

Universidad de las Ciencias Informáticas

Facultad 3



Título:

**Modelo de Factoría de Software para la
organización de Proyectos Productivos en el
Instituto Politécnico “Abel Santamaría”**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:

Yaneika Miranda Negrín

Tutor:

Ing. Rolando Pérez Pinto

Junio, 2007

Las Fábricas de Software son el próximo paso lógico en la continua evolución de los métodos y prácticas de desarrollo de software.

Jack Greenfield, 2004

Dedicatoria de Autoría

Declaratoria de Autoría.

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yaneika Miranda Negrín

Ing. Rolando Pérez Pinto

Agradecimientos

Agradecimientos

Dar las gracias al Comandante en Jefe Fidel Castro Rúz por haber hecho posible mi sueño.

A la Revolución Cubana por brindarme la valiosa oportunidad de estudiar en la Universidad del Futuro.

A los profesores de la Universidad de las Ciencias Informáticas que con tanta paciencia y conocimientos me han convertido en una persona mejor.

A mi familia que me ha ayudado siempre y me ha procurado apoyo, a la cual le estoy eternamente agradecida.

A Pascual Verdecia por ser mi profesor y amigo, por ayudarme siempre y dar buenos consejos.

A Rolando Pérez por guiarme en la realización de este trabajo, por la ayuda y el tiempo dedicado.

A mis amigos que tanto cariño y confianza me han brindado.

Dedicatoria

Dedicatoria

Dedicado en primer lugar al Comandante en Jefe Fidel Castro Rúz, por ser razón, virtud, vanguardia y ejemplo a seguir.

Dedicado especialmente a mi familia:

A mi madre que con tanto amor y sacrificio me enseñó a pensar en grande, a ser mejor cada día, gracias por estar ahí y por ser lo mejor del mundo.

A mi abuela y abuelo por ser mis guías e inspiración, por prepararme para la vida, por enseñarme a luchar por lo que quiero y es justo, porque esté es su sueño.

A mi tío por haber sido siempre, y hoy más que nunca, mi ídolo y un ejemplo a seguir, por confiar y ser el mejor padre.

A mi primo José Antonio, que le gustan las computadoras, por su cariño y por mantener viva la niña que hay en mí.

A Falcón por estos años de sacrificio y amor, por ser mi padre y ejemplo.

Dedicado también a Dayron, al cual amo, por la confianza y el apoyo brindado, por levantarme cada vez que he caído.

A mi querida Ileana por su cariño y apoyo, por su amor, sus consejos y por transmitirme desde el primer día los deseos de luchar y seguir adelante.

Abstract

Resumen

En la década de los noventa se comienza a reflexionar sobre la necesidad de un cambio en las formas y medios de producir software. Los ingenieros quieren evolucionar de una producción artesanal a una producción industrial.

La Industria Cubana del Software (INCUSOFT) es una rama que comienza a desarrollarse en estos tiempos y está llamada a convertirse en una significativa fuente de ingresos al país.

A los Institutos Politécnicos de Informática del país les corresponde la honrosa tarea de colaborar en esta misión, puesto que son centros de vanguardia creados para capacitar, no solo excelentes bachilleres sino también técnicos en informática que engrosaran las filas de los programadores cubanos, acrecentando así nuestro prestigio a nivel internacional.

En el presente trabajo se confeccionó un Modelo de Factoría de Software cuyo objetivo radica en la organización de los proyectos productivos en el Instituto Politécnico de Informática “Abel Santamaría”.

Una vez implementado el Modelo se tomó como estrategia para su adopción, realizar una prueba piloto a dos proyectos productivos del Instituto Politécnico de Informática “Abel Santamaría”. Aunque no se pudo llegar a la total adopción del Modelo en este primer intento, por falta de tiempo, los resultados obtenidos en la primera prueba fueron satisfactorios y alentadores, proporcionando la base y experiencia para continuar adelante con la implantación.

Índice

Índice

AGRADECIMIENTOS.....	IV
DEDICATORIA.....	V
RESUMEN.....	VI
INTRODUCCIÓN	1
CAPITULO 1: MARCO TEÓRICO.....	5
INTRODUCCIÓN	5
1.1 FACTORÍA	5
1.2 FACTORÍA DE SOFTWARE.....	6
1.2.1 <i>Objetivos de una Factoría de Software</i>	9
1.3 MODELOS DE FACTORÍA DE SOFTWARE	10
1.3.1 <i>Modelo Eureka</i>	11
<i>Aporte del Modelo</i>	11
1.3.2 <i>Modelo Clasificadorio</i>	12
<i>Aportes del Modelo</i>	13
1.3.3 <i>Modelo propuesto por Basili</i>	13
<i>Aporte del Modelo</i>	14
1.3.4 <i>Modelo basado en la norma ISO 9001 y CMM</i>	14
<i>Aporte del Modelo</i>	15
1.3.5 <i>Modelo Replicable</i>	15
1.3.6 <i>Modelo Funcional de la Factoría de Software de la UCI para la Línea Carrefour. [9]</i>	17
1.4 ESTÁNDARES DE CALIDAD.....	19
1.4.1 <i>Capability Maturity Model (CMM)</i>	19
<i>Aporte del Modelo</i>	22
1.4.2 <i>Modelo Integrado de Capacidad y Madurez (CMMI)</i>	23
<i>Aportes del Modelo</i>	25
1.4.3 <i>PSP y TSP</i>	25
1.5 METODOLOGÍAS DE DESARROLLO DEL SOFTWARE.....	26
1.5.1 <i>Metodología de desarrollo Rational Unified Process (RUP)</i>	26
1.5.2 <i>Metodología Programación eXtrema (XP)</i>	27
1.5.3 <i>Metodología para el Desarrollo Orientado a Objetos de Sistemas Informáticos (ADOOSI)</i>	28
1.6 SISTEMAS GESTORES DE BASE DE DATOS.....	29
1.6.1 <i>Microsoft SQL</i>	29

Índice

1.6.2 PostgreSQL.....	30
1.6.3 ORACLE.....	31
1.7 ESTILOS DE DIRECCIÓN	32
1.8 ORGANIGRAMAS DE EQUIPOS	34
1.9 REPOSITORIO DE COMPONENTE	34
1.10 HERRAMIENTAS DE GESTIÓN DE PROYECTO.	35
1.10.1 Microsoft Project 2003.....	35
1.10.2 Project KilckStart.....	35
1.11 HERRAMIENTAS DE GESTIÓN DE LA CONFIGURACIÓN.....	36
1.11.1 Concurrent Version System (CVS).....	36
1.11.2 Subversion (SVN)	37
1.12 INSTITUTO POLITÉCNICO DE INFORMÁTICA “ABEL SANTAMARÍA”	38
CONCLUSIONES.....	39
CAPITULO 2: MODELO ORGANIZATIVO PARA PROYECTOS PRODUCTIVOS (MOPP).....	40
INTRODUCCIÓN	40
2.1 POR QUÉ LA ADOPCIÓN DE MOPP. PARTICULARIDADES DE LA PRODUCCIÓN DEL INSTITUTO POLITÉCNICO DE INFORMÁTICA.	40
2.2 IDENTIFICACIÓN DE LOS ELEMENTOS DEL MODELO	41
2.3 ÁREA TRABAJADORES	45
2.3.1 Distribución de roles en la factoría de software.....	45
2.4 ORGANIZACIÓN DE LOS TRABAJADORES EN LA FACTORÍA.	50
2.5 ESTILO DE DIRECCIÓN	53
2.6 EL PSP (PROCESO DEL SOFTWARE PERSONAL) COMO INSTRUMENTO PARA ALCANZAR LA CALIDAD EN EL TRABAJO DE LOS PARTICIPANTES.	54
2.6.1 Cuaderno del Ingeniero.....	55
2.6.2 Cuaderno de Registro de Tiempo.....	56
2.6.3 Cuaderno de Resumen Semanal de Actividades.....	57
2.6.4 El Cuaderno de Trabajo.	58
2.7 PROCESO DE DESARROLLO DEL SOFTWARE.	60
2.8 ÁREA TÉCNICAS Y HERRAMIENTAS	60
2.8.1 HERRAMIENTAS PARA LA GESTIÓN DE PROYECTO.	61
2.8.2 Herramientas para la Gestión de Configuración	61
2.8.3 Tecnologías y herramientas de Construcción.....	62
2.8.4 Mecanismos de Control de la calidad.	64

Índice

2.8.5 Mecanismo de seguridad de la información.....	64
2.9 ÁREA REPOSITORIO DE COMPONENTE	65
CONCLUSIONES.....	66
CAPITULO 3: RESULTADOS DE LA IMPLANTACIÓN DEL MODELO ORGANIZATIVO PARA PROYECTOS PRODUCTIVOS	67
3.1 RESULTADOS EN LA VINCULACIÓN DE ESTUDIANTES A LA PRODUCCIÓN.....	67
3.2 RESULTADOS EN EL ÁREA TRABAJADORES.....	70
3.3 ÁREA TÉCNICAS Y HERRAMIENTAS	74
3.4 ÁREA PROCESO DE DESARROLLO.....	76
3.5 ÁREA REPOSITORIO DE COMPONENTES.....	76
CONCLUSIONES.....	76
CONCLUSIONES GENERALES	78
RECOMENDACIONES.....	79
REFERENCIAS BIBLIOGRÁFICAS	80
BIBLIOGRAFÍA.....	82

Introducción

Introducción

En la década de los noventa se comienza a reflexionar sobre la necesidad de un cambio en las formas y medios de producir software. Los ingenieros quieren evolucionar de una producción de software artesanal a una producción industrial. Es entonces cuando comienzan a aparecer en el mercado las herramientas y metodologías para el desarrollo de software. Desde entonces, las empresas que se dedican a la producción de software han ido avanzando en sus métodos y sistemas de producción.

La Industria Cubana del Software (INCUSOFT) es una rama que comienza a desarrollarse en estos tiempos, dada la importancia que juega a nivel mundial la producción de Software, está llamada a convertirse en una significativa fuente de ingresos al país, constituyendo así su desarrollo una de las principales tareas del Gobierno Cubano.

Es a partir de la Feria “Informática 2004” que el país comienza a ganar cierta imagen en el mercado externo, siendo las entidades productoras de Software vinculadas a las áreas de salud, educación y deporte, en su mayoría, las que han tenido mayor éxito en la actividad, en materia exportadora actualmente.[19]

Esa imagen a nivel Nacional e Internacional, principalmente en países latinoamericanos, se ha fortalecido con la creación de la Universidad de las Ciencias Informáticas (UCI), una Universidad de nuevo tipo que combina el estudio, la producción y la investigación, con una fuerte vinculación Universidad Empresa cuyo objetivo es, entre otros, introducir en los centros de enseñanza, un nuevo concepto basado en la relación Universidad-Empresa. Un centro donde la producción es un problema social, político, económico, donde estudiantes y profesores están vinculados a la producción.

Por otro lado, una fuerza que impulsa la producción de software y la informatización de la sociedad cubana en estos días, la protagonizan los estudiantes de los Institutos Politécnicos de Informática (IPI). Centros de vanguardia creados para capacitar no solamente excelentes bachilleres sino también técnicos en informática que pasaran a engrosar las filas de los programadores cubanos, acrecentando así nuestro prestigio internacional.

Introducción

Problemática

Actualmente en el Instituto Politécnico de informática “Abel Santamaría” la vinculación de los estudiantes con la producción es mínima, esto se convierte en un verdadero freno ya que la idea esencial que se tiene con respecto a estos centros es precisamente implementar, en la enseñanza politécnica, la vinculación con la producción y la utilización de los Modelos I+D.

Los estudiantes se sienten desubicados a la hora de enfrentarse a la construcción de un software, puesto que no saben qué hacer en cada momento ni a quién dirigirse, esto trae como consecuencia la desorganización de la producción afectando así la productividad en general.

Problema Científico

El problema identificado radica en la no existencia de una infraestructura productiva bien definida, lo cual se desglosa en:

1. Carencia de una organización que delimite las responsabilidades de cada persona involucrada con el proceso productivo (Asignación de roles).
2. Insuficiente dominio de las herramientas de trabajo por parte de los estudiantes.

Objeto de Estudio

Los modelos de Factoría de Software existentes actualmente.

Campo de Acción

Organización de Proyectos Productivos.

Objetivo General

Desarrollar un modelo de Factoría de Software que se ajuste a las particularidades de la producción en el Instituto Politécnico de Informática “Abel Santamaría”.

Introducción

Objetivos Específicos

1. Definir un modelo organizacional apropiado al Politécnico de Informática “Abel Santamaría”.
2. Definir buenas prácticas que garanticen la superación de los estudiantes implicados en el proceso de producción de software.

Tareas

- Realizar un levantamiento del estado del arte de los modelos de Factoría de Software, estándares de Calidad de Software, herramientas para la Gestión de Software, herramientas para la Gestión de la Configuración, diferentes Metodologías, Estilos de Dirección y Organigramas.
- Adoptar un criterio para la definición de Factoría de Software que se ajuste a las características de esta investigación.
- Evaluar la implantación del Modelo en el Instituto Politécnico de Informática “Abel Santamaría”.

El presente trabajo esta dividido en tres capítulos, a continuación se muestra el título de cada uno de ellos junto con un breve resumen de su contenido.

Capítulo 1: Marco Teórico.

En este capítulo se abordan conceptos que tienen relación con el tema a desarrollar y que son de gran importancia para el transcurso de la investigación. Se lleva a cabo un levantamiento del estado del arte de algunos de los modelos más representativos de Factoría de Software, Estándares de Calidad, Estilos de Dirección, Organigramas, Herramientas y Metodologías.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP).

En este capítulo se justifica el por qué de la creación de un Modelo específico para el Instituto Politécnico de Informática “Abel Santamaría”. Se identifican las entidades de dicho Modelo

Introducción

así como las relaciones que se establecen entre ellas, quedando descritas minuciosamente las características de dicho Modelo.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos.

El objetivo de este capítulo es exponer los resultados obtenidos en la adopción del Modelo Organizativo para Proyectos Productivos por el Instituto Politécnico de Informática “Abel Santamaría”.

Capítulo 1: Marco Teórico

Capítulo 1: Marco Teórico

Introducción

En la actualidad ha aumentado considerablemente la complejidad con la que se desarrollan sistemas informáticos para las empresas, por lo que resulta difícil generar productos que cumplan cabalmente con las expectativas del cliente.

Para responder a esta situación han surgido una serie de herramientas, técnicas y modelos que facilitan a las organizaciones encargadas de producir Software, generar productos que satisfagan las necesidades de los clientes, herramientas que prometen ser la solución a los problemas de calidad, costo y tiempo de desarrollo.

Haciendo honor a lo anteriormente expuesto en el presente capítulo se abordarán un grupo de conceptos y definiciones que constituyen la base de la presente investigación en la búsqueda de un Modelo de Factoría de Software para la organización de los proyectos productivos en el Instituto Politécnico de Informática "Abel Santamaría".

Quedando además despejadas una serie de incógnitas, tales como: qué es factoría, qué entendemos como Factoría de Software, qué son los Politécnicos de Informática, entre otros.

1.1 Factoría

Se denomina así, de forma genérica, a cualquier tipo de fábrica o industria, es decir, a cualquier tipo de instalación en la cual se produce la transformación de materias primas o productos semiterminados en otros productos, bien para otras industrias, bien para su uso o consumo final. [7]

Por extensión se está aplicando esta palabra para designar determinadas actividades en las cuales no se produce consumo y transformación de materias y que tienen como objeto final la obtención de productos intangibles: factoría de comunicación, factoría de cine, factoría de software. [7]

En la época colonial se denominaba "factoría" a cualquier establecimiento dedicado al comercio entre la colonia y la metrópoli y especialmente a los situados en la colonia. [7]

Capítulo 1: Marco Teórico

1.2 Factoría de Software

A modo de introducción podemos comenzar diciendo que el término de Factoría de Software fue utilizado por primera vez en la década del 60 en Japón. Pero varias empresas asociaron el término al mero desarrollo de software.

A continuación se exponen una serie de conceptos que diferentes autores dan al término de Factoría de Software:

Una empresa productora de software que no responda a características como: producción de software en gran escala, estandarización de tareas, estandarización del control, división del trabajo, mecanización y automatización, no puede ser considerada una factoría de software. El desarrollo de una factoría implica que las buenas prácticas de Ingeniería de Software sean aplicadas sistemáticamente [1].

Una organización fabril para el desarrollo de software debe tener claro el asunto del “software único”, es decir, todo software es único, pero algunas partes de ellos se pueden repetir en varios proyectos. El proceso industrial debe contener el desarrollo, almacenamiento y montaje de partes reutilizables en un producto [2].

Una factoría de software debe, para ser flexible, ser capaz de producir varios tipos de productos; llevar a cabo conceptos de Ingeniería de Software (metodología, herramientas, gestión de la configuración), y también ser capaz de estudiar, diseñar, implementar y mejorar sus sistemas y procesos [3].

Una factoría de software debe poseer un conjunto de herramientas estandarizadas para la construcción de software, bases históricas para ser usadas en la dirección de proyectos, y principalmente, poseer un alto grado de reutilización de código en el proceso de desarrollo de un determinado software, apoyado en una base de componentes reutilizables [4].

Una factoría de software -concepto que se contrapone a lo que es la manufactura artesanal de software- es una organización capaz de manufacturar productos con calidad aceptada en el ámbito mundial bajo criterios de rentabilidad, planificación, diseño y organización. El

Capítulo 1: Marco Teórico

software necesita una organización tras de sí, ya que se trata de uno de los principales bienes de capital de los negocios. [5]

Se podría pensar que Microsoft es una fábrica de software, pero no es así. Ellos basan su negocio en la creación de programas dirigidos a públicos masivos, más que a desarrollos especializados para cierto tipo de industria; hacen software genérico como hojas de cálculo, procesadores de palabra y sistemas operativos. Por el contrario, el modelo de fábricas de software se basa en que la empresa ofrezca servicios a la medida a sus clientes, es un concepto de outsourcing, en el cual se delega el diseño de software a una empresa dedicada totalmente a ese fin, la cual está encargada de desarrollar plataformas para sistemas administrativos, nóminas, control, procesos y muchos más. [5]

Se podría decir que poseer una biblioteca de objetos es lo que diferencia una fábrica artesanal de una industrial, porque las cosas no se vuelven a hacer cada vez, como una artesanía: cada nueva taza se vuelve a moldear y cada unidad es una pieza única. Cuando se hace un molde para construir millones de tazas, se marca la diferencia entre artesanía y producción industrial. En este caso, son moldes de conocimiento que se utilizan en la fabricación de software. [5].

Una factoría de software es una organización con procesos estructurados, controlados y mejorados de forma continua, considerando principios de Ingeniería Industrial, orientados a dar respuesta a múltiples demandas de distinta naturaleza y alcance. Dirigida a la creación de productos de software, conforme a los requerimientos documentados de los usuarios y clientes, de la forma más productiva y económica posible [6].

Una factoría de software puede tener varios dominios de actuación en dependencia de las fases de desarrollo del software en las que opere, desde un proyecto de software completo, hasta un proyecto físico o solo codificación del software. [6].

Es una línea de productos software que configura herramientas extensibles, procesos y contenido [...] para automatizar el desarrollo y mantenimiento de variantes de un producto arquetípico mediante la adaptación, ensamblaje y configuración de componentes basados en frameworks [12].

Capítulo 1: Marco Teórico

Las Factorías de Software se centran en el desarrollo de sistemas similares promoviendo la reutilización de arquitecturas, componentes software y conocimiento [15].

En las definiciones de Factoría de Software anteriormente expuestas encontramos un grupo de características importantes que no pueden ser ignoradas a la hora de definir un concepto de Factoría de Software, algunas de ellas a modo de resumen son:

- Proceso definido y estandarizado para el desarrollo de software.
- Producción de software a gran escala.
- Planificación y control de la producción.
- Aplicación sistemática de las buenas prácticas y conceptos de Ingeniería de Software.
- Alto grado de reutilización de código en el proceso de desarrollo de software.
- Construcción de software basada en componentes y la reutilización de los mismos.
- Control de los recursos humanos involucrados.
- Control y almacenamiento en bibliotecas de componentes de software (documentos, código, métodos, etc.)
- Control del estado de ejecución de todas las demandas.
- Capacitación de los recursos humanos.
- Herramientas estandarizadas para la construcción de software.
- Hardware y software ajustado a las necesidades del usuario.
- Interacción controlada con el cliente, alto poder de entendimiento.
- Estimación de costos y tiempo basados en el conocimiento real de la capacidad productiva, mediante métodos de obtención basados en datos históricos.
- El cliente dispone de una visión total del proceso.
- Garantía de la calidad del producto.
- Mecanismos de control de costos.

Capítulo 1: Marco Teórico

- Aporte de una mayor garantía respecto al resultado final.

Basado en las diferentes definiciones que dan los autores anteriormente citados y en las características que este trabajo ha señalado como fundamentales a la hora de referirse a Factoría de Software, se ha tomado como concepto de Factoría de Software en esta investigación el siguiente:

Organización estructurada y creada para el desarrollo de software, con procesos estandarizados, repetibles, gerenciales y mejorables continuamente. Con un grupo de herramientas estandarizadas, tanto para la construcción de software como para la gestión y administración de proyectos automatizando gran parte del trabajo, con una alta reutilización de código y de componentes (desarrollo basado en componentes), arquitectura y conocimiento en el desarrollo de un determinado producto de forma tal que haya una reducción del trabajo para la obtención final de mejores resultados en menor tiempo y con menos costos (esto es productividad).

Dado un determinado proyecto se debe contar con técnicas que predigan las actividades de desarrollo del mismo, es decir, la existencia de técnicas para la estimación de costos, plazos y tamaño de un equipo para un determinado proyecto, basado en el conocimiento real de la capacidad productiva, para lograr que los costos estimados y compromisos de cronograma puedan ser satisfechos en tiempo y forma.

Es importante tener establecida una política que garantice la calidad del software, basando el desarrollo en métodos y técnicas estandarizadas, y manteniendo capacitados sus recursos humanos. Debe tener establecidas líneas de productos que se enfoquen a segmentos de mercado, esto supone la creación de una arquitectura común para cada línea de producto.

1.2.1 Objetivos de una Factoría de Software

El enfoque de factoría de software viene a formalizar todos los procesos (etapas de producción) y sus productos, trabajando en líneas de producción, con etapas y tareas perfectamente definidas para cada tipo de profesional involucrado en el proceso, yendo de la productividad en la línea de producción a las rutinas de control de la calidad. Se busca la especialización de los profesionales, para que cada uno garantice la productividad de la fase

Capítulo 1: Marco Teórico

en la que está ocupado. Entre los principales objetivos trazados por una factoría de software están:

1. Industrializar el desarrollo de sistemas de software.
2. Producción de software a gran escala.
3. Lograr una alta productividad en el desarrollo de software.
4. Establecer una línea de producción.
5. Mejora continua de los procesos.
6. Estimación de costos y plazos extremadamente precisos.
7. Reducción de los costos de producción.
8. Lograr un buen control de la calidad.
9. Especializar al profesional en una tarea específica del proceso, concentrando sus esfuerzos en dicha tarea.

1.3 Modelos de Factoría de software

En el presente epígrafe se hace un análisis de algunos de los modelos de Factoría de Software más significativos encontrados en la bibliografía consultada, estos modelos constituyen la vía de muchas organizaciones de llevar a cabo el enfoque de Factoría de Software. Los modelos seleccionados son:

- Modelo Eureka.
- Modelo Clasificadorio.
- Modelo propuesto por Basili.
- Modelo basado en la norma ISO 9001 y CMM.
- Modelo Replicable.
- Modelo Funcional de la Factoría de Software de la UCI para la Línea de Carrefour.

Capítulo 1: Marco Teórico

1.3.1 Modelo Eureka

El Modelo Eureka surgió como el proyecto Eureka Software Factory. El objetivo del proyecto es crear un mercado para productos CASE¹. En el mismo son partícipes un conjunto de compañías europeas, dichas compañías actúan en las siguientes áreas: manufactura de computadoras, instituciones de investigación, producción de herramientas CASE y desarrollo de sistemas.

El modelo fabril propuesto por el proyecto Eureka está compuesto de proceso, reglas, herramientas, información, trabajadores y equipamiento (computadoras), (Ver Anexo 1).

El proceso de desarrollo está compuesto por reglas, las que son definidas por las personas involucradas en el ambiente de desarrollo de software y constituyen patrones a seguir, algoritmos, métodos de desarrollo de software. Las herramientas e información almacenada, soportan la automatización del proceso de desarrollo.

El modelo posee características giradas al proceso de desarrollo de software distribuido, en el mismo se sigue el enfoque software bus, que es un interfaz normalizado para la interconexión de herramientas. Enfoque que estipula reglas de conexión de componentes en la construcción de un software[7] (Ver Anexo 2).

Aporte del Modelo

- Es utilizado por un conjunto de empresas Europeas y hay que resaltar que el Viejo Continente presenta un alto nivel en el desarrollo de software.

¹ CASE por sus siglas en inglés Computer-Aided Software Engineering. Son herramientas que facilitan el desarrollo organizado del software aplicado técnicas de Ingeniería de Software. En sentido amplio podemos englobar en la tecnología CASE toda la variedad de herramientas aplicables en el desarrollo de software.

Capítulo 1: Marco Teórico

- Posee características que están presentes en la mayoría de las definiciones de Factoría de Software como son: la utilización de herramientas para la automatización del proceso de desarrollo y desarrollo basado en componentes.
- El modelo da una visión de cómo se puede desarrollar un producto en partes manejables y después la unión de estas para formar el producto final.

1.3.2 Modelo Clasificadorio

El Modelo Clasificadorio propuesto por Fernández y Teixeira [6] está dirigido a clasificar las factorías de acuerdo al alcance o ámbito de funcionamiento que tienen a lo largo del proceso de desarrollo de software [7], (Anexo 3). Una Fábrica de Software puede ser clasificada como:

1. Factoría de Proyectos Ampliada.
2. Factoría de Proyectos de Software.
3. Factoría de Proyectos Físicos.
4. Factoría de Programas.

La Factoría de Proyectos Ampliada comprende el concepto de arquitectura de solución. La arquitectura de solución es una etapa anterior al diseño conceptual del software, la que se ocupa de proyectar una solución en la que el software está formado por los componentes más significativos arquitectónicamente, se definen los principios que orientan el diseño y evolución del software. La arquitectura de solución puede contener, además del software, definición de procesos, definición de equipamiento, infraestructura de redes, plataforma de desarrollo, patrones a seguir.

La Factoría de Proyectos de Software abarca todo el ciclo de vida sistémico para la realización del software, correspondiente al análisis, diseño, implementación, prueba e implantación. En este tipo de factorías se tiene un conocimiento al detalle del negocio a automatizar.

Capítulo 1: Marco Teórico

La Factoría de Proyectos Físicos se abstrae del enfoque sistémico del software, se dedica al diseño, implementación y prueba. No se tiene un pleno conocimiento del negocio.

La Factoría de Programas, considerada la menor de las entidades, tiene como objetivos desarrollar componentes de código para la construcción del software. Esta factoría no se preocupa del contexto sistémico ni del diseño, se ocupa de producir código según las especificaciones del diseño. Posee como entrada la especificación del diseño de una parte del software y su salida es un componente de código que formará parte del software a desarrollar.

Aportes del Modelo

- Realiza una clasificación de las Factorías de Software de acuerdo con las actividades que realizan durante el proceso de desarrollo de software.
- Define hacia donde se puede ir avanzando en este enfoque

1.3.3 Modelo propuesto por Basili

El presente modelo divide una factoría de software en dos grandes entidades: Organización basada en proyectos y Factoría de componentes, (Ver Anexo 4). El autor plantea que una organización con características de Factoría de Software debe poseer una estructura de construcción de software basada en componentes.

Los componentes utilizados en la construcción del software pueden ser desarrollados por la factoría de componentes.

La Organización basada en proyectos realiza las solicitudes de productos (componentes para la construcción del software), de datos (estadística para la estimación de costo y plazos) o de planos (modelos, métodos para el análisis y diseño de software) a la factoría de componentes. La factoría de componentes posee una base de componentes reutilizables de la cual se apoya para dar respuesta a las solicitudes hechas por la unidad de producción de software. En respuesta a la solicitud, la organización basada en proyectos, recibe los modelos y componentes para la construcción del software, además de estadísticas y datos históricos que se encuentran en la base de componentes.

Capítulo 1: Marco Teórico

Aporte del Modelo

- División de la factoría en dos unidades.
- Es un Modelo que puede adaptarse con facilidad a las características de cualquier Factoría de Software.
- Da un enfoque a la reutilización durante el desarrollo del software.
- Implementa la utilización del desarrollo basado en componentes.

1.3.4 Modelo basado en la norma ISO 9001 y CMM

En este modelo se hace una división de los elementos fundamentales de una Factoría de Software en cinco entidades bien definidas, (Ver Anexo 5).

La arquitectura propuesta por el modelo, (Ver Anexo 6) propone que:

La Entidad Técnicas provee el soporte técnico y conceptual para la definición del proceso. Este es guiado por el estándar de calidad CMM².

La Entidad Gestión de la Factoría se divide en dos entidades, la Sub-Entidad Gestión de Calidad y la Sub-Entidad Organización del Modelo de Proceso.

La Entidad Trabajadores es la encargada de definir los trabajadores involucrados en el proceso de desarrollo de software y sus roles. Los trabajadores involucrados en el proceso de desarrollo del software son guiados por los modelos Personal Software Process (PSP)³ y Team Software Process (TSP)⁴.

La Sub-Entidad Organización del Modelo de Proceso define características y organización del proceso de desarrollo de software.

² Modelo de Capacidad y Madurez, CMM por sus siglas inglés Capability Maturity Model.

³ Proceso Personal de Software, PSP por sus siglas en inglés Personal Software Process.

⁴ Proceso de Software en Equipo, TSP por sus siglas en inglés Team Software Process.

Capítulo 1: Marco Teórico

Los activos del proceso, las herramientas y los componentes de código dan soporte al proceso de desarrollo de software.

Aporte del Modelo

- Define las entidades y las relaciones que forman el Modelo factoría de Software.
- Define la aplicación de normas y técnicas de calidad usadas actualmente en el mundo del software.

1.3.5 Modelo Replicable

El presente modelo fue desarrollado para ser replicado en una factoría de software. El proceso puede ser aplicado al desarrollo de software o al desarrollo de un componente. Se puede adaptar a las necesidades y recursos de una factoría determinada, actualmente es el modelo que más se desarrolla dado el gran alcance que tiene.

Este Modelo plantea que una factoría de software debe poseer (Ver Anexo 7):

- Un modelo de organización de la producción.
- Una unidad de producción de componentes y una unidad de producción de software.
- Tanto la unidad de producción de componentes como la de software poseen procesos.
- El proceso es guiado por un modelo de calidad de software.
- El proceso es compuesto de actividades que son compuestas de tareas.
- Las tareas utilizan los componentes, y estos son clasificados en infraestructura (o activos del proceso) y código.
- Las tareas usan un conjunto de herramientas para la automatización de las mismas.
- Por último el proceso puede ser aplicado al desarrollo de software o al desarrollo de un componente.

La organización de la producción esta dividida en cinco áreas:

Capítulo 1: Marco Teórico

1. Área de producción de análisis de sistema o modelado de negocio: Esta área es la encargada del estudio de un área de trabajo o negocio, llevando al análisis y especificación del sistema a desarrollar.
2. Área de producción de diseño de software: Es el área encargada de realizar el diseño a partir de la especificación proveniente del análisis del sistema. Un software automatiza un sistema o negocio, por lo que esta área se encuentra incluida en el área de análisis de sistema o modelado de negocio.
3. Área de construcción de software: Esta área es la encargada de construir el producto software de acuerdo a las especificaciones desarrolladas en el área de diseño. Es importante destacar que en una Factoría de Software la construcción se basa en la unión o ensamblado de componentes previamente desarrollados.
4. Área de producción de componentes de infraestructura o activos del proceso: Partiendo de que una Factoría de Software posee una unidad de producción de componentes, y estos pueden ser clasificados como componentes de infraestructura o de código, por lo tanto esta área tiene como principal responsabilidad producir componentes de infraestructura o activos del proceso solicitados por el área de construcción de software.
5. Área de producción de componentes de código: Esta área tiene la función de producir componentes de código solicitados por el área de construcción de software.

La unión de las áreas de análisis de sistemas, diseño de software y construcción de software forman el ámbito de negocio del modelo de producción. El ámbito de negocio incluye la interacción entre el cliente y la fábrica de software. Las áreas de producción de componentes de infraestructura y componentes de código forman el ámbito interno del modelo, el que es transparente a los ojos del cliente de la factoría. Este ámbito es el responsable de los subproductos creados, componentes para la construcción del sistema.

El modelo plantea que una Factoría de Software posee una unidad de producción de software y una de producción de componentes. La unidad de producción de software utiliza

Capítulo 1: Marco Teórico

componentes en la construcción del sistema, que son elaborados por la unidad de producción de componentes.

La unidad de producción de software posee las siguientes actividades: analizar, diseñar, construir, probar, implantar y revisar. Para la unidad de producción de componentes son las siguientes: diseñar componentes, construir, probar, almacenar y distribuir componentes, (Ver Anexo 8).

Este modelo plantea realizar la construcción del software mediante la integración de componentes producidos por la unidad de producción de componentes, lo que se asemeja a la línea de producción y montaje del proceso industrial tradicional (Ver Anexo 9).

Aportes del Modelo

- Resume varios modelos de Factoría de Software tomando de ellos los aspectos más representativos.
- Alto nivel de reutilización.
- Alto nivel de adaptabilidad a cualquier factoría.

1.3.6 Modelo Funcional de la Factoría de Software de la UCI para la Línea Carrefour. [9]

Este modelo fue creado para la Factoría de Software de la Universidad de las Ciencias Informáticas, específicamente para la Línea Carrefour.

Este modelo está dividido en cuatro entidades y sus relaciones, (Ver Anexo 10) las cuales son:

1. Técnicas y Herramientas: Comprende el contexto de las tecnologías, herramientas para dar soporte y automatización al proceso de desarrollo.
2. Proceso de Desarrollo: Comprende el conjunto de actividades que conforman el flujo de trabajo, el cual consta de: Recepción de Especificaciones, Diseño, Implementación, Prueba, Almacenamiento y Entrega.

Capítulo 1: Marco Teórico

3. Participantes: Comprende los recursos humanos involucrados en el proceso de desarrollo del software, la estructura organizativa y los roles que ocupan, está dividida en dos sub-entidades: Gestores de la Factoría y Grupo de desarrollo.
4. Repositorio de componentes: Activos del proceso y componentes de código. Entiéndase como activos del proceso formularios, documentos, patrones, algoritmos utilizados como artefactos en el proceso. Los activos del proceso también pueden ser denominados como componentes de infraestructura, componentes de valor en el proceso.

El resultado final de la adopción de este Modelo en un proyecto en la Factoría Software es un subsistema o componente de alto nivel, que toma forma durante su desarrollo gracias a la intervención de variados tipos de personas, estas personas están representadas en el modelo mediante la sub-entidad Grupo de desarrollo y Gestores de la Factoría. El equipo de desarrollo lo forman las personas involucradas directamente en el proceso, las cuales son quienes ejecutan las actividades o flujos de trabajo (Recepción de Especificaciones, Diseño, Implementación, Prueba, Almacenamiento y Destino), a su vez este equipo es guiado por el proceso de desarrollo de software, representado en el modelo mediante la entidad Proceso de desarrollo.

La sub-entidad Gestores de la Factoría comprende el equipo de dirección de la misma, encargados del control y gestión del grupo de desarrollo y del proceso. El proceso es automatizado y soportado por diversas tecnologías y herramientas, representados en la entidad “Técnicas” en el modelo.

La Entidad Repositorio de Componentes esta dividida en dos grandes grupos:

- Componentes de código: clases, procedimientos o funciones, módulos, subsistemas, aplicaciones.
- Activos del proceso: patrones de diseño, algoritmos, esquema de base de datos, manuales, documentación, modelos.

Capítulo 1: Marco Teórico

El modelo propone que la entidad Participantes haga uso de PSP (Personal Software Process) y TSP (Team Software Process).

Aportes del Modelo:

- Modelo confeccionado para ser implantado en una Factoría de Software cubana.
- Modelo creado para la Factoría de una universidad, donde la mayoría del capital humano esta representado por estudiantes.
- Presenta una detallada distribución de los elementos de una Factoría.
- Alto grado de adaptabilidad a las condiciones de cualquier Factoría.

1.4 Estándares de Calidad

En este epígrafe se realiza un estudio de los estándares de calidad más representativos encontrados en la bibliografía consultada.

Primeramente referenciar dos conceptos importantes y necesarios para la comprensión de este epígrafe:

Capacidad

Es un atributo de los procesos. El nivel de capacidad de un proceso indica si sólo se ejecuta, o si también se planifica se encuentra organizativa y formalmente definido, se mide y se mejora de forma sistemática [8].

Madurez:

Atributo de las organizaciones que desarrollan o mantienen los sistemas de software [8].

1.4.1 Capability Maturity Model (CMM)

El Modelo de Capacidad y Madurez o CMM (Capability Maturity Model) es un modelo para la evaluación de los procesos en una organización.

Capítulo 1: Marco Teórico

Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute)⁵.

A partir de noviembre de 1986 el SEI, desarrolla una primera definición de un modelo de madurez de procesos en el desarrollo de software, que se publicó en septiembre de 1987. Este trabajo evolucionó al modelo CMM o SW-CMM (CMM for Software), cuya última versión (v1.1) se publicó en febrero de 1993.

CMM es un modelo escalonado sobre el concepto de madurez que establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso, (Ver Anexo 11).

A su vez estas Áreas de Proceso se agrupan en cinco niveles o escalones para la clasificación de la madurez de una organización, de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez, (Ver Anexo 12).

Los niveles son:

Inicial o Nivel 1: Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobrecostos. El resultado de los proyectos es impredecible.

Repetible o Nivel 2: En este nivel las organizaciones disponen de unas prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable

⁵ SEI por sus siglas en inglés Software Engineering Institute. Es un Centro de Investigación y Desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon.

Capítulo 1: Marco Teórico

seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente.

Procesos que hay que implantar para alcanzar este nivel son:

- Gestión de requisitos
- Planificación de proyectos
- Seguimiento y control de proyectos
- Gestión de proveedores
- Aseguramiento de la calidad
- Gestión de la configuración

Definido o Nivel 3: Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detallada y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión por pares.

Procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos
- Solución Técnica
- Integración del producto
- Verificación
- Validación
- Desarrollo y mejora de los procesos de la organización
- Definición de los procesos de la organización
- Planificación de la formación
- Gestión de riesgos

Capítulo 1: Marco Teórico

- Análisis y resolución de toma de decisiones

La mayoría de las empresas que llegan al nivel 3 paran aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubierta la mayoría de sus necesidades.

Gestionado o Nivel 4: Se caracteriza porque las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos
- Mejora de los procesos de la organización

Optimizado o Nivel 5: La organización completa está volcada en la mejora de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional.
- Análisis y resolución de las causas.

Normalmente, las empresas que intentan alcanzar los niveles 4 y 5 lo realizan simultáneamente ya que están muy relacionados.

Aporte del Modelo

- Es específico para el desarrollo y mantenimiento de software.
- Definido como un conjunto de áreas clave de procesos.
- Tiene un modelo de evaluación.
- Clasifica a las empresas según el desarrollo alcanzado

Capítulo 1: Marco Teórico

- Provee a las organizaciones de una guía para incrementar el rendimiento de sus procesos de negocios, indicando que la productividad y la calidad de los productos y servicios, aumenta conforme crece el nivel de madurez de los procesos que lo crean.

1.4.2 Modelo Integrado de Capacidad y Madurez (CMMI)⁶

Surge a finales de los 90 cuando algunas organizaciones llevaban a cabo planes de calidad que integraban de forma simultánea varios modelos CMM.

Para facilitar la incorporación de varios CMM's, SEI desarrolla y publica en 2001 el modelo CMMI que integra:

- CMM (Capability Maturity Model).
- SE-CMM (Systems Engineering CMM).
- IPD-CMM (Integrated Product Development CMM).

Desde entonces estos tres modelos ya no evolucionan de forma separada.

CMMI plantea seis niveles para medir la capacidad de los procesos:

Incompleto o Nivel 0: El proceso no se realiza, o no se consiguen sus objetivos.

Ejecutado o Nivel 1: El proceso se ejecuta y se logra su objetivo.

Gestionado o Nivel 2: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.

Definido o Nivel 3: Además de ser un proceso “gestionado” se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.

Cuantitativamente gestionado o Nivel 4: Además de ser un proceso definido se controla utilizando técnicas cuantitativas.

⁶ CMMI por sus siglas en inglés Capability Maturity Model Integration.

Capítulo 1: Marco Teórico

Optimizado o Nivel 5: Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica para adaptarlo a los objetivos del negocio.

Niveles de madurez.

Son los mismos cinco niveles descritos en el modelo CMM.

Nivel 1: Inicial.

Nivel 2: Gestionado.

Nivel 3: Definido.

Nivel 4: Gestionado cuantitativamente.

Nivel 5: Optimizado.

Estructura del modelo CMMI

Los modelos de calidad que centran su foco en la madurez de la organización, presentan un modelo de mejora y evaluación “escalonado”. Los que enfocan las actividades de mejora y evaluación en la capacidad de los diferentes procesos presentan un modelo “continuo”.

CMMI nació integrando tres modelos diferentes, con representaciones diferentes:

- CMM: representación escalonada.
- SE-CMM: representación continua.
- IPD-CMM: modelo mixto.

En el equipo de desarrollo de CMMI había defensores de ambos tipos de representaciones. El resultado fue la publicación del modelo con dos representaciones: La escalonada y la continua. Son equivalentes y cada organización puede optar por utilizar la representación que se adapte a sus características y prioridades de mejora, (Ver Anexo 13).

La visión continua de una organización mostrará la representación de nivel de capacidad de cada una de las áreas de proceso del modelo.

La visión escalonada definirá a la organización dándole en su conjunto un nivel de madurez del 1 al 5.

Capítulo 1: Marco Teórico

Aportes del Modelo

- Posee un modelo de evaluación para la calidad y madurez de las organizaciones.
- Posee dos formas de representar, dando la posibilidad a las empresas de elegir la que se adapte a sus características.
- Es específico para el desarrollo y mantenimiento de software.
- Definido como un conjunto de áreas clave de procesos.
- Clasifica a las empresas según el desarrollo alcanzado.
- Provee a las organizaciones de una guía para incrementar el rendimiento de sus procesos de negocios, indicando que la productividad, la calidad de los productos y servicios aumenta conforme crece el nivel de madurez de los procesos que lo crean.

1.4.3 PSP y TSP

PSP muestra como aplicar métodos avanzados de ingeniería a las tareas diarias de cada individuo, proporciona métodos detallados de planificación y estimación, muestra a los implicados en el proceso como controlar su rendimiento y explica como los procesos definidos guían su trabajo [10].

Si se hace un buen uso de PSP por parte de todos los integrantes de un proyecto se reciben innumerables beneficios, puesto que se pueden obtener datos personales históricos que ayudan a un mejor desempeño de las actividades, logrando que los elementos de trabajo sean predecibles y mejorables.

TSP es un proceso que al igual que PSP esta basado en CMM. Es un modelo o proceso de trabajo en equipo, enfocado a aminorar problemas, tanto técnicos como administrativos, que se presentan en el desarrollo de software a la hora de formar equipos.

El TSP provee un esquema de trabajo donde cada desarrollador tiene perfectamente definido sus roles, sus actividades, y sus responsabilidades. Así mismo, TSP incluye procedimientos para la mejora continua del proceso de desarrollo, para mejorar la calidad del software producido, para mejorar la estimación del tiempo de desarrollo, para la disminución de

Capítulo 1: Marco Teórico

defectos en el producto y para promover la integración del equipo de desarrollo. Es decir, el TSP apoya tanto al equipo de desarrollo como a los administradores del proyecto para la culminación a tiempo y dentro del presupuesto de los proyectos de desarrollo de software. [1]

1.5 Metodologías de desarrollo del Software.

En un proyecto de desarrollo de software la metodología es quien define quién debe hacer qué, cuándo y cómo debe hacerlo.

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo. Toda metodología debe ser adaptada al contexto del proyecto: recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, entre otros.

En la actualidad encontramos dos clasificaciones para las metodologías:

Tradicional o Fuertes: Se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se crean y las herramientas y notaciones que se usarán. Dentro de esta metodología se destaca Rational Unified Process (RUP), como la metodología líder en el desarrollo de software.

Ligeras o Ágiles. Dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Dentro de estas metodologías se encuentran eXtreme Programming (XP), Crystal Methodologies, SCRUM, entre otras.

1.5.1 Metodología de desarrollo Rational Unified Process (RUP)

RUP es una metodología de desarrollo de software orientada a objetos que proporciona un método disciplinado para asignar las tareas y responsabilidades dentro del equipo de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que resuelva las necesidades del usuario dentro de un cronograma predecible y al menor costo posible. [11]

Está basado en componentes y utiliza UML⁷ para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

⁷ Lenguaje Unificado de Modelado. UML por sus siglas en inglés, Unified Modeling Language.

Capítulo 1: Marco Teórico

RUP es más que un proceso de trabajo, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes niveles de aptitud. [11]

Durante su ciclo de vida se desarrolla toda una metodología iterativa incremental que va eliminando los errores cometidos en las iteraciones previas, logrando que al final del proceso se obtenga como resultado un producto de calidad. Está dirigido por casos de uso que son los que guían todo el proceso de desarrollo, permitiendo obtener lo que los usuarios futuros necesitan y desean. Centrado en la arquitectura posibilitando describir los elementos más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente, mostrando al final, la visión del sistema completo. De esta manera se obtiene los resultados de un proceso de ingeniería de software entendible tanto por el equipo de desarrollo como por el cliente. Por otra parte, define los roles a jugar por cada miembro del equipo de desarrollo en cada una de las etapas por las que transcurre el sistema, facilitando la comunicación entre los diferentes miembros del equipo de desarrollo.

Esta metodología es ideal para la gestión de los requisitos, la verificación continua de la calidad del software, la gestión de los cambios y sobre todo, para proyectos grandes, a largo plazo y con un equipo de desarrollo numeroso, (Ver Anexo 15).

1.5.2 Metodología Programación eXtrema (XP)⁸

Permite establecer iteraciones muy cortas, apropiada para un entorno caracterizado por requerimientos cambiantes. Su objetivo principal es tener una nueva versión a cada instante, mostrarlo al cliente, ver lo que opina y seguir programando, su objetivo es tener una comunicación fluida con el cliente y el usuario final, es decir, define una manera de reunir a clientes y programadores en un equipo firmemente integrado con condiciones de trabajo que promueven la comunicación y solución de un problema. Se ha clasificado como una metodología ágil, ya que plantea aumentar constantemente la velocidad del proyecto.

⁸ XP por sus siglas en inglés eXtreme Programming.

Capítulo 1: Marco Teórico

El desarrollo bajo la metodología XP tiene características que lo distinguen claramente de otras metodologías:

- Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- Los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- Se obtiene retroalimentación de usuarios y clientes desde el primer día gracias a las baterías de pruebas.
- El software es liberado en entregas frecuentes tan pronto como sea posible.
- Los cambios se implementan rápidamente tal y como fueron sugeridos.
- Las metas en características, tiempos y costos son reajustadas permanentemente en función del avance real obtenido.

Esta metodología funciona mejor para pequeños equipos, (Ver Anexo 15).

1.5.3 Metodología para el Desarrollo Orientado a Objetos de Sistemas Informáticos (ADOOSI).

La metodología ADOOSI surgió como respuesta a la necesidad de poseer un conjunto de métodos e instrumentos que se unieran de forma tal que permitieran el desarrollo de proyectos que utilizaran el enfoque orientado a objetos, siguiendo cada una de las etapas que constituyen su ciclo de vida siendo estas el estudio preliminar, el análisis, diseño, construcción, prueba y la implantación.

En su versión actual aprovecha las facilidades que brindan los medios ambientes visuales, el auge de la técnica de prototipación, la extensión orientada a objetos del modelo relacional, el desarrollo de lenguajes de programación orientado a objetos que trabajan con bases de datos relacionales y la aceptación como estándar de la notación UML.

En los últimos años se ha incrementado el número de personas interesadas en conocer y dominar métodos para el desarrollo de proyectos orientado a objetos.

Capítulo 1: Marco Teórico

El desarrollo de las aplicaciones con esta metodología se realiza de forma incremental e iterativa, por ser una metodología que utiliza el prototipo como herramienta fundamental. [13]

1.6 Sistemas Gestores de Base de Datos.

Los Sistemas Gestores de Bases de Datos son un tipo de software específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan.

Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD (Sistema Gestor de Bases de Datos) y DBMS (DataBase Management System).

1.6.1 Microsoft SQL⁹

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL. Pone a disposición de muchos usuarios gran cantidad de datos de manera simultánea.

Entre sus características más importantes figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor, teniendo acceso a la información solamente las terminales o clientes de la red.

⁹ Lenguaje de Consulta Estructurado. SQL por sus siglas en inglés Structured Query Language.

Capítulo 1: Marco Teórico

- Permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños.

Desventajas

- Microsoft SQL Server no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

1.6.2 PostgreSQL

PostgreSQL es un motor de base de datos relacional libre, liberado bajo la licencia BSD, la cual permite el uso del código fuente en software no libre.

Entre sus principales características podemos citar:

- Alta concurrencia
- Amplia variedad de tipos nativos
- Llaves ajenas o Llaves Foráneas (*foreign keys*).
- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL.

PostgreSQL provee soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.

Capítulo 1: Marco Teórico

- Arrays.

1.6.3 ORACLE

Oracle es un sistema de gestión de base de datos fabricado por Oracle Corporation¹⁰, es el mayor y más usado Sistema Manejador de Base de Dato Relacional (RDBMS) en el mundo.

Este sistema gestor de Base de Datos corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Oracle soporta dos tipos de almacenamiento, por carácter (RAW) o por bloques (Files System), generalmente es recomendable que los sean colocados en Raw Device¹¹.

Es considerado además como uno de los sistemas de bases de datos más completos que existe, destacándose entre sus principales características:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.
- Almacena los códigos de los programas para empezar a ejecutarse.
- Almacena los datos necesarios durante la ejecución de un programa.
- Almacena información sobre cómo es la transferencia entre procesos y periféricos.

¹⁰ Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer, o JDeveloper.

¹¹ Raw Device es un dispositivo de caracteres disponibles en algunos sistemas operativos el cual es asignado directamente a Oracle.

Capítulo 1: Marco Teórico

- Soporta bases de datos de todos los tamaños, desde severas cantidades de bytes hasta gigabytes en tamaño.
- Oracle soporta un verdadero ambiente cliente servidor.

1.7 Estilos de dirección

Un estilo de dirección es la manera en que los gerentes interactúan con las personas bajo su mando. Existen tres estilos básicos que figuran en la bibliografía más usual de Gestión de Proyectos:

- El Autocrático.
- El Permisivo.
- El Democrático.

Para hacer un análisis de estos estilos y obtener sus diferencias, ventajas y desventajas se analizan en función de los flujos de información.

Estilo Autocrático

En este estilo los gestores de proyecto no se preocupan por procesar la información proveniente de afuera ni la retroalimentación que viene del equipo. En ocasiones sostienen una política de puertas abiertas, pero solo en la medida en que buscan la información del exterior, porque después no la tienen en cuenta, son autocráticos.

Ventajas

Este estilo es adecuado para proyectos de bajo riesgo, en los que el equipo se limita a llevar el plan como fue establecido, en este caso la retroalimentación no es tan decisiva como en un proyecto de alto riesgo.

Puede resultar eficaz cuando es preciso tomar decisiones rápidamente.

Desventajas

Puede llevar a la desmoralización del equipo, pues los desarrolladores se ven excluidos al no poder contribuir a la toma de decisiones.

Capítulo 1: Marco Teórico

Puede llevar a tomar decisiones erróneas, producto de que el jefe no obtiene suficiente información del exterior.

Estilo Permisivo

En este estilo la toma de decisiones es muy difusa, se basa en “dejar hacer”. Existe un escaso flujo de información o un flujo aleatorio, que no se canaliza adecuadamente. En un estilo permisivo el equipo de proyecto debe ser capaz de realimentar al equipo de gestión pero los directivos no suelen obrar adecuadamente con respecto a esta realimentación. El estilo autocrático y el permisivo comparten un rasgo común: en ninguno de los dos la información fluye desde el equipo de trabajo hacia los directores.

Ventajas

Puede ser eficaz en proyectos innovadores, en los que se promueve la creatividad. La libertad de acción suele levantar la moral del equipo de desarrollo.

Desventajas

Puede llevar a la desorganización por falta de dirección y resulta desastroso cuando se precisa una toma de decisión rápida.

Estilo Democrático

En este estilo los gestores antes de tomar decisiones, tratan de recibir información originada en el equipo. Generalmente este estilo es el más eficaz.

Ventajas

Posibilita la toma de buenas decisiones, porque refleja varios puntos de vista.

Aumenta el compromiso del equipo, ya que se saben partícipes en la toma de decisiones.

Desventajas

Puede ser inútil cuando se precisa la toma de decisiones rápidas.

Difícilmente podrá ser aplicado en sitios en los que ni siquiera conocen la faceta humana y política de la democracia [9].

Capítulo 1: Marco Teórico

1.8 Organigramas de Equipos

Mantei [17] sugiere tres organigramas de equipo genéricos:

- Descentralizado Democrático (DD)
- Descentralizado Controlado (DC)
- Centralizado Controlado (CC).

Descentralizado Democrático (DD)

Este tipo de equipo no posee un jefe permanente. Se nombran coordinadores de tareas a corto plazo y se sustituyen por otros para diferentes tareas. Las decisiones sobre problemas y los enfoques se hacen por consenso del grupo. La comunicación entre los miembros del equipo es horizontal.

Descentralizado controlado (DC)

Este tipo de equipo tiene un jefe definido que coordina tareas específicas y jefes secundarios que tienen responsabilidades sobre subtareas. La resolución de problemas sigue siendo una actividad del grupo, pero la implementación de la solución se reparte entre subgrupos por el jefe de equipo. Aunque la comunicación dentro de los subgrupos es básicamente horizontal, también hay comunicación vertical a lo largo de la jerarquía de control.

Centralizado Controlado (CC)

Tienen un jefe de equipo que se encarga de la resolución de problemas de alto nivel y la coordinación interna del equipo, mientras la comunicación entre el jefe y los miembros del grupo es vertical.

1.9 Repositorio de componente

Componente

Un componente de software es un elemento de software que se ajusta a un modelo de componentes y que puede ser independientemente desplegado y compuesto sin modificación, de acuerdo con un estándar de composición [14]

Capítulo 1: Marco Teórico

El repositorio de componentes constituye un almacén de componentes reutilizables que requiere ser mantenido y gestionado constantemente.

1.10 Herramientas de Gestión de Proyecto.

1.10.1 Microsoft Project 2003

Ventajas

- Provee facilidades para planificar, programar y controlar el proyecto.
- Permite el control en diferentes niveles de acuerdo a las necesidades del cliente.
- Permite la generación de diversos tipos de diagramas como son el de Gantt y Pert.
- Mejora la capacidad de organización del trabajo y se obtiene una comunicación eficaz entre los equipos de desarrollo del software.
- Permite que cada uno de los trabajadores cheque su estado de desarrollo y conozca el día que su trabajo será revisado.
- Permite ofrecer una panorámica real y actualizada al cliente sobre el software en general.
- En cuanto a seguridad ofrece el uso de contraseñas si el proyecto lo necesita para evitar que personal ajeno realice cambios.

Desventajas

- Es un software propietario.
- Puede no ser óptimo para proyectos muy grandes (de más de 100 tareas) pues resulta casi imposible mostrar en pantalla el desarrollo completo de los mismos.

1.10.2 Project KilckStart

Ventajas

- Provee facilidades para planificar, programar y controlar el proyecto.
- Se puede utilizar en proyectos de cualquier tamaño, hasta 1000 tareas y 100 recursos.

Capítulo 1: Marco Teórico

- Siete tipos de informes preestablecidos.
- Listado de sugerencias en los archivos de Objetivos, fases y Obstáculos, que usted puede arrastrar con el ratón y colocar en su proyecto.
- Gráfica de Gantt para programación "completa".
- Enlace dinámico con Word, Outlook, PowerPoint y Excel para incluir sus proyectos en la planeación de sus propuestas y planes de negocio.
- Se puede archivar como HTML.

Desventajas

- Software propietario.

1.11 Herramientas de Gestión de la Configuración.

1.11.1 Concurrent Version System (CVS)

Current Versions System (CVS) es un sistema para el control de versiones que se ha hecho muy popular en el mundo del software libre pues e difundido bajo la licencia GPL por sus desarrolladores. Es empleado en proyectos comerciales y prácticamente en todos lo proyectos de código abierto.

Es un software Open-Source que permite gestionar los cambios realizados sobre códigos fuentes de cualquier archivo en varias plataformas (C, C++, Java, etc.).

Ventajas:

- Sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- Maneja eficientemente archivos binarios.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.

Capítulo 1: Marco Teórico

- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP¹², PAM¹³, etc.).

Desventajas:

- El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.
- No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita el llevar la cuenta de qué cambios se han trasladado. Esto se resuelve siendo cuidadoso con los mensajes de commit.

1.11.2 Subversion (SVN)

Subversion es un sistema de control de versiones de código abierto, de propósitos similares al bien conocido, ampliamente extendido CVS. Está diseñado para proporcionar un sofisticado sistema de control de versiones, desarrollado con tecnología moderna [17].

Ventajas

- Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.
- La creación de ramas y etiquetas es una operación más eficiente; tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).

¹² LDAP por sus siglas en inglés Lightweight Directory Access Protocol, es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.

¹³ PAM, por sus siglas en inglés Pluggable Authentication Module es un mecanismo que proporciona una interfaz entre las aplicaciones de usuario y diferentes métodos de autenticación.

Capítulo 1: Marco Teórico

- Permite el acceso al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores, fomentando la colaboración.
- Si ocurre un cambio no deseado en los datos, da la posibilidad que se deshaga el ese cambio.
- Es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros.
- Ayuda al seguimiento de cambios en los ficheros de código fuente de un proyecto.
- Es fácil de instalar.
- A diferencia de CVS, hace las revisiones globales del repositorio, quiere decir, que un cambio en un fichero no introduce un salto de versión únicamente en este fichero, sino en todo el repositorio.

Desventajas

- Aún tiene problemas con merges de archivos movidos
- No soporta Win32

1.12 Instituto Politécnico de Informática “Abel Santamaría”.

El Politécnico de Informática “Abel Santamaría” es una institución creada al calor de la Batalla de Ideas, donde la tecnología y los conocimientos se ponen en pos del total cumplimiento de los objetivos a alcanzar:

- La formación de profesionales altamente calificados, comprometidos con la revolución, en la rama de la informática.
- Llevar la informatización a todos los sectores de la sociedad: Salud, Educación, Cultura, Deporte, Turismo, Prensa, etc.
- Convertir la industria del software cubana en un renglón fundamental de la economía e insertarnos en el mercado internacional.

Capítulo 1: Marco Teórico

Conclusiones

Una vez terminado el presente capítulo se concluye diciendo que se realizó un exitoso estado del arte, donde se citaron definiciones dadas por diferentes autores al concepto de Factoría y Factoría de Software, se estableció un concepto de Factoría de Software para una mejor comprensión de la presente investigación, siendo el concepto que se maneje a lo largo de todo el trabajo, se definió el concepto de Instituto Politécnico de Informática y algunos de sus objetivos principales. Se realizó con resultados satisfactorios la investigación de una gama de Modelos de Factoría de Software, analizándose también un Modelo de Factoría de Software creado específicamente para una línea de producción de la Universidad de las Ciencias Informáticas. Se analizaron también Modelos que son categorizados como Estándares de Calidad de Software, Herramientas de Gestión de Software y de Gestión de la Configuración, diferentes Metodologías, Estilos de Dirección y Organigramas.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Introducción

Dada la situación problemática planteada en el Capítulo anterior y después de hacer un análisis de los diferentes Modelos de Factoría de Software más representativos, según la bibliografía consultada, se propone el Modelo Organizativo para Proyectos Productivos (MOPP) para lograr la organización de los proyectos productivos en el Instituto Politécnico de Informática “Abel Santamaría” y de este modo dar solución a las dificultades relacionadas con la producción existente en dicho centro.

Este Modelo fue creado a partir de las características esenciales y moldeables de los siguientes modelos: Modelo Eureka, Modelo propuesto por Basili, Modelo Replicable y el Modelo Funcional de la UCI para la Línea de Carrefour, en su confección también intervinieron modelos que constituyen Estándares de la Calidad de Software como son, CMM y CMMI. Herramientas de Gestión de Software, Herramientas de Configuración del Software, Metodologías para el desarrollo del Software utilizadas a nivel mundial, entre otro bibliografía que se tubo en cuenta para la confección de dicho modelo de Factoría de Software.

2.1 Por qué la adopción de MOPP. Particularidades de la producción del Instituto Politécnico de Informática.

El motivo principal por el cual no se implementa uno de los modelo de Factoría de Software analizados anteriormente para la organización de los proyectos productivos en el Politécnico, está dado debido a que por lo general estos modelos están diseñados para grandes fábricas productoras de software, que poseen un alto nivel de desarrollo en la rama, experiencia y prestigio, que cuentan además con tecnología que avala dicho desarrollo; sin embargo, lo que se necesita en el IPI es un modelo que se adapte a las particularidades de un Instituto de enseñanza y producción que está surgiendo en estos momentos como productor de

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

software, con una matrícula pequeña, inexperta, un claustro de profesores que en su mayoría no son de la especialidad y los que lo son tienen muy poca experiencia en lo que respecta al desarrollo de software.

La ventaja que trae la creación de un Modelo para la organización de los proyectos productivos en el Instituto Politécnico de Informática está precisamente en que este Modelo ha sido diseñado sobre la base de las dificultades que están presentes hoy en día en la Institución, está implementado para satisfacer sus necesidades y para suprimir sus dificultades.

2.2 Identificación de los elementos del modelo

En la confección del Modelo Organizativo para Proyectos Productivos del Instituto Politécnico de Informática “Abel Santamaría” se han identificado cuatro áreas fundamentales y las relaciones que se establecen entre ellas. Estas áreas engloban todo el trabajo que se lleva a cabo en la Factoría y se han nombrado de la siguiente manera:

1. Área Técnicas y Herramientas: Comprende el contexto de las tecnologías, herramientas y métodos para dar soporte y automatización al proceso de desarrollo.
2. Área Proceso de desarrollo: Comprende el conjunto de actividades que conforman el flujo de trabajo, el cual consta de las siguientes fases: Captura de Requisitos, Análisis, Diseño, Implementación, Prueba y Entrega.
3. Área Trabajadores: Comprende los recursos humanos involucrado en el proceso de desarrollo de software, la estructura organizativa y los roles que ocupan cada uno de ellos, está dividida en dos Sub-Áreas: Gestores de la Factoría y Grupo de desarrollo.
4. Área Repositorio de Componentes: Esta compuesto por componentes reutilizables, los cuales se han clasificados en dos grupos: Activos del proceso, que también pueden llamarse Componentes de Infraestructura, que son componentes de valor en el proceso y Componentes de Código.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

En la figura 2.1 se muestran las Áreas del Modelo y las relaciones que se establecen entre ellas.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

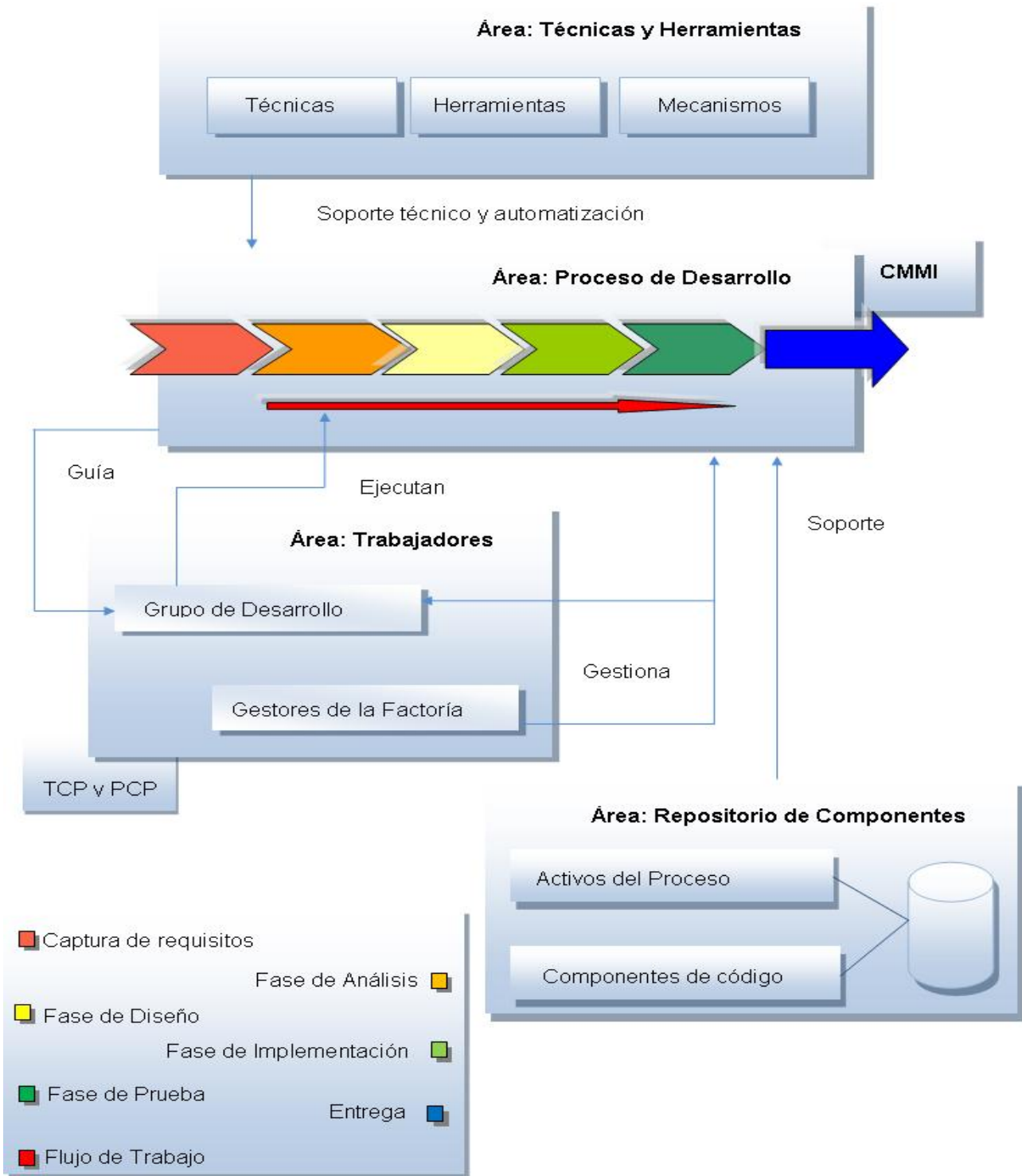


Figura 2.1_ Estructura del Modelo Organizativo para Proyectos Productivos en el IPI. "Abel Santamaría".

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

El resultado final de un proyecto en la Factoría del Politécnico será un producto de software de calidad, que toma forma gracias a la intervención de personas que juegan diferentes roles dentro de la Factoría, este recurso humano está representado en el Área Trabajadores que a la vez se divide en dos Sub-Áreas, Grupo de desarrollo y Gestores de la Factoría. El Grupo de desarrollo lo forman las personas involucradas directamente en el proceso de software, las cuales ejecutan las actividades o flujos de trabajo (Captura de Requisitos, Análisis, Diseño, Implementación, Prueba y Entrega), a su vez este equipo es guiado por un proceso de desarrollo de software, representado en el modelo en el Área Proceso de Desarrollo. La Sub-Área Gestores de la Factoría comprende el equipo de dirección de la Factoría, son las personas encargadas del control y gestión del Grupo de Desarrollo y del proceso de desarrollo del Software.

El modelo propone que el Área Trabajadores haga uso del PSP (Personal Software Process) y TSP (Team Software Process) con el objetivo de lograr un mayor rendimiento de las personas implicadas en el desarrollo del software, tanto de forma personal como a la hora de formar equipos.

El proceso es automatizado y soportado por diversas tecnologías, herramientas y mecanismos representados en el Área Técnicas y Herramientas, que a su vez se divide en tres Sub-Áreas:

- Sub-Área Técnicas: Comprende las técnicas a utilizar durante todo el proceso de desarrollo del software.
- Sub-Área Herramientas: Se localizan las herramientas, tecnologías y software sobre los cuales se implementa el proceso de desarrollo del Software.
- Sub-Área Mecanismos: Encierra todos aquellos mecanismos necesarios en el proceso de desarrollo del Software.

La reutilización tiene efectos positivos en el desarrollo del software, entre estos beneficios están el aumento en la productividad y calidad del software así como la reducción del tiempo

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

de desarrollo. La Factoría cuenta, para dar soporte al proceso de desarrollo del Software, con una base de componentes reutilizables, representada en el Área Repositorio de Componentes.

El proceso de desarrollo del Software es regulado por normas de calidad dando los primeros pasos hacia CMMI, puesto que es prioridad la obtención de productos con calidad, además CMMI es un modelo de calidad para la industria del software, que provee detalles importantes para la evaluación del proceso de desarrollo y la gestión de proyectos.

2.3 Área Trabajadores

Hay una serie de personas implicadas durante todo el proceso de desarrollo de software. La Factoría presenta una estructura organizativa donde cada persona implicada ocupa un rol determinado dentro de la misma en dependencia de sus conocimientos y capacidades. Las personas constituyen un factor importante en el éxito de un proyecto de software por lo que su organización resulta fundamental.

2.3.1 Distribución de roles en la factoría de software.

Un tema que se hace imprescindible en la implementación de la Factoría del Politécnico de Informática “Abel Santamaría” es la distribución de los roles a los trabajadores, pues es importante que el rol que se le asigna a un determinado trabajador este acorde a sus conocimientos y capacidades.

2.3.1.1 Sub-Área Gestores de la Factoría

En el caso de la Sub-Área Gestores de la Factoría no se hará una captación para desempeñar un determinado rol, simplemente se asignaran a personas con experiencia en el tema de dirección, como es el caso de los jefes de departamento o subdirectores de la Institución, la captación como tal se implementara en la Sub-Área Grupo de Desarrollo.

Los roles pertenecientes a la Sub-Área Gestores de la Factoría y las personas designadas son:

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Gestor de la Factoría: Es el encargado de dirigir y apoyar las distintas líneas de desarrollo de la factoría. Debe ser capaz de, en conjunto con la dirección de cada línea de desarrollo, estimar plazos, costos y riesgos para la realización de un determinado producto software, estableciendo las relaciones necesarias con el cliente. Además controlar el desarrollo de los distintos productos, debe verificar y tener noción de la fase en que se encuentran. Administrará con economía y eficacia los distintos recursos y procesos de la factoría, debe organizar el trabajo de forma tal que las personas bajo su mando cumplan su rol adecuadamente.

Persona designada: Profesor Jefe del Departamento de Programación.

En la figura 2.2 se muestra la jerarquía existente en la entre el Gestor de la Factoría, el Jefe de Línea de Desarrollo y el Jefe de Grupo de Asesoría y Servicios

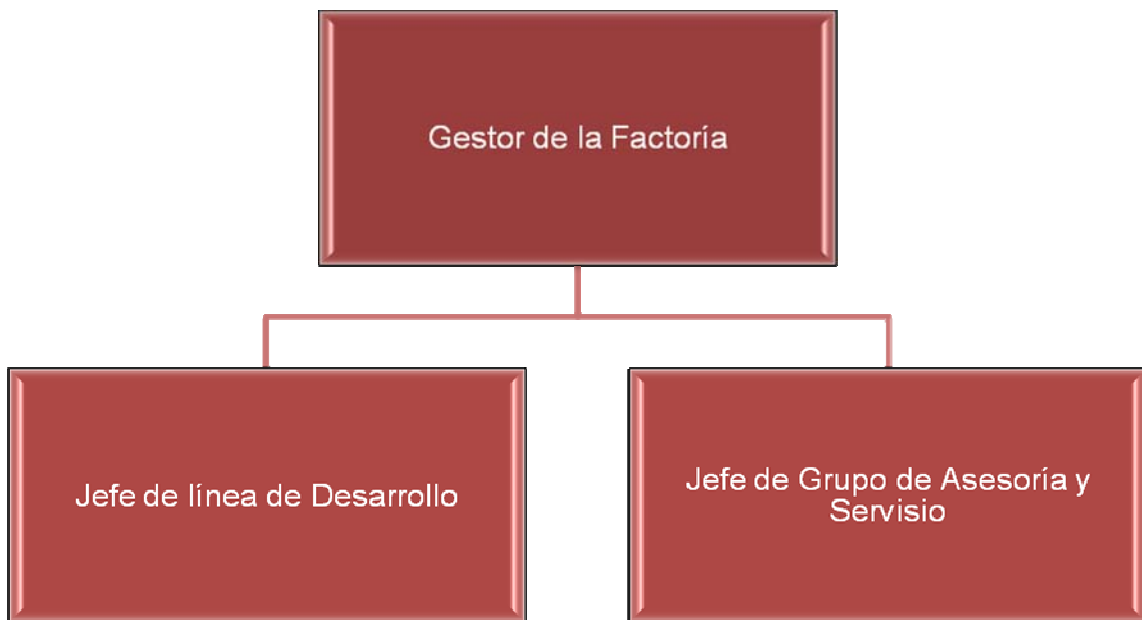


Figura 2.2_ Jerarquía de la Factoría de Software del Politécnico de Informática “Abel Santamaría”

Por cada línea de desarrollo dentro de la factoría habrá un equipo de dirección de la misma donde se identifica el siguiente rol:

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Jefe de Línea de Desarrollo: Es el encargado de dirigir el desarrollo de productos software en una línea determinada dentro de la Factoría. Tiene la tarea de velar por que los productos en su línea de desarrollo se realicen a tiempo y según las especificaciones establecidas por el cliente, controlando su estado de ejecución. Tiene bajo su mando a uno o varios equipos de desarrollo, es el intermediario entre estos y el jefe de la Factoría, constituye el canal a través del cual fluye la información desde los directivos hacia el equipo de desarrollo.

Persona designada: Profesor de Programación.

En la figura 2.3 se muestra la jerarquía que existe con relación al Jefe de Línea de Producción y los diferentes Jefes de Proyecto.

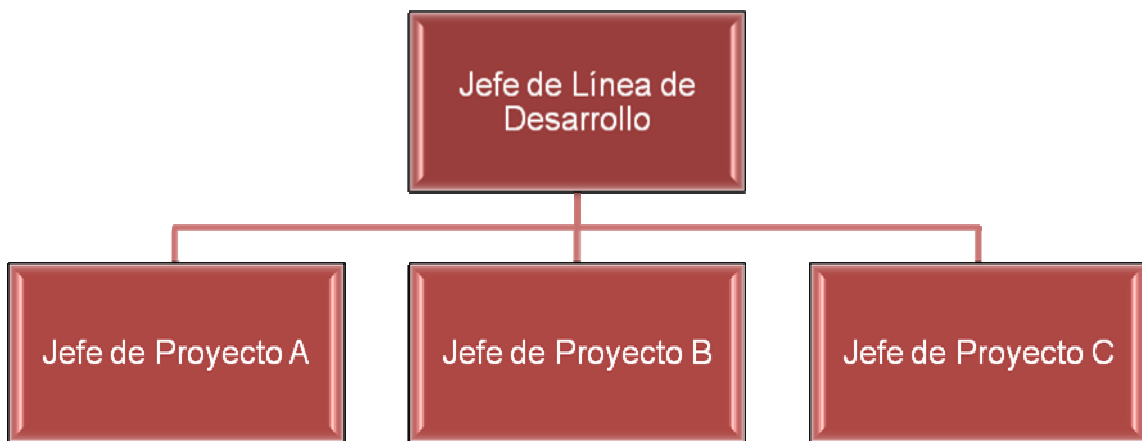


Figura 2.3_ Jerarquía entre el Jefe de línea de Desarrollo y los diferentes Jefes de Proyectos.

Además la Factoría contará con un grupo de soporte que tendrá como función brindar asesoría y servicio a las distintas líneas de desarrollo, en este se identifican los siguientes roles:

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Jefe de Grupo de Asesoría y Servicio: Guía y apoya al grupo de soporte de la Factoría, responde por este grupo ante la dirección de la misma. Es el encargado de identificar las necesidades de cada línea de desarrollo con el fin de poner a los especialistas del grupo en función de las mismas.

Persona designada: Jefe del Departamento de Proyecto Informático.

Administrador del Repositorio de Componentes Reutilizables: Es el responsable de gestionar el repositorio de componentes reutilizables de la Factoría.

Persona designada: Profesor Jefe del Departamento de Grafico.

Investigador: Encargado de realizar cualquier investigación, búsqueda de información en un tema determinado solicitado por cualquier línea de desarrollo. Desarrollar herramientas de apoyo a la programación y a los procesos de desarrollo con el objetivo de aumentar la productividad y la calidad en el trabajo, automatizando y optimizando los procesos, también se encargará de la capacitación de los trabajadores en la Factoría.

Persona designada: Estudiante aventajado (con más de 95 puntos de promedio general).

Jefe de Captación: Tiene la tarea de trazar una estrategia de captación y llevar a cabo la misma.

Persona designada: Subdirector docente del IPI..

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)



Figura 2.4_ Jerarquía entre el jefe de Grupo de Asesoría y Servicios, el Jefe de Capacitación, Administrador del Repositorio de Componentes y el Investigador.

2.3.1.2 Sub-Área Equipo de desarrollo.

En el caso del Equipo de Desarrollo se necesita una captación para designar qué rol desempeñará cada uno de los involucrados, el hecho de que la Factoría de Software del IPI se esté organizando actualmente hace necesario que se adopte una forma rápida para la captación de los trabajadores, sumándole a ello que este grupo va a estar integrado en su mayoría por estudiantes que no tienen experiencia en las tareas a realizar, es por ello que se implementará el siguiente proceso de captación:

Proceso de captación para la Sub-Área Equipo de desarrollo:

Comenzará la captación de trabajadores para un grupo piloto de dos proyectos, aproximadamente 60 estudiantes de tercer año, debido a que son los que llevan un número mayor de cursos en la escuela y por tanto los que más conocimientos han recibido, sumándole a ello el hecho de que es el año culminante y que parte de su trabajo de tesis puede ser el resultado de su desempeño e interesa en el proyecto.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Primeramente se hará una selección de los estudiantes con más de 90 puntos de índice académico general y con más de 90 puntos en las asignaturas de Proyecto Informático y Programación I.II.III.

Después de elegir a los estudiantes que cumplan con los requisitos anteriormente expuesto, se hace un escalafón, a los candidatos se les entrega una planilla (Ver Anexo 11) en forma de encuesta en la que podrán seleccionar el rol que desean desempeñar dentro del proceso de desarrollo.

Una vez que los estudiantes llenen la planilla debe ser entregada al Jefe de Captación quien otorga los roles teniendo en cuenta el escalafón, todo este proceso es necesario ya que es mayor la demanda por rol que la oferta, a los estudiantes se les hace llegar su rol por correo electrónico, citándolos para una reunión donde se les informará acerca de su trabajo en la Factoría.

A medida que la Factoría gane en madurez y el número de proyectos aumente se irán incorporando estudiantes de segundo año, con 85 puntos de promedio general, manteniendo los 90 puntos de promedio en las asignaturas de Proyecto Informático y Programación, la asignación de roles no sería por escalafón sino que se realizarían una serie de pruebas de nivel (Programación, Base de Datos, Ingeniería de Software) y según los resultados se seleccionarían a los estudiantes más capaces.

Los roles de esta Sub-Área se definen en el Área Proceso de Desarrollo.

2.4 Organización de los trabajadores en la Factoría.

Para la elección del organigrama de la factoría se realizó una valoración de los organigramas expuestos en el Capítulo 1, esta valoración arrojó como resultado que:

Los grupos descentralizados generan más y mejores soluciones que los individuales, por lo tanto estos equipos tienen más probabilidades de éxito en la solución de problemas complejos.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Partiendo de que el rendimiento de un equipo es inversamente proporcional a la cantidad de comunicación que se debe entablar, se puede plantear que los proyectos muy grandes son guiados mejor por grupos con estructura CC o DC, donde se puedan formar subgrupos fácilmente.

Cuando la modularidad es relativamente baja se aplica mejor el organigrama DD y cuando es posible una modularidad alta funcionan bien los organigramas CC o DC, además de que producen menos defectos que los equipos DD.

Hay que tener en cuenta que los equipos descentralizados requieren más tiempo para completar un proyecto que los centralizados y al mismo tiempo son mejores cuando se precisa una gran cantidad de comunicación.

Podemos ver que una estructura centralizada realiza las tareas más rápidamente, es la más adecuada a la hora de solucionar problemas sencillos en los cuales el equipo domina el problema a resolver, las tecnologías y herramientas a utilizar. Teniendo en cuenta el estado en que se encuentra la Factoría esta estructura organizativa no sería la más adecuada para aplicar en el Politécnico pues no existe una basta experiencia en el uso de las tecnologías y herramientas con que se trabajan, en un futuro no muy lejano cuando las condiciones estén creadas lo más seguro es que pueda optar por aplicar esta estructura de equipo en el desarrollo.

Por tanto el organigrama que se propuso para la organización de los trabajadores en el Politécnico de Informática “Abel Santamaría”, teniendo en cuenta las características de dicho centro y lo que se quiere lograr, fue el organigrama Descentralizado Controlado (DC).

La puesta en práctica de este organigrama requiere de múltiples conocimientos, por lo que se hace necesario reunir a personas con los conocimientos y habilidades necesarios y organizarlas en equipos de manera que puedan enfrentar el reto de desarrollar aplicaciones con éxito.

Cada equipo estará conformado de 8 a 10 trabajadores, posibilitando un mejor desempeño y cohesión del grupo de trabajo así como una buena interacción entre los integrantes, una vez

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

que los integrantes comiencen a comunicarse entre si da lugar a un aumento de confianza e interacción dentro del grupo, también las discusiones comienzan a centrarse más específicamente en las tareas de resolución de problemas y en el desarrollo de estrategias para cumplir las tareas.

En la figura 2.5 se muestra como quedaría organizada la Factoría una vez implantado el Organigrama Descentralizado Controlado. El Jefe Definido es el Gestor de la Factoría, el grupo de Jefes Secundarios esta dividido en dos sub-grupos, el Jefe de Línea de Producción y el Jefe de Grupo de Asesoría y Soporte. El administrador del Repositorio de Componentes, el Jefe de Captación y el Investigador son los subordinados del Jefe del Grupo de Asesoría y Soporte.

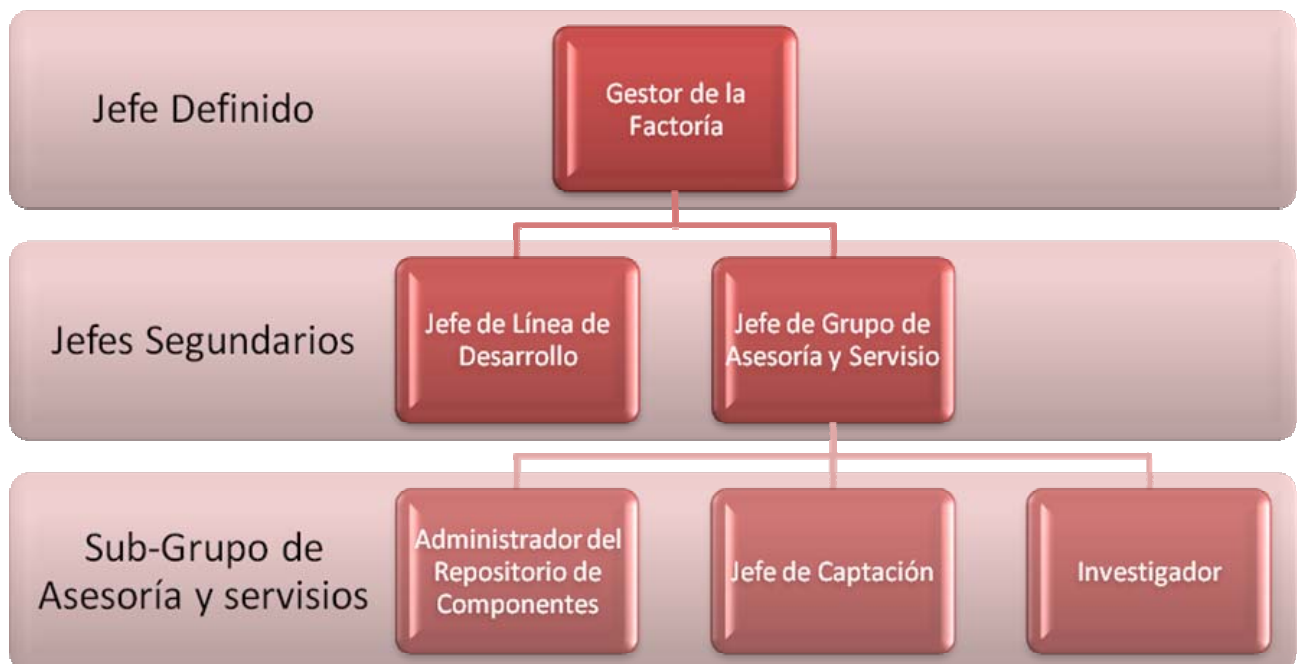


Figura 2.5_ Relaciones de comunicación entre los directivos y sus subordinados una vez aplicado el Organigrama Descentralizado Controlado en el IPI..

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

2.5 Estilo de dirección

Teniendo en cuenta que el estilo de dirección empleado tiene gran influencia sobre los resultados para la elección del estilo de dirección a implementar en el Politécnico de Informática “Abel Santamaría” se valoraron una serie de aspectos claves, como son:

- El IPI. es un centro de enseñanza donde los profesores que conviven y que conocen a los estudiantes (Capital Humano) pasan a ser, no en su totalidad, los directivos de la Factoría de Software.
- Los trabajadores son jóvenes de mucha creatividad y con muchos deseos de trabajar.
- La Factoría del Politécnico es una fábrica que está surgiendo actualmente .

Sobre la base de esta valoración MOPP propone la implantación del Estilo de Dirección Democrático para la factoría del IPI., puesto que en este Estilo de Dirección los directivos tratan siempre de recibir información originada por el equipo, permite la toma de buenas decisiones porque refleja varios puntos de vista, aumenta el compromiso del equipo con los directivos, ya que se saben partícipes en la toma de decisiones, se aprovecha al máximo la creatividad de los participantes ayudando así a la retroalimentación de todos, incluyendo a los directivos, que poseen poca experiencia en el tema.

En la figura 2.6 se muestra como queda implementado el estilo de dirección Democrático en el Instituto Politécnico de Informática “Abel Santamaría”.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

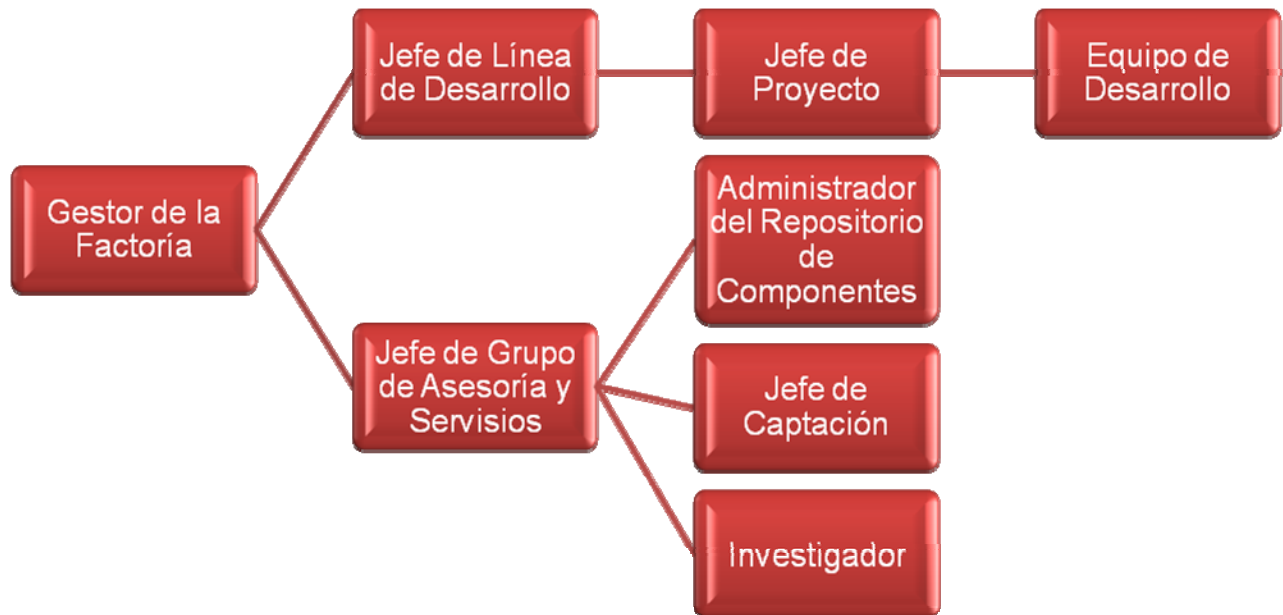


Figura 2.6_ Relaciones de comunicación que se establecen entre los directivos y sus subordinados una vez aplicado el Estilo de Dirección Democrático en el IPI.

2.6 El PSP (Proceso del Software Personal) como instrumento para alcanzar la calidad en el trabajo de los participantes.

Teniendo en cuenta que el personal de la Factoría del Politécnico es joven e inexperto en el desarrollo de software, el PSP ayuda en gran magnitud al trabajo organizativo de cada individuo en la factoría.

Sumándole a las responsabilidades del desarrollo del software, los trabajadores también tendrán la tarea de aprender a planificarse con eficiencia y a cumplir sus propios objetivos como seres independientes.

Consecuentemente este modelo propone algunos modelos referentes al tema, sin pretender englobar todo el trabajo organizativo individual, teniendo como objetivo incentivar, dar a conocer la importancia de estas herramientas y conducir los primeros pasos a la planificación personal.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

2.6.1 Cuaderno del Ingeniero.

El Cuaderno de ingeniero es un libro que debe de llevar personalmente cada uno de los involucrados en la factoría, donde puede:

- Registrar tiempos.
- Guardar cálculos.
- Tomar notas de las fases del desarrollo
- Registrar además cualquier tipo de información que afecte el horario de laboral en la factoría, teniendo en cuenta que quienes trabajan en ella tienen responsabilidades adicionales al desarrollo de software.

Al registrar las buenas ideas en cualquiera de las fases del desarrollo, los directivos y asociados al equipo podrán reconocer el merito alcanzado.

Una sugerencia se puede observar en el Anexo 16 y Anexo 17.

En la figura 2.6 se muestra las relaciones que se establecen entre el Trabajador, el Cuaderno del Ingeniero y el Jefe de Proyecto.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

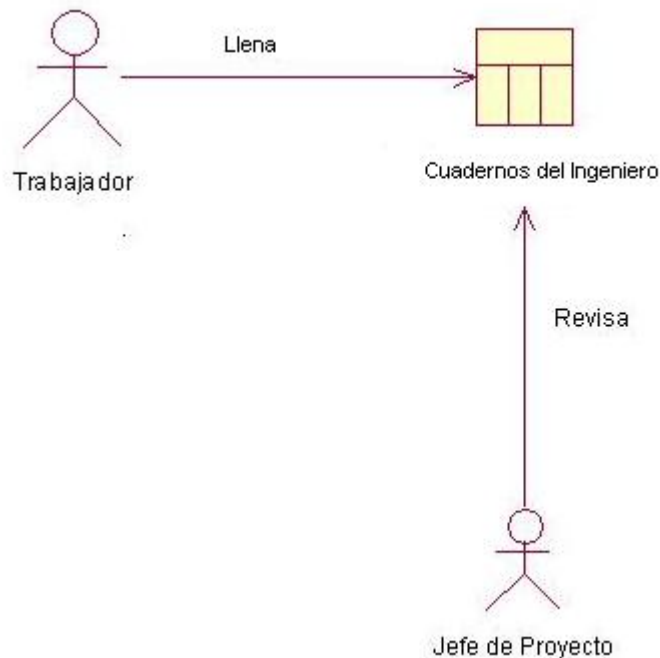


Figura 2.6_ Interacción del trabajador y el Jefe de Proyecto con el Cuaderno del Ingeniero.

2.6.2 Cuaderno de Registro de Tiempo.

El cuaderno de registro de tiempo tiene gran utilidad puesto que:

- Se registran diariamente las actividades realizadas en el día.
- Se contabilizan y describen las interrupciones ocurridas durante la jornada laboral, dejando almacenado el historial de las mismas.
- Ayuda al trabajador a controlar la duración y frecuencia de las interrupciones.
- Los directivos tienen la posibilidad de observar los comportamientos de los trabajadores y realizar análisis al respecto. Intentado evitar el aumento de los tiempos de interrupciones.

En la figura 2.7 se muestra las relaciones que se establecen entre el Trabajador, el Cuaderno de Registro de Tiempo y el Jefe de Proyecto.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

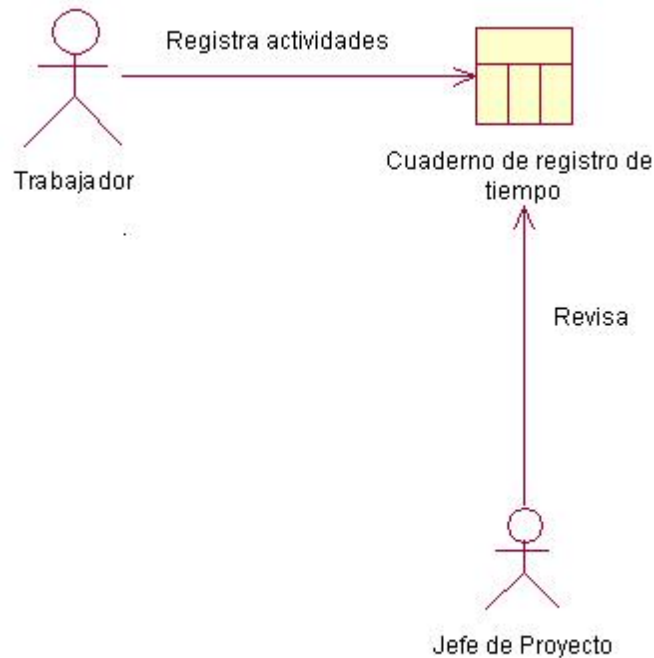


Figura 2.7_ Interacción del trabajador y el Jefe de Proyecto con el Cuaderno de registro del tiempo. El diseño del cuaderno se puede observar en el Anexo 18.

2.6.3 Cuaderno de Resumen Semanal de Actividades.

El cuaderno de resumen semanal de actividades es de mucha utilidad en la factoría pues los trabajadores agrupan en él las actividades que realizan diariamente, obteniendo totales de tiempos relacionados con los trabajos efectuados diariamente y los efectuados en la semana en general.

Para los directivos esta información es necesaria a la hora de planificar actividades relacionadas con los horarios laborales establecidos.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

En la figura 2.8 se muestra las relaciones que se establecen entre el Trabajador, el Cuaderno de resumen semanal de actividades y el Jefe de Proyecto.

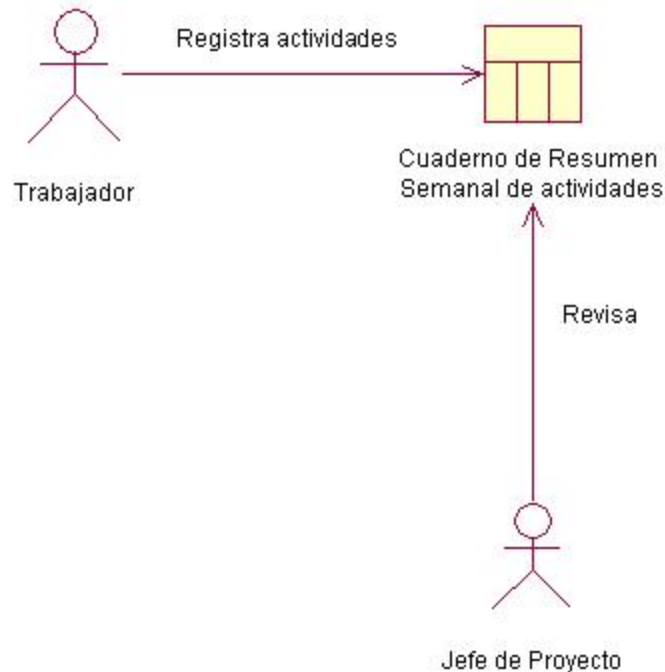


Figura 2.8 Interacción del trabajador y el Jefe de Proyecto con el Cuaderno de registro semanal de actividades.

En el Anexo 19 se puede encontrar el modelo de Registro semanal de actividades.

2.6.4 El Cuaderno de Trabajo.

Este cuaderno se diseñó para registrar datos de tiempos estimados y reales. Es un documento de planificación de producto que brinda dos grandes beneficios:

- Permite el cálculo de la velocidad media de un individuo para una tarea dada, de esa forma los trabajadores de la factoría pueden comenzar a estimar su velocidad mínima, media y máxima ante las tareas a realizar en dependencia del tipo que sean.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

- Luego de conocer la velocidad individual de cada individuo, el alcance de los proyectos y acumular cierta experiencia; los directivos podrían hacer planificaciones de tareas de mayor magnitud, estimando e informando fechas de entrega a los clientes con mayor exactitud.

En la figura 2.9 se muestra las relaciones que se establecen entre el Trabajador, el Cuaderno de Trabajo y el Jefe de Proyecto.

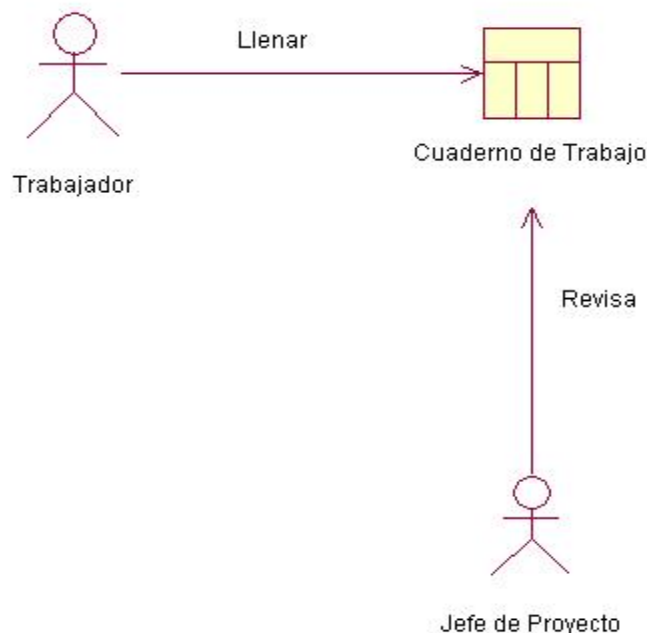


Figura 2.9_ Interacción del trabajador y el Jefe de Proyecto con el Cuaderno de trabajo.

El modelo está representado en el Anexo 20.

Existen muchas otras técnicas de PSP que pueden soportar el desempeño de los implicados en la factoría en general, pero inicialmente lo pretendido en el modelo productivo planteado es el aprendizaje de la estimación del tiempo necesario para lograr objetivos. Una vez logrado esto todas las demás cuestiones de planificación se lograrán con mayor facilidad.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

2.7 Proceso de Desarrollo del Software.

El objetivo fundamental de esta Área es elaborar normas, definir roles, artefactos y herramientas que guíen en cada fase el proceso productivo, para el desarrollo del software.

Después de realizar un análisis de algunas de las metodologías de desarrollo existentes, este modelo adoptó seguir durante todo el proceso de desarrollo del software la Metodología RUP, sintetizando este gran proceso a los aspectos claves que pueden ser útiles en el desempeño de la factoría, donde las fases de desarrollo serían: Captura de Requisitos, Análisis, Diseño, Implementación, Pruebas y Entrega.

Para cada fase del proceso de desarrollo serán válidos artefactos, flujo de trabajo y actividades fundamentales que especifica RUP, solamente habrá una variación en lo que respecta a los roles, con el objetivo de lograr un mayor acercamiento a las necesidades del IPI.

2.8 Área Técnicas y Herramientas

El proceso de desarrollo de software está basado en herramientas, métodos y mecanismos. En el momento de seleccionar las herramientas para trabajar en una determinada fase de desarrollo, se deben tener en cuenta una serie de aspectos, al igual que al diseñar las metodologías que optimicen los procesos.

En la organización de una Factoría de Software debe conocerse fundamentalmente que utilidad presenta cada herramienta, así como el consumo de recursos de la misma en la estación de trabajo. Además se debe crear una cultura de cumplimiento a totalidad de todas las normas disciplinarias y técnicas especificadas por los directivos.

Las herramientas que han sido seleccionadas por MOPP para brindar soporte al proceso de desarrollo del software en el Instituto Politécnico de Informática “Abel Santamaría” han sido sometidas a análisis (Capítulo 1), puntuando a la hora de su elección:

- Conocimiento previo, por parte de estudiantes y profesores, del funcionamiento de determinadas herramientas de trabajo.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

- Herramientas con las que se trabajan en el centro.
- Características y prestaciones de las herramientas.
- Rendimiento de las máquinas destinadas por el centro a la producción.

2.8.1 Herramientas para la Gestión de Proyecto.

Después de analizar diferentes herramientas para la Gestión de Proyecto en el Capítulo anterior, MOPP propone utilizar para llevar a cabo dicha actividad la herramienta Microsoft Project. Esta herramienta brinda una amplia gama de facilidades para la planificación y la gestión en general, también tiene a su favor el hecho de que a los estudiantes del Politécnico les enseñan a trabajar con ella, de forma práctica, en el primer año de la carrera, en la asignatura Proyecto Informático.

A pesar de que es un software propietario y es una política nacional la migración hacia Linux y software libres, el IPI. va a trabajar con el Microsoft Project por el momento, lo que no quita que más adelante utilice otra herramienta para la Gestión de Proyecto que sea libre.

Requerimientos para Instalar el software:

- Procesador Pentium II de 200 MHz de velocidad del microprocesador.
- Memoria RAM de 32 MB.

2.8.2 Herramientas para la Gestión de Configuración

La herramienta para la Gestión de Configuración que MOPP propone es CVS. Esta herramienta de Gestión de Configuración está categorizada como un estándar para el control de versiones y posee implementaciones en todos los sistemas operativos.

Si comparamos CVS con SVN en lo que respecta a la instalación del producto, la primera tiene las de ganar, ya que no necesita de una gran infraestructura y es muy sencilla a la hora de la instalación. CVS facilita en gran medida el manejo con las versiones y es fácil de utilizar. Aunque SVN es una herramienta más moderna y fue creada con el objetivo de ser óptima ante CVS, presenta desventajas que fueron consideradas a la hora de elegir la

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

herramienta para la Gestión de la Configuración en la Factoría. Los problemas que existen actualmente en el IPI con respecto al control de versiones son solucionables utilizando CVS. Otro aspecto importante es que en el centro se trabaja con esta herramienta, aunque no de forma generalizada y existen personas capacitadas en la utilización de la misma.

2.8.3 Tecnologías y herramientas de Construcción.

Cada tecnología de construcción tiene asociado un cúmulo de herramientas y metodologías que soportan su desarrollo. Un principio indispensable de cualquier Factoría de Software es que debe tener establecido para cada línea de producción el lenguaje de programación y el grupo de herramientas a utilizar.

Lenguaje de Programación:

Como lenguaje de programación se seleccionó Pascal Orientado a Objetos (OOP) y como entorno de programación Delphi 7.

Delphi 7: Entorno de programación que constituye una extensión orientada a objetos del lenguaje de programación Pascal, conocida como Object Pascal o Pascal Orientado a Objetos. Basado en el concepto de la programación orientada a objeto y en particular a la declaración de nuevos tipos de clases.

Para la elección de Delphi 7 no se realizó ninguna comparación pues esta herramienta se imparte a los estudiantes del Instituto Politécnico de Informática “Abel Santamaría” y la vinculación de los estudiantes a la producción esta vista, en este centro, como un complemento de lo que los estudiantes reciben en clases.

Requerimientos para Instalar el software:

- Procesador Pentium II 400 MHz.
- Memoria RAM de 256 MB.
- Sistema Operativo: Microsoft Windows 98, 2000, and XP.
- Capacidad mínima en el disco duro de 475 MB.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Herramienta CASE:

Rational Rose (Rational) 2001: Permite crear diagramas generados durante el proceso de Ingeniería en el desarrollo de un sistema informático. Brinda facilidades para generar la documentación necesaria del software en desarrollo. Posee un gran número de estereotipos predefinidos que agilizan el proceso de la modelación. Proporciona un soporte completo entre funciones a través de integraciones de productos y flujos de trabajo para optimizar el desarrollo incrustado y en tiempo real.

Requerimientos para instalar el software:

- Pentium III.
- Memoria RAM de 64 Mb.
- Espacio en el disco duro como mínimo de 200 MB.
- Sistema Operativo: Microsoft Windows NT 4.0, Windows 9X o Win2000.

Gestor de Base de Datos:

En el caso de la elección de un Gestor de Base de Datos para el IPI sucede igual que con la elección del lenguaje de programación. Anterior a esta investigación en el Politécnico se trabajaba con ORACLE como herramienta para la gestión de Base de Datos, este Modelo respeta la decisión de la dirección del Centro y propone continuar utilizando ORACLE como gestor de Base de Datos.

ORACLE proporciona innumerables beneficios a las entidades que trabajan con el ya que entre sus prestaciones encontramos posibilidades de almacenamiento de grandes volúmenes de datos, manejo rápido y seguro de los datos almacenados. Permite mayor números de conexiones a la Base de Datos con respecto a otros gestores de Base de datos, por ejemplo SQL Server.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

La herramienta que utiliza ORACLE para acceder a la base de datos es el lenguaje procedural SQL (PL/SQL) y este lenguaje es relacionamente completo, es decir, implementa prácticamente toda la funcionalidad y características del modelo relacional teórico.

Requerimientos para instalar el software:

- Memoria RAM de 512 MB, más memoria generalmente mejora.
- Pentium IV.

2.8.4 Mecanismos de Control de la calidad.

Los productos terminados deben cumplir las especificaciones de calidad, para esto son útiles las listas de chequeo. Estas listas especifican una serie de puntos a evaluar durante todas las fases del desarrollo con el objetivo de hacer el menor número de correcciones posibles al final (Ver Anexo 25).

2.8.5 Mecanismo de seguridad de la información.

Dentro de MOPP la seguridad de la información estará clasificada en dos grupos posibilitando así una mejor manipulación.

Estos grupos son:

- Plantillas y técnicas que ordenan el proceso, ya sean enfocadas a los servicios de administración, seguridad de la información, control de la calidad, formato, estilo de los archivos a recibir o de los entregables.
- Documentación referente al lenguaje en que se va a implementar el software, metodología de desarrollo de software, instaladores necesarios y repositorio de componentes de software reutilizables.

Para mantener la información es necesario establecer políticas de seguridad de la información (PSI), las cuales surgen como una herramienta organizativa para concienciar a cada uno de los miembros de la factoría sobre la importancia y sensibilidad de la información.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

Estas políticas describen aspectos como: accesos a la información básica, software que se deben instalar, garantía del producto en uso, restricciones sobre los dispositivos de acceso a la estación de trabajo y del personal ajeno a la Factoría. Estas se pueden observar en el Anexo 23.

Por el momento MOPP plantea una serie de herramientas y mecanismos para brindar soporte al proceso de desarrollo del software, lo que no quita que en un futuro, un determinado proyecto tenga que variar alguna de estas herramientas debido al interés particular de algún cliente en específico que necesite que se trabaje con una determinada herramienta.

2.9 Área Repositorio de componente.

Para el almacenamiento de los componentes en el Repositorio de Componentes se ha adoptado como forma organizativa, la clasificación de los componentes en dos grandes grupos, como se hace en el Modelo funcional de la factoría de Software de la UCI para la Línea de Carrefour, estos grupos son:

- Componentes de código.
- Activos del proceso.

Los componentes de código pueden ser:

- Una clase.
- Un procedimiento o función.
- Un módulo.
- Un subsistema.
- Una aplicación.

Los activos del proceso pueden ser:

- Patrones de diseño.

Capítulo 2: Modelo Organizativo para Proyectos Productivos (MOPP)

- Algoritmos.
- Un esquema de base de datos.
- Manuales y documentación.
- Un modelo.

Otra característica que deben cumplir los componentes del repositorio, es que, cada uno de ellos debe de estar documentado (Ver Anexo 22) con motivo de agilizar el proceso de búsqueda en el repositorio y tener la certeza que el componente localizado es el que verdaderamente se necesita.

Para reducir el tiempo de búsqueda de un determinado componente almacenado, existen diversas técnicas de clasificación y recuperación, las cuales son llevadas a un sistema que automatiza el proceso.

La Factoría del Politécnico de Informática “Abel Santamaría” es una Factoría que esta surgiendo en estos momentos, con carencia de elementos reutilizables, por lo tanto, se ha adoptado seguir el método Hojear, que garantiza la búsqueda en el repositorio, este método es sencillo de implementar, resuelve el problema en un inicio. Lo que no quiere decir que a medida que aumente la cantidad de elementos en el repositorio haya que hacer un nuevo estudio para mejorar el sistema, tal vez elegir una nueva práctica.

Conclusiones

Al finalizar este capítulo se logró la creación de un Modelo de Factoría de Software para la organización de los proyectos productivos en el Instituto Politécnico de Informática “Abel Santamaría”, un modelo cuyas características satisface las necesidades, en lo que respecta a la esfera productiva, del IPI. Este modelo divide la Factoría en cuatro Áreas fundamentales, Área Técnicas y Herramientas, Área Proceso de Desarrollo, Área Trabajadores, Área Repositorio de Componentes. Donde cada una de estas Aéreas están bien explicadas y enfocadas para suprimir las dificultades existentes en el IPI.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

Introducción

En el presente capítulo se muestran los resultados que hasta el momento han arrojado la adopción, por parte del Politécnico de Informática “Abel Santamaría”, del Modelo de Factoría de Software MOPP, estos resultados constituyen los frutos de una prueba piloto realizada a principio del mes de abril del presente año en dicho centro, para ello se crearon dos proyectos productivos, uno es la confección de un sistema para automatizar el proceso de la matrícula de los Estudiantes y el otro es un sistema para automatizar el proceso de la entrega de libros a los estudiantes, estos proyectos fueron compuestos por 60 de los mejores estudiantes de la institución.

3.1 Resultados en la vinculación de estudiantes a la producción.

Un estudio realizado a mediados de diciembre del pasado año, anterior a la adopción del Modelo de Factoría MOPP, muestra como deficiencias en el IPI:

- Mínima vinculación de los estudiantes a la producción.
- Desubicación por parte de los estudiantes a la hora de enfrentarse una tarea, no sabían qué hacer ni a quién dirigirse en cada momento.
- Carencia de una distribución de roles bien definida.
- La gestión de las dudas constituía un tema crítico e incremental.
- Los estudiantes que estaban vinculados a la producción no habían sido sometidos a un proceso de captación previamente.
- Los profesores encargados de guiar el proceso productivo no contaban con la debida preparación para desempeñar dicha tarea.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

- Dominio insuficiente de las herramientas de trabajo por parte tanto de profesores como de estudiantes.
- No existía un sistema de capacitación ni antes ni después de la incorporación a un proyecto.
- No se realizaba una estimación de plazos ni costos en los proyectos que se llevaban a cabo.
- Los recursos existentes (hardware, software, comunicaciones) no eran explotados al máximo.

Por tanto con todas estas insuficiencias no podía obtenerse un producto final con calidad y aceptación.

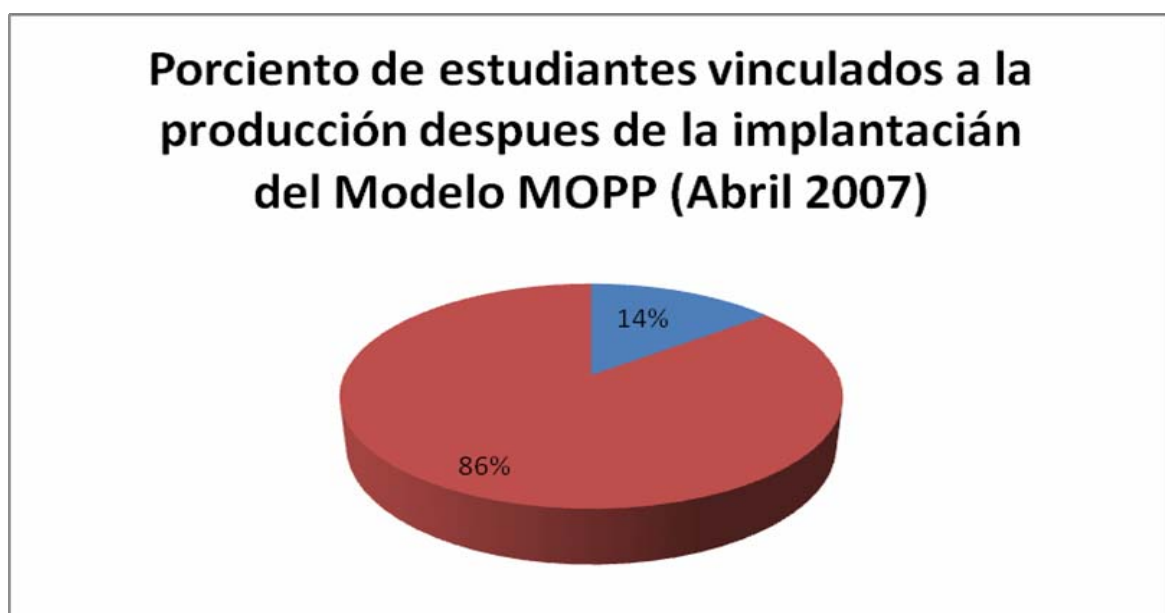
Uno de los temas críticos que presentaba el Politécnico era la vinculación de los estudiantes a la producción, en la gráfica 3.1 se observa que sólo un 5% de la matrícula de ellos estaban vinculados a la producción.



Gráfica 3.1_ Por ciento de estudiantes vinculados a la producción antes de la implantación del Modelo de Factoría MOPP.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

A mediados de Abril del presente año, mediante otro estudio realizado en el IPI valorando los primeros resultados de la adopción del Modelo de Factoría de Software MOPP, se comienza a observar resultados positivos con respecto a la incorporación de estudiantes a la producción, en la grafica 3.2 se observan estos resultados.



Gráfica 3.2_ Por ciento de estudiantes vinculados a la producción con la aplicación del Modelo de Factoría MOPP.

Si comparamos las gráficas 3.1 y 3.2 muestran una diferencia significativa, aunque pequeña, con respecto a la vinculación de estudiantes a la producción que va desde un 5% a un 14.7%, resultando positiva, en ese sentido, la implantación de MOPP en el IPI.

Una vez culminada la implantación del Modelo en el Politécnico, el aumento de los estudiantes vinculados a la producción será mayor, se aspira por la dirección de la Factoría llegar a un 90% de estudiantes, de segundo y tercer año, vinculados a la producción a mediados del próximo curso escolar 2007-2008, debido a que para esa fecha se contará con la experiencia de los meses anteriores y con un recurso humano con mayor capacitación que con el que se comenzó.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

3.2 Resultados en el Área Trabajadores.

En el Área Trabajadores los resultados han sido alentadores, se han logrado grandes avances en poco tiempo, la conformidad en todos los implicados en la Factoría, la calidad en el proceso de selección del personal y el otorgamiento de roles, constituyen una evidencia palpable de ello.

Sub-Área Gestores de la Factoría.

Los resultados en la Sub-Área Gestores de la Factoría han sido satisfactorios, las personas seleccionadas para desempeñar los roles de la dirección de la Factoría han demostrado una gran entereza y capacidad en el cumplimiento de las tareas encomendadas hasta el momento.

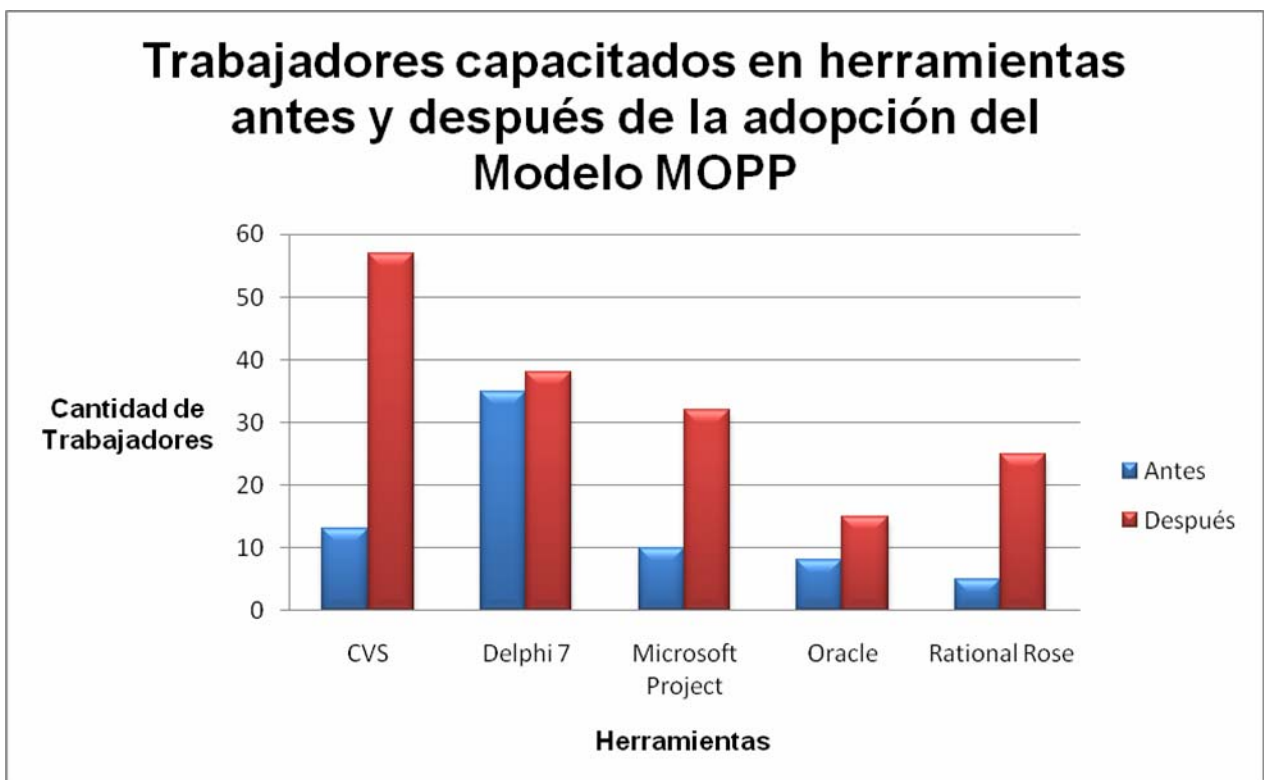
Por ejemplo podemos citar:

El Gestor de la Factoría es una persona con un carácter cordial, afable, con una capacidad especial para hacerse entender requisitos indispensables que debe tener un dirigente. A pesar de ser muy joven, inspira un gran respeto ante sus subordinados, ha demostrado en el corto tiempo que lleva desempeñando el rol, gran capacidad para la dirección y los negocios.

El estudiante designado para el rol de investigador ha demostrado gran responsabilidad y capacidad en el cumplimiento de su trabajo. Ha ideado, en el corto tiempo que lleva como investigador, un plan de capacitación que integra a todos los profesores y estudiantes que trabajan en la Factoría, unos imparten las clases y los otros se capacitan, gracias a este sistema, actualmente en el IPI se está desarrollando un grupo de cursos de capacitación a nivel de proyecto que van desde cursos de dirección hasta los referentes al manejo de las herramientas que se necesitan dominar para el trabajo en la factoría. Se ha implementado un sistema de competencia entre los estudiantes, cuyos resultados tributan a sus respectivos proyectos, logrando así mayor motivación a la hora de la capacitación, se le asignan puntos a los contenidos vencidos por los estudiantes y se les hace un reconocimiento en el matutino ante toda la escuela a los estudiantes destacados.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

En la grafica 3.3 se muestra una comparación entre la cantidad de personas que dominaban las herramientas seleccionadas por MOPP para el trabajo en la Factoría antes y después de implantado el proceso de capacitación que dicho Modelo propone.



Grafica 3.3_ Comparación entre el número de Trabajadores de la Factoría capacitados antes y después de la adopción de MOPP. Concurrent Versions System (2-57), Delphi 7 (35-38), Microsoft Project (10-32), Rational Rose (5-25), Oracle (8-15).

La elección del Jefe de captación no pudo ser mejor, pues es un profesor que lleva muchos años trabajando en la enseñanza politécnica, con una basta experiencia y metodología, conoce muy bien las características de los estudiantes y especialmente a los de tercer año ya que ha sido profesor de de ellos durante toda la carrera.

Captación

En lo que respecta a la captación, la adopción de MOPP fue de gran ayuda:

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

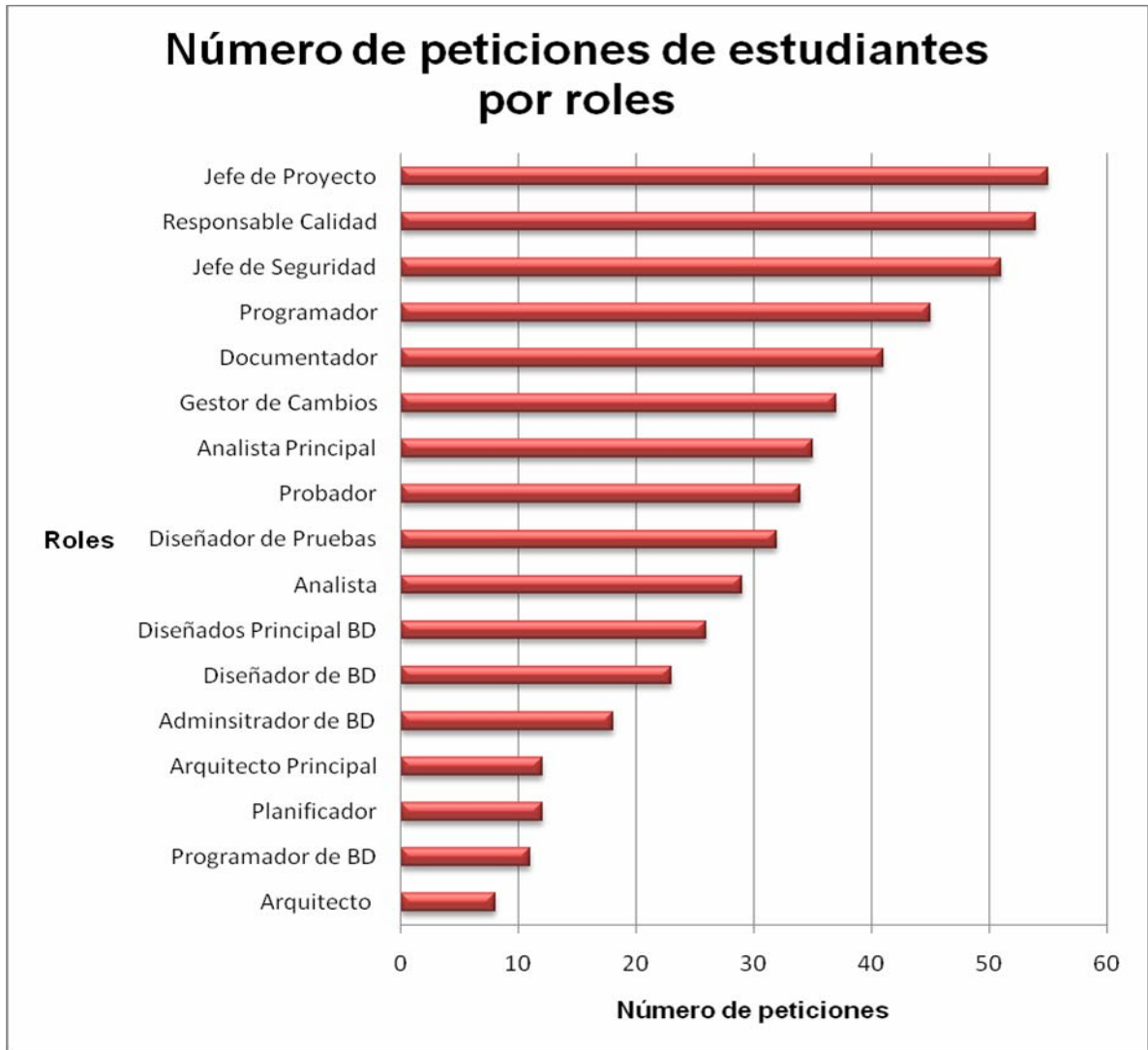
- Se logró organizar el proceso de captación de trabajadores a la Factoría.
- No hubo quejas por parte de los estudiantes sobre la forma en que se concibió el otorgamiento de los roles.

Resultó de gran ayuda en el proceso de confección del escalafón, el ya elaborado por la dirección del IPI, para la ubicación del servicio social de los estudiantes de tercer año.

La planilla que se les entregó a los estudiantes (Ver Anexo 22) para que eligieran el rol a desempeñar en la Factoría fue calificada por ellos mismos como una brillante idea, ya que le brindó la oportunidad de elegir según su criterio, capacidades, conocimientos y aptitudes el rol deseado, sumándole a ello que tenían tres posibilidades de elección en la planilla.

En la grafica 3.5 se muestran los resultados que arrojó la planilla elaborada para llevar a cabo el otorgamiento de roles en la Sub-Área Grupo de Desarrollo de la Factoría.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos



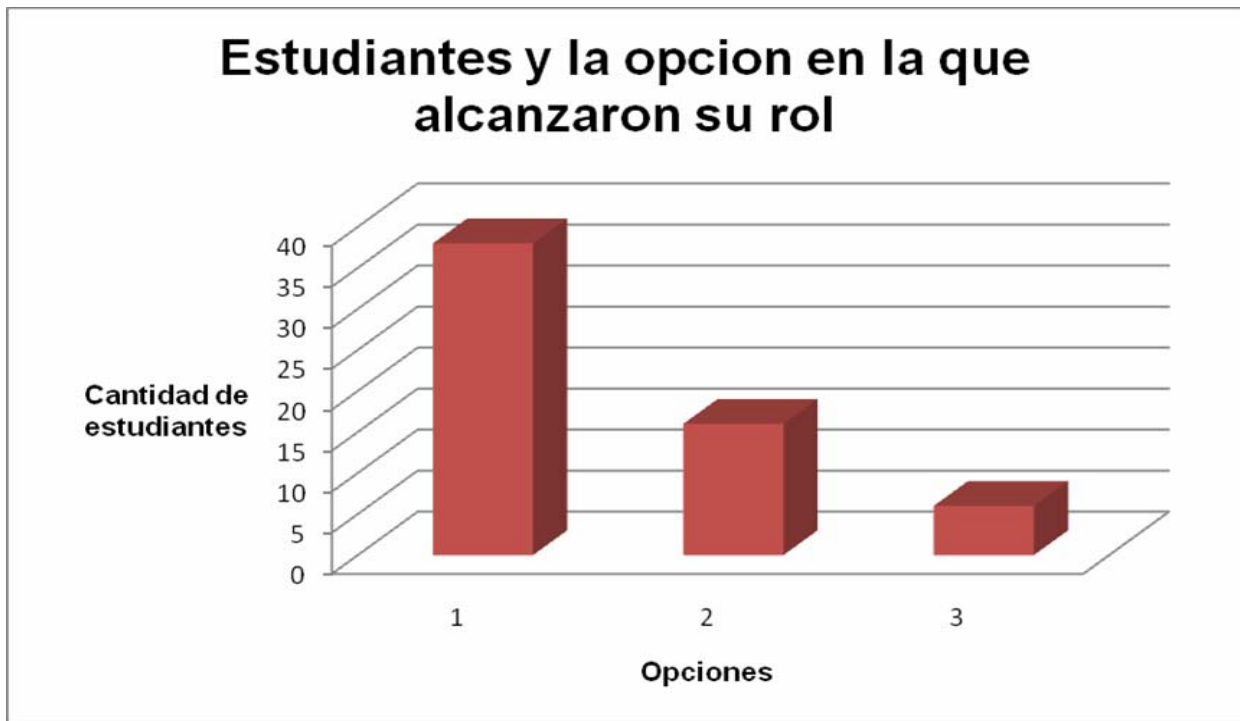
Grafica 3.5_ Cantidad de estudiantes optando por los distintos roles de la Factoría.

Sub-Área Grupo de Desarrollo.

Los trabajadores de la Sub-Área Grupo de Desarrollo en su mayoría están satisfechos con el rol que se les fue asignado.

En la grafica 3.6 se muestran los resultados del otorgamiento de roles en la Sub-Área Grupo de Desarrollo.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos



Gráfica 3.6_ Cantidad de estudiantes que se les otorgó el rol en la primera opción (38), segunda opción (16) y tercera opción (6).

3.3 Área Técnicas y Herramientas

En el Área Técnicas y Herramientas los avances han sido significativos, se destinaron dos laboratorios (201 y 202) cada uno con 18 máquinas que cumplen con los requerimientos necesarios para la instalación de software, entre ellas un servidor de base de datos y un servidor para los componentes reutilizables donde también se encuentran almacenados los instaladores de las herramientas de trabajo y documentación referente a ellas, en los laboratorios hay conexión.

Por el momento en los laboratorios de producción hay una máquina por cada dos trabajadores, con la excepción de que los jefes de proyecto debido al cúmulo de trabajo que este rol asume tienen una computadora a su servicio.

A la hora de hacer la distribución de las máquinas se tuvo en cuenta que los trabajadores que ocuparan la misma computadora desempeñarían el mismo rol ya que cada estación de

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

trabajo tiene instaladas las herramientas que necesita un trabajador para realizar su función en el proyecto, si los dos trabajadores utilizan las mismas herramientas, las prestaciones de la maquina son mejor utilizadas. El horario en que cada trabajador utilizará la máquina es decisión de ellos.

En lo que respecta a la utilización de las herramientas seleccionadas por MOPP en cada uno de los proyectos creados podemos citar los siguientes avances:

- Los planificadores están llevando a cabo la planificación con la herramienta de Gestión de Proyecto Microsoft Project obteniendo buenos resultados a la hora del establecimiento de metas, comunicación con los demás trabajadores y en lo que respecta a la construcción de diagramas.
- Los trabajadores involucrados en el proceso de desarrollo del Software están utilizando para la confección de los diferentes diagramas la herramienta CASE Rational Rose, reportando buenos resultados sobre todo en lo que respecta a disminución del tiempo de trabajo.

Las herramientas restantes definidas en esta Área aún no se han comenzado a utilizar ya que en la fase de desarrollo en que se encuentran inmersos los proyectos hasta el momento no es necesaria su utilización.

En los proyectos que se están realizando actualmente en el Instituto Politécnico “Abel Santamaría” los mecanismos de seguridad implementados son:

- Restricción de acceso a los laboratorios de producción, autorizando solamente a trabajadores del proyecto y Gestores de la Factoría.
- Acceso autorizado a las estaciones de trabajo solamente a los administradores de las mismas y Jefe de Proyecto.
- Firewall habilitado en todas las computadoras del laboratorio, asegurando de esta forma que no se pueda acceder a ella desde otras máquinas.

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

- Antivirus actualizado en cada una de las computadoras del laboratorio, evitando de esta forma la infección de las mismas y la propagación de los virus informáticos.

Con el objetivo de dar seguimiento y garantizar que se cumplan los mecanismos de seguridad establecidos en los laboratorios de producción:

- Se estableció un sistema de inspecciones a las estaciones de trabajo.
- Evaluación quincenalmente de los mecanismos de seguridad.

Los trabajadores que no cumplan con los requisitos establecidos serán sancionados, reduciéndosele el tiempo de máquina en el laboratorio o con la separación del proyecto.

3.4 Área Proceso de Desarrollo

En el Área Proceso de Desarrollo se lleva a cabo el Levantamiento de Requisitos, donde los analistas están realizando esta actividad satisfactoriamente, la herramienta utilizada en esta fase para el modelado, como se planteo en el epígrafe anterior, es el Rational Rose.

3.5 Área Repositorio de Componentes

En cada laboratorio de producción se encuentra un servidor destinado al almacenamiento de los componentes reutilizables y en el nodo central hay servidor central con copias de los componentes de cada uno de los servidores ubicados en los laboratorios de producción, esto responde a una medida de seguridad que se tomó para evitar la pérdida de componentes reutilizables o información almacenada en caso de fallo de la tecnología.

En estos momentos solamente están almacenados en dichos servidores los instaladores de las herramientas de trabajos y algunas de las versiones de los diagramas realizados en la captura de requisitos.

Conclusiones

Una vez finalizado el presente capítulo se ha llegado a la conclusión de que resultó positiva la adopción del Modelo de Factoría de Software MOPP por parte del Instituto Politécnico de

Capítulo 3: Resultados de la implantación del Modelo Organizativo para Proyectos Productivos

Informática “Abel Santamaría”, ya que en el corto tiempo de evaluación de los resultados se alcanzaron los siguientes logros:

- Aumentó en un 9.7 por ciento la vinculación de los estudiantes a la producción.
- Se asignaron roles a los estudiantes incorporados a la producción según sus capacidades y preferencias.
- Se llevó a cabo un sistema de capacitación a los trabajadores involucrados en la Factoría.
- Se establecieron medidas para la seguridad de la información y mecanismos para lograr un producto final con calidad.

Recomendaciones

Conclusiones Generales

El Modelo para la Organización de Proyectos Productivos (MOPP) es un Modelo de Factoría de Software creado con el objetivo de suprimir los problemas relacionados con la producción en el Instituto Politécnico de Informática “Abel Santamaría”.

Este Modelo fue confeccionado a partir del análisis de los diferentes Modelos de Factoría de Software más referenciados dentro de la bibliografía consultada, es un Modelo que se adapta a las características particulares del Instituto Politécnico de Informática “Abel Santamaría”.

MOPP esta dividido en cuatro Áreas fundamentales: Área Técnicas y Herramientas, Área Proceso de Desarrollo, Área Trabajadores y el Área Repositorio de Componentes.

La adopción de MOPP por el IPI tuvo gran trascendencia para este centro al ser mejorados los aspectos relacionados con la producción después de su implantación.

En el corto tiempo en que fue evaluado se arrojó los siguientes resultados:

- Aumentó en un 9.7 por ciento la vinculación de los estudiantes a la producción.
- Se definieron técnicas, mecanismos y herramientas que están presentes durante el proceso de desarrollo del software.
- Se confeccionó un proceso de captación para la elección de los trabajadores implicados en la Factoría.
- Se asignaron roles a cada uno de los implicados tanto en la Factoría de Software como en el proceso de desarrollo del software, teniendo en cuenta sus capacidades y afinidades.
- Se definió un proceso de capacitación para lograr la superación de los trabajadores implicados en la Factoría.

Por todo lo anteriormente expuesto se concluye que el objetivo general trazado al inicio de este trabajo fue cumplido en su totalidad.

Recomendaciones

Recomendaciones

Los objetivos del trabajo no abordan todo lo relacionado con el amplio concepto de Factoría de Software, por lo tanto, se recomienda que se continúe profundizando en el estudio del mismo puesto que es un enfoque práctico y con muchas garantías para lograr la calidad en el desarrollo del Software.

Con respecto al Modelo Organizativo para Proyectos Productivos, se recomienda:

- Continuar con la implantación del Modelo en el Instituto Politécnico de Informática “Abel Santamaría”.
- Profundizar en el Área Proceso de Desarrollo.
- Elaborar una herramienta para automatizar el modelo.
- Extender el Modelo a los diferentes Institutos Politécnico del país, teniendo presente siempre las características particulares de la producción en cada uno de estos centros.

Referencias Bibliográficas

Referencias Bibliográficas

- [1] Cusumano, M A. (1989). *Software Factory: A Historical Interpretation*. IEEE Software, (Vol. 6, No. 2). pp. 23-30. 2006.
- [2] Fernstrom, C.; Narfelt, K. H and Ohlsson, L. (1992). *Software Factory Principles, Architecture and Experiments*. IEEE Software. (Vol. 9, No. 2). pp. 36-44. 2006.
- [3] Cantone, G. (1992). *Software Factory: Modeling the Improvement*. 'Competitive Performance Through Advanced Technology'. Third International Conference on (Conf. Publ. No. 359). pp. 124 – 129. 27-29. 2006.
- [4] Li, C.; Li, H.; Li, M. A (2001). *Software Factory Model Based on ISO 9000 e CMM for Chinese Small Organization*. Second Asia-Pacific Conference on Quality Software (APAQS'01). Hong Kong. December. 2006.
- [5] Báez, M. P. (2002). *La Industria del Software, una oportunidad para México*. 2006.
- [6] Fernandes, A. A.; Teixeira, D. (2004). *Fábrica de Software: Implementação e Gestão de Operações*. Editora Atlas. 2006.
- [7] Casañola, Y. T. (2007). *Evaluación teórica de la adopción del enfoque de Factorías de Software en la Universidad de Ciencias Informáticas*. <http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion%20de%20trabajos/Yaim%ED%20Trujillo%20Casa%F1ola%20P.pdf>. 2006.
- [8] Palacios, J. (2006). *Sinopsis de los modelos SW-CMM y CMMI*. 2007.
- [9] La Hoz, Y. R. and Valdivida, M.M. (2005). *Modelo Funcional de la Factoría de Software de la UCI para la Línea de Carrefour, Universidad de las Ciencias Informáticas*. 2006.
- [10] Humphrey, W.S. (2001). *Introducción al Proceso de Software Personal*. Addison Wesley ed. 2007.
- [11] Jacobson I. G.B. Rumbaugh J. (2004). *“El Proceso Unificado de Desarrollo de Software”*. 2007.

Referencias Bibliográficas

- [12] Greenfield J. Short K. Cook S. Kent S. and Crupi J. (2004). Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools. John Wiley and Sons. 2007.
- [13] Alvarez, S. H. A. (2006). Metodología para el Desarrollo de Aplicaciones con Tecnología. 2007.
- [14] Councill, B. & Heineman, G. (2001). Definition of a Software Component and its Elements, Addison-Wesley, Boston. 2007
- [16] Muñoz J. and Pelechano V. (2005). Building a Software Factory for Pervasive Systems Development. M.E. Sharpe. 2007.
- [17] Garcia, S. R. (2002). Subversion Personal. In http://cronopios.net/Traducciones/subversion_personal.html . 2007.
- [18] Mantei, M. (1981). The Effect of Programming Team Structure on Programming Task. CACM. 2007.
- [19] La Industria del Software y la Dualidad Monetaria. (2005). In <http://cubaalamano.net/sitio/client/article.php?id=5780>. 2007

Anexos

Bibliografía

1. Fernström, C. *The Eureka Software Factory: Concepts and Accomplishment*. Proceedings Third European Software Engineering Conference. Berlin, 1991.
2. Fernandes, A. A.; Teixeira, D. (2004). *Fábrica de Software: Implementação e Gestão de Operações*. Editora Atlas. 2006.
3. La Hoz, Y. R. and Valdivida, M.M. (2005). *Modelo Funcional de la Factoría de Software de la UCI para la Línea de Carrefour, Universidad de las Ciencias Informáticas*. 2006.
4. Cusumano, M A. (1989). *Software Factory: A Historical Interpretation*. IEEE Software, (Vol. 6, No. 2). pp. 23-30. 2006.
5. Fernstrom, C.; Narfelt, K. H and Ohlsson, L. (1992). *Software Factory Principles, Architecture and Experiments*. IEEE Software. (Vol. 9, No. 2). pp. 36-44. 2006.
6. Cantone, G. (1992). *Software Factory: Modeling the Improvement*. 'Competitive Performance Through Advanced Technology'. Third International Conference on (Conf. Publ. No. 359). pp. 124 – 129. 27-29. 2006.
7. Álvarez, S. H.A. (2006). *Metodología para el Desarrollo de Aplicaciones con Tecnología*. 2007.
8. Muñoz J. and Pelechano V. (2005). *Building a Software Factory for Pervasive Systems Development*. M.E. Sharpe. 2007.
9. Basili, V. R.; Caldiera, G. and Cantone, G. (1992). *A Reference Architecture for the Component Factory*. ACM Transaction on Software Engineering and Methodology. Vol 1. pp 53-80. 2007
10. Casañola, Y. T. (2007). *Evaluación teórica de la adopción del enfoque de Factorías de Software en la Universidad de Ciencias Informáticas*. <http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion%20de%20trabajos/Yaim%ED%20Trujillo%20Casa%F1ola%20P.pdf>. 2006.

Anexos

11. Un caso para la fábrica de software. In <http://www.microsoft.com/spanish/msdn/articulos/archivo/041206/voices/TheCaseForSF.msp>
12. Desarrollo de la Metodología Extreme Programanig. In http://www.peruserver.com/des_metodologia.php. 2007
13. Berrueta, M. D. Herramientas colaborativas: Bugzilla, CVS y Wiki In http://www.asturlinux.org/archivos/jornadas2003/03_Charla_Bugzilla/herramientas-colaborativas-texto.pdf. 2007
14. Beaumont, A. (2006). Desarrollo en entornos UNIX libres in <http://det.bi.ehu.es/NQAS/jsp/down/herramientas1.pdf>.2007
15. Introducción a Subversion / Tutorial de Subversion (2006). In <http://www.kikov.org/subversion-tutorial-es-index>. 2007
16. Collins-Sussman, B; Fitzpatrick, B. W and Pilato, C. M. (2004). Control de versiones con Subversion. In <http://svnbook.red-bean.com/nightly/es/svn-book.html#svn-ch-1-sect-1>.2007
17. Moisset, D. F. (2004) Sistemas de control de revisión. In <http://www.grulic.org.ar/eventos/charlas/rcs-2004-10.html>. 2007
18. Palacio, J. (2006). Sinopsis de los modelos SW-CMM y CMMI. In http://www.qualitatis.org/files/1/sinopsis_cmm.pdf. 2007
19. Rojas Rivero, D.J. (2003). Aproximación a la Industria del Software en el Estado Lara. In http://bibcyt.ucla.edu.ve/edocs_bciucla/rojas//software.pdf. 2007
20. Muñoz, J. and V. Pelechano. *MDA vs Factorías de Software*. In <http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CInConferenceArticle%5C01-Munoz.pdf>