



TESINA DE LICENCIATURA

Título: Sugerencias para la implementación de Competisoft desde Scrum a través de un proyecto real.

Autores: Nora Beatriz Martínez

Director: Hugo Ramón

Codirector: Rodolfo Bertone

Asesor profesional:

Carrera: Licenciatura en Informática (Plan 1990)

Resumen

La industria de software latinoamericana está conformada por un número considerable de empresas PyMEs dedicadas al desarrollo. Estas empresas cuentan con presupuesto y recursos acotados, por lo cual les resulta indispensable contar con un proceso que guíe el desarrollo de software lo más simple y ajustado a sus necesidades como sea posible. Asimismo, para insertarse en el competitivo mercado de la industria de software, mantenerse en el mismo y cobrar notoriedad y renombre, las empresas requieren que, a través del proceso de desarrollo citado, los productos resultantes se obtengan con alta calidad. Por estos motivos, el presente trabajo tiene como objetivo sentar las bases de una definición de un proceso de desarrollo que se ajuste a las necesidades de una empresa PyME, el cual permita generar productos de alta calidad. El análisis de la metodología ágil Scrum, adoptada en un proyecto real y específico de una empresa PyME, en conjunto con el modelo de mejora de procesos Competisoft, facilitará este resultado.

Palabras Claves

Pequeña y Mediana Empresa (PyME)
Scrum
Competisoft
Proceso de desarrollo de software
Modelo de Mejora de Procesos

Conclusiones

En este trabajo se obtuvo la definición de un proceso metodológico para el desarrollo de software con las prácticas y actividades básicas e indispensables que una pequeña empresa de desarrollo de software debe implementar para incrementar la calidad de sus productos y para comenzar a transitar el camino hacia la implementación de un modelo de mejora de procesos, y posteriormente una certificación del mismo.

Trabajos Realizados

En este trabajo:

- Se presentaron las definiciones de las principales Metodologías Ágiles, entre ellas Scrum, así como también de Competisoft.
- Se especificó detalladamente el proyecto real, tomado como caso de estudio.
- Se analizaron similitudes y diferencias entre el proyecto real y Scrum, y éste con el Perfil Base de Proyecto, definido por Competisoft.
- Se propuso la definición de un proceso de desarrollo de software, resultado de estos análisis.

Trabajos Futuros

Como trabajos futuros se propone:

- La evaluación de las categorías Alta Dirección y Gerencia de Competisoft, con el fin de ampliar la definición del Perfil Base de Proyecto, incorporando las prácticas adecuadas al mismo.
- La inclusión de prácticas y actividades que se requieran para completar los Niveles de Madurez 1-Gestionado y 2-Administrado, parte de la estrategia de Competisoft hacia CMMI.

CONTENIDO

1.	Introducción.....	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Resultados Esperados	2
1.4	Tiempo de desarrollo.....	2
2.	Situación Actual.....	3
2.1	La empresa PyME: Tercer Planeta S.A.	3
2.2	Descripción del Proyecto	3
2.2.1	Relación con el Cliente.....	3
2.2.2	Especificación de Requerimientos.....	4
2.2.3	Equipo de Trabajo	15
2.3	Descripción del Proceso	16
2.3.1	Entrega al Cliente.....	17
2.3.2	Ambientes	18
2.3.3	Generación de incremento de funcionalidad.....	19
3.	Scrum.....	23
3.1	Metodologías Ágiles	23
3.1.1	Extreme Programming.....	24
3.1.2	Crystal.....	26
3.2	Scrum.....	27
3.2.1	El proceso	28
3.2.2	Roles	29
3.2.3	Reglas.....	30
3.2.4	Herramientas en Scrum.....	32
3.2.5	Valores de Scrum.....	33
4.	Competisoft	34
4.1	Introducción.....	34
4.2	Trabajos relacionados	34
4.3	Estándares internacionales	35
4.4	La mejora de procesos en PyMEs	35
4.4.1	Factores para el éxito de SPI en PyMEs	36
4.5	Proyecto COMPETISOFT	36
4.5.1	Objetivos del proyecto	37
4.5.2	Marco Metodológico	37
4.6	Modelo de Referencia de Procesos.....	38

4.6.1	Premisas del Modelo	38
4.6.2	Estructura del Modelo de Procesos	40
4.7	Modelo de Mejora.....	44
4.7.1	Visión general del Proceso de Mejora PmCOMPETISOFT	44
4.7.2	Visión general de la Metodología METvalCOMPETISOFT	46
4.8	Estrategias de implementación del Modelo de Procesos.....	49
4.8.1	Estrategia basada en objetivos de negocio	49
4.8.2	Estrategia para iniciar y crecer con éxito	52
4.9	Gestión de Proyectos	56
4.9.1	Proceso Gestión de Cartera de Proyectos.....	56
4.9.2	Proceso de Administración del Proyecto	58
4.9.3	Desarrollo de Software	62
5.	Aporte de la Tesina	64
5.1	Adaptación del Proceso Actual a Scrum.....	64
5.1.1	Reglas de Scrum aplicadas en forma completa.....	65
5.1.2	Reglas de Scrum aplicadas en forma parcial.....	69
5.1.3	Reglas de Scrum no aplicadas	70
5.1.4	Conclusiones.....	72
5.2	Adaptación Scrum a Competisoft	73
5.2.1	Metodologías Ágiles y COMPETISOFT	73
5.2.2	Scrum y COMPETISOFT	74
5.2.3	Conclusiones.....	78
5.3	Proceso Definido	79
5.3.1	Características del Proceso Real	80
5.3.2	Propuesta de Proceso Definido	81
5.4	Resultados Obtenidos	85
6.	Conclusiones.....	87
7.	Trabajos Futuros	88
8.	Bibliografía	89
9.	Tabla de Acrónimos.....	91
10.	Tabla de Traducción	92

1. INTRODUCCIÓN

1.1 Motivación

En la Industria del Desarrollo de Software de Latinoamérica, y en especial en Argentina, gran parte de las empresas dedicadas a ésta área son Pequeñas o Medianas Empresas (PyME). Aunque su definición depende de cada país, para unificar criterios se considerará como Pequeña empresa aquella que cuenta con hasta 20 empleados, y Mediana aquella entre 20 y 100 empleados (1).

Este sector especializado se caracteriza por su tamaño, por el volumen de venta anual y por contar con mano de obra calificada; ésta característica le permite disponer de una estructura adaptativa y flexible a los proyectos de software que desarrollen. Otras características que poseen estas organizaciones son: innovación y capacidad creativa, espíritu emprendedor, infraestructura acorde a sus necesidades, disponibilidad económica limitada, entre otras. Su mayor capital es el “*capital humano*” (2).

El sector informático es una de las áreas económicas más dinámicas, con mayor proyección y con un papel crecientemente importante en la dinámica de las economías de cada país. Dado que existe un numeroso grupo de firmas dedicadas al desarrollo de software, resulta cada vez más indispensable para las PyMEs ofrecer sus productos y servicios a precios competitivos, permitiendo generar de esa forma mejoras de rentabilidad. “Estas empresas tienen la necesidad de adaptarse fácil y rápidamente a los cambios del mercado, y en particular, a los cambios en las necesidades de sus clientes” (2).

Para desarrollar sus productos, las metodologías ágiles ofrecen a las PyMEs la posibilidad de contar con procesos livianos y simples, los cuales pueden adaptarse a su estructura o la “estructura” al proceso. Éstas persiguen principios como la *entrega incremental* de funcionalidad nueva al cliente, priorizándola según el valor de negocio que agrega (de esta forma el producto de software evoluciona en las diferentes entregas logrando la satisfacción del cliente), *mejora continua* y *hacer foco en lo humano* de la organización son puntos que ayudan a las PyMEs a satisfacer sus necesidades potenciando sus características (1).

Además del precio de sus productos, “la *calidad* (tanto en el software desarrollado como en su imagen empresarial) y la *reputación* empiezan a ser factores competitivos y de diferenciación claves en este segmento”. Dado que la calidad del producto de software desarrollado está estrechamente relacionada con la calidad del proceso utilizado, las PyMEs necesitan implementar proyectos para la mejora de sus procesos para, de esta forma, incrementar la calidad de sus productos (4). Tanto el proceso que utilicen para el desarrollo de software, como el modelo de mejora del mismo, tienen que adaptarse a sus necesidades, estructura, operatoria y realidad socio-económica en la que están inmersas, para responder rápida y competitivamente a los cambios del mercado.

En este escenario, se presenta el proyecto Competisoft como un modelo de mejora de procesos que se adapta a las necesidades de las pequeñas organizaciones permitiéndoles visualizar el proceso actual, comprender qué hace, porqué lo hace y en base a este estudio, optimizar el proceso para lograr alcanzar el nivel de calidad al que la empresa aspire. Su propósito es mejorar los procesos de la organización de manera sistemática y coherente (5).

Competisoft está basado en las recomendaciones ISO/IEC 12207 e ISO/IEC 15504 y en criterios coincidentes con los principios mencionados de las metodologías ágiles (mejora continua, entrega incremental, individuos motivados). Por esta razón, la implementación de Competisoft en una organización que utilice alguna metodología ágil para el desarrollo de sus

productos (para este trabajo se tomará Scrum), facilitará a las organizaciones la adopción de un modelo de mejora orientado a la agilidad, permitiéndoles evolucionar, crecer y afianzarse en el sector, obteniendo como resultado un proceso robusto y consistente con el cual generar productos de calidad.

1.2 Objetivos

Partiendo del supuesto que cualquier proceso de desarrollo de software que esté definido e identificado, es mejorable, éste trabajo se propone aplicar el modelo de mejora Competisoft a un *proyecto real de software* utilizando Scrum y evaluar los resultados de tal aplicación. De ésta evaluación se pretende obtener los requerimientos necesarios para poder aplicar Competisoft a Scrum y las sugerencias de adaptación que se tengan que realizar a Competisoft para que sea una opción válida como modelo de mejora sobre un proceso ágil.

Para lograr éste objetivo, se analizarán las reglas y prácticas definidas por Scrum y cómo las mismas fueron aplicadas al Proyecto real bajo estudio, de esta forma, se pretende analizar cuales prácticas se aplican y cuáles no, justificando en cada caso.

Un análisis similar se realizará entre el proceso Scrum y el modelo sugerido por Competisoft, realizando un mapeo de prácticas y reglas definidas por ambos para unir o crear puntos de contacto. De esta forma se pretende obtener como resultado los requerimientos necesarios para aplicar el modelo de mejora a una metodología ágil.

1.3 Resultados Esperados

De los análisis realizados surgirán aquellas prácticas de Competisoft que deben ser implementadas en el proceso del proyecto real bajo estudio. A partir del resultado de ésta implementación, se obtendrá la evaluación necesaria para indicar sugerencias a realizar a Competisoft para poder posicionarlo como un modelo de mejora del proceso ágil, en particular Scrum.

Una consecuencia de este posicionamiento es presentar a Competisoft como una alternativa concreta y viable, para las organizaciones tipo PyME de Iberoamérica, al momento de elegir un modelo de mejora de sus procesos, en pos de aumentar su productividad y competitividad.

1.4 Tiempo de desarrollo

El tiempo de desarrollo de la presente tesina será de seis (6) meses.

2. SITUACIÓN ACTUAL

2.1 La empresa PyME: Tercer Planeta S.A.

Tercer Planeta S.A. es una empresa PyME de sistemas especializada en la Arquitectura, Desarrollo e Implementación de sistemas informáticos y en ofrecer tutoría (o mentoring) en la elección de las tecnologías adecuadas a las necesidades de otras empresas del ramo informático y en la capacitación de sus equipos de trabajo.

En la actualidad, el staff está conformado por 11 integrantes con diferentes habilidades y responsabilidades. La dirección de la empresa se encuentra en manos de dos socios, encargados de la gerencia integral. Dentro del staff existe un Líder técnico quien brinda soporte y apoyo al equipo de desarrollo. Dos miembros del equipo son los responsables de las tareas administrativas, del área de finanzas y contabilidad. El staff se completa con diferentes perfiles tales como desarrolladores, arquitectos, investigadores, analistas y tester.

Dentro de los equipos de trabajo, asignados a los diferentes proyectos, participan los socios y el líder técnico como un miembro más. En éstos equipos no existe una estructura jerárquica definida, es decir, no se define el rol de Líder o Director de proyecto, pero sí existen diferentes roles para desempeñar una serie de responsabilidades y tareas, las cuales podrían extenderse a las propias de su perfil, por ejemplo, un analista podría hacer algún desarrollo o un programador podría realizar las pruebas de alguna funcionalidad, etc. Al manejar diferentes tipos de proyectos con diferentes características, puede suceder que algún integrante participe en más de un proyecto al mismo tiempo, de acuerdo al rol que sea requerido.

Cabe destacar que las funciones contractuales con el cliente son llevadas a cabo por los socios de la empresa. Ellos también se encargan de la negociación con el mismo de requerimientos nuevos, para incluirlos en el presupuesto o no, y en la decisión de los recursos que se destinan a cada proyecto en cada etapa.

En Tercer Planeta S.A. se manejan varios proyectos de desarrollo de software y mentoring de equipos de desarrollo. Todos los proyectos de desarrollo se realizan en las oficinas, es decir, los recursos no se encuentran *in situ* en el cliente. En cambio, los proyectos de mentoring pueden realizarse en el cliente o en las oficinas, dependiendo de la intensidad, la cantidad de participantes y el tipo de proyecto.

2.2 Descripción del Proyecto

2.2.1 Relación con el Cliente

En el proyecto analizado, el cliente es el Departamento de Sistemas de la cadena de canal Fox Latin American Channel, representado por su director. Este departamento se encarga del desarrollo, tercerización y posterior mantenimiento de aplicaciones utilizadas en los distintos departamentos y regiones donde la cadena FoxChannel posee oficinas (tales como Los Ángeles, Buenos Aires, Italia, Rusia, etc.).

La relación con el cliente surge de proyectos anteriormente desarrollados por Tercer Planeta S.A... Para este proyecto en particular, el contrato nace a partir de la necesidad de desarrollar una aplicación que satisfaga las necesidades no cubiertas por aplicaciones utilizadas hasta ese momento por los usuarios de las áreas de Programación, Finanzas y Legales.

No se realizó una estimación exhaustiva del proyecto al comienzo del mismo, aunque sí se realizó una estimación de la "puesta en marcha" del proyecto y de los requerimientos iniciales. El contrato se pactó por un tiempo fijo (6 meses) al cual se hicieron extensiones de plazo a medida que el cliente detectaba nuevas necesidades funcionales para ser desarrolladas, las cuales surgieron por la utilización del producto desarrollado o por la decisión de implementarlo en las regiones/países donde la cadena de canales tiene oficinas.

En cuanto al código fuente, la propiedad del mismo pertenece al Cliente, pero las decisiones de tecnología, arquitectura y herramientas fueron decididas por Tercer Planeta S.A. con el consentimiento de FoxChannel.

2.2.2 Especificación de Requerimientos

A continuación se presentará la especificación de requisitos del proyecto según el estándar *IEEE Std 830-1998 (12)*.

I. **Introducción**

I.1. **Propósito**

Este documento presenta la especificación de requisitos del proyecto.

Está dirigido al Cliente y a los miembros del grupo encargado del desarrollo en Tercer Planeta S.A. para la realización del diseño e implementación del proyecto.

I.2. **Ámbito del Sistema**

El nombre del proyecto, y del producto resultante, es *Program Assets Manager (PAM)*.

El objetivo principal del proyecto es desarrollar un sistema que permita la administración de derechos adquiridos de programas (series o películas) realizados por Fox Latin American Channel (FLAC), centralizando la información de los departamentos de Legales, Finanzas y Programación; permitiendo cubrir las necesidades no proporcionadas por el sistema Paradigm u otro.

Los puntos más destacables del sistema son:

- *Es el punto de inicio en la planificación de futuras adquisiciones de programas.*
- *Lleva la administración integral del presupuesto de programación, permitiendo planificar bajo distintos escenarios y ajustar los datos de cotización hasta que se concreta la adquisición.*
- *Guarda registro de todas las modificaciones realizadas dentro del sistema, permitiendo recuperar la auditoría de tales modificaciones.*
- *Permite calcular la amortización del costo de la adquisición de programas, según los parámetros elegidos por los usuarios al momento de realizar el cálculo respectivo.*
- *Divide el costo de los programas entre los diferentes canales pertenecientes a FLAC, mediante una distribución porcentual definida por los usuarios.*
- *La interfaz es consistente y amigable, diseñada según los requerimientos del cliente, para que sea de uso intuitivo y sencillo.*
- *Permite recuperar información de Paradigm mediante un proceso de importación de datos. Este proceso puede ser ejecutado en forma manual o automática (servicio programable).*
- *Permite agregar datos adicionales a la información recuperada desde Paradigm.*

- *Genera fácilmente reportes de datos consolidados, críticos al momento de realizar el presupuesto.*
- *La estructura del proyecto está preparada para incluir funcionalidad en forma incremental, según las prioridades establecidas por el cliente.*

1.3. Definiciones, Acrónimos y Abreviaturas

Definiciones

Programa: es un conjunto de episodios de una serie o un conjunto de películas.

Episodio: es un elemento de un programa; es un capítulo de una serie o una película.

Fiscal Year (FY): (año fiscal o financiero) es un período de 12 meses usado para calcular informes financieros anuales en negocios y otras organizaciones. Este período para FoxChannel comienza el 1 de julio y finaliza el 30 de junio del año siguiente.

Feed: es una señal que pertenece a un canal. Por ejemplo, FoxLat es un feed del canal Fox.

Amortización: se refiere al proceso de distribución en el tiempo del costo de un episodio o un programa. Este proceso calcula, para un valor a distribuir en varios períodos de tiempo, una amortización (monto), de modo que se reparte el costo entre todos los períodos.

Acrónimos

FLAC: Fox Latin American Channel

PAM: Program Assets Manager

FIC: Fox International Channel

Abreviaturas

FY: Año fiscal (o Fiscal Year).

ABM: siglas en castellano correspondiente a las operaciones de Alta, Baja y Modificación de datos maestros.

CRUD: siglas en inglés correspondientes a las operaciones alta (Creation), consulta (Read), modificación (Update) y borrado (Delete) de un conjunto de datos maestros.

II. Descripción General

II.1. Perspectiva del Producto

Si bien PAM permite recuperar información desde Paradigm, mediante una operación importación de datos (ver detalle en *Funciones del Producto – Interacción con Paradigm*), no depende de él para realizar su operatoria. La información básica de un programa de episodios se carga en ambos sistemas, pero PAM permite agregar información que no existe en Paradigm.

Dado que PAM aún no contempla todas las funcionalidades requeridas por los usuarios, no puede reemplazar completamente a Paradigm. Pero les permite contar con funcionalidad que no brinda Paradigm, la cual es importante para facilitar su trabajo (como reportes de presupuesto o el ingreso de información de derechos de adquisición). Hasta tanto no se definan los circuitos que cubran la funcionalidad completa y las interfaces necesarias que permitan llevarlos a cabo, será necesario cargar cierta información básica y común en ambos sistemas, y realizar la importación de sus datos a PAM.

La aplicación trabaja con una base de datos independiente de otros sistemas, esto le brinda flexibilidad para ser instalada en distintas regiones comerciales que FoxChannel posee en el mundo, por lo que cada instalación es independiente de otra.

II.2. **Funciones del Producto**

Para presentar las funciones del producto, se agruparán las mismas en módulos:

Administración de Futuras Adquisiciones: permite ingresar lo que se planea adquirir en los siguientes años fiscales (FY), trabajando con diferentes escenarios y calculando las amortizaciones según las diferentes planificaciones.

Control de Presupuesto y Amortizaciones: integra la información de futuras adquisiciones (según lo ingresado en el módulo anterior) con las adquisiciones ya realizadas. De esta forma se lleva un control mensual del estado del presupuesto de programación. Para realizar esto, se integra a Paradigm para ver todo lo adquirido.

Administración de Información Legal complementaria: se ingresan todas las restricciones contractuales de utilización y venta de los programas adquiridos (países en los que se puede distribuir, fechas, restricciones de formato, etc.).

Módulo de Reportes de Programación y Finanzas: desde este módulo se extraen todos los reportes que se necesiten para la operación de los departamentos involucrados. Este módulo se integra con Paradigm produciendo los reportes con la información que cada usuario necesite, en el formato que se necesite.

II.3. **Características de los usuarios**

El cliente pertenece al departamento de sistemas de FoxChannel, tiene conocimiento integral del negocio y posee los conocimientos técnicos necesarios para una correcta comunicación. El cliente, quien tiene un amplio conocimiento acerca de Paradigm en cuanto al manejo de la información, se convierte en el principal referente del equipo de desarrollo y el encargado de relevar las necesidades de los usuarios y establecer sus prioridades.

Los usuarios del sistema son gerentes y empleados de los tres departamentos a integrar; están familiarizados con el uso de aplicaciones de escritorio en un entorno Windows; conocen con autoridad la información del departamento al que pertenecen y tienen un conocimiento muy general del circuito de la información manejada por otros departamentos, por lo cual, al permitir la interacción y centralización de datos, el sistema tiene que ser lo suficientemente robusto para no generar inconsistencia de datos en el circuito completo de la información.

El usuario perteneciente al departamento de Programación actúa de referente en aquellos temas en los que el cliente no puede y es el principal usuario de la aplicación.

II.4. **Restricciones**

Debido a política de la empresa Tercer Planeta S.A., los desarrolladores asignados al proyecto no trabajan *in situ* en las oficinas de FLAC. Se coordinan reuniones de avance y para definiciones con el cliente o el usuario referente.

La instalación de nuevas versiones de la aplicación a los usuarios se realiza vía un cliente VPN provisto a Tercer Planeta S.A. por el cliente.

Para la importación de datos desde Paradigm es necesario configurar un cliente ODBC para realizar la conexión con AS/400.

La interfaz gráfica de la aplicación está estructurada de forma similar a la interfaz de “Outlook”, a fin de conservar el mismo formato a otras aplicaciones desarrolladas por la empresa Tercer Planeta S.A. para FLAC.

El lenguaje de programación utilizado es .Net Framework 3.5, tanto para el desarrollo de la interfaz gráfica como de acceso a la base de datos, para lo cual se utiliza un framework desarrollado por Tercer Planeta S.A... La capa de datos es SQL Server 2005.

El idioma de la interfaz gráfica de la aplicación es *inglés*, por esta razón la terminología utilizada en los requerimientos será mayormente ese idioma.

II.5. **Requisitos futuros**

Se tiene previsto la posibilidad de ampliar la sección de generación de reportes para que pueda realizarse a través de una página web, sin necesidad de instalar la aplicación.

Se podrá añadir un módulo para enviar los datos contractuales ingresados por el departamento de Programación al sistema WorkFlow (donde se generan y administran los contratos); y a Paradigm los datos correspondientes a la programación.

III. **Requisitos específicos**

Esta sección contiene la descripción detallada de la funcionalidad que brinda la aplicación a los usuarios, agrupada por módulo. Este detalle permite conocer en profundidad la aplicación.

III.1. **Funciones del producto**

Módulo de Administración

Este módulo abarca la carga de los datos maestros, utilizados a lo largo de todo el sistema. También facilita la configuración de ciertas variables utilizadas en operaciones específicas. Así mismo permite la administración de usuarios y permisos a la aplicación.

Seguridad

El entorno de seguridad permite dar de alta, modificar y eliminar los usuarios que pueden utilizar el sistema y asignar a cada uno los permisos que determinarán las operaciones a realizar; estas operaciones serán detalladas en el módulo respectivo.

La identificación de un usuario, al ejecutar la aplicación, se realiza a través de su cuenta Windows, la cual identifica unívocamente a un usuario en el sistema. Además, por cada usuario se le asigna su Nombre completo.

Los roles definidos en la aplicación son:

- **CRUD Administrator:** habilita al usuario a administrar las operaciones de Alta, Eliminación y Modificación de los datos maestros.
- **Inventory Administrator:** permite dar de alta y modificar Programas, Episodios y Contratos; crear y consultar Órdenes de Compra para un Contrato; crear, modificar y eliminar Costos de presupuesto.
- **Paradigm Administrator:** posibilita al usuario a realizar la Importación de datos desde Paradigm; administrar la asociación tanto de episodios como de canales entre los definidos en Paradigm y PAM.
- **Report Administrator:** el usuario puede emitir los reportes definidos en PAM.

- **System Administrator:** con este rol el usuario tiene acceso a la aplicación completa. Además de acceder a las operaciones de los roles expuestos, puede administrar Usuarios y las variables utilizadas como parámetro en la aplicación.

Parámetros

En la definición de parámetros se permite la configuración de variables utilizadas en las funcionalidades de Contrato y Programas.

Para el Contrato se define el formato del número del contrato (o Master Deal), los directorios donde se guardarán los archivos que se pueden adjuntar a un Contrato y las plantillas que se utilizan para sacar el reporte de un contrato específico.

En el Programa se utiliza el parámetro que configura el tipo de calendario (por mes fiscal o calendario) que se utiliza en el cálculo de las amortizaciones (ver *Administración de Futuras Adquisiciones*).

Datos maestros

Se dispone de una serie de módulos para Altas, Bajas y Modificaciones (ABM) que permiten cargar los datos maestros que serán utilizados por las diferentes funcionalidades incluidas en la aplicación.

No todos los datos maestros disponen de interfaz para ABM, sí los más importantes y aquellos que requieren de actualizaciones más frecuentes.

Para los datos maestros sin interfaz gráfica se realiza una carga inicial de datos y para las actualizaciones (nuevos datos o modificación de existentes) el equipo ejecuta un archivo de actualización que se ejecuta directamente en la base de datos. Debido a que estos datos no sufren modificaciones con frecuencia, se acordó junto al cliente retrasar el desarrollo de las interfaces gráficas, para priorizar, de esta forma, el desarrollo de interfaces con mayor valor para los usuarios, el cliente y el negocio.

Interacción con Paradigm

Debido a que, hasta el momento, los datos de episodios y programas se cargan en forma manual, tanto en Paradigm como en PAM, la interacción de PAM con Paradigm resulta fundamental para chequear que no existan diferencias en los datos básicos. Además, brinda a los usuarios la posibilidad de agregar información relevante para sus funciones, que hoy Paradigm no les permite.

A continuación se detallan las funciones principales que conforman esta interacción.

Importación de datos

Desde PAM se incorporan los datos básicos de programación cargados en Paradigm mediante un *Servicio de Importación*, el cual se configura para que sea ejecutado en forma automática. Existe la posibilidad de realizar el trabajo en forma manual cuando el usuario lo considere necesario.

El proceso de importación de datos desde Paradigm se puede dividir en tres partes, según la criticidad de los datos manipulados:

- *Importación de Episodios:* recupera de Paradigm la información básica de los episodios y los programas, descarta los paquetes que no tienen episodios con derechos a partir la fecha especificada en la importación; ésta fecha se guarda como un parámetro de la importación.

La información abarca datos de *Cabecera* (como Nombre, *Distribuidor* que lo provee, período durante el cual el episodio está activo, costos), *Feeds* (señales donde están distribuidas las alocaiones de los títulos), *Fechas adicionales de In y Out* y *Alocaciones de las amortizaciones* (se calculan en Paradigm, al finalizar un mes).

A los datos recuperados desde Paradigm tienen que poder ser consultados mediante reportes y comparar datos básicos y de cotización con los episodios y programas cargados en PAM.

- *Cálculo de Amortizaciones estimadas*: en Paradigm se calcula la amortización de un episodio al cierre de un mes, no se tiene la amortización del costo en su totalidad. En la importación de episodios, luego de recuperar los datos básicos, se procesa el cálculo de las amortizaciones para tener proyección completa de amortizaciones. El cálculo de amortizaciones proyectadas se realiza en base al costo de episodio, restándole lo ya amortizado, y se proyecta en los períodos que falten; estas amortizaciones se distinguen en PAM de las amortizaciones importadas desde Paradigm con el propósito de brindarle información al usuario al momento de emitir reportes. Para el usuario es necesario contar con ésta información proyectada para calcular el presupuesto del siguiente año.
- *Importación de Distribuidores*: se recupera desde Paradigm todos los Distribuidores cargados y se agregan a los distribuidores ingresados en PAM. El proceso actualiza los datos de los distribuidores que fueron importados anteriormente, agrega aquellos que aún no habían sido importados, y marca como eliminados (en forma lógica) aquellos que están en PAM pero no se trajeron en la Importación.

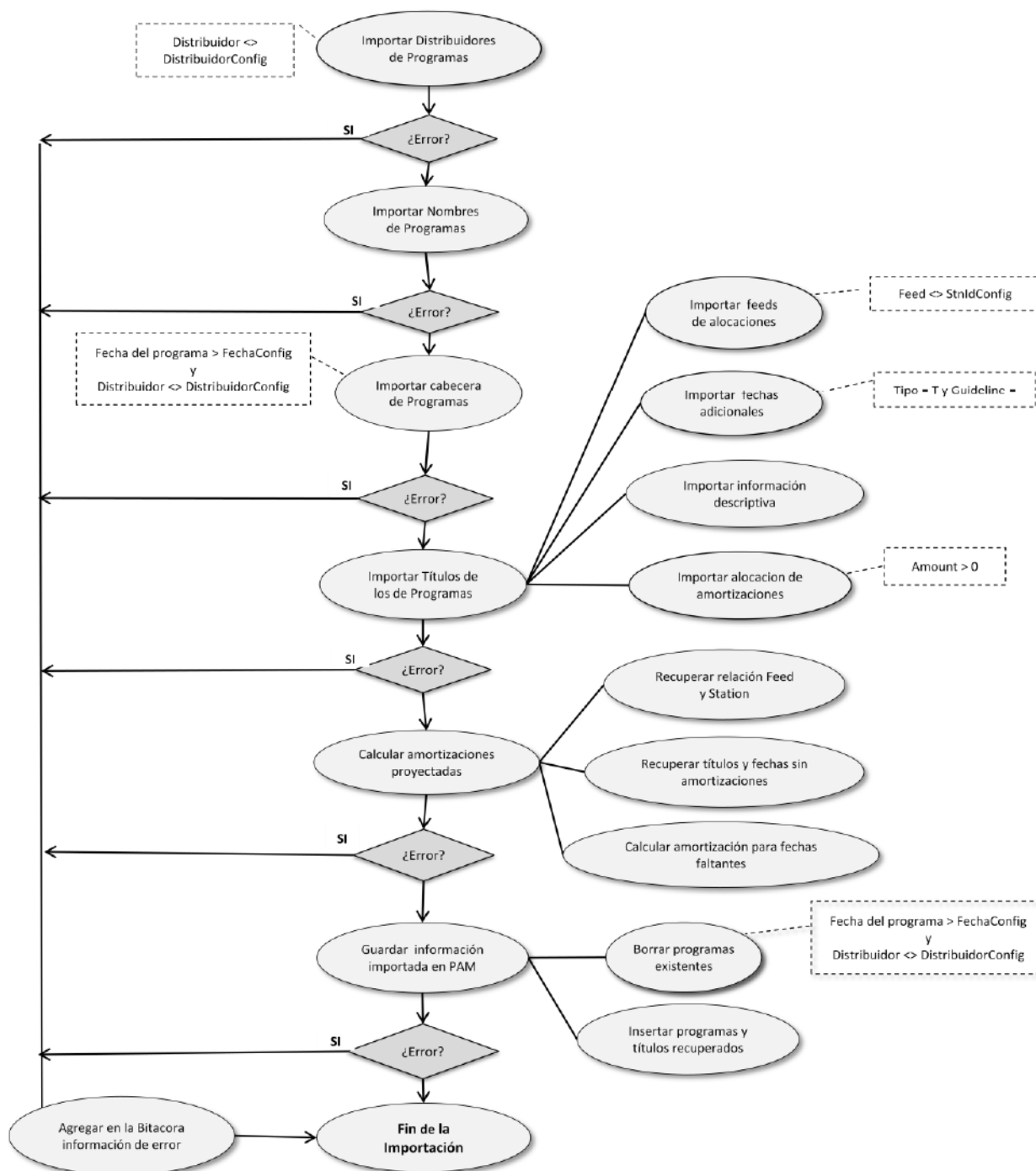


Diagrama de Importación de Datos desde Paradigm

Asociación de Episodios (PAM y Paradigm)

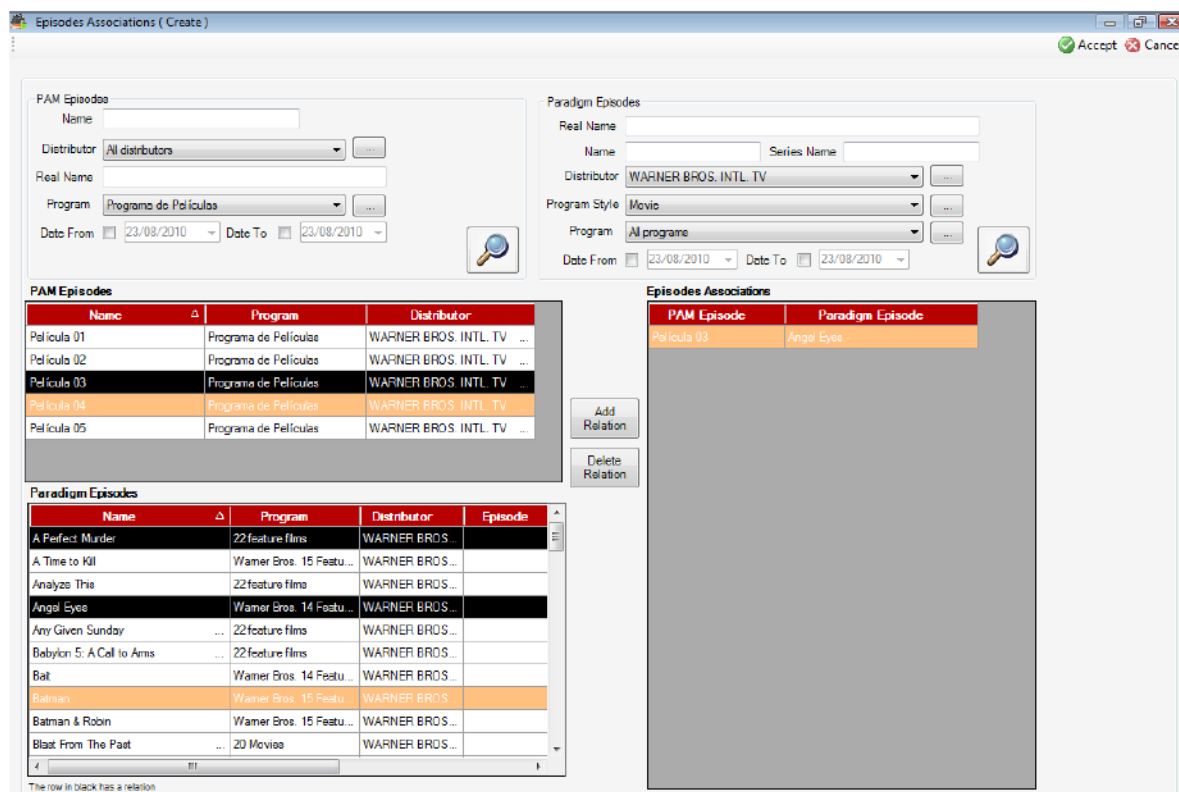
Un episodio dado de alta en PAM puede ser asociado con un episodio (y solo uno) que haya sido descargado desde Paradigm.

Esta asociación es necesaria para identificar los episodios que se encuentran en ambos sistemas, cuáles falta ingresar, conociendo el sistema donde falta, y además facilita la comparación de datos (como amortizaciones, períodos, cotización, etc.) al emitir los reportes.

Para poder realizar la asociación se diseñó la siguiente pantalla en la cual se buscan, teniendo la posibilidad de seleccionar filtros, los episodios cargados en PAM y aquellos

descargados desde Paradigm; nótese que los filtros de búsqueda se eligen de acuerdo al sistema donde están cargados.

Una asociación entre episodios se puede crear y eliminar.



Asociación de Canales y Feeds

Cada Canal cargado en PAM se corresponde con 1 o más Feeds de Paradigm.

Para reflejar en el sistema esta correspondencia, se dispone de una pantalla que permite realizar las asociaciones de Canales y Feeds necesarias.

Esta asociación es importante ya que se utiliza en la Importación de datos desde Paradigm, para adaptar las amortizaciones recuperadas a la estructura que maneja PAM, no solo para aquellas ya cerradas en Paradigm, sino también para las que se calcula su proyección o estimación.

Además, esta adaptación facilita la emisión de reportes en los que se muestra información de Costos y el resultado del cálculo de amortizaciones de los Programas y Episodios.

Reportes comparativos

Mediante una serie de reportes se puede realizar la comparación de datos entre PAM y Paradigm. Entre la información que se puede emitir con los reportes se encuentra:

- Episodios importados desde Paradigm.
- Episodios importados informando si se encuentran o no asociados a episodios de PAM.
- Para aquellos ingresados en PAM, se distingue cuáles están asociados a Paradigm y cuáles no.

Administración de Futuras Adquisiciones

Una adquisición es la compra de un conjunto de Películas y Series a un Distribuidor para ser emitidas por uno o más canales de la cadena.

Éste módulo define la funcionalidad necesaria para poder realizar una planificación de presupuesto de los siguientes años fiscales (Fiscal Year), convirtiéndose en el punto inicial de toda adquisición de películas o series que va a realizar FLAC.

La administración de futuras adquisiciones consta de dos partes:

- El ingreso de programas (conjunto de películas o series) con el detalle de la adquisición realizada.
- El ingreso de un contrato mediante el cual se refleja el acuerdo realizado con el Distribuidor para la compra de películas y/o series.

Ingreso de Programas

Desde el momento que planifica adquirir una serie o película la información con la que se cuenta de los mismos se ingresa al sistema mediante los ABM de Programs y Episodes. A medida que avanza la negociación con el Distribuidor, estos datos se irán ajustando.

Un programa agrupa un conjunto de episodios que tienen datos en común; por ejemplo, un programa de una serie contendrá un episodio por cada capítulo que conforma la temporada; para un programa de películas, un episodio corresponderá a una película que el distribuidor ofrece en un paquete de negociación.

La información de programas y episodios se divide en diferentes “secciones”, cada una de las cuales puede estar cargada a nivel de programa (todos los episodios activos contienen los mismos datos) o a nivel de episodio (uno o más episodios tienen algún dato diferente entre ellos).

En el alta de un programa, el sistema toma los datos ingresados en él para generar automáticamente los episodios, de esta forma el programa actúa como una “plantilla” de sus episodios. Luego el usuario podrá modificar los datos del programa o de algún episodio, optando por guardar los datos a nivel de programa o de episodio.

En la siguiente tabla se muestran las secciones principales con sus atributos más significativos:

Sección	Atributo
Cabecera	Distribuidor Año fiscal Título Cantidad de episodios del programa (total, activos, inactivos) Duración de un episodio (en horas o minutos) y total del programa (suma de la duración de los episodios activos) Tipo y género Número de contrato al que pertenece
Períodos	Períodos (“ventanas”) de licencia sobre el programa o el episodio. Fecha Desde y Hasta Cantidad de meses
Costos	Forma en que se cotizó el programa o episodio (Por Hora o Episodio) Moneda y Tipo de cambio a dólar

	Costo (valor sin impuesto) Impuesto Total (costo + impuesto) Costo total en dólar
Canales	Canal Porcentaje de distribución
Amortización	Calcula amortización (si/no) Calendario broadcast (si/no) Período Desde-Hasta Cantidad de meses Monto de la cuota en dólar Lista de canales en los cuales distribuye Porcentaje de distribución en cada canal Valor de la cuota por cada canal en dólar Plan de distribución de amortización (Fecha, Monto de cuota en dólar)
Episodios ¹	Lista de episodios del programa con: Identificador único (generado automáticamente por el sistema) Nombre (por default: nombre del Programa + número de episodio) Nombre real (será el nombre del episodio en Paradigm, si se encuentra asociado) Duración (en hora/minuto) Los episodios inactivos se marcan en la lista con otro color. No se pueden modificar.
Metadata ²	Información de: Sinopsis (sumario del programa o episodio) Ficha técnica (participes de los episodios)
Derechos legales ³	Contiene información acerca de los derechos legales de emisión asociados a las películas o series que se adquirirán
Adjuntos	Archivo adjunto Descripción/Comentario

Ingreso de Contratos

El ingreso de un Contrato completa la información relacionada a una futura adquisición. Este contrato describe aquello pactado con un Distribuidor, con cierto detalle de la negociación para la adquisición de películas o series.

Un contrato está compuesto de:

Datos de cabecera:

- Nombre y Fecha de contrato.
- Distribuidor, con información necesaria para la facturación (nombre legal, dirección).
- Tipo de contrato (Licencia, producción, etc.).

¹ Solo aparecen en el Programa.

² En este contexto, se refiere a la información que agrega más detalle al programa o episodio.

³ Mas detalle en Administración de Información Legal complementaria

- Estado (pendiente o aprobado).
- Persona que realizó la negociación con el distribuidor y el Responsable de la contratación.
- Entidad licenciataria dentro de FoxChannel.
- Fecha de vencimiento del contrato.

Datos de costos:

- Costo del contrato (moneda del contrato, tipo de cambio respecto al dólar).
- Costo del contrato en dólar.
- Costo de los programas incluidos en el contrato (en dólar).
- Costo de materiales.

Datos de Plan de Pagos:

Detalle del pago que se realizará al distribuidor por la adquisición de las películas o series. Este detalle de pago lo calcula el sistema tomando los datos de cantidad de cuotas, fecha de la primera cuota, días entre cuotas, los cuales ingresa el usuario. El detalle de pagos consiste en una lista de:

- Fecha de vencimiento.
- Porcentaje (respecto al costo en dólar del contrato).
- Monto de la cuota (en dólar).
- Detalle de la facturación del pago (cuando se efectiviza el pago de una cuota, se completa datos de la facturación en el pago).

Datos de Programas:

Se agrega el detalle de lo que se adquirió al Distribuidor, generándose un contrato por Distribuidor y grupo de programas.

El usuario elige si el detalle del contrato será Programa o Episodio, agregando aquellos que no se encuentren incluidos en otro contrato y que estén activos (para los episodios) o no cancelados (para los programas).

Si el contrato es de Programas, se muestra lo siguiente:

- Nombre.
- Cantidad de episodios (total e inactivo).
- Duración.
- Costo total del programa en dólar.

Ahora bien, si el contrato contiene Episodios, se ve la siguiente información:

- Nombre (de PAM).
- Duración.
- Costo del episodio en dólar.

En ambos casos, se puede ir a la pantalla de ABM de cada uno, para ver el detalle. También pueden excluirse del contrato.

Datos de Propuestas:

El departamento de Finanzas de FoxChannel proveyó dos plantillas de documentos de propuesta, Excel y Word, las cuales contienen información legal y financiera del contrato, además de datos específicos del mismo.

El usuario puede agregar al contrato un documento de propuesta, utilizando una de estas plantillas o cualquier documentación que posea en formato digital.

Si se agrega un documento basándose en alguna de las plantillas, el sistema completa en forma automática la información del contrato, guarda el archivo generado y lo adjunta al contrato para que pueda ser recuperado en cualquier momento. En este caso, el documento se genera con el nombre "ContractProposal-" y se concatena la fecha de generación del mismo.

Entonces, en esta sección del contrato se tiene:

- Las plantillas de propuesta, para ser seleccionadas adjuntando el documento generado al contrato
- El nombre del documento de propuesta
- Descripción
- Fecha en que se agregó el documento al contrato

Datos de Derechos legales:

Los datos de derecho legal contienen información acerca de los derechos legales de emisión asociados a las películas o series que se adquirirán.

Para ver el detalle de esta sección, remitirse a "*Administración de Información Legal complementaria*".

Reportes

Períodos de episodios

Programas de PAM, indica si están en un contrato

2.2.3 Equipo de Trabajo

Desde el inicio del proyecto hasta la actualidad, la conformación del equipo de trabajo se definió de acuerdo a las necesidades del proyecto y de la empresa.

A lo largo del ciclo de vida del proyecto se mantuvo un staff permanente, asignado exclusivamente al proyecto, el cual cubre los roles básicos: desarrollador, tester y analista funcional. El rol Líder de Proyecto no figura dentro del staff permanente; sin embargo, las tareas correspondientes al mismo eran desempeñadas por uno de los socios de la empresa, asistido por el analista funcional. Luego de un tiempo, fueron desempeñadas por el analista funcional, acudiendo a los socios en aquellos casos en los cuales era necesario asumir una responsabilidad mayor a su rol de analista.

A partir de éste staff permanente y según la necesidad de cubrir ciertos roles no existentes (por ejemplo, arquitecto) o reforzar los existentes (desarrollador y/o tester), se agrega al equipo, temporalmente, la persona más idónea para desempeñar las tareas requeridas.

Los cambios en la conformación del equipo se realizaron en forma controlada y planificada con cierta antelación para poder lograr un balance con otros proyectos de la empresa y para que el equipo pueda realizar las planificaciones de entrega de nueva funcionalidad al cliente con la mayor información posible.

Los motivos que dieron lugar a estas variaciones fueron diversos, entre ellos se puede citar: la necesidad de entregar determinada funcionalidad fundamental para el negocio; la toma de decisiones políticas por parte del Cliente (como ser justificación de extensión del proyecto,

unificación de criterios entre regiones, etc.), o necesidad, por parte de la Empresa, de asignar recursos a otros proyectos en curso.

Al inicio del proyecto, el equipo de trabajo estuvo conformado por:

- 1 analista funcional y tester; también podría desempeñar tareas de desarrollo en SQL, según las necesidades
- 3 desarrolladores (1 junior, 2 semisenior), cuentan con conocimiento fuerte en .Net y básico en SQL
- 1 referente técnico (desarrollador senior quien no formaba parte del staff permanente del proyecto sino que actuaba como consultor técnico)

Este equipo se encargó de realizar las tareas necesarias para la inicialización del desarrollo del Proyecto: análisis –funcional y técnico- de los primeros requerimientos, definición de la arquitectura, creación de la BD, creación de la solución en .Net y desarrollo de ABMs simples.

Al ser necesario el desarrollo de una funcionalidad compleja, el referente técnico pasó a integrar el staff del proyecto, durante el tiempo que llevó este desarrollo (el cual sería la base de los desarrollos siguientes). Al término de su tarea, el referente técnico vuelve a situarse como consultor.

Cuando el volumen de funcionalidad pendiente y urgente a entregar superó la capacidad de producción del equipo, se sumaron dos personas para reforzar los roles de Desarrollador (2 junior) y Tester (1 semi-junior), quien luego de un período corto de tiempo se desvinculó del equipo para formar parte de otro proyecto.

También se definió en el staff permanente el rol de Líder de proyecto, desempeñado por uno de los socios de la empresa. El objetivo fue ordenar al equipo y reforzarlo; pero no logró desempeñar ese rol debido a que era también responsable por otras tareas, más prioritarias para la empresa. Con lo cual, el analista funcional continuó realizando las tareas necesarias para llevar adelante el proyecto.

Luego de lograr un balance entre los requerimientos pendientes y la capacidad de producción del equipo, y ante la necesidad empresarial de contar con recursos en otros proyectos, el equipo de PAM pasa a contar con:

- 1 analista funcional y tester
- 1 desarrollador semisenior
- 1 referente técnico (como consultor)

Ésta conformación se mantiene hasta la actualidad.

2.3 Descripción del Proceso

La empresa, en sus orígenes, adoptó algunas prácticas descritas por la metodología ágil Crystal Clear porque se adecuaba a sus expectativas. Algunas de ellas fueron:

- mejorar la comunicación entre los integrantes de la empresa.
- desarrollar una relación cercana con el cliente (mediante entrega de funcionalidad temprana).
- integración continua.
- contar con una metodología que se adapte a diferentes tamaños de equipos y tipos de proyecto.

- focalizar en la parte humana del desarrollo de software; este motivo fue el de mayor peso al momento de decidir una metodología a adoptar, ya que el deseo era cambiar la forma de trabajo que conocían la cual se centraba en el proceso y los documentos.

Al comienzo del proyecto PAM se siguieron los lineamientos metodológicos adoptados por la empresa, aunque sin la debida capacitación. Por esta razón, surgieron algunos inconvenientes tales como:

- Dado el constante cambio en los requerimientos y sus prioridades, resultaba difícil llevar la lista de pendientes, la persona encargada de esta tarea era el analista funcional, por lo tanto era quien conocía el estado general del proyecto.
- El equipo de desarrollo desconocía el estado general del proyecto, solo conocía el tema propio de trabajo.
- Entrega de funcionalidad al cliente después de lo pactado.
- La calidad del producto entregado no era la mejor.

Para resolver éstos inconvenientes, algunos miembros de la gerencia buscaron asesoramiento en otra empresa de sistemas, aquí recibieron una pequeña capacitación en Scrum que luego presentaron al equipo completo de Tercer Planeta S.A... La idea era hacer la implementación de las prácticas de Scrum en el proyecto PAM, evaluar los resultados y, dependiendo de éstos, adoptarlas en otros proyectos de la empresa. Las prácticas o productos que se adoptaron fueron:

- Planificación de la iteración.
- Tablero de la iteración con la funcionalidad a entregar al cliente.
- Reunión diaria de sincronización del equipo.
- Retrospectiva, al finalizar cada iteración.

Estas prácticas permitieron contar con un proceso más organizado y claro para el proyecto y lograr cambios significativos:

- Mejoró la comunicación del equipo (retrospectiva y reunión diaria).
- El equipo comenzó a conocer el estado del proyecto (planificación de la iteración, lista de requerimientos del proyecto).
- Todo el equipo tomó conciencia de las tareas y tiempos involucrados en una entrega (pruebas y corrección de errores, instalación de una versión nueva en el cliente, etc.).
- Estimación de la entrega más realista, se eliminaron los desvíos en las entregas.
- Mejoró la organización y el equipo comenzó a auto-gestionarse.

2.3.1 Entrega al Cliente

La entrega de nueva funcionalidad al cliente se realiza al terminar cada iteración.

La duración de la misma, al comienzo del proyecto se definió de cuatro semanas. Una vez que se desarrolló lo necesario para “poner en marcha” la aplicación -con la funcionalidad básica-, y dado que los requerimientos podían ser desglosados para que en corto tiempo el cliente pudiera tener funcionalidad con valor de negocio, se cambió la duración de la iteración a una semana. En aquellas ocasiones en que la funcionalidad solicitada por el cliente requiere más tiempo de desarrollo, se extiende la iteración a dos o tres semanas, definiéndose al comienzo de la iteración.

Con entregas semanales o quincenales, se logró consolidar la relación con el cliente. En cada entrega, a partir del feedback obtenido, se pueden realizar los ajustes necesarios, y

principalmente permite mantener un canal de comunicación fluido e intenso con él. Además, en el equipo se logra enfocar en objetivos cortos y concretos.

Cada nueva versión de la aplicación consta de un ejecutable (.exe) y los archivos necesarios para actualizar la base de datos (stored procedures, modelo de datos, etc.). Estos se entregan al cliente en dos archivos comprimidos (.zip). Más adelante veremos cómo se actualiza la versión.

La instalación de nueva funcionalidad del producto al cliente se realizó de diferentes maneras:

- Al comienzo, parte del equipo de trabajo realizaba la instalación en las oficinas del cliente. En estos casos, finalizada la misma, se realizaba una pequeña demostración al cliente de la funcionalidad entregada.
- Previo acuerdo con el cliente, en algunas ocasiones se le envía vía e-mail los archivos necesarios para la instalación de la nueva versión, esto podía hacerse debido al conocimiento técnico que posee el cliente. Además, en el e-mail se incluía un detalle de la funcionalidad entregada.
- Luego de un tiempo, se obtuvo acceso remoto al servidor del cliente donde está instalada la aplicación, con lo cual la instalación se realiza desde las oficinas de la empresa Tercer Planeta S.A... Una vez ésta actualizada, se procede al envío de un mail al cliente con el detalle de la funcionalidad entregada.

En los casos en que se envía la nueva versión por e-mail o se realiza la instalación en forma remota, las reuniones con el cliente tienen como fin hacer un seguimiento del proyecto y recibir nuevos requerimientos o cambios, definiéndolos y priorizándolos junto al cliente.

2.3.2 Ambientes

Para el desarrollo de nueva funcionalidad, a ser entregada al cliente, se definieron tres ambientes de trabajo, con el fin de separar la generación del código de las pruebas de verificación, integración y validación.

Ambiente de Desarrollo

En este ambiente el equipo de programación realiza la codificación y las pruebas de unidad de la funcionalidad.

En la arquitectura definida en éste ambiente se tiene:

- Solución VisualStudio con los archivos fuentes de las librerías y ejecutable de la aplicación.
- Servidor propio SQLServer2005 con la base de datos que contiene datos, stored procedures y vistas.

Ambiente de QA (Quality Assurance)

El equipo encargado de las pruebas verificación es el encargado de actualizar este ambiente con la funcionalidad liberada por el equipo de programación. Esta actualización se realiza a medida que se libere funcionalidad para ser testeada, razón por la cual puede darse más de una vez en un mismo día. Con seguridad habrá al menos una nueva versión por día.

La arquitectura en este ambiente es:

- Aplicación ejecutable.
- Servidor propio SQLServer2005 con la base de datos que contiene datos, stored procedures y vistas.

Ambiente de Pre-Producción

Este ambiente es un “clon” del ambiente productivo del cliente, en las oficinas de Tercer Planeta S.A.; se mantiene con el objetivo de realizar la instalación del incremento completo del producto a ser entregado, previo a la instalación en el cliente, de esta forma se verifica la consistencia de la instalación. También se realizan pruebas de integración, aunque éstas no son tan fuertes como las realizadas en el ambiente de QA.

La arquitectura de este ambiente consta de:

- Aplicación ejecutable.
- Base de datos propia, instalada en el servidor SQLServer2005 del ambiente QA.
- Datos reales, importados desde el ambiente productivo del cliente, mediante un backup de la base de datos.

2.3.3 Generación de incremento de funcionalidad

Se entiende por *Incremento de funcionalidad* a la actualización de los componentes que integran la aplicación para incluir nueva funcionalidad o mejoras, solicitadas por el cliente y el usuario, al producto de software. En el caso del PAM, los componentes que se actualizan son: aplicación de escritorio (ejecutable), la base de datos (tablas, stored procedures, etc.) y el servicio de importación de datos desde Paradigm.

Se conoce, dentro del equipo de trabajo, como “*Nueva versión*” al paquete de componentes que permite instalar el incremento de funcionalidad, tanto en el ambiente Productivo, como en los ambientes internos (QA y Pre-Producción).

Un paquete de componentes está constituido por:

- Archivo comprimido (zip) con los archivos que conforman el ejecutable (.exe, librerías).
- Archivos .sql (o script) con Procedimientos almacenados y Vistas generados desde la base de datos.
- Archivo .sql (o script) para actualizar la versión del ejecutable en una tabla de la base de datos. Esta tabla, con nombre UpdVersion, permite la configuración de los parámetros que son utilizados por el mecanismo de actualización automática. La misma contiene:
 - número de la última versión de la aplicación;
 - número de la mínima versión que tiene que estar instalada en el cliente;
 - número de la versión del actualizador automático;
 - ruta de una carpeta compartida donde se guarda el paquete de componentes. Los usuarios que ejecuten la aplicación deben tener permisos de lectura sobre esta carpeta.
- Archivo .sql (o script) para actualizar los ítems de tipo “Fallos” en aplicación web Gemini⁴ con el número de versión en el que se encuentra disponible su resolución para ser testeada (únicamente para el ambiente QA).

A continuación se detallan las herramientas que facilitan, al equipo de trabajo, la generación de este paquete de componentes:

⁴ Gemini (desarrollada por Countersoft) es una plataforma web, basada en .NET, para la gestión de proyectos. Es una herramienta versátil ya que permite fácilmente adecuarla al proyecto de software para documentar y hacer un seguimiento tanto de los fallos, encontrados durante las pruebas, como de los requerimientos y funcionalidades que conforman el proyecto. De esta forma permite la gestión de proyectos y facilita la administración de las tareas del equipo de desarrollo.

2.3.3.1 FlexVersion

Herramienta desarrollada por Tercer Planeta S.A. para la administración de versiones de aplicaciones y componentes. Su código fuente reside en repositorios de SourceControl y FileSystems.

Metáfora del sistema:

- Existen repositorios (o depósitos) los cuales identifican el código fuente que contienen. Por ejemplo, "Team Foundation Server 01", "App.folders en ambiente PreProducción".
- Los repositorios están divididos en sectores (o "file groups"), por ejemplo: "Changesets en trunk⁵", "Branches para Release".
- Cada sector contiene uno o más paquetes (versiones) de un determinado producto (aplicación o componente). Por ejemplo, "Versiones para QA de la aplicación 'X'".
- Las acciones toman un paquete (versión), lo duplican o capturan su composición, lo llevan a otro sector y lo rotulan con un nuevo número de versión. Por ejemplo, "Crear branch para Release v.3.6.12 de la aplicación 'X', a partir de la versión para QA número 1132".

2.3.3.2 CruiseControl.Net (CC.Net)

CruiseControl.Net⁶ es un servidor para realizar *Integración continua* de manera automática. Ésta herramienta fue desarrollada por ThoughtWorks.

Es una herramienta versátil que permite la configuración de múltiples actividades para la generación de un incremento de la aplicación. Estas actividades van desde la creación de un archivo ejecutable hasta la ejecución de pruebas unitarias.

La configuración para el proyecto PAM consiste en:

- Build de Integración Continua: se programa un build⁷ diario para forzar la compilación de los módulos de la solución.
- Build a QA, se realiza en forma manual: luego de generar una versión con FlexVersion, se fuerza el build del proyecto. En este caso, se configura el CruiseControl para que realice una serie de tareas:
 - build ordenado de las dependencias y luego de los módulos que integran el producto (solución .Sln de Visual Studio).
 - construcción del archivo comprimido (zip) con el resultado del build realizado a la solución (se comprime el contenido del directorio Bin\Debug de la solución).
- generación del archivo .sql para actualizar la versión del ejecutable (necesario para la actualización automática de la aplicación en la máquina cliente). Para esto, se tiene un "template" de la sentencia en SQL (o query) que actualiza la versión, la tarea definida en CC.Net cambia el parámetro dinámico por el número de versión correspondiente al build lanzado (ver *Actualización de nueva versión*).
- generación del archivo .sql para actualizar los ítems tipo "Fallos" en Gemini (como el caso anterior, se tiene un "template" del query que actualiza los ítems de Gemini, la

⁵ Trunk es la denominación utilizada en VisualStudio para indicar la raíz de la solución VisualStudio.

⁶ CruiseControl.Net es una aplicación de "Open source", implementada usando el framework Microsoft .Net. Desarrollado por [ThoughtWorks](#), líder en el desarrollo de software ágil.

⁷ Se entiende por build a una nueva versión de la aplicación, la cual se genera a partir de los archivos que contienen el código fuente .Net.

tarea definida en CC.Net cambia el parámetro dinámico por el número de versión correspondiente al build lanzado).

Estos archivos generados por el build a QA se guardan, en forma automática, en una carpeta previamente configurada en CruiseControl.Net.

2.3.3.3 Servidor SQLServer2005

Este DMBS fue seleccionado como motor donde se encuentra la Base de datos de la aplicación. Existen dos servidores dedicados al ambiente Desarrollo y QA.

Al generar un build a instalar en el ambiente QA, se tiene que realizar en forma manual la generación del archivo .sql (script) con todos los procedimientos almacenados y las vistas de la base de datos desde el servidor de Desarrollo.

Luego, este archivo se ejecutará en el servidor del ambiente QA, en la base de datos asignada al ambiente respectivo.

Si, en cambio, se tiene que realizar la liberación de una nueva versión para ser instalada en el ambiente productivo, se toma como origen la base de datos correspondiente al ambiente Pre-productivo.

Las herramientas descritas previamente posibilitan el armado de un paquete de componentes para llevar a cabo la instalación de una nueva versión en la máquina cliente. Así, mediante la configuración de las herramientas y posterior ejecución se cuenta con un proceso semi automatizado.

2.3.3.4 Actualización de nueva versión

La aplicación de escritorio cuenta con un *mecanismo de actualización automática* que verifica la disponibilidad de nuevas versiones cada vez que se inicia la aplicación. Este mecanismo es un producto desarrollado por Tercer Planeta S.A., denominado *TPUpdater*, el cual es utilizado en todas las aplicaciones de escritorio desarrolladas por la empresa.

El mecanismo utiliza la tabla UpdVersion para discernir si se realiza o no la actualización de la aplicación. En PAM se actualizan los números de versiones última y obligatoria con el mismo valor para, de esta forma, lograr que la aplicación se actualice sin confirmación por parte del usuario, es decir, se realiza de forma obligatoria.

Durante la instalación de una nueva versión lo primero que es necesario realizar es la ejecución en la base de datos de los archivos .sql (actualización al modelo de datos, procedimientos almacenados, vistas y actualización de la versión del ejecutable en la tabla UpdVersion) en el ambiente correspondiente que quiere ser actualizado: QA, Pre-producción o Producción.

Luego, cuando el usuario ejecuta la aplicación, se “dispara” el mecanismo de actualización TPUdater, el cual realiza los siguientes pasos:

1. obtiene de la base de datos dos números de versiones: última y obligatoria.
2. recupera el número de versión del ejecutable actualmente instalado en el cliente.
3. compara las versiones, para determinar si es necesaria la actualización, de la siguiente manera:
 - si la versión del ejecutable actual es mayor a la versión obligatoria y es distinta a la versión última → pide confirmación por parte del usuario para realizar la actualización.

- si la versión del ejecutable actual es menor a la versión obligatoria → actualiza automáticamente.
 - Si la versión de ejecutable actual es igual a la versión obligatoria → continua con el paso 7.
4. al ser necesaria la actualización descarga el archivo comprimido –que contiene el exe y librerías de la aplicación- desde la carpeta compartida hacia la carpeta donde se encuentre instalada la aplicación en la máquina cliente.
 5. descomprime el archivo con el exe y librerías, actualizando de esta forma la aplicación.
 6. abre nuevamente la aplicación, vuelve a realizar el paso 1. (lo hace cada vez que ejecuta la aplicación).
 7. abre la aplicación.

3. SCRUM

3.1 Metodologías Ágiles

Las *Metodologías* ágiles nacen como una alternativa a las metodologías denominadas tradicionales (o formales), para el desarrollo de software.

Las metodologías tradicionales se basan en procesos predefinidos con una documentación exhaustiva y muy precisa del proyecto, y una detallada planificación que debe seguirse estrictamente, ambos definidos en la fase inicial del proyecto. Estas características causan un alto costo de implementación de cambios, por lo cual no resulta flexible en aquellos proyectos en los cuales el entorno es volátil y los requerimientos cambian constantemente (14).

En contraposición, las *metodologías* ágiles surgen como una respuesta a estos problemas, ya que buscan adaptarse a la realidad del entorno cambiante del desarrollo de software. Se basan principalmente en una planificación adaptativa (en lugar de predictiva) retrasando las decisiones y permitiendo potenciar aún más el desarrollo de software a gran escala; hacen foco en lo humano, dándole mayor valor al individuo y a la colaboración con el cliente; y al desarrollo incremental del software con iteraciones muy cortas (13).

En 2001, un grupo de miembros destacados de la comunidad de software, principales promotores de estas teorías, se reúne con el fin de debatir acerca de la necesidad de adoptar métodos menos formales para el desarrollo de software con el objetivo de resolver la problemática detallada. De esta reunión surge el término de *Metodologías Ágiles*; escriben y firman el *Manifiesto ágil*, el cual se trata de un compromiso público en buscar nuevas y mejores formas de desarrollar software poniendo énfasis en las personas y sus interacciones, la colaboración y la respuesta continua al cambio, explorando nuevas formas de hacer las cosas, y compartiendo experiencias.

“*Manifiesto por el Desarrollo Ágil de Software:*

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre Procesos y herramientas

Software funcionando sobre Documentación extensa

Colaboración con el cliente sobre Negociación contractual

Respuesta ante el cambio sobre Seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Seguimos estos principios:

Nuestra mayor prioridad es *satisfacer al cliente* mediante la entrega temprana y continua de software con valor de negocio.

Le damos la bienvenida a los *cambios de requerimientos*, incluso a finales del desarrollo. Los procesos ágiles controlan los cambios para la ventaja competitiva de los clientes.

Entrega frecuente de *software que funciona*, desde un par de semanas a un par de meses, prefiriendo la escala de tiempo más corta.

Personas con conocimiento del negocio y desarrolladores deben *trabajar juntos diariamente* a lo largo del proyecto.

Construir proyectos alrededor de *individuos motivados*. Proveerles el entorno y apoyo que ellos necesitan, y confiar en que pueden hacer el trabajo.

El método más efectivo y eficiente de transmisión de información hacia y dentro del equipo de desarrollo es la *conversación cara-a-cara*.

Los procesos ágiles promueven el *desarrollo sostenible*. Los promotores, desarrolladores y usuarios deben ser capaces de mantener una constante paz indefinidamente.

La atención continua a la *excelencia técnica y buenos diseños* mejora la agilidad.

Simplicidad, el arte de maximizar el porcentaje de trabajo no realizado, es esencial.

Las mejores arquitecturas, requerimientos y diseños emergen desde *equipos auto-organizados*.

En intervalos regulares, el *equipo reflexiona* en cómo convertirse en más efectivo, entonces *sintoniza y ajusta* su comportamiento en consecuencia." (8)

Entre los métodos ágiles más utilizados se encuentran Scrum y Extreme programming, siendo, en consecuencia, los más reconocidos. También puede citarse a la familia Crystal como otro método representativo de las metodologías ágiles (7). A continuación se presentarán las características destacadas de XP y Crystal, y se expondrá con más detalle a Scrum, dado que es uno de los pilares del presente trabajo.

3.1.1 Extreme Programming

Es el más destacado de los procesos ágiles. Formulado por Kent Beck, entre otros, se basa en las mejores características de la ingeniería de software para minimizar los típicos problemas que pueden surgir en todo el desarrollo de un proyecto de software.

XP se diferencia de las metodologías tradicionales principalmente en que pone énfasis en la adaptabilidad, en lugar de la previsibilidad. Es exitosa debido a que se direcciona hacia la *satisfacción del cliente*, entregando funcionalidad en la medida que el cliente la necesita.

Enfatiza el *trabajo en equipo*. Gerentes, clientes y desarrolladores son socios iguales en un equipo de colaboración. Permite crear un entorno de trabajo simple y efectivo, en el cual los equipos se convierten en altamente productivos. El equipo se auto-organiza alrededor del problema a resolver lo más eficientemente posible.

XP se fundamenta en cinco valores:

Simplicidad, solo se hace lo que se necesita y se solicita. Se toman pasos simples y pequeños hacia el objetivo y se mitigan los errores a medida que surjan. Esto maximiza el valor de la inversión realizada.

Comunicación, todos (gerentes, clientes, desarrolladores) son parte del equipo y se comunican diariamente cara-a-cara. Trabajan juntos tanto en los requerimientos como en la codificación, creando la mejor solución al problema.

Retroalimentación, se toma cada iteración como un compromiso serio, entregando software que funcione. En la demostración del software se escuchan cuidadosamente las necesidades y se realizan los cambios necesarios. Se habla sobre el proyecto y se adapta el proceso al mismo, no al revés.

Coraje, se dirá la verdad acerca del avance y las estimaciones. No se tendrán excusas para el fracaso ya que el plan es tener éxito. No se teme a nada porque ninguno trabaja solo. Cada vez que sucede un cambio, el equipo se adapta.

Respeto, cada uno brinda y siente el respeto que merece como un miembro valioso del equipo. Los desarrolladores respetan la experiencia de los clientes, y viceversa. La gerencia respeta el derecho del equipo de aceptar responsabilidades y de ser autoridad (9).

XP recomienda las siguientes prácticas:

- **Equipo en un mismo lugar.** Todos los integrantes de un proyecto tienen que sentarse juntos. El equipo debe incluir al Cliente (representante del negocio), quien provee los requerimientos y está disponible para responder a preguntas, resolver discusiones, fijar prioridades y dirigir el proyecto.
- El “**juego de la planificación**”. La planificación se compone de dos partes:
 - Planificación de entregas, el Cliente presenta los requerimientos que necesita a los desarrolladores, y estos estiman su dificultad. Con esta estimación y conociendo la importancia de los requerimientos, se establece un plan de entrega inicial para el proyecto. Este plan será revisado regularmente por el equipo para ajustarlo a su velocidad de trabajo.
 - Planificación de la iteración, cada dos semanas (una iteración) el equipo construye un nuevo software. Para esto, el Cliente presenta al equipo los requerimientos que desea en dos semanas, los desarrolladores los desglosan en tareas y realizan una estimación (es más fina que la realizada en la planificación de entregas). Tomando en cuenta la cantidad de trabajo realizada en la iteración previa, el equipo elige los requerimientos a desarrollar en la iteración actual.
- **Entregas pequeñas y frecuentes.** Cada iteración tiene que ser tan pequeña como sea posible. Al finalizar una iteración, el equipo libera software probado que funciona, y entrega valor de negocio, el cual fue elegido por el cliente.
- **Metáforas del sistema.** La metáfora es una simple descripción para describir cómo funciona el sistema. Así, el equipo desarrolla una visión común del sistema. Las metáforas ayudan a cualquier persona a entender el objetivo del sistema.
- **Diseño simple.** Se construye software a partir de un diseño simple, a través de pruebas y mejora de diseño el código evoluciona, manteniendo el diseño exactamente para la funcionalidad que se requiere en el momento. En un proceso incremental e iterativo como XP un buen diseño es esencial, ya que a partir de él, el proyecto crece y evoluciona.
- **Pruebas.** El cliente, para cada funcionalidad que necesita, define los criterios de aceptación; en base a éstos, los desarrolladores escriben las pruebas unitarias para chequear el correcto funcionamiento de lo desarrollado. El cliente realiza las pruebas funcionales. La automatización de estas pruebas es importante ya que permite implementar modificaciones y ejecutar las pruebas, asegurando que el correcto funcionamiento del sistema.
- **Recodificación.** La idea es reescribir código más simple del que está escrito sin perder funcionalidad. Este proceso se focaliza en remover código duplicado y en incrementar su cohesión, al tiempo que reduce el acoplamiento. La recodificación se apoya en las pruebas, para asegurar que, a medida que el diseño evoluciona, el sistema continúa funcionando.
- **Programación de a pares.** Todo el código desarrollado en XP es construido por dos programadores, sentados juntos en la misma máquina. Esta práctica asegura que todo el código producido es revisado al menos por otro programador, y da como resultado mejor código, diseño y pruebas. Cada miembro de la pareja juega su papel: mientras uno codifica y piensa la mejor manera de hacerlo, el otro piensa más estratégicamente:

¿va a funcionar?, ¿puede haber pruebas donde no funcione?, ¿hay forma de simplificar el sistema global para que el problema desaparezca? Además, la programación de a pares contribuye a la comunicación de conocimiento y al desarrollo de las habilidades.

- **Propiedad colectiva del código.** En XP cada programador puede mejorar cualquier parte del código en cualquier momento, ningún miembro es propietario del código. De esta forma, se evita la dependencia de un desarrollador.
- **Integración continua.** El código se debe integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del sistema. El principal beneficio de esta práctica es asegurar que el sistema, después de haber codificado nueva funcionalidad, continúe funcionando.
- **Ritmo sostenible.** El equipo trabaja junto por el tiempo que dure el proyecto, a un ritmo que resulte sostenible en el tiempo. Solo trabajan horas extras cuando es imprescindible y efectivo.
- **Estándares de codificación.** Dado que cualquier integrante puede modificar el código, se programa de a pares y se hace recodificación con frecuencia, es indispensable establecer un estándar de codificación que todos acepten e implementen. Lo más importante es que el código resulte familiar, como si lo hubiera escrito una misma persona, en apoyo a la propiedad colectiva. (10)

3.1.2 Crystal

Crystal es una familia de metodologías de desarrollo de software, creada por Alistair Cockburn (11). A diferencia de otros autores del *Manifiesto Ágil*, Alisar Cockburn fue hacia los principios ágiles por la necesidad de eficiencia, no por la necesidad de manejar rápidamente los cambios de requerimientos, por esta razón Crystal es adecuado para el contexto de precio-fijo.

La familia Crystal se caracteriza por estar centrada en las personas, ser “liviana” en cuanto a la documentación y burocracia, y ajustarse a las necesidades. Además, enfatiza la entrega frecuente, la comunicación cercana y la mejora continua mediante la reflexión.

Hay diferentes metodologías Crystal cada una para diferentes tipos de proyectos. El nombre Crystal viene de la caracterización de proyectos a través de dos dimensiones, tamaño y criticidad, comparable a los minerales en color y dureza.

El color oscuro será para proyectos que requieren más coordinación y comunicación, por lo general proyectos complejos y con mayor cantidad de integrantes. Esta diferenciación de colores se debe a que los proyectos de sistemas que pueden causar más daño necesitan agregar más rigidez en la metodología, así como más reglas de validación y verificación.

Aquí hay una clasificación por color de las metodologías Crystal, de acuerdo al número de personas a ser coordinadas:

Clear	Hasta 8 personas
Yellow	8 a 20 personas
Orange	20-50 personas
Red	50-10 personas

El equipo involucrado en el proyecto está orientado hacia 7 propiedades de seguridad, siendo las tres primeras el núcleo de Crystal, las restantes pueden ser agregadas en cualquier orden para incrementar el margen de seguridad:

- Entrega frecuente.
- Mejora reflexiva.
- Comunicación cercana.
- Seguridad personal (es el primer paso en la confianza).
- Enfoque.
- Fácil acceso a usuarios expertos.
- Entorno técnico con pruebas automáticas, administración de la configuración e integración frecuente.

Al comienzo de un proyecto se elige el miembro de la familia Crystal a adoptar, de acuerdo a las características que presente el proyecto en ese momento. Debido a que el entorno cambia con el tiempo, la metodología elegida se sintoniza nuevamente durante el transcurso del proyecto.

Los Principios que persigue la familia Crystal son:

- El porcentaje de detalle necesario en los documentos de requerimiento, diseño y planificación varía según las circunstancias del proyecto.
- Los documentos de requerimiento, diseño y plan del proyecto pueden ser reducidos en la medida en que vías de comunicación cercanas, ricas e informales están disponibles para el equipo y el software realizado y testeado es entregado rápido y frecuentemente.
- El equipo continuamente ajusta sus convenciones de trabajo de acuerdo a las personalidades particulares de los integrantes, el entorno de trabajo actual y las peculiaridades de la asignación específica.

Las principales prácticas definidas en Crystal son:

- Los integrantes del equipo tienen que estar sentados cerca, manteniendo una comunicación frecuente, amigable y con espíritu de ayuda.
- El equipo trabaja desde la comunicación tácita, es decir, las personas diariamente discuten acerca del proyecto: prioridades, estado, requerimientos y diseños.
- Mantener los procesos burocráticos lejos y dejar que el equipo diseñe libremente.
- Tener al usuario directamente involucrado en el proyecto.
- Tener una buena prueba de regresión, automatizada y disponible.
- Producir funcionalidad que pueda ser entregada pronto y frecuentemente.

3.2 Scrum

Scrum es un enfoque ágil para la construcción de software, desarrollado por Ken Schwaber y Mike Beedle (7).

Es una forma simple de manejar problemas complejos (en los cuales existe un alto grado de incertidumbre acerca de los requerimientos) a través de un marco de trabajo que impulsa la innovación y facilita la formación de equipos auto-organizados que entreguen productos de alta calidad en tiempos cortos.

Un tema clave de Scrum, para abordar la complejidad de un proyecto, es su énfasis en un enfoque empírico en lugar de un proceso definido; de aquí que los tres pilares que sostienen su implementación son: *visibilidad* (los aspectos del proceso que afectan el resultado deben ser visibles), *inspección* (el proceso debe ser inspeccionado con frecuencia para detectar variantes

inaceptables) y *adaptación* (según el resultado de la inspección, se tiene que ajustar el proceso tan rápido como sea posible) (4).

Este marco de trabajo consta de un conjunto de prácticas y reglas sencillas y fáciles de aprender, pero la forma de implementarlas dependerá de cada equipo de trabajo, e inclusive de cada proyecto. Por esta razón, la adopción de Scrum puede ser difícil dependiendo del contexto y de la “interpretación” de las reglas que cada equipo realice.

A continuación, se describen las prácticas y reglas definidas en Scrum (4).

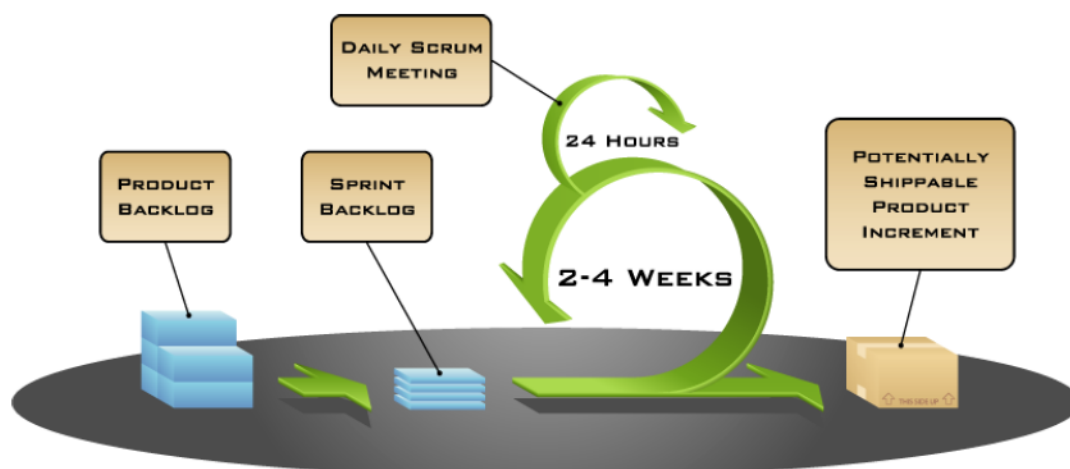
3.2.1 El proceso

El proceso de Scrum es iterativo e incremental, está compuesto por cuatro etapas: Planificación, Puesta en escena, Desarrollo y Liberación (13).

En la Planificación se define el producto a ser desarrollado, su visión y expectativas, se realiza el presupuesto y se crea una lista con todos los requerimientos conocidos, originados por el cliente o los desarrolladores del software, con una estimación inicial.

La etapa *Puesta en escena* requiere de identificar otros requerimientos. Se prioriza la lista de requerimientos completa, la cual se actualiza constantemente con nuevos requerimientos o más detalle de los existentes, más precisión en las estimaciones y actualización de prioridades. En esta etapa se incluye también la definición del equipo asignado al proyecto, recursos, evaluación de riesgos, control de problemas y capacitación necesaria. Se puede incluir un diseño de alto nivel del sistema incluyendo la arquitectura a utilizar.

El propósito de la etapa *Desarrollo* es construir un producto que esté listo para ser instalado. Mediante las prácticas definidas en Scrum, se controlan todos los elementos que pueden variar durante el desarrollo (como requerimientos, recursos, marco de tiempo, tecnologías, herramientas, etc.). De esta forma, los cambios se controlan constantemente con el fin de adaptarlos. La etapa se divide en una serie de bloques iterativos temporales cortos y fijos denominados iteraciones o “sprints” de 7 o 30 días de duración. Cada iteración, núcleo principal de Scrum, incluye las fases tradicionales del desarrollo de software con el fin de desarrollar la funcionalidad nueva o la mejora requerida para producir un nuevo incremento en el producto, el cual tiene valor de negocio para el cliente. La siguiente figura muestra las prácticas de una iteración, las cuales se verán en detalle más adelante:



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE (15)

La etapa *Liberación* consiste en el cierre del producto. Esta etapa sucede cuando se acuerda con el cliente que los requerimientos están completos en su totalidad, no existe ningún ítem pendiente. Entonces, el sistema está listo para ser liberado de la etapa de desarrollo.

3.2.2 Roles

En Scrum existen tres roles: el “Facilitador” de Scrum, Dueño del Producto y el Equipo. Todas las responsabilidades de gestión de un proyecto se dividen entre ellos (4).

3.2.2.1 Facilitador de Scrum (Scrum Master)

Este rol tiene como responsabilidad dirigir el proceso Scrum, para que todos los integrantes del equipo se focalicen en los objetivos a alcanzar. Esto lo logra enseñando Scrum a todos los involucrados en el proyecto, implementando Scrum de tal forma que se ajuste a la cultura de la organización para alcanzar y maximizar sus beneficios, y asegurando que todos los involucrados en el proyecto sigan las prácticas y reglas de Scrum. Es responsable por el éxito del proyecto: colaborando con el Dueño del Producto en la selección de la funcionalidad que mayor valor de negocio aporte y ayudando al Equipo a transformar ésta funcionalidad en un incremento en el producto. Es el nexo entre el Dueño del Producto y el Equipo.

Como facilitador es quien dirige el flujo de las discusiones y quien se encarga de propiciar un diálogo respetuoso, resumiendo los puntos de vista y fomentando el espíritu positivo y productivo dentro del equipo, por esto es importante que permanezca neutral en las discusiones. También se asegura que todos los impedimentos u obstáculos que se le presenten al Equipo sean removidos; para mantener al equipo trabajando lo más productivamente posible durante la iteración, protege al equipo de interferencias externas.

3.2.2.2 Dueño del producto (Product Owner)

Es el representante del cliente para el equipo de trabajo, siendo responsable por representar los intereses de todos aquellos que están interesados en el proyecto y en su producto resultante. Logra la financiación, inicial y permanente, para el proyecto mediante la creación de la lista de requerimientos, los objetivos del retorno de la inversión y el plan de entregas del producto.

Tiene la responsabilidad de llevar el control de los requerimientos y su priorización para asegurar que la funcionalidad con más valor es construida primero. Determina si la funcionalidad, entregada por el Equipo al finalizar una iteración, cumple con los criterios de validación, definidos por él para cada historia.

3.2.2.3 Equipo (Team)

Es el equipo del proyecto, responsable por desarrollar la funcionalidad; para lograrlo, posee la autoridad para decidir las acciones que sean necesarias llevar a cabo y de organizarse a sí mismo para alcanzar los objetivos de la iteración. El Equipo es auto-gestionado, auto-organizado y multidisciplinario. Es responsable de “descubrir” cómo transformar la lista de requerimientos (administrada por el Dueño del Producto) en un incremento de funcionalidad en una iteración y manejar su propio trabajo para lograrlo. Los miembros del Equipo son colectivamente responsables por el éxito de cada iteración, y de todo el proyecto.

3.2.2.4 Personajes: Cerdos vs. Gallinas (“Pigs” and “Chickens”)

Aquellos personas que ocupan los roles descritos son aquellas comprometidas con el proyecto. A ellos se los engloba en la caracterización de “Cerdos” (“Pigs”), dado que forman parte del equipo que trabaja activamente en el proyecto y en su concreción.

Otras personas interesadas en el proyecto, pero que no están comprometidas, tales como usuarios, otros clientes, sponsors, etc., se los generaliza como “Gallinas” (“Chickens”), dado que tienen algún interés en el producto generado a través del proyecto pero no participan comprometidamente del mismo.

Scrum hace una distinción clara entre estos dos grupos y asegura que aquellos responsables por el producto tengan la autoridad para realizar lo que sea necesario para el éxito y aquellos que no tienen responsabilidad alguna no pueden interferir innecesariamente. Esta distinción es importante en Scrum y relevante para la visibilidad total: tiene que estar claro quién está comprometido y quién es solo un “espectador”, para incrementar la productividad, crear impulso y poner fin a forcejeos

3.2.3 Reglas

El Facilitador de Scrum es responsable de asegurar que todos los que estén relacionados con el proyecto sigan las reglas definidas por Scrum, éstas facilitan que cada persona conozca en qué consiste su participación. A continuación se detallarán las reuniones de Planificación, Reunión Diaria, Demostración y Retrospectiva las cuales constituyen las prácticas empíricas de inspección y adaptación utilizadas en Scrum.

3.2.3.1 Reunión de Planificación de la iteración

O “Sprint Planning meeting”, se realiza al comienzo de cada iteración, en ésta participan el Dueño del producto, Facilitador de Scrum y el Equipo con el fin de planificar la siguiente iteración.

Esta reunión consta de dos partes: el Dueño del Producto presenta los posibles requerimientos a desarrollar; luego éstos son desglosados por el Equipo para preparar las tareas de desarrollo de los mismos.

En la primer parte, el Dueño del Producto presenta, en forma detallada al equipo, los requerimientos pendientes de su desarrollo. Estos requerimientos están priorizados por el dueño del producto, de acuerdo al valor de negocio que aporte al producto y el riesgo que podría acarrear su implementación. Durante la presentación, el Equipo analizará en profundidad todos los ítems, para lograr su comprensión.

Para el Equipo, el objetivo de éste segmento es determinar cuánto, de lo presentado por el dueño del producto, será capaz de desarrollar durante la siguiente iteración, realiza una estimación y selección inicial, y se compromete a ello; el equipo es responsable de estos puntos. En cuanto al dueño del producto, su responsabilidad radica en decidir qué ítems de la lista de requerimientos pendientes pueden formar parte de la iteración, de acuerdo a la prioridad asignada por él.

A continuación sucede la segunda parte de la reunión, en la cual el Equipo planifica cada requerimiento seleccionado. Toma cada uno y lo desglosa en las tareas que son necesarias realizar para completar su desarrollo. En este segmento, el dueño del producto debe estar disponible para contestar todas las preguntas que el equipo pudiera necesitar resolver. El equipo se hace cargo de resolver, durante ésta parte, cómo convertir los ítems seleccionados

en un incremento de funcionalidad al producto que sea potencialmente entregable. Aquellos requerimientos con un análisis vago o incompleto podrían no ser bien entendidos con el riesgo que el equipo no sea capaz de completar todos los ítems que seleccione y por los cuales se compromete.

Esta reunión es importante ya que determinará el éxito de la iteración: cuánto más entendimiento tenga el equipo de los requerimientos, más se acercará la funcionalidad entregada a lo que el dueño del producto desea; además, cuanto más certera sea la estimación que realice, será más cercano a lo que el equipo puede producir con el esfuerzo adecuado.

3.2.3.2 Reunión diaria (Daily Scrum meeting)

Mientras transcurre la iteración, cada Equipo se reúne diariamente por un lapso corto de tiempo, en general por 15 minutos, es recomendable hacerla de pie, con el fin de sincronizar el progreso del trabajo realizado por los miembros del equipo, y de informar acerca de cualquier impedimento u obstáculo al Facilitador de Scrum para que busque la forma de resolverlos inmediatamente.

En ésta reunión, dirigida por el Facilitador de Scrum, cada miembro del equipo contestará las siguientes preguntas:

- ¿Qué tareas hice desde la última reunión diaria?
- ¿Qué planeo hacer en el proyecto desde este momento hasta la próxima reunión diaria?
- ¿Qué obstáculos o impedimentos se interponen para continuar con las tareas a las cuales me comprometí?

Se pueden agregar las siguientes preguntas: (7)

- ¿Hay alguna tarea para agregar a la Lista de la iteración?
- ¿Se aprendió o decidió algo nuevo que sea relevante para los miembros del equipo?

La última pregunta permite abrir un foro para que el equipo constantemente aprenda y mejore.

El principal objetivo de esta reunión es sincronizar el trabajo de todos los miembros del equipo, actualizar el estado del proyecto y de las personas, y agendar cualquier reunión que el equipo necesite para continuar con su progreso.

Lo valioso de esta reunión radica en que las personas, al responder la segunda pregunta, expresan un “compromiso social” al equipo, incrementando la responsabilidad y el seguimiento. Permite crear y reforzar un lenguaje común, valores y prácticas que ayudarán al equipo de desarrollo.

3.2.3.3 Reunión de revisión (Sprint review meeting)

Al terminar la iteración, el equipo realiza una reunión de 4 horas en la que se presenta una demostración de la funcionalidad agregada al producto aquella que el equipo desarrolló durante la iteración, al Dueño del Producto, también puede participar cualquier otra parte interesada en ver lo que se logró (usuarios, clientes y sponsors, etc.).

El objetivo de esta reunión es lograr que el dueño del producto apruebe o no la funcionalidad presentada, tomando para esta evaluación los criterios de aceptación definidos en la planificación. De esta forma, el equipo tendrá una devolución rápida y efectiva del trabajo realizado.

Además, esta reunión tiene la intención de acercar a las personas y ayudarles a determinar, en forma colaborativa, lo que el equipo debe hacer en la siguiente iteración. Por lo tanto, podría generarse nuevos requerimientos pendientes e inclusive el cambio de dirección del sistema que se está desarrollando.

3.2.3.4 Reunión de Retrospectiva (Sprint Retrospective meeting)

Una vez la reunión de presentación, el Facilitador de Scrum reúne al Equipo para llevar a cabo la Reunión de Retrospectiva.

En esta reunión, cuya duración es de aproximadamente 3 horas, el Facilitador de Scrum estimula al Equipo a revisar, en el marco del proceso de Scrum y sus prácticas, la iteración que acaba de concluir, su proceso de desarrollo, buscando los puntos a mejorar para hacer más eficiente, agradable y productiva la próxima iteración; se definen las acciones a aplicar.

3.2.4 Herramientas en Scrum

Scrum introduce una serie de nuevas herramientas, o Artefactos, que son utilizadas a lo largo de su proceso. Seguidamente serán detalladas.

3.2.4.1 Lista de requerimientos del proyecto

Conocida en Scrum como Product Backlog, esta lista define todo lo que es necesario que incluya el producto final. En consecuencia, define el trabajo que tiene que realizarse en el proyecto para cumplir con las expectativas y cubrir las necesidades.

El Dueño del Producto es responsable por su contenido, priorización y disponibilidad.

Es una lista dinámica, la cual es constantemente actualizada y priorizada de acuerdo a los cambios que son necesarios realizar para que el producto sea el apropiado, competitivo y útil. Los ítems que contiene son tanto requerimientos concernientes al negocio así como técnicos, tales como errores a corregir, mejoras requeridas, actualizaciones de tecnología, o problemas que requieren solucionarse para continuar con otro ítem, entre otros.

La generación está a cargo del Dueño del Producto y puede haber otros participantes, como usuarios, el equipo de proyecto, marketing, ventas o gestión, y atención al cliente.

La lista está priorizada de acuerdo al valor de negocio que agrega al producto y al riesgo que implica implementar o no el requerimiento. De acuerdo a esta priorización, el Dueño del producto elige ítems a presentar al Equipo durante la reunión de planificación.

3.2.4.2 Lista de requerimientos de la Iteración (Sprint Backlog)

La lista de requerimientos de la iteración define el trabajo o tareas en las cuales el Equipo llevará a cabo durante la iteración con el fin de agregar un incremento funcional de valor al producto que resulte potencialmente entregable.

En la reunión de planificación, el Equipo, el Facilitador de Scrum y el Dueño del producto seleccionan ítems, desde la lista de requerimientos, basándose en su prioridad y el objetivo definido para la iteración en curso.

A diferencia de la Lista de requerimientos del producto, la lista de la iteración, que se arma en la planificación, se mantiene estable hasta que finaliza la iteración. Solo el Equipo puede

cambiarla, de ser estrictamente necesario, durante la Reunión Diaria. Cuando se completan todos los ítems de esta lista, se libera una nueva versión del producto.

Las tareas conforman esta son definidas en la segunda parte de la Reunión de Planificación; mientras transcurre la iteración, cada integrante del equipo elegirá libremente las tareas a realizar y por las cuales asume un compromiso y una responsabilidad. Por lo tanto, es necesario que la lista de requerimientos de la iteración se encuentre visible todo el tiempo, proporcionando la del trabajo que el equipo planeó llevar a cabo durante la iteración.

3.2.5 Valores de Scrum

Los valores que impulsa Scrum son: (7)

- **Compromiso** Todas las personas participantes del proyecto asumen un compromiso. El equipo se compromete, en la iteración, a lograr un objetivo y decidir por sí mismos la mejor forma de alcanzarlo. La gerencia y el Facilitador de Scrum se comprometen a no agregar trabajo nuevo durante una iteración y evitan dirigir al equipo; además proveen los recursos y remueven los obstáculos reportados en la reunión diaria. El Dueño del producto asume el compromiso de definir y priorizar la Lista de requerimientos del producto, guía la elección de los objetivos en cada iteración y brinda una rápida devolución del producto demostrado.
- **Foco** El equipo tiene que ser capaz de focalizarse solo en el objetivo de la iteración, sin distracciones. La gerencia y el Facilitador de Scrum hacen lo posible por colaborar que el equipo no sea interrumpido.
- **Honestidad** Las reuniones y herramientas que conforman las prácticas de Scrum permiten que toda la información concerniente al proyecto se encuentre visible y actualizada en todo momento.
- **Respeto** Los individuos en un equipo son respetados por sus diferentes fortalezas y debilidades, y no se los señala por los fracasos. Todo el equipo adopta la actitud de resolver problemas a través de la búsqueda grupal de soluciones.
- **Valor** La gerencia tiene el valor de planificar y orientar adaptativamente, y de confiar en los individuos y en el equipo para evitar establecer cómo lograr terminar la iteración. El valor del equipo es tomar la responsabilidad por auto-dirigirse y auto-administrarse.

4. COMPETISOFT

4.1 Introducción

La *mejora de procesos de software -Software Process Improvement (SPI)-* es un esfuerzo planeado, gestionado y controlado que tiene como *objetivo incrementar la capacidad de los procesos* de desarrollo de software de una empresa, permitiéndole gestionarlos de manera exitosa. Fundamentados en la premisa: “**la calidad de los productos está íntimamente ligada a la calidad de los procesos utilizados para desarrollarlos**”, las empresas han visto la mejora de procesos de software como una buena estrategia para el aseguramiento de la calidad de los productos de software. En particular, las PyMEs han buscado asegurar la calidad de sus productos de software a través de la mejora de sus procesos fundamentalmente debido a la *necesidad* de hacer de sus proyectos unidades administrativas eficientes y eficaces, y a la posibilidad de establecer y mantener una *posición prestigiosa* en el mercado informático local y global.

Las PyMEs pueden plantearse asegurar la calidad de sus productos a través de la mejora del proceso y la acreditación en modelos de calidad de organismos tales como *Software Engineering Institute (SEI)* e *International Organization for Standardization (ISO)*, sin embargo estos modelos de mejora y evaluación cuentan con recomendaciones complejas de ser aplicadas en las PyMEs, el motivo es que supone una gran inversión en dinero, tiempo y recursos, sumado a que el retorno de la inversión se produce a largo plazo. Otro efecto negativo en estas empresas se puede notar en el choque cultural que se produce al momento de importar y adoptar estos modelos (creados para ser aplicables en grandes empresas) en una organización pequeña.

El *proyecto COMPETISOFT -Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica-* surge como una iniciativa integradora de diferentes propuestas relacionadas con la mejora de procesos de software, aplicable a micro, pequeñas y medianas empresas (PyMEs) así como también a pequeños proyectos. Este proyecto, financiado por el *Programa Iberoamericano Ciencia y Tecnología para el Desarrollo (CYTED)*, tiene como fin crear un **marco metodológico** que pueda ser adoptado por aquellas PyMEs que tengan interés en la mejora de sus procesos.

4.2 Trabajos relacionados

El proyecto COMPETISOFT se basa en la iniciativa realizada en algunos países de Iberoamérica, donde diversos grupos de investigación se enfocaron en el desarrollo de estrategias nacionales para abordar la mejora de procesos en PyMEs, de los cuales surgen estándares y propuestas relacionadas con SPI para ser llevadas a cabo en pequeñas empresas. El proyecto se fortalece de la experiencia de estos trabajos desarrollados previamente, debido a que personas involucradas en el desarrollo de estas estrategias forman parte de los grupos participantes del proyecto COMPETISOFT. (19)

Entre estos trabajos se encuentran:

- El **Modelo de Referencia de Procesos (MoProSoft)** y el **Método de Evaluación de Procesos (EvalProSoft)**, desarrollados al interior del *Programa para el Desarrollo de la Industria del Software (ProSoft)*, bajo el convenio entre la Universidad Autónoma y la Secretaría de Economía de México. El objetivo es la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y producto de software que resulte apropiada a las características de las empresas mexicanas de desarrollo y mantenimiento de software, en su mayoría PyMEs.

- El **marco de trabajo Agile SPI** el cual incorpora la premisa esencial de que los modelos utilizados son ligeros y basados en estándares reconocidos internacionalmente. Fue desarrollado dentro del proyecto colombiano *Sistema Integral para la Mejora de los Procesos Software en Colombia-SIMEP-SW*, financiado por el Colciencias (Instituto colombiano para el desarrollo de la ciencia y la tecnología) y la Universidad del Cauca. El objetivo del proyecto es que las empresas pequeñas y medianas de Colombia sigan un modelo nacional coherente a las características propias de la idiosincrasia y adaptadas al contexto socio-económico del país.
- El **proyecto MPS.BR: Melhoria do Processo de software brasileiro**, financiado, desarrollado y coordinado por el gobierno de Brasil. El objetivo principal es definir e implementar un modelo para la mejora de procesos de software orientado a las micro, pequeñas y medianas empresas, de forma que obtengan un nivel de madurez 2 o 3 de CMMI a un costo accesible.

4.3 Estándares internacionales

Cabe resaltar que los organismos internacionales como el SEI e ISO han realizado esfuerzos para que sus estándares para la mejora de procesos de software puedan ser aplicados a micro y pequeñas empresas desarrolladoras de software.

El SEI se ha interesado en el área de mejora de procesos para “*Small Settings*” (*Improving Processes in Small Settings – IPSS*), haciendo referencia a equipos, proyectos u organizaciones pequeñas.

Por otro lado, ISO ha conformado un grupo de trabajo con el objetivo de establecer un marco común para describir perfiles evaluables del ciclo de vida de software para ser usados en *Very Small Enterprises (VSE)*, organización de menos de 25 empleados. Se planea producir un estándar y dos guías que ayuden a las VSEs a producir productos de software de una manera disciplinada y profesional.

A pesar de estos esfuerzos, dentro de la comunidad de Ingeniería de Software existe una creciente preocupación debido a que los estándares más utilizados y con mayor renombre internacional (por ejemplo CMMI), siguen siendo los estándares que muchas PyMEs continúan utilizando como opción para emprender sus esfuerzos de mejora, a pesar que no son fácilmente aplicables para ellas.

El proyecto COMPETISOFT no pretende ser una competencia de los modelos internacionales SEI e ISO, sino una alternativa y apoyo para que las PyMEs latinoamericanas y las empresas que desarrollen pequeños proyectos puedan iniciar y abordar la mejora de procesos con miras a obtener posteriormente certificaciones de este tipo de estándares.

4.4 La mejora de procesos en PyMEs

La comunidad de Ingeniería de Software a nivel mundial muestra un interés especial en abordar SPI en PyMEs, esto se evidencia en el crecimiento del número de publicaciones realizado con respecto a este tema. En aquellos países que reportan más esfuerzos en la mejora de procesos se nota un compromiso real de las empresas, el Estado, el sector académico para fortalecer la industria de software nacional.

En un programa de mejora se involucran diferentes tipos de modelos, entre ellos:

- El modelo que **gestiona y conduce la mejora**: describe infraestructura, actividades, ciclo de vida y consideraciones prácticas para guiar la mejora de procesos de software en una organización.
- El método de **evaluación de procesos**: especifica la ejecución de la evaluación formal para producir un resultado cuantitativo que caracterice el estado de la capacidad del proceso o la madurez de la organización.
- El modelo de **referencia de los procesos a seguir**: describe qué actividades son reconocidas como las mejores prácticas que una organización debe implementar para la producción de software.

Las propuestas de mejora sugeridas para guiar el esfuerzo de mejora son variadas. La mayor concentración de propuestas radica en: guiar un proyecto de mejora, priorizar la implementación de las mejoras, utilizar modelos de mejora existentes ajustándolos a las necesidades y evaluar las mejoras introducidas por un programa SPI. Otras propuestas de mejora se centran en la definición, evaluación y soporte de los procesos de software.

Los enfoques de mejora pueden presentarse a nivel organizacional o técnico. En el primer caso, son más económicos pero las mejoras no se ven a corto plazo; en cambio, a nivel técnico las mejoras son más costosas pero se ven a corto plazo. (19)

4.4.1 Factores para el éxito de SPI en PyMEs

Para que un SPI sea exitoso en una PyME, se pueden considerar los siguientes factores, obtenidos a partir de casos de estudio: (19)

- La organización debe ser estable para iniciar un programa SPI.
- Iniciar la mejora con una estructura simple del modelo SPI. Mejorar los procesos siguiendo una aproximación incremental. Priorizar los aspectos de mejora.
- Conseguir un rápido retorno de la inversión.
- Minimizar la resistencia al cambio, mediante la concienciación organizacional que la mejora de procesos beneficiará a los empleados y a la empresa.
- Involucrar a todas las personas de la empresa en la búsqueda permanente de la calidad: capacitación en el área SPI a los empleados.
- Supervisar y controlar el programa SPI.
- Establecer un mecanismo de comunicación eficiente entre los diferentes actores involucrados.
- Uso sistemático de métricas adaptadas a la organización.
- Comprometer a la gerencia de la empresa en el programa SPI.
- Lograr la asesoría de un experto para iniciar el programa SPI.
- Abordar el problema de mejora desde la perspectiva técnica.

4.5 Proyecto COMPETISOFT

El proyecto COMPETISOFT fue desarrollado con el **objetivo de mejorar e incrementar la competitividad de las PyMEs** de la industria de software de Latinoamérica mediante la creación y difusión de un **marco metodológico**, orientado a la **mejora continua de sus procesos**, que se ajuste a las necesidades y realidades socio-económicas de las empresas pequeñas y medianas desarrolladoras de software de Latinoamérica que tengan interés en la

mejora de sus procesos; y además, que llegue a ser la base para alcanzar una evaluación y certificación reconocida en el sector informático. (5)

El proyecto aborda la mejora de procesos en PyMEs de manera integral desde varios aspectos:

- Expresa qué hacer en mejora de procesos y sugiere cómo.
- Presenta un conjunto de procesos que describen buenas prácticas para crear software así como también una *guía para su introducción* en pequeñas empresas.

El método de investigación empleado en el proyecto COMPETISOFT ha sido *Investigación-Acción (I-A)*; es una forma de investigación de carácter *colaborativo* que busca unir la teoría y la práctica entre investigadores y profesionales. Una premisa fundamental en esta forma de trabajo es que los procesos sociales complejos (donde puede situarse el uso de tecnologías de la información en organizaciones) pueden ser estudiados mejor *introduciendo cambios en dichos procesos y observando los efectos* que se producen (20).

Los participantes de este proyecto se dividen en dos grupos: el primero conformado por investigadores de distintas universidades de Latinoamérica, y el segundo engloba a profesionales informáticos de PyMEs desarrolladores de software y a los organismos de estandarización, a este grupo se lo denomina *grupo de consulta crítica*.

4.5.1 Objetivos del proyecto

Para lograr el objetivo general del proyecto, fueron planteados los siguientes objetivos específicos: (5)

- *Desarrollar un Marco Metodológico común* ajustado a la realidad socio-económica de las PyMEs iberoamericanas, orientado a la *mejora continua de sus procesos*.
- *Difundir la cultura de la mejora de procesos* en el sector informático iberoamericano y, más específicamente, formar tanto a investigadores y/o docentes universitarios (formación de formadores) como a profesionales de un buen número de PyMEs productoras de software.
- *Incidir en los diferentes organismos de normalización y certificación*, para que asuman que los principios metodológicos objeto de este proyecto pueden ser la base para establecer un mecanismo común, y mutuamente reconocido, de evaluación y certificación de la industria del software Iberoamericana para las PyMEs y los pequeños proyectos de desarrollo de software.

4.5.2 Marco Metodológico

El marco metodológico del proyecto COMPETISOFT, está definido por tres modelos:

- **Modelo de Referencia de Procesos**, es una evolución y refinamiento del modelo MoProSoft.
- **Modelo de Evaluación**. COMPETISOFT sugiere que cada país interesado defina su propio Modelo de Evaluación, acorde a las necesidades de la industria de software y conforme a la norma internacional *ISO/IEC 15504 Information technology – Process assessment*, la cual define cinco *niveles de capacidad de procesos*: *Realizado, Gestionado, Establecido, Predecible y Optimizado*.
- **Modelo de Mejora**, basado en Agile SPI.

Los tres modelos han sido desarrollados y adaptados según las necesidades de la industria de software Iberoamericana a partir de la experiencia que han proporcionado los investigadores, las PyMEs y las unidades gubernamentales participantes del proyecto.

En el proyecto COMPETISOFT se ha definido un único *patrón de procesos* para estructurar todos los procesos de los modelos de la misma forma, haciéndolos más intuitivos y fáciles de utilizar.

4.6 Modelo de Referencia de Procesos

El Modelo de Referencia de Procesos (también denominado Modelo de Referencia o Modelo de Procesos) agrupa los procesos en tres categorías principales: **Alta Dirección, Gestión y Operación**.

El Modelo de Procesos está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software. Las organizaciones⁸ pueden utilizar el modelo ajustándolo a sus necesidades, en caso que no cuenten con procesos establecidos; en su defecto, pueden utilizarlo como punto de referencia para identificar los elementos que les hace falta cubrir. (5)

4.6.1 Premisas del Modelo

El desarrollo del Modelo de Procesos fue realizado y mejorado por personas con amplio conocimiento en el contenido de los modelos internacionales y que cuentan con experiencia en la implantación de estos modelos en PyMEs. La versión inicial del Modelo de Procesos fue probado en varias empresas de Iberoamérica, con el fin de corroborar que se adapte a sus necesidades, en su contexto social y económico. (19)

A continuación se enumeran las principales premisas que guiaron el desarrollo del modelo:

1. *Estar orientado a mejorar los resultados en las organizaciones de desarrollo de software, contribuyendo a los objetivos del negocio y no simplemente a ser un marco de referencia para una certificación o evaluación.*

Destaca el papel de la Alta Dirección, para establecer el rumbo de la organización, e incorpora la Planeación Estratégica, a través del proceso Gestión del Negocio.

2. *Estar dirigido a organizaciones dedicadas al desarrollo y/o mantenimiento de software.*

Los procesos de la categoría Operación (Desarrollo y Mantenimiento de Software) recogen las prácticas indispensables para estas actividades.

3. *No debe requerir de una estructura de organización compleja para poder ser aplicado. Debe adaptarse a la organización de cualquier tamaño, en especial a la pequeña y mediana.*

El modelo considera la estructura básica de una organización: Alta dirección, Gerencia y Operación. Define el concepto de *rol*, que no equivale necesariamente a un puesto, esto le permite a la micro y pequeña empresa crear la cultura de procesos desde el inicio, fomentar la disciplina en cambio de roles y ayuda a crecer ordenadamente.

4. *Servir de base para las organizaciones que no cuenten con procesos establecidos.*

Este modelo puede ser el primer acercamiento de las organizaciones para definir sus procesos debido a que involucran a toda la organización, establecen flujos de trabajo,

⁸ En el presente trabajo se utilizará Organización para hacer referencia a la Empresa o Área interna dedicada al desarrollo y/o mantenimiento de software.

productos y roles responsables, ofrecen información sobre qué hacer y cómo; incorpora guías de ajuste para adecuar los procesos a las necesidades de la organización.

5. *Servir de referencia para las organizaciones que ya tienen procesos establecidos.*

Las organizaciones pueden usarlo para identificar los elementos que les hace falta cubrir e incorporarlos en los procesos que la organización ya tiene establecidos.

6. *Mantener la excelencia en los recursos humanos.*

A través del proceso de Gestión de Recursos el modelo considera como parte fundamental el desarrollo de habilidades y competencias en los recursos humanos, destaca la importancia de los recursos humanos como activo principal de la organización. Personal satisfecho y capaz agrega valor a la ejecución de los procesos, lo que influye positivamente en los resultados que se desean obtener para la organización.

7. *Establecer un mecanismo para mantener el capital intelectual.*

Un punto fundamental para que una pequeña empresa pueda sostenerse es lograr que no dependa en su totalidad del conocimiento de individuos. Para resguardar y hacer disponible el capital intelectual generado por la organización el modelo propone el proceso de Conocimiento (Gestión de Recursos) mediante el cual se crea una base de conocimiento con los productos, mediciones y lecciones aprendidas generados por todos los procesos y proyectos de la organización. Su adecuada explotación puede significar grandes ahorros para la organización, facilitando el re-uso y disminuyendo el re-trabajo.

8. *Establecer un mecanismo para hacer un uso adecuado de los recursos materiales de la organización.*

El modelo considera la gestión de los recursos materiales, en el proceso Bienes, Servicios e Infraestructura (Gestión de Recursos), para la administración eficiente de los mismos, el gasto innecesario y el apoyo al logro de los objetivos financieros de la organización.

9. *Estar basado en los principales estándares internacionales, que aplican a la industria de software, pero siendo práctico.*

El objetivo del modelo es seleccionar y conjuntar las mejores prácticas de los modelos internacionales tomando en cuenta la experiencia práctica de su implantación en países iberoamericanos. El modelo ha integrado las mejores prácticas de estos modelos en un solo marco, para que las empresas las adopten para su beneficio.

10. *Fácil de aplicar.*

Ayuda a las organizaciones a iniciar el primer ciclo en un proceso de mejora, llevarlo a cabo de manera ágil y poder continuarlo, sin que se requieran grandes grupos de apoyo especializados para poder implementarlo.

11. *Estar acorde con el contexto económico de los países de Iberoamérica.*

COMPETISOFT busca constituir la base para que los países de habla hispana desarrollen sus propias competencias e instituciones, generando materiales y talleres de capacitación con el fin de desarrollar consultores locales, para que los precios estén al alcance de las organizaciones de acuerdo al contexto de cada país y no sea necesario solicitar los servicios de capacitación, implantación y evaluación a especialistas o instituciones de otros países.

4.6.2 Estructura del Modelo de Procesos

El Modelo de Procesos está conformado por tres categorías:

- **Categoría de Alta Dirección**, establece la razón de ser de la organización, lo que se desea alcanzar y las estrategias para lograrlo.
- **Categoría de Gerencia**, establece planes de acción para instrumentar las estrategias en cuanto a proyectos, procesos y recursos necesarios.
- **Categoría de Operación**, realiza proyectos de desarrollo o mantenimiento de software que cubran las necesidades del cliente en el tiempo y costo esperados.

Estas categorías se retroalimentan mutuamente de la siguiente forma:

- la Alta Dirección proporciona a la Gerencia los lineamientos para la definición de sus procesos y ésta brinda a la Alta Dirección información de utilidad para la toma de decisiones tendientes a la mejora continua;
- la Gerencia brinda a la Operación los elementos que necesita para el funcionamiento de sus procesos; recibe y evalúa la información y productos generados por los procesos de la Operación, sintetizando los resultados generados para ser comunicados a la Alta Dirección.
- la Operación realiza sus actividades de acuerdo a las directrices marcadas por la Gerencia, teniendo en cuenta siempre los objetivos de la organización; le entrega a la Gerencia toda la información y productos generados.

Como parte de la definición de cada categoría, también se especifican los roles y sus responsabilidades en la ejecución de las prácticas detalladas en cada proceso. Además se ha definido un conjunto de plantillas para la descripción de los productos que se crean en los distintos procesos, para ayudar a los usuarios a estructurar, clasificar y entender la información generada usando los modelos desarrollados por el proyecto. (5)

4.6.2.1 Categoría Alta Dirección (DIR)

La categoría Alta Dirección (DIR) aborda las prácticas relacionadas con la gestión del negocio, realizadas por el sector directivo de la organización. Define el proceso *Gestión del Negocio (GN)*.

Proceso Gestión de Negocio (GN)

El propósito de este proceso es establecer la razón de ser de la organización, identificar los objetivos que se desean alcanzar a corto, mediano y largo plazo, definir estrategias y las condiciones generales para lograrlos y tener una forma para medir el cumplimiento de las metas. Para establecer los objetivos estratégicos propone realizar un diagnóstico de la organización, analizando las necesidades del entorno y de competencia interna. Además, propone la mejora continua de tal manera que la organización paulatinamente pueda ir alcanzando los resultados deseados. Brinda las condiciones necesarias para que la organización responda a los cambios y a sus miembros para trabajar en función de los objetivos establecidos.

La actividad principal de este proceso es **Planeación Estratégica**, con técnicas para definir, comunicar, implantar, monitorear y mejorar las estrategias, buscando que sean claras y consistentes, de tal forma que permitan alcanzar los objetivos planteados a un costo accesible, en un tiempo y esfuerzo satisfactorio para la organización. Las técnicas de Planeación

Estratégica ayudan a tener una organización rentable, fortalece los aspectos relacionados con el cliente y la excelencia de los recursos humanos.

Para realizar el diagnóstico de la organización se puede utilizar una técnica de planeación estratégica conocida como FODA. Esta técnica tiene como función identificar cuatro variables: Fortalezas (F), Oportunidades (O), Debilidades (D) y Amenazas (A). Las fortalezas y debilidades son internas a la organización, siendo posible actuar directamente sobre ellas; en cambio oportunidades y amenazas son factores externos y por lo general resulta difícil modificarlas. Es importante analizar las fortalezas con las que se cuenta e identificar cuáles se tienen que desarrollar para minimizar el riesgo que una amenaza externa afecte a la organización. Además, las fortalezas se deben usar para aprovechar las oportunidades que ofrece el mercado, tratando de convertir las amenazas en oportunidades.

4.6.2.2 Categoría de Gerencia (GER)

La categoría de Gerencia (GER) provee a la organización de procesos, proyectos y recursos, tanto humanos como materiales, para instrumentar las estrategias establecidas por la Alta Dirección. Traduce estas estrategias en Planes detallados de acción.

La Gerencia asegura la provisión de recursos, establecimiento de procesos y realización de proyectos para alcanzar los objetivos de la organización. Vigila que los roles operativos estén trabajando acorde a los procesos, recursos y proyectos establecidos.

Los procesos de esta categoría se detallan a continuación:

Gestión de Procesos (GP)

Tiene como propósito establecer los procesos de la organización, definir, planificar e implantar las actividades de mejora.

Gestión de Cartera de Proyectos (GCP)

Se encarga de asegurar que los proyectos contribuyan al cumplimiento de los objetivos y las estrategias de la organización.

Dentro del alcance de este proceso se encuentra conseguir los proyectos externos e instrumentar cada uno de los proyectos, tanto externos como internos; también incluye la factibilidad de cada proyecto, tratando de asegurar que los beneficios superarán su costo.

Gestión de Recursos (GR)

Se enfoca en administrar de forma eficiente los recursos de la organización, tanto humanos como materiales y de conocimiento. Su propósito es proveer recursos y hacerlo de forma eficaz. Los subprocesos encargados de estas responsabilidades son: Gestión de Recursos Humanos, Gestión de Bienes, Servicios e Infraestructura y Gestión de Conocimiento.

En el desarrollo de software es de suma importancia buscar el crecimiento gradual del capital humano de la organización: hay que proporcionarles a sus miembros el conocimiento técnico, desarrollar las habilidades necesarias y proporcionarles un ambiente de trabajo con los servicios e infraestructura adecuados para que realicen su trabajo de manera eficiente.

Otro recurso importante de la organización es el conocimiento, el cual permite aprender de las experiencias pasadas al resguardar todos los productos de trabajo y documentar las lecciones aprendidas, para aprovechar lo que ya se hizo una vez, evitar cometer los mismos errores y

disminuir el re-trabajo. Este mecanismo ayuda a la organización a volverse más competitiva dentro del mercado.

4.6.2.3 Categoría de Operación (OPE)

La categoría de Operación se encarga de llevar a cabo los proyectos establecidos en la categoría de Gerencia. Se compone de tres procesos:

Administración del Proyecto (AP)

Busca cumplir con los objetivos establecidos en un proyecto en tiempo, costo y alcance establecidos, generando productos consistentes que sirvan para el fin que fueron solicitados. Para esto se deben coordinar eficientemente los recursos, mantener informados a todos los involucrados y atender las Solicitudes de Cambio. En consecuencia, está estrechamente relacionado con los otros dos procesos de la categoría: Desarrollo de Software y Mantenimiento de Software.

Las actividades principales son:

- *Planeación (P)*, el objetivo es realizar un Plan que sirva de guía al proyecto. En el plan se debe identificar los *ajustes* que se deben hacer al proceso de la organización para alcanzar los objetivos del Proyecto. Establecer claramente cuál será el *protocolo de entrega* del producto al cliente. Estimar y acordar con el cliente el número de entregas y establecer las fechas en un *calendario de trabajo*, el cual permite visualizar el alcance. Identificar los *recursos humanos y materiales* necesarios así como los requerimientos de capacitación. Identificar los riesgos y hacer un plan de mitigación, implementarlo y darle seguimiento para evitar que se vean afectados los objetivos del proyecto.
- *Realización (R)* de las actividades establecidas en el Plan de Proyecto, generar informes de avance para mantener informados a los involucrados.
- *Evaluación y Control (E/C)*, se evalúa y controla la calidad del producto que se está generando, se supervisa y evalúa el progreso para identificar desviaciones y realizar Acciones Correctivas cuando sea necesario. También se evalúa el cumplimiento del Plan de Proyecto.
- *Cierre (C)*, se realiza la entrega al cliente de los productos generados, de acuerdo al protocolo de entrega, y se formaliza la finalización del ciclo o proyecto.

Desarrollo de Software (DS)

Pretende lograr un entendimiento claro de las necesidades del cliente para generar un producto consistente que cumpla con sus necesidades, teniendo como propósito la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de producto de software nuevo, cumpliendo con los requisitos especificados. Las especificaciones del Sistema deben ser claras y consistentes en cada uno de los componentes que conforman el sistema, para facilitar la etapa de mantenimiento del producto.

En la actualidad, las buenas prácticas establecen una buena estrategia para mitigar los riesgos desde las fases iniciales del desarrollo es dividir el trabajo en ciclos, entregar al cliente los requerimientos mas prioritarios en los primeros ciclos e ir refinando en ciclos posteriores. Esto facilita al usuario la validación de la implementación de los requerimientos y facilita que el sistema se adapte rápidamente a sus necesidades.

El sistema puede ser desarrollado en uno o varios ciclos, cada ciclo de vida está conducido por los requerimientos, establece productos intermedios y los puntos de revisión. Al desarrollar un sistema por ciclos, el equipo y el cliente logran visibilidad a través del avance en los productos

intermedios, se revisa la consistencia con los requerimientos, a través de revisiones intermedias, y se pueden tomar decisiones de continuar o no con fundamento para los siguientes ciclos.

El Modelo establece las fases⁹ que deben incluirse. Al iniciar cada ciclo (CI) se lleva a cabo la fase de *Inicio (I)* donde el equipo técnico del proyecto revisa el alcance establecido del mismo. Luego, se contemplan las fases: *Requisitos (R)*, *Análisis y Diseño (AD)*, *Construcción (Co)*, *Integración y Pruebas (I/P)* y *Cierre (C)*.

Al finalizar el ciclo se prepara el producto para la entrega y se reúne el equipo para revisar lo que funcionó bien y aquellos puntos que pueden mejorarse en los siguientes ciclos o proyectos.

Cada una de las fases del ciclo se lleva a cabo a través de una serie de actividades¹⁰ realizadas por equipos de trabajo.

En las actividades se hace referencia a las verificaciones y validaciones necesarias para confirmar que el producto que se está desarrollando refleja apropiadamente los requerimientos (verificación) y es capaz de satisfacer los requerimientos para su aplicación o uso (validación). Este tipo de revisiones permiten identificar y remover defectos en los productos desde fases previas para disminuir el costo de las correcciones en fases posteriores. También se incluye actividades de control de versiones.

Pueden agregarse procesos en la capa de operación de acuerdo a las líneas de negocio que se requieran manejar para alcanzar los objetivos de la Organización.

Mantenimiento de Software (MS)

Tiene como objetivo llevar a cabo los cambios solicitados a un producto de software de tal forma que no pierda la consistencia y que cumpla con las necesidades del cliente.

El proceso considera mecanismos para recibir, analizar y dar seguimiento a las peticiones de cambio. Estas peticiones se atienden por grupos y en ciclos, los cuales se clasifican en dos tipos, según la cantidad de días en que se atiende y resuelve una petición de mantenimiento: planificables (1-7 días) y no planificables (más de 7 días).

A cada ciclo se lo denomina SprintM y considera la selección, análisis de las peticiones, intervención y pruebas, así como el seguimiento del registro de peticiones de modificación, así como una serie de reuniones con el fin de obtener el estado de avance y posibles problemas que puedan ocurrir dentro de su ejecución.

Al finalizar cada ciclo se valida con el usuario las modificaciones, se entrega al cliente y el equipo efectúa una reunión para identificar qué se puede mejorar en la solución del próximo grupo de peticiones.

Este proceso ágil permite el ingreso de cambios retroalimentándose constantemente con el cliente, por medio de entregas rápidas y periódicas de atención a las peticiones de mantenimiento.

⁹ Fase: conjunto de tareas que constituyen pasos significativos con la finalidad de generar un producto.

¹⁰ Actividad: conjunto de tareas que uno o varios roles deben ejecutar.

4.7 Modelo de Mejora

La implementación de un modelo de mejora en las pequeñas organizaciones es baja debido a los pocos trabajos relacionados con los modelos que guíen y gestionen la mejora de procesos de software en PyMEs. El Modelo de Mejora de COMPETISOFT guía y gestiona la implementación de proyectos de mejora de proceso en el contexto de pequeñas organizaciones de software. Este modelo está basado en algunos componentes de Agile SPI. (19)

El modelo de mejora ha sido desarrollado con el fin de:

- Guiar y gestionar la mejora de procesos en una pequeña organización de software, y lograr institucionalizar la cultura de la mejora continua al interior de una organización.
- Facilitar su aplicación en las pequeñas organizaciones de software de forma económica, con pocos recursos y en poco tiempo, buscando obtener resultados de mejora visibles a corto plazo.

Para guiar a las PyMEs de software en la conducción y gestión de proyectos de mejora, el Modelo de Mejora de COMPETISOFT define un conjunto de componentes:

- Proceso para guiar la mejora continua de procesos (**PmCOMPETISOFT**).
- *Metodología para la valoración de procesos (METvalCOMPETISOFT)*, evaluación interna de procesos.
- Una guía para formular y ejecutar mejoras utilizando Scrum.
- Una estrategia para la selección y priorización de procesos.

4.7.1 Visión general del Proceso de Mejora PmCOMPETISOFT

PmCOMPETISOFT es un proceso de mejora de procesos de software que guía la ejecución de un ciclo de mejora de procesos en PyMEs, se caracteriza por ser liviano en su aplicación.

Los propósitos de la mejora utilizando PmCOMPETISOFT son:

- **Satisfacer las necesidades de mejora de procesos de la organización**, mediante la entrega temprana y continua de mejoras significativas de los procesos de desarrollo de software. Obtener mejoras a corto plazo, siguiendo un ciclo de vida iterativo e incremental.
- **Diagnosticar continuamente los procesos de la organización**. Analizar, priorizar y llevar a cabo nuevos requerimientos de mejora de procesos en la medida que sea factible.
- **Establecer una colaboración efectiva entre los actores involucrados en el ciclo SPI**. Deben buscarse los mecanismos humanos y tecnológicos para facilitar la comunicación eficiente y efectiva dentro de un equipo de mejora.
- Construir proyectos individuales, grupales y organizacionales en torno a **individuos motivados a la mejora de procesos**. Brindarles la oportunidad, el respaldo y la confianza para que realicen las tareas de manera autónoma.

Además, para el éxito del ciclo de mejora promueve:

- El desarrollo sostenido del ciclo de mejora, a través del trabajo continuo.
- Una infraestructura técnica y de gestión adecuada para soportar la mejora de procesos. Promueve la conformación de una infraestructura organizacional dinámica basada en objetivos.

- El aprendizaje continuo como una disciplina clave para conocer el trabajo, reflexionar acerca de éste y ajustarlo a través de iteraciones cortas y concisas.

4.7.1.1 PmCOMPETISOFT: un proceso de mejora iterativo e incremental

PmCOMPETISOFT es un proceso iterativo e incremental, organizado con *iteraciones de mejora* que se llevan a cabo dentro de un *ciclo de mejora* con el fin de implementar *casos de mejora* de acuerdo a los *objetivos de mejora* de la organización. *El objetivo es obtener resultados rápidos de mejora.*

Una *iteración de mejora* es un incremento pequeño y concreto de mejora que permite adelantar la implementación y gestión de un *caso de mejora* de manera independiente. Las iteraciones pueden ser ejecutadas de manera paralela en áreas de procesos claramente independientes.

Un caso de mejora agrupa oportunidades de mejora concretas, las cuales pueden ser ejecutadas en corto plazo con el fin de entregar de manera temprana y continua mejoras significativas de los procesos de la organización. Los *casos de mejora* permiten gestionar la complejidad de las oportunidades de mejora encontradas en el proyecto de mejora que se ejecuta en la organización.

La filosofía detrás de obtener resultados rápidos permite que las mejoras sean visibles en iteraciones tempranas del ciclo SPI. Con esto se puede lograr mantener la motivación del personal en la mejora continua y *concentrar el esfuerzo en áreas que la organización considere importantes para el negocio*. Es importante establecer objetivos alcanzables que maximicen la posibilidad del éxito y minimicen los riesgos de no alcanzar la meta.

4.7.1.2 Descripción general de PmCOMPETISOFT

El proceso de mejora continua se compone de uno o más ciclos de mejora. Cada ciclo consta de cinco actividades, descritas a continuación en forma general:

- **Actividad 1: Instalación del ciclo.** Se crea o actualiza la *Propuesta de Mejora*, alineada con el *Plan Estratégico*. La propuesta guía a la organización en cada fase siguiente del ciclo. Debe ser aprobada y divulgada para garantizar la asignación de los recursos y el compromiso de los involucrados.
- **Actividad 2: Diagnóstico de procesos.** Se realiza la evaluación interna de procesos para conocer el estado general y analizar resultados para establecer las oportunidades de mejora de un proceso y su prioridad. Queda consignado en el *Informe de Valoración* que forma parte del *Plan General de Mejora*.

Se realiza una planificación preliminar y general del ciclo de mejora, para generar una primera versión del *Plan Preliminar de Mejora*, el cual forma parte del *Plan General del Mejora*.

- **Actividad 3: Formulación de mejoras.** Se planifica la iteración actual del ciclo y se define la estrategia a seguir para mejorar el proceso seleccionado. De la primera iteración se obtiene una medida del esfuerzo de conducir una determinada iniciativa de mejora, la cual es una base futuras estimaciones de esfuerzo, costo, tiempo, recursos, etc. Se registra en el *Plan de Implementación de Mejora*.

Esta actividad puede realizarse varias veces en un ciclo de mejora.

- **Actividad 4: Ejecución de mejoras.** Según la planificación, se gestionan y ejecutan los casos de mejora de la iteración actual. Si la planificación de la iteración se realizó satisfactoriamente, se aceptan e institucionalizan los nuevos procesos de la

organización. Se evalúa la iteración actual para analizar si se cumplieron los objetivos de mejora establecidos. La información de esta actividad se registra en el *Plan de Implementación de Mejora*.

La actividad puede realizarse varias veces en un ciclo de mejora.

- **Actividad 5: Revisión del ciclo.** Se corrigen o ajustan todos los elementos relacionados con la ejecución de cada una de las iteraciones de mejora. Se realiza nuevamente la evaluación interna de procesos para cuantificar las mejoras que se han realizado. Las medidas desarrolladas para evaluar el cumplimiento de objetivos, los procesos mejorados y las lecciones aprendidas se registran en el *Reporte de Mejora*. Si es el último ciclo de mejora, se realiza un análisis general de todo el proyecto y se registra también en el reporte de mejora.

De los casos de estudio conducidos en el proyecto COMPETISOFT se observó que dar este paso no es tan fácil y natural para las pequeñas empresas y es donde la mayoría de los proyectos SPI se paralizan. Esto se debe a que el mayor esfuerzo del trabajo relacionado con la formulación y ejecución de mejoras recae en los empleados de la organización, por lo tanto es importante ofrecer una visión de cómo organizar el trabajo para que el proyecto de mejora pueda seguir su flujo normal. Por esta razón PmCOMPETISOFT, siguiendo la filosofía de la metodología ágil Scrum, aprovecha la sinergia de la valoración de procesos y la encausa hacia la iniciación de la realización efectiva de mejoras al interior de la pequeña organización de software.

4.7.2 Visión general de la Metodología METvalCOMPETISOFT

La metodología de valoración METvalCOMPETISOFT pretende dar soporte a la actividad de diagnóstico de procesos de PmCOMPETISOFT. Apoya el principio de *diagnóstico continuo y rápido* de procesos propuesto por el Modelo de mejora de COMPETISOFT: proceso de valoración método de valoración.

4.7.2.1 Proceso de valoración – PvalCOMPETISOFT

Es un proceso para valorar procesos con el propósito de generar información confiable con la cual identificar, en forma general, fortalezas, debilidades y riesgos de los procesos de software de la organización. Esta información será la base para tomar decisiones relacionadas con el ciclo de mejora.

Los objetivos del proceso de valoración son:

- Realizar el diagnóstico de procesos mediante la valoración de procesos de manera disciplinada a través del cumplimiento y realización sistemática de las actividades y productos de trabajo propuestas.
- Definir objetivos y metas para la valoración de procesos con base en los objetivos de mejora descritos en la Propuesta de Mejora de la organización.
- Apoyar la evaluación de los resultados de cada ciclo de mejora con respecto a las mejoras introducidas en los procesos, supervisar el ciclo de mejora evaluando frecuentemente su eficiencia en la organización.

4.7.2.1.1 Descripción

El proceso para la valoración de procesos de software consta de cinco actividades:

- **Actividad 1: Planificación de la Valoración.** A partir de la Propuesta de Mejora, el Evaluador y el Responsable de Mejora desarrollan y documentan un *Plan de valoración*.
- **Actividad 2: Ejecución de la Valoración.** Según el alcance de la actividad anterior, el Evaluador recolecta y valida sistemáticamente los datos necesarios para la valoración de procesos mediante técnicas de recolección de evidencias.
- **Actividad 3: Generación y Socialización de resultados.** El Evaluador analiza y sintetiza los datos obtenidos en la actividad anterior para determinar el estado actual de los procesos mediante el hallazgo de fortalezas, debilidades y riesgos. Entrega al Responsable de Mejora de procesos el *Informe de Valoración* para su validación y refinamiento.
- **Actividad 4: Priorización de Procesos.** El Responsable de Mejora de procesos, con el Informe de valoración, prioriza las oportunidades de mejora y las organiza en casos de mejora, dependiendo de su complejidad.
- **Actividad 5: Planificación preliminar de Mejoras.** El Responsable de Mejora de procesos registra y planea las estrategias a seguir para abordar las oportunidades de mejora presentadas en el informe con las cuales se guía las diferentes iteraciones. Genera el *Plan Preliminar de Mejora*.

4.7.2.2 Método de Valoración – Light MECPDS

Este método fue desarrollado conforme al estándar internacional ISO/IEC 15004, utiliza el mapeo de los atributos de este proceso estándar como indicadores para la valoración en la dimensión de la capacidad del proceso. Para hacerlo liviano y fácil de utilizar solo se consideran los dos primeros niveles de capacidad de proceso del estándar internacional y sus tres atributos de proceso asociados.

Este método es adecuado para las pequeñas empresas que empiezan en el tema de la mejora de procesos, que no tienen experiencia en la valoración de procesos o que desean valorar solo hasta el nivel 2 de capacidad.

4.7.2.2.1 Indicadores

Light MECPDS está basado en un conjunto de indicadores que guían los propósitos y resultados de todos los procesos dentro del modelo de valoración de procesos. Estos indicadores son:

- **Para la dimensión de la capacidad del proceso:** las *prácticas de gestión y productos de trabajo genéricos* que están asociados a conseguir los resultados de los *atributos de proceso*.
- **Para la dimensión del cumplimiento del proceso:** las *prácticas base y productos de trabajo* asociadas a conseguir los resultados de los procesos definidos en el *Modelo de Referencia de Procesos*.

Los indicadores, para evaluar el nivel de implementación de las prácticas, pueden ser de tres tipos: Directos, Indirectos y Comentarios. Para la valoración de cada atributo en el proceso valorado, se utiliza una escala discreta para la medición, compuesta por:

- **CI – Completamente Implementado** (entre 86 y 100%). Hay evidencias de una completa y sistemática aproximación y logro total al cumplimiento del atributo. No hay debilidades significativas.
- **AI – Ampliamente Implementado** (entre 51 y 85%). Hay evidencias de una aproximación sistemática y un logro significativo al cumplimiento del atributo. La ejecución del proceso puede variar en algunas áreas o unidades de trabajo.

- **PI – Parcialmente Implementado** (entre 16 y 50%). Hay evidencia de alguna aproximación, algún logro al cumplimiento del atributo. Algunos aspectos del cumplimiento del atributo pueden ser impredecibles.
- **NI – No Implementado** (entre 0 y 15%). Hay muy poca o ninguna evidencia de cumplimiento del atributo.

4.7.2.2.2 Dimensión de la capacidad del Proceso

La dimensión de la capacidad del proceso es definida por una escala jerárquica de tres niveles, que representan el incremento de las capacidades de los procesos de desarrollo de software:

- **Nivel 0. Proceso Incompleto.** El proceso no está implementado o fallan los logros de su propósito. Hay poca o ninguna evidencia de algún logro sistemático del propósito del proceso. Hay grandes fallas. Hay muy pocos o ningún producto y/o salida identificados a lo largo del proceso.
- **Nivel 1. Proceso Realizado.** La implementación del proceso logra su propósito de proceso. El propósito del proceso es generalmente alcanzado, aunque no sea planificado o controlado. Los individuos dentro de la organización reconocen que se debe llevar a cabo una acción, la cual es ejecutada. Existen productos generados por el proceso y utilizados para medir el logro de objetivos.
- **Nivel 2. Proceso Gestionado.** A la realización del proceso se le implementa una manera de gestionarlo. Sus productos de trabajo se establecen, controlan y mantienen. El proceso genera productos capaces de ser liberados en tiempo y bajo planes controlables, están alineados con determinados estándares y requerimientos, cumplen con ciertas especificaciones de calidad.

Para determinar en qué nivel de capacidad se clasifica un proceso, se evalúa el cumplimiento de los *atributos del proceso*. Los atributos de proceso son elementos que permiten determinar las capacidades y habilidades de ese proceso; se componen de prácticas de gestión y productos de trabajo genéricos. Una *práctica de gestión* realiza la capacidad de realizar un proceso, soporta la implementación o gestión de un proceso y puede ser aplicada a cualquier proceso, permiten su medición individual para determinar el grado de alcance del atributo al que pertenecen y el nivel en que se encuentra el proceso en estudio. Cada uno de estos atributos, en forma individual, permite medir un aspecto específico de las capacidades y habilidades dentro de un proceso.

Cada nivel exige un grado de cumplimiento y/o un número de atributos de proceso para alcanzarlo. Los atributos de proceso son: Realización del Proceso, Gestión de la Realización y Gestión del Producto de Trabajo. En la siguiente tabla se muestra la relación entre el Nivel de Capacidad y el Grado de cumplimiento de los atributos del proceso evaluados:

Nivel de Capacidad	Atributos del Proceso	Grado de cumplimiento esperado
Nivel 1. Realizado	Realización del Proceso	AI o CI
Nivel 2. Gestionado	Realización del Proceso	CI
	Gestión de la Realización	AI o CI
	Gestión de los Productos de Trabajo	AI o CI

4.7.2.2.3 Dimensión del cumplimiento del proceso

El grado o dimensión del cumplimiento del proceso se caracteriza por enfocarse en las características y propósitos de un proceso específico determinado y definido por el Modelo de Referencia de Procesos. Los procesos están compuestos por *prácticas base* y *productos de*

trabajo. Una práctica base es una actividad de ingeniería de software que directamente guía el propósito de un proceso particular y contribuye a la generación de sus salidas. Es una actividad esencial de un proceso.

Al evaluar un proceso, se toman las prácticas base asociadas al mismo, éstas permiten su medición individual para poder determinar el grado de cumplimiento del proceso en estudio. El valor del cumplimiento de un proceso se obtiene de hallar el promedio de los valores porcentuales de sus prácticas base, las cuales son apoyadas por los productos de trabajo relacionados. Cada práctica base tiene el mismo peso dentro de un proceso específico.

4.8 Estrategias de implementación del Modelo de Procesos

La implementación de un programa de mejora en la organización, que utilice un modelo de referencia de procesos como guía, ayuda a lograr éxito en el cumplimiento de sus objetivos a corto, mediano y largo plazo. El proyecto COMPETISOFT propone usar un Modelo de Mejora de Procesos y el Modelo de Procesos, desarrollados para pequeñas y medianas empresas.

A continuación se especifican dos estrategias de implementación de un modelo de mejora, dirigido a micro y pequeñas organizaciones o de reciente creación, que comienzan un programa de mejora por primera vez. (19)

4.8.1 Estrategia basada en objetivos de negocio

Esta estrategia está dirigida a organizaciones ya establecidas y que se inician por primera vez en un programa de mejora. La estrategia se enfoca en el cumplimiento gradual de los objetivos estratégicos de la organización, y su propósito es implementar un programa de mejora de procesos guiado por el cumplimiento de los objetivos estratégicos establecidos.

Busca lograr un aprovechamiento máximo de los recursos con los que cuenta la organización con el fin de minimizar tiempos de ejecución en los procesos, disminuir costos y lograr mejor calidad en los proyectos.

A continuación se describen sugerencias prácticas de cómo seleccionar los procesos a implementar en función de los objetivos de negocio. La propuesta considera tres ciclos de mejora¹¹, si es bien aplicada permite obtener buenos resultados desde el primer ciclo de mejora.

4.8.1.1 Descripción de la estrategia

Todos los esfuerzos de mejora de procesos tienen que estar alineados al cumplimiento de los objetivos estratégicos de la organización. Esta filosofía se persigue desde los primeros ciclos de mejora, sin significar que para ello sea necesario tener implementados todos los procesos.

4.8.1.2 Identificación de objetivos de negocio

El primer paso es explicitar los objetivos que la organización desea lograr, expresándolos, describiéndolos y encontrando sus dependencias. Es necesario identificar sus relaciones y prioridades para establecer las tácticas que serán implementadas para su logro.

Una herramienta que ayuda a la identificación de dependencias es el *mapa estratégico*: es un conjunto de objetivos relacionados, ayudan a entender la coherencia de los mismos y a

¹¹ Un ciclo de mejora se entiende como un conjunto de actividades encaminadas a obtener un beneficio en la Organización.

determinar las prioridades, de acuerdo al impacto que puede tener una iniciativa sobre la organización. Estos mapas permiten establecer las hipótesis estratégicas.

4.8.1.3 Primer ciclo de mejora

Una vez identificadas las hipótesis estratégicas, se establecen las prioridades de los objetivos a lograr. Para establecer el primer ciclo, teniendo los objetivos priorizados, se eligen aquellos procesos del Modelo de Procesos de la Organización que estén encaminados a cumplir los objetivos seleccionados. Este conocimiento permite analizar qué prácticas de qué procesos del Modelo de Referencia colaboran en el logro de objetivos.

4.8.1.4 Sigüientes ciclos de mejora

En los sigüientes ciclos de mejora se agregan nuevos objetivos del negocio y se deben incluir nuevas prácticas o nuevos procesos de acuerdo a las prioridades establecidas. En cada nuevo ciclo también se refinan los procesos del ciclo anterior para obtener mejores resultados. Esto establece un proceso incremental de mejora continua para alcanzar los resultados deseados cada vez más ambiciosos.

4.8.1.5 Propuesta de perfiles de Proceso para ciclos de mejora

Se denomina *Perfil* al conjunto de procesos que se seleccionan para un ciclo de mejora. A continuación se presentan propuestas de tres perfiles preestablecidos que sirven de apoyo a las organizaciones que inician un programa de mejora. Cada organización puede establecer sus propios perfiles.

Los tres perfiles se detallan a continuación:

4.8.1.5.1 Perfil Base de Proyectos

El objetivo de este perfil es lograr que cada uno de los proyectos que tiene la organización cumplan los compromisos establecidos con el cliente en tiempo, costo y con un buen producto de software que cumpla con todos los requerimientos establecidos. Este perfil se apoya en el cumplimiento de los objetivos de negocio relacionados con la satisfacción del cliente y finanzas sanas de cada proyecto.

Las características de este perfil son:

- Finalizar el proyecto en tiempo, costo y alcance establecidos mediante el proceso Administración del Proyecto.
- Minimizar los costos al encontrar defectos en las fases iniciales en el desarrollo de proyecto mediante las verificaciones establecidas en el proceso de Desarrollo y/o Mantenimiento de Software.
- Generar productos que cumplan con las expectativas y los requisitos del cliente mediante la realización de la fase de Requerimientos establecida en el proceso de Desarrollo de Software o Análisis de Peticiones de modificación del proceso de Mantenimiento de Software.
- Minimizar las incidencias productivas, mediante las pruebas y validaciones con el cliente establecidas en los procesos de Desarrollo y de Mantenimiento de Software.

4.8.1.5.2 Perfil intermedio de cartera de proyectos

Busca administrar eficientemente los recursos para disminuir costos y ofrecer resultados equivalentes en todos los proyectos manejando de forma integral la cartera de proyecto y estandarizando la forma de operar los proyectos. Busca que se puedan re-usar los recursos y el conocimiento entre todos los proyectos de la organización, evitar el re-trabajo y usar eficiente los recursos humanos y materiales. Este perfil se apoya en el cumplimiento de los objetivos estratégicos relacionados con tener finanzas sanas, lograr el crecimiento y aprendizaje de los miembros de la organización y generar productos y servicios de calidad.

Características:

- Minimizar el re-trabajo mediante la utilización de “assets”¹² y lograr que el conocimiento de los proyectos y productos pertenezca a la organización mediante la Gestión de Conocimiento, evitando que el conocimiento se quede sólo en las personas.
- Minimizar costos al contar con plantillas y guías estándar y adecuar los procesos a las necesidades de la organización mediante la Gestión de Procesos.
- Minimizar costos al contar con un proceso eficiente de Gestión de Recursos Humanos, que permite compartir recursos entre los proyectos.
- Adquirir la infraestructura adecuada que permita a los grupos realizar eficientemente su trabajo mediante el proceso de Gestión de Bienes, Servicios e Infraestructura.

4.8.1.5.3 Perfil avanzado de negocio

Este perfil busca fortalecer a la organización y establecer los mecanismos para lograr un crecimiento sostenido. Se apoya en el cumplimiento de los objetivos de negocio relacionados con tener finanzas sanas, el crecimiento en forma sostenida, el aprendizaje de los miembros de la organización y generar productos y servicios de calidad.

Este perfil propone implantar el proceso de Gestión de Negocio de la categoría Alta Dirección y refinar las prácticas de los demás procesos.

Las características de este perfil son:

- Establecer la razón de ser de la organización mediante la actividad de Planeación Estratégica del proceso de Gestión de Negocio, significa definir la misión, visión, valores, objetivos y la forma de alcanzarlos usando estrategias, procesos y recursos.
- Establecer una estrategia de ventas en el proceso de Gestión de Cartera de Proyectos, para conseguir los proyectos que apoyen el cumplimiento de los objetivos del Plan Estratégico.
- Investigar nuevas propuestas tecnológicas e incorporarlas oportunamente en los procesos y productos.
- Realizar el trabajo orientado a procesos en toda la organización.

A continuación se mencionan las principales ventajas y desventajas de la implementación de la *estrategia de proceso basada en objetivos de negocio*:

Ventajas

- La mejora de procesos se realiza de acuerdo a las prioridades del negocio.

¹² Assets: Activos, se refiere a los componentes de Ingeniería de Software que pueden ser reutilizados.

- Al manejarse relativamente pocos conceptos, disminuye la complejidad de la implementación en la organización.
- Se hace foco en grupos específicos de la organización en cada ciclo.
- Se reducen los riesgos de la organización al, paulatinamente, ir cambiando la forma de operar.
- Se obtienen resultados visibles para la Alta Dirección relacionados con los objetivos de negocio, desde etapas tempranas de mejora.
- Se definen los procesos conforme la organización los requiere.

Desventajas

- Se requiere experiencia en el proceso iterativo e incremental para refinar las versiones anteriores de los procesos con base en los resultados obtenidos y para poder mejorar los procesos de los nuevos ciclos.

4.8.2 Estrategia para iniciar y crecer con éxito

La estrategia está ideada para las organizaciones en incubación o de reciente creación con el fin de ayudarlas en la adopción del Modelo de Referencia de Procesos, mediante la utilización de una estrategia de cuatro pasos. Cada paso corresponde a una etapa de crecimiento de la empresa: Incubación, Primer proyecto, Crecimiento y Éxito sostenido, aplicándose cada paso en forma incremental y secuencial.

La estrategia adapta los perfiles de procesos de la estrategia basada en objetivos de negocio, exceptuando el primer perfil, en función del cumplimiento de las condiciones en las que se encuentra la organización.

Para la implementación de la estrategia para iniciar y crecer con éxito, se define para cada paso las etapas que es necesario cumplir, con las condiciones de entrada, pasos a seguir y las condiciones de salida que permitan continuar con la siguiente etapa.

4.8.2.1 Perfil Inicial de Desarrollo de Software para la etapa de incubación

Para las empresas de desarrollo de software que se inician y que desean concretar la venta de un prototipo a un primer cliente lo fundamental es obtener la funcionalidad y la calidad suficiente para lograr esta primera venta, sin importar el esfuerzo y el tiempo invertidos.

Para alcanzar este objetivo se recomienda, como primer paso de la estrategia, que la organización fortalezca las prácticas de desarrollo de software y adopte prácticas de pruebas y verificación de manera sistemática. Además de estas prácticas, el Perfil Inicial de Desarrollo de Software consiste en adoptar otras prácticas técnicas del proceso de desarrollo de software, tales como especificación de requerimientos, análisis y diseño, construcción e integración, y, de ser posible, incluir las validaciones correspondientes.

Descripción

Definir, obtener y armar el Proceso de Desarrollo.

Condiciones de entrada, situación de la organización:

- No hay límite estricto de tiempo de entrega del producto de software.
- El cliente o usuario final son hipotéticos o están disponibles bajo demanda.

- Existe confianza dentro del equipo de trabajo. La comunicación es fluida y frecuente.
- Colaboración e intercambio de conocimiento sin restricciones.
- Control de versiones y de respaldo de los productos se realiza a nivel individual.

Pasos a seguir:

1. La Alta Dirección asigna a un Responsable de Gestión de Proceso (RGP).
2. RGP revisa las prácticas requeridas para el Perfil Inicial y las compara con las prácticas seguidas actualmente, identificando si se realizan o no. RGP discute con los miembros involucrados en desarrollo de software para obtener el consenso sobre adoptar las prácticas faltantes.
3. RGP coordina la generación de la versión documentada del proceso de Desarrollo de Software, se acuerdan técnicas para las prácticas y las plantillas para los productos de trabajo, adecuadas para la organización. En la definición de este proceso deben participar *todos* los involucrados. Se tiene que aprovechar al máximo las técnicas conocidas y los productos ya generados; si es necesario, investigar y capacitar para incorporar elementos nuevos.
4. RGP dirige la implementación del proceso de Desarrollo y/o Mantenimiento de Software definido en un ciclo completo de desarrollo.
5. La Alta Dirección y RGP revisan el resultado de la aplicación de los procesos con los participantes, se identifican y acuerdan oportunidades de mejora y se realizan ajustes en la definición del proceso.

Condiciones de salida

El proceso documentado de Desarrollo y/o Mantenimiento de Software, adecuado a las necesidades de la organización, fue probado y está listo para aplicarse en ciclos y proyectos.

4.8.2.2 Perfil Base del Proyecto para la etapa del Primer proyecto

La organización logra firmar un contrato con un cliente para el desarrollo de un nuevo producto o el mantenimiento de uno existente, por lo cual es necesario cuidar los costos de desarrollo para obtener ganancias.

Para lograr estos objetivos, como segundo paso de la estrategia, se recomienda que la organización agregue a las prácticas del Perfil Inicial las prácticas de Administración de Proyecto y de control de los productos de trabajo.

En el Perfil Base del Proyecto se incluyen prácticas del proceso Administración del Proyecto, como planeación, seguimiento, control y manejo de riesgos. Además se tiene que incorporar la creación de un repositorio para los productos de trabajo del proyecto, el control de versiones y el respaldo, todas prácticas incluidas en la Gestión de Conocimiento. Al proceso de Desarrollo de Software es necesario refinarlo para mejorarlo con actividades del proceso Administración del Proyecto: revisión del plan al inicio de cada ciclo, asignación de tareas al equipo de trabajo, reportes de actividades y cierre.

Descripción

Definir, Mejorar el Proceso de Desarrollo y el Proceso de Administración del Proyecto.

Condiciones de entrada:

- Se tiene el contrato con el cliente para un proyecto o se tiene un proyecto interno con fechas comprometidas.
- Se cuenta con los recursos humanos asignados y capacitados en el proceso de desarrollo.
- Se tienen los recursos de bienes, servicios e infraestructura suficientes para realizar el proyecto.

Pasos a seguir:

1. Alta Dirección asigna el Administrador del Proyecto (AP).
2. RGP revisa con AP las prácticas requeridas para el Perfil Base e identifican, al compararlas con las prácticas aplicadas actualmente por la organización, aquellas que se realizan y las que no.
3. RGP y AP discuten con los participantes del proyecto el resultado del diagnóstico para obtener el consenso sobre la pertinencia de adoptar las prácticas faltantes.
4. RGP coordina la generación de la versión documentada del proceso de Administración del Proyecto y la modificación del proceso de Desarrollo de Software. Se documentan las técnicas elegidas para las prácticas y las plantillas de los productos de trabajo, según las necesidades de la organización. Tienen que participar todos los involucrados y aprovechar al máximo las técnicas conocidas y los productos ya generados en la organización. Investigar y capacitar para incorporar nuevos elementos.
5. RGP dirige y vigila la implementación de los procesos Administración del Proyecto y Desarrollo y/o Mantenimiento definidos para la organización en un proyecto específico.
6. Alta Dirección y RGP revisan el resultado de la aplicación de los procesos con los participantes, se identifican y acuerdan oportunidades de mejora y se realizan los ajustes necesarios en la definición de los procesos.

Condiciones de salida:

Los procesos de Administración del Proyecto y Desarrollo y/o Mantenimiento de Software mejorados están listos para aplicar en proyectos subsecuentes.

4.8.2.3 Perfil Intermedio de la Cartera de Proyecto para la etapa de crecimiento

Cuando la organización tiene más de un proyecto, comienza la competencia por los recursos. Los procesos no siempre se desarrollan en el mismo orden en todos los proyectos, se necesita mayor control global del uso de los recursos, del avance de los proyectos, de la satisfacción de los clientes, así como también el mejor aprovechamiento del conocimiento de la organización.

El tercer paso de la estrategia consiste en que la organización agregue al Perfil Básico las prácticas de los procesos Gestión de la Cartera de Proyecto, Gestión de Recursos y Gestión de Proceso.

Descripción

Control satisfactorio de Recursos, Proyectos y Procesos.

Condiciones de Entrada:

- La organización tiene más de un proyecto, que deben realizarse al mismo tiempo.
- Existen lineamientos de Alta Dirección con respecto a los objetivos de negocio, estrategias, procesos requeridos, portafolio de proyectos, estrategia de recursos y presupuesto.

Pasos a seguir:

1. Alta Dirección asigna el Responsable de Gestión de Cartera de Proyectos (RGCP) y al de Gestión de Recursos (RGR).
2. RGP revisa con RGCP y RGR las prácticas requeridas para el Perfil Intermedio y hacen un diagnóstico comparándolas con las prácticas seguidas actualmente por la organización.
3. RGP, RGCP y RGR discuten con la Alta Dirección el resultado del diagnóstico para incluir o no las prácticas faltantes.
4. RGP coordina la generación de la versión documentada de los procesos de Gestión de Cartera de Proyectos, Gestión de Recursos, así como de Gestión de Proceso. Se agrega al proceso de Administración del Proyecto la conexión con los nuevos procesos. Se documentan las técnicas elegidas para las prácticas y las plantillas de los productos de trabajo. Deben participar *todos* los involucrados. Se tiene que aprovechar las técnicas conocidas y los productos generados. Si es necesario, investigar y capacitar para incorporar elementos nuevos.
5. RGP dirige y vigila la implementación de todos los procesos definidos para la organización, durante un período establecido por la Alta Dirección.
6. Al final del período, la Alta Dirección, RGP, RGCP y RGR revisan el resultado de la aplicación de los procesos, se identifican y acuerdan oportunidades de mejora, las cuales se asigna a RGP para su implementación.
7. RGP involucra a los miembros de la organización para incorporar las mejoras.

Condiciones de salida:

Se controlan satisfactoriamente los proyectos, recursos y procesos de la organización utilizando procesos ajustados a las necesidades de la organización.

4.8.2.4 Perfil Avanzado de Negocio para la etapa de Éxito sostenido

Para lograr que la organización pueda dirigir exitosamente su rumbo, es necesario que tenga el control cuantitativo y cualitativo sobre sus procesos y proyectos. Para lograrlo se recomienda, como cuarto y último paso, agregar a las prácticas del Perfil Intermedio las prácticas del proceso Gestión de Negocio.

Descripción

Mejora continua de prácticas y productos. Objetivos cada vez más ambiciosos.

Condiciones de entrada:

- Alta Dirección tiene expectativas de crecimiento con éxito y cuenta con la información sobre el mercado, la competencia y la tecnología.

Pasos a seguir:

1. Alta Dirección y RGP revisan las prácticas del Perfil Avanzado para compararlas con las prácticas seguidas actualmente por la organización. Se identifican las prácticas faltantes y las existentes.
2. RGP discute con la Alta Dirección y con otros miembros que participan de las decisiones del rumbo de la organización, la necesidad de adoptar las prácticas faltantes.
3. RGP coordina la generación del documento del proceso de Gestión de Negocio. Se modifican los documentos de procesos de gestión para conectarlos con el proceso Gestión de Negocio. Se documentan las técnicas elegidas para las prácticas y las plantillas de los productos de trabajo, según las necesidades de la organización. Tienen que participar *todos* los involucrados. Se tiene que aprovechar al máximo las técnicas conocidas y los productos ya generados en la organización. Investigar y capacitar para incorporar nuevos elementos.
4. RGP dirige y vigila la implementación de todos los procesos definidos por la organización, por un período definido en el Plan Estratégico.
5. Al finalizar el período, Alta Dirección, RGP, RGCP y RGR revisan el resultado de la aplicación de los procesos, se valora el cumplimiento del Plan Estratégico y se continúa con la aplicación del siguiente ciclo de todos los procesos de gestión.

Condiciones de salida:

No existe salida definida. La organización realiza la ejecución de procesos, con el objetivo de mejorar continuamente sus prácticas y productos de trabajo en función de los objetivos de negocio, cada vez más ambiciosos, definidos en el *Plan Estratégico*.

4.9 Gestión de Proyectos

Es importante la correcta gestión de los proyectos como elemento clave en la continuidad de las organizaciones. En algunas organizaciones, se considera la administración de proyectos como un mero trámite a realizar para dar al cliente (tanto externo como interno) información del proyecto (plazo de finalización, productos a entregar, etc.), no se la considera como una herramienta **para satisfacer los objetivos marcados por el cliente**.

4.9.1 Proceso Gestión de Cartera de Proyectos

Este proceso tiene como misión fundamental *establecer el enlace entre el nivel estratégico del negocio* (proceso Gestión de Negocio) y *el nivel de gestión operativa* (proceso Administración del proyecto).

En las PyMEs, dado que la consecución de la propia supervivencia tiene mayor visibilidad e inmediatez, la optimización de la eficiencia de la organización cobra especial importancia. Del mismo modo, el proceso Gestión de Cartera de Proyectos, como núcleo de la Planificación y el control operativo del negocio, ocupa una posición clave en las actividades de mejora y transformación de la organización.

4.9.1.1 Responsable de gestión de cartera de proyectos

El Responsable de Gestión de Cartera de Proyecto realiza las acciones necesarias para asegurar que las actividades de la organización, enmarcadas en la Gestión de Cartera de Proyectos, satisfacen los objetivos establecidos en el Plan Estratégico. Esta figura tiene competencia en todos los proyectos dedicados a la venta y además sobre aquellos cuya finalidad es la mejora de la organización.

Es necesario que el Responsable de gestión de cartera de proyectos tenga perfil de gestor, con habilidades de liderazgo y comunicación.

4.9.1.2 El plan de gestión de proyectos

El Plan de Gestión de Proyectos incluye las actividades que se realizan en el contexto del proceso Gestión de Cartera de Proyectos y en el proceso Administración de Proyecto. Este plan es una especie de guía en la que se detallan las actividades que la organización considera necesarias para una gestión eficiente de proyectos. Cada actividad es aconsejable que tenga asociado el responsable de su realización, las salidas (obligatorias u opcionales) y sus destinatarios. Si la organización dispone de guías o procedimientos para alguna actividad, se debe hacer referencia.

La cartera de proyectos, se puede estructurar sencillamente en las categorías: Ventas en firme, Ventas probables, Ventas posibles y Ventas inciertas. Esta agrupación permite ponderar cada una de ellas para proyectar las ventas netas de la organización. La ponderación depende mucho del tipo de empresa e incluso del tipo de proyecto de que se trata.

En el caso que la organización de desarrollo de software trabaje para otros departamentos de una empresa u organización de nivel superior, no es necesario realizar el plan de ventas; no obstante, es aconsejable realizar una valoración de la cartera de proyectos.

El Responsable de Gestión de Cartera de Proyectos tiene como misión hacer compatible la realización de los proyectos de desarrollo de software que permiten a la organización sobrevivir con aquellos proyectos de mejora que servirán para asegurar el futuro de ésta. La presentación de diferentes alternativas a la Alta Dirección debe servir como oportunidad para “vender” la necesidad de realización de este tipo de proyectos que, con frecuencia, son vistos en los niveles operativos de la organización como sobrecargas de trabajo impuestas por la gerencia.

La mayor parte de las pequeñas y medianas organizaciones tienen absorbida la totalidad de su actividad por el día a día. No obstante, toda organización que tenga vocación de permanecer y de crecer tiene que incluir en su estrategia proyectos de mejora de los diferentes procesos de la organización.

4.9.1.3 Planificación de los recursos

La planificación de los recursos se realiza en el Modelo de Referencia a través del Plan de Adquisiciones y Capacitación. El Responsable de Gestión de Cartera de Proyectos tiene como responsabilidad la elección de herramientas para la planificación de recursos en los proyectos, las cuales están basadas en la utilización de estimaciones que proceden del proceso Administración de Proyecto.

Los criterios claves para la elaboración del plan son:

- La determinación de los diferentes perfiles del personal a utilizar. Se tiene en cuenta las necesidades de perfiles no existentes en la actualidad.
- Las necesidades de formación del personal en la utilización de nuevas técnicas.
- La valoración de las adquisiciones de equipos y de posibles subcontrataciones de servicios.

Para las PyMEs, es indispensable diferenciar:

- Los recursos aplicables a ventas en firme, los proyectos en curso pendientes y los proyectos internos.

- Los recursos previstos para las ventas no firmes.
- En el caso de los recursos humanos de cada perfil, se comparará:
 - La cantidad de recursos necesarios para las ventas en firme, respecto a la disponibilidad de recursos actual, obteniendo las necesidades netas actuales.
 - La suma de recursos previstos (ventas firmes y no firmes) respecto a los recursos actualmente disponibles, obteniendo las **necesidades netas potenciales**.

También se debe relacionar los recursos materiales con los plazos de entrega críticos.

En las PyMEs, la adquisición de recursos debe realizarse sobre las necesidades netas actuales. No obstante, la información obtenida de las necesidades potenciales permite: anticipar la preselección de perfiles de personal más crítico, evaluar el mercado para adquisiciones o subcontrataciones potenciales, anticipar posibles necesidades de financiación, las distintas opciones de adquisición de ésta, entre otras.

A medida que las perspectivas evaluadas en la planificación son más precisas o inmediatas, los resultados del plan son mucho más exactos. El ejercicio de la planificación mejora los métodos de ajuste y, además la capacidad de la organización para afrontar las necesidades de sus clientes se incrementa notablemente, característica importante para las PyMEs.

4.9.1.4 Instrumentación interna

Una vez realizado el contrato del proyecto, el Responsable de Proyecto debe efectuar el nombramiento del Administrador del Proyecto. También es el momento de asignar el equipo de trabajo, tarea que es aconsejable que el Administrador del Proyecto y el Responsable de Proyecto realicen en forma conjunta. Una vez realizada la asignación, se elabora el registro del proyecto. Luego todo el personal de la organización debe conocer en qué nuevas actividades se ha comprometido la organización y quiénes son los responsables de llevarlas a buen fin.

La apertura del Registro del Proyecto no es un mero acto burocrático, sino que supone una autorización para utilizar recursos de la organización en actividades determinadas. El registro es importante que contenga algunas metas cuantitativas del proyecto, como: plazo de realización de los hitos más importantes, coste presupuestado de cada fase y tipo de recurso, cantidades máximas de problemas admitidos tras la entrega del producto.

Es importante que todo el equipo del proyecto conozca y se involucre en la consecución de los objetivos establecidos.

Es en el ámbito de la planificación donde se debe asegurar que el conjunto de hitos previstos permiten un adecuado seguimiento y control del proyecto. No se puede esperar a la finalización del proyecto para conocer el grado de desviación en plazo o coste, es necesario anticiparse a este hecho.

4.9.2 Proceso de Administración del Proyecto

La administración del proyecto puede definirse como el proceso que se compone de una serie de actividades, tiene una duración determinada, se realiza con unos recursos humanos y técnicos limitados, con un costo asignado y cuyo fin es cumplir los objetivos marcados por el cliente. (19)

El responsable del proceso se denomina, dentro del proyecto COMPETISOFT, Responsable de Administración de Proyecto.

Entre los factores que conducen al éxito de proyectos se pueden destacar:

- Metas y alcance del proyecto claramente definidos.
- Apoyo de la Alta Dirección.
- Elección de personas competentes del equipo de proyecto y del Responsable de la Administración de Proyecto.
- Asignación de recursos, humanos y materiales, suficientes.
- Comunicación adecuada.
- Mecanismos de seguimiento y control.

Todo proyecto debe pasar por las etapas: Planificación, Realización, Seguimiento y Control y Cierre.

4.9.2.1 Etapa Planificación

Al inicio del proyecto se trata de definir todo aquello que puede ser predecible, intentando reducir al máximo la incertidumbre sobre los aspectos: coste, plazo y funcionalidad del producto. La finalidad de esta etapa es obtener, controlar y mantener el plan de proyecto que regirá al proyecto específico, con base en la descripción establecida del proyecto.

Se elabora el documento *plan de proyecto* y, conjuntamente con el responsable de desarrollo y de otras áreas, establecerán las actividades que hay que realizar para lograr los objetivos marcados y satisfacer las necesidades del cliente.

Algunas actividades a realizar en la etapa de planificación son estimación, recursos humanos y métricas/indicadores, descriptas a continuación:

4.9.2.1.1 Estimación

Se tiene que establecer la estimación del esfuerzo y del tiempo que llevara la realización de las tareas del proyecto. La suma del tiempo de cada una de las actividades permitirá obtener la duración del proyecto y permitirá el cálculo del coste del mismo.

Esta etapa de planificación ha sido previamente elaborada a alto nivel conjuntamente entre el Responsable de Administración del Proyecto y el Responsable de Gestión de Cartera de Proyectos, para poder realizar la oferta a los clientes externos. Posteriormente a la aprobación por el cliente de la oferta, el plan de proyecto se describe con más detalle.

La estimación más sencilla es basarse en proyectos similares teniendo muy presente en esta estimación los recursos humanos y su capacitación. La actividad de estimación es una tarea a repetir según vaya avanzando la información adquirida del proyecto.

4.9.2.1.2 Recursos Humanos

Una de las tareas de esta actividad es asignar las personas que van a participar del proyecto, de acuerdo a su capacitación y experiencia. Si no se dispone de personal suficiente, será necesario realizar un plan de formación de personal.

Es importante que el equipo humano que participa en el proyecto esté informado de sus responsabilidades, de los objetivos del proyecto, los compromisos adquiridos con el cliente y de todo lo relacionado con el proyecto para que el equipo se involucre y esté motivado. Por esto es

fundamental la comunicación durante toda la vida del proyecto como uno de los factores de éxito del mismo.

4.9.2.1.3 Métricas/Indicadores

Los indicadores tienen un papel clave en el seguimiento y control de los proyectos ya que contribuyen a la mejora y a la medición del cumplimiento de objetivos o de la eficacia en el uso de recursos, además proporciona una visión del proceso.

El Modelo de Referencia COMPETISOFT establece tres objetivos para la administración de proyectos:

- **O1** Lograr los objetivos del proyecto en tiempo y costo.
- **O2** Mantener informado al Cliente.
- **O3** Atender las Solicitudes de Cambio.

Una actividad muy importante para las pequeñas y medianas empresas es comenzar a medir, con el fin de disponer de un histórico el cual servirá para poder compararse internamente o con otras organizaciones y poder encontrar oportunidades de mejora.

Es conveniente el uso de herramientas que faciliten la toma de medidas. Los valores de los indicadores que se tomen a lo largo del proyecto deberán ser analizados y realizar su seguimiento con la frecuencia establecida en el plan del proyecto.

4.9.2.1.4 Plan de proyecto

El propósito del plan de proyecto es establecer la estrategia a seguir para cumplir los objetivos del proyecto. El plan contendrá el conjunto de actividades y tareas a realizar, los hitos de control, los miembros del equipo del proyecto, las actividades y tareas que realizarán y el tiempo asignado a las mismas. El plan de proyecto es un documento dinámico a lo largo de la vida del proyecto que ayuda a la consecución de los objetivos. Una vez realizado el plan de proyecto, debe presentarse al Responsable de Gestión de Cartera de Proyectos para su aprobación.

Los proyectos tienen que planificarse, aún los pequeños, pero teniendo en cuenta que tiene que existir proporcionalidad entre el tamaño del proyecto y el esfuerzo de la actividad de planificación.

4.9.2.2 Realización

En esta etapa se llevan a cabo las actividades y tareas marcadas en el plan de proyecto para ser realizadas por el equipo de proyecto.

Dentro de esta etapa se encuentra:

- La realización de las actividades y tareas marcadas para cada grupo de trabajo.
- La realización de las comunicaciones con el cliente y con el equipo de trabajo con el objetivo de estar todos informados sobre el proyecto.
- Verificación de las condiciones de trabajo del equipo.
- La realización del plan de adquisiciones de equipos y capacitación.
- La administración del personal subcontratado y de la subcontratación de partes del proyecto.

- Registro de las mediciones realizadas y análisis de las mismas.
- Registro de los costes reales del proyecto.
- Realización del plan de gestión de la configuración.

Es importante la comunicación entre todo el equipo de proyecto con el Responsable de Administración de Proyecto.

En esta etapa está la base en la que se sustenta todo el desarrollo del proyecto y es el establecimiento de la gestión de configuración del mismo, junto con la gestión de los cambios requeridos por el cliente. Ante la petición de un cambio, el Responsable de Administración de Proyecto debe decidir si incorpora el cambio inmediatamente, lo pospone o lo rechaza debido al impacto en el proyecto. Una buena gestión de configuración y de cambios aumenta las posibilidades de éxito del proyecto.

4.9.2.3 Seguimiento y control del proyecto

Esta etapa consiste en asegurar que se cumplan los objetivos del proyecto y supervisar y evaluar el progreso del mismo, para identificar desviaciones y realizar acciones correctivas y preventivas.

El proyecto debe tener definidos seguimientos periódicos y estos deben consistir en comparar el progreso actual con el planificado, tanto en el cronograma como en costos. De encontrar desviaciones se deben tomar las acciones correctivas pertinentes, levantar una pequeña acta de la reunión de seguimiento (útil al cierre del proyecto) y, si fuera necesario, realizar la modificación del plan y su actualización en la base de datos documental y la comunicación a todo el equipo del proyecto. Las desviaciones se deben comunicar al Responsable de Gestión de Cartera de Proyectos, junto con las posibles acciones correctivas o preventivas. Esto permitirá que la organización apruebe acciones adecuadas para paliar o eliminar las desviaciones.

4.9.2.4 Cierre del proyecto

La última etapa consiste en proporcionar los entregables, de acuerdo al protocolo establecido, hacer una revisión del proyecto global entre el equipo de proyecto y el responsable del mismo y dar por concluido el proyecto.

Las tareas de esta etapa son:

- Aceptación por parte del cliente de los entregables (producto de software, documentación, formación, configuración del producto).
- Cierre de los contratos con los subcontratistas.
- Generación del informe de cierre del proyecto.

El informe de cierre del proyecto deberá contemplar las métricas realizadas en el proyecto, las incidencias más significativas encontradas, el estado de las acciones correctoras y su grado de eficacia, las desviaciones en tiempo, coste, plazo sufridas y las lecciones aprendidas.

Es importante que en el cierre de proyecto este el equipo de trabajo que participó y que pueda dar sus opiniones, dejándolas asentadas en el informe del proyecto como mejoras.

En caso que el proyecto no hubiese llegado a término, se debería realizar el mismo informe de cierre, indicando, agregando las causas del fracaso, con el objetivo de obtener las correspondientes lecciones aprendidas.

Una vez realizado el informe de cierre, el responsable de Administración de Proyecto lo trasladará al Responsable de Gestión de Cartera de Proyectos, quien establecerá los canales adecuados para la difusión de las lecciones aprendidas dentro de la organización y la incorporación de la base de datos de conocimiento.

La base de datos de conocimiento de la organización debe incorporar todos los informes, versiones del plan, actas de reuniones para que sirva de base de conocimientos históricos para otros proyectos.

El informe de cierre contribuirá activamente a la mejora de los planes de proyectos futuros dentro de la organización.

4.9.3 Desarrollo de Software

El propósito del proceso Desarrollo de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos cumpliendo con los requisitos especificados.

El proceso de Desarrollo de Software se compone de uno o más ciclos (iteraciones) de desarrollo. Cada ciclo está compuesto de las siguientes fases:

- **Inicio:** Revisión del *Plan del Proyecto* por los miembros del Equipo de Trabajo para lograr un entendimiento común del proyecto y obtener el compromiso de su realización.
- **Requisitos:** Conjunto de actividades cuya finalidad es obtener la documentación de la Especificación de *Requisitos y Plan de Pruebas de Sistema*, para conseguir un entendimiento común entre el Cliente y el Equipo del Proyecto.
- **Análisis y Diseño:** Ambas fases involucran un conjunto de actividades en las cuales se analizan los requisitos especificados para producir una descripción de la estructura de los *Componentes de software*, la cual servirá de base para la construcción. Involucra la concepción de la arquitectura o diseño de alto nivel y la especificación detallada considerando los lineamientos y decisiones para incluir las características de calidad a través de los requisitos no funcionales del producto. Como resultado se obtiene el documento de *Especificación del Sistema* y el *Plan de Pruebas de Integración*.
- **Construcción:** Conjunto de actividades para producir los Componentes de software que correspondan con la *Especificación del Sistema*, así como la realización de pruebas unitarias. Como resultado se obtienen los Componentes de software probados.
- **Integración:** Conjunto de actividades para integrar y probar los *Componentes* de software, basadas en el *Plan de Pruebas de Integración*, con la finalidad de obtener el *Sistema de Software* que satisfaga la *Especificación del Sistema* establecida. Se revisan los datos de entrada, el control de procesos internos, la integridad de los mensajes, la validación de los datos de salida y protección de los datos de prueba. Como resultado se obtiene el *Sistema de Software* para ser probado.
- **Pruebas:** Conjunto de actividades para probar el *Sistema de Software*, basadas en el *Plan de Pruebas de Sistema*, con la finalidad de obtener el *Sistema de Software* que satisfaga los requisitos especificados. Se genera la versión final del *Manual de Usuario*, *Manual de Operación* y *Manual de Mantenimiento*. Como resultado se obtiene el *Sistema de Software* probado y documentado.
- **Cierre:** Integración final para su entrega de la *Configuración de Software* generada en las fases anteriores. Identificación y documentación de las *Lecciones Aprendidas*. Generación del *Reporte de Mediciones y Sugerencias de Mejora*.

Para generar los productos de cada una de estas fases se realizan las siguientes actividades:

- Distribución de tareas, se asignan las responsabilidades a cada miembro del Equipo de Trabajo de acuerdo al Plan del Proyecto.
- Producción, verificación, validación o prueba de los productos, así como su corrección correspondiente.
- Generación del Reporte de Actividades.

Objetivos

- **O1** Lograr un entendimiento de las necesidades del cliente por el equipo de trabajo y estar de acuerdo con la solución propuesta mediante la ejecución de las actividades de la fase de Requisitos.
- **O2** 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, y mediante el control de cambios de los mismos.
- **O3** Llevar a cabo las actividades de las fases de un ciclo mediante el cumplimiento del *Plan del Proyecto* actual.
- **O4** Garantizar que al final del desarrollo del sistema todos los requisitos funcionales estén trazados a los *Componentes*.
- **O5** Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros mediante la integración de la *Configuración de Software* del ciclo actual.

(5)

5. APORTE DE LA TESINA

En capítulos previos se presentaron definiciones de los tres temas que fundamentales del presente trabajo: el proyecto de software utilizado para el análisis y su proceso metodológico, la metodología ágil Scrum y el proceso de mejora Competisoft.

Tomando como base las definiciones previas, el presente capítulo comienza con dos análisis comparativos:

1. Scrum versus el Proceso Metodológico
2. Scrum versus el Proceso de Mejora Competisoft

El primer análisis se centra en las características con que se realizó la adopción de Scrum en el proyecto real estudiado en este trabajo. Se toman como punto de origen las reglas y prácticas que conforman la definición Scrum y, por cada una, se determina si fue aplicada en forma completa, parcial o si su aplicación no pudo realizarse, detallando los motivos. Este análisis permite conocer, para este proyecto, las dificultades en que se pueden encontrar las empresas PyMEs al momento de adoptar esta metodología ágil como su proceso de desarrollo y cómo podrían ser resueltas.

En tanto, el segundo análisis permite evaluar si es posible aplicar Competisoft a un proceso de desarrollo que utilice la metodología Scrum como guía. Para realizar esta actividad se focaliza sobre las reglas y prácticas definidas en el proceso Operación de Competisoft y se las compara con las reglas y prácticas definidas en Scrum, haciendo un análisis similar al tomar Scrum y compararlo con Competisoft. El objetivo es detectar las coincidencias y diferencias entre ambas definiciones con el fin de considerar la viabilidad de aplicar Competisoft en un proceso que utilice Scrum como metodología de trabajo.

Contando con estos análisis comparativos y sus conclusiones, y tomando la definición de Scrum como punto de contacto entre ellos, se procederá a conectar Competisoft con el proceso metodológico del proyecto real con el fin de establecer las prácticas de Competisoft a implementar. Con estas prácticas detectadas es posible realizar la evaluación necesaria para determinar las sugerencias a Competisoft para que se convierta en un modelo de mejora que pueda ser adoptado en procesos ágiles, en particular Scrum, en empresas PyMEs.

5.1 Adaptación del Proceso Actual a Scrum

En el *capítulo 2: Situación actual* se realiza una descripción detallada del proyecto de software y se describe el proceso de desarrollo que guió al equipo de trabajo en la realización del mismo. Esta descripción nos plantea parte de la problemática a resolver con la implementación de la metodología ágil Scrum y presenta un escenario real en el cual fueron aplicadas las reglas y prácticas definidas por Scrum, facilitando la fundamentación del análisis de su adopción.

Tomando las reglas y prácticas definidas por Scrum y comparándolas con las implementadas por el proceso de desarrollo, se puede armar un mapa, el cual llamaremos "*Comparativa Scrum vs. Proceso Metodológico*", clasificando la aplicación de las reglas como: *Completa, Parcial o No aplicada*. Por cada regla aplicada en forma total o parcial, se especificará el modo en que esta aplicada. En otro caso, se fundamentará la razón de su no aplicación.

5.1.1 Reglas de Scrum aplicadas en forma completa

5.1.1.1 Lista de requerimientos priorizado (Product backlog)

En el momento en que se implementa Scrum, se genera formalmente la lista de requerimientos del proyecto, únicamente para aquellos requerimientos pendientes de definición.

La manera de priorizar cada requerimiento se basa inicialmente en una valoración descriptiva realizada por el cliente con la cual describe la criticidad del requerimiento, por ejemplo: *“Este punto tiene que estar lo antes posible”* o *“Este punto no es tan importante, dejémoslo para el final”* o *“Este punto tiene que estar antes que tal otro punto”*. En algunos casos el nivel crítico, impuesto por el cliente, está dado por la necesidad de implementar los requerimientos en una fecha límite, en esta situación es necesaria e imprescindible una priorización más específica por parte del cliente, además de un compromiso mayor por parte del equipo de trabajo.

Con esta valoración inicial, el responsable de recibir los requerimientos y el resto del equipo construyen la lista de requerimientos junto con su priorización, sumando diversos criterios para aquellos requerimientos sin valoración, entre los cuales se puede detallar:

- dependencia con otros requerimientos
- si la funcionalidad asociada es visible y puede ser de importancia para los usuarios (por ejemplo, un reporte de auditoría)
- dificultad, esfuerzo y recursos necesarios para llevar el requerimiento a cabo

Una vez completada la priorización, se define un cronograma *tentativo* con el fin de presentar al cliente fechas de entrega de cada requerimiento. El cliente realizará las modificaciones que crea necesario.

Si el cliente envía nuevos requerimientos, modificaciones o mejoras, de acuerdo a la valoración descriptiva de los mismos aportada por él, será necesario modificar el cronograma tentativo de entregas posteriores a la iteración en curso. Así se podrá incluir en forma ordenada los nuevos requerimientos enviados por el cliente.

Además de los requerimientos solicitados por el usuario, el equipo de trabajo puede agregar a la lista de requerimientos mejoras, tanto de performance como de usabilidad para el usuario, con el propósito de darle al aplicativo el mayor grado posible de calidad.

El equipo utiliza dos terminologías diferentes para diferenciar estos requerimientos: aquellos solicitados por el cliente los denomina “Historia de usuario”, y para los generados dentro del equipo se los conoce como “Deuda técnica”, aunque un nombre más representativo podría ser “Historia técnica”.

5.1.1.2 Iteración (Sprint)

En el proceso metodológico, utilizado para el desarrollo del producto de software, la duración estándar de cada sprint se define de 1 semana (o 5 días hábiles). Su principal objetivo es crear una interacción fluida con el cliente y fortalecer la relación con él, brindándole funcionalidad que considera crítica y prioritaria en forma rápida y recibiendo nueva funcionalidad, ajuste o aprobación de manera eficiente.

Desde el punto de vista del equipo de trabajo, al no tener experiencia en Scrum, ayuda a focalizar en la labor a realizar con mayor concentración y responsabilidad, se mejora en la planificación logrando ser más exacta a medida que se avanza en el tiempo y el equipo adquiere conocimiento.

En cuanto al producto de software resultante, al realizar entregas funcionalidad productiva en forma temprana y constante, el período de estabilización y puesta a punto también comienza temprano en la vida del proyecto, optimizando el tiempo total del proyecto y mejorando la inversión del cliente en el mismo.

Cabe aclarar que esta duración se amplía en caso de ser necesario, dependiendo de la funcionalidad a incluir.

Cada iteración comienza un lunes y finaliza el viernes siguiente. El primer día comienza con la planificación, la cual tiene una duración promedio de 3 horas.

El viernes se realizan las siguientes tareas correspondientes al cierre del sprint:

- generación del entregable para el cliente, el cual contiene un incremento de funcionalidad,
- instalación del entregable en el ambiente del Cliente,
- el equipo realiza la Retrospectiva de la iteración.

5.1.1.3 Lista de requerimientos de la iteración (Sprint backlog)

La lista de requerimientos de la iteración se completa durante la reunión de planificación, con aquellos requerimientos (“Historias de usuario” e “Historias técnicas”) que fueron pre-seleccionados para ser incluidos en el desarrollo de la iteración en curso. Estos requerimientos conformarán el incremento de funcionalidad que será entregado al cliente al finalizar la iteración.

La lista de requerimientos de la iteración se mantiene en un tablero, el cual se encuentra visible para todo el equipo durante toda la duración de la iteración. Además de los requerimientos, en el tablero se agregan las tareas que son necesarias realizar para completar cada uno de esos requerimientos.

En Scrum, el tablero utilizado dispone, en general, de cuatro columnas para representar los estados por los que transita una tarea, estas columnas son “Hacer”, “En progreso”, “Verificar” o “Testear” y “Listo”. En la adaptación a Scrum, la columna “Testear” se subdivide en dos sub-columnas con el fin de distinguir las tareas con el desarrollo terminado de aquellas que se encuentran en la versión instalada en el ambiente de QA.

El detalle de cada columna es el siguiente:

- **Primera columna “Hacer”:** también conocida como **Lista de requerimientos de la Iteración**, contiene las historias de usuario (requerimientos solicitados por el cliente o mejoras introducidas por el equipo) que se seleccionaron en la planificación de la iteración actual. Todos los ítems se toman de la Lista de requerimientos priorizada del Proyecto.
- **Segunda columna “En progreso”:** al comenzar el **desarrollo**, cada miembro del equipo elige una o más tareas a desarrollar, escribe su nombre y las coloca en la columna En progreso, con el fin de indicar que es el responsable de llevar a cabo la tarea.
- **Tercera columna “Prueba-Desa”:** una vez que el responsable de una tarea la habilita para ser liberada en la próxima versión de la aplicación a instalar en el ambiente de QA al ambiente QA, mueve la misma a esta columna.
- **Cuarta columna “Prueba-QA”:** al hacer un nuevo build al ambiente QA, las tareas que se encuentren en la columna Prueba-Desa pasan a la columna Prueba-QA, de esta

forma el responsable de las pruebas de verificación podrá comenzar a realizar las pruebas necesarias en esas tareas.

- **Quinta columna “Listo”:** aquellas tareas que fueron probadas en el ambiente QA y cumplen con las reglas de verificación definidas para la historia de usuario, se consideran cerradas y finalizadas, pasándolas a esta última columna. Una historia de usuario está finalizada cuando todas sus tareas lo están.

5.1.1.4 Planificación de la iteración (Sprint Planning meeting)

Al comienzo de cada iteración el equipo de trabajo se reúne para definir, discutir y seleccionar los requerimientos priorizados que formarán parte del próximo incremento de funcionalidad. Esta reunión es conocida en Scrum como Reunión de Planificación de la iteración.

Tomando como punto de partida el cronograma de entregas tentativo, el encargado de la relación funcional con el cliente presenta al resto del equipo los requerimientos que fueron pre-seleccionados para la iteración actual. Luego de la descripción funcional, el equipo le realiza todas las preguntas necesarias para entender “lo que hay que desarrollar”, pasando luego al análisis técnico de su implementación; en este análisis puede participar, a pedido del equipo, algún referente técnico de la empresa.

Luego de entenderlos requerimientos pre-seleccionados para la iteración, tanto de manera funcional como técnica, el equipo cuenta con más información para determinar su complejidad y así evaluar si podrá ser desarrollado o no durante la iteración en curso. Para esta actividad se utiliza un método conocido como cartas de Planificación (Planning PókerTM ¹³):

1. Cada integrante del equipo cuenta con un mazo de cartas con la numeración 0, 1, 2, 3, 5, 8, 13 y 20.
2. Tomando un requerimiento por vez, cada integrante elige una carta que represente la complejidad entendida del mismo (0: no es complejo, 20: es muy complejo).
3. Una vez que cada “estimador” eligió una carta, todas las cartas son mostradas al resto al mismo tiempo. Si todas las cartas son iguales, esa es la estimación.
4. Si la estimación no es la misma, el equipo discute las estimaciones, haciendo foco especialmente en los valores más bajo y más alto, de forma de “afinar” el análisis, tanto técnico como funcional.
5. Se repiten los pasos hasta que la estimación de todos converge a la misma carta.
6. Una vez que se votaron todos los requerimientos, se suman sus puntajes. Tomando en cuenta las iteraciones previas, se saca un promedio del puntaje total de las historias incluidas en cada planificación, de esta forma, el equipo selecciona, para la iteración actual, las historias cuya suma de puntaje sea cercano al puntaje promedio.

Se eligen los requerimientos que se desarrollaran en la iteración actual teniendo en cuenta: prioridad, velocidad del equipo (cantidad de puntos que se hace por iteración, varía entre 20 y 25 puntos), disponibilidad de los miembros del equipo (por ejemplo: integrantes que se encuentren de vacaciones, o pedidos por día de examen, licencia médica, etc.)

Una vez que se tienen los requerimientos que serán desarrollados, se pasa a escribir las tareas a realizar por cada uno.

¹³ Planning PokerTM es marca registrada de Mountain Goat Software^R (2007 Mountain Goat Software)

5.1.1.5 Reunión de Retrospectiva de la iteración (Sprint Retrospective meeting)

Para realizar la retrospectiva de la labor realizada, al finalizar la iteración, el equipo de trabajo utiliza el método conocido como “Tres columnas: Lo Bueno, Lo Malo, A Mejorar”.

Al comienzo de cada retrospectiva se revisa la anterior con el fin de chequear si las soluciones propuestas se realizaron y evaluar el resultado obtenido, o explicar las razones que motivaron que no realizara.

Posteriormente, y con un tiempo acotado, cada participante describe por escrito lo acontecido durante la iteración según su percepción positiva o negativa, ubicando cada papel escrito en la columna “Lo Bueno” o “Lo Malo”, respectivamente. Si bien se evalúa lo sucedido en el proyecto, los participantes pueden anotar sus percepciones a nivel de la organización; para estos casos se agregará a la retrospectiva algún integrante de la dirección de la empresa o se le notificará de alguna manera lo sucedido, si no estuviera disponible.

La consigna consiste en que todos los presentes en la retrospectiva participen aportando sus impresiones de lo acontecido, teniendo presente que durante el intervalo de tiempo en que se escriben las percepciones no se puede juzgar, discutir ni cuestionar lo que escriben los participantes, esto es conocido como “brainstorming”.

Una vez finalizado este intervalo se inicia uno nuevo para discutir acerca de lo que cada participante aportó. Para ello, se lee en voz alta lo escrito y se explica profundizando su percepción. Todo esto es debatido y cada miembro aporta su impresión.

La siguiente actividad en esta reunión es la elección de lo negativo que se aportó, en la columna “Lo malo”, para ejecutar tareas que faciliten su solución durante la siguiente iteración. Para esto, cada participante tiene 3 votos para asignar entre los “papeles de la columna Lo Malo” que vislumbra como candidatos a ser resueltos durante la siguiente iteración.

Entre los más votados, el equipo elige 2 o 3 ítems que considera más importantes o prioritarios a resolver, se llevan a la columna “A Mejorar” junto con un resumen del debate realizado en el intervalo anterior con el fin de buscar soluciones. El equipo se compromete entonces a colaborar en la resolución de estos puntos elegidos.

5.1.1.6 Equipo (Team)

Dentro de los roles propuestos por la metodología, se encuentra el Equipo. Si bien la metodología Scrum no detalla los roles a cubrir dentro del Equipo (dado que el objetivo del mismo es lograr ser multidisciplinario cubriendo todas las tareas involucradas en el desarrollo de un producto de software), en el proyecto real se pueden encontrar los siguientes roles:

- Analista funcional (senior).
- Desarrollador .Net (semisenior).
- Tester.
- Encargado de la comunicación con el cliente y los usuarios (recibe los requerimientos nuevos, las consultas o errores. Informa al cliente y los usuarios cuando hay una nueva versión de la aplicación y cuál es la funcionalidad agregada, etc.).
- Encargado de llevar el backlog del proyecto (ordena los requerimientos según prioridades, informa al equipo el detalle de una historia de usuario, agrega o saca historias de usuario del backlog -según lo que indique el cliente-).

- Cliente (está comunicado con el equipo, envía nuevos requerimientos -generados por él o solicitados/sugeridos por los usuarios- para ser incluidos en el backlog. Aunque no se encuentre presente físicamente en las reuniones, se debe mantener un contacto fluido mediante reuniones quincenales o mensuales, mails, comunicación telefónica, etc.).

Si bien el equipo de trabajo sufrió modificaciones desde su conformación, gran parte del mismo se mantuvo estable, permitiendo así crear una sinergia entre los integrantes al nutrirse de las diferentes experiencias, conocimientos y habilidades de los miembros, trabajando colaborativamente. Así, con la ejecución de ciertas prácticas de Scrum (como la Retrospectiva o Planificación) el equipo logra un nivel de madurez alto, lo que se percibe en su destreza para la auto-gestión, en la maximización de su capacidad para resolver problemas, en la aptitud de realizar conversaciones cara a cara con el consiguiente aumento de empatía y confianza entre sus miembros.

5.1.2 Reglas de Scrum aplicadas en forma parcial

Si bien la mayoría de las prácticas o reglas definidas por Scrum han sido adoptadas en el desarrollo del proyecto, existen algunas donde su implantación difiere de la forma en que Scrum lo plantea. A continuación se detallan dichos casos.

5.1.2.1 Dueño del Producto (Product Owner)

De acuerdo a la definición de Scrum, el Dueño del Producto es responsable de velar por los intereses de otros clientes y usuarios, maximizar el retorno de la inversión, planificar entregas, crear y mantener actualizada la lista de requerimientos y priorizarla. En definitiva, el dueño del producto cumple el rol de ser el representante del cliente en el equipo.

Este rol puede ser “interpretado” por más de una persona. Dentro del equipo de trabajo, lo interpreta aquella persona que se hace cargo de la comunicación con el cliente. Esta persona es quien mantiene y prioriza la lista de requerimientos y es quien detalla cada requerimiento al equipo en cada planificación.

Sin embargo, no es la persona encargada de maximizar el retorno de la inversión ni de financiar el proyecto; estas tareas las lleva a cabo el Cliente, quien contrató a la empresa Tercer Planeta S.A. (en el caso práctico analizado) como proveedor del servicio.

La persona del equipo que realiza algunas tareas de este rol, al no poseer el *know-how* completo del negocio, relacionado con el proyecto, no puede hacerse cargo de la toma de decisiones críticas en cuanto a la funcionalidad de requerimientos específicos; para esto necesita la colaboración del cliente, o de algún usuario clave que él asigne.

5.1.2.2 Facilitador de Scrum (Scrum Master)

Una dificultad que tuvo que ser superada al momento de adoptar Scrum, fue el hecho de no contar dentro de la Empresa con un integrante que actúe como facilitador en forma exclusiva. Esta situación se debe a la cantidad de personas de la empresa y su asignación a los distintos proyectos en curso, sumado a la falta de capacitación en este rol.

Por lo tanto, las responsabilidades fundamentales de este rol fueron asumidas por el equipo en su conjunto. Con esfuerzo y buena comunicación, el equipo fue logrando encontrar una dinámica que le permitió resolver obstáculos, respetar ciertas reglas de Scrum, logrando auto-gestionarse. Esto es, en definitiva, uno de los objetivos de la metodología ágil Scrum.

Se debe resaltar que la ausencia de una única persona que asuma este rol se hace evidente en ciertas situaciones, como por ejemplo algún agente externo que entorpezca el normal desarrollo de la iteración, la implementación o cumplimiento de algunas reglas o prácticas de Scrum.

5.1.2.3 Reunión diaria (Daily Scrum meeting)

Al inicio de la adopción de Scrum, el equipo lleva a cabo la reunión diaria con el fin de actualizar el estado del proyecto e informar su propio avance. La reunión se realiza al comienzo del día laboral. Si bien no se responde rigurosamente a las tres preguntas definidas por Scrum, cada uno de los integrantes comenta lo necesario para responderlas:

- Cuáles son las tareas que están listas para ser incluidas en el build al ambiente QA?
- Cuales tareas aún no se terminaron, cuál es el estado y si hace falta alguna definición o a alguien externo para ayudar a resolverla?
- Se evalúa si se hace un build al ambiente QA o si se espera la finalización de otra/s tarea/s.
- Se actualiza el estado de las pruebas: errores detectados o corregidos, haciendo hincapié en aquellos críticos y que pudieran obstaculizar el normal desarrollo de las pruebas o que perjudiquen el funcionamiento correcto de la aplicación.
- al acercarse el fin de la iteración se evalúa si se logra el objetivo; en caso que la evaluación sea negativa, se eligen los requerimientos para sacar de la lista de requerimientos de la iteración (serán incluidos en la lista de requerimientos de la próxima iteración).

Este "Reporte de estado" se puede realizar en otro momento del día.

Luego de un tiempo de comenzado el proyecto, cercano a su finalización y a la necesidad de asignar los recursos a otros proyectos de la empresa, se redujo la cantidad de personas asignadas al mismo. Por esta razón y, además porque el equipo compartía el mismo espacio físico, derivó en que la reunión diaria fue reemplazada por la actualización en tiempo real del estado de cada integrante; es decir, durante la jornada, en el momento de detectar un error durante las pruebas de verificación y/o validación, o de finalización de alguna tarea para su posterior prueba, etc., se informa al resto del equipo, permitiendo una sincronización del trabajo más dinámica y rápida.

Por lo tanto, y a pesar de no respetar el ritual de la reunión diaria (con la contestación de las tres preguntas) el equipo mantiene la sincronización.

5.1.3 Reglas de Scrum no aplicadas

5.1.3.1 Reunión de revisión (Sprint review meeting)

Dentro de la definición formal de Scrum se puede señalar, como una práctica no aplicada, a la reunión de demostración del incremento del producto al Cliente.

Al finalizar la iteración, la entrega de la nueva funcionalidad al cliente consiste en instalar la nueva versión de la aplicación –en los servidores del cliente en forma remota- y el envío de un mail detallando la lista de nuevas funcionalidades, mejoras o corrección de errores que forman parte del incremento del producto.

En algunas ocasiones se coordina una reunión post-instalación para revisar los puntos entregados y también la lista de puntos pendientes para una priorización; pero esto no se realiza rutinariamente al finalizar todas las iteraciones.

El cliente, si bien participa del proyecto, no lo hace en forma activa: no se encuentra presente en las reuniones de planificación ni en la reunión de demostración del producto resultante de la iteración. Esto se debe, entre otros puntos, a la confianza depositada, tanto en el equipo de trabajo como en la empresa, para llevar adelante el proyecto (para el caso de estudio analizado); a la falta de disponibilidad del cliente para reservar tiempo para esta reunión cada 1 o 2 semanas; y también a la falta de concientización, por parte de la empresa y del equipo, de la importancia de su participación activa en el proceso de desarrollo, solicitándole su presencia o la designación de un usuario clave.

5.1.3.2 Tiempo determinado (Timeboxing)

Cada reunión que se lleva a cabo en la formalidad de Scrum tiene una duración determinada, a esto se lo conoce como "Timeboxing".

En el proceso de desarrollo aplicado por el equipo de trabajo, para la reunión de planificación no se establece una duración específica y determinada de tiempo, sino que finaliza cuando el equipo termina de revisar la lista de requerimientos de la iteración, los cuales fueron previamente seleccionados en la reunión de planificación de la lista completa de requerimientos.

Si bien en la reunión de planificación de la iteración se pueden revisar otros requerimientos pendientes, los mismos pueden ser revisados rápidamente por el equipo.

5.1.3.3 Moderador en la Retrospectiva

La persona encargada de dirigir la reunión de Retrospectiva es el Facilitador (Scrum Master), según la definición de Scrum, quien no participa activamente de la reunión, permanece neutral en las discusiones, siendo su principal tarea en esta reunión la de evitar discusiones y lograr que el equipo pueda comunicarse y dialogar con tranquilidad, con la participación de todos.

Como se explicó con anterioridad, el rol de Facilitador en el proyecto no está desempeñado por una sola persona sino que sus responsabilidades son llevadas a cabo por el equipo de trabajo; es por esto que en la reunión de Retrospectiva no se cuenta con la figura de moderador que se encargue de dirigirla y de ser neutral. Es responsabilidad del equipo que todos sus miembros participen, que la comunicación sea fluida y sin roces ni discusiones.

En pocas ocasiones se ha evidenciado la falta de esta figura, especialmente cuando en la retrospectiva surgen temas delicados para discutir, en los cuales hay varias opiniones diversas y contrarias. Pero, gracias a la madurez lograda por el equipo, se puede resolver cualquier conflicto o discusión, no existe roces entre los integrantes.

5.1.3.4 Métricas

Debido a la forma de contratación de Tercer Planeta con FoxChannel –se realiza un contrato por 6 (seis) meses, renovable- no existe por parte de la empresa PyME la evaluación del Retorno de la Inversión (ROI) con respecto al proyecto, para evaluar si es rentable o no.

En cada iteración, de acuerdo a los requerimientos planificados y su avance en el desarrollo, se puede obtener un gráfico conocido como "*Burndown chart*", con él se visualiza rápidamente la evolución de la iteración y con un grafico el equipo puede decidir si se tiene que realizar algún ajuste a la planificación de la iteración para alcanzar el objetivo propuesto. Este grafico no es

utilizado por el equipo, para tomar decisiones acerca de los ajustes a realizar, los integrantes confían en su experiencia y sentido común.

Tampoco se lleva un registro de los errores detectados y resueltos, tanto durante las pruebas de verificación y validación, como durante el uso del producto en el ambiente de Producción. Esta medición permitiría detectar los puntos más débiles del ciclo de la iteración y la estabilización del producto en Producción, con el fin de armar un mapa de los puntos de falla para reforzarlos, definir acciones a tomar y así bajar la tasa de errores. De esta forma, se podría lograr un producto de alta calidad con bajo esfuerzo, cubriendo los puntos de falla detectados.

5.1.4 Conclusiones

Luego de experimentar durante cierto tiempo con Scrum es posible destacar a su definición como la más conveniente para este proyecto: define reglas y prácticas con claridad -lo cual facilitó su adopción-, y abarca la administración de un proyecto completo sin exigir una documentación exhaustiva y delegando ciertas tareas al equipo de desarrollo -por ejemplo, cada integrante es responsable de asignarse tareas a realizar durante la iteración-, lo cual se adapta a la "fisonomía" de la empresa.

En el proyecto bajo estudio, la adopción de Scrum se decidió a partir de la necesidad de resolver problemas de comunicación dentro del equipo y la administración de los requerimientos pendientes del proyecto. Las prácticas de Scrum que sin duda tuvieron que ser aplicadas, para que la adopción fuera satisfactoria, fueron las reuniones de Retrospectiva, Diaria y Planificación. Ambas mejoraron la comunicación del equipo y su conocimiento del proyecto: la primera focaliza principalmente en la integración de todos los miembros, y las otras posibilitan la participación de todos los integrantes y el entendimiento en el proyecto.

A pesar que no se aplicaron estrictamente todas las reglas, según lo especificado por Scrum, fue esta la metodología que adoptó el grupo para llevar a cabo el proceso de desarrollo del proyecto bajo estudio. Esto se debe a que el equipo pudo tener una mayor comprensión de cada una de las reglas, ajustándolas a la situación del grupo así como también de la empresa, para que el proceso se acerque lo más posible a la definición formal. La adopción de Scrum le permitió al equipo resolver, con trabajo y compromiso, las dificultades que existían y les permitió asentarse en sus tareas, crecer profesionalmente y madurar como grupo.

El logro en la adopción de la metodología se hace evidente principalmente en el equipo de trabajo, ya que pudo alcanzar uno de los objetivos más importantes de Scrum, y las metodologías ágiles: contar con un equipo motivado, multidisciplinario y auto-gestionado; además de generar un producto con calidad y con valor de negocio para el cliente y los usuarios.

Es importante el apoyo y compromiso de la empresa, ya que a nivel organizacional es necesario contar con pautas claras y precisas para ejecutar la adopción de Scrum, especialmente si no se cuenta con los recursos suficientes para cubrir todos los roles que se definen, o si falta una capacitación profunda en la metodología. Además, es fundamental que los participantes del proyecto participen activamente de todas las "rutinas" que se proponen en Scrum, convenciéndose, durante el proceso y la ejecución constante de las mismas, de su relevancia, así se logra adoptarlo en forma exitosa y, fundamentalmente, se obtiene un equipo que trabaja como tal.

5.2 Adaptación Scrum a Competisoft

El presente análisis tiene como fin comparar las definiciones de Scrum y COMPETISOFT – descritas en capítulos previos- con el fin de encontrar ítems de coincidencia y diferencia entre ambas definiciones de procesos.

Para realizar este análisis, de ambas definiciones se consideran aquellas prácticas y reglas que faciliten la gestión del desarrollo de un proyecto de software y al equipo involucrado en el mismo. En Scrum implica tomar su definición en forma completa: todas sus prácticas y reglas se comparan con Competisoft, para determinar cuales están cubiertas y cuáles no, además de indicar alguna consideración importante a considerar.

En cuanto a COMPETISOFT, para elegir el o los procesos a comparar en el análisis, y a su vez de estos las prácticas adecuadas, se tiene en cuenta las características del proyecto y de la organización. Es por este motivo que el *Perfil Base de Proyecto* resulta el más adecuado a considerar en este análisis. Se presenta como el *mínimo perfil* a implementar en una organización que cuenta con un proyecto en realización, por lo cual necesita cumplir con el alcance, tiempo y costos del proyecto, entre otros objetivos.

El *Perfil Base de Proyecto* contempla la implementación de prácticas y actividades de los procesos *Administración de Proyecto* y *Desarrollo de Software* de la categoría Operación, definidos en el Modelo de Referencia de procesos del proyecto COMPETISOFT.

Tanto Scrum como COMPETISOFT (en su estrategia de implementación del modelo de procesos) proponen un marco metodológico de trabajo que pueda ser adoptado por un equipo de producción en el desarrollo de un producto de software. Si bien ambos buscan construir un producto de software que cumpla con las necesidades del cliente, difieren levemente en el foco esencial: Scrum se centra principalmente en el cliente, busca conseguir su satisfacción al contar con un producto que cubra las necesidades de negocio del cliente, colaborando en el logro del retorno de la inversión exitoso; COMPETISOFT propone un modelo donde la organización es el centro, focalizando en sus objetivos de negocio para lograr su buen funcionamiento y estabilidad económica, pero sin perder de vista la importancia que el cliente y su satisfacción tienen para el crecimiento y estabilidad de una organización. Esto facilita la adopción de COMPETISOFT como un complemento a Scrum, con el fin de mejorar las prácticas, reglas y actividades de la organización para, de esta forma, aportar más herramientas a la hora de incrementar su productividad y competitividad, proponiéndose como una opción de calidad en el mercado informático.

5.2.1 Metodologías Ágiles y COMPETISOFT

El análisis comienza teniendo en cuenta las bases de las definiciones de las Metodologías Ágiles y del proyecto COMPETISOFT. Las Metodologías ágiles surgen a partir de una serie de *Principios*, los cuales dieron lugar al *Manifiesto Ágil*, guía fundamental en la definición de diferentes metodologías, y en particular de *Scrum*. Por otro lado, el modelo COMPETISOFT fue elaborado de acuerdo a un conjunto de *Premisas* que condujeron la generación del modelo.

Las Metodologías ágiles y COMPETISOFT coinciden en destacar la importancia de:

- *Recursos humanos*:

Las prácticas y actividades están orientadas a lograr un ambiente de trabajo en el cual los individuos involucrados en el proyecto se sientan motivados, con un buen flujo de comunicación e información, comprometidos con el proyecto y con la organización. De esta forma se reconoce a los individuos como el activo principal con que cuenta una organización (1).

- *Simplicidad y facilidad en su adopción:*
Prácticas, reglas y actividades claras, de fácil adopción.
- *Contribuir a alcanzar los objetivos del negocio:*
La definición de ambos promueve que el equipo de trabajo concrete el logro de los objetivos, del cliente y de la organización.
- *Mejora continua:*
Facilitan la búsqueda de la excelencia tanto en el proceso de desarrollo de software como en el proyecto gestionado por él, como del equipo de trabajo que lo lleva a cabo.

5.2.2 Scrum y COMPETISOFT

5.2.2.1 La Organización

Para considerar la adopción de Scrum, es preciso que la organización cumpla con ciertos requisitos, cuya responsabilidad recae sobre la dirección gerencial de la misma. En este sentido, COMPETISOFT comparte algunos requisitos detallados en la Categoría Alta Dirección de la definición del Modelo de Proceso. Si bien esta categoría no forma parte formal de la definición del *Perfil Base*, los requisitos son implícitos a una organización que desee adoptar la implementación de COMPETISOFT. (19)

Scrum y COMPETISOFT coinciden en que la Organización tiene que:

- Considerar al trabajo en equipo, la creatividad y la mejora continua como parte de su cultura.
- Comprometerse a resolver problemas endémicos y realizar los cambios organizativos que se consideren necesarios.
- Fomentar una cultura de gestión basada en la colaboración y en la facilitación de los miembros del equipo.
- Crear un entorno de trabajo óptimo para el trabajo en equipo.

5.2.2.2 El proceso

El eje sobre el que se realiza el análisis comparativo de las prácticas y reglas, entre Scrum y COMPETISOFT, se centra en la definición de Scrum, relacionándola con la definición del *Perfil Base de Proyecto* de COMPETISOFT.

Dueño del Producto

En Scrum este rol representa a todos los usuarios del proyecto ante el equipo, es el referente de las funciones del negocio sobre el que se realiza el proyecto, y está disponible para cualquier consulta. Sus principales responsabilidades son: mantener la lista de requerimientos del proyecto actualizada, establecer las prioridades de los requerimientos y explicarlos, y maximizar el retorno de la inversión, además de validar que el incremento funcional del producto cumpla con los requerimientos planificados. Participa activamente del proyecto (concorre a las reuniones de Planificación de la iteración y de Demo, al finalizar la misma). (6)

En los procesos *Administración del Proyecto* y *Desarrollo de Software* se define el rol *Cliente* quien realiza actividades similares al Dueño del Producto y tiene las mismas responsabilidades.

No obstante, no se manifiesta en la definición su participación activa en cada ciclo, ni su responsabilidad en maximizar el ROI.

La diferencia visible entre ambos reside en “el origen” de este representante: mientras que en Scrum el Dueño del Producto es el encargado de financiar el proyecto, pero no necesariamente forma parte de su staff, el *Cliente* de COMPETISOFT es “la persona u organización que financia el producto y usualmente (pero no necesariamente) decide los requisitos. El cliente es la persona u organización que desea usar el servicio, producto o resultado del proyecto.” (5) Esto que significa que pertenece al staff de aquel que financia el proyecto.

Además, COMPETISOFT define el rol *Usuario*, como la persona que va a utilizar el producto de software y solicita requerimientos directamente al equipo. (5) En contraposición, en Scrum los usuarios que tengan necesidades funcionales las elevan al Dueño del Producto, quien las informa al equipo como un nuevo requerimiento.

Tanto Scrum como COMPETISOFT hacen hincapié en la importancia de una *buena comunicación* entre el equipo de desarrollo y los clientes y usuarios del producto de software, así como mantenerlos informados del avance del proyecto.

Facilitador de Scrum

Algunos roles que están involucrados en actividades de los procesos *Administración del Proyecto y Desarrollo de Software* cubren ciertas responsabilidades del Facilitador de Scrum, no hay un único rol en COMPETISOFT que se corresponda con el Facilitador. Los roles, que pueden desempeñar actividades que en forma conjunta resuelvan algunas de las actividades del Facilitador de Scrum, son:

- Responsable de la Gestión de Proyectos.
- Responsable de la Administración del Proyecto.
- Responsable de Desarrollo de Software.

Sin embargo, la responsabilidad primordial del Facilitador de Scrum es: “velar por la adopción de Scrum” no se encuentra especificada explícitamente para ningún rol en COMPETISOFT, así como tampoco la “resolución de los obstáculos que tenga el equipo para alcanzar los objetivos de la iteración”.

Equipo

En este rol están involucradas las personas designadas a participar del proyecto como responsables de la construcción del Producto de Software. Scrum apunta a tener equipos multidisciplinarios, pero no detalla los roles que tiene que ser cubierto en el equipo, sino que da importancia a que los integrantes se comprometan a construir el producto de software acordado durante la iteración. De esta forma, el equipo tiene que ser capaz de *auto-gestionarse*, lo que significa que cada integrante elija la tarea a realizar (no necesariamente será de acuerdo a su perfil) cuidando en que todos los integrantes tengan una tarea y que todas las tareas asociadas a la construcción del producto estén cubiertas. (6)

En COMPETISOFT, en el proceso Desarrollo de Software se establecen detalladamente los roles dentro del equipo de trabajo, a quienes el Responsable de Desarrollo de Software y el Analista se encargan de distribuir las tareas de acuerdo al rol. Cabe aclarar que un rol puede ser desempeñado por una o más personas.

La diferencia, en cuanto a la forma de asignar las tareas al equipo, no inhabilita la adaptación de COMPETISOFT a Scrum. (5)

Iteración (Sprint)

Para Scrum una iteración es un período de tiempo fijo, conformado por las etapas: Planificación-Desarrollo/Pruebas-Incremento/Integración-Demo-Retrospectiva.

En el proceso *Desarrollo de software* de la Categoría Operación del Modelo de Procesos de COMPETISOFT puede encontrarse la definición de un *Ciclo de Desarrollo*, el cual se divide en diferentes etapas: *Inicio-Requisitos-Análisis y Diseño-Construcción-Integración-Pruebas-Cierre*.

Cada etapa del ciclo COMPETISOFT puede corresponderse con una etapa de la iteración de Scrum, como puede observarse en la siguiente tabla:

Etapas de una Iteración	Etapas de un Ciclo
Planificación	Inicio-Requisitos-Análisis-Diseño
Desarrollo/Pruebas	Construcción
Incremento/Integración	Integración-Pruebas
Demo/Retrospectiva	Cierre

Lista de Requerimientos del Proyecto

De acuerdo a la definición de Scrum, esta lista contiene la totalidad de requerimientos del proyecto, los cuales pudieron haber sido solicitados por el dueño del producto o ingresados por el equipo de trabajo, involucrado en el proyecto, en forma de mejoras técnicas o corrección de errores (postergado de la iteración anterior previa evaluación de riesgo en cuanto a la funcionalidad del producto de software).

Cada requerimiento que conforma la lista de requerimientos tiene un estado asociado el cual indica su lugar en la vida del proyecto (Pendiente, En Proceso –fue incluido en la iteración en curso-, Terminado –forma parte del producto de software-) y una prioridad asignada por el dueño del producto, la cual es importante al momento de elegir qué requerimientos se incluyen en la siguiente iteración.

Dentro de la Lista de Requerimientos, cada uno es representado en forma de “*Historia de Usuario*”. Una *Historia de Usuario* describe funcionalmente lo que será valorado por el usuario de un sistema o software. Se compone de una *descripción corta* escrita del requerimiento, la cual sirve como apunte o recordatorio; conversaciones acerca del requerimiento, que sirven para profundizar en el requerimiento; pruebas, detalle de documentos, etc. que pueden ser usados para determinar cuándo se completa el requerimiento. (22)

En COMPETISOFT, el proceso *Administración del Proyecto* recibe el documento formal *Plan de Proyecto* desde el proceso *Gestión de Proyectos*, perteneciente a la categoría Gerencia del modelo de procesos COMPETISOFT. Si bien este proceso no forma parte del *Perfil Base*, tomado para el análisis comparativo, es necesario generar al menos un Plan de Proyecto acotado que sea entrada al proceso *Administración del Proyecto* para poder comprender las necesidades del cliente, delimitar el alcance que tendrá el producto de software, asignar un equipo de trabajo, entre otras actividades asociadas a un proyecto de desarrollo de software. El plan acotado se puede limitar a la Descripción del Producto deseado por el cliente, el Alcance, Objetivos, Necesidades de Negocio, Supuestos, premisas y restricciones. (5)

Este *Plan de proyecto acotado* es el que puede asociarse a la *Lista de Requerimientos*. La información de ambos es dinámica, ya que se actualiza a medida que avanza el proyecto y se ejecutan ciertas prácticas y actividades. En COMPETISOFT particularmente son los procesos *Administración del Proyecto* y *Desarrollo de Software* donde se actualiza la información. En Scrum, el Dueño del producto es el responsable de mantener la lista correctamente actualizada.

Dentro de las salidas generadas en el proceso *Desarrollo de Software* se encuentra el documento *Especificación de Requisitos* el cual contiene la *Descripción de los requisitos funcionales* con necesidades que debe satisfacer el sistema de software. Esta descripción es otro componente de la definición del modelo de procesos de COMPETISOFT que puede relacionarse con la Lista de Requerimientos descrita en Scrum.

Lista de Requerimientos de la Iteración

La Lista de Requerimientos de la iteración se conforma por los ítems elegidos por el Dueño del Producto de la Lista de Requerimientos del Proyecto que se encuentren en estado *Pendiente*, con el fin de ser agregados al producto de software como parte de su incremento. Los ítems seleccionados son revisados junto al equipo de trabajo para determinar si es viable incluir su construcción en la iteración.

La lista de requerimientos de la iteración, que se confecciona durante la reunión de planificación de la iteración detallando las actividades y tareas que necesitan ser realizadas para completarla, es actualizada por el equipo de trabajo en el transcurso de la iteración.

Lo mismo sucede con el *Plan de Proyecto* en el modelo de procesos de COMPETISOFT. En la fase *Requisitos* de un *Ciclo de Desarrollo*, actividad del proceso *Desarrollo de Software*, una parte del equipo de trabajo revisa junto al cliente y al usuario el *Plan de Proyecto* con el objetivo de obtener los requerimientos a construir durante el ciclo en curso. En esta misma fase se genera la *Especificación de Requisitos* la cual contiene la descripción detallada de los requisitos necesarios para construir los requerimientos seleccionados para el ciclo.

Reunión de Planificación de la Iteración

Tanto en Scrum como en COMPETISOFT, para dar comienzo a la iteración o al ciclo, se concreta una reunión que tiene como objetivo principal obtener el compromiso del equipo de trabajo con el proyecto en general y con la iteración o el ciclo en particular, obteniendo las necesidades funcionales de negocio que se incluirán al producto de software, para, posteriormente, ser entregado al cliente y a los usuarios. Además, estas reuniones buscan lograr un entendimiento común, entre el cliente y el equipo, de las necesidades de negocio.

En Scrum denomina esta reunión como *Planificación de la iteración*, en la cual participa el Dueño del Producto, el Facilitador de Scrum y el Equipo de Trabajo, asignado al proyecto. La primer parte de esta reunión consiste en que el Dueño del Producto explica los requerimientos seleccionados por él, “que aportan mayor valor de negocio”, el Equipo realiza todas las preguntas necesarias para lograr entenderlos.

Esta primer parte tiene, en el proceso *Desarrollo de Software* de COMPETISOFT, su correlación con la fase de Requisitos del ciclo de Desarrollo. En esta fase participa parte del equipo de trabajo (los roles Analista y Equipo de Seguridad) junto al Cliente y Usuario. Esta es una diferencia importante con Scrum, donde la participación del equipo es obligatoriamente completa.

La segunda parte de la reunión de planificación consiste en un análisis más profundo de los requerimientos seleccionados por parte del equipo de trabajo junto con el Facilitador. Este análisis da por resultado el detalle de actividades que son necesarias realizar para completar cada requerimiento.

Asimismo, en las fases *Análisis y Diseño* del ciclo del proceso *Desarrollo de Software* se analizan detalladamente los requerimientos para armar la *Especificación de Requisitos*, documento técnico-funcional asociado al proyecto que servirá como guía en la construcción del producto de software asociado al proyecto.

Reunión diaria

En Scrum una de las prácticas que se llevan a cabo durante la iteración es la Reunión diaria, en la cual participa el Equipo de trabajo completo junto al Facilitador de Scrum y tiene como objetivo sincronizar la información y el trabajo del equipo con respecto al proyecto.

En el modelo de proceso de COMPETISOFT no se especifica en forma explícita ninguna reunión que se lleve a cabo diariamente durante el transcurso de un *Ciclo de Desarrollo*, pero al ser ésta parte del proceso Scrum, puede incorporarse a las actividades del proceso *Desarrollo de Software* con el fin de asegurar una buena comunicación e información del estado del proyecto por parte del equipo.

5.2.2.3 La documentación

Una diferencia sustancial entre Scrum y el Modelo de Procesos COMPETISOFT radica en la documentación que especifica cada uno en su definición.

El Modelo de Procesos COMPETISOFT especifica una serie de documentos que se tienen que generar y actualizar como salida de actividades de procesos, que luego se indican como entrada de otras actividades. Además, para algunos documentos se adjunta una plantilla de soporte para facilitar su generación.

Scrum en su definición no especifica documentación que deba generarse de un proyecto, así como tampoco ninguna plantilla soporte para la generación de documentación, pero tampoco anula la documentación de un proyecto. Con base en la declaración que el *Producto de Software funcionando es más valioso que la Documentación extensa* del Manifiesto Ágil se recomienda: (21)

- Evitar la documentación excesiva y difícil de actualizar. Generar aquella que aporte valor al producto de software.
- Definir el detalle y el tipo de documentación que se necesita mínimamente para tener en un proyecto.
- Evitar que reemplace la comunicación cara-a-cara de los participantes del proyecto.
- Facilitar la generación de documentación básica con el uso de plantillas sencillas.

Es importante destacar que *el beneficio de tener documentación debe ser mayor que el costo de crearla y mantenerla*, dado que si no se actualiza la documentación se convierte en estática y obsoleta.

5.2.3 Conclusiones

Scrum y el Modelo de Procesos COMPETISOFT se proponen como Marco Metodológico con el fin de ser adaptables al proyecto y al equipo de trabajo. En particular, el Perfil Base de Proyecto acota la definición del Modelo de Procesos de COMPETISOFT a una situación determinada de una organización, la cual necesita cumplir con los compromisos asumidos con un cliente de un proyecto de software específico, en este sentido, Scrum posibilita afrontar el mismo compromiso.

Al ser definidos como un marco metodológico, ambos proponen prácticas, actividades y reglas a adoptar por un equipo de trabajo. Coinciden principalmente en definir procesos iterativos e incrementales, definen períodos de tiempo, divididos en etapas, con un principio y fin determinados. Esta coincidencia facilita la inclusión o adaptación de prácticas definidas por el

Perfil Base de Proyecto con el fin de mejorar las prácticas de Scrum, dado que los dos apuntan a desarrollar software de una forma similar.

En cuanto a los participantes del proyecto, Scrum y el Perfil Base de Proyecto, destacan la importancia de lograr una buena comunicación, tanto dentro del equipo de trabajo como con el cliente. Una buena comunicación facilita la generación de un “lenguaje común” entre los participantes y además su integración, permite que el equipo de trabajo comprenda los requerimientos del cliente lo cual se refleja en el producto resultante, aumenta la motivación de los participantes, entre otros beneficios.

Asimismo sitúan al cliente como integrante activo del proyecto de software con el fin que el producto de software resultante sea realmente lo que solicita el cliente, que refleje y solucione lo más fielmente posible las necesidades de negocio del cliente para que sea un producto útil y el proyecto alcance sus objetivos.

Algunas prácticas de Scrum apuntan a lograr equipos auto-gestionados y auto-organizados que no necesiten un responsable o líder de proyecto que los lidere, aunque existe la figura de Facilitador de Scrum con la responsabilidad de ayudar al equipo en el gerenciamiento del proyecto. En cambio, el Perfil Base de Proyecto de COMPETISOFT define roles Responsables de Proyecto y Desarrollo quienes ejecutan tareas de gerenciamiento del proyecto, determinando tareas que el equipo de trabajo lleva a cabo, entre otras responsabilidades. Aunque es una diferencia conceptual importante, puede quedar a criterio de la organización y/o del equipo de trabajo la forma de gerenciamiento del proyecto específico que se adopte, para la elección se puede tomar en consideración la estructura de la organización, las características del proyecto y del equipo de trabajo asignado al mismo, así como también las características del cliente y la relación de este con la organización, entre otras.

Otra diferencia a destacar en la comparación de ambos procesos es en la documentación que definen. Como se indicó anteriormente, Scrum no define ninguna documentación, solo recomienda que ésta sea la mínima necesaria. En contraposición, el Perfil Base de Proyecto define una serie de documentos a generar y actualizar durante el ciclo de vida del proyecto. Es importante que se evalúe y se decida qué documentación es importante y necesaria generar y mantener para el proyecto, el cliente y usuarios, así como también para el equipo de trabajo y la organización.

Es importante concluir que tanto Scrum como el Perfil Base de Proyecto de COMPETISOFT se destacan por facilitar la adaptación de sus prácticas, reglas y actividades al contexto en el que se encuentra un proyecto de desarrollo de software, este contexto esta dado por diferentes factores, entre los que se puede enumerar el cliente que hace la solicitud del producto de software, la situación, económica, institucional y estructural de la organización, el equipo de trabajo asignado al proyecto, el tipo de proyecto, entre otros. Además, al coincidir ambas definiciones en premisas esenciales, se puede complementar Scrum con prácticas del Perfil Base de proyecto, eligiendo aquellas que sean adecuadas para mejorar y completar el proceso definido por Scrum.

5.3 Proceso Definido

Una vez realizados los análisis comparativos entre los diferentes procesos metodológicos, el siguiente paso consistió en diagramar la base de un proceso que pueda ser adoptado por una empresa PyME de la industria de software y que resulte flexible y adaptable según las particularidades de cada empresa interesada.

La empresa PyME descrita en *Situación Actual* describe algunas particularidades que forman parte de un común denominador entre las empresas pequeñas desarrolladoras de software,

hecho que puede facilitar la identificación del proceso como el adecuado a la estructura de una empresa.

La comparación entre *Scrum* y el *Proceso Real* brinda como resultado la clasificación de reglas y prácticas definidas por la metodología ágil, que se adoptaron y/o adaptaron en el proceso de desarrollo del proyecto de software en cuestión, además de un subconjunto de reglas y prácticas que no pudo adaptarse para el caso particular del proceso contemplado en el análisis.

Un resultado similar se obtuvo al evaluar las reglas y prácticas de *Scrum* contempladas o no en la definición del *Modelo de Procesos COMPETISOFT*, identificando las diferencias existentes entre ambas definiciones.

Ambas comparaciones permiten obtener un conjunto de prácticas que son posibles aplicar a una empresa PyME, indicando además consideraciones al momento de la adopción y detalles a tener en cuenta o evaluar para lograr una adaptación más adecuada. Además, puede indicarse un conjunto de prácticas del Modelo de Procesos COMPETISOFT sugeridas para ser implementadas de forma que se logre un proceso completo, consistente, adaptable, con el fin de lograr que la posterior aplicación del Modelo de Mejora de COMPETISOFT resulte lo más natural posible.

Para comenzar con la caracterización del Proceso Definido, se comienza con establecer las características principales de la organización y su proceso.

5.3.1 Características del Proceso Real

Las características la empresa PyME bajo estudio, las cuales pueden ser comunes a las empresas de desarrollo de software, son las siguientes:

- Empresa PyME del rubro informática.
- Posee una cartera de clientes reducida, entre 3 y 5 clientes, con 1 o 2 proyectos por cada uno.
- El staff de desarrollo se compone de menos de 20 miembros, distribuidos entre los diferentes proyectos, por lo cual es posible que algunos miembros se encuentren asignados a más de un proyecto simultáneamente.
- El staff es multidisciplinario, cubre los perfiles de arquitecto, analista funcional, encargado de pruebas, referente técnicos, desarrolladores senior, semi-senior y junior. También se encuentra el personal administrativo.
- Los socios de la empresa PyME participan de ciertos proyectos en forma activa, desempeñando tareas de desarrollo, arquitectura, pruebas de aceptación o análisis funcional.
- Los socios poseen poca experiencia en la gestión empresarial.
- No hay un plan de capacitación del staff definido. La capacitación de cada integrante se realiza a demanda.
- Empresa joven, con menos de 10 años en el mercado.
- No hay un modelo base de proceso que sea claro y aplicable a todos los proyectos. Sin embargo, existen algunas prácticas comunes a todos los proyectos pero implementadas de forma diferente.

El *Proceso Real* es una adaptación de la definición formal de Scrum que el equipo de trabajo logró a lo largo de la implementación de Scrum, que se realizó durante la gestión del proyecto real, para la “conformación” de su metodología de trabajo. A continuación se resumen las características principales del proceso real:

- Se definen iteraciones de una semana de duración. Al cierre de cada iteración se realiza una entrega de incremento de funcionalidad al cliente. En caso de ser necesario, se eleva a 2 semanas la duración de la iteración, el cambio en la duración se realiza en la planificación y con el consenso de los participantes.
- Fases de cada iteración: Planificación de la iteración, Codificación y Pruebas, Generación del entregable, Instalación de la nueva versión en el cliente, Retrospectiva.
- Se dispone de un “*tablero de la iteración*”, el cual contiene la lista de los requerimientos que serán entregados al término de la iteración como incrementos de funcionalidad.
 - Por cada ítem de la lista se detallan las tareas que es necesario llevar a cabo para completar el ítem.
 - Cada miembro del equipo se responsabiliza por la ejecución de una o más tareas.
 - Cada tarea transita entre diferentes estados durante la iteración, a medida que se avanza en su ejecución; estados: *Pendiente, En progreso, en Prueba desarrollo, en Prueba QA, Terminada*.
- Se definen una serie de reuniones que se suceden durante la iteración:
 - Planificación de la iteración, durante la cual se completa el “*tablero de la iteración*” con los incrementos funcionales.
 - de Avance, diariamente el equipo se reúne y comparte el avance en el proyecto, solicita ayuda si la necesita. El equipo en su conjunto evalúa los ajustes que sean necesarios realizar.
 - Retrospectiva, puesta en común del equipo acerca del desenvolvimiento de la iteración, se detectan las mejoras a realizar y las acciones a tomar durante la siguiente iteración.
- Equipo es *multidisciplinario*. Cada integrante, solo o en consenso con el resto del equipo, se asigna las tareas a realizar. El objetivo es lograr que el equipo sea auto-gestionado.
- Se asigna un *Representante del cliente* quien es el nexo entre el cliente y el equipo de trabajo. Sus responsabilidades son:
 - Se encarga de la comunicación con el cliente.
 - Prioriza junto al cliente los requerimientos a incorporar al producto de software.
 - Explica los requerimientos al equipo durante la reunión de Planificación.
 - Se encuentra a disposición del equipo para responder consultas durante la iteración.
 - Participa de las reuniones con el equipo.
 - Mantiene actualizada la *Lista de Requerimientos*.

5.3.2 Propuesta de Proceso Definido

5.3.2.1 Prácticas a adoptar

De acuerdo a la experiencia obtenida en el desarrollo del proyecto real, y a partir de la comparación realizada entre los procesos metodológicos, se puede obtener un conjunto de prácticas que son necesarias adoptar para asegurar una buena gestión del proyecto de software y facilitar la comunicación y el flujo de información entre los involucrados. A continuación, se enumeran las prácticas básicas del proceso a implementar en una organización:

- Documentar la información concerniente al proyecto:
 - Motivo por el cual se necesita un producto de software –Necesidad de Negocio-.
 - Descripción del proyecto.
 - Alcance y Objetivos.
 - Supuestos y premisas. Restricciones.
 - Como parte del presupuesto del proyecto, se estimará el Tiempo y Costo de desarrollo.
 - Elaborar un *Prototipo de Interfaz de Usuario* cuando sea necesario. El prototipo representa una aproximación a la interfaz de la herramienta que utilizará el usuario. En ciertos casos podría conformar una parte del presupuesto del proyecto.
 - Seguimiento de Defectos, registrar todos los defectos encontrados en el producto. Es conveniente separar los defectos encontrados durante las pruebas de Verificación de aquellos encontrados en las pruebas de Validación y Aceptación.

El formato y tipo de esta documentación se generará en base a, entre otras condiciones, el proyecto, las cláusulas establecidas por el cliente, el conocimiento –previo y adquirido- de la organización en herramientas de documentación.

- Descripción del Producto de software resultante del proyecto.
- Definir *Iteraciones* cortas, entre 1 a 4 semanas, para realizar el desarrollo de requerimientos. Cada iteración tendrá las fases:
 - Inicio –reunión de planificación-,
 - Realización/Construcción – reunión de avance, codificación y pruebas-,
 - Entrega –generación de entregable, demo al Cliente-,
 - Cierre –reunión de retrospectiva-.
- Recibir los *requerimientos*, nuevos o de modificación de funcionalidad incluida en el producto de software, por parte del cliente. Realizar un análisis en conjunto con el cliente y/o usuarios principales para determinar la criticidad e importancia de los requerimientos, otorgándoles una prioridad, la cual permite la selección en la reunión de planificación. Con esta información, crear o actualizar la Lista de Requerimientos del Proyecto.
- Realizar *entregas de Incremento Funcional* en el producto de software lo más asiduamente posible, preferentemente al término de cada iteración, esto puede definirse en conjunto con el cliente. Cada lapso de tiempo que transcurre entre las entregas se lo denominará como *ciclo*.
- Definir un Protocolo *de entrega* del Producto de Software, en el ambiente de QA y/ Productivo del cliente, con el fin de identificar tareas a desarrollar y sus responsables, tanto del lado del cliente como de la organización.
- Realizar una *Reunión de Planificación* al inicio de cada iteración, la misma tiene como fin seleccionar los requerimientos, según su prioridad, que formarán parte del próximo incremento funcional del producto, además del entendimiento de los requerimientos por parte del equipo de trabajo, encargado de la producción y agregado del incremento al producto de software. Es fundamental que participe todo el equipo de desarrollo. Es deseable que participe el cliente o un usuario clave.
- Al finalizar una iteración realizar la *Reunión de Cierre/Retrospectiva* del equipo con el fin de identificar sugerencias de mejora y las acciones a llevar a cabo durante la siguiente iteración o en un tiempo corto, dependiendo de la acción que se haya decidido seguir.

- Durante el transcurso de la iteración, el equipo participará de una *Reunión de Avance* con el objetivo que el equipo comunique el estado de las tareas que se comprometió a hacer, solicitar ayuda, detectar retrasos o falta de tareas o requerimientos. Además, cada miembro del equipo se compromete a realizar una o más tareas para avanzar en el desarrollo de la iteración. Esta reunión se realiza diariamente, aunque su periodicidad puede ser evaluada por el equipo en su conjunto.
- Coordinar una *Reunión de Cierre/Demo con el Cliente* con el propósito de mostrar el incremento funcional del producto de software. Es preferible que esta reunión se lleve a cabo al finalizar cada iteración, de no ser posible, se puede determinar que se realice cada cierta cantidad de iteraciones. La reunión de Cierre/Demo con el cliente le permite al equipo recibir la impresión del cliente en cuanto al trabajo realizado en el incremento entregado. Además, aumenta el compromiso del equipo de trabajo en entregar un producto de buena calidad y favorece el fortalecimiento de una comunicación fluida con el cliente y a la confianza mutua.

Considerar que la información que sea documentada tiene que ser actualizada a medida que evoluciona, resulta de vital importancia garantizar su consistencia y coherencia durante el ciclo de vida completo del proyecto. Por esta causa, es conveniente que la forma de documentar resulte simple y comprensible y accesible; asimismo, que todos los involucrados en el proyecto conozcan su ubicación.

5.3.2.2 Roles y Responsables

Dentro de una organización es necesario contar con recursos humanos que cubran las tareas relacionadas con el desarrollo del producto de software. De acuerdo a sus habilidades y a sus responsabilidades, estos recursos estarán asignados a lo largo de todo el proyecto o bien podrían ser asignados en la medida que se necesiten en el proyecto. Lo importante es que la organización garantice la continuidad del proyecto y la entrega continua de incremento de funcionalidad al cliente del producto de software.

Los roles básicos que se necesitan en el desarrollo de software son:

Responsable de la Gestión del Proyecto	Encargado de realizar las actividades concernientes a la gestión de un proyecto, capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software.
Responsable de pruebas de verificación y validación	Con conocimiento en la planificación y realización de pruebas de integración y de sistema y en técnicas de revisión. Se sugiere asignar un responsable de pruebas de aceptación por parte del cliente, con asistencia de un responsable idóneo del equipo de trabajo, de no ser posible, será designado por la organización.
Representante del Cliente	Encargado de recibir los requerimientos, validarlos y priorizarlos en conjunto con el cliente. Además, actúa como nexo entre el equipo de desarrollo y el Cliente, detallando los requerimientos al equipo.
Analista funcional	Encargado de interpretar los requerimientos del cliente y tener una visión más cercana al negocio, con conocimiento y experiencia en la especificación y análisis de los requisitos.
Programador	Con conocimiento y/o experiencia en programación, integración y pruebas unitarias.
Arquitecto	Encargado de definir la plataforma tecnológica sobre la cual se desarrollara el producto de software, conocimiento de los recursos

	existentes que pueden ser reutilizados, necesita tener una visión global del negocio y de las soluciones de arquitectura que garantizan la evolución del sistema.
--	---

Cada rol puede ser desempeñado por 1 o más personas.

5.3.2.3 Condiciones necesarias

Conforme la evaluación del aprendizaje adquirido durante el progreso de la implementación del proceso y a las observaciones realizadas, se puede señalar una serie de buenas prácticas para facilitar una adopción exitosa del proceso.

- Es responsabilidad de la gerencia de la organización impulsar la definición formal de un Proceso Base de Desarrollo de Software que funcione como común denominador a todos los proyectos, y que pueda ser adaptado, de ser necesario, en cada proyecto específico. La adaptación se realizará tomando en consideración el Proceso Base de la organización y el acuerdo realizado con el Cliente.
- Resulta indispensable que la organización se alinee con el proceso y brinde los recursos necesarios, tanto humanos como materiales, para facilitar la implementación. La capacitación del staff en el proceso se encuentra dentro de las responsabilidades de la gerencia de la organización, así como también brindar el apoyo y la contención para allanar el camino hacia la institucionalización de un proceso robusto y consistente que se ajuste a sus necesidades y que sirva para mejorar la gestión de los proyectos, aumentar la productividad del staff y garantizar la calidad del producto de software resultante.
- Una parte central de un proceso basado en metodologías ágiles reside en la buena interacción de los individuos involucrados en el proyecto, sin dejar de lado la interacción entre los miembros de la organización. Para lograrlo, es necesario trabajar en la confianza y respeto de los miembros con el fin de facilitar la fluidez de la comunicación y el entendimiento y la comprensión mutua. Las reuniones y la participación activa de los miembros en las mismas juegan un papel importante en la creación de una buena interacción, razón por la cual es clave que las mismas se lleven a cabo con la participación activa de todos los integrantes y con el conocimiento de los participantes de sus reglas, condiciones y objetivos. Es importante que la gerencia de la organización sea responsable de incentivar estas prácticas y auxiliar al equipo en la resolución de problemas, obstáculos o conflictos que pudiera afrontar.
- Entre los involucrados en el proyecto de software se encuentra el cliente, cuya participación activa en las reuniones claves es fundamental para el éxito del proyecto, ya que de él depende la priorización de los requerimientos, la explicación clara de éstos al equipo y la validación del cumplimiento de los requerimientos en el incremento funcional. Por lo tanto, es necesario contar con una representación del cliente en el proyecto.

En la ejecución de las prácticas del proceso es necesario que el equipo lo haga con disciplina, especialmente al inicio de la implementación, con el fin que el proceso se internalice, que se establezca una rutina en la ejecución de los pasos y se cumplan las reglas definidas por el proceso. Por esta razón, es fundamental comenzar con la capacitación en el proceso al equipo, además es de utilidad contar con un facilitador, designado por la gerencia de la organización, que acompañe al equipo durante el progreso del proyecto. Su presencia también resulta de importancia en aquellos equipos que no están familiarizados con el proceso, ya que se encarga de la capacitación del proceso.

5.3.2.4 Prácticas sugeridas

Dentro de las prácticas que propone COMPETISOFT en el Perfil Base de Proyecto, se pueden recomendar las siguientes para su implementación, con el objetivo de mejorar el Proceso Base:

- Documentar y actualizar un *Plan de Manejo de Riesgos* donde se identifique, describa y evalúen los riesgos que puedan afectar al proyecto. Los riesgos pueden estar relacionados con el equipo de trabajo, Cliente y usuarios, con la tecnología o la metodología, con la organización del proyecto (costo, tiempo, alcance y recursos) o riesgos externos al proyecto. Se procura identificar la probabilidad e impacto de cada riesgo estimando sus implicaciones en los objetivos del proyecto y priorizando sus efectos. Desarrollar procedimientos para reducir el impacto de los riesgos.
- Elaborar un Plan de Entregas estimativo a largo plazo que contenga, por cada ciclo de entrega el incremento funcional que se agregará al producto de software resultante y que podría ser entregado al cliente al término de cada ciclo. Este plan tiene que ser actualizado cada vez que se reciben nuevos requerimientos. Lo actualiza el Representante del Cliente en conjunto con Responsable de la Gestión del Proyecto y el Equipo de Trabajo.
- Elaborar los Casos de Prueba, apunta a documentar las pruebas de Validación y Aceptación que es necesario ejecutar en el sistema.
- Al finalizar cada ciclo, realizar las siguientes actividades:
 - Capacitación del Cliente, y otros recursos que la requieran, en la herramienta. Esta capacitación debería realizarse en el entorno del Cliente. Solicitar evaluación de la capacitación.
 - Realizar una Encuesta de Satisfacción del Cliente en la herramienta. La organización analizará los resultados de la encuesta con el objetivo de identificar mejoras a realizar en futuras iteraciones. El análisis realizado también se incorpora como aprendizaje del equipo de trabajo y de la organización.

La encuesta puede realizarse al finalizar el proyecto, aportando aprendizaje para futuros proyectos que afronte la organización.
- Evaluar el progreso del proyecto, en cuanto al cumplimiento de sus objetivos.

5.4 Resultados Obtenidos

El principal beneficio que se obtuvo fue la **formalización del proceso metodológico** para el desarrollo de software. La proposición de prácticas y reglas derivó en la obtención, por parte del equipo de trabajo, de un método coordinado, formal y ordenado para desempeñar sus funciones. A través del hábito en la ejecución de este proceso y de una definición clara en el papel que desempeña cada integrante y sus responsabilidades, entre otros puntos, se consolidó el proceso, convirtiéndose en cimiento de un proceso del que pueda apropiarse la organización, para ser utilizado, con los cambios que se consideren necesarios, en el desarrollo de proyectos de software actuales y futuros.

Asimismo, la gerencia de la organización supo reconocer las responsabilidades que debe afrontar para apoyar y facilitar la formación del proceso, su aprendizaje y adiestramiento, así como también su posterior consolidación dentro del equipo de trabajo. Cabe destacar que, a pesar de ello, debido a limitaciones de recursos e infraestructura que posee la organización, así como la relación contractual con el cliente, ciertas prácticas sugeridas no pudieron completarse. Estas prácticas están relacionadas con la creación de documentos, y posterior su actualización, que son requeridos para llevar un mejor control del proyecto y evaluación. En la organización,

debido a la poca cantidad de recursos humanos disponibles para ser incorporados a un proyecto, diferentes roles debían ser desempeñados por la misma persona, quien debía priorizar qué tareas realizar, ponderando el valor comúnmente en el impacto que el resultado de sus tareas tenía en el producto de software, carta visible del trabajo del equipo hacia el cliente y los usuarios.

Un punto decisivo en la implementación del proceso metodológico es la reunión de Retrospectiva, la cual se lleva a cabo al cierre de la iteración o ciclo. El aporte de esta reunión se centra en el objetivo que persigue: el equipo, en conjunto, analiza cómo fue el progreso de la iteración o ciclo y debate las sugerencias de mejora que podrían resolver los problemas detectados. Esta dinámica permitió ajustar las prácticas del proceso para que se adecuen al equipo de trabajo y al proyecto específico, ya que en ella el equipo encontró el espacio para tratar temas concernientes al equipo, método de trabajo y proyecto, incentivando el dialogo entre los integrantes, para conformar un equipo colaborativo, comprometido (con el proyecto y la organización), maduro y auto-gestionado, en el cual la mejora continua forma parte de su idiosincrasia.

6. CONCLUSIONES

Ante el crecimiento año a año del sector informático, las empresas pequeñas tienen que concentrar sus esfuerzos en continuar en el mercado informático, una estrategia es mantener y afianzar la relación con el cliente, respondiendo satisfactoria y rápidamente a sus requerimientos. También, ampliar su cartera de clientes o acceder a proyectos de mayor envergadura, sería una estrategia de consolidación y expansión, y motivaría su aspiración a realizar una certificación en alguna norma internacional de conocimiento masivo, y así mejorar su currículo para competir con empresas que posean una mayor infraestructura y capacidad de producción.

Para realizar una certificación de su proceso de desarrollo en alguna norma internacional, las empresas pequeñas tienen que afrontar el desafío de ejecutar el amplio abanico de tareas y actividades que las normas internacionales reconocidas proponen, pero que no se ajustan a su estructura. En algunos casos, a las empresas se les añade la dificultad previa de no contar con un proceso metodológico, o el mismo no está formalmente definido a nivel organización, por lo cual necesitan incorporar un proceso que guíe el desarrollo de sus proyectos de software.

Para esto, precisan imperiosamente de un proceso metodológico que asegure la calidad de los productos que genere, por lo tanto, como primer paso necesitan la adopción de este proceso, el cual debe ser acorde a su estructura. Scrum y COMPETISOFT presentan un proceso metodológico sencillo con prácticas que focalizan en la mejora continua y con la premisa que puede ser adaptado a las necesidades, no solo de la empresa, sino de un proyecto de software. Además, COMPETISOFT extiende su definición para abarcar a procesos específicos a la gerencia de la organización. También, COMPETISOFT propone un modelo de mejora de procesos que asegura la calidad de los procesos y, en consecuencia, de los productos generados.

A pesar que el modelo COMPETISOFT hace foco en empresas PyMEs, su definición en aquellas que son pequeñas puede resultar un poco extensa para implementar en forma completa. Es por esta razón que acota la definición a un perfil básico de los procesos Administración y Desarrollo de un proyecto de software. Sin embargo, las empresas pequeñas que reconozcan la importancia de contar con un proceso de calidad tienen que realizar el esfuerzo para implementar las prácticas y actividades que las lleven a mejorar su proceso de desarrollo y en consecuencia los productos que con él generen. Por lo tanto, la gerencia de la organización debe situar a la mejora de procesos como una inversión, necesaria en la estrategia de consolidación, expansión e incremento de competitividad, cuyo rédito puede obtenerse en poco tiempo.

Es imprescindible hacer partícipe del proyecto de mejora a todo el staff de la organización, ya que serán los ejecutores de las prácticas elegidas para ser implementadas. Que comprendan la importancia de un proceso de calidad es el paso inicial para que se comprometan a unificar esfuerzos y a trabajar cooperativa y colaborativamente en pos del logro del objetivo de mejora.

7. TRABAJOS FUTUROS

Para extender la definición del Perfil Base de proyecto, definido en COMPETISOFT, se propone impulsar trabajos en:

- La evaluación de los procesos de las Categorías Alta Dirección y Gerencia para identificar las prácticas que puedan ser implementadas por una empresa pequeña acotándolas a su estructura y necesidades.

Estos procesos definen prácticas concernientes a los aspectos directivos, la planeación estratégica, gestión de recursos y conocimiento, entre otras.

- La inclusión de prácticas y actividades que se requieran para completar los Niveles de Madurez 1: Gestionado y 2: Administrado, presentados en el capítulo 10: "COMPETISOFT como estrategia hacia CMMI" (19).

8. BIBLIOGRAFÍA

1. **Fundación Observatorio PyME.** *Informe 2005-2006: Situación actual y desafíos futuros de las PyME de Software y Servicios Informáticos.* Informe realizado con el auspicio y la colaboración de la Cámara de Empresas de Software & Servicios Informáticos (CESSI), Abril 2006.
2. **Chudnovsky Daniel, López Andrés y Melitsko Silvana.** El sector de software y servicios informáticos (SSI) en la Argentina: Situación actual y perspectivas de desarrollo. Julio 2001.
3. **Pino F. J., Garcia F., Ruiz F. y Piattini. M.** Adaptación de las Normas ISO/IEC 12207:2002 e ISO/IEC 15504:2003 para la Evaluación de la Madurez de Procesos Software en Países en Desarrollo. IEEE LATIN AMERICA TRANSACTIONS, April 2006.
4. **Schwaber, Ken.** *Agile project management with Scrum.* Microsoft Press, 2004.
5. **Proyecto COMPETISOFT 506PI287** - Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica. *Financiado por Ciencia y Tecnología para el Desarrollo (CYTED).* Código de proyecto 3789. Agosto 2007. Perfil 1: Administración de un proyecto Específico, Desarrollo de Software y Mantenimiento de Software. Elaborado con base en COMPETISOFT_v04 (Diciembre 2008. Versión 1.0).
6. **Kniberg, Henrik.** *Scrum and XP from the Trenches. How we do Scrum.* InfoQ Enterprise Software Development series, 2007. ISBN: 978-1-4303-2264-1.
7. **Larman, Craig.** *Agile & Iterative development. A manager's guide.* Addison-Wesley. Pearson Education, Inc., 2004. ISBN: 0-13-111155-8.
8. **Ken Schwaber, Jeff Sutherland y otros.** Manifiesto for Agile Software Development. URL: <http://agilemanifesto.org>. Febrero 2001.
9. **Wells, Don.** Extreme Programming: A gentle introduction. URL: <http://www.extremeprogramming.org>. 2009.
10. **Jeffries, Ronald.** XProgramming. An agile software development resource. URL: <http://xprogramming.com>. Copyright 1999-2010.
11. **Cockburn, Alisar.** *Crystal Clear: A human-powered methodology for small team.* Addison-Wesley Professional, 2004. ISBN: 978-0201699470.
12. **Software Engineering Standards Committee of the IEEE Computer Society.** IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998.* Institute of Electrical and Electronics Engineers, Inc., 25 de Junio de 1998. ISBN: 0-7381-0332-2.
13. **Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J..** Agile software development methods. Review and analysis. Technical Research Centre of Finland, VTT Electronics, September 2002. ISBN: 951-38-6010-8. URL: <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>.
14. **Canós, J., Letelier, P. y Penadés, M.C..** *Metodologías Ágiles en el Desarrollo de Software.* DSIC - Universidad Politécnica de Valencia.
15. **Mountain Goat Software.** URL: <http://www.mountaingoatsoftware.com/scrum/images> (Licencia Creative Commons Attribution 2.5)
16. **Rivera Martínez, Francisco, Hernández Chávez, Gisel.** *Administración de proyectos. Guía para el aprendizaje* Prentice Hall, 2010. ISBN: 978-607-442-620-5.
17. **Pressman, Roger.** *Ingeniería del software. Un enfoque práctico.* McGraw Hill, 1998. ISBN: 84-481-1186-9.
18. **Sommerville, Ian.** *Ingeniería del Software.* Pearson Addison Wesley, 2005. ISBN: 10-84-7829-074-5.
19. **Oktaba, Hanna, Piattini Mario, Pino Francisco J., Orozco, María Julia, Alquicira, Claudia.** *Competisoft Mejora de procesos de software para pequeñas y medianas empresas y proyectos.* Ra-Ma, 2008. ISBN: 978-84-7897-901-1.
20. **Baskerville, Richard L.,** *Investigating Information Systems with action research.* Computer Information Systems Department, Georgia State University. Volumen 2, article 19. Octubre 1999.

21. **Ambler, Scott W.** *Agile Modeling (AM)*. Effective Practices for Modeling and Documentation. URL: <http://www.agilemodeling.com/>. Licensing Material Copyrighted by Scott W. Ambler and Ambysoft Inc. 2001-2011.
22. **Cohn, Mike.** *User Stories Applied*. For agile software development. Addison-Wesley. Pearson Education, July 2010. ISBN: 0-321-20568-5.

9. TABLA DE ACRÓNIMOS

Sigla	Definición
CCNet	CruiseControl.Net (aplicación para integración continua)
CMMI	Capability Maturity Model Integration
CYTED	Ciencia y Tecnología para el Desarrollo
FLAC	Fox Latin American Channel
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
PAM	Program Assets Manager
PyME	Pequeña y Mediana Empresa
QA	Quality Assurance
SEI	Software Engineering Institute
SPI	Software Process Improvement
XP	Extreme Programming

10. TABLA DE TRADUCCIÓN

Idioma	Frase	Traducción
Inglés	Branch	Rama (en Visual Studio: paquete del código fuente).
Inglés	Feedback	Retroalimentación
Latín	<i>in situ</i>	En el lugar, en el sitio.
Inglés	<i>Know-how</i>	Saber-cómo. Se refiere al conocimiento o habilidad con respecto a un tema. Relacionado con la experiencia.
Inglés	Mentoring	Tutoría
Inglés	Release	Liberación (de una nueva versión de un producto).
Inglés	Sprint	Carrera corta, en Scrum significa Iteración.
Inglés	Tester	Persona encargada de realizar las pruebas de verificación e integración de un producto de software.
Inglés	Timeboxing	Período de tiempo fijo.