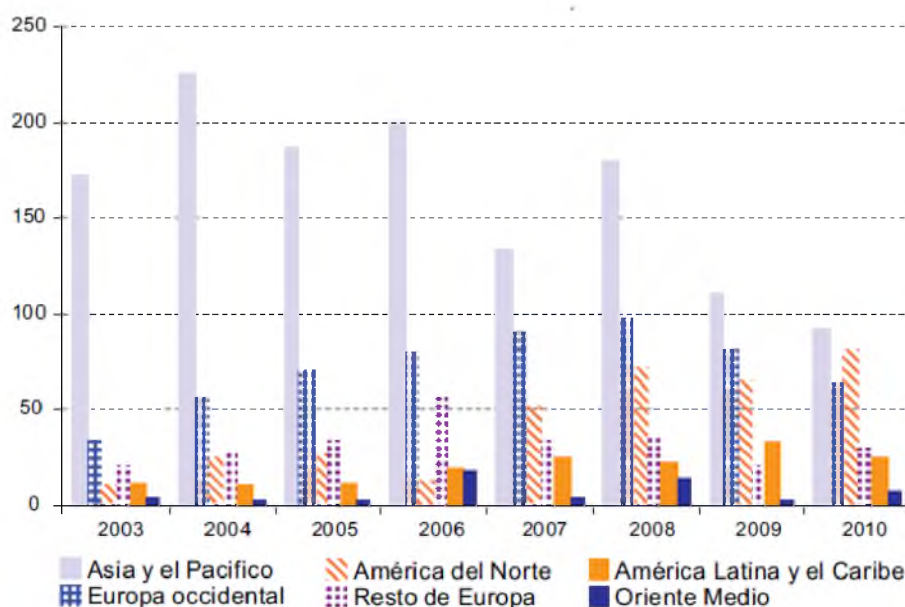


Figura 1.1. Evolución del número de proyectos de software anunciados por región, 2003-2010. Fuente: CEPAL

Los tres principales países de origen de los proyectos registrados durante el último año fueron los Estados Unidos (53%), la India (15%) y España (6%). En este sentido, de acuerdo con la Secretaría de Economía, México ocupó el 2° lugar en Latinoamérica como destino de inversión con el 23% del total de proyectos de software en la región. Las principales ciudades donde se localizaron los proyectos fueron Monterrey, Guadalajara y la Ciudad de México. De forma complementaria, José Luis Sampedro Hernández² afirma que en América Latina, México ocupa el segundo lugar en todos los sectores de Tecnologías de la Información (TI) en cifras absolutas, después de Brasil, aunque en el periodo 2005-2014 creció menos que este país y otros latinoamericanos.

Sin embargo, el crecimiento de la industria mexicana de software se ha visto afectado por varios factores tales como la educación, una infraestructura inadecuada y la ausencia de políticas que ayuden a motivar y facilitar el desarrollo e implementación del software. La carencia de formación académica ha generado que las empresas no cuenten con personal altamente capacitado para dominar lenguajes avanzados y nuevas tecnologías. De acuerdo con Mochi (2004), las empresas mexicanas engrosarían sus filas con egresados de licenciaturas, pero contarían con poco personal con maestrías y tendrían una ausencia casi absoluta de profesionales con doctorado. En este sentido Sampedro (2011) afirma que en la actualidad la industria de software en México se encuentra en una etapa de desarrollo y que sus problemas más importantes se acentúan en una limitada formación de recursos humanos especializados, infraestructura inadecuada, ausencia de un marco legal, modelos de evaluación y certificación de procesos, entre otros.

En este sentido, la investigación realizada en (García, Pacheco, & Calvo-Manzano, 2014) sobre la educación de la Ingeniería de Software en México, identificó cinco disfunciones

² El Dr. José Luis Sampedro Hernández es profesor investigador del Departamento de Estudios Institucionales (DESIN) de la Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Egresado de la Maestría en Economía y Gestión del Cambio Tecnológico (2003) y del Doctorado en Ciencias Sociales (2008), con especialidad en Economía y Gestión de la Innovación.

importantes: falta de atención en el proceso de software, falta de conciencia sobre la calidad del proceso de software, poca preocupación para planificar y controlar los proyectos de software, un abismo entre los cursos impartidos en las universidades y la industria de software y la falta de especialización en habilidades específicas. Dicha investigación recalca que uno de los problemas cruciales en la industria mexicana de software (y en general) es que las empresas normalmente deben invertir grandes recursos económicos en la formación de nuevos trabajadores debido a sus deficiencias de conocimientos.

En este contexto, desde los inicios de los años noventa, la industria y los investigadores interesados en la Ingeniería de Software han expresado un interés especial en la Mejora del Proceso de Software (*Software Process Improvement*) explorándola como una posible solución a los problemas mencionados anteriormente y que evidencian que la Ingeniería de Software dista mucho de ser un proceso maduro porque no es escalable y no puede satisfacer la demanda actual (Pino, García, & Piattini, 2008).

La Mejora del Proceso de Software es pues una actividad que las empresas de desarrollo de software implantan con el principal objetivo de adquirir capacidad y calidad para sus procesos, la cual se vea reflejada finalmente en sus productos. De acuerdo con (Oktaba & Piattini, 2008), un importante número de MiPyMEs dedicadas al desarrollo de software, carecen de una forma de trabajo adecuada y eficiente, lo cual da como resultado la generación de productos de baja calidad y el uso inadecuado de los recursos.

1.1. Importancia del problema y necesidad de la solución

Es verdad que a nivel de las grandes empresas existen varios modelos de mejora, tales como CMMI-DEV v1.3 (2010) e ISO/IEC 15504:2008 (ISO/IEC, 2008), que han adquirido gran importancia dentro de la industria del software de cualquier país. Sin embargo, aplicar estos modelos dentro de las MiPyMEs es una tarea difícil porque los costos son demasiado elevados para este tipo de empresas, además de que, en el contexto de éstas, estos modelos son difíciles de entender, generan mucha documentación y asumen que el personal de la empresa que los desea adoptar cuenta con habilidades y conocimientos ya consolidados.

Ante esta situación, el Gobierno Federal Mexicano, a través de la Secretaría de Economía, lanzó en el 2002 el Programa para el Desarrollo de la Industria del Software (PROSOFT) [URL-2] con la finalidad de acelerar y mejorar la calidad de los procesos y productos de la industria de software. Es decir, se da la primera aproximación de México a la Mejora del Proceso de Software como parte integral del éxito de cualquier empresa. PROSOFT fue lanzado oficialmente el 9 de octubre de 2002, con una visión a diez años (2003-2013) y estableció tres metas y siete estrategias para alcanzar su objetivo. En el proyecto han participado instituciones de educación técnica y superior, los gobiernos de todas las entidades federativas, las dependencias de la Administración Pública y Federal y la iniciativa privada (Mochi et al., 2009). Entre las metas para el año 2013 figuraron las siguientes:

- Dotar a México de capacidad exportadora de software y servicios para alcanzar los 5,000 millones de dólares anuales.
- Aumentar el nivel promedio de gasto en TI respecto del PIB (1.4%) para equipararse al promedio de los países de la OCDE (4.3%).

- Alcanzar la vanguardia latinoamericana de soporte y desarrollo de servicios basados en tecnologías de la información.

Las estrategias establecidas fueron las siguientes:

1. Promover las exportaciones y atraer inversiones.
2. Proporcionar educación y formación al personal en el desarrollo de software en cantidad y calidad convenientes.
3. Contar con un marco legal promotor de esa industria.
4. Desarrollar el mercado interno.
5. Fortalecer a la industria local.
6. Alcanzar niveles internacionales en capacidad de procesos.
7. Promover el desarrollo de agrupamientos empresariales.

En el año 2002 se inició un esfuerzo encabezado por el Gobierno Federal, por medio de la Secretaría de Economía, con los tres órdenes de gobierno, organismos empresariales y la academia, para elaborar un diagnóstico del sector de TI y buscar alternativas para impulsar su crecimiento. Organismos como la Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de la Información (CANIETI), AMITI y la Asociación Mexicana de Internet (AMIPCI) fueron parte de este proyecto con el objetivo de establecer los fundamentos para impulsar el crecimiento de la industria. De esta forma nació PROSOFT, que desde entonces ha contribuido al desarrollo de esta industria, intentando posicionarla a nivel mundial a través de MoProSoft®.

De acuerdo con Oktaba (2006), MoProSoft® es el Modelo de Procesos de Software desarrollado para la industria mexicana, el cual agrupa los procesos de una empresa en tres grandes categorías: (i) Alta Dirección, que contiene el proceso de Gestión de Negocios; (ii) Gerencia, que está integrada por los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos y por último, (iii) Operación, que está integrada por dos procesos; Administración de Proyectos Específicos y Desarrollo y Mantenimiento de Software. Cada uno de estos procesos define los roles responsables de la ejecución de las prácticas, los cuales son asignados al personal de la empresa de acuerdo a sus habilidades y capacitación para desempeñarlos.

MoProSoft® proporciona a las pequeñas y medianas empresas desarrolladoras de software, un modelo basado en las mejores prácticas internacionales con las siguientes características: fácil de entender, fácil de aplicar, no costoso en su adopción, e intenta constituirse como la base para alcanzar evaluaciones exitosas con otros modelos o normas, como ISO 9001:2000 (ISO, 2005) o CMMI v1.1 (CMMI, 2002). De acuerdo con la Dra. Oktaba³, el modelo MoProSoft®, a diferencia

³ Hanna Oktaba tiene el doctorado en Ciencias de la Computación otorgado por la Universidad de Varsovia, Polonia. Su actividad académica, iniciada en 1974 en Polonia, ha continuado desde 1983 en la Universidad Nacional Autónoma de México en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS). Oktaba ha escrito varios artículos de investigación y difusión sobre estos temas y ha dictado múltiples conferencias en foros nacionales e internacionales, específicamente en el área de la Ingeniería de Software, Tecnología Orientada a Objetos, Modelos de Procesos de Software y Mejora de Procesos. Oktaba fue fundadora y vicepresidenta de la ahora extinta

de otros modelos existentes, tiene la categoría de Alta Dirección que busca ordenar el trabajo de los directivos de una MiPyME desarrolladora de software, introduciendo prácticas actuales de gestión.

Las empresas que desean certificarse bajo este modelo de procesos deben lanzar una iniciativa de mejora que les permita obtener uno de los seis niveles de madurez que dicho modelo propone a través de un órgano certificador oficial. Así, en la actualidad, la Agencia Mexicana de Normalización y Certificación (NYCE) indica que 439 empresas están certificadas bajo los lineamientos de MoProSoft® de la siguiente manera: Nivel 0 = 8, Nivel 1 = 267, Nivel 2 = 152, Nivel 3 = 12, Nivel 4 = 0 y Nivel 5 = 0 [URL-3]. Es evidente que desde el 2005 que se inició con el proceso de certificación, las cifras más altas de empresas se han concentrado en los Niveles 1 y 2 puesto que representan los escalones iniciales de una mejora integral; sin embargo, lograr esto no es fácil cuando las empresas carecen del conocimiento sobre la Mejora del Proceso de Software, problema principal que será abordado en esta tesis. Aunado a esto, en México la información sobre iniciativas de Mejora del Proceso de Software aún es escasa y no existen guías puntuales para que una empresa que carece de este conocimiento pueda realizarla exitosamente.

Así pues, esta tesis se enfoca a establecer un marco de trabajo⁴ para que una empresa Oaxaqueña, denominada en adelante “EmpSoft” por cuestiones de confidencialidad, pueda alcanzar los Niveles 1 y 2 en la categoría de Operación de MoProSoft®.

1.2. Hipótesis de la tesis

La hipótesis de este trabajo se plantea como una relación causal y se enuncia de la siguiente manera:

“La creación de un marco de trabajo alternativo puede ayudar a que una pequeña empresa desarrolladora de software pueda implantar un modelo de referencia con el objetivo de alcanzar una mejora significativa en sus procesos.”

1.3. Objetivos de la tesis

Teniendo en mente la hipótesis anterior, se establecen los objetivos de la tesis a través de un objetivo general y varios objetivos secundarios, los cuales son descritos a continuación:

1.3.1. Objetivo general

Diseñar un marco de trabajo para una empresa pequeña de tal forma que le permita alcanzar una mejora significativa en sus procesos de desarrollo.

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

Asociación Mexicana para la Calidad en la Ingeniería de Software (AMCIS) y sigue siendo hasta el día de hoy coordinadora de la definición del Modelo de Procesos para la Industria de Software en México (MoProSoft®).

⁴ En el contexto de este trabajo de tesis, un *marco de trabajo* define, en términos generales, un conjunto estandarizado de conceptos, prácticas, guías y criterios para abordar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. En este sentido, un marco de trabajo tiene como objetivo ofrecer una funcionalidad definida, supervisada, siendo construidos usando documentos de soporte, y su característica principal es su alta cohesión y bajo acoplamiento.

1.3.2. Objetivos específicos

1. Analizar el modelo MoProSoft® proporcionando una descripción práctica y detallada para organizaciones Oaxaqueñas que persigan objetivos similares al propuesto en esta tesis.
2. Proporcionar mecanismos y pautas para MiPyMEs mexicanas que deseen consolidarse y crecer dentro del mercado competitivo del desarrollo de software a través de la adopción de MoProSoft®.
3. Identificar factores de éxito en “EmpSoft” que sean utilizados en casos de estudio similares.
4. Aportar información que ayude a las instituciones educativas de nivel superior a fortalecer los temas relacionados con la Mejora del Proceso de Software.

1.4. Delimitaciones de la tesis

Esta investigación se enfoca en brindar un marco de trabajo para las MiPyMEs desarrolladoras de software interesadas en adoptar el modelo MoProSoft®, con el propósito de mejorar la calidad de sus procesos/productos y así propiciar un ciclo de mejora en los mismos.

La implementación de este marco de trabajo estará delimitada a:

- Una pequeña empresa desarrolladora de software del estado de Oaxaca, México.
- La categoría de Operación de MoProSoft®, compuesta por los procesos de Administración de Proyectos Específicos y Desarrollo y Mantenimiento de Software.
- Los Niveles 1 y 2 del modelo MoProSoft® puesto que es recomendable que aquí comience una empresa que no tiene conocimiento alguno sobre la Mejora del Proceso de Software.

1.5. Limitaciones de la tesis

- Los mecanismos y guías diseñadas podrán ser implementados en MiPyMEs desarrolladoras de software las cuales no tienen implementada alguna versión del modelo MoProSoft® y que por consiguiente desconocen su estructura.

1.6. Solución propuesta

Las iniciativas de mejora suelen fundamentarse en tres tipos de modelos muy diferentes. La Figura 1.2 muestra que estos modelos se enfocan a la evaluación de los procesos actuales de la empresa para determinar su nivel de madurez/capacidad (e.g., CBA-IPI (Dunaway & Masters, 1996), ISO/IEC 15504:2008, SCAMPI (2011)). Dichas evaluaciones comparan la forma actual de trabajo de la empresa contra las prácticas y actividades recomendadas en modelos de referencia o de procesos (e.g., ITIL (Hochstein, Zarnekow, & Brenner, 2005), CMMI-DEV v1.3, CMMI-ACQ (Phillips, 2011), CobiT (Bernroider & Ivanov, 2011) con el objetivo de determinar fortalezas y debilidades. Por último, una vez que la empresa ha determinado sus deficiencias, un modelo de mejora (e.g., IDEAL (McFeeley, 1996), CMMI-DEV v1.3, ISO/IEC 15504:2008) le indica cómo alcanzar un incremento en la calidad de sus procesos y por ende de sus productos. En este sentido, los modelos de evaluación y de procesos se encuentran inmersos dentro de las fases del modelo de mejora, por lo que a la dificultad en su uso se suma la correcta elección de estos por el responsable de conducir la iniciativa de mejora.

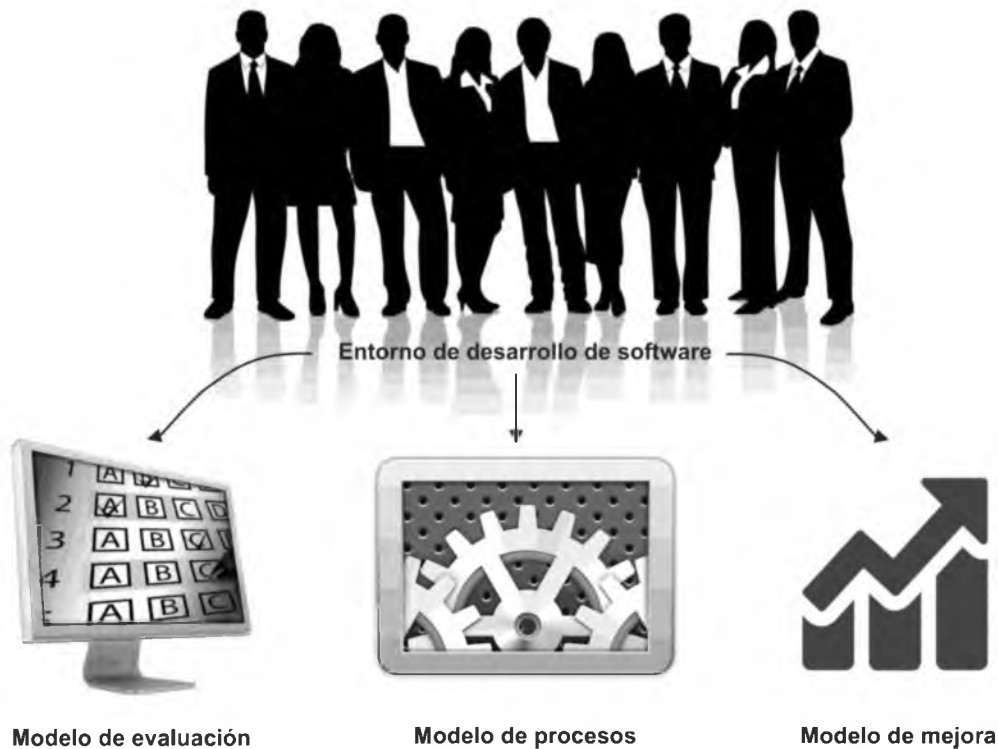


Figura 1.2. Categorías de modelos relacionados con una iniciativa de mejora

En este contexto, esta tesis pretende utilizar el modelo EvalProSoft® (NYCE, 2005^a) para diseñar un mecanismo de evaluación enfocado a MiPyMEs. En este sentido, existen muchas técnicas o métodos de evaluación que se utilizarán para este fin: entrevistas, revisión intensiva de documentos, cuestionarios, etc. El proyecto plantea la implementación del método de Evaluación basada en Cuestionarios (QBA, por sus siglas en inglés) (Marciniak & Sadauskas, 2003) puesto que permite aplicarse a muchas personas al mismo tiempo, es efectiva en costo y no invasiva, proporciona información cuantitativa, y los resultados son analizados rápidamente. Por otro lado, tal y como se menciona en el título de la tesis, se plantea un enfoque hacia el uso y adopción del modelo MoProSoft® (NYCE, 2005^b) puesto que pretende convertirse en el estándar de facto para la industria mexicana. Cabe mencionar que ambos modelos, EvalProSoft® y MoProSoft®, fueron escogidos puesto que han sido creados específicamente para el contexto de las MiPyMEs de México. Finalmente, dado que en la actualidad no existe un modelo de mejora creado específicamente para las MiPyMEs, esta tesis propone la creación de un marco de trabajo que permita que estas empresas adopten y usen un modelo de procesos (concretamente MoProSoft®) para mejorar su nivel de madurez/capacidad. Así, tomando como base al modelo IDEAL (McFeeley, 1996), se pretende diseñar el marco de trabajo DERE (Definir, Evaluar, Realizar, Extender) para simplificar el trabajo de los responsables de las iniciativas de mejora de un entorno pequeño (véase la Figura 1.3).



Figura 1.3. Propuesta del marco de trabajo para conducir iniciativas de mejora en MiPyMEs

Este marco de trabajo estará compuesto por las siguientes etapas:

- **Definir** el alcance de la iniciativa de mejora a través del establecimiento de objetivos alcanzables y compromisos que aseguren el éxito del esfuerzo por mejorar.
- **Evaluar** el estado actual de la empresa con el objetivo de determinar fortalezas y debilidades en su forma de trabajo. Esta valoración cualitativa/cuantitativa permite establecer un nivel de madurez para la empresa y un nivel de capacidad para su personal. Como resultado de la evaluación se genera un plan de mejora que busca solucionar todas las debilidades halladas durante la evaluación y diseñar (o modificar con base al actual) un proceso efectivo para la empresa.
- **Realizar** proyectos piloto con equipos piloto que utilicen el nuevo proceso con el objetivo de medir logros tempranos y mantener el compromiso e interés mostrados en la etapa inicial. Es obvio que es necesario establecer medidas de éxito que permitan determinar el grado de éxito de la nueva forma de trabajo.
- **Extender** el nuevo proceso al resto del personal con el objetivo de institucionalizar las prácticas a través de su despliegue controlado.

Adicionalmente, este marco de trabajo deberá ser validado en un entorno real para demostrar su factibilidad. En este sentido, actualmente los resultados que se obtienen con la aplicación de metodologías, técnicas y herramientas que se utilizan para el desarrollo de software son

impredicibles. De acuerdo con (Basili, Selby, & Hutchens, 1986), la experimentación en la Ingeniería de Software promovería el avance en el área a través de un proceso iterativo de aprendizaje. De acuerdo con Juristo y Moreno (2010), la experimentación en Ingeniería de Software se refiere a demostrar con hechos las suposiciones, hipótesis, especulaciones y creencias que abundan alrededor de la construcción y uso del software. Por lo tanto, es necesario analizar, estudiar y utilizar los métodos de investigación cuando se trata de validar científicamente alguna propuesta en el campo de la Ingeniería de Software. Así, en (Wohlin, Hoest, & Henningsson, 2003) se indica que las estrategias empíricas para el desarrollo de experimentos en Ingeniería de Software incluyen a las encuestas, casos de estudio y experimentos. Particularmente un caso de estudio se utiliza para investigar una única entidad o fenómeno dentro de un espacio de tiempo específico. Durante la realización del caso de estudio, es posible aplicar una variedad de procedimientos para la recopilación de los datos (Creswell, 2008). Sin embargo, si se desea comparar dos métodos, puede ser necesario organizar el estudio como un caso de estudio o un experimento. La elección depende de la escala de evaluación. Así pues, los casos de estudio son adecuados para la evaluación industrial de métodos y herramientas de Ingeniería de Software porque evitan problemas de escalado (Juristo & Gómez, 2012). De acuerdo con Yin (2009), una ventaja de los casos de estudio es que son más fáciles de planificar, pero la principal desventaja es que los resultados son difíciles de generalizar y más difíciles de interpretar, es decir, es posible mostrar los efectos en una situación típica, pero no puede generalizarse a cualquier situación.

Es evidente así, que se ha escogido un caso de estudio como la estrategia empírica experimental a seguir en la validación de las hipótesis establecidas en esta tesis.

1.7. Estructura de la tesis

La estructura del documento de tesis se detalla a continuación:

El Capítulo 2 presenta el marco conceptual sobre el modelo MoProSoft® y expone un análisis sobre propuestas similares relacionadas con la creación de guías o herramientas para facilitar la adopción del modelo en el contexto de las MiPyMEs.

El Capítulo 3 describe el diseño del marco de trabajo DERE.

El Capítulo 4 presenta el caso de estudio en “EmpSoft”, las principales lecciones aprendidas y todos los artefactos generados durante la experimentación.

El Capítulo 5 presenta las conclusiones y líneas futuras sobre este trabajo.

El Anexo A presenta una lista de acrónimos que son mencionados a lo largo de la presente tesis.

El Anexo B presenta el repositorio de la documentación de soporte a DERE, como son las plantillas definidas para el uso del marco de trabajo.

El Anexo C presenta los cuestionarios diseñados para la fase de evaluación del caso de estudio.

El Anexo D presenta los formatos diseñados para la fase de realización del caso de estudio.

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

2. Marco Teórico

En la actualidad el software se ha convertido en un elemento sumamente importante sobre el cual yace la estructura, información y procesos de negocio de las empresas que se dedican al desarrollo e implementación del mismo. En este sentido, de acuerdo con (Richardson & Gresse von Wangenheim, 2007), para detectar la carencia de gestión de los procesos y su mantenimiento dentro de una empresa, la forma de trabajar debe analizarse desde dos vertientes: las características del propio proceso y los resultados del mismo.

Desde el punto de vista de las características del proceso, se deben observar:

- Roles del equipo de desarrollo, incluido el de jefe de proyecto, poco definidos o no coordinados.
- Improvisación sobre la marcha de los profesionales del desarrollo.
- Pérdidas de rendimiento del equipo, que se encuentra sometido a crisis constantes (apaga fuegos).
- Adecuación inexistente, en muchos casos, entre la capacidad y habilidades del equipo de desarrollo y la tarea encomendada.
- Falta de un seguimiento adecuado del avance del desarrollo y del punto exacto en el que se está dentro del proceso.

Desde el punto de vista de los resultados obtenidos con el proceso de software, se debe observar otro tipo de problemas:

- El producto suele retrasarse en su plazo de entrega.
- El producto no cumple con la funcionalidad completa requerida por el cliente.
- Es usual la existencia de sobrecostos de todo tipo.
- Las actividades dedicadas a garantizar la calidad, tales como revisiones y pruebas, suelen recortarse o eliminarse directamente.

Estos dos enfoques claramente permiten visualizar que el problema radica en la falta de capacidad (técnica y de recursos) para gestionar el proceso de software, y que el éxito de la empresa

dependerá de su disponibilidad por mejorarlo. Así, para diseñar una iniciativa de mejora es necesario entender primero en qué consiste la Mejora del Proceso de Software:

- Zahran (1998) afirmó que un programa de mejora representaba *“un conjunto integrado de iniciativas cuyo objetivo es el mejoramiento de las actividades de desarrollo y/o mantenimiento de productos basados en software”*.
- De acuerdo con la definición proporcionada por (Cuevas, De Amescua Seco, San Feliu, Calvo-Manzano, Arcilla, García, & Cerrada, 2002), *“la Mejora del Proceso de Software es un mecanismo que establece un ciclo continuo de mejora de la calidad de un producto de software”*.
- Finalmente, para el CMMI-DEV v1.3 (2010), una iniciativa de mejora *“puede utilizarse cuando una empresa desea evaluar la capacidad de sus contratistas, auditar sus desarrollos internos de software, y planificar su estrategia de Ingeniería de Software”*.

Con base a las definiciones anteriores se puede entender que el valor de la Mejora del Proceso de Software, a nivel de una empresa de desarrollo de software, es significativamente importante. Este proceso, la mejora en sí, debe de considerar las relaciones de todas las tareas requeridas, los métodos y herramientas utilizadas, así como las habilidades y motivaciones de las personas involucradas. En este sentido, durante la última década las empresas pequeñas inmersas en la industria de software han estado cada vez más interesadas en beneficiarse de las ventajas que proporciona la mejora de sus procesos (Pino, García, & Piattini, 2008). La Figura 2.1 evidencia que, durante los últimos años, la comunidad de investigadores relacionados con la Ingeniería de Software ha realizado estudios enfocados a aumentar la calidad de los productos y la productividad de las pequeñas empresas de desarrollo a través de iniciativas de mejora. Sin embargo, aún existe una tendencia generalizada a subrayar que el éxito de una iniciativa de Mejora del Proceso de Software es posible sólo en las grandes empresas. Las pequeñas empresas, que dominan el mercado internacional, también pueden fortalecer sus prácticas de Ingeniería de Software con las mejores prácticas que han funcionado en las grandes empresas y que se han adaptado a su tamaño y tipo de negocio.

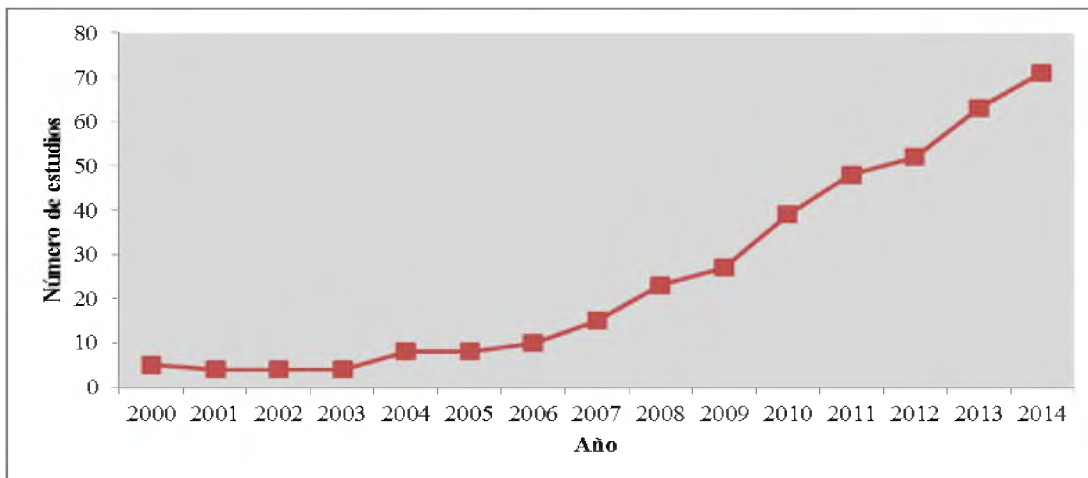


Figura 2.1. Tendencia de publicación de estudios realizados sobre la Mejora del Proceso de Software. Modificada de (Pino, García, & Piattini, 2008).

Sin embargo, uno de los mayores retos para que una iniciativa de mejora alcance el éxito en una pequeña empresa reside en que ésta entienda cómo deben implementarse los nuevos procesos. En este sentido, de acuerdo con Dyba (2005) existen cinco factores críticos que pueden ayudar a una empresa a entender e implementar iniciativas de mejora de sus procesos: orientar la iniciativa con los objetivos de negocio, comprometerse a nivel de liderazgo, obtener la participación activa del personal, establecer un conjunto de indicadores para evaluar el progreso de la mejora, y establecer diferentes estrategias de aprendizaje para cubrir las deficiencias técnicas. Así, para mejorar el proceso de software de cualquier empresa, es necesario tener clara la meta que se desea conseguir y al mismo tiempo dar un seguimiento continuo del progreso alcanzado en el camino hacia ésta. De acuerdo con Humphrey (2008), al establecer la posición de la empresa en una estructura de madurez, los profesionistas del software y los gestores del proceso están listos para identificar áreas donde las acciones de mejora serían más productivas.

Es verdad que la Mejora del Proceso de Software es una actividad compleja y costosa, ya que requiere de una gran cantidad de recursos humanos y financieros y principalmente del compromiso a largo plazo. Por ejemplo, abundando un poco más sobre el “compromiso”, numerosas investigaciones han demostrado que existe una cantidad importante de problemas relacionados con esta etapa inicial de la mejora (e.g., Baddoo & Hall, 2003; Niazi, Wilson, & Zowghi, 2006; Niazi, Babar, & Verner, 2010). Entre estos problemas se ha identificado que, por muy pequeño que sea el cambio en los procesos de una empresa, se puede generar una sobrecarga de trabajo. Se ha hecho referencia también a la carencia de recursos internos, ya que algunos procesos requieren más trabajo por parte del personal y muchas actividades terminan por no realizarse. En este sentido, de acuerdo con la revisión de literatura presentada en (Unterkalmsteiner, Gorschek, Islam, Chow-Kian, Permadi, & Feldt, 2012), los problemas que afectan el desarrollo de una iniciativa de mejora son bien conocidos y se resumen de la siguiente manera:

- Es verdad que cada vez son más las empresas que se atreven a iniciar un programa de Mejora del Proceso de Software, pero éstas no consiguen despegar en su totalidad ya que gastan dinero y tiempo en talleres de consultoría y en la evaluación de su proceso.
- Regularmente los especialistas contratados crean un manual amplio de procedimientos en donde se explican paso a paso los cambios a realizar inmediatamente, es decir se proporciona el qué, pero no el cómo.
- Es común que la falta de compromiso por parte de la alta dirección, la falta de capacidad de los niveles gerenciales para entender que la mejora no intenta introducir un proceso nuevo sino mejorar el actual, y la mala formación del personal ocasionen un rechazo cultural hacia la mejora.
- Por último, haciendo énfasis en el rechazo cultural, el miedo a las nuevas tecnologías, métodos y herramientas que puede generarse a través de la mejora, incrementa la frustración en la empresa y se establece una barrera para no dejar la antigua forma de trabajo.

Con base a lo anterior existe un reto crítico que enfrentan las empresas de software, el de mantener sus capacidades tanto de eficiencia como de flexibilidad. Esta situación que requiere la administración del cambio y la estabilidad, ha conducido a investigadores y profesionistas a establecer esfuerzos dirigidos a mejorar aquellas capacidades que resulten más fundamentales. Es decir, las empresas de software pueden implementar dos tipos de estrategias de aprendizaje con miras a la mejora: (1) orientarse hacia la adopción y utilización de conocimiento y experiencias

existentes (explotación), o (2) hacia la búsqueda de nuevo conocimiento, ya sea a través de imitación o innovación (exploración). Así, la *explotación* involucra capacidades existentes para refinar, estandarizar y elaborar paradigmas, tecnologías, estrategias e ideas establecidas. Por otro lado, la *exploración* involucra el aprendizaje por medio del descubrimiento y la experimentación de nuevas ideas, paradigmas, tecnologías, estrategias y conocimientos, con la idea de encontrar otras alternativas y oportunidades que resulten mejores para las prácticas actuales. Por lo tanto, se define a las *estrategias de aprendizaje* como el grado en el que una organización de software está comprometida con la explotación de conocimiento existente y la exploración de nuevo conocimiento (Dyba, 2005).

A pesar de las complicaciones descritas anteriormente, cada vez son más las empresas que se atreven a comenzar una iniciativa de mejora ya que, si son guiadas correctamente, pueden obtener beneficios que impactan directamente en la mejora de la calidad, productividad, desarrollo y mantenimiento del software que producen. En este sentido, la investigación de (Sulayman, Urquhart, Mendes, & Seidel, 2012) identifica que una empresa que establece y desarrolla correctamente un programa de mejora puede obtener los siguientes beneficios:

- Reducción de los costos de desarrollo y mantenimiento del software. Los costos de la implementación de las mejoras en el proceso de software están altamente contrarrestados con los ahorros de costo en retrabajo (al ser menor) y en desarrollo (al reducirse).
- Mejora la satisfacción del cliente. Lo normal es que las empresas lancen sus productos con un 15% de defectos que permanecen ocultos hasta que el cliente los encuentra. De ninguna manera ningún cliente está satisfecho con tantos problemas. Las mejoras sobre el proceso de software reducen los defectos post-lanzamiento hasta casi cero. Así pues, mejorar la satisfacción del cliente da como resultado conservarlo y mejorar la imagen de la empresa.
- Reducción del tiempo de ciclo. Los esfuerzos de mejora reducen normalmente la duración de los proyectos (calendarios) entre un 30% y 40%. Esto se convierte en mayores beneficios, pues permite a las empresas abordar un mayor número de proyectos.
- Mejora el beneficio. El Retorno de la Inversión (ROI) para la mejora del software es muy alto. Muchas empresas han informado un ROI de 7 a 1. Este ROI alto se consigue reduciendo los costos de retrabajo. Las ventas del producto aumentan debido a que la calidad del software es más alta, además de que no se incurre en penalizaciones y los proyectos de la empresa aumentan.
- Mejora la profesionalidad del personal. La Mejora del Proceso de Software mejora la moral del empleado y aumenta la confianza de los desarrolladores. Una de las consecuencias directas de lo anterior, es que existe menos pérdida de tiempo, menos crisis, menos rotación de personal y se mejora la competitividad. Solamente con la reducción de costos por una menor rotación del personal podrían pagarse los costos de la mejora.

En este contexto, la Mejora del Proceso de Software requiere evaluar el proceso actual de las empresas para identificar las debilidades y fortalezas del proceso actual, y después establecer un plan de mejora que debe ejecutarse para reducir la problemática entre los procesos actuales y los que deberán realizarse en el futuro. De igual forma esto ayudará a determinar el nivel de madurez de los procesos y así alcanzar sin problemas la planeación, las metas de costos y el objetivo de calidad establecidos. Obviamente que una pequeña empresa, por ejemplo, necesita establecer un método de

evaluación adaptado a su medida y que le ayude a realizar evaluaciones a costos razonables. No obstante, la realización de esta tarea requiere rotundamente contar con una guía y ayuda constantes. Es decir, aplicar las prácticas recomendadas por los especialistas es esencial para el crecimiento de cualquier empresa; sin embargo, realizar este esfuerzo sin experiencia previa puede delimitar el aspecto de innovación de una empresa pequeña. El cambio planteado debe dar respuesta de tal forma que arroje resultados a la medida de la cultura y entorno de la empresa, considerando por supuesto los objetivos de negocio establecidos en ese momento.

2.1. Fundamentos de la Mejora del Proceso de Software

Desafortunadamente en empresas inmaduras (como las pequeñas empresas) es difícil medir los beneficios de una iniciativa de mejora del proceso debido a que raramente almacenan datos de sus proyectos anteriores (relacionados con los costos, calendarios, habilidades empleadas, y demás). En este tipo de empresas, el principal beneficio obtenido por la Mejora del Proceso de Software sería la capacidad para cumplir con el calendario planificado del proyecto, evitando desviaciones que generan los sobrecostos.

En este sentido, de acuerdo con Cuevas et al. (2002), para guiar, facilitar y asegurar la Mejora del Proceso de Software en una empresa se han desarrollado una serie de modelos que detallan los pasos a seguir, con el fin de lograr el éxito en el programa de mejora (véase Figura 2.2). Estos modelos permiten establecer un ciclo de actividades que introducen la mejora de manera paulatina (modelo de mejora), a partir de la determinación de fortalezas y debilidades actuales (modelo de evaluación), y siguiendo las pautas de un conjunto de mejores prácticas que ha sido reconocido por la comunidad internacional (modelo de proceso o de referencia). A continuación, se expone una breve definición de cada uno de estos modelos.



Figura 2.2. Modelos involucrados en una iniciativa de mejora

2.1.1. Modelos de proceso

Un proceso representa una base de conocimiento, y puesto que el conocimiento está esparcido, el desarrollo de software implícito e incompleto es en gran medida un proceso de aprendizaje social. Así, el proceso proporciona un medio de interacción entre los usuarios y el equipo de trabajo (recursos humanos), las herramientas de desarrollo (tecnología), y las tareas a desarrollar (actividades). En este sentido, se entiende que el proceso de software es un marco de trabajo que involucra a las actividades necesarias a realizar para la construcción de un producto de software de buena calidad.

En este sentido, un modelo de proceso, o de referencia, para la Ingeniería de Software es un marco abstracto u ontología de un dominio específico que consiste en un conjunto interrelacionado de conceptos, claramente definidos y producidos por un experto o grupo de expertos, con el fin de

fomentar la práctica eficiente del arte. A través de un modelo de referencia se pueden identificar funciones y responsabilidades claras. De acuerdo con (INTENCO, 2009), el uso de un modelo de proceso genera las siguientes ventajas a las empresas:

- Desde el punto de vista de gestión:
 - Facilita la tarea de planificación.
 - Facilita las tareas de control y seguimiento de los proyectos.
 - Mejora la relación costo/beneficio.
 - Optimiza el uso de los recursos disponibles.
 - Facilita la evaluación de los resultados y el cumplimiento de los objetivos.
 - Facilita la comunicación efectiva entre los usuarios y desarrolladores.
- Desde el punto de vista de los ingenieros de software:
 - Ayuda a la comprensión del problema.
 - Optimiza el conjunto y cada una de las fases del proceso de desarrollo.
 - Facilita el mantenimiento del producto final.
 - Permite la reutilización de partes del producto.
- Desde el punto de vista del cliente o usuario:
 - Garantiza un determinado nivel de calidad en el producto final.
 - Genera confianza sobre los plazos de tiempo fijados en la definición del proyecto.
 - Define el ciclo de vida que más se adecue a las condiciones y características del desarrollo.

2.1.2. Modelos de mejora

En los últimos años, los modelos para la Mejora del Proceso de Software han sido una alternativa para lograr el incremento de la calidad de los servicios y productos que ofrece una empresa de software. Dichos modelos tratan de aprovechar las oportunidades de mejora y de esta forma, establecer una mejora por medio del conocimiento, costos y comprensión de una implementación. A lo largo de la literatura se mencionan tres modelos de reconocido prestigio internacional:

- Modelo IDEAL (McFeeley, 1996) desarrollado por el Instituto de Ingeniería de Software (SEI, por sus siglas en inglés).
- Modelo de Mejora del Proceso de Software desarrollado por el Instituto para la Mejora del Proceso de Software (ISPI, por sus siglas en inglés) (Wilmoth, Prigmore, & Bray, 2002), que define las siguientes etapas:
 - Compromiso para la mejora: Esta etapa indica que la alta dirección de la empresa debe comprometerse e involucrarse en el proyecto de mejora, proporcionando personal necesario, tiempo y otros recursos que sean necesarios para obtener éxito en el despliegue de la mejora propuesta.

- Evaluación del proceso: Esta etapa determina cuál es el proceso de software que la empresa emplea en la actualidad, ayuda a encontrar las fortalezas y debilidades del mismo mediante un comparativo de las prácticas existentes con las de un modelo de proceso.
 - Establecimiento de infraestructura y planes de acción para la mejora: Esta etapa tiene como objetivo definir las funciones y responsabilidades de cada integrante de la infraestructura requerida para la mejora del proceso. Los resultados obtenidos en la evaluación determinarán el plan de ejecución para implementar las mejoras correspondientes.
 - Implantación e institucionalización de la mejora: Esta etapa permite realizar actividades que involucren una mejora, las cuales serán previamente definidas en un plan de ejecución. También se deben ejecutar proyectos pilotos para asegurar que las nuevas actividades funcionan y mejoran el proceso de la empresa.
- ISO/IEC 15504/2008 Tecnología de la Información - Evaluación del Proceso Software – Parte 7: Guía para utilización en la mejora del proceso (ISO/IEC, 2008).

2.1.3. Modelos de evaluación

La evaluación de un proceso se define como el examen disciplinado de los procesos usados en una empresa frente a un conjunto de criterios para determinar la capacidad de dichos procesos, con el objetivo de determinar si éstos son realizados dentro de los objetivos de calidad, costo y planificación (García, Pacheco, & Calvo-Manzano, 2010). El propósito es caracterizar la práctica actual, identificando debilidades y fortalezas y la habilidad del proceso para controlar o evitar las causas de baja calidad, desviaciones en costo o planificación. La actividad de evaluación muestra el cambio que se obtuvo mediante la adopción de un modelo de mejora. En los últimos años se han propuesto varios enfoques para la evaluación y mejora del proceso de software con la finalidad de determinar la capacidad del proceso de una empresa. Por ejemplo, en (Pressman, 2010) se definen algunos de los principales métodos y modelos de evaluación del proceso de software:

- Método de Evaluación basada en CMM® para la Mejora Interna del Proceso (CBA-IPI, por sus siglas en inglés): Este método proporciona una herramienta de diagnóstico que permite a las organizaciones y proyectos el poder determinar las fortalezas y debilidades de sus procesos de desarrollo de software. El método utiliza el Modelo de Madurez de Capacidad para Software desarrollado por el SEI, el SW-CMM v1.1. A diferencia de CBA-IPI®, SCAMPI® tiene un alcance más amplio al poder evaluar múltiples funciones de la organización relacionadas con las disciplinas del modelo (Dunaway & Masters, 1996). Es decir, es un sólo método para evaluaciones de varias disciplinas (software, sistemas, etcétera).
- Método Estándar de Evaluaciones basadas en CMMI® para la Mejora de Procesos (SCAMPI, por sus siglas en inglés): Una evaluación SCAMPI® es la forma de evaluar si una empresa de desarrollo de software cumple con un modelo de la familia de CMMI®. Es decir, una evaluación SCAMPI® es una forma para identificar qué hace la empresa, cómo lo hace, verificar que hace lo que dice, como dice que lo hace, y verificar que esta forma de ejecutar los procesos se apega a lo que dice algún modelo de CMMI®. En este sentido, una evaluación se puede utilizar para desarrollar diagnósticos internos y/o para la evaluación de

proveedores, y es capaz de producir un perfil de capacidades o bien un nivel de madurez para una organización (SCAMPI, 2011).

- EvalProSoft®: El propósito de este modelo de evaluación consiste en definir un método para obtener un nivel de la capacidad de sus procesos y un nivel de madurez de capacidades de la organización, tomando como referencia MoProSoft®. La evaluación de cada proceso conlleva a un examen disciplinado, el cual se apoya en una escala, criterios de evaluación, conjunto de estándares mejores prácticas y un mecanismo claro para exponer los resultados obtenidos (NYCE, 2005^a).

Es importante mencionar que un modelo de evaluación no es garantía de que el software se entregue dentro de los tiempos establecidos, ni que cumpla con las necesidades de los usuarios, solamente muestra los resultados de evaluar al proceso ejecutado.

En este contexto, las siguientes secciones describirán a detalle los modelos MoProSoft® y EvalProSoft® como el modelo de proceso y el modelo de evaluación, respectivamente, creados específicamente para la industria mexicana de software. Por último, se discutirá el uso de un modelo de mejora que permita que las MiPyMEs adopten el modelo MoProSoft® en sus actividades cotidianas.

2.2. Modelo de proceso para la industria mexicana de software: MoProSoft®

Tal y como se ha visto a lo largo de esta tesis, la industria de software a nivel mundial se ha convertido en una actividad económicamente importante, ya que se caracteriza por generar un valor adicional en el mercado, y contribuye a la modernización del país y de las empresas a través de la innovación tecnológica y empleos mejor remunerados. En este contexto, la Secretaría de Economía del Gobierno Mexicano creó el Programa de Desarrollo de la Industria de Software (PROSOFT) con el objetivo de *“Lograr una producción anual de software de 5,000 millones de dólares, alcanzar el promedio mundial de gasto en tecnologías de información y además, convertir a México en el líder latinoamericano de desarrollo de software y contenidos digitales en español”* [URL-4].

Así, para alcanzar los niveles internacionales en capacidad de procesos se propuso la definición de un modelo de referencia y evaluación de los procesos para la industria de software mexicana. Para lograrlo se definió un equipo de trabajo conformado por empresarios mexicanos y académicos especializados, dirigidos por la Dra. Hanna Oktaba –profesora de la Facultad de Ciencias de la UNAM– con la finalidad de crear un modelo de procesos de software, tomando en consideración las mejoras prácticas de la industria y otros modelos de proceso tales como SW-CMM (Paulk, 1995), ISO/IEC TR 15504 (ISO/IEC, 1998) e ISO 9000:2000 (ISO, 2005). Es así como en diciembre del 2002 se genera la primera versión del Modelo de Procesos para la Industria de Software (mejor conocido como MoProSoft®). En mayo del 2003 se crea la versión v1.1 y para agosto del 2005 se libera la versión v1.3, actualmente vigente, y que da sustento a la Norma mexicana NMX-I-059/02-NYCE-2005 (NYCE, 2005b), la cual fue publicada el 15 de agosto del 2005 en el Diario Oficial de la Federación.

De acuerdo a sus creadores, la adopción de MoProSoft® permitiría elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. En este sentido, este modelo está dirigido a las empresas o departamentos internos dedicados al desarrollo y/o mantenimiento de software. De esta manera aquellas organizaciones que no cuenten con procesos establecidos, 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 elementos que les hace falta cubrir. La elaboración de MoProSoft® contempló los siguientes criterios:

- Generar una estructura de procesos que estuviera acorde con la estructura de las organizaciones que conforman la industria mexicana de software (es decir, que abarcara los niveles de Alta Dirección, Gestión y Operación).
- Destacar el papel de la Alta Dirección en la planificación estratégica, y la importancia de su revisión y soporte a la mejora continua fungiendo como el promotor del buen funcionamiento de la organización.
- Considerar al nivel de Gestión como proveedor de recursos, procesos y proyectos, y responsable de vigilar el cumplimiento de los objetivos estratégicos de la organización.
- Considerar al nivel de Operación como ejecutor de los proyectos de desarrollo y mantenimiento de software.
- Integrar de manera clara y consistente los elementos indispensables para la definición de los procesos y las relaciones entre éstos.
- Integrar los elementos para la administración de proyectos en un sólo proceso.
- Integrar los elementos para la ingeniería de productos de software en un solo marco que incluya los procesos de soporte (verificación, validación, documentación y control de configuración).
- Destacar la importancia de la gestión de los recursos, en particular aquellos que componen la base de conocimiento de la organización, tales como: productos generados por proyectos, datos de los proyectos (incluyendo las mediciones), documentación de los procesos y los datos recaudados a partir de su uso y lecciones aprendidas.
- Basar el modelo de procesos (MoProSoft®) en ISO 9000:2000 y los niveles 2 y 3 de CMMI® V.1.1. Adicionalmente, usar como marco general de evaluación el estándar ISO/IEC 15504 - Software Process Assessment (ISO/IEC, 1998) e incorporar las mejores prácticas de otros modelos de referencia tales como PMBoK y SWEBOK (IEEE, 2004).

Dado que se pretendía establecer un modelo que las MiPyMEs de software pudieran entender, se establecieron algunas definiciones básicas para reducir la complejidad y aumentar la comprensión del modelo, por ejemplo:

- Categoría de procesos: Un conjunto de procesos que abordan la misma área general de actividad dentro de una organización.
- Proceso: Conjunto de prácticas relacionadas entre sí, llevadas a cabo a través de roles y por elementos automatizados, que utilizando recursos y a partir de insumos producen un satisfactor de negocio para el cliente.
- Objetivo: Fin a que se dirige o encamina una acción u operación.
- Indicador: Mecanismo que sirve para mostrar o significar una cosa con evidencias y hechos.

- Rol: Es responsable por un conjunto de actividades de uno o más procesos. Un rol puede ser asumido por una o más personas de tiempo parcial o completo.
- Producto: Cualquier elemento que se genera en un proceso.
- Práctica: Un conjunto de elementos, tales como actividades, roles, infraestructura y mediciones, que al llevarse a cabo describen la ejecución de un proceso.
- Actividad: Conjunto de tareas específicas asignadas para su realización a uno o más roles.
- Verificación: Actividad para confirmar que el producto refleja propiamente los requerimientos especificados para él.
- Validación: Actividad para confirmar que el producto resultante es capaz de satisfacer los requerimientos para su aplicación especificada o uso previsto.
- Flujo de trabajo: Esquema que expresa las relaciones entre las actividades de un proceso. Una relación puede ser secuencial, paralela, cíclica, de selección o anidada.
- Guía de ajuste: Modificación a las prácticas, entradas y salidas de un proceso, siempre y cuando no afecten al cumplimiento de sus objetivos.
- Gestión: Hacer diligencias conducentes al logro de un negocio.
- Administración: Organizar trabajo y disponer recursos.
- Organización: Empresa o área interna de una organización dedicada al desarrollo y/o mantenimiento de software.
- Infraestructura: Conjunto de elementos o servicios que se consideran necesarios para la creación y funcionamiento de una organización.
- Medición: Acción o efecto de medir.
- Base de conocimiento: Es un repositorio de todos los productos tales como productos de software, planes, reportes, registros, lecciones aprendidas y otros documentos.
- Situación excepcional: Circunstancia que impide el desarrollo de una actividad.
- Lección aprendida: Experiencia positiva o negativa obtenida durante la realización de alguna actividad.
- Prospección: Estudio de la potencialidad o de la capacidad que tiene alguna cosa para producir o dar resultados en el futuro, a partir del análisis de los datos reunidos previamente.

Como mecanismo de simplificación de los modelos considerados para su creación, MoProSoft® introdujo el concepto de “patrón de procesos” para definir un esquema de elementos para la documentación de los procesos. El patrón está constituido por las siguientes partes:

- La *Definición general del proceso* identifica el nombre de un proceso, la categoría a la que pertenece, su propósito, una descripción general de sus actividades, sus objetivos e indicadores, las metas cuantitativas, responsabilidad y autoridad, los subprocesos (en caso de tenerlos), los procesos con los que se relaciona, sus entradas y salidas, los productos internos y referencias bibliográficas.

- Las *Prácticas* identifican los roles involucrados en el proceso y la capacitación requerida. Describen también a detalle las actividades, asociándolas a los objetivos del proceso. Como ayuda adicional el modelo utiliza diagramas de flujo de trabajo, describe las verificaciones y validaciones requeridas para introducir calidad al trabajo, lista los productos que se deben incorporar a la base de conocimiento, identifica los recursos de infraestructura necesarios para apoyar las actividades, establece las mediciones del proceso, así como las prácticas para la capacitación, manejo de situaciones excepcionales y uso de lecciones aprendidas.
- Las *Guías de ajuste* sugieren modificaciones al proceso que no deben afectar los objetivos del mismo.

Tal y como se indicó anteriormente, y como lo muestra la Figura 2.3, MoProSoft® fue creado para cubrir tres categorías de procesos que reflejan la estructura de cualquier organización:

- La categoría de Alta Dirección contiene el proceso de Gestión de Negocio.
- La categoría de Gerencia 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, Servicios e Infraestructura y Conocimiento de la Organización.
- La categoría de Operación está integrada por los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software.

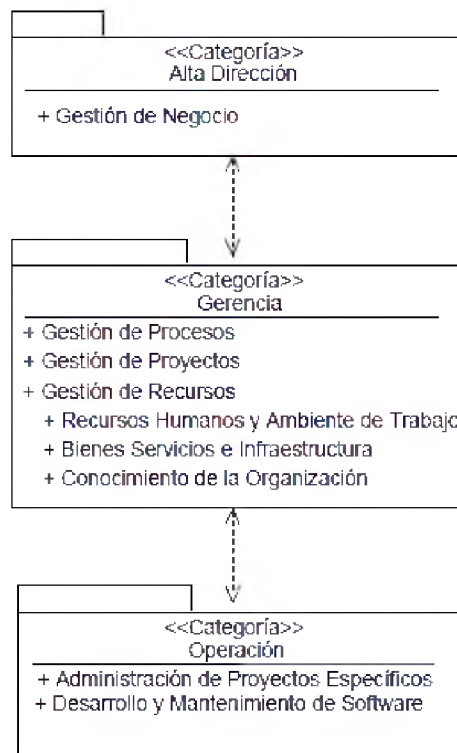


Figura 2.3. Representación gráfica de las categorías de procesos definidas en MoProSoft®

Cada uno de los procesos mostrados en la Figura 2.3 están definidos en términos de los roles responsables por la ejecución de sus prácticas. Es importante mencionar que estos roles se asignan al personal de la organización de acuerdo a sus habilidades para desempeñarlos de forma eficiente. En este sentido, MoProSoft® clasifican los roles como *Grupo Directivo*, *Responsable de Proceso* y *otros roles involucrados*. Además, se consideran tanto al cliente como al usuario como roles externos a la organización. Con el propósito de proporcionar una idea general sobre aquellos aspectos que son cubiertos por MoProSoft®, la Tabla 3 describe brevemente el propósito de cada proceso mostrado en la Figura 2.3.

Tabla 3. Definición de los procesos establecidos por MoProSoft®.

Categoría	Proceso	Propósito
Dirección	Gestión de Negocio	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.
Dirección	Gestión de Procesos	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.
Gerencia	Gestión de Proyectos	Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.
Gerencia	Gestión de Recursos	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.
Gerencia	Recursos Humanos y Ambiente de Trabajo	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.
Gerencia	Bienes, Servicios e Infraestructura	Proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.
Gerencia	Conocimiento de la Organización	Mantener disponible y administrar la base de conocimientos que contiene la información y los productos generados por la organización.
Operación	Administración de Proyectos Específicos	Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
Operación	Desarrollo y Mantenimiento de Software	Realizar sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software (nuevos o modificados) cumpliendo con los requerimientos especificados.

Para implementar este modelo en una organización que no cuente con procesos establecidos ni documentados, se debe 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 los 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 y 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.

En caso de que la organización ya cuente con procesos definidos y/o documentados, se deberá realizar una evaluación de procesos que permita identificar las coincidencias existentes entre los procesos de la empresa y los establecidos por MoProSoft®, y éstos deberán ajustarse al modelo. En este sentido, MoProSoft® cuenta con su propio modelo de evaluación (denominado EvalProSoft®) el cual será descrito a continuación.

2.3. Modelo de evaluación para la industria mexicana de software: EvalProSoft®

El modelo de evaluación de procesos para la industria de software en México (mejor conocido como EvalProSoft®) proporciona a la empresa un nivel de capacidad de procesos (relacionado con las habilidades operativas) y un nivel de madurez de capacidades de la organización (relacionado con las habilidades directivas). EvalProSoft® fue creado con la intención de ser utilizado únicamente en empresas dedicadas al desarrollo y/o mantenimiento de software y, de manera muy particular, en aquellas que utilizan o han utilizado MoProSoft® para conducir una iniciativa de mejora de sus procesos. De acuerdo a sus creadores, el modelo también puede ser utilizado para evaluar los procesos de empresas que no sigan como modelo de referencia a MoProSoft®.

Así, la evaluación de cada proceso implica un examen disciplinado, el cual se debe apoyar en una escala de calificación, criterios de evaluación, conjunto de estándares, mejores prácticas y un mecanismo claro para exponer los resultados obtenidos. La Figura 2.4 muestra que el *evaluador certificado* es el único que tiene la capacidad para llevar a cabo una evaluación siguiendo el proceso definido por EvalProSoft®. El método de evaluación involucra al organismo rector que proporciona el paquete de evaluación, resguarda la información y da seguimiento al método y a la organización a evaluar. Al término del proceso de evaluación, el evaluador debe entregar un reporte estadístico que contiene las evidencias documentadas que respaldan el cumplimiento de un atributo perteneciente al producto y que se verá representado en los resultados finales. Por lo tanto, la empresa debe seleccionar a un evaluador certificado y reconocido por el organismo rector para que éste dirija el proceso de evaluación con base a la información proporcionada. Como resultado del proceso de

evaluación se obtendrá un reporte para la empresa y un reporte estadístico para el organismo rector. El reporte de resultados documentará el perfil del nivel de capacidad de los procesos y el nivel de madurez de capacidades. En el reporte estadístico se detallará también la información general de la empresa que fue evaluada, los resultados y las lecciones aprendidas sobre el modelo de evaluación y su modelo de procesos, MoProSoft®.

En este contexto, existen tres distintos escenarios donde EvalProSoft® puede ser utilizado:

- Evaluación de una empresa: Este escenario se presenta cuando una empresa solicita a un evaluador certificado la aplicación de la evaluación para obtener como resultado un perfil del nivel de capacidades de los procesos y un nivel de madurez de capacidades.
- Evaluación de capacidades de un proveedor: Este escenario sucede cuando un cliente solicita a un evaluador certificado la aplicación de una evaluación para determinar cuál es el mejor proveedor de desarrollo y mantenimiento de software, y
- Autoevaluación de capacidades de procesos: Este escenario asume que una empresa decide aplicar una evaluación utilizando su propio personal o bien personal externo, el cual no debe ser precisamente un evaluador certificado.

Evaluador certificado/Paquete de evaluación

Equipo de evaluación/Información inicial

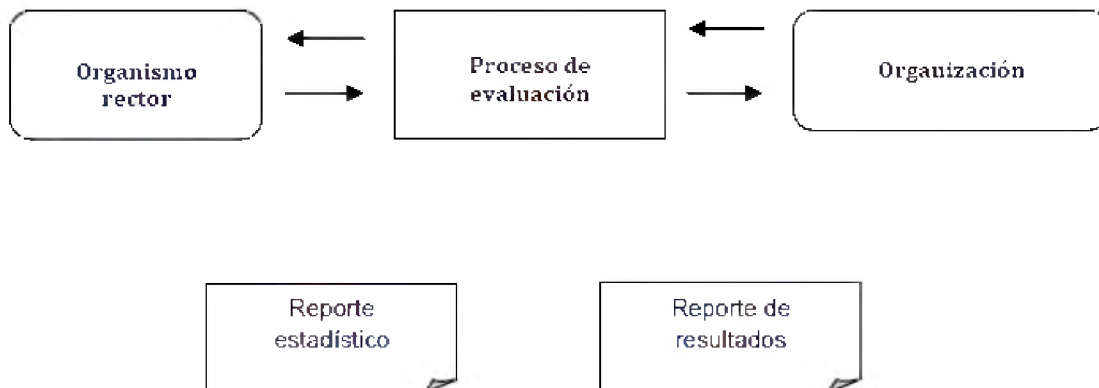


Figura 2.4. Relación entre los elementos de EvalProSoft®

Tal y como se indicó, EvalProSoft® se basa en el modelo de procesos MoProSoft® (como referencia comparativa entre las prácticas establecidas por el modelo y las que la empresa ejecuta actualmente) y en un modelo propio de capacidades que se utiliza para determinar el nivel de capacidad de procesos en el cual se encuentra la empresa. Así, la Figura 2.5 presenta la relación que existe entre los elementos que intervienen en una evaluación conducida por EvalProSoft®.

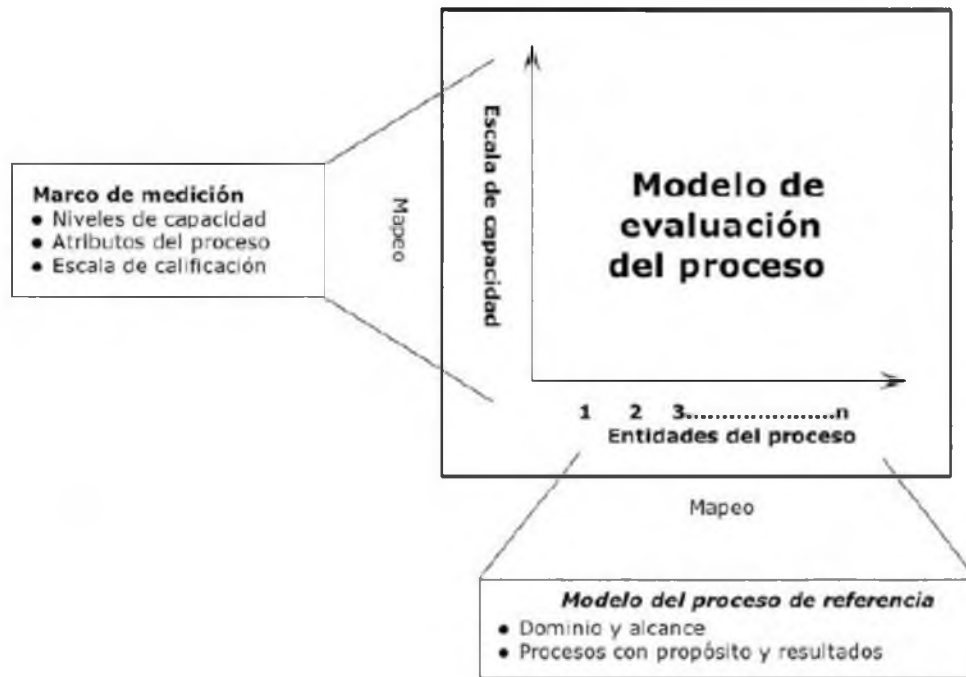


Figura 2.5. Modelo de capacidades establecido por EvalProSoft®

La capacidad de proceso se evalúa en una escala de cinco niveles que van de 0 a 5. El 0 se asocia al nivel más bajo, lo que significa que no alcanza el propósito del proceso, mientras que el 5 se asocia al nivel más alto a través del cual se demuestra el cumplimiento de todas las metas.

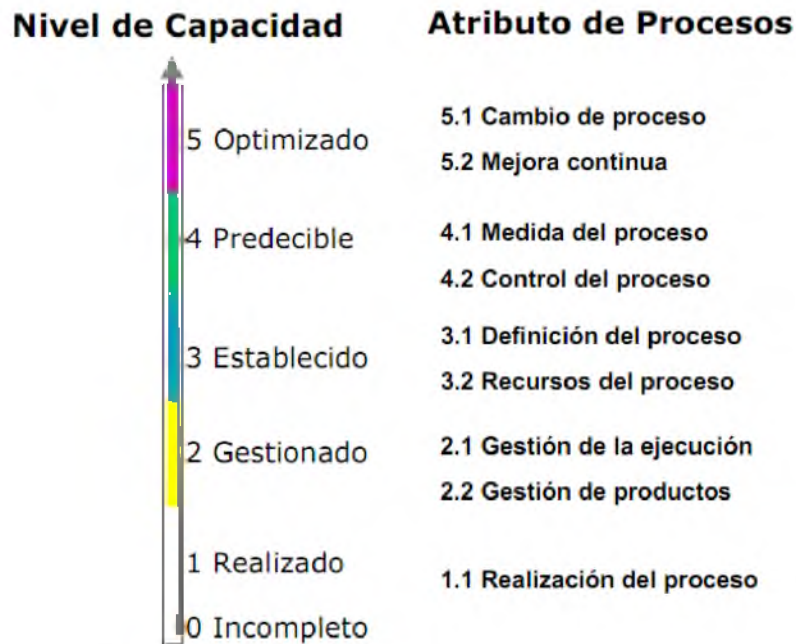


Figura 2.6. Niveles de capacidad definidos por EvalProSoft®

Estos niveles de capacidad se definen de la siguiente manera:

- Nivel 0 (Incompleto): Este nivel implica que el proceso no está implantado y no alcanza el propósito del proceso.
- Nivel 1 (Realizado): Este nivel significa que el proceso implantado logra el propósito fijado.
- Nivel 2 (Administrado): Este nivel reconoce que el proceso realizado se ejecuta de manera administrada y sus productos de trabajo están establecidos, controlados y mantenidos de manera correcta.
- Nivel 3 (Establecido): Este nivel asegura que el proceso realizado y administrado es implementado por medio de pasos definidos ya estandarizados.
- Nivel 4 (Predecible): Este nivel involucra que el proceso establecido opera dentro de límites establecidos, los cuales permiten lograr los resultados esperados.
- Nivel 5 (Optimizado): Este nivel define que el proceso es predecible y se mejora constantemente para lograr sus metas.

Ahora bien, la combinación de las calificaciones de los atributos determina la capacidad de cada proceso para generar los productos, y en conjunto determinan la madurez de la organización al ser evaluada. Las calificaciones de cada proceso se determinan de acuerdo a los siguientes cuatro lineamientos.

- N – No alcanzado. Una cobertura que va del 0% al 15% sobre las prácticas de MoProSoft®. Esto quiere decir que existe poca documentación, que no existe o hay muy poca evidencia sobre el alcance del proceso evaluado en relación al atributo definido.
- P – Parcialmente alcanzado. Una cobertura que va del 16% a un 50% sobre las prácticas de MoProSoft®. Es decir, se afirma que existe una evidencia y un alcance sobre el atributo definido.
- L – Ampliamente alcanzado. Una cobertura que va del 51% al 85% sobre las prácticas de MoProSoft®. Esto implica que existe un enfoque sistemático y un alcance significativo del atributo definido.
- F – Totalmente alcanzado. Una cobertura que va del 86% al 100% sobre las prácticas de MoProSoft®, ya que se tiene una evidencia plena de un enfoque completo y sistemático, y ya se alcanza por completo el atributo definido. Además, no existen debilidades significativas en la realización del proceso.

La Figura 2.7 muestra que formalmente el proceso de evaluación consta de cinco etapas.

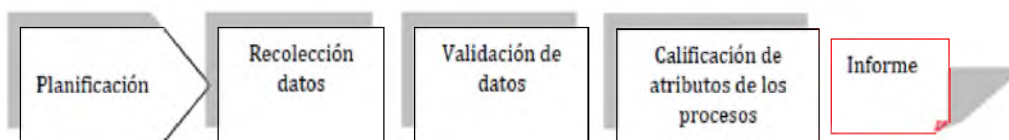


Figura 2.7. Etapas del proceso de evaluación establecido por EvalProSoft®

Las etapas se describen de la siguiente manera:

- **Planificación:** En esta etapa se definen las actividades que se llevarán a cabo para realizar la evaluación. Además, es necesario definir las actividades de cada participante en la evaluación, así como también los criterios para verificar que se han cumplido los requisitos.
- **Recolección de datos:** En esta etapa se recogen los datos y se registran para posteriormente verificar los resultados. Es necesario también establecer una estrategia y técnica para la selección, recolección, análisis de datos y justificación de las calificaciones. Con esto se obtiene una evidencia objetiva para cada proceso evaluado.
- **Validación de datos:** En esta etapa se juntan todos los datos obtenidos para consolidarlos y hacer la validación de los mismos.
- **Calificación de atributos de los procesos:** En esta etapa se le asigna una calificación a cada atributo de los procesos analizados, tomando en consideración los datos ya validados.
- **Informe:** En esta etapa se realiza un informe de la evaluación que indique la situación actual de la organización. Dicho informe es preparado por el evaluador, el cual tiene la obligación de presentarlo ante los participantes y área directiva de la organización. El informe que se entrega a la organización debe contener la siguiente información:
 - Nombre de la organización evaluada.
 - Nombre del promotor y su rol dentro de la organización.
 - Nombre del evaluador certificado, equipo de evaluación y sus roles dentro de la evaluación.
 - Versión del método de evaluación.
 - Procesos evaluados.
 - Fecha de la evaluación.
 - Tabla de perfiles de calificaciones de atributos de cada proceso evaluado.
 - Perfil del nivel de capacidad de los procesos implantados y un nivel de madurez de capacidades. Resumen de hallazgos detectados por cada proceso.
 - Resumen de hallazgos que aplican a varios procesos.
 - Hallazgos que no están directamente relacionados con los requisitos de los procesos, pero que afectan a la implantación.

Adicionalmente, el evaluador certificado elabora y entrega al organismo rector el reporte estadístico de la evaluación que contiene la siguiente información:

- Tipo de evaluación.
- Versión del método de evaluación.
- Fechas de la evaluación.
- Datos de la organización evaluada.
- Unidades administrativas de la organización que fueron evaluadas.

- Datos del promotor.
- Datos del evaluador certificado.
- Datos del representante de la organización.
- Datos del facilitador.
- Equipo de evaluación.
- Participantes entrevistados.
- Resumen de resultados de la evaluación.
- Grado de apego al proceso de evaluación.
- Lecciones aprendidas sobre el método de evaluación y los requisitos de los procesos.
- Documentos a ser enviados al organismo rector.

Como se puede observar, la industria mexicana de software cuenta con un modelo de proceso o referencia (MoProSoft®) y su respectivo modelo de evaluación (EvalProSoft®), por lo que es necesario determinar si existe un modelo de mejora que marque la pauta para implantar cualquiera de los nueve procesos definidos por MoProSoft® con base a la evaluación realizada siguiendo las pautas establecidas por EvalProSoft®.

2.4. Modelo de mejora para implantar MoProSoft®

Tal y como se ha mencionado anteriormente, el modelo MoProSoft® se encuentra definido dentro de la norma mexicana NMX-059-NYCE-2005: Tecnología de la Información-Software - Modelos de procesos y de evaluación para desarrollo y mantenimiento de software, la cual está conformada de la siguiente manera:

- Parte 01: Definición de conceptos y productos.
- Parte 02: Requisitos de procesos (MoProSoft®).
- Parte 03: Guía de implantación de procesos.
- Parte 04: Método de evaluación (EvalProSoft®).

En este sentido, la *Parte 03: Guía de implantación de procesos* describe de manera muy general cómo implementar los procesos descritos anteriormente dentro de aquellas empresas que deseen mejorar su forma habitual de trabajo y, al mismo tiempo, obtener una certificación bajo una norma de calidad. Esta guía hace mención que es necesario que la organización que desea obtener la certificación tenga una buena comprensión de ¿por qué se quiere implementar un modelo?, ¿cuál es el objetivo?, ¿qué utilidad tiene estar certificado bajo una norma?, y demás.

De manera similar, esta guía recomienda que las empresas que desean implantar los procesos desde cero, dispongan de una persona encargada de integrar toda la documentación generada durante la implantación, difundirla y tenerla disponible para su consulta en cualquier momento, asegurándose de que el acceso a ésta sea restringido. En este sentido, dicha guía define, en términos

muy generales, los pasos a considerar cuando se pretende implantar el modelo dentro de una empresa:

1. Dar a conocer a la organización la necesidad de implantar un modelo de calidad; de esta manera, será más fácil la cooperación de todos los involucrados. Es importante que se entienda que la adopción del modelo se relaciona con un proceso de mejora continua y no con un proceso burocrático.
2. Dar a conocer a MoProSoft® dentro de la organización y poner a disposición del personal la documentación del mismo.
3. Definir la estructura del repositorio donde se irá almacenando la información generada. Este repositorio contendrá los formatos, la descripción de los procesos, ejemplos, etc.
4. Definir la forma de nombrar a los archivos y el formato que deben seguir (por ejemplo, tipo de encabezado, pies de página, tipo y tamaño de letra para títulos, subtítulos y texto), numeración dentro de los documentos, etc. De esta forma la información que sea almacenada tendrá un formato uniforme.
5. Una vez que se haya definido cómo generar los documentos, se debe comenzar con el plan estratégico, el cual definirá las metas a alcanzar. La alta dirección de la organización debe ser la primera en implantar este modelo.
6. Definir los procesos de operación de proyectos específicos. Estos procesos, de una u otra forma, se llevan a cabo en la organización para el desarrollo del software y es muy probable que no se tengan documentados o algunas actividades no se realicen. Al realizar este paso ya se estará mejorando el proceso.

Además de estos pasos, la guía argumenta que es necesario que la organización comience la implantación con proyectos pilotos, los cuales ayudarán a definir los procesos necesarios y tienen la finalidad de lograr un mejor entendimiento del por qué se debe documentar cada fase del proceso.

Como se puede ver claramente, la guía que proporciona la norma NMX-059-NYCE-2005 para la implantación del modelo MoProSoft® no ofrece la suficiente ayuda y soporte para que las empresas conduzcan eficientemente una iniciativa de mejora. Aunado a esto, la norma asume que todo el personal de la empresa está capacitado de tal forma que le será fácil entender cada una de las prácticas recomendadas. Sin embargo, la realidad es otra puesto que es común que sin la ayuda correcta todos los niveles de la organización (alta dirección, gerencial y operativo) tengan problemas para realizar la implantación de MoProSoft®. En este sentido, es verdad que existe literatura que hace referencia al uso del modelo IDEAL (McFeeley, 1996) como guía a seguir para establecer iniciativas de mejora en pequeñas empresas (e.g., Kautz, Hansen, & Thaysen, 2000; Guerrero & Eterovic, 2004; Pino, Alegría, Vidal, García, F., & Piattini, 2009), pero estas experiencias documentan esfuerzos centrados en otros modelos de proceso como CMM (e.g., Guerrero & Eterovic, 2004), CMMI (e.g., García et al., 2010; Lester, Wilkie, McFall, & Ware, 2010), y CMMI y TSP (e.g., Serrano, Montes de Oca, & Cedillo, 2006). Sin embargo, no existe evidencia documentada de que IDEAL pueda ser utilizado en la implantación de MoProSoft®, y aunque así fuera este modelo de mejora fue creado por el SEI para conducir iniciativas relacionadas con los modelos creados para las grandes empresas no para las pequeñas, por lo que su uso requeriría de conocimiento sólido previo.

En este sentido, en México se han desarrollado algunas alternativas que pretenden agilizar el tiempo de implantación y reducir, al mismo tiempo, la complejidad de la iniciativa de mejora cuando se intenta seguir las prácticas marcadas por MoProSoft®. Así, la siguiente sección de este documento presenta una revisión detallada de aquellas propuestas que se relacionan con la creación de métodos para facilitar la adopción de la norma. Al final de esta revisión se presentará un análisis sobre las ventajas y desventaja de cada una de las propuestas.

2.5. Estado del arte

Con el objetivo de determinar el avance de la investigación relacionada con el desarrollo de un modelo o marco de trabajo que permita guiar a las MiPyMEs desarrolladoras de software en la implantación del modelo MoProSoft®, a continuación se presenta una revisión de los principales estudios realizados en diferentes universidades de México. Con esto se pretende establecer un *benchmarking*⁵ y determinar aspectos que podrían ser útiles en el planteamiento de la solución propuesta en esta tesis.

2.5.1. Manejador de documentos de MoProSoft® (MDM)

2.5.1.1. Objetivo

En el 2005 se desarrolló una investigación con el propósito de auxiliar a las empresas a lograr una implantación exitosa de MoProSoft® (Caballero, 2005). Dicha investigación fue presentada como tesis para obtener el título de Licenciatura en Ingeniería en Sistemas Computacionales en la Universidad de las Américas Puebla (UDLA) en México. El principal producto desarrollado con esta tesis consistió de una aplicación llamada Manejador de Documentos de MoProSoft® (MDM) que planteó la automatización de la Base de Conocimientos que la norma establece como elemento principal de la implantación. Así, el objetivo consistió en definir un mecanismo computacional para ayudar a las empresas mexicanas a documentar concretamente los procesos de Gestión de Negocio (GN), Gestión de Procesos (GP) y Administración de Proyectos Específicos (APE) que forman parte de MoProSoft®.

2.5.1.2. Descripción de la propuesta

De acuerdo con su creadora, MDM consiste de un módulo a través del cual cada uno de los usuarios, dependiendo del rol asignado en el sistema, puede crear, guardar y/o consultar documentos relacionados con los procesos de MoProSoft®. Es importante mencionar que el acceso a este manejador se realiza a través de una página de Internet. Los documentos relacionados con los procesos de GN y GP, únicamente son controlados al inicio de un nuevo proyecto; mientras que los documentos involucrados en APE son gestionados a lo largo del proyecto. De esta forma, un usuario puede seleccionar, al iniciar un proyecto, el documento que requiere utilizar/consultar y el sistema le presentará la plantilla correspondiente en formato Microsoft Word®.

⁵ 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 que se reconocen como representantes de las mejores prácticas (Spendolini, 1992). El análisis *benchmarking* en esta tesis se enfoca en analizar comparativamente las investigaciones realizadas previamente con el objetivo de mejorar la solución propuesta.

Adicionalmente, MDM define roles que restringen el acceso a los procesos, es decir, un usuario solamente podrá acceder al proceso asignado en el proyecto. En este sentido, únicamente el usuario administrador tendrá acceso a los tres procesos cubiertos, y podrá dar de alta, modificar y dar de baja a otros usuarios y a proyectos a través de un módulo administrativo. Con el objetivo de ampliar la explicación sobre el funcionamiento de MDM, a continuación se presentan algunos diagramas de secuencia que ejemplifican su dinámica de trabajo. Por ejemplo, la Figura 2.8 describe la interacción y secuencia cronológica de los objetos relacionados con los procesos de GN y GP. En este sentido, los objetos definidos para ejecutar estos procesos son: sistema, usuario, proceso, empresa, y documento. La figura muestra que inicialmente el usuario debe ingresar al sistema proporcionando sus datos de ingreso. En caso de que esta verificación resulte incorrecta la sesión terminará, si los datos de ingreso son correctos el usuario podrá acceder al proceso en el que va a trabajar. Dependiendo del proceso al que ingrese, el usuario podrá acceder a las funcionalidades para crear, consultar o guardar documentos. Por último, la sesión terminará cuando el usuario lo decida.

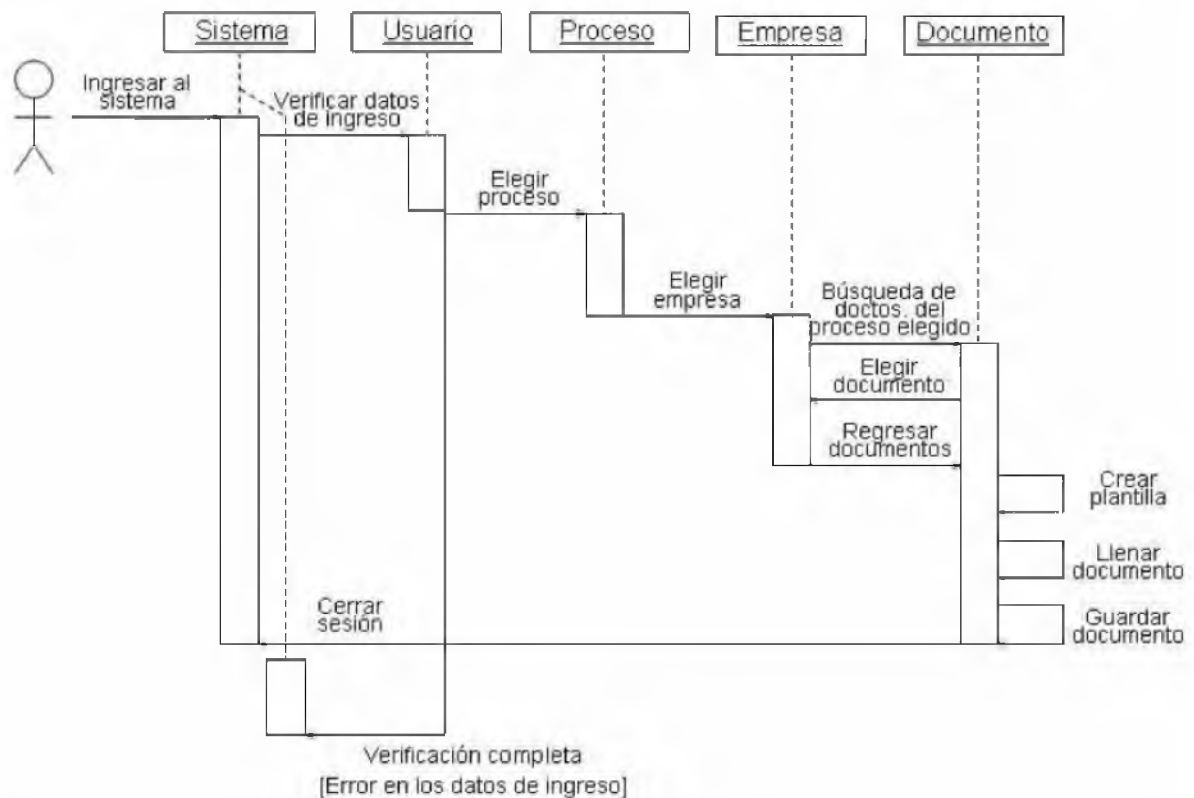


Figura 2.8. Diagrama de secuencia de los procesos de Gestión de Negocios y Gestión de Procesos (Caballero, 2005)

De manera similar, los objetos que interactúan en el proceso de APE son: sistema, usuario, empresa, proyecto, y documento (véase Figura 2.9). Se observa que en un principio el usuario accede al sistema proporcionando sus datos de ingreso y posteriormente elige la empresa a la que pertenece. De esta forma este usuario obtiene acceso a los proyectos vinculados con dicha empresa y, obviamente, en los que puede trabajar. Dependiendo del proyecto que el usuario elija éste puede crear documentos o consultar y guardar versiones modificadas. Finalmente, cuando el usuario decida cerrar el sistema terminará su sesión de trabajo.

La implementación de MDM utilizó Tomcat como servidor de la página web, y fue desarrollado sobre el lenguaje de programación Java. Adicionalmente, la creación de las pantallas se realizó con el editor de HTML Dreamweaver y Flash ambos de Macromedia Adobe®. Para almacenar la información se utilizó el gestor de base de datos MySQL.

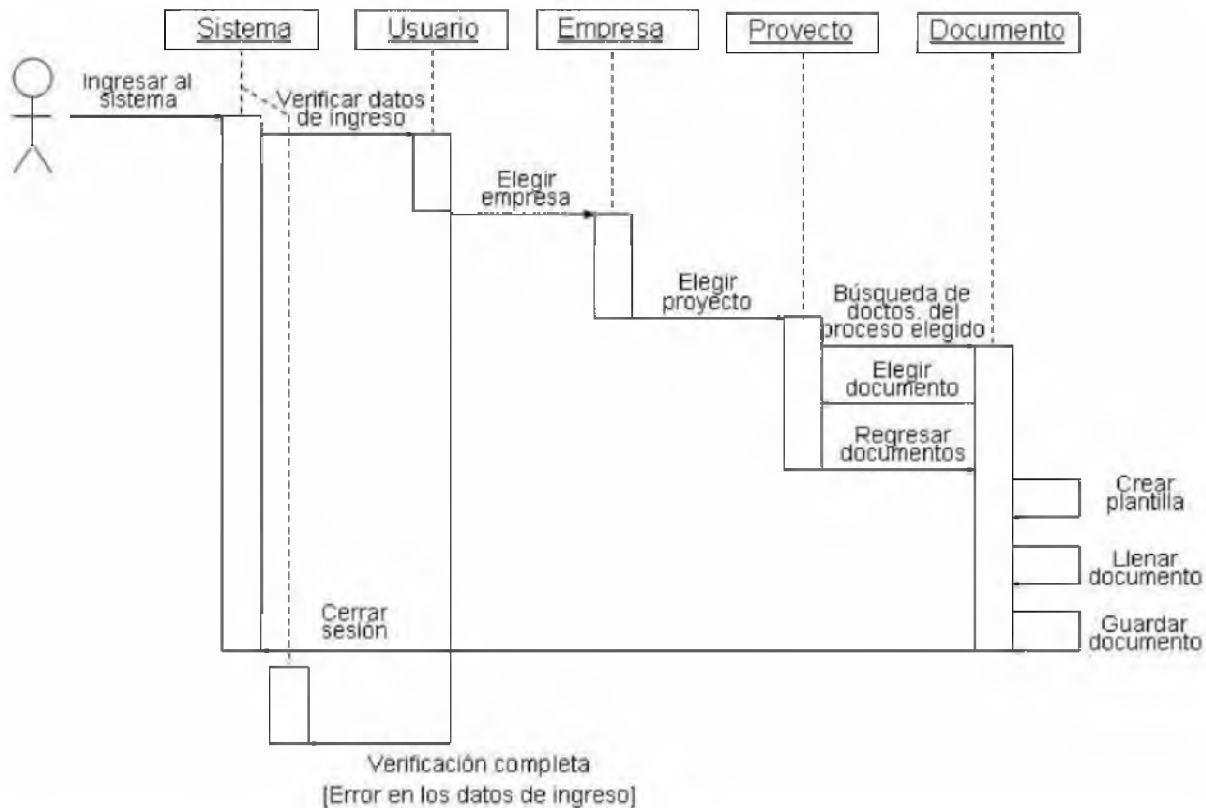


Figura 2.9. Diagrama de secuencia del proceso de Administración de Proyectos Específicos (Caballero, 2005)

En este sentido, las dos características más importantes de MDM son:

1. El sistema maneja los documentos de MoProSoft® de tal forma que si este modelo de procesos sufre alguna alteración o se liberan nuevas versiones, basta aumentar un componente en el sistema.
2. MDM permite a sus usuarios crear documentos, modificarlos, compartirlos, crear reportes, y administrarlos. Esto incluye controlar accesos, modificaciones y autorizaciones de impresión.

2.5.1.3. Resultados alcanzados

MDM fue utilizado por diez estudiantes dentro del curso “Administración de la Calidad del Software”. Estos estudiantes, que poseían conocimientos previos sobre la norma mexicana NMX-I-059/02-NYCE-2005, fueron los encargados de probar las funcionalidades del sistema y respondieron una encuesta para obtener resultados de tipo cualitativo. De acuerdo a la creadora de

MDM, la técnica de evaluación basada en cuestionarios siguió los lineamientos establecidos por Nielsen (1999), los cuales evalúan de forma concreta lo siguiente:

- Visibilidad del estado del sistema: Mide si el usuario siempre sabe qué está haciendo el sistema.
- Similitud entre el sistema y el mundo real: Mide si la aplicación se expresa de una manera comprensible para el usuario.
- Control y libertad del usuario: Mide si los usuarios que se equivocan al hacer algo, tienen forma de recuperarse de esos errores.
- Prevención de errores: Permite validar si se cuenta con mecanismos que aseguren que el ingreso de cualquier información por parte del usuario, permitan evitarle errores. Para ello, se verifica si en las áreas en que los usuarios deben interactuar con el sistema, se les explica claramente lo que se espera de ellos.
- Preferencia al reconocimiento que a la memorización: Permite validar si la aplicación ayuda al usuario a recordar cómo se hacía una operación, o bien le obliga a aprenderse los pasos cada vez que ingresa.
- Flexibilidad y eficiencia de uso: Permite validar si la aplicación es eficiente teniendo en cuenta el objetivo para el que fue desarrollada.
- Estética y diseño minimalista: Pide que los elementos que se ofrezcan en la pantalla correspondan a una buena razón para estar presentes. Así pues, se verifica la existencia de elementos irrelevantes (texto, sonido e imagen), que no aportan ni ayudan a que el usuario distinga lo importante de lo superfluo.

Siguiendo estos lineamientos, la aplicación del cuestionario arrojó los resultados mostrados en las Tablas 4 y 5.

Tabla 4. Resultados sobre la utilidad del Manejador de Documentos de MoProSoft® (Caballero, 2005).

Preguntas	Sí	No
1.- La herramienta es consistente con el objetivo de la norma MoProSoft®.	10	0
2.- Las pantallas del sistema contienen la información necesaria para completar las tareas que se requieren.	10	0

Tabla 5. Resultados sobre la evaluación cualitativa del Manejador de Documentos de MoProSoft® (Caballero, 2005).

Preguntas	E	B	R	M	N
3 ^a .- Claridad de MDM al informar sobre operaciones fallidas o exitosas.	8	2	0	0	0
3b.- Pertinencia del sistema MDM en el momento de los mensajes.	9	1	0	0	0
4.- Manejo de errores del sistema MDM.	9	1	0	0	0
5.- Calificación del manual de usuario.	8	2	0	0	0

6.- Facilidad que da el sistema al usuario para recordar los procedimientos para llevar a cabo las funciones del sistema.	7	3	0	0	0
7.- Eficiencia del sistema para la creación, consulta y almacenamiento de documentos.	10	0	0	0	0

E = Excelente, B = Bueno, R = Regular, M = Malo, N = No sé

Observando las tablas se puede determinar que, de acuerdo a los estudiantes que participaron en la evaluación, el sistema de MDM cumple con lo propuesto en la norma y proporciona información adicional que puede ayudar a las empresas a documentar continuamente cada una de las actividades que se requieren durante su implementación. Sin embargo, estos resultados debieron ser sustentados con un caso de estudio en un entorno real de tal forma que fuera posible afirmar que su uso reduce la complejidad de la implementación en MiPyMEs.

2.5.2. Implementación de MoProSoft® en una empresa dedicada al desarrollo de software

2.5.2.1. Objetivo

De manera similar que la tesis analizada anteriormente, en el 2008 se realizó un estudio relacionado con la implementación de MoProSoft® en una empresa Veracruzana (Barradas, 2008). Este trabajo fue presentado como tesis para obtener el título de Licenciado en Administración en la Universidad Veracruzana en México. Dicho estudio presentó a MoProSoft® como un modelo que permite alinear los procesos con los objetivos y estrategias de la empresa, promoviendo cambios al diseño de las organizaciones, introduciendo mejores prácticas y estandarizando los procesos, con el fin de incrementar la competitividad de las empresas. En este sentido, se definió la importancia que un modelo como MoProSoft® tiene para reflejar cambios significativos en los procesos de las empresas que se dedican al desarrollo de software y de esta forma adoptar mejores prácticas para alcanzar altos niveles de competitividad.

2.5.2.2. Descripción de la propuesta

De acuerdo a lo descrito en (Barradas, 2008), la empresa en la cual se realizó la implementación de MoProSoft® fue fundada en febrero del 2002 con el objetivo de ofrecer soluciones (productos y servicios) de TI a través de la obtención paulatina de niveles internacionales de calidad. Así, la Figura 2.10 muestra la estructura organizativa definida antes de la implementación, en ésta es posible observar que la empresa está compuesta por los departamentos de Mercadotecnia, Ventas, Soporte Técnico y Administración de Proyectos.

Sin embargo, antes de iniciar con el proceso de adaptación fue necesario que la empresa tomara en cuenta los siguientes recursos:

- Materiales: Se requirió un ejemplar de la norma NMX-I-059-NYCE-2005.
- Humanos: No fue necesario contratar otros recursos humanos, ya que las personas que participaron en la implementación ya laboraban en la empresa.
- Tecnológicos: Al personal que participó en la implementación del modelo se le asignó una computadora laptop para uso personal, una herramienta para manejo de documentos y control de versiones (Microsoft Visual SourceSafe), herramienta para redacción de documentos (Microsoft Office Word), herramienta para la planeación de proyectos (Microsoft Project), herramienta para el desarrollo de software (Microsoft Visual Basic),

herramienta para cálculos (Microsoft Office Excel), y un método uniforme para administrar las actividades del proyecto: la Estructura Desglosada de Trabajo (WBS, por sus siglas en inglés).

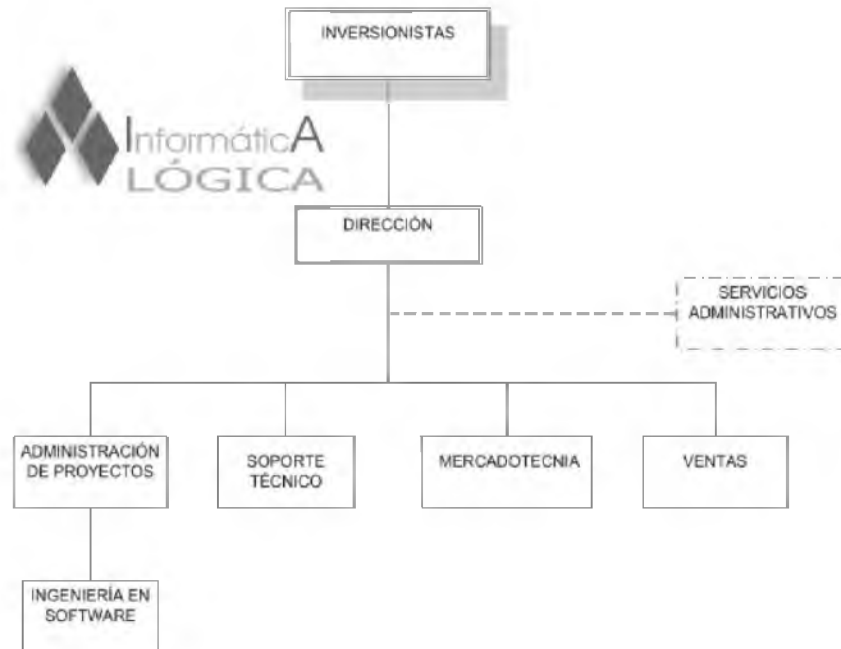


Figura 2.10. Estructura organizativa de Informática Lógica S. C. (Barradas, 2008)

De manera adicional, la empresa realizó un plan de trabajo en donde se programaron las siguientes actividades a realizar:

- **Capacitación:** Fue necesario que las personas responsables de los procesos asistieran a un taller donde recibieron capacitación y explicaciones sobre éstos y los conceptos que manejan, así como de los productos de trabajo que se generan como resultado de cada actividad.
- **Diseño:** Se elaboraron las plantillas de los productos generados como resultados de cada una de las actividades correspondientes a cada proceso. En esta etapa las personas involucradas ayudaron a determinar qué información debería contener cada plantilla.
- **Pruebas:** Se realizaron pruebas sobre los documentos y procesos generados, siguiendo lo establecido en la Norma NMX-I-059-NYCE-2005.
- **Ajustes:** Una vez realizadas las pruebas, los responsables de los procesos observaron la problemática que se presentó durante la prueba de procesos y documentos, y se definieron los ajustes necesarios para iniciar con la implementación del proceso.
- **Implantación:** Se realizaron ajustes a los procesos definidos previamente, tales ajustes correspondieron a las necesidades específicas de la empresa.
- **Implantación de procesos de soporte:** Se utilizaron los procesos de soporte con el objetivo de ayudar a la empresa a contar con todo lo necesario para realizar correctamente las actividades.

Así, a cada persona se le asignó un rol considerando los procesos definidos por MoProSoft®. Esta tarea fue realizada mediante una reunión de trabajo en la cual estuvieron presentes todos los involucrados en la inicia de mejora, y mediante la firma de una minuta se comprometieron a realizar las actividades del proceso. En este sentido, la capacitación fue genérica y específica puesto que intentó cubrir aquellos aspectos de aplicación general y temas más relevantes en cuanto a la ejecución de cada actividad de los procesos. De esta manera fue necesario que la empresa identificara todas las entradas y salidas que se debían contemplar por cada actividad del modelo, quién debería ser su responsable o qué rol debería ejecutarla. La Figura 2.11 muestra el diagrama que resume dicha información.

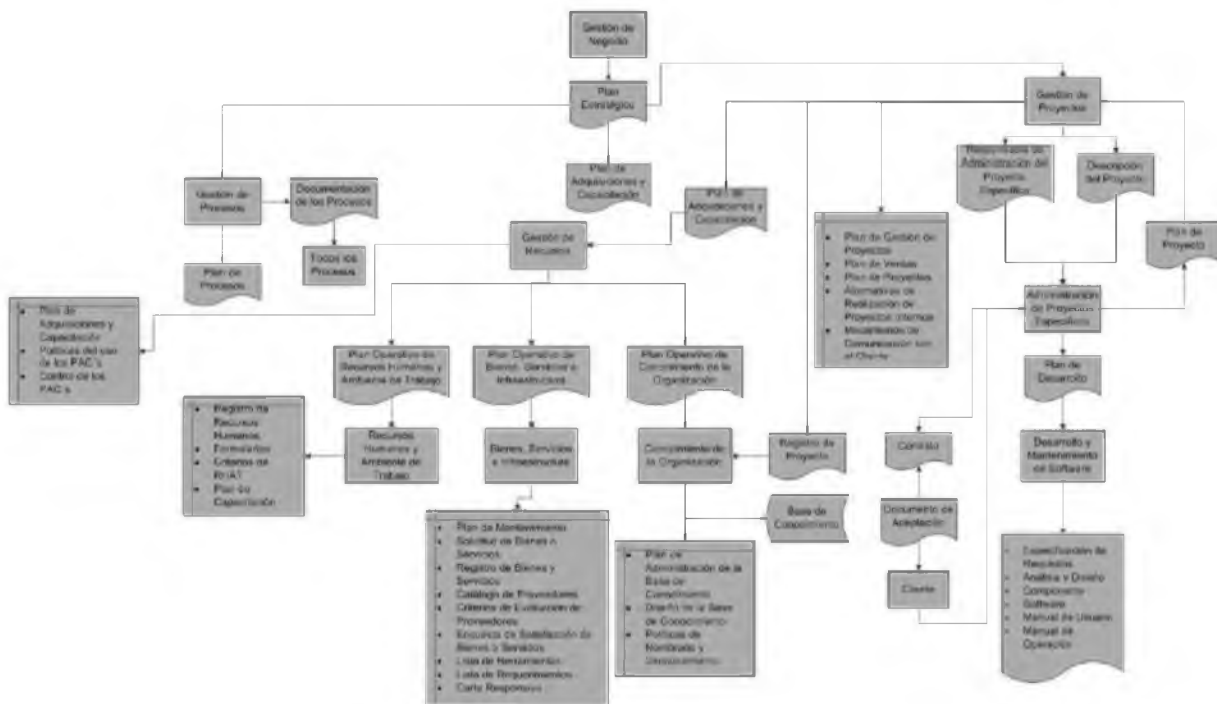


Figura 2.11. Flujograma de productos de trabajo entre procesos (Barradas, 2008)

2.5.2.3. Resultados alcanzados

Una vez realizado el proceso de adaptación de MoProSoft® dentro de la empresa, ésta adquirió nuevas formas de trabajo para el desarrollo y mantenimiento de sus productos de software que le aseguran que cada uno de éstos posee la calidad necesaria para salir al mercado. Así pues, el nuevo proceso se basa en las buenas prácticas de un modelo de calidad. No obstante, como consecuencia del cambio, la empresa cambió totalmente su estructura organizacional, logrando con esto que cada proceso relacionado con el desarrollo de software se realice por una persona diferente, asignando así responsabilidades a cada miembro de la organización (véase Figura 2.12). Es decir, cada uno de los integrantes del equipo de desarrollo ahora cuenta con los conocimientos necesarios para realizar sus actividades, y han logrado la absoluta comprensión de sus roles y responsabilidades.

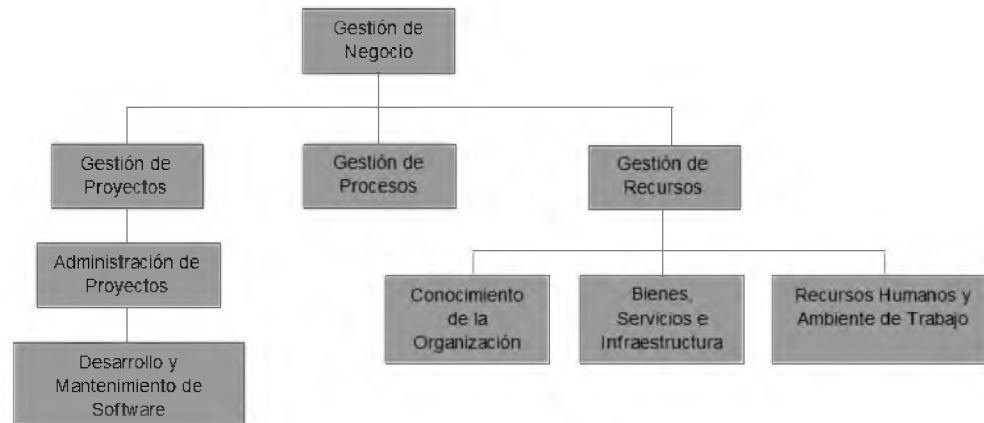


Figura 2.12. Estructura organizacional resultante con la implementación de MoProSoft® (Barradas, 2008)

A continuación se muestra el contenido de cada uno de los procesos (en términos de actividades y documentos) que la empresa decidió implementar después de realizar el análisis y las pruebas piloto.

- **Gestión de Negocio (GN).**

Las actividades realizadas para este proceso fueron:

- Planeación estratégica.
- Preparación para la realización.

Los documentos necesarios para realizar estas actividades fueron:

- Plan Estratégico.
- Plan de Adquisiciones y Capacitación.

- **Gestión de Procesos (GP).**

Las actividades realizadas para este proceso fueron:

- Planificación.
- Preparación para la implantación.

Los documentos necesarios para realizar estas actividades fueron:

- Plan de Procesos.
- Documentación de los 9 Procesos Normativos.

- **Gestión de Proyectos (GPY).**

Las actividades realizadas para este proceso fueron:

- Planificación.
- Realización.

Los documentos necesarios para realizar estas actividades fueron:

- Plan de Adquisiciones y Capacitación.
- Plan de Gestión de Proyectos.
- Plan de Ventas.
- Plan de Proyectos.
- Alternativas de Realización de Proyectos Internos.
- Contrato.
- Registro de Proyecto.
- Descripción del Proyecto.

Además de que fue necesario crear el siguiente rol:

- Responsable de Administración del Proyecto Específico.
- Gestión de Recursos (GR).

La actividad realizada para este proceso fue:

- Planificación.

Los documentos necesarios para realizar esta actividad fueron:

- Plan Operativo del proceso de Recursos Humanos y Ambiente de Trabajo.
- Plan Operativo del proceso de Bienes, Servicios e Infraestructura.
- Plan Operativo del proceso de Conocimiento de la Organización.
- Plan de Adquisiciones y Capacitación (PAC).
- Naturaleza y Especificidad de los PAC's.
- Recursos Humanos y Ambiente de Trabajo (RHAT).

Las actividades realizadas para este proceso fueron:

- Preparación.
- Instrumentación.

Los documentos necesarios para realizar estas actividades fueron:

- Plan de Capacitación.
- Registro de Recursos Humanos.

- Bienes, Servicios e Infraestructura (BSI).

Las actividades realizadas para este proceso fueron:

- Preparación.
- Instrumentación.

Los documentos necesarios para realizar estas actividades fueron:

- Plan de Mantenimiento.
- Solicitud de Bienes o Servicios.

- Registro de Bienes o Servicios.
- Catálogo de Proveedores.
- Criterio de Evaluación de Proveedores.
- Encuesta de Satisfacción de Bienes o Servicios.
- Conocimiento de la Organización (CO).

Las actividades realizadas para este proceso fueron:

 - Preparación.
 - Instrumentación.

Los documentos necesarios para realizar estas actividades fueron:

 - Plan de Administración de la Base de Conocimiento.
 - Diseño de la Base de Conocimiento.
 - Base de Conocimiento.
 - Políticas para nombrado y control de versiones.
- Administración de Proyectos Específicos (APE).

Las actividades realizadas para este proceso fueron:

 - Planificación.
 - Realización.
 - Cierre.

Los documentos necesarios para realizar estas actividades fueron:

 - Plan de Desarrollo.
 - Plan de Proyecto.
 - Documento de Aceptación.
- Desarrollo y Mantenimiento de Software (DMS).

Las actividades realizadas para este proceso fueron:

 - Realización de la fase de inicio.
 - Realización de la fase de requerimientos.
 - Realización de la fase de análisis y diseño.
 - Realización de la fase de construcción.
 - Realización de la fase de integración y pruebas.

Los documentos necesarios para realizar estas actividades fueron:

 - Especificación de Requerimientos.
 - Análisis y Diseño.
 - Componente.

- Software.
- Manual de Usuario.
- Manual de Operación.

Por último, este trabajo menciona que, partiendo de la premisa de que MoProSoft® permite alinear los procesos con los objetivos y las estrategias de la empresa, la iniciativa de mejora realizada facilitó que la empresa en cuestión obtuviera la primera certificación en el Estado de Veracruz. De esta manera, la empresa obtuvo una posición competitiva preferencial y serviría como referente para las demás empresas del Estado que se orientan al mismo ramo.

2.5.3. Caso de estudio de mejora continua de procesos de software en una PYME mexicana

En el año 2009 se realizó una tesis más que desarrolló un caso de estudio enfocado a una PyME mexicana enfocada al desarrollo de software, denominada UTR, con la finalidad de introducir un marco de referencia basado en los beneficios obtenidos en dicha empresa (García, 2009). Este caso de estudio fue presentado como tesis para obtener el grado de Maestro en Ingeniería en la en la Universidad Nacional Autónoma de México (UNAM). El objetivo de esta tesis consistió en determinar un grupo de factores de éxito para la mejora continua (a largo plazo) de los procesos de UTR, mediante el análisis de la información y los datos recogidos en la empresa.

2.5.3.1. Descripción de la propuesta

Para el desarrollo del caso de estudio se utilizó un método cualitativo de investigación, el cual se caracteriza por realizar un trabajo metódico y bien planificado para obtener resultados útiles y significativos. Dicha investigación siguió los pasos de concepción, diseño, preparación, ejecución, análisis y diseminación, y toma de decisiones. Esta metodología permitió identificar los beneficios obtenidos y los elementos principales que han sido mejorados en UTR durante un periodo de diez años. A continuación se explica brevemente la aportación de cada uno de estos pasos al contexto de la tesis desarrollada.

Concepción: Este paso requirió de una investigación bibliográfica relacionada con casos de estudio relacionados con la Ingeniería de Software en el contexto de una PyME. Una vez realizada la investigación, se logró obtener información relevante que facilitó el entendimiento del estudio, además de que fue posible establecer un marco complementario de comparación que facilitó la toma de decisiones.

Concretamente la concepción permitió la obtención de las siguientes premisas que facilitaron la focalización del problema:

1. Las PyMEs no son una representación a menor escala de las grandes empresas.
2. Debido a su tamaño, estas empresas pueden ser más vulnerables a factores externos, como la situación económica del país donde se encuentran, las decisiones de los clientes, las situaciones con socios, etc.
3. Las decisiones del(los) dueño(s) son fundamentales para la supervivencia de la empresa.
4. No todas las PyMEs desean convertirse en “mega empresas”.

5. Los conceptos de “éxito” o “fracaso” no necesariamente significan lo mismo para las PyMEs como para las empresas grandes.

Dentro de este paso se realizaron ciertas actividades que ayudaron a conducir una entrevista preliminar que ayudó a establecer un marco de referencia para definir la pregunta base de investigación, la cual está orientada hacia la descripción de procesos. Esta pregunta fue: ¿Cuáles son los factores que han hecho de UTR una PyME de Ingeniería de Software exitosa en México?

Diseño: Una vez que el objetivo de la investigación fue establecido de manera clara y antes de encontrar datos empíricos por primera vez, el siguiente paso consistió en la generación de un caso de estudio que proporcionaría las respuestas a la pregunta clave y las premisas establecidas.

En este sentido, para planificar la entrevista se tomó en consideración la simplicidad y la disponibilidad del personal de UTR. Así, fue posible diseñar dos tipos de entrevistas por separado: la entrevista formalmente definida como “Entrevista Piloto”, que proporcionaría la evidencia inicial y la cual ayudaría a realizar otra entrevista llamada “Entrevista Retrospectiva”, cuya finalidad era validar, confirmar y complementar la información obtenida en la entrevista piloto. Cada entrevista siguió los siguientes objetivos:

1. Entrevista piloto: Obtener la mayor cantidad de información básica sobre UTR relacionada con:
 - a. Sus orígenes.
 - b. Su estado actual.
 - c. Su transición desde el origen hasta su estado actual.
 - d. Sus hitos más importantes en la historia desde su creación.
2. Entrevista retrospectiva: Obtener información específica de los diferentes puntos de vista del resto de los participantes sobre los aspectos más sobresalientes de las entrevistas piloto.

Preparación: Este paso se enfoca en la preparación de la entrevista piloto con el objetivo de recoger información base que permitiera identificar el contexto actual de la empresa.

La preparación se enfocó en las siguientes tres actividades.

1. Presentación.

Sección 1. Presentación:

- a. Presentación e identificación del entrevistador.
- b. Breve descripción del objetivo del estudio.
- c. Exposición del objetivo de la entrevista piloto.

- d. Beneficios para UTR con la investigación.
- e. Información para establecer contacto con el entrevistador.

Sección 2. Instrucciones Generales:

- a. Descripción general de las secciones de la entrevista piloto.
- b. Identificación de preguntas opcionales.

2. Formulación de preguntas:

Sección 3. Antecedentes:

- a. Indicaciones para responder la sección de preguntas.
- b. Preguntas referentes a información previa a la creación de UTR (procedencia de los fundadores de UTR).
- c. Preguntas referentes al momento de la creación de UTR.
 - Motivación para la creación de UTR.
 - Primeros socios de UTR.
 - Manera de financiar la creación de la empresa.
 - Preparación profesional de los principales creadores de UTR.

Sección 4. Preguntas generales:

- a. Respecto a los productos/servicios que ofrece y ha ofrecido UTR.
- b. Respecto al tipo de clientes que maneja UTR.
- c. Respecto al enfoque de la calidad que maneja UTR.
- d. Respecto a la asesoría externa que recibe UTR.

Sección 5. Hitos:

- a. Pregunta abierta acerca de los eventos más sobresalientes en la historia de UTR (enfrentamiento de retos importantes y aprovechamiento de oportunidades sobresalientes).

3. Evaluación del instrumento de recolección de información:

Sección 6. Evaluación de la entrevista:

- a. Claridad y amigabilidad del instrumento (estructura, facilidad para responder las preguntas, manejo del lenguaje, manejo de instrucciones).
- b. Problemas generales del entrevistado respecto al instrumento.
- c. Opiniones adicionales del entrevistado.

Ejecución: Una vez que la entrevista fue preparada, el siguiente paso se enfoca en la conducción tanto de la entrevista piloto como de la entrevista retrospectiva.

La entrevista piloto fue realizada bajo supervisión y tuvo una duración de 90 minutos. Al finalizar, los resultados de la entrevista fueron transcritos, analizados y evaluados para de esta forma dar entrada a la segunda entrevista que duró 150 minutos. La preparación de la entrevista retrospectiva consistió de las siguientes actividades:

1. Presentación:

Sección 1. Instrucciones: Las preguntas en la entrevista se manejaron en un mismo formato, de esta manera no fue necesario especificar instrucciones para cada sección.

2. Preguntas:

Sección 2. Recursos humanos (personal y capacitación):

- a. Preguntas referentes a los cambios en cantidad de personal.
- b. Preguntas referentes a la importancia que han tenido capacitaciones específicas en UTR.

Sección 3. Recursos físicos y herramientas:

- a. Cambios en los recursos físicos e infraestructura durante la historia de UTR.
- b. Información de las herramientas más importantes que utiliza UTR y en los tipos de clientes que maneja la empresa.

Sección 4. Recursos económicos:

- a. Cambios en las fuentes de financiamiento de UTR y en los tipos de clientes que maneja la empresa.
- b. Crecimiento económico de UTR.

Sección 5. Oportunidades de mercado y perspectivas de negocio:

- a. Cambios en las oportunidades de mercado.
- b. Papel que desempeña la innovación y la vanguardia en UTR.
- c. Información acerca de las cuatro perspectivas de negocio a las que se enfoca UTR.

Sección 6. Estandarización:

- a. Cambios en la importancia que UTR le otorga a la estandarización de los procesos.
- b. Cambios en los elementos principales a los que se ha enfocado UTR para la estandarización de los procesos.

Sección 7. Situaciones extraordinarias: Comportamiento de UTR ante situaciones que no correspondían al entorno acostumbrado.

- a. Devaluación del peso mexicano en 1995.
- b. Experiencia con licitación de General Electric en 1995.

- c. Lecciones aprendidas de estas experiencias.

Sección 8. Evaluaciones:

- a. Generación de la idea.
- b. Preparación para la primera evaluación.
- c. Incorporación de MoProSoft®.
- d. Evaluación más reciente (2008): CMMI 1.2 Nivel 4.

- 3. Evaluación del instrumento de recolección de información:

Sección 9. Evaluación de la entrevista:

- a. Claridad y amigabilidad del instrumento.
- b. Problemas generales del entrevistado respecto al instrumento.
- c. Opiniones adicionales del entrevistado.

Análisis y diseminación: Una vez obtenida la información necesaria mediante las entrevistas, los resultados fueron plasmados mediante un análisis.

En este sentido, las respuestas de las entrevistas fueron analizadas para encontrar puntos comunes o de complemento, y la información fue interpretada de acuerdo a los criterios que se aprendieron durante el desarrollo de la investigación. Una vez que se terminó el proceso de revisión, los datos fueron agrupados en nuevas categorías correspondientes a periodos que describían la evolución de UTR.

2.5.3.2. Resultados alcanzados

El análisis de las respuestas permitió conocer la evolución histórica de la empresa UTR, en la cual se pueden observar las oportunidades de negocio que han contribuido para su consolidación en el mercado y los diferentes procesos que han seguido para alcanzar el éxito. Los principales hallazgos resumen que UTR ha sido una empresa que se ha preocupado por las habilidades de su recurso humano, por lo cual lo ha sometido a diferentes capacitaciones para aumentar su conocimiento en el área correspondiente a la Ingeniería de Software dentro de la empresa. Además, se pudo determinar que durante la historia de la empresa ésta ha experimentado un periodo importante de crecimiento y maduración. En el momento en que se realizó este caso de estudio, UTR era una empresa que contaba con un Nivel 4 de madurez en CMM y CMMI 1.2; sin embargo se observó que inició con la incorporación de MoProSoft® y a través de la constante capacitación en diferentes ámbitos relacionados con calidad del software, administración de proyectos de software, administración de requerimientos, administración de riesgos en los proyectos de software, procesos de desarrollo de software, entre otros, logró abrir sus posibilidades de éxito en el mercado internacional a través de un modelo como CMMI.

2.5.4. Propuesta de una guía para interpretar los procesos de MoProSoft® con métodos ágiles

2.5.4.1. Objetivo

En el año 2010 se diseñó la propuesta de una guía para facilitar la interpretación de los procesos de MoProSoft®, específicamente de la categoría de Operación, usando una combinación de métodos ágiles (Rueda, 2010). Esta propuesta fue presentada como tesis para obtener el grado de Maestro en Ciencias en Informática en el Instituto Politécnico Nacional en México. El objetivo de esta tesis consistió en proporcionar una guía para agilizar/facilitar la implementación de los procesos correspondientes a la categoría de Operación de MoProSoft®. Así pues, la categoría de Operación incluye dos procesos: Administración de Proyectos Específicos (APE) y Desarrollo y Mantenimiento de Software (DMS). Por otro lado, los métodos ágiles que se utilizaron para complementar dicha categoría fueron Scrum y Programación Extrema (XP, por sus siglas en inglés).

Adicionalmente, la implementación de esta guía fue llevada a cabo en una empresa que cuenta con un área específica para el desarrollo de software y sistemas de información. Esta validación se realizó a través de un proyecto piloto que consistió en el desarrollo de un sistema de información en línea.

2.5.4.2. Descripción de la propuesta

La guía propuesta pretende ayudar a las empresas en la interpretación de los procesos a través del uso de las prácticas, valores, y principios resumidos por los métodos ágiles más utilizados a nivel mundial: Scrum y XP. En este sentido, las siguientes consideraciones fueron realizadas previamente con el objetivo de establecer el contexto real de la iniciativa de mejora:

- Fue necesario tener presente que los métodos ágiles hacen énfasis en aspectos técnicos y no de negocio, como medir el rendimiento de los equipos y los proyectos.
- MoProSoft® presenta un marcado enfoque hacia los procesos, mientras que los métodos ágiles se enfocan a los valores de trabajo de las personas.
- Cuando se utiliza el desarrollo ágil las fases de un proyecto son consideradas actividades.
- No se consideró que los métodos ágiles fueran modelos completos de procesos.
- El uso de un método ágil se justifica cuando los proyectos de software están propensos a cambios constantes (i.e., principalmente en los requerimientos) que no fueron contemplados al inicio.
- Para alcanzar el objetivo de la guía propuesta Scrum se enfocó en las prácticas de APE y XP en las de DMS. Así, ambos métodos se complementaron y combinaron para abordar áreas diferentes a través de aspectos comunes como las iteraciones, los incrementos, el desarrollo de prototipos rápidos y funcionales en cada iteración y la auto-organización de los equipos.

Así, para comenzar con la iniciativa de mejora fue necesario realizar un análisis que permitiera determinar qué actividades, tanto de APE como de DMS, podían ser realizadas a través de la adición de prácticas ágiles. Con el objetivo de medir el grado de cumplimiento de cada práctica o valor ágil a nivel de objetivos y de actividades se definió la nomenclatura resumida en la Tabla 6.

Utilizando la nomenclatura anterior, se procedió a analizar primeramente a APE. En este sentido, APE establece los siguientes objetivos:

- O1. Lograr los objetivos del proyecto en tiempo y costo mediante la coordinación y el manejo de los recursos del mismo.
- O2. Mantener informado al cliente mediante la realización de reuniones de avance del proyecto.
- O3. Atender las solicitudes de cambio del cliente mediante la recepción y análisis de las mismas.

Tabla 6. Nomenclatura definida para el análisis entre procesos de MoProSoft® y métodos ágiles (Rueda, 2010).

Calificador	Significado	Descripción
x	Insatisfactoria	La actividad no se puede conducir a través de prácticas ágiles.
√	Parcialmente satisfactoria	La actividad se puede conducir a través de prácticas ágiles, sin embargo, es necesario utilizar una actividad o práctica complementaria para cumplir con el requisito en el contexto de MoProSoft®.
√√	Completamente satisfactoria	La actividad puede conducirse completamente a través de prácticas ágiles.

Además, de acuerdo con la norma mexicana NMX-I-059/02 (Requisitos de Proceso), APE cuenta con 4 niveles de capacidad (Proceso Realizado, Proceso Administrado, Proceso Establecido y Proceso Predecible) y cada nivel está compuesto por diferentes actividades que podrían ser apoyadas por prácticas ágiles. La Tabla 7 resume los resultados de dicho análisis para APE.

Tabla 7. Análisis de APE y Scrum (Rueda, 2010).

Objetivo	Calificación	Descripción de cumplimiento
O1	√√	Scrum se enfoca en la forma de trabajo del equipo para construir un sistema flexible dentro de un entorno cambiante. Es en sí un método ágil para administrar proyectos.
O2	√√	Scrum realiza la reunión para la planificación y revisión de cada sprint donde interviene el cliente, usuario, etc., para mantenerlos informados.
O3	√√	Scrum responde a los cambios que surgen a lo largo del proyecto.

En relación a los niveles de capacidad de APE, el análisis indicó que es posible alcanzarlos con el apoyo de Scrum. En este sentido, la Tabla 8 resume esta información.

Tabla 8. Niveles de capacidad de MoProSoft® y cobertura de Scrum (Rueda, 2010).

Nivel	% de Cobertura	Justificación
1	100%	Todas las actividades descritas por MoProSoft® pueden ejecutarse de manera ágil utilizando Scrum.
2	90%	Los métodos ágiles no pueden corregir su ciclo de desarrollo, ya que éste ha sido previamente definido por el manifiesto ágil. Además, Scrum no maneja un protocolo de relación de contratistas. Es decir, a pesar de que el manifiesto ágil indica que lo más

		importante son las personas, y de que algunos de los principios básicos son la comunicación y el respeto, esto no implica que los contratistas estén familiarizados con estos términos.
3	30%	No es posible determinar la complejidad de un proyecto de software usando prácticas ágiles, ya que el producto evoluciona con respecto al tiempo. Por otro lado, el análisis y la medición no son procesos definidos por los métodos ágiles, ya que la única unidad de medida de un proyecto ágil es el código. Finalmente, en los métodos ágiles no existe una base de conocimiento, ya que la comunicación entre los equipos de desarrollo es directa (i.e., cara a cara), lo que permite transmitir el conocimiento con mayor precisión.
4	10%	No es alcanzable usando una combinación de métodos ágiles.

Posteriormente se procedió a realizar un análisis similar con DMS. Así, de acuerdo a MoProSoft® el propósito de DMS es la realización sistemática de las actividades de análisis, diseño, construcción, integración, y pruebas de los productos de software cumpliendo con los requerimientos especificados. Los objetivos del proceso establecidos por DMS son:

- O1. Lograr que los productos de salida sean consistentes con los productos de entrada en cada fase de un ciclo de desarrollo mediante las actividades de verificación, validación o prueba.
- O2. Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros mediante la integración de la configuración de software del ciclo actual.
- O3. Llevar a cabo las actividades de las fases de un ciclo mediante el cumplimiento del plan de desarrollo actual.

De manera similar que APE, DMS también se ve influenciado por los mismos 4 niveles de capacidad. Así, la Tabla 9 resume los resultados del análisis para DMS.

Este análisis considero que XP tiene como objetivo principal “construir software” utilizando grupos pequeños o medianos de programadores, en un entorno donde los requerimientos no están muy claros y cambian rápidamente o suelen ser de alto riesgo. Por lo tanto, para la implementación de DMS sería recomendable utilizar las prácticas de XP en combinación con algunas de Scrum. De acuerdo con los resultados, a través de XP se pueden alcanzar los objetivos de DMS de manera general, ya que estas prácticas permiten desarrollar un producto consistente mediante la integración continua y la planificación de pruebas. Además, al igual que Scrum, XP no se basa en un plan fijo, sino que se adapta a los cambios a través de las diferentes iteraciones requeridas para construir cada parte del producto de software.

Tabla 9. Análisis de DMS, Scrum y XP (Rueda, 2010).

Objetivo	Calificación	Descripción de cumplimiento
O1	√√	El producto se desarrolla en pequeñas iteraciones y antes de su liberación es sometido a una revisión continua mediante la planeación de pruebas e integración continua, lo que asegura que el producto sea consistente.
O2	√√	La realización de ciclos posteriores está relacionada, al igual que en Scrum, a los requerimientos del cliente, ya que él decide las historias que se seleccionarán para cada

		iteración. Además, XP hace uso de la refactorización y el uso de estándares de codificación, lo que hace posible el mantenimiento de un proyecto.
O3	√	Al igual que en Scrum no existe un plan fijo para el proyecto. Antes de iniciar un nuevo ciclo los programadores estiman cuanto esfuerzo requiere cada historia y a partir de ahí se define un cronograma.

En relación a los niveles de capacidad establecidos por MoProSoft®, DMS podría ayudar a alcanzarlos a través de una combinación de métodos Ágiles sin perder de vista el concepto de agilidad (véase Tabla 10).

Tabla 10. Niveles de capacidad de MoProSoft® y cobertura de Scrum y XP (Rueda, 2010).

Nivel	% de Cobertura	Justificación
1	100%	Todas las actividades descritas por MoProSoft® pueden ejecutarse de manera ágil utilizando una combinación de Scrum y XP.
2	90%	De acuerdo con el segundo principio del Manifiesto Ágil, el software que funciona es más importante que la documentación excesiva. Por esta razón, se requiere hacer un mayor esfuerzo de interpretación para alcanzar el Nivel 2 de madurez, ya que la mayoría de las actividades dan un seguimiento mediante la documentación de acuerdo a MoProSoft®, pero muchas de estas actividades ya están implícitas en las prácticas de Scrum y XP.
3	25%	La medición no está definida por ninguno de los métodos ágiles ya que la única unidad de medida en un proyecto ágil es el código. Por otra parte, en los métodos ágiles no existe una base de conocimiento, ya que la comunicación entre los equipos de desarrollo es directa, lo que permite transmitir conocimiento con mayor precisión y riqueza.
4	0%	No es alcanzable usando una combinación de métodos ágiles.

Finalmente, los roles propuestos tanto por MoProSoft® como por los métodos ágiles son pocos, por lo tanto, fue posible hacer una correspondencia directa que cumpliera con los roles que propone la norma (véase Tabla 11). Por otro lado, de acuerdo con las necesidades que tenía el área de Ingeniería de Costos e Informática de la empresa donde se llevó a cabo la implementación, se decidió implementar los procesos de APE y DMS con la finalidad de crear un entorno de trabajo flexible que dependiera más de las personas y donde los procesos se asemejaran más a procedimientos o rutinas de trabajo. Así, la implementación de APE requirió de la interpretación de las mejores prácticas de Scrum y XP.

Tabla 11. Mapeo entre roles de MoProSoft® y Scrum (Rueda, 2010).

Nombre	Acrónimo	Descripción
Jefe del Área Informática	JAI	Encargado de la Jefatura del Área de Ingeniería de Costos, Informática y Desarrollo de Aplicaciones (<i>Product Owner</i>).
Equipo de Desarrollo	EQU	Equipo de Desarrollo (<i>Scrum Team</i>).
Scrum Manager	SM	Responsable de que Scrum sea aplicado correctamente durante el desarrollo de un producto de software (<i>Scrum Manager</i>).

Área Usuaría	AU	Área que solicita el desarrollo o modificación de un producto de software (Cliente).
--------------	----	--

En este sentido, la Figura 2.13 representa al proceso APE en función de su relación con los roles presentados en la Tabla 11, de tal manera que refleja los elementos necesarios para la administración de los proyectos que se desarrollan en la empresa donde se llevó a cabo la implementación. Es importante hacer notar que este proceso puede cambiar conforme se desarrollen más proyectos, para contribuir así a la mejora continua del mismo.

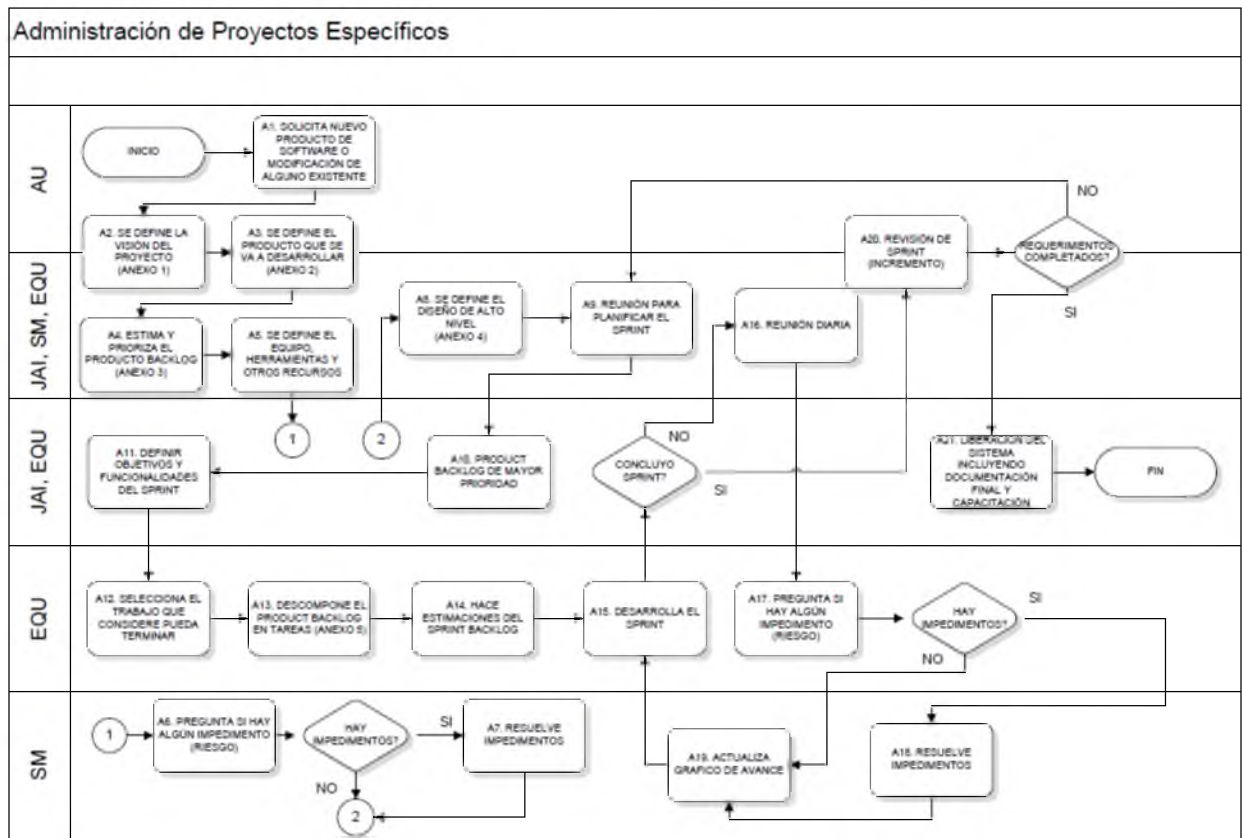


Figura 2.13. Diagrama definido para APE (Rueda, 2010)

De manera similar, el proceso de DMS fue implementado a través de la interpretación de las mejores prácticas de Scrum y XP para identificar los elementos necesarios para el desarrollo y mantenimiento de los productos de software (véase Figura 2.14).

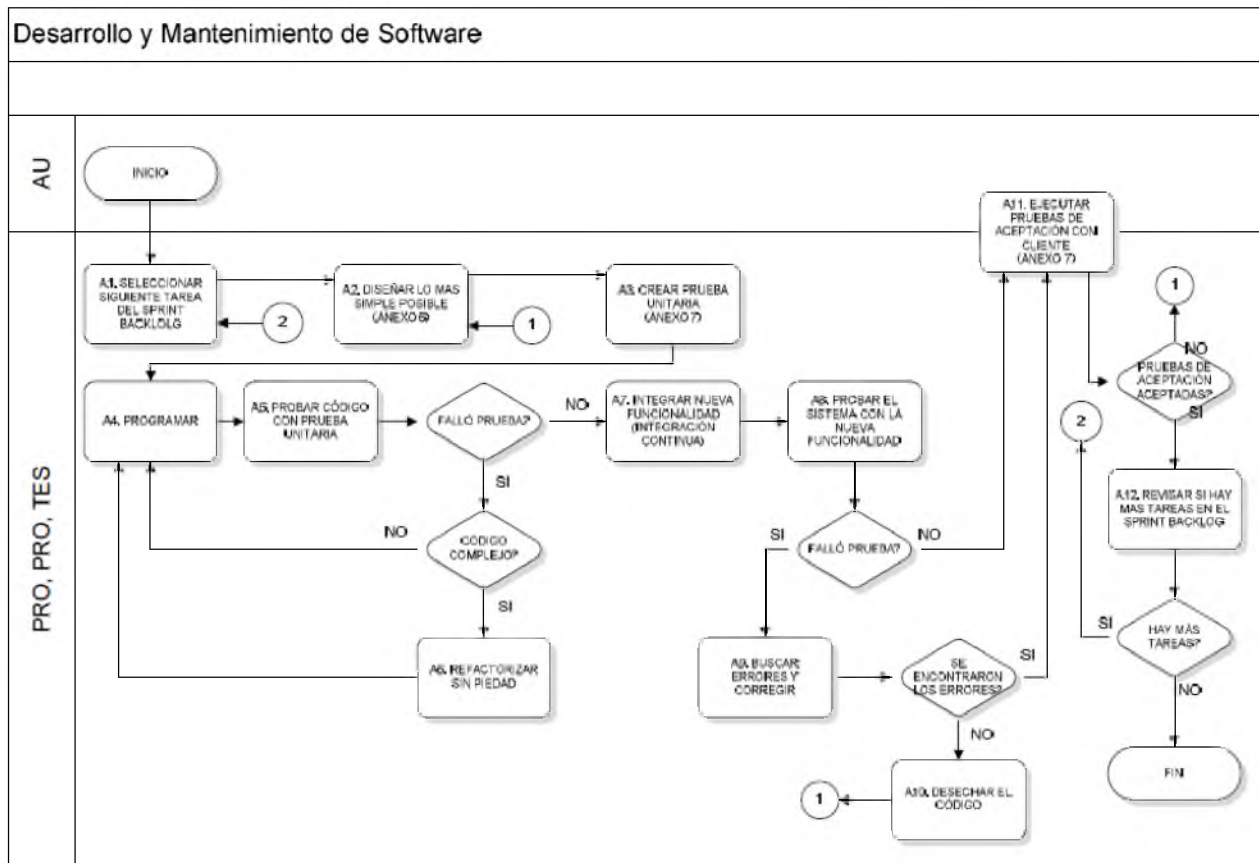


Figura 2.14. Diagrama definido para DMS (Rueda, 2010)

2.5.4.3. Resultados alcanzados

La guía creada en esta tesis fue implementada en el área de desarrollo de software de una empresa pública que brinda los servicios de energía eléctrica en la zona del centro del país. Dicho departamento requería mejorar la forma en que desarrollaba las aplicaciones informáticas para que otros departamentos pudieran realizar eficientemente tareas administrativas, técnicas y operativas de manera permanente. En este sentido, se definió un proyecto piloto con el área de Bienes Informáticos de dicha empresa, con la finalidad de utilizar los procesos propuestos en la guía.

Así pues, los resultados obtenidos en el proyecto piloto evidenciaron que es posible desarrollar software que satisfaga las necesidades del cliente y que genere poca documentación con base a la combinación con prácticas ágiles de Scrum y XP. Sin embargo, se menciona también que existen factores que deben refinarse, por ejemplo: los integrantes del equipo deben estar muy bien relacionados con las prácticas recomendadas por los métodos ágiles, también éstos deben estar comprometidos con los valores ágiles para lograr una comunicación efectiva y eliminar los malos hábitos. También se descubrió que fue posible incrementar la visibilidad tanto del seguimiento del proyecto como de algunas situaciones personales que de alguna manera pueden afectar a un proyecto. Esto permitió comprender mejor la categoría de Operación de MoProSoft® en los niveles de dirección y gerencial.

Se concluye que las metodologías ágiles son de gran ayuda para desarrollar software de manera incremental, mediante iteraciones que facilitan la obtención de versiones funcionales del producto que el cliente puede revisar para satisfacer rápidamente sus necesidades. Estos métodos ágiles son utilizados en gran parte por equipos pequeños de trabajo. De manera similar, MoProSoft® puede ser utilizado por empresas que cuentan con equipos pequeños de desarrollo, que cuentan con procesos más controlados por políticas internas. Mejor aún, esta tesis confirma que la combinación de los métodos ágiles con MoProSoft® contribuye a mejorar la verificación, validación, integración, documentación y configuración de un producto en la categoría de Operación.

2.5.5. Diseño e implementación de la guía de autoevaluación sobre la categoría de operación del Modelo de Procesos para la industria de software (MoProSoft®)

2.5.5.1. Objetivo

En el año 2010 se realizó una propuesta para diseñar e implementar una guía de autoevaluación sobre MoProSoft® en su categoría de Operación. Dicha propuesta fue presentada como tesis para obtener el grado de Maestro en Informática y Tecnologías Computacionales en la Universidad Autónoma de Aguascalientes (Preciado, 2010). El objetivo de esta tesis planteó la idea de incrementar la sensibilidad dentro de las empresas mexicanas de desarrollo de software en relación a la importancia de la implementación de buenas prácticas, y ayudarlas a determinar el estado en que se encuentran con respecto a MoProSoft®. Así, este trabajo de investigación se enfocó a tratar los problemas relacionados con la mayoría de los proyectos: no son terminados en las fechas indicadas en el plan de trabajo, los costos siempre son más elevados de lo estimado, y los procesos que se utilizan no siguen estándares que aseguren la efectividad de la práctica.

En este sentido, uno de los objetivos principales de este trabajo consistió en evaluar la utilidad, ventaja relativa, y facilidad de uso de los entregables, y diseñar conceptualmente una guía de autoevaluación para la categoría de Operación del modelo de procesos.

2.5.5.2. Descripción de la propuesta

La guía de autoevaluación tuvo como objetivo proporcionar una herramienta para que una empresa fuera capaz de analizar cómo se realizan los procesos de la categoría de Operación (APE y DMS concretamente) en relación a los niveles de madurez, y de esta forma evaluar la efectividad en la integración de mejores prácticas basadas en MoProSoft®. Esta guía, tal y como lo menciona su autor, fue creada por medio de una interfaz “amigable” para el usuario, la cual brindó, a través de preguntas y entregables para cada fase de los procesos, una idea clara sobre las actividades que se debían desarrollar. De esta forma se pretendió ayudar al usuario a mejorar la organización y administración de la información relacionada con sus proyectos.

Para generar y diseñar esta guía fue necesario realizar un análisis previo sobre los procesos de APE y DMS, de tal forma que una vez que fueran clasificados todos los entregables de sus prácticas, éstos fueran introducidos a la herramienta. De esta manera, el usuario podría utilizar dicha herramienta para cargar su información y utilizar los entregables para ejecutar correctamente las actividades correspondientes a cada fase de los procesos. Así pues, la guía conceptual se basa en las actividades de MoProSoft® para emitir un diagnóstico en relación con los procesos de la empresa y de esta forma poder establecer medidas de mejora. En este sentido, se diseñó un cuestionario de aproximadamente 200 preguntas para APE y otro de la misma cantidad de preguntas para DMS.

La Tabla 12 muestra que la validación de la herramienta requirió de tres tipos de constructos: utilidad, ventaja relativa, y facilidad de uso. Estos constructos fueron validados subjetivamente por expertos a través del instrumento denominado “*face validity*”.

Tabla 12. Constructos y variables de estudio (Preciado, 2010).

Constructo	Criterios	Variables
Utilidad	Definición (Du)	La herramienta ayuda a definir los requerimientos de un proyecto determinado.
	Planeación (Pu)	La herramienta ayuda a definir el análisis y diseño de un proyecto determinado.
	Creación (Cu)	La herramienta ayuda a obtener la fase de construcción de un proyecto.
	Integridad (Iu)	La herramienta ayuda a definir la integración y pruebas de un proyecto.
	Resultados (Ru)	La herramienta ayuda a definir el cierre de un proyecto determinado.
Ventaja relativa	Identificación (Ivr)	El uso de la herramienta ayuda a incrementar la calidad en el desarrollo y mantenimiento de sistemas.
	Calidad (Cvr)	El uso de la herramienta ayuda a incrementar la calidad en el desarrollo y mantenimiento de sistemas.
	Administración (Avr)	El uso de la herramienta ayuda llevar una mejor administración en el desarrollo y mantenimiento de sistemas.
	Confiabilidad (Covr)	El uso de la herramienta ayuda a prevenir errores posteriores.
	Interpretación (Ivr)	El uso de la herramienta ayuda a prevenir malas interpretaciones.
	Redundancia (Rvr)	El uso de la herramienta ayuda a prevenir duplicidad de la información.
	Evaluación (Evr)	El uso de la herramienta ayuda a evaluar el desarrollo y mantenimiento de sistemas.
Facilidad de uso	Claridad (Cf)	La interacción con la herramienta es clara y entendible.
	Interpretación (If)	La interacción con la herramienta es amigable.
	Simplicidad (Sf)	Existe un fácil manejo de la herramienta.

La herramienta que implementa la guía sistematizada de autoevaluación fue desarrollada para un entorno Web con el fin de que las empresas que desearan utilizarla pudieran hacerlo fácilmente. Los lenguajes que se utilizaron para su construcción fueron HTML y PHP. La base de datos fue diseñada en el lenguaje MySQL y para su implementación se utilizó el software WAMP Server (véase Figuras 2.15, 2.16 y 2.17).



Figura 2.15. Pantalla inicial de la herramienta (Preciado, 2010)



Figura 2.16. Pantalla que muestra un ejemplo de una pregunta de los cuestionarios de evaluación (Preciado, 2010)

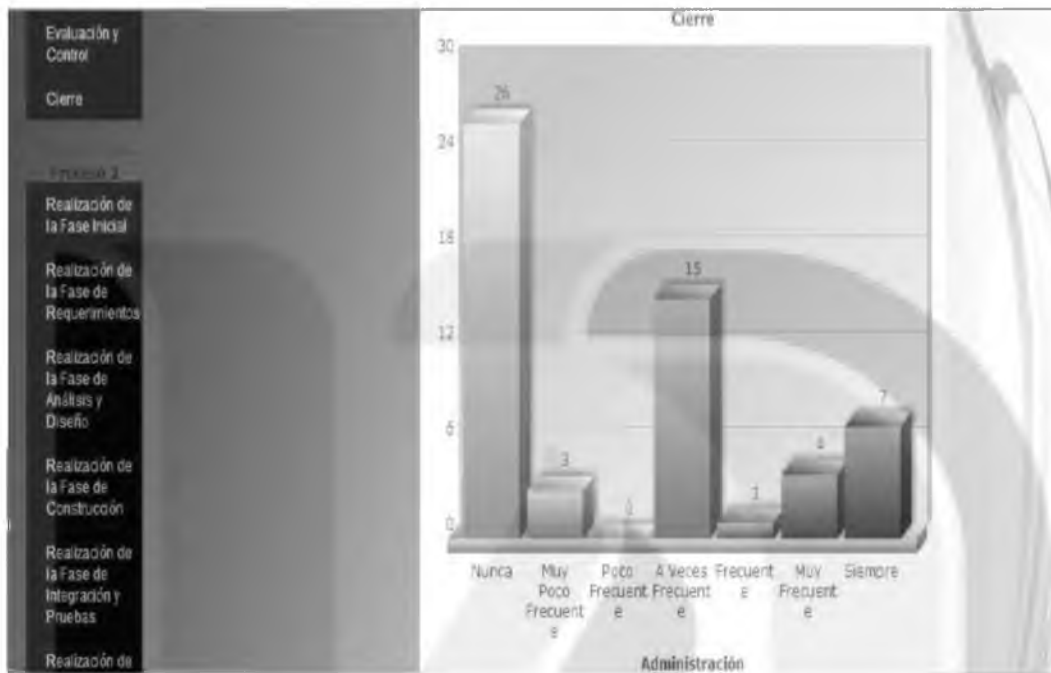


Figura 2.17. Pantalla que muestra un ejemplo de resultados de la evaluación (Preciado, 2010)

2.5.5.3. Resultados alcanzados

De acuerdo a su creador los resultados obtenidos con la aplicación de la guía de autoevaluación fueron positivos, ya que se logró cumplir con todos los objetivos planteados. En este sentido, la tesis argumenta que se realizaron pruebas piloto con diferentes empresas que no implementaban la autoevaluación para medir el nivel de madurez de sus procesos de acuerdo a MoProSoft®. Por último, la Tabla 13 resume los entregables que fueron generados para cada una de las fases de los dos procesos de la categoría de Operación de MoProSoft®.

Tabla 13. Entregables generados para OPE de MoProSoft® (Rueda, 2010).

Para APE	Para APE y DMS	Para DMS
Revisión de la descripción del proyecto	Reporte de actividades	Revisión del plan de desarrollo
Formato de descripción del proyecto		Especificación de requerimientos
Plan del proyecto		Plan de pruebas del sistema
Definición del proceso del sistema		Manual de usuario
Plan de adquisiciones		Manual de operación
Equipo de trabajo		Manual de mantenimiento
Plan de iteración del proceso		
Protocolo de entrega de avance		

Riesgos		
Plan de desarrollo		
Plan de comunicación e implantación		
Revisión de descripción del producto		
Calendario		
Reporte de mediciones y sugerencias		
Reporte de seguimiento de proyectos		

Finalmente, con la guía de autoevaluación se generaron diferentes entregables en la herramienta para cumplir con las actividades propuestas por MoProSoft®, teniendo como principal beneficio la reducción de tiempo puesto que solamente se requiere el vaciado de la información del proyecto.

2.5.6. Modelo de calidad para la microempresa basado en MoProSoft®

2.5.6.1. Objetivo

En el año 2012 se diseñó un modelo de calidad para la microempresa basado en MoProSoft®. Este modelo fue presentado como tesis para obtener el grado de Maestro en Computación en la Universidad de Colima (Flores, 2012). El objetivo de esta tesis consistió en plantear un modelo que estuviera basado en los tres niveles del modelo: Alta Dirección, Gerencia, y Operación. La creación de este modelo pretendió apoyar a las microempresas a mejorar sus estándares de calidad para que sean más competentes en el mercado laboral.

Así pues, uno de los objetivos principales de esta tesis fue identificar aquellas características de MoProSoft® que son viables en un entorno de desarrollo de una microempresa y a su vez conocer los resultados de la implementación de MoProSoft® en empresas de mayor tamaño.

2.5.6.2. Descripción de la propuesta

Para crear el modelo de calidad para la microempresa se empezó con el análisis de MoProSoft. MoProSoft®. En este sentido, es bien sabido que en el primer nivel (Alta Dirección) los directivos toman decisiones y definen la misión y valores de la empresa. Así, en una microempresa la alta dirección se compone por el dueño, quien realiza diferentes funciones al mismo tiempo, tales como la Alta Dirección y la Gerencia. Es por ello que la tesis decidió que para el desarrollo de este trabajo se unificaran en un solo nivel ambas actividades, mientras que para el nivel de Operación no se haría ningún cambio.

Mediante un análisis exhaustivo se determinó que era necesario descartar fases o procesos de MoProSoft®, debido a que en las microempresas la capacitación del personal es individual puesto que los tiempos de desarrollo son reducidos para poder subsistir, el personal con el que se cuenta es poco, entre otros factores. En este sentido, la Tabla 14 muestra la comparativa realizada entre MoProSoft® y el modelo propuesto por la tesis analizada.

Tabla 14. Comparativa entre MoProSoft y el modelo propuesto (Flores, 2012).

Categoría	MoProSoft®	Modelo propuesto
Alta Dirección	Si	Unido a la Gerencia
Gerencia	Gestión de Proyectos Gestión de Procesos Gestión de Recursos	Gestión de Proyectos Gestión de Procesos Gestión de Recursos
Gerencia / Gestión de Recursos	Plan de Capacitación Plan Operativo de Conocimiento de la Organización	Sin Plan de Capacitación Sin Plan Operativo de Conocimiento de la Organización
Operación / Administración de Proyectos Específicos	Si	Modificado, planificación dentro del Desarrollo y Mantenimiento de Software Sin Plan de Adquisiciones y Capacitación
Operación / Desarrollo y Mantenimiento de Software	Fase de Inicio Fase de Requerimientos Fase de Análisis y Diseño Fase de Construcción e Integración Manual de Usuario (preliminar y final)	Fase de Exploración Fase de Planificación Fase de Diseño Fase de Construcción Fase de Implementación Manual de Usuario Final

En este sentido, el nivel de Gerencia del nuevo modelo es una adopción de la categoría original de Gerencia en MoProSoft® que se divide en Gestión de Proyectos, Gestión de Procesos, y Gestión de Recursos Humanos de la siguiente manera:

- **Gestión de Proyectos:** su propósito es cerciorarse que los proyectos de las organizaciones cumplan con todos los objetivos y estrategias previamente establecidos. Básicamente en esta primera etapa las actividades a realizar son la base de una buena planificación de proyectos, debido a que se incorpora un plan estratégico, un plan de gestión de proyectos, un plan del proyecto, y un plan de mantenimiento.
- **Gestión de Procesos:** su propósito es definir los procesos de la organización en función de los Procesos Requeridos los cuales fueron identificados en la Gestión de Proyectos. Otra de las actividades a realizar es establecer, planificar, e implantar las actividades de mejora en los mismos. En esta segunda etapa se realiza la definición de elementos de procesos, se realiza un calendario de actividades que deberá incluir los procesos a realizar y su responsable, también se genera un plan de evaluación de procesos y un plan de manejo de riesgos que ayude a identificar y evaluar los riesgos, así como los planes de contención y de contingencia correspondientes.
- **Gestión de Recursos Humanos:** se define claramente que para ejecutar esta última etapa se debe tener conocimiento de la organización con el propósito de mantener disponible y

administrar correctamente la base de conocimiento, la cual contiene información de la organización. Dentro de este nivel, se deben definir diferentes mecanismos tales como el diseño de modelo conceptual, los mecanismos de alimentación, los mecanismos de consulta, los mecanismos de control de acceso, los mecanismos de mantenimiento, y los mecanismos de respaldo.

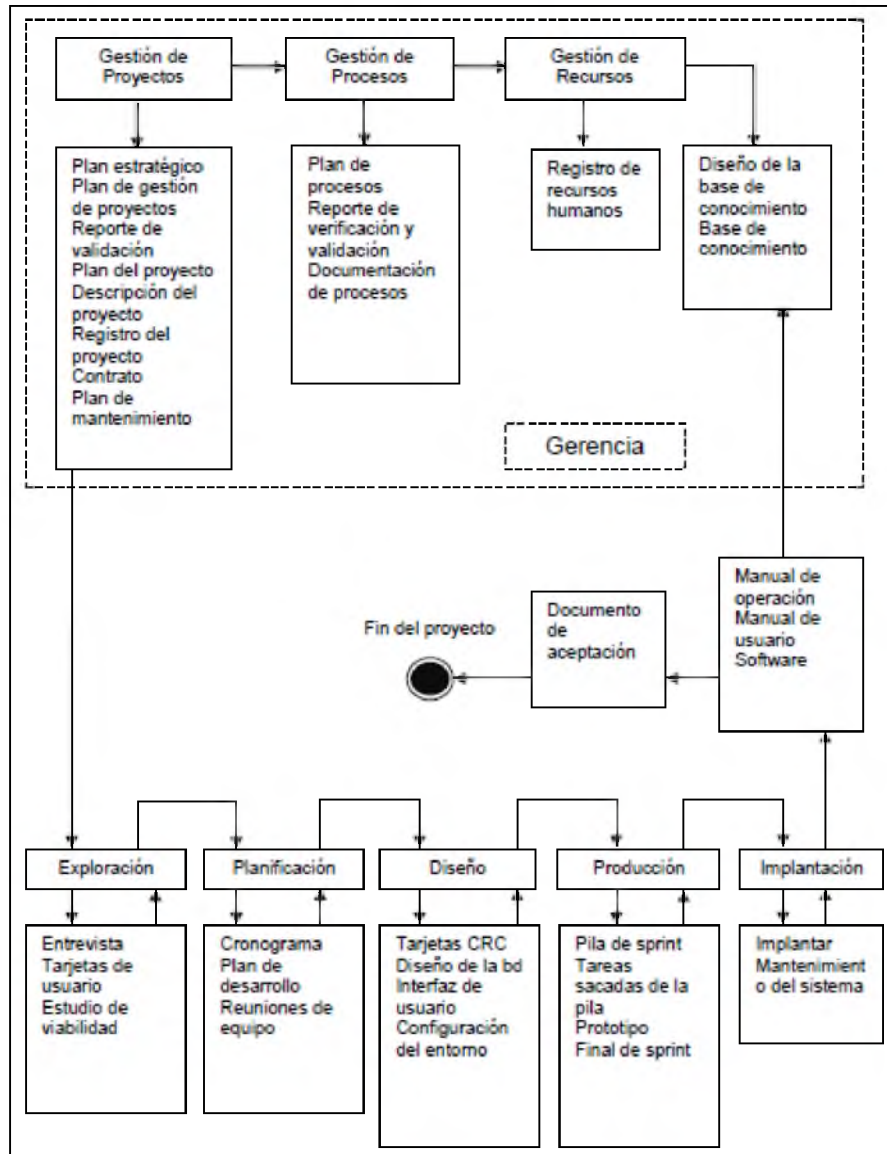


Figura 2.18. El modelo de calidad para la microempresa basado en MoProSoft (Flores, 2012)

El nivel de Operación está basado también en MoProSoft®, específicamente en el proceso de Administración de Proyectos Específicos, considerando una nueva metodología de desarrollo para el proceso de Desarrollo y Mantenimiento de Software que se generó tomando en cuenta la metodología de *eXtreme Programming* (XP), con SCRUM y el Método de Desarrollo de Sistemas Dinámicos (DSDM, por sus siglas en inglés). Esta metodología de desarrollo incluyó las siguientes fases:

- Fase de Exploración: dentro de esta fase se realizan entrevistas con el usuario con el objetivo de conocer lo que el cliente requiere, una vez que se obtuvieron los requerimientos, se genera un estudio de viabilidad y de negocios. Lo requerido por el cliente le ayuda al equipo de desarrollo a definir las herramientas, tecnología y las prácticas a utilizar en el desarrollo del proyecto.
- Fase de Planificación: el objetivo de esta fase es ejecutar reuniones entre el equipo de trabajo para planificar las actividades obtenidas de las entrevistas con el cliente. Las actividades tendrán que seccionarse por prioridades de entrega y se deberá generar un cronograma, serán los programadores quienes estimen cuánto esfuerzo se requiere para cada actividad y a partir de allí se debe definir el cronograma.
- Fase de Diseño: para esta fase se define la configuración del entorno y la interfaz de usuario, además de realizar el diseño de la base de datos de acuerdo a las tarjetas de clase de responsabilidad-colaboración (CRC), estas tarjetas facilitan la interacción entre los participantes del proyecto ya que sirven para representar clases con métodos, atributos, relaciones de herencia, composición o dependencia.
- Fase de Producción: esta fase está relacionada con la metodología ágil llamada SCRUM donde se realizan *sprints* de entre 2 y 4 semanas y pasa a mantenimiento. El desarrollo de software se realiza mediante iteraciones, llamadas *sprints*, que duran 30 días. Cada sprint da como resultado una versión del sistema ejecutable para el cliente. Con SCRUM las reuniones con el cliente se realizan a lo largo de proyecto.
- Fase de Implantación: como última fase del proceso es aquí donde se entrega el producto al cliente y se implanta, y será el cliente quien determine cuándo se realizará el mantenimiento al sistema. Dentro de esta etapa se realiza la documentación necesaria y no se deberán realizar cambios de arquitectura, diseño, o código elaborado. Para realizar pruebas será necesario implantar el sistema en un entorno real. Para cerrar el proyecto se deberá realizar un documento de aceptación que será entregado al cliente y firmado por ambas partes; en este documento se deberán establecer las condiciones para el mantenimiento del sistema y la conformidad del cliente con el producto entregado.

2.5.6.3. Resultados alcanzados

Se realizó una comparativa de las fases entre las diferentes metodologías ágiles tomando en cuenta los tiempos recomendados. Las tres metodologías que se compararon fueron XP, DSDM, y la metodología propuesta. De acuerdo a (Flores, 2012), los resultados obtenidos fueron los siguientes:

- XP: el tiempo estimado para las primeras fases antes del desarrollo tuvo como resultado 22-31 días.
- DSDM: el tiempo estimado para las primeras tres fases antes del desarrollo tuvo como resultado 26-28 días.
- Metodología propuesta: se obtuvo un tiempo menor en sus primeras fases antes del desarrollo.

La metodología DSDM dio como resultado un tiempo menor a XP pero no incluye en su proceso un plan de desarrollo y la metodología propuesta, a pesar de presentar un tiempo menor a

las anteriores, no presentó un estudio de viabilidad el cual permita conocer si el proyecto es viable. De igual forma el tiempo máximo fue mayor debido a un mayor número de subfases; sin embargo, se considera que éstas son importantes de ejecutar debido a que incluyen un plan de desarrollo, el diseño de base de datos y la interfaz del usuario.

El modelo propuesto en comparación con MoProSoft®, adopta nuevas fases en el nivel de Operación para el Desarrollo y Mantenimiento de Software: Exploración, Planificación, Diseño, Producción e Implantación, las cuales están basadas en metodologías ágiles XP, DSDM y SCRUM. Estas metodologías son las más utilizadas en las microempresas debido a que son fáciles de adoptar en los procesos de desarrollo.

Finalmente, de acuerdo a sus creadores, con base a los procesos de análisis para comparar MoProSoft® con la metodología propuesta se obtuvo como conclusión que se lo logró reducir MoProSoft® y adecuar el modelo a las microempresas por medio de un nuevo modelo que permite agilizar los procesos, de igual manera se reducen los tiempos de trabajo, debido a la inclusión de metodologías ágiles.

2.6. Conclusiones finales sobre el marco teórico

Como se puede observar, este capítulo ha evidenciado que la mayoría de las tesis analizadas se centran en proveer soluciones que facilitan la adopción de MoProSoft® en el contexto de las MiPyMEs. Sin embargo, no se encontró evidencia de trabajos que busquen desarrollar modelos o marcos de trabajo alternativos para ayudar a estas empresas en la conducción de una iniciativa de mejora.

En este sentido, es claro que estos trabajos han orientado sus esfuerzos ya sea a facilitar la evaluación de los modelos de evaluación o a simplificar los modelos de procesos, pero poco trabajo se ha realizado en el contexto de los modelos de mejora orientados a las MiPyMEs. Así pues, la propuesta de esta tesis pretende desarrollar un marco de trabajo como punto inicial en el contexto de los modelos de mejora actuales, demasiado complejos para que una pequeña empresa pueda realizar una iniciativa de mejora. Por lo tanto, el siguiente capítulo presenta de manera detallada la solución planteada para abordar la problemática identificada al inicio de esta tesis.

3. Marco de trabajo para guiar iniciativas de mejora con MoProSoft®

Con el objetivo de ayudar a las MiPyMEs a implantar un programa de mejora de sus procesos bajo el modelo MoProSoft®, esta tesis plantea el desarrollo de DERE. Este marco de trabajo está basado en las experiencias documentadas (tanto gubernamentales, como de clientes e industria) relacionadas con el uso de la Norma mexicana NMX-I-059/02-NYCE-2005 (NYCE, 2005b). En este sentido, el capítulo anterior de esta tesis ha resaltado que a pesar de que el modelo MoProSoft® ha sido implantado exitosamente en cerca de 400 empresas, no existe información alguna relacionada con un modelo o marco de trabajo que capacite al resto de empresas (las cuales regularmente desconocen qué hacer para conducir una iniciativa de mejora) para realizar una certificación oficial. Así, el propósito de DERE (llamado así por sus cuatro fases: Definir, Evaluar, Realizar, y Extender) es servir como una guía para aquellas organizaciones que se inician en la Mejora del Proceso de Software, así como para aquellas que ya tienen establecido MoProSoft® y buscan iniciar un plan de mejora para otros procesos o niveles de madurez. DERE proporciona cuatro fases que establecen un ciclo continuo para realizar una iniciativa de mejora; el número de ciclos que se requieren para alcanzar la mejora varía de organización a organización dependiendo la disponibilidad de infraestructura y recursos humanos, su nivel de compromiso. Como se verá más adelante, estos factores desempeñan un papel importante puesto que representan el factor principal para medir el éxito de un programa de mejora.

En este sentido, DERE reconoce tres elementos determinantes para el éxito de una iniciativa de mejora: la motivación del personal por adquirir mejores habilidades y por consiguiente madurar como una organización desarrolladora de software; el compromiso con la iniciativa de mejora, tanto de los directivos para proporcionar los recursos necesarios como del personal para realizar las actividades que la mejora requiera; y la capacitación, puesto que la empresa debe asumir que su personal carece de conocimiento técnico y que el tiempo destinada a ésta impactará directamente en el calendario establecido.

La Figura 3.1 muestra que DERE establece un ciclo continuo de trabajo para realizar una iniciativa de mejora, el cual es conducido por tres roles principales: la alta dirección, el jefe de proyecto y el equipo de trabajo. Es evidente pensar que estos roles cubren los niveles de trabajo definidos por la pirámide organizacional establecida por Frederick Taylor a través de sus principios de administración científica; sin embargo, es importante mencionar que la Norma mexicana NMX-I-059/02-NYCE-2005 está orientada, por definición, a estos tres niveles y que por ende un marco de trabajo orientado a conducir su implantación debe respetarlos. El jefe de proyecto, por ejemplo, deberá realizar las labores de “consultor” interno para ejecutar los ciclos de DERE con el objetivo de

que la empresa tenga una clara visión de la mejora antes de llamar al órgano certificador. De esta manera, DERE pretende ayudar a las empresas a realizar “experimentos de laboratorio” para asegurar la mejora antes de solicitar una certificación oficial y reducir así de manera considerable la inversión económica.

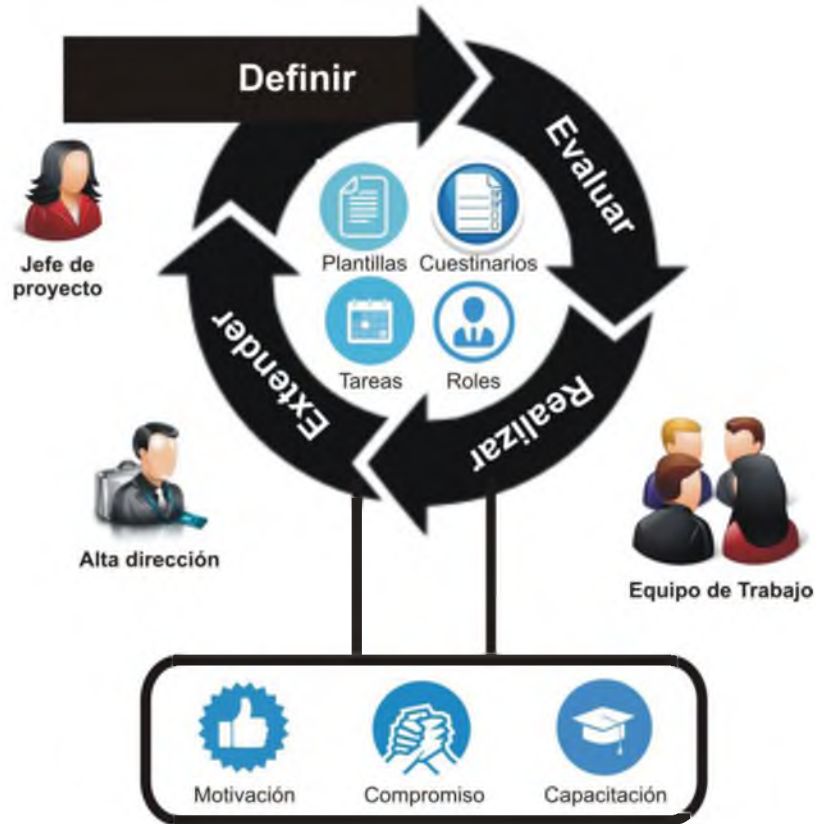


Figura 3.1. Fases del marco de trabajo DERE

Se pretende que DERE establezca los lineamientos para conducir una iniciativa de mejora y proporcionar ayuda a los jefes de proyecto a través de plantillas, cuestionarios, tareas a realizar, roles que desempeñar y demás, con el objetivo de que sea la misma empresa quien entrena antes de requerir la visita del organismo de evaluación. A continuación se proporciona una descripción detallada de cada fase.

3.1. Fase Definir

En esta fase la alta dirección de la empresa de software debe comprender la necesidad de incorporar un programa de mejora a través de un análisis costo/beneficio. En este sentido, es necesario definir los roles y las responsabilidades como infraestructura inicial, además de gestionar el compromiso de la alta dirección y los mandos intermedios (o gerenciales) para comunicar los objetivos y metas de la empresa, así como para garantizar la disponibilidad de recursos y la priorización del proyecto de mejora. Por lo tanto, es necesario especificar los objetivos del programa de mejora y el personal involucrado debe comprometerse a ejecutar las soluciones propuestas al interior de la empresa como corrección a los problemas encontrados. Por consiguiente, es necesario definir un Grupo Directivo y un Grupo de Procesos (véase Figura 3.2) que deben integrarse con

personas con ciertas habilidades y conocimientos que les permitan asumir la responsabilidad en el desarrollo de los procedimientos, planes y calendarios que conduzcan a la organización a través del proceso de mejora.

El Grupo Directivo asegura que los recursos necesarios estén disponibles cuando sean requeridos, recursos tales como humanos, financieros, de infraestructura, etc. Este grupo debe monitorizar la iniciativa de mejora y aprobar los nuevos procedimientos creados o modificados para el proceso de la empresa.



El Grupo de Trabajo desarrolla la mayor parte de las actividades de mejora (desde la evaluación, definición y documentación del proceso, pasando por la generación de planes de acción y su posterior ejecución y monitorización, hasta el establecimiento de procedimientos y medición de la mejora). El responsable del Grupo de Trabajo debe ser un jefe de proyecto con mayor antigüedad.



Figura 3.2. Roles involucrados en la fase de definición

Al final de esta fase inicial y una vez que se han establecido correctamente los criterios mencionados, el Grupo de Trabajo habrá alineado la iniciativa de mejora con los objetivos de negocio de la empresa, y habrá creado un documento de compromiso que deberá seguirse en las fases posteriores del modelo. De manera más específica, en esta fase se identifican diez actividades que deberán ser completadas de la siguiente forma:

1. *Inicio*: El propósito de esta actividad consiste en establecer los lineamientos para gestionar la fase inicial del programa de mejora. Para esto se requiere documentar información que comprende las necesidades actuales, políticas organizacionales, regulaciones que puedan afectar al programa, actividades similares de mejora que ya existan dentro de la organización o que estén planeadas a futuro, etc. El objetivo es básicamente identificar a los departamentos que participarán en la evaluación, de tal forma que sea posible seleccionar un enfoque que conduzca el programa de mejora. En el caso de DERE, el enfoque está determinado por lo que el modelo SCAMPI (SCAMPI, 2011) y el método EvalProSoft® (NYCE, 2005^a) definen como enfoque “experimental” o de “prácticas de laboratorio”. Es decir, se define un marco de trabajo que permite la experimentación en el campo de la Mejora del Proceso de Software y genera por ende el conocimiento necesario para que, llegado un momento, la empresa pueda solicitar una certificación con la plena seguridad de que la pasará sin exceder el limitado presupuesto con el que cuenta. Para esto, es necesario que se seleccione a un líder que cumpla con habilidades específicas sobre el área de mejora y que posea conocimientos sobre la planificación de proyectos. En este sentido, se recomienda que se inicie con la dirección de un consultor externo o especialista sobre Mejora del Proceso de Software que esté adscrito a una institución educativa, para que los jefes de proyecto aprendan a utilizar el marco de trabajo DERE y puedan utilizarlo posteriormente sin problema. Esto último es un factor determinante puesto que es necesario que un especialista brinde entrenamiento sobre aquellos aspectos que sean identificados como problemas o deficiencias del proceso que se pretende mejorar.

2. *Identificar las necesidades de negocio y conductores para la mejora:* El propósito de esta actividad consiste en entender, desde una perspectiva gerencial, las necesidades clave de negocio para vincularlas con los requerimientos del programa de mejora. Entre las tareas que se desarrollan en esta actividad se encuentran la revisión de la visión organizacional y las necesidades de mejora en el contexto gerencial. Es necesario también conducir entrevistas con los participantes de nivel gerencial para que éstos comprendan cómo el programa de mejora satisfará (completa o parcialmente) sus necesidades. En este sentido, todo el personal involucrado debe entender por qué las empresas invierten tanto esfuerzo y tiempo en un programa de mejora, dado que a medida que su entendimiento crezca también lo hará su apoyo. Por lo general, las motivaciones exitosas eliminan el “dolor” de la situación actual, a diferencia de promesas sobre un estado deseado. Estas motivaciones deben documentarse y extenderse a toda la empresa.
3. *Desarrollar una propuesta de mejora:* El propósito de esta actividad es establecer una propuesta que explique a la alta dirección por qué debe iniciarse un programa de mejora, cuánto costará realizarlo, cuánto tiempo se llevará hasta visualizar los primeros resultados, y qué tipo de resultados serán obtenidos. Esta actividad representa un punto crucial para decidir si el programa continúa o no. El borrador de la propuesta deberá enviarse continuamente a los participantes para su revisión y obtención de comentarios. Adicionalmente, el borrador debe establecer los objetivos para el programa de mejora, asegurando el alineamiento con los objetivos de la organización y necesidades críticas previamente identificadas. Es necesario también determinar el alcance del programa mediante la identificación de los departamentos de la empresa que serán incluidos en el programa de mejora a través de, por ejemplo, el establecimiento de roles y responsabilidades. Por último, se desarrolla un plan de alto nivel que resume las actividades y calendarios que constituye los requerimientos básicos en cuanto a personal, necesidades de capacitación (si es que la hay), equipo de cómputo, etc.
4. *Educar y brindar soporte:* Los propósitos de esta actividad son crear conciencia sobre lo que implica la iniciativa de mejora, establecer expectativas alcanzables, y definir el soporte al programa de mejora dentro de la empresa. Para esto, es necesario establecer sesiones informativas orientadas a la alta dirección y los mandos gerenciales (jefes de proyectos) para explicar el esfuerzo que está por iniciar. En este sentido, es recomendable que durante el programa de mejora se establezca directamente el diálogo con el personal que participará en todas las actividades establecidas por DERE.
5. *Obtener la aprobación para la propuesta de mejora y los recursos iniciales:* El propósito de esta actividad consiste en presentar la propuesta de mejora ante la alta dirección para obtener su aprobación y lograr la asignación de tiempo y recursos requeridos para lanzar el programa. DERE establece que esta aprobación se obtenga a través de iteraciones hasta que se alcance el acuerdo sobre la asignación de recursos para continuar con la iniciativa de mejora, o bien se decida abandonarla por la falta de éste. Una vez que se obtiene la aprobación de la alta dirección, los recursos iniciales son asignados para iniciar el trabajo y se establece una estrategia de lanzamiento que identifica a las responsabilidades sobre los recursos, la distribución de recursos (incluyendo el tiempo requerido de la alta dirección para participar en las actividades), y la actualización del documento final de compromiso.
6. *Establecer la infraestructura para la mejora:* Para gestionar exitosamente el programa de mejora es necesario que se defina una infraestructura que incluya tanto deberes como

responsabilidades. Esta infraestructura puede proveer recursos cuando sea necesario, mantener la visibilidad del programa de mejora, y obtener y mantener las lecciones aprendidas. La infraestructura se proporciona en términos de personal específico, entidades organizativas, documentos, responsabilidades, y lugares de trabajo. En resumen, el propósito de esta actividad consiste en asignar las responsabilidades necesarias que aseguren que la información referente a los avances del programa de mejora se mantenga visible y sea compartida entre todos los participantes.

7. *Evaluar el clima para la mejora:* El propósito de esta actividad consiste en identificar las barreras que afecten negativamente al programa de mejora y desarrollar planes eficaces que garanticen que las mejoras realizadas perduren.
8. *Definir las metas generales de la mejora:* La Mejora del Proceso de Software es una inversión a largo plazo. Se requiere de metas mensurables, claramente definidas, para orientar y ayudar en el desarrollo de tácticas de mejora. Estas metas también permiten la medición objetiva de los resultados de mejora. La creación de buenas metas requiere la comunicación bidireccional entre diferentes grupos de gestión y entre la dirección y los especialistas. Así, el propósito de esta actividad consiste en definir qué indicadores serán necesarios para determinar la satisfacción del objetivo, de esta manera el programa de mejora puede ser una estrategia para determinar una visión más clara y vincularla con el plan de negocios.
9. *Definir los principios guía del programa de mejora:* El propósito de esta actividad es definir como principio rector del programa de mejora el experimentar con los procesos de las MiPyMEs.
10. *Lanzar el programa:* El propósito de esta actividad es iniciar la segunda fase de DERE (la Evaluación) y comenzar con el ciclo continuo de mejora.

En este sentido, DERE introduce la “Plantilla para la Definición del Compromiso” (véase Anexo B.1) para satisfacer las diez actividades propuestas y definir los compromisos sobre la iniciativa de mejora.

3.2. Fase Evaluar

En esta fase el Grupo de Trabajo deberá comprender el proceso de software actual de la empresa de tal forma que pueda desarrollar un plan que cubra los cambios especificados en las metas de mejora. Así, las actividades realizadas en la fase “Evaluar” proporcionan esta información a través de la planificación de la mejora y el proceso de priorización. El conocimiento sobre las fortalezas y debilidades de la empresa es un prerrequisito esencial para identificar y priorizar un programa efectivo de mejora. La salida principal de esta fase resume las conclusiones finales y un reporte de recomendaciones que es generado como resultado de la evaluación. La información sobre el estado actual de la empresa es usada por el Grupo de Trabajo durante la fase “Realizar” para desarrollar soluciones de mejora.

En este sentido, el propósito de esta fase es obtener una idea clara de las fortalezas y debilidades actuales de la empresa, es decir obtener una línea base. Esta línea base proporcionará información sobre cómo y qué tan bien realiza el personal sus actividades (software) en la actualidad. Esta información es útil para iniciar el desarrollo de un plan de acción que proporcione

guía y dirección al programa de mejora en los años siguientes. Las actividades de la empresa deben ser auto-verificadas, es decir la credibilidad de éstas depende de la habilidad percibida para extraer la información importante y real de la empresa y presentarla en una forma coherente y viable. Esta fase debe realizarse a través de las seis actividades siguientes:

1. *Determinar el alcance de la evaluación:* La organización tiene razones importantes para emprender un programa de mejora. El propósito de decidir el alcance de la evaluación es asegurar que el enfoque del programa de mejora esté ligado a las necesidades de negocio de la empresa. En este sentido, DERE considera que la evaluación debe determinar el porcentaje de cobertura que una empresa muestra en relación a las actividades o prácticas recomendadas por MoProSoft®.
2. *Planificar la evaluación:* Para realizar las actividades de la evaluación se requiere la coordinación de personal, datos, instalaciones, entrenamiento y servicios de soporte. Puede ser que esta actividad requiera el repetir algo del trabajo realizado en la fase “Definir”. En otras ocasiones no es necesaria esta repetición, pero a menudo, el Grupo de Trabajo se forma con diferentes miembros a los que crearon el programa de mejora, y tendrán que cubrir algunos de los mismos temas para desarrollar su propio entendimiento y estrategia. Cuando esta actividad es introducida como resultado de un ciclo posterior a través de DERE, esta repetición no es necesaria.
3. *Realizar la evaluación:* El propósito de realizar la evaluación es reunir información factible para apoyar al esfuerzo de mejora. La información recopilada proporcionará una “radiografía” de las fortalezas y debilidades de la empresa relacionadas con su proceso de software y a sus prácticas de gestión del desarrollo de software.
4. *Presentar los hallazgos de la evaluación:* El Grupo de Trabajo, al final de la recogida de datos, presenta a los participantes los problemas que ha encontrado. En esta reunión se describen el método usado para evaluar, los participantes, las áreas de investigación y las fortalezas y debilidades que se hayan encontrado. Como medida de éxito, se recomienda empezar a trabajar en la guía del plan de acción con la intención de que la presentación de los hallazgos no desanime a los oyentes. Este documento, la guía del plan de acción, será moldeada en la siguiente fase de DERE, pero en esta fase basta con preparar un conjunto de acciones que aborde los problemas detectados.
5. *Desarrollar las conclusiones finales y el reporte de recomendaciones:* Las conclusiones finales y el reporte de recomendaciones documentan los hallazgos de la evaluación y representa el estado actual de la empresa. En este sentido, el Grupo de Trabajo debe desarrollar un conjunto de recomendaciones con base a las conclusiones que fueron descubiertas durante la evaluación. Regularmente estas actividades permiten identificar inconsistencias y generar recomendaciones basadas en un consenso más amplio. Estos resultados son incorporados en el plan de acción como una estrategia para mejorar las actividades actuales.
6. *Comunicar las conclusiones y recomendaciones a la empresa:* En este punto el personal se preguntará sobre todas las actividades que ocurrieron durante el diagnóstico. Para aliviar cualquier temor que pueda haber surgido, o que pudiera surgir, se recomienda que los resultados de las actividades del diagnóstico sean comunicados a toda la empresa. La organización puede lograr esto mediante la realización de una serie de reuniones, de tal manera que todos los miembros de la empresa escuchen el mismo mensaje. Esto contribuirá a la construcción del patrocinio y del apoyo al programa de mejora.

Como se mencionó anteriormente, las actividades de la fase “Evaluar” establecen el uso de un mecanismo de diagnóstico con el fin de caracterizar el estado inicial del proceso de las MiPyMEs y facilitar la definición de un nuevo proceso acorde a sus necesidades. En este sentido, DERE incorpora un mecanismo de evaluación que permite que el Grupo de Trabajo evalúe sus procesos con el fin de identificar debilidades y fortalezas en los mismos. A través de esta evaluación, el Grupo de Trabajo recibe una ayuda inicial sobre qué actividades debe incorporar a su nuevo proceso. De esta manera, los jefes de proyectos deben responder a una serie de cuestionarios (Anexo C) para obtener un nivel de cobertura por cada proceso evaluado (i.e., que servirá posteriormente como línea base), tomando como base el proceso actual y MoProSoft®. Así pues, en el contexto de la Mejora del Proceso de Software, suele elegirse una técnica basada en cuestionarios (Anexo C) ya que proporciona soluciones rápidas dentro de una metodología de investigación, dado que es el investigador quien determina las preguntas a formular y la gama de respuestas que se pueden dar. Esto hace que el resultado obtenido sea más preciso y fácil de analizar desde el punto de vista del investigador (Young, Fang, & Hu, 2006; Habra, Alexandre, Desharnais, Laporte, & Renault, 2008; Pino, Pardo, García, & Piattini, 2010). Así, el mecanismo de evaluación utilizado en DERE se basa en dos tipos de actividades: prácticas específicas (relacionadas con los jefes de proyectos) y prácticas genéricas (relacionadas con la alta dirección de la empresa) (García, Calvo-Manzano, Cuevas, & San Feliu, 2007). De esta manera, DERE define un cuestionario para cada una de las áreas de proceso de MoProSoft® abordadas en esta tesis (dichos cuestionarios pueden ser consultados en el Anexo C de este documento). En este sentido, el Grupo de Trabajo debe analizar los resultados de la evaluación dando un peso específico a cada tipo de respuesta e identificado las prácticas que están institucionalizadas dentro de las empresas y aquellas que no se realizan en absoluto (véase Tabla 15).

Tabla 15. Clasificación de niveles de rendimiento durante la evaluación del proceso (García et al., 2007).

Posible respuesta	Nivel de rendimiento	Descripción
Siempre	4	La actividad está documentada y establecida en la empresa. Siempre se lleva a cabo, entre el 76% y 100% de las veces, en sus proyectos de software.
Usualmente	3	La actividad está establecida en la empresa, pero rara vez es documentada. Se realiza por lo general, entre el 51% y 75% de las veces, en sus proyectos de software.
Algunas veces	2	La actividad está débilmente establecida en la empresa. Se realiza a veces, entre el 26% y 50% de las veces, en sus proyectos de software.
Rara vez	1	La actividad se realiza con poca frecuencia en la empresa. Rara vez se lleva a cabo, entre el 1% y el 25% de las veces, en sus proyectos de software.
Nunca	0	La actividad no se lleva a cabo en la organización (0%). Ninguna persona o grupo realiza la actividad de la organización.
No lo sé	--	La persona no está segura de cómo responder a la pregunta.
No aplica	--	La pregunta no se aplica a la empresa.

La técnica implementada utiliza un promedio ponderado para calcular el nivel de cobertura para cada pregunta (número 1 en la Figura 3.3). Por lo tanto, el análisis se traslada de un punto de vista cualitativo (gama de respuestas Siempre (S), Usualmente (U), Algunas veces (A), Rara vez (R), y Nunca (N)) a una visión cuantitativa (nivel de cobertura por pregunta) (número 2 en la Figura 3.3). Además, la desviación estándar también es considerada mediante el cálculo del nivel de cobertura para cada proceso (número 3 en la Figura 3.3).

El porcentaje de nivel de cobertura para cada pregunta (Ncp) se calcula considerando las respuestas de todos los jefes de proyecto de la siguiente manera:

$$Ncp_{ij} = Cr_{iS} * 1 + Cr_{iU} * 0.75 + Cr_{iA} * 0.50 + Cr_{iR} * 0.25 + Cr_{iN} * 0 \quad 1)$$

Donde Cr es la cobertura de la respuesta, i es el número de pregunta, j es el número de jefes de proyectos, y los números representan los porcentajes definidos en la Tabla 15 para cada tipo de respuesta. La razón de esta ponderación es porque se considera que las afirmaciones más fuertes deben tener más importancia que las débiles, dado que, por ejemplo, algo que siempre se hace no puede influir de la misma manera que algo que se hace raramente. Sin esta ponderación todas las preguntas tendrían siempre una cobertura del 100%, lo cual no aclararía nada.

Además, para cada pregunta, la media matemática y la desviación estándar son calculadas de la siguiente manera:

$$M_i = \frac{\#A * 4 + \#U * 3 + \#S * 2 + \#R * 1 + \#N * 0}{jp} \quad 2)$$

$$Desv_i = \sqrt{\frac{[\#jp * (\#A * 4^2 + \#U * 3^2 + \#S * 2^2 + \#R * 1^2) - \#jp^2 * M_i^2]}{jp^2}} \quad 3)$$

Donde i es el número de pregunta, jp es el número de jefes de proyectos que participan en la evaluación de los procesos y las letras con # son los niveles de desempeño definidos en la Tabla 15. Los valores de 1, 0.75, 0.5, 0.25 y 0 también podrían ser utilizados y el valor máximo de la media matemática en lugar de 4 sería 1. Sin embargo, es recomendable utilizar valores más altos de ponderación (4, 3, 2, 1, y 0) mediante la aplicación de un factor, por lo tanto las diferencias existentes son más visibles. Es importante tener en cuenta que existe una correspondencia biunívoca o el mismo significado entre el porcentaje de cobertura de la pregunta y su media matemática (la única diferencia es que los factores de escala son diferentes), de tal forma que un alto porcentaje de cobertura también reflejará una media matemática alta y en la misma proporción que el porcentaje.

Por último, el nivel de cobertura total (NCT) del proceso se calcula de la siguiente manera:

$$NCT(p) = \frac{\sum_{i=1}^n Cp_i}{n} \quad 4)$$

Donde p es el proceso evaluado, i es el número de pregunta, y n es el total de preguntas del cuestionario. De esta manera, el Grupo de Trabajo determina el nivel de cobertura de su empresa y el promedio de cobertura para identificar las fortalezas y debilidades de los procesos seleccionados. En este sentido, para analizar las fortalezas y debilidades (o problemas), el Grupo de Trabajo tiene que usar las respuestas obtenidas en las evaluaciones. Aquellas preguntas con una cobertura inferior

a cierto porcentaje (número 4 en la Figura 3.3) implican que la actividad de MoProSoft® (número 5 en la Figura 3.3) no está bien implementada en la empresa y tiene que ser considerada como una cuestión a mejorar (o una debilidad) en el proceso. Este porcentaje de cobertura varía de acuerdo al modelo de procesos que se pretenda adoptar, por ejemplo para obtener un nivel de madurez del CMMI, el órgano certificador solicita una cobertura mínima del 75%, en caso de MoProSoft® este valor varía a 85%.

Administración de Proyectos Específicos	Jefes de proyecto		1					2		
	1	2	#S	#U	#A	#R	#N	Ncp	M	D
1. ¿Se identifican hitos importantes para el proyecto?	S	N	1 50%	0	0	0	1 50%	50%	2	2.00
2. ¿Se identifican suposiciones sobre el calendario del proyecto?	A	R	0	0	1 50%	1 50%	0	37%	1.5	0.50
3. ¿Se identifican las limitaciones del proyecto?	A	A	0	0	2 100%	0	0	50%	2	0
4. ¿Se identifican las dependencias de las tareas del proyecto?	U	N	0	1 50%	0	0	1 50%	37%	1.5	1.50
5. ¿Se definen y mantienen el presupuesto y calendario del proyecto?	S	A	1 50%	0	1 50%	0	0	75%	3	1
6. ¿Se establecen acciones correctivas para el proyecto?	R	A	0	0	1 50%	1 50%	0	37%	1.5	0.50
7. ¿Se identifican los riesgos del proyecto?	A	A	0	0	2 100%	0	0	50%	2	0
8. ¿Se documentan los riesgos?	R	R	0	0	0	2 100%	0	25%	1	0
9. ¿Se evalúan los riesgos cuando es necesario?	R	R	0	0	0	2 100%	0	25%	1	0
10. Do you establish requirements and procedures to ensure privacy and security of the project's data?	U	R	0	1 50%	0	1 50%	0	50%	2	1
11. ¿Se establecen mecanismos para archivar los datos y acceder a los datos archivados?	U	U	0	2 100%	0	0	0	75%	3	0
12. ¿Se determinan los datos del proyecto que serán identificados, recogidos y distribuidos?	U	U	0	2 100%	0	0	0	75%	3	0
13. ¿Se determinan los requisitos de instalaciones y equipo?	R	R	0	0	0	2 100%	0	25%	1	0
14. ¿Se determinan los requisitos de personal?	R	R	0	0	0	2 100%	0	25%	1	0
15. ¿Se identifican el conocimiento y las habilidades necesarias para desarrollar el proyecto?	U	U	0	2 100%	0	0	0	75%	3	0
TOTALES			2 6.7%	10 33.3%	6 20%	12 40%	0	47%		

Figura 3.3. Ejemplo de análisis de respuestas realizado por los jefes de proyectos

Por otro lado, las preguntas con cobertura igual o mayor al porcentaje marcado por el órgano certificador deben ser analizadas por el Grupo de Trabajo a través de la desviación estándar de la siguiente manera:

- Si la desviación estándar es inferior a 0.8, la actividad de MoProSoft® relacionada con la pregunta representa una fortaleza del proceso evaluado (número 6 en la Figura 3.3).
- Si la desviación estándar es igual o superior a 0.8 (número 7 en la Figura 3.3), la actividad del modelo de referencia relacionada con esa pregunta debe ser profundamente explorada por el Grupo de Trabajo a través de entrevistas para aclarar si se trata de una fortaleza o de un aspecto a mejorar. Este caso significa que ha habido fuertes discrepancias en las respuestas dadas por la empresa al completar los cuestionarios (las respuestas de los jefes de proyectos son muy divergentes y, por lo tanto, deben ser exploradas a mayor profundidad).

Así, la Figura 3.3 resume el proceso que el Grupo de Trabajo tiene que realizar para tomar una decisión y definir un proceso que tome en cuenta todas las deficiencias detectadas.

Adicionalmente, los resultados obtenidos en las empresas se muestran con gráficos de barras, que resumen el porcentaje de prácticas efectivas que es cubierto por los procesos evaluados, y gráficas de Kiviati, que muestren el nivel de madurez/capacidad de todos los procesos evaluados en la empresa (véase Figura 3.4). Finalmente, DERE introduce la “Plantilla para entrega de los Resultados de la Evaluación” (véase Anexo B.3) para reunir y presentar los hallazgos de esta fase.

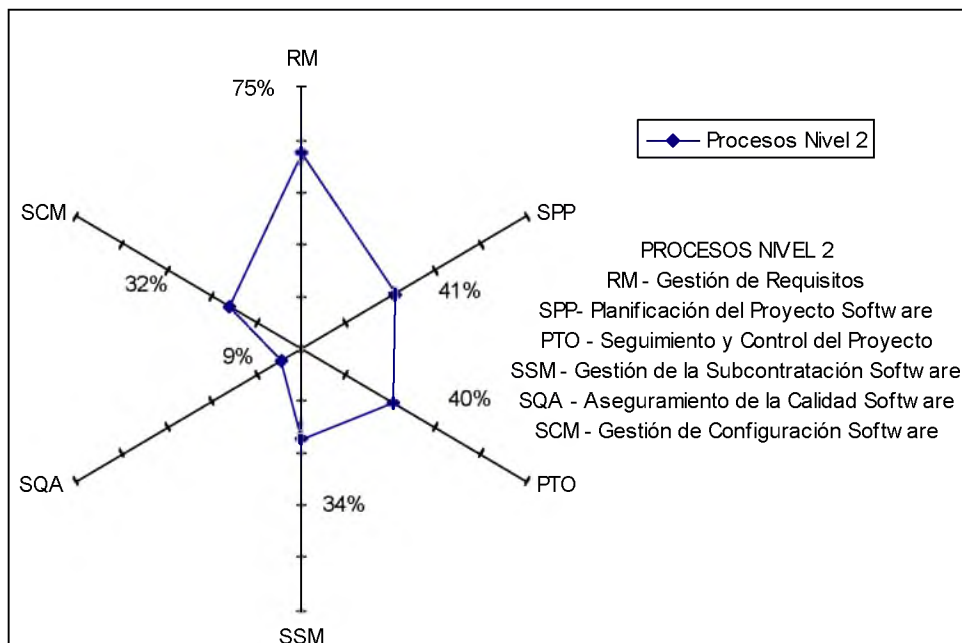


Figura 3.4. Ejemplo de una gráfica de Kiviati para mostrar los resultados de evaluar diferentes procesos

3.3. Fase Realizar

La creación del plan de acción para conducir la Mejora del Proceso de Software es una de las actividades más importantes en la iniciativa –y una de las más frecuentemente rechazadas por el

personal. En esta fase es donde el Grupo de Trabajo desarrolla o actualiza un plan de acción basándose en la visión de la empresa, junto con los hallazgos obtenidos en la fase “Evaluar”.

Por lo tanto, esta fase se deberá repetir cuantas veces sea necesario dado que es común que la empresa carezca de un plan de acción en el primer ciclo de DERE. Para aquellas organizaciones que ya estén en un ciclo posterior de DERE, esta fase puede iniciarse por la necesidad de actualizar planes, objetivos o directrices previas. Así, el Grupo de Trabajo tiene la responsabilidad de crear un plan estratégico de acción. En cierto sentido, se trata de crear una línea base a través de los procesos o técnicas evaluadas en la fase “Evaluar” (fortalezas y debilidades). Posteriormente, considerando esta línea base, las mejoras son desarrolladas, puestas en práctica, y desplegadas en la empresa. En este sentido, las mejoras propuestas por el Grupo de Trabajo deben ser probadas en la empresa a través del “pilotaje”. Dado que se pretende “pilotar” las soluciones antes de implementarlas en toda la empresa, es importante que el Grupo de Trabajo esté atento a los efectos producidos por los cambios introducidos en cada área, puesto que este es el momento en que se le pide al personal que cambie ciertas cosas que hacía previamente y puede presentar el rechazo cultural. Considerando lo anterior, las mejoras deben ser creadas a través de dos enfoques básicos: (1) el Grupo de Trabajo se enfoca en problemas específicos y desarrolla una solución usando proyectos piloto para validarla y refinarla, o (2) el Grupo de Trabajo se enfoca en un proceso en particular y desarrolla refinamientos incrementales con proyectos piloto para probarlos. Así pues, esta fase identifica doce actividades que deben realizar de la siguiente manera:

1. *Revisar la misión/visión de la empresa:* El propósito de esta actividad es vincular claramente el plan de acción con la visión de la empresa, de modo que el programa de mejora esté correctamente alineado y sea consistente con otras actividades que se estén realizando dentro de la misma. Dado que DERE está orientado a MiPyMEs será común que los mismos integrantes del Grupo de Trabajo sean los que crearon el programa de mejora, y hayan participado en todas las actividades anteriores por lo que no será necesaria la capacitación adicional.
2. *Determinar problemas claves en la empresa:* Las necesidades clave de la empresa deben ser claramente definidas, medibles, y entendidas para ofrecer una visión común al Grupo de Trabajo. De esta manera los problemas detectados en la fase “Evaluar” deben usarse para priorizar los esfuerzos en la mejora. Esta priorización puede hacerse de manera sencilla reuniendo al Grupo de Trabajo, al jefe de proyectos, y a los representantes de la alta dirección y determinar qué es más importante en ese momento en relación a los objetivos de la empresa.
3. *Identificar esfuerzos de mejora (planificados) actuales y anteriores:* Es recomendable que antes de empezar a mejorar se identifiquen otras iniciativas que la empresa puede tener en marcha. Esto se realiza durante la revisión de la visión y esfuerzos anteriores de mejora de las empresas. Por lo general, la mayoría de las empresas tienen muchos diferentes esfuerzos de mejora en curso, y a menudo estas iniciativas son descoordinadas y compiten entre sí por los escasos recursos de las MiPyMEs. Si una empresa desea maximizar la eficacia de su inversión en la Mejora del Proceso de Software, deberá evaluar todas las iniciativas en curso y determinar cuánto se está invirtiendo en cada una y en total, y posiblemente priorizar dada la limitada cantidad de personal.
4. *Establecer metas de Mejora del Proceso de Software que sean realistas y mensurables:* Ahora que los resultados de la evaluación se han entendido y se han priorizado las soluciones a éstos, existen suficientes datos para tomar los objetivos generales a largo y corto plazo,

desarrollados en la fase “Definir”, y hacerlos más específicos. Esto se realiza mediante la incorporación de la medición en la “Plantilla para la Definición del Compromiso” (véase Anexo B.1), específicamente el su punto 3.5 Indicadores de Proceso. Tales indicadores permiten establecer valores alcanzables para las mejoras paulatinas que permitan mantener el compromiso definido al inicio de la iniciativa de mejora.

5. *Crear/actualizar la guía del plan de acción:* Esta guía de acción será inútil si es construida en el vacío y solamente un pequeño grupo cree en ella. Para que sea útil, ésta tiene que ser “vendida” y aceptada en toda la empresa mediante el consenso. Es decir, la guía del plan de acción que se desarrolle debe ser comunicada a la empresa. Es claro que al inicio de la iniciativa de mejora fue necesario establecer los componentes y el personal de la empresa para que este plan funcionara, por lo tanto es justo que se les comunique qué hay en él y lo que se espera de las personas involucradas. En este sentido, DERE introduce la “Plantilla para la Guía del Plan de Acción” (véase Anexo B.4) para realizar estas actividades.
6. *Capacitar al Grupo de Trabajo para las tareas técnicas:* El Grupo de Trabajo en una MiPyMEs debe estar compuesto por voluntarios de la misma empresa que serán afectados directamente por el proceso de mejora y que están interesados en trabajar sobre la mejora. Este grupo ya ha sido identificado desde la fase “Definir”. Dado que se introducirán mejoras que modifican la forma de trabajo habitual a través del uso de técnicas y procedimientos formales, es necesario que el Grupo de Trabajo sea capacitado antes de “pilotar” los proyectos.
7. *Desarrollar las soluciones:* En esta actividad se desarrollan soluciones a los problemas encontrados durante la fase “Evaluar”. El propósito de estas soluciones es mejorar los procesos que la empresa ha determinado importantes para satisfacer las necesidades actuales de negocio. Es importante que la solución seleccionada sea compatible con la cultura de la organización, de modo que sea fácilmente aceptada e institucionalizada. En este sentido, DERE introduce los “Formatos para Mejorar los Procesos” (véase Anexo D) para proporcionarle a las MiPyMEs la ayuda necesaria para realizar de forma correcta las actividades recomendadas por MoProSoft®. A modo de ejemplo, la Figura 3.5 muestra un reporte de actividades requerido por la actividad 1.3 del proceso DMS y su guía de uso. Cabe mencionar que todas las plantillas creadas en esta tesis no están disponibles para cualquier empresa que desee implementar MoProSoft® en sus tareas diarias, por lo que fue necesaria crearlas para que cualquier MiPyMEs las pueda utilizar.
8. *Refinar el proceso (enfoque centrado en el proceso):* El enfoque centrado en el proceso aborda la comprensión de un proceso específico, identificado en la fase “Definir”, y aplica mejoras incrementales al mismo. Este enfoque es útil para alcanzar mejoras a largo plazo en dicho proceso. Sin embargo, debido a las presiones inmediatas e incertidumbres típicas de las MiPyMEs con niveles bajos de madurez; sin embargo, es difícil mantener este enfoque por lo que se requiere del compromiso organizacional sólido para mantener la motivación y entusiasmo del personal.
9. *Pilotar las soluciones potenciales:* Los proyectos piloto se utilizan para probar las soluciones definidas. Estas soluciones requieren de adaptación y refinamiento para que encajen en los proyectos de la MiPyMEs, y los proyectos piloto ayudan precisamente a determinar las necesidades de adaptación y las directrices para el resto de la empresa. En este sentido, es posible que el Grupo de Trabajo realice varios proyectos pilotos para probar una solución y pueden existir varias iteraciones entre el desarrollo de ésta y los pasos del “pilotaje” para su

implementación en toda la empresa. Es normal que durante este trabajo sea necesario refinar toda la documentación generada para el proceso hasta asegurarse que, por ejemplo, el conjunto de “Formatos para Mejorar los Procesos” (véase Anexo D) se ajusta a las necesidades de la MiPyMEs.

10. *Capacitar al resto del personal:* A medida que las soluciones son aceptadas, éstas deben ser implementadas en otras partes de la empresa, por lo que el Grupo de Trabajo deberá entrenar al personal que utilizará el nuevo proceso. Es claro que en este proceso surgirán problemas y puede requerirse de modificaciones adicionales a los procedimientos y/o plantillas definidas en la actividad 7. Por lo tanto, es posible que se requiera de capacitación adicional en términos de conocimientos y habilidades técnicas necesarias para cumplimentar las plantillas creadas. En este sentido, debe existir el soporte continuo del Grupo de Trabajo (quien creó toda la documentación del proceso) para explicar cualquier herramienta, método, procedimiento, formato, etc.

Reporte de actividades No. [Inserte aquí el texto]						
Datos generales						
Nombre del proyecto	[inserte aquí el texto]					
Referencia del proyecto	[inserte aquí el texto]					
Nombre del RDM	[inserte aquí el texto]					
Fecha del reporte	[Fecha]	Período comprendido	[Fecha] al [Fecha]			

Registro de actividades							Responsable
Actividad del Plan de Desarrollo	Fecha de inicio		Fecha de fin		Esfuerzo (horas)		
	Planeado	Real	Planeado	Real	Planeado	Real	
[inserte aquí el texto]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[inserte aquí el texto]
Tiempo total de producción					[Rev]	[Rev]	

Instrucciones de llenado de la plantilla

Campo	Descripción
<i>Datos generales</i>	
Nombre del proyecto	Nombre que le fue asignado al proyecto en el Plan de Desarrollo.
Referencia del proyecto	Referencia única que se le da a cada proyecto para referirse a él de forma abreviada. Estas referencias suelen tener identificadores que ayudan al personal a determinar la naturaleza del proyecto sin consultar el nombre o descripción completa del mismo. El estilo de las referencias debe ser definido por la empresa y debe ser respetado por todo el Equipo de Trabajo.
Nombre del RDM	Nombre del Responsable del Desarrollo y Mantenimiento de Software que fue asignado al proyecto en cuestión.
Fecha de reporte	Fecha en la que se está entregando el reporte de avance. Es obvio que cada modificación de avance requerirá la modificación del reporte existente y por ende la fecha será actualizada.
Periodo comprendido	Periodo de tiempo que se está reportando. Por ejemplo, la Fecha del reporte puede ser 08/06/2013 y se están reportando las actividades que se realizaron entre el 01/04/2013 y el 07/06/2013.
<i>Registro de actividades</i>	
Actividad del plan de desarrollo	Las actividades listadas en esta columna deben corresponder con las actividades que hayan sido registradas en el Plan de Desarrollo.
Fecha de inicio – Planeado	Fecha planeada por el RAPE para iniciar la actividad correspondiente.
Fecha de inicio – Real	Fecha real en que inició la actividad correspondiente.
Fecha de fin – Planeado	Fecha planeada por el RAPE para finalizar la actividad correspondiente.
Fecha de fin – Real	Fecha real en que finalizó la actividad correspondiente.
Esfuerzo (horas) – Planeado	Horas planeadas por el RAPE para realizar la actividad correspondiente.
Esfuerzo (horas) – Real	Horas reales para realizar la actividad correspondiente.
Responsable	El responsable de cada actividad tomada del Plan de Desarrollo y que acordó comprometerse a realizarla en tiempo y forma a través de la Minuta de Lanzamiento.

Figura 3.5. Ejemplo de formato para mejorar el proceso de DMS y guía de uso

11. *Reunir toda la información del proceso y subirla a la base de conocimiento:* Es probable que durante la capacitación del personal haya sido necesario modificar o crear algunas plantillas, formatos o procedimientos y artefactos intermedios. En este sentido, es necesario que el

Grupo de Trabajo reúna en la base de conocimiento toda la documentación asociada con el proceso para su mantenimiento y soporte a largo plazo.

12. *Desplegar la solución:* El propósito de esta actividad es diseminar en toda la empresa las soluciones probadas. Es decir, las mejoras deben ser institucionalizadas en toda la empresa. Las actividades de la institucionalización podrían ir desde crear nuevas herramientas de software, adecuar las instalaciones, documentar todo lo relativo al nuevo proceso, hasta contratar nuevo personal. Por otro lado, el personal de software debe ser capaz de continuar sin mucha orientación y soporte, pero debe ser capaz de recurrir al asesoramiento cuando sea necesario. En este sentido, los jefes de proyectos que fueron integrados al Grupo de Trabajo asumirán el rol de guardia de los procesos y deberán hacerse cargo de las asesorías futuras.

3.4. Fase Extender

Antes de comenzar con el siguiente ciclo de DERE es necesario revisar qué ha pasado durante el ciclo que termina y prepararse para el siguiente, antes de comenzar. Por lo tanto, esta fase consiste en revisar y analizar las lecciones aprendidas de las fases anteriores, incorporar mejoras en las actividades de DERE (e.g., qué no se hizo, qué se puede hacer mejor, qué se puede omitir), revisar la información obtenida, revisar y evaluar las metas, evaluar el compromiso del Grupo Directivo, y revisar el plan de mejora para determinar qué objetivos falta por cubrir. Con el objetivo de asegurar el compromiso del Grupo Directivo durante el nuevo ciclo, el Grupo de Trabajo debe informar al Grupo Directivo sobre los resultados del primer ciclo, los indicadores de mejora y las metas alcanzadas hasta ahora, los cambios que ocurren en la infraestructura, etc. En este sentido, las siguientes cuatro actividades enmarcan la realización de esta fase:

1. *Reunir las lecciones aprendidas:* El propósito de esta actividad es asegurar que toda la información sobre las lecciones aprendidas esté disponible para su revisión antes de comenzar el siguiente ciclo de DERE. Sin esta información será difícil recordar las actividades que se realizaron durante las fases anteriores. Así pues, el Grupo de Trabajo debe reunir estas lecciones aprendidas a lo largo de todas las actividades previas y almacenarlas en la base de conocimientos de la empresa; si no fuera así, las lecciones deben reunirse al entrar a la fase “Extender”.
2. *Analizar las lecciones aprendidas:* El propósito de esta actividad es asegurar que el marco de trabajo que se está usando para conducir la mejora (DERE) es el mejor que se puede utilizar. Ahora que se cuenta con toda la información del ciclo previo es necesario reflexionar sobre las actividades que se hicieron o que no se hicieron. En este sentido, es necesario aprender de los errores u omisiones que se pudieron realizar en el ciclo anterior y modificar y cambiar el enfoque para no repetirlos. Cualquier mejora que se haga a DERE permitirá realizar cambios con mayor eficacia, reducirá la resistencia al cambio y permitirá que la mejora del proceso avance a un ritmo más rápido
3. *Revisar el compromiso:* Se ha establecido desde el inicio que el compromiso del Grupo Directivo y del personal operativo es fundamental para el éxito de la iniciativa de mejora. Al igual que como se realizó la primera vez a través de la fase “Definir”, es necesario que el Grupo de Trabajo se asegure de que se sigue teniendo el patrocinio y compromiso adecuados para comenzar el nuevo ciclo de mejora. En este sentido, se puede utilizar la misma “Plantilla para la Definición del Compromiso” (véase Anexo B.1) para analizar y documentar los compromisos que ya han sido cubiertos y los que falta por cubrir.

4. *Continuar con la mejora:* El propósito de esta actividad es moverse hacia la fase principal del programa de mejora (Evaluar) y comenzar el nuevo ciclo. Para esto es necesario revisar y/o actualizar los enfoques documentados en el documento de compromiso, revisar los indicadores de mejora faltantes, y obtener la aprobación para continuar con la iniciativa.

4. Caso de estudio

En este capítulo se presentan los resultados obtenidos al utilizar el marco de trabajo DERE dentro de una MiPyMEs del Estado de Oaxaca. Estos resultados provienen de la ejecución de un caso de estudio que permitió establecer una iniciativa de mejora basada en MoProSoft® dentro un entorno real. Como parte del caso de estudio fue necesario definir proyectos piloto dentro de la MiPyMEs, para implementar las mejoras propuestas y comparar los resultados antes y después de utilizar a DERE. En este sentido, un caso de estudio se utiliza habitualmente como una estrategia de investigación empírica en el campo de los Sistemas de Información (SI) y a menudo es utilizado para investigar una única entidad o fenómeno dentro de un espacio de tiempo específico. Así, el investigador puede recopilar información detallada sobre, por ejemplo, un único proyecto durante un periodo de tiempo. Por lo tanto, los casos de estudio son adecuados para la evaluación de métodos y herramientas de la Ingeniería de Software en un entorno industrial, dado que pueden evitar problemas de escalado [Wohlin et al., 2012]. Sin embargo, a pesar de que los casos de estudio son fáciles de planificar, su principal desventaja es que los resultados que se obtengan no se generalizan dado que es posible que los efectos observados en una situación típica no sean aplicables a otra [Yin, 2009].

Así pues, existen tres diferentes estrategias para desarrollar un caso de estudio:

1. Comparar los resultados obtenidos de una nueva propuesta y una línea base.
2. Desarrollar dos proyectos 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 el contexto de esta tesis se optó por elegir la primera de las estrategias anteriores, para lo cual se consideraron como línea base los resultados obtenidos en una primera evaluación sobre los procesos de la empresa, tal y como lo establece la fase “Evaluar” de DERE. Posteriormente, cuando los planes de acción fueron implementados para subsanar las deficiencias encontradas, de acuerdo a lo indicado por la fase “Realizar” de DERE, la evaluación fue repetida para comparar los resultados de la línea base contra los obtenidos después de utilizar DERE para establecer la iniciativa de mejora dentro de la MiPyMEs.

En este sentido, con este caso de estudio se busca validar la hipótesis presentada al inicio de esta tesis, la cual plantea la idea de que a través del diseño de un marco de trabajo es posible que una

empresa pequeña alcance con mayor facilidad una mejora significativa de sus procesos de desarrollo.

4.1. Definición del caso de estudio

El caso de estudio se enfocó en el establecimiento de una iniciativa de Mejora del Proceso de Software usando como guía al marco de trabajo DERE. Un aspecto importante en relación a la elección de la empresa participante fue la disposición de ésta a colaborar con las actividades requeridas para que pudiera alcanzar la certificación en los Niveles 1 y 2 de MoProSoft®. Es decir, a diferencia de cualquier otra empresa esto facilita el establecimiento de un compromiso serio, junto con la disposición adecuada de recursos en materia de personal e instalaciones, y el tiempo necesario para llevar por buen camino la iniciativa de mejora. Estos factores son cruciales puesto que la falta de compromiso es el principal factor en el fracaso de iniciativas de Mejora del Proceso de Software dentro de las empresas. Otro factor importante que fue considerado en cuanto a la elección de la empresa participante, fue el delimitar apropiadamente las expectativas que ésta hubiera podido tener sobre este tipo de proyectos, ya que, si los objetivos y resultados esperados por los directivos no son realistas, el esfuerzo de mejora puede resultar un fracaso. Por último, es necesario mencionar que la consecución de un nivel de MoProSoft®, jamás debe convertirse en el principal objetivo de la iniciativa de mejora, dado que esto puede ser perjudicial para el negocio. Al contrario de esta idea, la Mejora del Proceso de Software debe verse como un mecanismo para alcanzar los objetivos reales de la empresa, lo que a la postre tendrá como resultado natural la consecución de un nivel específico del modelo.

Así pues, después de establecer los objetivos reales del caso de estudio con la empresa, denominada en adelante “EmpSoft” por motivos de confidencialidad, se recogió información relacionada con la naturaleza de la misma (véase Tabla 16).

Tabla 16. Perfil de la empresa participante en el caso de estudio.

Nombre de la MyPyMEs	Número de empleados	Actividad principal que desarrolla
EmpSoft	20	Desarrollo de software contable y páginas web

En este sentido, EmpSoft está conformada por tres departamentos:

- Departamento administrativo: se encarga de realizar todas las gestiones administrativas con los clientes y con la plantilla de trabajadores. Este departamento es controlado por un grupo de directivos relacionados con la gestión empresarial, quienes son apoyados por personal de recursos humanos y logística.
- Departamento contable: se encarga de realizar todas las gestiones de índole contable con los clientes y al interior de la empresa. Este departamento es controlado por un directivo relacionado con la contabilidad organizacional.
- Área de desarrollo: es el área encargada de desarrollar los proyectos de software requeridos por los clientes. Dentro de este personal se encuentra un jefe de proyectos quien lidera al equipo de trabajo.

En relación a la experiencia previa de EmpSoft con iniciativas de mejora, los responsables de la empresa expresaron lo siguiente:

- La empresa se ha enfocado por más de cuatro años al desarrollo de software contable con el objetivo de brindar soluciones tecnológicas al mercado estatal. Sin embargo, a pesar de que se han obtenido beneficios económicos importantes, regularmente los problemas durante el desarrollo de los proyectos les impide terminar el producto a tiempo y sin errores, lo que evita que pueda comprometerse con un mayor número de proyectos. De hecho, se argumentó que la gravedad era tal que se habían recibido ya dos demandas por incumplimiento en la entrega de dos productos.
- La empresa argumentó poseer un conocimiento limitado sobre las iniciativas de mejora y los modelos de procesos, particularmente MoProSoft®. De acuerdo a la información vertida por su director general, la empresa reconoció que debe mejorar su forma de trabajo para crecer y por eso la intención de probar con el marco de trabajo DERE para facilitar esta labor.

Una vez que se definieron las características de EmpSoft y que se decidió el desarrollar un caso de estudio como método de experimentación para alcanzar una mejora significativa, fue necesario establecer la forma en que las fases de DERE serían aplicadas. En este sentido, la Figura 4.1 muestra que después de establecer un compromiso con EmpSoft fue necesario obtener datos sobre la línea base inicial, generar un plan de acción que guiara las mejoras que subsanaran las deficiencias halladas en el proceso, probar dichas soluciones antes de su institucionalización dentro de la organización y, por último, comparar los resultados de la línea base con los resultados obtenidos después de la implementación de las mejoras. De esta forma se tendría suficiente información para validar la hipótesis establecida al inicio de esta tesis.



Figura 4.1. Proceso establecido para la incorporación de DERE al caso de estudio

Es importante mencionar que antes de iniciar el proceso mostrado en la Figura 4.1, se organizó una reunión de trabajo con la alta dirección de EmpSoft con la intención de conocer al Consejo Directivo y que fuera éste quien, de primera mano, brindará la información relacionada con el interés por mejorar. De esta manera fue posible observar una motivación real que provenía del nivel más alto de la organización, algo que es bastante difícil de encontrar puesto que con frecuencia la

motivación proviene del personal técnico y la directiva no está consciente de, o no reconoce, la existencia de los problemas.

4.1.1. Definir el compromiso de mejora

La fase “Definir” de DERE inició formalmente con la planeación de dos talleres orientados a los dos grupos de personas identificados en la Figura 3.2: el personal directivo y el equipo de trabajo (incluido el jefe de proyectos). El taller orientado a la alta dirección pretendió concientizar a los directivos sobre la importancia de su compromiso para con la iniciativa de mejora que está por comenzar. Este taller suele tener una duración aproximada de cinco horas, dependiendo el nivel de inquietud de los directivos con el trabajo que se planea hacer. Durante dicho taller se suelen utilizar diapositivas que proporcionan información relevante sobre la Mejora del Proceso de Software, sus beneficios, su enfoque ajustado a las particularidades de la empresa, y cómo éste puede cambiar la forma tradicional de trabajo que la empresa ha tenido hasta ese momento (véase Figura 4.2).



Figura 4.2. Ejemplo de diapositivas utilizadas en EmpSoft para concientizar a la alta dirección sobre la importancia del compromiso

Por otro lado, el taller orientado al equipo de trabajo buscó concientizar al personal técnico sobre la importancia de su participación en la iniciativa de mejora que está por comenzar. Así pues, el taller debe proporcionar información relevante sobre las actividades que serán realizadas con el

objetivo de reducir el rechazo cultural. En este sentido, es común que durante una iniciativa de mejora el personal técnico se sienta amenazado porque cree que será “juzgado” y tiene miedo de perder el trabajo. Sin embargo, uno de los puntos cruciales de este taller es aclarar que las iniciativas de mejora intentan identificar problemas y no culpables, por lo que debe quedar claro que las actitudes negativas del personal pueden llevar a resultados incorrectos que con seguridad incrementarán el tiempo dedicado a la mejora y requerirán cambios más drásticos. La duración de este taller puede requerir un poco más de tiempo, posiblemente siete u ocho horas, puesto que la mayoría de las veces son más las inquietudes del personal técnico que las de la alta dirección.

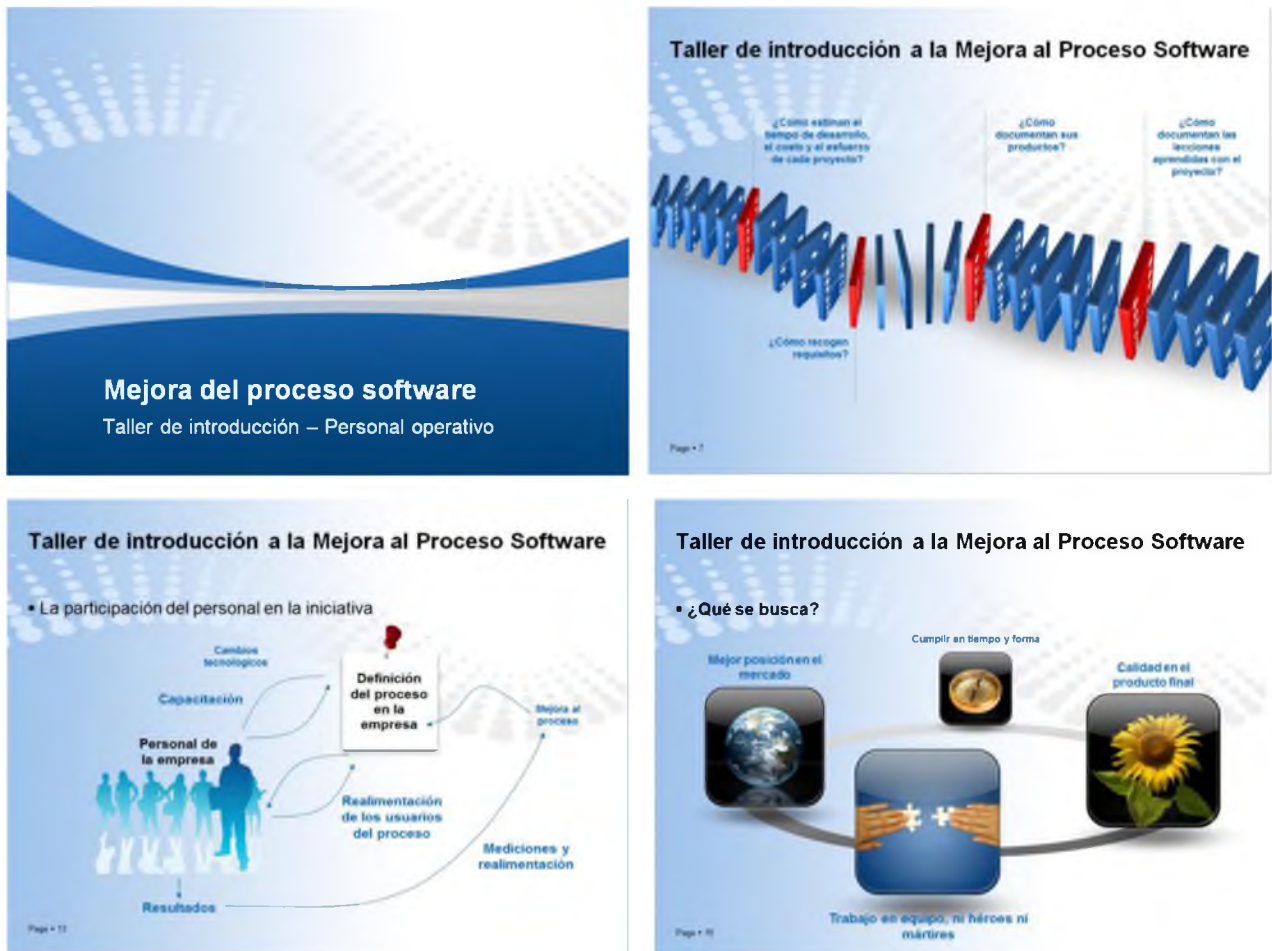


Figura 4.3. Ejemplo de diapositivas utilizadas en EmpSoft para motivar al personal técnico a participar en la mejora

De manera adicional, se organizó un taller de capacitación para la alta dirección y el jefe de proyectos (como representante del personal técnico) sobre MoProSoft® con la intención de cumplir con las actividades marcadas por DERE, específicamente las actividades 1, 3, y 6 de la fase “Definir”. Esta capacitación proporciona una visión general sobre la estructura de MoProSoft® y cómo éste puede ser ajustado a las características de una empresa pequeña (véase Figura 4.4). Durante y después de esta capacitación debe ser posible recibir ruegos y preguntas para que sea posible delimitar correctamente la iniciativa de mejora, entendiendo el nivel de compromiso y recursos que requiere cada esfuerzo. De esta forma fue posible que el personal de EmpSoft, alta dirección –representada por el consejo directivo- y operativo/técnico –representado por el jefe de proyectos, analizara a través del consenso en qué lugar se encontraban, a dónde querían llegar, y

cómo se visualizaban a largo plazo. Este análisis es recomendado cuando la empresa no posee experiencia previa con modelos de proceso, como MoProSoft®, puesto que a través de la discusión se establece un curso claro y real que aborde las áreas primordiales de la organización.



Figura 4.4. Ejemplo de diapositivas utilizadas en la capacitación del personal directivo de EmpSoft

Así, como resultado de los talleres organizados y la capacitación proporcionada fue posible establecer los elementos requeridos por la fase “Definir”, concretamente los siguientes:

- Se establecieron las principales motivaciones para iniciar una iniciativa de mejora, éstas se resumen de la siguiente forma:
 - El cálculo del costo de cada producto regularmente es obtenido sin utilizar procedimientos formales (o bien datos históricos) que permitan obtener el mayor beneficio de cada proyecto. Además de lo anterior, el costo es determinado por el equipo de logística sin contemplar las peculiaridades de un proyecto de software, lo que ocasiona que se ponga en riesgo la relación costo-beneficio para la empresa.
 - La determinación de un calendario de trabajo de acuerdo al tamaño, complejidad, y alcance de los proyectos es una tarea ineficiente puesto que se tiene conocimiento de retrasos importantes que afectan: (1) la obtención del

- beneficio económico (i.e., mientras más tiempo se retrase la entrega, más dinero se gasta en salarios que ya no son cubiertos por el costo del producto), (2) la imagen de EmpSoft frente a los clientes actuales, clientes potenciales, y su competencia, (3) la motivación del personal de desarrollo dado que deben repetir continuamente las tareas para alcanzar una entrega inmediata. De manera similar que en el cálculo del costo del producto, el calendario es generado por el equipo de logística que no tiene conocimiento sobre estimaciones para proyectos de software.
- El aseguramiento de la calidad de los productos creados en EmpSoft es un proceso que se realiza de forma ineficiente. Regularmente los productos son entregados con errores que el cliente detecta al momento de recibir su producto. En concreto, las pruebas no están siendo realizadas correctamente y se pone en riesgo la calidad del producto final.
 - Derivado del problema anterior, EmpSoft está a merced de cualquier cliente al modificar continuamente el producto sin determinar el impacto o importancia de cada cambio requerido por el mismo. Este problema afecta: (1) el calendario de entrega y por ende el costo del producto, (2) la motivación del personal al verse obligados a ceder ante cualquier petición del cliente, y (3) la gestión del proyecto dado que se pierde el control de las actividades.
 - Finalmente, el personal posee habilidades diferentes y regularmente es difícil integrar equipos que funcionen. Aunado a esto, la contratación de personal nuevo afecta el trabajo cotidiano en EmpSoft puesto que muchas veces su integración, a la forma de trabajo actual, se realiza con bastante retraso afectando así los tiempos de entrega establecidos
- Se identificaron las necesidades de negocio y conductores para la mejora, concretamente se especificó que era crucial corregir los problemas a nivel técnico que originaban las complicaciones relacionadas con la mala calidad de los productos y lo tardío de su entrega. Cabe mencionar que uno de los factores cruciales para determinar lo anterior fue el miedo a recibir más demandas por atrasos en la entrega de productos.
 - Se planteó la necesidad de que el personal de EmpSoft recibiera capacitación técnica en aquellas áreas problemáticas que fueran primordiales para alcanzar las necesidades actuales de negocio.
 - Se definieron las metas generales de la iniciativa de mejora las cuales delimitaron, de cierta manera, el alcance del caso de estudio. Así pues, se estableció lo siguiente:
 - Orientarse a la Categoría de Operación de MoProSoft®, específicamente a los procesos de Administración de Proyectos Específicos (APE) y Desarrollo y Mantenimiento de Software (DMS). Esta selección se debió a que APE y DMS eran los procesos que inciden directamente en los problemas técnicos que afrontaba EmpSoft en ese momento.
 - Alcanzar la mejora en los Niveles 1 y 2 de MoProSoft® puesto que ambos representan los escalones iniciales de una mejora integral. Es decir, se escogió un enfoque de adentro hacia afuera que resolviera primero los problemas a nivel equipo de desarrollo –puesto que al cubrir las actividades de estos niveles

es posible generar productos con la calidad correcta, a tiempo y con el costo establecido- y posteriormente resolver los problemas de la alta dirección.

- Disponer de un espacio de trabajo con una computadora por cada integrante del equipo de trabajo, un proyector para la capacitación de todos los integrantes del mismo, una sala de juntas de fácil acceso y alejada del resto del personal. Cabe mencionar que la norma NMX- la proporcionó el co-director de la tesis, por lo que no fue necesario adquirirla.

Con el objetivo de establecer un compromiso real con los puntos identificados anteriormente, se propuso un calendario de trabajo, y se diseñó una tentativa de propuesta de mejora a nivel general que incluyó indicadores de mejora establecidos por el personal directivo de EmpSoft (i.e., objetivos de mejora a corto plazo que les indicaran que en realidad se estaba mejorando). Toda esta información fue recogida en la “Plantilla para la Definición del Compromiso” (véase Anexo B.1), la cual fue firmada por todos los involucrados para lanzar de inmediato la iniciativa de mejora.

4.1.2. Evaluar los procesos

De acuerdo a la fase “Evaluar”, es necesario identificar un equipo de evaluación que determine cuáles son los problemas que están afectando gravemente el rendimiento del proceso utilizado por la empresa. En este sentido, DERE recomienda utilizar un mismo equipo para desarrollar todas las actividades de la iniciativa de mejora puesto que las pequeñas empresas, que a menudo carecen de personal, no pueden cubrir las recomendaciones de los modelos de mejora creados para las grandes empresas en el sentido que debe utilizarse personal diferente para cada equipo. Por lo tanto, se decidió formar un Grupo de Trabajo de cinco participantes para realizar la evaluación. La decisión de escoger a un grupo de personas de la misma empresa como equipo de evaluación corresponde a las experiencias que se han documentado en casos anteriores en diferentes partes del mundo, las cuales evidencian que es más recomendable que el personal sea evaluado por personas conocidas (i.e., del mismo lugar de trabajo) puesto que se reduce considerablemente el rechazo a responder con la verdad. Este Grupo de Trabajo se formó considerando el conocimiento, las habilidades, experiencia, y la antigüedad del personal en el desarrollo de sus labores como miembros del equipo de trabajo, y la motivación para realizar las labores que implican ser parte de un equipo de mejora. Ser parte del Grupo de Trabajo es un trabajo de tiempo completo durante los periodos de capacitación y ejecución de la evaluación. Por lo tanto, este personal debió estar libre del trabajo diario para desarrollar sus deberes como equipo de evaluación.

Así, el Grupo de Trabajo fue capacitado en el método de evaluación mostrado en la fase “Evaluar”, el cual fue complementado con la técnica de modelado grupal y entrevistas para contrastar la información recogida (véase Figura 4.5). Adicionalmente, fue necesario solicitar acceso a la base de datos donde el personal de EmpSoft almacenaba todos los activos del proceso que han utilizado para desarrollar sus productos. Esta información, considerada como “evidencia objetiva del proceso” es útil cuando existen discrepancias significativas entre las respuestas proporcionadas por el personal. Así, fue necesario utilizar la “Plantilla para el Plan de Evaluación” (véase Anexo B.2) para cumplir con las actividades 1 y 2 de la fase “Evaluar”.

En este sentido, los lineamientos establecidos por EvalProSoft® fueron seguidos para conducir una autoevaluación (i.e., previa a la evaluación oficial) considerando los valores establecidos por NYCE para la obtención de cualquiera de los niveles manejados por la norma

(véase Tabla 17). Es decir, EmpSoft debía alcanzar un porcentaje de cobertura mayor al 85% para obtener una calificación de “Completamente alcanzado” y obtener su certificación.



Figura 4.5. Ejemplo de diapositivas utilizadas en la capacitación del equipo de evaluación

Tabla 17. Escala normativa de calificación definida por EvalProSoft®.

Calificación	Descripción	Porcentaje de cobertura
N	No alcanzado	0% al 15%
P	Parcialmente alcanzado	>15% al 50%
A	Ampliamente alcanzado	>50% al 85%
C	Completamente alcanzado	>85%

Dado que la evaluación pretende evaluar al personal en cuanto al cumplimiento de las actividades recomendadas por MoProSoft®, no en el proceso que hasta ahora habían utilizado, se consideró importante dar una capacitación sobre los aspectos generales de los procesos escogidos en la fase “Definir”. Así, antes de comenzar con la evaluación se impartió capacitación adicional al

personal técnico de EmpSoft para que éste se comenzara a relacionar con las actividades y terminología utilizada por el modelo.

Al mismo tiempo que se impartió la capacitación sobre APE y DMS (véase Figura 4.6), en conjunto con el personal se procedió a redactar un diccionario de términos que permitió identificar si ya existían algunas cosas en la empresa que estuvieran relacionadas con MoProSoft® pero que tuvieran diferente nombre. Esta sencilla práctica elimina también de forma considerable el rechazo cultural del personal, puesto que éste se ve implicado en el proceso de mejora y empieza a creer que su participación es importante.

The figure consists of four slides from a presentation titled "Introducción a la norma NMX-I-059/2-NYCE-2005".

- Slide 1 (Top Left):** Title slide: "Introducción a la Norma NMX-I-059/2-NYCE-2005" and "Módulo avanzado".
- Slide 2 (Top Right):** "Tareas de APE:"
 - A1 Planificación. Establecimiento del proceso metódico diseñado para obtener el producto del proyecto a través de un plan.
 - A2 Realización. El proceso metódico es ejecutado para alcanzar los objetivos planeados en el plan.
 - A3 Evaluación y Control. Monitorización del plan contra lo actual para aplicar acciones correctivas si fuere necesario.
 - A4 Cierre. Se termina un ciclo del proyecto o bien el proyecto es finalizado.
- Slide 3 (Bottom Left):** "Realización"
 - Actividades: 1. Asignar las tareas al Equipo de Trabajo, incluyendo a los subcontratistas.
- Slide 4 (Bottom Right):** "Cierre"
 - Entradas: Plan del Proyecto (N1), Configuración de Software (N1 y N2), Plan de Procesos (N2).
 - Salidas: Configuración de Software (N1 y N2), Plan del Proyecto (N1), Plan de Desarrollo (N1), Minuta(s): Documento que describe el objetivo de las reuniones realizadas, los puntos tratados y los acuerdos (N2).

Figura 4.6. Ejemplo de diapositivas utilizadas para la capacitación sobre APE y DMS

Posteriormente las técnicas de evaluación fueron aplicadas de la siguiente manera. En primer lugar, se organizó un taller con todo el personal técnico para aplicar la técnica de modelado grupal siguiendo una estrategia de "coaching". De esta manera, se dirigió al personal en la obtención de un modelo que representara, a través de la notación de un diagrama de flujo, el proceso utilizado para desarrollar los productos de software. En este sentido, el "coaching" asume que una persona toma el control del taller y propone ideas de un modelo que es ajustado a través de la participación del resto de participantes. La Figura 4.7 muestra el resultado obtenido después de haber terminado el taller de modelado.

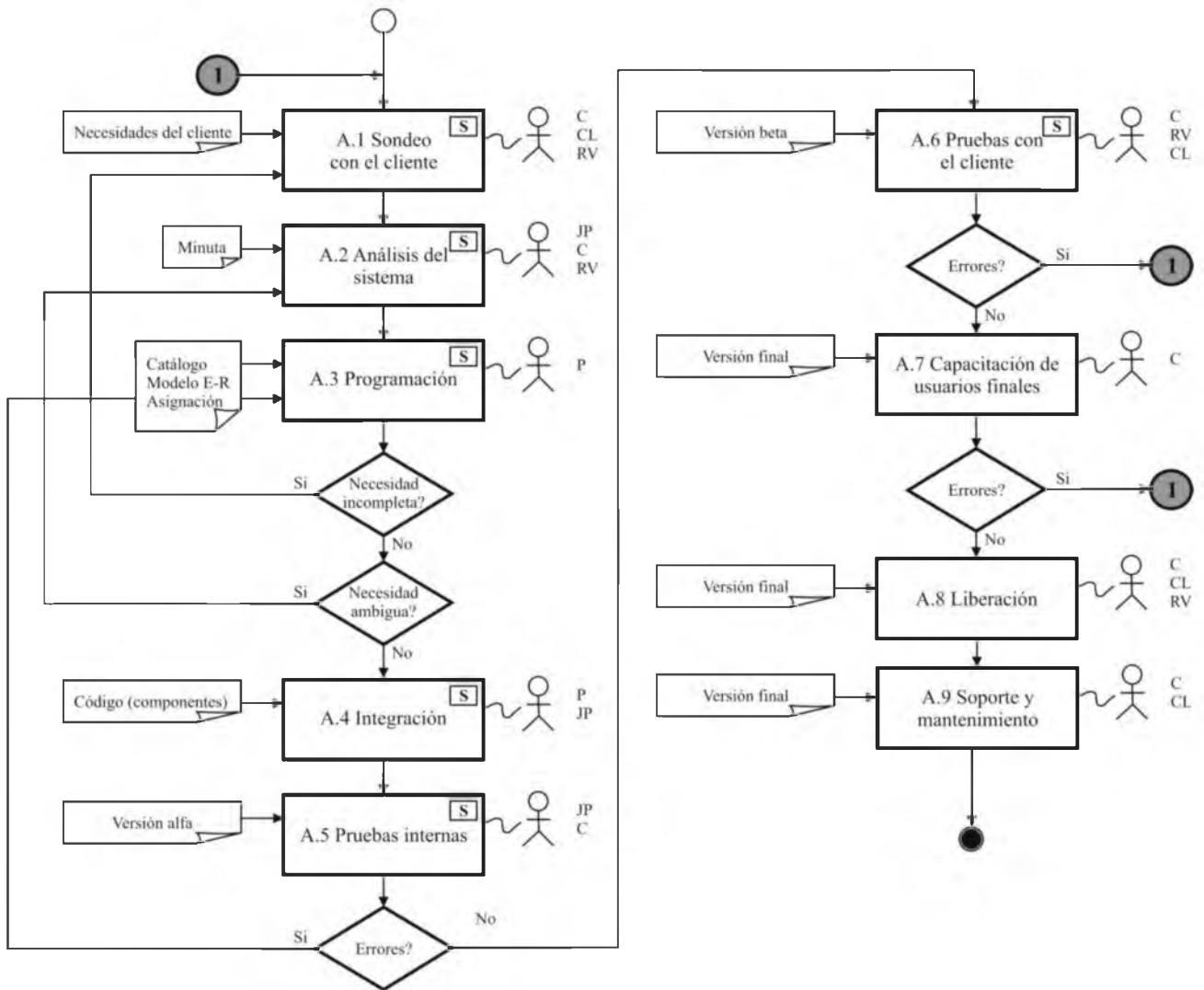


Figura 4.7. Proceso modelado por el personal técnico de EmpSoft

Con el modelo terminado se procedió a analizar cuidadosamente las diferencias existentes entre éste y el modelo propuesto por MoProSoft®. La Figura 4.8 muestra que fue fácil intuir que la mala calidad del producto se podía deber a la ausencia de una fase destinada a la obtención de requisitos, tal y como lo establece MoProSoft®. En este sentido, lo anterior proporcionó una pista sobre que era importante analizar, a través de la aplicación de los cuestionarios, si la fase “A.1 Sondeo con el cliente” realizada por EmpSoft (véase modelo a la izquierda de la Figura 4.8) cubría los aspectos más importantes establecidos por las fases “A.1 Realización de la fase de inicio” y “A.2 Realización de la fase de requerimientos” de MoProSoft® (véase modelo a la derecha de la Figura 4.8). De manera similar, este ejercicio sirvió para determinar que algunos entregables del modelo establecido por MoProSoft® no estaban siendo contemplados en las fases o bien no tenían el mismo nombre, por lo que fue posible determinar que iba a ser necesario analizar con cuidado la evidencia objetiva entregada por la empresa. Así, sería posible determinar si, por ejemplo, el entregable “Necesidades del cliente”, asociado a A.1, era equivalente al entregable “Especificación de requerimientos”, definido por MoProSoft® en A.2.

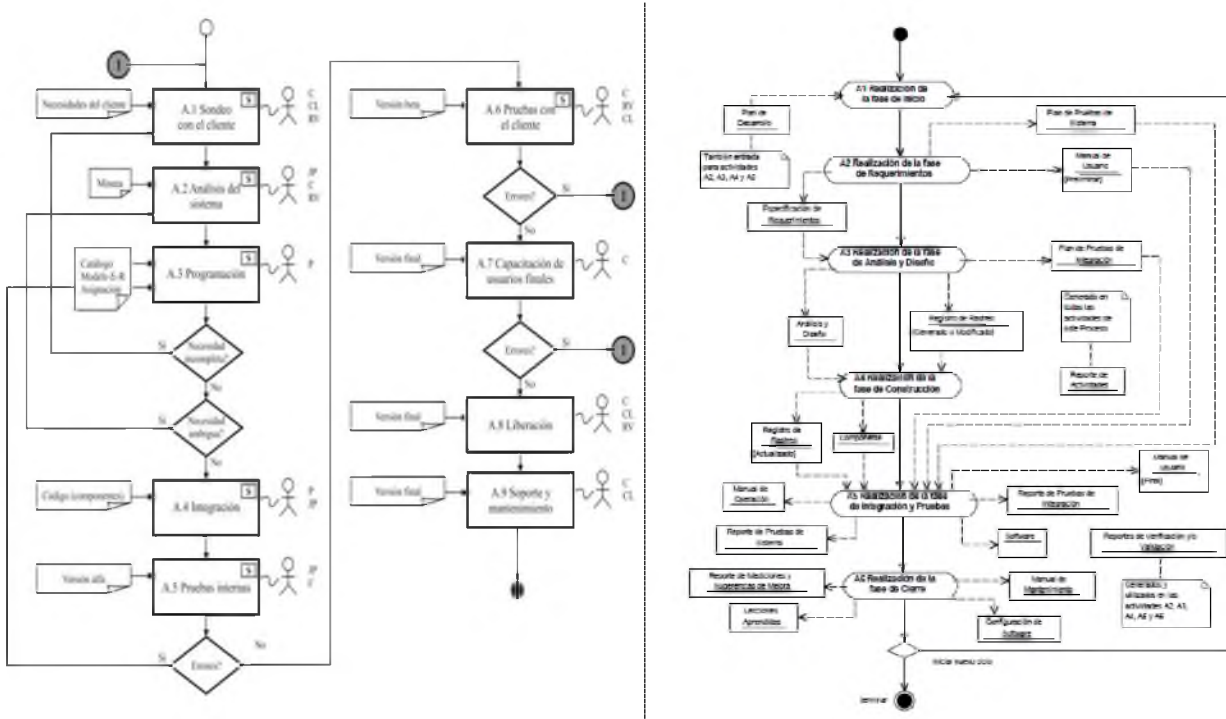


Figura 4.8. Proceso modelado vs proceso establecido por MoProSoft®

A continuación se preparó el paquete de evaluación que incluyó a los cuestionarios de evaluación creados para EmpSoft, y que se muestran en el Anexo C “Cuestionarios de evaluación”, y se dirigió al Grupo de Trabajo en su aplicación. Cabe mencionar que, para obtener mejores resultados, es recomendable impartir una charla antes de la evaluación a los encuestados para explicar, de nueva cuenta, que no se intenta culpar a nadie de la mala ejecución del proceso actual, sino que se pretende identificar dónde se encuentran los problemas. De no realizar esta charla se correría el riesgo de obtener resultados ajenos a la verdadera forma de trabajo puesto que el personal tiende a mentir cuando se le pregunta, por ejemplo, si realiza tal actividad o utiliza tal herramienta como soporte en su trabajo diario. Las Tablas 18 y 19 resumen los resultados obtenidos con la aplicación de los cuestionarios en la evaluación de APE y DMS, respectivamente.

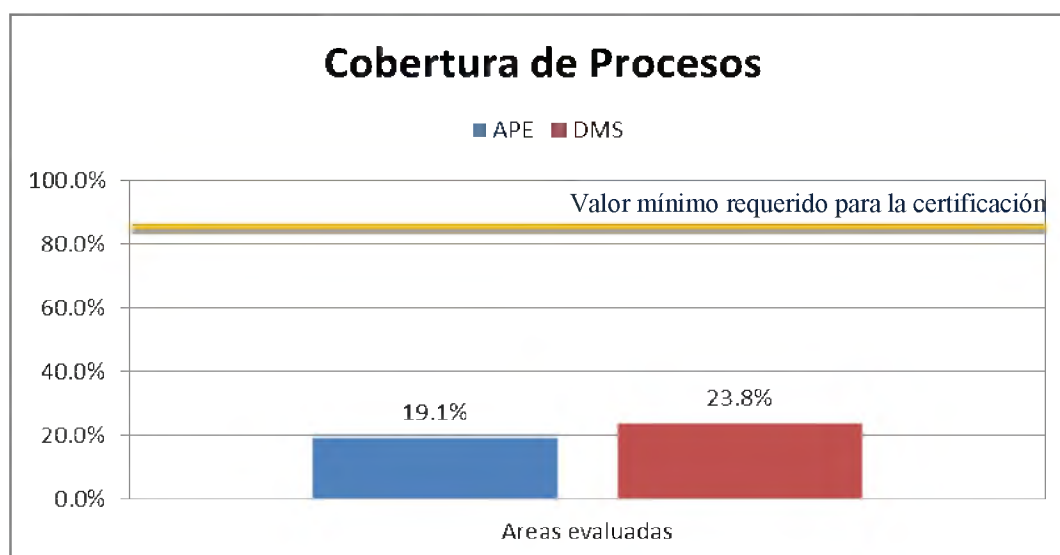
Tabla 18. Porcentajes de cobertura obtenidos con la evaluación de las fases de APE.

Fases	Escala de calificación manejada por NYCE			
	N	P	A	C
A1. Planificación	19.3%			
A2. Realización	28.6%			
A3. Evaluación y Control	9.1%			
A4. Cierre	19.6%			

Tabla 19. Porcentajes de cobertura obtenidos con la evaluación de las fases de DMS.

Fases	Escala de calificación manejada por NYCE			
	N	P	A	C
A1. Inicio	21.4%			
A2. Requisitos	27.1%			
A3. Análisis y Diseño	20.0%			
A4. Construcción	34.3%			
A5. Integración y Pruebas	30.9%			
A6. Cierre	8.9%			

De esta forma, la fase “Evaluar” proporcionó información útil relacionada con el estado actual de los procesos de EmpSoft, identificando aquellas prácticas que requerían atención inmediata. A través del análisis se determinó de que a pesar de que existía un proceso para el desarrollo y mantenimiento de software dentro de la empresa, éste no estaba documentado, no contaba con una descripción estándar, no utilizaba artefactos definidos para su correcta ejecución, y no disponía de personal con roles adecuados para desarrollar las actividades que eran pobremente definidas. En este sentido, se explicó que era necesario incorporar mejoras al proceso actual para que la adherencia con lo establecido en la norma NMX-I-059/02-NYCE se incrementara a los niveles requeridos por el Órgano Certificador (véase Figura 4.9). Además de estas mejoras, sería necesario analizar las funciones de cada integrante del equipo de trabajo para determinar habilidades de acuerdo a su motivación, conocimientos y experiencia.

**Figura 4.9.** Coberturas totales de APE y DMS al finalizar la evaluación

Con esta información fue posible rellenar la “Plantilla para entrega de los Resultados de la Evaluación” (véase Anexo B.3) para presentar los hallazgos a la empresa, considerando la cobertura y desviación estándar por cada actividad. Con base a estos resultados se obtuvieron los puntos débiles, puntos fuertes, y las actividades que debían explorarse con mayor profundidad para poder dilucidar si constituían un punto fuerte o un aspecto a mejorar. Los hallazgos fueron evidencia clara de que los procesos evaluados (en los Niveles 1 y 2 de MoProSoft®) se encontraban pobremente implementados en EmpSoft.

En este sentido, con la intención de concientizar al Consejo Directivo de EmpSoft sobre la gravedad de los resultados, se utilizó la información recogida a través de los cuestionarios y las entrevistas para definir posibles consecuencias en caso de no corregir los problemas. Concretamente se definieron las siguientes:

- La empresa posiblemente podría seguir realizando proyectos para los clientes habituales. Sin embargo, como una empresa desarrolladora de software que busca la consolidación en el mercado estatal y nacional, no sería capaz de llevar a buen término proyectos externos puesto que se seguirían presentando desvíos sin control en el calendario, lo que terminaría por elevar el costo estimado del producto impactando negativamente en las ganancias de la empresa.
- Los productos que la empresa genere seguirían presentando bajos niveles de calidad puesto que se seguirían realizando pruebas unitarias o de integración desordenadamente y sin la incorporación del cliente o usuario final.
- La culminación de los proyectos seguiría dependiendo de que el cliente desee aceptar la versión final del producto o no, puesto que sin procedimientos establecidos éste seguirá modificando los requisitos a su gusto hasta que logre funcionalidades que no habían sido acordadas al inicio y que por ende afectarían el calendario y costo iniciales.
- Los productos creados en la empresa seguirían abordando funcionalidades diferentes a las requeridas por los clientes. Lo anterior se establece bajo la base de que no existe actualmente un procedimiento formal para capturar correctamente las necesidades del cliente.

Finalmente, y una vez que se tuvieron cubiertas las actividades 1, 2, 3 de la fase “Evaluar”, se procedió a presentar los hallazgos de la evaluación a los dos grupos de participantes: alta dirección y personal técnico/operativo. DERE, a través de la actividad 4 de la fase “Evaluar”, recomienda preparar un bosquejo de la guía del plan de acción que plantee en esta misma presentación cómo serán abordados los problemas. Así, durante la presentación de los hallazgos se utilizaron diapositivas que explicaban la naturaleza del problema detectado y una posible solución (véase Figura 4.10). A partir de dicha evidencia, la siguiente fase de DERE estableció soluciones que fueron probadas en proyectos piloto y que se enfocaron, a través de iteraciones, en las áreas más urgentes de acuerdo al razonamiento del Consejo Directivo.

4.1.3. Realizar las mejoras

Culminada la evaluación, la fase “Realizar” comenzó con la revisión de la misión y visión de la EmpSoft puesto que, como se ha dicho anteriormente, cualquier esfuerzo de mejora debe estar alineado estrechamente con los objetivos de negocio de la empresa. En este sentido fue necesario modificar ambas puesto que, considerando objetivos a largo plazo, la empresa definió una misión

demasiado ambiciosa que afirmaba “*ser el mejor proveedor de soluciones informáticas contables de Latinoamérica*”. Es importante mencionar que esta modificación es obligatoria puesto que en esta fase se pretende establecer metas realistas que generen rápidamente mejoras. Por lo tanto, establecer una estrategia de mejora que se alinea con la misión actual iría más allá de este trabajo de tesis. Así pues, el cambio sugirió establecer una misión realista a corto plazo que permitiera visualizar a EmpSoft en una mejor posición competitiva dentro del mecano Estatal, y a partir de una evaluación sobre los objetivos establecidos aumentar la cobertura de su mercado.



Figura 4.10. Ejemplo de diapositivas utilizadas para la presentación de los hallazgos de la evaluación

El determinar qué problemas atender primero dentro de EmpSoft requirió un análisis minucioso de los problemas junto con el Consejo Directivo. Es decir, de todos los problemas detectados en la evaluación es necesario que la empresa, junto con el asesor en Mejora del Proceso de Software, decidiera cuáles son más importantes de acuerdo a los objetivos actuales de negocio. Este análisis estableció el siguiente razonamiento: se sabía que APE no estaba implantada en EmpSoft y que la calificación sería baja al inicio (un 19.1%, de acuerdo a los resultados mostrados anteriormente), pero DMS (o su variante dentro de la empresa) se había estado desarrollando todos los días y no existió una diferencia importante entre su calificación (23.8%, de acuerdo a los resultados mostrados anteriormente) y la de APE. Por lo tanto, se establecieron las siguientes conclusiones:

- Muchos de los problemas con las fases iniciales del proceso actual de EmpSoft se relacionaban con el personal de logística, que son los responsables de establecer el contacto inicial con el cliente.
- Los procedimientos del proceso no estaban documentados, y si lo estaban la documentación era ambigua.
- Los proyectos no fueron delimitados y se terminó por rebasar el tiempo de entrega y el costo planificado.
- Existían problemas graves de comunicación e integración.
- No existían roles específicos para cada persona de la plantilla de software, y
- Se carecía de capacitación específica en todo el proceso técnico.

En este sentido, el Consejo Directivo decidió que era apremiante resolver primero las deficiencias halladas en DMS, y se procedió a priorizar los problemas encontrados utilizando una tabla de análisis (véase Tabla 20).

Tabla 20. Ejemplo de priorización de los problemas clave detectados en EmpSoft.

Proceso	Prioridad	Orden	Causa de fracaso	Recomendación
DMS	Alta	1	El equipo de trabajo no revisa el plan de desarrollo para un proyecto, y no todos sus integrantes tienen el mismo conocimiento.	Adaptación de directrices para crear un entorno de trabajo participativo e incluyente.
	Alta	2	La identificación y consulta de fuentes de información es pobre, por lo que los requerimientos no son obtenidos correctamente.	Implementación de las guías básicas para la obtención de una especificación formal de requerimientos.
	Media	1	La descripción de la estructura interna del sistema y su descomposición en subsistemas no coincide plenamente con su descripción o no se encuentra totalmente alineada con los requerimientos obtenidos.	Implementación de actividades relacionadas con el análisis de la especificación de los requerimientos para generar una descripción de la estructura del sistema, que incluya a sus subsistemas y componentes y que defina las interfaces entre ellos.
	Baja	1	No existe personal asignado a roles específicos para el desarrollo, mantenimiento o actualización de manuales de sistema.	Establecer una descripción de roles que facilite el delegar las responsabilidades con base a roles establecidos por MoProSoft®.

Con los problemas priorizados se procedió a reorganizar a la plantilla de software, de tal forma que fuera posible formar un grupo, con responsables de cada área, que haría la labor del Grupo de Trabajo en la fase “Realizar”. Cabe mencionar que, cómo se dijo anteriormente, estos participantes son los mismos que realizaron la evaluación en la fase anterior de DERE (véase Figura

4.11). Así, el esfuerzo de mejora se centró en preparar soluciones para los problemas que fueron previamente analizados y priorizados junto con el Consejo Directivo.



Figura 4.11. Formación del grupo de trabajo para la fase "Realizar"

Posteriormente se desarrolló la guía del plan de acción utilizando la "Plantilla para la Guía del Plan de Acción" (véase Anexo B.4). Esta guía fue la base para que la alta dirección tomara decisiones en cuanto al establecimiento de compromisos visibles para las siguientes etapas del ciclo de mejora, asegurando que la iniciativa de mejora siguiera estando alineada con la visión corporativa y los objetivos de negocio de EmpSoft. Así, se establecieron los siguientes objetivos para DMS:

- Refinar el proceso actual de EmpSoft y documentarlo con mejoras de tal forma que fuera posible realizar sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software (tanto nuevos como aquellos que pudieran ser modificados), cumpliendo siempre con los requisitos especificados por el cliente.
- Establecer prácticas eficientes para recoger los requisitos del cliente.
- Trabajar bajo los umbrales de calidad establecidos por la empresa para que cualquier producto que sea creado garantice la satisfacción de su usuario.

Considerando estos objetivos, se establecieron las siguientes metas:

- Establecer un estándar de documentación que permitiera gestionar toda la información que el equipo de trabajo genere y comparta.
- Establecer una cultura de trabajo de calidad total mediante la validación y verificación continuas sobre el trabajo de cada integrante del equipo de trabajo.

- Aumentar la participación del cliente para que reconozca la importancia de analizar cualquier petición de cambio a los requerimientos de un producto.
- Lograr una mayor participación del usuario final en la definición de los requerimientos.
- Mejorar las habilidades de documentación de los programadores para que realicen pruebas unitarias sobre sus componentes siguiendo el estándar de documentación definido.
- Reducir el número de defectos que los productos contienen al momento de ser liberados para entrega.
- Mejorar la comunicación entre el equipo de trabajo y los responsables de gestión.

En relación a la infraestructura necesaria para alcanzar estas metas, la guía del plan de acción estableció que la alta dirección debería comprometerse a respetar los nuevos roles y a proporcionar lo siguiente:

- Un salón de reuniones con capacidad para que el Grupo de Trabajo revisara y discutiera los procedimientos establecidos para el nuevo proceso. Este mismo salón sería ocupado posteriormente para desarrollar los proyectos piloto. Se requirió de un proyector de tiempo completo.
- Acceso a un servidor específico para que se pudiera montar la base de conocimientos (requerida por MoProSoft®) de la empresa.
- Un proyector electrónico adicional para que fuera posible impartir la capacitación al personal y revisar los procedimientos y plantillas creadas para el nuevo proceso. Dicha capacitación incluyó los siguientes temas:
 - Para analistas/diseñadores: obtención y especificación de los requerimientos del cliente con base al estándar IEEE-1998 Std. 830 (e.g., obtención de diagramas de casos de uso, traducción de diagramas de casos de uso a requerimientos formales), uso del estándar de documentación, uso de procedimientos y plantillas del proceso DMS, (específicamente aquellas relacionadas con la fase de Análisis de MoProSoft®).
 - Para programadores: documentación de su trabajo y realización de pruebas unitarias y de integración, uso del estándar de documentación, uso de procedimientos y plantillas del proceso DMS (específicamente aquellas relacionadas con la fase de Construcción de MoProSoft®).
 - Para revisores/documentadores: uso del estándar de documentación, uso de procedimientos y plantillas del proceso DMS (específicamente aquellas relacionadas con las actividades de validación y verificación del proceso).

Con la capacitación definida fue necesario establecer un calendario que definiera el tiempo necesario para la adopción del nuevo proceso definido para EmpSoft. Dicho proceso corresponde a

las fases de trabajo establecidas por MoProSoft® con una pequeña modificación al final, la cual incorporó a la fase de capacitación (véase Figura 4.12).

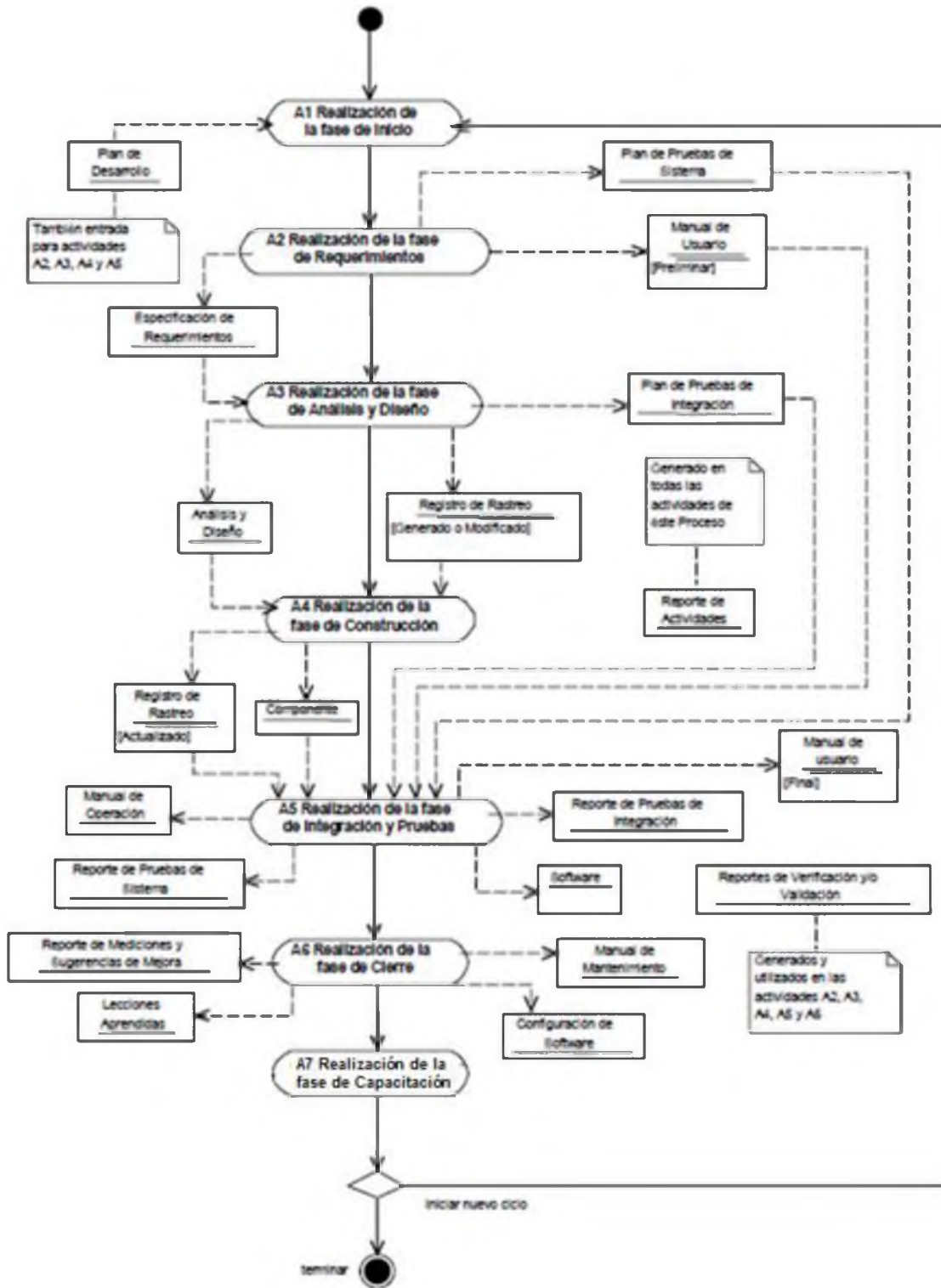


Figura 4.12. Diseño del nuevo proceso para EmpSoft

La implantación de este nuevo proceso requirió de la ejecución de tres proyectos piloto:

- Sistema de Administración de Bibliotecas (réplica del caso de estudio UWON): Se planteó el diseño y desarrollo de un sistema para hacer más eficiente la gestión de la biblioteca de una universidad en término de préstamos, reservaciones, consultas y compra/venta de libros, revistas o material multimedia.
- Sistema de Gestión de Nómina: Se consideró el desarrollar un sistema que fuera capaz de gestionar la nómina de una empresa incluyendo el cálculo de impuestos, retenciones, primas vacacionales, estímulos, y demás.
- Sistema de Control Escolar: Se construyó un sistema enfocado a administrar los datos escolares y financieros de un plantel educativo.

Por último, se definió que el éxito en el establecimiento y capacitación del nuevo proceso de EmpSoft se podía ver afectado por los siguientes riesgos:

- Incorporación o despido de personal de la plantilla técnica de la empresa una vez que la ejecución de la guía del plan de acción hubiera iniciado.
- Falta de apoyo o interés de la alta dirección o del personal técnico/operativo.
- Otras iniciativas de mejora que actualmente se estuvieran desarrollando (que no hubieran sido notificadas a la tesista) y que pudieran interferir con los tiempos programados en esta tesis.
- Exceso de capacitación dado que el Grupo de Trabajo podría desconocer la mayor parte de las actividades recomendadas por la iniciativa para mejorar la situación actual de la empresa. Este exceso generaría, sin duda alguna, un atraso en el calendario propuesto.

De manera similar, se contempló que la única limitación para el establecimiento del nuevo proceso de EmpSoft se relacionaba con la misma decisión de la empresa en el sentido de escoger un solo proceso para la mejora (DMS en su Nivel 2). Esto implicó que los ocho procesos restantes para alcanzar el Nivel de madurez 1 que marca la norma NMX-I-59-NYCE-2005 deberían ser mejorados en una iniciativa posterior.

Por último, el Grupo de Trabajo controló el avance sobre las actividades de mejora, el término de las actividades y los entregables, la información sobre las próximas entregas y el desfase en los tiempos de terminación de las actividades; de esta manera se cubrieron las diez actividades propuestas por la fase “Realizar”.

4.1.4. Extender el proceso y continuar con la mejora

La última fase de DERE se enfocó a recoger y analizar las lecciones aprendidas durante cada ciclo de mejora, de forma tal que fuera posible identificar y corregir problemas en el proceso y eliminarlos en ciclos posteriores. Algunos ejemplos de las lecciones aprendidas, recogidas por el Grupo de Trabajo y personal de la empresa, son mostrados en las Tablas 21 y 22. Cabe mencionar que estos errores fueron de mucha utilidad para corregir y adecuar las plantillas propuestas en la fase “Realizar” (y mostradas en el Anexo B) a las necesidades de EmpSoft.

Tabla 21. Ejemplo de lecciones aprendidas para el proyecto “Sistema de Administración de Bibliotecas”.

Fase(s):	Requerimientos/Diseño
Proyecto(s) piloto relacionado(s):	Sistema de Administración de Bibliotecas
Errores encontrados (ambigüedad del proceso, inconsistencia de las plantillas, gestión operativa)	
Error	Sugerencia de corrección
1. La plantilla de <i>Registro de Rastreo</i> está incompleta.	Al requisitar la información de la plantilla, ésta me funciona para verificar los requerimientos y los casos de uso que le corresponden a cada requerimiento. Sin embargo, considero que si la plantilla incluyera información sobre los subsistemas y las clases que le corresponden quedaría más completa.
2. La información de los documentos no coincide.	En la revisión me faltó prestar más atención a lo plasmado por las AN y verificar que las necesidades si estuvieran siendo cubiertas por lo plasmado en el documento de requerimientos. En este sentido considero que debe existir un reporte de verificación previo a la fase de diseño.
3. Al momento de redactar los requerimientos muchas cosas fueron consideradas como obvias.	Las acciones, mensajes y botones que no se contemplaron en el documento de requerimientos, por ser consideradas como obvias, afectaron al momento de realizar el diseño de sistema ya que no se comprendía muy bien lo plasmado en la especificación de requerimientos. Se recomienda que el responsable de AN colaboré con los programadores para que éste resuelva las dudas que pudieran surgir.
Inconformidades (aspectos negativos) / Comentarios generales (aspectos positivos)	
1. La mayoría de la documentación es tediosa de rellenar y siento que muchas cosas se repiten, pero al final es parte del estar acostumbrados a realizar las cosas de una manera más “práctica” y desordenada. Los cambios siempre son buenos y en este caso particular siento que nos ha beneficiado mucho porque hemos aprendido muchas cosas nuevas, y recordado otras en las que hace tiempo no estábamos involucrados como equipo de trabajo.	
2. Siento que nos sigue fallando un poco la comunicación y a veces nos brincamos el orden de las cosas o las órdenes de los responsables. En el caso de las AN, por ejemplo, debe existir un mecanismo que permita que expresen y defiendan sus ideas de diseño, y los programadores deben respetarlas o establecer el consenso.	
3. Por último, he observado que al haber adquirido nuevo conocimiento algunos integrantes del equipo de trabajo han expresado inconformidades salariales y la lucha de egos se hacen presente.	
En este punto de la iniciativa, ¿cuál es tu compromiso con la mejora?	
Alto	Justificación: Mucho tiempo trabajamos con la forma antigua, en el poco tiempo que llevamos con este nuevo proceso he notado cambios importantes que poco a poco se van acentuando más, eso es lo que me motiva a continuar.
Medio	--
Nulo	--

Tabla 22. Ejemplo de lecciones aprendidas para el proyecto “Sistema de Control Escolar”.

Fase(s):	Requerimientos/Diseño
Proyecto(s) piloto relacionado(s):	Sistema de Control Escolar
Errores encontrados (ambigüedad del proceso, inconsistencia de las plantillas, gestión operativa)	
Error	Sugerencia de corrección
1. En la <i>Especificación de Requerimientos</i> no se le dio importancia a pequeños aspectos que en ese momento parecieron insignificantes pero que en lo posterior repercutieron, al grado de solicitar cambios.	En la especificación de requerimientos se deben contemplar todos los aspectos que pudieran parecer insignificantes, por más mínimos y absurdos que parezcan.
2. Muchos requerimientos no fueron entendibles, claros y presentaron ambigüedad. Específicamente cuando se hacía referencia de un requerimiento a otro, éste no era nombrado de forma correcta.	Se debe ser más claros y menos ambiguos en la redacción de nuestro trabajo.
3. La presentación de los documentos no fue la correcta.	Debemos ser más profesionales en la elaboración de los documentos (redacción y estandarización) y por supuesto en la revisión y aprobación de los mismos. Se debe seguir al pie de la letra el estándar de documentación que se ha definido.
4. Hubo incongruencias en la <i>Especificación de Requerimientos</i> y la descripción de los casos de uso porque no se verificó la congruencia de los contenidos de ambos documentos.	La RAN debe leer detalladamente los documentos y asegurarse de que los requerimientos coincidan con la descripción de los casos de uso para que en la realización de los diagramas de secuencia no tengamos que detenernos, porque existe incongruencia.
Inconformidades (aspectos negativos) / Comentarios generales (aspectos positivos)	
1. Cada persona debe conocer detalladamente las actividades y obligaciones que le competen de acuerdo a su rol.	
2. Los responsables deben documentarse sobre conceptos “básicos” que tienen que ver con el desarrollo de software, por ejemplo: diagrama de clases, diagrama E-R, casos de uso.	
En este punto de la iniciativa, ¿cuál es tu compromiso con la mejora?	
Alto	Justificación: Me encuentro dispuesta a mejorar en todos los aspectos necesarios que ayuden a perfeccionar el proceso de desarrollo de software de la empresa, puesto que es un trabajo en equipo y requiere mucha coordinación, responsabilidad y disciplina. Me encuentro abierta a cualquier crítica constructiva que me ayude a mejorar mi desempeño.
Medio	--
Nulo	--

Estas lecciones fueron recogidas a lo largo de seis ciclos de mejora por cada proyecto piloto durante los once meses que duró toda la iniciativa de mejora. Al final de cada iteración, se revisó el compromiso de la alta dirección y, una vez que éste era renovado, se continuaba con el siguiente ciclo.

4.1.5. Comparación evaluación final con la línea base inicial

De acuerdo a lo indicado por la Figura 4.1, una iniciativa de mejora debe realizar un n número de iteraciones (o ciclos) hasta alcanzar los indicadores de mejora establecidos por la empresa. En el caso particular de este caso de estudio, EmpSoft decidió guiarse por los indicadores establecidos en la Tabla 16. Dado lo anterior, se decidió junto con la empresa recoger información relacionada con el proceso de tal forma que fuera posible evidenciar una cobertura total de, por lo menos, el 86% requerido por el Órgano Certificador. Así pues, la comparación inició con la recogida de datos de todos los ciclos de mejora con el objetivo de establecer una base comparativa que permitiera determinar la eficiencia (o no) del marco de trabajo DERE en la conducción de iniciativas de mejora.

Así, la Figura 4.13 muestra que, después de seis ciclos de mejora, EmpSoft logró alcanzar el indicador establecido en un tiempo de once meses. Utilizando esta figura como referencia comparativa es posible observar los beneficios de utilizar DERE mediante la comparación del valor obtenido antes de utilizar dicho marco de trabajo contra el valor obtenido con su adopción. La línea amarilla denota el porcentaje de cobertura mínima requerido por el Órgano Certificador.

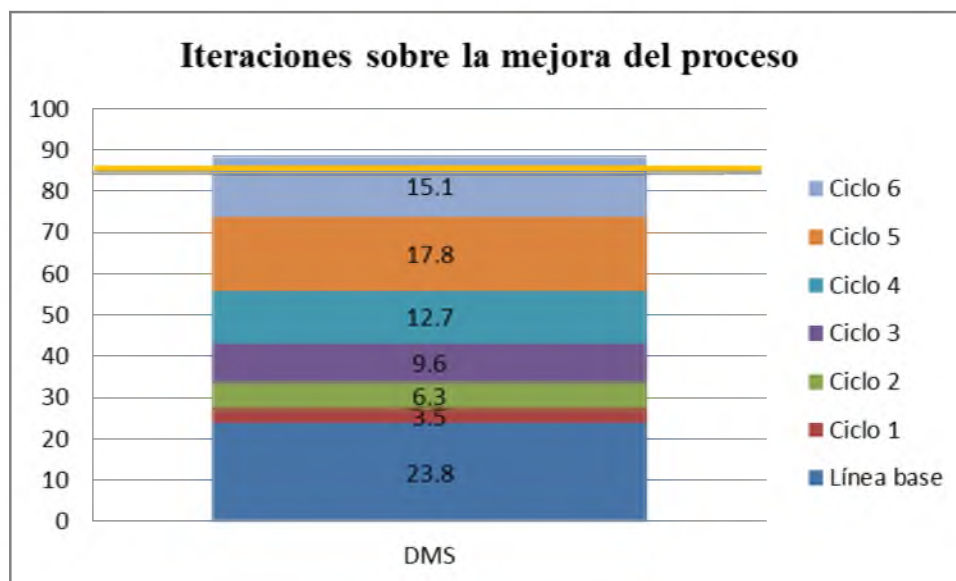


Figura 4.13. Incremento de la cobertura de DMS a través de la iniciativa de mejora

En relación al tiempo dedicado a la iniciativa de mejora, es importante mencionar que ésta requirió un poco más del tiempo considerado al inicio puesto que no se contempló que fuera necesario capacitar al personal de EmpSoft en todos los aspectos técnicos del desarrollo de software. Sin embargo, cabe mencionar también que el tiempo destinado a esta capacitación solamente afectaría a los primeros proyectos piloto realizados en EmpSoft y que posteriormente éste sería nulo. Por último, para apoyar a EmpSoft en la adopción del nuevo proceso, fue necesario programar reuniones semanales (en sábado y domingo) con el Grupo de Trabajo (incluido el jefe de proyectos)

para evaluar el calendario y los objetivos de la iniciativa de mejora. Este personal participó desde la formulación del plan inicial y trabajó de acuerdo a los objetivos establecidos por la fase “Definir”.

Aunado a lo anterior, se puede considerar que los resultados obtenidos con la aplicación de DERE en una iniciativa real cumplen el objetivo establecido al inicio de esta tesis: *“Diseñar un marco de trabajo para una empresa pequeña de tal forma que sea posible alcanzar una mejora significativa en sus procesos de desarrollo”*. En este sentido, es importante decir que la solución propuesta en esta tesis no requiere de una gran cantidad de recursos ni de una estructura organizativa específica para poder utilizarse en el entorno de las MiPyMEs. Otro aspecto relevante es el factor de la inversión económica, dado que, sin el requerimiento de una inversión importante destinada a la definición de los procesos dentro de una empresa pequeña como EmpSoft, se comprobó que el uso de DERE puede ayudar a este tipo de empresas en la mejora de sus actividades cotidianas.

Para finalizar, es importante mencionar que la hipótesis establecida en el contexto de la tesis (véase pág. 6 de esta tesis), es aceptada únicamente para la empresa que participó en el caso de estudio. De ninguna manera es posible que el marco de trabajo aquí propuesto produzca los mismos resultados en cualquier otra empresa debido a que esto requiere de la ejecución de más iniciativas de mejora definidas para una muestra más significativa. Sin embargo, de acuerdo con Juristo y Moreno (2013), en el contexto de la experimentación en Ingeniería de Software se afirma que los resultados positivos sobre una efectividad parcial, mostrados en un trabajo de investigación o reporte, evidencian que es posible establecer un caso de estudio con un mayor número de participantes y proyectos.

5. Discusión y conclusiones

Detectar que existen muchas MiPyMEs que no cuentan con los procesos adecuados para desarrollar software de calidad, fue lo que motivó e impulsó el desarrollo de este proyecto de tesis.

EmpSoft, fue de esas MiPyMEs que mostró interés para obtener la certificación en los Niveles 1 y 2 de MoProSoft, ya que detectó que sus procesos actuales para desarrollar software no eran los correctos para obtener buenos resultados.

En este sentido, EmpSoft al ser una empresa con más de cuatro años de experiencia en el desarrollo de software y al tener institucionalizados sus procesos, fue capaz de reconocer que éstos no eran correctos y decidió aplicar una iniciativa de mejora que le ayudará a crecer y a ser reconocida dentro y fuera del Estado de Oaxaca.

Así pues, la disposición y compromiso de esta empresa permitió validar la utilidad del Marco de Trabajo DERE. Es importante mencionar que durante el tiempo de capacitación que se tuvo con el personal de EmpSoft, se encontró que diseñar un marco de trabajo que permitiera guiar a la empresa en la mejora de los procesos de desarrollo de software no es una tarea fácil, ya que los problemas organizacionales encontrados requerían de un cambio radical. En este sentido, fue difícil que el personal se adaptara y entendiera el nuevo proceso a aplicar, principalmente al momento de documentar los requerimientos del cliente en relación con el sistema, ya que no se contaba con la suficiente experiencia y el conocimiento necesario para llevar a cabo dicha actividad. El equipo estaba acostumbrado a obviar muchas cosas que en este proceso eran más que indispensables.

Otra de las dificultades que se presentó fue la falta de comunicación entre el equipo de trabajo, la lucha de egos y la falta de interés que algunos integrantes del equipo fueron presentando debido a la impaciencia y al lento avance que se iba teniendo. Al presentarse todo esto fue difícil continuar con el plan de trabajo establecido.

Al término del proyecto se detectó que DERE podría ser mejorado para obtener resultados satisfactorios en menos tiempo. Adicionalmente se observó que sería correcto que antes de la implementación de DERE se capacite al equipo de trabajo en cuestiones básicas del área, tales como: análisis, diseño y arquitectura de software; puesto que en el caso particular de EmpSoft, la capacitación al personal hizo que los tiempos se incrementaran. Otro punto a considerar es definir un plan de motivación continua que ayude a que tanto el equipo de trabajo de desarrollo como los altos mandos no pierdan el interés en mejorar sus procesos.

En este contexto, se considera que la principal contribución de esta tesis reside en que ha sido posible desarrollar un marco de trabajo orientado a las MiPyMEs que proporcionará un conjunto de

guías y plantillas para que la conducción de una iniciativa de mejora no sea una labor tan complicada. Cabe mencionar que, en este sentido, hasta el día de hoy no existe un modelo o marco de trabajo similar al creado en esta tesis, por lo que se considera que este trabajo puede ser utilizado por futuras investigaciones para consolidar los esfuerzos de mejora en las MiPyMEs de México.

Finalmente, a pesar de los retos enfrentados se logró con el objetivo definido al inicio de la tesis, el cual consistía en *“Diseñar un marco de trabajo para una empresa pequeña de tal forma que sea posible alcanzar una mejora significativa en sus procesos de desarrollo”*. Es importante mencionar que actualmente, gracias a este trabajo de tesis, EmpSoft continúa trabajando con DERE para obtener la certificación oficial de los Niveles 1 y 2 de MoProSoft®.

Anexo A. Acrónimos

AMCIS	Asociación Mexicana para la Calidad en la Ingeniería de Software
AMIPCI	Asociación Mexicana de Internet
AMITI	Asociación Mexicana de la Industria de Tecnologías de la Información
APE	Administración de Proyectos Específicos
BSI	Bienes, Servicios e Infraestructura
CANIETI	Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de la Información
CBA-IPI	Método de Evaluación basada en CMM para la Mejora Interna del Proceso
CEPAL	Comisión Económica para América Latina y el Caribe
CMM	Modelo de Madurez y Capacidad
CMM-SW	Modelo de Madurez y Capacidad para Software
CMMI	Modelo de Madurez y Capacidad Integrado
CMMI-ACQ	Modelo de Madurez y Capacidad Integrado para la Adquisición
CMMI-DEV	Modelo de Madurez y Capacidad Integrado para el Desarrollo
CO	Conocimiento de la Organización
DERE	Marco de Trabajo para Definir, Evaluar, Realizar, y Extender el Proceso de Software

DESIN	Departamento de Estudios Institucionales
DMS	Desarrollo y Mantenimiento de Software
EvalProSoft	Método de Evaluación del Proceso de Software
GN	Gestión de Negocio
GP	Gestión de Procesos
GPY	Gestión de Proyectos
GR	Gestión de Recursos
HTML	Lenguaje de Marcas de Hipertexto
IEC	Comisión Electrotécnica Internacional
IIMAS	Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
INEGI	Instituto Nacional de Estadística y Geografía
INTECO	Instituto Nacional de Tecnologías de la Comunicación
ISO	Organización Internacional de Normalización
ISPI	Instituto para la Mejora del Proceso de Software
ITIL	Biblioteca de Infraestructura de Tecnologías de la Información
MDD	Millones de Dólares
MDM	Manejador de Documentos de MoProSoft
MiPyMEs	Micro, Pequeñas y Medianas Empresas
MoProSoft	Modelo de Procesos para la Industria del Software
MySQL	Sistema de Gestión de Bases de Datos Relacionales
NYCE	Agencia Mexicana de Normalización y Certificación
OCDE	Organización para la Cooperación y el Desarrollo Económicos
PIB	Producto Interno Bruto
PMBok	Cuerpo de Conocimiento sobre la Gestión de Proyectos
PROSOFT	Programa para el Desarrollo de la Industria del Software
PyME	Pequeña y Mediana Empresa

QBA	Evaluación basada en Cuestionarios
RHAT	Recursos Humanos y Ambiente de Trabajo
ROI	Retorno de la Inversión
SCAMPI	Método de Evaluación Estándar para CMMI para la Mejora del Proceso
SEI	Instituto de Ingeniería de Software
SWEBOK	Cuerpo de Conocimiento de la Ingeniería de Software
TI	Tecnologías de la Información
UDLA	Universidad de las Américas Puebla
WBS	Estructura Desglosada de Trabajo
XP	Programación Extrema

Anexo B. Plantillas definidas para las actividades de DERE

B.1. Plantilla para la Definición del Compromiso

Este documento describe el compromiso que se requiere establecer para iniciar el programa de Mejora del Proceso Software que será realizado en la Empresa [Inserte aquí el texto]. Así mismo, será utilizado para desarrollar un plan integrado de corto alcance para iniciar y gestionar la iniciativa de mejora. Este documento está compuesto por tres apartados: la Elaboración y Revisión del Compromiso de Mejora, la Descripción del Compromiso, y las Firmas de Conformidad.

1. Elaboración y Revisión del Compromiso de Mejora

1.1. Objetivo

El compromiso de mejora establece uno de los insumos más importantes para fortalecer la decisión de mejorar por parte de una empresa de desarrollo de software. El objetivo de este documento es comunicar a la alta dirección las actividades que serán realizadas y que constituyen en sí a la iniciativa de mejora.

1.2. Alcance

Este documento describe las actividades de la fase de definición del marco de trabajo DERE, el cual será utilizado para guiar, iniciar y gestionar la iniciativa de mejora de la Empresa [Inserte aquí el texto]. DERE proporciona a las empresas una descripción genérica de la secuencia de pasos recomendados para experimentar con iniciativas de Mejora del Proceso Software. Así pues, la fase inicial del modelo consiste en el establecimiento de un compromiso de trabajo, el alcance de éste se enfoca en dos niveles:

- Nivel organizativo: Se establece la infraestructura inicial de mejora, se definen las funciones y responsabilidades para esta infraestructura, se definen roles y participantes, y se asignan los recursos iniciales.
- Nivel de los objetivos del modelo objetivo de la certificación: Esta iniciativa de Mejora se enfoca en experimentar con el nivel [Inserte aquí el texto] del modelo de procesos [Inserte aquí el texto]. De acuerdo a este modelo, en el nivel [Inserte aquí el texto] el proceso implantado en la Empresa [Inserte aquí el texto] logrará su propósito y obtendrá los resultados definidos.

La descripción actual del proceso de software de la Empresa [Inserte aquí el texto] se actualizará a medida que se vayan implantando nuevas actividades e indicadores de progreso que se irán estableciendo con el tiempo en el desarrollo y despliegue del modelo en la organización.

1.3. Documentos de referencia

El proceso de la Empresa [Inserte aquí el texto] partirá de su definición actual y se esperan aportaciones documentadas o de prácticas identificadas que se realizan cotidianamente y que se relacionan con el proceso que se pretende mejorar. Adicionalmente, se hará uso de la siguiente documentación:

Código	Fecha	Título
DERE	[Inserte aquí el texto]	Marco de trabajo para establecer iniciativas de mejora del proceso de software con MoProSoft®: caso de estudio en una pequeña empresa oaxaqueña.
MoProSoft®	[Inserte aquí el texto]	Norma Mexicana NMX-I-059/01-04-NYCE-2005
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

1.3. Definiciones y acrónimos

Concepto	Definición
Proceso	Un conjunto organizado de actividades realizadas para transformar una entrada(s) en una salida(s) con un propósito dado.
Calidad	Totalidad de atributos y características de un proceso que permiten satisfacer las necesidades establecidas y las implícitas.
Mejora de Proceso de Software	El arte y la ciencia de cambiar el proceso de software de una organización para construir software de mejor calidad y de forma más productiva.
DERE	Marco de trabajo para promover la experimentación con iniciativas de mejora enfocadas a la implantación de MoProSoft®.
Compromiso de Mejora	Fase inicial de DERE en la cual se define el compromiso de los participantes y se delegan responsabilidades y tareas para realizar a lo largo del programa.
Evaluación	Examen y determinación del grado de cumplimiento de un proceso frente a normas, planes y procedimientos que le afecten.
[Inserte aquí el texto]	[Inserte aquí el texto]

Acrónimo	Concepto
AD	Alta Dirección
JP	Jefe de Proyecto
GT	Grupo de Trabajo
MPS	Mejora del Proceso de Software
MiPyMEs	Micro, Pequeña y Mediana Empresa de software
[Inserte aquí el texto]	[Inserte aquí el texto]

2. Descripción del Compromiso: Alta dirección, Gerencia, y Operación

2.1. Misión de la empresa

“*[Inserte aquí el texto]*”.

2.2. Visión de la empresa

“*[Inserte aquí el texto]*”.

2.3. Relación de la iniciativa de mejora con la misión y visión de la empresa

La iniciativa de mejora definida para la Empresa *[Inserte aquí el texto]* estará alineada con la misión y visión de la siguiente manera:

- Con respecto a la misión, el proceso de software que será definido para la Empresa *[Inserte aquí el texto]* estará soportado por la experiencia de diversas metodologías de desarrollo de software que permitirán establecer un proceso de acuerdo a las características de la organización para generar productos de calidad. *Nota: Es evidente que para fortalecer esta justificación de alineación de la mejora con los objetivos de negocio, será necesario agregar información relacionada con la misión particular de la empresa en cuestión.*
- En relación a la visión, el proceso software que será definido para la Empresa *[Inserte aquí el texto]* le permitirá incrementar los niveles de satisfacción de sus clientes alcanzando así una ventaja competitiva dentro del mercado en el cual ha estado participando. Esta competitividad permitirá que la Empresa *[Inserte aquí el texto]* mejore su posición actual. *Nota: Es evidente que para fortalecer esta justificación de alineación de la mejora con los objetivos de negocio, será necesario agregar información relacionada con la visión particular de la empresa en cuestión.*

2.4. Documentación de la necesidad de mejora

Es necesario identificar el origen de la necesidad por mejorar con el objetivo de establecer indicadores coherentes de mejora. Así pues, dicha necesidad se define por: *Nota: Deberán establecerse justificaciones claras para cada pregunta establecida en las viñetas posteriores.*

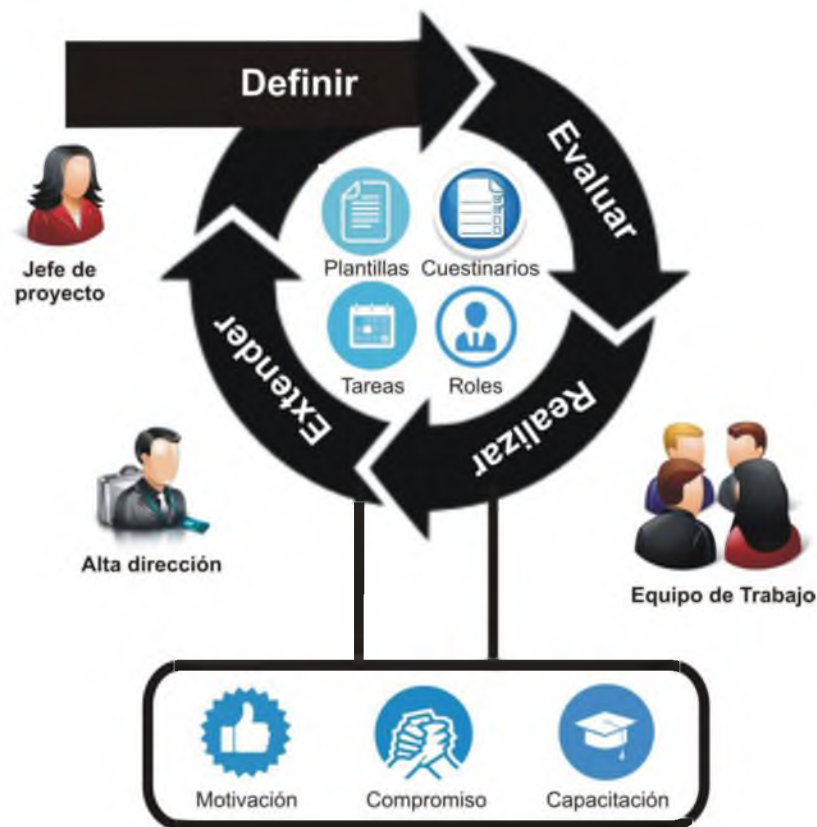
- ¿Quién identificó la necesidad de la mejora en la empresa? ¿Alta dirección, los mandos gerenciales, o el personal operativo? *[Inserte aquí el texto]*.
- ¿Qué problemas fueron detectados en el proceso actual que motivaron a la empresa a realizar una iniciativa de mejora? *[Inserte aquí el texto]*.
- Con el conocimiento de estos problemas, ¿qué espera la empresa de la iniciativa de mejora? *[Inserte aquí el texto]*.

2.5. Roles y responsabilidades

- Alta Dirección (AD): Se considera como AD a los directivos con cargos más altos en la empresa: el Presidente, el Gerente General y los Directores de las distintas áreas. En las empresas, la AD forma parte del Comité de Dirección o Comité Ejecutivo que se reúne con una frecuencia determinada y marcan la línea estratégica a seguir. Es regular que en empresas de tipo MiPyMEs la AD esté dividida en Dirección Administrativa y Dirección Técnica.
- Jefe de Proyecto (JP): Es el responsable de la gestión de los proyectos de software en la empresa y por lo tanto de la ejecución de las acciones correctivas sobre los mismos y de la cumplimentación y cierre de la documentación. Lidera al Grupo de Trabajo.
- Grupo de Trabajo (GT): Personal que debe llevar a cabo de manera concreta la implementación de las actividades propuestas por el proceso definido para la empresa, así como las acciones correctivas que el JP proponga.

2.6. Estrategia metodológica

El compromiso está inmerso en el ciclo continuo que propone el marco de trabajo DERE (específicamente en su fase Definir), el cual será utilizado para conducir la iniciativa de mejora de la Empresa [Inserte aquí el texto].



El marco de trabajo DERE es creado con la intención de conducir la mejora del proceso software en las empresas del Estado de Oaxaca y fue creado en la Universidad Tecnológica de la

Mixteca, con la colaboración de la tesista. Cabe mencionar que este modelo servirá únicamente para guiar iniciativas de mejora con MoProSoft® y será utilizado para definir el nuevo proceso software que será implantado en la Empresa [Inserte aquí el texto].

3. Definición del compromiso

3.1. A nivel general

Actividad	EPG	AD	JP	TWG
A1. Proporcionar seminario(s) para estimular la mejora en la AD	R	P		
A2. Proporcionar seminario(s) para estimular la mejora en el TWG (incluyendo al JP)	R		P	P
A3. Establecer la infraestructura de mejora	P	R	P	
A4. Evaluar y caracterizar el estado actual de la práctica (evaluación sobre el proceso software actual)	R		P	P
A5. Desarrollar recomendaciones y documentar los resultados	R			
A6. Presentar resultados preliminares	R	P	P	
A7. Establecer estrategias y prioridades	R		P	
A8. Generar planes de acción	R			
A9. Diseñar y establecer los activos del nuevo proceso	R		P	
A10. Planear y ejecutar proyectos piloto	R		P	P
A11. Capacitar al equipo piloto en los activos del nuevo proceso	R		P	
A12. Definir procesos y medidas para validar el nuevo proceso	R		P	
A13. Documentar y analizar las lecciones aprendidas	R			
A14. Presentar los resultados de la mejora a la alta dirección	R	P		
A15. Revisar el enfoque organizacional	R	P		
A16. Capacitar al resto de la organización en el uso del nuevo proceso	P		R	P
A17. Desplegar el proceso	P	R	P	

R: Responsable de la actividad – P: Participa en la actividad.

3.2. A nivel de infraestructura

Recurso	EPG	AD	JP	TWG
R1. Salón de reuniones con capacidad para todos los participantes de la iniciativa de mejora (actividades A1, A2, A6, y A14)		R		
R2. Equipo de cómputo suficiente para el desarrollo de los proyectos piloto (se recomienda un equipo por cada participante)		R		
R3. Servidor específico para establecer la base de conocimiento de la empresa		R		
R4. Proyector electrónico para la impartición de los seminarios y capacitación del personal (actividades A1, A2, A6, A11, A14, A16)		R		
R5. Pizarrón de acrílico para el trabajo en equipo durante la actividad A4		R		
R6. Adquisición de la documentación relacionada con MoProSoft® y para uso interno de la empresa		R		

R: Responsable de la actividad – P: Participa en la actividad.

3.3. A nivel de personal

La iniciativa de mejora requiere no solamente el compromiso de la AD sino también del TWG que es quien utilizará en sus tareas cotidianas el proceso de software definido para la Empresa [Inserte aquí el texto]. En este sentido, la iniciativa requiere lo siguiente:

- Se requiere la incorporación de uno de los JP de la Empresa [Inserte aquí el texto] al EPG con el objetivo de que éste se relacione estrechamente con el proceso que se está definiendo para la organización.
- Adicionalmente se requiere la integración del Equipo de Trabajo para desarrollar los planes de acción del proceso a mejorar. Además, éste ayudará a definir y desarrollar los nuevos procesos, estableciendo los estándares y procedimientos apropiados de acuerdo a las indicaciones del EPG, para lo cual se necesita la colaboración activa de, por lo menos, dos JP de la Empresa [Inserte aquí el texto], y dos miembros del TWG con la mayor antigüedad.
- Por último, todos los integrantes del TWG formarán el Equipo Piloto para realizar las actividades de prueba sobre el proceso definido para la Empresa [Inserte aquí el texto].

Además de estos requerimientos, es necesario que el TWG y los JP sepan que la iniciativa de mejora se sustenta en su compromiso para desempeñar las labores que el EPG le indique, dedicando tiempo extra a éstas, además de desempeñar sus tareas diarias. La AD debe asumir que posiblemente el personal se sienta presionado o saturado de trabajo y es compromiso de ésta resolver estos conflictos en pro de la mejora.

3.4. Procesos relacionados

Nota: Se debe incluir el listado de los procesos que serán abordados por la iniciativa de mejora. Estos procesos corresponden con el nivel de MoProSoft®, establecido en el punto 1.2 de este documento.

[Inserte aquí el texto]

3.5. Indicadores del proceso

Nota: Con base a la información proporcionada en el punto 2.4, la AD y los JP deben definir indicadores que les permitan comprobar que la iniciativa de mejora fue exitosa. Por ejemplo, si uno de los problemas que motivan la realización de la mejora es la desviación en el calendario - lo que provoca que los productos sean entregados tardíamente - el indicador seleccionado debe dar una idea de ¿cuánto quieren reducir ese retraso? (en porcentaje). Otro ejemplo puede ser que si los productos llevan un 15% de defectos insertados, ¿a cuánto lo quieren reducir? Estos indicadores son muy importantes dado que servirán al final para comprobar que se alcanzó tanto la mejora como el objetivo buscado.

[Inserte aquí el texto]

3.6 Periodo de ejecución

Es importante tener en cuenta que la duración en tiempo para completar un ciclo a través de DERE puede variar de organización en organización. Las organizaciones encuentran, dependiendo de los recursos destinados al programa de mejora, que muchas actividades pueden realizarse de

forma paralela. En este sentido, se plantea un tiempo no mayor a los [Inserte aquí el texto] meses para implantar la mejora en el proceso actual de la Empresa [Inserte aquí el texto].

4. Firmas de conformidad con el compromiso establecido

4.1. Por parte de la alta dirección

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

4.2. Por parte de la gerencia

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

4.3. Por parte del EPG (consultor externo o interno)

[Inserte aquí el texto]

B.2. Plantilla para el Plan de Evaluación

Este documento describe esencialmente la preparación y planificación de la fase de evaluación, además de identificar a los participantes, definir las actividades y tiempos para realizar el diagnóstico sobre los procesos de [Inserte aquí el texto] para determinar el grado de cumplimiento en la Empresa [Inserte aquí el texto]. La información recogida a través de este diagnóstico será soportada por el modelo del proceso obtenido a través del taller de motivación realizado anteriormente.

1. Introducción

1.1. Objetivo

Este documento servirá de guía en la identificación de fortalezas y debilidades de los procesos actuales de la Empresa [Inserte aquí el texto], en relación a la norma NMX-I-059/02-NYCE-2005.

1.2. Alcance

El alcance del (los) proceso(s) aquí descrito(s) comprende:

- A nivel directivo: La evaluación de una posible Dirección Técnica que resuelva las dudas relacionadas con la gestión directiva en el desarrollo de software.
- A nivel de gestión: La evaluación de los responsables de la gestión del proceso de desarrollo de software para determinar errores de administración de recursos, personal, y proceso.
- A nivel operativo: La evaluación del equipo de trabajo para obtener de primera mano información relevante sobre la implantación y documentación del proceso de desarrollo de software.

1.3. Documentos de referencia

La planificación de la evaluación se sustenta en los siguientes documentos:

Código	Fecha	Título
DefiniciónDeCompromiso	[Inserte aquí el texto]	Documento que define el compromiso de la alta dirección y los jefes de proyecto para con la iniciativa de mejora.
AcuerdoDeConfidencialidad	[Inserte aquí el texto]	Documento que establece un acuerdo de confidencialidad para asegurar los datos que serán recogidos en la empresa.
ModeladoDeProcesoActual	[Inserte aquí el texto]	Representación gráfica del proceso actual utilizado en la empresa para desarrollar y mantener software.

DefinicionDeProcesoActual	[Inserte aquí el texto]	Documento que define en términos de tareas, roles, productos de entrada y de salida el proceso actual utilizado en la empresa para desarrollar y mantener software
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

1.4. Definiciones y acrónimos

Concepto	Definición
Mejora del Proceso Software	El arte y la ciencia de cambiar el proceso software de una organización para construir software de mejor calidad y de forma más productiva.
Evaluación	Examen y determinación del grado de cumplimiento de un proceso frente a normas, planes y procedimientos que le afecten.
Proceso	Un conjunto organizado de actividades realizadas para transformar una o más entradas en una o más salidas que tienen un propósito dado.
Administración de Proyectos Específicos	Proceso cuyo propósito es establecer y realizar sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
Desarrollo y Mantenimiento de Software	Proceso cuyo propósito es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software (nuevos o modificados), cumpliendo con los requisitos especificados por el cliente.
Equipo de Trabajo	Recursos humanos asignados al proyecto.
Objetivos de negocio	Paso específico o medida que debe ser tomada para alcanzar la misión de la empresa.
Calidad	Totalidad de atributos y características de un proceso que le capacitan para satisfacer las necesidades establecidas y las implícitas.
Rol	Función que el personal técnico desarrolla en el área de acuerdo con su perfil y experiencia.
Entrada	Elemento que permite iniciar una actividad, es de carácter obligatorio.
Salida	Elemento que permite terminar una actividad. Algunas veces una salida se relaciona con un producto que será utilizado posteriormente en una actividad no secuenciada o a través de todo el ciclo de desarrollo.
Producto	Documento, componente, asignación de trabajo, o cualquier resultado generado por la actividad.
Criterios de medición	Conjunto de datos e información que se toman como referencia para evaluar una actividad en una valoración, auditoría, prueba, revisión y

	otros aspectos a comprobar de un proceso.
--	---

2. Plan de evaluación

2.1. Objetivos de negocio identificados

A través del análisis del Plan Estratégico, Misión, y Visión de la Empresa [Inserte aquí el texto], se vislumbran los siguientes objetivos de negocio:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

2.2. Metas de calidad establecidas en relación a la iniciativa de mejora

Las metas de calidad fueron establecidas durante la Elaboración y Revisión del Compromiso de Mejora y definen los resultados esperados por la iniciativa de mejora realizada en la Empresa [Inserte aquí el texto], en relación a la implantación de una nueva forma de trabajo. Las metas establecidas se resumen como:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

2.3. Unidad o área organizativa sujeta a evaluación

El proceso de evaluación se enfoca únicamente en el área responsable del desarrollo y mantenimiento de software dentro de la Empresa [Inserte aquí el texto]. Por la naturaleza de la norma NMX-I-059/02-NYCE-2005, este personal será dividido en tres categorías para su evaluación:

- Dirección técnica. En la evaluación participará [Inserte aquí el texto], como representante del equipo de trabajo quien fue seleccionado como representante de la Dirección Técnica considerando sus habilidades, experiencia, antigüedad y motivación para con la iniciativa de mejora.
- Gestores. En la evaluación participarán [Inserte aquí el texto], como representantes del equipo de trabajo quienes fueron seleccionados como jefes de proyectos considerando sus habilidades, experiencia, antigüedad y motivación para con la iniciativa de mejora.
- Personal operativo. El resto del personal será evaluado como personal operativo.

2.4. Capacitación del equipo de evaluación

El equipo de evaluación se forma considerando el conocimiento, las habilidades, experiencia, y antigüedad del personal en el desarrollo de sus labores como miembros del equipo de trabajo, y la motivación para realizar las labores que implican ser parte del equipo de evaluación. Ser parte del equipo de evaluación es un trabajo de tiempo completo durante los periodos de capacitación y ejecución de la evaluación. Este personal deberá estar libre del trabajo diario para desarrollar sus deberes como equipo de evaluación y no debe esperarse que asistan a reuniones, redacten documentos, o viajen, por ejemplo, durante el periodo que dure la evaluación. En este sentido, el equipo de evaluación estará formado por el siguiente personal:

- Personal externo a la Empresa [Inserte aquí el texto] (si lo hubiera):
 - Líder del equipo de evaluación: [Inserte aquí el texto].
- Personal de la Empresa [Inserte aquí el texto]:
 - A nivel de jefes de proyectos:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - A nivel de personal operativo:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].

2.5. Establecimiento de los procesos a evaluar

La realización de los talleres con el personal de la empresa permitió identificar que de los nueve procesos requeridos por la norma NMX-I-059/02-NYCE-2005 para alcanzar la certificación en el Nivel [Inserte aquí el texto], [Inserte aquí el texto] no se encuentran implementados actualmente. Por lo que la evaluación se enfocará en [Inserte aquí el texto] procesos que se encuentran medianamente establecidos en la empresa por el personal operativo y que se definen como:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

2.6. Diagrama del proceso actual de la Empresa [Inserte aquí el texto]

Nota: Deberá agregarse el modelado del proceso actual de la empresa. Cabe mencionar que este modelado fue obtenido a través del taller de colaboración como lo indican las actividades de la fase de evaluación.

2.7. Diagrama de la norma NMX-I-059/02-NYCE-2005 para el proceso [Inserte aquí el texto]

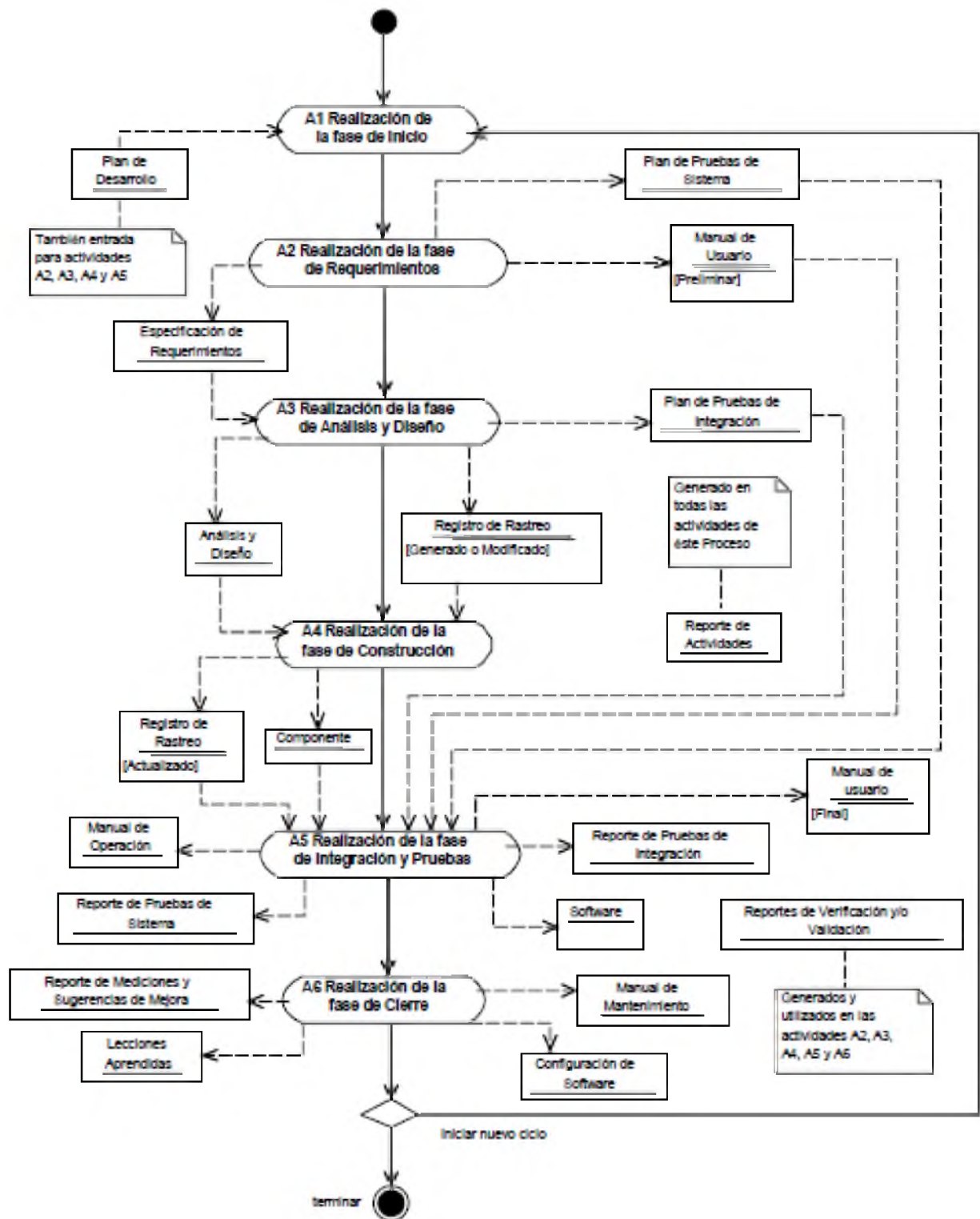


Figura 1. Ejemplo de modelo para DMS obtenido de la norma NMX-I-059/02-NYCE-2005

2.8. Comparación entre los procesos

2.8.1 A nivel de actividades

Proceso de la empresa [Inserte aquí el texto]	Proceso NMX-I-059/02-NYCE-2005
[Inserte aquí el texto]	A1. Fase de Inicio
[Inserte aquí el texto]	A2. Fase de Requisitos
[Inserte aquí el texto]	A3. Fase de Análisis y Diseño
[Inserte aquí el texto]	A4. Fase de Construcción
[Inserte aquí el texto]	A5. Fase de Integración y Pruebas
[Inserte aquí el texto]	
[Inserte aquí el texto]	
[Inserte aquí el texto]	A6. Fase de Cierre

2.8.2 A nivel de roles

Proceso de la empresa [Inserte aquí el texto]	Proceso NMX-I-059/02-NYCE-2005
[Inserte aquí el texto]	Cliente
[Inserte aquí el texto]	Analista, Equipo de Trabajo
[Inserte aquí el texto]	No aplica para el proceso
[Inserte aquí el texto]	Programador, Equipo de Trabajo
[Inserte aquí el texto]	Responsable de Administración de Proyectos Específicos, Responsable de Desarrollo y Mantenimiento
[Inserte aquí el texto]	Diseñador
[Inserte aquí el texto]	Diseñador de Interfaces
[Inserte aquí el texto]	Responsable de Pruebas
[Inserte aquí el texto]	Revisor
[Inserte aquí el texto]	Responsable de Manuales
[Inserte aquí el texto]	Usuario
[Inserte aquí el texto]	[Inserte aquí el texto]

2.8.3 A nivel de productos

Proceso de la empresa [Inserte aquí el texto]	Proceso NMX-I-059/02-NYCE-2005
[Inserte aquí el texto]	Plan de Desarrollo
[Inserte aquí el texto]	Especificación de Requisitos
[Inserte aquí el texto]	Plan de Pruebas de Sistema
[Inserte aquí el texto]	Manual de Usuario
[Inserte aquí el texto]	Análisis y Diseño
[Inserte aquí el texto]	Plan de Pruebas de Integración
[Inserte aquí el texto]	Reporte de Actividades
[Inserte aquí el texto]	Registro de Rastreo
[Inserte aquí el texto]	Componente
[Inserte aquí el texto]	Manual de Operación
[Inserte aquí el texto]	Reporte de Pruebas de Sistema
[Inserte aquí el texto]	Software
[Inserte aquí el texto]	Configuración de Software
[Inserte aquí el texto]	Manual de Mantenimiento
[Inserte aquí el texto]	[Inserte aquí el texto]

2.9. Metas de la evaluación

Las metas establecidas para esta evaluación son las siguientes:

- Identificar fortalezas y debilidades del proceso actual de [Inserte aquí el texto] en relación a la norma NMX-I-059/02-NYCE-2005.
- Cuantificar el nivel de cobertura actual de la Empresa de [Inserte aquí el texto] en relación a la norma NMX-I-059/02-NYCE-2005.
- Establecer una línea base para comparar el nivel de cobertura una vez que los procesos hayan sido mejorados.
- Establecer la base para desarrollar un plan de mejora.
- [Inserte aquí el texto].

2.10. Proyectos considerados en la evaluación

Los proyectos considerados por la Empresa [Inserte aquí el texto] para evaluar la adherencia al proceso establecido actualmente por la empresa para el desarrollo de software son los siguientes:

Actividad	JP	AD	EPG	EEVA	ET	LEEVA
A1. Capacitación al equipo de evaluación.						
A2. Aplicación de la evaluación.						
A3. Recogida y análisis de resultados.						
A4. Envío de resultados al líder del equipo de evaluación.						
A5. Realización de entrevistas y revisión de proyectos escogidos.						
A6. Preparación de informes.						
A7. Presentación de resultados a alta dirección y a equipo de trabajo.						
A8. Cierre de la evaluación.						

2.13. Infraestructura necesaria para la evaluación

Recurso	EPG	AD	LEEVA	EEVA
R1. Salón de reuniones con capacidad para todos que el EEVA sea capacitado (actividad A1)				
R2. Salón con suficiente espacio para aplicar la evaluación (actividad A2)				
R3. Servidor específico para establecer la base de conocimiento de la empresa (actividad A5)				
R4. Proyector electrónico para la impartición de la capacitación del personal y la presentación de resultados (actividades A1, A5, A7)				
R5. Pizarrón de acrílico para el trabajo en equipo (actividades A1, A5)				
R6. Adquisición de la Norma Mexicana NMX-I-059/NYCE-2005 para uso de la empresa				

3. Firmas de conformidad con el compromiso establecido

3.1. Por parte de la alta dirección

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

3.2. Por parte de la gerencia

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

3.3. Por parte del EPG (consultor externo o interno)

[Inserte aquí el texto]

B.3. Plantilla para entrega de los Resultados de la Evaluación

Este documento describe los hallazgos encontrados durante la evaluación de los procesos evaluados en la Empresa [Inserte aquí el texto] en el contexto de una iniciativa de Mejora del Proceso Software. Así mismo, estos resultados permitirán la priorización de soluciones que se enfocan a resolver los problemas más urgentes de la empresa. Este documento está compuesto por tres apartados: la Historia y Desarrollo de la Evaluación, los Resultados, y las Firmas de Conformidad.

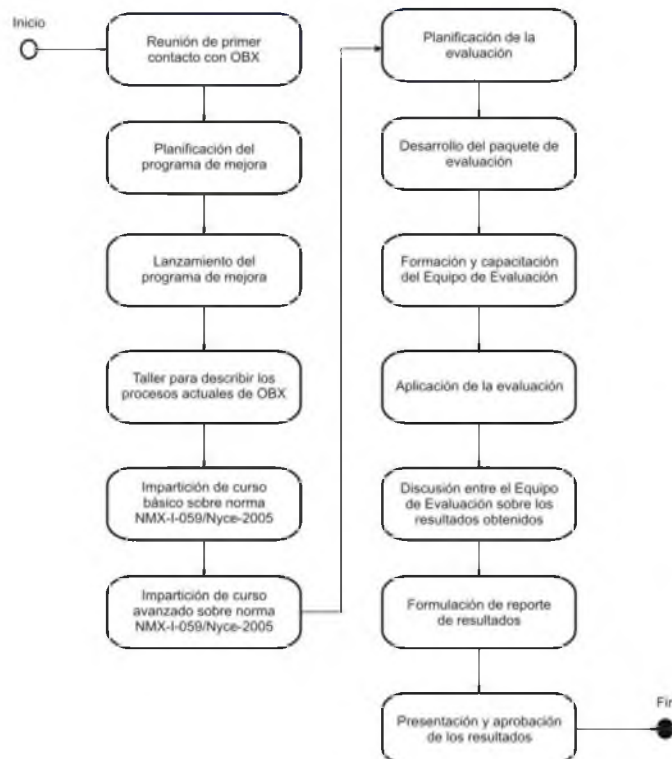
1. Historia y desarrollo de la evaluación

La necesidad de un esfuerzo de mejora fue identificada por la alta dirección y apoyada por los mandos intermedios (o de gestión). En este sentido, los acontecimientos que motivaron el lanzamiento de un programa de mejora se resumen como:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

1.1. Actividades de la evaluación

El siguiente diagrama resume las actividades realizadas para determinar las fortalezas y debilidades actuales de la Empresa [Inserte aquí el texto]:



Los tiempos en que cada actividad fue realizada se resumen en el siguiente calendario:

Actividad	Abril				Mayo				Junio				Julio		
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3
A1. Reunión de primer contacto con la Empresa [Inserte aquí el texto] .															
A2. Planificación del programa de mejora.															
A3. Lanzamiento del programa de mejora.															
A4. Taller para describir los procesos actuales de la Empresa [Inserte aquí el texto] .															
A5. Impartición de cursos básico sobre Norma NMX-I-059/NYCE-2005.															
A6. Impartición de cursos básico sobre Norma NMX-I-059/NYCE-2005.															
A7. Planificación de la evaluación.															
A8. Desarrollo del paquete de evaluación.															
A9. Formación y capacitación del Equipo de Evaluación.															
A10. Aplicación de la evaluación.															
A11. Discusión entre el Equipo de Evaluación sobre los resultados obtenidos.															
A12. Formulación de reporte de resultados.															
A13. Presentación y aprobación de los resultados.															

1.2. Equipo evaluador

El equipo de evaluación se formó considerando el conocimiento, las habilidades, experiencia, y antigüedad del personal en el desarrollo de sus labores como miembros del equipo de trabajo, y la motivación para realizar las labores que implican ser parte del equipo de evaluación. Ser parte del equipo de evaluación es un trabajo de tiempo completo durante los periodos de capacitación y ejecución de la evaluación. Este personal debió estar libre del trabajo diario para desarrollar sus deberes como equipo de evaluación. En este sentido, el equipo de evaluación estuvo formado por el siguiente personal:

- Personal externo a la Empresa [\[Inserte aquí el texto\]](#).
 - Líder del equipo de evaluación: [\[Inserte aquí el texto\]](#).
- Personal de la Empresa [\[Inserte aquí el texto\]](#).
 - A nivel de jefe de proyectos.
 - [\[Inserte aquí el texto\]](#).
 - [\[Inserte aquí el texto\]](#).

- [Inserte aquí el texto].
- A nivel de personal técnico/operativo.
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].

1.3. Capacitación impartida

El equipo de evaluación recibió capacitación relacionada con los siguientes tópicos:

- Modelo de evaluación EvalProSoft®.
- Mecanismo de evaluación basado en cuestionarios y entrevistas.
- Mecanismo de calificación a través de niveles de rendimiento por pregunta.
- Cálculo de cobertura total de procesos.
- [Inserte aquí el texto].
- [Inserte aquí el texto].

1.4. Alcance de la evaluación

La realización de los talleres con el personal de la Empresa [Inserte aquí el texto] permitió identificar los procesos que actualmente están implantados y que son requeridos por la norma NMX-I-059/NYCE-2005 para alcanzar la certificación en el Nivel [Inserte aquí el texto]. Por lo que la evaluación se enfocó en los siguientes procesos, dado que éstos se encuentran medianamente establecidos en la empresa por el personal técnico/operativo:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

2. Resultados

2.1. Hallazgos para el proceso de [Inserte aquí el texto]

La cobertura para la fase de [Inserte aquí el texto] del proceso de [Inserte aquí el texto] (barras de color azul) muestra que todas las actividades establecidas por la norma NMX-I-059/NYCE-2005 (denotadas de P1 a [Inserte aquí el texto]) están [Inserte aquí el texto] del valor mínimo requerido por el Órgano Certificador (línea amarilla que representa al 85% de cobertura).

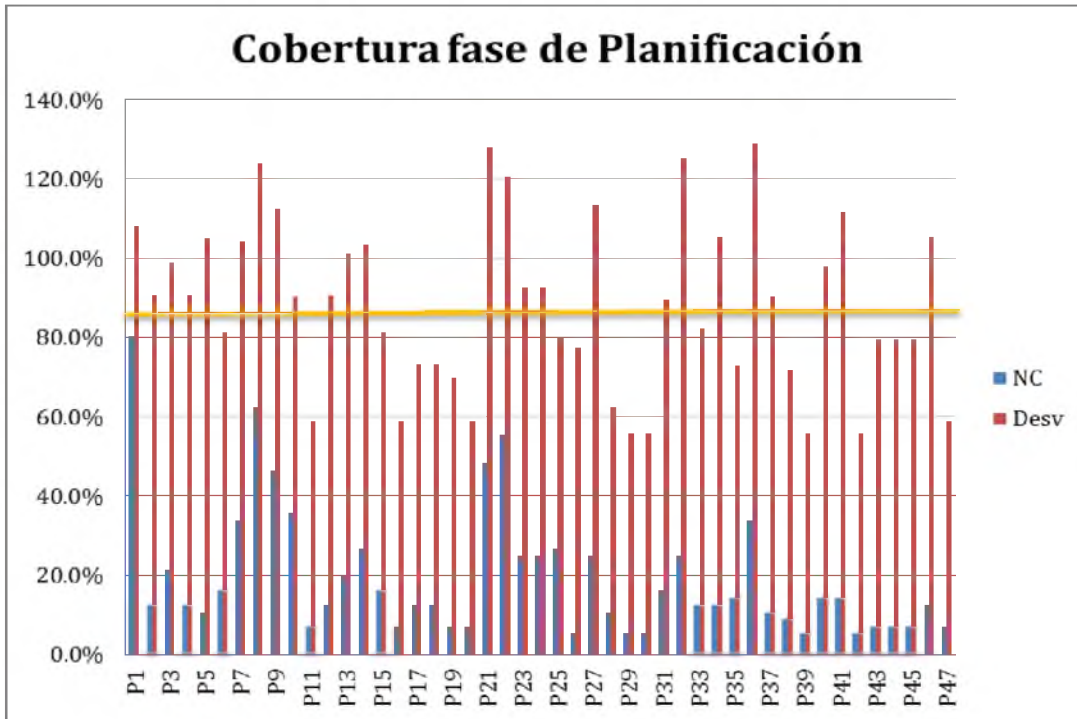


Figura 1. Ejemplo de gráfica de barras para cada proceso evaluado

Nota: Deberá agregarse una explicación con gráfica de barras por cada fase del proceso, y una sección por cada proceso evaluado.

2.2. Coberturas totales de los procesos [Inserte aquí el texto]

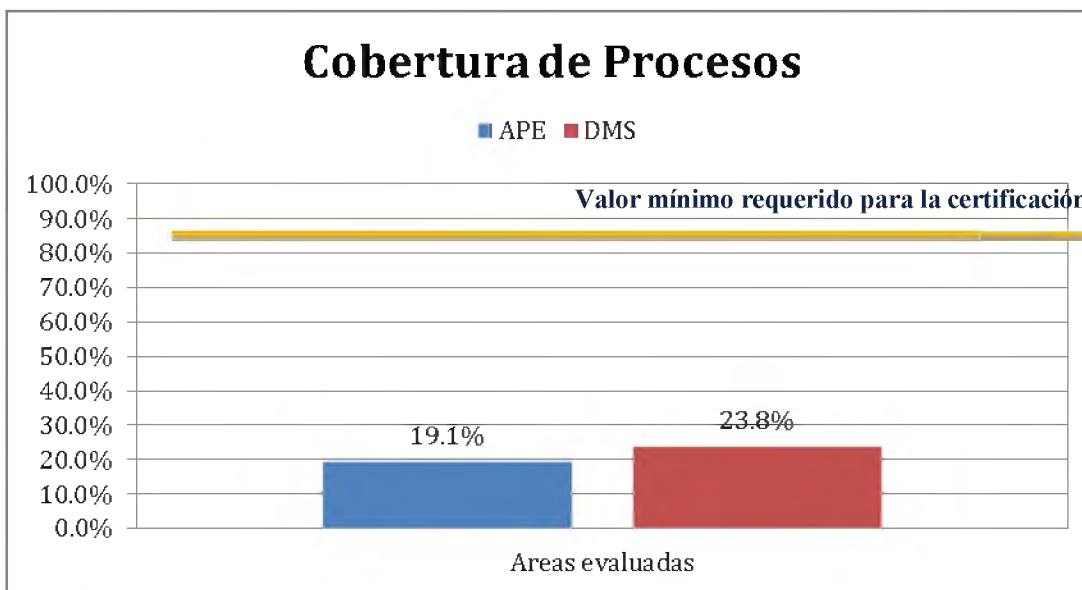


Figura 2. Ejemplo de gráfica de barras para mostrar las coberturas de los procesos evaluados

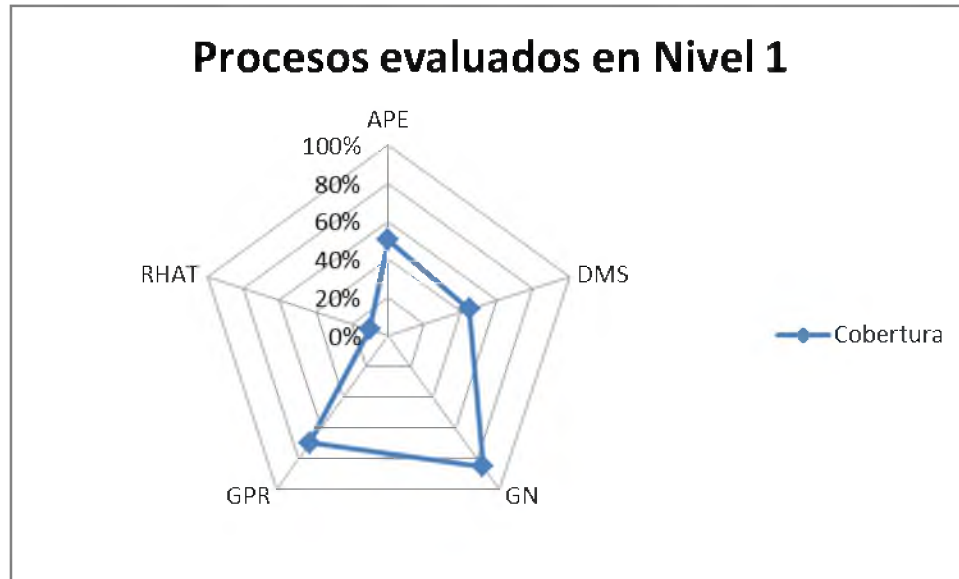


Figura 3. Ejemplo de gráfica de Kiviat para mostrar las coberturas de todos los procesos evaluados

Nota: La gráfica de Kiviat deberá llevar una justificación sobre la decisión de escoger, a lo más, tres procesos para mejorar durante una iniciativa.

2.3. Consecuencias sobre la decisión de no implantar la mejora

Con base al análisis de los resultados obtenidos durante la evaluación de los procesos actuales de la Empresa [Inserte aquí el texto], se recomienda realizar acciones inmediatas para subsanar todas las deficiencias encontradas, de lo contrario puede ocurrir lo siguiente:

- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].
- [Inserte aquí el texto].

3. Firmas de conformidad con los resultados presentados

3.1. Por parte de la alta dirección

[Inserte aquí el texto]

[Inserte aquí el texto]

3.2. Por parte de la gerencia

[Inserte aquí el texto]

[Inserte aquí el texto]

3.3. Por parte del personal operativo

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

3.4. Por parte del EPG (consultor externo o interno)

[Inserte aquí el texto]

B.4. Plantilla para la Guía del Plan de Acción

La guía del plan de acción proporciona a la alta dirección una visión de qué es necesario hacer para eliminar los problemas detectados en la fase “Evaluar”, quién tiene que estar involucrado en la creación de las soluciones y qué puede suponer para la Empresa [Inserte aquí el texto] la consecución de las mejoras propuestas. Este documento está compuesto por tres apartados: la Introducción, la Guía del Plan de Acción, y las Firmas de Conformidad.

1. Introducción

1.1. Objetivos y expectativas sobre la mejora para los procesos de [Inserte aquí el texto]

Esta guía será la base para que la alta dirección tome decisiones en cuanto al establecimiento de compromisos visibles para las siguientes etapas del ciclo de mejora, haciendo que la mejora resultante esté alineada con la visión corporativa y objetivos de negocio de la Empresa [Inserte aquí el texto].

1.2. Alcance

El alcance de la fase “Realizar” de los nuevos procesos de la Empresa [Inserte aquí el texto] comprende lo siguiente:

- A nivel directivo: La confirmación del compromiso establecido en la fase “Definir” por parte de la alta dirección, para apoyar la consecución del plan propuesto en este documento.
- A nivel de gestión: El entendimiento y aceptación de las soluciones diseñadas para eliminar los problemas detectados a nivel de gestión de los proyectos.
- A nivel operativo: La motivación para aprender a realizar eficientemente su trabajo diario y asumir la responsabilidad de una política de calidad total para todos los proyectos que realicen.

1.3. Documentos de referencia

La guía del plan de acción se sustenta en los siguientes documentos.

Código	Fecha	Título
DefinicionDeCompromiso	[Inserte aquí el texto]	Documento que define el compromiso de la alta dirección y los jefes de proyecto para con la iniciativa de mejora.
AcuerdoDeConfidencialidad	[Inserte aquí el texto]	Documento que establece un acuerdo de confidencialidad para asegurar los datos que serán recogidos en la Empresa [Inserte aquí el texto].
ModeloDelProcesoActual	[Inserte aquí el texto]	Modelo gráfico que representa el proceso actual utilizado en la Empresa [Inserte aquí el texto] para desarrollar y

		mantener software.
DefinicionDelProcesoActual	[Inserte aquí el texto]	Documento que define en términos de tareas, roles, productos de entrada y de salida el proceso actual utilizado en la Empresa [Inserte aquí el texto] para desarrollar y mantener software.
ResultadosDeEvaluacion	[Inserte aquí el texto]	Documento que resume los resultados obtenidos en la fase de evaluación.
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

1.4. Definiciones y acrónimos

Concepto	Definición
Mejora del Proceso Software	El arte y la ciencia de cambiar el proceso software de una organización para construir software de mejor calidad y de forma más productiva.
Evaluación	Examen y determinación del grado de cumplimiento de un proceso frente a normas, planes y procedimientos que le afecten.
Proceso	Un conjunto organizado de actividades realizadas para transformar una o más entradas en una o más salidas que tienen un propósito dado.
Administración de Proyectos Específicos	Proceso cuyo propósito es establecer y realizar sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
Desarrollo y Mantenimiento de Software	Proceso cuyo propósito es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software (nuevos o modificados), cumpliendo con los requisitos especificados por el cliente.
Equipo de Trabajo	Recursos humanos asignados al proyecto.
Equipo Piloto	Recursos humanos a realizar las pruebas sobre el/los proceso(s) definido(s).
Objetivos de negocio	Paso específico o medida que debe ser tomada para alcanzar la misión de la empresa.
Calidad	Totalidad de atributos y características de un proceso que le capacitan para satisfacer las necesidades establecidas y las implícitas.
Rol	Función que el personal técnico desarrolla en el área de acuerdo con su perfil y experiencia.

Acrónimo	Concepto
AD	Alta Dirección
RGPY	Responsable de la Gestión de Proyectos
RAPE	Responsable de Administración de Proyectos Específicos
RDM	Responsable de Desarrollo y Mantenimiento de Software
EPG	Grupo de Ingeniería de Software
EEVA	Equipo de Evaluación
ET	Equipo de Trabajo
EP	Equipo Piloto
LEEVA	Líder de Equipo de Evaluación
PEVA	Participantes en la Evaluación
CL	Cliente
APE	Administración de Proyectos Específicos
DMS	Desarrollo y Mantenimiento de Software

2. Plan de acción

2.1. Objetivos y expectativas sobre la mejora para el proceso de [\[Inserte aquí el texto\]](#)

Los objetivos establecidos para el proceso de [\[Inserte aquí el texto\]](#) son los siguientes:

- [\[Inserte aquí el texto\]](#).
- [\[Inserte aquí el texto\]](#).
- [\[Inserte aquí el texto\]](#).
- [\[Inserte aquí el texto\]](#).

Nota: Deberá agregarse una sección por cada proceso seleccionado para la mejora.

2.2. Tareas para alcanzar los objetivos y las expectativas de mejora

A continuación se resumen las tareas que han de realizarse para alcanzar los objetivos y cumplir las expectativas definidas en el punto anterior.

2.2.1 Infraestructura requerida

La infraestructura mínima necesaria para realizar las actividades de la guía del plan de acción se resume como:

Recurso	EPG	AD	ET	EP
R1. Salón de reuniones con capacidad para que el ET revise y discuta los procedimientos establecidos para el nuevo proceso. Se requiere de un proyector de tiempo completo.		R		
R2. Salón de reuniones con capacidad para que el EP sea capacitado en el uso del nuevo proceso. Se		R		

requiere de un proyector de tiempo completo.				
R3. Servidor específico para establecer la base de conocimiento de la Empresa [Inserte aquí el texto].		R		
R4. Proyector electrónico para la impartición de la capacitación del personal y la definición de procedimientos, plantillas, etc.		R		
R5. Adquisición de la Norma Mexicana NMX-I-059/NYCE-2005 para uso de la Empresa [Inserte aquí el texto].		R		

R: Responsable, P: Participa

2.2.2. Entrenamiento requerido

El Equipo de Trabajo (ET) se forma considerando las habilidades técnicas, la experiencia desarrollando el rol al que se le pretende asignar a cada individuo, la antigüedad en el desarrollo de sus labores diarias, y la motivación por aprender a realizarlas de manera eficiente. El equipo está compuesto por voluntarios de la Empresa [Inserte aquí el texto] (los que en última instancia, adoptan los procesos que resultan de la mejora) que están entusiasmados por trabajar y aprender sobre la mejora. Este grupo de personas fue identificado durante la fase de Diagnóstico. En este sentido, el ET fue reasignado de la siguiente manera:

- Personal externo a la Empresa [Inserte aquí el texto]:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
- Personal de la Empresa [Inserte aquí el texto].
 - Responsables:
 - RGPY: [Inserte aquí el texto].
 - RAPE: [Inserte aquí el texto].
 - RDM: [Inserte aquí el texto].
 - RAN: [Inserte aquí el texto].
 - RPR: [Inserte aquí el texto].
- Analistas:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
- Analistas/Diseñadores:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
- Programadores:
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
 - [Inserte aquí el texto].
- Revisor/documentador:
 - [Inserte aquí el texto].

roles del ET.																						
A3. Ajuste de procedimientos, plantillas y guías de implantación.																						
A4. Preparación y ejecución de pruebas piloto por roles.																						
A5. Recogida y análisis de pruebas piloto.																						
A6. Planificación de proyectos piloto.																						
A7. Cierre de la fase de establecimiento de nuevos procesos.																						

Es posible que el tiempo planificado sea reducido puesto que el ET y el EP tiene los mismos integrantes y puede que algunas actividades sean realizadas de forma paralela.

2.3.4. Diseño de los nuevos procesos

Es posible que el tiempo planificado sea reducido puesto que el ET y el EP tiene los mismos integrantes y puede ser que algunas actividades sean realizadas de forma paralela.

2.3.4.1. Modelado del proceso de [Inserte aquí el texto] para la Empresa [Inserte aquí el texto]

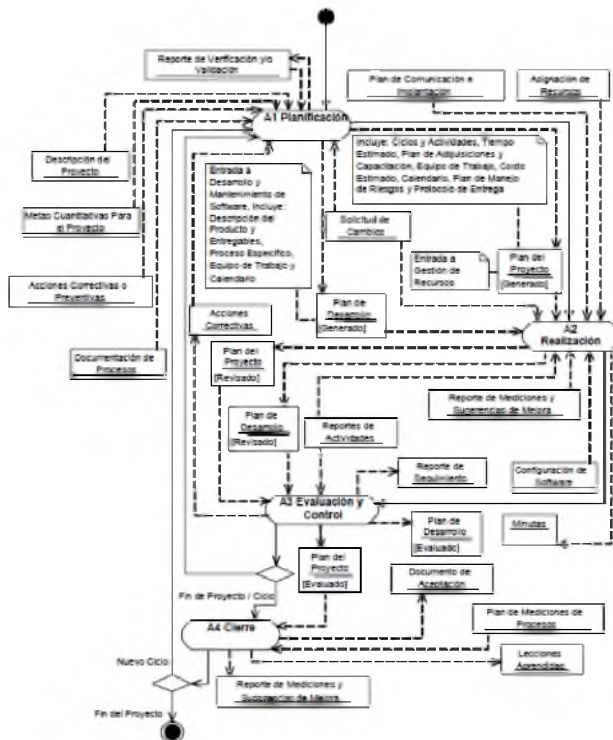


Figura 1. Ejemplo de modelado del proceso obtenido a través del taller de trabajo

Nota: Deberá agregarse un modelo por cada proceso seleccionado para la mejora. Este proceso, o procesos, será posteriormente definido en términos de actividades, roles, responsabilidades, entradas, salidas, y criterios de medición para que la empresa cuente con una descripción documentada de sus procesos.

2.3.5. Selección de proyectos piloto

Los procesos definidos anteriormente deberán ser probados en los proyectos piloto que la Empresa [Inserte aquí el texto] considere necesarios para experimentar con las nuevas descripciones entregadas. Los proyectos piloto escogidos son los siguientes:

Referencia del proyecto	Descripción	Fecha de inicio	Fecha de fin	Motivo de cancelación
<i>Referencia interna que se le dio al proyecto</i>	<i>Descripción del proyecto. Además se debe mencionar la cantidad de personal que trabajó en él, los lenguajes utilizados, y el tiempo y costo estimados</i>	<i>Cuándo inició</i>	<i>Cuándo se canceló o terminó</i>	<i>Por qué se canceló, si fuera el caso</i>
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

2.3.6. Involucración y nivel de esfuerzo requerido

Considerando las actividades definidas en el calendario de trabajo, se requiere del siguiente nivel de participación (indicado en porcentaje):

Actividad	AD	EPG	ET	EP	EEVA
A1. Definición de procesos, procedimientos y guías de implantación.					
A2. Capacitación por roles del ET.					
A3. Ajuste de procedimientos, plantillas y guías de implantación.					
A4. Preparación y ejecución de pruebas piloto por roles.					
A5. Recogida y análisis de pruebas piloto.					
A6. Planificación de proyectos piloto.					
A7. Cierre de la fase de establecimiento de nuevos procesos.					

2.3.7. Dependencias, riesgos y limitaciones

El éxito en el establecimiento y capacitación de los nuevos procesos de la Empresa [Inserte aquí el texto] se puede ver afectado por las siguientes dependencias y riesgos:

- [Inserte aquí el texto].

- [Inserte aquí el texto].
- [Inserte aquí el texto].

Por último, las limitaciones para el establecimiento del (los) nuevo(s) proceso(s) de la Empresa [Inserte aquí el texto] se resumen como:

- [Inserte aquí el texto].
- [Inserte aquí el texto].

3. Firmas de conformidad con la Guía del Plan de Acción

3.1. Por parte de la alta dirección

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

[Inserte aquí el texto]

3.2. Por parte de la gerencia

[Inserte aquí el texto]

[Inserte aquí el texto]

3.3. Por parte del EPG (consultor externo o interno)

[Inserte aquí el texto]

Anexo C. Cuestionarios de Evaluación

C.1. Evaluación del Proceso de Administración de Proyectos Específicos

PLANIFICACIÓN (A.1)

I. La planificación se refiere al conjunto de actividades cuya finalidad es obtener y mantener el *Plan del Proyecto* y el *Plan de Desarrollo* que regirán al proyecto específico, con base en la *Descripción del Proyecto*.

1. ¿Existe en la organización un *Responsable de la Gestión de Proyectos*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica
Dependiendo del tipo de producto, la persona que lidera el proyecto puede cambiar de un proyecto a otro.

Comentarios:

2. ¿Existe en la organización un documento que defina los objetivos y el alcance de un proyecto? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica
Este documento, la Descripción del Proyecto, debe contener la Descripción del propósito y del producto, el Alcance del Proyecto, los Objetivos del mismo, los Entregables esperados, los Supuestos y premisas, la Necesidad de negocio, entre otros.

Comentarios:

3. ¿El Responsable de la Gestión de Proyectos revisa la *Descripción del Proyecto*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica
La Descripción del Proyecto es revisada con la finalidad de que pueda definirse una estrategia que cubra los objetivos y entregables establecidos para el proyecto.

Comentarios:

4. ¿Existe un *Protocolo de Entrega* para cada proyecto aceptado por la organización? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica
El Protocolo de Entrega debe establecer los términos en que cada entregable del proyecto será recibido en tiempo y forma por el Cliente.

Comentarios:

5. ¿Este *Protocolo de Entrega* es definido por el *Responsable de Administración de Proyectos Específicos* y por el *Cliente*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica
El Protocolo de Entrega debe ser establecido y aceptado por todos los involucrados en esta actividad del proyecto.

Comentarios:

6. ¿Existe en la organización un proceso que identifique los *Ciclos y Actividades* específicas a realizar para producir los *Entregables* de un proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Este proceso resume las actividades a realizar para cumplir con el Protocolo de Entrega. La determinación de las fases del proyecto incluye, en la mayoría de los casos, la selección y refinamiento de un modelo de desarrollo de software para establecer las interdependencias y la secuencia apropiada de las actividades del proyecto.

Comentarios:

7. ¿Se consensa cada una de las actividades con aquellos que las llevarán a cabo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Para esta respuesta debe reflexionarse ¿El personal es considerado para seleccionar aquellas actividades que es capaz de realizar?

Comentarios:

8. ¿Se identifica a un responsable para cada una de las actividades que serán realizadas?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cada tarea a realizar debe tener un responsable que asegure que fue realizada correctamente y que se comprometa a corregirla si existieran problemas.

Comentarios:

9. ¿Se identifican la relación y dependencia de todas las actividades a realizar?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comúnmente, las actividades de un proyecto se pueden lograr en una cierta secuencia que minimice la duración del mismo. Esto implica identificar las tareas predecesoras y sucesoras para determinar el orden óptimo.

Comentarios:

10. ¿Se estima el tiempo de desarrollo para cada actividad identificada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cuando los calendarios se desarrollan inicialmente, es común hacer suposiciones sobre la duración de ciertas actividades. Estas suposiciones se hacen con frecuencia con elementos para los cuales existen pocos, o no existen, datos de estimación. Identificar estas suposiciones proporciona entendimiento del nivel de confianza (incertidumbre) en el calendario general.

Comentarios:

11. ¿Existe información histórica disponible en la organización para estimar estos tiempos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La información histórica incluye costo, esfuerzo, y tiempos de proyectos previamente realizados, además de datos de escala apropiados para ser tomados en

Comentarios:

18. ¿Existe en la organización un *Plan de Capacitación para el uso de Herramientas* en caso de que el proyecto lo requiera?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

19. ¿Se utiliza toda la información de los planes anteriores para elaborar el *Plan de Adquisiciones y Capacitación* de la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan de Adquisiciones y Capacitación concentra las solicitudes de adquisición de recursos. Regularmente se incluye necesidades de capacitación al personal, requisitos a proveedores, necesidades de infraestructura y herramientas, así como requerimientos de capacitación en general.

Comentarios:

20. ¿Se cuenta con fichas de personal que permitan identificar roles y habilidades para formar un *Equipo de Trabajo* por cada proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Las fichas de personal deberán concentrar toda la información relacionada con las habilidades, certificaciones, educación, y entrenamiento de cada miembro del Equipo de Trabajo para la asignación correcta de roles.

Comentarios:

21. ¿Existen roles definidos por la organización que establezcan tareas y responsabilidades dentro del *Equipo de Trabajo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

22. ¿Se establecen responsables dentro del *Equipo de Trabajo* para cada actividad establecida en la *Descripción del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

23. ¿Se establece un mecanismo que asegure que las actividades de la *Descripción del Proyecto* sean coordinadas y realizadas?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los hitos son establecidos a menudo para asegurar la terminación de ciertos entregables en el proyecto. Los hitos pueden estar basados en eventos o en el calendario.

Comentarios:

24. ¿Se planifican las fechas de inicio y fin por cada una de estas actividades para generar el *Calendario* de trabajo considerando tiempos estimados?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Establecer y mantener el calendario del proyecto incluye comúnmente lo siguiente: Determinar la disposición por fase de las actividades; Definir las dependencias entre actividades; Definir las actividades e hitos del calendario para apoyar la exactitud en la medición del progreso; Identificar los hitos para la entrega de productos al cliente; Definir actividades de duración apropiada; Usar datos históricos apropiados para verificar el calendario.

Comentarios:

25. ¿Se planifican las fechas de inicio y fin del *Calendario* considerando la disponibilidad comprometida o esperada de recursos e instalaciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

26. ¿La planificación de las actividades se realiza a través de una herramienta automatizada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Algunos ejemplos de estas herramientas incluyen a las siguientes: Microsoft Project, Primavera Enterprise, Deltek Open Plan, etc.

Comentarios:

27. ¿Se estima el *Costo* de cada proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Establecer y mantener el costo del proyecto incluye comúnmente lo siguiente: Competencias y roles críticos necesarios para realizar el trabajo; Costo de productos adquiridos externamente; Valoraciones de costos del ciclo de vida; Capacidad de las herramientas proporcionadas en el entorno de la ingeniería; Niveles de habilidad de los administradores y el personal necesario para realizar el trabajo; Conocimientos, habilidades, y necesidades de entrenamiento; Facilidades necesarias (i.e., oficinas, espacios y sitios de trabajo); Viajes; Acuerdos del nivel de servicios de garantía y call-centers; Gastos directos e indirectos.

Comentarios:

28. ¿Se identifican, describen y evalúan los riesgos que pueden afectar a cada proyecto de la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La identificación de riesgos incluye identificar situaciones, peligros, amenazas, vulnerabilidades, etc., potenciales que puedan afectar negativamente los esfuerzos y planes de trabajo. Los riesgos deben ser identificados y descritos de manera comprensible antes de que puedan ser evaluados. Al identificar los riesgos, es bueno utilizar un método estándar para su definición. Las herramientas de identificación y evaluación de riesgos se pueden utilizar para ayudar a identificar posibles problemas.

Comentarios:

29. ¿Se analiza el impacto y la probabilidad de ocurrencia de cada riesgo identificado?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cada riesgo identificado debe ser documentado claramente en el Plan de Manejo de Riesgos, incluyendo las consecuencias posibles y su impacto en consecuencia, su probabilidad de ocurrencia y las acciones tanto preventivas como correctivas que se deben llevar a cabo en cada situación.

Comentarios:

30. ¿Son priorizados los efectos de cada riesgo identificado sobre los <i>Objetivos</i> del proyecto?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

31. ¿Se documenta una <i>Solicitud de Cambios</i> sobre los requisitos del proyecto?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La Solicitud de Cambios se relaciona con el proceso de control de cambios mediante el cual se propone, se evalúa, se aprueba o se rechaza, se programa y se le da seguimiento a la modificación del software. El fundamento básico es un proceso de control de cambios, un estado de los componentes, presentación de informes, y un proceso de evaluación. El control de cambios en el software es un proceso de decisión utilizado en el control de los cambios realizados al mismo. Algunos cambios propuestos son aceptados e implementados durante este proceso, otros son rechazadas o pospuestos, y no se aplican.

Comentarios:

32. ¿Se determina el origen de este cambio?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

33. ¿Existe en la organización un responsable de gestionar el cambio del requisito desde su inicio hasta su cierre?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

34. ¿Se actualiza el *Plan del Proyecto* con cada petición de cambio?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan del Proyecto es un documento formal usado como guía para la ejecución y control del proyecto. En este punto, este documento está formado por Ciclos y Actividades, Tiempo Estimado, Plan de Adquisiciones y Capacitación, Equipo de Trabajo, Costo Estimado, Calendario, Plan de Manejo de Riesgos, y Protocolo de Entrega.

Comentarios:

35. ¿Se definen hitos para la actualización del *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

36. ¿Se establece el control de versiones ante la gestión de los cambios requeridos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados Sistemas de Control de Versiones o SVC (System Version Control).

Comentarios:

37. ¿Existe en la organización un *Plan de Desarrollo* independiente del *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan de Desarrollo define una estrategia de alto nivel para el desarrollo de los productos requeridos por el Cliente que contiene: Descripción del Producto y Entregables, Proceso Específico, Equipo de Trabajo, Calendario.

Comentarios:

38. ¿Se verifica el *Plan del Proyecto* cada que es actualizado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La verificación es la actividad que confirma que el Plan del Proyecto o el Plan de Desarrollo reflejan propiamente los requerimientos especificados para él.

Comentarios:

39. ¿Se verifica el *Plan de Desarrollo* cada que es actualizado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan de Desarrollo define una estrategia de alto nivel para el desarrollo de los

productos requeridos por el Cliente que contiene: Descripción del Producto y Entregables, Proceso Especifico, Equipo de Trabajo, Calendario.

Comentarios:

40. ¿Son corregidos los errores/defectos encontrados en la verificación del Plan del Proyecto y el Plan de Desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

41. ¿Se comunican al resto del Equipo de Trabajo los errores/defectos corregidos en la verificación del Plan del Proyecto y el Plan de Desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

42. ¿Se documentan las correcciones realizadas al Plan del Proyecto y al Plan de Desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

43. ¿Se valida el Plan del Proyecto cada que es actualizado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La validación es la actividad que confirma que el Plan del Proyecto o el Plan de Desarrollo resultante es capaz de satisfacer los requerimientos para su aplicación especificada o uso previsto.

Comentarios:

44. ¿Se valida el Plan de Desarrollo cada que es actualizado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

45. ¿Son corregidos los errores/defectos encontrados en la validación del Plan del Proyecto y el Plan de Desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

apropiado?

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Comentarios:

5. Una vez que la *Asignación de Recursos* es aceptada o rechazada, ¿se documenta la distribución de los recursos a los miembros del *Equipo de Trabajo* para que éstos realicen las actividades?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La definición de los recursos (habilidades/conocimiento, maquinaria/equipo, materiales, métodos, herramientas) y sus cantidades necesarias para realizar las actividades del proyecto se basa en la estimación inicial y proporciona información adicional que se puede aplicar para extender los Ciclos y Actividades usados en la administración del proyecto.

Comentarios:

6. ¿Se planifican, revisan y auditan las actividades de los subcontratistas, asegurando la calidad de los productos o servicios contratados y el cumplimiento con los estándares y especificaciones acordadas?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

7. ¿Se generan *Reportes de Actividades*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Un Reporte de Actividades es un registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como: Tiempo de producción, de Corrección, de Verificación y de Validación, Defectos encontrados en verificación, Defectos encontrados en validación, Defectos encontrados en pruebas, Tamaño de Productos.

Comentarios:

8. ¿Se registran todos los costos reales del ciclo/proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El término "real" hace alusión al costo, recurso, esfuerzo, etc., que en la actualidad se ha invertido en el ciclo/proyecto.

Comentarios:

9. ¿Se registran todos recursos reales del ciclo/proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

Comentarios:

16. ¿Se realizan reuniones de revisión con el *Equipo de Trabajo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

17. ¿Se documentan en *Minutas* los puntos tratados y los acuerdos tomados entre el *Equipo de Trabajo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Una Minuta es un resumen escrito de las reuniones, la cual describe quiénes asistieron, qué asuntos se trataron y los acuerdos o compromisos a que se llegó, resaltando quién y cuándo tendrá lista alguna solución.

Comentarios:

18. ¿Se realizan reuniones de revisión con el *Cliente*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

19. ¿Se documentan en *Minutas* los puntos tratados y los acuerdos tomados con el *Cliente*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

EVALUACIÓN Y CONTROL (A.3)

III. La evaluación y el control se aseguran que los *Objetivos* del proyecto sean cumplidos a través de la supervisión y evaluación del progreso del mismo. En caso de que sean detectadas desviaciones en el *Calendario original*, se establecen *Acciones Correctivas* según sea apropiado. Como resultado de estas actividades el *Plan del Proyecto* y el *Plan de Desarrollo* serán actualizados.

1. ¿Se realiza una evaluación sobre el avance del *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los parámetros de la planificación del proyecto constituyen indicadores típicos del progreso y rendimiento del mismo e incluyen atributos de productos de trabajo tales como actividades, costos, esfuerzo y tiempo. El seguimiento normalmente implica

medir los valores reales de los parámetros de la planificación del proyecto, comparar estos valores con los estimados en el Plan del Proyecto e identificar las desviaciones significativas.

Comentarios:

2. ¿Se evalúa el cumplimiento actual del alcance del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Esta evaluación mide periódicamente la completitud real de las actividades e hitos, y se compara contra el Plan del Proyecto.

Comentarios:

3. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El documentar los valores reales de los parámetros de la planificación del proyecto incluye el registrar la información contextual asociada para ayudar a comprender la comparación e identificar desviaciones importantes que deben ser tratadas.

Comentarios:

4. ¿Se evalúa el cumplimiento actual del Costo estimado en el Plan del Proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La evaluación del costo empleado en el proyecto normalmente incluye: Comparar periódicamente los costos reales incurridos, Comparar los costos reales con las estimaciones y Costo estimado en el Plan del Proyecto, Identificar las desviaciones significativas del Costo estimado en el Plan del Proyecto.

Comentarios:

5. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La documentación incluye a las desviaciones significativas encontradas.

Comentarios:

6. ¿Se evalúa el cumplimiento actual del Calendario establecido en el Plan del Proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La evaluación del progreso incluye: Comparar el grado de avance real de las actividades e hitos frente al Calendario del Plan del Proyecto, Identificar las desviaciones significativas de las estimaciones del Calendario del Plan del Proyecto.

Comentarios:

7. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La documentación incluye a las desviaciones significativas encontradas.

Comentarios:

8. ¿Se evalúa el cumplimiento actual del *Equipo de Trabajo* establecido en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La evaluación del esfuerzo realizado por el Equipo de Trabajo normalmente incluye: Comparar periódicamente el esfuerzo del personal asignado al Equipo de Trabajo, Comparar el esfuerzo con las estimaciones del Plan del Proyecto, Identificar las desviaciones significativas del esfuerzo estimado para el Equipo de Trabajo en el Plan del Proyecto.

Comentarios:

9. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La documentación incluye a las desviaciones significativas encontradas.

Comentarios:

10. ¿Se evalúa el cumplimiento actual sobre los *Ciclos y Actividades* establecidos en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los directivos, el Equipo de Trabajo, los Clientes, los Usuarios finales, los subcontratistas y otras partes interesadas se incluyen en la evaluación según sea apropiado. La evaluación del progreso sobre los Ciclos y Actividades se realizan con regularidad (p.ej., semanal, mensual o trimestralmente).

Comentarios:

11. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La documentación incluye a las desviaciones significativas encontradas.

Comentarios:

12. ¿Se establecen *Acciones Correctivas* para resolver los problemas encontrados en las evaluaciones anteriores sobre el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Las Acciones Correctivas son establecidas y gestionadas hasta su cierre para corregir una desviación o problema con respecto al cumplimiento del Plan del Proyecto y el Plan de Desarrollo. Algunos ejemplos de Acciones Correctivas son:

Modificar el Plan de Desarrollo, Modificar los requisitos, Modificar las estimaciones y los planes, Renegociar los compromisos, Añadir recursos, Cambiar los procesos, Modificar los riesgos del proyecto.

Comentarios:

13. ¿Se realiza el seguimiento del *Plan de Manejo de Riesgos* para identificar, controlar o actualizar los riesgos durante el desarrollo del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Este plan contiene la identificación y evaluación de riesgos, así como los planes de contención y de contingencia correspondientes. A medida que avanzan los proyectos (especialmente aquellos de larga duración), surgen nuevos riesgos por lo que es importante identificarlos y evaluarlos. Por ejemplo, el software, el equipo y las herramientas en uso pueden volverse obsoletos; o los miembros del Equipo de Trabajo pueden perder gradualmente las habilidades en áreas de particular importancia a largo plazo para el proyecto y la organización.

Comentarios:

14. ¿Se genera el *Reporte de Seguimiento* del proyecto considerando los *Reportes de Actividades* generados?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Seguimiento contiene el registro del avance de las actividades realizadas incluyendo las realizadas en el Plan de Manejo de Riesgos. El avance se registra por ciclo, incluyendo fecha de inicio y fin. Además, este reporte contiene el registro periódico de las mediciones como: Costo real del proyecto, Esfuerzo realizado, Cambios implementados y clasificados por tipo, Tiempo real invertido, Defectos encontrados, Tamaño de los productos y Trabajo duplicado.

Comentarios:

CIERRE (A.4)

IV. El cierre consiste en entregar los productos de acuerdo al *Protocolo de Entrega* y dar por concluido el ciclo o proyecto. Como resultado final se genera el *Documento de Aceptación* del *Cliente* que da por terminado el desarrollo del proyecto.

1. ¿Se establece y se cumple con un *Protocolo de Entrega* en cada proyecto realizado por la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Protocolo de Entrega resume los términos establecidos por la organización y el Cliente para entregar y recibir, respectivamente, el producto terminado.

Comentarios:

2. ¿Se documenta el seguimiento y evaluación de dicho *Protocolo de Entrega*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Esta evaluación mide la completitud real de los elementos establecidos en el

Protocolo de Entrega, y se compara contra lo establecido en el Plan del Proyecto.

Comentarios:

3. Una vez terminado el proyecto, ¿se efectúa el cierre de las actividades subcontratadas con terceros?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Es importante asegurarse de tratar todas las cuestiones relativas a la propiedad intelectual del producto (o componente) adquirido antes de que el proyecto lo acepte.

Comentarios:

4. ¿Se obtiene el *Documento de Aceptación del Cliente* para dar por finalizado el proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Este documento establece la aceptación del Cliente de los entregables establecidos en el proyecto y da por terminada la responsabilidad de la organización en todo lo relacionado con el desarrollo del proyecto.

Comentarios:

C.2. Evaluación del Proceso de Desarrollo y Mantenimiento de Software

REALIZACIÓN DE LA FASE DE INICIO (A.1)

I. Durante la fase de inicio los miembros del *Equipo de Trabajo* revisan el *Plan de Desarrollo* para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.

1. ¿Se revisa con los miembros del *Equipo de Trabajo* el *Plan de Desarrollo* actual para entender el proyecto y comprometerse con su desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan de Desarrollo, generado por el proceso de Administración de Proyectos Específicos, puede omitirse en el caso de que los roles de Responsable de Administración de Proyectos Específicos y de Responsable de Desarrollo y Mantenimiento sean desempeñados por una misma persona. Si éste fuera el caso, el documento que registrará el proceso de Desarrollo y Mantenimiento de Software será el Plan del Proyecto.

Comentarios:

2. ¿Se elabora un Reporte de Actividades registrando las actividades realizadas en la revisión anterior, las fechas de inicio y fin, y el responsable por cada actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Un Reporte de Actividades es un registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como: Tiempo de producción, de Corrección, de Verificación y de Validación, Defectos encontrados en verificación, Defectos encontrados en validación, Defectos encontrados en pruebas, Tamaño de Productos.

Comentarios:

REALIZACIÓN DE LA FASE DE REQUISITOS (A.2)

II. Durante la fase de requisitos se desarrolla el conjunto de actividades cuya finalidad es obtener la documentación de la *Especificación de Requisitos* y el *Plan de Pruebas de Sistema* para conseguir un entendimiento común entre el *Cliente* y el proyecto.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los analistas de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los analistas poseen el conocimiento y experiencia para la obtención, especificación y análisis de los requisitos.

Comentarios:

2. ¿Estos analistas identifican y consultan fuentes de información (clientes, usuarios, sistemas anteriores, documentos existentes, etc.) para obtener los nuevos requisitos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

3. ¿Son analizados los requisitos identificados en la actividad anterior para delimitar su alcance y factibilidad, considerando las restricciones del ambiente de negocio del *Cliente* o del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Aquellos requisitos que no sean factibles de desarrollo serán descartados y no formarán parte de la Especificación de Requisitos.

Comentarios:

4. ¿Se elabora o modifica un prototipo de interfaz de usuario que permita facilitar el entendimiento sobre el proyecto entre el *Cliente* y la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Esta interfaz de usuario contiene elementos de diseño visual que le proporcionan al usuario una idea de las interfaces que mostrará el sistema, con el fin de obtener realimentación sobre los requisitos del mismo. Es posible generar unas cuantas imágenes de pantalla o bien un esqueleto ejecutable de interfaces de usuario.

Comentarios:

5. ¿Los analistas generan o actualizan la *Especificación de Requisitos*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La Especificación de Requisitos regularmente se compone de una introducción y una descripción de requisitos. Los requisitos son descritos como: Funcionales, de Interfaz de usuario, de Confiabilidad, de Eficiencia, de Mantenimiento, de Portabilidad, de Restricciones de diseño y construcción, Legales y reglamentarios, etc.

Comentarios:

6. ¿Se verifica la *Especificación de Requisitos* antes de ser aprobada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La verificación consiste en revisar la claridad de la redacción de la Especificación de Requisitos y su consistencia con la Descripción del Producto y con el estándar de documentación requerido en el Proceso Específico. Adicionalmente se revisa que los requisitos sean completos y no ambiguos o contradictorios.

Comentarios:

7. ¿Se corrigen los defectos encontrados en la *Especificación de Requisitos* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

8. ¿Se valida la *Especificación de Requisitos* antes de ser aprobada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La validación es la actividad que confirma que la Especificación de Requisitos resultante cumple con las necesidades y expectativas acordadas, incluyendo las pruebas sobre la interfaz de usuario. Los defectos encontrados se documentan en un Reporte de Validación.

Comentarios:

9. ¿Se corrigen los defectos encontrados en la *Especificación de Requisitos* con base en el *Reporte de Validación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Validación contiene el registro de los participantes en la validación, la fecha y lugar en que se realiza, y la duración.

Comentarios:

10. ¿Se elabora o modifica el *Plan de Pruebas de Sistema* que será utilizado en la entrega del software al *Cliente*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Plan de Pruebas de Sistema identifica las pruebas requeridas para el cumplimiento de los requisitos especificados. Una práctica común para obtener el Plan de Pruebas de Sistema es hacer que el Cliente participe en la definición de los casos de prueba.

Comentarios:

11. ¿Se verifica el *Plan de Pruebas de Sistema* antes de ser aprobado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La verificación consiste en revisar la consistencia del Plan de Pruebas de Sistema en relación con la Especificación de Requisitos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

12. ¿Se corrigen los defectos encontrados en el *Plan de Pruebas de Sistema* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

13. ¿Se documenta la versión preliminar del *Manual de Usuario* o bien se modifica uno ya existente?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Esta versión preliminar del Manual de Usuario servirá también para que el Cliente y la organización logren el entendimiento del proyecto y los objetivos establecidos para éste.

Comentarios:

14. ¿Se verifica el *Manual de Usuario* antes de ser aprobado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La verificación consiste en revisar la consistencia del Manual de Usuario en relación con la Especificación de Requisitos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

15. ¿Se corrigen los defectos encontrados en el *Manual de Usuario* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

16. Una vez que ya han sido verificados los documentos, ¿se incorporan la *Especificación de Requisitos*, los *Planes de Pruebas de Sistema*, y el *Manual de Usuario* como líneas base a la *Configuración de Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas del Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.

Comentarios:

17. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

REALIZACIÓN DE LA FASE DE ANÁLISIS Y DISEÑO (A.3)

III. Durante la fase de construcción se desarrolla el conjunto de actividades para analizar los requisitos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado de esta fase se obtendrá la documentación del *Análisis y Diseño* y *Plan de Pruebas de Integración*.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los analistas o diseñadores de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los analistas poseen el conocimiento y la experiencia para la obtención, especificación y análisis de los requisitos; mientras que el diseñador posee el conocimiento y la experiencia en el diseño de la estructura de los componentes de software. En empresas pequeñas, ambos roles suelen ser desempeñados por una misma persona.

Comentarios:

2. ¿Se analiza la *Especificación de Requisitos* 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?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

El Reporte de Validación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

10. ¿Se genera o actualiza el *Registro de Rastreo* del proyecto?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Registro de Rastreo describe la relación entre los requisitos, elementos de análisis y diseño, componentes y planes de pruebas.

Comentarios:

11. ¿Se verifica el *Registro de Rastreo* antes de ser aprobado?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La verificación consiste en comprobar que el Registro de Rastreo contenga las relaciones adecuadas entre los requisitos y los elementos de Análisis y Diseño. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

12. ¿Se corrigen los defectos encontrados en el *Registro de Rastreo* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

13. ¿Se genera o actualiza el *Plan de Pruebas de Integración* del proyecto?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

Este plan contiene el orden de integración de los componentes o subsistemas en función de la parte arquitectónica del Análisis y Diseño. Este documento resume las pruebas que serán aplicadas para verificar la interacción entre los componentes.

Comentarios:

14. ¿Se verifica el *Plan de Pruebas de Integración* antes de ser aprobado?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La verificación consiste en asegurar la consistencia del Plan de Pruebas de Integración con el Análisis y Diseño y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

15. ¿Se corrigen los defectos encontrados en el *Plan de Pruebas de Integración* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

16. Una vez que ya han sido verificados los documentos, ¿se incorporan el *Análisis y Diseño*, el *Registro de Rastreo*, y el *Plan de Pruebas de Integración* como líneas base a la *Configuración de Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.

Comentarios:

17. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

REALIZACIÓN DE LA FASE DE CONSTRUCCIÓN (A.4)

IV. Durante la fase de construcción se desarrolla el conjunto de actividades para producir el (los) *Componente(s)* de software que correspondan al *Análisis y Diseño*, así como la realización de pruebas unitarias. Como resultado de esta fase se obtendrán el (los) *Componente(s)* de software ya probados.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los programadores de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los programadores poseen el conocimiento y/o la experiencia en la programación, integración y realización de pruebas unitarias.

Comentarios:

2. ¿Se implementa o se modifica el (los) *Componente(s)* de software con base al *Análisis y Diseño* obtenido en la fase anterior?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Un Componente es un conjunto de unidades de código relacionadas con un fin.

Comentarios:

3. ¿Son definidas y aplicadas por los programadores las pruebas unitarias que verifiquen el funcionamiento de cada *Componente* de acuerdo con el *Análisis y Diseño*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

4. ¿Son corregidos los defectos encontrados hasta lograr pruebas unitarias exitosas (es decir, sin defectos)?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

5. ¿Se actualiza el *Registro de Rastreo* incorporando el (los) *Componente(s)* construidos o modificados?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

6. ¿Se verifica el *Registro de Rastreo* antes de ser aprobado?

La verificación consiste en asegurar que el Registro de Rastreo contenga las relaciones adecuadas entre los elementos de Análisis y Diseño y el (los) Componente(s). Los defectos encontrados se documentan en un Reporte de Verificación.

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

7. ¿Se corrigen los defectos encontrados en el *Registro de Rastreo* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

8. Una vez que ya han sido verificados los documentos, ¿se incorporan el (los) *Componente(s)* y el *Registro de Rastreo* como líneas base a la *Configuración de Software*?

La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

9. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

REALIZACIÓN DE LA FASE DE INTEGRACIÓN Y PRUEBAS (A.5)

V. Durante la fase de integración y pruebas se realiza el conjunto de actividades para integrar y probar el (los) *Componente(s)* de software, con base a los *Planes de Pruebas de Integración y de Sistema*, con la finalidad de obtener el *Software* que satisfaga los requisitos especificados. Se genera también las versiones finales del *Manual de Usuario*, *Manual de Operación* y *Manual de Mantenimiento*. Como resultado final se obtiene el producto de *Software* ya probado y documentado.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los programadores o responsables de pruebas de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Los responsables de pruebas tienen conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema.

Comentarios:

2. ¿Son integrados el (los) *Componente(s)* en subsistemas o en el sistema de *Software* y se aplican las pruebas siguiendo el *Plan de Pruebas de integración*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

3. ¿Son documentados los resultados de las pruebas en un *Reporte de Pruebas de Integración*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Reporte de Pruebas de Integración registra a los participantes, la fecha y lugar en que se realiza, la duración y los defectos encontrados.

Comentarios:

4. ¿Son corregidos los defectos encontrados, con base en el Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (es decir, sin defectos)?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

5. ¿Se actualiza el *Registro de Rastreo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comentarios:

6. ¿Se documenta el *Manual de Operación* o bien se modifica uno ya existente?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El Manual de Operación es un documento electrónico o impreso que contiene la información indispensable para la instalación y administración del software, así como el entorno de operación (sistema operativo, base de datos, servidores, etc.).

Comentarios:

13. ¿Se verifica el *Manual de Usuario* antes de ser aprobado?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La verificación consiste en asegurar la consistencia del Manual de Usuario con el Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

14. ¿Se corrigen los defectos encontrados en el *Manual de Usuario* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

15. Una vez que ya han sido verificados los documentos, ¿se incorporan el *Software*, el *Reporte de Pruebas de Integración*, el *Registro de Rastreo*, el *Manual de Operación* y el *Manual de Usuario* como líneas base a la *Configuración de Software*?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas del Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.

Comentarios:

16. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

Comentarios:

REALIZACIÓN DE LA FASE DE CIERRE (A.6)

VI. Durante la fase de cierre se realiza la integración final de la *Configuración de Software* generada en las fases para su entrega. Si fuera el caso, se identifican y documentan las *Lecciones Aprendidas* y se genera el *Reporte de Mediciones y Sugerencias de Mejora*.

1. ¿Se genera o actualiza el *Manual de Mantenimiento del Software*?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Manual de Mantenimiento es un documento electrónico o impreso que describe la Configuración de Software y el entorno usado para el desarrollo y realización de pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este manual deberá ser redactado en términos comprensibles para el personal de mantenimiento.

Comentarios:

2. ¿Se verifica el *Manual de Mantenimiento* antes de ser aprobado?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La verificación consiste en asegurar la consistencia del Manual de Mantenimiento con la Configuración de Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Comentarios:

3. ¿Se corrigen los defectos encontrados en el *Manual de Mantenimiento* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.

Comentarios:

4. Una vez que ya han sido verificado el documento, ¿se incorpora el *Manual de Mantenimiento* como línea base a la *Configuración de Software*?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.

Comentarios:

5. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

Comentarios:

Anexo D. Formatos diseñados para mejorar los procesos de la empresa

D.1. Formato para Minuta de Lanzamiento (DMS A.1.1)

Fecha:	[Fecha]	Hora de inicio:	[Rev]	Hora de fin:	[Rev]
Objetivo de la reunión	El objetivo de la reunión es obtener la minuta de revisión después de haber analizado con los miembros del <i>Equipo de Trabajo</i> el <i>Plan de Desarrollo</i> actual para lograr un entendimiento común y obtener su compromiso con el proyecto.				
Participantes de la reunión	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software. ET como participantes del proceso de Desarrollo y Mantenimiento de Software.				
Seguimiento de compromisos pendientes	[Inserte aquí el texto]				
Puntos tratados	[Inserte aquí el texto]				
Acuerdos obtenidos	[Inserte aquí el texto]				
Asuntos pendientes por resolver	[Inserte aquí el texto]				
Aprobación	Los presentes aseguran haber entendido el proyecto y se comprometen a realizarlo con profesionalismo y responsabilidad de acuerdo al Plan de Desarrollo revisado. [Inserte aquí el texto]				

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha	Fecha en la que se realiza la reunión.
Hora de inicio	Hora exacta en la que comienza la reunión.
Hora de fin	Hora exacta en la que termina la reunión.
Objetivo de la reunión	El objetivo de la reunión para el caso de la revisión del Plan de Desarrollo y el consecuente lanzamiento del proyecto ya se establece en el documento. Si fuera necesario, este objetivo puede modificarse para efectos de un mejor entendimiento.
Participantes de la reunión	Los roles de los participantes en la reunión y que han sido definidos como Equipo de Trabajo en el Plan de Desarrollo. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
Seguimiento de compromisos pendientes	Es responsabilidad del RDM documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
Puntos tratados	<p>El RDM debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo:</p> <ol style="list-style-type: none"> 1. Pase de lista. 2. Revisión del Plan de Desarrollo. 3. Ruegos y preguntas. 4. Obtención de compromiso del ET. 5. Firma de minuta. 6. Cierre de la reunión.
Acuerdos obtenidos	El RDM debe documentar con una lista los acuerdos finales o compromisos, estableciendo que los responsables y fechas que fueron establecidos por el RAPE en el Plan de Desarrollo son aceptados.
Asuntos pendientes por resolver	El RDM debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY o al RAPE) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
Aprobación	El RDM debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

D.2. Formato para Reporte de Actividades (DMS A.1.3)

Reporte de actividades No. [Inserte aquí el texto]			
Datos generales			
Nombre del proyecto	[Inserte aquí el texto]		
Referencia del proyecto	[Inserte aquí el texto]		
Nombre del RDM	[Inserte aquí el texto]		
Fecha del reporte	[Fecha]	Período comprendido	[Fecha] al [Fecha]

Registro de actividades							
Actividad del Plan de Desarrollo	Fecha de inicio		Fecha de fin		Esfuerzo (horas)		Responsable
	Planeado	Real	Planeado	Real	Planeado	Real	
[Inserte aquí el texto]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Inserte aquí el texto]
Tiempo total de producción					[Rev]	[Rev]	

Defectos encontrados			
Productos verificados	Registro de tiempos (esfuerzo en horas)		Número de defectos
	En verificación	En corrección	
Minuta de Lanzamiento	[Rev]	[Rev]	[Inserte aquí el texto]
Necesidades del Sistema	[Rev]	[Rev]	[Inserte aquí el texto]
Especificación de Requerimientos	[Rev]	[Rev]	[Inserte aquí el texto]
Plan de Pruebas de Sistema	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Usuario	[Rev]	[Rev]	[Inserte aquí el texto]
Registro de Rastreo	[Rev]	[Rev]	[Inserte aquí el texto]
Plan de Pruebas de Integración	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Operación	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Usuario	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Mantenimiento	[Rev]	[Rev]	[Inserte aquí el texto]
Productos validados	Registro de tiempos (esfuerzo en horas)		Número de defectos
	En validación	En corrección	
Especificación de Requerimientos	[Rev]	[Rev]	[Inserte aquí el texto]
Análisis y Diseño	[Rev]	[Rev]	[Inserte aquí el texto]
Productos probados	Registro de tiempos (esfuerzo en horas)		Número de defectos
	En prueba	En corrección	
[Inserte aquí el texto]	[Rev]	[Rev]	[Inserte aquí el texto]
	[Rev]	[Rev]	[Inserte aquí el texto]

Instrucciones de llenado de la plantilla

Campo	Descripción
<i>Datos generales</i>	
Nombre del proyecto	Nombre que le fue asignado al proyecto en el Plan de Desarrollo.
Referencia del proyecto	Referencia única que se le da a cada proyecto para referirse a él de forma abreviada. Estas referencias suelen tener identificadores que ayudan al personal a determinar la naturaleza del proyecto sin consultar el nombre o descripción completa del mismo. El estilo de las referencias debe ser definido por la empresa y debe ser respetado por todo el Equipo de Trabajo.
Nombre del RDM	Nombre del Responsable del Desarrollo y Mantenimiento de Software que fue asignado al proyecto en cuestión.
Fecha de reporte	Fecha en la que se está entregando el reporte de avance. Es obvio que cada modificación de avance requerirá la modificación del reporte existente y por ende la fecha será actualizada.
Periodo comprendido	Periodo de tiempo que se está reportando. Por ejemplo, la Fecha del reporte puede ser 08/06/2013 y se están reportando las actividades que se realizaron entre el 01/04/2013 y el 07/06/2013.
<i>Registro de actividades</i>	
Actividad del plan de desarrollo	Las actividades listadas en esta columna deben corresponder con las actividades que hayan sido registradas en el Plan de Desarrollo.
Fecha de inicio – Planeado	Fecha planeada por el RAPE para iniciar la actividad correspondiente.
Fecha de inicio – Real	Fecha real en que inició la actividad correspondiente.
Fecha de fin – Planeado	Fecha planeada por el RAPE para finalizar la actividad correspondiente.
Fecha de fin – Real	Fecha real en que finalizó la actividad correspondiente.
Esfuerzo (horas) – Planeado	Horas planeadas por el RAPE para realizar la actividad correspondiente.
Esfuerzo (horas) – Real	Horas reales para realizar la actividad correspondiente.
Responsable	El responsable de cada actividad tomada del Plan de Desarrollo y que acordó comprometerse a realizarla en tiempo y forma a través de la Minuta de Lanzamiento.
Tiempo total de producción	Sumatoria de todos los campos registrados para cada actividad en Esfuerzo (horas) – Planeado y Esfuerzo (horas) – Real, respectivamente.
<i>Defectos encontrados</i>	
Registro de tiempos (esfuerzo en horas) – En verificación	Tiempo total (en horas) que el responsable de la verificación tardó en revisar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
Registro de tiempos (esfuerzo en horas) – En corrección	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
Número de defectos	Total de defectos encontrados por cada producto verificado.
Registro de tiempos (esfuerzo en horas) – En validación	Tiempo total (en horas) que el responsable de la validación tardó en revisar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.

Registro de tiempos (esfuerzo en horas) – En corrección	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
Número de defectos	Total de defectos encontrados por cada producto validado.
Productos probados	Se deberán listar todos los prototipos, módulos, componentes, etc., que los PR someten a pruebas unitarias o pruebas de integración.
Registro de tiempos (esfuerzo en horas) – En prueba	Tiempo total (en horas) que el PR tardó en probar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
Registro de tiempos (esfuerzo en horas) – En corrección	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
Número de defectos	Total de defectos encontrados por cada producto probado.
Tamaño de los productos	
Producto	Se deberán listar todos los productos generados en el proceso de DMS. Por ejemplo: Casos de Uso, Especificación de Requerimientos, Análisis y Diseño, Componente de Software, Manual de Usuario, etc.
Tamaño real	Se deberá registrar el tamaño real (es decir, cuando el producto ya esté terminado) para cada producto listado anteriormente. Por ejemplo, para los Casos de Uso se podría registrar: # de casos de uso y # de páginas; para la Especificación de Requerimientos: # de requerimientos y # de páginas de la Especificación de Requerimientos; para el Análisis y Diseño: # de diagramas de clases, # de tablas de la BD, # de páginas del Análisis y Diseño; para el Componente de Software: LOC (Líneas de Código); y para el Manual de Usuario: # de páginas del Manual de Usuario.

D.3. Formato para Necesidades del Sistema (DMS A.2.3)

1. Alcance

[Inserte aquí el texto]

El contenido de este apartado debe proporcionar una breve introducción del documento, su relación con otros posibles documentos, su finalidad y los destinatarios en relación a la información que contiene.

1.1. Identificación

[Inserte aquí el texto]

Identificación del sistema, proporcionando su nombre y abreviatura (si procede).

1.2. Visión general del documento

[Inserte aquí el texto]

Explicación del propósito, audiencia, y consideraciones de seguridad o privacidad del documento.

- *Generalmente, el propósito del documento suele ser uno de los siguientes:*
- *Comunicar las necesidades y expectativas del cliente.*
- *Comunicar el entendimiento del proyecto.*
- *Obtener acuerdos entre las partes implicadas (personal representante de la parte cliente, personal del equipo de desarrollo, etc.).*

1.3. Visión general del sistema

[Inserte aquí el texto]

Resumen del propósito del sistema o subsistema propuesto (al cual se aplica la descripción del sistema).

1.4. Personal involucrado

Nombre	[Inserte aquí el texto]
Rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]
Información de contacto	[Inserte aquí el texto]
Aprobación	[Inserte aquí el texto]

Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el RDM pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requerimientos, validaciones de seguimiento, etc.

2. Documentos referenciados

N°	Título	Ruta	Versión	Fecha	Autor
	[Título]	[Ruta]	[Rev]	[Fecha]	[Autor]

Relación de los documentos a los que se hace referencia en las necesidades del sistema, indicando, según proceda: el nombre del manual, documento, código del documento, título, ruta, revisión, fecha y origen.

3. Situación actual

[Inserte aquí el texto]

Cuando el sistema propuesto tenga como finalidad la sustitución o ampliación de un sistema o situación existente, este apartado contendrá una descripción suficiente del actual (aunque éste último sea un sistema manual).

3.1. Antecedentes

[Inserte aquí el texto]

Visión general del sistema o situación actual, incluyendo: misión, objetivos, alcance, modo de uso o aplicación, etc.

3.2. Políticas y restricciones operacionales

[Inserte aquí el texto]

Relación de políticas o restricciones de cualquier índole impuestas sobre el sistema o situación actual. Algunos ejemplos pueden ser:

- *Restricciones sobre el número de usuarios capaces de utilizar el sistema.*
- *Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)*
- *Restricciones de seguridad o relativas a protección de datos.*
- *Restricciones de software (uso obligado de una determinada BD, Sistema Operativo, etc.)*
- *Restricciones de recursos operacionales como espacio físico.*

3.3. Descripción del sistema o situación actual

[Inserte aquí el texto]

Descripción detallada del sistema o situación actual y de su funcionamiento, incluyendo los siguientes subapartados:

- *Enumeración y descripción de funciones, características y capacidades del sistema o situación actual.*
- *En el caso de considerarse apropiado puede aportarse información adicional como:*
 - *Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema o situación actual.*
 - *Interacción entre componentes del sistema.*
- *Descripción del entorno de operación y sus características.*
- *Interacción del sistema con otros sistemas externos.*
- *Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, y demás.*
- *Atributos de calidad como disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, y demás.*
- *Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.*
- *Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, coste de las operaciones, y demás.*

Es importante que la descripción del sistema o situación actual sea lo más simple y clara posible, para que todos los lectores del documento logren entenderla completamente. Es importante que ésta se realice usando la terminología del usuario.

3.4. Tipos de usuarios

Tipo de usuario	[Inserte aquí el texto]
Responsabilidad	[Inserte aquí el texto]
Formación	[Inserte aquí el texto]
Habilidades	[Inserte aquí el texto]
Actividades	[Inserte aquí el texto]
Interacción con el sistema	[Inserte aquí el texto]

Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual interactúa con el sistema. Factores como la responsabilidad, habilidades, competencias, etc., distinguen a los distintos tipos de usuarios.

3.5 Mantenimiento / soporte

[Inserte aquí el texto]

Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.

3.6. Necesidad y naturaleza de los cambios

[Inserte aquí el texto]

Descripción de las carencias, defectos o debilidades del sistema o situación actual y que motivan al desarrollo de un nuevo sistema o, a una modificación del existente.

De no existir un sistema anterior (ni tan siquiera manual), en este apartado y subapartados deben quedar reflejadas las justificaciones que llevan al cambio.

3.7. Descripción de los cambios deseados

[Inserte aquí el texto]

Enumeración de las capacidades, funciones, procesos, etc. que deben generarse o modificarse para satisfacer las nuevas necesidades.

Los cambios deben basarse en el sistema descrito en el punto 3.3. Si no existe un sistema anterior, en este apartado se enumeran las capacidades requeridas del nuevo sistema.

De forma apropiada podrá hacerse referencia a:

- *Cambios en la capacidad. Descripción de las funciones y características que deben añadirse, eliminarse o modificarse para conseguir los objetivos y requisitos del nuevo sistema.*
- *Cambios en el proceso del sistema. Descripción de los cambios en el proceso o procesos de transformación de datos que darán lugar a nuevos resultados con los mismos datos, mismos resultados con nuevos datos o ambos.*
- *Cambios de interfaces. Descripción de los cambios en el sistema que provocan cambios en los interfaces y cambios en los interfaces que causan cambios en el sistema.*
- *Cambios de personal. Descripción de cambios en personal causados por nuevos requisitos, cambios en tipos de usuarios o ambos.*
- *Cambios de entorno. Descripción de cambios en el entorno operacional que causan cambios en las funciones, procesos, interfaces, o personal del sistema y/o cambios que deben realizarse en el entorno por causa de cambios de las funciones, procesos, interfaces, o personal del sistema.*
- *Cambios operacionales. Descripción de cambios a procedimientos, métodos, rutinas de trabajo del usuario causados por los cambios mencionados.*
- *Cambios de soporte. Cambios en los requisitos de soporte causados por cambios en las funciones, procesos, interfaces o personal del sistema y/o cambios en las funciones, procesos, interfaces o personal del sistema causados por cambios de soporte.*
- *Otros cambios. Descripción de otros cambios que afectarán a los usuarios.*

Es necesario identificar las prioridades entre los cambios deseados y las nuevas características. Esta identificación puede realizarse clasificando cada uno de los cambios como esencial, deseado u opcional. Esta información será útil en la toma de decisiones durante el desarrollo y en el caso de interrupciones o invasiones de agendas o presupuestos.

4. Sistema propuesto

[Inserte aquí el texto]

La descripción del sistema propuesto debe ceñirse a los conceptos básicos que indican las nuevas capacidades operacionales sin entrar en especificaciones de diseño, a no ser que sean restricciones impuestas.

4.1. Antecedentes

[Inserte aquí el texto]

Visión general del sistema propuesto, incluyendo: misión, objetivos, alcance, modo de uso o aplicación, etc.

4.2. Políticas y restricciones operacionales

[Inserte aquí el texto]

Relación de políticas o restricciones de cualquier índole aplicables al sistema propuesto. Algunos ejemplos pueden ser:

- *Restricciones sobre el número de usuarios capaces de utilizar el sistema.*
- *Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)*
- *Restricciones de seguridad o relativas a protección de datos.*
- *Restricciones de software (uso obligado de una determinada BD, Sistema Operativo, etc.)*
- *Restricciones de recursos operacionales como espacio físico.*

4.3. Descripción del sistema propuesto

[Inserte aquí el texto]

Descripción detallada del sistema propuesto y de su funcionamiento, incluyendo los siguientes subapartados:

- *Enumeración y descripción de funciones, características y capacidades del sistema propuesto.*
- *Deben identificarse las prioridades entre funcionalidades y características. Esta identificación puede realizarse clasificando cada una las funciones, características y capacidades como esencial, deseado u opcional.*
- *Deben incluirse en la medida de lo posible:*
 - *Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema propuesto desde el punto de vista del usuario.*
 - *Identificación de interacciones entre componentes del sistema.*
- *Descripción del entorno de operación y sus características.*
- *Interacción del sistema con otros sistemas.*
- *Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, etc.*

- *Atributos de calidad como disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, etc.*
- *Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.*
- *Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, costo de las operaciones, etc.*

Es importante que las funcionalidades del sistema propuesto sean lo más simples y claras posibles, para que todos los lectores del documento logren entenderlos completamente. Debe realizarse usando la terminología del usuario.

Nota: El RA debe organizar la información en este apartado como considere apropiado para el sistema o situación. Aquellas partes de la descripción que sean extensas podrán incluirse como apéndices del documento.

4.4. Tipos de usuarios

Tipo de usuario	[Inserte aquí el texto]
Responsabilidad	[Inserte aquí el texto]
Formación	[Inserte aquí el texto]
Habilidades	[Inserte aquí el texto]
Actividades	[Inserte aquí el texto]
Interacción con el sistema	[Inserte aquí el texto]

Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual un usuario interactúa con el sistema. Factores como la responsabilidad, habilidades, competencias, etc. distinguen a los distintos tipos de usuarios.

4.5. Mantenimiento / soporte

[Inserte aquí el texto]

Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.

4.6. Escenarios operacionales

[Inserte aquí el texto]

Relación de escenarios operacionales. Un escenario es una descripción paso a paso de cómo el sistema propuesto debe operar e interactuar con sus usuarios y sus interfaces externas, bajo unas determinadas circunstancias.

Los escenarios operacionales proporcionan información sobre cómo interactúan todas las partes del sistema propuesto, los usuarios y otras entidades externas. También pueden utilizarse para describir lo que no debe hacer el sistema.

4.7. Futuras evoluciones

[Inserte aquí el texto]

Descripción de las previsiones de evolución que se tienen previstas para el sistema (si se tiene).

Esta información resulta útil para el personal involucrado en el desarrollo del nuevo sistema, y podrá preparar la arquitectura del sistema para asimilar cambios futuros con el menor impacto.

4.8. Cambios considerados pero no incluidos

[Inserte aquí el texto]

Identificación de cambios considerados pero no incluidos en el sistema propuesto y el motivo por el que no han sido incluidos.

5. Resumen de mejoras

[Inserte aquí el texto]

Resumen de los beneficios proporcionados por el sistema propuesto. Este resumen puede incluir las siguientes secciones:

- *Nuevas características o funcionalidades.*
- *Actualización de características existentes.*
- *Eliminación de características obsoletas, no útiles o confusas.*
- *Mejoras de rendimiento como tiempo de respuesta, mejoras de calidad, etc.*

6. Información adicional

[Inserte aquí el texto]

Cualquier información adicional que facilite la comprensión del documento en sí.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RE y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RAN y RE tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RE hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RAN y RE que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar dado lo definido en los Criterios de Verificación de la documentación de procesos. Por ejemplo: Claridad de redacción, Consistencia o adherencia con la Descripción del producto, Conformidad con el estándar de documentación. Se recomienda incluir todos los elementos que forman el documento de las Necesidades del Cliente y aplicar los criterios definidos.
Descripción del defecto encontrado	El RAN y RE deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento está redactado claramente”. Un ejemplo de cómo describir un defecto puede ser: “Las necesidades no son consistentes con la descripción del producto dado que solo considera uno de los objetivos del producto y no los dos.”
Fecha de aprobación	Fecha en la que RAN y RE se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

D.5. Formato para Especificación de Requerimientos (DMS A.2.6)

1. Introducción

[Inserte aquí el texto]

La introducción de la Especificación de Requerimientos (SRS) debe proporcionar una visión general de la misma. Debe incluir el propósito, el alcance, las definiciones y acrónimos, y las referencias, de acuerdo al proyecto.

1.1. Propósito

[Inserte aquí el texto]

- *Propósito del documento.*
- *Audiencia a la que va dirigido.*

1.2. Alcance

[Inserte aquí el texto]

- *Identificación del producto(s) a desarrollar mediante el nombre asignado en las Necesidades del Sistema o bien la propuesta de uno diferente.*
- *Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir.*

1.3. Personal involucrado

Nombre	[Inserte aquí el texto]
Rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]
Información de contacto	[Inserte aquí el texto]
Aprobación	[Inserte aquí el texto]

Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el RDM pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requerimientos, validaciones de seguimiento, etc.

1.4. Definiciones, acrónimos y abreviaturas

[Inserte aquí el texto]

Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.

1.5. Referencias

Referencia	Título	Ruta	Fecha	Autor
[Ref.]	[Título]	[Ruta]	[Fecha]	[Autor]

Relación completa de todos los documentos relacionados en la especificación de requerimientos, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.

1.6. Resumen del documento

[Inserte aquí el texto]

- Descripción del contenido del resto del documento.
- Explicación de la organización del documento.

2. Descripción general

2.1. Perspectiva del producto

[Inserte aquí el texto]

Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, el uso de un diagrama que sitúe al producto dentro del sistema e identifique sus conexiones facilita la comprensión.

2.2. Funcionalidad del producto

[Inserte aquí el texto]

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en detalle. En ocasiones la información de esta sección puede tomarse de un documento de mayor nivel (ej. Necesidades del Sistema).

Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor puedan entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.

2.3. Características de los usuarios

Tipo de usuario	[Inserte aquí el texto]
Formación	[Inserte aquí el texto]
Habilidades	[Inserte aquí el texto]
Actividades	[Inserte aquí el texto]

Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

2.4. Restricciones

[Inserte aquí el texto]

Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.

2.5. Suposiciones y dependencias

[Inserte aquí el texto]

Descripción de aquellos factores que, si cambian, pueden afectar a los requerimientos. Por ejemplo, una suposición puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la Especificación de Requerimientos debería modificarse.

2.6. Evolución previsible del sistema

[Inserte aquí el texto]

Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

2.7. Diagrama de contexto de casos de uso para el sistema

[Inserte aquí el texto]

Este diagrama de casos de uso debe ser general y debe resumir las funcionalidades identificadas en las Necesidades del Sistema. Los Analistas deben partir del documento de Necesidades del Sistema para resumir con un diagrama de casos de uso las funcionalidades requeridas (y confirmadas) por el Cliente.

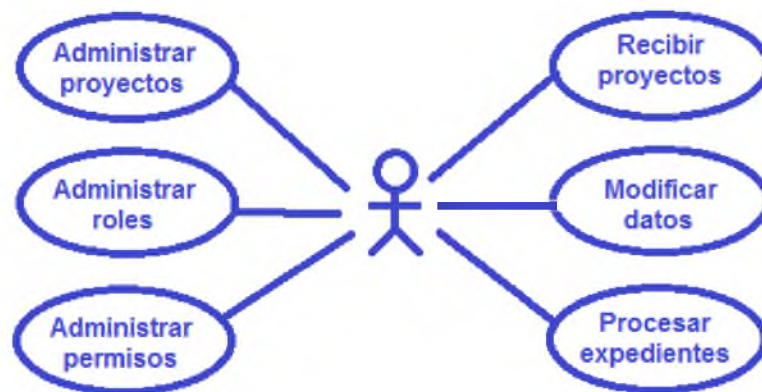


Figura 1. Ejemplo de diagrama de contexto de casos de uso

3. Requerimientos funcionales

Esta es la sección más extensa y más importante del documento. Debe contener una lista detallada y completa de los requerimientos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requerimientos debe ser el suficiente para que los Diseñadores realicen un diseño del sistema que los satisfaga y los encargados de las pruebas (Programadores) puedan determinar si éstos se satisfacen.

Los requerimientos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (RF 10.1, RF 10.2, etc.). Es muy importante tener presente que el primer número después de RF corresponderá a cada funcionalidad identificada en el diagrama de contexto de casos de uso para el sistema. El segundo número corresponderá al número de requerimiento en sí.

Para cada requerimiento debe rellenarse la siguiente tabla, obviamente deberán ser agregadas tantas tablas como número de requerimientos que se desee especificar, por ejemplo:

Número y nombre del requerimiento	RF 1.12 Modificación de contraseña de usuario		
Origen del requerimiento	Usuario final		
Prioridad del requerimiento	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional		
Descripción	Este requerimiento indica que el sistema podrá permitir la modificación de la contraseña del usuario.		
Entrada	Se mostrará el nombre del usuario y se requerirán los siguientes datos: Contraseña (ob), Nueva Contraseña (ob), Repetir nueva contraseña (ob).		
Salida	Se modifica y registra la nueva contraseña con la que el usuario accede al sistema. Se mostrarán mensajes de error en caso de que la modificación haya sido fallida, bien por problemas sintácticos, por falta de datos, o por problemas de autenticación con la antigua contraseña.		
Proceso	<p>El sistema muestra la pantalla para introducir los datos requeridos. Una vez rellenos los campos, y mediante una opción de modificación se procederá a la introducción de estos datos en la BD del sistema. Se dispondrá de una opción de borrado que permita hacer correcciones antes del envío de datos.</p> <p><i>Se mostrará al usuario una pantalla con los nuevos datos con los que ha sido registrado.</i></p>		
Referencias	Con otros requerimientos	RF 1.10, RF 1.11	
	Con casos de uso	Gestión de Usuarios	
Fecha de creación	12/03/2013	Último cambio	06/05/2013

Número y nombre de requerimiento	[Inserte aquí el texto]
---	-------------------------

Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		
Referencias	Con otros requerimientos	[Inserte aquí el texto]	
	Con casos de uso	[Inserte aquí el texto]	
Fecha de creación	[Fecha]	Último cambio	[Fecha]

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		
Referencias	Con otros requerimientos	[Inserte aquí el texto]	
	Con casos de uso	[Inserte aquí el texto]	
Fecha de creación	[Fecha]	Último cambio	[Fecha]

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		

<i>Referencias</i>	Con otros requerimientos		[Inserte aquí el texto]
	Con casos de uso		[Inserte aquí el texto]
<i>Fecha de creación</i>	[Fecha]	<i>Último cambio</i>	[Fecha]

<i>Número y nombre de requerimiento</i>	[Inserte aquí el texto]		
<i>Origen del requerimiento</i>	[Inserte aquí el texto]		
<i>Prioridad del requerimiento</i>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		
<i>Descripción</i>	[Inserte aquí el texto]		
<i>Entrada</i>	[Inserte aquí el texto]		
<i>Salida</i>	[Inserte aquí el texto]		
<i>Proceso</i>	[Inserte aquí el texto]		
<i>Referencias</i>	Con otros requerimientos		[Inserte aquí el texto]
	Con casos de uso		[Inserte aquí el texto]
<i>Fecha de creación</i>	[Fecha]	<i>Último cambio</i>	[Fecha]

<i>Número y nombre de requerimiento</i>	[Inserte aquí el texto]		
<i>Origen del requerimiento</i>	[Inserte aquí el texto]		
<i>Prioridad del requerimiento</i>	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		
<i>Descripción</i>	[Inserte aquí el texto]		
<i>Entrada</i>	[Inserte aquí el texto]		
<i>Salida</i>	[Inserte aquí el texto]		
<i>Proceso</i>	[Inserte aquí el texto]		
<i>Referencias</i>	Con otros requerimientos		[Inserte aquí el texto]
	Con casos de uso		[Inserte aquí el texto]
<i>Fecha de creación</i>	[Fecha]	<i>Último cambio</i>	[Fecha]

3.1. Requerimientos de las interfaces

[Inserte aquí el texto]

Descripción detallada de todas las entradas y salidas del sistema de software.

3.1.1. Interfaces de usuario

[Inserte aquí el texto]

Describir los requerimientos de la interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas de interfaz. Por ejemplo, posiblemente el cliente ha especificado el estilo y los colores de su producto. En este sentido, el AN debe describir exactamente cómo el producto aparecerá a su usuario previsto.

Descripción	El producto debe ser fácil de usar cuando se trate de un usuario sin entrenamiento previo.
Justificación	El cliente indica que los usuarios potenciales pueden nunca antes haber usado este tipo de productos.

Descripción	[Inserte aquí el texto]
Justificación	[Inserte aquí el texto]

Descripción	[Inserte aquí el texto]
Justificación	[Inserte aquí el texto]

3.1.2. Interfaces de hardware

[Inserte aquí el texto]

El AN debe especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema si éste fuera el caso. Se incluirán características de configuración. Por ejemplo:

Descripción	El producto debe obtener la información de llegada a la oficina de los trabajadores de un dispositivo lector de huellas digitales.
Justificación	Es importante que para registrar la entrada al trabajo se tenga un control automático y se reduzca el tiempo de retardos. El sistema lector debe ser configurado de acuerdo a las especificaciones técnicas del fabricante.

Descripción	[Inserte aquí el texto]
Justificación	[Inserte aquí el texto]

Descripción	[Inserte aquí el texto]
--------------------	-------------------------

<i>Justificación</i>	[Inserte aquí el texto]
----------------------	-------------------------

3.1.3. Interfaces de software

[Inserte aquí el texto]

El AN indicará si es necesario integrar el producto requerido por el cliente con otros productos de software. Para cada producto de software debe especificarse lo siguiente:

- *Descripción del producto software utilizado.*
- *Propósito de la interfaz.*
- *Définition de la interfaz: contiendo y formato.*

3.1.4. Interfaces de comunicación

[Inserte aquí el texto]

Describir los requerimientos de interfaces de comunicación si existen comunicaciones con otros sistemas y cuáles son los protocolos de comunicación. Estos requerimientos son registrados con descripciones textuales y deberá usarse una viñeta por cada uno. Por ejemplo:

- *El sistema debe obtener la lectura de las ventas del dispositivo móvil que se entrega a cada empleado a través de una comunicación Bluetooth.*
- *Las lecturas de pedidos serán descargadas a través del puerto RS232 entre dispositivos.*

3.2. Requerimientos no funcionales

3.2.1. Requerimientos de rendimiento

[Inserte aquí el texto]

Especificación de los requerimientos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. Todos estos requerimientos deben ser medibles.

Estos requerimientos son registrados con descripciones textuales y deberá usarse una viñeta por cada uno. Por ejemplo:

- *El 95% de las transacciones deben realizarse en menos de 1 segundo.*
- *Los registros de todas las ventas realizadas deben ser actualizados cada 3 horas.*

3.2.2. Seguridad

[Inserte aquí el texto]

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requerimientos pueden especificar:

- *Empleo de técnicas criptográficas.*
- *Registro de ficheros con “logs” de actividad.*
- *Asignación de determinadas funcionalidades a determinados módulos.*
- *Forma de acceso al sistema.*
- *Restricciones de comunicación entre determinados módulos.*
- *Comprobaciones de integridad de información crítica.*

3.2.3. Fiabilidad

[Inserte aquí el texto]

Especificación de los factores de fiabilidad necesaria del sistema que son subjetivos y no medibles. Estos se expresan generalmente como el tiempo entre los incidentes permisibles, o el total permisible de incidentes. Se recomienda utilizar la siguiente clasificación de incidentes en el funcionamiento de un sistema:

- *Transitorio: Ocurre solamente con ciertas entradas.*
- *Permanente: Ocurre con todas las entradas.*
- *Recuperable: El sistema puede recuperarse sin la intervención del operador.*
- *Irrecuperable: Es necesaria la intervención del operador para recuperarse del fallo.*
- *No corruptivo: El fallo no corrompe el estado del sistema o los datos.*
- *Corruptivo: El fallo corrompe el estado del sistema o los datos.*

<i>Clase de incidente</i>	Permanente, no corruptivo.
<i>Descripción</i>	Incidente al leer tarjeta.
<i>Ejemplo</i>	El sistema deja de funcionar con cualquier tarjeta que se le introduzca. El software debe reiniciarse para corregir el fallo.
<i>Métrica de fiabilidad</i>	ROCOF (Rate Of Change Of Frequency) 1 ocurrencia/1,000 días

<i>Clase de incidente</i>	[Inserte aquí el texto]
<i>Descripción</i>	[Inserte aquí el texto]
<i>Ejemplo</i>	[Inserte aquí el texto]
<i>Métrica de fiabilidad</i>	[Inserte aquí el texto]

<i>Clase de incidente</i>	[Inserte aquí el texto]
<i>Descripción</i>	[Inserte aquí el texto]
<i>Ejemplo</i>	[Inserte aquí el texto]
<i>Métrica de fiabilidad</i>	[Inserte aquí el texto]

3.2.4. Disponibilidad

[Inserte aquí el texto]

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad. Por ejemplo:

Descripción	Disponibilidad total
Justificación	La disponibilidad del sistema debe ser del 100% con un nivel de servicio para los usuarios de 7 días X 24 horas, garantizando un esquema adecuado que permita ante una posible falla en cualquiera de sus componentes, contar con una contingencia y generación de alarmas.

3.2.5. Mantenibilidad

[Inserte aquí el texto]

Identificación del tipo de mantenimiento necesario del sistema. Especificación de quién debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador. Especificación de cuándo deben realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanal y mensual.

Descripción	Documentación de software.
Justificación	Todo el sistema deberá estar complemente documentado, cada uno de los componentes de software que forman parte del producto final deberán estar debidamente documentados tanto en el código fuente como en los manuales de uso y de mantenimiento.

Descripción	Errores futuros.
Justificación	El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema

3.2.6. Portabilidad

[Inserte aquí el texto]

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

- *Porcentaje de componentes dependientes del servidor.*
- *Porcentaje de código dependiente del servidor.*
- *Uso de un determinado lenguaje por su portabilidad.*
- *Uso de un determinado compilador o plataforma de desarrollo.*
- *Uso de un determinado sistema operativo.*

3.3. Otros requerimientos

[Inserte aquí el texto]

Cualquier otro requerimiento que no encaje en ninguna de las secciones anteriores, por ejemplo:

- *Requerimientos culturales y políticos*
- *Requerimientos legales*
- *Requerimientos ambientales.*

Descripción	El producto deberá respetar la norma vigente para el cálculo de impuesto y exenciones.
Justificación	El cliente establece que los cálculos de impuestos deber respetar la ley fiscal actual del país.

Descripción	El producto deberá mostrar los textos en Zapoteco y Quechua, además del Español.
Justificación	Es importante mantener el equilibrio cultural entre los tipos de usuario planeados para el producto.

4. Apéndices

[Inserte aquí el texto]

Este apartado es opcional y puede contener todo tipo de información relevante para la Especificación de Requerimientos pero que, propiamente, no forme parte de la misma. (ej. Documentos entregados por el cliente, Manuales utilizados, Políticas, etc.).

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RE y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RE y RAN que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Descripción del producto, Descripción de requerimientos, etc. Se recomienda incluir todos los elementos que forman el documento de la Especificación de Requerimientos y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Requerimientos Funcionales, Requerimientos de Interfaz, etc.
Descripción del defecto encontrado	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los requerimientos del sistema”. Un ejemplo de cómo describir un defecto puede ser: “Los requerimientos de mantenibilidad no consideran quién es el responsable del mantenimiento del sistema.”
Fecha de aprobación	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
Lugar	Lugar en donde el RDM y el CL y US realizan la validación del documento.
Hora de inicio	Hora exacta en la que comienza la validación del documento.
Hora de fin	Hora exacta en la que termina la validación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RDM, CL y US que participan en la validación.
Puesto (rol)	Puestos o roles de los participantes en la validación.
Elemento a validar	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Descripción del producto, Requerimientos entendibles, etc. Se recomienda incluir todos los elementos que forman el documento de la Especificación de Requerimientos y aplicar los criterios definidos para su validación.
Descripción del defecto encontrado	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los requerimientos del sistema”. Un ejemplo de cómo describir un defecto puede ser: “El requerimiento que refleja la necesidad de imprimir un comprobante de ventas en una impresora térmica no ha sido agregado.”
Fecha de aprobación	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Definición de los Casos de Prueba

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Nombre del caso de prueba	[Inserte aquí el texto]
Propósito	[Inserte aquí el texto]
Pre-requerimientos	[Inserte aquí el texto]
Datos de Prueba	[Inserte aquí el texto]
Pasos	[Inserte aquí el texto]
Notas y Preguntas	[Inserte aquí el texto]

Instrucciones de llenado de la plantilla

Campo	Descripción
<i>Definición de pruebas</i>	
Fecha	Fecha en la que se realiza el plan de pruebas.
Hora de inicio	Hora exacta en la que comienza la redacción del plan de pruebas.
Hora de fin	Hora exacta en la que termina la redacción del plan de pruebas.
Identificación del requerimiento	Mediante su referencia deberán listarse todos los requerimientos que fueron aceptados en la Especificación de Requerimientos.
Tipo de requerimiento	De acuerdo a su especificación, se deberá indicar si es un requerimiento funcional, de interfaz, de rendimiento, de mantenimiento, etc.
Tipo de prueba	<p>Los tipos de pruebas que se podrán manejar son las siguientes:</p> <ul style="list-style-type: none"> • Pruebas de integridad sobre la base de datos e información. Las bases de datos y los procesos de acceso a la misma deben ser evaluados como un subsistema. Esta prueba debe probar los subsistemas independientemente de la interfaz de usuario. • Prueba funcional. Esta prueba se enfoca en verificar que el sistema cumpla con los requerimientos definidos, esta es una prueba de caja negra que se basa en los resultados obtenidos a través de la interfaz gráfica. • Prueba de interfaz de usuario. Verifica la interacción con el software, asegurando que el usuario cuenta con el acceso y navegación adecuada para desarrollar las funcionalidades del sistema. Además revisa que los objetos de la interfaz gráfica se comporten de manera adecuada y cumplan con los estándares establecidos por la empresa o la industria. • Pruebas de rendimiento. Se verifica el desempeño del sistema para cumplir con los requerimientos establecidos. Por ejemplo: Tiempo de Respuesta, Número de Transacciones procesadas por unidad de tiempo, etc. • Pruebas de carga. Se verifica la funcionalidad del sistema en diferentes situaciones de carga de trabajo, ya sea esperada o más allá del límite. Se verifica tiempo de respuesta, número de transacciones procesadas, etc., para picos de carga, carga alta sostenida, simulación de carga en un periodo de tiempo (ej. Cierre mensual, Comportamiento de las cargas en diferentes configuraciones de equipo, etc.). • Pruebas de estrés. Se verifica la funcionalidad del sistema bajo condiciones de recursos que no se presentan de manera normal, por ejemplo, poca memoria disponible, usuarios realizando la misma transacción a la vez, ancho de banda disponible. Algunas de estas pruebas pueden haberse considerado en pruebas de funcionalidad (si es un requerimiento de la aplicación) o pruebas de carga (al haber probado situaciones de carga fuera de los límites esperados, ej. Número de usuarios conectados más allá del límite esperado). • Pruebas de control de acceso y seguridad. Pruebas a nivel de acceso al sistema, a fin de verificar que éste es adecuado para los datos o funciones del negocio, según se requiera. Se requiere de pruebas de seguridad a nivel sistema que aseguren que solamente los usuarios autorizados tienen acceso al sistema y son capaces de acceder al sistema a través de los canales apropiados. • Pruebas de falla y recuperación. Pruebas que permiten asegurar que el sistema es capaz de recuperarse ante una falla de software o hardware con la consecuente pérdida de integridad en los datos. Es decir, se prueba que existan los mecanismos adecuados de respaldo y recuperación de información, o de autocorrección de transacciones que no fueron completadas, identificación de datos inconsistentes en la base de datos y su respectiva corrección. • Pruebas de configuración. Verifican que el sistema se comporta adecuadamente en

	<p>diferentes plataformas de hardware o software para las cuales fue realizada. Incluyendo uso de diferentes versiones del sistema operativo, marcas de browsers, etc.</p> <ul style="list-style-type: none"> • Pruebas de instalación. Verifican que la aplicación y sus actualizaciones sean instaladas adecuadamente.
Descripción de la prueba	Representación y vinculación con el caso de prueba que corresponda en la Definición de los Casos de Prueba.
Resultado esperado	Salida esperada del sistema y que corresponde con la funcionalidad reflejada en el requerimiento relacionado. Es necesario indicar la tolerancia a fallas o criterios para decidir si el sistema es satisfactorio en este tipo de pruebas.
Calendarización	Fecha propuesta para realizar la prueba, ésta debe ser consistente con las fechas establecidas en el Calendario del Plan de Desarrollo.
Recursos para establecer el entorno de pruebas	
Recurso	Cualquier tipo de recurso (infraestructura o equipo electrónico, de cómputo o tecnológico, documentos, manuales, etc.) que debe estar disponible para realizar las pruebas sin contratiempo. Por ejemplo: Servidor de Base de Datos, Nombre del Servidor, Nombre de la Base de Datos, Computadoras Personales, Impresoras, etc.
Nombre/Tipo/Descripción	Descripción textual del recurso y su justificación en relación con las pruebas.
Herramientas necesarias para ejecutar el plan de pruebas	
Herramienta	Definición de las herramientas que serán utilizadas para ejecutar las pruebas descritas en la Definición de Pruebas (ej. Diseño de Casos de Prueba, Registros de Defectos, Reporte de Defectos, Gráficas/Métricas de Defectos, Automatización de Pruebas, etc.). Para este caso se ha definido por default el uso de Casos de Pruebas, pero queda abierto a las herramientas usadas actualmente en la empresa.
Descripción	Descripción textual de la herramienta y su justificación en relación con las pruebas.
Recursos que el cliente debe proporcionar para realizar las pruebas	
Recurso	Lista de recursos necesarios para realizar las pruebas y que deben ser provistos por el cliente. Por ejemplo: el sistema asume que el cliente tiene un servidor propio para instalar y probar el sistema.
Información relevante	Justificación de la importancia del recurso en relación con la realización de las pruebas.
Fecha compromiso	Fecha en la que deben estar disponibles los recursos y cualquier información relevante con respecto a los mismos.

Definición de los casos de prueba	
Nombre del caso de prueba	Nombre único que le es asignado a cada caso de prueba. Es común que el nombre tenga relación con la funcionalidad que se desea probar.
Propósito	Una o dos oraciones cortas sobre el aspecto del sistema que está siendo probado. Si esto toma mucho tiempo, se recomienda interrumpir el caso de prueba o proporcionar más información en la descripción del caso de prueba.
Pre-requerimientos	Suposiciones que deben cumplirse antes de que correr el caso de prueba. Por ejemplo: "el usuario <i>admin</i> ya está registrado", "inicio de sesión como invitado permitido", etc.
Datos de Prueba	Lista de variables y sus posibles valores usados en el caso de prueba. Es posible enlistar valores específicos o describir rangos de valores. El caso de prueba deberá ser ejecutado una vez por cada combinación de valores. Estos valores se escriben con notación de asignación, uno por línea. Por ejemplo: <ul style="list-style-type: none"> • loginID = {loginID válido, loginID inválido, email válido, email inválido, vacío} • password = {válido, inválido, vacío}
Pasos	Pasos ordenados para ejecutar la prueba. Estos pasos deben respetar el formato establecido por ejemplo en: <ol style="list-style-type: none"> 1. visitar LoginPage 2. teclear usernameOrEmail

	<ol style="list-style-type: none">3. teclear password4. hacer click en Entrar5. ver: la página de los términos de uso6. hacer click hasta el fondo de la página7. hacer en click Aceptar8. ver: PersonalPage9. verificar el mensaje de bienvenida si el inicio de sesión es correcto
Notas y Preguntas	Cualquier información adicional que extienda la descripción del caso de prueba o bien alguna pregunta sobre la funcionalidad y que servirá para determinar algún error en el sistema.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RE y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RE y RAN que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. . Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Descripción de los Casos de Prueba, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas del Sistema y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Descripciones de recursos, Identificación de herramientas requeridas para la ejecución de las pruebas, etc.
Descripción del defecto encontrado	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todas las pruebas que serán realizadas”. Un ejemplo de cómo describir un defecto puede ser: “Los recursos no contemplan quién tendrá la responsabilidad de proporcionar el servidor para instalar el sistema y proceder con las pruebas.”
Fecha de aprobación	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
Lugar	Lugar en donde el RDM y el CL y US realizan la validación del documento.
Hora de inicio	Hora exacta en la que comienza la validación del documento.
Hora de fin	Hora exacta en la que termina la validación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RDM, CL y US que participan en la validación.
Puesto (rol)	Puestos o roles de los participantes en la validación.
Elemento a validar	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Recursos requeridos, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas de Sistema y aplicar los criterios definidos para su validación.
Descripción del defecto encontrado	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento cubre claramente los requerimientos establecidos por el cliente en relación al sistema”. Un ejemplo de cómo describir un defecto puede ser: “No se ha incluido un caso de prueba que incluya la generación de cortes de caja.”
Fecha de aprobación	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

D.11. Formato para Manual de Usuario (DMS A.2.20)

1. Datos de control

Estos datos deben ser llenados por el RM al momento de empezar y terminar con la generación del Manual de Usuario. Así, esta información sirve únicamente para que el RDM controle el proyecto, una vez que este manual esté listo para entregarse al CL deberán ser borrados del mismo.

Preparado por	[Inserte aquí el texto]	Fecha	[Fecha]
Hora de inicio	[Rev]	Hora de fin	[Rev]

2. Derechos de autor de la Empresa [Inserte aquí el texto]

El CL es libre de:

Copiar, distribuir y comunicar públicamente este Manual de Usuario.

Realizar obras derivadas del mismo bajo las condiciones siguientes:

- Reconocimiento: Debe reconocer y citar a la Empresa [Inserte aquí el texto] como autor original.
- Fines no comerciales: No puede utilizar este Manual de Usuario para fines comerciales.
- Compartir bajo la misma licencia: Si altera o transforma este Manual de Usuario, o genera una obra derivada de éste, solamente puede distribuir la obra generada bajo una licencia de la Empresa [Inserte aquí el texto].

3. Objetivo del manual

El objetivo primordial de este Manual de Usuario es ayudar y guiar al usuario en la instalación y uso del Sistema [Inserte aquí el texto] a través de información resumida que le permitirá despejar todas las dudas existentes y facilitar su entendimiento sobre el mismo.

4. Descripción general del sistema

[Inserte aquí el texto]

Este apartado debe proporcionar a la audiencia una introducción y visión general del sistema para el cual se está generando el Manual de Usuario.

5. Audiencia

Es necesario definir al personal específico que se dirige este Manual de Usuario. Sería recomendable que esta misma audiencia tenga los roles del personal involucrado en la obtención de los requerimientos.

Descripción de rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]

Descripción de rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]

Descripción de rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]

6. Requerimientos mínimos para usar el manual

[Inserte aquí el texto]

Se deberán listar los requerimientos relacionados con el conocimiento previo que debe tener la audiencia de este Manual de Usuario. Por ejemplo:

- *Conocimientos básicos sobre programas utilitarios.*
- *Conocimientos básicos de Navegación Web.*
- *Conocimientos básicos de Internet.*
- *Conocimientos básicos de Windows.*



7. Convenciones y estándares

Es necesario especificarle a la audiencia las convenciones de formato que encontrará en el texto del Manual de Usuario, por ejemplo:

Formato	Tipo de información
Viñeta numérica	Numeración de procedimientos paso a paso
Negrita	Términos de importancia que requieren especial atención.
Menor que (<) Mayor que (>)	Nombre de opciones que el usuario puede elegir ya sea haciendo uso del teclado o el mouse.
! (signo de admiración)	Consejo en caso de error.

Se deberá especificar también la convención para el uso del mouse y del teclado, por ejemplo:

Término	Significado
Señalar	Colocar el extremo superior del mouse sobre el elemento que se desea señalar.
Hacer clic.	Presionar el botón principal del mouse (generalmente el botón izquierdo) y soltarlo lentamente.
Desplegar el menú contextual	Presionar el botón secundario del mouse (generalmente el botón derecho) y soltarlo lentamente.

Término	Significado
Teclas de método abreviado 	Teclas que se utilizan como método abreviado para ejecutar un proceso. Por ejemplo: F1 para mostrar la ayuda.
Tecla Enter 	Tecla utilizada para ejecutar un proceso cuando el manual indique “Presionar Enter”.

8. Actividades de instalación

[Inserte aquí el texto]

Utilizando las convenciones y estándares definidos en el apartado anterior, se enlistan los pasos para realizar la correcta instalación del sistema. El RM deberá agregar imágenes que le sirvan de soporte para describir los pasos y por consecuencia mejorar el entendimiento del CL y US.

9. Descripción de usos y funcionalidades

[Inserte aquí el texto]

Utilizando las convenciones y estándares definidos en el apartado anterior, se enlistan los pasos para realizar las funcionalidades generales del sistema. El RM deberá agregar imágenes que le sirvan de soporte para describir los pasos y por consecuencia mejorar el entendimiento del CL y US. En este caso, el RM deberá apoyarse en el RAN para establecer una secuencia de presentación que resuma las funcionalidades del sistema.

10. Preguntas frecuentes

[Inserte aquí el texto]

Es recomendable agregar una sección de preguntas y respuestas que ayuden al CL o US a encontrar soluciones rápidas a problemas sencillos.

11. Glosario de términos y acrónimos

[Inserte aquí el texto]

Para un mejor entendimiento se recogen todos aquellos términos del Manual de Usuario que pueden confundir a la audiencia y afectar el entendimiento del sistema.

Término	Acrónimo	Descripción
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RM y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RM y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RM y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RM y RAN que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. . Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos. Se recomienda incluir todos los elementos que forman el documento de Manual de Usuario y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Descripciones de pasos de instalación del sistema, Uso de convenciones y estándares definidos para describir los usos o funcionalidades, etc.
Descripción del defecto encontrado	El RM y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los pasos que el US debe realizar”. Un ejemplo de cómo describir un defecto puede ser: “El manual no describe correctamente cómo realizar el alta de un empleado.”
Fecha de aprobación	Fecha en la que RM y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RDM y el RAPE actualizan la Configuración de Software. Es importante considerar que cada actualización realizada genera un cambio de fecha si es necesario.
Proyecto	Referencia al proyecto para el cual se han generado todos los productos específicos, ya sea por ciclo o en su totalidad.
Hora de inicio	Hora exacta en la que comienza la realización del documento.
Hora de fin	Hora exacta en la que termina la realización del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RDM y RAPE tardaron en realizar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como RAPE hayan tenido interrupciones realizando la actividad y esta relación no sea del todo cierta.
Nombre completo	Nombres del RDM y RAPE que participan en la documentación de los productos.
Producto	Todo aquel producto generado durante el ciclo/proyecto y que debe ser identificado y consultado fácilmente. La lista de productos ya está definida por default.
Ubicación del producto	Ubicación de los productos dentro de la estructura de carpetas de la Base de Conocimiento (servidor) que la empresa haya generado para actualizar toda la información relacionada con el proyecto.
Última versión activa	Última versión generada mediante el control de versiones de esta plantilla y que evitará utilizar documentos duplicados o con información incorrecta.

Instrucciones de llenado de la plantilla

Campo	Descripción
Datos generales	
Nombre del proyecto	Nombre que le fue asignado al proyecto en el Plan de Desarrollo.
Referencia del proyecto	Referencia única que se le da a cada proyecto para referirse a él de forma abreviada. Estas referencias suelen tener identificadores que ayudan al personal a determinar la naturaleza del proyecto sin consultar el nombre o descripción completa del mismo. El estilo de las referencias debe ser definido por la empresa y debe ser respetado por todo el Equipo de Trabajo.
Nombre del RAN	Nombre del Responsable de Analistas que fue asignado al proyecto en cuestión.
Nombre del RPR	Nombre del Responsable de Programadores que fue asignado al proyecto en cuestión.
Registro de rastreo	
#	Identificador numérico único para cada requerimiento a rastrear.
Requerimiento	Nombre dado al requerimiento en la Especificación de Requerimientos (ej. RF 1.13).
Origen del requerimiento	Proveedor del requerimiento que está siendo rastreado (ej. Gerente de Ventas, Dueño, Secretaría).
Referencias	Relaciones que tiene el requerimiento con otros requerimientos (ej. RF 1.1, RF 1.2).
Caso de uso	Nombre del caso de uso que incluye al requerimiento y que ha sido obtenido en el Análisis y Diseño.
Especificación de diseño	Diagrama de caso de uso que incluye al requerimiento en cuestión.
Programado	Funcionalidad que es descrita por el requerimiento y pantalla en la que cual se implementa.
Tabla de la Base de Datos	Nombre de la tabla de la BD donde se ve reflejada la implementación del requerimiento.
Caso de prueba	Nombre del caso de prueba con el que se relaciona el requerimiento. Este nombre se obtiene del Plan de Pruebas de Sistema.
Estado	Situación actual del requerimiento (<i>Activo</i> para decir que está en implementación, <i>Terminado</i> para indicar que se ha terminado su implementación, <i>Pendiente</i> para decir que se está realizando un análisis más exhaustivo sobre él, y <i>Eliminado</i> para indicar que el requerimiento ha sido borrado de la especificación).
Observaciones	Cualquier información adicional que se desee incorporar al registro de rastreo.

D.15. Formato para Análisis y Diseño (DMS A.3.3)

1. Introducción

Los requerimientos documentados en la Especificación de Requerimientos son analizados meticulosamente para producir una descripción detallada de la estructura de los componentes de software, la cual servirá de base para la construcción del sistema.

2. Análisis y Diseño

Este documento contiene la descripción textual y gráfica de la estructura de los componentes de software divididos en dos partes: Arquitectónica (Diseño de alto nivel) y Detallada (Diseño de bajo nivel).

2.1. Estructura arquitectónica

[Inserte aquí el texto]

La estructura arquitectónica comienza por describir con mayor detalle los casos de uso que representen las funcionalidades descritas en la Especificación de Requerimientos a través del Diagrama de Contexto de Casos de Uso. Por ejemplo, el diagrama ejemplo de la Especificación de Requerimientos debería descomponerse en funcionalidades más específicas, una de ellas para el caso de uso Administrar proyectos quedaría como:

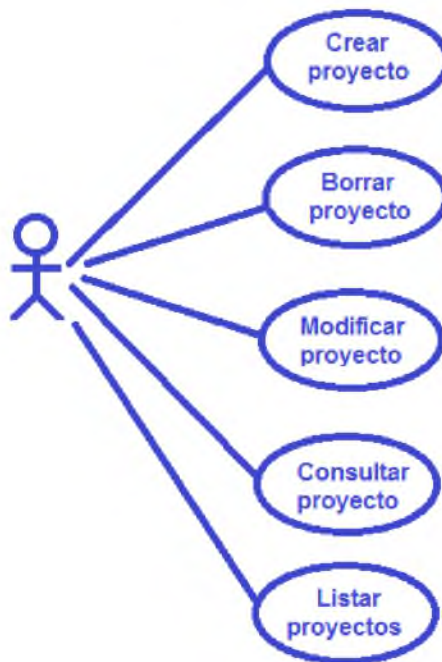


Fig 1. Ejemplo de casos de uso para Administrar proyectos

Utilizando estos diagramas como punto de partida, los AN deberán detallar cada caso de uso de la siguiente forma.

Caso de uso	<i>Crear proyecto</i>	
Actores	<i>Administrador</i>	
Propósito	<i>Dar de alta o crear un nuevo proyecto o grupo de proyectos</i>	
Visión general	<i>El Administrador crea un proyecto o grupo de proyectos de acuerdo con un formato establecido</i>	
Referencias	<i>RF 1.1, RF 1.2</i>	
Curso típico de eventos		
	<i>Administrador</i>	<i>Sistema</i>
	<i>1. El Usuario accede a la pantalla de alta de proyecto desde el menú principal del sistema.</i>	<i>2. El sistema muestra una lista de proyectos existentes (si los hubiera).</i>
	<i>3. El Administrador rellena los datos de identificación del proyecto seleccionando el número de proyectos a insertar, los ciclos y el patrón que usará.</i>	<i>4. El sistema comprueba que los datos son correctos, los almacena en la BD y vuelve a mostrar la lista con todos los proyectos creados.</i>
Cursos alternativos		

Caso de uso	[Inserte aquí el texto]	
Actores	[Inserte aquí el texto]	
Propósito	[Inserte aquí el texto]	
Visión general	[Inserte aquí el texto]	
Referencias	[Inserte aquí el texto]	
Curso típico de eventos		
	[Inserte aquí el texto]	[Inserte aquí el texto]
	[Inserte aquí el texto]	[Inserte aquí el texto]
Cursos alternativos		
[Inserte aquí el texto]		

Caso de uso	[Inserte aquí el texto]	
Actores	[Inserte aquí el texto]	
Propósito	[Inserte aquí el texto]	
Visión general	[Inserte aquí el texto]	
Referencias	[Inserte aquí el texto]	
Curso típico de eventos		
	[Inserte aquí el texto]	[Inserte aquí el texto]
	[Inserte aquí el texto]	[Inserte aquí el texto]
Cursos alternativos		
[Inserte aquí el texto]		

2.2. Arquitectura del software

[Inserte aquí el texto]

La arquitectura de software representa el enlace entre la Especificación de Requerimientos y el diseño en sí. A efectos de agilizar el ciclo de desarrollo, esta tarea se puede realizar en paralelo con las actividades para la Especificación de Requerimientos; sin embargo cuando la empresa carece de madurez en el uso de estas prácticas lo mejor es ir paso a paso. El diseño de la arquitectura del software implica un esfuerzo creativo por parte de los AN, de forma que las actividades a realizar pueden cambiar de acuerdo a la naturaleza del sistema a desarrollar.

2.3. Descomposición en subsistemas

La descomposición modular o con subsistemas proporciona pequeños componentes que proveen servicios a otros componentes y que no se consideran un sistema separado. Es decir, con base al entendimiento de los requerimientos funcionales el sistema se descompone en objetos que interactúan para realizar una tarea específica. El paso inicial consiste en obtener un diagrama de bloques que resuma la estructura del sistema en términos de los componentes que lo forman. Por ejemplo:

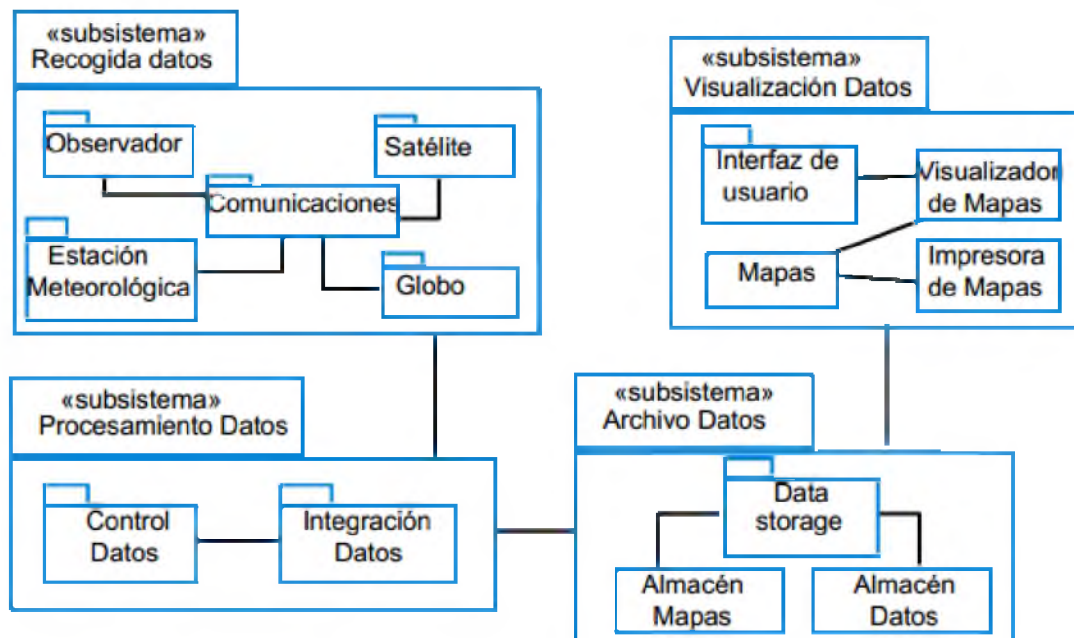


Fig. 1. Ejemplo de diagrama de bloques

Nombre del subsistema	Recogida de datos
Descripción	Este subsistema se encargará de recibir la información a través de una estación meteorológica, un observador, el satélite, el globo lanzado a la atmósfera o bien a través de un protocolo de comunicaciones.

Nombre del subsistema	[Inserte aquí el texto]
Descripción	[Inserte aquí el texto]

Nombre del subsistema	[Inserte aquí el texto]
Descripción	[Inserte aquí el texto]

2.4. Arquitectura por capas

Descomposición del diagrama de bloques en entidades que proporcionan información concisa sobre cada subsistema. Se recomienda no tener más de siete entidades en un modelo arquitectónico. Por ejemplo:

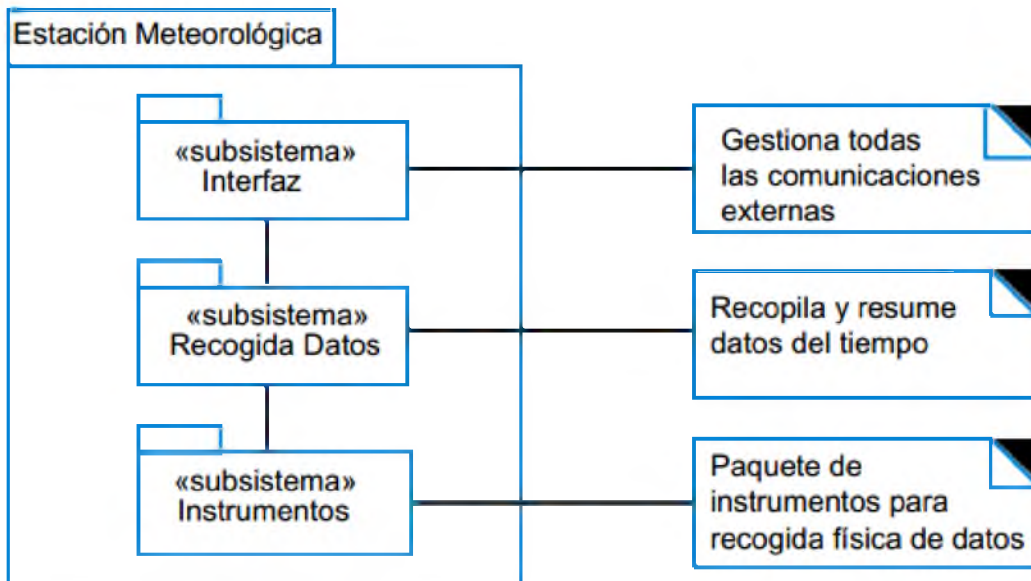


Fig. 2. Ejemplo de descomposición del diagrama de bloques

2.5. Representación estructural o estática del software

[Inserte aquí el texto]

Agregar el diagrama de clases y objetos para representar el conjunto de componentes (objetos) y sus interrelaciones. Este modelo de objetos mostrará los componentes o subsistemas principales del software.

2.5.1. Identificación de objetos

[Inserte aquí el texto]

La representación estructural inicia con la identificación de objetos para posteriormente relacionarlos en un modelo estático. Por ejemplo:

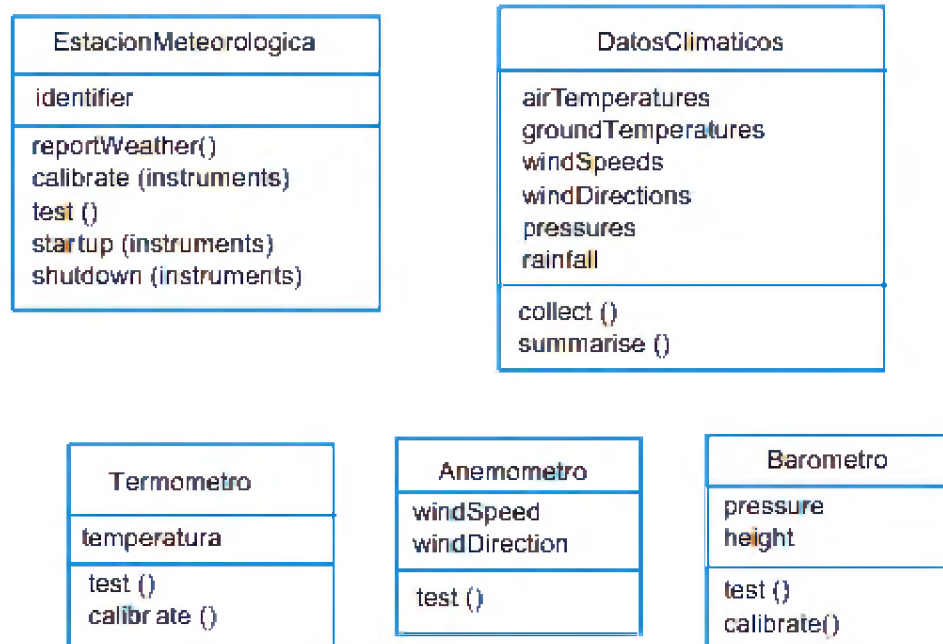


Fig. 3. Ejemplo de objetos identificados

2.5.2. Modelo estático

El modelo estático empezará a sustituir los subsistemas por objetos que mejoren la comprensión para su implementación.

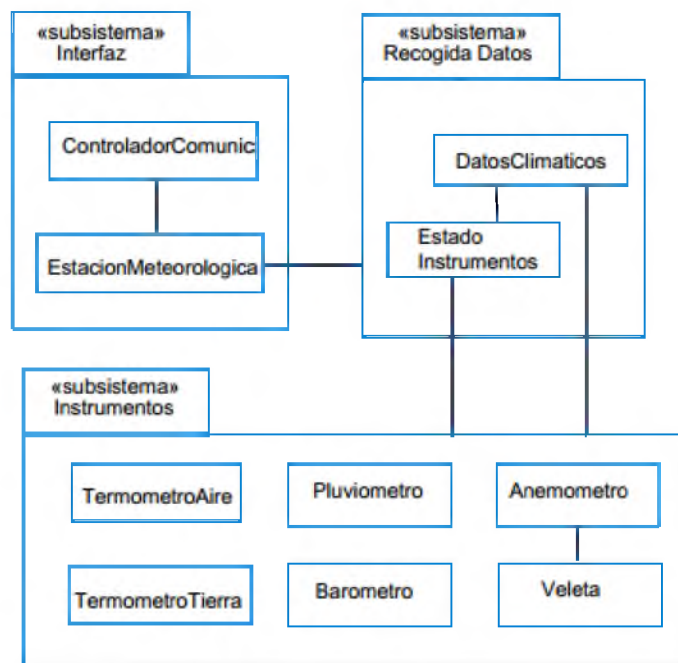


Fig. 4. Ejemplo de modelo estático

2.6. Representación dinámica del software

[Inserte aquí el texto]

Con base a la arquitectura que se está estableciendo, los AN comenzarán a diseñar modelos dinámicos de diseño con base a los casos de uso descritos en 2.1. Se recomienda obtener de entre 3 a no más de 36 diagramas de secuencia por caso de uso. Por ejemplo:

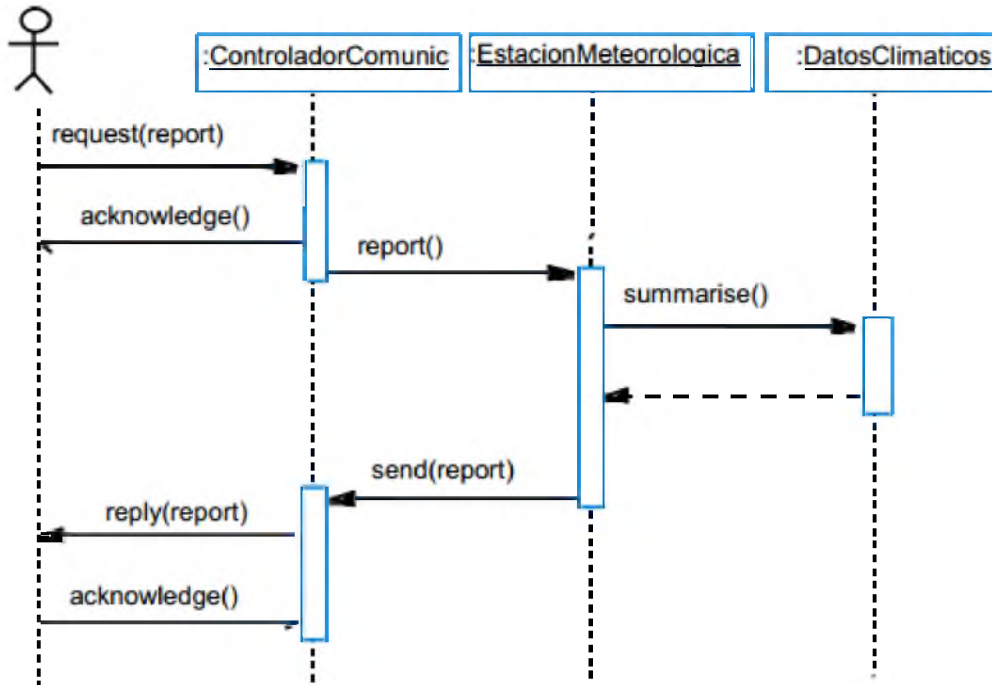


Fig. 5. Ejemplo de modelo dinámico con diagrama de secuencias

2.7. Representación por interfaces

La representación por interfaces define con claridad las decisiones de los AN para facilitar la construcción del sistema. Por ejemplo, una interfaz para la EstacionMeteorologica podria ser:

```

interface WeatherStation {
public void WeatherStation ();
public void startup ();
public void startup (Instrument i);
public void shutdown ();
public void shutdown (Instrument i);
public void reportWeather ();
public void test ();
public void test ( Instrument i );
public void calibrate ( Instrument i);
public int getID ();
} //WeatherStation.
  
```

2.8. Modelo de datos (E-R)

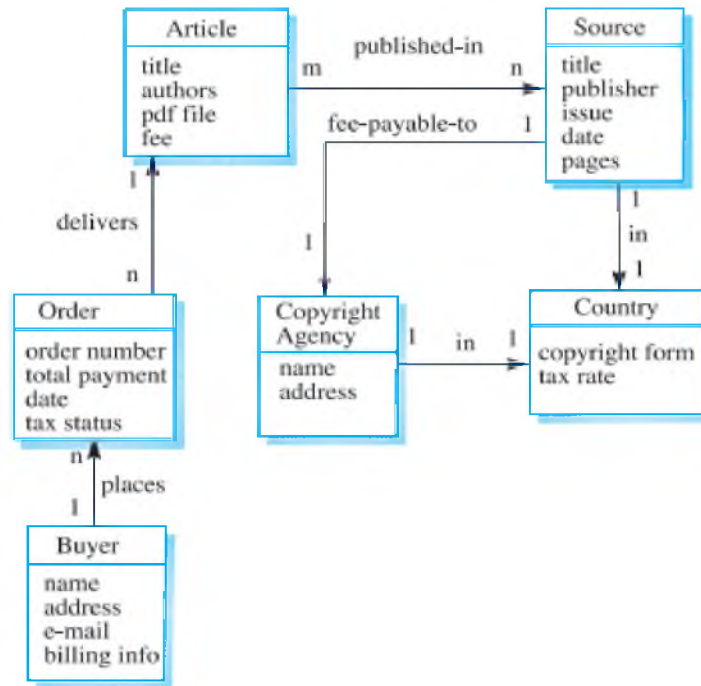


Fig. 6. Ejemplo de modelo entidad-relación

2.8.1. Diseño de tablas y atributos

La siguiente descripción presenta en orden alfabético las diversas tablas que componen al modelo completo de datos (E-R). Para cada tabla se debe hacer la descripción de los campos, así como las claves propias y ajenas que la componen. En algunos casos, se muestra una representación de la tabla y se incluye el código SQL que la crea en la BD. Cada tabla se describe de la siguiente manera.

Nombre de la tabla		calendario			
Campo	Tipo	Nulo	PK	FK	Descripción
IdCiclo	Integer	No	Si	Si	Número de ciclo dentro del proyecto
IdProyecto	Integer	No	Si	Si	Proyecto al que se asocia el calendario
IdSemana	Integer	No	Si	No	Identificativo de la sema del proyecto
Fecha	Date	No	No	No	Fecha que representa el lunes de cada semana
Horas	Integer	No	No	No	Horas reales en la semana

Nombre de la tabla		[Inserte aquí el texto]			
Campo	Tipo	Nulo	PK	FK	Descripción
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]

Nombre de la tabla		[Inserte aquí el texto]			
Campo	Tipo	Nulo	PK	FK	Descripción
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]

3. Apéndices

[Inserte aquí el texto]

Este apartado es opcional y puede contener todo tipo de información relevante para el Análisis y Diseño pero que, propiamente, no forme parte de la información requerida.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RE y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RE y RAN que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Descripción detallada de los Casos de Uso, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas del Sistema y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Modelo estructural, Modelo dinámico, etc.
Descripción del defecto encontrado	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todas las decisiones de diseño”. Un ejemplo de cómo describir un defecto puede ser: “El diagrama de secuencia no contempla todos los casos de uso descritos.”
Fecha de aprobación	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
Lugar	Lugar en donde el RE y el RAN realizan la verificación del documento.
Hora de inicio	Hora exacta en la que comienza la verificación del documento.
Hora de fin	Hora exacta en la que termina la verificación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RE y RAN que participan en la verificación.
Puesto (rol)	Puestos o roles de los participantes en la verificación.
Elemento a verificar	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Incorporación de todos los requerimientos funcionales, etc. Se recomienda incluir todos los elementos que forman el documento del Registro de Rastreo y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Incorporación de los elementos de diseño, etc.
Descripción del defecto encontrado	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente la trazabilidad de cada requerimiento funcional”. Un ejemplo de cómo describir un defecto puede ser: “El RF 19.5 no está siendo rastreado.”
Fecha de aprobación	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

Instrucciones de llenado de la plantilla

Campo	Descripción
Fecha del reporte	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
Lugar	Lugar en donde el RDM y el CL y US realizan la validación del documento.
Hora de inicio	Hora exacta en la que comienza la validación del documento.
Hora de fin	Hora exacta en la que termina la validación del documento.
Duración total	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
Nombre completo	Nombres del RDM, CL y US que participan en la validación.
Puesto (rol)	Puestos o roles de los participantes en la validación.
Elemento a validar	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Recursos requeridos, etc. Se recomienda incluir todos los elementos que forman el documento del Análisis y Diseño y aplicar los criterios definidos para su validación.
Descripción del defecto encontrado	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento cubre claramente los requerimientos establecidos por el cliente en relación al sistema”. Un ejemplo de cómo describir un defecto puede ser: “No se ha incluido un diagrama de secuencia que represente la cancelación de una venta.”
Fecha de aprobación	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

D.19. Formato para Plan de Pruebas de Integración (DMS A.3.13)

Fecha:	[Fecha]	Hora de inicio:	[Rev]	Hora de fin:	[Rev]
--------	---------	-----------------	-------	--------------	-------

Nombre del subsistema		[Inserte aquí el texto]			Versión:	[Rev]
<i>Componentes y subsistemas que lo integran</i>						
Componente 1	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]	
Componente 2	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]	
Componente 3	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]	
Componente n	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]	
<i>Requerimientos funcionales cubiertos por el subsistema</i>						
Referencia		Descripción				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
[Rev]		[Inserte aquí el texto]				
<i>Casos de prueba</i>						
Escenario relacionado		[Rev]				
# de caso de prueba		[Rev]	Descripción:	[Inserte aquí el texto]		
Procedimiento de prueba		[Inserte aquí el texto]				
Entrada		[Inserte aquí el texto]				
Salida esperada		[Inserte aquí el texto]				
Errores encontrados		[Inserte aquí el texto]				
Sugerencias de corrección		[Inserte aquí el texto]				
# de caso de prueba		[Rev]	Descripción:	[Inserte aquí el texto]		
Procedimiento de prueba		[Inserte aquí el texto]				
Entrada		[Inserte aquí el texto]				
Salida esperada:		[Inserte aquí el texto]				
Errores encontrados:		[Inserte aquí el texto]				
Sugerencias de corrección:		[Inserte aquí el texto]				
# de caso de prueba:		[Rev]	Descripción:	[Inserte aquí el texto]		
Procedimiento de prueba		[Inserte aquí el texto]				
Entrada:		[Inserte aquí el texto]				
Salida esperada:		[Inserte aquí el texto]				
Errores encontrados:		[Inserte aquí el texto]				
Sugerencias de corrección:		[Inserte aquí el texto]				
Escenario relacionado:		[Rev]				
# de caso de prueba:		[Rev]	Descripción:	[Inserte aquí el texto]		
Procedimiento de prueba		[Inserte aquí el texto]				
Entrada:		[Inserte aquí el texto]				

Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		

<i>Cobertura de requisitos funcionales</i>					
Escenario con datos	Entrada	Salida esperada	Salida obtenida	Error	Caso de prueba
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]

Nombre del subsistema	[Inserte aquí el texto]			Versión:	[Rev]
<i>Componentes y subsistemas que lo integran</i>					
Componente 1	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]
Componente 2	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]
Componente 3	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]
Componente n	[Inserte aquí el texto]	Versión	[Rev]	Implementador	[Inserte aquí el texto]
<i>Requerimientos funcionales cubiertos por el subsistema</i>					
Referencia	Descripción				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				
[Rev]	[Inserte aquí el texto]				

<i>Casos de prueba</i>			
Escenario relacionado	[Rev]		
# de caso de prueba	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida esperada	[Inserte aquí el texto]		
Errores encontrados	[Inserte aquí el texto]		
Sugerencias de corrección	[Inserte aquí el texto]		
# de caso de prueba	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
Escenario relacionado:	[Rev]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		
# de caso de prueba:	[Rev]	Descripción:	[Inserte aquí el texto]
Procedimiento de prueba	[Inserte aquí el texto]		
Entrada:	[Inserte aquí el texto]		
Salida esperada:	[Inserte aquí el texto]		
Errores encontrados:	[Inserte aquí el texto]		
Sugerencias de corrección:	[Inserte aquí el texto]		

<i>Cobertura de requisitos funcionales</i>					
Escenario con datos	Entrada	Salida esperada	Salida obtenida	Error	Caso de prueba
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]

[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]

Interacción de la Integración

Subsistemas involucrados	[Inserte aquí el texto]
Entrada	[Inserte aquí el texto]
Resultado esperado	[Inserte aquí el texto]
Resultado obtenido	[Inserte aquí el texto]
Errores encontrados	[Inserte aquí el texto]
Sugerencias de corrección	[Inserte aquí el texto]

Subsistemas involucrados	[Inserte aquí el texto]
Entrada	[Inserte aquí el texto]
Resultado esperado	[Inserte aquí el texto]
Resultado obtenido	[Inserte aquí el texto]
Errores encontrados	[Inserte aquí el texto]
Sugerencias de corrección	[Inserte aquí el texto]

Subsistemas involucrados	[Inserte aquí el texto]
Entrada	[Inserte aquí el texto]
Resultado esperado	[Inserte aquí el texto]
Resultado obtenido	[Inserte aquí el texto]
Errores encontrados	[Inserte aquí el texto]
Sugerencias de corrección	[Inserte aquí el texto]

Evaluación

Evaluación final	[Inserte aquí el texto]
-------------------------	-------------------------

Instrucciones de llenado de la plantilla

Campo	Descripción
<i>Descripción de pruebas de integración</i>	
Fecha	Fecha en la que se redacta/ejecuta el plan de pruebas.
Hora de inicio	Hora exacta en la que comienza la redacción/ejecución del plan de pruebas.
Hora de fin	Hora exacta en la que termina la redacción/ejecución del plan de pruebas.
Nombre del subsistema	Mediante su referencia deberán listarse todos los requerimientos que fueron aceptados en la <i>Especificación de Requerimientos</i> y que se relacionan con el subsistema descrito.
<i>Componentes y subsistemas que lo integran</i>	
Componente	Cada uno de los componentes que integran al subsistema de acuerdo a lo establecido en el <i>Análisis y Diseño</i> de la iteración (ciclo).
Versión	Indicador numérico para llevar el control de versiones cuando se empiece a modificar el componente, si fuera necesario.
Implementador	Nombre del programador que fue asignado para implementar el componente.
<i>Requerimientos funcionales cubiertos por el subsistema</i>	
Referencia	Se hace referencia a las funcionalidades identificadas en la <i>Especificación de Requerimientos</i> y que fueran relacionadas con los subsistemas en el <i>Análisis y Diseño</i> . Concretamente, se debe colocar la referencia que se le asignó al requerimiento desde la <i>Especificación de Requerimientos</i> .
Descripción	Descripción del requerimiento.
<i>Casos de prueba</i>	
Escenario relacionado	Se refiere a los escenarios identificados con los diagramas de secuencia que mostraron la relación de los subsistemas definidos en el <i>Análisis y Diseño</i> .
# de caso de prueba	Número que se le asigna a cada caso de prueba (p.ej., 1, 2, 3, 4, ..., n).
Procedimiento de prueba	Descripción de los procedimientos de prueba para la realizar la funcionalidad de cada # de caso de prueba.
Entrada	Se describen los datos de prueba (condiciones de entrada) necesarios para realizar la prueba.
Salida esperada	Se describe la salida esperada para las condiciones de entrada especificadas en la entrada.
Salida obtenida	El RPR describe la salida real obtenida para cada valor de entrada especificado anteriormente.
Errores encontrados	El RPR describe los errores encontrados, si es posible la ubicación de cada uno y su gravedad e impacto en el sistema.
Sugerencias de corrección	El RPR proporciona sugerencias de corrección para los errores encontrados, si esto es posible.
<i>Cobertura de requisitos funcionales</i>	
Escenario con datos	Se hace referencia a cada # de caso de prueba planteado anteriormente, indicando la funcionalidad a probar y detallando los casos de prueba con datos asociados.
Entrada	Datos reales que debe recibir el sistema (p.ej., CONTRASEÑA: inválida, Nombre: Juan Pérez, Fecha de préstamo: 45/89/2130, CONTRASEÑA: válida, etc.).
Salida esperada	Respuesta esperada del sistema de acuerdo a la entrada recibida (p.ej., Entrada->CONTRASEÑA: inválida, Salida->El sistema envía mensaje de error y solicita de nuevo la autenticación).
Salida obtenida	El RPR describe la respuesta real de acuerdo a la entrada recibida (p.ej., Entrada->CONTRASEÑA: inválida, Salida->El sistema no hace nada).

Error	El RPR describe a detalle los errores encontrados, si fuera el caso.
Caso de prueba	Referencia al # de caso de prueba con el que se relaciona la prueba sobre los requisitos funcionales.
<i>Interacción de la integración</i>	
Subsistemas involucrados	Se debe representar en la matriz de pruebas la relación uno-a-uno entre los diferentes subsistemas que componen la arquitectura presentada en <i>Análisis y Diseño</i> .
Entrada	Se describen los valores utilizados para realizar la interacción entre los subsistemas.
Resultado esperado	Se describe el resultado esperado para cada valor de entrada especificado anteriormente.
Resultado obtenido	El RPR describe el resultado real obtenido para cada valor de entrada especificado anteriormente.
Errores encontrados	El RPR describe los errores encontrados, si es posible la ubicación de cada uno y su gravedad e impacto en el sistema.
Sugerencias de corrección	El RPR brinda sugerencias de corrección para los errores encontrados, si esto es posible
<i>Evaluación</i>	
Evaluación final	El RPR evalúa en forma global el comportamiento de los subsistemas y establece el estado del mismo utilizando: no aprobado (cuando las pruebas a los subsistemas no son satisfechas en su totalidad) o aprobado (cuando las pruebas a los subsistemas son satisfechas en su totalidad).

6. Bibliografía

Baddoo, N. & Hall, T. (2003). “De-motivators for software process improvement: an analysis of practitioners’ views” *Journal of Systems and Software*, 66(1): 23-33.

Baetjer, H. (1998). *Software as capital: an economic perspective on software*. 1st Edition, Los Alamitos, CA: IEEE Computer Society.

Basili, V. R., Selby, R. W., & Hutchens, D. H. (1986). “Experimentation in software engineering” *IEEE Transactions on Software Engineering*, 12(7): 733-743.

Bernroider, E. W. N. & Ivanov, M. (2011). “IT project management control and the Control Objectives for IT and related Technology (CobiT) framework” *International Journal of Project Management*, 29(3): 325-336.

Caballero, D. (2005). “Manejador de Documentos MoProSoft®” Tesis para obtener el título de Licenciatura en Ingeniería en sistemas Computacionales. Universidad de las Américas Puebla (UDLA). Puebla, México. 2005.

CMMI Product Team. (2002). *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/PPD/SS, V1.1). Continuous Representation*. CMU/SEI-2002-TR-011, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

CMMI Product Team. (2010). *Capability Maturity Model Integration for Development (CMMI-DEV) version 1.3*, CMU/SEI-2010-TR-033, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Creswell, J. W. (2008). *Research design, qualitative and quantitative approaches*. 3rd Edition, Thousand Oaks, CA: Sage Publications.

Cuevas, G., De Amescua Seco, A., San Feliu, T., Calvo-Manzano, J. A., Arcilla, M., García, M. & Cerrada, J. A. *Gestión del proceso software*. 1^a Edición. Centro de Estudios Ramón Areces, S. A: Madrid, España. 2002.

Dyba, T. (2005). “An empirical investigation of the key factors for success in software process improvement” *IEEE Transactions on Software Engineering*, 31(5): 410-413.

Dunaway, D. K. & Masters, S. (1996). *CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description*, Technical Report (CMU/SEI-96-TR-007), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Flores, S. (2012), "Modelo de Calidad para la Microempresa Basado en MoProSoft®" Tesis para obtener el título de Maestro en Computación. Universidad de Colima. Colima, México. 2012.

Garcia, I., Calvo-Manzano, J. A., Cuevas, G., & San Feliu, T. (2007). "Determining practice achievement in project management using a two-phase questionnaire on small and medium enterprises" In: P. Abrahamsson, N. Baddoo, T. Margaria, and R. Messnarz (Editors), *Software Process Improvement*, Springer-Verlag: Heidelberg, Germany, pp. 46-58.

Garcia, I., Pacheco, C. & Calvo-Manzano, J. A. (2010). "Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting" *IET Software*, 4(4): 237-251.

Guerrero, F. & Eterovic, Y. (2004). "Adopting the SW-CMM in a small IT organization" *IEEE Software*, 21(4): 29-35.

Habra, N., Alexandre, S., Desharnais, J. M., Laporte, C. Y., & Renault, A. (2008). "Initiating software process improvement in very small enterprises: Experience with a light assessment tool" *Information and Software Technology*, 50(7-8): 763-771.

Hochstein, A., Zarnekow, R., & Brenner, W. (2005). "ITIL as common practice reference model for IT service management: formal assessment and implications for practice" *Proc. of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, IEEE Computer Society, pp. 704-710.

Humphrey, W. S. (2008). "The software quality challenge" *CrossTalk: The Journal of Defense Software Engineering*, 21(6): 4-9.

IEEE. (2004). Guide to the Software Engineering Body of Knowledge. New York. Disponible en: <http://www.swebok.org>.

Instituto Nacional de Tecnologías de Comunicación. (2009). *Ingeniería del software: metodologías y ciclos de vida*. 1ª Edición, Madrid, España: Laboratorio Nacional de Calidad del Software – INTECO.

ISO. (2005). "ISO 9000: Quality management systems – Fundamentals and vocabulary" 3rd Edition. Stage: 60.60. TC 176/SC 1.

ISSO/IEC. (1998). ISO/IEC TR 15504:1998(E): 'Information technology –software process assessments. Parts 1–9' (International Organization for Standardization, Geneva).

ISO/IEC. (2008). ISO/IEC TR 15504-7:2008. Information technology - Process assessment - Part 7. Assessment of organizational maturity. International Organization for Standardization. Available from www.iso.org.

Juristo, N. & Moreno, A. (2013). Basics of software engineering experimentation. Dordrecht, The Netherlands: Kluwer Academic Press.

Juristo, N. & Gómez, O. (2012). "Replication of software engineering" B. Meyer and M. Nordio (Editors), *Empirical Software Engineering and Verification*, Berlin, Heidelberg, Germany: Springer-Verlag, pp. 60-88.

Kautz, K., Hansen, H. W., & Thaysen, K. (2000). "Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise" *Proc. of the 22nd International Conference on Software engineering (ICSE '00)*, ACM, pp. 626-633.

Lester, N. G., Wilkie, F. G., McFall, D., & Ware, M. P. (2010). "Investigating the role of CMMI with expanding company size for small- to medium-sized enterprises" *Journal of Software: Evolution and Process*, 22(1): 17-31.

Marciniak, J. & Sadauskas, T. (2003). "Use of Questionnaire-Based Appraisals in Process Improvement Programs" *Proc. of the Second Annual Conference on the Acquisition of Software-Intensive Systems*. Carnegie-Mellon University. Software Engineering Institute. Arlington, Virginia. USA.

McFeeley, B. (1996). *IDEAL: A user's guide for software process improvement*, (CMU/SEI-96-HB-001), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Members of the Assessment Method Integrated Team. (2006). *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI)*, version 1.2, CMU/SEI-2006-HB-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Mochi, A. (2004). "La industria del software en México. Problemas de desarrollo" *Revista latinoamericana de Economía*, 35(137): 42-58.

Mochi, A., Prudencio, O., & Hualde, A. (2009). "México: Producción interna e integración mundial" In: *Desafíos y Oportunidades de la Industria del Software en América Latina*, pp. 171-203, Colombia: ONU-CEPAL.

Niazi, M., Wilson, D. & Zowghi, D. (2006). "Critical success factors for software process improvement implementation: an empirical study" *Software Process: Improvement and Practice*, 11(2): 193-211.

Niazi, M., Babar, M. A., & Verner, J. M. (2010). "Software Process Improvement barriers: A cross-cultural comparison" *Information and Software Technology*, 52(11): 1204-1216.

Nielsen, J. (1999). *Designing web usability: The practice of simplicity*, 1^a Edition. Indianapolis, IN: New Riders Publishing.

Normalización y Certificación. (2005a). NMX-I-059/04-NYCE-2005. Tecnología de la Información -Software- Modelos de procesos y evaluación para desarrollo y mantenimiento de software. Parte 04: Directrices para la evaluación de procesos (EvalProSoft®). México, DF: NYCE.

Normalización y Certificación. (2005b). NMX-I-059/02-NYCE-2005. Tecnología de la Información -Software- Modelos de procesos y evaluación para desarrollo y mantenimiento de software. Parte 02: Requisitos de procesos (MoProSoft®). México, DF: NYCE.

Oktaba, H. (2006). "MoProSoft®: A software process model for small enterprises," *Proc. of the First International Research Workshop for Process Improvement in Small Settings*, Carnegie Mellon University -Software Engineering Institute, pp. 93-101.

Oktaba, H. & Piattini, M. (2008). *Software process improvement for small and medium enterprises: Techniques and case studies*. Hershey, PA: Information Science Reference Publisher.

Paulk, M. C., Curtis, M. B., Chrissis, M. B., & Weber, C. V. (1995). *The Capability Maturity Model Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley.

Phillips, M. (2011). *CMMI for Acquisition (CMMI-ACQ) Primer*, Version 1.3, CMU/SEI-2011-TR-010, ESC-TR-2011-010, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Pino, F. J., García, F., & Piattini, M. (2008). "Software process improvement in small and medium software enterprises: a systematic review" *Software Quality Journal*, 16(2): 237-261.

Pino, F. J., Alegría, J. A., Vidal, J. C., García, F., & Piattini, M. (2009). "A Process for Driving Process Improvement in VSEs" In Wang, Q., Garousi, V., Madachy, R. & Pfahl, D. (Editors), *Trustworthy Software Development Processes*, Springer Berlin Heidelberg, pp. 342-353.

Pino, F. J., Pardo, C., García, F., & Piattini, M. (2010). "Assessment methodology for software process improvement in small organizations" *Information and Software Technology*, 52(10): 1044-1061.

Preciado, R. (2010). "Diseño e implementación de la guía de autoevaluación sobre la categoría de operación del Modelo de Procesos para la Industria de Software (MoProSoft®)" Tesis de para obtener el grado de Maestro en Informática y Tecnologías Computacionales. Universidad Autónoma de Aguascalientes (UAA). Aguascalientes, México. 2010.

Pressman, R. S. (2010). *Ingeniería de Software: Un enfoque práctico*, 7ª Edición, Mc Graw Hill.

Richardson, I. & Gresse von Wangenheim, C. (2007). "Why are small software organizations different?", *IEEE Software*, 24(1): 18-22.

Sampedro, J. L. (2011). *Conocimiento y empresa: la industria del software en México*. México, D.F: Editorial Plaza y Valdés, S. A. de C.V. / UAM Cuajimalpa.

Serrano, M. A., Montes de Oca, C., & Cedillo, K. (2006). "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*, Carnegie Mellon University -Software Engineering Institute, pp. 81-92.

SCAMPI Upgrade Team. (2011). Standard CMMI appraisal method for process improvement (SCAMPI) A, Version 1.3: Method definition document, (CMU/SEI-2011-HB-001), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Spendolini, M. (1992). *The benchmarking book*. New York: Amacom.

Sulayman, M., Urquhart, C., Mendes, E. & Seidel, S. (2012). "Software process improvement success factors for small and medium Web companies: A qualitative study" *Information and Software Technology*, 54(5): 479-500.

Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Chow-Kian, C., Permadi, R. B. & Feldt, R. (2012). "Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review" *IEEE Transactions on Software Engineering*, 38(2): 398-424.

Wilmoth, F. S., Prigmore, C., & Bray, M. (2002). "HPT Models: An overview of the major models in the field" *Performance and Improvement*, 41(8): 16-24.

Wohlin, C., Hoest, M., & Henningsson, K. (2003). "Empirical research methods in software engineering" R. Conradi and A. Wang (Editors), *Empirical Methods and Studies in Software Engineering*, Lecture Notes in Computer Science, vol. 2765, Berlin-Heidelberg, Germany: Springer-Verlag, pp. 145-165.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslén, A. (2012). *Experimentation in software engineering*. Boston, MA: Kluwer Academic Publishers.

Yin, R. K. (2013). *Case study research: Design and methods*. 4th Edition, Thousand Oaks, CA: Sage Publications.

Young, H., Fang, T., & Hu, C. (2006). "A successful practice of applying software tools to CMMI process improvement" *Journal of Software Engineering Studies*, 1(2): 78-95.

Zahran, S. (1998). *Software process improvement: practical guidelines for business success*. Addison-Wesley, Essex, England.

6.1. Sitios de Internet

[URL-1] Programa de desarrollo informático: IV industria de la tecnología de la información y las comunicaciones. Disponible en: www.inegi.org.mx – Último acceso: Septiembre 2013.

[URL-2] PROSOFT: La política pública que ha transformado una industria. Disponible en: www.prosoft.economia.gob.mx – Último acceso: Febrero 2014.

[URL-3] Resultados de evaluaciones en certificaciones sobre MoProSoft®. Disponible en: <https://dl.dropboxusercontent.com/u/44916249/Padr%C3%B3n%20Nacional%20de%20Empresas%20Dictaminadas.htm> – Último acceso: Mayo 2016.

[URL-4] Programa para el desarrollo de la industria del software (ProSoft). Disponible en: <http://www.economia.gob.mx> – Último acceso: Marzo 2015.