

Universidad de las Ciencias Informáticas

Facultad # 3



Título: Propuesta de marco de Fábricas de Software basado en las buenas prácticas de Métodos Ágiles

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

Autor: Jasmín Fuentes Padrón
Tutores: Ing. Rene Lazo Ochoa
MSc. Rolando Quintana Aput

La Habana, Junio de 2008

Declaración de autoría

Declaro que soy la única autora de este trabajo y autorizo a 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 _____

Jasmín Fuentes Padrón

Ing. René Lazo Ochoa

MSc. Rolando Quintana Aput

Datos de los contactos

Nombre y Apellidos: Ing. Rene Lazo Ochoa.

Fecha de nacimiento: 10 febrero 1982.

Institución: Universidad de las Ciencias Informáticas.

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 82021020245

Correo: rlazo@uci.cu

Situación laboral: Profesor.

Institución: Universidad de las Ciencias Informáticas.

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba.

Código Postal 19370.

Resumen del Currículo

Ingeniero en Ciencias Informáticas en La Universidad de Ciencias Informáticas desde julio de 2007. Al graduarse pasa a ser profesor adiestrado de la Universidad de las Ciencias Informáticas, en la Disciplina de Técnicas de programación donde Imparte la asignatura Inteligencia Artificial. Se desempeña como Arquitecto principal del proyecto de software Enterprise Resource Planning (ERP) Cuba.

Ha publicado en la revista Cubana de Ciencias Informáticas.

Nombre y Apellidos: MSc Rolando Quintana Aput

Fecha de nacimiento: 19 de Junio de 1976

Institución: Universidad de las Ciencias Informáticas

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 76061921366

Correo: rqaput@uci.cu

Situación laboral: Profesor Instructor, Departamento de la Especialidad de la facultad #4

Institución: Universidad de las Ciencias Informáticas

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba.

Código postal 19370.

Resumen del Currículo

Ingeniero Informática del Instituto Superior Politécnico José Antonio Echeverría desde Julio del 2004. Máster en Gestión de Proyectos Informáticos desde Diciembre de 2007 de la universidad de Las Ciencias Informáticas. Al graduarse pasa a ser profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Técnicas de programación. Obtiene la categoría de Instructor en octubre del 2005 y actualmente está en el proceso de categorización optando por la categoría de Asistente. Tiene una experiencia como profesor de cinco cursos en las asignaturas: Introducción a la Programación, Programación I, Programación II, Seguridad Informática, Historia de la Informática y metodología de la Investigación además de cursos optativos sobre PHP y HTML. Ha trabajado en investigaciones relacionadas a la investigación de operaciones y competencias de las personas que ocupan roles en los proyectos multimedia. Tiene dos artículos publicados en memorias de eventos científicos nacionales y uno internacional. Ha ejercido como tutor de varias tesis de pregrado, así como miembro de tribunal y en estos momentos participa como tribunal en tesis de postgrado.

*“El secreto del éxito está
en la constancia del
propósito”.*

AGRADECIMIENTOS

A ti Jehová por permitirme haber estudiado en esta universidad de excelencia, por permitirme ser parte de su brillante tropa del futuro, por darme la oportunidad de vivir y regalarme una familia tan maravillosa. Por enseñarme el camino correcto de la vida, guiándome y fortaleciéndome cada día con tu espíritu.

*Principalmente a mis adorados padres: **Coralía C. Padrón y Eduardo F. Fuentes de León** por ser los mejores, los que me dieron la vida. Por su determinación, entrega y humildad que me han enseñado tanto. Por ser la razón de mí existir. Gracias por todo papi y mami, por apoyarme en todas las decisiones que he tomado en la vida y por creer siempre en mí. Estos son mis resultados, se los obsequio hoy. Los amo con todo mi corazón.*

*A mi precioso hermanito: **Edy Julian** mi vida te adoro.*

*A mis madrinas **Gladys** y **Rosita** por siempre ayudarme cuando mas lo necesite*

***A Zayda** por ser mi esperanza de seguir adelante. Por ser esa personita que me alentó tanto y me escucho siempre. Profe gracias por ayudarme tanto, le debo mucho le adoro.*

*A mi hermana que la vida me regaló: **Lesyanis Pompa** y que siempre ha estado a mi lado en buenas y malas, ayudándome y sacándome de los baches, por siempre confiar en mi, te quiero mucho. Gracias pro ser mi amiga del alma.*

*A mis buenas amigas de que la vida me ha regalado como mis hermanitas: **Ana, Susy, Petry Sheila, Thaily, Lidy Chan, Linet, Maylen** a todas por ayudarme tanto en lo personal como en los estudios, por hacerme reír y por saber que las tengo en mi vida. Gracias por ser mis amigas y recuerden que siempre las llevaré presente.*

*A mis abuelas: **Edelmira Padrón y Micaela de León**, por estar siempre pendientes de mí y por hacerme tan feliz con su presencia y ayudarme y darme tanto apoyo.*

*A mí querido Vartolito (**Pascual Verdecia**) por su amor y apoyo siempre en todo.*

*A **Rolando Quintana Aput** por haberme ayudado en este trabajo y apoyarme cuando más lo necesite, por ayudarme a alcanzar este sueño, que significa hoy por hoy el primer escalón de una vida exitosa. Por tenerme la paciencia necesaria. Gracias Roly, por tantas noches de desvelo a mi lado, apoyándome y dándome optimismo.*

*A **Rene Lazo Ochoa** por su paciencia y conocimientos.*

*A **Made** con todo mi corazón, a **Migue** por formar parte en mi vida, a **Niurka**, a **Jose Ramón**, a **Barriel**, a **Roberto**, a **Valia** a **Hector Rodríguez**, a **Eduardo**, a **Las Muchachitas de la Biblioteca**, a **Orlandito** mi primo querido, a **Irita**, a **Mis tíos Ady y Jose**, a **Ramoncito**, a **Handy (hdalmau)**, a **Anita** y a todos por su ayuda y gracias por todo.*

En fin, a todas las personas que quiero y que de una u otra forma hicieron que mi etapa universitaria fuera enriquecedora tanto emocional como intelectualmente. Sin intención de haber dejado de mencionar a alguien.

Jasmín

*Dedico con todo mi amor y mi cariño mi tesis a mi
mamita linda de mi corazón y a mi adorado padre,
que son para mí lo más sagrado que tengo en la vida
y que son la luz de mis ojos.*

*Gracias por existir y sentirme tan orgullosa de
ustedes.*

*Gracias por estar siempre a mi lado y ser los mejores
padres del mundo.*

*Gracias por ayudarme a alcanzar este sueño, por ser
mi guía y mi inspiración
Los ama su hija del alma:
Jasmín*

RESUMEN

En materia de desarrollo de software Cuba no se ha quedado atrás y se ha propuesto producir con calidad óptima sus productos de software, teniendo en cuenta que en la actualidad ha surgido una nueva concepción en la manera de producir software, se trata de la “Fábrica de Software”, donde su principal objetivo es lograr una mejor organización, control, desarrollo y producción. Con esta investigación luego de un profundo análisis sobre el tema se propone una nueva estructura organizacional, es decir un marco de Fábrica de Software basado en las buenas prácticas de los métodos ágiles para obtener una mayor productividad en los proyectos de La Universidad de Ciencias Informáticas(UCI), logrando que los productos sean adaptables a los diferentes entornos, además de elevar la eficiencia y eficacia del proceso productivo, el cual es de gran importancia para lograr la excelencia en la producción. Siendo así de gran necesidad una nueva estructura que elimine las irregularidades existentes dentro del proceso de desarrollo de software y se eleve la producción dentro de la organización, influyendo positivamente en el desarrollo de la economía del país.

Palabras claves: Fábricas de software, Proceso de desarrollo, Metodologías Ágiles.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Conceptualizaciones de la investigación.....	5
1.3. Antecedentes de una Fábrica de Software.....	7
1.4. Objetivos de una Factoría de Software.	8
1.5. Línea de Producción y Montaje	9
1.6. Modelos de Factoría de Software.....	10
1.6.1. Modelo basado en la norma ISO 9001 y CMM.....	10
1.6.2. Modelo Eureka.....	12
1.6.3. Modelo Clasificadorio.	14
1.6.4. Modelo propuesto por Basili.	15
1.6.5. Modelo Replicable.	17
1.7. Modelos propuestos en la UCI de Fábrica de Software	18
1.7.1. Modelo funcional de la Factoría de Software de la UCI para la Línea Carrofeur.	18
1.7.2. Modelo Organizativo para los Proyectos Productivos (MOPP).	19
1.7.3. Modelo de Factoría de Software aplicando Inteligencia para el Polo Productivo Gestión de Proyecto.	19
1.8. Análisis comparativo de los modelos.....	20
1.9. Experiencia Internacional	22
1.10. Ventajas en la aplicación de un modelo de factoría.	23
1.11. Proceso de desarrollo de Software.....	24
1.11.1. ¿Qué es un proceso de desarrollo?	24
1.11.2. Principales características	25
1.12. Metodologías de desarrollo de Software.	25
1.12.1. Metodologías tradicionales (no ágiles)	26
1.12.2. Metodologías ágiles.....	27
1.13. Conclusiones del Capítulo 1.	34
CAPÍTULO 2. DESARROLLO DEL MARCO DE FÁBRICA DE SOFTWARE	36
2.1. Introducción	36
2.2. Marco Fábrica de software. Aproximación práctica del proceso productivo.....	36
2.3. Identificación de los elementos del modelo de Fábrica de Software.....	37
2.3.1. Observatorio	37
2.3.2. Área Productiva	40
2.3.2.1. Guía de flujo de procesos para el desarrollo de software en el Área Productiva.....	42
2.3.2.2. Detalles de las fases del ciclo de vida de un producto informático en la Fábrica de Software.....	45
2.3.3. Área Servicios de la Fábrica.....	53
2.3.4. Interrelación sistémica entre las áreas.	65
2.4. Flujos verticales y horizontales.....	66
2.5. Despliegue de la Fábrica de Software.....	67
2.5.1. Fases para la implantación de la Fábrica de Software.....	69
2.6. Valoración de la propuesta del marco de Fábrica de Software.	71
2.6.1. Comparación de los modelos de producción industrial v/s marco de Fábrica de Software.....	71
2.6.2. Arquitectura empresarial v/s modelo de Fábrica de Software.	73
2.6.3. Principios de las metodologías que se tienen encuentra en el marco de Fábrica de software propuesto.	74
2.7. Conclusiones del Capítulo 2.	75

CONCLUSIONES GENERALES.....	76
RECOMENDACIONES	77
REFERENCIAS BIBLIOGRÁFICAS.....	78
ANEXO 1. COMPARACIÓN DE LOS MODELOS DE FACTORÍA DE SOFTWARE.....	81
ANEXO 2. COMPARACIÓN ENTRE MODELOS DE PROCESO DE SOFTWARE.....	83
ANEXO 3. DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES.....	84
GLOSARIO DE TÉRMINOS Y SIGLAS	85

INTRODUCCIÓN

Uno de los aspectos mas importantes del mundo contemporáneo es el acelerado desarrollo de la ciencia y la tecnología dentro de esta la producción de software se ha convertido en una importante vía de ganancias económicas. En dicha actividad han surgido un gran número de modelos, procesos, tecnologías, teorías, y normas a seguir en busca de lograr mejoras en la productividad del proceso de desarrollo.

Las Fábricas de Software se encuentran dentro de las concepciones contemporáneas para producir el software, que permite explotar las potencialidades del modelo industrial orientandolo hacia una economía de escala. También fomenta todos los procesos 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.

La producción de software es una tarea compleja que presenta mucha incertidumbre durante su realización. Parte de la solución ha sido el desarrollo de metodologías y modelos de procesos, así como todo un arsenal en las áreas de la gestión de calidad total.

Para mejorar la productividad y calidad del software producido por las empresas, los ingenieros de software han trabajado por más treinta años buscando soluciones en estos aspectos, una de las cuales es la idea de Fábricas de Software.

Según las tendencias actuales los procesos de desarrollo de software se vieron abordados por metodologías tradicionales, las cuales son eficientes en proyectos de gran envergadura, sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante y en donde se exige reducir drásticamente los tiempos de desarrollo. Como solución al problema, hace menos de una década, aparecieron las metodologías ágiles, orientadas a simplificar y complementar el trabajo del desarrollador, permitiendo gestionar la incertidumbre en la estimación de los proyectos aprovechando así un poco más de lo aprendido en otros proyectos que se han realizado, es decir aplicar la reutilización y ofrecer un mejor servicio.

El desarrollo de la economía cubana está vinculado en los momentos actuales a los avances relacionados con Las Tecnologías de la Información y las Comunicaciones(TIC). La producción de software se convierte en una actividad cada vez más de demanda, lo que

provoca que las empresas dedicadas a la producción de bienes y servicios informáticos tiendan a buscar las vías idóneas para el perfeccionamiento de la gestión empresarial.

El ministro cubano de Informática y Comunicaciones, Ignacio González, opina que Cuba debe aspirar a hacerse un hueco en el mercado mundial de software. Un negocio que mueve más de 440.000 millones de euros al año y que podría suponer una importante fuente de recursos para Cuba. Antes de conseguirlo, la Isla debe prepararse y profundizar tanto en la capacitación profesional de sus trabajadores, como en la dotación tecnológica de que dispone (ALBA 2004).

La UCI, surgida al calor de La Batalla de Ideas es un pilar fundamental para el desarrollo del país y constituye un proyecto que se concibió con el objetivo de contribuir a la formación de ingenieros informáticos como soporte a La Industria Cubana del Software, convirtiéndose de la misma forma en una fuente de ingresos para el país. Dicho centro es un desafío importante que impone muchos retos, de ellos el más significativo es encontrar un modelo de desarrollo, eficiente, productivo, adaptativo, que permita irrumpir en ese mercado monopolizado y ganar un espacio en él. Por las razones anteriores que investigaciones de este tipo tienen un aporte significativo.

En la actualidad el centro de altos estudios UCI se encuentra inmerso en un proceso de transformación y cambios estructurales en cuanto a la forma en que lleva a cabo la producción, pues no se cuenta aún con el modelo ideal de "Formación-producción" que le permita convertirse en una Universidad ejemplo de industria de software con altos índices de productividad y eficiencia para Cuba y el Tercer Mundo. Por las razones antes expuestas se presenta la siguiente Propuesta de Marco de Fábrica.

En la UCI existen algunas irregularidades para El Proceso de Desarrollo de Software. Cómo son la docencia variable, los cambios sistemáticos en los horarios, las evaluaciones están concentradas lo que provoca alto grado de estrés en los estudiantes. También la existencia del desarrollo con estudiantes y profesores noveles en la Industria de Software. La ausencia de un proceso de certificación y capacitación que permita la superación del equipo técnico para así lograr productos de óptima calidad. La inestabilidad en el trabajo con los clientes y la capacidad gerencial reducida, son algunos de las referencias que se pueden hacer.

El uso de la metodología Rational Unified Process (RUP) exige de un nivel de madurez y de disciplina laboral en el ajuste a los procesos normados, que se complejiza con las

características de la UCI y en ocasiones imposibilitan lograr una gestión eficiente de los proyectos productivos.

La producción basada en productos a la medida requiere de largos plazos de desarrollo y luego dificultan el proceso de parametrización hacia otros clientes, los productos no son adaptables a distintos escenarios o clientes, pues su propio proceso de desarrollo no los concibe así. De este modo se afecta la productividad de la industria.

No se utiliza un modelo de producción orientado a la reutilización de componentes parametrizables, donde se simplifiquen las actividades y se eleve el nivel de especialización de los involucrados.

No se concibe la producción en aras de obtener tecnologías, que se parametricen y conviertan en productos según clientes concretos, sino que se busca un resultado determinado, en un escenario determinado.

EL trabajo con los procesos productivos es muy pobre, se busca la solución en las metodologías y las buenas ideas, encontrando soluciones incompletas.

No existe casi conciencia económica de la producción.

La gestión de hitos de negocio e hitos tecnológicos en los proyectos es deficiente. Es limitado e insuficiente el manejo del concepto de valor agregado. Por lo que no se estratifica el alcance de las soluciones, por tanto no existen estrategias de gancho en los productos, se dan soluciones muy a la medida que son muy difícil de generalizar o aplicar en circunstancias parecidas. Los productos que se desarrollan poseen poca adaptabilidad.

Falta el trabajo estratégico, para interrelacionar de forma sistémica y a través de procedimientos la especialización entre las soluciones tecnológicamente integrales y las modelaciones de los procesos en busca de soluciones a transformaciones organizacionales, falta modelar soluciones tecnológicas centradas en arquitecturas robustas.

Ante la situación descrita anteriormente se plantea el siguiente problema: La baja productividad existente en los proyectos de software y la poca adaptabilidad de los productos que se construyen, afecta la eficiencia y eficacia del proceso de desarrollo de software.

El objetivo general es proponer un marco de Fábricas de Software basado en las buenas prácticas de los métodos ágiles.

Como objetivos específicos de la presente investigación se plantea:

- Realizar el estudio del arte de los Modelos Industriales del Desarrollo de Software y los métodos ágiles.
- Identificar y describir la interrelación entre los elementos que integran un marco de fábricas de software basado en las buenas prácticas de los métodos ágiles de desarrollo.
- Proponer una guía de flujo de procesos para el desarrollo en el Área Productiva.
- Valorar las características que cumple el marco de Fábricas de Software propuesto en cuanto a arquitectura de empresa y características de las metodologías de desarrollo de software.

El objeto de estudio es el Proceso de Desarrollo de Software y como campo de acción los Modelos de Desarrollo Basados en Ambientes Industriales de Fábricas de Software.

El presente trabajo se ha dividido en dos capítulos, en el Capítulo 1, se muestra la fundamentación teórica. Aborda conceptos relacionados con el tema de Fábricas de Software, proceso de desarrollo, métodos ágiles, entre otros. Se mencionan los modelos de Fábricas de Software más representativos a nivel internacional, las características esenciales de cada uno y su experiencia. El Capítulo 2, constituye el desarrollo del marco de Fábricas de Software basado en las buenas prácticas de los métodos ágiles de desarrollo. Se identifican y describe la interrelación entre los elementos del marco de Fábricas de Software propuesto, la guía del flujo de procesos para el desarrollo de software así como la valoración del marco de Fábricas de Software teniendo en cuenta la arquitectura empresarial y las características de las metodologías de desarrollo de software.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordarán un grupo de conceptos importantes relacionados con la obtención de un marco de desarrollo de una Factoría de Software, basado en las buenas prácticas de los métodos ágiles. Se describen los modelos más representativos encontrados durante la revisión bibliográfica, sobre Fábricas de Software y las metodologías para el desarrollo de software. También se presentarán definiciones de marco, los enfoques y niveles que caracterizan a las Fábricas de Software; así como las diferentes tecnologías a utilizar.

1.2. Conceptualizaciones de la investigación

Definición de Marco

Según el diccionario de la Real Academia Española (*Diccionario de la Real Academia Española* 2008) marco se define como:

- Es un ambiente o paisaje que rodea algo.
- Límites en que se encuadra un problema, cuestión o etapa histórica.
- Patrón o tipo por el cual debían regularse o contrastarse ciertas cosas.
- Conjunto de circunstancias, ámbito. Ej.: El marco de unas negociaciones.

Marco de desarrollo

El Marco de Desarrollo no es más que una configuración para amoldarse a las necesidades del sistema. Su objetivo fundamental consiste en proveer ayuda y soporte a los miembros del proyecto de desarrollo de software. (MERINDE 2008)

Un marco de desarrollo es la configuración de elementos en el que se expresan los contratos y la interrelación de las partes en si de los elementos que lo integran. Se definen cada uno de los pedazos (que son los elementos) defino las interrelaciones entre ellos (no ha nivel de proceso) y las interrelaciones de las dependencias (yo produzco tu produces) y por último se define el contrato de trabajo de cada una de las cosas.

Factoría

Se denomina "Factoría", de forma genérica, a una fábrica o complejo industrial (Diccionario de la Real Academia Española 2008)

Factoría de software

El concepto de Software Factory o Fábrica de Software es un modelo de servicios cuya forma de trabajo es una analogía de los procesos de producción industriales, permitiendo una reducción de los niveles de incertidumbre en el desarrollo y mantención de proyectos de software. Ofrece una serie de beneficios directos sobre la confiabilidad y nivel de satisfacción de los productos entregados, proporcionando una mayor claridad presupuestaria y un calendario acotado de los proyectos e incidencias asociadas a las aplicaciones de software de la empresa. (LABORDA 2007)

“La fábrica es una organización habitada por personas comprometidas en un esfuerzo común, el trabajo es organizado de una manera o de otra, se usa la estandarización para la coordinación y formalización, y la sistematización es importante.”(AAEN and MATHIASSEN 1997)

La Fábricas de Software refiere el sentido de producir con rapidez y calidad a través de procesos conocidos, repetibles y gerenciables, y principalmente, mejorables continuamente, no sólo por la incorporación de técnicas y herramientas en el desarrollo del software, sino porque se mantiene constantemente el objetivo sobre el mejoramiento del proceso de producción y cada uno de los pasos que esto acarrea. Aaen, Bøttcher y Mathiassen indican que el "término fábrica indica un compromiso a largo plazo, esfuerzos integrados, por encima de proyectos individuales para mejorar las operaciones relativas al software".

Entre los significados de la Fábrica de Software, mencionan Aaen, Bøttcher y Mathiassen, "mejorar la efectividad del proceso, reducir la cantidad de trabajo y reusar el ciclo de vida de los productos". De forma tal que se puedan obtener mejores resultados en menor tiempo y con menos costos, pero además que sea una industria en la cual las actividades del desarrollo de software sean predecibles, lo que significa que los costos estimados y los compromisos de cronograma pueden ser satisfechos y confiables, logrando que la capacidad del proceso sea conocida, llevando una mejora continua y concentrándose en mejorar el proceso. Donde el conocimiento y las habilidades para mejorar están establecidos.(AAEN and MATHIASSEN 1997)

El proceso fabril tradicional utiliza la técnica de línea de producción y montaje con el objetivo de reducir los costos de producción por la vía de la estandarización de productos y procesos. La producción por montaje se caracteriza por encadenar secuencias de procesos que

convergen hacia una línea continua en la que se ensamblan los productos finales. Entre las industrias que trabajan por montaje se cuentan algunas de las actividades productivas de mayor relevancia para la economía actual, principalmente las mecánicas: automóviles, motores, tractores, electrodomésticos, electrónicos, etc.

Arnoldo Díaz, director general de Certum plantea que “el fundamento para establecer una FS, fue basado en tratar de obtener los beneficios que las líneas de producción industrial produjeron en la calidad de los productos así como la productividad lograda.”(OLAVARRIETA and PADRÓN 2006)

Se decidió que es un pilar fundamental para su éxito que las Fábricas de Software se nutran de esta experiencia, aplicándola a su producción a través de la especialización del trabajo por líneas de producción, donde cada una tenga un proceso y un flujo de trabajo correctamente definidos. Conllevando así a lograr una mayor productividad y adaptabilidad al cambio.

1.3. Antecedentes de una Fábrica de Software.

A fines de los años 1960 e inicios de 1970, surge en la industria del software el concepto de FS, como una respuesta a la necesidad de aliviar la incertidumbre que se tenía en el desarrollo de los proyectos de software en aspectos como:

- Confiabilidad de los productos.
- Mantener en presupuesto y calendario los proyectos de desarrollo de software.
- Falta de una definición y seguimiento adecuado a los procesos de producción, así como un medio efectivo de medir su desempeño y la productividad de las personas que lo ejecutan.
- Falta de estandarización en los métodos y herramientas empleados en los procesos. Además de la nula reutilización de los productos que los mismos procesos de producción generan.
- Falta de herramientas para hacer rastreables los productos que generan los procesos.

El primer antecedente de una Fábrica de Software lo encontramos en 1968 cuando General Electric desarrollo una Fábricas de Software para mejorar la productividad de sus programadores mediante el uso de herramientas estandarizadas, una interfaz gráfica de usuario y una base de datos con datos financieros y de control de la administración.

El concepto de Fábricas de Software, fue originalmente propuesto para mejorar la productividad del desarrollo de software hacia las herramientas estandarizadas, métodos y componentes reutilizables.

La necesidad de una estructura y de prácticas de software predecibles ha llevado a muchos a buscar la forma de imponer rigor y disciplina en sus operaciones para desarrollar software. De esta manera ha crecido el interés en introducir algunos métodos que son usados en fábricas convencionales, para el control de desarrollo de software. Claramente, la esperanza es que al imitar a la fábrica, la calidad y productividad del desarrollo de software se aproximará a los otros elementos del negocio.(VALDELAMAR 2007)

Con el auge de la industria del software el concepto de Fábricas de Software ha evolucionado a medida que se han definido técnicas y métodos, además de la aparición de diversas herramientas de software. Aunque el término Fábricas de Software no es nuevo, desde hace varias décadas ha adquirido una especial relevancia con el desarrollo de modelos. A continuación se enuncian varios conceptos de Fábricas de Software dados por distintos autores: Cusumano plantea en 1989 que: “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 Fábricas de Software. El desarrollo de una factoría implica que las buenas prácticas de Ingeniería de Software sean aplicadas sistemáticamente.”(CUSUMANO 1989)

Cantone enuncia en 1992 que “Una Fábricas 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.” (CANTONE 1992)

Una Fábricas de Software, tal y como se define 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. Por lo tanto, las FS se centran en el desarrollo de sistemas similares promoviendo la reutilización de arquitecturas, componentes software y conocimiento. (KUTVONEN 2008)

1.4. Objetivos de una Factoría de Software.

El enfoque de Fábricas de Software viene a formalizar todos los procesos o 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, abarcando desde 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 en la que está ocupado. Entre los principales objetivos trazados por Fábricas Software están:

- Industrializar el desarrollo de sistemas de software.
- Producción de software a gran escala.
- Lograr una alta productividad en el desarrollo de software.
- Establecer líneas de producción.
- Mejora continua de los procesos.
- Estimación de costos y plazos extremadamente precisa.
- Reducción de los costos de producción.
- Lograr un buen control de la calidad.
- Especializar al profesional en una tarea específica del proceso, concentrando sus esfuerzos en dicha tarea.

1.5. Línea de Producción y Montaje

Una línea de productos es un tipo de aplicación que puede agruparse mediante una familia de productos como por ejemplo sistemas de software de seguros, software para entidades bancarias, de control aéreo y cualquier otro. Es decir, pueden agruparse de distintas maneras, por segmento de negocios, tipo de soluciones, plataforma, etc. La importancia que tiene es poder identificar aquellas características comunes. Tiene tres aspectos claves los cuales son: los alcances, la variabilidad y la extensibilidad. Los alcances determinan que tipo de soluciones pueden construirse a partir de la línea de productos y define por lo general la arquitectura base para la familia de aplicaciones, la variabilidad hace referencia a las partes comunes definidas en los alcances y aquellas que son variables por medio de configuración de la línea y la extensibilidad determina los puntos en los que es posible añadir funcionalidades.(ONTIVERO 2007)

La línea de producción y montaje o cadena de montaje es una técnica utilizada en el proceso fabril tradicional. Esta es una técnica que fue Henry Ford el inventor principal, responsable de que estas prácticas se generalizaran en el desarrollo industrial. Consiste en instalar una cadena de montaje a base de mecanismos de transmisión y guías de deslizamiento que van desplazando automáticamente las unidades en montaje por los puestos de trabajo donde sucesivos grupos de operarios van realizando tareas específicas hasta lograr la integración del producto final. Este sistema de piezas intercambiables fue el aplicado por Ford en la fabricación de automóviles, revolucionando así la industria automovilística, por lo que a partir de entonces esto ha sido aplicado a toda la industria en general, es una técnica que reduce al máximo los costos de producción por la vía de la estandarización de los productos y los

procesos. Estos descubrimientos se convirtieron en un importante aumento de la productividad de la empresa.

En la línea de producción y montaje Henry Ford relacionó importantes principios de administración científica y organización racional del trabajo. Como fue la división del trabajo, especializando a cada trabajador en una tarea específica. La estandarización de los métodos de trabajo, materias primas, herramientas y los componentes con el propósito de aumentar la eficiencia y eficacia. Otro principio fue el de la productividad aumentando la capacidad de producción del hombre en el mismo periodo de tiempo.

Estudiando la declaración de Ford nos permite llegar a la conclusión de que el software se podría industrializar permitiendo producir software en grandes cantidades para un determinado sector del mercado. Donde la calidad de los recursos humanos empleados en la producción de software influye directamente en la calidad final del producto. Donde la simplicidad en el proceso de desarrollo influye y determina el precio final de la productividad, adaptabilidad del proceso y el producto.(ONTIVERO 2007)

1.6. Modelos de Factoría de Software

Los modelos de fábricas de Software son la forma en que se han llevado a la práctica el enfoque de Fábricas de Software por distintas empresas y entidades que los han adoptado. Se describen una selección estos modelos más representativos encontrados en la bibliografía consultada, los mismos servirán como base de la investigación a partir de los elementos más importantes identificados en cada uno de ellos. Se abordarán cinco modelos, los mismos son:

- Modelo basado en la norma ISO 9001 y CMM.
- Modelo Eureka.
- Modelo Clasificadorio.
- Modelo propuesto por Basili.
- Modelo Replicable.

En el Anexo 1 puede verse de forma resumida una comparación entre los modelos de Factoría de Software.

1.6.1. 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: Técnicas, Proceso, Trabajadores involucrados, Gestión de la factoría y Activos del proceso, herramientas y componentes de código. El Modelo basado en

la Norma ISO 9001 y CMMI es aplicable a cualquier Fábricas Software; pueden adaptarse las entidades que componen el mismo de acuerdo a las características y necesidades del entorno. En este modelo se hace una división de los elementos fundamentales de una Fábricas de Software en cinco entidades bien definidas las cuales son:

- Técnicas: Comprende el contexto de las técnicas que sirven de soporte al proceso de desarrollo, técnicas para la reutilización de software, para el desarrollo basado en componentes y otras técnicas utilizadas por la factoría.
- Proceso: Representa el proceso de desarrollo de software, los flujos de trabajo y actividades que componen el mismo.
- Trabajadores involucrados: Personas que actúan directamente en el desarrollo de software.
- Gestión de la factoría: Define la estructura organizacional de la Fábricas de Software, el proceso fabril y la gestión de calidad.
- Activos del proceso, herramientas y componentes de código: Entiéndase como activos del proceso modelos, 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.

La arquitectura propuesta por el modelo se puede observar a continuación en la figura 1.1.

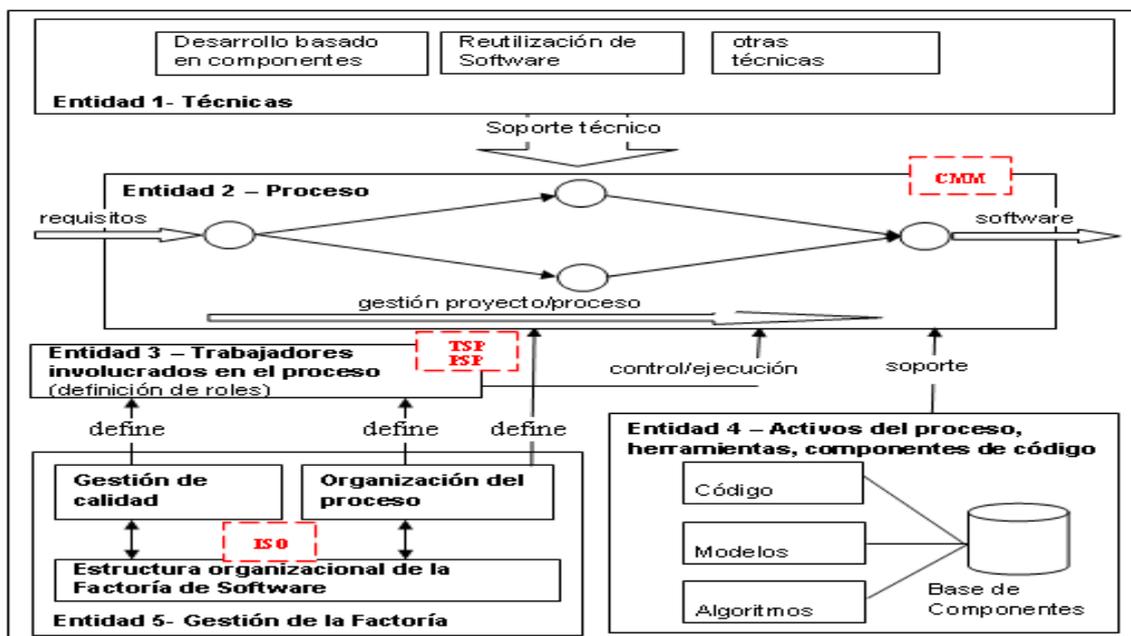


Figura 1.1. Arquitectura del Modelo basado en la norma ISO 9001 y CMM.

“En la misma se observa que la entidad Técnicas provee el soporte técnico y conceptual para la definición del proceso. Este es guiado por el estándar de calidad CMM, los requisitos de calidad para la organización de la factoría son definidos por la norma ISO 9001. El modelo

toma la norma ISO 9001 como un estándar utilizado en el contexto industrial cuyo enfoque está en el sistema de calidad organizacional, propone un conjunto de principios probados para mejorar la calidad final del producto mediante mejoras en la organización de la empresa. CMM es designado para la industria del software, de este modo las áreas claves proveen detalles importantes para la evaluación y mejora del proceso de desarrollo, su propósito es guiar a las organizaciones en la selección de estrategias de mejora determinando la madurez del proceso actual e identificando los puntos importantes que se deben estudiar y trabajar para mejorar tanto el proceso como la calidad del software.” (HOZ, Y. R. 2005)

“La entidad Gestión de la factoría define, a través de las sub-entidades Gestión de Calidad y Organización del Modelo de Proceso, los trabajadores involucrados en el proceso de desarrollo de software y sus roles. La sub-entidad Organización del Modelo de Proceso define características y organización del proceso de desarrollo de software. Los trabajadores involucrados son guiados por los modelos PSP (Personal Software Process) y TSP (Team Software Process). Los activos del proceso, las herramientas y los componentes de código dan soporte al proceso de desarrollo de software.”(HOZ 2005)

Lo más relevante del modelo presentado anteriormente es que define objetivamente y describe las entidades que forman el modelo y las relaciones que se establecen entre ellas, así como la aplicación de normas y técnicas de calidad usadas hoy en el mundo del software, aunque CMM es un estándar que se abolió en diciembre del 2005 por lo que una de las mejoras que se le propone es el uso de CMML, además no desglosa las actividades, objetivos y características de cada una de las entidades que lo conforman.

1.6.2. Modelo Eureka.

La iniciativa intergubernamental EUREKA ha movilizó a la comunidad europea para innovar. Creada en 1985, esta Red descentralizada permite a las pequeñas, medianas y grandes empresas, centros de investigación, universidades y administraciones nacionales reunir fuerzas para llevar a cabo proyectos de investigación y desarrollo (I+D) cercano al mercado a través de proyectos de colaboración transnacional.

Uno de los resultados más relevantes es el proyecto Eureka Software Factory.

“El modelo Eureka surgió como el proyecto Eureka Software Factory. El objetivo del proyecto es crear un mercado para productos CASE. En el mismo participan un conjunto de compañías europeas; tales compañías actúan en las siguientes áreas: manufactura de computadoras,

instituciones de investigación, producción de herramientas CASE y desarrollo de sistemas.” (HOZ, Y. R. 2005)

El modelo está compuesto de los elementos: proceso, reglas, herramientas e información, personas y computadoras, ver figura 1.2.

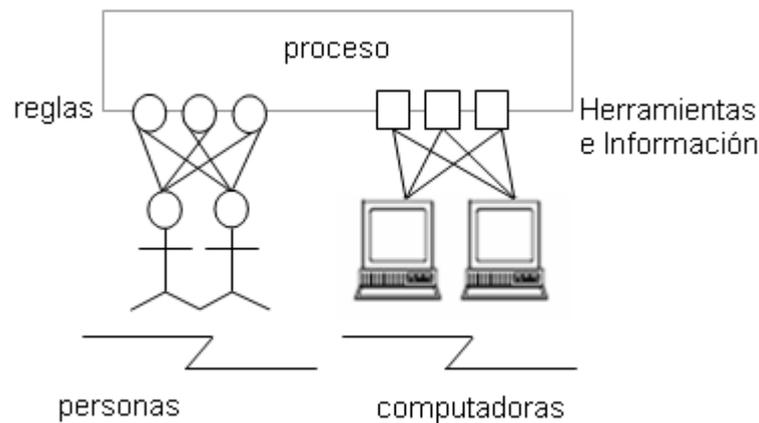


Figura 1.2. Arquitectura del Modelo Eureka.

Este modelo propone un proceso de desarrollo de software distribuido, siguiendo el enfoque software bus. Lo que permite que a través de las reglas se puedan unir los componentes realizados por distintos equipos para formar el producto.

En la figura 1.3 se muestra que utilizando este modelo los componentes pueden ser desarrollados por diversas FS en localidades diferentes, y su conexión se realiza de acuerdo a reglas establecidas. Esto permite además que cada área se pueda especializar en diferentes líneas de producción y que las factorías que usen ese modelo puedan compartir y reutilizar los componentes para la construcción del software. Además se clasifican los componentes en los de servicios y los de interfaz, reflejándose un desarrollo de software por capas.

La contribución de este modelo está en el desarrollo distribuido de software, da una visión de cómo se puede llevar la construcción de un producto software entre diferentes factorías, y después realizar la unión de los componentes elaborados por cada una para formar el producto final todo esto soportado por sistemas automatizados y un conjunto de reglas que establecen las condiciones para cada uno de los artefactos que se van obteniendo. Pero este modelo no enuncia el cómo se organiza la producción, el proceso y los desarrolladores, cómo gestionar un proyecto, cómo se define el proceso y no usa estándares de calidad.

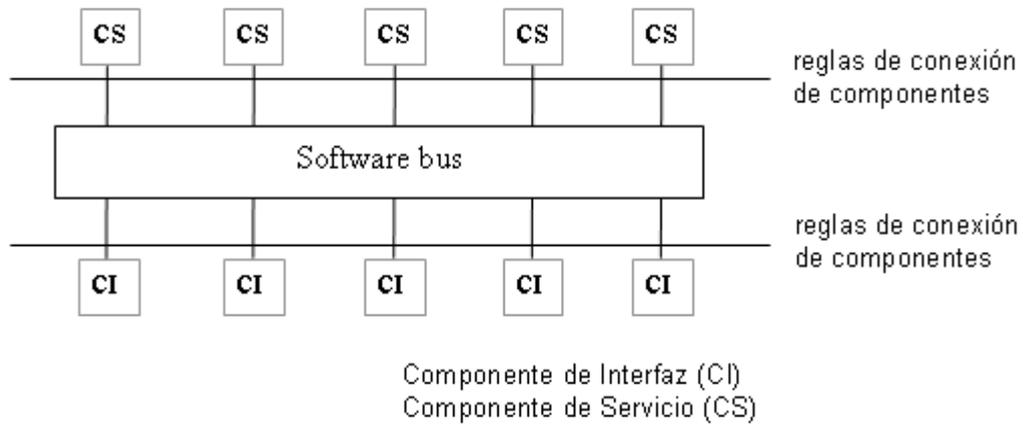


Figura 1.3: Enfoque Software bus en el Modelo Eureka.

1.6.3. Modelo Clasificadorio.

El Modelo Clasificadorio propuesto por Fernández y Teixeira 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. Una Fábricas de Software puede ser clasificada en:

- Factoría de Proyectos Ampliada.
- Factoría de Proyectos de Software.
- Factoría de Proyectos Físicos.
- Factoría de Programas.

En la figura 1.4 se puede observar la clasificación según el alcance del proyecto.

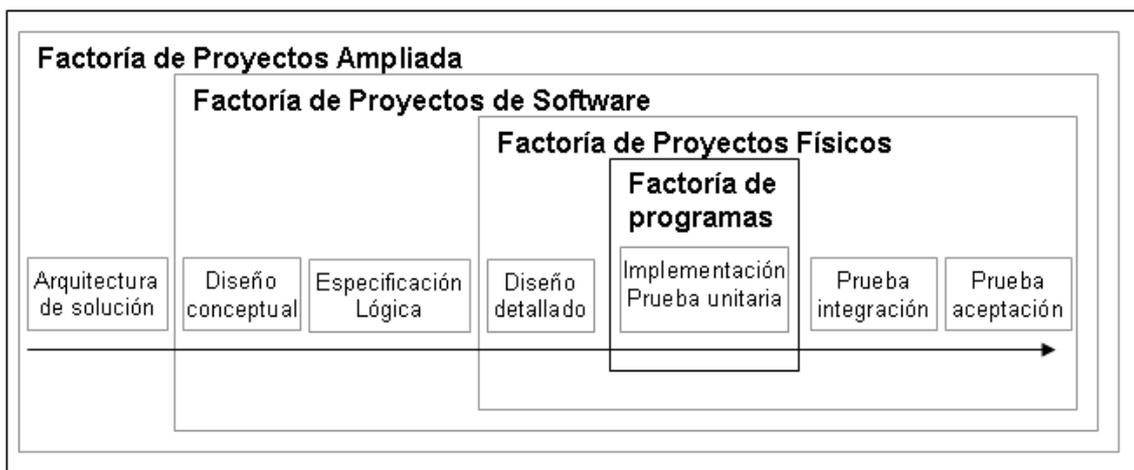


Figura 1.4.- Arquitectura del Modelo Clasificadorio

Una 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 cual se

ocupa en 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.

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.” (FABRI et al. 2008)

Este modelo aporta una clasificación para las factorías en dependencia de la magnitud del proyecto, da una visión de cómo puede ser clasificado en correspondencia a las fases del proceso de desarrollo que se lleven a cabo, así mismo da la idea de que en la factoría se realiza un ciclo de vida de un producto o parte de él y cómo se puede ir avanzando en este enfoque, ya que en un futuro se pudiera pasar a una factoría de mayor o menor alcance. Este modelo solo se enmarca en el proceso, no define las demás áreas involucradas en la producción, la gestión del proyecto, la organización de la producción y de los desarrolladores, el uso de herramientas para la automatización de los procesos de construcción, soporte y gestión, entre otras. (FABRI et al. 2008)

1.6.4. Modelo propuesto por Basili.

El presente modelo se plantea que un área de producción con características de Factoría debe estar basada en componentes y que los mismos pueden ser desarrollados por una factoría de componentes. O sea, propone dividir el área de producción en dos subáreas: la de producción de software y la de componentes.

Como muestra la figura 1.5, el modelo se divide en organización basada en proyectos de software (producción de software), y factoría de componentes (unidad de producción de componentes).

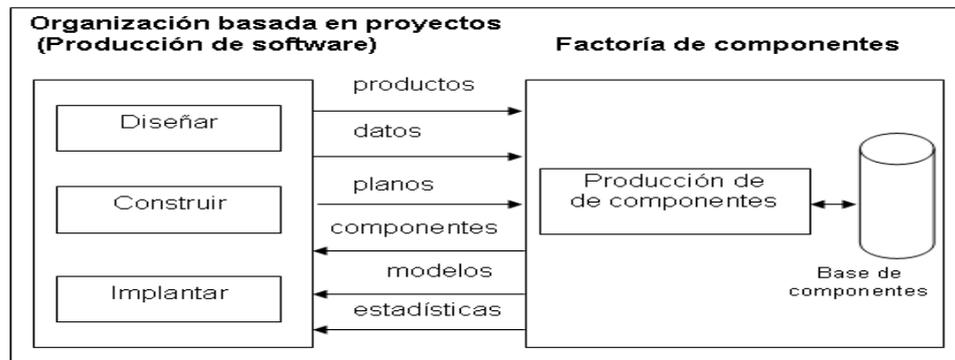


Figura 1.5.Arquitectura del Modelo propuesto por Basili.

“La organización basada en proyectos realiza las solicitudes de productos (componentes para la construcción del software), de datos (estadística para la estimación de costos y plazos) y 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.”(HOZ, Y. R. 2005)

“Comparando el presente modelo con la idea de línea de producción y montaje del proceso industrial, es posible notar también una fuerte presencia de intercambio e integración de piezas, en el caso de la línea de producción de software, esas piezas son denominadas componentes de software.” (HOZ 2005).

Este modelo se basa en la división de la factoría en dos unidades aumentando la eficiencia y especialización en la producción, y también se enfoca en la reutilización durante el desarrollo; para esto se propone tener una base de componentes reutilizables. Pero se encuentra incompleto pues solo describe la relación entre las dos subáreas y los procesos que se ejecutan entre ellas, aunque se pueden adaptar a las características de una determinada Fábrica de Software, pero solo se enmarca en el área de producción dejando sin definir las demás áreas involucradas en la producción, la gestión del proyecto, la organización de la producción y de los desarrolladores, el uso de herramientas para la automatización de los procesos de construcción, soporte y gestión, entre otras.

1.6.5. Modelo Replicable.

El modelo a describir fue concebido para ser aplicado a cualquier factoría. Es fácil de adaptar a cualquier entorno, a las necesidades y recursos de una factoría. Reúne en él las características más importantes de los modelos anteriores, es el que más detallado se encuentra en la bibliografía consultada.

Este modelo plantea que una FS debe poseer:

- 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 un proceso.
- 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.” (HOZ 2005).

El modelo basa su descripción en determinar las relaciones entre conceptos a diferencia del Modelo basado en la norma ISO 9001 y CMM que lo realiza entre entidades, ver figura 1.6.

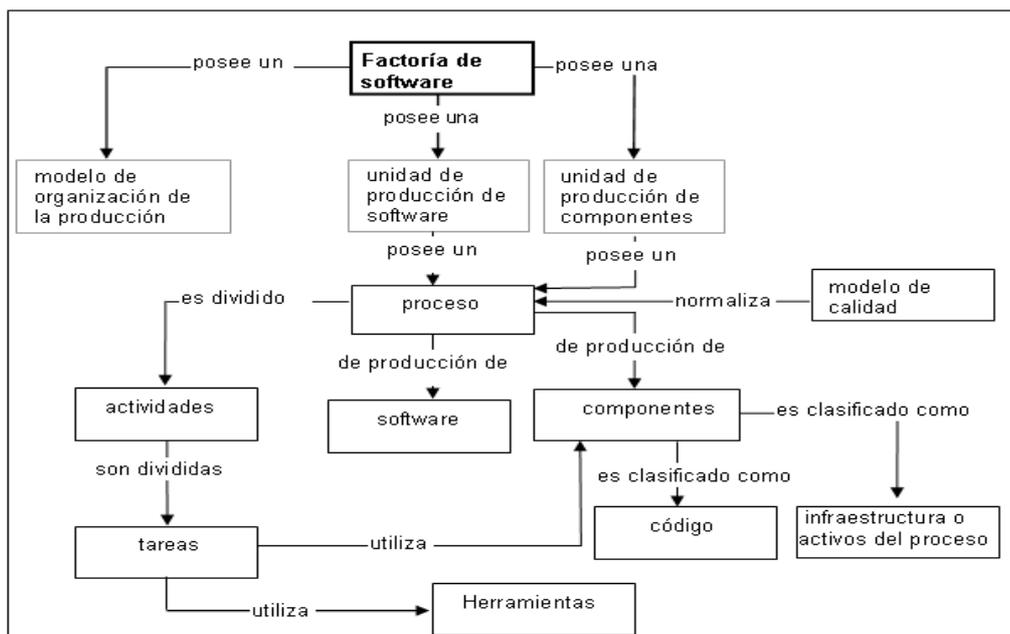


Figura 1.6. - Arquitectura del Modelo Replicable

En él se define la organización de la producción dividiendo la misma en cinco áreas y define las actividades que intervienen en cada una de ellas, los objetivos de los mismos y la relación que existe entre ellas. Estas son:

- Área de producción de análisis de sistema o modelado de negocio.
- Área de producción de diseño de software.
- Área de construcción de software.
- Área de producción de componentes de infraestructura o activos del proceso.
- Área de producción de componentes de código.

Donde 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.”(HOZ 2005).

Describe además las propuestas de roles a asignar para cada una de las actividades aunque no los organiza estructuralmente. Describe además las técnicas y herramientas a utilizar. Este modelo define mejor el proceso de producción pero no se enmarca en el uso de alguna de las metodologías estandarizadas, ni utiliza estándares de calidad, presenta el área de producción dividiéndolas como la propone Basili aportándole más actividades. Aún deja incompleta la gestión de proyecto.

1.7. Modelos propuestos en la UCI de Fábrica de Software

1.7.1. Modelo funcional de la Factoría de Software de la UCI para la Línea Carrofeur.

Brinda una guía con el fin de lograr una producción de software industrializada. Basada en las entidades y los mecanismos de control(HOZ, YANOSKY RIOS LA and VALDIVIA 2005).

En este modelo aun no abarca todos los temas relacionados con el enfoque de Fábrica de Software al ser estos muy amplios. Este propone la optimización y seguimiento de un modelo o marco que se acerque a la solución más óptima y productiva de proceso de desarrollo.

1.7.2. Modelo Organizativo para los Proyectos Productivos (MOPP).

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”.(NEGRÍN 2007)

Basada en los problemas existentes de esta organización pero aun así los objetivos de este trabajo no abarcan todos los temas relacionados con el enfoque de factoría de software se recomienda profundizar y darle continuidad a esta investigación.

1.7.3. Modelo de Factoría de Software aplicando Inteligencia para el Polo Productivo Gestión de Proyecto.

En el modelo aplicando inteligencia se hace una división de la factoría de software en seis entidades y sus relaciones(MARTÍNEZ 2007)

1. Bases tecnológicas: Comprende el contexto de las bases tecnológías y 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 depende de la metodología que se utilice para guiar el desarrollo del proyecto. Personas: Comprende el capital humano involucrado con el proceso de desarrollo de software, la estructura organizativa y los roles que ocupan, está dividida en dos sub-entidades: Gestores de la Factoría y Grupo de desarrollo.
3. 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.
4. Gestión de la Factoría: Comprende todos las áreas de la gestión de proyecto.
5. Centro de Inteligencia: Tiene el objetivo de utilizar herramientas de Vigilancia Tecnológica, Inteligencia Empresarial, Prospectiva que permitan la orientación estratégica.

La descripción de la arquitectura y composición del modelo se basa en que el resultado final de un proyecto es un producto, que toma forma durante su desarrollo gracias a la intervención de varios tipos de personas, que utilizan PSP y TSP para la planificación personal y en equipo. Estas personas están representadas en el modelo por la Entidad Personas, la que se divide en Grupo de desarrollo y Gestores de la Factoría. El Grupo de Desarrollo lo conforman las personas

1.8. Análisis comparativo de los modelos.

El análisis de los modelos de Fábricas de Software antes vistos, según las características especificadas para una Fábrica de Software, permitió adquirir elementos para seleccionar el modelo para el cual se le diseñará la estrategia. En el anexo#1 aparece una tabla comparativa que resume las características de los modelos antes vistos. El modelo basado en la norma ISO 9001 y CMM se puede adaptar según las necesidades de la factoría. Reflejando la necesidad de la reutilización de componentes y agrupando sus actividades por áreas. El modelo utiliza estándares para la calidad y la mejora continua del proceso.

El modelo Eureka al igual que el anterior, guía su proceso de desarrollo por reglas y tiene como característica principal que se enfoca a la producción de software de forma distribuida o por componentes. El modelo Clasificadorio y el propuesto por Basili se caracterizan por la profundización en la especialización de las factorías. El primero las define según el alcance del proceso de desarrollo mientras que Basili agrupa las actividades de una factoría en dos unidades, buscando la especialización y el aumento de la reutilización de los componentes producidos.

Los modelos restantes, el Replicable y el aplicando Inteligencia, tienen como característica esencial que se sustentan de los elementos más sobresalientes de los anteriores, definen con mayor precisión las actividades y tareas a realizar dentro de cada flujo de trabajo, aunque el modelo de Fábricas de Software aplicando Inteligencia se dedica a definir el área de gestión del proyecto, aspecto fundamental que establece el éxito de un proyecto de software. Este no propone ver como núcleo de la Fábrica de Software al Área Productiva. Tampoco establece la relación de las demás áreas de la fábrica. No se compone por Áreas de servicios, Área productiva y no tiene un observatorio que controla y verifica todo el proceso de desarrollo.

Existen enfoques sobre las Fábricas de Software que resulta interesante revisar y en las siguientes figuras 1.7 y 1.8 se puede ver a través de comparaciones.

Comparación de modelos de determinadas fábricas en el mundo			
	Desarrollada por	Objetivo	Estrategia
Fábrica Industrializada (Japón)	Toshiba	Aumentar la calidad y productividad en el desarrollo y el mantenimiento.	Infraestructura Combinación de infraestructura física, organizacional y basada en herramientas
Fábrica Genérica (Europa)	Proyecto Eureka	Ambientes de desarrollo de software integrados hechos a la medida.	Basada en herramientas Estandarización de componentes y homogeneización de componentes y procesos.
Fábrica de componentes basada en la experiencia (EU)	Software Engineering lab.	Mejor efectividad en el proceso, menos trabajo y más reuso	Continua Mejora basada en la experiencia y automatización flexible.
Organización Madura de Software (EU)	US DoD (basada en CMM)	Un proceso efectivo, predecible, y susceptible de mejora.	Incremental Mejora basada en escalar niveles de madurez en el proceso

Figura1.7. Comparación modelos de fábricas de software en el mundo

Comparación de modelos propuestos en la UCI			
Modelo funcional de la Factoría de Software de la UCI para la Línea Carrofeur.	UCI	Brinda una guía con el fin de lograr una producción de software industrializada	Basada en las entidades y los mecanismos de control.
Modelo Organizativo para los Proyectos Productivos (MOPP).	UCI	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".	Basada en los problemas existentes de esta organización.
Modelo de Factoría de Software aplicando Inteligencia	UCI	Recoge los resultados de la evaluación teórica de la adopción del enfoque en la UCI.	Para esto se tomaron los aspectos positivos de los modelos anteriores y se adaptaron al contexto del entorno de desarrollo específico de la UCI.
Modelo de Factoría de Software aplicando Inteligencia para el Polo Productivo Gestión de Proyecto.	UCI	Propone hacer uso de modelos de factoría para organizar la producción de software en cada Facultad.	Los objetivos de este trabajo no abarcan todos los temas relacionados con el enfoque de factoría de software.

Figura 1.8 Análisis de los Modelos de Factorías en la UCI

Se ha podido constatar mediante el estudio de las tesis ubicadas en la biblioteca de la UCI que ninguno de los modelos existente en el mundo pueden ser aplicados a la UCI, incluso los que se proponen aplicar para la UCI, además ninguno concibe la formación de especialistas de la informática dentro del mismo proceso de desarrollo y no se presenta un área de formación ubicada dentro de la fábrica, en algunos casos se habla de modelo de factoría dentro del proyecto, precisamente en el marco de desarrollo de fábrica de software a proponer se pretende eliminar la palabra proyecto y comensar a utilizar la palabra fábrica.

Mediante la observación de los proyectos se ha podido apreciar que no está aplicado en la práctica un modelo de factoría de software y de estar aplicado estos resultados no están publicados.

1.9. Experiencia Internacional

Como parte de la investigación se han encontrado varias organizaciones que ponen en práctica el enfoque de FS. Estas empresas se ubican en varios países y han obtenido una serie de resultados que les ha reportado grandes beneficios en todos sus sectores.

Ivan Aaen afirma que “la primera compañía en el mundo que nombró una organización del software como una factoría fue Hitachi en 1969, la segunda fue implantada entre los años 1975-1976 bajo el nombre de Corporación de Desarrollo de Sistemas, por un líder americano en el campo del software personalizado.(AAEN and MATHIASSEN 1997)

En la década de los 70 se destacaron varias empresas japonesas que adoptaron trabajar con Factorías de Software, entre ellas están Toshiba y Fujitsu.

Otro ejemplo es AZERTIA, un Grupo multinacional español de empresas de tecnologías de la información que pertenece a La Corporación IBV (Instituto de Biomecánica de Valencia). Tiene más de 3.300 profesionales que se distribuyen por nueve países (España, Portugal, EE.UU. Argentina, Brasil, Colombia, México, Puerto Rico y Venezuela).Esta investiga e innova mediante alianzas y acuerdos estratégicos con los principales proveedores de tecnologías líderes en el mercado, que le permiten complementar sus soluciones y adaptarlas a las necesidades concretas de cada cliente. Surgió en el 2001 como la integración de Centrisa, Keon y TeleInformática, dependencias de IBV también.(AZERTIA 2008)

Ibermática es una de las principales compañías de servicios en tecnologías de la información en el mercado español. Su actividad se centra en las siguientes áreas: consultoría TIC,

equipamientos e infraestructuras, integración de sistemas de información, outsourcing e implantación de soluciones integradas de gestión empresarial para pequeñas y medianas empresas (PYME). En sus más de treinta años de experiencia en el sector de las TIC, Ibermática se ha distinguido por la búsqueda permanente de la excelencia y una apuesta decidida por la calidad y la innovación. Surgió en 1973 para brindar servicios informáticos al mercado español y desde entonces evoluciona constantemente para adaptarse a los cambios del mercado informático mundial. Por esto, en el 2005 Ibermática creó el Instituto Ibermática de Innovación, un centro de investigación que dirige la aplicación de modelos de innovación tanto en la propia Ibermática como en otras organizaciones, a través de la implantación de soluciones y la creación de una metodología para medir la innovación. Este instituto basó su funcionamiento en el enfoque de Fábrica de Software, aplicando de forma sistemática los conceptos más avanzados en materia de estandarización e industrialización del software, permitiéndole a Ibermática ganar mayor prestigio a nivel nacional e internacional. (IBERMATICA 2008)

Vector Software Factory es otra de las empresas que han aplicado el enfoque de FS como solución para sus procesos. Esta red de producción de soluciones de software se basa en el desarrollo de componentes y construcción de componentes de software avanzados, basados en componentes reutilizables. Surgió en el 2000 con el objetivo de desarrollar una Red de Centros de Producción de Componentes Software orientada al desarrollo y mantenimiento de componentes y soluciones complejas para empresas y organizaciones de diversos sectores. Centra su atención en el cliente por lo que tiene un equipo de consultoría directamente donde éste radica y dispone de centros de producción remotos con la infraestructura necesaria para el desarrollo de las aplicaciones.(VECTORSF 2008)

Desde la implantación de la primera Fábrica de software Hitachi hasta la fecha se puede observar que las empresas que han adoptado este enfoque han obtenido importantes beneficios, sobre todo en el cumplimiento de los costos y tiempos estimados, la calidad de los productos y la satisfacción de sus clientes. Estas empresas cuentan con equipos especializados y repositorios de componentes reutilizables(RAMOS *et al.* 2002).

1.10. Ventajas en la aplicación de un modelo de factoría.

Los modelos de factorías proporcionan una guía metodológica completa a los desarrolladores de los proyectos. Con las FS se precisa de forma clara los pasos que deben llevarse a cabo para producir software a gran escala. Se recomiendan explícitamente líneas de productos, frameworks de implementación, patrones de diseño lenguajes de modelado, etc. Las FS integran muchas de las aéreas de La Ingeniería de Software debido a que han

sido investigadas y llevadas a la práctica. Las principales actividades que promueven las FS son la construcción de familias de software similar, la cual soporta el análisis y diseño de una arquitectura común para un conjunto de sistemas y el desarrollo de un frameworks que soporte esta arquitectura. También se desarrollan lenguajes de modelado y herramientas específicas para el dominio, que permiten a los desarrolladores utilizar estos lenguajes para describir los requisitos específicos de un miembro de la familia de sistemas. Estas también hacen énfasis en el uso de una planificación basada en restricciones y guía activa que define que todos los pasos del proyecto de desarrollo deben realizarse de acuerdo a un proceso bien definido y adaptado al dominio. Otro aspecto a tener en cuenta es el ensamblado de componentes que especifica que la construcción de un nuevo sistema supone el uso, ensamblado o configuración de los componentes proporcionados por el frameworks.

1.11. Proceso de desarrollo de Software

1.11.1. ¿Qué es un proceso de desarrollo?

Es importante que para poder lograr el éxito dentro de una FS y lograr una mayor productividad y calidad potenciar bien el desarrollo de software, el cual ha pasado a ser el factor determinante en la producción de software, puesto que define el proyecto en su totalidad. A continuación se muestran varios conceptos de proceso de desarrollo dados por diferentes autores: Pressman(PRESSMAN 1992) expresa en 1992 que: “La capa de proceso es el fundamento de la Ingeniería de Software (IS).

El proceso es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la IS. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se producen resultados del trabajo, se establecen hitos, se asegura la calidad y se gestiona el cambio de manera adecuada.” (PRESSMAN 1992)

En 1998 Isabel Ramos plantea que: “El proceso de desarrollo de software puede definirse como un conjunto de herramientas, métodos y prácticas que se emplean para producir software.” (CARREIRA *et al.* 2008)

Basado en las definiciones anteriores este trabajo define el proceso de desarrollo como aquel en el cual se toman las necesidades o requisitos que persigue el cliente, y se definen a partir de una metodología diferentes actividades, tareas, roles y herramientas que serán empleados para construir el producto final.

1.11.2. Principales características

De acuerdo al software que deseemos crear, debemos tomar el modelo o metodología necesaria para su desarrollo. El proceso se puede definir como un marco de trabajo de las tareas que se requieren para construir software de alta calidad. Se podría decir que la ingeniería del software tal vez pueda ser como un sinónimo de lo que es el "proceso", porque el proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería de software.

Un proceso de software se puede caracterizar definiendo actividades del marco de trabajo, con independencia de su tamaño o complejidad. Un conjunto de tareas que permiten que las actividades del marco de trabajo se adapten a las características del proyecto del software y a los requisitos del equipo del proyecto. En los últimos años, se ha dicho mucho énfasis en la madurez del proceso y para determinar el estado actual de madurez, según el SEI (SEI 2008) se realiza un cuestionario de evaluación y un esquema de grados que determina la conformidad con un modelo de capacidad de madurez.

El desarrollo de Software es un proceso complejo y a menudo difícil que requiere la síntesis de muchos sistemas. Desde el modelado y diseño hasta el código, administración del proyecto, pruebas, despliegue, administración de cambios y más.

1.12. Metodologías de desarrollo de Software.

Dentro del proceso de desarrollo un papel muy fundamental lo conforman las diferentes metodologías de software por tanto un proceso de software detallado y completo suele denominarse "Metodología". Las metodologías se basan en una combinación de los modelos de proceso genéricos, ejemplos de estos son: cascada, evolutivo, incremental, etc... Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término "método" para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una o varias actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis o diseño.

En el Anexo 2 se expone un cuadro comparativo de acuerdo con algunos criterios básicos para la selección de un modelo de proceso, la medida utilizada indica el nivel de efectividad del modelo de proceso de acuerdo al criterio (Por ejemplo: El modelo Cascada responde con un nivel de efectividad Bajo cuando los Requisitos y arquitectura no están predefinidos). Es importante que cada proyecto de software requiere de una forma de particular de abordar el

problema. Las propuestas comerciales y académicas actuales promueven procesos iterativos, donde en cada iteración puede utilizarse uno u otro modelo de proceso, considerando un conjunto de criterios, por ejemplo: grado de definición de requisitos, tamaño del proyecto, riesgos identificados, entre otros.(Soft, 2002)

La comparación y clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. A grandes rasgos, si tomamos como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, podemos clasificar las metodologías en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o como suelen llamarse peyorativamente Metodologías Pesadas, o Peso Pesado. Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.(BALZER 1985; FABRI *et al.* 2008).

En el Anexo 3 se pueden observar varias de las características que distinguen a estos dos grupos.

A continuación se revisan brevemente cada una de estas categorías de metodologías.(BALZER 1985)

1.12.1. Metodologías tradicionales (no ágiles)

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales. Aunque en el caso particular de RUP, por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto mediante su configuración previa a aplicarse, realizando una configuración adecuada, podría considerarse ágil.

1.12.2. Metodologías ágiles

Un proceso es ágil cuando el desarrollo de software es incremental cuando hacen entregas pequeñas de software, con ciclos rápidos, cooperativo cuando el cliente y los desarrolladores trabajan juntos constantemente con una buena comunicación, sencillo mientras el método en sí mismo sea fácil de aprender y modificar, y este bien documentado, y adaptable cuando permita realizar cambios de último momento.(ABRAHAMSSON 2002)

Las metodologías ágiles de desarrollo de software son una agrupación de las prácticas tradicionales pero llevadas al extremo, tomando la esencia y aplicándolas buscando la calidad en el desarrollo desde el inicio, entregas oportunas y la entrega final del sistema, teniendo en cuenta el soporte, mantenimiento, auditoría y capacitaciones al usuario final.

Principios Agiles

Según el manifiesto ágil(BECK and BEEDLE 2001), los principales aspectos a tener en cuenta son:

- La mayor prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
- Aceptar requisitos cambiantes, incluso en etapas avanzadas. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregar software frecuentemente, con una periodicidad desde un par de semanas a un par de meses, con preferencia por los periodos más cortos posibles.
- Los responsables de negocio y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
- Construir proyectos con profesionales motivados. Dándoles el entorno y soporte que necesitan, y confiando en ellos para que realicen el trabajo.
- El método más eficiente y efectivo de comunicar la información a un equipo de desarrollo y entre los miembros del mismo es la conversación cara a cara.
- Software que funciona es la principal medida de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Esponsors, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y los buenos diseños mejoran la agilidad.
- Simplicidad, el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de equipos que se autoorganizan.

- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo, entonces mejora y ajusta su comportamiento de acuerdo a sus conclusiones.

Ejemplos de Metodologías Ágiles:

Entre las metodologías ágiles identificadas se encuentran:

- Extreme Programming (XP).
- Scrum.
- Familia de Metodologías Crystal.
- Feature Driven Development.
- Proceso Unificado Rational, una configuración ágil.
- Dynamic Systems Development Method.
- Adaptive Software Development.
- Open Source Software Development.

Extremen Programming (XP):

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

La programación extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.(ESCRIBANO 2002)

La metodología XP fue desarrollada por Kent Beck. El cual dijo: «Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» Kent Beck. (ESCRIBANO 2002)

Esta surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos, se plantea como una metodología a emplear en proyectos de riesgo y aumenta la productividad.

Roles XP

Los roles son los siguientes(CANÓS and LETELIER 2008):

Programador. El programador escribe las pruebas unitarias y produce el código del sistema.

Cliente. Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

Encargado de pruebas (Tester). Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker). Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (Coach). Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor. Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (Big boss). Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Proceso XP

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.(ESCRIBANO 2002)

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

El juego de la planificación. Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de tres meses.

Metáfora: El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

Diseño simple: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

Pruebas: La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.

Refactorización (Refactoring). Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.

Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

Propiedad colectiva del código. Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

Cuarenta horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.

Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

Estándares de programación: XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras, donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica para un análisis histórico de ideas y prácticas que sirven como antecedentes a las utilizadas por las metodologías ágiles. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

Adaptive Software Development (ASD)

Adaptive Software Development(PALACIO 2006). Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

SCRUM

SCRUM(PALACIO 2006): Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos diez años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Este maximiza utilidad y en Scrum se construye primero la funcionalidad que resulte en el mayor beneficio para el negocio y se evita en todo momento desarrollar cosas que no serán de utilidad para el cliente. De esta manera, se puede desarrollar un sistema en menos tiempo evitando el trabajo innecesario.

Scrum tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa. Por

otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta en un incremento en la motivación de los integrantes del equipo.

Crystal Clear

Crystal Methodologies(PALACIO 2006). Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear.

Microsoft Solutions Framework (MSF)

Es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos, se ha convertido posteriormente en un modelo práctico que facilita el éxito de los proyectos tecnológico(GATTACA 2008).

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.

- Framework que describe las mejores prácticas en términos de principios básicos, modelos conceptuales, y disciplinas.
- Provee las bases descriptivas desde las cuales puede derivar cualquier metodología específica.
- MSF 3.0 contiene ocho principios básicos:
 - Promover comunicaciones abiertas.
 - Trabajar para una visión compartida.
 - Fortalecer los miembros del equipo.

- Establecer responsabilidades claras y compartidas.
 - Focalizarse en agregar valor al negocio.
 - Permanecer ágil, y esperar los cambios.
 - Invertir en calidad.
 - Aprender de todas las experiencias.
- MSF 4.0
- También un Framework descriptivo similar en muchos aspectos, pero la gran diferencia es incluye dos metodologías prescriptivas:
 - *MSF para el desarrollo de Aplicaciones Ágiles.*
 - *MSF para el proceso de mejora CMMI.*
- MSF 4.0 se denomina metamodelo, para evitar confusiones.

Metodología tradicional RUP

Esta se centra en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán. Dentro de estas metodologías se destaca RUP (Rational Unified Process).

Jacobson(JACOBSON et al. 2000) utiliza para esta metodología el Lenguaje Unificado de Modelado (UML) y se basa en tres ideas básicas: guiado por casos de uso, centrado en la arquitectura y el desarrollo es iterativo e incremental.

Después de analizar las metodologías anteriores y el artículo "Procesos activos: Una nueva era para el desarrollo de software de calidad"(PASCO et al. 2008) donde se hace un estudio comparativo entre las características de las metodologías de Software tradicionales y las no tradicionales, además que en el desarrollo de software no existe una metodología universal sino que toda metodología debe ser adaptada al contexto del proyecto. Se pudo finalmente hacer la propuesta de marco de Fábrica de Software.

1.13. Conclusiones del Capítulo 1.

En este capítulo se han seleccionado los conceptos fundamentales para el desarrollo de la investigación sobre la propuesta de marco de Fábrica de Software, basado en las buenas prácticas de métodos ágiles, para el desarrollo de software, concluyendo en este sentido que

el enfoque de Fábrica de Software busca perfeccionar el proceso actual aplicando principios de ingeniería de software a la producción.

Se analizaron varios modelos donde cada uno presenta características importantes y están orientados a perfeccionar la producción. Se mencionan las experiencias que han tenido algunas empresas trabajando con el enfoque de Fábrica de Software, principalmente en la reducción de los costos de producción, tiempo de desarrollo y la satisfacción de sus clientes por el aumento de la calidad del producto. Se estudiaron conceptos relacionados con la Fábrica de Software, secuencia de pasos lógicos para alcanzar los objetivos propuestos. Se investigó sobre varias metodologías de desarrollo ágiles y tradicionales.

CAPÍTULO 2. DESARROLLO DEL MARCO DE FÁBRICA DE SOFTWARE

2.1. Introducción

En este capítulo se presenta un marco de desarrollo de software el cual es un esquema organizativo que busca la generalización de las actividades y la especialización de las mismas, para así lograr una mayor calidad elevando la eficiencia y la eficacia al aumentar la productividad en el proceso de desarrollo y la adaptabilidad de los productos.

Este marco se basa en las principales tendencias de desarrollo de software del mundo. Cuya esencia es adaptarse a las características UCI a partir de la problemática en cuestión. Fue consultado CMMI y PMBOOK entre otros, como referencia de que elementos hay que tener en cuenta a la hora de desarrollar un proceso de desarrollo de software.

2.2. Marco Fábrica de software. Aproximación práctica del proceso productivo

El marco de desarrollo propuesto se basa en la vista empresarial y en las características de la universidad como industria. La razón por la que se propone un nuevo enfoque es debido a que los modelos vistos anteriormente están diseñados para grandes empresas productoras de software, que poseen un alto nivel de desarrollo en la rama, y estas cuentan además con tecnología que avala dicho desarrollo; sin embargo, lo que se necesita en la UCI es un modelo que se adapte a sus particularidades como productor de software con un personal de desarrollo, inexperto, con muy poca experiencia.

Durante el proceso de desarrollo, un aporte importante en las fases de entendimiento y modelado de los requisitos del sistema lo constituye el enfoque basado en el modelado de los procesos de negocio. Mediante esta estrategia, se pretende que antes de empezar a desarrollar cualquier tipo de software, se deba realizar un análisis y un modelado exhaustivo de los procesos de negocio de la organización. De ésta forma, se conseguirá un mayor conocimiento de ellos y, además, mediante su análisis se podrá intentar mejorar el rendimiento de cada uno de ellos. La importancia de este enfoque se convierte en crucial desde el momento en que ayuda a las empresas a estar constantemente mejorando y adaptándose a los continuos cambios del mercado.

La ventaja que trae la creación de un marco Fábrica de Software para la organización del proceso de desarrollo de software está precisamente en que este marco busca la especialización y por tanto aumenta la adaptabilidad y la productividad de los productos,

mejorándose así la eficiencia y eficacia en el proceso de desarrollo, ya que este modelo es adaptativo y sobre todo muy novedoso. Este rompe con las concepciones y propone una nueva vía de desarrollo.

2.3. Identificación de los elementos del modelo de Fábrica de Software

En la confección del marco Fábrica de Software se han identificado todas las 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:

- Observatorio.
- Área productiva.
- Áreas de servicios.

Para la determinación de los objetivos, funciones, las entradas y las salidas de información de cada una de las áreas del marco de Fábrica de Software se tuvo en cuenta las opiniones de un grupo de directivos de la UCI que fueron entrevistados, estos son:

- Vicerrector de Tecnologías
- Vicerrectora de Formación
- Director de Contabilidad y Finanzas
- Dirección de Producción Infraestructura Productiva (IP)
- Directivos de Las Direcciones de Asuntos Legales
- Dirección de Capital Humano

2.3.1. Observatorio

Objetivos

- Observar y monitorear los indicadores de desempeño que defina la Fábrica de Software concreta.
- Observar y monitorear las factibilidades de mercado así como el impacto de los productos distribuidos por la Fábrica de Software.
- Tomar de decisiones relativas al funcionamiento de la Fábrica de Software y monitorear y controlar toda su actividad.

Funciones

- Monitorear y registrar toda la información que se genera en la Fábrica de Software.
- Coordinación entre las áreas de servicio y el área productiva de la Fábrica de Software.
- Brindar la información necesaria para la toma de decisiones sobre el funcionamiento de la Fábrica de Software.

Entradas

- Necesidades de capacitación (Proceden del Área Productiva, Gestión de la Calidad, Recursos humanos).
- Necesidades tecnológicas (Proceden del Área de Tecnología).
- Necesidades de investigación (Proceden del Área de Tecnología y de Formación).
- Informes de análisis legal (Proceden del Área de Asuntos Legales de todas las áreas de la Fábrica de Software).
- Informes de la evaluación de la calidad del trabajo realizado por el personal (Proceden del Área de Gestión de la Calidad).
- Planes de capacitación para (Proceden del Área de Formación).
- Información de la plantilla de la Fábrica de Software (Proceden del Área de Recursos Humanos).
- Informes del aprovechamiento de la jornada laboral y organización del trabajo (Proceden del Área de Recursos Humanos).
- Informes de la actividad de Recursos Humanos (Proceden del Área de Recursos humanos).
- Informes de análisis financiero (Proceden del Área de Economía).
- Informe de solicitud de nuevas líneas de investigación (Proceden del Área de Tecnología).
- Plan de gestión de la calidad (Proceden del Área de Gestión de la Calidad).

Salidas

- Aprobación de todas las solicitudes de capacitación (Dirigidas al Área Productiva de Gestión de la Calidad y de Recursos Humanos para Área de Formación).
- Aprobación de las necesidades tecnológicas (Dirigidas al Área de Producción para el Área de Tecnología).
- Aprobación de las necesidades de investigación (Dirigidas al Área de Formación para el Área de Investigación).

- Aprobación de todos los Informes de análisis legal (Dirigidas al Área de Asuntos Legales).
- Aprobación de Informes de la evaluación de la calidad del trabajo realizado por el personal (Dirigidas al Área de Gestión de la Calidad).
- Aprobación del Plan de capacitación (Dirigidas al del Área de Formación).
- Aprobación de la plantilla de la Fábrica de Software (Dirigidas al Área de Recursos Humanos).
- Aprobación de Informes del aprovechamiento de la jornada laboral y organización del trabajo (Dirigidas para el Área de Economía).
- Aprobación de Informes de la actividad de Recursos Humanos (Dirigidas al Área de Asuntos Legales).
- Aprobación de Solicitud de nuevas líneas de investigación (Dirigidas al Área de Tecnología para el Área de Investigación).
- Aprobación de Plan de Gestión de la Calidad (Dirigidas al Área de Gestión de la Calidad).

Descripción

Desde el Observatorio se dirige la Fábrica de Software y es como el tablero de control de ésta, aquí llegan todas las informaciones que circulan por la misma y son aprobados informes y solicitudes. Estas informaciones permiten la toma de decisiones.

El Observatorio analiza el mercado, desde que el que se esta creando el producto, hasta con quien va a competir, es quien decide que hay que ponerle que hay que quitarle. Cuando llega, éste se encarga de ver como llegó, con quien compite. Este se encarga de estudiar, valorar y analizar cuando hay que retirarlo, cuando hay que moverlo, cuando hay que ponerlo, etc. Incluso ayuda a decidir que producto debe llevar y en que momento. Este también observa los indicadores de ganancia.

A continuación se muestra de forma gráfica en la figura 2.3 la posición jerárquica del observatorio en el marco de Fábrica de Software.

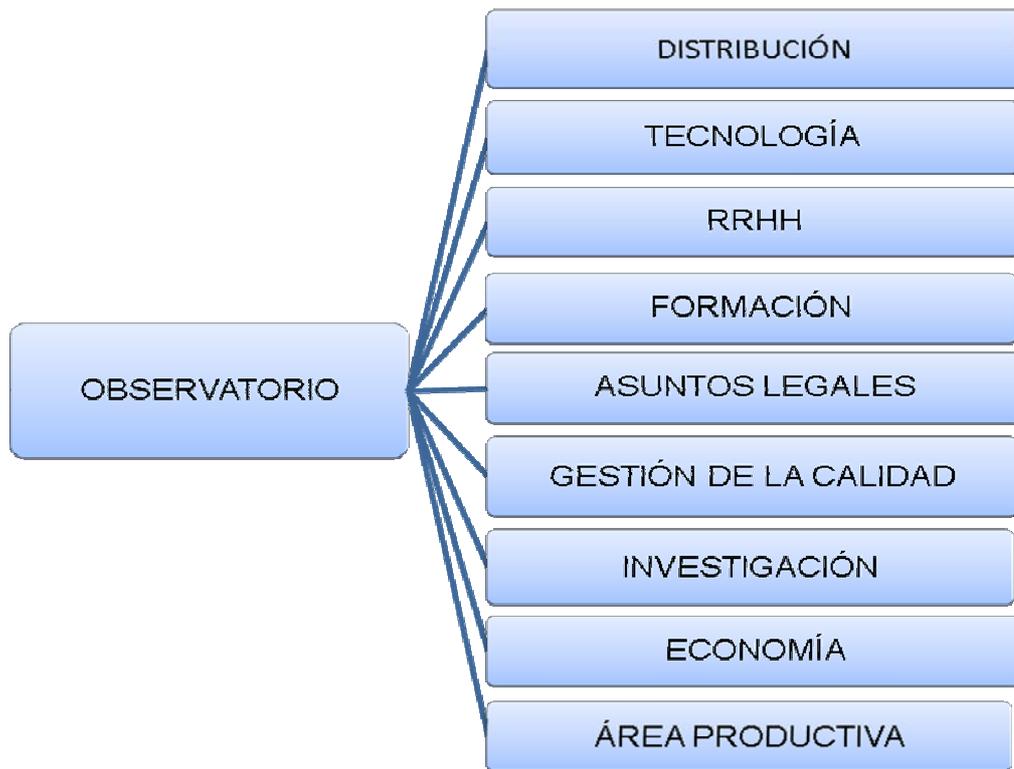


Figura 2.1. Organigrama de la Fábrica de Software.

2.3.2. Área Productiva

Descripción

Se compone por los recursos, los procesos de producción, líneas de producción, la metodología y el producto. Los recursos: están compuestos por las personas y las tecnologías, las primeras se gestionan en el área de recursos humanos y las tecnologías en el área de tecnología. El proceso esta regido por una metodología de desarrollo ágil y tiene varias líneas de producción orientadas a los componentes especializados y ensamblaje del producto final.

Los cinco elementos antes mencionados constituyen el núcleo productivo de toda la fábrica y para que funcione necesita de las áreas de servicio.

Objetivo

- Lograr la producción de productos informáticos de alta calidad, adaptables bajo un proceso de desarrollo eficiente y eficaz respondiendo a las necesidades del mercado.

Funciones

- Establecer planes de producción.
- Diseño de nuevas líneas de producción.
- Producir componentes de software.
- Ensamblaje de productos informáticos.
- Identificar y demandar nuevas tecnologías que favorezcan la productividad.
- Identificar necesidades de capacitación en el personal.

Entradas

- Solicitudes de producción.

Salidas

- Necesidades tecnológicas (Dirigidas al Área de Tecnología).
- Necesidades de investigación (Dirigidas al Área de Investigación).
- Necesidades de personal (Dirigidas al Área de Recursos Humanos).
- Necesidades de capacitación (Dirigidas al Área de Formación).

En la siguiente figura 2.2 se puede ver de forma grafica el marco de Fábrica de Software.



Figura 2.2 Marco Fábrica de Software.

2.3.2.1. Guía de flujo de procesos para el desarrollo de software en el Área Productiva.

La propuesta presentada aquí como guía del flujo de proceso presenta las siguientes características:

- 1- Hay un rol que ejecuta las tareas y otro supervisa. Por tanto hasta que el que supervisa no apruebe la propuesta, esta no va a Revisión Técnica Formal (RTF).
- 2- Se utilizan solamente los artefactos necesarios para documentar el producto.
- 3- Se basa en la reutilización de componentes.
- 4- Se desarrollan partes pequeñas y se ensambla después el producto,
- 5- Los flujos se integran a través de la arquitectura de software y de negocio. Todos los flujos de desarrollo de cada fase se integran siempre detrás de la arquitectura de software y de negocio.
- 6- Se hacen pruebas continuas sobre los compones y/o productos y los cambios se hacen a tiempo. Antes de poner un componente en el repositorio se hacen pruebas unitarias, y cuando se va a utilizar como parte de otro producto se hacen pruebas de

integración, al igual que antes de liberar el producto también. Todo esto demuestra que se está probando en todo el proceso de desarrollo.

- 7- Las áreas de servicio están especializadas, en temas de apoyo a la producción que es el elemento fundamental de la fábrica.

En la **fase de ámbito** se reúne el grupo de especialistas, para analizar la posibilidad de abrir una nueva línea temática. Una vez tomada la decisión y especificada la línea temática se hacen investigaciones sobre el estado del arte, conceptos importantes a tener en cuenta, se hace el estudio de factibilidad económica y/o social y luego se estudia el entorno del negocio.

En la **segunda fase o desarrollo** se definen los requisitos, se realiza el diseño la implementación y las pruebas.

En la **tercera fase** se realiza el ensamblaje modular de los componentes, la estabilización/ las pruebas de integración de los componentes y las pruebas de liberación.

En la **cuarta fase o transición** se cierra el ciclo de desarrollo del producto. Aquí se hacen las pruebas piloto para la certificación y luego el despliegue, finalmente se pasa a darle seguimiento al producto.

En la siguiente figura 2.3 se muestra de forma gráfica la guía del flujo de procesos para el desarrollo de software en el área productiva.

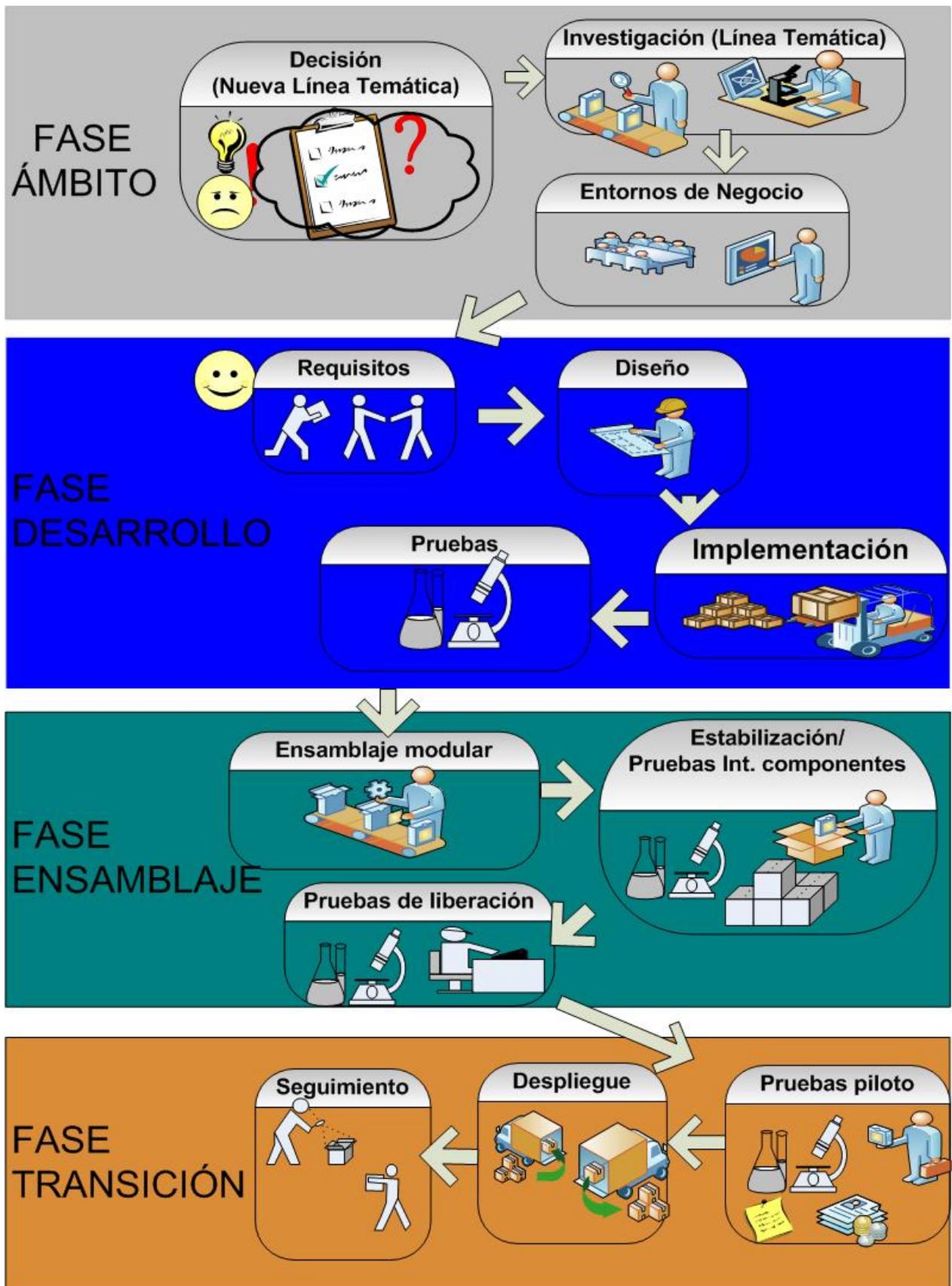


Figura 2.3 Guía del flujo de procesos par el desarrollo de software en el área productiva.

2.3.2.2. Detalles de las fases del ciclo de vida de un producto informático en la Fábrica de Software.

Fase 1. Ámbito

Flujo 1. Estudio sobre la nueva línea temática

Descripción:

En esta fase se reúne un equipo para tomar la decisión de abrir una nueva línea temática.

Artefactos

Documento: Informe sobre nueva línea temática.

Objetivo

Dejar documentados los fundamentos sobre la decisión de la apertura de una nueva línea temática.

Rol que ejecuta

Analista principal (del área productiva).

Especialista económico (del área de economía).

Especialista de investigación (del área de investigación).

Especialista de tecnología (del área de tecnología).

Especialista de formación (del área de formación).

Rol que supervisa

Revisor técnico (del observatorio de la fábrica).

Líder de factoría (del observatorio de la fábrica).

Flujo 2. Investigación

Descripción

Se realiza el estudio del estado del arte, de artículos de investigación, son definidos términos y conceptos a manejar, además de dejar clara sobre la factibilidad de desarrollar el producto.

Artefactos

Documento: Estado del arte.

Objetivo

Informe sobre toda la información existente relacionada al desarrollo del producto a desarrollar.

Documento: Estudio factibilidad.

Objetivo

Realizar un análisis económico y social sobre la factibilidad de desarrollar el producto.

Documento: Glosario de términos.

Objetivo

Dejar documentados todos los términos específicos del área del conocimiento, objeto de estudio.

Rol que ejecuta

Analista principal (del área productiva).

Especialista económico (del área de economía).

Especialista de investigación (del área de investigación).

Rol que supervisa

Revisor técnico (del observatorio de la fábrica).

Líder de factoría (del observatorio de la fábrica).

Flujo 3. Entorno de negocio

Descripción:

En esta fase se estudia el entorno de negocio del cliente que solicita el software.

Artefactos

Documento: Diagrama de procesos.

Objetivo

Modelar (BPMN o IDEF) los procesos de negocio identificados en el entorno cliente como objeto de análisis.

Documento: Especificación de procesos.

Objetivo

Especificación de cada proceso identificado. Cada proceso lleva un documento.

Documento: Visión.

Objetivo

Dejar claras las características básicas del producto, la oportunidad de negocio a partir del alcance, el perfil de los usuarios involucrados, la posición en el mercado así como los plazos de entrega con los precios y costos.

Documento: Glosario de término.

Objetivo

Este documento creado en el primer flujo se actualiza.

Rol que ejecuta

Analista principal (del área productiva).

Especialista económico (del área de economía).
Especialista de investigación (del área de investigación).
Líder de factoría (del observatorio de la fábrica).

Rol que supervisa

Revisor técnico (del observatorio de la fábrica).
Líder de factoría (del observatorio de la fábrica).

Fase 2. Desarrollo

Flujo 1. Requisitos

Descripción:

Se obtienen los requisitos del producto a desarrollar.

Artefactos

Documento: Glosario de término.

Objetivo

Este documento creado en el primer flujo se consulta y actualiza.

Documento: Especificación de requisitos.

Objetivo

En este documento se utiliza para listar y detallar las características que el cliente desea en su producto.

Rol que ejecuta

Analista (del área productiva).
Analista principal (del área productiva).

Rol que supervisa

Revisor técnico (del observatorio de la fábrica).
Líder de factoría (del observatorio de la fábrica).

Flujo 2. Diseño

Descripción:

Se describe como va a ser el sistema propuesto que componentes existen y que papel juegan empleando diagramas y notaciones formales

Artefactos

Documento: Diagrama de diseño de clases.

Objetivo

Se muestra todo el diseño de clases del componente.

Documento: Realización de requisito.

Objetivo

Contiene todos los diagramas de interacción (secuencia y/o colaboración).

Documento: Modelo relacional de datos.

Objetivo

Contiene el modelo de datos (Diagrama Entidad Relación).

Documento: Documento integración.

Objetivo

Especifica como se integran los elementos de diseño identificados como son los nodos de integración y las clases y estructuras comunes.

Rol que ejecuta

Revisor técnico (área productiva).

Desarrollador (área productiva).

Rol que supervisa

Especialista de calidad (área de calidad).

Arquitecto (área productiva).

Analista (área productiva).

Especialista de bases de datos (área productiva).

Especialista de bases de datos (área productiva).

Arquitecto principal (área productiva).

Flujo 3. Implementación

Descripción

Se codifica el sistema diseñado.

Artefactos

Código fuente.

Manual de usuario.

Documento de integración.

Objetivo

Se muestran los elementos de implementación que integran.

Registro de errores de implementación (Métricas).

Objetivo

Registrar los errores de la implementación (Bug).

Modelo de componentes

Objetivo

Mostrar la relación entre los componentes.

Rol que ejecuta

Desarrollador (área productiva).

Documentador (área productiva).
Desarrollador (área productiva).
Revisor técnico (área productiva).
Desarrollador (área productiva).
Revisor técnico (área productiva).

Rol que supervisa

Desarrollador (área productiva).
Analista (área productiva).
Arquitecto principal (área productiva).
Especialista de calidad (área de calidad).

Flujo 3. Pruebas

Descripción

Se realizan pruebas unitarias a cada componente diseñado e implementado.

Artefactos

Documento Escenario(s) de prueba unitaria.
Archivo de Bugs.
Documento Plan de prueba.
Documento de gestión de Bugs.

Rol que ejecuta

Especialista de calidad.

Rol que supervisa

Especialista principal de calidad (área de calidad).
Administrador de la gestión de configuración (área productiva).

Fase 3. Ensamblaje

Flujo 1. Ensamblaje modular

Descripción:

Se ensambla los sistemas y subsistemas.

Artefactos

Documento integración.
Realese de subsistemas a integrar ajustados.
Documento de vista del despliegue integración.
Modelo de datos de subsistemas a integrar ajustados.
Diseño de clases de integración.
Documento resultado de la RTF.

Rol que ejecuta

Arquitecto (área productiva).

Desarrollador (área productiva).
Arquitecto (área productiva).
Especialista de base de datos (área productiva).
Revisor técnico (observatorio).

Rol que supervisa

Arquitecto principal (área productiva).
Arquitecto (área productiva).
Especialista principal de bases de datos (área productiva).
Especialista de calidad (área de calidad).

Flujo 2. Estabilización y pruebas de componentes

Descripción

Se realizan pruebas al sistema ensamblado y se estabilizan los componentes de software integrados en el ensamblaje.

Artefactos

Archivo de Bugs (Especificando los bugs de los elementos arquitectónicos inestables en la integración).
Documento de solicitud de cambio.
Orden de cambio.
Release del subsistema reparado.
Reporte de auditoria de la gestión de cambio.

Rol que ejecuta

Especialista de calidad (área de calidad).
Gestor de configuración de software (área productiva).
Desarrollador (área productiva).
Arquitecto (área productiva).

Rol que supervisa

Administrador de la gestión de configuración (área productiva).
Especialista principal de calidad (área de calidad).
Arquitecto principal (área productiva).

Flujo 3. Pruebas de liberación

Descripción

Se verifica que el producto obtenido satisface los requisitos especificados.

Artefactos

Escenario(s) de prueba de integración.

Plan de prueba.
Archivo de Bugs.
Documento de gestión de Bugs.
Documento de solicitud de cambio.
Orden de cambio.
Release del subsistema reparado.
Reporte de auditoria de la gestión de cambio.

Rol que ejecuta

Especialista de calidad.
Gestor de configuración de software.
Desarrollador.
Arquitecto.

Rol que supervisa

Especialista principal de calidad.
Administrador de la gestión de configuración principal de calidad.
Arquitecto principal.

Fase 4. Transición

Flujo 1. Pruebas piloto (Certificación)

Descripción

Se hacen pruebas de funcionamiento del producto en entornos del cliente.

Artefactos

Escenarios de pruebas pilotos.
Plan de prueba piloto.
Archivo de Bugs.
Documento de gestión de Bugs.
Orden de cambio.
Release del subsistema reparado.
Reporte de auditoria de la gestión de cambio.

Rol que ejecuta

Especialista de calidad.
Gestor de configuración de software.
Desarrollador.
Arquitecto.
Gestor de configuración de software.

Rol que supervisa

Especialista principal de calidad.

Administrador de la gestión de configuración.

Arquitecto principal.

Flujo 2. Despliegue**Descripción**

Se procede a la implantación del sistema de forma generalizada en el entorno cliente, además se capacita a los usuarios del sistema.

Artefactos

Documento: Plan de despliegue.

Objetivo

Mostrar información sobre el inventario de componentes, así como el aseguramiento tecnológico, procedimientos y cronograma de ejecución.

Rol que ejecuta

Arquitecto principal (área productiva).

Rol que supervisa

Líder de la factoría (observatorio).

Flujo 3. Seguimiento**Descripción**

Se monitorea el funcionamiento del producto implantado en el entorno del cliente durante el período de tiempo que se ha negociado, con el propósito de resolver cualquier fallo en el funcionamiento del mismo.

Artefactos

Documento de no conformidades.

Objetivo

Mostrar todos los problemas en el funcionamiento del software.

Rol que ejecuta

Arquitecto (área de calidad).

Rol que supervisa.

Líder de la factoría (observatorio).

2.3.3. Área Servicios de la Fábrica.

EL Area de Servicios, de la Fábrica de Software comprende los elementos siguientes: Área de Formación Docente, Economía, Recursos Humanos (RRHH), Tecnología, Investigación, Gestión de la Calidad, Asuntos legales, y Distribución. Los cuales se interrelacionan en un trabajo sistémico demandando y entregando entradas y salidas de los procesos con el fin de conformar un carácter dialéctico en el proceso de desarrollo. Cada uno de los elementos responde a responsabilidades concretas dentro del proceso de desarrollo. Así como la importancia de sus actividades dentro del desarrollo sistémico productivo de la fábrica.

Área de Formación Docente:

Objetivo

- Coordinar las actividades de superación, actualización e intercambio académico y de formación docente de la Fábrica de Software hasta el nivel de puestos de trabajo, brindando el apoyo académico y administrativo requerido.

Funciones

- Proponer a la Secretaría General el programa anual de actividades de la Fábrica de Software y presentar la evaluación correspondiente.
- Supervisar la planeación, el desarrollo y la evaluación de las actividades de superación, actualización e intercambio académico y formación docente de la Fábrica de Software.
- Fomentar y mantener comunicación e intercambio con instituciones de formación y apoyo docente a fin de obtener información y material de interés para el desarrollo de las actividades del área.
- Presentar a la secretaría general el informe del ejercicio presupuestal asignado a la Fábrica de Software para el desarrollo de las actividades de superación, actualización e intercambio académico.
- Supervisar la difusión oportuna de los programas del área y los ofrecidos por otras instituciones.
- Diseñar, organizar, instrumentar y evaluar los programas de los cursos de formación, capacitación, actualización pedagógica e intercambio en las disciplinas englobadas al interior de la Fábrica de Software.
- Preparar el material requerido para los cursos de formación, capacitación y actualización pedagógica.

- Asesorar a los órganos de la secretaría general en la selección, diseño y puesta en práctica de nuevos métodos de enseñanza.
- Gestionar la autorización de los programas de formación, actualización y perfeccionamiento, en lo general.
- Vigilar que las actividades del departamento se realicen conforme a la legislación universitaria.
- Llevar a cabo las actividades y comisiones que se le deleguen por acuerdo superior.

Entradas

- Necesidades de capacitación (Proceden del Área productiva, Gestión de la Calidad o Recursos Humanos).
- Resultados de la investigación (Proceden del Área de Investigación).
- Necesidades de capacitación por asimilación de nuevas tecnologías para el desarrollo (Proceden del Área de Tecnología).

Salidas

- Necesidades de investigación (Dirigidas al Área de Investigación).
- Plan de capacitación y certificación de roles y conocimientos (Dirigidas al Área Recursos Humanos).
- Plan de carrera pregrado y posgrado a certificar (Dirigidas a Área Calidad).

Descripción

El Área de Formación posee un componente que es pregrado, otro componente que es postgrado.

Pregrado: Se define el programa de carrera con el que se forma a los estudiantes.

Postgrado: Se definen las maestrías, los doctorados, las conferencias, los diplomados.

Gestiona las necesidades de carrera del proceso de desarrollo. Entre su principal característica se propone cinco fases formativas en el proceso de formación de los recursos humanos. La primera relacionada con el ciclo básico del estudiante, donde se impartirán las asignaturas horizontales de las ingenierías y en la que se incluyen un grupo de asignaturas de corte formativo para la producción. En la segunda fase, y previa definición de las necesidades formativas el recurso humano de pregrado recibe las asignaturas en forma rotativa que recrean los diferentes roles que un ingeniero podría rotar en la fábrica, además de las

asignaturas maestras del programa de ingeniería informática, las fases restantes están asociadas al postgrado, maestría y doctorado.

Área de Economía:

Objetivos

- Control y seguimiento de la contabilidad presupuestaria, patrimonial y de costos, así como facilitar los procesos de conciliación y presentación de estados de situación financiera con consolidación a diferentes niveles, dependiendo de las demandas específicas de información.
- Gestiona y norma los procesos contables de los centros de costo de la fábrica.
- Gestiona el plan económico de los diferentes procesos productivos desarrollados en la fábrica.

Funciones

- Seguimiento y análisis de la contabilidad generada en el área productiva y las áreas de de servicio.
- Control de ingresos y pagos, así como la ejecución, seguimiento y control de cuantas tareas exijan la gestión de la tesorería de la Fábrica de Software.
- Elaboración del anteproyecto de Presupuesto anual de la Fábrica de Software conforme a las instrucciones recibidas del Observatorio, así como la ejecución de cuantas modificaciones se autoricen sobre el mismo por parte de los órganos competentes.
- Seguimiento y control de la ejecución del presupuesto de la Fábrica de Software y su proceso de desarrollo.
- Elaboración de informes y estados de carácter periódico y extraordinario referidos al presupuesto, la contabilidad y la tesorería a otras materias de su competencia.
- Elaboración y análisis de la cuenta de liquidación del presupuesto y del informe económico anual.
- Coordinación de las administraciones y unidades administrativas periféricas en aquellas materias de su competencia.
- Contabilización, seguimiento y control de las obligaciones de carácter tributario en los que incurra la Fábrica de Software.

Entradas

- Informes de aprovechamiento de la jornada laboral, pre nóminas de pago, y vacaciones (Proceden del Área de Recurso Humanos).

- Plan de presupuesto para salarios (Proceden del Área de Recursos Humanos).
- Salidas del proceso de gestión de los centros de costo (Proceden de todas las Área con centro de costo definido).

Salidas

- Informes de presupuesto para las áreas que lo soliciten.
- Informes de análisis financiero (Dirigidas al Área Observatorio).

Descripción

Esta se encarga de ilustrar al futuro profesional respecto a la contribución práctica, donde el pensamiento económico puede ofrecer una alta gestión a la fábrica. Permite familiarizarse en el manejo de ciertos modelos de toma de decisión. Como por ejemplo la lógica competitiva de los mercados, los determinantes de las oportunidades y los beneficios, la formulación de estrategias competitivas y la sostenibilidad de ventajas competitivas.

Área de RRHH

Objetivos

- Gestionar Capital humano, para alcanzar los objetivos definidos de la manera más eficaz y eficiente posible.

Funciones

- **Planeación de personal:** Determina las necesidades de personal, determina los objetivos, políticas, procedimientos y programas de administración de personal dentro de la Fábrica de Software. Consiste en realizar estudios tendientes a la proyección de la estructura de la organización en el futuro, incluyendo análisis de puestos proyectados y estudio de las posibilidades de desarrollo de los trabajadores para ocupar estas.
- **Empleo (reclutamiento, selección, contratación e inducción):** Lograr que todos los puestos sean cubiertos por personal idóneo, de acuerdo a una adecuada planeación de recursos humanos.
 - **Reclutamiento:** buscar y atraer solicitantes capaces para cubrir las vacantes que se presente: Técnica encaminada a proveer de recursos humanos a la Fábrica de Software en el momento oportuno.

- **Selección:** Analizar las habilidades y capacidades de los solicitantes a fin de decidir, sobre bases objetivas, cuál tienen mayor potencial para el desempeño de un puesto y posibilidades de un desarrollo futuro, tanto personal como de la **Fábrica de Software**. Proceso que trata no solamente de aceptar o rechazar candidatos, sino conocer sus aptitudes y cualidades con objeto de colocarlo en el puesto más fin a sus características.

- **Contratación:** Formalizar con apego a la ley la futura relación de trabajo para garantizar los intereses, derechos y deberes tanto del trabajador como de la **Fábrica de Software**.

- **Inducción:** Dar toda la información necesaria al nuevo trabajador y realizar todas las actividades pertinentes para lograr su rápida incorporación a los grupos sociales que existan en su medio de trabajo, a fin de lograr una identificación entre el nuevo miembro y la organización viceversa. Consiste en llevar al individuo al puesto que va a ocupar, presentarlo con su superior y compañeros con el objeto de lograr una adaptación de grupo que evite una baja en el rendimiento, que obtenga una visión de la empresa. Así mismo se le mostraran las instalaciones donde de la empresa y principalmente de su área de trabajo. Empleo (reclutamiento, selección, contratación e inducción)

- **Capacitación y desarrollo:** Tiene por objeto ampliar, desarrollar y perfeccionar al hombre para su crecimiento profesional en determinado puesto en la empresa o para estimular su eficiencia y productividad. Debe basarse en el análisis de necesidades que parta de una comparación del desempeño y la conducta actual con la conducta y desempeño que se desean. Con base a este análisis, se identifican los métodos y necesidades de capacitación que son enviados al área de formación.

- **Administración de sueldos y salarios:** Parte de la administración, de personal que estudia los principios o técnicas para lograr que la remuneración global que recibe el trabajador sea adecuado a la importancia de su puesto, a su eficiencia personal, a sus necesidades ya las posibilidades de la empresa. Consiste en asignar valores monetarios a los puestos, en tal forma que sean justos y equitativos en relación a otras posiciones de la organización y a puestos similares en el mercado de trabajo.

- **Prestaciones y servicio de personal.** Son todas aquellas actividades que realiza la empresa enfocadas a proporcionar al trabajador un beneficio, ya sea en dinero o en especie. Satisfacer las necesidades de los trabajadores que laboran en la organización y tratar de ayudarles en los problemas relacionados a su seguridad y bienestar personal.

- **Seguridad e Higiene en el trabajo:** Es el conjunto de conocimientos y técnicas dedicadas a reconocer, evaluar y controlar aquellos factores del ambiente, psicológicos o tensionales, que provienen del trabajo y pueden causar enfermedades, accidentes o deteriorar la salud. Desarrollar y mantener instalaciones y procedimientos para prevenir accidentes de trabajo y enfermedades profesionales.

- **Relaciones laborales:** Parte de la administración de Recursos Humanos que se ocupa de negociar con el sindicato los términos del contrato o convenio de trabajo, interpretar la ley laboral en lo que se refiere a las políticas y prácticas de la organización, así como el arreglo arbitrario de cualquier agravio que surja de tales contratos.

Entradas

- Solicitud de personal (Proceden del Área Productiva).
- Informes de análisis legal (Proceden del Área de Asuntos Legales).
- Informes de la evaluación de la calidad del trabajo realizado por el personal (Proceden del área de Gestión de la Calidad).
- Plan de capacitación para el personal reclutado (Proceden del Área de Formación).
- Solicitud de información de la plantilla de la Fábrica de Software (Proceden del Área de Distribución).
- Entradas análisis estratégico referente a RRHH del Área Observatorio de la Fábrica de Software.
- Plan de capacitación, certificación de roles y conocimientos (Proceden del Área formación).

Salidas

- Plan de presupuesto para salarios (Dirigida al Área de Economía).
- Informes del aprovechamiento de la jornada laboral y organización del trabajo (Dirigida al Departamento de Economía).
- Informes de la actividad (Dirigida al Área de Asuntos Legales).

- Necesidades de capacitación del personal reclutado (Dirigida al Área de Formación).
- Información sobre la plantilla de la Fábrica de Software (Dirigida al Área de Distribución).
- Informe de análisis del capital humano (Dirigida al Área Observatorio).

Descripción

Esta área está regida por las leyes de la administración y en ella se gestionan los recursos humanos, ofreciendo información y control estratégicamente desde la planeación hasta la operación de dichos recursos.

Área de Tecnología

Objetivos

- Gestionan la política de desarrollo tecnológica de la Fábrica de Software. De modo que garantice un estado competitivo y rentable de la ciencia y tecnología de la institución.
- Rige la definición de nuevas líneas de investigación referentes a tecnología.

Funciones

- Definir y mantener en buen estado la tecnología.
- Administrar las redes y seguridad informática dentro de la fábrica.
- Administra los servicios tecnológicos.
- Facilitar los medios tecnológicos.
- Controlar el buen uso y desarrollo de la tecnología.
- Definir nuevas líneas de investigación tecnológicas.
- Elaborar el plan con las políticas de gestión tecnológica.

Entradas

- Solicitud de nuevas tecnologías (Proceden de las necesidades de la Fábrica de Software y su plan estratégico).
- Informe de necesidades de mantenimiento o remodelación de la tecnología de producción (Procede del Área de Producción).

Salidas

- Solicitud de nuevas líneas de investigación (Dirigidas al Área de Investigación).

- Necesidades de capacitación por asimilación de nuevas tecnologías para el desarrollo (Dirigidas al Área de Formación).
- Propuesta de tecnologías (Dirigidas al Área Productiva).

Descripción

La actividad tecnológica influye en el progreso social y económico. Esta es un área de servicio, la cual le brinda el soporte para su buen desarrollo a las demás áreas. Desde el punto de vista de los productores de bienes y de los prestadores de servicios, las tecnologías son el medio indispensable para obtener un buen resultado. Desde el punto de vista de los consumidores, las tecnologías les permiten obtener mejores bienes y servicios.

Área de Investigación

Objetivos

- Lograr resultados científico-técnicos de impacto en el proceso de desarrollo y en la formación de personal.

Funciones

- Promoción del desarrollo de líneas de investigación enfocado a hitos tecnológicos e hitos de negocio.
- Apoyo en la realización de tesis doctorales en el área de conocimiento.
- Asesorar y apoyar a trabajos de investigación que puedan generar nuevas líneas de investigación.
- Promoción de encuentros de expertos en el tema de investigación de interés.

Entradas

- Solicitud de nuevas líneas de investigación (Proceden de la gerencia de la Fábrica de Software).
- Solicitud de nuevas investigación (Proceden del Área de Tecnología, Formación y Proceso productivo).

Salidas

- Resultados de investigación (Dirigidas al Área de Tecnología, Formación y proceso producción).

Descripción

La investigación estaría centrada en generar nuevos conocimientos, en crear una nueva prospectiva tecnológica y desarrollar innovación tecnológica.

La importancia que tiene es que se elevaría la competitividad y la innovación de la Fábrica de Software incrementando la inversión, promoviendo la gestión tecnológica influyendo en el mejoramiento del proceso de desarrollo de software, contribuye a la preparación de personal a un alto nivel científico; generando promover y apoyar el buen uso de las tecnologías de la información.

Área de Gestión de la calidad

Objetivos

- Dar seguimiento y evaluar el sistema de calidad implantado en la Fábrica de Software para garantizar la expedición de resultados confiables, veraces y oportunos.
- Promover el desarrollo de los programas de calidad.
- Definir las principales líneas de investigación en temáticas relacionadas a la calidad.

Funciones

- Promover el desarrollo de procesos de formación, diagnóstico y certificación en el área de mejoramiento continuo de la calidad, para elevar la calidad de la producción de software contribuyendo al aumento de la productividad y la calidad en los productos.
- Contribuir a la identificación, generación, promoción y adopción de estándares, normas y mejores prácticas relacionadas con la calidad en la Ingeniería de Software.
- Promover la investigación y la búsqueda de soluciones de los principales problemas en el Área de Ingeniería y la Calidad de Software.
- Desarrollar la oferta de servicios de diagnóstico de calidad.
- Desarrollar la oferta de servicios de formación en áreas de calidad.
- Desarrollar acuerdos con todo tipo de instituciones nacionales e internacionales con el objeto de realzar las actividades, proyectos y programas inherentes a la gestión de la calidad.

Entradas

- Necesidades de gestión de la calidad (Proceden del Área Productiva).
- Resultados de investigación (Proceden del Área de Investigación).

Salidas

- Solicitud de nuevas líneas de investigación (Dirigidas al Área de Investigación).
- Necesidades de capacitación (Dirigidas al Área de Formación).
- Plan de gestión de la calidad (Dirigidas al Área de Producción y entorno de la Fábrica de Software).
- Informes de la evaluación de la calidad del trabajo realizado por el personal (Dirigidas al Área de Recursos Humanos).

Descripción

Se controla la adecuación del producto al uso correspondiente. También debe responder a la conformidad con requisitos y confiabilidad en el funcionamiento para así obtener un producto dentro de la industria con cero defectos. Evitándose la pérdida económica que un producto supone para la sociedad.

Esta área se caracteriza por estar interrelacionada con las demás, ya que se encarga de controlar la eficacia y eficiencia dentro del proceso de la fábrica. Como por ejemplo vinculándose con la Formación, los RRHH, la Metodología, la Gestión y la Infraestructura para así optimizar el producto final.

La gestión debe estar basada en la las vistas de la calidad, debe ser trascendente, es decir que sea de de calidad de excelencia, estar basada en el producto, controlando su economía, basada en el usuario y al cliente adecuándose a la necesidad del propósito, para así dar conformidad y que tenga siempre un valor importante dentro del equipo de desarrollo.

Área de Asuntos legales

Objetivos

- Control de los Asuntos Legales de la Fábrica de Software.

Funciones

- Realizar estudios sobre licencias de software.
- Registro y control de las patentes.
- Diseño de contratos.
- Define derechos legales sobre los productos.

Entradas

- Informe de la actividad de Recursos Humanos (Proceden del Área de Recursos Humanos).
- Informe de la actividad del área de formación (Proceden del Área de Formación).
- Informe de la actividad del Área de Gestión de la Calidad (Proceden del Área de Gestión de la Calidad).
- Informe de la actividad del Área de Investigación (Proceden del Área de Investigación).
- Informe de la actividad del Área de economía (Proceden del Área de Economía).
- Solicitud de estudios sobre licencias, patentes, y derecho de autor (Proceden del Área de Tecnología).
- Solicitud de diseño de contratos (Proceden del Área de Distribución).

Salidas

- Informe de análisis legal sobre actividad de Recursos humanos (Dirigidas al para Recursos Humanos).
- Informe de análisis legal de la actividad del Área de Gestión de la Calidad (Dirigidas al Área de Gestión de la Calidad).
- Informe de análisis legal de la actividad del Área de Formación (Dirigidas al Área de Formación).
- Informe de análisis legal de la actividad del Área Investigación (Dirigidas al Área de Investigación).
- Informe de análisis legal de la actividad del Área de Economía (Dirigidas al Área de Economía).
- Informes sobre licencias, patentes, y derecho de autor (Dirigidas al Área de Tecnología).
- Informe sobre diseño de contratos (Dirigidas al Área de Distribución).

Descripción

Al producirse software se consumen licencias y recursos en función a lo que se desea desarrollar. Este trata el tema sobre el consumo de las licencias, del software propietario y las patentes. Este rige que es lo libre y que lo propietario viendo así la interrelación de los proveedores y las licencias.

Este es quien tiene que controlar todo el tema sobre patentes, asuntos jurídicos, etc. Le proporciona un apoyo al subsistema de economía como por ejemplo cuando existe algún problema con la facturación o alguna patente este lo apoya y lo defiende.

Su objeto social controlar los asuntos legales en la UCI de la Fábrica de Software.

Área de Distribución

Objetivo

- Administrar compras, ventas, comercialización, almacenes relacionados a la Fábrica de Software.
- Dar seguimiento al producto que se distribuye.

Funciones

- Encontrar todo lo necesario para administrar los bienes y consumibles, adquiridos e ingresados ya en los diferentes almacenes, permitiendo distribuir dichos bienes en las diferentes áreas basados en los requerimientos de operación.
- Proporcionar la información necesaria sobre toda la gestión.
- Gestionar la distribución de los productos a comercializar o en prestación de colaboración.
- Gestionar la evolución del producto distribuido.

Entradas

- Solicitud de compras y ventas de cualquier área.
- Informe de análisis legal actividades (Procede del Área de Asuntos Legales).

Salidas

- Solicitud de autorización de compras y ventas al Área de Economía.
- Informe de actividades (Dirigido al Área de Asuntos Legales).
- Informe de operaciones al Observatorio.
- Solicitud de información de la plantilla de la Fábrica de Software (Dirigido al Área de Recursos Humanos).

Descripción

Relación con todas las áreas operativas de un proceso de compra y que este maximice los recursos de información, está desarrollado específicamente para la administración de compras del sector ya que maneja los requisitos e interacciones de las dependencias áreas internas y externas como recursos financieros, unidades administrativas, almacenes, etc. Y procesos propios desde los requerimientos de compra hasta el la entrega del bien o servicio, pasando por la consolidación de pedidos, generación de bases, proceso de licitación la necesidades de descentralización y centralización de casi toda la administración pública.

Debe contener un poderoso administrador de almacenes físicos y lógicos para un mejor control de entradas, salidas, devoluciones etc. Dando como resultado una existencia exacta de cantidad y ubicación.

2.3.4. Interrelación sistémica entre las áreas.

La Fábrica de Software es un conjunto de áreas relacionadas entre si coordinadas por su área principal (Observatorio) y teniendo como núcleo central el área productiva, esta se dedica a la producción de determinados productos según la línea temática. Entre estas áreas existe un intercambio de información y cada una de ellas tiene sus funciones específicas por lo que si falla una el proceso de desarrollo se compromete considerablemente la producción.

Es el área productiva la que demanda personal bien formado y capacitado, nuevas tecnologías, planes de gestión de la calidad, un buen control económico, administración de recursos (Distribución), control legal y buen control de los recursos humanos. De la coordinación entre las áreas de servicio y el área de producción dependen los altos resultados productivos. En la siguiente figura puede apreciarse de forma gráfica la interrelación sistémica.



Figura 2.4 Interrelación de las áreas.

2.4. Flujos verticales y horizontales.

Los flujos verticales contienen artefactos por fases y los horizontales están durante todas las fases. Por ejemplo se integra todo el tiempo con la arquitectura, desde el inicio hasta el final. La calidad se evoluciona desde el comienzo hasta que se termina. Desempeño la evaluación de los RRHH desde que se inicia y hasta que termina y el proceso se gestiona desde que inicia hasta que se termina. Permitiendo capacitar al personal en función de la productividad, de inicio a fin. Por lo tanto si se evalúa se puede capacitar.

A continuación una representación gráfica de estos flujos (Figura 2.5).

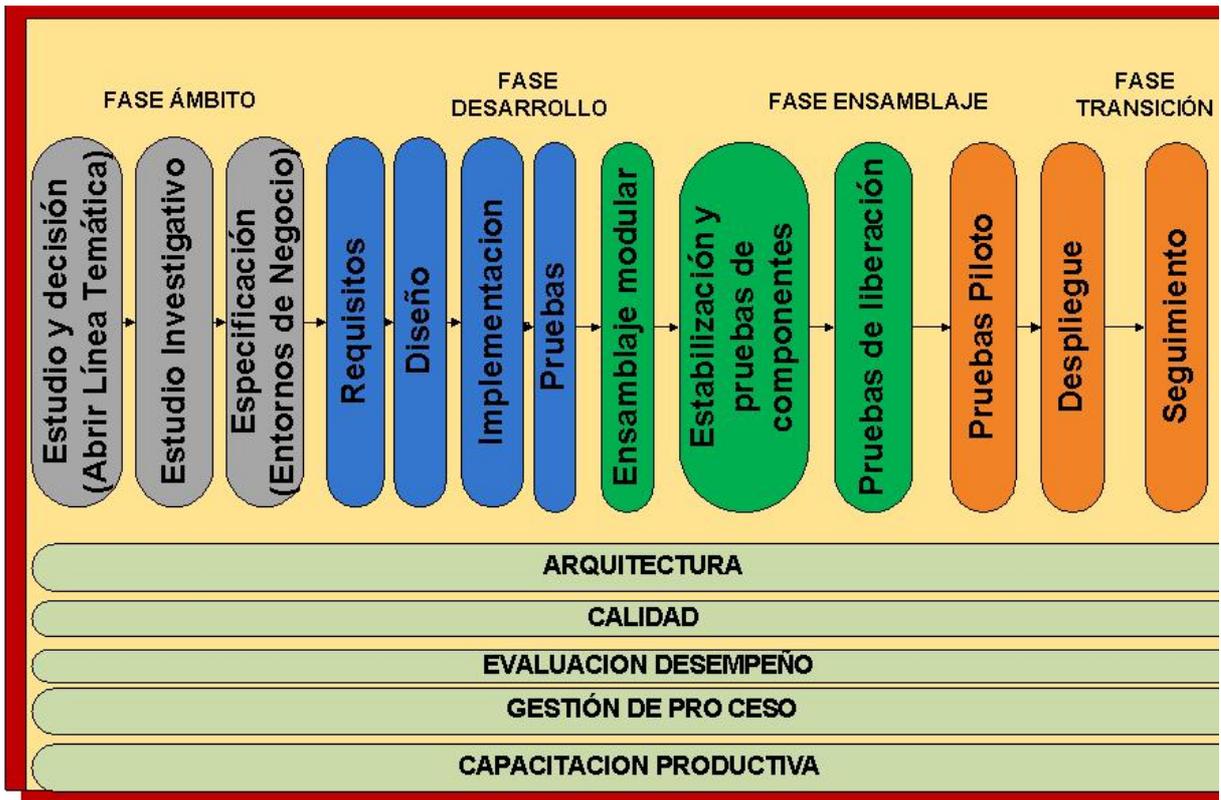


Figura 2.5. Flujo horizontal y vertical.

2.5. Despliegue de la Fábrica de Software.

Para comprender como se ubica la Fábrica de Software dentro de la UCI, la primera idea es que la UCI contiene los Polos y cada uno puede tener diferentes tipos de productos como por ejemplo: Gestión Empresarial, Gubernamental y Financiera. Para cada tipo de producto existe una fábrica especializada.

La fábrica es un área donde se forma personal, se produce e investiga sobre un tipo de producto para un entorno de negocio determinado y es a lo que se le denomina línea temática. Dentro de la fábrica pueden existir varias líneas de producción para componentes y otras para ensamblaje. En la siguiente figura 2.7 se puede tener una idea gráfica de la ubicación jerárquica de la fábrica dentro de la UCI.

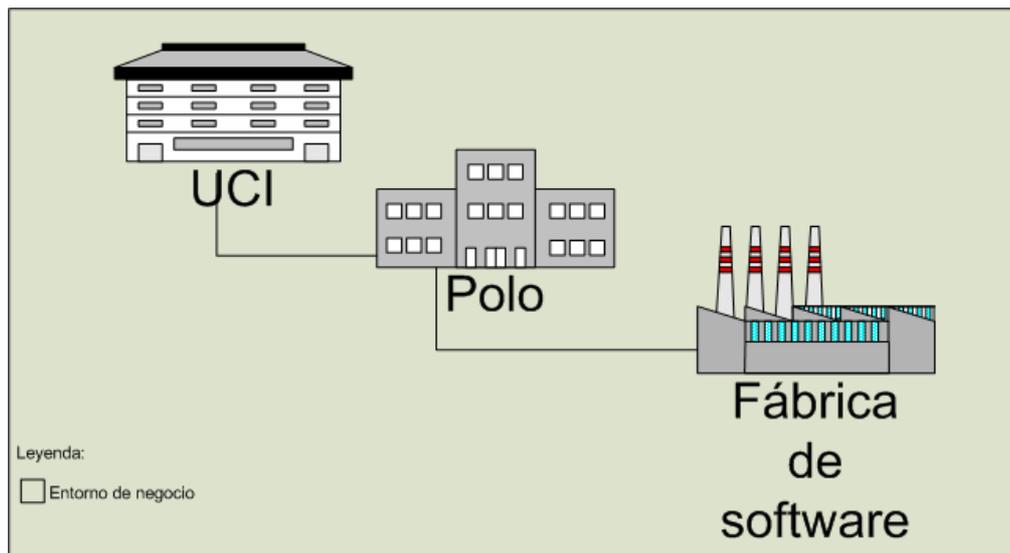


Figura 2.6. Ubicación jerárquica de la FS dentro de la UCI.

El Polo de Gestión de Recursos de la Facultad 3, tratando de ejemplificar el modelo tiene como objeto de producción hacer software relacionado con la gestión de recursos, por ejemplo los ERP o la contabilidad del material de los ERP, los software de comercio electrónico y otros software o contratos.

Actualmente la UCI esta dividida en trece (13) facultades donde existen polos dentro de los cuales están ubicados los proyectos productivos.

La facultad tiene una estructura por departamentos y dicha estructura ha sido fuente de diversos problemas y críticas mundialmente.

Según Veritas(VERITAS 2005) estas criticas se deben a:

- El establecimiento de objetivos locales o individuales en ocasiones incoherentes y contradictorios con lo que deberían ser los objetivos globales de la organización.
- La proliferación de actividades departamentales que no aportan valor al cliente ni a la propia organización, generando una injustificada burocratización de la gestión.
- Fallos en el intercambio de información y materiales entre los diferentes departamentos (especificaciones no definidas, actividades no estandarizadas, actividades duplicadas, indefinición de responsabilidades)
- Falta de implicación y motivación de las personas, por la separación entre “los que piensan” y “los que trabajan” y por un estilo de dirección autoritario en lugar de participativo.

Actualmente las organizaciones tienden a la Gestión por procesos que supera a la gestión tradicional. Dicha Gestión por procesos se basa en la modelación de los sistemas como un conjunto de procesos interrelacionados mediante vínculos causa-efecto. El propósito final de la Gestión por Procesos es asegurar que todos los procesos de una organización se desarrollan de forma coordinada, mejorando la efectividad y la satisfacción de todas las partes interesadas (clientes, accionistas, personal, proveedores, sociedad en general).

El marco de Fábrica de Software propuesto se basa en una Gestión por procesos, y la docencia se subordina a la producción. En la fábrica trabajan estudiantes que se forman mientras producen dirigidos por ingenieros y especialistas de la producción profesionales.

Dentro de los elementos que pudieran dificultar el despliegue de la Fábrica de Software se encuentra la estructura por departamentos de la facultad unida a la organización de la producción por proyectos actual.

2.5.1. Fases para la implantación de la Fábrica de Software

Fase 1. Se define el alcance de lo que se va a hacer, el alcance de la fábrica a que entorno de negocio va a estar enfocada.

Fase 2. Se define el marco de dicha fábrica, como cuales componentes de lo que como modelo se va a proponer y se concluye cada uno de los componentes.

Fase 3. Se pasa a montar la plataforma tecnológica que da soporte a todos esos componentes y automáticamente se comienza a montar las líneas de producción, donde hay una tributación en paralelo con las tecnologías. Pues las líneas de producción están muy emparejadas al montaje y la plataforma tecnológica. Esta tributación continua de las líneas de producción a las tecnologías es muy importante ya que no puedo evolucionar en uno si no tengo desarrollado el otro.

Fase 4. Una vez montada la plataforma tecnológica de los componentes y montar las líneas de producción puedo empezar un proceso de producción de productos genéricos que son los componentes especializados, donde la tecnología una vez mas vuelve a dar soporte. Y una vez obtenidos todos los productos genéricos, ya listos voy a la comercialización y seguimiento a los productos.

Fase 5. Una vez obtenidos los productos se procede a ensamblar y a comercializar el producto.

Una fábrica salta de un proceso de definición, a un proceso de desmontaje, luego pasa a un proceso productivo y posteriormente a un proceso de comercialización y despliegue.

El orden de las fases se puede ver en la siguiente figura 2.7:

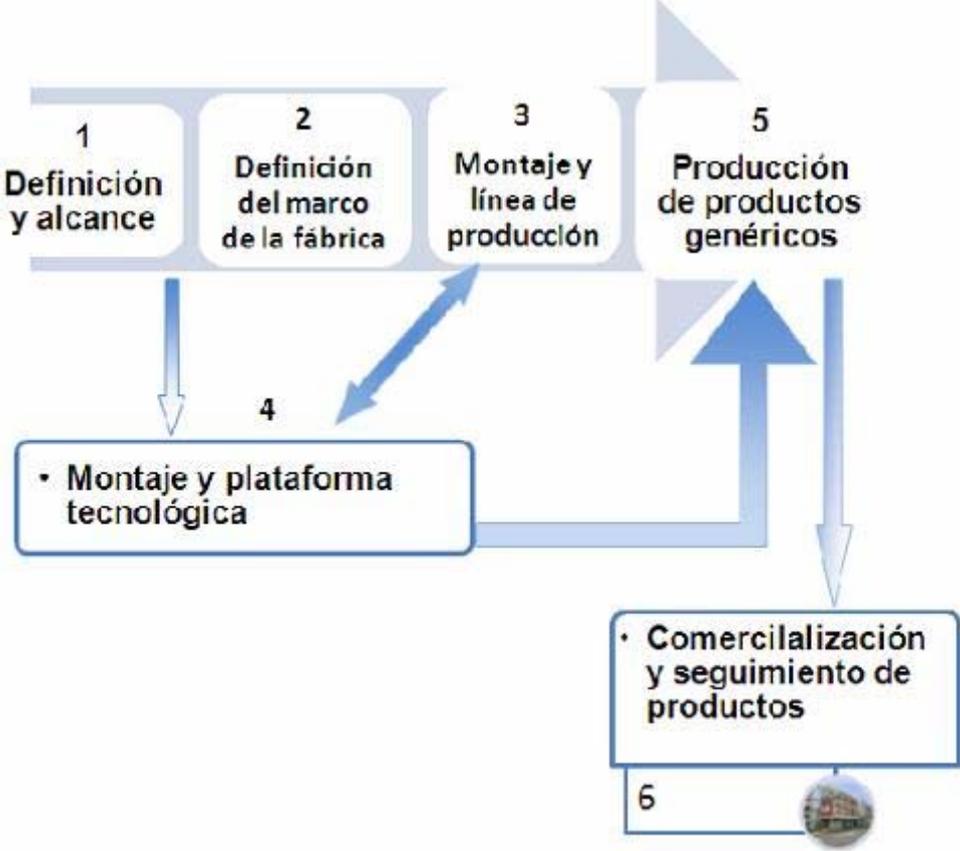


Figura 2.7. Orden de las fases para implantación de la fábrica de software.

2.6. Valoración de la propuesta del marco de Fábrica de Software.

2.6.1. Comparación de los modelos de producción industrial v/s marco de Fábrica de Software

La siguiente tabla 1 presenta la comparación de algunas características de los modelos de producción industrial(MON 2008).

	Artesanado	Taylorismo	Fordismo	Toyotismo
Tipo de Organización del Trabajo	Por oficio	Organización científica del trabajo	Cinta transportadora o cadena de montaje	Módulos flexibles y transfuncionales
Métodos de Control de Producción	Sin control sistemático del tiempo de producción	Asignación de tiempos de producción <i>Tiempos asignados</i>	Imposición de tiempos de producción <i>Tiempos impuestos</i>	Revisión constante de tiempos y tareas <i>Tiempos Compartidos</i>
División Técnica del Trabajo	Trabajo de desarrollo completo de producción	Fraccionamiento de tareas. Repetición de gestos parcelarios	Parcelación Máximo Fraccionamiento de tareas. Especialización	Polifunción Desespecialización con Pluriespecialización
Conocimientos	Dominan los conocimientos técnicos individuales	Reducción de conocimientos técnicos individuales	Especialización de conocimientos fraccionados. Separación entre concepción y ejecución del trabajo	Ampliación de conocimientos, Pluriespecialización
Planificación de la Producción	Sin Planificación	Por tareas	Por Puestos	Por módulos polifuncionales
Conceptos de Calidad	Basado en el conocimiento y capacidad individual	Basado en la rapidez de producción	Basado en la rapidez y cantidad de producción	Detención del proceso productivo en cada módulo ante fallos

Tabla 1. Características de los Modelos de Organización.

El marco de Fábrica de Software basado en las buenas prácticas de las metodologías de desarrollo de software ágil y teniendo en cuenta los aspectos descritos por Alicia Mon(MON 2008) presenta las siguientes características.

	Proceso de desarrollo de software	Marco de Fábrica de software propuesto
Tipo de Organización del Trabajo	Organización científica del trabajo	Existe una metodología que describe las principales fases por las que pasa un producto informático, con los principales flujos de trabajo, los roles y artefactos
Métodos de Control de Producción	Asignación de tiempos de producción <i>Tiempos Asignados</i>	En dependencia de los plazos de entrega del producto en la primera fase de la metodología, se planifica el tiempo para cada fase del desarrollo del producto
División Técnica del Trabajo	Parcelación Máximo Fraccionamiento de tareas Especialización	Existen áreas dedicadas a tareas específicas y a su vez se concibe la especialización a través de los roles
Conocimientos	Especialización de conocimientos fraccionados Separación entre concepción y ejecución del trabajo	Aquí los que investigan y ejecutan tareas no están en la misma área de la fábrica
Planificación de la Producción	Por tareas	Cada rol tiene sus tareas específicas en un puesto de trabajo específico.
Conceptos de Calidad	Los conceptos de Calidad que se utilizan en el Proceso Software, si bien incorporan elementos del <i>Toyotismo</i> en la concepción de la Calidad del Proceso, combina elementos de las diversas concepciones organizacionales, aunque pareciera no responder tan claramente a ninguno de los modelos descriptos.	Está previsto el control de la calidad que es realizado por el área de Gestión de la Calidad que cuenta con un plan de calidad, el producto no sale hasta que tenga cero defectos.

Tabla 2. Características tomadas de los modelos de producción industrial presentes en el desarrollo de software que presenta el marco de desarrollo de Fábrica de Software.

Teniendo en cuenta la comparación de la tabla anterior la Fábrica de software presenta características propias de los modelos industriales de producción y que le son inherentes a cualquier proceso de desarrollo de software bien organizado.

2.6.2. Arquitectura empresarial v/s modelo de Fábrica de Software.

Los elementos de la arquitectura empresarial según Lloréns(FÁBREGAS 2008) son:

➤Arquitectura del negocio.

- La arquitectura del negocio incluye los procesos y la forma en que estos han sido implementados o serán implementados.

➤Arquitectura de sistemas.

- Es la organización fundamental de un sistema, que incluye sus componentes, las relaciones entre ellos y el ambiente, y los principios que gobiernan su diseño y evolución.

➤Arquitectura de información.

- El mapa de todas las necesidades de información derivadas de la Arquitectura de Sistemas.

➤Arquitectura de computación.

- Es la infraestructura tecnológica que se deriva de la arquitectura de información.

El marco de Fábrica de Software propuesto presenta de los cuatro elementos de la arquitectura empresarial los tres primeros. En el caso de la arquitectura del negocio la propuesta de fases del ciclo de vida de un producto puede ser objeto de análisis para el diseño de flujos de procesos. En el caso de arquitectura de sistemas se puede ver a través de la interrelación entre los elementos del marco de Fábrica de Software propuesto. En el caso de la arquitectura de la información, aunque no existe un mapa de necesidades son presentadas las necesidades de información de las áreas.

2.6.3. Principios de las metodologías que se tienen en cuenta en el marco de Fábrica de software propuesto.

Las metodologías ágiles tienen características que hacen el desarrollo de software mucho más rápido y eficiente, aspecto descrito por Pasco (PASCO *et al.* 2008) pero es importante tomar aspectos de las metodologías tradicionales, en la siguiente tabla 2. Se muestran las características que se proponen tener en cuenta de las metodologías de desarrollo de software ágiles y no ágiles.

Aspectos	Metodología Ágil	Metodología No Ágil (Tradicional)	Características seleccionadas
Artefactos	<i>Pocos artefactos</i>	Más artefactos	Metodología Ágil
Roles	<i>Pocos roles</i>	Más roles	Metodología Ágil
Contratos	No existe un contrato tradicional o al menos es bastante flexible	<i>Existe un contrato prefijado</i>	Metodología No Ágil
Interacción del cliente	<i>El cliente es parte del equipo de desarrollo (además in-situ)</i>	El cliente interactúa con el equipo de desarrollo mediante reuniones	Metodología Ágil
Tamaño del grupo de trabajo	<i>Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio</i>	Grupos grandes	Metodología Ágil
Énfasis en la arquitectura	Menos énfasis en la arquitectura y se va definiendo y mejorando a lo largo del proceso	<i>La arquitectura es esencial y se define tempranamente</i>	Metodología No Ágil
Normas a seguir	Basadas en heurísticas provenientes de prácticas de producción de código.	<i>Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.</i>	Metodología No Ágil
Cambios	<i>Se esperan cambios durante el proyecto.</i>	Se espera que no ocurran cambios de gran impacto durante el proyecto.	Metodología Ágil

Tabla 3. Características seleccionadas de las metodologías de desarrollo de software a tener en cuenta en el marco de Fábrica de Software propuesto.

2.7. Conclusiones del Capítulo 2.

Se concluye que los objetivos propuestos para el presente capítulo fueron cumplidos satisfactoriamente. La novedad está en la nueva estructura organizacional, mediante el marco de desarrollo propuesto, las áreas por las cuales este está compuesto y su interrelación. Se logró proponer la guía del flujo de procesos basada en las características positivas de los métodos ágiles y tradicionales. Se valoró y se comparó el modelo propuesto con los diferentes modelos de producción industrial existentes en el mundo, demostrándose que perfectamente se puede producir software como una industria, pero teniendo en cuenta sus particularidades.

CONCLUSIONES GENERALES

Con la presente investigación se hizo un profundo análisis de los modelos industriales de la producción del software y de los diferentes enfoques que existen para su correcto desarrollo. Siendo éste la base para llegar a la propuesta del marco de desarrollo de software. Se logró la identificación de cada uno de los componentes que lo integran y las relaciones entre dichos componentes. La propuesta presenta características de las metodologías ágiles, pues con esta manera de organizar se pretende lograr altos índices de productividad, eficiencia y eficacia en el proceso de desarrollo de software así como una gran adaptabilidad de los productos.

Para dar certeza y fiabilidad a la investigación se hizo una valoración basada en la arquitectura empresarial y las características de las metodologías de desarrollo de software donde queda evidente que ninguna de las existente por si solas pueden aplicarse y sin embargo se pueden utilizar características o principios de todas.

RECOMENDACIONES

Es importante señalar que se le debe dar seguimiento a esta investigación la cual es de gran necesidad, para el buen funcionamiento del proceso de desarrollo de software, para esto se debe implantar y probar el marco propuesto con toda su novel estructura organizacional y su flujo de trabajo basado en los procesos de desarrollo de las fábricas que existen hoy día en el mundo.

Por el corto período de tiempo en el que se desarrollo la investigación se trabajo solamente en las fases por las cuales pasa un producto en la fábrica de software, pero existen otros flujos que perfectamente pueden se objeto de análisis y junto a este son la base de la organización del trabajo.

Detallar y especificar las funciones y la información de cada área de la fábrica requiere de un estudio profundo que no fue objetivo de este trabajo pero se recomienda realizar con el propósito de hacer un buen diseño de arquitectura de información, el cual es esencial para diseñar la arquitectura de computación.

La propuesta realizada debe ser objeto de análisis y valoración por parte de todas aquellas entidades o grupos de personas que deseen adaptarla para producir software.

REFERENCIAS BIBLIOGRÁFICAS

- AAEN, P. B. I. and L. MATHIASSEN *The Software Factory: Contributions and Illusions*, in Proceedings of the Twentieth Information Systems Research Seminar in Scandinavia, Oslo, 1997.
- ABRAHAMSSON, P. S. *Agile Software Development Methods. Review and Analysis*, 2002.
- ALBA, P. R. *Cuba se hará un sitio en el mercado del software*, 2004. [Disponible en: <http://www.americaeconomica.com/numeros3/182/reportajes/rafa182.htm>]
- AZERTIA. *Información general de AZERTIA*, 2008. [2008]. Disponible en: <http://www.ibv.es/pages/empresas/azertia.html>
- BALZER, R. *A 15 Year Perspective on Automatic Programming. IEEE Transactions on Software Engineering* 1985. 11: 1257-1268.
- BECK, K. and M. BEEDLE. *Manifiesto for Agile Software Development*, 2001. [2008]. Disponible en: <http://agilemanifesto.org/>
- CANÓS, J. H. and P. LETELIER. *Métodologías Ágiles en el Desarrollo de Software*, 2008. [2008]. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- CANTONE, G. *Software Factory: Modeling the Improvement. 'Competitive Performance Through Advanced Technology'*. Third International Conference onon (Conf. Publ. No. 359), 1992. p.
- CARREIRA, M. R.; I. R. ROMAN, *et al. Modelado y Simulación del Proceso de Desarrollo de Software: una Técnica para la Mejora de Procesos. Técnicas Cuantitativas para la Gestión en la Ingeniería del Software*, 2008. [2008]. Disponible en: http://investigacion.us.es/sisius/sis_showpub.php?idpers=3274
- CUSUMANO, M. A. *Software Factory: A Historical Interpretation*, 1989. *Diccionario de la Real Academia Española*. 2008. [2008]. Disponible en: <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>
- ESCRIBANO, G. F. *Introducción a Extreme Programming*, 2002. [2008]. Disponible en: <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>
- FÁBREGAS, J. L. *Arquitectura Empresarial* 2008. [Disponible en: <http://www.da.com.ve/publicaciones/arquitecturae.htm>]
- FABRI, J. A.; A. L. P. TRINDADE, *et al. Un Estudio Comparativo entre as Fábricas de Softwares Brasileiras e Japonesas*, 2008. [2008]. Disponible en: <http://engenhariosoftware.files.wordpress.com/2008/04/brasiljapao.pdf>

- GATTACA. *Presentacion de Metodoligoa MSF (Microsoft Solution Framework)*, 2008. [2008].
Disponible en: <http://www.e-gattaca.com/eContent/library/documents/DocNewsNo50DocumentNo6.PDF>
- HOZ, Y. R. *Modelo funcional de la Factoría de Software de la UCI para la línea Carrefour.*, 2005. p.
- HOZ, Y. R. L. and M. M. VALDIVIA. *Modelo Funcional de la Factoría de Software de la UCI para la línea Carrefour*, Universidad de las Ciencias Informáticas, 2005. p.
- IBERMATICA. *GNSIS-Los métodos y técnicas a emplear*, 2008. [2008]. Disponible en:
<http://www.ibermatica.com/ibermatica/integracion/gnsismetodos>
- JACOBSON, I.; G. BOOCH, et al. *El Proceso Unificado de Desarrollo de Software*. Madrid, Pearson Education, 2000. 464 p. 84-7829-036-2
- KUTVONEN, L. *Relating MDA and inter-enterprise collaboration management*, 2008.
[Disponible en:
<http://www.cs.kent.ac.uk/projects/kmf/mdaworkshop/submissions/Kutvonen.pdf>
- LABORDA, E. *Vates y América XXI están desarrollando software para ChileCompra 2.0 con los máximos estándares de certificación mundial 2007*. [2008]. Disponible en:
https://www.chilecompra.cl/Portal/PortalInvitado/noticias/noticias_detalle.aspx?IDC=397
- MARTÍNEZ, N. C. *Estrategia de implantación del Modelo de Factoría de Software aplicando Inteligencia para el Polo Productivo Gestión de Proyecto.*, Universidad de las Ciencias Informáticas, 2007. p.
- MERINDE. *Marco de Desarrollo 2008*. [2008]. Disponible en:
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=538&Itemid=291
- MON, A. *Estudio de la madurez de la producción de software en comparación con la producción industrial*, 2008. [2008]. Disponible en:
<http://is.ls.fi.upm.es/doctorado/Trabajos20022003/Mon.doc>
- NEGRÍN, Y. M. *Modelo de Factoría de Software para la organización de Proyectos Productivos en el Instituto Politécnico "Abel Santamaría"*, Universidad de las Ciencias Informáticas, 2007. p.
- OLAVARRIETA, A. D. and A. B. PADRÓN. *Fábrica de software Un modelo de negocio certificable basado en Estructura y Capacidades*, 2006. [2008]. Disponible en:
<http://www.certum.com/Publicaciones/FabSoft.pdf>
- ONTIVERO, L. *Software Factories (Parte 2)*, 2007. [2008]. Disponible en: <http://software-factories.blogspot.com/2007/09/software-factories-introduccion-parte-2.html>
- PALACIO, J. *Gestión de proyectos ágil: conceptos básicos*, 2006. [2008]. Disponible en:
http://www.navegapolis.net/files/s/NST-003_01.pdf

- PASCO, J. M.; E. M. VERGARA, *et al.* *Procesos activos: Una nueva era para el desarrollo de software de calidad*, 2008. [2008]. Disponible en:
http://www.insovina.cl/seminarios/romero/ROMERO_PROCESOS_ACTIVOS_UNA_NUEVA_ERA_PARA_EL_DESARROLLO_DE_SOFTWARE_DE_CALIDAD.doc
- PRESSMAN, R. S. *Ingeniería del software. Un enfoque práctico*. Mc Graw Hill, 1992. 601 p.
- RAMOS, P. A. L.; J. C. A. FRANCO, *et al.* *Fábrica de Software*, 2002. [Disponible en:
http://www.ci.ulsu.mx/~elinod/docencia/ctrldesa/fabsof_pres.pdf
- SEI. *Software Engineering Institute*, 2008. [Disponible en: <http://www.sei.cmu.edu/>
- VALDELAMAR, E. J. H. *La oportunidad de las Fábricas de Software*, 2007. [2007]. Disponible en: <http://www.rosenblueth.mx/InterFAR/Vol1Num3/doc/Vol1Num3-50.htm>
- VECTORSF. *Página de inicio de la Web Corporativa*, 2008. [Disponible en:
<http://www.vectorsf.com/h/index2.html>
- VERITAS, D. R. P. D. B. *La gestión tradicional y la gestión por procesos*, 2005. [2008]. Disponible en: <http://www.gestiopolis.com/recursos4/docs/ger/gestitra.htm>

ANEXO 1. COMPARACIÓN DE LOS MODELOS DE FACTORÍA DE SOFTWARE.

Aspectos	Modelos de Factoría de Software					
	ISO 9001 y CMM	Eureka	Clasificadorio	Basili	Replicable	Inteligencia
Proceso	Mejora continua del proceso de producción, pero sin definir el proceso	Desarrollo de software distribuido, guiado por reglas, no define su organización.	Especifica el proceso según el alcance	Especialización del área de producción de software y la de producción de componentes	Especialización del área de producción de software y la de producción de componentes y un modelo de calidad. Define un flujo de procesos pero no se guía por una metodología. No sirve para todo tipo de producto.	Define el proceso según una metodología de desarrollo. Vincular la mejora del proceso de producción a la gestión de proyectos y la orientación estratégica.
Clasificación de la Factoría	No menciona nada al respecto	No menciona nada al respecto	Clasifica la factoría según el alcance en: Proyectos Ampliada. Proyectos de Software. Proyectos Físicos. Programas.	No menciona nada al respecto	No menciona nada al respecto	Clasifica la factoría según el alcance de los flujos de trabajo dentro de una metodología de desarrollo de software.
Estándares de calidad	ISO 9001 y CMM, aunque este último se abolió en el 2005	Reglas	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Sí, ISO 9001, CMMI, PSP y TSP
Gestión de proyectos	Define los roles, proceso y la calidad con la ISO 9001	Es responsabilidades de las reglas	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Define los roles, procesos, alcance, costo, tiempo de desarrollo, plan de riesgos y la calidad.
Estructura organizacional	Los roles se agrupan en una entidad y utilizan PSP y TSP	Proceso modelado con reglas, tareas, actividades, y apoyada por	No menciona nada al respecto	No menciona nada al respecto	Los roles se establecen en función de los procesos	Las actividades y tareas de cada rol dependen de la

		herramientas automatizadas, pero no enuncia como organizarlo.				metodología, se apoya en herramientas automatizadas.
Bases tecnológicas	Define una entidad para el soporte técnico	Las herramientas soportan la automatización	No menciona nada al respecto	No menciona nada al respecto	Define una entidad para el soporte técnico y propone solo las herramientas iniciales.	
Reutilización de componentes	Para dar soporte al proceso.	La integración se realiza basado en reglas	No menciona nada al respecto	Establece los procesos para la gestión de la reutilización en el repositorio	Los utiliza para apoyar el proceso y los clasifica en activos del proceso o de infraestructura	Los utiliza para apoyar el proceso. Los clasifica en activos del proceso o de infraestructura y establece los procesos para la gestión de la reutilización en el repositorio
Gestión del conocimiento	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Plantea el uso de la inteligencia empresarial, vigilancia tecnológica, estudio de mercado, prospectiva, estudio de tendencias.

ANEXO 2. COMPARACIÓN ENTRE MODELOS DE PROCESO DE SOFTWARE.

Modelo de proceso	Funciona con requisitos y arquitectura no predefinidos	Produce software altamente fiable	Gestión de riesgos	Permite correcciones sobre la marcha	Visión del progreso por el Cliente y el Jefe del proyecto
Codificar y corregir	Bajo	Bajo	Bajo	Alto	Medio
Cascada	Bajo	Alto	Bajo	Bajo	Bajo
Evolutivo exploratorio	Medio o Alto	Medio o Alto	Medio	Medio o Alto	Medio o Alto
Evolutivo prototipado	Alto	Medio	Medio	Alto	Alto
<i>Desarrollo formal de sistemas</i>	<i>Bajo</i>	<i>Alto</i>	<i>Bajo a Medio</i>	<i>Bajo</i>	<i>Bajo</i>
Desarrollo orientado a reutilización	Medio	Bajo a Alto	Bajo a Medio	Alto	Alto
Incremental	Bajo	Alto	Medio	Bajo	Bajo
Espiral	Alto	Alto	Alto	Medio	Medio

ANEXO 3. DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES

Metodologías Ágiles	Metodologías Tradicionales
Pocos artefactos. El modelado es prescindible, modelos desechables.	Más artefactos. El modelado es esencial, mantenimiento de modelos
Pocos roles, más genéricos y flexibles.	Más roles, más específico.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
Cliente es parte del equipo de desarrollo (además in-situ).	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.

GLOSARIO DE TÉRMINOS Y SIGLAS

A

Artefactos: Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Actividades: Es la suma de tareas, normalmente se agrupan en un procedimiento para facilitar su gestión. La secuencia ordenada de actividades da como resultado un subproceso o un proceso. Normalmente se desarrolla en un departamento o función.

Arquitecturas informáticas: Es la visión común en la que todos los empleados (desarrolladores y otros usuarios) deben de estar de acuerdo, la arquitectura da una clara perspectiva del sistema. Guía el trabajo de los desarrolladores con el sistema.

B

Bases Tecnológicas: Se orientan para llevar acabo el proyecto, sobre todo plantean conocimiento en las tecnologías para la construcción del software, la gestión y el soporte del mismo.

C

Componentes: Los componentes de software son los recursos desarrollados para un fin concreto y que puede formar solo o junto con otros un entorno funcional requerido por cualquier proceso predefinido.

Cliente: Son las personas para las cuales se elabora un producto determinado.

CMM: Capability Maturity Model

CMMI: Capability Maturity Model Integration

CNTI: Centro Nacional de Tecnologías de la Información

Calidad: Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.

Calidad del software: Es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. La calidad del software ha pasado de una simple inspección y detección de errores a un cuidado total en su proceso de fabricación, desarrollo y mantenimiento; y es que el correcto funcionamiento de éste es fundamental para el óptimo comportamiento de los sistemas informáticos.

Capacitación: Es toda acción organizada y evaluable que se desarrolla en una empresa para modificar, mejorar y ampliar los conocimientos, habilidades y actitudes del personal en

conductas produciendo un cambio positivo en el desempeño de sus tareas. El objetivo es perfeccionar al profesional en su puesto de trabajo.

D

Despliegue: Producir una versión del producto y entregar el software a sus usuarios finales.

E

Eficiencia: Capacidad para lograr un fin empleando los mejores medios posibles

Eficacia: Capacidad para obrar o para conseguir un resultado determinado.

Estrategia: Conjunto de acciones que se llevan a cabo para lograr un determinado fin.

Estructura Organizacional. Estructura que descompone la labor de la compañía en tareas especializadas, asigna éstas a personas y departamentos y coordina las tareas mediante la definición de vínculos formales entre personas y departamentos (Y) estableciendo línea de autoridad y comunicación.

Entorno empresarial: es el área que rodea a la empresa en el que desarrolla su actividad. De este modo, la empresa puede considerarse como un sistema abierto al medio en el que se desenvuelve, en el que influye y recibe influencias

Equipo de desarrollo: Es un grupo de trabajo constituido por una serie de profesores, investigadores, colaboradores y alumnos unidos en la ilusión de acometer un determinado proyecto o avanzar en el conocimiento y en la investigación teórica y aplicada.

ERP: Sistema de administración de un negocio, (Enterprise Resource Planning).

Estándar: Lo que es establecido por la autoridad, la costumbre o el consentimiento general. En este sentido se utiliza como sinónimo de norma.

Evento sistémico: Es un evento de alto nivel generado por un actor externo. Se asocia a operaciones del sistema: las que se emiten en respuesta a los eventos del sistema. Por ejemplo, cuando un cajero que usa una terminal de punto de venta oprime el botón "terminar venta", está generando un evento sistémico que indica que "la venta ha terminado". Del mismo modo, cuando alguien que usa un editor de texto pulsa el botón "revisar ortografía", está produciendo un evento del sistema.

F

Factoría: 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.

Factoría de software: Una Factoría de Software es el sentido de producir con rapidez y calidad a través de procesos conocidos, repetibles y mejorables continuamente. Incorpora técnicas, metodologías y herramientas en el desarrollo del software, que mantiene una mejora continua de procesos y trae como resultados la industrialización en la producción de software.

Flujo de trabajo: Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Fases: Cada uno de los estados sucesivos de una algo que cambia o se desarrolla. Una diferencia verdadera de tiempo.

FDD: Feature Driven Development

FS: Fábricas o Factorías de software

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Flujos de Trabajo: es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las actividades, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, como fluye la información que soporta las tareas y como se le realiza seguimiento al cumplimiento de estas.

G

Gestión de proyectos: Es la aplicación de varios conocimientos, habilidades, herramientas y técnicas para planificar y satisfacer las actividades de un proyecto.

Gestión del Capital Humano: Función administrativa en la que se maneja el reclutamiento, asignación, capacitación y el desarrollo de los miembros de una organización o empresa.

Gestión del Capital Humano: Función administrativa en la que se maneja el reclutamiento, asignación, capacitación y el desarrollo de los miembros de una organización o empresa.

Gestión: Gestión es la acción y efecto de gestionar o la acción o efecto de administrar. Comprende todas las actividades de una organización que implican el establecimiento de metas u objetivos, así como la evaluación de su desempeño y cumplimiento; además del desarrollo de una estrategia operativa que garantice la supervivencia de la misma, según al sistema social correspondiente.

H

Herramientas: Utensilios o provisiones necesarias para poder emprender un proyecto de software. Soportan los procesos de desarrollo de software modernos.

I

ISO 9001: La norma ISO 9001, es un método de trabajo, con el fin de mejorar la calidad y satisfacción de cara al consumidor. Esta dirigido a mejorar los aspectos organizativos de una empresa.

ISO: Organización Internacional de Estándares.

L

Línea temática: La fábrica es un área donde se forma personal, se produce e investiga sobre un tipo de producto para un entorno de negocio determinado.

M

Metodología: Es un conjunto de métodos de investigación apropiados al quehacer de una ciencia determinada. Es la parte de un proceso de investigación que permite sistematizar los métodos y técnicas para realizar el proceso con eficiencia.

Modelo de Factoría de software: Es una forma de representar el enfoque de Factoría de Software.

P

Programación Extrema: Es una metodología de desarrollo de software ágil centrada en potenciar las relaciones interpersonales como clave para el desarrollo de software.

Producto: es cualquier cosa que puede ser ofrecida al mercado para su compra, para su utilización o para su consideración. Es cualquier bien, servicio o idea capaz de motivar y satisfacer a un comprador.

Producto de software: Son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

Proceso: Es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin.

Proceso de software: Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto para transformar los requisitos de usuario en un producto.

Prueba: Es un proceso que se desarrolla a lo largo del ciclo de vida de un software con la intención de encontrar errores previos a la entrega final. Es una actividad en la cual un sistema o componente es eficiente bajo unas condiciones o requerimientos específicos.

PYME: Pequeñas y Medianas Empresas

PMI: Project Management Institute

PMBOK: Project Management body of knowledge.

Planificación: La planificación es el establecimiento de objetivos, y la decisión sobre las estrategias y las tareas necesarias para alcanzarlas.

Parametrizar: hacerlo característico para cada entorno, pero que a la vez sea reutilizable

PSP: Proceso de Software Personal.

Productividad del trabajo: Es un indicador que refleja que tan bien se están usando los recursos de una economía en la producción de bienes y servicios

R

Reutilización: Reutilizar es la acción de volver a utilizar los bienes o productos ya elaborados y probados. Puede venir propiciada por una mejora o restauración o sin modificarse, usarlo en la creación de un nuevo producto.

Repositorio: Es un sitio centralizado donde se almacena y mantiene información digital, como bases de datos o archivos informáticos.

Rol: Es el papel que desempeña una persona o grupo de ellas en cualquier actividad. Una persona puede tener varios roles, así como un rol puede ser desempeñado por varias personas.

RUP: Rational Unified Process. Es un proceso iterativo e incremental para el desarrollo del software creado por Rational Software.

Recursos: Son todos aquellos elementos necesarios, tanto tangibles como intangibles, para que una organización cumpla con sus objetivos.

RTF: Revisión Técnica y Formal

RRHH: Recursos Humanos

S

Software: Todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Salidas: Resultado del procedimiento.

SEI: Software Engineering Institute

SOA: Arquitectura Orientada a Servicios.

T

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

TIC: Tecnologías de la Información y las Comunicaciones.

TSP: Proceso de Software en Equipo.

U

Usuario: Individuo que suele llamarse consumidor, usufructuario, beneficiario o cliente que habitualmente utiliza algo ajeno por derecho o por concesión.

UCI: Universidad de Ciencias Informáticas.

X

XP: Extreme Programming

XML: Extensible Markup Language