



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA INFORMÁTICA

AAA4532
TRAB
II2005
F8

**La aplicabilidad de las nuevas tecnologías
utilizando Ingeniería del Software**

TRABAJO DE ASCENSO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Realizado por:
Jacinto Fung León

Caracas, 15 Abril 2005

I. Pensamientos y Citas

I. Pensamientos y Citas

"Los problemas no se pueden solucionar si los enfrentamos con la misma forma de pensar que tuvimos cuando se crearon".

Albert Einstein

"Siempre hay una manera mejor de hacer las cosas... nuestro reto es encontrarla".

Jay Goltz

"La experiencia no es lo que te ha pasado, sino lo que haces con lo que te ha pasado".

Aldous Huxley

"Lo importante no es obsesionarse con la obtención de buenos resultados; concéntrese en seguir con disciplina los principios y conceptos adecuados, y verá cómo naturalmente aparecerán los resultados positivos."

James Citrin

"La innovación y la creatividad crean la oportunidad...la calidad crea la demanda".

Norman Vincent Peale

"No mires a lo lejos, descuidando lo que tienes cerca".

Eurípides

"Los problemas surgen a menudo en nuestra vida. Pero los problemas, por sí solos no provocan automáticamente el sufrimiento. Si logramos abordar con decisión nuestros problemas y centrar nuestras energías en encontrar una solución, el problema puede transformarse en un desafío ..."

Dalai Lama. "El arte de la felicidad" Pag.137

II. Dedicatorias

*“Gracias a ti, Dios, por mi existencia.
Gracias a mis padres por mi formación y mi vida.
Gracias a mi familia, por el apoyo.
Y a la UCAB, por incentivar me en este trabajo.*

*Gracias Liliana Rosado, por tenerme paciencia,
y por ser mi compañera en la vida.
Dedicada en la corrección,
en la redacción de este trabajo.”*

Jacinto Fung

Abril 2005

III. Índices

III. Índice de contenido	Pag.
1. Introducción	<u>1</u>
2. Entornos de MANTIS	4
1.1. Conceptos básicos de entornos	4
1.2. Tecnología de procesos de Software	5
1.2.1. Características de los procesos del software	6
1.2.2. Tecnología de Procesos del Software	6
1.3. Entornos de Ingeniería del Software	7
1.3.1. Integración	8
1.3.2. Orientación a Procesos	10
1.3.3. Modelo de Servicios: ISO 15940	12
1.4. Modelos de Procesos Software	15
1.4.1. Elementos Básicos	18
1.4.2. Niveles y Dominios	18
1.4.3. Vistas	20
1.4.4. Sistemas de Procesos de Software	20
1.5. Evolución de Procesos Software	21
1.6. Lenguajes de Modelado de Procesos	22
1.7. El Rol Humano en los Procesos	25
1.7.1. Interacción de Usuario	26
1.7.2. Interacción Interpersonal	26
1.8. Modelos de Procesos de Mantenimientos	27
1.8.1. Ciclo de Vida: ISO 12207	27
1.8.2. Modelo de Referencia para Procesos: ISO 15504	32
1.8.3. El proceso de Mantenimiento: ISO 12207 y 14764	35
3. CMMI	39
3.1. Selección de Modelo	40
3.1.1. Representación Continua	40
3.1.2. Representación de escenario	41
3.1.3. Modelo Integrar	41
3.2. Nivel de Capacidad	42

III. Índices

3.2.1. Nivel de capacidad 0: Incompleto	43
3.2.2. Nivel de capacidad 1: Ejecutado	43
3.2.3. Nivel de capacidad 2: Gestionado	43
3.2.4. Nivel de capacidad 3: Definido	44
3.2.5. Nivel de capacidad 4: Gestionado cuantitativo	45
3.2.6. Nivel de capacidad 5: Optimizado	45
3.3. Nivel de Madurez	46
3.4. Áreas de Procesos	46
3.4.1. Gestión de Procesos	47
3.4.2. Gestión de Proyecto	50
3.4.3. Ingeniería	55
3.4.4. Soporte	59
4. Aplicabilidad de las nuevas tecnologías	63
4.1. Procesos de Gestión	65
4.1.1. Ingeniería del Software en el proceso de Gestión Organizacional	67
4.1.2. Gestión de procesos de Software	71
4.2. Procesos de Producción	75
4.2.1. Modularidad de las Nuevas Tecnologías	76
4.2.2. Arquitectura de las Nuevas Tecnologías	78
4.2.3. Reusabilidad de las Nuevas Tecnologías	81
4.2.4. Comunicación las Nuevas Tecnologías	82
4.2.5. Entornos de la Ingeniería del Software las Nuevas Tecnologías	85
4.2.6. Uso de Lenguaje Modelado de Procesos	86
4.2.7. Mantenimiento del Software utilizando Nueva Tecnología	87
4.2.8. Desarrollo de Software utilizando Nueva Tecnología	89
4.2.9. Prototipos	91
4.2.10. Selección, generación de alternativas y tomas de decisiones	93
4.2.11. Tomas de decisiones en Selección de una Nueva Tecnología	94
5. Conclusiones	96
6. Apéndices	98
6.1. Lista de Acrónimos	98
6.2. Referencia Bibliográficas	102

IV. Índice de Tabla

Pag.

Tabla 2-1. Servicios de un Entorno de Ingeniería del Software	13
---	----

V. Índice de Figuras

Pag.

Figura 4.1 Impacto de la tecnología de Proceso del Software	64
Figura 4.2. Multicapas	81

1. INTRODUCCIÓN

La aplicabilidad de las nuevas tecnologías utilizando Ingeniería del Software, conlleva a una necesidad en las empresas de mantenerse actualizadas y continuar en el mercado tanto local como global. Las personas involucradas en la tecnología tienen el gran esfuerzo de mantenerse dentro del conocimiento y la tendencia de lo último en tecnología. Esto determina gran parte del esfuerzo en los proyectos llevados por una empresa y el personal para mantenerse al día. Los estudiantes en esta área tienen dificultad a menor escala, en adaptarse a la nueva tecnología, pero el esfuerzo depende de la aplicabilidad del conocimiento que se adquiere en un recinto universitario. Este trabajo busca minimizar estos esfuerzos con la aplicabilidad de Ingeniería del Software y disminuir el nivel de incertidumbre. Como lo describe Luis Fernández Sanz (2000) *“¿Renovarse o morir? o ¿renovarse para morir?. Una de las sensaciones más habituales de los profesionales de la informática en general, y del desarrollo de software en particular, consisten la impresión de desbordamiento ante la avalancha constante de nuevas tecnologías, entornos, siglas, etc. Parece como si los seres humanos fueran incapaces de gestionar con sentido el mundo tecnológico que han generado. La renovación tecnológica parece dejar exhaustos a muchos profesionales independientemente de las ventajas y oportunidades que supone o de la aceptación de su inevitabilidad.”*.

No existe una metodología genérica o un modelo conceptual para todos los proyectos con uso de las nuevas tecnologías, solo resolvemos los problemas con la utilización de métodos en los diferentes casos que se presenten. En este caso, nuestra herramienta de apoyo es con los conocimientos de Ingeniería del Software y la aplicabilidad en esta área.

En todos los niveles ya sea en el campo laboral o en el universitario, la nueva tecnología se observa como una variable que genera un nivel de incertidumbre alta. El nivel de incertidumbre es alta cuando hablamos de lo último en tecnología aplicado en un proyecto. En gran medida, el nivel de incertidumbre y los posibles puntos débiles que se presentan en un proyecto se ven disminuidos por la tendencia de aplicar metodología de desarrollo. Se han desarrollado nuevas metodologías de desarrollo que a medida que transcurre el tiempo se mantienen y llegan a ser métodos clásicos, lo que varía en el tiempo es la tecnología de la época. Un ejemplo es el modelo cascada o lineal secuencial; en los años 60-70 la tendencia de la tecnología aplicada en los software era muy limitada y centrada en ciertas personas, las cuales desarrollaron este modelo. Se sabe que el software era un complemento del hardware (Pressman 1998, pag. 4-5), el trabajo se centra en fabricación de computadoras de uso particular, el área del conocimiento en el software era de flexibilizar a los cambios propios de las necesidades de esos momentos hacia esos equipos. El nivel de incertidumbre de los proyectos era baja y se aplicaba este modelo en su desarrollo. Los requerimientos de software era bien descrito por los fabricantes de las computadoras, cuyo desarrollo eran esa misma empresa. En los actuales momentos, este modelo se aplica en algunos proyectos donde en el proceso de desarrollo del software se

detecta, en su análisis, que el nivel de incertidumbre es baja. Se aplica este modelo, aún sabiendo que el hardware no es fabricado por los mismos desarrolladores del software.

En los nuevos proyectos, aplicando una nueva tecnología, impacta de gran modo la parte económico y la técnica. Las compañías de éxito se mantienen por el esfuerzo de su personal, la labor es grande por parte del personal técnico dependiendo de la capacidad del personal del conocimiento no tanto de la tecnología sino la metodología que ha ayudado en el manejo del proyecto con nuevas tecnologías, como los mencionan en su trabajo Karen Lepe, Paula Mora y Carolina Rojas (2001):

“Manejo de la tecnología. En la primera etapa, una nueva tecnología es difícil de emplear, sus beneficios no son todavía obvios y su atracción es mayormente para aquellas personas que se han fascinado con ella por su propia causa. Podemos encontrar entusiastas personas que aman compartir historias sobre cómo ellos lucharon contra las dificultades y pudieron superarlas. El público en general se ve como que no tiene suficiente comprensión o mérito para realmente usar las nuevas invenciones. La radio de onda corta es un buen ejemplo de una tecnología que fue adoptada por un pequeño pero dedicado grupo en la etapa de manejo de tecnología. En la misma línea, las leyendas del Silicon Valley incluyen muchas historias de los primeros días de los computadores y los valientes pioneros que levantaron el Altair o el Osborne.”

Como también menciona Robert Hojman y Benardita Muñoz (2001) en su informe:

“Toda nueva tecnología no tiende a reemplazar a la anterior, sino que llena un espacio dejado por sus antecesoras, mejorando la tecnología preexistente o respondiendo a necesidades generadas por otras tecnologías. Por ejemplo, la radio no reemplazó al diario, ni la televisión a la radio, ni el vídeo a la televisión ni el computador a los dos anteriores.”

Este trabajo no involucra el efecto de las nuevas tecnologías a la sociedad, como el estudio realizado por Julio Cabero Almenara (2002). Por lo tanto, no evaluaremos el impacto de las nuevas tecnologías en la sociedad, debido a que se tendría que desarrollar cada área de la sociedad y se volvería muy extenso.

Esta investigación busca disminuir el impacto en el desarrollo de software; tomando en consideración que existen diversidades de procesos para desarrollar software, estudiaremos solo dos grandes procesos, los Entornos de MANTIS y CMMI, de los cuales se extraerá información del entorno que rodea a ambos procesos con el propósito de adaptarla al proceso del uso de las nuevas tecnologías, tomando en cuenta que los entornos de MANTIS y CMMI son nuevas tecnologías de desarrollo, por consiguiente el uso de éstas ayudan a entender así misma el uso de las nuevas tecnologías. MANTIS se basa en procesos de desarrollo en la fase de mantenimientos del software, y CMMI integra proceso de optimización organizacional, sistema y el software.

En este trabajo se describen, en el segundo capítulo, los conceptos de Ingeniería del Software y los Entornos de MANTIS, se profundiza en conceptos de la literatura académica del área y da practicidad a estos conceptos en los mantenimientos del software. En el tercer capítulo se reutilizan los conceptos anteriores para describir el modelo CMMI en la optimización de procesos organizacionales y procesos de desarrollo de software. Este

1. INTRODUCCIÓN

capítulo se realiza una traducción personal de los manuales originales del modelo. El cuarto capítulo se basa en los dos capítulos anteriores para elaborar el impacto y riesgo del uso de la nueva tecnología en el mantenimiento y desarrollo de un nuevo software. Se da un enfoque de la implementación de una nueva tecnología, basado en actividades de gerencias (gestión) y de actividades de producción para la minimización de la incertidumbre y riesgo. Se agrega las observaciones de conceptos tradicionales de los autores de Sommerville (2002) y de Pressman (1998) enfocado a la nueva tecnología.

2. Entornos de MANTIS

2.1 Conceptos básicos de Entornos de MANTIS

Los entornos de MANTIS que se describe a continuación, se extrae de un capítulo del trabajo realizado por Francisco Ruiz González (2003), en su trabajo "MANTIS: Definición de un Entorno para Gestión del Mantenimiento de Software". La selección de este capítulo, es debido, a su gran labor en describir claramente los diferentes conceptos que involucra el área de Ingeniería del Software en su modelo de desarrollo. Adicionalmente, es uno de los modelos de desarrollo de Software enfocado excelentemente en el proceso de mantenimiento.

Los entornos de MANTIS se apoyan en estudios donde se indica que el ciclo de vida de un software se consume desde un 60% a 80% en el mantenimiento. El costo de corregir un error durante la fase mantenimiento es cien veces mayor que en la etapa inicial de requisitos. La propuesta es abordar el problema del mantenimiento desde la perspectiva de "proceso de negocio", es decir, integrando los aspectos de la ingeniería con la organizacional y de gestión. El mantenimiento del software es clave de la competitividad entre organizaciones, ya que es la parte central actual de los cambios de los sistemas.

Los Entornos de MANTIS se basa en los conceptos de "Entornos de Ingeniería del Software (EIS) extendido", que integra y amplía los conceptos tradicionales de las metodología y de entorno de Ingeniería del Software (una colección de herramientas técnicamente integradas para automatizar los procesos de Ingeniería del Software). Al explicar el concepto de EIS extendido es "una colección de herramientas conceptuales, metodológicas y técnicas cuya finalidad es poder abordar los procesos del software (desarrollo o mantenimiento) desde la perspectiva global de proceso de negocio".

La gestión de los desarrollos del software y mas en el proceso de mantenimiento donde se considera como un problema con múltiples áreas. Por esta razón se ataca en diferentes disciplinas: Ingeniería del Software (tecnología de procesos software: EIS, sistemas de procesos, herramientas CASE, etc.), tecnologías y sistema de información (arquitecturas conceptuales, metamodelado, etc.), métodos de computación (ontologías, gestión del conocimiento, etc.) y gestión de proyectos.

La definición de Entornos de MANTIS, posee las siguientes características:

- integración por medio de herramientas
- orientación a procesos
- especialización de mantenimientos y
- escalabilidad y adaptabilidad

Existen tres clases de elementos en los Entornos de MANTIS:

- 1) Un marco conceptual para gestión de proyectos de mantenimiento.

- 2) Una colección de procedimientos para los procesos de gestión y organizacionales.
- 3) Una colección de prototipos definidos como componentes software del entorno. La arquitectura del sistema MANTIS se establece tres componentes:
 - a. Verticales.- herramientas para automatizar un determinado tipo de actividad.
 - b. Horizontales.- soporte en todo el entorno.
 - c. Externas.- no incluidas en MANTIS, pero se pueden invocar desde sus herramientas internas.

Prototipos Horizontales desarrollados han sido un gestor del repositorio integrado de datos y metadatos, una herramienta para metamodelización, un gestor de la base de conocimientos, y un interfaz de integración.

Prototipos Verticales incluyen herramientas gestor de peticiones de modificaciones, gestor de recursos humanos en el mantenimiento o gestor de cuestionario de evaluación de la madurez de un servicio de mantenimiento.

Prototipos Externas son los "Sistema de gestión de Flujos de trabajo", para automatizar el seguimiento y control de los proyectos.

Reificación de un ente según John McCarthy (McCarthy) es "hacer una cosa" de ella (donde re en latín se traduce como cosa). Desde un punto de vista lógico, una cosa es aquello que una variable puede recorrer para asignarle un valor. Así, la inteligencia artificial lógica necesita reificar esperanzas, intenciones y cosas desubicadas. Algunos filósofos deploran la reificación, señalando que con ella aparece una ontología sesgada. La inteligencia artificial necesita más cosas que las que habían soñado las filosofías de los filósofos. En general, la reificación le da al lenguaje un poder más expresivo. Permite referirse a entidades directamente que fueron anteriormente mencionados exclusivamente en metalenguajes.

La ontología según John McCarthy (McCarthy) es el capítulo de la filosofía que estudia qué cosas existen. El punto de vista de W.V.O. Quine es aquello sobre lo cual las variables buscan su rango. La ontología ha sido usada en forma diversa por la inteligencia artificial, pero en mi opinión lo que afirma Quine es lo mejor para el uso en esa especialidad. Tanto la ontología como la reificación tratan del mismo fenómeno. Es de lamentar que la voz "ontología" se usa en inteligencia artificial en un sentido más difuso y vago que la voz reificación.

2.2. Tecnología de proceso de Software

La mejora de la calidad de los productos del software se basa en el estudio y mejora de los procesos mediante los cuales el software es desarrollado y mantenido. Existe una correlación entre calidad de proceso y calidad de producto obtenido. En la Ingeniería del Software, aborda esta área y es conocida como "Tecnología de Proceso Software (TPS)", o "Proceso Software". La primera contribución importante de TPS fue la confirmación de

que el desarrollo y mantenimiento del software son procesos complejos, que requieren un esfuerzo colectivo y creativo. Por lo tanto, la calidad de un producto software depende fuertemente de las personas, la organización y los procedimientos utilizados para crearlo, entregarlo y mantenerlo.

2.2.1. Características de los Procesos Software

La definición de proceso de software complementa el concepto de “ciclo de Vida” en el sentido de que éste último define el esqueleto y la filosofía para llevar a cabo un proceso de software, pero no es suficiente para guiar y controlar un proyecto de desarrollo y/o mantenimiento. Un proceso software (PS) es “un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener producto software”.

La participación humana de forma creativa y de ausencia de acciones repetitivas hacen que ni el desarrollo ni el mantenimiento del software sean procesos de fabricación, pero existen algunas similitudes entre ambos tipos de procesos que son útiles para comprender los procesos software con una perspectiva más amplia. Al igual que el proceso de fabricación, los procesos software constan de dos sub-procesos interrelacionados: el proceso de producción y el proceso de gestión. El proceso de producción se relaciona con la producción y el mantenimiento del producto propiamente dichos, mientras el proceso de gestión proporciona los recursos necesarios para el proceso de producción y lo controla. Esto último es posible si el proceso de producción devuelve información al proceso de gestión sobre su comportamiento. El entorno externo también afecta estos dos sub-procesos, lo cual conlleva a peticiones sobre el producto desde el mundo exterior, es decir, el entorno exterior es quien justifica la existencia del proceso de producción. Además la gestión tiene que cumplir con estándares actuales que existen en el entorno, y con el entorno exterior influye también indirectamente en el proceso de producción. Los procesos de producción y de gestión explotan tecnologías de gestión y tecnologías de producción que vienen del entorno.

2.2.2. Tecnología de Proceso Software

La estructura básica de la tecnología de Proceso de Software (TPS) permite la integración de tecnologías de producción y de gestión en un nuevo entorno de trabajo, conocido como “Entorno de Ingeniería del Software orientado al Proceso” (PSEE, Process-centered Software Engineering Environment), que da soporte a los procesos de gestión y de producción de forma integrada. PSEE impacta con esta nueva tecnología al proceso de gestión y proceso de producción, el PSEE implementa, controla y mejora los flujos de información con los cuales el proceso de gestión controla al proceso de producción. PSEE integra la tecnología de Gestión y de producción, y explota la tecnología de Procesos.

El TPS genera ventajas en diversas áreas y conceptos:

- *Tecnologías* de desarrollo y mantenimiento de software: Aportan las herramientas, infraestructuras y entornos que necesitamos para hacer posible y económicamente

factible crear y mantener productos software complejos que satisfagan las necesidades sociales actuales y futuras.

- *Métodos y técnicas* para el desarrollo y mantenimiento de software: Suponen el soporte metodológico esencial para aprovechar de manera eficiente las tecnologías y realizar con éxito las actividades de desarrollo y mantenimiento del software.
- *Comportamiento organizacional*: la ciencia de las organizaciones y las personas. Útil en TPS porque, en general, los proyectos software se llevan a cabo por equipos de personas que tienen que ser coordinados y dirigidos dentro de una estructura organizacional eficiente.
- *Marketing y economía*: Los proyectos de desarrollo y mantenimiento de software no son esfuerzos autónomos ya que, como pasa con cualquier otro producto, el software debe estar dirigido a satisfacer las necesidades de clientes reales. Por ello, en los proyectos software son necesarias etapas (especificaciones de requisitos, etc.) para tener en cuenta el contexto donde se supone que el software será vendido y utilizado.

En conclusión, el desarrollo y el mantenimiento de software es basado en la interrelación que se produce entre los factores organizacionales, culturales, tecnológicos y económicos.

2.3. Entornos de Ingeniería del Software

El uso de herramientas para ayudar a los desarrolladores en la producción de software ha existido, de una u otra manera, desde los días iniciales de la Informática, el concepto de "*Software Engineering Environment*" (SEE), es decir, Entorno de Ingeniería del Software (EIS), es bastante reciente. Un EIS se define como "*una colección de herramientas que proporcionan un soporte automático, parcial o total, a las actividades de ingeniería del software*". Habitualmente, estas actividades se llevan a cabo en el marco de un proyecto software, y se refieren a aspectos tales como la especificación, desarrollo, reingeniería o mantenimiento de sistemas software. En la bibliografía los EIS también se han conocido con otros nombres (Hanrahan et al, 1994): IPSE (*Integrated Project Support Environment*), ISEE (*Integrated Software Engineering Environment*), coalición de herramientas CASE, herramientas CASE federadas, o ISF (*Integrated Software Factory*).

El término EIS se puede aplicar a sistemas de un alcance muy diferente: desde un conjunto de unas pocas herramientas ejecutándose sobre el mismo sistema, hasta un entorno totalmente integrado capaz de gestionar y controlar todos los datos, procesos y actividades del ciclo de vida de un producto software. Gracias a la automatización de actividades (de forma parcial o total), un EIS puede aportar importantes beneficios a una organización: reducción de costes (alta productividad), mejora en la gestión y mayor calidad en el producto final. Por ejemplo, la automatización de actividades repetitivas - como la ejecución de casos de prueba- no sólo mejora la productividad, sino que también ayuda a asegurar la completitud y consistencia de las actividades de prueba. Habitualmente, un EIS maneja información relacionada con:

- a) El software en desarrollo o mantenimiento (especificaciones, datos de diseño, código fuente, datos de pruebas, planes de proyecto, ...);

- b) Los recursos del proyecto (costos, recursos informáticos, personal, responsabilidades y obligaciones, ...); y
- c) Los aspectos organizacionales (políticas de la organización, estándares y metodologías empleados, ...).

Un EIS da soporte a actividades humanas mediante una serie de servicios que describen las capacidades del entorno. Los servicios proporcionan una correspondencia entre un conjunto de procesos escogidos, relativos al ciclo de vida del software, y su automatización mediante el uso de herramientas. En la mayoría de los casos la funcionalidad de una herramienta está relacionada con uno o más servicios. El interés investigador en el campo de los EIS comenzó alrededor del año 1990, cuando surgieron las primeras propuestas de modelos de referencia (Zelkowitz, 1993) y se propusieron las primeras clasificaciones de los servicios que deberían incluirse (Zelkowitz, 1996). Pero ha sido a comienzos del siglo XXI cuando se ha empezado a trabajar en propuestas de entornos, es decir, EIS orientados a un dominio específico (Lédeczi et al, 2001) e integrando diversas tecnologías que han sido desarrolladas recientemente (lenguajes de modelado de procesos, metamodelos, XML, etc.). Una prueba de lo anterior se encuentra en la revisión general de la situación de la investigación sobre EIS publicada por Ossher et al (2000). Estos autores indican que las principales líneas de investigación actuales se centran en los siguientes aspectos:

- a) La integración de las herramientas (en las tres áreas de datos, presentación y control), especialmente en base a la utilización de un repositorio que aproveche las potencialidades de XML y tecnologías web complementarias; y
- b) La orientación a procesos, los cuales deben integrar el soporte a herramientas para producir artefactos software con el soporte al modelado y ejecución de los procesos de ingeniería del software que producen dichos artefactos.

2.3.1. Integración.

El concepto que más diferencia un EIS de un simple conjunto de herramientas ejecutándose en una computadora bajo un mismo sistema operativo es el grado de integración que el entorno provee. El concepto de integración aplicado a EIS puede significar varias cosas relacionadas pero diferentes:

- El grado en que diferentes herramientas pueden comunicarse eficazmente entre sí dentro del marco de trabajo del EIS.
- Una medida de las relaciones entre los componentes de un EIS.
- La facilidad, interoperabilidad, portabilidad, escalabilidad, productividad, etc., producidas por la interacción "sin parches" entre un conjunto de componentes de un EIS.

Compartir un mismo sistema de gestión de objetos (gestor del repositorio) en vez de un sistema de archivos separado para cada herramienta es un aspecto importante de la integración, pero no es el único. Un EIS debe disponer de un conjunto de interfaces que permitan la cooperación arbitraria entre herramientas de fabricantes diversos. Por esta razón, la integración implica los tres aspectos siguientes:

- Un conjunto de servicios. Muchos de los servicios que se describen en apartados posteriores son aplicables a la integración. Por ejemplo, utilizar un sistema de gestión de objetos (SGO) común con esquemas comunes permite que las herramientas compartan objetos; utilizar características de presentación globales en el interfaz de usuario permite disponer de un "aspecto de visualización" similar en todas las herramientas; o los servicios de gestión de procesos y de comunicación son necesarios para que las herramientas puedan comunicarse unas con otras.
- Una nueva dimensión para cada servicio. Tener servicios comunes permite pero no obliga a la integración (los constructores de herramientas no están obligados a utilizarlos). Esta nueva dimensión indica el grado en que un servicio puede contribuir a aumentar la integración.
- Una política. También se requiere implantar políticas para que los constructores de las herramientas, marcos de trabajo y plataformas utilicen los servicios de integración eficientemente. Un ejemplo de esto son las "guías de estilo" para constructores de herramientas.

En base a la definición anterior, la integración es una propiedad de la relación entre los componentes de un EIS. Esta relación puede ser entre herramienta y marco de trabajo, herramienta e interfaz de usuario, herramienta y herramienta, herramienta y SGO, etc.; por eso la integración se puede considerar desde varias perspectivas diferentes:

- Para el *usuario*, un EIS integrado provee una vista común del sistema. El entorno entero funciona como una herramienta consistente en vez de como una colección de operaciones diferentes o invocadas separadamente.
- Para el *desarrollador de herramientas*, un EIS integrado provee un interfaz consistente para construir herramientas. De esta manera, las funciones para interactuar con el SGO, gestionar procesos y demás servicios deben estar claramente especificadas; y las herramientas podrán pasar información a otras herramientas de forma fácil. En general, la necesidad de integración en un EIS abarca varias dimensiones diferentes (Wasserman, 1990):
 - Datos. La integración de los datos es la habilidad de compartir la información dentro del EIS. El grado de integración de datos puede ser alto (las herramientas usan una base de datos común con un esquema común), mediano (formatos de datos comunes) o bajo (utilizar mecanismos de traducción). Otra característica que la integración de datos puede incluir es la composición.
 - Control. La integración del control es la habilidad de combinar las funcionalidades ofrecidas en un entorno de forma flexible. Las combinaciones pueden corresponder a preferencias de un proyecto y estar dirigidas por los procesos software subyacentes.

- **Presentación.** La integración de la presentación es la habilidad de interactuar con las funcionalidades del entorno mediante pantallas de apariencia similar y modos de interacción similares.
- **Procesos.** La integración de procesos es la habilidad de acceder a las funcionalidades del entorno utilizando un proceso software reificable predefinido.
- **Marco de Trabajo.** Esto se refiere al grado en que las herramientas están integradas, es decir, hacen un uso de el marco de trabajo. Thomas y Nejme (1992) ampliaron el concepto de integración representado por las dimensiones anteriores. Estos autores se centraron en las relaciones entre las herramientas y por ello no incluyeron la integración del marco de trabajo, aunque sugieren que esta integración deberá consistir en disponer de mecanismos que mejoren el grado de integración de las otras cuatro dimensiones.

2.3.2. Orientación a Procesos.

Ya se ha comentado la importancia que los EIS orientados a procesos (o PSEE's) tienen en la tecnología de proceso software (ver apartado 2.2), de hecho, el principal rol de un EIS es dar soporte para llevar a cabo los PS de forma eficaz. Este punto de vista está ganando peso porque los procesos de desarrollo y mantenimiento de software se han convertido, cada vez más, en actividades complejas y laboriosas de carácter intelectual, con un alto potencial para las mejoras en la calidad y la productividad basadas en la disciplina, la gestión, y la ayuda de EIS y otras tecnologías informáticas. Muchas organizaciones tienen problemas para definir y realizar los pasos que transforman las necesidades del usuario en un producto software de maneras que sean repetibles, medibles con respecto a su impacto en los objetivos de calidad, y adaptables o mejorables. Por tanto, la ayuda de un EIS para implantar un proceso definido durante la realización de un proyecto software puede proporcionar beneficios sustanciales a corto plazo. En este ámbito, un **proyecto** puede ser considerado como una instancia de un proceso que está siendo reificado (Feiler y Humphrey, 1993).

Dentro de un EIS, los servicios de gestión de procesos contribuyen a este soporte eficaz de los PS proporcionando facilidades orientadas al usuario final para definir y utilizar procesos que puedan reemplazar la invocación indisciplinada, difícil de controlar, y tediosa de herramientas individuales. Estos servicios también pueden ayudar a mejorar la arquitectura software de los marcos de trabajo EIS, en el sentido que los PSEE pueden ser diseñados más fácil y eficazmente (lo que en sí mismo supone un buen ejemplo de mejora de procesos). Kobialka (1998) ha realizado un estudio en profundidad sobre cómo los procesos software pueden ser soportados por un EIS y, en concreto, cómo implementar el soporte y los cambios de los procesos durante su reificación. Garg y Jazayeri (1996) han considerado que el soporte a procesos en un EIS está basado en las siguientes funcionalidades:

- *Definición de procesos.* Los ingenieros de software utilizan el EIS para definir un proceso de cara a su utilización en uno o más proyectos.

- *Análisis de procesos.* Dentro del EIS, un modelo de proceso (ver apartado 2.4) puede ser analizado para verificar su consistencia, completitud y corrección.
- *Presentación de procesos.* El EIS incluye soporte para la visualización gráfica de los PS (flujos de actividades) y de los productos (diagramas estructurados).
- *Simulación de procesos.* El EIS soporta el uso de simulaciones para poder evaluar la idoneidad de un proceso antes de consumir recursos en su realización real.
- *Automatización de procesos.* Una vez que un proceso ha sido definido, las actividades que no requieren intervención humana pueden ser identificadas y automatizadas por el EIS.
- *Supervisión de procesos.* El EIS supervisa la reificación de un proceso y registra la historia de las actividades realizadas. Esta historia del proceso puede utilizarse después para futuros desarrollos de nuevos procesos o mejora del existente.
- *Soporte de cambios en procesos.* El EIS permite que una organización cambie sus definiciones de procesos sin tener que interrumpir el trabajo (ver apartado 2.5).
- *Apertura.* El EIS provee herramientas para intercambiar datos y metadatos con herramientas no integradas o con otros EIS.
- *Soporte multiusuario.* Habitualmente, los proyectos de ingeniería del software son realizados por grupos de personas con diferentes roles, por tanto, el EIS debe dar servicio a todas las personas que trabajan juntas en un proceso.
- *Dirección de procesos.* Los ingenieros de software utilizan el EIS para llevar a cabo diferentes etapas de un proceso. El EIS debe ofrecer ayuda para elegir las etapas siguientes en base al modelo del proceso y del estado actual.
- *Interfaz de usuario específico para cada tarea.* Basándose en el modelo del proceso, el EIS puede adaptar la interfaz de usuario a las necesidades de cada tarea y así evitar un exceso de información presentada al usuario.

Los EIS orientados a procesos pueden ofrecer servicios comunes que saquen provecho de la similitud que existe entre los procesos y los productos software: los PS tienen ciclos de vida similares a los ciclos de vida de un producto software porque ambos tienen las siguientes características comunes:

- Pueden ser especificados, diseñados, implementados, reificados (ejecutados), analizados, medidos, modificados y mejorados.
- Pueden tener *arquitecturas de proceso*, que son marcos conceptuales para incorporar, relacionar y adaptar elementos de procesos de forma consistente, incluyendo la capacidad para indicar si un elemento de proceso es o no compatible con la arquitectura. Las arquitecturas de procesos pueden especificar interfaces entre subprocesos, pautas para su composición, y mecanismos de comunicación entre los elementos de procesos. También pueden permitir propiedades orientadas a un dominio de aplicación o a los objetivos de un proyecto concreto, como por ejemplo, el soporte de la reutilización, el prototipado o el desarrollo evolutivo.
- Los *activos de proceso (assets)*, es decir, procesos completos, elementos de procesos, modelos de procesos, arquitecturas de procesos o diseños de procesos, pueden organizarse en bibliotecas y ser reutilizados.

Algunos EIS proporcionan servicios de gestión de procesos utilizando facilidades comunes con otros servicios diferentes. Esto ocurre, por ejemplo, con las facilidades de gestión de objetos, ya que la información de los PS es compleja y además pueden tener que manejar la información del proyecto (así como las representaciones del estado de los procesos) durante la reificación de los procesos.

Cada vez es más frecuente que el desarrollo y mantenimiento de un producto software es realizado con la colaboración de varias empresas u organizaciones. Por ello, en los últimos años ha cobrado auge el estudio de los problemas que surgen cuando se desea que varios PSEE's diferentes y separados colaboren, y más concretamente, que exista interoperabilidad entre los procesos que soportan. Entre las diversas propuestas formuladas para abordar este problema, destacan las federaciones de PSEE's (Martínez et al, 2001a). En esta línea, Ben-Shaul y Kaiser (1998) han propuesto usar la metáfora de la alianza internacional, donde cada organización gestiona sus propios procesos (igual que cada país tiene sus leyes) y los procesos inter-organizacionales actúan de forma semejante al de los tratados entre países. En la bibliografía se han propuesto dos tipos de arquitecturas conceptuales para federaciones de PSEE's: basadas en el control, que favorecen la centralización al existir modelos de procesos comunes; y las basadas en el estado, que disponen de un espacio de trabajo donde se almacena el estado común (Estublier et al, 1998). En (Martínez et al, 2001b) se dispone de una panorámica general sobre el modelado y ejecución de procesos inter-organizacionales.

2.3.3. Modelo de Servicios: ISO 15940.

Para describir la relación de posibles servicios que puede soportar un determinado EIS (aunque también se puede aplicar a una única herramienta CASE), vamos a basarnos en la propuesta de estándar ISO 15940 (ISO/IEC, 2001b). Esta norma provee un modelo de referencia y una descripción de todos los servicios que soportan a los procesos del ciclo de vida del software. Un servicio se define como "*una descripción abstracta del trabajo hecho por una o varias herramientas software*". Cada descripción de un servicio es de carácter general y no asume ningún dominio de aplicación, modelo de ciclo de vida o herramienta en un proyecto.

Cada servicio es definido mediante las tres partes siguientes:

- a) *Concepto del servicio* : proporciona una descripción del servicio sin referirse a ninguna implementación concreta.
- b) *Operaciones básicas*: es una lista de las operaciones que deberían ser incluidas en un servicio. La lista no pretende ser exhaustiva y, en muchos casos, incluye sólo conceptos de servicio elementales o primarios.
- c) *Soporte Automatizado*: provee una lista de las operaciones que deben ser automatizadas para cumplir satisfactoriamente con el estándar. Esta lista es un subconjunto de las referidas en el punto anterior.

2. Entornos de MANTIS

Los servicios de un EIS se agrupan en seis categorías, que reflejan la gran diversidad y amplitud que tienen hoy en día las diferentes actividades de ingeniería del software. Estas categorías son:

1. Ingeniería Técnica: Soportan actividades -propias de los ingenieros de software relacionadas con la especificación, diseño, implementación, prueba y mantenimiento de software.
2. Gestión Técnica: Dan soporte a actividades mixtas comunes a ingenieros software y gestores de proyectos.
3. Gestión del Proyecto: Ofrecen soporte a actividades relacionadas con la planificación y ejecución de un proyecto software.
4. Gestión de Procesos: Ayudan a los proyectos software a alcanzar disciplina, control y comprensión clara de sus procesos y actividades.
5. Soporte : Útiles a todos los usuarios, están relacionados con procesar, formatear y distribuir datos en formato manejable por las personas.
6. Globales: Ayudan a que la infraestructura del EIS de soporte a las aplicaciones y herramientas. Estos servicios forman el núcleo del marco de trabajo (*framework*) de un EIS.

Servicios de un Entorno de Ingeniería del Software	
Categoría	Servicio
Ingeniería Técnica	IT.1 Ingeniería de requisitos software
	IT.2 Diseño software
	IT.3 Simulación y modelado software (para determinar la eficacia de diseños alternativos)
	IT.4 Verificación de software
	IT.5 Generación de software basado en componentes
	IT.6 Generación de código fuente
	IT.7 Compilación
	IT.8 Análisis estático de software (análisis de código fuente)
	IT.9 Depuración
	IT.10 Prueba de software
	IT.11 Integración de componentes
	IT.12 Ingeniería inversa de software (capturar información de diseño a partir del código)
	IT.13 Reingeniería de software
	IT.14 Trazabilidad de software (revisar relaciones entre artefactos a lo largo del ciclo de vida)

2. Entornos de MANTIS

	IT.15 Pruebas de cualificación de software (comprobar que el producto satisface los requisitos y está listo para su uso)
	IT.16 Prototipado software
	IT.17 Documentación de usuario
Categoría	Servicio
Gestión Técnicas	GT.1 Gestión de la configuración
	GT.2 Gestión de cambios
	GT.3 Gestión del repositorio EIS
	GT.4 Reutilización (almacenar, inspeccionar y reutilizar activos -assets-)
	GT.5 Colección y análisis de métricas
	GT.6 Aseguramiento de la Calidad
	GT.7 Auditoría
Categoría	Servicio
Gestión del Proyecto	GP.1 Planificación
	GP.2 Estimación
	GP.3 Análisis de riesgos
	GP.4 Seguimiento (evolución de costes, tiempos y requisitos)
	GP.5 Evaluación (análisis y toma de decisiones en función de los datos de seguimiento)
Categoría	Servicio
Gestión de Procesos	PR.1 Definición de procesos
	PR.2 Biblioteca de procesos
	PR.3 Iniciación de procesos
	PR.4 Utilización de procesos en proyectos
	PR.5 Supervisión de procesos
	PR.6 Mejora de procesos
	PR.7 Documentación de procesos
Categoría	Servicio
Soporte	SO.1 Soporte global (crear y manipular objetos dentro del EIS)
	SO.2 Publicación (crear y editar documentos)
	SO.3 Soporte al trabajo en grupo (trabajo cooperativo distribuido)
	SO.4 Soporte a la comunicación de usuarios (facilidad para comunicar con clientes y usuarios)
	SO.5 Administración del EIS (definir y controlar accesos al EIS, controlar su funcionamiento)
	SO.6 Cumplimiento de políticas (implantar políticas de seguridad, accesos, integridad y auditoría)

2. Entornos de MANTIS

Categoría	Servicio
Globales	GL.1 Gestión de la infraestructura del EIS (gestionar los recursos del EIS)
	GL.2 Comunicación inter-proceso (mecanismo de comunicación básico entre herramientas y/o servicios)
	GL.3 Gestión de objetos (definir, almacenar, mantener y acceder a objetos y sus interrelaciones)

Tabla 2-1. Servicios de un Entorno de Ingeniería del Software.

En la Tabla 2-1 se muestra una lista de todos los servicios EIS incluidos en la propuesta ISO 15940. A continuación se detallan los servicios que están dentro del alcance de este trabajo, es decir, aquellos que se corresponden con las categorías de procesos de gestión y de organización en la norma ISO 15504-2. En esta lista se incluyen dos servicios de gestión técnica, todos los servicios de gestión del proyecto y de gestión de procesos, y un servicio de soporte (están enunciadas en la Tabla 2-1).

2.4. Modelos de Procesos Software

En una organización o en un dominio de aplicación, los procesos de diferentes proyectos tienden a seguir patrones comunes, o bien porque las mejores prácticas son formalmente reconocidas o bien por la existencia de estándares. Por lo tanto, se hace necesario intentar capturar estos aspectos comunes en una representación del proceso, la cual describe estas características comunes y fomenta la homogeneidad.

Durante las tres últimas décadas, el estudio de los procesos de producción del software ha llevado al desarrollo de varios *Ciclos de Vida* del software que pueden ser empleados en las actividades de ingeniería del software. Estos son, por ejemplo, los modelos en Cascada, Evolutivo, Transformacional o en Espiral. Las funciones primarias de un ciclo de vida son determinar las fases envueltas en el desarrollo y mantenimiento del software, y establecer los criterios de transición para la progresión de una fase a la siguiente. Estos últimos incluyen a los criterios de compleción para la fase actual y a los criterios de entrada para la siguiente. Estos modelos de ciclo de vida ayudan a los ingenieros y a los gestores a comprender mejor el proceso software, y a determinar el orden de actividades globales envueltas en la producción de software. Sin embargo, una limitación de estos modelos es que no dan mayor importancia a procesos que son cruciales para el éxito de proyectos software. Por ejemplo:

- La *Gestión de la Configuración* es crítica, sin embargo ¿cómo debe hacerse la gestión de la configuración para que un producto X sea integrado correctamente en un proceso específico en el proyecto Y?
- Identificar medidas y métricas de acuerdo al nivel de calidad requerido y a las políticas de control definidas es fundamental para obtener productos software de alta calidad. Pero, ¿Cómo son integradas estas medidas y métricas en el proceso de producción y mantenimiento de software?

- La Gestión de Proyectos es esencial para el éxito de un proyecto. ¿Cómo puede ser recogida y presentada la información al jefe del proyecto para que tenga una visión global del proceso actual?
- Durante el mantenimiento, todas las actividades que van desde "Control de Cambios" a "Corrección de Errores" han de estar bien coordinadas. ¿Cómo se garantiza la consistencia cuando dos "correctores" trabajan simultáneamente en la misma versión de una configuración?

Como ya se comentó, el último objetivo de la TPS es alcanzar el punto donde la representación de procesos puede ser usada para conducir los actuales procesos de desarrollo y mantenimiento del software. Como primer paso en esa dirección, la tecnología de procesos introduce la noción de **Modelo de Proceso (MP)**, una representación abstracta de una familia de procesos expresada en una adecuada notación de modelado de procesos (formalismo). De esta forma, la TPS hace uso de la considerable experiencia existente con representaciones formales y construcción de modelos. La utilidad de los modelos de procesos es múltiple: desde un simple soporte informal (ayuda a incrementar la comprensibilidad) hasta la asistencia directa en la evaluación y mejora de los procesos. Para facilitar la comprensión de los PS, Murer et al (1996) proponen utilizar una perspectiva tridimensional al consultar y "visualizar" los MP; con dos dimensiones estructurales (dependencias de objetos y flujos de trabajo concurrentes) y otra dimensión de evolución del proceso. La primera dimensión se refiere a que un PS y su ciclo de vida pueden representarse como un conjunto parcialmente ordenado de objetos (contratos, especificaciones, implementaciones, documentos, código, pruebas, etc.). La segunda dimensión tiene que ver con las transiciones de estado de los objetos (con restricciones indicando las permitidas). La tercera dimensión se refiere a poder ver la historia del proceso, incluidas las diferentes versiones que han existido de cada objeto.

En general, la disponibilidad de un MP computarizado (es decir, representable y manipulable mediante un computador) proporciona capacidades adicionales para:

- *Completar procesos*: dando soporte directo a los desarrolladores para el control de su trabajo, su coordinación con otros, etc.;
- *Automatizarlos*: permitiendo la invocación automática de herramientas no interactivas;
- *Dirigirlos*: facilitando el soporte indirecto, mediante la información del estado actual del proceso, el significado de los puntos de decisión, etc.
- *Incrementar su eficiencia*: puesto que un MP preciso es primordial en el aumento de la efectividad, ya que proporciona una base no ambigua para la comunicación entre los procesos.

Los MP también juegan un rol esencial en diversas actividades de ingeniería del software, entre las que merece destacar las siguientes:

- *Seguimiento*: porque los MP permiten una clara comprensión de lo que puede ser observado y por qué;
- *Simulación*: porque el comportamiento de un proceso puede ser estudiado al menor coste sin desarrolladores reales o herramientas.

- *Validación*: porque la simulación y supervisión, junto con la inspección de modelos, permiten que las propiedades de los modelos sean determinadas.
- *Verificación*: porque los MP son necesarios para probar formalmente las propiedades de interés del proceso.
- *Mejora*: porque todas las actividades anteriores son necesarias para la mejora de procesos.

El modelado de procesos puede entenderse desde dos puntos de vista distintos (McChesney, 1995): Un modelo descriptivo describe cómo un proceso es realizado en un entorno particular; y un modelo prescriptivo describe cómo un proceso debería ser realizado.

Los MP descriptivos tienen que ser exactos, es decir, deben describir fielmente los procesos reales. Una organización implicada en un proceso de mejora necesita utilizar ambos tipos de modelos: un modelo descriptivo para representar los procesos actuales a ser evaluados (actividad necesaria para mejorarlos); y un modelo prescriptivo para representar los procesos mejorados que se tienen como objetivo a alcanzar.

McChesney (1995) divide los MP prescriptivos en dos categorías: manuales y automatizados. Los MP manuales pueden ser estándares, metodologías y métodos centrados en los procesos del ciclo de vida del software (gestión, desarrollo, mantenimiento, evaluación, soporte organizacional, etc.). Los MP automatizados son especificaciones computerizadas de estándares de PS. Su principal objetivo es actuar como guía en el proceso de modelado, es decir, ayudar a los agentes de proceso para interpretar mecánicamente los MP (Lonchamp, 1994). Los MP automatizados se subdividen en dos categorías: orientados a actividades (centrados en las funciones, actividades y partes de los PS) y orientados a personas (centrados en la especificación de las personas participantes y sus relaciones). En Acuña y Ferré (2001) se puede consultar una taxonomía completa de los MP prescriptivos, con ejemplos de cada una de las categorías y subcategorías.

Los tipos de información en un MP también pueden estructurarse desde diferentes puntos de vista. En este sentido, Curtis et al (1992) establecieron las siguientes perspectivas de información:

- *Funcional*: que representa los elementos de proceso que están siendo implementados y los flujos de información que son importantes para dichos elementos.
- *De Comportamiento*: que representa cuando y bajo qué condiciones son implementados los elementos de proceso.
- *Organizacional*: que representa donde y por quién (en una organización) son implementados los elementos de proceso.
- *Informativa*: que representa las entidades de información resultantes o manipuladas por un proceso, incluyendo su estructura y relaciones.

2.4.1. Elementos Básicos.

Cualquier MP debe incluir un conjunto mínimo de elementos o conceptos. En ingeniería del software se considera que dichos componentes mínimos son los siguientes (Derniame et al, 1994):

- Una **actividad** es una operación atómica o compuesta, o un paso de un proceso. Las actividades se encargan de generar o modificar un determinado conjunto de **artefactos**; para lo cual siguen o están basadas en unos determinados procedimientos, reglas y políticas (**normas**). Además, una actividad es un concepto con un componente funcional fuerte ya que implica entradas, salidas, y resultados intermedios.
- El conjunto de artefactos a ser desarrollados, entregados y mantenidos en un proyecto es lo que se denomina el **producto**. Normalmente existe algún tipo de correspondencia entre la descomposición de actividades en sub-actividades y del producto en subproductos, pero no es obligatorio.
- Un **recurso** es un activo que una actividad necesita para poder ser llevada a cabo. En ingeniería del software hay dos clases principales de recursos: los **desarrolladores**, es decir, los **agentes** humanos que intervienen en el proceso; y las **herramientas**, es decir, los agentes computarizados (software) que tradicionalmente han sido usados en desarrollo y mantenimiento del software (herramientas CASE como editores especializados, compiladores, etc.) o herramientas de propósitos generales como hojas de cálculo, editores de diagramas, etc., que pueden ser utilizados para gestionar el proceso.
- Normalmente, las herramientas tienen una relación fuerte con las actividades en las que son usadas, mientras que los desarrolladores están vinculados indirectamente con una actividad por medio de sus **roles**, es decir, el conjunto de responsabilidades, obligaciones y tareas (por ejemplo, diseñador, jefe de proyecto, revisor, etc.). El carácter de la **organización** impacta en el proceso indirectamente por medio de los roles definidos, y directamente por medio de las **normas** (políticas, reglas, y procedimientos) que gobiernan las actividades. Las normas suelen aparecer en forma de documentos, y por tanto, suele existir una jerarquización entre normas y sub-normas. La TPS ayuda a hacer cumplir estas directrices proporcionando soporte computacional para guiar y controlar las iniciativas de los desarrolladores, para coordinar sus actividades, o para soportar la cooperación y/o automatización. De esta forma, la tecnología de procesos contribuye a aumentar el nivel de confianza en la calidad del producto, ya que asegura mayor adherencia a los estándares de calidad de procesos.

No todos los autores consideran el mismo conjunto básico de elementos de proceso, aunque existe consenso en cuanto a considerar los siguientes: Agentes/Actores/Desarrolladores, Roles, Actividades, y Artefactos/Productos (Dowson et al, 1991), (Feiler y Humphrey, 1993). En (Acuña y Ferré, 2001) se puede consultar una comparativa de las principales propuestas.

2.4.2. Niveles y Dominios.

Los procesos pueden ser representados con diferentes niveles de detalle, capturando clases cada vez más pequeñas de sub-procesos, correspondientes a asuntos cada vez más detallados. Normalmente, las descripciones más genéricas, como los ciclos de vida, suelen

ser informales o semi-formales, mientras que en el otro extremo los procesos operacionales de soporte suelen necesitar una descripción formal en un nivel bastante detallado. En función de este nivel de detalle o abstracción, en la bibliografía han sido establecidas las siguientes distinciones:

- Un ciclo de vida es una representación general e informal de procesos software. El propósito de estos modelos es dirigir aspectos metodológicos globales. Por ejemplo, resaltar que se debe especificar antes de codificar (modelo en cascada; Royce, 1993), o que se deben calcular los riesgos antes de la especificación (modelo en espiral; Boehm, 1981), o ir desde la especificación al código y no al contrario (modelo transformacional; Balzer et al, 1983).
- Un MP genérico es una representación abstracta, que puede ser usada en muchos proyectos similares y organizaciones que comparten propiedades y características comunes. Por ejemplo, puede ser una representación formal de un ciclo de vida o de un proceso de reingeniería del software.
- Un MP adaptado (*customized*) es bastante más detallado que un MP genérico. Normalmente se deriva de un modelo genérico teniendo en cuenta características específicas locales, es decir, dominios de aplicación. En esta clase se incluyen modelos con diferentes niveles de detalles, por ejemplo: un estándar internacional, adaptación (*tayloring*) de un estándar a las necesidades de una organización, o particularización para ajustarse a las necesidades de proyectos específicos.
- Un MP reificable (*enactable*) tiene el mayor nivel de detalle, de forma que define el proceso concreto y exacto que debe ser seguido en un proyecto específico. Desde la perspectiva de la tecnología de procesos software, un MP reificable es una representación lista para ser cargada en una máquina y proporcionar soporte automático a la gente que lleva a cabo procesos de desarrollo o mantenimiento de software.
- Finalmente, un MP reificado (*enacting*) es un modelo que dá soporte a un proyecto real y concreto, para lo cual incluye algún tipo de representación del estado actual del proceso.

De forma ortogonal a la anterior clasificación, se distinguen dos dominios de aplicación en cada uno de dichos niveles (Dowson y Fernström, 1994). El dominio de realización de procesos se refiere al acto de participar las personas y las herramientas en un proceso. El dominio de reificación de procesos está relacionado con el acto de conducir automáticamente un proceso, es decir, interpretar con más detalle el modelo del proceso. En el ámbito de la tecnología de procesos software, las herramientas y las personas también pertenecen al dominio de reificación de procesos ya que son controladas por el entorno. En otras palabras, mientras que un proceso se lleva a cabo en el mundo real, es reificado en el mundo abstracto del modelo computarizado, pero puesto que los computadores son parte del mundo real, se dice que la reificación es parte de la realización. Es precisamente por la pertenencia a ambos dominios por lo que, contrariamente a lo que se pudiera pensar, los desarrolladores humanos pueden contribuir creativamente en el desarrollo y mantenimiento de software, incluso cuando se trabaja bajo las restricciones de la reificación en un PSEE. Esta multiplicidad de dominios y la idea de múltiples vistas, son aspectos inherentes a la naturaleza compleja de los procesos software.

2.4.3. Vistas.

De forma similar al concepto de vistas en el campo de las bases de datos, las vistas de un MP expresan un punto de interés particular en vez del MP completo. Ejemplos típicos son:

- El *modelo de actividades*, que se centra en los tipos, estructura y propiedades de las actividades del proceso y sus relaciones. Esta vista es relevante, por ejemplo, al jefe de proyecto para propósitos de organización del calendario.
- El *modelo de productos*, que describe los tipos, estructura y propiedades de los elementos software de un proceso. Además de para los miembros del equipo del proyecto, esta vista puede ser de interés también para el usuario, por ejemplo, para comprender qué tipo de documentación se elaborará.
- El *modelo de recursos*, que describe los recursos que se necesitan o se suministran a los procesos, que es relevante desde una perspectiva de gestión de proyectos (costes, calendarios, riesgos).
- El *modelo de roles*, que describe un conjunto de recursos peculiar, como son las habilidades que los desarrolladores poseen y las responsabilidades que aceptan. Esto es relevante, además de para el equipo de desarrollo del proyecto, para la organización responsable y, en particular, para el personal de aseguramiento de la calidad.

Obviamente, una vista no puede ser definida sin usar conceptos de otras. Por ejemplo, una vista de actividades puede necesitar algunos elementos de una vista de productos para que sea comprensible, o bien, una vista de productos puede llevar información sobre la estructura de un producto, reduciendo las actividades a meras entradas y salidas. Cabe destacar que hay dos tendencias con respecto a como utilizar la idea de diferentes vistas en un MP comprensivo. La primera, con una perspectiva *top-down* inspirada en la experiencia con la gestión de bases de datos, considera las vistas como submodelos de un modelo único global y obliga a la consistencia de las diferentes vistas. La segunda aproximación, de tipo *bottom-up*, asume como punto de partida una colección de vistas (posiblemente inconsistentes), que el diseñador puede combinar negociando las inconsistencias mediante un soporte automático (Nuseibeh et al, 1993).

2.4.4. Sistemas de Procesos Software.

Algunos autores establecen una clasificación en niveles de los procesos de ingeniería del software. En este sentido, Wang y King (2000) establecen la siguiente taxonomía (en orden de menor a mayor jerarquía):

- *Práctica*: Una actividad o un estado en un PS para llevar a cabo una tarea específica. Es la unidad mínima que puede ser modelada.
- *Proceso*: Es un PS tal como ya se ha definido. Expresado en función del concepto anterior, un proceso es un conjunto de prácticas funcionalmente coherentes y reutilizables para un proyecto software.
- *Categoría*: Es un conjunto de procesos funcionalmente coherentes y reutilizables en algún aspecto de la ingeniería del software.

- *Subsistema*: Es un conjunto de categorías funcionalmente coherentes y reutilizables en alguna parte principal de la ingeniería del software.
- *Sistema*: Conjunto completo de PS estructurados.

Algunos de los sistemas de PS más conocidos usando niveles de procesos son: CMM, Bootstrap e ISO 15504 incluyen 18, 23 y 35 procesos, respectivamente.

Complementario a la taxonomía anterior, los miembros de un sistema de PS se pueden clasificar según una jerarquía funcional. Wang y King (2000) proponen la jerarquía, donde un sistema de PS se puede analizar desde tres puntos de vista: el modelo de procesos, el modelo de evaluación y el modelo de mejora. A su vez, el modelo de procesos se subdivide en tres subsistemas: organización, producción (desarrollo y mantenimiento), y gestión.

2.5. Evolución de Procesos Software.

En un entorno de producción de software se deben considerar dos aspectos:

1. Un proceso, P, es decir, el proceso de producción del mundo real que incluye actores humanos, herramientas CASE, etc.; y que sirve de guía para llevar a cabo los actividades de desarrollo y/o mantenimiento de productos software, y
2. Un modelo de procesos, MP, que es una representación del mundo real, y captura el estado actual de las actividades para dirigir, hacer cumplir o automatizar partes del proceso de desarrollo o de mantenimiento.

Idealmente, P y MP deberían estar perfectamente alineados en el sentido de que el estado interno del modelo de proceso debe ser una fiel representación del estado actual de los asuntos en el mundo real, es decir, "*P es una instancia de MP*". Sin embargo, cualquier proceso software en el mundo real es un proceso creativo y dinámico que abarca numerosos actores humanos y no puede ser reducido a la mera programación de autómatas. Puesto que un grado de inconsistencia entre P y MP puede traer consecuencias negativas para la calidad final del producto, el modelo de procesos debe adaptarse a cualquier evolución del proceso en el mundo real, además de coordinar las actividades y gestionar los flujos de información. Son varias las razones por las que puede cambiar un proceso software (Madhavji, 1991):

- ser erróneo;
- ciertos pasos importantes no están previstos, lo que hace ineficaz al proceso;
- el MP puede ser genérico y necesita ser detallado para obtener resultados específicos;
- las presunciones sobre las cuales se construyó el MP ya no son válidas; ó
- las dinámicas políticas, humanas y tecnológicas pueden inducir el cambio.

Como resultado de lo anterior, el proceso P del mundo real y el modelo de procesos MP deberían evolucionar de forma conjunta y coherente; aunque esta afirmación no siempre se puede cumplir; En esta línea, Heimbigner (1993) establece dos clases de cambios:

- *Cambios Definicionales*: que ocurren cuando se detecta un error o inadecuación en el MP. El problema es modificar el MP y a continuación propagar los cambios a todas las instancias de ejecución (procesos P del mundo real) de dicho MP.
- *Cambios en Instancias*: que ocurren cuando, por razones imprevistas, el proceso real P debe ser alterado aunque el MP sigue considerándose válido (por ejemplo, una revisión de diseño no se lleva a cabo por problemas de tiempo). La ocurrencia repetida del mismo cambio en varias instancias puede implicar la necesidad de un cambio definicional. En este ámbito, Kobialka (1998) ha realizado una propuesta muy completa para dar soporte al cambio de procesos durante su reificación.

Como conclusión general de lo anterior, se puede decir que la evolución de un proceso software es, en sí misma, un proceso completo de alto nivel, conocido como **meta-proceso**, cuyo principal objetivo es asegurar que P y MP son consistentes entre sí. Este meta-proceso incluye los pasos para cambiar los procesos del mundo real y los pasos para introducir cambios en el MP. En consecuencia, para llevarlo a cabo correctamente, el director del proyecto necesitará implicar los servicios del modelador de procesos para diseñar un modelo aumentado, validar este nuevo modelo, y después decidir cuando comenzar a reificarlo. Conradi y Jaccheri (1999) proponen las siguientes fases en el ciclo de vida del meta-proceso:

1. Elicitación y especificación de requisitos del proceso.
2. Análisis del proceso.
3. Diseño del proceso.
4. Implementación del proceso.
5. Reificación del proceso.
6. Evaluación de la calidad y rendimiento del proceso.

2.6. Lenguajes de Modelado de Procesos.

Un lenguaje de modelado de procesos (LMP) software permite expresar los procesos software en base a un determinado modelo de procesos utilizando una determinada sintaxis. Para ello, es necesario poder describir todos los elementos de proceso (actividades, roles, recursos, etc.) y los elementos del meta-proceso (relacionados con la evolución del proceso).

Son 6 las clases de elementos de proceso (elementos primarios) que debe poder modelar un LMP: actividades, productos, roles, personas, herramientas, y soporte para la evolución. El LMP debe ofrecer soporte para la evolución del MP, tanto a nivel técnico (por ejemplo, mediante la reflexión o interpretación) como a nivel conceptual (mediante un metamodelo asociado). Además de estos elementos primarios o principales, los LMP suelen incluir la posibilidad de representar elementos propios del meta-proceso (elementos secundarios).

Los más habituales son:

- **Proyectos/Organizaciones:** Una organización está formada por personas relacionadas con otras y elementos de otros tipos. Un proyecto es una estructura temporal de una organización montada para poder alcanzar un objetivo específico.
- **Contextos de Trabajo :** Están formados por espacios de trabajo, cada uno de los cuales engloba y controla artefactos para un (sub)proceso. Estos artefactos suelen ser ficheros de un repositorio.
- **Vistas de Usuario:** Son el interfaz general para ayudar al usuario a comprender el MP y guiarlo durante su reificación. A este nivel se distingue entre el llamado “modelo interno” (que muestra cómo funciona el proceso) y el “modelo externo” (que muestra cómo se hacen las cosas).
- **Modelo de Cooperación:** En el se indican los modos de cooperación (secuencial o paralelo), los protocolos de comunicación entre objetos y la coordinación de acciones (ordenación o sincronización).
- **Modelo de Versionado/Transacciones.**
- **Modelo de Calidad/Rendimiento.**
- **Modelo de Calidad del Producto:** Representa los objetivos de calidad del producto y las métricas asociadas.
- **Modelo de Rendimiento del Proceso:** Para expresar el cumplimiento con respecto a tiempos, costes, roles, etc.

Otro aspecto que va a influir en las características y naturaleza de los LMP es que pueden ser utilizados con propósitos diversos:

- **Comprender el PS:** Un LMP puede ser utilizado para representar de forma precisa cómo está estructurado y organizado un proceso. Esto es una ayuda para eliminar inconsistencias en la especificación del proceso.
- **Diseñar el PS :** Proactivamente, un LMP puede usarse para diseñar un nuevo PS, describiendo su estructura y organización.
- **Entrenamiento y formación:** Una descripción precisa del PS puede ser útil para enseñar los procedimientos y operaciones de una empresa al personal recientemente incorporado.
- **Simulación y optimización del PS:** Una descripción de un PS puede ser simulada para evaluar los posibles problemas, cuellos de botella, y oportunidades de mejora.
- **Soporte al PS reificado:** Una descripción precisa de un PS puede ser interpretada por una computadora y, por tanto, utilizada para proveer diferentes niveles de soporte a las personas que llevan a cabo el proceso.

Por otro lado, un MP es desarrollado, analizado, refinado, transformado y reificado dentro del meta-proceso. Por tanto, además de modelar adecuadamente el proceso del mundo real, un LMP se usa de diferente forma por diferentes roles durante las diferentes fases del ciclo de vida del meta-proceso. En consecuencia, en cada fase interesan unas características diferentes en el LMP. Por ejemplo, durante la “Especificación de requisitos” interesa un lenguaje orientado al modelado conceptual, intuitivo y con notación fácil para los usuarios no técnicos (por ejemplo, basado en gráficos). En cambio, durante la “Implementación del proceso” el lenguaje debe permitir el suficiente detalle para que el MP sea reificado, es decir, el LMP debe ser ejecutable (formal). En general, las propiedades deseables en un LMP son (Conradi y Jaccheri, 1999):

- *Formalidad*: Tal como ya se ha dicho, el nivel adecuado de formalidad (es decir, la elección de un lenguaje formal, semi-formal o informal) depende de la fase del meta-proceso en que nos encontremos.
- *Expresividad*: Su máximo nivel se consigue cuando todos los elementos del MP pueden ser directamente representados por el LMP. Franch y Ribó (1999b) han propuesto algunas ideas para mejorar la expresividad en base al uso de lenguajes estándares modulares y flexibles.
- *Comprensibilidad* (facilidad de comprensión): Se ha comprobado que lo mejor es utilizar lenguajes gráficos basados en metáforas en el caso de usuarios no técnicos; y aprovechar la experiencia con lenguajes de programación en el caso de usuarios técnicos.
- *Abstracción*: El uso de modelos abstractos que se instancian en MP concretos permiten manejar modelos a distintos niveles (ciclos de vida, modelos genéricos o detallados) y establecer correspondencias entre ellos.
- *Modularidad*: Estructurar un MP en sub-modelos interrelacionados ayudará a la reutilización de partes de un modelo.
- *Ejecutabilidad*: Definición de modelos operacionales que son ejecutables y fácilmente reificables.
- *Analizabilidad*: Definición de modelos descriptivos (por ejemplo, utilizando la lógica de predicados), que son fácilmente analizables.
- *Soporte de evolución*: La reflexión es muy importante para soportar la evolución de MP.
- *Múltiples vistas*: Soportar la definición de vistas consistentes desde diferentes perspectivas del MP. Para ello es necesario disponer de mecanismos de integración de vistas.

Al diseñar LMP que satisfagan las propiedades anteriores suficientemente se presentan bastantes problemas. En (Sutton et al, 1995a) se realiza un repaso de todos ellos y se detallan los más importantes: las dependencias entre los requisitos del proceso y los requisitos del lenguaje; la representación de los requisitos de ejecución; la amplitud, nivel y profundidad semánticos del lenguaje; y las meta-capacidades del proceso (metadatos, reflexión y dinamismo).

Son múltiples los campos de la Informática y de otras áreas de conocimiento que han aportado técnicas y herramientas útiles en los LMP. Los más significativos han sido:

- *Lenguajes de programación.*
- *Sistemas basados en reglas* (sistemas de producción, programación lógica, etc.).
- *Grafos y gramáticas.*
- *Redes de Petri.*
- *Orientación a objetos* (por ejemplo, utilizando extensiones de UML).
- *Notaciones estáticas* (entidad-interrelación).
- *Notaciones de comportamiento* (diagramas de transición de estados).
- *Notaciones funcionales* (diagramas de flujo de datos).
- *Gestión de proyectos* (diagramas de barras, redes de actividades, etc.).
- *Lenguajes de bases de datos* (bases de datos activas).
- *Herramientas CASE* y mecanismos de integración entre ellas.
- *Flujos de trabajo* (WorkFlows).

- Herramientas de *trabajo en grupo* (Groupware).

2.7. El Rol Humano en los Procesos.

La investigación en los aspectos sociales y humanos de los PS ha sido llevada a cabo principalmente por las comunidades de investigadores en Sistemas de Información, Ingeniería del Software e Interacción Humano-Computadora (Arbaoui et al, 1999). En este sub-capítulo lo que más interesa es la perspectiva aportada por la segunda, y más concretamente, la problemática originada al intentar aplicar un punto de vista humano (en vez del puramente tecnológico) en el diseño e implementación de EIS. En esta línea, a continuación se presenta el marco conceptual de Dowson (Dowson y Fernström, 1994), que provee una base útil para representar las relaciones clave entre agentes humanos, EIS orientados a procesos (es decir, PSEE's), y modelos de proceso; y tener en cuenta estas relaciones al implementar el soporte a procesos en los EIS. Dowson establece tres dominios diferentes en relación al PS:

- a) Definición del Proceso, que contiene las caracterizaciones de los procesos y fragmentos de procesos, expresadas en alguna notación, en términos de cómo dichos procesos podrían o deberían ser realizados.
- b) Reificación del [Modelo del] Proceso, que tiene que ver con lo que ocurre en un PSEE para soportar la realización del proceso.
- c) Realización del Proceso, que abarca las actividades reales o acciones dirigidas por agentes humanos (personal del proyecto: ingenieros software, gestores, etc.) y agentes no humanos (programas, herramientas del entorno, etc.).

Dowson distingue cuatro roles diferentes desempeñados por agentes humanos en un PS (son roles abstractos que no tienen correspondencia unívoca con personas o grupos concretos):

- *Diseñador del Proceso*: actúa en el dominio de definición del proceso. Define MP genéricos y gestiona su evolución. Estos modelos definen una estrategia global para el desarrollo o mantenimiento del software independientemente del contexto específico de un proyecto (Benali y Derniame, 1992). Ejemplos de este rol son un diseñador de una metodología, un estratega técnico en una empresa de software, un consultor especializado en implantar programas de mejora de procesos software, etc.
- *Gestor del Proceso*: actúa en los dominios de definición y de reificación del proceso. Adapta los MP genéricos a un proyecto específico y los instancia con el fin de obtener un modelo reificable. En un caso real, este cometido puede ser realizado por un gestor de proyecto trabajando en conjunción con el director técnico de la empresa.
- *Agente del Proceso*: actúa en el dominio de realización del proceso. Puede ser un gestor de proyecto, un programador, un analista de sistemas, un auditor de calidad, un responsable de pruebas, etc.
- *Usuario Final*: actúa en el dominio de realización del proceso. Es el cliente, es decir, el usuario del producto que está siendo desarrollado o mantenido. La realización del proceso pone en juego a los tres dominios citados, ya que todos los agentes mencionados, humanos o no, participan en un flujo dinámico de interacción (comunicando, coordinando, negociando, aconsejando, preguntando, etc.) dentro y fuera de los límites de los tres

dominios. Estas interacciones pueden ser de dos tipos: interacción de usuario e interacción interpersonal.

2.7.1. Interacción de Usuario.

Esta interacción se realiza entre los dominios de reificación y realización, es decir, entre los agentes de proceso y el MP reificado. Se refiere a actividades de carácter individual, como editar un módulo o leer un documento. Esta interacción es bidireccional:

a) En el sentido “de *reificación a realización*”, se refiere a la influencia del mecanismo de reificación de procesos sobre los actores en orden a determinar la forma en que un proceso es realizado. En este asunto se plantea el problema de decidir quién ejerce el control durante la realización del proceso, el PSEE o un actor humano. Esta relación de poder pueda adoptar las formas siguientes (enumeradas de menor a mayor control por parte del PSEE):

- *Guía Pasiva*: el mecanismo de reificación facilita a los actores, cuando lo piden, información para ayudarles a llevar a cabo el proceso.
- *Guía Activa*: el mecanismo de reificación facilita ayuda sin que sea pedida por los agentes de proceso, es decir, el MP especifica cuándo y cómo debe ofrecerse dicha ayuda.
- *Proceso Dirigido*: los agentes son obligados a realizar una parte del proceso de una determinada manera (por ejemplo, controlando su acceso a los datos y herramientas).
- *Proceso Automatizado*: alguna parte del proceso es realizada automáticamente bajo el control del mecanismo de reificación sin ninguna participación de los actores humanos.

b) En el otro sentido, “de *realización a reificación*”, se refiere al enlace recíproco entre los agentes del proceso y el MP reificado en orden a proveer realimentación sobre el estado actual de la realización del proceso. Esta realimentación es crítica para asegurar que los “estados” de estos dos dominios estén alineados entre sí y sean tan consistentes como sea posible. Se pueden distinguir dos tipos de realimentación:

- De primer orden, que se refiere simplemente a informes del progreso de acuerdo con el modelo prescrito; y
- De segundo orden, que se refiere a los cambios locales que han sido hechos en el proceso en respuesta a exigencias prácticas (adaptación del proceso).

2.7.2. Interacción Interpersonal.

Este tipo de interacción se refiere a la interacción entre diferentes actores de proceso, por ejemplo, un analista comentando un documento de diseño con un programador, o una reunión para inspección de código. Se distinguen dos clases distintas de interacción interpersonal:

- *Formal*, si ocurre siguiendo unos protocolos bien definidos y formalizados. Un ejemplo de esta clase es una guía escrita que indica que si un programador tiene algún problema deberá informar al ingeniero de diseño.
- *Informal*, que ocurre cuando no está regulada por ninguna clase de protocolo de comunicación formal. Ejemplos de esta clase son: una llamada telefónica, un correo

electrónico o una charla en la cafetería. La mayoría de interacciones en un proyecto software son de esta naturaleza y tienen lugar dentro del dominio de realización, ya que no están mediatizadas o reguladas por el mecanismo de reificación y no se prevén en el diseño del proceso. Estas interacciones son cruciales para el éxito de un proyecto software ya que permiten que la información y la experiencia sean intercambiadas.

Las herramientas de soporte a las actividades interpersonales pueden tener un importante papel en el proceso software. De hecho, el campo del CSCW (*Computer Supported Collaborative Work*) ha sido la fuente de bastantes aportaciones útiles para dar soporte a las interacciones formales en un PS. En cambio, para las interacciones informales lo más importante es disponer de facilidades de comunicación en el dominio de realización (correo electrónico, boletines electrónicos, audio y/o videoconferencia, etc.).

2.8. Modelos de Proceso para Mantenimiento.

Entre las diversas propuestas o normas, publicadas a nivel internacional, útiles para modelar el proceso de mantenimiento del software (PMS), nos vamos a centrar en los estándares ISO correspondientes. Para su selección hemos tenido en cuenta que, desde una perspectiva funcional, se distinguen siete categorías diferentes de estándares para ingeniería del software (ISO/IEC, 2000a): proceso, producto, herramientas, tecnologías, recursos (humanos y materiales), y datos (de requisitos, de producto o de ingeniería). Una visión muy completa de los diferentes estándares de ingeniería del software propuestos por las principales organizaciones mundiales (IEEE, ISO/IEC, ESA, EIA, AIAA, PMI, etc.) puede consultarse en Moore (1998).

Por razones evidentes, los estándares de más interés para este trabajo, que presentaremos a continuación, pertenecen a la categoría de proceso: ISO 12207 (1995) para los procesos que forman el ciclo de vida de un producto software e ISO 14764 (1998e) para un modelo del proceso de mantenimiento. También incluimos la norma ISO 15504 para la evaluación de procesos porque, aunque está incluida en la categoría de tecnología, incluye un modelo de referencia para procesos software que completa el estándar 12207.

2.8.1. Ciclo de Vida: ISO12207.

La norma ISO 12207 “establece un marco de referencia común para los procesos del ciclo de vida software, con una terminología bien definida, que puede ser referenciada por la industria software” (ISO/IEC, 1995). En este marco se definen los procesos, actividades (que forman cada proceso) y tareas (que forman cada actividad) presentes en la adquisición, suministro, desarrollo, operación y mantenimiento del software. Según esta norma, un proceso es un conjunto de actividades interrelacionadas que transforman entradas en salidas, definiendo quién está haciendo qué y cuándo, y cómo alcanzar un determinado objetivo. Se presentan los tres documentos oficiales referidos a esta norma: el propio estándar, una mejora posterior y una guía de utilización.

ISO 12207 establece una arquitectura de alto nivel del ciclo de vida de un producto software, desde su concepción hasta su retirada. La arquitectura está construida con un

conjunto de procesos e interrelaciones entre ellos. La definición de dichos procesos está basada en dos principios elementales (ISO/IEC, 1998a):

- *Modularidad*: ya que todas las partes de un proceso están fuertemente relacionadas (cohesión fuerte) y el número de interfaces entre procesos se mantiene al mínimo (acoplamiento bajo).
- *Responsabilidad*: Cada proceso es responsabilidad de una parte, es decir, de un rol.

Una organización puede realizar uno o varios procesos. Un proceso puede ser realizado por una o varias organizaciones (pero con una de ellas identificada como responsable). Este principio de responsabilidad facilita la adaptación a organizaciones y proyectos de cualquier tamaño y características.

Cada proceso se considera formado por una o varias actividades. A su vez, cada actividad está formada por una o varias tareas. Cada tarea se expresa en la forma de un requisito, una declaración, una recomendación o una acción permitida. Esta norma clasifica los procesos que pueden ser realizados durante el ciclo de vida del software en cuatro categorías:

- **Procesos Principales**: Son procesos que dan servicio a las partes principales durante el ciclo de vida del software. Una parte principal es la que inicia o lleva a cabo el desarrollo, operación o mantenimiento de productos software. Estas partes principales son el adquisidor, el suministrador, el desarrollador, el operador y el mantenedor de productos software.
- **Procesos de Soporte**: Apoyan a otro proceso, como parte esencial del mismo, con un propósito bien definido y contribuyen al éxito y calidad del proyecto software. Un proceso de soporte se emplea y ejecuta por otro proceso según sus necesidades.
- **Procesos Organizacionales**: Se emplean por una organización para establecer e implementar una infraestructura constituida por procesos y personal asociados al ciclo de vida, y para mejorar continuamente esta estructura y procesos.
- **Proceso de Adaptación**: Este proceso especial define las actividades básicas necesarias para llevar a cabo adaptaciones de esta norma, es decir, adecuarla a las características especiales de cada proyecto u organización.

2.8.1.1. Procesos Principales.

Como su nombre indica, son los procesos fundamentales en el ciclo de vida de un producto software. Los cinco procesos de esta clase son:

- **Adquisición**: Define las actividades del adquisidor (organización que adquiere un sistema, producto software o servicio software). Su objetivo es obtener el producto y/o servicio software que satisfaga una necesidad expresada por el cliente.
- **Suministro**: Define las actividades del suministrador (organización que proporciona el sistema, producto software o servicio software al adquisidor). Su objetivo es proveer un producto o servicio software que cumpla los requisitos acordados.
- **Desarrollo**: Define las actividades del desarrollador (organización que define y desarrolla el producto software). Su objetivo es transformar un conjunto de requisitos en un

producto software funcional o un sistema basado en software, que satisfagan las necesidades establecidas por los clientes. Las actividades de este proceso son llevadas a cabo por uno de dos roles diferentes, el Desarrollador del Sistema o el Desarrollador del Software, aunque alguna actividad debe ser realizada por ambos (Evaluación del Producto).

- **Operación:** Define las actividades del operador (organización que proporciona el servicio de operar un sistema informático en su entorno real a los usuarios). Su objetivo es operar (explotar) el producto software en el entorno deseado y dar soporte a los clientes de dicho producto.
- **Mantenimiento:** Define las actividades del mantenedor (organización que proporciona el servicio de mantenimiento del producto software, esto es, la gestión de las modificaciones al producto software para mantenerlo actualizado y operativo). La finalidad de este proceso es ser sensible a las peticiones del cliente y gestionar las modificaciones, migraciones y retiradas de productos asociadas con ellas. El objetivo concreto es modificar y/o retirar sistemas o productos software preservando la integridad de las operaciones organizacionales.

2.8.1.2. Procesos de Soporte.

Los procesos de soporte se aplican en cualquier momento del ciclo de vida. Se pueden utilizar para dar ayuda a otro proceso de cualquier clase: principal, otro de soporte, organizativo o de adaptación. Los nueve procesos de esta categoría son los siguientes:

- **Documentación:** Define las actividades para el registro de la información producida por un proceso del ciclo de vida. Su propósito es elaborar y mantener registrada la información producida por un proceso.
- **Gestión de la Configuración:** Define las actividades de gestión de la configuración. Su objetivo es establecer y mantener la integridad de todos los productos de trabajo de un proceso o proyecto, y hacerlos disponibles a las partes interesadas.
- **Aseguramiento de la Calidad:** Define las actividades para asegurar, de una manera objetiva, que los productos de trabajo y los procesos son conformes a la especificación de requisitos y se ajustan a los planes establecidos. Son técnicas útiles en este proceso las revisiones conjuntas, auditorías, verificaciones y validaciones.
- **Verificación:** Define las actividades (para el adquirente, proveedor u otra tercera parte independiente) para verificar los productos software hasta un nivel de detalle adecuado al proyecto software de que se trate. Su objetivo es confirmar que cada producto de trabajo y/o servicio software de un proceso o proyecto cumple adecuadamente los requisitos especificados.
- **Validación:** Define las actividades (para el adquirente, proveedor u otra tercera parte independiente) para validar los productos software adecuados para un determinado proyecto. Su propósito es confirmar que los requisitos para un uso específico deseado de un producto de trabajo software son satisfechos.
- **Revisión Conjunta :** Define las actividades para evaluar el estado y productos de una actividad. Este proceso puede ser empleado por dos partes cualesquiera, donde una de las partes (la revisora) revisa a la otra parte (la revisada), de una manera conjunta. El objetivo de este proceso es mantener un conocimiento compartido con el cliente de los progresos realizados frente a los objetivos del contrato, así como de lo que debería hacerse para

garantizar el desarrollo de un producto que satisfaga al cliente. Las revisiones conjuntas pueden ser a nivel técnico o de gestión del proyecto y ocurren a lo largo de toda la vida del proyecto.

- **Auditoria:** Define las actividades para determinar –de forma independiente- el cumplimiento por parte de los productos y procesos seleccionados de los requisitos, planes y contratos. Este proceso puede ser empleado por dos partes cualesquiera, donde una parte (la auditora) audita los productos software o actividades de otra parte (la auditada).
- **Resolución de Problemas:** Define las actividades para analizar y eliminar los problemas (incluyendo las no conformidades) que sean descubiertos durante la ejecución de otros procesos (desarrollo, operación, mantenimiento, etc.), cualesquiera que sea su naturaleza o causa. Su objetivo es asegurar que todos los problemas descubiertos son analizados y resueltos.
- **Usabilidad:** Incluye las actividades necesarias para asegurar que se están teniendo en cuenta los intereses y necesidades de los individuos y /o grupos que trabajarán con el sistema (usuarios). Este proceso garantiza la calidad de uso del software.

El proceso de Usabilidad es posterior al estándar oficial (ISO/IEC, 1995). Este proceso ha sido incorporado en la mejora del estándar (ISO/IEC, 2000b).

2.8.1.3. Procesos Organizacionales.

Los procesos de esta clase ayudan a establecer, implementar y mejorar la estructura y los procesos de una organización, contribuyendo a hacerla más eficiente. Habitualmente se llevan a cabo a nivel organizacional (corporativo), fuera del ámbito de proyectos y contratos específicos. Los siete procesos de esta clase son:

- **Gestión:** Define las actividades básicas de gestión (gestión de proyectos, gestión de calidad, gestión de riesgos, ...) durante un proceso del ciclo de vida. El objetivo general es organizar, supervisar y controlar la iniciación y realización de cualquier proceso o función para alcanzar sus objetivos en consonancia con los objetivos de negocio de la organización. Este proceso es establecido para asegurar la aplicación consistente de prácticas por parte de la organización en cada uno de los proyectos que lleva a cabo.
- **Mejora:** Define las actividades básicas que una organización (adquisidor, suministrador, desarrollador, operador, mantenedor o el gestor de otro proceso) lleva a cabo para establecer, evaluar, medir, controlar y mejorar un proceso del ciclo de vida del software.
- **Gestión de Recursos Humanos:** Define las actividades para conseguir personal formado adecuadamente. El objetivo es proveer a la organización y a los proyectos con individuos que posean habilidades y conocimientos para realizar sus roles de forma eficiente y para poder trabajar juntos como un grupo cohesionado.
- **Infraestructura:** Define las actividades básicas para establecer la infraestructura de un proceso del ciclo de vida. El propósito de este proceso es mantener una infraestructura estable y fiable necesaria para poder realizar cualquier otro proceso. Son infraestructuras para el ciclo de vida del software las siguientes: hardware, software, métodos, herramientas, técnicas, estándares, y otros medios útiles para desarrollar, operar o mantener software.
- **Gestión de Activos:** Define las actividades necesarias para gestionar la vida de activos (*assets*) reutilizables, desde su concepción hasta su retirada. Consiste en aplicar

procedimientos administrativos y técnicos para su identificación, definición, certificación, clasificación, modificación, almacenamiento, recuperación y retirada.

- **Gestión de la Reutilización:** Define las actividades para planificar, establecer, gestionar, controlar y supervisar un programa de reutilización en la organización, y el aprovechamiento sistemático de las oportunidades de reutilización.
- **Ingeniería de Dominios:** Define las actividades para desarrollar y mantener modelos de dominios, arquitecturas de dominios y activos para los dominios. La mejora del estándar (ISO/IEC, 2000b) ha cambiado el nombre del proceso de "Formación" a "Gestión de Recursos Humanos". También ha añadido tres procesos nuevos: Gestión de Activos, Gestión de la Reutilización, e Ingeniería de Dominios.

2.8.1.4. Roles y Organizaciones participantes.

Se representan los procesos del ciclo de vida del software y sus relaciones bajo diferentes puntos de vista de uso: contrato, gestión, operación, ingeniería y soporte. Cada punto de vista viene determinado por los intereses de un rol (parte) distinto. Desde la visión contractual, los roles adquisidor y suministrador negocian y se someten a un contrato empleando los procesos de Adquisición y Suministro, respectivamente. Bajo el punto de vista de gestión, una de las partes (adquisidor, suministrador, desarrollador, operador, mantenedor u otro) gestiona sus respectivos procesos. Bajo la visión de operación, el operador proporciona el servicio de operación del software para sus usuarios. Bajo el punto de vista de ingeniería, el desarrollador o mantenedor lleva a cabo sus respectivas tareas de ingeniería para producir o modificar los productos software. Por último, desde la visión de soporte, unos roles proporcionan servicios de apoyo a otros roles para completar tareas únicas y específicas. Los procesos organizacionales, que se emplean por una organización, a nivel corporativo, para establecer e implementar la estructura subyacente compuesta por los procesos y el personal asociados al ciclo de vida y mejorarlos continuamente.

Se puede considerar otro punto de vista de gestión de la calidad que engloba a los siguientes procesos de soporte: Aseguramiento de la Calidad, Verificación, Validación, Revisión Conjunta, Auditoría y Usabilidad. Todos estos procesos se emplean para gestionar la calidad a lo largo del ciclo de vida del software. Los procesos de Verificación, Validación, Revisión Conjunta y Auditoría se pueden emplear por diferentes roles separadamente (auditor, desarrollador, cliente, etc.) o como técnicas del proceso de Aseguramiento de la Calidad.

Los procesos y organizaciones (o roles) están sólo relacionados funcionalmente, es decir, los puntos de vista anteriores no implican ninguna estructura para ninguna organización o rol. Una organización (o un rol) toma el nombre del proceso que lleva a cabo; por ejemplo, se le llama adquisidor cuando lleva a cabo el proceso de Adquisición. Una organización puede llevar a cabo uno o varios procesos, es decir, puede desempeñar uno o varios roles. Un proceso puede ser llevado a cabo por una o varias organizaciones, es decir, un mismo rol puede ser desempeñado por varias organizaciones. La única restricción a lo anterior es que una misma parte no debería llevar a cabo simultáneamente los procesos de Adquisición y Suministro del mismo producto o recurso.

En este modelo de ciclo de vida, las relaciones entre procesos son sólo estáticas. Las relaciones dinámicas más importantes de la vida real, entre procesos, entre roles y entre procesos y roles se establecen automáticamente cuando este modelo se aplica en proyectos software concretos.

2.8.2. Modelo de Referencia para procesos: ISO 15504.

Para mejorar la calidad de un producto, una organización debe tener un método (probado, consistente y fiable) para evaluar el estado de sus procesos, y debe tener medios para usar los resultados como parte de un programa de mejora. Dentro de una organización, la evaluación de procesos incentiva tres aspectos importantes: la cultura de la mejora constante y el establecimiento de los mecanismos apropiados para apoyarla y mantenerla; la ingeniería de procesos para satisfacer los requisitos del negocio; y la optimización de recursos. Por tanto, la evaluación de procesos es importante para que las organizaciones maximicen su sensibilidad frente a los clientes y los requisitos de mercado, minimicen los costes del ciclo de vida de sus productos y, como resultado de ambos, mejore la satisfacción del usuario final. El estándar ISO 15504 (ISO/IEC, 1998b, 1998c, 1998d) aborda la estimación de procesos software desde dos perspectivas diferentes, la mejora de procesos y la determinación de la capacidad de las organizaciones. Aunque está formado por 9 partes diferentes, las que son de interés en este trabajo son las partes 1 (introducción y conceptos), 2 (modelo de referencia para procesos y capacidad de procesos), y 9 (vocabulario). Estos documentos junto con una mejora posterior del modelo de referencia de procesos software extendiendo los procesos propios del rol de adquisidor (ISO/IEC, 2001a).

De los dos contextos citados para la estimación de procesos, en este trabajo nos centramos únicamente en el de mejora, que es el que tiene relación más directa con el modelado de procesos software. En este contexto, la estimación de procesos provee un medio de caracterizar las prácticas actuales de una unidad organizacional en términos de capacidad de los procesos seleccionados. El análisis de los resultados de la estimación en relación con las necesidades de negocio de la organización ayuda a identificar fortalezas, debilidades y riesgos inherentes a los procesos. Esto, a su vez, posibilita determinar si los procesos son eficientes logrando sus objetivos, e identificar las causas más significativas de la falta de calidad o los excesos en tiempo y/o costes. Por último, conocidas dichas causas, se pueden determinar los cambios necesarios en el proceso propiamente dicho (y en consecuencia, en su modelo correspondiente) para mejorarlo.

El modelo de referencia de procesos propuesto en ISO 15504-2 (ISO/IEC, 1998c) tiene una arquitectura con dos dimensiones:

- La dimensión del proceso, caracterizada por descripciones del propósito de los procesos, las cuales, en esencia, son objetivos medibles de los procesos. Por tanto, define los procesos que pueden ser estimados.
- La dimensión de la capacidad de proceso, caracterizada por una serie de atributos del proceso, aplicables a cualquier proceso, que representan las características medibles necesarias para gestionar un proceso y mejorar la capacidad de realizarlo. Es decir, define la escala para medir las capacidades de los procesos.

Cada proceso en este modelo de referencia es definido por una descripción de propósito. Estas descripciones incluyen los objetivos funcionales del proceso cuando es instanciado en un entorno particular. También incluyen material adicional identificando los resultados que se obtienen al aplicar con éxito cada proceso. El modelo de referencia está basado en la norma ISO 12207 ya presentada, a la cual completa y mejora en algunos aspectos. De esta forma, en la dimensión del proceso se considera que los procesos del ciclo de vida del software se dividen en los mismos tres grupos que en ISO 12207 y, además, se establecen cinco categorías en las que se subdividen:

1. Los **procesos primarios** están divididos en dos categorías:

- Cliente-Suministrador (CUS), con los procesos que afectan directamente al cliente y permiten la operación y uso adecuado del producto y/o servicio.
- Ingeniería (ENG), que consiste en los procesos que directamente especifican, implementan o mantienen el producto software, su relación con el sistema y su documentación.

2. Los **procesos de soporte** forman una única categoría de igual nombre:

- Soporte (SUP), formado por los procesos que pueden ser empleados por cualquiera de los otros procesos (incluidos otros procesos de soporte) en diferentes momentos del ciclo de vida del software.

3. Los **procesos organizacionales** se dividen en dos categorías:

- Gestión (MAN), que engloba a los procesos que contienen prácticas de carácter general que pueden usarse por cualquiera que gestiona cualquier tipo de proyecto o proceso dentro del ciclo de vida de un producto software.
- Organización (ORG), formada por los procesos que establecen los objetivos de negocio de la organización y desarrollan los activos (de proceso, de producto o recursos) que ayudan a la organización a alcanzar dichos objetivos de negocios cuando son utilizados en los proyectos que lleva a cabo.

El modelo de referencia no define cómo, o en qué orden, deben conseguirse los objetivos de un proceso. Estos objetivos se alcanzan a través de diversas actividades detalladas, tareas y prácticas que deben ser llevadas a cabo para producir los productos de trabajo. La realización de estas actividades, tareas, y prácticas, junto con las características de los productos de trabajo obtenidos, demostrarán cuando está siendo alcanzado el propósito de un proceso específico.

Para detectar fácilmente las similitudes y diferencias con ISO 12207, consiste en la existencia de dos niveles de procesos (grupos y categorías) frente a únicamente grupos en ISO 12207. La segunda diferencia importante es la existencia de los siguientes nuevos procesos o cambios en procesos de ISO 12207:

- Elicitación de Requisitos (nuevo, principal/cliente-suministrador): El propósito de este proceso es recopilar, procesar y hacer un seguimiento de la evolución de las necesidades y

requisitos del cliente a lo largo de la vida del producto o servicio software, así como establecer una línea base (*baseline*) de requisitos que sirva como fundamento para definir los productos de trabajo necesarios.

- **Gestión del Proyecto** (nuevo, organizacional/gestión): Su objetivo es identificar, establecer, coordinar y supervisar las actividades, tareas y recursos necesarios en un proyecto para producir un producto o servicio que cumpla los requisitos.
- **Gestión de la Calidad** (nuevo, organizacional/gestión): Este proceso busca supervisar la calidad de los productos y/o servicios de un proyecto y asegurar que satisfaga al cliente. El interés se debe centrar en la calidad de los productos y procesos, tanto a nivel de proyecto como de la organización en su conjunto.
- **Gestión de Riesgos** (nuevo, organizacional/gestión): El propósito es identificar y mitigar los riesgos de un proyecto de forma continua a lo largo del ciclo de vida de un proyecto. Para ello se debe realizar una supervisión de los riesgos a nivel de proyecto y a nivel de la organización en general.
- **Alineamiento Organizacional** (nuevo, organizacional/organización): Su objetivo es asegurar que los individuos de la organización comparten una visión, cultura y conocimiento de los objetivos del negocio comunes, que les faculte para funcionar de forma eficiente.
- **Medida** (nuevo, organizacional/organización): El propósito de este proceso es recoger y analizar datos relativos a los productos desarrollados y a los procesos implementados por una unidad organizacional, para ayudar a la gestión eficaz de los procesos y a la demostración objetiva de la calidad de los productos.
- **Mantenimiento del Sistema y del Software** (cambio, principal/ingeniería): Mientras que en ISO 12207 este proceso se refiere únicamente al mantenimiento del software, ahora incluye también el hardware, redes, etc.
- **Reutilización** (cambio, organizacional/organización): Equivale al proceso de Gestión de la Reutilización en ISO 12207 cambiando su objetivo. Ahora el propósito es promover y facilitar la reutilización de productos de trabajo software desde una perspectiva organizacional y de producto/proyecto.
- **Gestión** (cambio, organizacional/gestión): El proceso de Gestión incluido en ISO 12207 ha sido dividido en varios: Gestión (en general), Gestión del Proyecto, Gestión de la Calidad, Gestión de Riesgos, Alineamiento Organizacional y Medida.
- Los procesos de Usabilidad, Gestión de Activos e Ingeniería de Dominios no están incluidos en la versión actual de ISO 15504 porque se incluyeron en la mejora de ISO 12207 elaborada con posterioridad. Cabe suponer que en la reforma de ISO 15504, en realización en la actualidad, se producirá una convergencia del modelo de procesos entre ambas normas corrigiendo estas divergencias.

Otra diferencia menor entre ambas normas tiene que ver con la nomenclatura utilizada. ISO 15504 establece dos niveles jerárquicos de procesos, de forma que algunos procesos (llamados básicos) están formados por otros procesos más simples (llamados componentes). En cambio, en ISO 12207 no existen procesos componentes sino actividades.

El documento de mejora de ISO 15504-2 (ISO/IEC, 2001a) se dedica únicamente a conceder más importancia al rol de adquirente de software. Para ello propone que el

proceso de Adquisición pase a ser una categoría dentro de los procesos principales formada por 17 procesos diferentes, que se corresponden con las actividades y/o tareas del proceso original.

2.8.3. El Proceso de Mantenimiento: ISO 12207 y 14764.

Como hemos visto anteriormente, la norma ISO 12207 considera el mantenimiento como uno de los procesos principales del ciclo de vida del software, que define las actividades y tareas del mantenedor (la organización que proporciona el servicio de mantener el producto software). A continuación se presenta el modelo de PMS establecido a grandes rasgos en ISO 12207 y detallado en el documento borrador del nuevo estándar ISO 14764 (ISO/IEC, 1998e).

Al igual que pasa con los demás estándares, en estos documentos únicamente se especifica cuáles deben ser las actividades y tareas del proceso de mantenimiento pero no se indica cómo realizarlas.

En ISO 14764 se considera que el PMS se activa cuando se modifica un producto software después de su entrega. Se distinguen cuatro **tipos de mantenimiento** diferentes según los objetivos de dicha modificación:

- Corregir problemas detectados (Correctivo).
- Mejorar la funcionalidad, o el rendimiento, mantenibilidad u otros atributos del software (Perfectivo).
- Hacer que siga siendo utilizable en un entorno nuevo o cambiado (Adaptativo).
- Detectar y corregir errores latentes en el software antes de que se conviertan en fallos reales (Preventivo).

Cuando el mantenedor recibe una **petición de modificación** (propuesta de cambio de un producto software que está siendo mantenido), según su naturaleza, implicará una corrección o una mejora en el software. Una corrección supone un mantenimiento de tipo correctivo, mientras que una mejora puede suponer un mantenimiento preventivo, adaptativo o perfectivo.

2.8.3.1. Actividades y Tareas.

En la definición de ISO (ISO/IEC, 1995 y 1998e), el PMS consta de las seis siguientes actividades:

A1: Implementación del proceso.

En esta actividad se desarrollan los planes (por ejemplo, el plan de mantenimiento) y procedimientos correspondientes para llevar a cabo las actividades y tareas del PMS. También se deben definir los procedimientos necesarios para la gestión de los problemas (empleando el proceso de resolución de problemas) y de las peticiones de modificación, e implementar el proceso de gestión de la configuración para gestionar los cambios en los elementos software existentes. Consta de tres tareas:

1. Elaborar, documentar y ejecutar planes y procedimientos para conducir las actividades y tareas del proceso.
2. Establecer procedimientos para recibir, almacenar y controlar los informes de problemas y solicitudes de modificación de los usuarios, y proporcionar a éstos realimentación.
3. Implementar o establecer una interfaz organizacional con el proceso de Gestión de la Configuración, para gestionar las modificaciones del sistema existente.

A2: Análisis de problemas y modificaciones.

Antes de modificar el software mantenido, el mantenedor deberá analizar los problemas o peticiones de modificación, elaborar y documentar distintas soluciones potenciales, y obtener la aprobación para implementar la opción elegida. En consecuencia, durante esta actividad, el mantenedor realiza las siguientes tareas:

4. Analizar el impacto de la petición o informe en la organización, el sistema existente y las interfaces con otros sistemas, determinando el tipo de mantenimiento (uno de los cuatro anteriores), el alcance (tamaño, coste y tiempo para hacer la modificación) y su criticidad (impacto en el rendimiento, la seguridad, etc.).
5. Replicar o verificar el problema.
6. Elaborar opciones para implementar la modificación.
7. Documentar el informe o petición, los resultados del análisis, y las opciones de implementación.
8. Obtener aprobación para la opción de modificación elegida.

A3: Realización de las modificaciones.

En esta actividad se incluyen todas las tareas relativas a determinar qué elementos del sistema software (documentación, unidades o módulos y versiones) deben modificarse, y se utiliza el proceso de desarrollo para implementar las modificaciones. Las tareas que lleva a cabo el mantenedor durante esta actividad son las siguientes:

9. Realizar un análisis para determinar qué documentación, unidades de software y versiones necesitan ser modificadas. Documentar esta información.
10. Entrar en el proceso de Desarrollo para implementar las modificaciones. Este proceso, según es definido en ISO 12207 (ISO/IEC, 1995), debe complementarse con lo siguiente:
 - a) definir y documentar criterios para probar y evaluar las partes del sistema modificadas y no modificadas;
 - b) comprobar la implementación completa y correcta de los requisitos nuevos o modificados;
 - c) asegurarse de que los requisitos originales no modificados no han sido afectados; y
 - d) documentar los resultados de estas pruebas.

A4: Revisión y aceptación del mantenimiento .

En esta actividad se asegura que las modificaciones al sistema son correctas y que cumplen los estándares aprobados utilizando una metodología correcta. Las dos tareas que lleva a cabo el mantenedor durante esta actividad son:

11. Realizar revisiones conjuntas entre el mantenedor y la organización que debe aprobar la modificación para determinar la integridad del sistema.
12. Obtener la aprobación, según los términos indicados en el contrato o acuerdo de mantenimiento, de que la modificación se ha completado satisfactoriamente.

A5: Migración.

Esta actividad se realiza cuando el sistema tiene que ser modificado para ser ejecutado en un entorno diferente. En ese caso, el mantenedor necesita determinar las acciones necesarias para lograr la migración, llevarlas a cabo, y documentar los efectos. Las tareas incluidas en esta actividad son las siguientes:

13. Asegurar que cualquier producto software o dato producido o modificado durante la migración lo ha sido de conformidad a lo establecido en la norma ISO 12207.
14. Elaborar y documentar un "plan de migración" en el que se incluirán, al menos, las siguientes cuestiones:
 - Análisis y definición de los requisitos de la migración.
 - Herramientas para la migración.
 - Conversiones del software y de los datos.
 - Ejecución de la migración.
 - Verificación de la migración.
 - Soporte futuro del entorno antiguo.
15. Notificar a los usuarios las intenciones de migración previstas, indicando lo siguiente:
 - a) explicación de porqué no se puede seguir soportando el entorno antiguo;
 - b) descripción del nuevo entorno, indicando la fecha de disponibilidad prevista;
 - y
 - c) descripción de otras opciones de soporte cuando el entorno antiguo haya sido eliminado.
16. Realizar operaciones en paralelo en ambos entornos, el viejo y el nuevo, para que la transición sea suave. A la vez, proveer a los usuarios de la formación necesaria.
17. Notificar a todos los afectados de que se ha completado la migración. Toda la documentación, elementos del sistema y datos del entorno antiguo deberá ser recopilada y archivada.
18. Realizar una revisión post-operación para evaluar el impacto del cambio al entorno nuevo. Los resultados serán enviados a las autoridades adecuadas.
19. Establecer la accesibilidad a los datos utilizados o asociados con el entorno antiguo de acuerdo con los requisitos del contrato en cuanto a protección de datos y auditoría.

A6: Retirada.

Esta actividad se realiza cuando un producto software ha alcanzado el final de su vida útil. Para ello, se deben llevar a cabo las siguientes tareas:

20. Elaborar y documentar un "plan de retirada", incluyendo los siguientes apartados:
 - Cese total o parcial del soporte después de un cierto tiempo.
 - Archivo del producto software y su documentación asociada.
 - Responsabilidad sobre cuestiones de soporte residual futuro.
 - Transición al nuevo producto, si lo hubiera.
 - Verificación de la migración.
 - Accesibilidad a las copias de los datos archivados.

21. Notificar a los usuarios las intenciones de retirada previstas, indicando lo siguiente:
 - a) descripción de la sustitución o actualización previstas, indicando la fecha de disponibilidad;
 - b) explicación de porqué no se puede seguir soportando el producto retirado; y
 - c) descripción de otras opciones de soporte disponibles una vez el producto se haya retirado.

22. Realizar operaciones para una transición suave a la nueva situación, incluyendo la formación necesaria a los usuarios.
23. Notificar a todos los afectados de que se ha completado la retirada. Toda la documentación, elementos del sistema y datos deberá ser recopiladas y archivados.
24. Establecer la accesibilidad a los datos utilizados o asociados con el producto retirado de acuerdo con los requisitos del contrato en cuanto a protección de datos y auditoría.

Las actividades de implementación del proceso y de retirada se deben realizar obligatoriamente una sola vez por parte del mantenedor al inicio y final del servicio o proceso de mantenimiento. La actividad de migración podrá ocurrir varias veces o ninguna en momentos aleatorios, según se produzcan cambios en el entorno o no (sistema operativo, SGBD, red, hardware, etc.). Las tres actividades intermedias ocurren una vez por cada petición de modificación o informe de problemas recibido por el mantenedor. Estas tres actividades forman un ciclo en el sentido de que el resultado de una revisión y aceptación del mantenimiento puede conducir a nuevas peticiones de modificación o informes de problemas.

3. CMMI

En el Sistema de Procesos Software se nombran algunos sistemas mas conocidos, como el CMM, Bootstrap e ISO 15504 donde se presenta con niveles de procesos en la Ingeniería del Software. Se habla CMMI, que es la continuación o actualización del CMM. En este capítulo se describe en forma general sobre este proceso de Software.

El CMMI (Capability Maturity Model Integration) esta orientado para mejorar la dirección de la organización de cualquier estructura. Desde 1991, CMM fue desarrollado para una cantidad de disciplinas. Algunos modelos que incluye para Ingeniería de Sistema, Ingeniería de Software, Requerimiento de Software, Gestión y Desarrollo, e integración de producto y desarrollo de procesos.

El CMMI fue desarrollado para resolver el problema de uso múltiple de CMM. Combina los tres modelos fuentes: (1) CMM para Software (SW-CMM) V2.0, (2) Electronic Industries Alliance Interim Standard (EIA/IS) 731, y (3) Integrated Product Development Capability Maturity Model (IPD-CMM) V0.98 en un framework para la mejora de los procesos, esto incluye el soporte a futuras integraciones de otras disciplinas especificas de los modelos CMMI. El trabajo tiene como objetivo que todos sus desarrollos sean consistente y compatibles con International Organization for Standardization/International Electrotechnical Comisión (ISO/IEC) 15504.

Es un proceso de punto base para una organización con mejora continua. El objetivo de Integración CMM o CMMI es proporcionar la dirección para mejorar los procesos de su organización y en su capacidad en la gestión, desarrollo, adquisición, y mantenimiento de productos o servicios. CMMI coloca accesos disponibles en una estructura que ayuda a su organización valorar su madurez de organización o procesa la capacidad del área, estableciendo prioridades para la mejora, y colocar en práctica estas mejoras.

Existen conceptos generales sobre este modelo los cuales describiremos mas adelante.

CMMI Product Suite

El "CMMI Product Suite" es el conjunto completo de los productos desarrollado alrededor del concepto CMMI. Estos productos incluye el framework en si, modelos, métodos de evaluación, materiales de evaluación, y varios tipos de entrenamiento que son producidos desde el framework CMMI.

CMMI Framework

El "CMMI Framework" es la estructura básica que organiza los componentes de CMMI, incluyendo elementos comunes de los modelos actuales de CMMI así como reglas y métodos para generar modelos, sus evaluaciones, métodos, y su material de entrenamiento. El framework permite agregar nuevas disciplinas en el CMMI de modo que las nuevas disciplinas serán integradas con lo existente.

Modelo CMMI

Desde el CMMI Framework puede generar diferentes modelos basados en las necesidades de la organización, hay múltiples modelos de CMMI. El Modelo CMMI puede ser muchas colecciones de información. La frase "Modelos CMMI" se refiere a uno, más de uno, o a la colección completa de posibles modelos que pueden ser generados desde el CMMI framework.

El CMMI "Product Suite" contiene y produce desde un framework que provee la capacidad de generar múltiples modelos y entrenamientos asociados y material de evaluación. Estos modelos deben reflejar contenido el área del conocimiento (Por ejemplo: Ingeniería de Sistemas, Ingeniería del Software, Productos Integrados y Procesos de Desarrollo) en combinación con la utilidad (por ejemplo: CMMI-SE/SW, CMMI-SE/SW/IPPD/SS).

Áreas de Procesos

Un área de procesos es un grupo de prácticas relacionadas en un área que, cuando se realiza colectivamente, satisface un conjunto de objetivos considerados importantes para realizar mejoras significativas en esa área. Todas las áreas de proceso de CMMI son comunes para ambas representaciones continuas y escenarios. En la representación continua, las áreas de procesos son organizadas por categoría de áreas de procesos.

Stakeholder

Un Stakeholder es un grupo o individuo que es afectado, o en algunas vías es responsable del resultado de un compromiso. Los Stakeholders debe ser incluidos en miembros de proyectos, generadores, clientes, usuarios finales y otros.

3.1 Selección del Modelo

Existen varios modelos disponibles en CMMI, generado en el framework CMMI. Debe el personal estar preparado para decidir cual de los modelos CMMI es el mejor para la organización según sus necesidades. Se selecciona una representación, y se debe determinar el área de conocimiento que se debe incluir en el modelo que la organización va a usar dependiendo de las representaciones; las representaciones son: escenario y continua.

3.1.1 Representación continua

- Se estudia el orden de los objetivos de negocios de la organización.
- Realiza comparaciones estudios de áreas básicas y comparaciones de escenarios.
- Realiza la conversión de "Electronic Industries Alliance Interim Standard (EIA/IS)" 731 a CMMI.
- Realiza comparaciones de procesos con "International Organization for Standardization and International Electrotechnical Comisión (ISO/IEC)" 15504.

3.1.2. Representación de escenarios

- Provee una secuencia de mejoras, comenzando con práctica de gestión básica y progresos a través de camino predefinidos en sucesivos niveles, cada nivel sirve al próximo.
- Permite comparaciones a través y entre la organización por el uso de niveles de madurez.
- Realiza la conversión de SW-CMM a CMMI
- Suministra resumen de comparaciones de resultados.

3.1.3. Modelo Integrar

Las áreas de conocimientos disponibles en un modelo CMMI son:

- Ingeniería de Sistemas
- Ingeniería del Software
- Producto Integrado y Proceso de desarrollo
- Generador de fuentes

3.1.3.1. Ingeniería de Sistema

Ingeniería de Sistema abarca el desarrollo total del Sistema, el cual puede o no incluir el Software. Ingeniería de Sistema se enfoca en la transformación de las necesidades de los clientes, expectativa, y solución del producto y soportar ese producto en su ciclo de vida. Esto involucra Gestión de Procesos, Gestión de proyectos, soporte y área de Ingeniería de procesos.

3.1.3.2. Ingeniería del Software

Ingeniería del Software abarca el desarrollo de sistema Software. Ingeniería del Software se basa en la aplicación sistemática, disciplinada y medibles enfocado para en el desarrollo, operación y mantenimiento del software. Este modelo contiene la Gestión de procesos, Gestión de Proyectos, soporte y área de Ingeniería de procesos.

3.1.3.3. Producto Integrado y procesos de desarrollo

Producto Integrado y Procesos de desarrollo o en su siglas inglesa "Integrated Product and Process Development (IPPD)" es un enfoque sistemático que logra oportunamente colaboración de Stakeholders en todo el ciclo de vida del producto para mejor satisfacción de las necesidades del cliente, expectativa y requerimientos. El proceso para soportar un IPPD enfoca estar integrado con otros procesos de la organización. El área de procesos IPPD, los objetivos específicos, prácticas específicas solas no puede alcanzar IPPD. Si un proyecto selecciona IPPD, realiza prácticas específicas IPPD concurrentemente con otras prácticas específicas usado para producir el producto. Este modelo contiene Gestión de Procesos, Gestión de proyectos, soporte y área de Ingeniería de procesos que aplicado a ambas IPPD y a otras disciplinas.

3.1.3.4. Generador de fuentes

Los generadores se deben usar para realizar funciones y agregar modificaciones al producto que es especialmente necesitado para el proyecto. Estas actividades son críticas, el beneficio del proyecto desde realizar análisis de fuentes y desde agregar actividades de monitoreo antes que el producto sea liberado. La disciplina de generadores de fuentes cubre productos adquiridos desde generadores bajo esas circunstancias.

Cuando se selecciona la disciplina de generador de fuentes para el modelo, el modelo contiene Gestión de Procesos, Gestión de Proyecto, soporte y área de Ingeniería de procesos que aplica para ambos, generadores de fuentes y las otras disciplinas que se selecciono en el modelo. El área de procesos de Gestión Generador Integrado esta incluido bajo la categoría de área de procesos Gestión de Proyectos y especificaciones de amplificadores de disciplina para generadores de fuentes están presente en otras áreas de procesos para ayudar interpretar especificaciones prácticas para el generador de fuentes.

Amplificadores de Disciplinas

Provee guía para interpretar modelo de información para disciplinas específicas (por ejemplo: IPPD, Ingeniería de Sistema, o Ingeniería de Software) son llamados "Amplificadores de Disciplinas". Los amplificadores de disciplinas son adicionados para otros componentes modelos donde son necesitados.

3.2. Nivel de capacidad

Todos los modelos CMMI tienen una representación continua que refleja el nivel de capacidad en su diseño y contenido. Un nivel de capacidad consiste de especificaciones relacionadas y prácticas genéricas para un área del proceso que puede mejorar los procesos de la organización asociándolo con esas áreas de procesos. Satisfecho los objetivos genéricos y específicos para un área de proceso en un nivel de capacidad en particular, y alcanza ese nivel de capacidad, cosecha el beneficio del proceso. Nivel de capacidad enfoca sobre la habilidad de crecimiento de la organización para realizar, controlar y mejorar este rendimiento en un área de proceso. Los niveles de capacidad son construidos sobre otro nivel, dando recomendaciones ordenadas para aprovecharlo en mejoras del proceso.

Existe 6 niveles de capacidad, diseñado desde el número 0 hasta el 5:

0. Incompleto
1. Ejecutado
2. Gestionado
3. Definido
4. Cuantitativo Gestionado
5. Optimizado

El nivel de capacidad y componentes modelos genérico enfoca en la construcción de habilidades de la organización para lograr mejoras de procesos en múltiples áreas de procesos. Usando los niveles de capacidad, objetivos genéricos, y prácticas genéricas, los usuarios son capaces para mejorar sus procesos, también demostrar y evaluar los progresos de la organización como sus mejoras.

Niveles de capacidad en la representación continua provee un orden de recomendaciones para aprovechar mejoras de procesos con cada área de procesos.

3.2.1. Nivel de capacidad 0: Incompleto

Un proceso incompleto es un proceso que no fue realizado o parcialmente ejecutado. Uno o más objetivos específicos en el área de procesos no fueron satisfechos.

3.2.2. Nivel de capacidad 1: Ejecutado

Un proceso de nivel de capacidad 1 esta caracterizado como un "Proceso ejecutado".

Un proceso ejecutado es un proceso que satisface los objetivos específicos de un área de procesos. Apoya y permite el trabajo necesitado para producir productos identificados del trabajo de la salida usando productos identificados del trabajo de la entrada. Una distinción crítica entre un proceso incompleto y un proceso ejecutado es que un proceso ejecutado satisface todo los objetivos específicos de un área de procesos.

3.2.2.1. Objetivo General

El proceso soporta y permite logros de objetivos específicos de un área de procesos por la transformación identificables de productos de trabajo de entradas para producir producto de trabajos identificables de salidas.

3.2.2.2. Prácticas Genéricas

Ejecución de prácticas básicas del área de procesos para desarrollar productos de trabajo y proveer servicios para alcanzar los objetivos específicos de un área de procesos.

3.2.3. Nivel de capacidad 2: Gestionado

Esta caracterizado por gestión de procesos.

3.2.3.1. Objetivo General

Institucionaliza una Gestión de procesos. El proceso es institucionalizado como una gestión de procesos.

3.2.3.2. Prácticas Genéricas

- 1.) Establecer y mantener una política organizacional.
- 2.) Planificar el Proceso.
- 3.) Proveer recursos adecuado para la ejecución del proceso, desarrollo del producto de trabajo y proveer los servicios de los procesos.
- 4.) Asignar responsabilidad y autorización para ejecutar el proceso, desarrollo del producto de trabajo y proveer los servicios de los procesos.
- 5.) Entrenar personal para ejecutar o soportar el proceso requerido.
- 6.) Gestionar la configuración. Colocar los productos de trabajo del proceso bajo niveles apropiados de la configuración gerencial.
- 7.) Identificar y comprometer a relevantes "stakeholder" tal como lo planificado.
- 8.) Monitorear y controlar el proceso. El monitoreo y controla del proceso contra el plan para ejecutar el proceso y tomar acciones correctivas apropiadas.
- 9.) Evaluar el objetivo de la adherencia del proceso contra la descripción, estándares y procedimientos, y no acatamiento de la dirección del proceso.
- 10.) Revisar las actividades, estatus, y resultados del proceso con la Gerencia del nivel superior y editar los resultados.

3.2.4. Nivel de capacidad 3: Definido

Un proceso del nivel de capacidad 3 se caracteriza por ser un "proceso definido". Un proceso definido es manejado (proceso del nivel de la capacidad 2) que se adapta al sistema de la organización de procesos estándares según las pautas de adaptación de la organización, y contribuye los productos del trabajo, las medidas, y la otra información del proceso de mejorar a los activos de proceso de la organización.

3.2.4.1. Objetivo General

Institucionalizar un proceso definido.

3.2.4.2. Prácticas Genéricas

- 1.) Establecer un proceso definido.
- 2.) Recolectar la información de la mejora. Recogen productos del trabajo, medidas, resultados de las medidas, y la información de la mejora derivada del planeamiento y de realizar el proceso para apoyar el uso y las mejoras futuras de los procesos y de los activos de proceso de la organización.

3.2.5. Nivel de capacidad 4: Cuantitativo Gestionado

Un proceso del nivel de capacidad 4 se caracteriza por ser un "proceso cuantitativo gestionado". Un proceso cuantitativo gestionado es definido (proceso del nivel de capacidad 3) que es usar control estadístico y otras técnicas cuantitativas. Los objetivos cuantitativos para el funcionamiento de la calidad y del proceso se establecen y se utilizan como criterios en el manejo del proceso. El funcionamiento de la calidad y del proceso se entiende en términos estadísticos y se maneja a través de la vida del proceso.

3.2.5.1. Objetivo General

Institucionaliza un proceso cuantitativo gestionado.

3.2.5.2. Prácticas Genéricas

- 1.) Establecer Objetivos cuantitativos para el proceso. Establece y mantiene los objetivos cuantitativos para el proceso que tratan con la calidad y el funcionamiento del proceso basados en las necesidades del cliente y en los objetivos del negocio.
- 2.) Establecer el funcionamiento de los subprocessos para determinar la capacidad del proceso y para alcanzar los objetivos cuantitativos establecidos sobre la calidad del proceso-funcionamiento.

3.2.6. Nivel de capacidad 5: Optimizado

Se caracteriza un proceso del nivel de capacidad 5 por ser un "proceso óptimo". Un proceso óptimo es manejado en forma cuantitativo (proceso del nivel de capacidad 4) que se cambia y se adapta para resolver objetivos relevantes de negocios actuales y futuros. El enfoque de proceso óptimo es continuamente mejorar el funcionamiento de proceso con mejoras tecnológicas incrementales e innovadoras. Dichas mejoras se identifican en tratar la raíz de las causas con la variación de proceso y cuantificarlas de los procesos de la organización, evaluado, y desplegado como apropiado. Se seleccionan estas mejoras basándose en una comprensión cuantitativa de su contribución prevista a alcanzar los objetivos del proceso-mejora de la organización contra el coste y el impacto a la organización.

3.2.6.1. Objetivo General

Institucionaliza un proceso optimizado.

3.2.6.2. Prácticas Genéricas

- 1.) Asegurar la mejora continua del proceso para satisfacer los objetivos de negocios relevantes de la organización.
- 2.) Corregir las causas raíz de los problemas. Identifique y corrija las causas raíz de los defectos y otros problemas en el proceso.

3.3. Nivel de madurez

Es la representación de escenarios aplicados a la madurez total de una organización. Existen cinco niveles de madurez, cada nivel de madurez comprende un conjunto predefinido de área de procesos.

Nivel	Representación de escenario
1.	Inicial
2.	Gestionado
3.	Definido
4.	Cuantitativo Gestionado
5.	Optimizado

La representación continua tiene mas de prácticas específicas que la representación de escenario, motivado a que la representación continua tiene dos tipos de prácticas específicas: base y avanzado; y la representación de escenario tiene solo un tipo de práctica específica. En las representaciones continuas, existen prácticas genéricas para los niveles de capacidad 1-5, y en la representación de escenarios, solo se presenta la práctica genérica en los niveles de capacidad 2 y 3; no hay práctica genérica en los niveles de capacidad 1, 4 y 5.

3.4. Áreas de procesos

Existen cuatro áreas de procesos, las cuales se nombran a continuación, para luego describirlos en forma general mas adelante:

- Gestión de procesos
- Gestión de Proyectos
- Ingeniería
- Soporte

Cada área se ejecuta en forma secuencial dos objetivos: los objetivos específicos y luego los objetivos globales o generales. Los objetivos generales tienen las mismas actividades en todas las áreas del proceso de la ingeniería de CMMI, las diferencias son los

procedimientos internos en cada actividad dependiendo de cada área del proceso que se describe con detalle en la documentación de CMMI (* direccionar a documentación CMMI). Los objetivos generales posee las siguientes actividades:

Tabla de actividades de Objetivos Globales

- GG 1 Lograr Objetivos específicos
 - GP 1.1 Ejecución de prácticas básicas
- GG 2 Institucionalizar una Gestión de procesos
 - GP 2.1 Establecer una política organizacional
 - GP 2.2 Planificar el proceso
 - GP 2.3 Proveer Recursos
 - GP 2.4 Asignar responsabilidades
 - GP 2.5 Entrenar Personal
 - GP 2.6 Gestionar configuraciones
 - GP 2.7 Identificar y comprometer relevantes Stakeholders
 - GP 2.8 Monitorear y Controlar el proceso
 - GP 2.9 Evaluar objetivo de la adherencia
 - GP 2.10 Revisión de estatus con los gerentes de alto nivel
- GG 3 Institucionalizar un proceso definido
 - GP 3.1 Establecer un proceso definido
 - GP 3.2 Recolectar Información de mejoras
- GG 4 Institucionalizar un proceso cuantitativo gestionado
 - GP 4.1 Establecer objetivos cuantitativos para el proceso
 - GP 4.2 Establecer el funcionamiento de los subprocesos
- GG 5 Institucionalizar un proceso optimizado
 - GP 5.1 Asegurar la mejora de proceso continuo
 - GP 5.2 Corregir de las causas raíz de los problemas

En los siguientes sub-capítulos al culminar los objetivos específicos de cada área de proceso, se debe ejecutar los objetivos globales identificados en la “Tabla de actividades de Objetivos Globales”. Se debe reseñar que las actividades “Tabla de actividades de Objetivos Globales” (GG) son diferentes en sus contenidos para cada área del proceso.

3.4.1. Gestión de procesos

Esta sección involucra todas las áreas de proceso que pertenecen a la categoría de la gerencia o Gestión de proceso. Estas áreas son:

- Enfoque de proceso organizacional
- Definición de proceso organizacional
- Entrenamiento organizacional
- Funcionamiento de proceso organizacional
- Innovación y despliegue Organizacional

3.4.1.1. Enfoque de proceso organizacional

El propósito del enfoque de proceso organizacional es planificar e implementar la mejora en ejecución basada en una comprensión cuidadosa de las fuerzas, de las debilidades actuales y de los activos de proceso de la organización.

Tabla de la relación Objetivo a Práctica

- SG 1 Determinar procesos – Oportunidades de mejoras
 - SP 1.1-1 Establecer necesidades de procesos organizacional
 - SP 1.2-1 Evaluar los procesos organizacionales
 - SP 1.3-1 Identificar las mejoras de los procesos organizacionales
- SG 2 Planificar e Implementar Actividades de Mejoras de procesos
 - SP 2.1-1 Establecer procesos de Planes de acciones
 - SP 2.2-1 Implementar Procesos planes de acciones
 - SP 2.3-1 Despliegue de los Activos de proceso organizacional
 - SP 2.4-1 Incorporar las experiencias de Procesos relacionadas a los Activos de proceso organizacional.

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.1.2. Definición de proceso organizacional

El propósito de la definición de proceso de organización es establecer y mantener un sistema usable de activos de proceso organizacional.

La biblioteca de activo de proceso organizacional es una colección de artículos mantenidos por la organización para el uso de las personas y los proyectos de la organización. Esta colección de artículos incluye descripciones de procesos y de elementos del proceso, descripciones de los modelos del ciclo de vida, el proceso que adapta pautas, la documentación proceso-relacionada, y datos. La biblioteca de activo de proceso organizacional de la organización soporta el aprendizaje y el proceso de mejora, permitiendo compartir las mejores prácticas y lecciones aprendidas a través de la organización.

Tabla de la relación Objetivo a Práctica

- SG 1 Establecer activos de procesos organizacional
 - SP 1.1-1 Establecer proceso estándar
 - SP 1.2-1 Establecer descripciones de Modelos de ciclos de vida
 - SP 1.3-1 Establecer Criterios y lineamientos adecuados
 - SP 1.4-1 Establecer depósito de medidas organizacional
 - SP 1.5-1 Establecer biblioteca de Activos de procesos organizacional

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.1.3. Entrenamiento organizacional

El propósito del entrenamiento organizacional es desarrollar las habilidades y el conocimiento del personal para así puedan realizar sus actividades con eficacia y eficiencia.

Tabla de la relación Objetivo a Práctica

SG 1 Establecer capacidad de entrenamiento organizacional

SP 1.1-1 Establecer Necesidades de entrenamiento táctico

SP 1.2-1 Determinar que necesidades de entrenamiento son responsabilidad de la organización

SP 1.3-1 Establecer un plan de entrenamiento táctico organizacional

SP 1.4-1 Establecer capacidad de entrenamiento

SP 1.5-1 Establecer biblioteca de Activos de procesos organizacional

SG 2 Proveer entrenamientos necesarios

SP 2.1-1 Entregar entrenamiento

SP 2.2-1 Establecer registros de entrenamientos

SP 2.3-1 Determinar la eficacia del entrenamiento

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.1.4. Funcionamiento de proceso organizacional

El propósito del funcionamiento de proceso organizacional es establecer y mantener una comprensión cuantitativa del funcionamiento del sistema de la organización de los procesos estándares en la ayuda de los objetivos de calidad y del proceso-funcionamiento y proporcionar los datos, las líneas fundamentales y los modelos de proceso del funcionamiento para manejar cuantitativamente los proyectos de la organización.

El funcionamiento de proceso es una medida de los resultados reales alcanzados siguiendo un proceso. El funcionamiento de proceso es caracterizado por las medidas del proceso (por ejemplo, esfuerzo, duración de ciclo, y eficacia del retiro del defecto) y las medidas del producto (por ejemplo, confiabilidad y densidad del defecto).

Tabla de la relación Objetivo a Práctica

SG 1 Establecer representación de líneas fundamentales y modelos

SP 1.1-1 Seleccionar procesos

SP 1.2-1 Establecer Procesos de representación de medidas

SP 1.3-1 Establecer calidad y procesos de representación de objetivos

SP 1.4-1 Establecer procesos de realización de líneas fundamentales

SP 1.5-1 Establecer procesos de realización de Modelos

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.1.5. Innovación y despliegue Organizacional

El propósito de la innovación y del despliegue de organización es seleccionar y desplegar las mejoras incrementales e innovadoras que son las medidas de las mejoras de los procesos y de las tecnologías de la organización. Las mejoras apoyan los objetivos del funcionamiento de la calidad y del proceso de la organización según lo derivado de los objetivos de negocio de la organización.

Tabla de la relación Objetivo a Práctica

SG 1 Seleccionar las mejoras

SP 1.1-1 Recolección y análisis de propuestas de mejoras

SP 1.2-1 Identificación y análisis de Innovaciones

SP 1.3-1 Piloto de Mejoras

SP 1.4-1 Seleccionar mejoras para el despliegue

SG 2 Despliegue de mejoras

SP 2.1-1 Plan de despliegue

SP 2.2-1 Gestión de despliegue

SP 2.3-1 Medición de efectos de despliegue

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2. Gestión de proyectos

La sección siguiente contiene todas las áreas de proceso que pertenecen a la categoría del área del proceso de la gerencia de proyecto o gestión de proyecto. Las áreas del proceso de la gerencia de proyecto de CMMI son las siguientes:

- Planificación de proyecto
- Control y monitoreo de proyecto
- Gestión de contrato de suministro
- Gestión de Integración de proyecto por IPPD
- Gestión de riesgo
- Teaming Integrado
- Gestión Integrada de suministro
- Gestión de proyecto cuantitativo

3.4.2.1. Planificación de proyecto

El propósito de la planificación del proyecto es establecer y mantener los planes que definen actividades del proyecto.

Tabla de la relación Objetivo a Práctica

SG 1 Establecer estimaciones

SP 1.1-1 Estimar el alcance del proyecto

- SP 1.2-1 Establecer Estimaciones del producto de trabajo y atributos de las tareas
- SP 1.3-1 Definir Ciclo de vida del Proyecto
- SP 1.4-1 Determinar estimaciones de esfuerzos y costos

SG 2 Desarrollo de un plan del proyecto

- SP 2.1-1 Establecer el presupuesto y fechas
- SP 2.2-1 Identificar riesgos del proyecto
- SP 2.3-1 Plan para gestión de datos
- SP 2.4-1 Plan para recursos del proyecto
- SP 2.5-1 Plan para necesidades de conocimientos y habilidades
- SP 2.6-1 Plan para implicar a Stakeholder
- SP 2.7-1 Establecer el plan del proyecto

SG 3 Obtener compromiso para el plan

- SP 3.1-1 Revisión de planes que afectan el proyecto
- SP 3.2-1 Reconciliación del proyecto y niveles de recursos
- SP 3.3-1 Obtener el plan compromiso

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.2. Control y monitoreo de proyecto

El propósito de la supervisión y del control del proyecto es proporcionar una comprensión del progreso del proyecto para poder tomar acciones correctivas apropiadas cuando el funcionamiento del proyecto se desvíe perceptiblemente del plan.

Tabla de la relación Objetivo a Práctica

SG 1 Monitoreo del proyecto contra el plan

- SP 1.1-1 Monitoreo parámetros de la planificación del proyecto
- SP 1.2-1 Monitoreo de compromisos
- SP 1.3-1 Monitoreo de riesgos del proyecto
- SP 1.4-1 Monitoreo de gestión de datos
- SP 1.5-1 Monitoreo de la implicación del Stakeholder
- SP 1.6-1 Revisión de la conducta del progreso
- SP 1.7-1 Revisión de la conducta de “Milestone”

SG 2 Gestión de acción correctiva para la clausura

- SP 2.1-1 Análisis de la edición
- SP 2.2-1 Tomar acción correctiva
- SP 2.3-1 Gestión de acción correctiva

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.3. Gestión de contrato de suministro

El propósito de la gestión de contrato de suministro es manejar la adquisición de productos de los surtidores para quienes existe un acuerdo formal.

Tabla de la relación Objetivo a Práctica

- SG 1 Establecer surtidores formales
 - SP 1.1-1 Determinar el tipo de la adquisición
 - SP 1.2-1 Seleccionar a surtidores
 - SP 1.3-1 Establecer surtidores formales
- SG 2 Satisfacción de surtidores formales
 - SP 2.1-1 Revisión COTS (“*Commercial Off-The-Shelf*”) del producto
 - SP 2.2-1 Ejecutar el surtidor formal
 - SP 2.3-1 Aceptación del producto adquirido

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.4. Gestión de Integración del proyecto por IPPD

El propósito de la Gestión de Integración del proyecto es establecer y manejar el proyecto y la implicación de los Stakeholders relevantes según un proceso integrado y definido que se adapte al sistema de la organización de procesos estándares. Para el desarrollo integrado del producto y del proceso (IPPD), la gerencia de proyecto integrada también cubre el establecimiento de una visión compartida para el proyecto y de una estructura de equipo para los equipos integrados que realizarán los objetivos del proyecto.

Tabla de la relación Objetivo a Práctica

- SG 1 Uso del proceso definido del proyecto
 - SP 1.1-1 Establecer el proceso definido del proyecto
 - SP 1.2-1 Uso de Activos de proceso organizacional
 - SP 1.3-1 Integrar planes
 - SP 1.4-1 Gestionar el proyecto usando el plan integrado
 - SP 1.5-1 Contribuir para Activos de proceso organizacional
- SG 2 Coordinar y colaborar con los relevantes Stakeholders
 - SP 2.1-1 Gestión de involucrar Stakeholders
 - SP 2.2-1 Gestión de dependencias
 - SP 2.3-1 Resolver la distribución de la coordinación
- SG 3 Uso de Visión compartida del proyecto por IPPD
 - SP 3.1-1 Definir contexto de Visión compartida del proyecto
 - SP 3.2-1 Establecer Visión compartida de proyecto
- SG 4 Organizar equipos integrados por IPPD
 - SP 4.1-1 Determinar estructura de equipos integrados para el proyecto
 - SP 4.2-1 Desarrollo de un preliminar distribución de requerimientos para los equipos integrados
 - SP 4.3-1 Establecer equipos integrados

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.5 Gestión de riesgo

El propósito de la gerencia de riesgo es identificar problemas potenciales antes de que ocurran, para poder ser planeado e invocar el manejo-riesgo de actividades según lo necesitado a través de la vida del producto o del proyecto y atenuar impactos adversos en la realización de objetivos.

Tabla de la relación Objetivo a Práctica

- SG 1 Preparación para la gestión de riesgo
 - SP 1.1-1 Determinar fuentes de riesgos y categorías
 - SP 1.2-1 Definir parámetros de riesgos
 - SP 1.3-1 Establecer una estrategia de gestión de riesgo
- SG 2 Identificar y Analizar Riesgo
 - SP 2.1-1 Identificar riesgos
 - SP 2.2-1 Evaluar, categorizar y priorizar riesgos
- SG 3 Mitigar Riesgo
 - SP 3.1-1 Desarrollo de Planes de Mitigación de riesgos
 - SP 3.2-1 Planes de implementación de Mitigación de riesgos

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.2.6 Teaming Integrado

El propósito de Teaming integrado es formar y sostener a un equipo integrado para el desarrollo de los productos del trabajo.

Componen a un equipo integrado (también conocido como un "equipo integrado del producto" o IPT) de los Stakeholders relevantes que generan y ponen las decisiones en ejecución el producto del trabajo que es desarrollado. Los miembros del equipo integrado son colectivamente responsables de entregar el producto del trabajo. El equipo integrado recibe su asignación de su patrocinador. El patrocinador de un equipo integrado es una persona o un grupo (por ejemplo, encargado de proyecto u otro equipo integrado) que pueden asignar tareas de trabajo y proporcionar recursos.

Tabla de la relación Objetivo a Práctica

- SG 1 Establecer composición del equipo
 - SP 1.1-1 Identificar las tareas de los equipos
 - SP 1.2-1 Identificar el conocimiento y habilidades necesarios
 - SP 1.3-1 Asignar miembros del equipo apropiadamente
- SG 2 Gobierna la operación del equipo
 - SP 2.1-1 Establecer una visión compartida
 - SP 2.2-1 Establecer una carta de equipos
 - SP 2.3-1 Definir reglas y responsabilidades
 - SP 2.4-1 Establecer Procedimientos operacionales
 - SP 2.4-1 Colaboración de interfases entre equipos

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.7 Gestión Integrada de suministro

El propósito de la gestión integrada de suministro es pro-activamente identificar fuentes de los productos que se pueden utilizar para satisfacer los requisitos del proyecto y para manejar a surtidores seleccionados mientras que mantienen una relación cooperativa del proyecto-surtidor.

Tabla de la relación Objetivo a Práctica

- SG 1 Analizar y seleccionar fuentes de productos
 - SP 1.1-1 Analizar potenciales fuentes de productos
 - SP 1.2-1 Evaluar y determinar fuentes de productos
- SG 2 Coordinar Trabajo con Surtidores
 - SP 2.1-1 Monitorear Proceso de selección de Surtidores
 - SP 2.2-1 Evaluar productos de trabajo del surtidor seleccionado
 - SP 2.3-1 Revisar el acuerdo o relaciones con el surtidor

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.2.8 Gestión de proyecto cuantitativo

El propósito del área Gestión de proyecto cuantitativa es manejar el proceso cuantitativamente definido en el proyecto para alcanzar los objetivos establecidos en calidad y del proceso-funcionamiento del proyecto.

Tabla de la relación Objetivo a Práctica

- SG 1 Gestión de proyecto cuantitativo
 - SP 1.1-1 Establecer los objetivos del proyecto
 - SP 1.2-1 Construir procesos definidos
 - SP 1.3-1 Seleccionar los subprocesos que serán manejados estadísticamente
 - SP 1.4-1 Gestión de Ejecución del proyecto
- SG 2 Gestionar estadísticamente la ejecución de los subprocesos
 - SP 2.1-1 Seleccionar Medidas y técnicas analíticas
 - SP 2.2-1 Aplicar métodos estadísticos para entender la variación
 - SP 2.3-1 Monitorear la ejecución de los subprocesos seleccionados
 - SP 2.4-1 Registrar los datos de la gestión estadísticas

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.3 Ingeniería

La siguiente sección contiene todas las áreas de proceso que pertenecen a la categoría del área del proceso de la ingeniería. Las áreas del proceso de la ingeniería de CMMI son las siguientes:

- Gestión de requerimientos
- Gestión de desarrollo
- Solución técnica
- Integración del Producto
- Verificación
- Validación

3.4.3.1. Gestión de requerimientos

El propósito de la gestión de los requerimientos es manejar los requisitos de los productos del proyecto y de los componentes del producto e identificar inconsistencias entre esos requisitos y los productos de los planes y del trabajo del proyecto. Los procesos introductorios de la gestión de los requerimientos manejan todos los requisitos recibidos o generados por el proyecto, incluyendo requisitos técnicos y no técnicos así como esos requisitos impuestos en el proyecto por la organización. En detalle, si se pone en ejecución el área del proceso del desarrollo de los requisitos, sus procesos generarán los requisitos del producto y del producto-componente que también serán manejados por los procesos de la gestión de los requisitos. Cuando se coloca en ejecución a la gestión de requerimientos, el desarrollo de los requisitos, y las áreas técnicas todas del proceso de la solución, sus procesos asociados se pueden atar y realizar de cerca concurrentemente.

Tabla de la relación Objetivo a Práctica

SG 1 Gestión de requerimientos

SP 1.1-1 Obtener un entendimiento del requerimiento

SP 1.2-2 Obtener compromiso para el requerimiento

SP 1.3-1 Gestión de cambios de requerimientos

SP 1.4-2 Mantener traza Bidireccional de los requerimientos

SP 1.5-1 Identificar inconsistencia entre el proyecto y los requerimientos

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.3.2. Desarrollo de los requerimientos

El propósito del desarrollo de los requisitos es analizar y producir al cliente, los requisitos del producto y del producto-componente. Esta área de proceso describe tres tipos de requisitos: requisitos del cliente, requisitos del producto y requisitos del producto-componentes. Tomado en conjunto, la dirección de estos requisitos, las necesidades de Stakeholders relevantes, incluyendo estos en las diferentes fases del ciclo de vida de producto (criterios de la prueba de aceptación, etc.) y a las cualidades de producto (seguridad, confiabilidad, capacidad de mantenimiento, etc.). Los requisitos también tratan

los efectos causados por la selección del diseño de la solución (integración de productos disponibles comerciales).

Tabla de relación de los Objetivos a Práctica

SG 1 Desarrollo de requerimientos del cliente

SP 1.1-1 Recolectar necesidades del Stakeholder

SP 1.1-2 Extraer necesidades

SP 1.2-1 Desarrollo de requerimientos del cliente

SG 2 Desarrollo Requerimientos del producto

SP 2.1-1 Establecer producto y componentes-productos requeridos

SP 2.2-1 Localizar Producto-componente requeridos

SP 2.3-1 Identificar requerimientos de Interfaces

SG 3 Analizar y validar el requerimientos

SP 3.1-1 Establecer Conceptos Operacionales y Escenarios

SP 3.2-1 Establecer una definición de funcionalidad requerido

SP 3.3-1 Analizar requerimientos

SP 3.4-3 Analizar requerimientos para alcanzar Balance

SP 3.5-1 Validar Requerimiento

SP 3.5-2 Validar Requerimientos con Métodos Extenso

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.3.3. Solución Técnica

El propósito de la solución técnica es diseñar, desarrollar e implementar soluciones de los requisitos. Las soluciones, los diseños y las implementaciones abarcan los productos, componentes del producto y el ciclo de vida relacionado con el producto procesa solo o en combinaciones como apropiadas. El área técnica del proceso de la solución es aplicable en cualquier nivel de la arquitectura del producto y a cada producto, componente del producto, producto relacionado al ciclo de vida y servicio.

Tabla de relación de los Objetivos a Práctica

SG 1 Seleccionar Soluciones Productos-Componentes

SP 1.1-1 Desarrollo soluciones alternativas y Criterio de Selección

SP 1.1-2 Desarrollo detallado de soluciones alternativas y Criterio de Selección

SP 1.2-2 Envolver Conceptos Operacionales y Escenarios

SP 1.3-1 Selección de Soluciones Productos-Componentes

SG 2 Desarrollo del Diseño

SP 2.1-1 Diseño del producto o Producto-Componente

SP 2.2-3 Establecer una técnica de datos empaquetado

SP 2.3-1 Establecer Descripción de Interfase

SP 2.3-3 Diseño de Interfase Usando Criterios

SP 2.4-3 Ejecución de fabricación, compra o Análisis de Reuso

SG 3 Implementación de Diseño del Producto

SP 3.1-1 Implementar el Diseño

SP 3.2-1 Desarrollo Documentación de soporte del producto

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.3.4. Integración del Producto

El propósito de la integración del producto es ensamblar el producto con los componentes, se asegura de que el producto, según lo integrado, funcione correctamente y se haga la entrega del producto. Las direcciones de esta área del proceso en la integración de los componentes del producto en componentes más complejos o en productos completos. El término "integración" se utiliza en este sentido a través de esta área de proceso y no debe ser confundido con la integración de la gente o de las actividades que pueden ser descritas a otra parte en el modelo. El alcance de esta área de proceso es lograr la integración completa del producto a través del ensamblaje progresivo de los componentes, en etapa o en etapas incrementales, según una secuencia definida y procedimientos de la integración.

Un aspecto crítico de la integración del producto es la gerencia de las interfaces internas y externas y de los componentes del producto para asegurar compatibilidad entre las interfaces. La atención se debe prestar para interconectar a la gerencia a través del proyecto. La integración del producto es más que apenas un montaje de una sola vez de los componentes del producto en la conclusión del diseño y de la fabricación. La integración del producto se puede conducir en forma incremental, usando un proceso iterativo de montar componentes del producto, de evaluarlos y después de montar más componentes del producto. Este proceso puede comenzar con el análisis y las simulaciones (por ejemplo: hilos de rosca, prototipos rápidos, prototipos virtuales y prototipos físicos) y progresar constantemente en una funcionalidad incremental cada vez más realista hasta que se alcanza el producto final. En cada estructura sucesiva, se construyen, se evalúan, se mejoran, y se reconstruyen los prototipos (virtual, rápido o comprobación) basado sobre el conocimiento ganado en el proceso de la evaluación. El grado virtual contra prototipo físico requerido depende de la funcionalidad de las herramientas de diseño, de la complejidad del producto y de su riesgo asociado. Hay una alta probabilidad que el producto, integrado de este modo, pasará la verificación del producto y validación. Para algunos productos, la fase pasada de la integración ocurrirá cuando el producto se despliega en su sitio operacional previsto.

Tabla de relación de los Objetivos a Práctica

SG 1 Preparación para Integración del Producto

SP 1.1-1 Determinar la secuencia de la Integración

SP 1.2-2 Establecer el ambiente de la Integración del Producto

SP 1.3-3 Establecer Procedimientos y Criterios en Integración del Producto

SG 2 Garantizar Compatibilidad de las Interfases

SP 2.1-1 Revisar Descripción de la Interfase para la Completitud

SP 2.2-1 Gestionar (o Gerencia) las Interfases

SG 3 Ensamblar Componentes del Producto y Entregar el Producto

SP 3.1-1 Confirmar Rapidez de los Componentes del Producto para la Integración

SP 3.2-1 Ensamblaje de los Componentes del Producto

SP 3.3-1 Evaluar el Ensamblado de los componentes del producto

SP 3.4-1 Empaquetar y Entregar el producto o Componente del producto

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.3.5. Verificación

El propósito de la verificación es asegurarse de que los productos seleccionados del trabajo resuelvan los requisitos especificados. El área del proceso de verificación implica lo siguiente: preparación de la verificación, ejecución de la verificación e identificación de la acción correctiva. Este proceso incluye la verificación del producto y de los productos intermedios del trabajo contra todos los requisitos seleccionados, incluyendo cliente, producto y requisitos del producto-componente. La verificación es intrínsecamente un proceso incremental porque ocurre a través del desarrollo de los productos, comenzando con la verificación de los requisitos, progresando en la verificación de los productos de desarrollo del trabajo, y culminando en la verificación del producto terminado.

Las áreas de proceso de la verificación y de la validación son similares, pero tratan diversas ediciones. La validación demuestra que el producto, en la manera prevista (o pues será proporcionado), satisfará su uso previsto, mientras que la verificación trata si el producto del trabajo refleja correctamente los requisitos especificados. Es decir, la verificación se asegura de que "usted la construyera correctamente" mientras que, la validación se asegura de que "usted construyera la cosa correctamente".

Tabla de relación de los Objetivos a Práctica

SG 1 Preparación para la Verificación

SP 1.1-1 Seleccionar los productos del trabajo para la verificación

SP 1.2-2 Establecer el ambiente de verificación

SP 1.3-3 Establecer Procedimientos y Criterios en la verificación

SG 2 Ejecutar ambas revisiones (validación y verificación)

SP 2.1-1 Preparar las dos revisiones

SP 2.2-1 Conducir las dos revisiones

SP 2.3-2 Analizar datos de las dos revisiones

SG 3 Verificar los productos de trabajo seleccionados

SP 3.1-1 Ejecutar la verificación

SP 3.2-2 Analizar resultados de la verificación e Identificar acción de corrección

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.3.6. Validación

El propósito de la validación es demostrar que un producto o un componente del producto satisfacen su uso previsto, cuando está colocado en su ambiente.

Las actividades de la validación se pueden aplicar en todos los aspectos del producto y en cualquiera de sus ambientes previstos, tales como operación, entrenamiento, fabricación, mantenimiento y servicios de ayuda. Los métodos empleados para lograr la validación se pueden aplicar a los productos del trabajo, así como al producto y los componentes del producto. Los productos del trabajo (por ejemplo: requisitos, diseños, los

prototipos) se deben seleccionar en base a cuales son los mejores predispuestos para el bien del producto y el componente del producto que satisfarán las necesidades del usuario.

El ambiente de la validación debe representar el ambiente previsto para el producto y los componentes del producto así como representan el ambiente previsto conveniente para las actividades de la validación con los productos del trabajo.

La validación demuestra que el producto, en la manera prevista, satisfará su uso previsto; mientras que, la verificación trata si el producto del trabajo refleja correctamente los requisitos especificados. El uso de las actividades de la validación se acerca a la verificación (por ejemplo: prueba, análisis, inspección, demostración o simulación). A menudo, los usuarios finales están implicados en las actividades de la validación. Las actividades de la validación y de la verificación funcionan a menudo concurrentemente y pueden utilizar porciones del mismo ambiente.

En lo posible, la validación se debe lograr usando el producto o el componente del producto que funciona en su ambiente previsto. El ambiente completo se puede utilizar o solamente parte de él. Sin embargo, las ediciones de la validación se pueden descubrir al inicio de la vida del proyecto usando productos del trabajo.

Cuando se identifican las ediciones de la validación, se refieren a los procesos asociados al desarrollo de los requisitos, a la solución técnica o a las áreas de proceso de supervisión del proyecto y de control para la resolución.

Tabla de relación de los Objetivos a Práctica

SG 1 Preparación para la Validación

SP 1.1-1 Seleccionar los productos del trabajo para la validación

SP 1.2-2 Establecer el ambiente de validación

SP 1.3-3 Establecer Procedimientos y Criterios en la validación

SG 2 Validar producto o producto-Componente

SP 2.1-1 Preparar las validaciones

SP 2.2-1 Analizar los resultados de las validaciones

GG Ejecución de "Tabla de actividades de Objetivos Globales"

3.4.4. Soporte

La sección siguiente contiene todas las áreas de proceso que pertenecen a la categoría del área del proceso de soporte. Las áreas del proceso de Soporte de CMMI son las siguientes:

- Gestión de configuración
- Proceso y garantía de la calidad del producto
- Medición y Análisis
- Análisis de Decisión y resolución
- Ambiente Organizacional para la Integración
- Análisis causal y Resolución

3.4.4.1. Gestión de configuración

El propósito de la gestión de configuración es establecer y mantener la integridad de los productos del trabajo, usando la identificación de la configuración, el control de configuración, la contabilidad del estado de la configuración y la auditoria de la configuración.

SG 1 Establecer líneas de bases

SP 1.1-1 Identificar Ítem de Configuración

SP 1.2-1 Establecer un Sistema de Gestión de Configuración

SP 1.3-1 Crear o entregar líneas de bases

SG 2 Vías y control de cambios

SP 2.1-1 Requerimientos de vías de cambios

SP 2.2-1 Control de Ítem de configuración

SG 3 Establecer Integridad

SP 3.1-1 Establecer registros de Gestión de Configuración

SP 3.2-1 Ejecución de Auditoria de configuración

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.4.2. Proceso y garantía de la calidad del producto

El propósito del proceso y garantía de la calidad del producto es proveer del personal y de la gerencia en la introducción objetiva en los procesos y productos asociados del trabajo.

SG 1 Evaluar los objetivos de los procesos y productos del trabajo

SP 1.1-1 Evaluar los objetivos de los procesos

SP 1.2-1 Evaluar los objetivos de los productos de los trabajos y servicios

SG 2 Proporcione La Penetración Objetiva

SP 2.1-1 Comunique y asegure la resolución de las ediciones no acatadas

SP 2.2-1 Establecer registros

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.4.3. Medición y Análisis

El propósito de la medición y del análisis es desarrollar y sostener una capacidad de medida que se utilice para apoyar necesidades de información de la gerencia.

SG 1 Alineación de mediciones y actividad de análisis

SP 1.1-1 Establecer Objetivos de Medición

SP 1.2-1 Especificar mediciones

SP 1.3-1 Especificar recolección de datos y procedimientos de almacenamiento

SP 1.4-1 Especificar procedimientos de análisis

- SG 2 Proveer Resultados de la medición
 - SP 2.1-1 Recolectar datos de la medición
 - SP 2.2-1 Analizar datos de la medición
 - SP 2.3-1 Almacenar datos y resultados
 - SP 2.4-1 Comunicar Resultados

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.4.4. Análisis de Decisión y resolución

El propósito del análisis de la decisión y de la resolución es analizar decisiones posibles usando un proceso formal de la evaluación que estudie alternativas identificadas contra criterios establecidos.

- SG 1 Evaluar alternativas
 - SP 1.1-1 Establecer las pautas para el análisis de la decisión
 - SP 1.2-1 Establecer Criterio de Evaluación
 - SP 1.3-1 Identificar Soluciones alternativas
 - SP 1.4-1 Seleccionar Métodos de evaluación
 - SP 1.5-1 Evaluar las alternativas
 - SP 1.6-1 Seleccionar Soluciones

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.4.5. Ambiente Organizacional para la Integración

El propósito del ambiente de organización para la integración es proporcionar una infraestructura integrada del desarrollo del producto y del proceso (IPPD) y manejar al personal para la integración.

- SG 1 Proveer Infraestructura IPPD
 - SP 1.1-1 Establecer la Visión Compartida organizacional
 - SP 1.2-1 Establecer un ambiente de trabajo integral
 - SP 1.3-1 Identificar requerimiento de habilidad IPPD-único
- SG 2 Gestión al personal para la integración
 - SP 2.1-1 Establecer mecanismo de lideres
 - SP 2.2-1 Establecer Incentivos para la integración
 - SP 2.3-1 Establecer Mecanismo para balancear los equipos y responsabilidades en la organización

GG Ejecución de “Tabla de actividades de Objetivos Globales”

3.4.4.6. Análisis causal y Resolución

El propósito del análisis causal y de la resolución es identificar causas de defectos y otros problemas, y tomar la acción para evitar que ocurran en el futuro.

- SG 1 Determinar las causas del defecto

SP 1.1-1 Seleccionar datos con defectos para el análisis

SP 1.2-1 Analizar las causas

SG 2 Direccionamiento causas de los defectos

SP 2.1-1 Implementar propuestas de acciones

SP 2.2-1 Evaluar el efecto del cambio

SP 2.3-1 Registrar los datos

GG Ejecución de "Tabla de actividades de Objetivos Globales"

4. Aplicabilidad de las nuevas tecnologías

El concepto de "Tecnologías", se deriva del momento histórico en que se ha sido utilizado. Desarrollaremos este concepto hasta llegar al de "Nuevas Tecnologías" (UCM). Existen diferentes definiciones de tecnologías, de las cuales se pueden nombrar las siguientes:

- La palabra tecnología proviene del termino griego *teknites* que eran los artesanos; más tarde *tékne*, el artesanado y también la práctica, la habilidad de hacer las cosas con cierta desenvoltura. Más adelante *tékne* pasa a ser "arte".
- En la revolución industrial en el siglo XVIII, las tecnologías son los artefactos, la maquinaria, el conjunto de los conocimientos propios de los oficios mecánicos y de las artes, o el tratado de términos técnicos.
- Según Gouldner la vinculación entre la tecnología y la ciencia se ha intensificado, rutinizado e institucionalizado, logrando con ello un éxito considerable. Indica "*... esta fusión la que dota a la tecnocracia con la mística de la ciencia, lo que le permite definirse como algo más que el ingenio, la inteligencia, la diligencia y la disciplina del honesto artesano, como algo más que un astuto espíritu práctico animado por la esperanza de la ganancia y el reconocimiento*".
- Según el Diccionario Básico de Comunicación, la tecnología es: "la mutación cualitativa que resulta del agregado de una gran cantidad de actividades artesanales".
- Varios autores señalan la tecnología como una estrategia metodológica mediante la cual se aplica el conocimiento organizado a la solución de problemas. Este conocimiento se desarrolla mediante la ejecución de reglas, modos y procedimientos de la disciplina o actividad, reflejando la pericia o habilidad del individuo, que es en sí la técnica.

Las definiciones de las Nuevas Tecnologías que se han ofrecido, son diversas. Así para Gisbert y otros (1992, 1), hace referencia al "*conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información*". Por su parte Bartolomé (1989, 11), desde una perspectiva abierta, señala que su expresión se refiere a los últimos desarrollos tecnológicos y sus aplicaciones. En esta misma línea en el diccionario de Santillana de Tecnología Educativa (1991), se las definen como los "*últimos desarrollos de la tecnología de la información que en nuestros días se caracterizan por su constante innovación*." Castells y otros (1986) indica que "*comprenden una serie de aplicaciones de descubrimiento científico cuyo núcleo central consiste en una capacidad cada vez mayor de tratamiento de la información*".

Las nuevas tecnologías están sustentadas en la concepción de que las tecnologías son el resultado del conocimiento científico, en la transformación de objetos y en la utilidad, en donde uno de sus principios es la adecuación de medios a fines. Se considera nueva tecnología, cuando la modificación introducida en una tecnología rebasa ciertos límites, la estructura del sistema se resiente y tiene que modificarse dando lugar a la aparición de un nuevo sistema.

La aplicabilidad de las nuevas tecnologías en este trabajo, consiste en el uso de nuevos sistemas en el sistema actual, si utilizamos Ingeniería del Software en este proceso,

4. Aplicabilidad de las nuevas tecnologías

se reduce el estudio del apoyo de la Ingeniería del Software en el uso de estos nuevos sistemas.

Se estudia en dos contextos la aplicabilidad de las nuevas tecnologías, uno es el proceso de Gestión, y otro es el proceso de producción, como se describe en el capítulo 2.2.2. de TPS (ver figura 4.1). Viendo estos dos enfoques, se analiza la influencia de la ingeniería del Software en cada contexto, donde se describe un proceso de mantenimiento en el primer capítulo (donde se describen los conceptos en la Ingeniería del Software aplicable a este trabajo) y en el mejoramiento de proceso en una organización a que se refiere el segundo capítulo en el modelo de procesos CMMI. Ambos capítulos anteriores tiende a describir en forma detallada actividades y procesos que envuelve en los procesos del software en el área de gestión o gerencia en la parte del rol humano. Seguidamente, en este capítulo se describe en forma genérica, la parte proceso de producción del software abarcando en si la Ingeniería del Software.

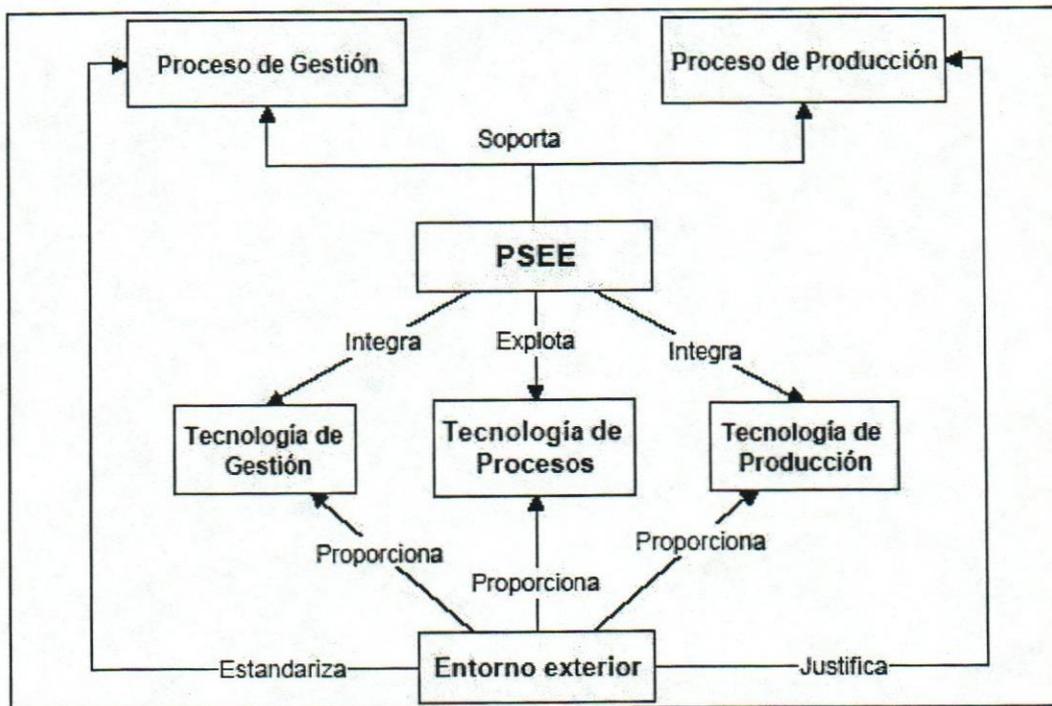


Figura 4.1 Impacto de la tecnología de Proceso del Software

Las nuevas tecnologías vienen desde el entorno exterior, la tecnología de Gestión, tecnología de procesos y/o tecnologías de producción. PSEE se encarga de suministrar y administrar la información hacia los procesos de gestión y producción, aquí impacta cualquier cambio de las tecnologías.

Delimitar en cada proceso descrito el área de Ingeniería del Software, y limitar la Ingeniería del Software con respecto a las demás áreas, es un trabajo arduo. Dado que los dos capítulos anteriores involucra la Ingeniería del Software en forma indirecta o directa, partimos de ellos para generar y tomar elementos propios para la Ingeniería del Software en

4. Aplicabilidad de las nuevas tecnologías

apoyar cambios a las nuevas tecnologías. El otro punto es que en esos procesos esta involucrada explícita e implícita la Ingeniería del Software que influye sobre las demás áreas. Esto es un camino bidireccional, donde permite alimentar todas las áreas con la Ingeniería del Software y viceversa. En este último caso, se define los PSEE definido y detallado en todo el capítulo 2.2. para abarcar y no buscar esa limitación entre las áreas. Si planteamos esta perspectiva, el capítulo 3 esta involucrado en lo que define en sistemas de procesos de software (capítulo 2.4.4.) donde identifican las actividades que se deben realizar por parte de los seres humanos.

Parte de este trabajo, es el estudio de las partes de las nuevas tecnologías, su impacto en los dos contextos (proceso de Gestión y de producción), y lo que debe realizar en cada uno de los contextos. Gran parte del trabajo en la gestión y conceptos en la Ingeniería del Software y su influencia están descritas en el primer capítulo donde se centra el uso de la nueva forma de gestión en el proceso de mantenimiento de un Software, el segundo capítulo sobre CMMI es el uso de modelos de gestión organizacional y modelo de gestión técnicas que involucra la Ingeniería de Software en sus procesos, en la bibliografía de CMMI, se describe claramente en cada uno de su contenido la separación de los PIS con respecto a las demás áreas. Los entornos de MANTIS y CMMI posee en si el concepto de gestión de procesos y proyectos, si existe un cambio a una nueva tecnología a ¿qué afectaría a estos dos sistemas?. Otro punto de estudio es el cambio de la nueva tecnología en la tecnología de producción. Mas adelante se desarrollara en forma mas profunda.

En ambos procesos al ingresar una nueva tecnología el nivel de incertidumbre llega a ser alto, donde impacta el desarrollo y el mantenimiento de cualquier software y finalmente puede impactar al usuario final, por esto, se debe gestionar la incertidumbre, es decir, existe un riesgo latente en el proceso de incluir nuevas tecnologías en los proyectos y por ello conlleva a realizar actividades humanas que no impacten en los proyectos.

Cuando se desarrolla un nuevo proyecto y/o se mantiene un proyecto, y simultáneamente agrega una nueva tecnología el impacto es menor en el mantenimiento, debido al tamaño del cambio del proyecto que afecta la nueva tecnología y la magnitud de su uso.

4.1 Procesos de Gestión

Las fallas encontrada en los años 60 y principios de los 70 en la gestión o administración del proyecto no eran por los administradores o desarrolladores incompetentes, las fallas eran por el enfoque de la gestión utilizada (Sommerville I. 2002, Pag. 72), una buena gestión no garantiza el éxito del proyecto. La gestión del software abarca no solo el ser administradores, adicionalmente debe plantearse diferencias sustanciales como:

- a) El producto intangible.- no se puede ver el progreso en forma tangible.
- b) No existen procesos del software estándares.- se aplicaba administración de otras áreas de la ingeniería pero no son efectivas para el desarrollo del software.
- c) A menudo los proyectos grandes de software son únicos.- la nueva tecnología hace obsoleta la experiencia previa, que no es transferibles a nuevos proyectos.

4. Aplicabilidad de las nuevas tecnologías

En el caso de un nuevo desarrollo y mantenimiento se debe medir dos características importante la complejidad de la nueva tecnología de gestión y el tamaño o escala de esta tecnología. Adicionalmente, se debe evaluar las dos características de complejidad y tamaño del software. Realizar un cambio a nuevas tecnologías antes del desarrollo de un proyecto se debe medir el impacto y si es transferible el proceso a este proyecto. Una forma de medición es el desarrollo en paralelo de dos procesos específicos con dos gestiones diferentes, la actual y la nueva tecnología de gestión. Se debe validar y verificar los resultados de las gestiones. El impacto es menor a un proyecto que este bien avanzado, si el proyecto esta avanzado el impacto es grande, debido al paradigma de la horda mongoliana (Pressman 1998, pag. 12). En el caso de mantenimiento se asume que el producto esta terminado, la gestión es crear un proceso nuevo para el mantenimiento, donde se involucra una nueva gestión, el impacto es mínimo, debido a que la complejidad del resultado debe ser de menor escala al desarrollo total del software.

En los últimos años, se ha desarrollado una variedad de procesos de software de forma estándar, que los administradores de software se pueden apoyar en su labor. Existen dos formas de desarrollo de estos procesos:

- Nuevos procesos
- Cambios por adaptaciones

Tenemos dos ejemplos claros que hemos descritos, los entornos de MANTIS donde se plantea un proceso de software para el mantenimiento y el CMMI que es el resultado de adaptaciones y mejoras del CMM. Los entornos de MANTIS es un planteamiento en el que se basan en los conceptos de los Entornos de Ingeniería del Software y los diferentes ISO que son empleados en las gestiones en los procesos de mantenimiento y calidad de los software. CMMI basado en ISO/IEC 15504 que integran las diferentes disciplinas de ingeniería que manejan el uso múltiples que se maneja en el CMM, lo permite con el uso de framework.

Si comparamos los entornos de MANTIS y CMMI, ambos toman como referencias ISO/IEC 15504, son extensión de este estándar, aún partiendo del mismo estándar, se diferencia por la complejidad y escala o tamaño de estas tecnologías. El proceso de gestión entre ambos vienen bien limitados en dos diferentes procesos, mantenimiento y optimización organizacional en los diferentes procesos. El cual uno puede abarcar el otro. Si aplicamos los entornos de MANTIS en un desarrollo completo de optimización organizacional, puede que funcione esta gestión, pero es altamente factible que falle porque no esta concebido para este proceso. De la misma manera el desarrollo de la gestión en una organización grande al aplicar MANTIS puede que no resulte, debido a que esta gestión esta orientada a mantenimiento de software. Del mismo modo, al aplicar CMMI en un pequeño proyecto puede que sea tan engorroso que es probable que la gestión falle, debido a su complejidad o por sub-utilizar esta gestión. Si vemos estos dos procesos como tecnologías nuevas en las empresas a nivel de gestión, se debe estudiar muy bien el objetivo de la organización o las necesidades de este.

Si comparamos CMM y CMMI, el CMM se desarrolla en varias disciplinas como se menciona en el capítulo 3. El combinar las diferentes disciplinas en un proyecto conlleva al

4. Aplicabilidad de las nuevas tecnologías

manejo de complejidad alta de la integración de éstas. De este modo el CMMI hace la integración de forma menos complicada y en forma automatizada.

Tanto CMMI y los entornos de MANTIS provienen de cambios de necesidades del entorno externo, esa necesidad es de simplificar y automatizar los procesos, que necesariamente sea una evolución de esa tecnología o combinación de varias tecnologías, en el caso de CMMI proviene del CMM, los entornos del MANTIS provienen de diferentes combinaciones de tecnologías de gestión como PCTE (Portable Common Tool Environment), realiza con el PSEE el complemento y amplía la propuesta de ECMA (European Computer Manufacturers Association), y hace referencia a ISO 12207, ISO 15504, ISO 1476.

Existirán, evolucionarán y se crearán nuevas tecnologías en el proceso de gestión desde pasados años, presente y en el futuro, esto viene dado por el mismo progreso de las investigaciones de cada tecnología, nuevos descubrimientos, y nuevas necesidades del entorno externo que empuja a ser cambiante nuestra forma de gestionar. El detalle es la aplicabilidad y la selección de estas nuevas tecnologías en nuestro entorno organizacional, para el mejoramiento de los procesos. Si realizamos la analogía de los procesos y meta-procesos (capítulo 2.5) de la diferencia entre procesos y meta-procesos, para los procesos son todas las variantes en el entorno externos y los meta-procesos son los procesos de investigación, adopción de nuevas tecnologías y descubrimientos de estas tecnologías. Si se determina en estas tecnologías sea coherente y consistente con el entorno externo se mantendrá por si misma en el tiempo, sino, padecerá.

En conclusión, la escogencia de estas tecnologías depende del estudio del entorno organizativo real y de la consistencia de la representación del modelo organizativo de estos procesos. Por otro lado, se debe medir estas tecnologías en dos formas, en su tamaño y complejidad. Un punto importante es el objetivo por el que fue creada esta nueva tecnología aplicada a la organización.

4.1.1. Ingeniería del Software en el Proceso Gestión Organizacional

Cuando se usa Ingeniería del Software en el modelo CMMI se ingresan diferentes áreas como: Proceso de Gestión, Gestión de proyecto, soporte e Ingeniería de procesos. En los entornos de MANTIS el uso de la Ingeniería del Software abarca en todo sus procesos. En rasgos generales, la pregunta sería ¿Qué tanto ayuda la Ingeniería del Software en el proceso de Gestión?. Esta pregunta debe ser enmarcado en dos tipos de gestiones: gestión de procesos de software, y gestión alrededor de estas o gestión organizacional (que incluye en la mayoría de los casos a la gestión de procesos del software), mas adelante se tratarán estos temas. La otra pregunta que debemos hacer, si existe una nueva forma de gestión organizativa como un nueva tecnología, es ¿Qué aporta la Ingeniería del Software en el proceso de gestión?. Para responder este punto se describe en los siguientes párrafos al responder la primera pregunta.

El proceso de gestión de software es administrar el desarrollo del software en una forma eficiente, de alta calidad y de bajo costo, este concepto aparece en los diferentes textos académicos y no difieren tanto en las consecuencias o resultados de esta gestión; la

4. Aplicabilidad de las nuevas tecnologías

gestión alrededor de estas, son gestiones propias de las organizaciones, del cual, las gestiones del software están atadas y restringidas a estas gestiones externas, existe un vínculo entre ambas gestiones (Sommerville I. 2002, Pag. 72). El caso descrito por Sommerville en estar restringidas a presupuesto y calendarización de los proyectos a la que debe ajustarse a la organización que se desarrolla el software. El manejo de la gestión de proceso de software afecta las otras gestiones, en el caso de la velocidad de entrega de productos, con calidad y bajo costo, permite la gestión de negociación y ventas del producto tomar decisiones a sus acciones con sus clientes (usuario finales, otras áreas de la empresa, o a otras empresas). Al tener una deficiencia en la gestión de procesos del software, podemos nombrar numerables efectos de esta relación, tomamos un ejemplo como, el producto realizado con una gestión pobre produce un alto nivel de esfuerzo para las ventas o mercadeo de estas, por otro lado, el mantenimiento tendrá un costo elevado, mas adelante se replantea el efecto de las gestiones de procesos del software a la gestión organizacional. En el sentido contrario, la gestión de la organización afecta la gestión de los procesos de software en gran medida, cuando Sommerville habla de presupuestos, afecta en alguna medida la asignación de recursos de los proyectos, la gestión de suministro, etc. Afecta de alguna forma el recurso humano dentro de los proyectos de desarrollos de software, donde influye directamente en los producto el factor humano (Pressman 1998, pag. 39-43). Por esta razón, existen procesos de gestión diversos, como ejemplo, en la aplicabilidad de los procesos de CMMI involucra la institucionalización de los cambios, los objetivos y el compromiso de la institución. Existe una gestión de proceso de software y gestión organizacional que permite el éxito del producto y servicio (ver "Gestión de procesos" capítulo 3.4.1.). En los entornos de MANTIS se describe en los "Procesos organizacionales" (Capítulo 2.8.1.3.) y "Roles y Organizaciones participantes" (Capítulo 2.8.1.4.). Esto indica que la gestión organizacional impacta a la gestión de procesos de software, sobre este tema existe una gran cantidad de trabajo y áreas de estudio en la gerencia.

Replanteando el vínculo de la gestión Organizacional con la gestión de procesos de software desde el punto de vista de dos o mas organizaciones diferentes, una organización que realiza la gestión de proceso de software y otra organización quien la utiliza para efecto de gestión organizacional. En párrafos anteriores, se describe el impacto de la gestión organizacional sobre la gestión del proceso del software dentro de una organización, en otra perspectiva, el efecto e impacto de la gestión de proceso de software sobre la organización viene dado en el apoyo a la labor de gestión organizacional, es decir, la gestión organizacional debe plantearse el hecho de responder, adecuarse, innovar y anticiparse al entorno externo, para esto se basa en la información u acceso a la información, ya sea en forma electrónico o por procesos humanos para las tomas de decisiones. En este caso el aporte de la Ingeniería del Software esta en los productos desarrollados para el apoyo de estas decisiones y la automatización de la información, generar una integración del conocimiento humano y la integración de soluciones basadas en la información. En donde los productos generados por los entornos de la Ingeniería del Software en los procesos planificados y previstos son los referentes a los desarrollos de los sistemas de información, y en los casos de situaciones no planificadas y anticiparse al entorno externo, se apoya en el desarrollo de los sistemas del conocimiento. Desde este punto de vista, existe una gran relación de gestión organizacional y gestión de procesos del software, en donde este último, debe integrar ese conocimiento organizacional y la gestión organizacional como producto.

4. Aplicabilidad de las nuevas tecnologías

Describiremos mas adelante la influencia o el aporte de la Ingeniería del Software en la gestión organizacional. En la gestión organizacional se pueden describir actividades en las que la Ingeniería del Software influye como:

- En la definición del ciclo de vida del proyecto. La determinación de las fases del ciclo de vida del proyecto en las tomas de decisiones de los períodos de evaluación y realizar decisiones. La Ingeniería del Software determina las fases del proyecto para el software que incluye típicamente la selección y el refinamiento de un modelo del desarrollo del software para tratar interdependencias y para apropiarse en la ordenación de las actividades del proyecto del software (SEI 2002). Determina en el proceso de la gestión organizacional en la evaluación de la selección del software apropiado, la colocación ordenada del uso del software.
- En la estimación de esfuerzos y costos. Dentro del área de la Ingeniería del Software, muchos modelos paramétricos se han desarrollado para ayudar en estimar coste y horario. El uso de estos modelos como la fuente única de la estimación no se recomienda mientras que estos modelos se basan en los datos históricos del proyecto que pueden o no pueden ser pertinentes a su proyecto. Los modelos múltiples y/o los métodos se pueden utilizar para asegurar un alto nivel de la confianza en la estimación. Adicionalmente, se considera el estudio estimación de los recursos críticos de computadora en el ambiente de Host, en el ambiente de la prueba, en el ambiente de producción, o en cualquier combinación de éstos. Las estimaciones incluyen: identificar recursos críticos de computadoras para el proyecto del software; y estimar los requerimientos de recursos críticos de computadoras (SEI 2002). Estima en la gestión organizacional el costo y el esfuerzo de los recursos requeridos en relación al uso de las computadoras y el uso de esta en las estimaciones.
- Establecimiento de plan del proyecto. Planificación y documentación de los planes de desarrollo del software, de proyecto y del software. Un plan documentado que trata todos los artículos relevantes del planeamiento que es necesario alcanzar para la comprensión, el compromiso y el funcionamiento mutuo entre los individuos, de grupos y de las organizaciones que deben ejecutar o apoyar los planes. El plan generado para el proyecto define todos los aspectos del esfuerzo, atando junto de una manera lógica: consideraciones del ciclo de vida del proyecto; tareas técnicas y de la gerencia; presupuestos y horario; gerencia de datos, identificación del riesgo, recursos y requisitos de la habilidad; e identificación e interacción de los Stakeholders. Las descripciones de la infraestructura incluyen las relaciones de la responsabilidad y de la autoridad para el personal del proyecto, la gerencia y las organizaciones de los soportes (SEI 2002). El estudio de la planificación en la gestión de los procesos de software y ayuda a la gestión organizacional en el desarrollo de software.
- Monitoreo de recursos provistos y usados. Los recursos usados por la Ingeniería del Software como las computadoras, periféricos y software usado en el diseño, manufacturación, pruebas y operación; Herramientas de Software, redes, etc. (SEI 2002). La Ingeniería del Software ayuda en la gestión organizativa en las

4. Aplicabilidad de las nuevas tecnologías

mediciones, servicios, operaciones básicas y soporte automatizado en los cambios del área de la información y administración de la información. De aquí se destaca, por ejemplo, el uso de la Intranet, Internet y Extranet en las organizaciones.

En resumen, basando por los procesos organizacionales de los entornos de la Ingeniería del Software (capítulo 2.8.1.3.), donde se detalla el apoyo a la gestión organizativa por el EIS en el modelo de ciclo de vida ISO 12207 de la siguiente clase:

- Gestión: organizar, supervisar y controlar la inicialización y ejecución de cualquier función o proceso para alcanzar el objetivo de negocio de la organización, aplicando gestión de proyectos, gestión de riesgo, gestión de calidad y gestión de medición. Gestiona con herramientas y procesos del EIS para los cambios de nuevas tecnologías, adicionalmente determina la gestión de la necesidad de la organización en materia de procesos de software y en cambios con la nueva tecnología.
- Mejora: lleva las actividades de selección, adquisición, desarrollo, mantenimiento y gestor con otros procesos. Al colocar una nueva tecnología de gestión, permite a la Ingeniería del Software evaluar la tecnología computacional que se adapte al cambio, permitiendo que la selección de la tecnología computacional se adapte a los cambios organizacionales, fácil desarrollo de nuevas funciones (software) y rápida integración con otros procesos y que adicionalmente tenga esta tecnología computacional herramientas de mediciones y controles para los resultados reales hacia la organización. Todo estos componentes deben ser integrales para un fácil mantenimiento, control, desarrollo y expansión hacia el futuro.
- Gestión de recursos humanos: selección de personal con conocimientos y habilidades en las nuevas tecnologías. Donde abarcar tanto personal en la nueva tecnología de procesos de gestión y producción (tecnología computacionales). Permite enfocar los tipos de entrenamientos al personal en la nueva tecnología en el proceso de producción.
- Infraestructura: Gestiona las actividades para el mantenimiento y selección de la infraestructura para el software: HW, SW, métodos, herramientas, técnicas, estándares y cualquier medio para el desarrollo, mantenimiento y operación de los SW de una forma de alta calidad y confiable para el soporte de la nueva tecnología.
- Gestión de Activos: gestiona la vida útil de los activos en forma técnica desde su incorporación hasta su retirada, procedimiento administrativo en los activos para la implementación de la nueva tecnología en el proceso de gestión.
- Gestión de la reutilización: Gestionar planes para establecer, controlar y supervisar el programa de reutilización en la organización, para la tecnología actual y a la nueva tecnología a implantar.
- Ingeniería de dominio: Gestiona el desarrollo, mantenimiento y mejoras del dominio con la integración de la nueva tecnología.

En conclusión, el aporte de la Ingeniería del Software hacia la gestión organizativa es el apoyo de la gestión técnica al implementar la nueva tecnología de gestión. La gestión técnica permite los planes de la selección, implementación, mantenimiento y desarrollo de la tecnología computacional que soporte la nueva tecnología de gestión organizacional.

4.1.2. Gestión de Proceso del Software

En la gestión de proceso del software al utilizar nuevas tecnologías se debe plantear de la misma forma que el desarrollo y mantenimiento de un software, utilizando los modelos de procesos adecuados. Para gestionar el proyecto se plantea de dos clases: gestión de desarrollo y gestión de mantenimiento del software, la gestión de desarrollo, generalmente, es mayor escala que la gestión de mantenimiento. En la descripción de CMMI y los entornos de MANTIS se describe en forma breve las gestiones que se realizan en ambos modelos, con las clases de gestión de proyecto. En ambas gestiones existen coincidencia en las actividades, y si estudiamos otros modelos de desarrollo las consecuencias u objetivos finales son los mismos. En la gestión el manejo de las nuevas tecnologías lo tomamos como un requerimiento básico para la gestión en el proceso de software.

4.1.2.1. Gestión en Proceso desarrollo del Software

Tomamos como base el modelo CMMI, donde se identifican los procesos de gestión del proyecto. Están incluidos en el áreas procesos: Gestión de procesos, Gestión de proyecto, Ingeniería y Soporte.

4.1.2.1.1. Gestión de Proceso de desarrollo del Software

En la gestión de procesos de desarrollo del software se basa en el capítulo 3.4.1. donde identifican claramente las siguientes áreas:

- Enfoque proceso organizacional
- Definición de proceso organizacional
- Entrenamiento organizacional
- Funcionamiento de proceso organizacional
- Innovación y despliegue Organizacional

Al definir la gestión de estos proceso como una tarea básica donde hay que gestionar evaluar y adaptar las nuevas tecnologías que soporten las necesidades, ejecución, crecimientos y mejoras de los objetivos trazados. Esta gestión de evaluar, seleccionar y medir la nueva tecnología en forma macro debe soportar la necesidad de la organización.

4.1.2.1.2 Gestión de proyecto en Proceso desarrollo del Software

En la gestión de proyecto por el CMMI, indicado en el capítulo 3.4.2., se identifican varias áreas de procesos como se indica:

- Planificación de proyecto.- estimar el ciclo de vida de la nueva tecnología a implementar o estimar el ciclo de vida de la plataforma de la nueva tecnología. Planificación de acceso y arquitectura de almacenamiento de datos, identificar debilidades de la nueva tecnología, estimar el esfuerzo y tiempo de esta nueva tecnología contra otra tecnología ofrecida, costo de la nueva tecnología, estimación

4. Aplicabilidad de las nuevas tecnologías

de nivel de especialización de conocimiento o habilidades para el uso de la nueva tecnología y su estudio de portabilidad otra tecnología. Planificación del plan del proyecto y obtener compromiso para el plan.

- Control y monitoreo de proyecto.- Planificación de monitoreo para evaluación de la nueva tecnología.
- Gestión de contrato de suministro.- estudio y selección de suministradores de la nueva tecnología, garantía de existencia de los suministradores de la tecnología por lo menos hasta la estimación del ciclo de vida del proyecto.
- Gestión de Integración de proyecto por IPPD.- Gestionar y establecer el uso y la adaptación de la nueva tecnología en la organización, determinar stakeholders relevantes y equipos.
- Gestión de riesgo.- Gestión de identificación, determinación, análisis y contingencia de los problemas y debilidades de la nueva tecnología.
- Teaming Integrado.- Identificar, establecer y asignar los Stakeholders y equipos que soporten, y tenga la habilidad y el conocimiento en las nuevas tecnologías.
- Gestión Integrada de suministro.- Selección, e identificación de los suministradores y productos de la nueva tecnología confiable, adicionalmente, evaluar los productos de la nueva tecnología que permita llegar al objetivo de la organización.
- Gestión de proyecto cuantitativo.- Gestión de selección, construcción y ejecución de procesos que permiten medir en forma estadística la nueva tecnología. Dividir los procesos en subprocesos que permitan la medición y aplicación de controles. Gestión de tipos de mediciones y técnicas analíticas, almacenamiento de información y monitoreo de los subprocesos.

4.1.2.1.3. Ingeniería

En la gestión de la ingeniería por el CMMI, indicado en el capítulo 3.4.3., se identifican varias áreas:

- Gestión de requerimientos.- Manejar los requisitos y ventajas de la nueva tecnología y sus componentes e identificar inconsistencia contra los planes del proyecto, requisitos y productos del proyectos. Identificar del mismo modo las consistencia entre los componentes de la nueva tecnología y los requerimientos.
- Gestión de desarrollo.- gestión de desarrollo y análisis de los requisitos de los clientes, producto y producto-componente bajo esta nueva tecnología, recolección de información de los Stakeholders y la parte técnica, extraer necesidades de los stakeholders y/o técnicas de la organización. Establecer, localizar e identificar componentes o productos de la nueva tecnología. Análisis y validación de los escenarios técnicos, funcionales y conceptos operacionales con el uso de la nueva tecnología. Analizar, validar y comparar con otra tecnología y la actual.
- Solución técnica.- Desarrollo de soluciones alternativas y criterio de selección. Diseño de escenarios con la nueva tecnología. Seleccionar los componentes de la nueva tecnología que cumplan con los requisitos de las soluciones. Gestionar el diseño, desarrollo e implementación de las soluciones de los requisitos basado en la nueva tecnología.
- Integración del Producto.- Manejo de ensamblar los diferentes componentes seleccionados que funcionen y la entrega del producto. Integración con productos

4. Aplicabilidad de las nuevas tecnologías

mas complejos y nuevos componentes en forma incremental que soporten la nueva tecnología e Integración con otras tecnologías. Gestionar las interfases dentro de la misma tecnología y con otras tecnologías. Esta integración se puede determinar de componentes a componentes, de componentes a productos, de productos a productos de la misma tecnología y de productos a productos de diferentes tecnologías.

- Verificación.- Gestión de verificación de integración de las diversas combinaciones.
- Validación.- Gestión de la satisfacción de los diversas integraciones.

4.1.2.1.4. Soporte

En la gestión del soporte por el CMMI, indicado en el capítulo 3.4.4., se identifican varias áreas:

- Gestión de configuración.- Gestión de identificar y mantener la integridad, usando configuraciones. Las configuraciones son para tener niveles de granulación, permite trabajar el software en diferentes tipos de configuraciones, en ellos se basa por la Ingeniería del Software en los códigos fuentes diferentes tipos de compiladores, herramientas, etc. que soporte en la nueva tecnología. En esto existe el manejo de identificación, control, contabilidad de estado y auditoria de la configuración. La nueva tecnología debe permitir el fácil manejo o herramientas de control de versiones, administración de versiones, entregas y cambios, y construcción del sistemas.
- Proceso y garantía de la calidad del producto.- Gestionar que el recurso humano y la gerencia en la penetración de la nueva tecnología.
- Medición y Análisis.- Gestiona el establecimiento, recolección y almacenamiento de datos para las mediciones y análisis de estas en el uso de la nueva tecnología.
- Análisis de Decisión y resolución.- Analizar las diferentes decisiones posibles y el uso de la forma de evaluación formal versus los criterios establecidos. Resolución del problema.
- Ambiente Organizacional para la Integración.- Gestionar la integración de la infraestructura de la nueva tecnología a la organización. Gestionar el recurso humano para la integración.
- Análisis causal y Resolución.- Canalizar los defectos y soluciones del problema y acción para que no vuelva ocurrir con la nueva tecnología. Puede que surja de una nueva versión de la tecnología o por controles de los componentes.

4.1.2.1. Gestión en Proceso mantenimiento del Software

Tomamos como base el modelo entornos de MANTIS en el modelo de referencias para procesos: ISO 15504 (capítulo 2.8.2.), donde se identifican tres aspectos importantes para el mantenimiento: Cultura de mejoramiento y establecimiento de mecanismo para apoyarlo y mantenerlo; la ingeniería de procesos para satisfacer los requisitos del negocio y la optimización de recursos.

Si existe la cultura de mejoramiento en la organización existe como regla en la gestión organizativa. Un punto de mejoramiento en la organización es estar

4. Aplicabilidad de las nuevas tecnologías

tecnológicamente actualizado, esto implica el compromiso de la organización en la adquisición de las nuevas tecnologías. Esta nueva tecnología debe permitir cambiar en forma incremental o facilitar los cambios, donde es una característica importante en el mantenimiento y permite alargar el ciclo de vida de la nueva tecnología. Debe ser rico en componentes y de arquitectura abierta para apoyar los mecanismos.

La nueva tecnología debe satisfacer los requisitos del negocio, por medio del manejo del ciclo de vida de los requerimientos debe ser de menor costo y tiempo en su desarrollo, y fácilmente integrable y de forma integral.

En la optimización de los recursos, la nueva tecnología debe permitir herramientas de evaluación de los componentes, productos y su ambiente. La filosofía de trabajar de esta tecnología es que permita el uso eficiente de los recursos. El uso de esta tecnología debe permitir disminuir el costo, tiempo de respuestas rápidas de los recursos y de calidad.

En el capítulo "4.1.1. Ingeniería del Software en el Proceso Gestión Organizacional" se menciona la influencia de la gestión afectada por la Ingeniería del software en ISO 12207, mas adelante se describe la nueva tecnología vista por el mantenimiento del software haciendo diferencia con ISO 15504 en agregar nuevos procesos al ISO 12207 de los cuales se pueden nombrar:

- Elicitación de requerimientos: gestión de recompilar, procesar y hacer seguimiento de la evolución de las nuevas tecnologías. Establecer si la nueva tecnología satisface a los requerimientos de los usuarios y a la organización.
- Gestión de proyecto: Su objetivo es identificar, establecer, coordinar y supervisar las actividades, tareas y recursos necesarios de la nueva tecnología que permita producir un producto o servicio que cumpla con los requisitos.
- Gestión de la Calidad: Este proceso busca supervisar la calidad de los productos, componentes y/o servicios de la nueva tecnología y asegurar que satisfagan a la organización. El interés se debe centrar en la calidad de la nueva tecnología.
- Gestión de Riesgos: El propósito es identificar y evaluar los riesgos de la nueva tecnología.
- Alineamiento Organizacional. Se identifica esta tecnología con las necesidades organizacionales.
- Medida: El propósito de este proceso es planificar la recolección y análisis de los datos relativos a los productos desarrollados y a los procesos implementados por una unidad organizacional con la nueva tecnología.
- Mantenimiento del Sistema y del Software: gestión de mantenimiento del software con la nueva tecnología y, adicionalmente, la infraestructura tecnológica que soporte esta tecnología.
- Reutilización: gestión del proceso de la Reutilización en ISO 12207, usando la nueva tecnología. Reutilización de la tecnología actual en la nueva tecnología y viceversa.

Independientemente del modelo que se debe escoger para la gestión de mantenimiento del software con la nueva tecnología, se debe evaluar para determinar su aplicación, por

4. Aplicabilidad de las nuevas tecnologías

ejemplo el CMM y CMMI es adecuado en software de la larga vida y largo plazo por organizaciones grandes (Sommerville, I. 2002, pag. 573).

En ambos modelos tanto el CMMI y MANTIS contiene en forma de objetivos y actividades principales a nivel de calidad, se describe en tres grandes niveles (Somerville, I. 2002, pag. 537): aseguramiento de la calidad, Planeación de la calidad y control de la calidad. En las actividades de gestión tanto el desarrollo y mantenimiento se ven reflejadas estas actividades y procesos.

4.2 Procesos de Producción

El proceso de producción se relaciona en la producción y mantenimiento del producto (capítulo 2.2.1. características de procesos del software), el impacto que se genera el uso de la nueva tecnología en la producción y mantenimiento del producto se minimiza con las gestiones y actividades de estas. En este capítulo destallaremos estas actividades con el uso de la Ingeniería del Software, en los primeros subcapítulos describiremos a lo que nos enfrentamos al hablar de nuevas tecnologías, luego separamos en dos procesos: mantenimiento y desarrollo. Si enfocamos las actividades de desarrollo o mantenimiento del software en un conjunto de actividades para incorporar la nueva tecnología, determinamos a simple vista que la escala y magnitud del trabajo es diferente en ambos casos. El caso de desarrollo la escala y magnitud es mayor que el mantenimiento.

Si partimos que un software se descompone en funciones o módulos altamente cohesivo, modular y de bajo acoplamiento, en el caso de desarrollo es acoplar todos estos componentes, módulos o funciones en un producto, al usar la nueva tecnología desde el comienzo del proyecto, partimos que estos componentes, funciones o módulos están bajo esta nueva tecnología. El impacto es general en el software a desarrollar, puede o no afectar los demás software ya implementados, o mas general a tomo el sistema.

En el caso de mantenimiento del software se estima que el impacto es menor, debido a que el software esta desarrollado bajo una tecnología actual que se asume que esta probada, al ingresar el nuevo módulo, función o componente con una nueva tecnología, la proporción entre la tecnología actual y la nueva tecnología, es mayor el primero, el control de riesgo es basado en la tecnología actual y bajo control de la nueva tecnología adquirida, al menos, que la nueva tecnología absorba la tecnología actual e influya de forma radical. Si las condiciones última sucede, de que el cambio de un componente con la nueva tecnología afecta a la tecnología actual.

Se puede inferir varios puntos y los separamos en varios niveles de abstracción en cualquiera de los aspectos tanto de desarrollo como de mantenimiento, los cuales describiremos:

- La función o módulo con la nueva tecnología colocado en la tecnología actual, solo afecta al módulo o a la función, y no afecta al software completo.
- La función o módulo con la nueva tecnología colocado en la tecnología actual, afecta al software completo.

4. Aplicabilidad de las nuevas tecnologías

- La función o módulo con la nueva tecnología colocado en la nueva tecnología, afecta al software completo.
- La función o módulo con la nueva tecnología colocado en la nueva tecnología, solo afecta al módulo o función, y no al software completo.
- Software desarrollado completamente con la nueva tecnología en la tecnología actual, afecta solo al software.
- Software desarrollado completamente con la nueva tecnología en la tecnología actual, afecta a otros software.
- Software desarrollado completamente y puesto en el ambiente de la nueva tecnología, afectan a los software de la tecnología actual.
- Otras combinaciones que afecta o no al hardware.

El estudio de cada caso puede ser largamente discutido y medir su impacto en cada opción y condición, adicionalmente, el estudiar desde los códigos hechos o creación de nuevos códigos hasta estudiar la tecnología, tanto la actual y la nueva tecnología, es un trabajo arduo, al estudiar todos los casos que puede suceder, que aparezca nuevas tecnologías en el entorno externo y el estudio pierda vigencia. Hacer un estudio mas eficiente de todos estos casos en particular no entra en este trabajo, pero si podemos plantear una forma mas puntual que ayude el estudio de cada caso. Si nos basamos en el estudio de enfocar el impacto en cada ente que hemos mencionado, como:

- Funciones
- Módulos
- Software
- Sistema operativo
- Hardware
- Tecnología

La nueva tecnología puede ser agregada en uno o varios entes que enumeramos anteriormente. Se debe agregar al estudio la forma de conexión, comunicación o acoplamiento entre cada uno de los entes. Dependiendo la comunicación entre ellas y con ellas misma, afecta o no la nueva tecnología, del mismo modo, la comunicación entre los diferentes entes también es una tecnología por si sola. Basando en la Ingeniería del Software vemos que existen varios conceptos involucrados en este estudio, como la modularidad, arquitectura, reusabilidad y la comunicación. Empecemos a describir cada concepto aplicado a este estudio.

4.2.1. Modularidad de las nuevas tecnologías

La modularidad según Pressman (1998, pag. 239-240), puede ser efectivo o eficiente, el caso de ser eficiente se basa en el nivel de la cohesión y el acoplamiento de las funciones o módulos, es eficiente si existe una cohesión alta y acoplamiento bajo. Si aplicamos este concepto a las funciones y a los módulos con el uso de una nueva tecnología, el efecto debe afectarse en la función, es decir, al ocurrir cualquier error, este error debe quedar e impactar en la función o módulo. Existen herramientas de desarrollo que no permite una cohesión baja, por ejemplo, el caso de Visual C++ en el uso de objeto y clases en el desarrollo, y existen herramientas que permite trabajar con cohesión baja como

4. Aplicabilidad de las nuevas tecnologías

Turbo Pascal, Borland C. En ambas herramientas, dependiendo de la habilidad del desarrollador puede manejar el nivel de cohesión a su antojo, es decir, aunque la herramienta desarrollo no permita el desarrollo de funciones y módulos con una cohesión baja, el desarrollador lo haga y viceversa, si la herramienta de desarrollo solo permite una cohesión baja, el desarrollador enfoca en sus códigos un nivel de cohesión alta. En conclusión la decisión esta en las manos del desarrollador, aun que la tecnología lo permita o no.

Existe otro concepto referente a la modularidad, pero de descomposición modular de la estructura del sistema (Sommerville, I. 2002, pag. 219-232). En este explica que existe dos modelos que son: modelo orientado a objetos y un modelo de flujo de datos. Explica que estos modelos son de bajo acoplamiento, no especifica el nivel de cohesión. Dependiendo de la estructura del sistema podemos manejar las dos hipótesis: de alta cohesión y baja cohesión. Esto se aplicaría a los entes de software y sistema operativo, y en algunos casos la descomposición modular puede llegar hasta las funciones del software. Aquí depende de la implementación y el uso en cada uno de los entes en esa tecnología en el tema de cohesión. El efecto de la cohesión debe ser análoga, según lo presentado por Presuman, enfocado a los sistemas operativos y software. Los ejemplos enfocados al software y los sistemas operativos UNIX y los de Windows, donde el sistema operativo UNIX es afectado directamente por el software y el software permite adicionalmente el acceso directo a los recursos del hardware, mientras que en los sistemas Windows, el software solo permite acceso al sistema operativo y no al recurso del hardware en forma directa, sino por medio del sistema operativo. En estos casos se detecta que el software puede ser altamente cohesivo con el sistema operativo en el caso de Windows y medianamente cohesivo en el caso de UNIX. El caso de la cohesión de las funciones, en el párrafo anterior, se aplica de la misma manera en este estudio, depende del desarrollador en crear software altamente cohesivo o no, pero la diferencia en los sistemas operativos, es que lo permita o no, en el caso de Windows podemos simular el nivel de cohesión alta, colocando un software o componente entre el software a desarrollar y el sistema operativo.

El estudio del nivel de uso de los recursos de hardware y del sistema operativo, el desarrollar un software se debe no acceder con frecuencia al hardware o al sistema operativo, a medida que se haga esta actividad el nivel de cohesión se vuelve alta, colocando el uso de hardware o sistema operativo a componentes, funciones o módulos individuales que es llamado por el software. Ejemplos de la cohesión entre el sistema operativo y hardware, tenemos grandes ejemplos entre la tecnología "propietarias" y la tecnología actual, los sistemas operativos de la tecnología "propietaria" funcionan en hardware fabricado por los mismos diseñadores del sistema operativo, mas el sistema operativo no funciona en otro hardware. En la tecnología actual, la mayoría de los sistemas operativos puede estar funcionando independientemente del hardware y del fabricante.

Si manejamos el nivel de tecnología el nivel de cohesión y acoplamiento, abarca todos los entes nombrados anteriormente e inclusive la filosofía de la tecnología a usar, la tecnología contiene estándares, actividades y uso de esa tecnología en específico. Dependiendo de esta abstracción, el nivel de cohesión en todos sus ámbitos, puede afectar e impactar el uso de esta tecnología enormemente o puntualmente en el sistema o parte del sistema.

4. Aplicabilidad de las nuevas tecnologías

El otro punto a estudiar en la modularidad es el acoplamiento, en párrafos anteriores hemos mencionado algo sobre el acoplamiento. Los niveles de acoplamientos, es el nivel de interconexión entre dos funciones o módulos en una estructura de programa (Pressman 1998, pag. 240-241). En este concepto se aplica en forma análoga entre software (que puede ser sistema operativo), aunque Pressman no mencione el acoplamiento entre software. Mientras el nivel de acoplamiento sea mayor, el nivel de reusabilidad disminuye.

4.2.2. Arquitectura de las nuevas tecnologías

La arquitectura de las nuevas tecnologías difieren dependiendo de como la estructura del sistema este diseñada. Al estudiar la base de la estructura del sistema, estamos estudiando la base de la arquitectura de las tecnologías. Esto es debido a que el reflejo de una tecnología plasmada en las computadoras, da como resultado el producto en software y/o hardware, herramientas de desarrollo, etc. de la tecnología. Podemos nombrar tres grandes modelos, basándonos en la "Estructura del Sistema" de Sommerville (2002, pag. 219-224): el modelo de depósito, el modelo cliente – servidor, el modelo de máquina abstracta. Las nuevas tecnologías pueden generar una nueva estructura diferente o reutilizar y combinar las existentes.

4.2.2.1. Modelo de depósito

Los subsistemas que componen un sistema deben intercambiar información con el fin de que pueda trabajar en forma conjunta y efectiva (Sommerville, I. 2002, pag. 222). Existen dos formas para lograrlo:

- Base de datos central compartida para ser accedido por cada subsistema.
- Cada subsistema tiene su propia base de datos. Los datos son intercambiados entre subsistema por mensajes.

En el caso de Base de datos centrales, son las herramientas CASE, los sistemas CAD, los sistemas de mando y control, y sistemas de administración de información. Este tipo de arquitectura tiene ventajas y desventajas, las cuales al ser implementados o adquiridos hay que tenerlo en cuenta lo siguiente:

- Eficiencia de compartir los datos. No es necesario de transmitir datos de un subsistema a otro.
- Los subsistemas deben estar acordes con el modelo. Puede ser difícil o imposible integrar nuevos subsistemas acordes al modelo.
- Los subsistemas que producen datos no necesitan saber cómo son utilizados por otros subsistemas.
- Difícil de evolucionar si existen muchos datos. Traducir a un nuevo modelo es difícil o imposible de realizar.
- Respaldo, control de acceso, seguridad y recuperación de errores son centralizadas.
- Cada subsistema tiene su propias políticas de seguridad, recuperación y respaldo, a diferencia del centralizado que utiliza todos los subsistemas con la misma política.
- El modelo de compartir es visible a lo largo del esquema. Facilidad de nuevas herramientas.

4. Aplicabilidad de las nuevas tecnologías

- Si existe distribución de datos, habrá problema de redundancia e inconsistencia de los datos.

Las nuevas tecnologías basadas en este modelo, al ser agregadas a la tecnología actual puede generar los siguientes:

- Compatibilidad de los datos. Si la tecnología actual soporta este modelo, el cambio es mínimo y es heredada la nueva tecnología a su subsistema. De lo contrario, la nueva tecnología no es soportada, produciendo un gran impacto.
- Dificultad de migrar a otras tecnologías si posee muchos datos.
- Dificultad de crecimiento.
- Funciona todo o no funciona nada, por tener los datos centralizados. Al existir un error o haber incompatibilidad de los datos de este modelo con la tecnología actual, la tecnología falla.
- Una mala definición de las políticas centrales genera un colapso en sus subsistemas.
- Fácil reuso del modelo en desarrollo de nuevas herramientas.
- Los esfuerzos de la tecnología esta centrado en la disponibilidad, redundancia y recuperación.

4.2.2.2. Modelo de cliente – servidor

Es el modelo de sistema distribuido que muestra cómo los datos y el procesamiento se distribuyen en varios procesadores.

Este modelo esta compuesto de servidores independientes que ofrecen servicios, clientes que llaman los servicios ofrecidos por los servidores y la red que permite la conexión entre cliente - servidor (Sommerville, I. 2002, pag. 222).

El servidor y el cliente varían dependiendo de la función que ejercen en el procesamiento de la presentación, la lógica y los datos del sistema. Desde el esquema en que el servidor procesa todos los datos, toda la lógica y organiza los datos para ser presentado al cliente hasta el esquema en que el servidor solo administra los datos centralizados y el resto del procesamiento los realiza el cliente (parte de datos, toda la lógica y toda la presentación). Dependiendo del esquema el cliente puede, hasta cierto punto, ser independiente al servidor u operar a media si el servidor no este disponible.

La nueva tecnología basada en este modelo posee ventajas y desventajas de las cuales mencionamos:

- Distribución de sus componentes en los clientes, deben tener un nivel de control y administración para distribución del software, estructura de datos lógicos y datos.
- Distribución de datos, puede producir redundancia e inconsistencia.
- Excelentes condiciones de la red. Puede colapsar el sistema si toda la red no esta disponible. La condición de la red dependerá del desempeño de las respuestas del sistema.
- Mala distribución de los datos genera tráfico en la red.
- Facilidad de agregar nuevos servidores o clientes sin afectar a la estructura actual.

4. Aplicabilidad de las nuevas tecnologías

- Al no estar un servidor disponible los demás pueden seguir trabajando y no afectar todo el sistema.
- Puede soportar diversidad de tecnología, cada servidor y cliente no necesariamente deben tener la misma tecnología, solo depende del estándar de comunicación entre todos sus elementos. Convivir varias tecnologías.
- El nivel de integración es dependiendo de las diferentes tecnologías que conviven entre si, y es la suma de el nivel de integración de cada tecnología (ver capítulo 3.2.1.).

4.2.2.3. Modelo de máquina abstracta

El modelo de máquina abstracta de una arquitectura (algunas veces llamados modelo de capas) modela la interacción entre los subsistemas. Organiza un sistema en una serie de capas cada una de las cuales suministra un conjunto de servicios. Cada capa se utiliza para implementar el siguiente nivel de la máquina abstracta. Un ejemplo es el enfoque OSI para los protocolos de red (Sommerville, I. 2002, Pag. 223-224).

Las ventajas y desventajas de este modelo son:

- Permite desarrollo incremental.
- Es portable y cambiante. Una capa puede ser reemplazada por otra.
- Cuando una capa cambia afecta las capas adyacentes.
- Puede cambiar de máquinas en reimplementar la capa mas interna con dependencia al hardware a otro.
- Difícil estructurar los sistemas de esta forma.
- El desempeño es un problema si el recurso necesitado no esta en la capa inmediata, sino, en la interna.
- Si existen demasiadas capas el nivel de administración es grande.

4.2.2.4. Modelo de híbrido

El modelo híbrido es la combinación de los modelos anteriores. Un ejemplo de este modelo es combinar Modelo Cliente – Servidor y el modelo de máquina abstracta, donde cada capa puede estar definida en un servidor, de esta forma, el servidor dependiendo de las condiciones se vuelve cliente para el siguiente servidor (otra capa).

Las ventajas y desventajas de este modelo, viene dada por la suma de los modelos de la máquina abstracta y cliente - servidor, adicionando:

- Cada función de presentación al usuario, datos y lógica, puede ser representada en un servidor cada uno, es decir, un servidor para la presentación, un servidor para la lógica y otro para los datos. Puede realizar la implementación en uno o mas servidores, dependiendo del procesamiento de cada función.
- Facilidad de estructuración si se representa por funciones bien definidas (o capas).

Un ejemplo mas práctico de este modelo es Microsoft DNA/.NET. Ver la siguiente figura:

4. Aplicabilidad de las nuevas tecnologías

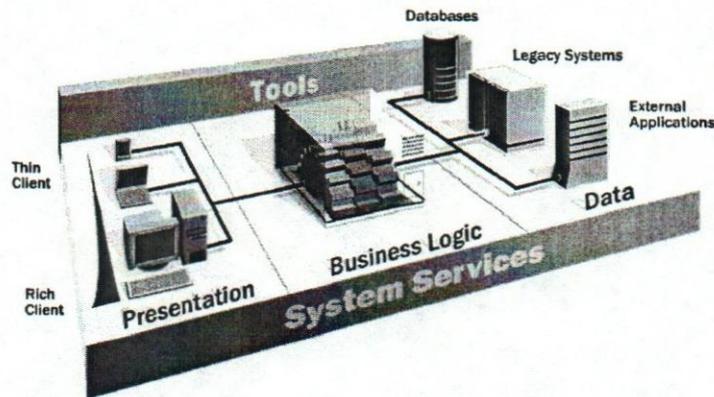


Figura 4.2. Multicapas

4.2.3. Reusabilidad de las nuevas tecnologías

El concepto de reusabilidad es descrito tanto por Pressman como por Sommerville como el concepto de “reutilización”. Sommerville (2002, pag. 307) define la Ingeniería del Software como un enfoque de desarrollo que trata maximizar la reutilización de software existente. Mientras Pressman (1998, pag. 485) señala a Peter Freeman el concepto de reutilización como cualquier procedimiento que produce (o ayuda a producir) un sistemas mediante la reutilización de algo procedente de algún esfuerzo anterior. Ambos autores presentan la reutilización del software en dos sentidos, por la unidad de tamaño y artefactos reutilizables. En nuestro estudio para las nuevas tecnologías, se usa los dos sentidos en reutilizar o implantar esta tecnología en la tecnología actual de la organización.

4.2.3.1. Reusabilidad por unidad de tamaño de las nuevas tecnologías

El concepto de reutilización descrito por Sommerville (2002, pag. 307) como unidades de tamaño se presenta de esta forma:

- Reutilización de funciones: es lo común en reutilizar una función o algoritmo de las librerías estándar de la nueva tecnología; este algoritmo, de alguna forma, lo podemos transformar en la tecnología actual, con el uso de reingeniería inversa o el uso de un modelo de lenguaje de proceso y adaptarlo a los lenguajes de desarrollo de la tecnología actual. Por ejemplo: cálculos matemáticos, algoritmo de reconocimiento de patrones, etc.
- Reutilización de componentes: es la utilización de un conjunto de funciones, módulos, objetos y subsistema de la nueva tecnología acoplados a la tecnología actual. Por ejemplo: uso de DCOM de MS en el manejo de un sistema de inventario, uso de módulo de comunicación J2EE de IBM, etc.
- Reutilización de sistemas de aplicaciones: es el uso de la totalidad de los componentes de la nueva tecnología o de los programas de la nueva tecnología. Por

4. Aplicabilidad de las nuevas tecnologías

ejemplo: Uso de control, política, y conexión de ORACLE en un ambiente MS o IBM.

Este subcapítulo permite el uso de componentes internos en forma abstractas (filosofía de uso) hasta todos los sistemas de la nueva tecnología en el desarrollo y/o mantenimiento de un software.

4.2.3.2. Reusabilidad por Artefactos Reutilizables

El concepto de reutilización descrito como Artefactos Reutilizables (Pressman 1998, pag. 488-489) se presenta los siguientes candidatos para la reutilización:

- Planes de proyectos: La nueva tecnología presenta un conjunto de planes y actividades para la gestión de los proyectos bajo esta tecnología.
- Estimaciones de coste: La nueva tecnología posee actividades y estimaciones generales y puntuales de proyectos anteriores o los laboratorios del suministrador de la tecnología. Existen actividades para estimaciones en condiciones para un nuevo cliente o estimaciones de clientes que ya han sido implementados con esta tecnología.
- Arquitectura: La nueva tecnología presenta programas básicos, datos o plantillas que son reutilizados y modificados a la necesidad de la organización.
- Especificaciones y modelos de requisitos: Los modelos y especificaciones de clases y objetos de la nueva tecnología son candidatos de la reutilización. Modelos de análisis de la nueva tecnología enfocado a la Ingeniería del Software.
- Diseños: Diseños arquitectónicos, de datos, de interfaz y de procedimientos desarrollados presentados en la nueva tecnología.
- Código fuentes: Consideremos lo descrito en el capítulo 4.2.3.1. de unidades de tamaños.
- Documentación del usuario y técnica: Uso de la documentación de la nueva tecnología en adicionar o modificar a los cambios que han surgido en el ciclo de vida dentro una organización. Institucionalizar o adquirir la forma de documentar de esta nueva tecnología en la organización.
- Interfaces Humanas: Uso del IGU de la nueva tecnología en los sistemas y en la tecnología actual.
- Datos: Uso de las estructuras de archivos, datos, archivos, listas o toda la base de datos y las políticas de la nueva tecnología.
- Casos de pruebas: Siempre al utilizar un candidato de reutilización de la nueva tecnología, el caso de prueba es relevante. Implica una aplicación práctica y visible del cambio.

4.2.4. Comunicación de la nueva tecnología

La comunicación es una tecnología que abarca una característica fundamental en los procesos, desde el punto de vista de gestión hasta la parte de proceso de producción. Los procesos de Gestión (capítulos 4.1.1.) se basan en la organización de la comunicación eficiente entre el recurso humano del manejo de la información, las actividades y los objetivos que se deben desplegar en toda organización o en los equipos de desarrollo del

4. Aplicabilidad de las nuevas tecnologías

proyecto. En este subcapítulo estudiaremos, en el aspecto de producción, como una tecnología que es involucrada en todos los procesos afecta la interconexión entre dos o mas subsistema o entes.

Las telecomunicaciones son parte del estudio en un área de la ingeniería que va a la par con el área computacional. Desde los principios de transferencias de datos hasta nuestro día que involucra el manejo de transmisión de data, voz e imagen, han alcanzado un avance significativo y de madurez que permite a los desarrolladores de software verlo de otra perspectiva. El uso de las conexiones vía satelital ayuda en la reducción de costos y acceso a todos los lugares del mundo cosa que en la tecnología de antes no se permitía a nivel computacional (por cableado). Un ejemplo típico de reutilización de los recursos es, vista desde el punto computacional, la utilización de transmisión de información vía telefónica y la infraestructura ya instalada desde hace años con tecnología análoga donde es reutilizado en la tecnología digital. Replantea la reutilización de los recursos ya implantados de una forma eficiente. La Ingeniería del Software plantea los enfoques de utilización de esta tecnología, desde el punto de vista que el desarrollador no advierte el uso de las telecomunicaciones en sus proyectos como una incapacidad en los proyectos. Este punto de vista para la Ingeniería de Software lo identifica como un componente mas para su uso.

Las nuevas tecnologías en esta área se plantean desde el punto de vista de mantener viejas estructuras y reutilizarlas con las nuevas tendencias de comunicación. Un punto de estudio de esta nueva tendencia es el uso de datos, imagen y voz, usando infraestructura análogas de años anteriores, este trabajo se centra en tecnología de eficiencia de uso del recurso de señales análogas en la transmisión de esta información; algoritmo mas eficiente en el recurso y modelos de comunicación. El otro punto de estudio es la transmisión en nuevos recursos o nuevas tecnologías de conexión, como señales satelitales y señales de ondas sin el uso de cables o usar materiales mas eficiente en los cables de transmisión.

La reutilización de conexiones es una de las partes que debemos evaluar en este trabajo, se enfoca en dos caminos: en la reutilización de una arquitectura ya existente y la reutilización de una nueva tecnología en la tecnología actual, como se enfoca en el subcapítulo 4.3.2. "reusabilidad de las nuevas tecnologías".

En la reutilización de una arquitectura ya existente tenemos como el caso del protocolo de comunicación como el SNA (Systems Network Architecture) desarrollado en 1970 por IBM. Esta es una conexión punto a punto en su concepción (APPN, Advanced Peer To Peer Networking), es decir, es un enlace que físicamente va desde un computador a otro, esta filosofía de conexión es obligatoria en los años anteriores, este protocolo no es enrutable, no permite direccionar en forma dinámica el paquete de envío o de recepción, esta arquitectura era y es una tecnología completa en comunicación para su época. El avance de esta tecnología tuvo la necesidad de adaptarse al uso de otros tipos de acceso al medio como canales mainframe como SDLC (Synchronous Data Link Control), X.25, Token Ring, etc. El SDLC se utiliza para la interconexión de controladores de comunicación y controladores establecimiento. X.25 se utiliza en redes WAN para interconexión de nodos SNA. IBM dispone en la actualidad la posibilidad de usar otros estándares, como Ethernet (IEEE802.3), FDDI o Retransmisión de tramas o paquetes (Frame Relay), entre otros (Sánchez-López, pag. 160-162). El uso de la red digital permite

4. Aplicabilidad de las nuevas tecnologías

el uso de la WAN y LAN para este tipo de protocolo, como TCP/IP, estos protocolos tienen que ser enrutables. Ambos protocolos (X.25 y TCP/IP), puede convivir junto, como otros diversos protocolos para manejar en forma eficiente el mismo recurso de la red. Las primeras versiones de SNA, al tener una falla de conexión de punto a punto, la comunicación muere entre los dos equipos. Al usar el protocolo X.25 con tecnología Ethernet, donde permite que físicamente existe una falla en la conexión, este se reconfigura en forma de conseguir otra vía física para la conexión y reparar la falla en forma paralela, sin perder la disponibilidad de la comunicación entre los dos equipos. El mantenimiento del SNA en ciertas organizaciones es obligatorio, por sus programas que es imposible de adquirir nuevas tecnologías o el impacto en grande al cambiar sus programas. Esto obliga que las organizaciones actualicen sus plataformas de comunicación con el protocolo hacia TCP/IP (u otras nuevas tecnologías que lo soporten) y no sus programas, colocando en los datos bajo TCP/IP, el formato de SNA y los datos de los programas, es decir, encapsula la información de SNA en el protocolo de TCP/IP. Para la organización es adquirir equipos y software para el manejo de SNA con TCP/IP para que los programas no sean modificados.

La otra área de estudio en las telecomunicaciones es la utilización de la infraestructura telefónica en el manejo de imágenes, voz y datos. La mayoría de la plataforma telefónica esta construida bajo tecnología analógica. Con el uso de las tecnologías digitales actuales y la necesidad de reutilizar la tecnología analógica se hace un trabajo de investigación constante sobre el uso eficiente de este recurso, paralelamente el ingreso de tecnología digital en el área telefónica y el uso de ambos, abarcan extensas áreas de estudios en las telecomunicaciones. Del mismo modo, el uso de la tecnología digital en la simulación de la tecnología analógica para el mantenimiento de comunicaciones anteriores, que muchas organizaciones todavía mantienen, es decir, el uso de la tecnología digital para el mantenimiento de las redes análogas. En este trabajo, el aporte de la Ingeniería del Software es fundamental, en el manejo de la administración de las conexiones vía software, las transformaciones de señales digitales análogas por programas digitales, la parte de gestión en forma automatizada, ha tomado peso en el uso eficiente de los software en esta tecnología. Otro punto es la transparencia del uso del medio de comunicación en el sentido que la utilización del medio de transporte de paquetes sea para los desarrolladores de software independiente en el uso de estos componentes, esto es debido al uso de los estándares del modelo ISO de las siete diferentes capas de protocolo de comunicación.

Se detallan diferentes protocolos de comunicación hoy en día, como XML, SOAP, etc. y abran mas protocolos que no son familiares para el desarrollador, mas lo que se están desarrollando como nueva tecnología. La mayoría de este tipo de protocolo se maneja a nivel de capa de aplicación. Su base, se comunica con TCP/IP como medio de transporte. Es importante tener en cuenta el uso de TCP/IP en la tecnología actual y en las nuevas tecnologías.

El desarrollador al manejar las nuevas tecnologías en comunicación, tiende a disminuir su complejidad a nivel de abstracciones de programación como por ejemplo, el uso de librerías públicas de la tecnología para realizar conexiones TCP/IP. Caso específico es el uso de estas librerías de comunicación de TCP/IP en el laboratorio de la materia de Ingeniería del Software de la UCAB, aunque los estudiantes no tienen las habilidades y

4. Aplicabilidad de las nuevas tecnologías

conocimientos de comunicación en este caso de TCP/IP, el uso de esta librería, enfocado como un componente de uso en el proyecto y práctica de manejo y desarrollo de librerías, ha tenido gran éxito en los talleres de esta materia; por un lado, demuestra el fácil uso de esta tecnología sin conocimiento previo de comunicación (uso de una la nueva tecnología como caja negra), e incorporación de esta tecnología en los proyectos sin gran impacto. Esto es un ejemplo típico de utilizar componente nuevo en un proyecto.

Existen el uso de comunicación en forma implícita, como por ejemplos, las conexiones de DCOM, ODBC, OLE DB, y CORBA para el manejo de bases de datos y control de los proyectos. Estas conexiones se basan en el manejo de control y base de datos bajo estándares de manejo de un lenguaje definido por las empresas desarrolladoras de tecnologías. Aún teniendo diversidad de bases de datos como MS SQL, ORACLE, DB2, SYBASE, etc. en el mercado tecnológico, todas trabajan bajo la misma filosofía de comunicación y de aplicación. Existe el lenguaje PL/SQL (Procedural Language / Structured Query Language) que es un lenguaje flexible utilizado para acceder y manipular Base de datos relacionales (Urman, S. 2002, pag.3-6) que admite estándares ANSI (American National Standards Institute). En esta tecnología en la parte de comunicación, permite conexiones transparentes a otros equipos o servidores de base de datos sin que el desarrollador sepa el tipo de conexiones existentes, ya sea por la misma filosofía de la tecnología o seguir un estándar para la conexión para conectar otras tecnologías. El desarrollador maneja las bases de datos e implícitamente maneja la parte de comunicación.

Existen conexiones que no son tan directas o fácil de realizar, debido a la incompatibilidad de tecnologías, en este caso los especialitas en comunicación y en sus tecnologías entran en el concepto de Middleware, que es un software que administra, comunica y que intercambia datos, que se sitúa a la mitad de los diversos componentes. Todo lo descrito en la comunicación implícita en el párrafo anterior es un Middleware o conjunto de Middleware, que es ofrecido por la nueva tecnología. En el caso que no existe el Middleware que facilitad la comunicación entre las tecnologías, la nueva tecnología debe desarrollar este software. Existe diversos casos del uso de Middleware en todas las tecnologías actuales, pero para el desarrollo y uso de este software, se justifica en los casos de sistemas heredados (Sommerville, I., 2002, pag. 581-598).

En resumen, los casos que se realicen comunicaciones directas (utilizando las librerías de conexión) o indirectas (OLE DB, ODBC, CORBA, etc.). El desarrollador lo maneja como componentes que se integra en sus proyectos. En caso contrario hay que desarrollar esas conexiones con la nueva tecnología. Adicionalmente, esta tecnología debe tener la facilidad de desarrollar estas conexiones.

4.2.5. Entornos de la Ingeniería del Software en la nueva tecnología

Los PSEE se basan en los "Entornos de la Ingeniería del Software" (SEE o EIS) donde son soportes de las actividades humanas en el desarrollo del ciclo de vida del software. El EIS es un conjunto de herramientas que permite el soporte (capítulo 2.3) a estas actividades. Un EIS maneja las siguientes informaciones:

4. Aplicabilidad de las nuevas tecnologías

- a) El software en desarrollo o mantenimiento: colocar las nuevas tecnologías, significa incluir nuevos componentes, nuevas especificaciones técnicas o nuevos códigos fuentes de la nueva tecnología, dependiendo del nivel del ciclo de vida del software.
- b) Los recursos del proyecto: dependiendo del nivel de desarrollo del software se replantean los costos o se considera en el desarrollo del software en la inclusión de partida para la nueva tecnología.
- c) Los aspectos organizacionales: se debe medir el nivel de impacto de cambio y de riesgos de la nueva tecnología.

La aplicabilidad de las nuevas tecnologías conllevan en descubrir en los procesos actuales, la automatización de procesos no cubiertos por la tecnología actual, el redefinir en forma mas eficiente los procesos automatizados, llevar un proceso de control de calidad, los procesos de verificación y validación, minimizar los riesgos e impacto, y la evaluación de los costos.

En esta labor se debe cuidar los aspectos de Integración de herramientas y la orientación de procesos (capítulo 2.3) de la nueva tecnología.

4.2.6. Uso de Lenguajes de modelado de procesos

En este capítulo se determina el uso que podemos darle al lenguaje de modelado de procesos en las nuevas tecnologías. En el estudio realizado en (capítulo 2.6) "Lenguajes de modelado de procesos" existen propiedades deseables en los LMP, en las nuevas tecnologías al poseer esta propiedad, dependiendo de la madurez y la riqueza de la nueva tecnología, las 6 clases de elementos de proceso deben estar incluidas en que pueda modelar un LMP con tecnología propuesta. Existen tres grandes categorías según Ruiz G., F. (2003, pag. 31-38):

- Formales: Tienen sintaxis y semánticas formales, proveen soporte para la verificación y análisis formales, simulación y ejecución. Un ejemplo tipo es MVP-L "Multi View Process Modeling Language" basado en reglas formales desarrollado por la Universidad de Maryland (USA) y Kaiserslautern (Alemania).
- Semiformales: Tiene notación formal pero no tiene semántica formal y no son ejecutables. Un ejemplo es el UML (Lenguaje Unificado de Modelado) lenguaje orientado a objetos para modelar PS (Rumbaugh et al. 1999).
- Informales: sin sintaxis ni semánticas formales y no es ejecutables. Ejemplo el lenguaje natural.

Los requerimientos de los usuarios se pueden redactar en lenguaje natural puesto que debe ser comprendidos por personas no técnicas, mientras que a nivel de sistema los requerimientos son más técnicos. Estas técnicas de usar lenguaje modelado de procesos permite la integración de las visiones tanto técnica como la no técnica. Los modelos se utilizan en el proceso de análisis para desarrollar una comprensión del sistema existente a remplazar o a mejorar, o para especificar el sistema requerido (Sommerville, I., 2002, pag. 149).

4. Aplicabilidad de las nuevas tecnologías

En esta actividad de modelar los procesos, permite el entendimiento de los requerimientos y procesos que se deben desarrollar e implementar en el software, de tal forma que usando la nueva tecnología se verifica que tan susceptible o fácil es implementar con respecto a la tecnología actual y la riqueza de herramientas que permite en algunas tecnologías de pasar de un modelado a otro.

De este subcapítulo terminamos el estudio de lo que nos enfrentamos en estimar la nueva tecnología. En los dos subcapítulos siguientes se describe el proceso de mantenimiento y el proceso de desarrollo en el uso de la nueva tecnología, basándonos en "Características de procesos del software" (subcapítulo 2.2.1.), donde se determina que el proceso de producción consta de estos dos subprocesos.

4.2.7. Mantenimiento del Software utilizando la nueva tecnología

Se presenta en cuatro tipos de mantenimientos (capítulo 2.8.3. El Proceso de Mantenimiento: ISO 12207 y 14764) basados en ISO 14764:

- Corregir el problema detectado (Correctivo): se presentan errores que no son detectados en las pruebas y control de calidad del software. Generalmente, las correcciones son realizadas en la misma tecnología en que fue desarrollado el software. Si se presenta que el software no posee solución con la tecnología usada en el desarrollo, se debe estudiar las nuevas tecnologías que tenga la capacidad de solucionar el problema. Si existen las dos opciones (solución con tecnología actual y la nueva tecnología), es un punto de evaluación entre ambas tecnologías y medir el menor esfuerzo, riesgo e impacto del cambio.
- Mejorar la funcionalidad, o el rendimiento, mantenibilidad u otros atributos del software (Perfectivo): el mejoramiento de la funcionalidad y rendimiento, dependerá de gran modo de las características de la nueva tecnología. Generalmente, las nuevas tecnologías involucran las características de mejoramiento de funcionalidad, rendimiento y/u otras condiciones que la tecnología actual no posee. En caso contrario, no vale la pena el cambio de tecnología.
- Hacer que siga siendo utilizable en un entorno nuevo o cambiado (Adaptativo): el realizar cambios puntuales a la nueva tecnología, permite a la larga el cambio total del software o que el cambio total no sea traumático. El cambio a nueva tecnología permite otra visión de cambio del entorno del software, en donde si es factible y funciona correctamente, se alarga mas el ciclo de vida del software.
- Detectar y corregir errores latentes en el software antes de que se conviertan en fallos reales (Preventivo): cambios en puntos críticos (estructural) del software, permite corregir a largo y mediano plazo errores latentes. Permite enfocar en el futuro del uso de la nueva tecnología para fallas reales de la tecnología actual.

4.2.7.1. Actividades y tareas usando la nueva tecnología

Siguiendo las actividades definidas en (subcapítulo 2.8.3.1.) “Actividades y tareas”, la ejecución de estas actividades y tareas se deben enfocar en ingresar una nueva tecnología en el ciclo de vida de mantenimiento de un software. Se debe determinar en gran parte del estudio e implementación en la “Reusabilidad de la nueva tecnología” (subcapítulo 4.2.3), en que tamaño del cambio se debe hacer y que componente vamos a reutilizar de la nueva tecnología. El tamaño de la unidad a cambiar o a mejorar indicado en la “Reusabilidad por unidad de tamaño de las nuevas tecnologías” (subcapítulo 4.2.3.1.) es importante, pues, gran parte de los cambios en los mantenimientos se basan en las funciones y componentes, y muy poco a nivel del sistema por completo. De alguna forma el estudio de tamaño del impacto se mide por el tamaño del cambio. Por otro lado, el impacto y riesgo que se debe medir es en los artefactos reutilizables indicado en el subcapítulo 4.2.3.2., el de mayor impacto o riesgos es el cambio a nivel de Arquitectura, modelos de requisitos y diseños que las modificaciones de los códigos fuentes. De una u otra forma todos los artefactos de reutilización afectan de gran o pequeña medida en el uso de la nueva tecnología. El estudio del tipo de comunicación de la nueva tecnología se debe evaluar, en el sentido que mientras el ambiente de la nueva tecnología esta separado del ambiente de la tecnología actual, el impacto o riesgo es menor, y mientras que la comunicación de la nueva tecnología esta en el mismo ambiente o hardware del cambio el riesgo es mayor, debido a que componentes o artefactos reutilizables son instalados en el ambiente de la tecnología actual y puede afectarlo para mejorar o degradarlo. Hasta este punto medimos el tamaño del cambio.

Si evaluamos la complejidad de la nueva tecnología debemos estudiar la arquitectura de las nuevas tecnologías (subcapítulo 4.2.2), en este estudio se debe ver las ventajas y desventajas de la arquitectura de la nueva tecnología, verificar las ventajas que se ofrecen en la nueva tecnología, y validar las desventajas de ésta. La validación de las desventajas en la nueva tecnología, puede subsanarlas con ciertos mecanismos creados por esa tecnología, dependiendo del nivel de maduración de esta tecnología, esto se debe a que las tecnologías mejoran de la misma forma que el software mejore en uso de estas nuevas tecnologías. Adicionalmente, es importante el estudio de la arquitectura de la tecnología actual, donde se aplica el mismo concepto de mejoramiento de la tecnología, las empresas o instituciones de estas tecnologías (nuevas y actuales) desarrollan versiones actualizadas donde vienen estos cambios. Evaluar el ingreso de una nueva tecnología, implica evaluar la implantada en la organización, debido a que puede suceder que ambas tecnologías en sus últimas versiones estén en el mismo estado, de lo cual se evalúa el costo y riesgo de la adquisición de una nueva tecnología y la nueva versión de la tecnología actual.

Tomando en consideración los puntos anteriores, el mantenimiento con el uso de la nueva tecnología se debe considerar en realizar prototipo (si es factible y que sea corto en el tiempo) para cambios menores (nivel funciones o componentes). Nos basamos en el uso de prototipo que describe y aconseja Pressman (1998, pag. 24) en las interfaces al usuario. En la nueva tecnología, adicional de realizar este prototipo a nivel de presentación, debemos realizar también prototipo a nivel de integración, estos prototipos se encuentran definidos en el subcapítulo 2.3.1. de “Integración” como: prototipo de datos, prototipo de desarrollo a baja escala, prototipo de control, prototipo de procesos, Prototipo del marco de trabajo.

4. Aplicabilidad de las nuevas tecnologías

Estos prototipos podemos usarlos tanto en las actividades y tareas de baja escalas indicada en el subcapítulo 2.8.3.1. De la evaluación de todos los prototipos se debe generar un nivel macro de tareas y actividades englobando los resultados de todos los prototipos, repitiendo la misma actividad, involucrando todos los puntos de la integración. En este sentido se aplica el concepto del modelo incremental dicho en los capítulos anteriores, basándose en los prototipos refinados (lo recomendable es no usar los prototipos de pruebas anteriores, sino a nivel de refinamiento en el resultado de la evaluación). Estas actividades y tareas deben estar alineadas con el proceso de gestión del proceso de mantenimiento (subcapítulo 4.1.2.1.).

Para la realización de estos prototipos hay que tener al alcance esta nueva tecnología, para esto existe en el mercado versiones de evaluación que duran cierto periodo de tiempo, el uso de estas versiones es recomendable en todo lo casos, de tener un ambiente aparte para las pruebas y tener respaldo de la tecnología actual. De esta forma tiene el mecanismo de evaluar ambas tecnologías en forma separada y aislada. Por otro lado, tiene el mecanismo de restablecer el ambiente original sin impacto ni riesgo.

En el caso de no tener el alcance de esta tecnología o que al implementar la nueva tecnología de gran escala o de gran tamaño, existe forma de evaluar ambas tecnologías en forma conceptual, existe en el mercado información de caso de uso en otras organizaciones, resultados y tendencias de esa tecnología. Se desarrollan las pruebas de conceptos o casos de pruebas si es de gran tamaño. Desde otra óptica, si el tamaño y la complejidad de la nueva tecnología posee gran tamaño, es el límite de considerarlo como un modelo de desarrollo de software o como un proyecto nuevo, tomando en consideración el concepto de la reutilización en la tecnología actual y no en la nueva tecnología. Para esto se desarrollan los siguientes subcapítulos.

4.2.8. Desarrollo de Software utilizando la nueva tecnología

El enfoque de este subcapítulo se desarrolla en el proceso de producción de un nuevo software con una nueva tecnología. El modelo a escoger en el desarrollo depende mucho de la característica de las variables de incertidumbre en el proyecto. El trabajo de la Ingeniería del software es minimizar estos impactos y producir menos riesgos. En la práctica es buscar de disminuir todas las variables de incertidumbres a la mínima expresión y tomar alternativas y controles para el manejo de estas variables. La gestión que se debe aplicar se estudio en los subcapítulos anteriores (subcapítulos 4.1.1. y 4.1.2.) con las incorporación de una nueva tecnología. El estudio se centrará en desarrollar un software con una variable de incertidumbre adicional que es la nueva tecnología.

Las actividades son parecidas al mantenimiento del software pero manejado a gran escala, es decir, el desarrollo normal de un software y el trabajo adicional de estar pendiente de la nueva tecnología. En este caso nos enfocamos en las actividades y tareas definidas con el Modelo CMMI en el desarrollo del software.

En el CMMI los puntos de desarrollo del software con nuevas tecnologías esta localizado en el concepto Ingeniería del Software (subcapítulo 3.1.3.2.), esto contiene la gestión de procesos, gestión de proyectos, soporte y área de Ingeniería de proceso.

4.2.8.1. Ingeniería utilizando la nueva tecnología

En el subcapítulo 3.4.3 “Ingeniería” donde están identificadas las áreas de procesos del CMMI, se debe cumplir y realizar lo descrito en cada de las áreas nombradas, se agrega en cada punto las observaciones en el uso de la nueva tecnología, y son:

- Gestión de requerimientos (subcapítulo 3.4.3.1.): el estudio de los requerimientos para el proyecto y los componentes del producto. Se estudian en los componentes del producto y se asignan al estudio de los componentes de la nueva tecnología y de la arquitectura de ésta (subcapítulo 4.2.2.). El entendimiento abarca los requerimientos y requerimientos de los componentes de la nueva tecnología. Se determinan las ventajas y desventajas reales dependiendo del modelo de la tecnología. Identificar inconsistencia entre el proyecto y los requerimientos.
- Desarrollo de requerimientos (subcapítulo 3.4.3.2.): Se aplica lo indicado en el desarrollo de requerimientos del producto, los componentes de Interfaces y que cubren los requerimientos con la nueva tecnología debe estar identificado, conceptualizar escenarios tanto en el proyecto y en la arquitectura de esta tecnología. Establece la definición con la funcionalidad con los componentes. Analizar y validar el requerimiento. Por ejemplo: La validación puede identificar que los componentes de interfaces y de base de datos trabajen correctamente, tanto individualmente como en su conjunto, debido a que se puede complicar al juntar los dos componentes.
- Solución técnica (subcapítulo 3.4.3.3.): Se diseña producto-componente con la nueva tecnología. Desarrollo detallado de soluciones de las alternativas y selección de soluciones Productos-Componentes en el ambiente de la tecnología. Debe estar bien definida la arquitectura y partición estructural de esta tecnología. Identificar bien los componentes o fuentes de reusos. La compatibilidad de las nuevas versiones en el futuro. El método de codificar de la nueva tecnología esta identificado (programación estructurada, programación Orientada a Objeto, diseño de patrones, Generación de código en forma automática, etc.).
- Integración del Producto: Se evalúa en la práctica la integración y ensamblaje de los componentes de esa tecnología y que tan rápido se hace. Mide la complejidad de uso de la nueva tecnología. Se evalúan las interfaces de comunicación entre los componentes y su cohesión. Se detalla el método de entrega y empaquetado del producto.
- Verificación: Selección de los componentes de la nueva tecnología en la que cumplan con los requisitos de requerimientos y los requerimientos técnicos. Se usa prueba en diferentes caminos; prueba de estrés, carga y capacidad; pruebas basadas en tablas de decisión; prueba de descomposición funcional; prueba de reuso; prueba de aceptación, etc.

4. Aplicabilidad de las nuevas tecnologías

- Validación: Demuestra que la nueva tecnología funciona en el ambiente que refleje los requisitos especificados tanto de las necesidades técnicas como organizacional.

4.2.8.2. Soporte utilizando la nueva tecnología

Este capítulo se basa en el (subcapítulo 3.4.4.) “Soporte”, donde se indica en cada área las observaciones en la nueva tecnología:

- Gestión de configuración. Prueba de la integridad de configuración bajo la nueva tecnología. Por ejemplo: prueba de módulos y códigos, compiladores.
- Proceso y garantía de la calidad del producto. Personal y gerencia bajo la nueva tecnología.
- Medición y Análisis. Facilidad y riqueza de herramientas en la nueva tecnología que permiten a la gerencia recolectar y analizar los resultados.
- Análisis de Decisión y resolución. Herramientas para la evaluación del problema y presentación de alternativas de soluciones.
- Ambiente Organizacional para la Integración. Herramientas de distribución de información.
- Análisis causal y Resolución. Herramientas de ayuda de solución de errores y captura de defectos de la nueva tecnología.

4.2.9. Prototipos

Se ha nombrado prototipos en algunas actividades y tareas en los métodos de desarrollo y mantenimiento usándolo en la nueva tecnología. Sommerville (2002, pag. 171-174) describe el apoyo de las actividades en el procesos de ingeniería en los requerimientos, tanto en la obtención de requerimientos y validación de los requerimientos. Se deriva varias ventajas como:

- 1) Demostrar las funciones del sistema donde se identifica las discrepancias entre los desarrolladores y usuarios.
- 2) El desarrollador puede encontrar requerimientos inconsistentes y/o incompletos.
- 3) Dispone rápidamente de un sistema que funciona y demuestra la factibilidad y usabilidad de la aplicación.
- 4) Ayuda a escribir las especificaciones de producción de un sistema de calidad.

En el caso de la nueva tecnología, el uso de prototipo posee las ventajas indicadas anteriormente más la siguientes:

4. Aplicabilidad de las nuevas tecnologías

- 1) Demostrar las discrepancias y semejanzas entre la tecnología de la organización y la nueva tecnología.
- 2) Demuestra la completitud o no de la nueva tecnología con los requerimientos de la organización.
- 3) Demuestra una funcionalidad rápida de la nueva tecnología. Permite también evaluar el nivel de rapidez de desarrollo con esta tecnología y compararlo con la actual.
- 4) Determina las vías para el proceso de calidad de esta tecnología.

En la construcción de prototipos se diferencia por dos objetivos (según Sommerville Sommerville, I. 2002, pag. 174-180): Construcción de prototipos desechables y prototipos evolutivos.

- 1) Prototipos desechables.- el objetivo es validar y derivar los requerimientos del sistemas.
- 2) Prototipos evolutivos.- se entrega a usuarios finales en un sistema funcional.

El último tipo de prototipo esta en contradicción de las recomendaciones de Pressman (1998, pag. 24-25), donde determina que todos los prototipos se deben desechar para evitar llevar los errores al sistema final.

Esta discrepancia se debe llevar a un acuerdo en la práctica, donde se propone la definición de ambos prototipos, pero con limitantes en los prototipos evolutivos. Estas limitantes se apoya en la misma práctica de Ingeniería del Software. Existen varios problemas de desarrollo de prototipos evolutivos como lo menciona Sommerville (2002, Pag. 177) en que este desarrollo tiene un problema de administración, mantenimiento y contractuales.

El problema de administración que plantea Sommerville es que al crear prototipos evolutivos y desarrollar cambios en un sistema grande, se generan gran cantidad de documentación. El desarrollo rápido usa herramientas no tan familiares. Para el caso nos basamos en el punto M de equilibrio de modularidad propuesta por Pressman (1998, Pag. 233-235), donde la cantidad de módulos deben estar acorde al esfuerzo de interconexión de los módulos, por otro lado, el tener un prototipo de una gran cantidad de código no es lo aconsejable. En conclusión, este trabajo propone que el prototipo evolutivo sea válido cuando tenga un tamaño esperado de código y funcionalidad limitada, que permita el fácil control y mantenimiento, se propone como un prototipo de un solo componente. Al crecer este prototipo se debe generar otro prototipo con funcionalidad diferente (partición de funcionalidad) y como otro componente, crear un tercer componente que controle a las dos anteriores. En vista del sistema, el componente de mayor importancia es el de mayor jerarquía de los tres, si el número de componentes crece de forma que es imposible de mantener, este conjunto de componentes se convierte en un subsistema y no como componentes del sistema actual, respetando el nivel de acoplamiento bajo con alta cohesión (por esta razón se debe convertir el prototipo a un componente). Al ser un prototipo

4. Aplicabilidad de las nuevas tecnologías

evolutivo y al ser utilizado en el sistema final, debe tener este un nivel de cohesión alta, para controlar los errores y no propagarlo al sistema completo.

Los problemas de mantenimiento que plantea Sommerville que los cambios continuos tienden a corromper la estructura y que nadie, excepto los desarrolladores originales, pueden entenderla. La tecnología se vuelve obsoleta por el desarrollo rápido de prototipo. Se propone en este trabajo, que los prototipos no se realicen nuevos cambios, de alguna forma este cumple un requerimiento actual. Si el prototipo es modular y si el requerimiento cambia completamente, la mejor decisión es desecharlo y construir uno nuevo con la nueva tecnología, si solo cambian ciertas particularidades del requerimiento, el componente debe mantenerse y se debe desarrollar nuevos componentes o prototipos con la nueva tecnología con la reutilización de los componentes de la tecnología anterior.

Los problemas contractuales, según Sommerville, basados en las especificaciones del sistema entre el desarrollador y el usuario, al no existir estas especificaciones bien definida debilita el funcionamiento del sistema y sobrepasan el presupuesto. Los desarrolladores no aceptan un contrato con precio fijo y no pueden controlar los cambios requeridos. La propuesta de este trabajo, se basa en los casos anteriores, es decir, no hacer cambios en lo que esta hecho o que funcione, los nuevos requerimientos se desarrolla en un prototipo evolutivo o desechables, en el cual si el resultado es satisfactorio, el prototipo es candidato a un componente del sistema, considerando que este componente debe ser alta cohesión y bajo acoplamiento.

El prototipo se basa en cualquier componente de la tecnología como lo indicamos en la reusabilidad (subcapítulo 4.2.3), en la arquitectura (subcapítulo 4.2.2.) y en la comunicación de la nueva tecnología (subcapítulo 4.2.4.). Una tarea al usar los prototipos son las pruebas de integridad, debido a que dependiendo de la arquitectura este componente o prototipo afecte al resto del sistema.

4.2.10. Selección, generación de alternativa y tomas de decisiones

En los diferentes métodos de CMMI, Entornos de MANTIS y los procesos de aplicar la nueva tecnología, no se describe la toma de decisiones en selección ni la generación de alternativas. Se propone el uso del árbol de decisión, tablas de decisión, matriz de pago y diagrama de flujo (Davis y Olson 1989) definidos como herramientas en el área de Ingeniería del Software para representar las diferentes alternativas y el entorno del mundo completo que puede representar el sistema. Estas herramientas describen como pasar de la representación gráfica a algoritmo y optimización de código. El proceso de desarrollo de método de documentación basado en estas herramientas, permite, a parte de optimizar código de programación, el desarrollar de las alternativas que se presentan en ese entorno. Al definir las variables del mundo de entorno que afecta los resultados, permite ampliar condiciones no esperadas antes del diseño e implementación del software, creando nuevas alternativas de acción no contemplado desde su inicio. El proceso involucra intrínsecamente la evaluación de las variables de los entornos del mundo real que afecta el sistema, por ejemplo: si las variables no afectan las condición de comportamiento y estado del sistema, estas variables se debe desechar debido a que no influye en la toma de decisión

4. Aplicabilidad de las nuevas tecnologías

dentro del sistema, se debe estudiar de nuevo el proceso y recolectar nuevas variables, del caso contrario, se debe reducir la cantidad de variables a su mínima expresión, esto ayuda al acoplamiento bajo entre componentes y ayuda en generación de módulos mas controlables.

Dependiendo de las condiciones del entorno externo e interno del proyecto, el estudio de toma de decisiones con estas herramientas, no son los mas adecuados, en estos casos se direcciona a otras áreas del conocimiento como se describe:

1. Técnica de optimización: decisiones bajo certeza. Se utiliza Sistemas de ecuaciones, Programación Lineal, Programación Entera, Programación Dinámica, Modelo de Teorías de Colas, Métodos de Inventario, Análisis de Presupuesto Capital, Análisis de Punto de Equilibrio.
2. Matrices de Pago en la teoría de decisiones estadísticas. Todas las alternativas y resultados son conocidos y el objetivo maximizar beneficio o utilidades. Esta son: Matriz de Pago y árbol de decisión.
3. Curva de utilidad e indiferencia. Se utiliza el término UTIL unidad de medida en función de utilidad.
4. Clasificación, ponderación o eliminación de aspecto. Involucran un número de factores o aspectos, donde a cada aspecto se le asigna una importancia.
5. Teoría de juego. Si uno pierde el otro gana.
6. Inferencias estadísticas clásicas. Enfoque contrario al bayessiano y son: a) Muestreo: tomar muestra para determinar variancia, esperanza, etc.; b) Distribución de probabilidad: aproximación por distribución a una función conocida (chi-cuadrado, Poisson, etc.)

Las generaciones de alternativas en la práctica, se produce en el desarrollo de prototipos e inclusive afecta en las tomas de decisiones. El desarrollo de prototipos permite ayudar a investigar, detectar nuevas alternativas o nuevos problemas y tener el primer contacto con la nueva tecnología.

4.2.11. Tomas de decisiones en selección de una nueva tecnología

La organización en algún momento debe tomar la decisión sobre seleccionar una tecnología en un conjunto de tecnología. Esta área es nueva en base a la tecnología de Información. Todavía no existe una forma general de evaluar y comparar varias tecnologías y escoger una de ellas. En todo este trabajo, estudiamos como incluimos una nueva tecnología en el mantenimiento y desarrollo de un nuevo software, pero no detallamos en la selección de esta. El estudio de esta área merece un trabajo mas completo que no puede ser desarrollado en este trabajo. Pero, se menciona como actividad inicial o propuesto para un nuevo trabajo de investigación. Al realizar este estudio de impacto y riesgo de la nueva tecnología, vemos a groso modo puntos básicos que debemos tomar en cuenta en la selección.

4. Aplicabilidad de las nuevas tecnologías

Existen actividades básicas que se plantean en el anexo del material de auditoría de Sistema, en el tema "Introducción a los controles internos" en el Grupo Alarcos (2003-2004) donde plantea:

Objetivos de controles:

1. Tecnología de la información como parte del Plan a largo y corto plazo de la Organización.
2. Plan de la tecnología de la Información a largo plazo.
3. Planificación de la tecnología de la Información a largo plazo - Enfoque y Estructura.
4. Cambios en el Plan a largo plazo de la tecnología de la información.
5. Planificación a corto plazo de la Tecnología de la Información.
6. Evaluación de los sistemas existentes.

Objetivos de controles detallados:

- Obtención de conocimiento e Información.
- Evaluación de controles.
- Evaluación de suficiencia.
- Evaluación del riesgo que no se cumplan los objetivos de controles.

Vale destacar que en cada objetivo de controles se debe aplicar los Objetivos de controles detallados.

Otros autores, como De Nardi di et al (2000) describe colocar cada componente en relación a la vida útil y costo/beneficio que optimiza al sistema, adicionalmente, las características técnicas y costos de cada componentes de los diferentes proveedores, suministradores y/o fabricantes. Luego se pondera con un método definido y se selecciona la mejor opción. En el trabajo no se define ni describe el método de ponderación.

Sommerville (2002, pag. 37-38) estudia en forma general, en el estudio de "Adquisición del sistema", esto nos ayuda en la evaluación en la nueva tecnología en la parte computacional. Donde plantea en forma paralela e integrada las actividades de obtener requerimiento de usuario/organización y el estudio de los sistemas actuales disponibles en el mercado. Una nueva tecnología no solo abarca la parte computacional sino proceso de gestión y herramientas que faciliten la gestión.

Gartner (2003) plantea un método para la selección de una nueva tecnología, el método RHA (Refined Hierarchical Analysis) o Método Jerárquico Refinado de Análisis, que posee prácticas para adquisición de tecnologías para las necesidades de largos y cortos plazos de una empresa.

Existen actualmente muchos mas modelos de evaluar una nueva tecnología, pero todos estos modelos están basados en el estudio de minimizar el impacto de la nueva tecnología.

5. Conclusiones

La gestión y la producción del software en el uso o ingreso de la nueva tecnología se define, principalmente, en el alcance del proyecto tanto a nivel organizacional y de tecnología computacional como en el estudio de la nueva tecnología, basado en la magnitud y complejidad de ambos. Las actividades y tareas que se involucran en el proceso de gestión y producción deben tener en consideración diferentes temas que abarquen la Ingeniería del Software como la modularidad, reutilización, comunicación, arquitectura, uso de lenguaje modelado de procesos, herramientas de tomas de decisiones, modelo de desarrollo.

La nueva tecnología puede abarcar los procesos de gestión y/o de producción. El proceso de gestión deriva en actividades y tareas que aseguran la calidad del producto, la optimización de procesos, las medidas de contingencia, la gestión de medición y el control, la gestión de configuración, la gestión de desarrollo y el mantenimiento del software. La gestión de producción y mantenimiento del software es afectado directamente por la gestión organizacional y viceversa. En el proceso de producción se ejecutan los planes definidos en las áreas de gestión apoyándose de la Ingeniería del Software.

La nueva tecnología se puede medir, en la parte computacional, estudiando la analogía en la arquitectura de sistema, evaluando los impactos y riesgos que pueden producir esa arquitectura, evaluando las ventajas y desventajas de esa tecnología. La selección de una nueva tecnología del mercado abarca un estudio mayor o nuevos trabajos de investigación.

Los modelos CMMI y Entornos de MANTIS son desarrollados para dos condiciones y realidades diferentes; CMMI se aplica en el desarrollo, optimización de procesos organizacionales y se involucra en el desarrollo de productos altamente de calidad, aplicado en una organización grande; mientras que los entornos de MANTIS se enfoca en el mantenimiento del software. A nivel de ciclo de vida del software el CMMI se involucra en todo este proceso, pero para los entornos de MANTIS solo se involucra en el ciclo de vida pero a nuevos requerimientos en el mantenimiento. En ambos modelos y en cualquier modelo de desarrollo de software, el uso de la nueva tecnología impacta e incrementa el riesgo si no se aplican controles y actividades para la disminución de estos. El uso de prototipos permite la disminución de riesgos, pero es un arma de doble filo sino se aplican correctamente, cada prototipo debería convertirse en un componente del sistema total, en forma modular de alta cohesión y bajo acoplamiento, si los cambios de requerimientos son muchos o son muchos los requerimientos no definidos, los prototipos son de gran ayuda, intentando evitar realizar cambios a los ya existentes, sino, desarrollando nuevos prototipos y acoplándolo a las ya existente. Si existen demasiados componentes en los cambios, es punto de particionar las funcionalidades y reorganizar en subsistemas con estos componentes, usando un medio de comunicación de acoplamiento bajo y que no impacte a todo el sistema. El nivel de acoplamiento y de cohesión de una tecnología influye en el concepto de Integración o Integrado de esta, si la tecnología esta bien diseñada, los efectos de errores deben ser controlados por estas, en su concepto base y sus implementaciones, gran parte del trabajo del personal de tecnología de computación

5. Conclusiones

debe vigilar, conservar y procesar esta filosofía en todas las etapas de ciclo de vida del proyecto. El personal de tecnología y la organización debe equilibrar la Integración de sus productos, esta entre realizar un cambio con mínimo esfuerzo y que abarque a todos los componentes de la tecnología, y el manejo y gestión de los errores en disminuir el impacto de los cambios.

Las apariciones de nuevas tecnologías intenta abarcar el desarrollo rápido de Software de calidad y de bajo costo, por medio de desarrollo de nuevos protocolos de aplicación y comunicación entre aplicativo, como herramientas CASE para el análisis y diseño de los productos, y todo un conjunto de componentes que ayuda a controlar y gestionar el proyecto. Pero todas estas tecnologías no valen nada sin las actividades de gestiones de controles por parte del personal que lo involucra y sin la aplicación de los conceptos básicos de la Ingeniería del Software.

Apéndices.**6.1. Lista de Acrónimos.**

4GL Lenguaje de Cuarta Generación
 ACID Atomic, Consistent, Isolation, and Durable
 ACL *Agent Communication Language*
 ACM *Association for Computing Machinery*
 AIAA *American Institute of Aeronautics and Astronautics*
 ANS Acuerdo de Nivel de Servicio
 ANSI *American National Standards Institute*
 API *Application Programming Interface*
 APSE *Ada Programming Support Environment*
 APSEC *Asia Pacific Software Engineering Conference*
 APPN *Advanced Peer To Peer Networking*
 ARIS *Architecture of Integrated Information Systems*
 ASCII *American Standard Code for Information Interchange*
 ASE *Annals of Software Engineering*
 ATIS *A Tool Integration Standard*
 BPR *Business Process Reengineering*
 CAIS-A *Common APSE Interface Set*
 CAISE *Conference on Advanced Information Systems Engineering*
 CASE *Computer Aided Software Engineering*
 CCS *Calculus of Communicating Systems*
 CDIF *CASE Data Interchange Format*
 CENPRI Centro Provincial de Informática
 CICS *Customer Information Control System*
 CICYT Comisión Interministerial de Ciencia y Tecnología
 CM3 *Corrective Maintenance Maturity Model*
 CMM *Capability Maturity Model*
 CMMi *Capability Maturity Model Integration*
 CNS Contrato de Nivel de Servicio
 COCOMO *Constructive Cost Model*
 CORBA *Common Object Request Broker Architecture*
 COTS *Commercial Off-The-Shelf*
 CPU *Central Processing Unit*
 CRM *Customer Relations Management*
 CSCW *Computer Supported Colaborative Work*
 CSI Consejo Superior de Informática
 CSMR *European Conference on Software Maintenance and Reengineering*
 CSP *Communicating Sequential Processes*
 DOM *Document Object Model*
 DRAE Diccionario de la Real Academia Española de la Lengua
 DTD *Document Type Definition*
 EBNF *Extended Backus Naur Form*
 EC *European Comission*
 ECMA *European Computer Manufacturers Association*
 EIA *Electronic Industries Alliance*
 EIA/IS *Electronic Industries Alliance / Interim Standard*
 EIS Entornos de Ingeniería del Software
 ESA *European Space Agency*
 ESE *Empirical Software Engineering*
 ESEC *European Software Engineering Conference*
 ESP Entornos de Soporte a Proyectos
 EWSPT *European Workshop on Software Process Technology*
 FASE *International Conference on Fundamental Approaches to Software Engineering*

6.1. Lista de Acrónimos

FEDER Fondo Europeo de Desarrollo Regional
FIPA *Foundation for Intelligent Physical Agents*
FRISCO *Framework of Information Systems Concepts*
FT *Flujo de Trabajo*
GAD Grafo Acíclico Dirigido
GCR Grupo Crítico de Referencia
GCS Gestión de la Configuración Software
GDPA *Graphical Development Process Assistant*
GME *Generic Modeling Environment*
GQM *Goal-Question-Metric*
HIP *Human Information Processing*
HMM *Herramienta de Meta-Modelado*
HTML *HyperText Markup Language*
I+D Investigación y Desarrollo
IA-SI Investigación-Acción en Sistemas de Información
ICEIS *International Conference on Enterprise Information Systems*
ICSE *International Conference on Software Engineering*
ICSM *International Conference on Software Maintenance*
ICSR *International Conference on Software Reuse*
IDL *Interface Definition Language*
IEC *International Electrotechnical Commission*
IEEE *Institute of Electrical and Electronics Engineers*
IEEE-CS *IEEE – Computer Society*
IFIP *International Federation for Information Processing*
IPSE *Integrated Project Support Environment*
IPSEN *Integrated, Incremental, Interactive Project Support Environment*
IRDS *Information Resource Dictionary System*
IS Ingeniería del Software
ISACF *Information Systems Audit and Control Foundation*
ISEE *Integrated Software Engineering Environment*
ISESS *International Software Engineering Standards Symposium*
ISF *Integrated Software Factory*
ISO *International Standardization Organization*
ISPL *Information Services Procurement Library*
IT *Information Technologies*
ITM *Information Technology and Management*
ITS-CMM *IT Service Capability Maturity Model*
IWPC *International Workshop on Program Comprehension*
IWPSE *International Workshop on Principles of Software Evolution*
IWSM *International Workshop on Software Measurement*
JADE *Java Agent DEvelopment framework*
JSME *Journal of Software Maintenance and Evolution*
JSS *Journal of Systems and Software*
KLCF Kilo-Líneas de Código Fuente
KPA *Key Process Area (Área de Proceso Clave)*
LCF Líneas de Código Fuente
LMP Lenguaje de Modelado de Procesos
LNCS *Lecture Notes in Computer Science*
MDC *Meta-Data Coalition*
MOF *Meta-Object Facility*
MP Modelo de Proceso
MSI Mantenimiento de Sistemas de Información
MT-ECMA Marco de Trabajo ECMA
MUC Mantenimiento Correctivo Urgente

MVP-L *Multi View Process Modeling Language*
 MwR *Maintenance with Reuse*
 NASA *National Aeronautics and Space Administration*
 NATO *North Atlantic Treaty Organization*
 NIST *National Institute of Standards and Technology*
 OCD *Objetivos de Control Detallados*
 OCG *Objetivos de Control Generales*
 OCL *Object Constraint Language*
 OG *Objetivo General*
 OIM *Open Information Model*
 OMG *Object Management Group*
 OO *Orientado a Objetos*
 OP *Objetivo Parcial*
 PCIS *Portable Common Interface Set*
 PCTE *Portable Common Tool Environment*
 PF *Puntos Función*
 PIE *Process Instance Evolution*
 PIF *Process Interchange Format*
 PM *Petición de Modificación (o de Mantenimiento)*
 PMBOK *Project Management Body of Knowledge*
 PMI *Project Management Institute*
 PMS *Proceso de Mantenimiento del Software*
 PPD *Dependencias Producto/Proceso*
 PPD-EP *Paquete de Experiencia PPD*
 PPS *Programa del Proceso Software*
 PROFES *International Conference on Product Focused Software Process Improvement*
 PROMENADE *PROcess-oriented Modelling and ENactment of software Developments*
 PS *Proceso Software*
 PSEE *Process-centered Software Engineering Environment*
 PSL *Process Specification Language*
 PT *Producto de Trabajo*
 PYMES *Pequeñas y Medianas Empresas*
 Q-MOPP *Qualitative Evaluation of Maintenance Organizations, Processes and Products*
 RDF *Resource Description Framework*
 REFSENO *Representation Formalism for Software Engineering Ontologies*
 RPC *Remote Procedure Call*
 RUP *Proceso Unificado de desarrollo de software de Rational*
 SCI *Science Citation Index*
 SDL *Specification and Description Language*
 SDLC *Synchronous Data Link Control*
 SDS *Schema Definition Sets*
 SEE *Software Engineering Environment*
 SEN *ACM SIGSOFT Software Engineering Notes*
 SGBD *Sistema de Gestión de Bases de Datos*
 SGC *Sistema de Gestión de Conocimiento*
 SGFT *Sistema de Gestión de Flujos de Trabajo*
 SGO *Sistema de Gestión de Objetos*
 SI *Sistema de Información*
 SNA *Systems Network Architecture*
 SOFT *IEEE Software*
 SPEM *Software Process Engineering Metamodel Specification*
 SQA *Software Quality Assurance (Aseguramiento de la Calidad del Software)*
 SQL *Structured Query Language*
 STAR *Software Technology for Adaptable, Reliable Systems*

SWEBOK *Software Engineering Body of Knowledge*
TI *Tecnologías de la Información*
TIC *Tecnologías de la Información y Comunicaciones*
TPS *Tecnología de Proceso Software*
TSE *IEEE Transactions on Software Engineering*
UCLM *Universidad de Castilla -La Mancha*
UML *Unified Modeling Language* (Lenguaje Unificado de Modelado)
USAF *United States Air Force*
W3C *World Wide Web Consortium*
WfMC *Workflow Management Coalition*
WSDL *Web Services Description Language*
XMI *XML Metadata Interchange*
XML *Extensible Markup Language* (Lenguaje de Marcas Extensible)
XP *eXtreme Programming*
XPDL *XML Process Definition Language*

6.2. Referencias Bibliográficas.

- Acuña, S.T. y Ferré, X. (2001): Software Process Modelling. Proceedings of the *International Conference on Information Systems, Analysis and Synthesis (ISAS'2001)*, volume 1. Orlando, Florida (Estados Unidos), pp. 237-242.
- Arbaoui, S., Lonchamp, J., Montangero, C. (1999): The Human Dimension of the Software Process. En "*Software Process: Principles, Methodology and Technology*". LNCS 1500, Springer-Verlag, pp. 165-199.
- Balzer, R., Cheatham, T.E., y Green, C. (1983): Software Technology in the 1990's: Using a New Paradigm". *IEEE Computer*, 16(11), pp. 39-45.
- Benali, K. y Derniame, J.C. (1992): Software Process Modeling: What, Who and When. *2nd European Workshop on Software Process Technology (EWSPT'92)*. Trondheim (Noruega), LNCS 635, Springer-Verlag, pp. 21-25.
- Ben-Shaul, I.Z. y Kaiser, G.E. (1998): Federating Process-Centered Environments: the Oz Experience. *Automated Software Engineering*. 5(1), Kluwer Academic Publishers, pp. 97-132.
- Boehm, B.W. (1981): *Software Engineering Economics*. Prentice-Hall.
- Cabero A., J. (2002): "Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones educativas.", 15/06/2002. <http://tecnologiaedu.us.es/bibliovir/pdf/75.pdf>
- Conradi, R. y Jaccheri, M.L. (1999): Process Modelling Languages. En "*Software Process: Principles, Methodology and Technology*". LNCS 1500, Springer-Verlag, pp. 27-52.
- Curtis, B., Kellner, M. y Over, J. (1992): Process Modeling. *Communications of ACM*, 31(11), pp. 1268-1287.
- Davis y Olson (1989): "Sistemas de Información Gerencial". 2da. Edición. 1989.McGraw-Hill.USA.
- De Nardi, M., Dorgan, M., Graffigna, J. P. y Romo, R. (2000): "Estrategias para el diseño sistemático de un PACS Institucional". Universidad Nacional de San Juan. 2000. Disponible en: http://www.sis.org.ar/sis2000/pacs_institucional.pdf
- Derniame, J.C. et al (1994): Life-Cycle Process Support in PCIS, or It Is Time to Think about Software Process Formalisms Standardization. *Proceedings of the PCTE'94*, Technical Journal No.2, San Francisco, USA, November 1994.
- Dowson, M., Nejme, B. y Riddle, W. (1991): Fundamental Software Process Concepts. En Ambriola, V., Fugetta, A. y Conradi, R. (editors); *First European Workshop on Software Process Modeling*. AICA Press, pp. 16-37.
- Dowson, M. y Fernström, C. (1994): Towards Requirements for Enactment Mechanisms. *Third European Workshop on Software Process Technology (EWSPT'94)*. Villad-de-Lans, France, February 1994. LNCS 772, Springer-Verlag, pp. 90-106.
- Estublier, J., Cunin, P.Y. y Belkhatir, N. (1998): Architectures for Process Support System Interoperability. Proceedings of the *Fifth International Conference on the Software Process (ICSP'98)*, 15-17 Junio, Chicago (Estados Unidos), pp. 137-147.
- Feiler, P.H. y Humphrey, W.S. (1993): Software Process Development and Enactment: Concepts and

6.2. Referencias Bibliográficas.

Definitions. Proceedings of the Second *International Conference on the Software Process*. IEEE Computer Press, pp. 41-53.

Fenandez S., L. (2000): "El futuro de la Ingeniería del Software o ¿Cuándo será el software un producto de la ingeniería?". <http://www.ati.es/novatica/2000/145/luifer-145.pdf>.

Franch, X. y Ribó, J.M. (1999b): Some Reflexions in the Modelling of Software Processes. *International Process Technology Workshop (IPTW)*. 1-3 septiembre, Villars de Lans (Francia).

Garg, P.K. y Jazayeri, M. (1996): Process-centered Software Engineering Environments: A Grand Tour. En Fuggetta, A. y Wolf, A. (editores); *Software Process*. John Wiley & Sons.

Gartner (2003): "Developing Best Practices For Technology Selection". 24 Julio 2003. Disponible en: <http://www.gartnerweb.com/pages/story.php.id.9289.s.8.jsp>

Grupo Alarco (2003-2004): Anexo del curso en Auditoria y seguridad Informatica , el tema INTRODUCCION A LOS CONTROLES INTERNOS. 2003-2004. Grupo Alarcos de la Universidad de Castilla- La Mancha. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/Auditoria/AUDAnexoT3.pdf>

Hanrahan, R., Daud, C., Meiser, K. y Peterson, J. (1994): *Software Engineering Environment Technology Report*. USAF, Software Technology Support Center.

Heimbigner, D. (1993): A Revisionist Approach to Process Change. Proceedings of the Eight *International Software Process Workshop (ISPW)*. Schloss Dagstuhl (Alemania), 2-4 marzo, pp. 95-97.

Hojman, R. y Muñoz, B. (2001): "Centros de Recursos Educativos Integrales Compartidos: Globalización y Diversidad como un Complemento Ineludible de las Reformas Educativas". Pag. 13 Abril de 2001. <http://www.cnea.gov.ar/semi-inter-2001-trabajos/HOJMAN.pdf>

ISO/IEC (1995): *12207: Information Technology - Software Life Cycle Processes*.

ISO/IEC (1998a): *15271: Information Technology - Guide for ISO/IEC 12207 (Software Life Cycle Processes)*.

ISO/IEC (1998b): *15504-1: Information Technology - Software Process Assessment - Part 1: Concepts and Introductory Guide*.

ISO/IEC (1998c): *15504-2: Information Technology - Software Process Assessment - Part 2: A Reference Model for Processes and Process Capability*.

ISO/IEC (1998d): *15504-9: Information Technology - Software Process Assessment - Part 9: Vocabulary*.

ISO/IEC (1998e): *FDIS 14764: Software Engineering - Software Maintenance (draft)*, Dec-1998.

ISO/IEC (2000a): *Framework for ISO/IEC System and Software Engineering Standards*, draft 6.1, July 2000.

ISO/IEC (2000b): *Amendment to ISO/IEC 12207:1995 - Information Technology - Software Life Cycle Processes*.

ISO/IEC (2001a): *Amendment to ISO/IEC TR 15504-2 - Information Technology - Software Process Assessment - Reference Model Extensions For Acquirer Processes*.

ISO/IEC (2001b): *CD 15940: Information Technology -Software Engineering Environment Services*,

6.2. Referencias Bibliográficas.

working draft 5, May-2001.

Kobialka, H.U. (1998): Implementing Support for Software Processes in a Process-centered Software Engineering Environment. Tesis doctoral. GMD Research Series No. 15/1998 (Alemania). Disponible en <http://www.gmd.de/publications/research/1998/015/Text.ps>.

Lédeczi, A., Bakay, A., Maróti, M., Völgyesi, P., Nordstrom, G.; Sprinkle, J. y Karsai, G. (2001): Composing Domain-Specific Design Environments. *IEEE Computer* 34(11), pp. 44-51.

Lepe, K., Mora, P. y Rojas, C. (2001): "De la programación de ambientes a ambientes para diseñar". Universidad de Concepción Facultad de Ingeniería, <http://www.inf.udec.cl/~ingsoft/traduccion.doc>

Lonchamp, J. (1994): An Assesment Exercise. En Finkelstein, A., Kramer, J. y Nuseibeh, B. (editors); *Software Process Modelling and Technology*. Research Studies Press, Wiley and Sons, pp. 335-356.

Madhavji, N.H. (1991): The Process Cycle. *Software Engineering Journal*, 6(5), pp. 234-242, 1991.

Martínez, A., Canós, J.H. y García-Consuegra, J.D. (2001a): Un Estilo Arquitectónico para la Federación de Sistemas Basados en Procesos. *VI Jornadas de Ingeniería del Software y Bases de Datos*. Almagro (España), pp. 99-111.

Martínez, A., Canós, J.H. y García-Consuegra, J.D. (2001b): *Estudio del Estado del Arte en Modelado y Ejecución de Procesos Interorganizacionales*. Universidad de Castilla-La Mancha, Departamento de Informática, informe técnico #DIAB-01-06-21.

McCarthy: referente al Concepto "reificación", Disponible: <http://club.telepolis.com/ohcop/mcreific.html>

McChesney, I.R. (1995): Toward a Classification Scheme for Software Process Modelling Approaches. *Information and Software Technology*, 37(7), pp. 363-374.

Moore, J.W. (1998): *Software Engineering Standards: A User's Road Map*. IEEE Computer Society Press.

Murer, T., Würtz, A., y Scherer, D. (1996): A 3D Model for a Common Understanding of the Software Process. *Proceedings of Asia-Pacific Conference on Computer Human Interaction (APCHI 96)*. Singapur, pp. 318-323.

Ossher, H., Harrison, W., y Tarr, P. (2000): Software Engineering Tools and Environments: a Roadmap. *International Conference on Software Engineering (ICSE) - Future of SE Track*. Limerick (Ireland), pp. 261-277.

Pressman, R.S. (1998): *Ingeniería del Software. Un Enfoque Práctico* (4ª edición). McGraw-Hill Interamericana, España.

Ruiz G., F. (2003): MANTIS: Definición de un entorno para Gestión del Mantenimiento de Software. Capítulo 3. Universidad de Castilla-La Mancha, Dep. de Informática. Tesis Doctoral, Junio 2003. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/masi/doc/lec/parte2/fruiz-cap3.pdf>.

Rumbaugh, J., Jacobson, I. y Booh, G. (1999): *The UML Reference Manual*. Addison-Wesley.

Royce, W. (1993): Why Software Costs so Much?. *IEEE Software*, 10(3), pp. 90-91, May 1993.

Sánchez-López (2000): *Redes Iniciación y Referencia*. Jesús Sánchez / Joaquín López Lérica. McGraw-Hill. México.

6.2. Referencias Bibliográficas.

SEI (2002): *Capability Maturity Model Integration (CMMI)*, version 1.1. March 2002. Disponible en <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr011.pdf>

Sommerville, I. (2002) *Ingeniería de Software (6ta Edición)*. Pearson Educación,. México

Sutton, S.M., Tarr, P.L. y Osterweil, L.J. (1995a): *An Analysis of Process Languages*. TR 95-78, Computer Science Department, University of Massachusetts, USA.

Thomas, I. y Nejme, B.A. (1992): Definitions of Tool Integration for Environments. *IEEE Software*, 9(2), pp. 29-35.

UCM (2000): Las nuevas Tecnologías. <http://www.ucatolicamz.edu.co/centros/cuvirtual/tecnologias.htm>
Universidad de Castilla -La Mancha, Dep. de Informática. Proyecto fin de carrera, septiembre 2000.

Urman, Scott, (2002): "Oracle 9i Programación PL/SQL". McGraw-Hill Interamericana. España.

Wang, Y. y King, G. (2000): *Software Engineering Processes: Principles and Applications*. CRC Press.

Wasserman, A. (1989): Tool Integration in Software Engineering Environments. En Long, F. (editor), *Software Engineering Environments: Proceedings of the International Workshop on Environments*. Chinon (Francia), LNCS 467, Springer-Verlag, pp. 137-149.

Zelkowitz, M.V. (1993): Use of an Environment Classification Model. Fifteenth *ACM/IEEE International Conference on Software Engineering*, Baltimore (USA), pp. 348-357.

Zelkowitz, M.V. (1996): Software Engineering Environment Capabilities. *Journal of Systems and Software*. Elsevier Science, 35(1), pp. 3-14.