

Universidad de las Ciencias Informáticas



Título: Propuesta de modelo de desarrollo para líneas de productos de software en centros de producción

Trabajo de Diploma para optar por el título de Máster en Gestión de Proyectos Informáticos

Autor: Ing. Henrik Pestano Pino

Tutor: Dr.C Pedro Yobanis Piñero Pérez

CIUDAD DE LA HABANA, FEBRERO 2011

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Inq. Henrik Pestano Pino soy el autor principal de la tesis de maestría PROPUESTA DE MODELO DE DESARROLLO PARA LÍNEAS DE PRODUCTOS DE SOFTWARE EN CENTROS DE PRODUCCIÓN desarrollada como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio así como los derechos patrimoniales de la misma con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Henrik Pestano Pino

Nombre del autor

Firma

Dedicatoria

A mis padres que siempre me han apoyado y gracias a ellos he llegado hasta aquí...

Muy especialmente a mi novia Mónica, por estar a mi lado en las buenas y en las malas, por comprenderme y aceptarme como soy, por ayudarme a encontrar la verdadera esencia de la vida...

A toda mi familia (la de siempre y la nueva) por darme su apoyo y demostrarme que sin ustedes no valdría la pena vivir...

Agradecimientos

A Pedro mi tutor, por ser mi amigo y haber confiado siempre en mí...

A todos mis amigos y compañeros de trabajo, de la Facultad 3, del ERP, de CENTALAD y más recientemente de la Dirección Técnica, por haberme enseñado lo importante que es compartir.

A la Universidad de las Ciencias Informáticas, mi segunda casa y el lugar donde me convertí en el profesional que soy.

“Un proceso del software es un conjunto de actividades que conducen a la creación de un producto software. Estas actividades pueden consistir en el desarrollo de un software desde cero en un lenguaje de programación estándar como Java o C. Sin embargo, cada vez más, se desarrolla nuevo software ampliando y modificando los sistemas existentes y configurando e integrando software comercial o componentes del sistema.”

“... un modelo del proceso del software es una representación abstracta de un proceso del software”

Sommerville

RESUMEN

En la presente investigación se realiza un estudio del estado del arte de los modelos de desarrollo basados en la reutilización, analizando las principales tendencias en Europa, Norte América y Sur América. Toma de cada una de estas corrientes los aspectos positivos y conforma un modelo basado en los principios de las líneas de productos de software, la arquitectura de empresas y la mejora continua. Además incluye un guía para su implantación donde se detallan cada uno de los pasos a seguir. Toma como punto de partida las experiencias obtenidas en la organización del Centro de Tecnologías de Almacenamiento y Análisis de Datos y de la Dirección Técnica de la Producción. En ambas áreas existía un elevado número de compromisos productivos y una necesidad urgente de reutilizar los resultados obtenidos, para poder cumplir con los compromisos productivos, motivando el surgimiento del modelo. Se presentan los resultados de su aplicación en función del aumento de la productividad de los equipos de trabajo y de la cantidad de productos desarrollados.

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN..... | 1 |
| DISEÑO TEÓRICO | 2 |
| DISEÑO METODOLÓGICO..... | 3 |
| CAPITULO 1 FUNDAMENTACIÓN TEÓRICA | 5 |
| 1.1. PRINCIPALES ENFOQUES SOBRE EL CONCEPTO DE REUTILIZACIÓN EN LA INDUSTRIA DEL SOFTWARE..... | 5 |
| 1.2. CONCEPTO DE LÍNEAS DE PRODUCTOS DE SOFTWARE | 6 |
| 1.3. ANÁLISIS DE ALGUNOS MODELOS DE LÍNEAS DE PRODUCTOS DE SOFTWARE | 7 |
| 1.3.1. Modelos de Europa | 7 |
| 1.3.1. Modelos de Suramérica | 14 |
| 1.3.2. Modelos de Norteamérica | 17 |
| 1.4. ARQUITECTURA DE EMPRESAS, VISIÓN GENERAL..... | 21 |
| CONCLUSIONES DEL CAPÍTULO | 23 |
| CAPÍTULO 2 DESARROLLO DEL MODELO PARA LÍNEAS DE PRODUCTOS DE SOFTWARE | 25 |
| 2.1. PRINCIPIOS GENERALES DEL MODELO | 25 |
| 2.1.1. Modelo centrado en arquitectura de empresas..... | 25 |
| 2.1.2. Modelo centrado en la mejora continua..... | 26 |
| 2.1.3. Modelo centrado en líneas de productos de software..... | 27 |
| 2.2. PASOS FUNDAMENTALES PARA LA APLICACIÓN DEL MODELO | 28 |
| 2.2.1. Paso 1: Diagnóstico del estado actual de la organización..... | 28 |
| 2.2.2. Paso 2: Definición del modelo ideal a alcanzar en la organización. | 30 |
| 2.2.3. Paso 3: Definición del cronograma de los modelos de transición..... | 39 |
| CONCLUSIONES DEL CAPÍTULO | 44 |
| CAPITULO 3 VALIDACIÓN DEL MODELO PROPUESTO..... | 45 |
| 3.1 DESCRIPCIÓN DE LA MUESTRA | 45 |
| 3.2 DATOS Y EXPERIENCIA PRÁCTICA DE LA APLICACIÓN EN EL CENTALAD..... | 45 |
| 3.2.1 Diagnóstico del estado actual de la organización..... | 45 |
| 3.2.2 Modelo ideal definido..... | 46 |
| 3.2.3 Evaluación de los resultados de la aplicación del modelo en CENTALAD..... | 48 |
| 3.3 DATOS Y EXPERIENCIA PRÁCTICA EN LA DIRECCIÓN TÉCNICA. | 52 |
| 3.3.1 Diagnóstico del estado actual de la organización..... | 52 |
| 3.3.2 Modelo ideal definido..... | 52 |
| 3.3.3 Evaluación de los Resultados de la aplicación del modelo en la DT..... | 58 |
| CONCLUSIONES DEL CAPÍTULO | 64 |
| CONCLUSIONES | 65 |
| RECOMENDACIONES..... | 66 |
| BIBLIOGRAFÍA REFERENCIADA | 67 |
| BIBLIOGRAFÍA CONSULTADA | 69 |
| ANEXOS | 71 |

INTRODUCCIÓN

La Ingeniería de Software surge como la aplicación de modelos y métodos de las ingenierías tradicionales en el contexto del desarrollo de productos de software, situación que ha condicionado su evolución por tener como guías los conceptos que en otros ámbitos tiene la ingeniería. Históricamente han surgido varios enfoques que buscan abordar de manera sistemática, la planificación, análisis, diseño e implementación de los proyectos de desarrollo de software, sean estos a gran escala, pequeñas aplicaciones o herramientas, contruidos a la medida o no. Cada uno de estos enfoques tiene su origen en las concepciones dominantes en su momento histórico y sobre todo, en la búsqueda incesante de mejoras a los enfoques precedentes.

Desde los mismos inicios de la construcción de programas computacionales, el concepto de reutilización estuvo presente, evolucionando a partir de las subrutinas en la década del 60 y la programación estructurada en los 70, cuando se comenzaron a organizar los programas a partir de módulos o pequeños programas como librerías matemáticas. A partir de allí la programación orientada a objetos comienza a tomar fuerza y supone hasta nuestros días uno de los niveles más altos de concentración de funciones en una única entidad, las clases. El término más importante en el ámbito de la reutilización y que ha cambiado la ingeniería de software es el concepto de componentes. Comienza a diferenciar dos estilos de desarrollo: la construcción de software basado en la reutilización y el desarrollo de software para la reutilización. A partir de comienzos de este siglo ha emergido, como elemento central de la reutilización, el concepto de activos de software reutilizable, que no es más que un artefacto diseñado para ser utilizado múltiples veces en el desarrollo de diferentes aplicaciones. Las características de estos activos de software han obligado a crear nuevos modelos de desarrollo de software basados en las líneas de productos.

Las líneas de productos de software (LPS) rápidamente se han convertido en una importante forma de organización de procesos de desarrollo de software centrado en la reutilización y en una solución viable a las demandas más exigentes, posibilitando que pequeñas empresas aumenten considerablemente la eficiencia de los tiempos de entrega al mercado, los costos, la productividad y la calidad sus productos. Producir siguiendo este enfoque también permite rápidamente la exploración y entrada a nuevos mercados, puesto que construir sistemas flexibles que proporcionan la capacidad de personalización y adaptación a nuevos requerimientos, aumenta el espectro de posibilidades de negocios.

La presente investigación tiene como propósito principal, presentar un modelo de desarrollo basado en los conceptos de Líneas de Productos de Software y está orientada por las corrientes más actuales de reutilización y desarrollo basado en componentes. Toma como punto de partida las experiencias obtenidas en el desarrollo de productos en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD) y en la Dirección Técnica de la Infraestructura Productiva (DT).

Situación problemática

La mayoría de los modelos de desarrollo de software utilizados hasta el momento en el contexto de la Universidad estaban basados en los modelos evolutivos. Lo más común en estos casos era la creación de equipos multidisciplinarios para el desarrollo de proyectos que implementaban soluciones a la medida para determinados problemas. En general el nivel de reutilización que genera este tipo de soluciones es

muy bajo, producto en muchos casos a la urgencia de los cronogramas, la inexperiencia de los equipos de trabajo y a la duplicidad de esfuerzos en la búsqueda de soluciones para problemas comunes como la generación de reportes, la réplica de datos, la autenticación o la gestión documental. De esta manera se producían aplicaciones informáticas sin reutilizar los activos generados durante el proceso de desarrollo. Esta manera de hacer software funcionaba mientras los compromisos de negocios no superaran los límites de la organización en cuanto a su capacidad para crear equipos de trabajo para cada uno de los proyectos. Además de crear islas de conocimientos alrededor de la experiencia que se generaba en el desarrollo de los proyectos y donde por lo general los activos de software que conformaban los productos desarrollados quedaban archivados como historia de la organización y no dentro de un marco de reutilización y de ensamblaje, que permitiera construir paralelamente varias soluciones similares como productos específicos.

En este marco se encontraba el centro de producción CENTALAD durante su creación y posteriormente la Dirección Técnica durante su reordenamiento. Con un elevado número de compromisos productivos con la dirección de la Universidad, insuficiente fuerza laboral y un ajustado cronograma para cumplirlos.

Diseño teórico

Formulación del problema

No disponer de un modelo de desarrollo de software, basado en la reutilización de los activos de software y los recursos de la organización, afecta los niveles de productividad desde el enfoque de la obtención de nuevos productos y la eficiencia del desarrollo en el Centro de Tecnologías de Almacenamiento y Análisis de Datos y en la Dirección Técnica de la Producción.

Objeto de Estudio

Modelos de desarrollo de software.

Objetivo General

El objetivo general de este trabajo es:

- Desarrollar un modelo basado en las líneas de productos de software y en los principios de la arquitectura de empresas que permita incrementar la productividad en la construcción de aplicaciones informáticas.

Campo de Acción

Modelos de desarrollo basados en la reutilización

Objetivos Específicos

- Formalizar un estudio del estado del arte de los modelos de desarrollo de software y de las mejores prácticas que potencian la reutilización en la producción de aplicaciones informáticas.
- Desarrollar un modelo basado en las líneas de productos de software que posibilite un aumento en la cantidad de productos y un incremento de la productividad.
- Evaluar el modelo desarrollado a partir de su aplicación en el Centro de Tecnologías de Almacenamiento y Análisis de Datos y en la Dirección Técnica de la Producción.

Hipótesis

Si se desarrolla un modelo basado en líneas de productos de software que potencie la reutilización de los activos de software y los recursos de la organización, se obtendrá una mayor productividad,

expresada en el aumento de la cantidad de productos y en el aumento de la eficiencia del desarrollo en las áreas CENTALAD y DT.

Variables

Variables independientes: Modelo de desarrollo basado en líneas de productos de software.

Variables dependientes: Productividad

Operacionalización de las variables

| Variables dependientes | Dimensiones | Indicadores | UM |
|------------------------|-------------------------------|-------------------------|---|
| Productividad | Obtención de nuevos productos | Productos desarrollados | Cantidad |
| | Eficiencia del desarrollo | Eficiencia | $\frac{\text{Tamaño del producto obtenido}}{\text{Recursos}}$ |

Tabla 1 Operacionalización de la variable dependiente.

| Variables independientes | Dimensiones | Indicadores | UM |
|--|--------------------|---|-----------------------------|
| Modelo de desarrollo basado en líneas de productos de software | Calidad del modelo | Calidad de la documentación | Adecuada Media Baja |
| | | Capacidad de ser generalizado | Alta Media Baja |
| | | Complejidad (áreas del conocimiento que abarca) | Todas Algunas Ninguna |

Tabla 2 Operacionalización de la variable independiente.

Diseño Metodológico

Unidad de estudio

El proceso de organización del desarrollo de proyectos de software en centros de producción de la Universidad de las Ciencias Informáticas.

Población

Los departamentos de desarrollo de software del Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD) y la Dirección Técnica de la producción (DT).

Selección muestral

Técnicas de muestreo

La población a analizar está constituida por tres departamentos del centro de producción CENTALAD y por la Dirección Técnica. En todos los casos se generan productos con características similares al ser todos sistemas de información, aunque existen dos grupos bien marcados de sistemas de información: los sistemas decisionales (herramientas para el análisis y soluciones de almacenes de datos) desarrollados en CENTALAD y los sistemas transaccionales (herramientas de gestión de proyectos)

construidos en la DT. Como técnica de muestreo se utilizó una selección prescriptiva donde se escogieron tres unidades de producción que representan el 75% de la población:

- CENTALAD - Departamento de desarrollo de herramientas de análisis datos.
- CENTALAD - Departamento de desarrollo de almacenes de datos.
- DT – Grupo de desarrollo de soluciones de gestión de proyectos.

Métodos y Técnicas Investigativas a utilizar

Se utilizó una estrategia descriptiva en el desarrollo de la investigación

Métodos Teóricos

- Histórico Lógico
- Hipotético deductivo

Los modelos de desarrollo orientados a líneas de productos fueron revisados desde un enfoque histórico lógico, realizando un estudio del estado del arte del tema; analizando las ventajas y limitaciones de cada uno de los modelos. La presente investigación sigue además un método hipotético deductivo porque desde un inicio el problema fue analizado como un todo, revisando los aspectos más generales y los datos tomados de los proyectos que forman parte de la muestra. A partir de este análisis se descompuso el problema en elementos particulares de cada una de las áreas de la gestión de proyectos según diferentes centros de conocimiento en el mundo, principalmente el Project Management Institute (PMI), el Software Engineering Institute (SEI) y el Programa de Mejora de la Universidad de las Ciencias Informáticas. Por otra parte se utilizarán métodos empíricos ya que estos describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional, específicamente las entrevistas y la revisión de documentos (ROLANDO ALFREDO HERNÁNDEZ LEÓN 2002).

Aporte teórico

- Propuesta de una guía para el diseño de líneas de productos de software en entidades productoras de soluciones informáticas.

Aporte práctico

- Aumento de la productiva en CENTALAD y en la DT a partir de la implantación de modelo de producción basado en líneas de productos de software.
- Aumento de la capacidad de obtener nuevos productos potencialmente reutilizables y comercializables.

Estructura del trabajo

Esta investigación está estructurada en tres capítulos de ellos el primero se refiere a la fundamentación teórica del tema, particularmente incluye un estudio del estado del arte sobre la posición de los modelos de desarrollo basados en la reutilización y el concepto de arquitectura de empresa, caracterizando y analizando desde una posición crítica los modelos estudiados y concluyendo con la selección de los principios a seguir. El capítulo dos presenta el desarrollo de una guía metodológica para la implantación del modelo propuesto, según los principios de arquitectura de empresas. Definiendo tres etapas: diagnóstico del estado actual de la organización, diseño del modelo ideal a alcanzar y diseño de los modelos de transición que identifican el camino hasta el estado ideal. En el capítulo tres se analizan todos los datos y resultados de su aplicación en CENTALAD y en la Dirección Técnica.

CAPITULO 1 FUNDAMENTACIÓN TEÓRICA

El presente capítulo tiene como objetivo, presentar la fundamentación teórica de la investigación comenzando por describir elementos generales de la evolución del término *reutilización* y algunos conceptos relacionados con líneas de productos de software. Además se presenta un resumen de las características de algunos modelos de desarrollo de software basados en la reutilización y una visión general de la arquitectura de empresa.

1.1. Principales enfoques sobre el concepto de reutilización en la Industria del Software.

La reutilización de software no es un proceso nuevo, se ha venido utilizando desde inicios de los años 60 con las subrutinas. En los años 70 cuando surge la programación estructurada y la posibilidad de organizar los programas a partir de módulos que eran superfunciones o pequeños programas que podían ser llamados en el momento en que se necesitasen por el programador (ISORNA 1999), durante esta etapa comienzan a comercializarse librerías matemáticas y estadísticas tales como ISML, SAS y SPSS. A principios de los 80 comienza a tomar fuerza un nuevo concepto en la programación, el de los objetos, que si bien es justo aclarar que es utilizado por primera vez a principios de los 70 con un lenguaje de programación denominado SIMULA, no es hasta la década de los 80 que se dan importantes pasos en el nuevo paradigma de la programación orientada a objetos que supone hasta nuestros días uno de los niveles más altos de condensación de funciones en una entidad unitaria, las clases (ISORNA 1999).

En los albores de los 90 surge el término más importante que ha cambiado y sigue cambiando los procesos de ingeniería de software, el concepto de “componente”. Inicialmente surge ante la necesidad de reutilizar partes o módulos de software existentes que podían ser utilizadas para el desarrollo de nuevas funcionalidades a las aplicaciones o para la construcción de nuevas aplicaciones. Pero esto suponía un gran esfuerzo, pues había que localizar estas partes reutilizables y almacenarlas en repositorios especiales que más tarde pudieran ser consultadas en fase de diseño. Con el término componente se empieza a diferenciar dos estilos de desarrollo de software, por un lado el desarrollo de software basado en reutilización, donde las aplicaciones se construyen a partir de otras partes software ya existentes y accesibles en repositorios conocidos y por otro lado el desarrollo de software de reutilización, donde se ponen en práctica procesos de ingeniería para la elaboración de partes eficientes de software que luego pueden ser utilizadas para la construcción de nuevas aplicaciones. A estas partes software se las conoce como componentes software y han dado lugar a los paradigmas de programación de componentes top-down o descendente (para reutilizar) y bottom-up o ascendente (basado en reutilización) (IRIBARNE 2003).

Otro avance tecnológico que ha influido significativamente en la reutilización fue el surgimiento de las arquitecturas orientadas a servicios (del inglés Service-Oriented Architecture SOA) que comienzan a tomar fuerza desde finales de la década de los noventa y principios de este siglo, cuando la industria del software se planteó la necesidad de crear software débilmente acoplado, orientado fundamentalmente el desarrollo a satisfacer la rapidez operativa del entorno comercial en que se desenvuelven las organizaciones. Para lograrlo fue necesario que los componentes de esta infraestructura fuesen tan reutilizables y tan poco interdependientes que permitirán una rápida reestructuración de los

misimos(GALLEGOS 2007). En general SOA trasciende el concepto de una tecnología en particular, como web services por ejemplo, las aplicaciones construidas bajo estos principios tienen todas sus funciones definidas como servicios independientes con interfaces invocables bien definidas y pueden ser llamados en secuencias definidas para conformar procesos de negocio (B2B-SOLUTIONS-GROUP 2009).

En este sentido el concepto de líneas de producción resurgió a comienzos de este siglo aplicándose a la reutilización en la industria del software, convirtiéndose rápidamente en un importante paradigma del desarrollo de sistemas informáticos y en una solución viable a las demandas más exigentes, posibilitando que pequeñas organizaciones aumentasen considerablemente la eficiencia de los tiempos de entrega de sus productos, los costos, la productividad y la calidad de los mismos. En el caso de la industria del software las líneas de producción se ven reflejadas en las “líneas de productos de software” que permiten construir sistemas flexibles que proporcionan la capacidad de personalización y adaptación a nuevos requerimientos, aumenta el espectro de posibilidades de negocios.(SEI 2009)

1.2. Concepto de Líneas de Productos de Software

Una línea de productos software es un conjunto de sistemas de software de un entorno de negocios determinado, que tienen funcionalidades en común. El desarrollo basado en Líneas de Producto busca aprovechar esta parte común para desarrollar de forma eficiente y sistemática nuevos miembros de la familia de productos. Mediante la ingeniería de Líneas de Producto, las organizaciones pueden reducir su esfuerzo de desarrollo, acortar los tiempos de entrega al mercado de nuevos productos, facilitar su mantenimiento y evolución, además de poder planificar de forma conjunta el desarrollo y el mantenimiento.

Para entender el término, debería partirse de la definición del concepto que tienen varios autores:

...tener una línea de productos implica tener una arquitectura genérica a partir de la cual se derivan de alguna forma todos sus componentes. Tiene que ser posible, tanto abstraer una arquitectura general a partir de los componentes de la familia, como derivar los diferentes productos individuales a partir de la arquitectura general.”(PERRY 1998)

“... es un conjunto de sistemas de software que comparten un conjunto común y gestionado de aspectos que satisfacen las necesidades específicas de un segmento de mercado o misión y que son desarrollados a partir de un conjunto común de activos fundamentales de software de una manera pre-escrita”(CLEMENTS, PAUL and NORTHROP 2002)

“...consiste de una familia de sistemas de software que tienen una funcionalidad común y alguna funcionalidad variable”(GOMAA 2005)

- La funcionalidad común descansa en el uso recurrente de un conjunto común de activos reutilizables
- Los activos son reutilizados por todos los miembros de la familia

“...se refieren a técnicas de ingeniería para crear un portafolio de sistemas de software similares, a partir de un conjunto compartido de activos de software, usando un medio común de producción”. (KRUEGER 2006)

Las líneas de productos de software (ver Fig. 1) representan a familias de productos en un entorno de negocio o de conocimiento específico con un alcance bien definido. Las aplicaciones que pertenecen a uno de estos dominios, comparten aspectos comunes como: la arquitectura, las tecnologías base y

algunas funcionalidades. Están construidas a partir de grupos de componentes reutilizables o servicios y el centro de la reutilización son los Activos de Software¹. (NORTHROP 2008)

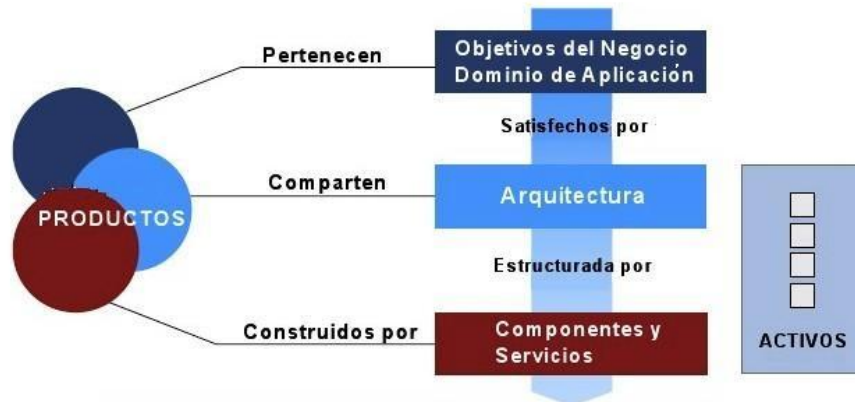


Fig. 1 Líneas de productos de software.

1.3. Análisis de algunos modelos de Líneas de Productos de Software

Los modelos de desarrollo de líneas de productos se han desarrollado a lo largo de todo el mundo, a continuación se realiza un análisis de algunas regiones y se exponen las ventajas y limitaciones de algunos de los modelos más significativos.

1.3.1. Modelos de Europa

Proceso de desarrollo PuLSE

PuLSE (Product Line Software Engineering) fue creado en el Fraunhofer IESE (Institute for Experimental Software Engineering) una de las principales organizaciones que investigan el tema en Europa (KNAUBER *et al.* 2000). Está organizada en tres conjuntos de elementos, ver Fig. 2, el primero de ellos recoge el conjunto de etapas que comprenden el ciclo de vida, el segundo, un conjunto de componentes técnicos que dan soporte a cada una de estas etapas y el tercero incluye un conjunto de componentes de soporte que proporcionan información adicional al proceso de desarrollo.(IESE 2004)

Las fases del ciclo de vida PuLSE describen las distintas actividades que deben llevarse a cabo en una línea de productos.(CHICOTE 2005)

1. **Inicialización:** Durante esta etapa se adaptan los componentes al dominio específico de las aplicaciones que se van a desarrollar a partir de la LPS.
2. **Construcción de un núcleo de infraestructuras comunes y reutilizables:** En esta fase se analizan los aspectos comunes y variables de los productos que se van a derivar de la LPS. Se divide a su vez en las tres etapas siguientes:
 - Definición del alcance de la LPS: Durante esta etapa se acota la línea estableciéndose qué productos podrán derivarse o partir de ella.

¹ Artefacto diseñado expresamente para ser utilizado múltiples veces en el desarrollo de diferentes sistemas o aplicaciones, pueden ser: componente de software, una especificación de requisitos, un algoritmo, un esquema de base de datos, una especificación de diseño, etc.

- Modelado de análisis: Durante esta etapa se modelan los elementos comunes y variantes de los productos incluidos en el alcance de la línea y se genera un mapa de las distintas opciones y alternativas de configuración de los distintos productos.
 - Definición de la arquitectura software de la línea: En esta etapa se describe la arquitectura genérica compartida por todos los productos.
3. **Construcción de nuevos productos a partir de las infraestructuras creadas:** En esta fase se instancian los nuevos productos a partir de las infraestructuras previamente creadas.
 4. **Mantenimiento y evolución de la LPS:** Los requisitos de la línea pueden evolucionar a lo largo del tiempo incluyendo nuevos productos, excluyendo otros o incorporando nuevas características a los productos considerados dentro del alcance de la línea.

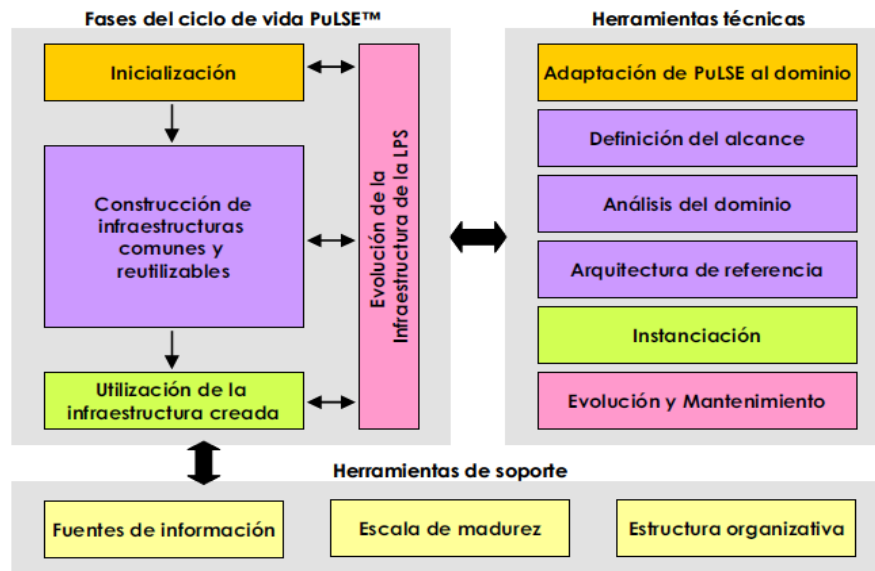


Fig. 2 Vista general del método PuLSE.

La metodología PuLSE proporciona un conjunto de seis componentes o herramientas técnicas que facilitan el desarrollo de las etapas de su ciclo de vida:

1. **Adaptación de PuLSE al dominio:** Esta herramienta permite establecer el contexto empresarial en el que se desarrolla la línea y adaptar convenientemente la metodología.
2. **Delimitación del alcance:** Este componente técnico está formado por un conjunto de herramientas que permiten identificar, describir y acotar la línea, determinando las características de los productos que se construirán a partir de ella.
 - Plantilla de descripción de productos.
 - Herramientas para la identificación y descripción de las áreas funcionales (dominios).
 - Tabla de características.
 - Plantilla de descripción de dominios.
 - Mapa de la estructura de dominios.
 - Mapa inicial de productos.
 - Herramientas de valoración y selección de dominios.
 - Estudio de la relevancia.

- Estudio del potencial de reutilización.
 - Selección de dominios.
3. **Análisis del dominio:** Este componente técnico está formado por un conjunto de herramientas que permiten modelar los dominios previamente seleccionados.
 - Modelos de Casos de Uso
 - Diagrama de Caso de Uso.
 - Plantilla para la descripción de Caso de Uso.
 - Diagrama de variabilidad.
 - Tabla de decisión.
 4. **Arquitectura de referencia:** Esta herramienta permite especificar la arquitectura de referencia de la línea de productos mediante las distintas vistas arquitectónicas.
 5. **Instanciación:** Esta herramienta permite especificar, construir y validar distintos productos a partir de los resultados obtenidos en las fases previas de la metodología.
 6. **Evolución y mantenimiento:** Esta herramienta guía al resto de componentes técnicos de centrándose en tres aspectos fundamentales.
 - Aspectos de gestión.
 - Aspectos de evolución.
 - Aspectos de aprendizaje.

Las componentes de soporte proporcionan directrices de ayuda a los distintos componentes técnicos por lo que pueden considerarse herramientas de apoyo transversales:

1. **Puntos de entrada de información:** Fuentes de información externas (estudios de mercado, estudios de la competencia, etc.) o internas (análisis de productividad, información relativa a productos desarrollados previamente, etc.).
2. **Escala de madurez:** Herramienta utilizada para evaluar la calidad tanto del proceso como de los resultados obtenidos durante la aplicación de la metodología PuLSE a fin de identificar y mejorar los puntos débiles detectados.
3. **Estructura organizativa:** Esta herramienta es la encargada de estudiar si la estructura organizativa que sustenta la LPS es la más adecuada y eficiente, proponiendo los cambios necesarios.

Análisis de ventajas

- El modelo está clara y extensamente documentado, tanto de manera teórica como con casos prácticos.
- Apuesta por el uso de notaciones estándar como UML (Unified Modeling Language).
- El modelo introduce dentro de los componentes de soporte, actividades relacionadas con la gestión documental, la gestión de los recursos humanos y con la mejora continua.
- Sigue un enfoque orientado a componentes que resulta compatible y fácil de integrar con otras metodologías.
- Incluye dentro del componente técnico de Evolución y Soporte la gestión de integración.

Limitaciones del modelo

- No especifica un proceso de gestión de repositorios de activos de software reutilizables.

- El modelo evalúa la calidad tanto del proceso como de los resultados obtenidos durante la aplicación de la metodología, pero no contempla como principio la Calidad Total.
- La función del componente de soporte relacionado con los recursos humanos se limita a controlar el desempeño de la estructura organizativa que sustenta la línea y no se enfoca en la gestión de los recursos humanos de manera integral.
- La gestión de los recursos materiales no está contemplada en el modelo.

Proceso de desarrollo PRAISE

PRAISE (Product-line Realisation and Assessment in Industrial Settings) es un proceso definido en el ESI (European Software Institute) que propone un esquema parecido al ciclo de vida en cascada, añadiéndole una fase de análisis de dominio cuyo resultado es la elaboración de una arquitectura de referencia. Este proyecto constituyó una referencia para el paradigma de desarrollo de Familias de Productos en Europa y es la base del modelo CAFE. (ITEA 2001)

La implementación del modelo tiene como resultado la producción de componentes reutilizables, ver Fig. 3.

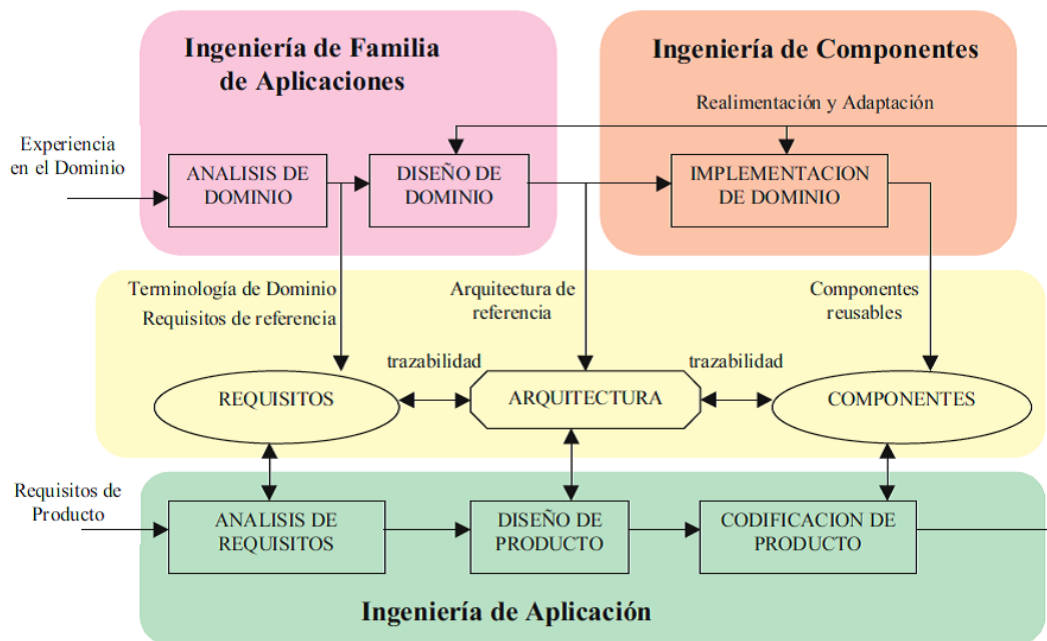


Fig. 3 Proceso de desarrollo PRAISE.

La Ingeniería de Familias de Aplicaciones produce los activos que el proceso de Ingeniería de Aplicaciones reutilizará, involucrando las siguientes actividades:

- **Análisis del Dominio:** Determina las características de la familia de productos.
- **Diseño del Dominio:** Decide qué plataforma de componentes se necesita.
- **Implementación del Dominio:** Construye o adquiere los componentes y la infraestructura de soporte necesarias.

Actividades que comprende la Ingeniería de Aplicación:

- **Análisis de los requisitos de la aplicación:** Determina las funcionalidades y características que un producto particular debe tener.
- **Diseño de la aplicación:** Selecciona los componentes para hacer el producto.
- **Codificación de la aplicación:** Integra los componentes seleccionados usando la arquitectura definida, adicionando adecuaciones o desarrollando nuevos componentes.

Análisis de ventajas

- Están bien definidas en el modelo las áreas de procesos relacionadas con Ingeniería de Dominio y la Ingeniería de Aplicaciones.
- Establece la trazabilidad desde los requisitos hasta los componentes.
- Establece un conjunto de decisiones del diseño de los componentes y productos de la línea en una Arquitectura de Referencia.
- Incluye elementos de retroalimentación y adaptación de los dominios definidos a partir de los productos desarrollados.

Limitaciones del modelo

- El proceso no contempla los principios de la calidad total.
- No se definen actividades relacionadas con la gestión del repositorio de activos de software reutilizables.
- No está identificado el proceso de soporte de los productos de la línea.
- No se incluyen en el modelo actividades relacionadas con la gestión de los costos.
- El modelo no especifica cómo realizar la gestión de los recursos humanos.
- El modelo no explica cómo realizar la gestión de los riesgos de la línea.

Proceso de desarrollo CAFE

El proyecto europeo CAFE (From Concepts to Application in System-Family Engineering) (ITEA 2009) es la continuación del proyecto ESAPS (Engineering Software Architectures, Processes and Platforms for System-Families) (ITEA 2001). Este modelo es una combinación de cuatro modelos diferentes:

- **Ciclo de vida de una organización que trabaja en el campo de las Tecnologías de la Información:** Ofrece una visión global de alto nivel de la organización, identificando dos actividades principales, el uso de la familia de productos (desarrollo, despliegue y mantenimiento) y el preuso de la familia de productos (planificación, análisis económico y delimitación).
- **Entorno de Procesos ISO/IEC 15504:** Clasifica todos los procesos de una entidad en cinco grupos: Cliente- Suministrador, Ingeniería, Organización, Gestión y Soporte.
- **Modelo de referencia CAFE (CAFE CRM):** Define seis grandes paquetes básicos de trabajo en una línea de productos.
- **Modelo de activos CAFE (CAFE ARM):** Clasifica los diferentes activos de la línea de productos según el modelo de referencia anterior.

Como se muestra en la Fig. 4 el modelo de referencia CAFÉ incluye las actividades de prueba de la línea de productos, tanto de la ingeniería del dominio como en la ingeniería de aplicaciones.(MELLADO 2004)

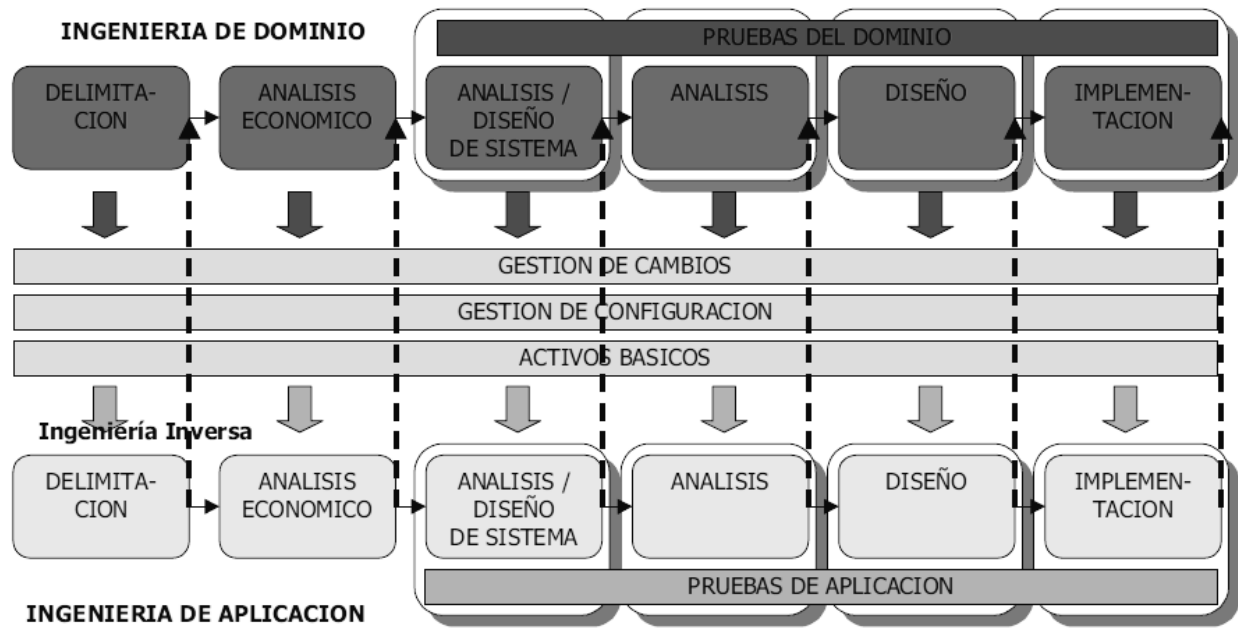


Fig. 4 Actividades de Ingeniería en el Modelo de Referencia CAFE.

El modelo de activos que aparece en la Tabla 3, asocia los distintos tipos de activos de software a cada una de las fases del modelo de referencia. Los artefactos de prueba pertenecen a las fases que son propiamente de desarrollo de software (análisis, diseño e implementación).

Tabla 3 Modelo de Activos Básicos Software proyecto CAFE.

| Delimitación | Análisis Económico | Análisis y Diseño de Sistema | Análisis | Diseño | Implementación |
|--|--------------------|---|------------------|--------------------------------------|----------------|
| Pruebas | | | | | |
| Gestión de Cambios | | | | | |
| Gestión de Configuración | | | | | |
| Documentación de Gestión (Estimaciones, Planificaciones) | | | | | |
| Diccionario | | | | | |
| | Plan de Negocio | Requisitos del Sistema | Otros Requisitos | Modelo de Datos | Clases |
| | | Plataforma de Hardware y Software | | Interfaces | Componentes |
| | | Arquitectura del Sistema | | Arquitectura de Software | |
| | | Artefactos de Prueba (Planes de Prueba, Casos de Prueba) | | | |
| | | Otros documentos (Económicos, marketing, desarrollo, formación) | | | |
| | | | | Librerías | |
| | | | | COTS | |
| | | | | Fuentes libres de distribución (GPL) | |
| | | | | Servicios Web | |

Análisis de ventajas

- Establece el proceso de pruebas tanto de los dominios como de las aplicaciones.
- Incluye procesos horizontales de la línea que abarcan la gestión de configuración, la gestión de cambios y la planificación.
- Define como actividad inicial la delimitación del alcance tanto en los procesos de Ingeniería de Dominio y como en los de Ingeniería de Aplicación.
- Incluye el análisis de factibilidad económica como paso previo a la Ingeniería de Dominio y de Aplicación.
- Establece claramente un conjunto de artefactos básicos en cada una de las fases del modelo.
- Gestiona el proceso de adquisiciones y contempla la relación cliente suministrador.

Limitaciones del modelo

- No define actividades que incluyan la gestión del repositorio de los activos generados.
- No incluye explícitamente elementos de retroalimentación a partir de los activos o productos desarrollados.
- El modelo no contempla todos los principios para el aseguramiento de la calidad total.
- El proceso de desarrollo no está orientado a la mejora continua de la organización.
- La gestión de los recursos humanos no se define de manera explícita en el modelo.

Modelo TWIN

El modelo de procesos de dos fases o modelo TWIN Life Cycle (SAMETINGER 1997) incorpora explícitamente la reutilización de software en el proceso de desarrollo de aplicaciones. La esencia de este modelo son los procesos de desarrollo de activos reutilizables y de desarrollo de aplicaciones basadas en la reutilización:

- Desarrollo de software para la reutilización: donde el propósito es producir componentes de software reutilizables. (Este proceso se denomina Ingeniería de Dominio)
- Desarrollo de software con reutilización: cuyo propósito es el desarrollar software reutilizando componentes existentes. (Este proceso recibe el nombre de Ingeniería de Aplicación)

Tanto la Ingeniería de Dominios como la Ingeniería de Aplicación proponen métodos que implican la reutilización, ya sea de activos reutilizables o de componentes reutilizables. Los procesos del modelo TWIN se representan a continuación en la Fig. 5.

- La Ingeniería de Dominio se dirige a la creación sistemática de modelos de dominios, arquitecturas, componentes y activos de software reutilizables. También soporta la Ingeniería de Aplicaciones, la cual utiliza los modelos y las arquitecturas para construir sistemas.
- La Ingeniería de Aplicación es la parte de la Ingeniería de Software que se basa en la reutilización de componentes existentes y en el conocimiento del dominio. Las aplicaciones son construidas mediante el ensamblaje de componentes.

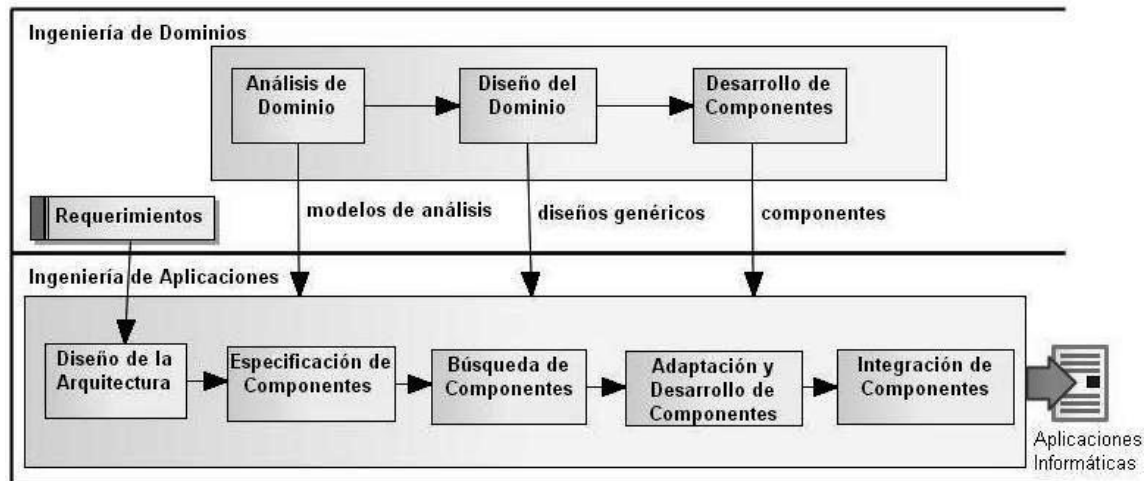


Fig. 5 Modelo de procesos del modelo TWIN.

Análisis de ventajas

- Propone un sistema de gestión Activos de Software Reutilizables.
- Identifica claramente los procesos de ingeniería de dominio y de ingeniería de aplicaciones.
- Incluye actividades relacionadas con el aseguramiento de la calidad.

Limitaciones del modelo

- No especifica las actividades relacionadas con la gerencia del proceso.
- El modelo no incluye componentes relacionados con el soporte y mantenimiento de los productos liberados.
- A pesar de contemplar el aseguramiento de la calidad, el modelo no cumple con todos los principios de calidad total.
- No incluye explícitamente actividades relacionadas con la gestión de los recursos humanos.
- La gestión de riesgos en la línea no es parte esencial del modelo TWIN.
- En el modelo no aparecen actividades vinculadas explícitamente con la gestión de los costos.

1.3.1. Modelos de Suramérica

Modelo WATCH

Surge en la Universidad de Los Andes (Venezuela) para el desarrollo de aplicaciones empresariales y como medio instruccional empleado en la enseñanza de los procesos de la Ingeniería de Dominios y de la Ingeniería de Aplicaciones (MONTILVA, J. *et al.* 2000). Es un modelo de procesos orientado al desarrollo de proyectos de software de pequeño o mediano tamaño. Integra aspectos de los métodos: modelo espiral, desarrollo incremental y por versiones y del método Orientado a Objetos (BRUEGGE and DUTOIT 2002). El modelo está estructurado en 4 grupos de procesos, ver Tabla 4.

Tabla 4 Procesos propuestos por el modelo WATCH.

| Ingeniería de Métodos | Procesos Gerenciales |
|---|-------------------------|
| Definición del Proyecto | Iniciación del proyecto |
| Adaptación y conversión del modelo de procesos en un método | Gerencia del Proyecto |
| | Gestión de la Calidad |

| | |
|--|---|
| Documentación del método de desarrollo | Gestión de la Configuración Verificación y Validación Adiestramiento Documentación |
| Procesos de Desarrollo | Procesos de Post-desarrollo |
| Análisis del Dominio de Aplicación Descubrimiento de Requerimientos Análisis y Especificación de Requerimientos Diseño del Sistema Diseño de Componentes Implementación del Sistema Pruebas del Sistema Entrega del Sistema | Operación y soporte Mantenimiento Retiro |

Análisis de ventajas

- El modelo integra los procesos gerenciales y de desarrollo.
- Emplea el UML (Unified Modeling Language) como lenguaje de modelado.
- Utiliza el estándar IEEE 1074 para crear la estructura de procesos del modelo.
- Incluye procesos para el mantenimiento y soporte post-desarrollo de los productos.
- El modelo es simple, completo y además es adaptable.

Limitaciones del modelo

- No se definen en el modelo actividades encaminadas a la gestión de los riesgos.
- No están definidas en el modelo actividades relacionadas con la mejora continua de la organización.
- El proceso de desarrollo aplica actividades relacionadas con el aseguramiento de la calidad, pero no los principios de calidad total.
- No contempla explícitamente actividades de reutilización de los componentes desarrollados.
- La gestión de los recursos humanos está enfocada solo el adiestramiento de la fuerza laborar.
- La gestión de los costos no está identificada como parte del modelo.

Modelo WATCH Component

Para satisfacer la necesidad de tener un modelo que proporcionara mayor detalle del proceso y que a su vez tomara en cuenta la reutilización de componentes existentes, se creó esta variación del modelo WATCH, basada en los principios que este método posee y en su facilidad para integrar diferentes aspectos metodológicos del desarrollo de software basado en componentes. Básicamente el modelo cubre el ciclo de vida de un componente, tomando en cuenta la utilización de componentes existentes y es visto como un elemento metodológico que se puede integrar al modelo WATCH original.(HAMAR 2003) Este método basado en componentes permite planear, organizar, controlar y desarrollar aplicaciones y se describe mediante tres modelos metodológicos, el modelo de producto, el modelo de procesos y el modelo del grupo de desarrollo.(MONTILVA, J. *et al.* 2004)

Modelo de Productos: representa los conceptos asociados y formas en las que pueda tener el concepto para el cual se está desarrollando el método. En este caso, se representan los conceptos asociados a un componente de software en cada una de sus etapas o formas.

Modelo de Procesos: representa y define los procesos o actividades que debe seguir el grupo de desarrollo de componentes, guía tanto los procesos gerenciales como los de desarrollo del ciclo de vida de un componente, ver Fig. 6.

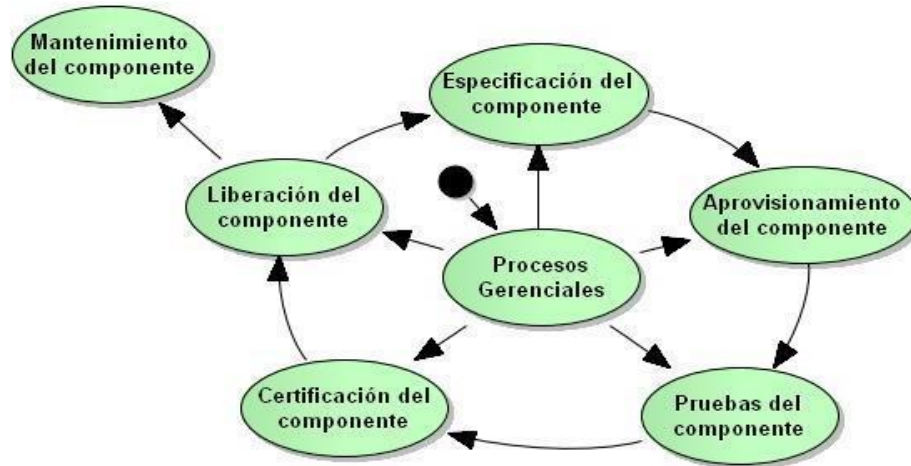


Fig. 6 El modelo de procesos de WATCH-Component.

Modelo de Grupo de Desarrollo, se hace con el fin de representar los equipos de personas involucradas en el desarrollo de los componentes, ver Fig. 7.



Fig. 7 El modelo de grupo de desarrollo de WATCH-Component.

Análisis de ventajas

- Como parte del modelo se propone un componente relacionado con la organización del grupo de personas involucradas en el desarrollo.
- Incluye actividades relacionadas con el aseguramiento de la calidad y la liberación de los productos.
- Su estructura permite al grupo de desarrollo, adaptar el modelo a las características particulares del producto.
- Define un repositorio donde gestionar productos y activos reutilizables de la línea.

Limitaciones del modelo

- Es un método desarrollado formalmente para producir componentes de software reutilizable.
- A pesar de que contempla el aseguramiento de la calidad no cumple con los principios de la calidad total.
- No contempla elementos relacionados con gestión de los riesgos de la línea de productos.
- La gestión de los costos no está definida dentro del modelo.
- La gestión del alcance no es parte del modelo WATCH-Component.

1.3.2. Modelos de Norteamérica

Modelo ESPLEP

El modelo ESPLEP (Evolutionary Software Product Line Engineering Process) es un proceso de desarrollo de software que elimina la distinción tradicional entre el desarrollo y mantenimiento de software. Con este modelo los productos evolucionan a través de varias iteraciones, por lo tanto los sistemas desarrollados con este enfoque son capaces de adaptarse a los cambios en los requisitos en cada iteración. Además, como los nuevos productos son a menudo evoluciones de los ya existentes o están conformados por partes reutilizables de estos, el modelo ESPLEP permite el desarrollo de las familias de productos de software, ver Fig. 8.(CATAL 2009)

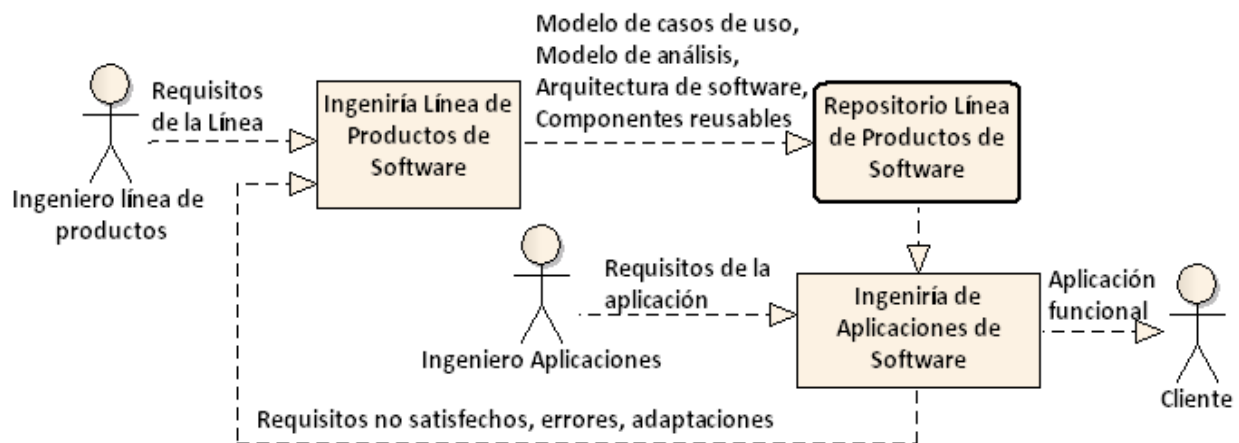


Fig. 8 Modelo de Líneas de Productos de Software ESPLEP.

El modelo ESPLEP es un proceso de desarrollo de software altamente interactivo basado en el concepto de casos de uso. Las fases del proceso ESPLEP se muestran en la Fig. 9.

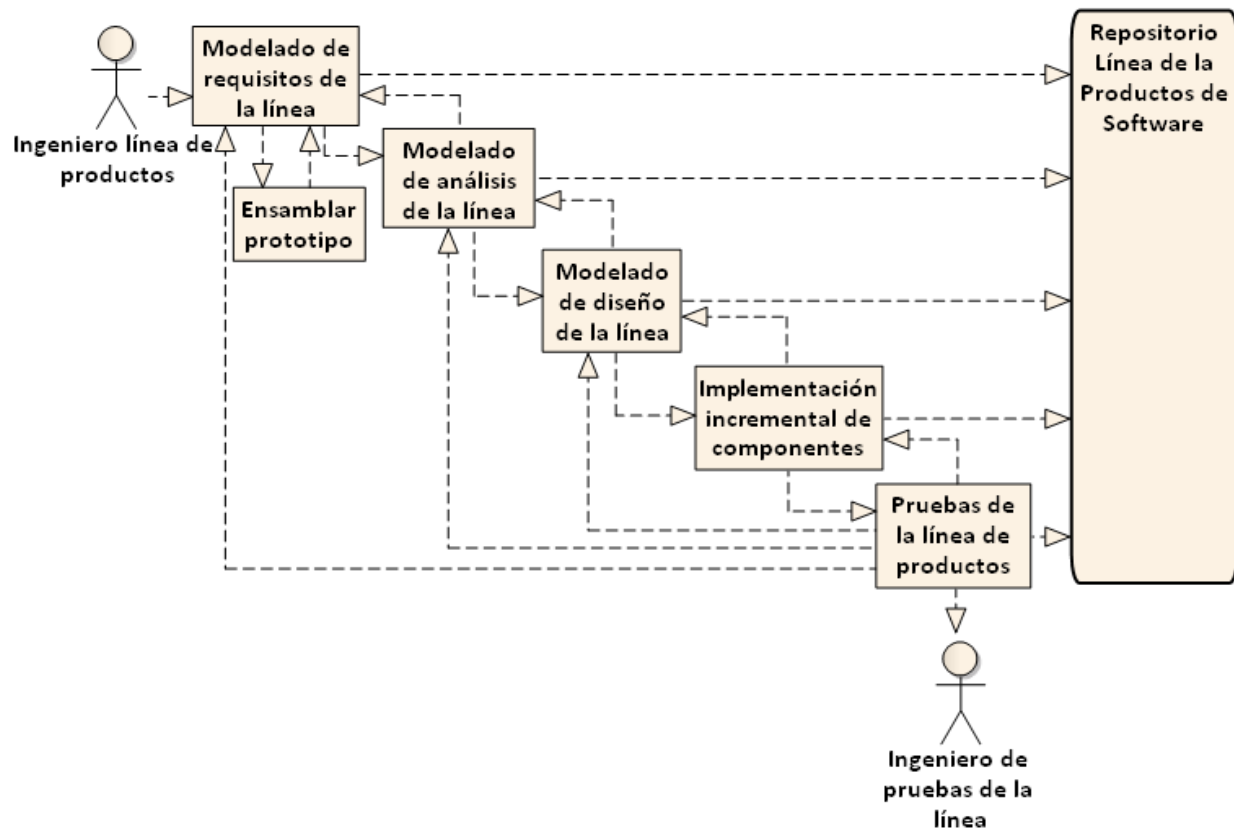


Fig. 9 Fases del modelo ESPLEP.

Durante el modelado de requisitos, los requisitos funcionales de la línea de productos se definen en términos de actores y casos de uso. Durante el modelado del análisis, cada caso de uso se describe en función de los objetos que intervienen en el caso de uso y sus interacciones. Durante el modelado del diseño, se desarrollan los componentes de la arquitectura de software de la línea de productos. Todos los artefactos producidos durante cada fase se almacenan en el repositorio de software de la línea de productos.

Análisis de ventajas

- El modelo promueve la utilización de un repositorio de productos donde se gestionan todos los activos generados en el proceso de desarrollo de software.
- El modelo utiliza la notación estándar UML para representar muchos de los artefactos generados en el desarrollo de software (actores, casos de uso, etc.).
- El modelo propone un ciclo de vida de los productos que permite su evolución, adaptándose a los cambios en los requisitos en cada iteración.
- Dentro de las actividades que propone el modelo se incluye el aseguramiento de la calidad.

Limitaciones del modelo

- No contempla actividades relacionadas con la mejora continua del modelo y la retroalimentación del proceso.
- No abarca las áreas relacionadas con la gestión del alcance de la línea y la gestión de los costos.

- El modelo a pesar de contemplar actividades relacionadas con el aseguramiento de la calidad no cumple con los principios de calidad total.
- No contempla el modelo elementos relacionados con gestión de los riesgos de la línea de productos.
- No incluye actividades relacionadas con la gestión de los recursos humanos.

Modelo propuesto por el SEI

Las líneas de productos, según el SEI (Software Engineering Institute) se apoyan sobre tres pilares: el desarrollo de activos de software, el desarrollo de productos y la gestión de la línea de productos.(CLEMENTS, PAUL C. *et al.* 2005)

Gestión de la Línea: Tiene como objetivo principal proporcionar los recursos, coordinar y supervisar el desarrollo de activos y productos. Para garantizar un exitoso desempeño de la línea de productos, más allá de una mera coordinación se requiere de una gestión fuerte y con una larga visión. Esta debe garantizar también, que se inviertan recursos en el desarrollo y el mantenimiento de los activos de software básicos y se fomenten los cambios culturales necesarios para ver los principales activos en el contexto de los nuevos productos.

Desarrollo de Activos de Software (Ingeniería de Dominio): Tiene como objetivo construir los activos de software de la línea a partir de las entradas relacionadas a continuación.

Entradas:

- Restricciones del producto: Definen cuanto tienen los productos en común y cuanto puede variar en ellos.
- Restricciones del desarrollo: Definen el marco de desarrollo sobre el cual se trabajará en la línea.
- Estrategia de producción: Precisa los hitos a cumplir en el desarrollo activos y productos.
- Inventario de activos pre-existentes: Especifica que sistemas legados, componentes o artefactos están disponibles a partir de desarrollos anteriores.

Salidas

- Alcance de la línea: Describe los productos existentes o potenciales de la línea y por implicación los que excluye. Indica lo que todos los productos tienen en común y la forma en que varían.
- Activos de Software: Son la base para la producción en la línea y puede ser cualquier artefacto generado en el proceso (La arquitectura de software, componentes, especificaciones de requisitos, planes de prueba o artefactos de gestión como los cronogramas o los presupuestos).
- Plan de producción: Describe paso a paso cómo los productos se fabrican utilizando los activos de software. Suelen variar dependiendo del nivel de organización y de automatización en el proceso de producción.

Desarrollo de Productos (Ingeniería de Aplicaciones): Tiene como objetivo elaborar los productos de la línea a partir del ensamblaje de activos de software fundamentalmente, siguiendo el plan de producción y las restricciones establecidas. Al igual que la ingeniería de dominio tiene varias iteraciones donde suele haber retroalimentación, porque el diseño de un nuevo producto puede ser similar a un producto existente lo que implica una búsqueda en los activos existentes cada vez que se desarrolla un nuevo producto.

Entradas:

- Los requisitos del producto concreto, se expresan a menudo como una descripción genérica del producto.
- Un proyecto de desarrollo de productos que incluye los resultados de un proyecto de desarrollo de activos (que pueden constituir sub-proyectos).
- El alcance.
- Los activos de software.
- El plan de producción.

Salidas:

- Productos

Además de las actividades esenciales anteriormente explicadas, existen tres áreas prácticas (Un área de práctica es un conjunto de actividades que una organización debe dominar para llevar a buen término la labor esencial de una línea de productos). El área de *Ingeniería de Software* incluye las actividades relacionadas con la ingeniería de dominio y de aplicaciones. El proceso de gestión de la línea se divide en dos áreas prácticas: la *Gestión Técnica*, orientada a los grupos que desarrollan activos y productos y la *Gestión Organizacional*, orientada a los aspectos organizacionales (estructura, relaciones, recursos, financiamiento, etc.) ver Tabla 5.

Tabla 5 Principales tareas que se incluyen en las áreas de procesos.

| Ingeniería de Software | Gestión Técnica | Gestión Organizacional |
|---------------------------------------|-----------------------------|--|
| Definición de la arquitectura | Gestión de la configuración | Construcción de casos de uso del negocio |
| Evaluación de la arquitectura | Análisis de factibilidad | Gestión de clientes |
| Desarrollo de componentes | Seguimiento y control | Gestión de adquisición |
| Minería de activos existentes | Definición del proceso | Gestión de costos |
| Ingeniería de requisitos | Gestión del alcance | Implantación y soporte |
| Integración de sistemas | Planificación técnica | Análisis de mercado |
| Pruebas | Gestión de riesgos técnicos | Operaciones |
| Comprensión de Dominios relacionados | Herramientas de soporte | Planificación de la organización |
| Reutilización de software de terceros | | Gestión de riesgos de la organización |
| | | Estructuración de la organización |
| | | Observatorio tecnológico |
| | | Entrenamiento |

Análisis de ventajas

- El modelo propuesto por el SEI abarca actividades relacionadas con la planificación, y el análisis de factibilidad de los productos desarrollados.
- La gestión de costos está incluida dentro de las actividades del modelo.
- Los riesgos se gestionan a nivel de la organización y a nivel técnico en los productos y activos desarrollados.

- Se gestionan los clientes y las adquisiciones de la línea.
- Se incluyen procesos de soporte y mantenimiento de los productos.
- Está soportando por una plataforma de trabajo bien documentada que cuenta además con un conjunto patrones y buenas prácticas para la implantación del modelo, define doce patrones y once variantes.

Limitaciones del modelo

- Requieren para su implantación un esfuerzo considerable de la organización por la complejidad y alcance del modelo.

1.4. *Arquitectura de Empresas, visión general*

Hoy en día las organizaciones se enfrentan a un entorno complejo y cambiante, donde la necesidad de mantener los niveles de competitividad es vital para poder rivalizar en el mercado internacional. En este contexto se hace necesario establecer mecanismos que permitan coordinar las decisiones para obtener la ventaja necesaria y lograr la integración empresarial con los modelos de producción. Para entender esta visión completa de la organización en todas sus dimensiones, surge el concepto de Arquitectura de Empresa. Que es básicamente la identificación de los componentes principales de la organización y su relación en función de conseguir los objetivos de la entidad, integrando aspectos de planificación, aspectos de ejecución, aspectos tecnológicos y otros. Una característica o principio específico de la Arquitectura de Empresa es la definición de sus vistas (MARTIN and ROBERTSON 2004), elemento que ayuda a comprender la complejidad de una empresa bajo varias perspectivas. Normalmente no hay una única vista de la arquitectura de empresa, sino que ésta se definirá en función de las vistas que la componen y el estado en que se encuentre la organización. Entre las variantes de arquitecturas o vistas de una arquitectura de empresa destacan las siguientes:

TOGAF (The Open Group Architecture Framework)

Desarrollada por los miembros del Open Group en 1995 es un método y un conjunto de herramientas para desarrollar arquitecturas de empresa (OPEN-GROUP 2010). TOGAF incluye la vista de negocio como un requisito previo desde cualquier punto de vista (datos, aplicaciones, tecnología) y es la primera que debe ser completada. Contempla la arquitectura de datos que tiene como objetivo definir los principales tipos y fuentes de datos necesarios para dar soporte a las actividades de la empresa, de manera que sea entendible, completa, consistente y estable. El modelo incluye además la vista tecnológica y la vista de aplicación que define qué clase de aplicaciones son relevantes para la empresa y lo que estas aplicaciones necesitan para gestionar los datos y presentar la información.

IE-GIP (Integración Empresarial-Gestión Integrada de Procesos)

Es una propuesta para el desarrollo de Programas de Integración Empresarial, llevada a cabo en el Departamento de Organización de Empresas de Universidad Politécnica de Valencia (ORTIZ et al. 1999). Presenta una metodología y una arquitectura que cubre el ciclo de vida de un programa de integración empresarial (nivel de negocio, nivel de modelos, nivel de implementación y nivel de operación), en esta propuesta se incluye la vista de función que puede corresponderse con la vista de negocio; la vista de

datos que permite la representación y modificación de la información de la empresa y no define una arquitectura tecnológica porque incluye estos términos en una vista de recursos.

Zachman Framework para arquitecturas de empresas

Fue publicado en 1987 por John Zachman. Proporciona la arquitectura, para la infraestructura de la información de una organización, el marco contiene una matriz de seis filas por seis columnas que componen treinta y seis aspectos. No tiene un flujo establecido para las secuencias o proceso y su objetivo es asegurarse de que todos los aspectos de la empresa estén cubiertos. Incluye la vista de negocio donde define la dirección de la empresa y el propósito de su negocio (Alcance); la vista de datos que define las funciones de negocio en términos de información; Define además una vista tecnológica y una vista de la organización que indica las relaciones dentro de la empresa, indicando los roles y responsabilidades (ZACHMAN 2003).

Standard ISO/DIS 1939(Enterprise Integration-Framework for Enterprise Modeling)

El modelo definido como estándar, se estructura de acuerdo a tres dimensiones, siendo una de ellas la dimensión de las vistas, en la que se incluye la vista de función, que permite la representación y modificación de los procesos de la empresa, su funcionalidad y comportamiento; correspondiéndose con la vista de negocio. Además define la vista de datos que representa la información de la empresa y no define una vista tecnológica pero incluye algunos elementos en una vista de recursos.

Se establece una comparación entre los distintos modelos de arquitecturas de empresa bajo la perspectiva de las vistas que los componen, ver Tabla 6. (GONZÁLEZ *et al.* 2005)

Tabla 6 Análisis de los modelos de AE.

| | TOGAF | IE-GIP | ZACHMAN | ISO/DIS 19439 |
|------------------------|--------------|---------------|----------------|----------------------|
| Vista de negocio | +++ | ++ | ++ | ++ |
| Vista de Información | ++ | +++ | ++ | +++ |
| Vista de datos | +++ | + | + | + |
| Vista de aplicación | +++ | + | + | + |
| Vista tecnológica | +++ | + | ++ | + |
| Vista recursos humanos | + | +++ | ++ | +++ |

+++ : Indica que la vista está incluida con el mismo nombre en el modelo analizado.

++ : Indica que la vista está incluida pero con distinto nombre en el modelo analizado.

+ : Indica que la vista no está incluida de manera explícita pero puede ser completada a partir de otras.

Conclusiones del capítulo

A partir del estudio del estado del arte del ciclo de vida de las Líneas de Producto Software en diferentes metodologías podemos arribar a las siguientes conclusiones:

- Se cumplió el primer objetivo específico, evidenciando la necesidad de elaborar un modelo basado en la reutilización, capaz de ser generalizable en diferentes escenarios relacionados con el desarrollo de software y capaz de potenciar la productividad.
- Como entrada básica de cada uno de los modelos estudiados están los requisitos y como resultado del proceso de desarrollo de productos además de los propios productos, se elabora toda la infraestructura de reutilización de software que va a permitir obtenerlos con menos esfuerzo.
- La fase de análisis del dominio es muy importante en el desarrollo de líneas de productos, puesto que en ella se definen los aspectos comunes a todos los miembros de la familia y los particulares de cada uno.
- Hay que definir los activos básicos de la línea de producto (ingeniería del dominio) y la forma de instanciar cada producto particular a partir de dichos activos básicos (ingeniería de aplicación).

Del análisis de algunos de los métodos existentes para organizar el desarrollo de una Línea de Productos de Software se pueden resumir algunas ventajas y limitaciones.

Ventajas de los modelos:

- Muchos modelos apuestan por el uso de notaciones estándar como UML (Unified Modeling Language) y las utilizan para representar muchos de los artefactos generados en el proceso de desarrollo de software (actores, casos de uso, etc.).
- Están bien definidas en casi todos los modelos las áreas de procesos relacionadas con Ingeniería de Dominio y la Ingeniería de Aplicaciones.
- Se establecen en la mayoría de los modelos un conjunto de decisiones del diseño de los componentes y productos a través de las Arquitecturas de Referencia.
- Algunos modelos establecen un conjunto de artefactos básicos para cada una de las fases del proceso.
- Muchos modelos están soportados por una amplia documentación y algunos como el propuesto por el SEI cuenta además con un conjunto de patrones y buenas prácticas para su implantación.

Limitaciones de los modelos:

- No especifican en muchos casos procesos relacionados con la gestión de los repositorios de activos de software reutilizables.
- Los modelos introducen elementos de soporte en muchos casos, pero no tienen bien definidas las actividades necesarias para dar seguimiento a los productos desplegados.
- La mayoría de los modelos evalúa la calidad tanto del proceso como de los resultados obtenidos, pero no contemplan como principio la Calidad Total.
- La gestión de los recursos humanos se limita en muchos casos a controlar un área y no se enfocan en la gestión de los recursos humanos de manera integral.

- La gestión de los recursos materiales y las adquisiciones no están contempladas en muchos modelos.
- No se incluyen explícitamente en los modelos actividades relacionadas con la gestión de los costos.
- No contemplan en algunos casos actividades relacionadas con la mejora continua del modelo y la retroalimentación del proceso.
- Muchos modelos no contemplan explícitamente actividades relacionadas con la gestión de los riesgos.
- Requieren para su implantación un esfuerzo considerable de la organización por la complejidad y alcance del modelo.

Del análisis de los modelos para organizar la arquitectura de empresas se pueden concluir:

- En los modelos estudiados existen diferencias en la sintaxis utilizada para llamar cada vista: como el caso entre la arquitectura de negocio, la arquitectura operacional y la vista de función; o como el caso entre la arquitectura de aplicación y la de sistemas.
- De los modelos estudiados se han seleccionado cinco vistas por su presencia de una u otra forma en las diferentes arquitecturas: la vista de negocio, la vista de datos, la vista de aplicación, la vista tecnológica y la vista de recursos humanos.

CAPÍTULO 2 DESARROLLO DEL MODELO PARA LÍNEAS DE PRODUCTOS DE SOFTWARE

En este capítulo se presenta el modelo de desarrollo para líneas de productos de software en centros de producción, a partir de los elementos analizados en el estudio del estado del arte. El modelo propuesto se ha organizado siguiendo los principios de Arquitectura de Empresas (EA), las líneas de productos de software y los lineamientos relacionados con el Programa de Mejora de la Universidad. Se enfoca en organizar y mejorar el proceso productivo de la organización, de forma tal que se potencie la reutilización y el incremento de la productividad. Para facilitar su introducción se definen una serie de pasos para implantar el modelo y se recomiendan algunas técnicas y herramientas. También se proponen los procesos básicos que deberían implementarse en cada una de las áreas y cómo deberían establecerse los modelos de transición para garantizar la evolución de la organización en el tiempo.

2.1. Principios generales del modelo

2.1.1. Modelo centrado en arquitectura de empresas

El modelo propuesto está basado en los principios de la Arquitectura de Empresa y se presenta dividido en cinco vistas (Negocio, Datos, Aplicaciones, Tecnología y Recursos Humanos) para facilitar la flexibilidad en los cambios del negocio y los sistemas, de manera coherente con la implantación y el desarrollo de líneas de productos de software y soportando además la mejora continua, ver Fig. 10. Por su importancia la vista relacionada con los recursos humanos, se desarrolló en una investigación independiente (TORRES 2010).

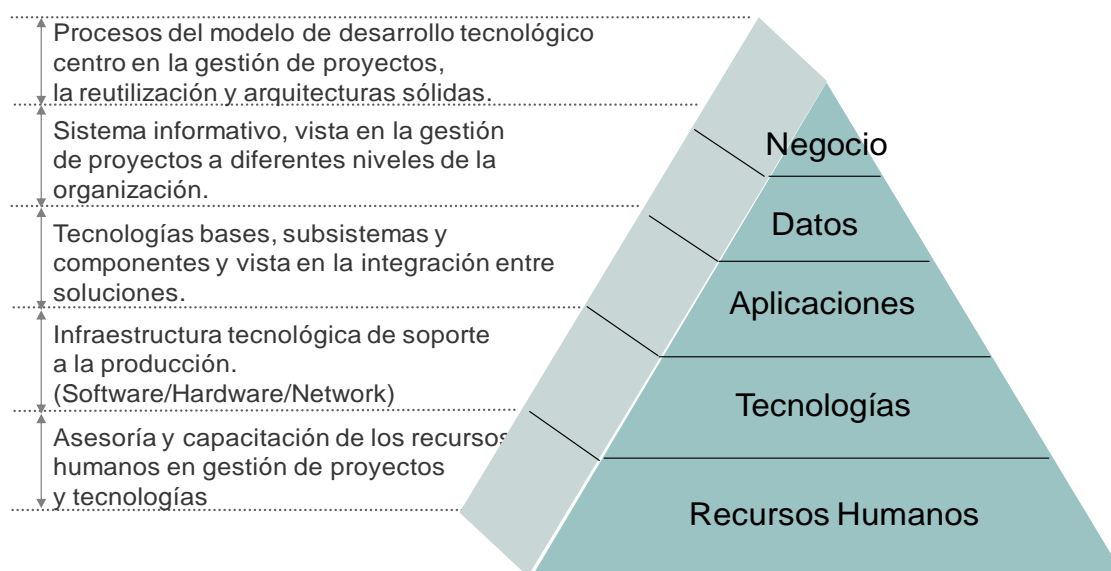


Fig. 10 Vista de desarrollo integral basado en los principios de la arquitectura de empresas.

Todo el modelo está basado en los principios de mejora continua y en este sentido se identifican claramente cuatro momentos que se repiten constantemente en forma de iteraciones representando los ciclos evolutivos de la organización. En cada ciclo se identifican los cuatro procesos siguientes, ver Fig. 11:

1. Diagnóstico del estado actual de la organización: Representa el estado de cada una de las variables de la organización en un momento específico y se repite al inicio de cada ciclo de mejora del proceso. Por ejemplo cada año puede hacerse un balance del estado de la organización respecto a las cinco variables para proponer los elementos a mejorar en la siguiente transición.
2. Diseño del modelo ideal a alcanzar: Representa el estado al que se quiere llegar en función de los objetivos estratégicos de la organización. En cada período se define un estado ideal alcanzable al final del mismo, por ejemplo se define respecto a las cinco variables que elementos se pueden lograr al final de cada año en ejercicio de planeación estratégica.
3. Diseño de los modelos de transición: Los estados de transición que identifican el camino hasta el estado ideal (Representan estados intermedios a alcanzar y se recomienda que sean trimestrales). Las transiciones indican qué elementos tangibles pueden ser mejorados cubriendo las cinco vistas.
4. Conducir, monitorear y controlar los estados de transición: El estado de la conducción del proceso de transición se basa en el control interno de la organización y es realizado por la dirección de la línea de productos. Proporciona una seguridad razonable en el cumplimiento de los objetivos e incluye las siguientes actividades: efectividad y eficiencia de las operaciones, confiabilidad de la información financiera y cumplimiento de las normas y regulaciones. Se apoya en un sistema de medidas e indicadores del estado de los objetivos y actividades, pero sólo es efectivo si la información fluye de forma pertinente, clara y precisa a través de la organización, facilitando las correcciones y acciones correctivas pertinentes.

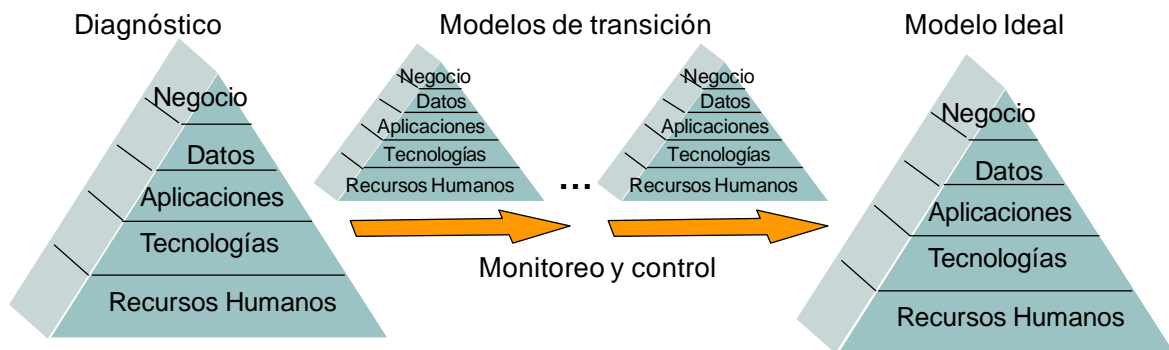


Fig. 11 Vista general de los principios de la arquitectura de empresas.

2.1.2. Modelo centrado en la mejora continua

Es importante resaltar que la mejora continua no es un paso sino la aproximación al modelo ideal por iteraciones, donde se repiten cada uno de los pasos partiendo de lo aprendido en la iteración anterior. Estos ciclos se repiten de forma permanente durante el tiempo de vida de la organización y son la esencia de la mejora continua, ver Fig. 12. La responsabilidad de guiar la mejora continua, es en primer lugar de la alta gerencia de la organización, que deberá estar alineada además con elementos del control interno y la medición constante del proceso.

Se recomienda particularmente para los centro de producción de la UCI, hacer coincidir los ciclos de transición con el año escolar, para poder vincular a estudiantes y profesores al ciclo profesional del

modelo de formación. La iteración parte de la evaluación de la organización al principio del mismo (Diagnóstico situación actual), se define el estado esperado de la organización al final del curso (Modelo ideal a alcanzar), se definen modelos concretos de transición a alcanzar con frecuencia trimestral o mensual y se definen acciones concretas a realizar para lograr la mejora de la organización.

Otro variante sería aplicarlo en períodos de tiempo más largos entre los tres y cinco años, dependiendo de la organización. Se podría hacer coincidir un ciclo con el tiempo estimado para alcanzar la visión estratégica de la organización, se realiza una evaluación de la organización al principio del mismo (Diagnóstico situación actual), se define el estado esperado de la organización al final del período (Modelo ideal a alcanzar) y se definen modelos concretos de transición a alcanzar con frecuencia anual o semestral que determinan los objetivos por año en la evolución de la organización.

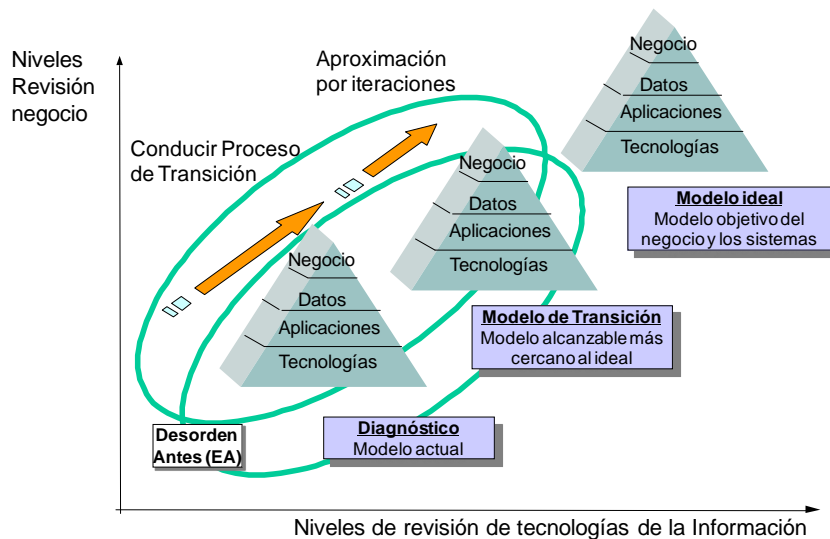


Fig. 12 Representación de los ciclos de mejora dentro de la organización.

2.1.3. Modelo centrado en líneas de productos de software

El modelo propuesto define cinco elementos fundamentales que se revisan en cada una de las vistas anteriormente definidas.

- Dominio: Se refiere al área de aplicación de productos de software del entorno de negocio donde se despliega la línea de productos y su posible integración a ecosistemas de software.
- Familia de productos: Define el conjunto de software asociados a un dominio determinado. Los miembros de la familia comparten aspectos comunes tales como:
 - Diseño arquitectónico común.
 - Conjunto componentes reutilizables.
 - Capacidades y servicios comunes.
 - Tecnologías comunes.
- Modo de producción: Se refiere a la forma de organización del proceso productivo de la línea.
- Arquitectura: Incluye toda la información referente a la línea base de la arquitectura de las soluciones, organizadas en nueve capas (los Procesos, Presentación, Sistema, Datos, Integración, Seguridad, Tecnología, Infraestructura, Despliegue, los Roles y las Responsabilidades).

- Activos de software: Son todos los artefactos y componentes que se producen en la línea con el propósito de ser reutilizados múltiples veces en el desarrollo de diferentes aplicaciones.

2.2. Pasos fundamentales para la aplicación del modelo

A continuación se describe en forma de guía un grupo de pasos generales para implantar el modelo propuesto, indicando en algunos casos técnicas y herramientas que se pueden utilizar.

2.2.1. Paso 1: Diagnóstico del estado actual de la organización

Básicamente consiste en introducir guías de diagnóstico y los diagramas apropiados para cada capa de la arquitectura de empresas de acuerdo a los elementos fundamentales relacionados con las líneas de productos de software. El objetivo de esta etapa es tener una visión del estado actual y lograr una comprensión completa de la organización por parte del equipo de dirección de la línea de productos.

Paso 1.1: Desarrollar diagnóstico de la vista de negocio.

Lista de chequeo para el diagnóstico:

Dominio

- ¿Cuál es el dominio de aplicaciones que forma parte del objetivo de la organización?

Familia de productos

- ¿Qué productos posee la organización?

Modo de producción

- ¿Cuáles son las entidades que conforman el modo de producción actual?
- ¿Cuáles son las relaciones entre las entidades que conforman el modo de producción actual, representadas como procesos?
- ¿Cuáles son los roles y las responsabilidades que están definidos en la organización?

Técnicas y herramientas:

- DMM (Diamante de Mandala) Diagrama basado en el concepto del análisis estructurado que describe las funciones lógicas en el negocio a través de una tabla de tres columnas y tres filas, ver Anexo 1.
- DFD (Diagramas de flujo de datos) Basado en el análisis estructurado, representa complementemente las funcionalidades del negocio desde la perspectiva de funciones lógicas y del intercambio de información (Procesos, Datos almacenados, Entidades externas, Flujo de datos), ver Anexo 2.
- WFA (Diagramas de flujo de trabajo) Representa completamente el flujo del negocio de una organización a través de la comunicación entre los siguientes elementos, ver Anexo 3.
 - Secciones de la organización y personal
 - Operaciones
 - Información que se mueve en cada operación
 - Flujo de información
- IDEF0 (Diagrama IDEF0 descripción de procesos) Diagrama para la modelación de procesos a través de los siguientes elementos: Actividades, Entradas, Salidas, Control y mecanismos regulatorios, ejecutores del proceso, ver Anexo 4.

- Estudios de mercado: Proporciona información relevante sobre los clientes potenciales de los productos, la competencia y las prácticas generales en el sector que se estudia.

Paso 1.2: Desarrollar diagnóstico de la vista de datos.

Comprende la asimilación de todo el sistema de información necesaria para el negocio, la modelación de los datos de entrada y salida en la estructura organizativa deseada, por ejemplo el expediente del proyecto.

Lista de chequeo para el diagnóstico:

Dominio

- ¿Qué información se guarda sobre estudios de mercado en el dominio específico de la organización?

Modo de producción

- ¿Qué tipo de información relevante se almacena sobre el modo de producción actual de la organización en el expediente de proyecto?
- ¿Qué información referente a las medidas y métricas del proceso de producción se almacenan?

Arquitectura

- ¿Qué información referente a la línea base de la arquitectura se almacena?

Activos de software

- ¿Qué activos de software reutilizables se almacenan?
- ¿Qué metadatos se registran de los activos de software?
- ¿Cómo se almacenan los activos de software?

Técnicas y herramientas:

- Herramientas para la presentación de la información
 - ERD (Diagramas entidad relación) Es el mismo diagrama utilizado en el diseño de bases de datos, representa la información utilizada en la gestión de la empresa a partir de la definición de las Entidades, Relaciones, Cardinalidades y Atributos, ver Anexo 5.
 - IDEF1X (Modelo de datos IDEF) Representan la información utilizada en la gestión de la organización siguiendo un enfoque de entidades y las relaciones, ver Anexo 6.
- Listas de chequeo de elementos requeridos.

Paso 1.3: Desarrollar diagnóstico de la vista de aplicaciones.

Identificar todo lo relacionado con las funciones de la organización, la misión en función de los productos y servicios de cara a los clientes. Se debe realizar un análisis atendiendo a las posibilidades de las tecnologías expresado a través de las funciones de la empresa.

Lista de chequeo para el diagnóstico:

Dominio

- Identificación de la integración de tecnologías actuales de la organización con los ecosistemas en los que se integra la solución.

Familia de productos

- ¿Qué productos desarrolla la organización?
- ¿Qué servicios brinda la organización?

Arquitectura

- ¿Qué herramientas se utilizan como soporte del desarrollo del dominio?
- ¿Cuáles son las herramientas que conforman el entorno de desarrollo de los productos de la organización?
- ¿Qué herramientas se utilizan como soporte del desarrollo de los productos?

Activos de software

- ¿Qué activos reutilizables se generan en la organización?

Técnicas y herramientas:

- Diagrama de interrelaciones entre activos del sistema de Información. Describe las interrelaciones entre estos activos vista desde la reutilización, ver Anexo 7.
- Catálogo de herramientas utilizadas para el desarrollo tecnológico.

Paso 1.4: Desarrollar diagnóstico de la vista de tecnologías.

Explorar la estructura de las tecnologías para el soporte a los proyectos (software, hardware y redes).

Lista de chequeo para el diagnóstico:

Arquitectura:

- ¿Qué tecnologías de redes posee la organización y como está organizada esta?
- ¿Qué tecnologías de hardware tiene instalada la organización?
- ¿Qué tecnologías de software base tiene la organización?

Activos de software:

- ¿Existe algún repositorio para los activos reutilizables?

Técnicas y herramientas:

- Diagrama de configuración de redes. Representa la configuración de las tecnologías de red que soportan al sistema de trabajo incluyendo dispositivos de la red y las conexiones entre estos elementos.
- Diagrama de configuración de software base. Representa la configuración de los sistemas de información y la interacción entre ellos, basa su modelación en los siguientes elementos: productos de software, conexión e interrelaciones entre los sistemas.
- Diagrama de configuración de Hardware. Representa la configuración de los componentes de hardware y la interacción entre ellos. Baza su modelación en los siguientes elementos: Productos de hardware y conexión e interrelaciones entre los productos de hardware.

2.2.2. Paso 2: Definición del modelo ideal a alcanzar en la organización.

El modelo ideal se basa en los tres principios explicados en la sección 2.1, sobre los cuales estará sustentada la organización de la línea de productos. Para una mejor comprensión del modelo se detallan a continuación una serie de guías para la implantación según las vistas de la arquitectura de empresas, cada uno de los elementos del modelo y el proceso de mejora.

Paso 2.1: Definición del modelo desde la vista de negocio

Definición del dominio de aplicaciones:

Parte de la caracterización del dominio y del mercado potencial, además describe la necesidad y el objetivo del montaje de la línea de producción, la factibilidad técnico-comercial y la sostenibilidad en el tiempo. Para ello se definen los siguientes pasos:

Actividad 1: Definición de la necesidad a partir de la caracterización del mercado potencial que tendrán los productos de la línea (Cuales tienen mayor demanda, cuál es la competencia, cuál es el tamaño del mercado y cuáles son las oportunidades).

Actividad 2: Entendimiento de los dominios relevantes proporcionando los requisitos y los aspectos comunes y variables a todos los sistemas (aplicaciones) que forman parte del dominio.

Definición de la familia de productos:

Actividad 1: Proyección tecnológica sobre los productos que pueden llegar a ser factibles en el futuro cercano.

Actividad 2: Análisis de la variabilidad de la familia de productos. Para ellos puede utilizarse un diagrama de variabilidad (Herramienta que permite representar gráficamente el conjunto de elementos variantes identificados en cada componente así como las relaciones existentes entre ellos). Para la elaboración de estos diagramas existen diversas notaciones cuyo uso está más o menos extendido: FODA(KANG 1990), FORM (KANG 1998), RSEB (GISS *et al.* 1998).

Actividad 3: Construcción de un caso de negocios (Proporciona una justificación de la selección de productos y del enfoque que se usará para construirlos).

Actividad 4: Definición del alcance de la línea productos (Cuales productos serán incluidos en la línea y cuáles no).

- Definir en forma de objetivos específicos el alcance de lo que se quiere desarrollar, dejando claramente identificados los productos de software a obtener.
 - Además se pueden definir criterios de calidad reflejando los requisitos de adaptabilidad de estos productos para su incorporación a la línea.
- Descripción de la factibilidad técnico comercial y la sostenibilidad en el tiempo.
- Definir la estrategia para la comercialización de los productos y servicios de la línea.

Definición del modo de producción:

Para facilitar la implantación del modelo, se definen en la guía varias actividades relacionadas con las políticas, los roles, los registros, estructuras de dirección o administrativas y los procedimientos, mecanismos reguladores del dominio de aplicación, de las conductas internas que ayudan a asegurar que las directivas se lleven a cabo y aseguran que se tomen las acciones necesarias, además de orientar los riesgos hacia la consecución de los objetivos de la entidad.

Actividad 1: Definición de la estrategia de producción de la línea de productos, ver Tabla 7.

Tabla 7 Estrategias de producción de la línea.

| Nombre de la estrategia | Descripción |
|-------------------------|-------------|
| | |

| | |
|---|--|
| Proactiva | Los activos de la línea son construidos por necesidades internas bien planificadas alineadas con los objetivos estratégicos y no por la influencia de necesidades externas o emergentes. |
| Reactiva | Los activos de la línea son construidos a partir de necesidades emergentes y generalmente por necesidades de productos requeridos por los clientes |
| Modelo híbrido, esencialmente proactiva | Este es el modelo recomendado que emplearemos en este centro básicamente es un modelo híbrido donde la mayoría de los activos se construyen por necesidades internas bien planificadas alineadas con los objetivos estratégicos y no por la influencia de necesidades externas o emergentes. No obstante algunos de los activos se construye de forma reactiva |
| Modelo híbrido esencialmente reactiva | Modelo híbrido donde la mayoría de los activos se construyen por necesidades emergentes de productos que con frecuencia son productos requeridos por los clientes externos, este modelo también incluye el desarrollo de algunos activos de forma proactiva. |

Actividad 2: Definición en la organización para la ejecución de la actividad fundamental un número suficiente pero reducido de roles. Ver más detalles en (TORRES 2010).

Actividad 3: Definición de las entidades que conforman el modelo de desarrollo. El proceso de desarrollo está representado por seis entidades que soportan completamente el montaje de líneas de productos de software como proceso de producción. Cada una de estas entidades cubre un área importante dentro de la línea y garantizan la involucración de todos los miembros en el modelo y las responsabilidades individuales de cada miembro de la organización en la calidad y la productividad. Además de las entidades que conforman el modelo, se detallan en la sección otros procesos asociados a la actividad fundamental, importantes a la hora de definir el modo de producción de la línea, ver Fig. 13.

Entidad 1: Dedicada a la Ingeniería de dominio, es la entidad donde se construyen los activos que soportan la línea de productos de software. Básicamente todos los componentes y artefactos que se generan en el proceso de ingeniería de dominio constituyen los activos de software de la línea (algoritmos, componentes, planes, documentación, especificaciones de requisitos, etc.). Se recomienda que se conformen proyectos para el desarrollo de los activos generados en la ingeniería de dominio. Las actividades de la entidad se concentran en:

- **Análisis del dominio:** En este proceso a partir de un adecuado estudio del estado del arte y de la experiencia se adquiere el conocimiento, se analizan e identifican las aplicaciones relevantes para el dominio, se identifican los requisitos de las mismas y se procede al modelado del dominio.
- **Diseño del dominio:** En este paso se diseña construye y evalúa la arquitectura que soporta las aplicaciones del dominio específico.
- **Ingeniería de componentes:** Se construyen los componentes que soportan completamente la arquitectura y se gestiona y maneja el repositorio de componentes que soporta la línea.

Entidad 2: Dedicada a la Ingeniería de Aplicaciones, esta entidad funciona básicamente bajo un modelo de desarrollo basado en componentes. Se recomienda que se conciba un proyecto para el desarrollo de las aplicaciones independiente de los proyectos previstos para la ingeniería del dominio. Se sugiere además que las relaciones entre la ingeniería de aplicaciones y la ingeniería del dominio, para el caso de los requerimientos y diseño de los activos de forma reactiva, se realice siguiendo los procesos de gestión de proveedores.

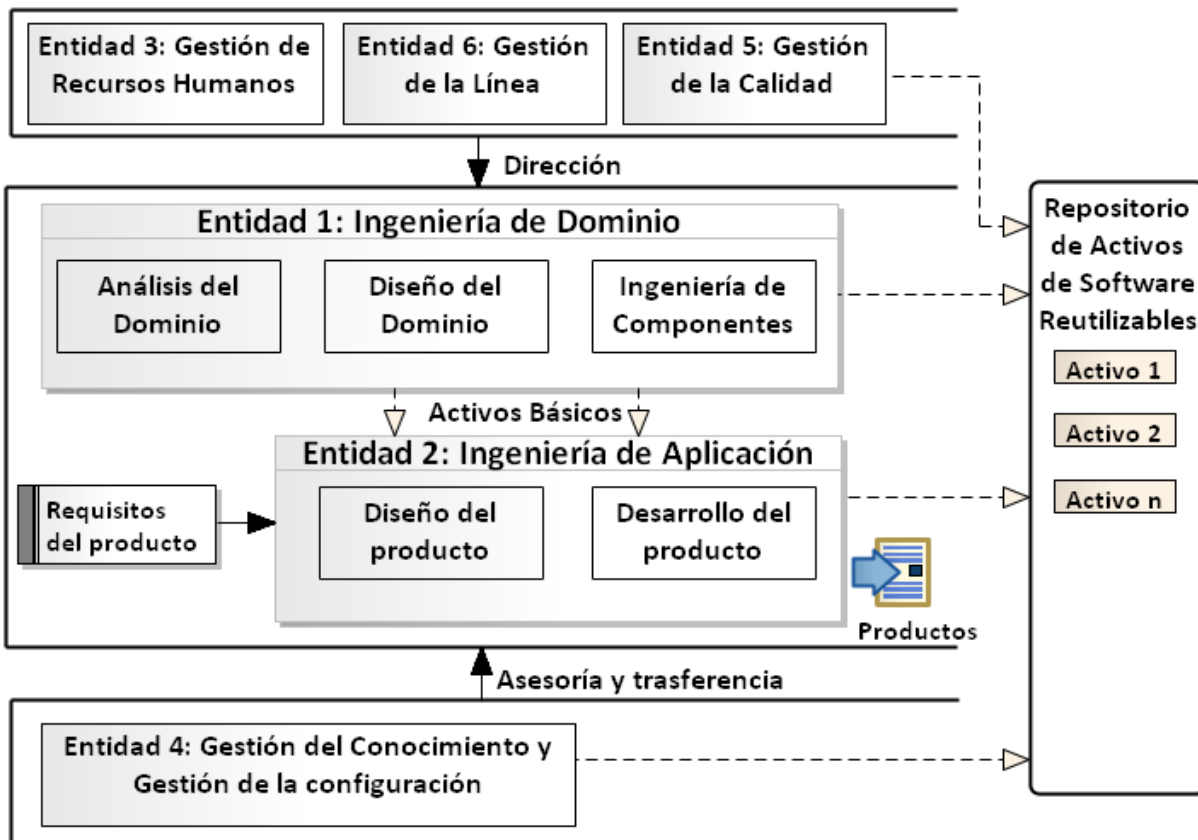


Fig. 13 Vista de desarrollo integral basado en los principios de líneas de productos de software.

Entidad 3: Dedicada a la Gestión de los Recursos Humanos (TORRES 2010), es la entidad que cumple las funciones de obtener los recursos humanos necesarios para crear los equipos de trabajo y demás puestos de trabajo de la línea. Desarrolla las habilidades del personal de la línea y se encarga del seguimiento del rendimiento de los miembros de los equipos. Además es la responsable de determinar la estructura de la organización, los roles y las responsabilidades del centro. Las actividades que incluyen son:

- Adquisición de los recursos humanos.
- Desarrollo de los recursos humanos.
- Planificación de los recursos humanos.
- Gestión de los recursos humanos.

Entidad 4: Dedicada a la gestión del conocimiento, a la gestión de configuración de los activos de software y en general al soporte de los productos de la línea. Tiene dentro de sus responsabilidades la

gestión del repositorio donde se almacenan los componentes y el mantenimiento de los activos que allí se encuentran. Su objetivo fundamental es asegurar la disponibilidad de los componentes para apoyar la ingeniería de aplicaciones.

Entidad 5: Dedicada a la gestión de la calidad, tiene dentro de sus principales responsabilidades garantizar que los requerimientos sean comprendidos y asumidos exactamente, con plena satisfacción de las necesidades y expectativas de los clientes (internos y externos). En su funcionamiento debe involucrar a todos los miembros de la organización de forma activa, bajo los principios de la calidad total garantizando la mejora continua de la organización. Además es la encargada de garantizar el aseguramiento de la calidad de todos los activos y productos de la línea y verificar el alcance de las soluciones desarrolladas (MATIAS 2011).

Entidad 6: Dedicada a la gestión del proceso, esta entidad representa a la alta gerencia de la organización. La actividad fundamental de esta entidad es la gestión de integración, la mejora continua del modelo, el control interno y la medición constante del proceso. Además se encarga de otras actividades relacionadas con la gestión de los proyectos como:

- Determinar las responsabilidades de autorizar, ejecutar, registrar y comprobar cada transacción y transferencia entre las distintas áreas de responsabilidad definidas en la entidad.
- Evaluar la actualización del sistema de indicadores de rendimiento implementando en la entidad para la puesta en marcha de acciones correctivas que disminuyen o eliminan las desviaciones importantes.
- Revisar que el plan de prevención elaborado ha tenido en cuenta el diagnóstico de los riesgos internos o peligros potenciales, el análisis de las causas que lo provocan o propician y las propuestas de medidas para prevenir o contrarrestar su ocurrencia.

Actividad 4: Definición de los procesos para el funcionamiento de la línea de productos. Se presentan los procesos a documentar organizados en las cuatro fases principales del desarrollo de proyectos, se insiste en que aunque estén bien determinados los procesos deben ser lo más simples y sencillos posible. Adicionalmente es importante la categorización de los procesos y la alineación de estos con las áreas de conocimientos relacionadas con la gestión de proyectos, cumplimiento con las políticas y niveles de calidad y desempeño planeados. Además es importante definir las entidades responsables de cada proceso y la correspondencia de estos con los procesos propuestos por el Project Management Institute (P), el Software Engineering Institute (S) y el Programa de Mejoras de la UCI (U), ver Tabla 8.

Tabla 8 Grupos de procesos propuestos por entidades.

| Procesos a ejecutar en la fase de inicio | | | |
|--|---|-----------|-------|
| Áreas | Procesos | Entidad | |
| Ingeniería de Dominio | Adquisición de conocimiento del dominio | Entidad 1 | |
| Alcance | Análisis de mercado | Entidad 4 | S,P,U |
| Comunicaciones | Identificar interesados | Entidad 3 | P |
| Integración | Análisis de factibilidad | Entidad 4 | P |
| | Desarrollar el plan de gestión del proyecto | Entidad 6 | S,P,U |
| Riesgos | Planificar la gestión de riesgos | Entidad 5 | S,P,U |
| Calidad | Planificar el aseguramiento de la calidad | Entidad 5 | S,P,U |

| Procesos a ejecutar en la fase de elaboración | | | |
|--|--|-----------|-------|
| Ingeniería de Dominio | Colectar nuevos requerimientos del dominio | Entidad 1 | S,P,U |
| | Actualizar el modelo de dominio | Entidad 1 | |
| Alcance | Gestión del alcance | Entidad 4 | S,P,U |
| | Crear estructura de desglose de trabajo | Entidad 6 | S,P |
| RRHH | Conformar equipo de proyecto | Entidad 3 | S,P |
| | Desarrollar plan de recursos humanos | Entidad 3 | S,P |
| | Desarrollar el equipo de proyecto | Entidad 3 | S,P |
| Riesgos | Identificar los riesgos | Entidad 5 | S,P,U |
| | Realizar análisis de los riesgos | Entidad 5 | S,P |
| | Planificar respuesta a los riesgos | Entidad 5 | S,P |
| Tiempo | Desarrollar el cronograma | Entidad 6 | S,P,U |
| Costos | Estimar los costos | Entidad 6 | S,P,U |
| | Determinar el presupuesto | Entidad 6 | S,P,U |
| Comunicaciones | Planificar las comunicaciones | Entidad 3 | S,P |
| Adquisiciones | Planificar la gestión de adquisiciones | Entidad 6 | S,P |
| Conocimiento | Evaluación y diagnóstico | Entidad 4 | P |
| Procesos a ejecutar en la fase de ejecución | | | |
| RRHH | Gestionar el equipo de proyecto | Entidad 3 | S,P |
| Comunicaciones | Distribuir la información | Entidad 3 | P |
| | Gestionar expectativas de los interesados | Entidad 3 | S,P |
| Adquisiciones | Conducir las adquisiciones | Entidad 6 | S,P |
| Conocimiento | Aplicación del conocimiento | Entidad 4 | |
| Ingeniería de Dominio | Diseño de la arquitectura de dominio | Entidad 1 | |
| | Evaluación de la arquitectura de dominio | Entidad 1 | |
| | Desarrollo de nuevos activos | Entidad 1 | |
| | Gestión del repositorio de activos | Entidad 4 | |
| Ingeniería de Aplicación | Análisis de requisitos del producto | Entidad 2 | |
| | Selección de la arquitectura del producto | Entidad 2 | |
| | Minería de activos existentes | Entidad 2 | S |
| | Adaptación de activos existentes | Entidad 2 | |
| | Integración del producto | Entidad 2 | |
| Procesos a ejecutar para el control y seguimiento | | | |
| Integración | Supervisión y control del trabajo | Entidad 6 | S,P,U |
| | Control integrado de cambios | Entidad 6 | S,P |
| Calidad | Aseguramiento de la calidad | Entidad 5 | S,P,U |
| Alcance | Verificación del alcance | Entidad 5 | S,P |
| | Control integrado de cambios | Entidad 4 | S,P |
| Tiempo | Control del cronograma | Entidad 6 | S,P,U |
| Costos | Control de los costos | Entidad 6 | S,P |
| Comunicaciones | Informar el rendimiento | Entidad 3 | P |

| | | | |
|---|---|-----------|-------|
| Riesgos | Monitorear y controlar los riesgos | Entidad 5 | S,P |
| Adquisiciones | Administrar las adquisiciones | Entidad 6 | S,P,U |
| Procesos a ejecutar para el cierre | | | |
| Calidad | Pruebas de liberación del producto | Entidad 5 | U |
| Ingeniería de Aplicación | Implantación y soporte del producto | Entidad 4 | S,U |
| Adquisiciones | Cerrar las adquisiciones | Entidad 6 | S,P |
| Integración | Archivar proyecto | Entidad 6 | S,P |
| Conocimiento | Generalización y socialización del conocimiento | Entidad 4 | |

Paso 2.2: Definición del modelo desde la vista de datos.

En esta vista se definen cuales son los datos de entrada y salida para la definición de la línea de productos de software. Se organiza básicamente según los elementos que definen una línea de productos.

Definición del modo de producción:

Actividad 1: Configuración del expediente de la línea.

En él se identifica la estructura y organización de la entidad. Se refleja a través de un expediente digital que contiene toda la información que se deberá guardar de cada proyecto y de la línea. Quedando completamente documentado el proceso de desarrollo para una futura certificación. Debe incluir los siguientes elementos, ver Fig. 14:

- [-] 100 Vista Negocio
 - [+] 100 Disenno Estrategico y Estructuras
 - [+] 110 Leyes Res Polit y Lineamientos
 - [+] 130 Modelos y Plantillas
 - [-] 140 Proc de la organización
 - [+] 120 Proc Asesoría Tec
 - [+] 121 Proc Desarrollo Tecnológico
 - [+] 122 Proc Desarrollo Infraestructura
 - [+] 141 Procesos G.Conocimiento
 - [+] 142 Procesos G.Calidad
 - [+] 143 Procesos G.RRHH
 - [+] 144 Procesos G.Adquisiciones
 - [+] 145 Procesos G.Comunicaciones
 - [+] 146 Procesos G.Costos
 - [+] 147 Gestion de la Configuración
 - [+] 148 Plan de Seguridad Informática
 - [+] 150 Servicios de la organización
- [-] 300 Vista Aplicaciones
 - [+] G.Activos
 - [+] G.Configuración
 - [+] G.Documental
 - [+] G.Indicadores
 - [+] G.Procesos
 - [+] G.Proyecto
 - [+] G.Riesgos
 - [+] 400 Vista Tecnología y Soporte
 - [-] 500 Vista RRHH (TPI y Registros)
 - [+] 1 Convenio colectivo
 - [+] 2 Planes de Adiestramiento
 - [+] 3 Planes de Trabajo
 - [+] 4 Evaluaciones trabajadores
 - [+] 5 Guardia Obrera
 - [+] 6 Plantilla y Nóminas
 - [+] 7 Reservas de cuadro
 - [+] 8 Salidas al exterior
 - [+] 9 Sindicato
 - [+] 10 UJC
 - [+] 11 TPI
 - [+] 12 Gestion Conocimiento
 - [+] 13 Curriculums

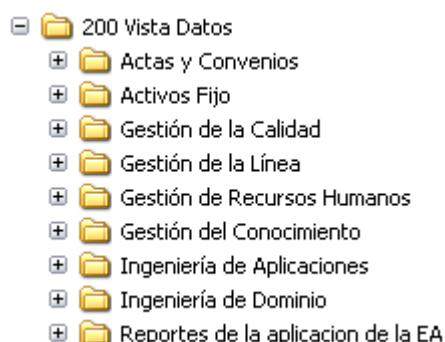


Fig. 14 Diseño del expediente de la Línea.

Actividad 2: Definición de los reportes para la toma de decisiones.
Estos reportes definen la dimensión de los datos que son requeridos para la toma de decisiones y se pueden representar en tres niveles fundamentales: la persona, el proyecto y la línea, ver Tabla 9.

Tabla 9 Propuesta de reportes para complementar el modelo.

| Planes y Evaluaciones (Nivel de Persona) | Gestión de Tiempo (Nivel de Proyecto) |
|---|---|
| Plan de Trabajo en el Proyecto Plan de Trabajo en el Centro Plan de Trabajo Hitos Centro Evaluación de Desempeño del Trabajador Evaluación de Tareas del Trabajador Incidencias del Trabajador Desviaciones del Trabajador Evaluación de Tareas del Estudiante Incidencias del Estudiante | Desviaciones del Proyecto Índice de Ejecución y Rendimiento Hitos del Proyecto Tareas del Proyecto Actividades en iteración Actividades en período |
| Gestión de Calidad (Nivel de Proyecto) | Gestión Riesgos (Nivel de Proyecto) |
| Reporte de No Conformidades Estado Pruebas por Iteraciones | Estado de los Riesgos Criticidad de los riesgos |
| Gestión RRHH (Nivel de Proyecto) | Gestión de Integración (Nivel Centro) |
| Asistencia al Proyecto Distribución por Roles Eficiencia y Productividad | Proyectos del Centro Estado de los Proyectos |
| Gestión RRHH (Nivel de Centro) | Gestión de Tiempo (Nivel de Centro) |
| Asistencia al Centro Miembros con pocas tareas Estudiantes del Centro por Proyecto Trabajadores del Centro por Proyecto Distribución por Roles | Hitos generales Fases por proyecto Desviaciones por Proyecto |

Definición de la arquitectura

Actividad 1: Definición de la arquitectura de referencia de la línea.

Para complementar esta actividad se recomienda utilizar los principios contenidos en la “*Guía base para la especificación de arquitecturas de software*”(LAZO and PIÑERO 2010) con sus nueve vistas de la arquitectura, agrupadas en tres áreas principales como se relaciona a continuación:

- Vistas de Arquitectura de Sistema
 - Vista de Procesos
 - Vista de Sistema
 - Vista de Datos
 - Vista de Integración
 - Vista de Presentación
- Vistas de Arquitectura de Tecnología
 - Vista de Seguridad
 - Vista de Entorno de desarrollo tecnológico
- Vistas de Arquitectura de Infraestructura
 - Vista de Infraestructura
 - Vista de Despliegue

Definición de los activos de software

Actividad 1: Clasificación de los activos de software reutilizables.

En su configuración y desarrollo los activos cumplen estrictos criterios de adaptabilidad para garantizar su integración. Es por ello que deben ser diseñados y concebidos detalladamente en función del propósito para el que se desarrollan, así mismo se deben garantizar su gestión en el repositorio de la organización. Se pueden clasificar de la siguiente manera:

- Activos que representan artefactos que formen parte del expediente de la línea o del proyecto.
- Activos que constituyen componentes de software y sus conectores (Componentes en tiempo de ejecución y cómo se comunican).
- Activos que constituyen los módulos o unidades de desarrollo.

Paso 2.3: Definición del modelo desde la vista de aplicaciones

Esta vista se refiere a las aplicaciones o herramientas informáticas requeridas para dar soporte a las vistas anteriores y se organiza en función de lo que estas aplicaciones necesitan para gestionar los datos y presentar la información de los elementos fundamentales de la línea. Son de extraordinaria importancia para el control interno y para garantizar la calidad. Por la importancia que reviste, se propone utilizar el documento “*Arquitectura Vista Entorno de Desarrollo Tecnológico*” para completar su definición (LAZO et al. 2010a).

Paso 2.4: Definición del modelo desde la vista de las tecnologías

Esta vista está compuesta por tres elementos fundamentales asociados a la base tecnológica de la línea de productos: hardware, redes y software.

Definición de las tecnologías que soportan la arquitectura de la línea

Actividad 1: Durante esta etapa se deberán puntualizar todos los detalles relacionados con los medios básicos que se encuentren a disposición de la línea para hacer cumplir las normas establecidas para el control interno de los medios. En esta vista se agrupan los recursos en tres áreas fundamentales:

Hardware: Representa la infraestructura que soportará el proceso productivo, incluyendo servidores de desarrollo y estaciones de trabajo.

- Nodos de comunicaciones, servidores dedicados y servicios de terceros.
- Infraestructura para el desarrollo (estaciones de trabajo).

Redes: Este elemento soporta la infraestructura de redes necesaria para enlazar todos los sistemas que forman parte de la Vista de Aplicaciones y garantizar la conectividad y visibilidad.

- Redes de comunicación por fibra óptica o acceso telefónico.

Software base: Este elemento describe los sistemas operativos y las herramientas que se utilizan para el soporte de la Vista de Aplicaciones y para la administración de los servicios.

- Sistema operativo de los servidores dedicados y de las estaciones de trabajo.
- Servidores web para múltiples plataformas, por ej. TomCat 6.0 es un servidor web con soporte de servlets y JSPs, no es un servidor de aplicaciones. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets, el motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache (En este caso es la manera que se utiliza).

Se recomienda utilizar el documento “*Arquitectura Vista de Infraestructura*” (LAZO et al. 2010b).

Definición de las tecnologías que soportan la gestión de los activos de software

Actividad 1: Montar un repositorio de activos de reutilizables utilizando para ello alguna herramienta que permita catalogar y realizar búsquedas dentro del mismo. Preferentemente que permitan gestionar los cambios que se realizan en los activos y que sea accesible desde la red (PUPO 2011).

2.2.3. Paso 3: Definición del cronograma de los modelos de transición

Para establecer el cronograma a seguir en la implantación del modelo propuesto de líneas de productos de software es necesario seguir los pasos representados en la Fig. 15.

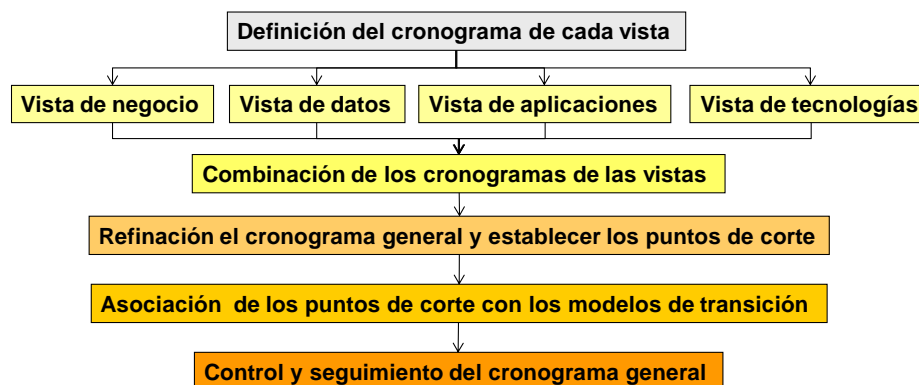


Fig. 15 Definición del cronograma de los modelos de transición.

Paso 3.1: Definición del cronograma de cada una de las vistas del modelo

A partir de las siguientes acciones se construye el cronograma necesario para implantar el modelo propuesto en cada una de las vistas, ver Fig. 16. Las entradas de este proceso son todos los pasos

identificados en las cinco vistas del modelo ideal y las salidas son los cronogramas individuales para la implantación del modelo en cada una de las vistas.

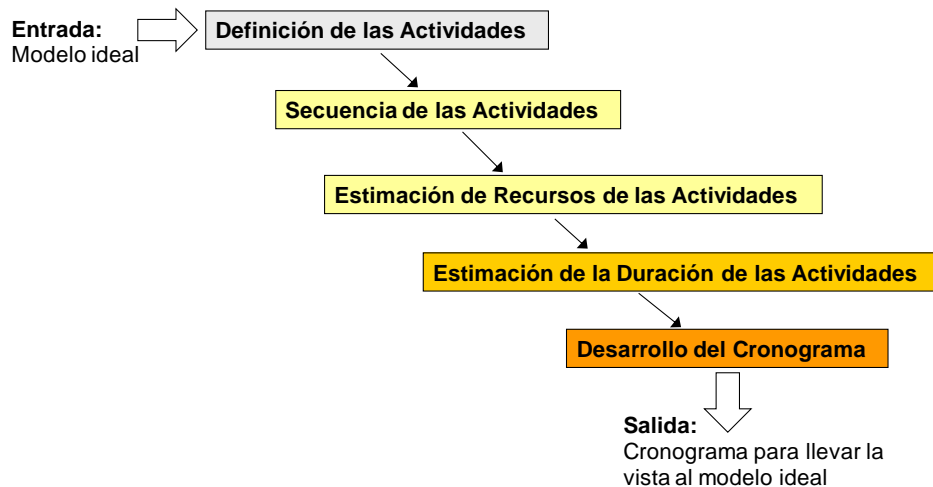


Fig. 16 Fases de una iteración de la aplicación del modelo de líneas de productos.

Actividad 1 (Definición de las actividades): Se identifican todas las actividades que son necesarias ejecutar para lograr cada uno de los pasos definidos en el modelo ideal, con vistas a crear una guía con ellas. Para ello se puede utilizar la técnica de descomposición, que consiste en la construcción de un árbol empleado como estrategia el desglose de los pasos del modelo ideal en tareas más sencillas y repetir este algoritmo recursivamente con ellas.

Comprensión del modelo e identificación del ambiente de la vista

Si (paso es sencillo y soluble)

Entonces resolverlo y terminar

Sino

División del paso en tantas tareas como sea necesario

Mientras (no se hayan resuelto todos los pasos)

Resolver el paso del modelo, utilizando este mismo algoritmo basado en la estrategia divide y vencerás

Integrar la solución de todos los pasos del modelo ideal en una única lista de tareas.

Actividad 2 (Secuencia de las actividades): Se define la precedencia de las actividades identificadas en la acción anterior y se genera una versión preliminar del cronograma para la implantación del modelo en la vista. Implica identificar y documentar las relaciones lógicas entre las actividades del cronograma, estas actividades pueden estar ordenadas de forma lógica con relaciones de precedencia adecuadas, así como también adelantos y retrasos, para respaldar el desarrollo posterior de un cronograma realista y factible. Para lograr este objetivo puede utilizarse la técnica de diagramación por precedencia que es un método para crear un diagrama de red del cronograma del proyecto que utiliza casillas o rectángulos, denominados nodos, para representar actividades, que se conectan con flechas que muestran las dependencias. Incluye cuatro tipos de dependencias o relaciones de precedencia (Final a Inicio, Final a Final, Inicio a Fin, Inicio a Fin).

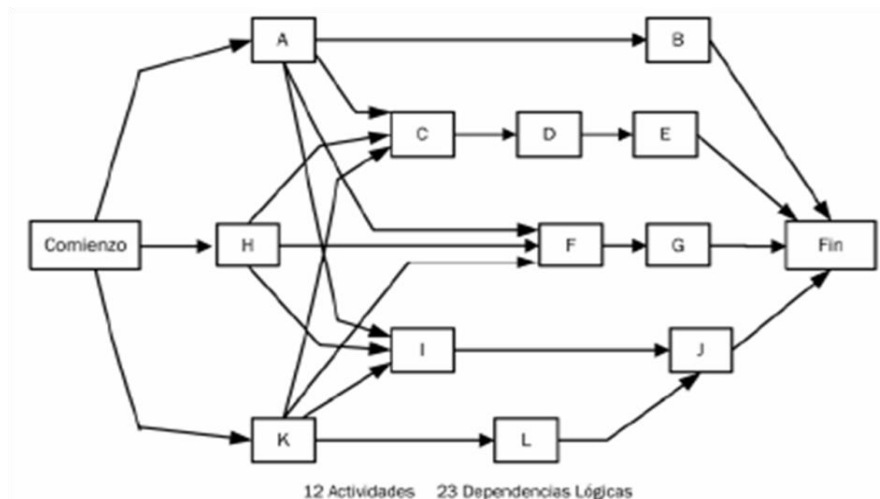


Fig. 17 Secuenciamiento de las actividades a través de un diagrama de precedencia.

Actividad 3 (Estimación de recursos de las actividades): Se identifican los recursos (Personas-Equipos-Materiales) necesarios para la ejecución de las actividades definidas en los pasos anteriores. Para ello se utilizan como entradas los requisitos de recursos de las actividades, los atributos de las actividades, la estructura de desglose de los recursos y el calendario de los recursos. Puede utilizarse el método de estimación ascendente que analiza el trabajo que aparece dentro de la actividad del cronograma descomponiéndolo con más detalles. Se estiman las necesidades de recursos de cada una de las partes inferiores y más detalladas del trabajo, y estas estimaciones se suman luego en una cantidad total para cada uno de los recursos de la actividad del cronograma.

Actividad 4 (Estimación de la duración de las actividades): Se definen los tiempos necesarios para ejecutar las actividades definidas a partir de las estimaciones de la duración de cada actividad con un rango de variación posible de $(2 \pm 3 \text{ días})$ y los atributos de las actividades, ver Fig. 18. Puede utilizarse la estimación por tres valores. En este método la precisión de la estimación de la duración de la actividad puede mejorarse teniendo en cuenta la cantidad de riesgo de la estimación original. Las estimaciones por tres valores se basan en determinar tres tipos de estimaciones:

- Más Probable (CM): La duración de la actividad se basa en una evaluación realista del esfuerzo necesario para el trabajo requerido y cualquier gasto previsto.
- Optimista (CO): La estimación de la duración de la actividad se basa en el análisis del mejor escenario posible para esa actividad.
- Pesimista (CP): El tiempo estimado de la actividad se basa en el análisis del peor escenario posible para esa tarea.

Se puede elaborar una estimación de la duración de la actividad utilizando un promedio de las tres duraciones estimadas, la fórmula utilizada puede ser la siguiente $(CO + 4CM + CP)/6$

| Id | Nombre de tarea | Duración | Dur. optimista | Dur. esperada | Dur. pesimista |
|----|--|------------|----------------|---------------|----------------|
| 0 | Plan de renovación empresa | 66,75 días | 67,75 días | 64,75 días | 73,75 días |
| 1 | Comienzo estudio proyecto renovación | 0 días | 0 días | 0 días | 0 días |
| 2 | Diseño | 26,25 días | 27,25 días | 24,25 días | 33,25 días |
| 3 | Designación de los 5 mejores arquitectos | 1,5 días | 2 días | 1 día | 3 días |
| 4 | Entrevistas a los arquitectos | 1,25 días | 1,25 días | 1,25 días | 1,25 días |
| 10 | Selección del arquitecto | 1,67 días | 2 días | 1 día | 4 días |
| 11 | Preparación del primer borrador | 1,83 días | 2 días | 1 día | 5 días |

Fig. 18 Estimaciones por tres valores.

Actividad 5 (Desarrollo del cronograma): En la construcción del cronograma se revisan y se corrigen las estimaciones de duración y las estimaciones de los recursos para crear un cronograma para la implantación del modelo en cada una de las vistas y que pueda servir como guía base con respecto a la cual poder medir el avance posteriormente, ver Fig. 19. Para este propósito se puede analizar la red del cronograma (Diagrama Gantt), a través de diversas técnicas analíticas, como el método ajustes de adelantos y retrasos que puedan distorsionar el cronograma.

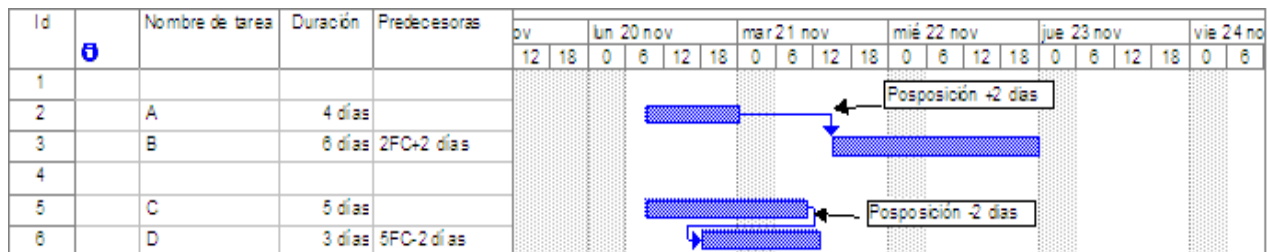


Fig. 19 Ajuste de adelantos y retrasos.

Paso 3.2: Combinación de los cronogramas de las vistas

Una vez obtenidos los cronogramas para implantar el modelo propuesto en cada una de las vistas, es necesario combinar todas las planificaciones en un solo cronograma general que sirva de hilo conductor del proceso y que mezcle las actividades en función de las prioridades de la organización, identificando claramente todas las actividades que coincidan en la misma fecha.

Paso 3.3: Refinar el cronograma general y determinar los puntos de corte

El proceso de ajuste del cronograma implica un esfuerzo importante para lograr optimizar los recursos y las prioridades de la organización en función de la implantación del modelo propuesto. Es recomendable partir de las siguientes preguntas para determinar las prioridades de la organización:

- ¿Qué vistas del modelo debemos organizar primero?
- ¿A quién asignar las responsabilidades de cada vista?
- ¿Cuáles son las áreas más importantes para la organización?
- ¿Cómo reutilizar los resultados obtenidos en las demás vistas del modelo?

A partir de las prioridades de la organización y partiendo del cronograma general se ajusta la planificación de todas las actividades balanceando cada una de las vistas del modelo según las prioridades. Pueden utilizarse varias técnicas para el refinamiento del mismo:

- La nivelación de recursos se usa para ajustar las actividades del cronograma que deben realizarse para cumplir con las fechas previstas, para resolver situaciones en las que se dispone

de recursos compartidos o críticos necesarios sólo en ciertos momentos o para mantener el uso de recursos seleccionados a un nivel constante durante períodos específicos. Este enfoque de nivelación del uso de recursos puede hacer que cambie el camino crítico original de la planificación.

- El método del camino crítico calcula las fechas de inicio y finalización tempranas y tardías teóricas para todas las actividades del cronograma, sin considerar las limitaciones de recursos, realizando un análisis de recorrido hacia adelante y un análisis de recorrido hacia atrás a través de los caminos de red del cronograma del proyecto.
- El método de la cadena crítica agrega *buffers* de duración que son actividades del cronograma no laborables. Una vez que se determinan estos *buffers* en el cronograma, las actividades planificadas se programan para las fechas de inicio y finalización planificadas más tardías posibles.

Paso 3.4: Asociación de los puntos de corte con los modelos de transición

Alineado con la mejora continua y partir del cronograma general, las prioridades de la organización y los puntos de corte identificados se propone que los modelos de transición que identifican las etapas intermedias, se establezcan en intervalos fijos semestrales o anuales, indicando los elementos tangibles que pueden ser mejorados en ese período cubriendo las cinco vistas de la arquitectura de empresas y cada uno de los elementos que definen la línea de productos. Estos intervalos de tiempo dependerán del nivel de la organización y del contexto donde se desea implantar la línea de productos. Podrán hacerse coincidir también con resultados tangibles como por ejemplo la definición de las estructuras que soportan las entidades del modelo.

Paso 3.5: Control y seguimiento del cronograma general

A través de este paso se gestionan los cambios que suceden en el estado actual del cronograma del proyecto y se detectan problemas que pudieran ocurrir con la planificación de las actividades. Influyendo sobre los factores que originan estos cambios y tomando las medidas pertinentes para corregirlos. Los informes del estado actual del cronograma pueden incluir información sobre las fechas de inicio y finalización reales, y las duraciones restantes para las actividades del cronograma no completadas. Si se usa una medición del avance por ejemplo, el valor ganado, también podrá incluirse el porcentaje completado de las actividades en curso del cronograma. Para facilitar el informe periódico del avance del proyecto, se puede usar una plantilla creada al efecto para estandarizar la información que puede hacerse en papel o de manera digital, en este caso pueden emplearse herramientas informáticas.

Conclusiones del capítulo

En este capítulo se propone un modelo para la organización de una empresa con un enfoque centrado en líneas de productos de software y que toma elementos positivos del PMI y del SEI fundamentalmente. En el análisis de las precondiciones, debilidades y fortalezas del modelo propuesto se arriban a las siguientes conclusiones:

- Precondiciones para utilizar el modelo propuesto
 - Es necesario que exista un repositorio de activos de software reutilizables, con los códigos fuentes de los activos, la documentación técnica y la certificación de calidad de los activos de ser posible, ver (PUPO 2011).
 - Debe existir en la organización disposición a la reutilización y al menos algún procedimiento para reutilizar componentes o activos en general.
- Fortalezas del modelo
 - Se definen seis entidades que conforman el modelo: Ingeniería de Dominio, Ingeniería de Aplicación, Gestión de Recursos Humanos, Gestión del Conocimiento y de la Configuración, Gestión de la Calidad y Gestión de la Línea en concordancia con las tendencias identificadas durante la fundamentación teórica.
 - El modelo propuesto tiene la novedad de combinar los principios de la Arquitectura de Empresas, las Líneas de Productos de Software y la mejora continua de la organización.
 - El modelo define claramente una guía de pasos para su implantación, fijando tres etapas esenciales: Diagnóstico del estado actual de la organización, definición del modelo ideal a alcanzar y definición del cronograma de los modelos de transición.
 - Como parte del modelo se proponen un grupo de procesos que abarcan las actividades de la línea y se agrupan según las fases de los proyectos. Estos procesos se basan fundamentalmente en los conceptos del PMI y el SEI sobre las líneas de productos de software, la gestión de proyectos y la mejora continua.
 - El modelo está concebido a partir del empleo de herramientas informáticas para automatizar la gestión de la línea, facilitar su implantación, seguimiento y control.
- Oportunidades de mejora del modelo
 - El modelo propuesto puede seguir mejorando a partir de su integración con los conceptos de Ecosistemas de Software.
 - El modelo puede ser perfeccionado detallando los grupos de procesos definidos para cada entidad según el área de dominio específica de la organización.

CAPITULO 3 VALIDACIÓN DEL MODELO PROPUESTO

En este capítulo se realiza un análisis de la aplicación del modelo de desarrollo para líneas de productos de software. Para evaluar el modelo propuesto se mostrarán los resultados obtenidos en su aplicación en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD) y en la Dirección Técnica de la Producción (DT). En esencia se analiza la influencia del modelo propuesto en la productividad, para estimarla se decidió tener en cuenta como indicador la cantidad de productos desarrollados y la eficiencia del desarrollo. Para analizar la calidad del modelo se utilizaron tres indicadores: calidad de la documentación, capacidad de ser generalizado y completitud (áreas del conocimiento que abarca).

3.1 Descripción de la muestra

El modelo propuesto en el capítulo dos ha sido aplicado en dos entornos, en un primer momento en el Centro de Tecnologías de Análisis y Almacenamiento de Datos (CENTALAD) desde octubre del 2008 hasta enero del 2010 donde comenzó a desarrollarse producto a la necesidad de organizar el proceso de productivo y más recientemente en la reorganización de la Dirección Técnica de la Infraestructura Productiva desde febrero del 2010 hasta la fecha.

En CENTALAD durante el período 2008-2010 en que se aplicó el modelo basado en líneas de productos de software, se trabajó en función de la obtención de cinco sistemas de información. Dos de estos proyectos pertenecieron al Departamento de Desarrollo de Herramientas de Análisis de Datos. Los tres productos restantes se desarrollaron en el Departamento de Desarrollo de Almacenes de Datos siguiendo una adaptación de la metodología Kimball (HERNÁNDEZ LÓPEZ *et al.* 2009) para el desarrollo de almacenes de datos. En la DT fue aplicado el modelo durante el transcurso del período 2010 en función de la obtención de un producto genérico enfocado en la gestión de proyectos.

3.2 Datos y experiencia práctica de la aplicación en el CENTALAD

A partir de la estructuración de un grupo de centros productivos en la Universidad, en septiembre del 2008 se creó el Centro de Tecnologías de Análisis y Almacenamiento de Datos (CENTALAD) que tenía la misión de impulsar el desarrollo relacionado con la línea de investigación relacionada con las tecnologías de almacenamiento y procesamiento de datos.

3.2.1 Diagnóstico del estado actual de la organización

El diagnóstico del estado actual de la organización se realizó desde septiembre hasta diciembre del 2008, obteniéndose los resultados siguientes en cada una de las vistas del modelo.

Dominio

- No existía un dominio bien definido sino un área temática amplia centrada en el desarrollo de tecnologías de bases de datos.
- En el centro recién creado no existían estudios de mercado sobre el área temática.

- Existían un incipiente desarrollo de la tecnología PostgreSQL y seis soluciones de réplica dispersas en la Universidad, además existían varias herramientas para la gestión de proyectos en la Universidad utilizándose en el centro el dotproject.

Familia de productos

- El centro heredó dos soluciones un Generador de Reportes (GRD) y el Sistema de Gestión Estadística (SIGE) que se perfilaban dentro del dominio de sistemas de ayuda a la toma de decisiones.
- Se utilizaban como herramientas para el desarrollo de los productos heredados los frameworks de desarrollo Symfony de PHP y .NET de C#, además los gestores SQLServer y MySQL que formaban parte de las soluciones mencionadas.
- No estaba definido ningún servicio de la organización.

Modo de producción

- El modelo de producción que existía en el centro era el clásico modo de producción, centrado en la gestión de proyectos de forma individual donde se asignaban los recursos humanos y materiales a cada proyecto sin potenciar la reutilización o la implantación de modelos industriales de desarrollo de software.
- Se almacenaba únicamente la información relacionada con el expediente de proyecto establecido en la Universidad, no se registraban datos relacionados con el proceso de producción y el control y seguimiento de los proyectos.

Arquitectura

- No existía una línea base de la arquitectura en el centro sino un documento en el expediente de los proyectos que recogía algunos elementos relacionados con la arquitectura de cada proyecto y que no tenían elementos de integración.
- Las herramientas que conformaban el entorno de desarrollo de la organización eran fundamentalmente Eclipse IDE y NetBeans IDE.
- El centro poseía un laboratorio con 30 puestos de trabajo con maquinas de 512 MB de RAM.
- Además de la red convencional del laboratorio existía un Access Point de acceso inalámbrico en el mismo laboratorio.
- En la organización existían varios sistemas operativos concentrados entre Windows y Linux

Activos de software

- Prácticamente no se hacía uso del concepto de activos de software porque los desarrollos no estaban orientados a la reutilización.

3.2.2 Modelo ideal definido

Se estableció como fecha para la implantación del modelo ideal a alcanzar por la organización en el primer ciclo de mejora diciembre del 2009 fijándose para ello un período de un año para alcanzar este estado.

Dominio

Se definió como parte del modelo implantado un dominio de aplicaciones relacionado con los sistemas de información.

Familia de productos

Se definieron dos familias de productos una relacionada con herramientas de análisis y procesamiento de datos y otra con soluciones de almacenes de datos.

Modo de producción

En el centro quedaron reflejadas las entidades propuestas en el modelo ideal (ver Fig. 13), distribuidas en los departamentos en algunos casos y centralmente en otros casos:

Departamento de Herramientas de Análisis de Datos:

- Entidad 1. (Grupo de Análisis y Grupo de Arquitectura) Encargado de la ingeniería de dominio, básicamente realizaba el análisis y seguimiento de los requisitos hasta la obtención del producto final, generando activos de software básicos. El segundo encargado de definir la arquitectura de referencia de la línea y desarrollar los componentes de la línea base.
- Entidad 2. (Grupo de Desarrollo) Responsable de construir los productos a partir de los activos de software generados, desarrollando los componentes necesarios.

Departamento de Soluciones de Almacenes de Datos:

- Entidad 1. (Grupo de Análisis y Grupo de Extracción, Transformación y Carga de Datos (ETL)) El primero de ellos encargado básicamente de realizar el análisis y seguimiento de los requisitos hasta la obtención del producto final, generando los activos de software primarios. El grupo de ETL era el encargado de procesamiento de las fuentes de los datos a partir de las necesidades de información definidas en los requisitos.
- Entidad 2. (Grupo de Almacenes de Datos y Grupo de Inteligencia de Negocio) El primero diseñaba los activos necesarios para poder almacenar la información y el segundo era el responsable de construir los reportes y análisis necesarios para poder satisfacer los requisitos de información a partir de los activos de software generados.

Entidades definidas centralmente en el centro:

- Entidad 3. (Departamento de gestión de los recursos humanos y del conocimiento) Encargada de la gestión de los recursos humanos del centro, básicamente gestionaba los procesos de formación tanto de pregrado como de postgrado. Además se ocupaba de la gestión de las competencias técnicas de los equipos y de algunos procesos relacionados con la capacitación de los clientes.
- Entidad 4. (Departamento de servicios de postventa y soporte) Encargada de la asesoría técnica, la personalización de nivel 1, los servicios de soporte de los productos del centro y del manejo del repositorio de activos reutilizables. Se encargaba además de las relaciones con las entidades externas al centro y de la búsqueda de clientes y/o mejoramiento de los productos a los clientes ya existentes.
- Entidad 5. (Departamento de Calidad) Se encargaba del aseguramiento de la calidad de los productos y de los procesos del centro, basado en los principios de calidad total. Encargada además de establecer las políticas y regulaciones necesarias para un correcto funcionamiento del resto de las entidades.
- Entidad 6. (Subdirección de Producción) Entidad que funcionaba disgregada en los diferentes departamentos del centro, representada por el subdirector de producción, los jefes de

departamento y los jefes de grupos de trabajo. Se encargaba de controlar el trabajo de la línea gestionando las prioridades y el desarrollo de los productos.

Arquitectura

La arquitectura del centro se manejaba a nivel de los departamentos dependiendo del tipo de sistema del que se trate. Se documentaba a partir de lo establecido por la Dirección de Calidad de la Universidad (Calisoft) aunque en el caso de las almacenes de datos fue necesario incluir algunas adecuaciones por las características de la familia de productos.

Activos de software

Durante el transcurso de la construcción de los productos se generaban una serie de componentes y artefactos que constituían el núcleo de la reutilización. Estas salidas podían ser tan variadas o granuladas como se estimara: requerimientos, modelos de dominio, diseños, arquitecturas, pruebas de rendimiento, documentación, casos de prueba, componentes, servicios, procesos, etc. La mayoría de estos activos quedaban registrados en el expediente de cada proyecto o en expediente de familia de productos que recogía los elementos generales a todos los productos.

3.2.3 Evaluación de los resultados de la aplicación del modelo en CENTALAD

Análisis de la variable productividad, dimensión obtención de nuevos productos

Después de implantado el modelo en el *Departamento de desarrollo de herramientas de análisis de datos* y el *Departamento de desarrollo de almacenes de datos*, se obtuvieron los siguientes productos. Incluido el registro de uno de los productos (Generador de Reportes Dinámico) en el CENDA.

- Generador de Reportes Dinámico, versión 1.5 (Registro CENDA)
- Sistema de Gestión Estadística, versión 1.0
- Modelo 005 de la Oficina Nacional de Estadística ONE, versión 0.9
- Consumo de energía UCI, versión 0.9

Análisis de la variable productividad, dimensión eficiencia del desarrollo

Se produjo como consecuencia de la implantación del modelo ideal un aumento del índice de productividad en los proyectos analizados en el *Departamento de desarrollo de herramientas de análisis de datos* (Generador de Reportes Dinámico GRD y el Sistema de Gestión Estadística SIGE) con respecto al estado en el diagnóstico inicial. Específicamente el indicador utilizado para analizar la variable productividad fue la eficiencia, estimada a partir de la productividad en la etapa de desarrollo y la productividad en la etapa de soporte, considerando además un coeficiente para cada etapa según la complejidad de la misma.

- Coeficiente de desarrollo (CD)= 0.7
- Coeficiente de soporte (CS)= 0.3
- Productividad etapa de desarrollo (PD) se calcula a partir del Tamaño del Producto (TP) expresado en la cantidad de requisitos a desarrollar entre los Recursos (R) expresados en el tiempo en meses necesario para desarrollar el producto, ver Tabla 10.

$$PD = \frac{TP}{R}$$

- Productividad etapa de soporte (PS) se calcula a partir del Tamaño del Producto (TI) expresado en la cantidad de incidencias detectadas entre los Recursos (R) expresados en el tiempo de respuesta en meses necesario para resolverlas, ver Tabla 11.

$$PS = \frac{TI}{R}$$

- Eficiencia (E) la eficiencia se calcula a partir de la suma de la productividad en la etapa de desarrollo y la productividad en la etapa de soporte, multiplicadas por sus respectivos coeficientes, ver Tabla 12.

$$E = (CD \times PD) + (CS \times PS)$$

Tabla 10 Productividad de los proyectos durante el desarrollo.

| Proyectos | TP | R | PD |
|----------------|----|----|------|
| GRD Dic. 2008 | 49 | 16 | 3.06 |
| GRD Dic. 2009 | 94 | 13 | 7.23 |
| SIGE Dic. 2008 | 55 | 7 | 7.86 |
| SIGE Dic. 2009 | 89 | 11 | 8.09 |

Tabla 11 Productividad de los proyectos durante el soporte y mantenimiento.

| Proyectos | TI | R | PS |
|----------------|-----|----|-------|
| GRD Dic. 2008 | 0 | 1 | 0 |
| GRD Dic. 2009 | 16 | 12 | 1.33 |
| SIGE Dic. 2008 | 0 | 1 | 0 |
| SIGE Dic. 2009 | 189 | 12 | 15.75 |

Tabla 12 Eficiencia de total de los proyectos analizados.

| Proyectos | PD | PS | E |
|----------------|------|-------|-------|
| GRD Dic. 2008 | 3.06 | 0 | 2.14 |
| GRD Dic. 2009 | 7.23 | 1.33 | 5.46 |
| SIGE Dic. 2008 | 7.86 | 0 | 5.50 |
| SIGE Dic. 2009 | 8.09 | 15.75 | 10.39 |

A partir de estos resultados se demuestra un aumento de la productividad en los proyectos GRD y SIGE después de aplicado del modelo de desarrollo en la organización, con un factor de crecimiento de la eficiencia de 1.88 y 2.54 respectivamente, ver Fig. 20.

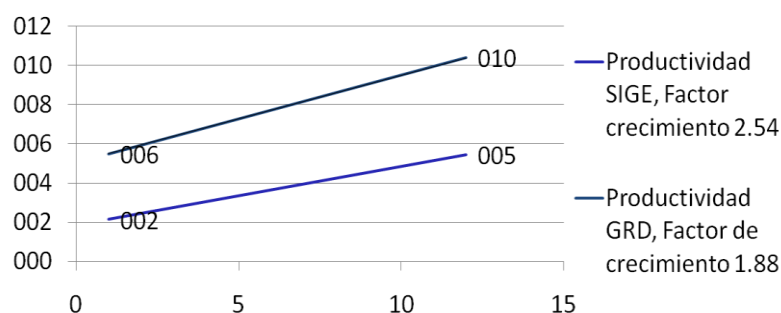


Fig. 20 Aumento de la productividad en el Departamento de desarrollo de herramientas de análisis de datos.

En el caso del *Departamento de desarrollo de almacenes de datos*, la variable productividad también se analizó en función del indicador eficiencia, estimada en este caso a partir del tamaño del producto por capas entre los recursos expresados en el tiempo en días necesario para desarrollar el producto.

- Tamaño del producto (TP) se calcula a partir de la estimación del volumen de requisitos a desarrollar por capas: Análisis (AN), Almacén de Datos (AL), Extracción Transformación y Cargar (ETL) e Inteligencia de Negocio (BI) en una escala de 1 a 5 (1 poco esfuerzo y 5 mucho esfuerzo), ver Tabla 13.

$$TP = AN \times AL \times ETL \times BI$$

Tabla 13 Análisis de la productividad en el departamento de Desarrollo de Almacenes de Datos.

| Soluciones de Almacenes de Datos | AN | AL | ETL | BI | TP |
|----------------------------------|----|----|-----|----|-----|
| 005 ONE (enero-marzo 2009) | 5 | 5 | 5 | 2 | 250 |
| Prototipo INE (marzo 2009) | 5 | 5 | 2 | 1 | 50 |
| Energía UCI(abril-mayo 2009) | 3 | 3 | 4 | 5 | 180 |

- Eficiencia (E) se calcula a partir del Tamaño del Producto (TP) expresado en el indicador del volumen de requisitos a desarrollar para cada capas entre los Recursos (R) expresados en el tiempo en días necesario para desarrollar el producto, ver Tabla 14.

$$E = \frac{TP}{R}$$

Tabla 14 Análisis de la productividad en el departamento de Desarrollo de Almacenes de datos.

| Soluciones de Almacenes de Datos | TP | R | E |
|----------------------------------|-----|----|------|
| 005 ONE (enero-marzo 2009) | 250 | 90 | 2.78 |
| INE (marzo 2009) | 50 | 15 | 3.33 |
| Energía UCI(abril-mayo 2009) | 180 | 45 | 4.00 |

La aplicación del modelo de producción permitió aumentar la productividad del *Departamento de desarrollo de almacenes de datos*. Los resultados positivos en la productividad se demuestran a través del indicador eficiencia en los resultados obtenidos con un aumento de 0,61 puntos como promedio con cada nuevo producto, ver Fig. 21.

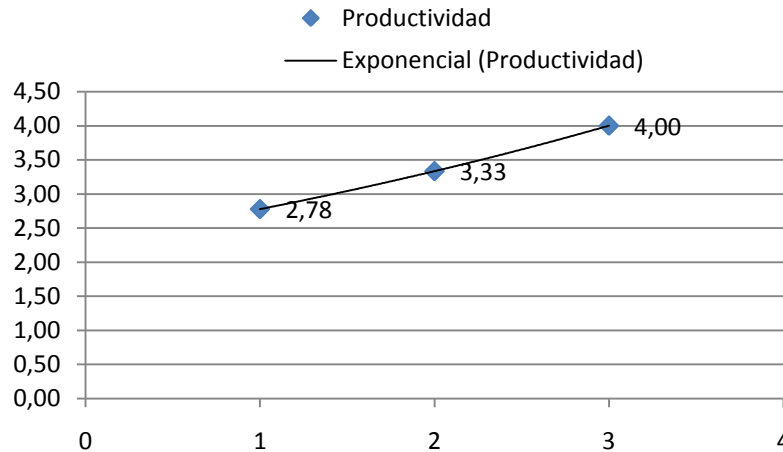


Fig. 21 Aumento de la productividad en el departamento de Desarrollo de Almacenes de Datos.

Análisis de la variable calidad del modelo

- Calidad de la documentación (Media): En el modelo se propone una guía de pasos donde se relacionan todas las acciones necesarias para la implantación del modelo, así como las entidades que los implementan, ver epígrafe 2.2.
- Capacidad de ser generalizado (Alta): Se implantó con éxito dentro de CENTALAD en dos familias de productos con diferentes tipos de sistemas, una enfocada al desarrollo de herramientas de análisis de datos y otra en la construcción de almacenes de datos.
- Completitud (Todas): Los procesos que se incluyen en el modelo abarcan la mayoría de las áreas propuestas por el PMI, el SEI y con el Programa de Mejora de la UCI, ver Fig. 22.

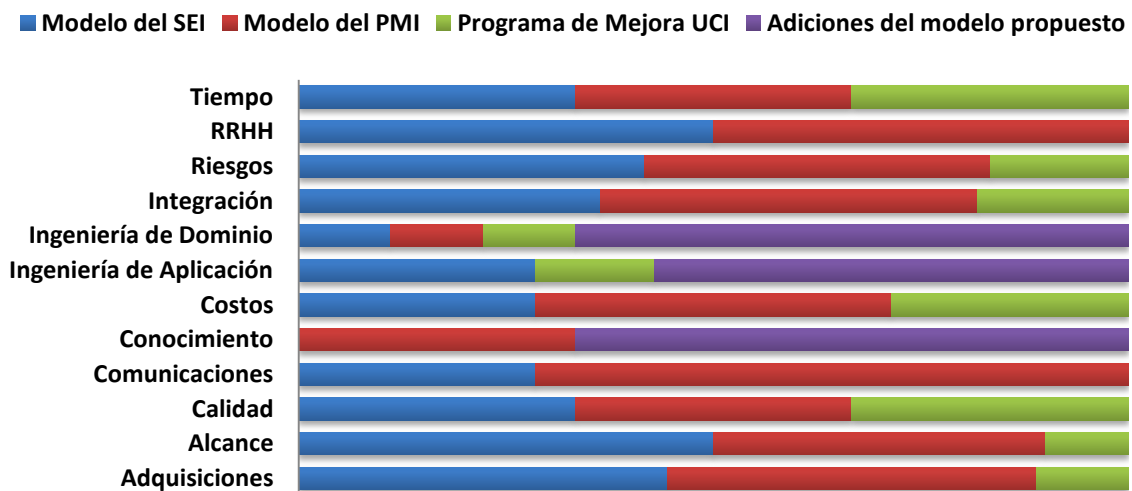


Fig. 22 Relación de los procesos del modelo propuesto y con otros modelos.

3.3 Datos y experiencia práctica en la Dirección Técnica.

Para lograr cumplir con los compromisos productivos desde el enfoque de la gestión de los proyectos y el con un nuevo modelo de formación se producen en la Universidad a partir del curso 2009-2010 modificaciones en el plan de estudio y en el modelo de producción simultáneamente potenciando la sinergia entre las áreas de producción, formación, investigación y postgrado. El centro de estas modificaciones vista desde la producción se enfocaba en la modificación del modelo de gestión de los proyectos, de las estructuras productivas y de las herramientas para lograr avances significativos en la organización del personal y por ende un aumento en la productividad. En este contexto surge la necesidad de reorganizar la Dirección Técnica de la producción según el nuevo modelo y las nuevas necesidades. Para lograr este propósito se aplicó la guía metodológica propuesta en y se creó una línea de producto de software enmarcada en el dominio de la gestión de proyectos.

3.3.1 Diagnóstico del estado actual de la organización

A partir de la aplicación del diagnóstico inicial realizado a la organización durante los meses de enero y febrero del 2010 se obtuvieron los siguientes resultados:

Dominio

- El dominio de aplicaciones que formaba parte del objetivo de la organización no estaba claro, solamente se tenía la orientación de crear una infraestructura que soportara la gestión de los proyectos fundamentalmente.
- No existían estudios relacionados con el dominio propuesto por encontrarse la dirección en plena transformación.

Familia de productos

- Existían algunos desarrollos vinculados a la herramienta Redmine.

Modo de producción

- El modelo de producción que existía en la organización estaba organizado por proyectos individuales.
- La información referente a los desarrollos de la dirección estaba organizada en forma de un portafolio que recogía los datos esenciales de cada proyecto.
- No se almacenaban datos que evaluaran el avance de los proyectos o métricas que permitieran mantener un control sobre estos.

Arquitectura

- No existían datos de la línea base de la arquitectura al no estar definida.

Activos de software

- No se recogía ninguna información relativa a los activos de software reutilizables.

3.3.2 Modelo ideal definido

Dominio

Teniendo en cuenta los nuevos objetivos que seguiría la dirección se definió como parte del dominio de aplicaciones de la línea de productos de software, las herramientas para la ayuda a la toma de decisiones a diferentes (nivel de persona, nivel de proyecto, nivel de centro de producción, nivel de la

alta gerencia UCI) ver Fig. 23. (PIÑERO, PEDRO 2010) tomando como base del dominio de aplicaciones de gestión de proyectos el modelo propuesto por el Project Management Institute (PMI 2004; 2008).



Fig. 23 Vista de los diferentes niveles para la ayuda a la toma de decisiones en la Dirección Integrada de Proyectos.

Familia de aplicaciones

Dentro de este dominio se idéntico una familia de productos, el Paquete de Gestión de Proyectos (GESPRO). Agrupando un conjunto de herramientas utilizadas para potenciar el control y seguimiento de los proyectos de la universidad y la implantación del nuevo modelo de desarrollo tecnológico. GESPRO agrupa un conjunto de herramientas representadas, ver Fig. 24 (No Registro CENDA Cuba 1540-2010) (PIÑERO, PÉREZ PEDRO *et al.* 2010).

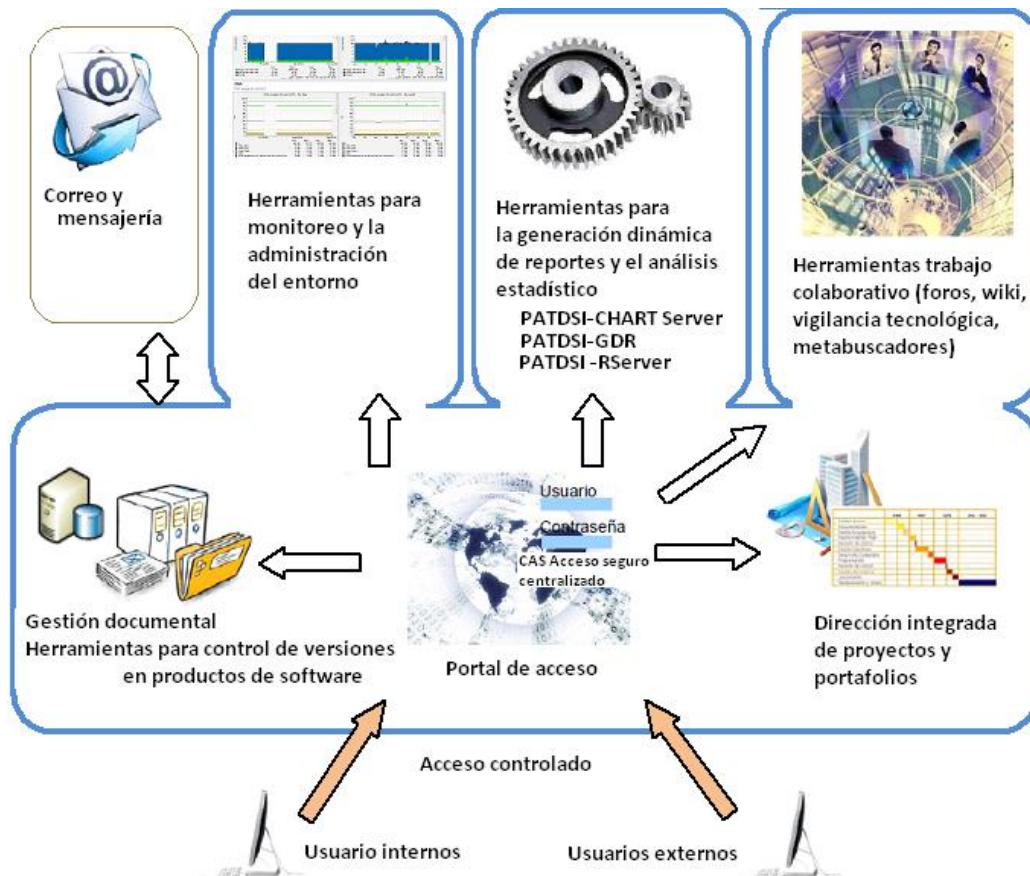


Fig. 24 Vista integrada de los componentes fundamentales de GESPRO

Modo de producción

El modelo de producción implantado en la línea de productos se conformó utilizando los principios de la guía metodológica propuesta y para ello se conformaron las siguientes entidades o grupos.

- Entidad 1. (Grupo de desarrollo tecnológico) Encargado de la Ingeniería de dominio, es quién concibe el desarrollo de proyectos tecnológicos que potencialmente se convertirán en activos reutilizables o en componentes COTS fundamentales, determina además los nuevos requisitos y gestiona su evolución.
- Entidad 2. (Grupo de soporte a la producción) Encargada de la construcción de GESPRO y de sus modificaciones para diferentes entornos y escenarios de trabajo. Además diseña y construye los activos y componentes fundamentales, construye además los conectores para la integración entre los diferentes componentes COTS y con otros sistemas externos.
- Entidad 3. (Grupo asistencia técnica) Encargado de la gestión de los recursos humanos de la línea, las competencias técnicas y el desarrollo integral de los equipos.
- Entidad 4. (Grupo de desarrollo tecnológico) Es la entidad encargada del manejo del repositorio de activos desarrollados y ayuda en la identificación de nuevos componentes para ser convertidos en activos de la línea de productos.
- Entidad 5. (Grupo asesoría técnica) Grupo encargado del control de la calidad de la línea toma como base el desarrollo basado en los principios de calidad total, vinculándose además a la gestión de los requisitos. También es la encargada de la asesoría técnica, la personalización de nivel 1 y los servicios de soporte de GESPRO.
- Entidad 6. (Grupo asesoría técnica) Gestiona las prioridades de la línea en función de las necesidades de los clientes y de los nuevos requisitos que puedan surgir. Controla el trabajo del resto de los grupos y maneja los compromisos externos de la línea.

Arquitectura

Algunos de los principios de la línea base son la comunicación basada en servicios y el desarrollo de extensiones a la plataforma como componentes COTS.

- Se estructura la arquitectura en las siguientes siete capas: infraestructura, sistema, presentación, datos, seguridad, tecnología e integración.
- Se adoptan para el desarrollo y la integración entre los componentes COTS los siguientes estilos arquitectónicos:
 - Estilos de Llamada y Retorno: Model-View-Controller (MVC) y arquitecturas basadas en componentes.
 - Estilos Peer-to-Peer: arquitecturas orientadas a servicios (se establecen API para potenciar la conexión orientada a servicios).
 - Estilos Centrados en Datos: arquitecturas de Pizarra o repositorio tomando en algunos casos la base de datos como área de intercambio de información entre los sistemas.

Activos de software

Los módulos que conforman GESPRO y los activos que soportan la línea base son los siguientes:

- *Módulo Gestión de Portafolios de Proyectos*

Descripción: Permite la gestión de portafolios de proyectos. Plan de proyecto, la ejecución del plan de proyecto y en el control integrado de cambios (PMI 2004). Incluye más de 35 tipos de reportes personalizados para la toma de decisiones y facilidades para la generación dinámica de nuevos reportes a partir de solicitudes de clientes.

Activos: Subsistema Redmine v 1.0, Subsistema UCI PATDSI Generador Reportes 1.6, UCI Plugin Reportes.
- *Módulo Gestión de Tiempo*

Descripción: Permite la definición de las actividades, la secuencia de las actividades, estimación de la duración de las actividades, desarrollo del programa y control del programa.

Activos: Subsistema Redmine v 1.0.
- *Módulo de apoyo a la Gestión de la Calidad*

Descripción: Esta consiste en el apoyo en la planificación de la calidad, aseguramiento de la calidad y control de calidad.

Activos: Subsistema Redmine v 1.0.
- *Módulo Gestión de Alcance*

Descripción: Facilidades para la gestión del alcance del proyecto, la identificación de requisitos y la construcción de la estructura de desglose de trabajo. Se toman como base para desarrollo futuro de este módulo los elementos presentados en (REUBENSTEIN and WATERS 1991; RICH and WATERS 1986).

Activos: Subsistema Redmine v 1.0, Subsistema UCI CAXTOR, Subsistema para la Gestión de Alcance.
- *Módulo Gestión de Costos*

Descripción: La planificación de los recursos, estimación de los costos, preparación de presupuestos de costos y control de costos.

Activos: Subsistema Redmine v 1.0, Subsistema UCI SIGE.
- *Módulo Gestión de los Recursos Humanos*

Descripción: Procesos requeridos para realizar un uso más eficiente y eficaz de las personas involucradas con el proyecto. Esta consiste en la planificación organizacional, la adquisición de personal y el desarrollo del equipo.

Activos: Subsistema Redmine v 1.0, Subsistema UCI DECOM, UCI Plugin Redmine Asistencia.
- *Módulo Gestión de Adquisiciones*

Descripción: Gestión de los recursos materiales, la planificación de la adquisición hasta la asignación a proyectos.

Activos: Subsistema Redmine v 1.0.
- *Módulo Gestión de comunicaciones*

Descripción: Procesos requeridos para asegurar la generación, recopilación, disseminación, almacenamiento y disposición final de la información del proyecto. Facilita la planificación de las comunicaciones, distribución de la información, reporte del rendimiento, desempeño, gestión de acuerdos, gestión de compromisos, foros, noticias, wiki.

Activos: Subsistema Redmine v 1.0, Subsistema UCI PATDSI Generador Reportes 1.6, Subsistema ORION v1.0, UCI Plugin Redmine Side News v 1.0, Plugin Redmine Sidebar v1.0.

- *Módulo Gestión de Riesgos*

Descripción: Procesos para la identificación, análisis y respuesta al riesgo del proyecto. Esta consiste en la planificación de la gestión de riesgos, identificación de los riesgos, análisis cualitativo de los riesgos, análisis cuantitativo de los riesgos, planificación de las respuestas a los riesgos, y monitoreo y control de los riesgos.

Activos: Subsistema Redmine, UCI Plugin Gestión Riesgos v 2.0.

- *Módulo Gestión Documental*

Descripción: Permite la gestión documental de la organización potenciando el uso de flujos documentales y facilidades para la gestión de los documentos.

Activos: Subsistema UCI Excriba, UCI Gestión documental Plugin v1.0, UCI Plugin Components v1.0.

- *Módulo Control de Versiones*

Descripción: Módulo de especial interés en los proyectos de desarrollo de software, permite el control de versiones del código y potencia el trabajo colaborativo.

Activos: Subsistema Control de versiones, Subversion v1.4.5, GIT v1.0.

- *Módulo para las salvas de seguridad*

Descripción: Módulo que de forma integrada y controlada permite la salva de seguridad de los datos de la plataforma potenciando respaldo para: la gestión documental, para el control versiones y las actividades del sistema de dirección integrada de proyectos.

Activos: Subsistema Salvas de seguridad, Bacula Enterprise Edition 2.6.1.

- *Módulo de monitoreo y administración*

Descripción: Permite la administración centralizada de todos los recursos dispuestos en la red y que son explotados por la plataforma GESPRO, permite además el monitoreo permanente de la plataforma monitoreando más de 30 variables por cada servidor.

Activos: Subsistema UCI Monitoreo basado en Munin, Virtual Center: VMware vSphere 4.

- *Módulo de autenticación*

Descripción: Permite la autenticación centralizada tanto para usuarios internos como externos, garantiza además el manejo de la conexiones y la navegación por el paquete de soluciones. Autenticación contra LDAP.

Activos: Jasig CAS v3.3.1, LDAP, UCI Plugin Conexión CAS.

- *Módulo de herramientas de apoyo al desarrollo*

Descripción: Incluye un paquete de herramientas auxiliares para el desarrollo de los proyectos de desarrollo de software. Incluye herramientas para la protección del código abierto, y herramientas para el desarrollo de interfaces web basadas en extjs.

Activos: Subsistema ODEC v1.0, Subsistema CAXTOR 1.0, PATDSI-Chart Server 1.0.3, UCI Plugin Links.

- *Módulo de gestión de incidencias para gestión de no conformidades*

Descripción: Incluye como parte de la propia plataforma funcionalidades para la gestión de las propias incidencias de la plataforma facilitando su resolución como parte de un sistema de alta disponibilidad y rendimiento. Posibilita el soporte a 3 niveles.

Activos: Subsistema Redmine v 1.0.

- *Módulo de ayuda*

Descripción: Incluye como parte de la propia plataforma funcionalidades para la gestión de las propias incidencias de la plataforma facilitando su resolución como parte de un sistema de alta disponibilidad y rendimiento. Posibilita el soporte a 3 niveles.

Activos: Manual de usuarios de cada uno de los componentes COTS establecidos, Manual de usuario genérico, Banco de preguntas y respuestas para el soporte a la aplicación, Modelo de Gestión de Proyectos.

Activos que constituyen Componentes COTS:

- Apache 2, Phusion Passenger
- TomCat 6.0
- Máquina virtual Java 6.0
- Framework Ruby on Rails v2.3.5
- Postgresql 9.0
- Munin
- Bacula Enterprise Edition 2.6.1
- Jasig CAS v3.3.1
- Redmine v1.0
- Excriba v1.0
- Control de versiones SVN, GIT
- Subsistema PATDSI-Generador de Reportes v1.6
- Subsistema PATDSI-Chart Server 1.0.3
- Subsistema PATDSI-R-Server 1.0
- Subsistema PATDSI SIGE 2.0
- Subsistema CAXTOR 1.0
- Subsistema DCOM v1.0
- Subsistema ODEC v1.0
- Subsistema ORION v1.0

Activos conectores

- Plugin Advanced roadmap
- UCI Plugin Redmine CAS
- UCI Plugin Redmine Asistencia
- UCI Plugin Redmine Components
- UCI Plugin Reportes
- UCI Plugin Redmine Risks
- UCI Plugin Redmine Side News

- UCI Plugin Redmine Cambios
- UCI Plugin Documental
- UCI Plugin SVN Estadísticas
- Plugin Redmine ezSidebar
- Plugin Redmine Links

Activos que constituyen artefactos reutilizables:

- Manual de usuarios de cada uno de los componentes COTS establecidos.
- Manual de usuario genérico.
- Banco de preguntas y respuestas para el soporte a la aplicación.
- Documento de Requisitos de GESPRO
- Documentos de diseño y línea base de la arquitectura de GESPRO
- Casos de prueba de GESPRO
- Modelo de Gestión de Proyectos

3.3.3 Evaluación de los Resultados de la aplicación del modelo en la DT

A partir de la conformación de la línea de productos de software para el desarrollo de GESPRO, se desarrolló la primera versión del producto y se completaron personalizaciones combinando diferentes activos. Aplicar este modelo significó un aumento la productividad del equipo de trabajo y las oportunidades de negocios. Se presentan las personalizaciones realizadas del producto genérico:

Ejemplo aplicación GESPRO v1.0 en UCI

Se aplicó ajustado para dar soporte al modelo integrado de formación producción de la universidad. Se montó una granja de cinco servidores virtualizados y se montó sobre la misma un paquete GESPRO por cada uno de los trece centros de la Red de centros de la UCI ubicados en la sede central. Se montó además el paquete en las tres facultades regionales de la universidad ubicadas en varias regiones del país. Todos los centros interconectados por red, ver en la Tabla 15 los indicadores en el caso de la red de centros.

Tabla 15 Indicadores de aplicación de GESPRO y la LPS en la Red de Centros de la UCI.

| Indicador | Valor |
|--|--|
| Producto desplegado | GESPRO-UCI |
| Cantidad de desarrolladores de la solución personalizada | 10 |
| Tiempo en días para el desarrollo | 45 |
| Cantidad de usuarios que lo han usado | Más de 6000 |
| Cantidad de proyectos | Más de 200 incluidos proyectos con clientes y proyectos de innovación internos |

Elementos positivos:

- Se montó un sistema que garantiza una alta disponibilidad del sistema global por encima del 95%. Calculada a partir de (OKINAWA INTERNATIONAL CENTER 2010).

- Se logró estandarizar la herramienta de gestión de proyectos nacionales y de exportación que desarrolla la red de centros de la UCI facilitando la toma de decisiones y la agilidad en las informaciones para la dirección integrada de proyectos.
- Se logró integrar a través de la herramienta los modelos de formación del profesional y la producción de la UCI básicamente el control y seguimiento de las tareas de proyectos constituyen las tareas de entrenamiento de los estudiantes que son evaluadas por tutores y contribuyendo además al establecimiento de los principios de la calidad total.
- Se logró un cambio significativo en la organización del proceso de producción aunque se sigue trabajando en este sentido.

Elementos que constituyen oportunidades de mejora:

- La disponibilidad de los paquetes que brindan servicios a cada uno de los centros se comportó por debajo de la disponibilidad global siendo en algunos casos alrededor del 87%. Calculada a partir de (OKINAWA INTERNATIONAL CENTER 2010). Se realizaron mejoras y actualizaciones al paquete en función de elevar la misma. Se promueve la compra de equipamiento adicional para garantizar elevar a más de 95 % la disponibilidad en cada uno de los centros.

Ejemplo aplicación GESPRO en ALBET

Se aplicó para dar soporte a las necesidades en el control de proyectos y la gestión documental de la empresa ALBET SA, empresa comercializadora de los productos de la UCI. Esta empresa tiene varias sucursales fuera del país que constituyen parte de la misma, ver en la Tabla 16 los indicadores en el caso de la empresa ALBET.

Tabla 16 Indicadores de aplicación de GESPRO y la LPS en la Empresa ALBET.

| Indicador | Valor |
|--|--------------|
| Producto desplegado | GESPRO-ALBET |
| Cantidad de desarrolladores de la solución personalizada | 3 |
| Tiempo en días para el desarrollo | 10 |
| Cantidad de usuarios que lo han usado: | 58 |

Elementos positivos:

- Se montó un sistema que apoya la informatización de la empresa y a un mayor nivel de control de los proyectos y el estado de la organización.
- Se logró un cambio significativo en la organización de la gestión documental y se continúa avanzado en la organización del proceso de producción.
- Se montó un sistema que garantiza una alta disponibilidad del sistema global por encima del 95%. Calculada a partir del método presentado en (OKINAWA INTERNATIONAL CENTER 2010).

Elementos que constituyen oportunidades de mejora:

- Se trabaja en el despliegue de la solución en las sucursales de la empresa y el establecimiento de vías seguras para la comunicación y la transmisión de las informaciones.

Ejemplo aplicación GESPRO en el LIPS

El LIPS constituye el laboratorio industrial de prueba dónde se combinan la liberación de las soluciones informáticas de la universidad con la formación de los estudiantes de la universidad en el rol de revisor de la calidad. Los probadores pasan por el laboratorio y trabajan activamente con la herramienta durante la liberación de los productos, ver en la Tabla 17 los indicadores que en el caso del Laboratorio Industrial de Pruebas, la estructura y las necesidades de este laboratorio son diferentes de las del resto de las aplicaciones.

Tabla 17 Indicadores de aplicación de GESPRO y la LPS en el LIPS.

| Indicador | Valor |
|--|-------------|
| Producto desplegado | GESPRO-LIPS |
| Cantidad de desarrolladores de la solución personalizada | 2 |
| Tiempo en días para el desarrollo | 7 |
| Cantidad de usuarios que lo han usado: | 1275 |

Elementos positivos:

- Se establece sistema que potencia el desarrollo de las pruebas de liberación de los productos de la universidad. Mejorando la automatización del proceso.
- Se montó un sistema que garantiza una alta disponibilidad del sistema global por encima del 95%. Calculada a partir del método presentado en (OKINAWA INTERNATIONAL CENTER 2010).

Elementos que constituyen oportunidades de mejora:

- Se trabaja en el despliegue de nuevos reportes personalizados y en la capacitación de nivel avanzado en los usuarios administradores del sistema.

Ejemplo de aplicación Dirección de Supervisión y Control UCI

La dirección de supervisión y control (DSC) es una dirección de apoyo para el control interno utiliza el sistema para el control y seguimiento de sus actividades y en la organización de proyectos de auditoría a las diferentes áreas de la universidad. Ver en la Tabla 18 los indicadores de la aplicación de la LPS en el caso de estudio.

Tabla 18 Indicadores de aplicación de GESPRO y la LPS en la Dirección de Supervisión y Control.

| Indicador | Valor |
|--|------------|
| Producto desplegado | GESPRO-DSC |
| Cantidad de desarrolladores de la solución personalizada | 2 |
| Tiempo en días para el desarrollo | 4 |
| Cantidad de usuarios que lo han usado: | 10 |

Elementos positivos:

- Se logra la personalización de la herramienta en un plazo corto y su aplicación en un escenario diferente del inicial que se había concebido la aplicación GESPRO.

Elementos que constituyen oportunidades de mejora:

- Se trabaja en el despliegue de nuevos reportes personalizados y en la capacitación de nivel avanzado en los usuarios administradores del sistema.

Ejemplo aplicación Centros Regionales de Producción UCI

Se aplicó para dar soporte a las necesidades de los centros de desarrollo de software regionales (CR) de la Red de Centros de la Universidad de las Ciencias Informáticas. Estos centros son diferentes de los vinculados a las facultades porque no tiene estudiantes asociados. Esto implica una redefinición de los campos personalizados, de los reportes y de las herramientas. Ver los indicadores de la aplicación de la LPS en el caso de estudio, en la Tabla 19.

Tabla 19 Indicadores de aplicación de GESPRO y la LPS en los Centros de Producción Regionales.

| Indicador | Valor |
|--|-----------|
| Producto desplegado | GESPRO-CR |
| Cantidad de desarrolladores de la solución personalizada | 3 |
| Tiempo en días para el desarrollo | 10 |

Elementos positivos:

- Se completó el montaje del sistema de gestión de proyectos en toda la red de centros de la universidad elemento que permite el control de los proyectos incluso en los centros más distantes.
- Se logró la personalización de la herramienta para las condiciones específicas de los centros y se les brinda actualizaciones remotas a los mismos.

Elementos que constituyen oportunidades de mejora:

- Se trabaja en el despliegue de nuevos reportes personalizados y en la capacitación de nivel avanzado en los usuarios administradores del sistema.

Análisis de variable productividad, obtención de nuevos productos

Después de implantado el modelo en la Dirección Técnica de la Producción, se obtuvieron las siguientes personalizaciones del producto GESPRO, incluido el registro del producto genérico GESPRO en el CENDA.

- Aplicación en la Red de Centros de producción de la UCI.
- Aplicación en la empresa ALBET
- Aplicación en el Laboratorio Industrial de Pruebas
- Aplicación en la Dirección de Supervisión y Control UCI
- Aplicación en Centros Regionales de Producción UCI

Análisis de variable productividad, eficiencia del desarrollo

La implantación del modelo ideal estimuló un aumento de la productividad en la línea. Específicamente la variable productividad fue analizada a través del indicador eficiencia, calculada a partir del tamaño del producto dado la cantidad de componentes que lo integran así como su peso entre la cantidad de recursos utilizados, en este caso el tiempo en días para el desarrollo y la cantidad de personas involucradas en la personalización.

- Tamaño del producto (TP) se calcula con el indicador del volumen de trabajo para desarrollar las soluciones, a partir del producto del peso (P) estimado de cada uno de los componentes, según la complejidad de la integración en una escala de 1 a 5 (1 poco esfuerzo y 5 mucho esfuerzo), ver Tabla 20.

$$TP = \prod P$$

*Componentes: Jasig CAS (CAS), Redmine (RED), Excriba (EXC), Control de versiones (SVN), PATDSI-Generador de Reportes (GR), PATDSI-R-Server (R), PATDSI-Chart Server (CH), DCOM (D) y Sistema de Monitoreo (M).

Tabla 20 Estimación del tamaño de las personalizaciones de GESPRO.

| Producto/Componentes | CAS | RED | EXC | SVN | GR | R | CH | D | M | TP |
|----------------------|-----|-----|-----|-----|----|---|----|---|---|----|
| GESPRO-UCI | 2 | 4 | 2 | 1 | 4 | 1 | 1 | 1 | 1 | 64 |
| GESPRO-CR | 0 | 4 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 16 |
| GESPRO-ALBET | 2 | 4 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 32 |
| GESPRO-LIPS | 2 | 4 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 32 |
| GESPRO-DSC | 2 | 4 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 32 |

- Eficiencia (E) se calcula a partir del Tamaño del Producto (TP) expresado en el indicador del volumen de trabajo a desarrollar entre los Recursos expresados en el tiempo en días (T) necesario para desarrollar el producto más la cantidad de personas (CP) involucradas en la personalización, ver Tabla 21.

$$E = \frac{TP}{T + CP}$$

Tabla 21 Análisis de la productividad en la DT.

| Productos | TP | CP | T | E | Incremento de E |
|--------------|----|----|----|------|-----------------|
| GESPRO-UCI | 64 | 10 | 45 | 1,16 | - |
| GESPRO-CR | 16 | 3 | 10 | 1,23 | 0,07 |
| GESPRO-ALBET | 32 | 3 | 10 | 2,46 | 1,23 |
| GESPRO-LIPS | 32 | 2 | 7 | 3,56 | 1,1 |
| GESPRO-DSC | 32 | 2 | 4 | 5,33 | 1,77 |

Los resultados muestran un aumento de la productividad en la personalización del paquete GESPRO a partir de la aplicación del modelo de desarrollo en la organización, con un incremento promedio de la productividad de 1,04 puntos por cada nueva personalización, ver Fig. 25.

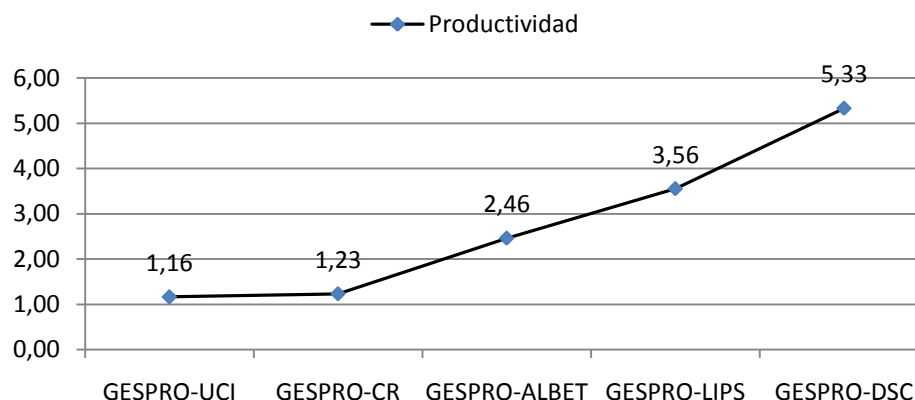


Fig. 25 Aumento de la productividad en la DT

Análisis de la variable calidad del modelo

- Calidad de la documentación (Media): En el modelo se propone una guía de pasos (ver epígrafe 2.2) bien definidos, relacionando todas las acciones necesarias para la implantación del modelo, así como los detalles de las entidades que lo definen. Además se proponen los procesos que deberían incluirse en cada una de las entidades (ver Tabla 8) y se establecen un grupo de principios de trabajo.
- Capacidad de ser generalizado (Alta): Se implantó con éxito en la Dirección Técnica en una familia de productos, relacionada con la gestión de proyectos. En la Fig. 26 se muestra un resumen del seguimiento de las no conformidades registradas de todos los sistemas GESPRO en todos los escenarios de la aplicación, demostrando con más de 6000 usuarios un índice de no conformidades por usuario menor de 0.05.

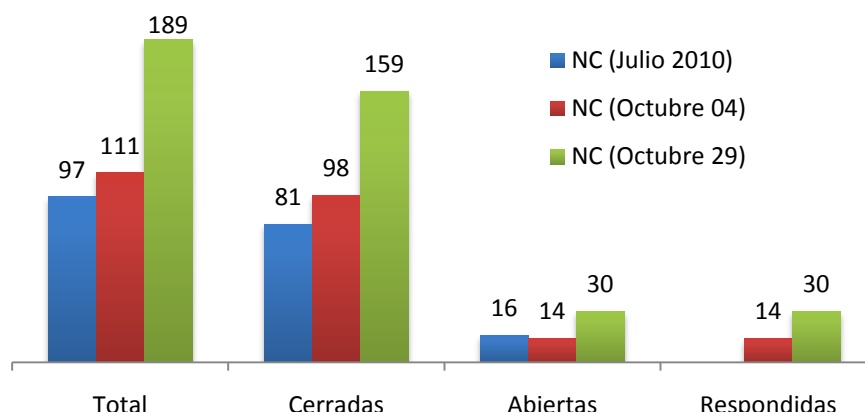


Fig. 26 Muestra del seguimiento a las no conformidades del sistema.

- Completitud (Todas): En la propuesta de procesos que se incluyen en el modelo se cubren la mayoría de las áreas que se definen en el PMI, en el SEI y en el Programa de Mejora de la UCI. Además los procesos propuestos están organizados según las fases del desarrollo de los productos, ver Fig. 27.

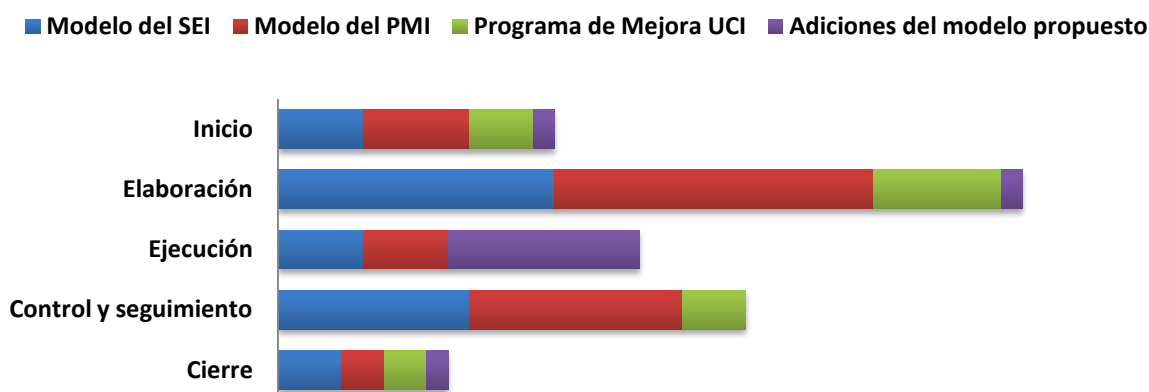


Fig. 27 Cantidad de procesos incluidos en cada fase.

Conclusiones del capítulo

A partir del análisis de resultados anteriormente presentado se arriba a las siguientes conclusiones divididas en las dos áreas de resultados:

- Después de haber implantado el modelo en CENTALAD se lograron terminar cuatro productos: Sistema de Gestión Estadística, Consumo de energía UCI, Modelo 005 de la ONE y Generador de Reportes Dinámico, este último incluyó su registro en el CENDA.
- Después de implantado el modelo en la Dirección Técnica de la Producción, se obtuvieron varias personalizaciones del producto GESPRO, registrado como producto en el CENDA: Aplicación en la Red de Centros de producción de la Universidad, en la empresa ALBET, en el Laboratorio Industrial de Pruebas de la UCI, en la Dirección de Supervisión y Control de la UCI y en los Centros Regionales de Producción de la UCI.
- Los costos asociados al desarrollo de las aplicaciones y su personalización en los diferentes escenarios son bajos y se reflejan en la disminución de los recursos involucrados en los desarrollos y las personalizaciones.
- Todos los productos así como sus activos son comercializados bajo licencia GPL potenciando el aumento del concepto de ingresos por servicios informáticos.
- Los productos y los activos que lo componen son dominados completamente por la Universidad de las Ciencias Informáticas potenciando la independencia tecnológica del modelo de producción.
- En el *Departamento de desarrollo de herramientas de análisis de datos* de CENTALAD, aumentó la productividad 2,21 puntos como promedio en cada solución a partir de la aplicación del modelo.
- En el *Departamento de desarrollo de almacenes de datos* de CENTALAD, se evidenció un aumento de la productividad a través del indicador eficiencia en los resultados obtenidos con un aumento de 0,61 puntos como promedio con cada nuevo producto después de haberse aplicado el modelo.
- Se incrementó como promedio la productividad en la Dirección Técnica 1,04 puntos en cada nueva personalización de GESPRO, empleando menos recursos humanos y menos tiempo.

CONCLUSIONES

- A partir del estudio del estado del arte del ciclo de vida de las Líneas de Producto Software se pudo observar la mayoría de los modelos estudiados tienen como entrada básica los requisitos y como resultado del proceso de desarrollo de productos además de los propios productos, toda la infraestructura de reutilización que va a permitir obtenerlos con menos esfuerzo.
- Muchos de los modelos estudiados tienen bien definido las áreas de procesos relacionadas con Ingeniería de Dominio y la Ingeniería de Aplicaciones, pero en muchos casos no especifican los procesos relacionados con la gestión del repositorio de activos de software, la gestión de los recursos humanos, el soporte de los productos desplegados o la mejora continua del proceso.
- Se propuso un nuevo modelo donde se definieron seis entidades: Ingeniería de Dominio, Ingeniería de Aplicación, Gestión de Recursos los Humanos, Gestión del Conocimiento y de la Configuración, Gestión de la Calidad y Gestión de la Línea en concordancia con las tendencias identificadas durante la fundamentación teórica.
- El modelo propuesto tiene la novedad de combinar los principios de la Arquitectura de Empresas, las Líneas de Productos de Software y la mejora continua de la organización.
- Como parte del modelo se proponen un grupo de procesos que abarcan las actividades de la línea y se agrupan según las fases de los proyectos. Estos procesos se basan fundamentalmente en los conceptos del PMI y del SEI sobre las líneas de productos de software, la gestión de proyectos y la mejora continua.
- Como parte del modelo se sugiere el empleo de herramientas informáticas para automatizar la gestión de la línea, facilitar su implantación, seguimiento y control.
- El modelo propuesto se aplicó en CENTALAD y posteriormente en la Dirección Técnica comprobándose su aplicabilidad en diferentes dominios de negocio.
- En el caso de los departamentos de CENTALAD se evidenció un aumento de la productividad 1,41 puntos como promedio con cada nuevo producto y en la Dirección Técnica se incrementó la productividad 1,04 puntos en cada nueva personalización de GESPRO, empleando menos recursos humanos y menos tiempo.

RECOMENDACIONES

- Continuar aplicando el modelo de desarrollo valorando aplicar su integración con los conceptos de Ecosistemas de Software.
- El modelo propuesto puede ser perfeccionado a partir de potenciar el análisis económico en el comportamiento de la línea de productos de software.
- Para fortalecer el modelo podrían proponerse nuevos procesos relacionados con la gestión de los activos reutilizables y detallar la manera de documentarlos.

BIBLIOGRAFÍA REFERENCIADA

- B2B-SOLUTIONS-GROUP. *Arquitectura Orientada a Servicios (SOA)*. Bogotá DC, 2009.
- BRUEGGE, B. and A. H. DUTOIT. *Ingeniería de software orientado a objetos*. Pearson Education, 2002. p. 9780134897257
- CATAL, C. *Barriers to the Adoption of Software Product Line Engineering*. ACM SIGSOFT Software Engineering Notes. Kocaeli, TURKEY, TUBITAK (The Scientific and Technological Research Council of TURKEY), 2009.
- CLEMENTS, P. and L. NORTHROP. *Software Product Lines: Practices and Patterns*. Addison Wesley, 2002. p.
- CLEMENTS, P. C.; L. G. JONES, et al. *Project Management in a Software Product Line Organization*, IEEE Computer Society, 2005.
- CHICOTE, C. V. *Desarrollo Integral de Sistemas de Procesamiento de Información Visual: Un Enfoque Multiparadigma basado en Líneas de Productos, Componentes y Generación Automática de Software*. Departamento de Tecnologías de la Información y Comunicaciones. Cartagena, Colombia, Universidad Politécnica de Cartagena, 2005. p.
- GALLEGOS, R. *SOA un nuevo modelo de arquitectura en TI*, 2007.
- GISS, M.; J. FAVRO, et al. *Integrating Feature Modeling with the RSEB*. 5th International Conference on Software Reuse. Vancouver, Canada, 1998. 76-85.
- GOMAA, H. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley, 2005. p. 0201775956
- GONZÁLEZ, L. C.; Á. O. BAS, et al. *Arquitectura de Empresa. Visión General*. Valencia, España, Centro de Investigación Gestión en Ingeniería de Producción. Universidad Politécnica de Valencia, 2005.
- HAMAR, V. *Aspectos metodológicos del desarrollo y reutilización de componentes de software*. Mérida, Mérida, 2003. p.
- HERNÁNDEZ LÓPEZ, A.; Y. GONZÁLEZ, et al. *Propuesta de metodología para el desarrollo de soluciones de inteligencia de negocio*. Ciudad de las Habana, Cuba, Universidad de las Ciencias Informáticas, 2009.
- IESE. *Annual Report 2004*. Fraunhofer, Alemania, Fraunhofer Institute for Experimental Software Engineering, 2004.
- IRIBARNE, L. *Un Modelo de Mediación para el Desarrollo de Software Basado en Componentes COTS*. Departamento de Lenguajes y Computación. Almería, España, Universidad de Almería, 2003. p.
- ISORNA, J. M. *Aplicaciones informáticas en arquitectura*. Ediciones UPC, 1999. p. 8483013282, 9788483013281
- ITEA, I. T. F. E. A. *CAFÉ Framework*, 2009. [Disponible en: <http://www.esi.es/Cafe/overview.html>]
- . *ESAPS Engineering Software Architectures, Processes and Platforms for System-Families*, 2001. [2010]. Disponible en: <http://www.esi.es/esaps/>
- KANG, K. C. *Feature-oriented Domain Analysis (FODA) Feasibility Study*, Carnegie Mellon Software Engineering Institute, 1990.
- . *FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures*. *Annals of Software Engineering*, 1998. 5: 143-168.
- KNAUBER, P.; D. MUTHIG, et al. *Applying Product Line Concepts in Small and Medium-Sized Companies*, IEEE Software, 2000. 88-95.
- KRUEGER. *Introduction to Software Product Line*, 2006. [Disponible en: www.softwareproductlines.com]
- LAZO, O. R.; M. G. JORRÍN, et al. *Arquitectura Vista de Entorno de Desarrollo Tecnológico*. Habana, Universidad de las Ciencias Informáticas, 2010a.

- . *Arquitectura Vista de Infraestructura*. Habana, Universidad de las Ciencias Informáticas, 2010b.
- LAZO, O. R. and P. P. PIÑERO. *Guía para la especificación de arquitecturas de software*. Universidad de las Ciencias Informáticas, Dirección Técnica de la Producción, 2010.
- MARTIN, R. and E. ROBERTSON. *Frameworks: comparison and correspondence for three archetypes*. Indiana, 2004.
- MATIAS, M. I. *Modelo para la Gestión de la Calidad en un centro de desarrollo de software*. Dirección Técnica de la Producción. Habana, Cuba, Universidad de las Ciencias Informáticas, 2011. p.
- MELLADO, J. *Estrategias de prueba de líneas de producto de sistema de tiempo real especificados con diagramas de estados jerárquicos*. Departamento de Ingeniería de Sistemas Telemáticos. Madrid, Universidad Politécnica, 2004. p.
- MONTILVA, J.; J. BARRIOS, et al. *Aspectos metodológicos del desarrollo de componentes de software reutilizable. SEPG Conferencia Latinoamérica*. Guadalajara, México, 2004.
- MONTILVA, J.; K. HAZAM, et al. *The Watch Model for Developing Business Software in Small and Midsize Organizations. Proceedings of the IV World Multiconference on Systemics, Cybernetics and Informatics - SCI'2000*. Orlando, Florida, 2000.
- NORTHROP, L. *Software Product Lines Essentials. Software Product Lines Essentials*. INSTITUTE, S. E. Carnegie Mellon University Pittsburgh, 2008.
- OKINAWA INTERNATIONAL CENTER, O. *Project Management Specialist*, 2010. [Disponible en: www.jica.go.jp/english]
- OPEN-GROUP, T. *The Open Group Architecture Framework*, 2010. [Disponible en: <http://www.opengroup.org/togaf/>]
- ORTIZ, A.; F. LARIO, et al. *A proposal for a Methodology to Develop Enterprise Integration Program: Computers in Industry*, 1999. 40: 155-171.
- PERRY, D. E. *Generic Architecture Descriptions for Product Lines*. Las Palmas de Gran Canaria, España, Second International ESPRIT ARES Workshop, 1998.
- PIÑERO, P. *Modelo de producción de la Universidad de las Ciencias Informáticas*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2010. 100.
- PIÑERO, P. P.; S. TORRES, et al. *Paquete de Herramientas para la Gestión de Proyectos*. Cuba, 2010. 1540-2010: 100. p.
- PMI. *Guía de los Fundamentos de la Dirección de Proyectos*. Newtown Square Pennsylvania, Project Management Institute, 2004. 409 p. 1-930699-73-5
- . *A Guide to Project Management Body of Knowledge (Draf)* Project Management Institute, 2008. 409 p.
- PUPO, I. I. P. *Propuesta de metodología para el diseño y explotación de repositorios de activos de software reutilizables*.: Centro de Desarrollo de Informática Industrial (CEDIN). Habana, Cuba, Universidad de las Ciencias Informáticas, 2011. p.
- REUBENSTEIN, H. and R. WATERS *The Requirements Apprentice: Automated Assistance for Requirements Acquisition IEEE Trans. Softw. Eng*, 1991, 17: 226-240.
- RICH, C. and R. WATERS. *Toward a Requirements Apprentice: On the Boundary between Informal and Formal Specifications*, Massachusetts Institute of Technology, 1986. [Disponible en: <http://hdl.handle.net/1721.3/5516>]
- ROLANDO ALFREDO HERNÁNDEZ LEÓN, S. C. G. *El paradigma cuantitativo de la investigación científica*. Ciudad de la Habana, EDUNIV, Editorial Universitaria, 2002. p. 959-16-0343-6
- SAMETINGER, J. *Software Engineering with Reusable Components*. Springer-Verlag, 1997. p.
- TORRES, I. S. *Modelo para el desarrollo de los recursos humanos*. Dirección Técnica de la Producción. Habana, Cuba, Universidad de las Ciencias Informáticas, 2010. p.

ZACHMAN, J. A. *The Zachman Framework for Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing*. 2003. p.

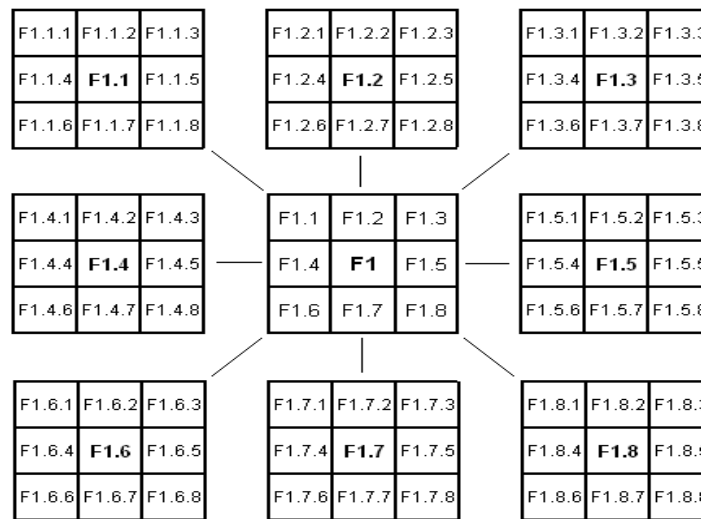
BIBLIOGRAFÍA CONSULTADA

- AJILA, S. A. *Change Management: Modeling Software Product Lines Evolution*. Ottawa, Ontario, Canada, Department of Systems & Computer Engineering, Carleton University, 2002.
- BERTO, M. F. and A. VALLECILLO. *Atributos de Calidad para Componentes COTS*. Málaga, España, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2002.
- BEUCHE, D. and M. DALGARNO. *Software Product Line Engineering with Feature Models*, 2006.
- BROWNSWORD, L.; C. C. WOODY, et al. *A Framework for Modeling the Software Assurance Ecosystem: Insights from the Software Assurance Landscape Project*, Carnegie Mellon University, 2010.
- CRESPO, Y.; M. Á. LAGUNA, et al. *Integrando un modelo de reutilización en la producción de software: entorno distribuido para el desarrollo basado en reutilización*. Valladolid, España, Departamento de Informática, Universidad de Valladolid, 2004.
- CRNKOVIC, I. and M. LARSSON. *Building Reliable Component-Based Software Systems*. Artech House, 2002. p. 1-58053-327-2
- CUEVAS, D. B. *Técnicas avanzadas para el razonamiento automático sobre modelos de características: Aplicación a la ingeniería del dominio en líneas de producto software*.: ETSI Informática. Sevilla, España, Universidad de Sevilla, 2005. p.
- CHASTEK, G.; P. DONOHOE, et al. *A Study of Product Production in Software Product Lines*, Carnegie Mellon University, 2004.
- CHASTEK, G. J.; P. DONOHOE, et al. *Formulation of a Production Strategy for a Software Product Line*, Carnegie Mellon University, 2009.
- DENG, G.; D. C. SCHMIDT, et al. *Evolution in Model-Driven Software Product-line Architectures*. Nashville, USA, Department of EECS, Vanderbilt University, 2007.
- DÍAZ, Ó. *Líneas de Producto Software*. España, Facultad de Informática, Universidad del País Vasco, 2010.
- FULLER, T. Fundamento para los Pilares de las Fábricas de Software *The Architecture Journal*, 2006, 9.
- GIL, F. P. *Razonamiento Basado en Ontologías y Componentes Reutilizables para la Dirección Automática de Narración Digital e Interactiva*. Departamento de Sistemas Informáticos y Programación, Facultad de Informática. España, Universidad Complutense de Madrid, 2006. p.
- HURTADO, J. A. and C. BASTARRICA. *Hacia una Línea de Procesos Ágiles Agile SPsL*. Colombia, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, 2005.
- JAVIER MUÑOZ, V. P. *MDA vs Factorías de Software*. Dept. de Sistemas Informáticos y Computadores Universidad Politécnica de Valencia, 2005.
- JUSTO, J. L. B. *Estudio comparativo de técnicas para la clasificación/recuperación de componentes software reutilizables*. España, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Vigo, 2004.
- KÄKÖLÄ, T. and J. C. DUEÑAS. *Software Product Lines: Research Issues in Engineering and Management*. Springer, 2006. p. 9783540332527
- KRUEGER, C. W. *The 3-Tiered Methodology: Pragmatic Insights from New Generation Software Product Lines*. 11th International Software Product Line Conference. SOFTWARE, B. Austin, Texas, 2007.
- KULOOR, C. and A. EBERLEIN. *Requirements Engineering for Software Product Lines*. Calgary, Alberta, Canadá, 2002.

- LAMANCHA, B. P. and M. P. USAOLA. *Pruebas basadas en modelos para Líneas de Producto Software. Jornadas de Ingeniería del Software y Bases de Datos*. Ciudad Real, España, Grupo Alarcos, Departamento de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, 2009.
- LEE, K.; K. C. KANG, *et al.* *Concepts and Guidelines of Feature Modeling for Product Line Software Engineering*. Pohang, Korea, Department of Computer Science and Engineering, Pohang University of Science and Technology, 2003.
- MONTILVA, J. *Desarrollo de Software Basado en Líneas de Productos de Software. IEEE Computer Society Región 9 Capítulo Argentina Programa DVP*, 2006.
- POHL, K.; G. BÖCKLE, *et al.* *Software Product Line Engineering: Foundations, Principles, and Techniques*. Alemania, Springer-Verlag, 2005. p. 10 3-540-24372-0; 13 978-3-540-24372-4
- ROJAS, M. A. and J. C. M. GARCÍA. *Introducción y principios básicos del desarrollo de software basado en componentes*. Murcia, España, 2004.
- ROSELLÓ, E. G.; J. G. DACOSTA, *et al.* Una propuesta para la reutilización de componentes en el proceso de desarrollo de software educativo, 2002.
- SCHEKKERMAN, J. *Extended Enterprise Architecture Framework Essentials Guide*. Holland, Institute for Enterprise Architecture Developments, 2006.
- SNOOK, C.; M. POPPLETON, *et al.* *Rigorous engineering of product-line requirements: a case study in failure management*. Highfield, Reino Unido, School of Electronics and Computer Science, University of Southampton, 2007.
- SOUSA, P. G. D. *Software Product Lines*, 2008.
- SPRINGER, J. A.; R. A. MARTIN, *et al.* *Architectural Principles for Enterprise Frameworks*. Bloomington, Indiana, USA, Computer Science Department, Indiana University, 2004.
- THE-REUSE-COMPANY. *Presente y Futuro de la Reutilización de Software*. Madrid, España, The Reuse Company, 2006.
- VRIES, M. D. Medidas del Nivel de Éxito con Fábricas de Software *The Architecture Journal*, 2006, 9.

ANEXOS

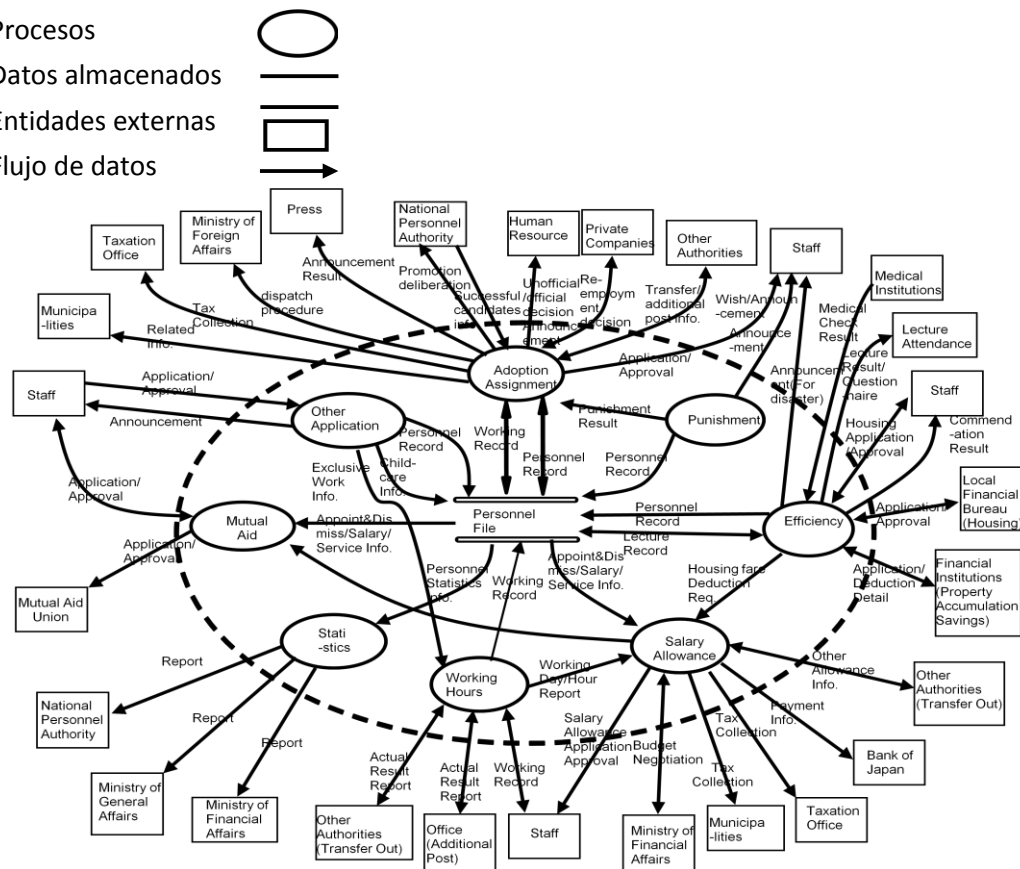
Anexo 1



Anexo 1 Ejemplo de un diagrama diamante de mandala.

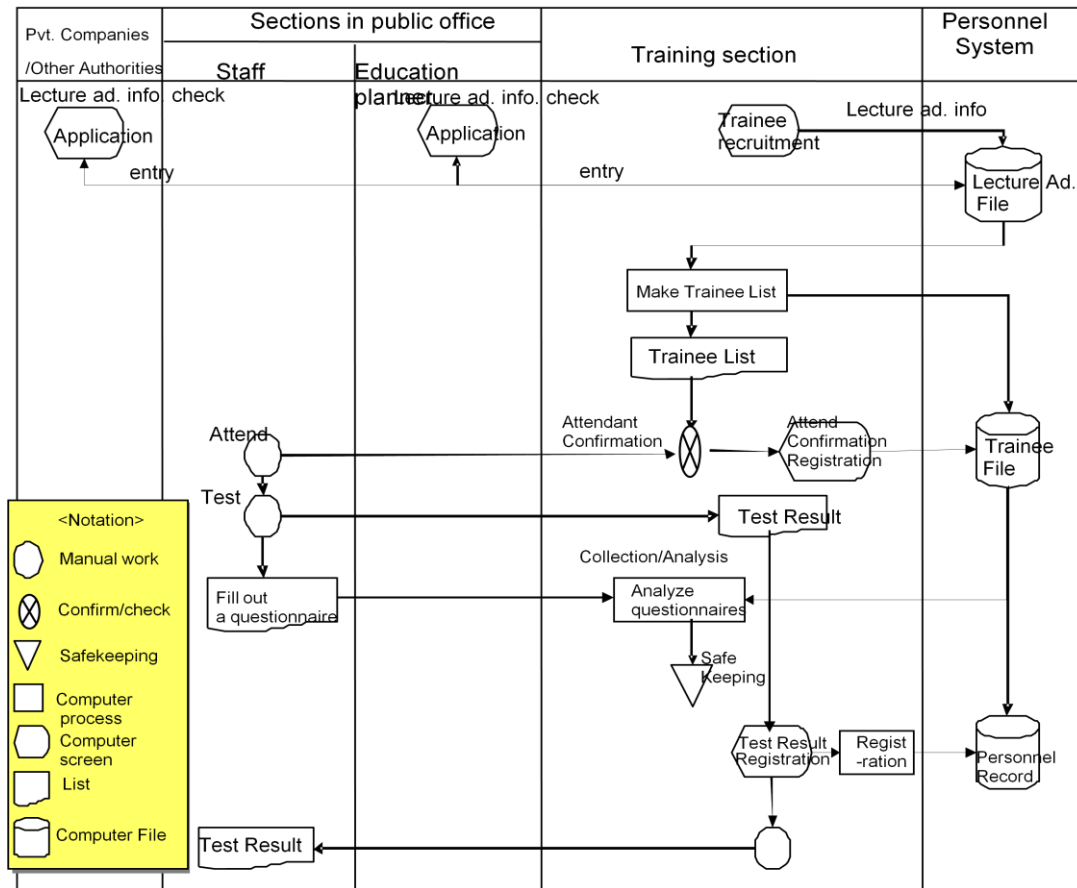
Anexo 2

- Procesos
- Datos almacenados
- Entidades externas
- Flujo de datos



Anexo 2 Ejemplo de Diagrama de flujo de datos.

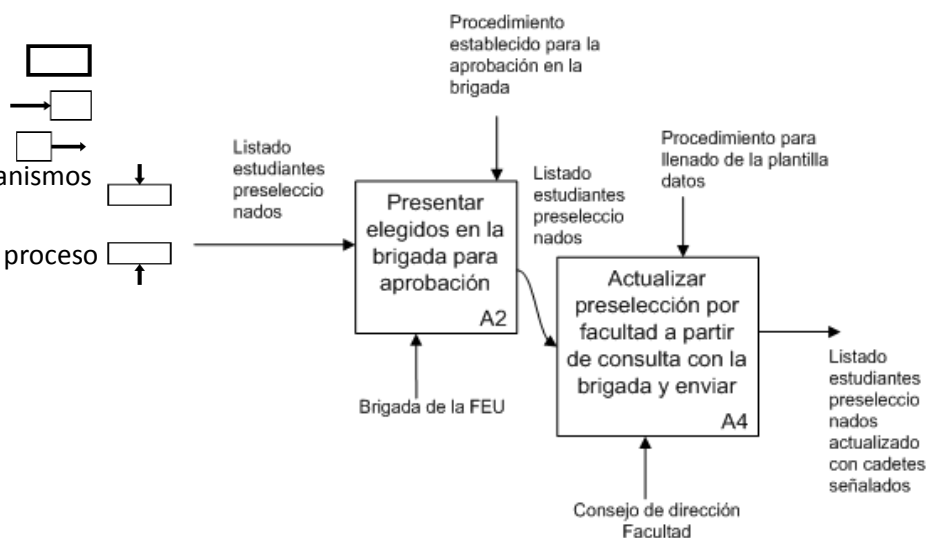
Anexo 3



Anexo 4 Ejemplo de diagrama de flujo de trabajo (WFA)

Anexo 4

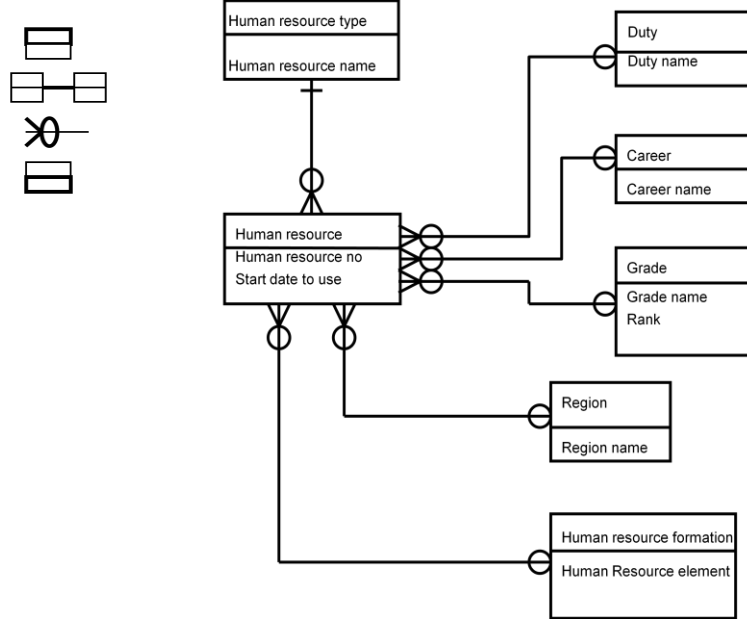
- Actividades
- Entradas
- Salidas
- Control y mecanismos regulatorios
- Ejecutores del proceso



Anexo 3 Ejemplo de diagrama IDEF0.

Anexo 5

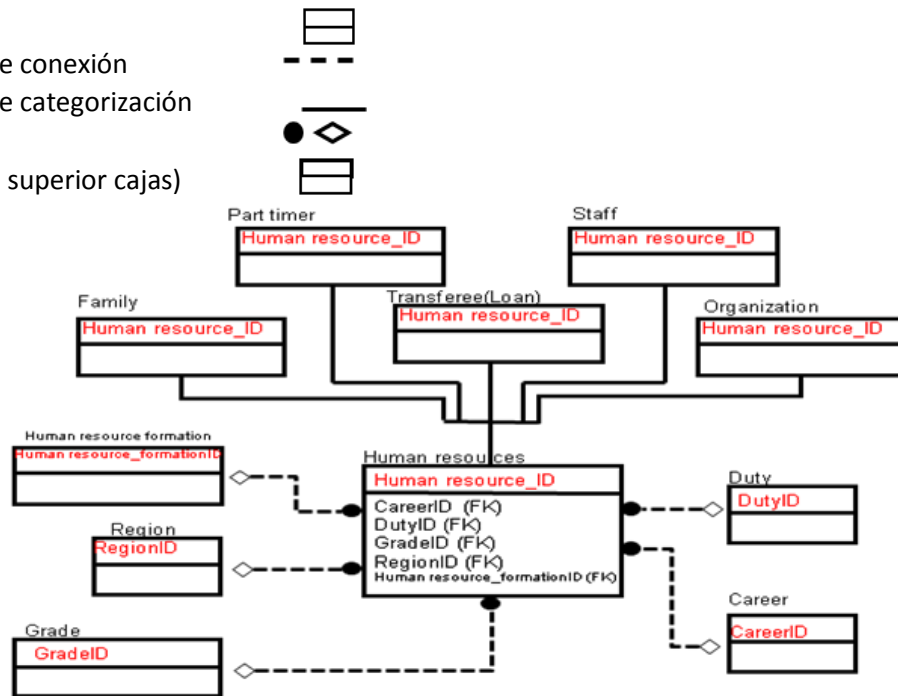
- Entidades
- Relaciones
- Cardinalidad
- Atributos



Anexo 5 Ejemplo de ERD (Diagrama Entidad Relación)

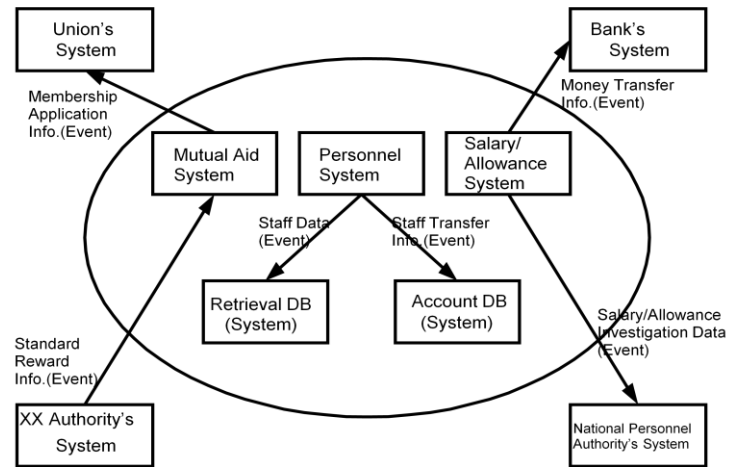
Anexo 6

- Entidades
- Relaciones de conexión
- Relaciones de categorización
- Cardinalidad
- Llaves (Parte superior cajas)



Anexo 6 Ejemplo de IDEF1 (Definición integrada para modelos de información)

Anexo 7



Anexo 7 Diagrama de relaciones entre sistemas de información.

Sistemas

Información entre sistemas

Dirección de la información



A

