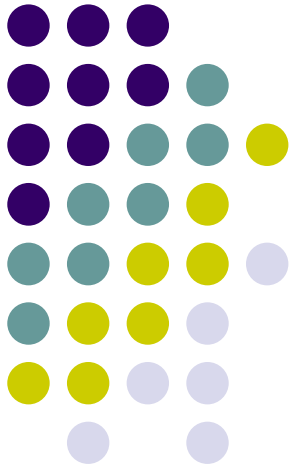
IEEE Computer Society Región 9
Capítulo Argentina
Programa DVP

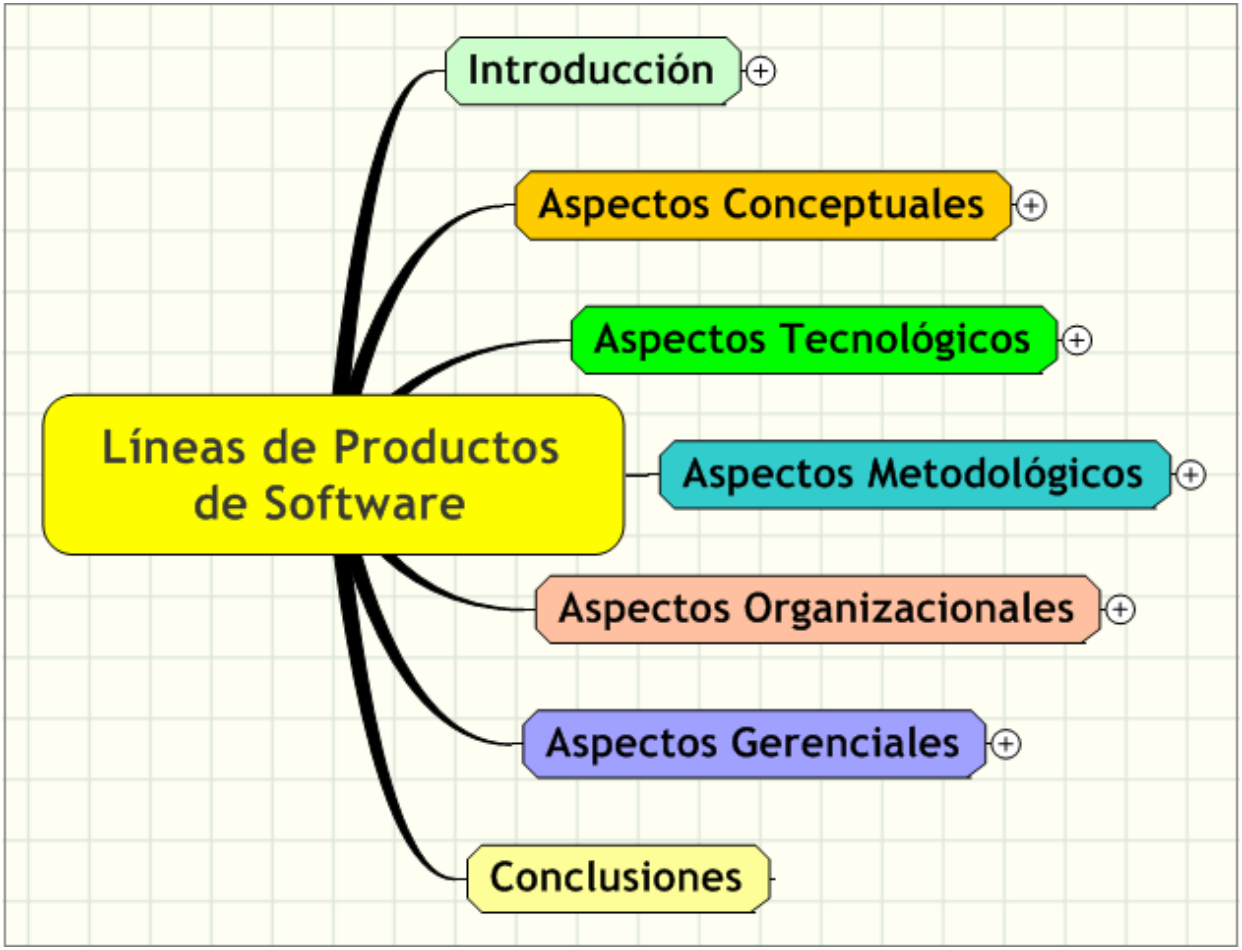
Desarrollo de Software Basado en Líneas de Productos de Software

Jonás A. Montilva C., Ph.D.
IEEE Member

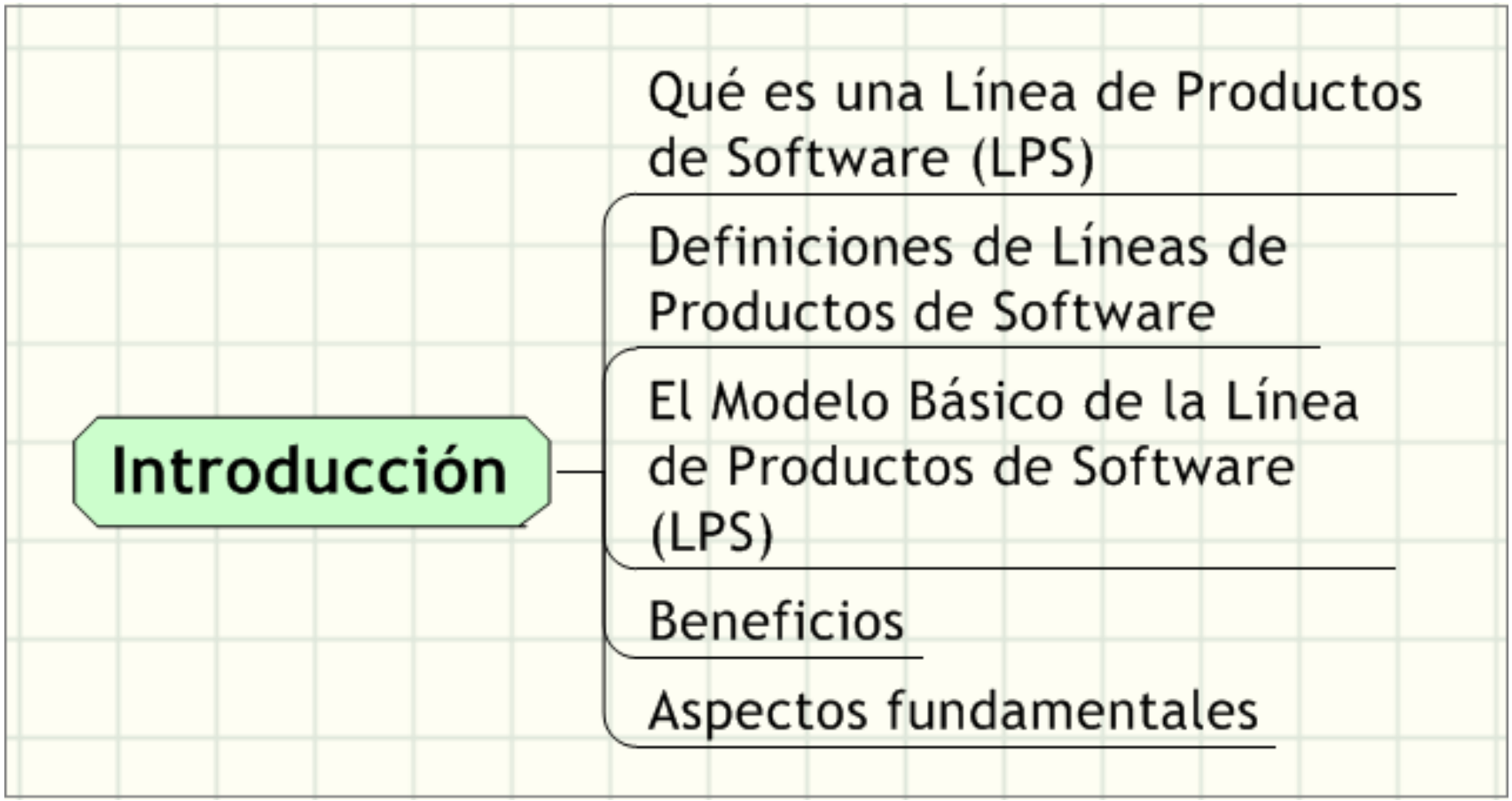
Universidad de Los Andes
Facultad de Ingeniería
Departamento de Computación
Mérida – Venezuela



Contenidos

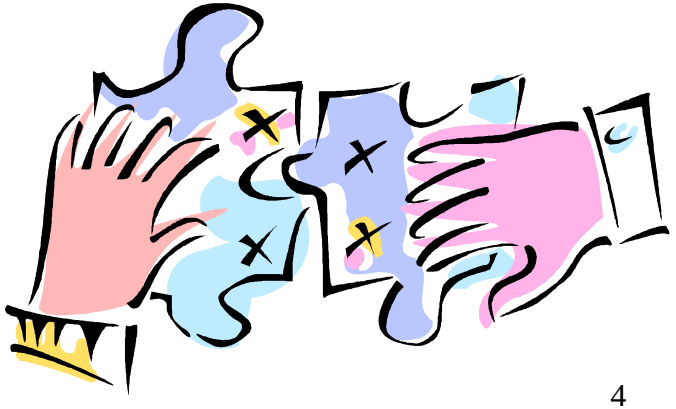

Líneas de Productos de Software
Introducción
Aspectos Conceptuales
Aspectos Tecnológicos
Aspectos Metodológicos
Aspectos Organizacionales
Aspectos Gerenciales
Conclusiones

Desarrollo de Software Basado en Línea de Productos



Qué es una Línea de Productos de Software (LPS)

- La idea básica:
  - Ensamblaje de partes de software previamente elaboradas
  - Inspirada en los procesos de producción de sistemas físicos
    - Producción de aviones, vehículos, computadores, aparatos electrónicos, etc.
  - Fundamentada en la Reutilización de Software
  - Asume la existencia de una industria de partes



Antecedentes

- Reutilización de software
  - “La reutilización de software es el proceso de implementar o actualizar sistemas de software usando activos de software existentes”

    (Sodhi & Sodhi, 1999)

  - "Reutilización de software es el proceso de crear sistemas de software a partir de software existente, en lugar de desarrollarlo desde el comienzo"

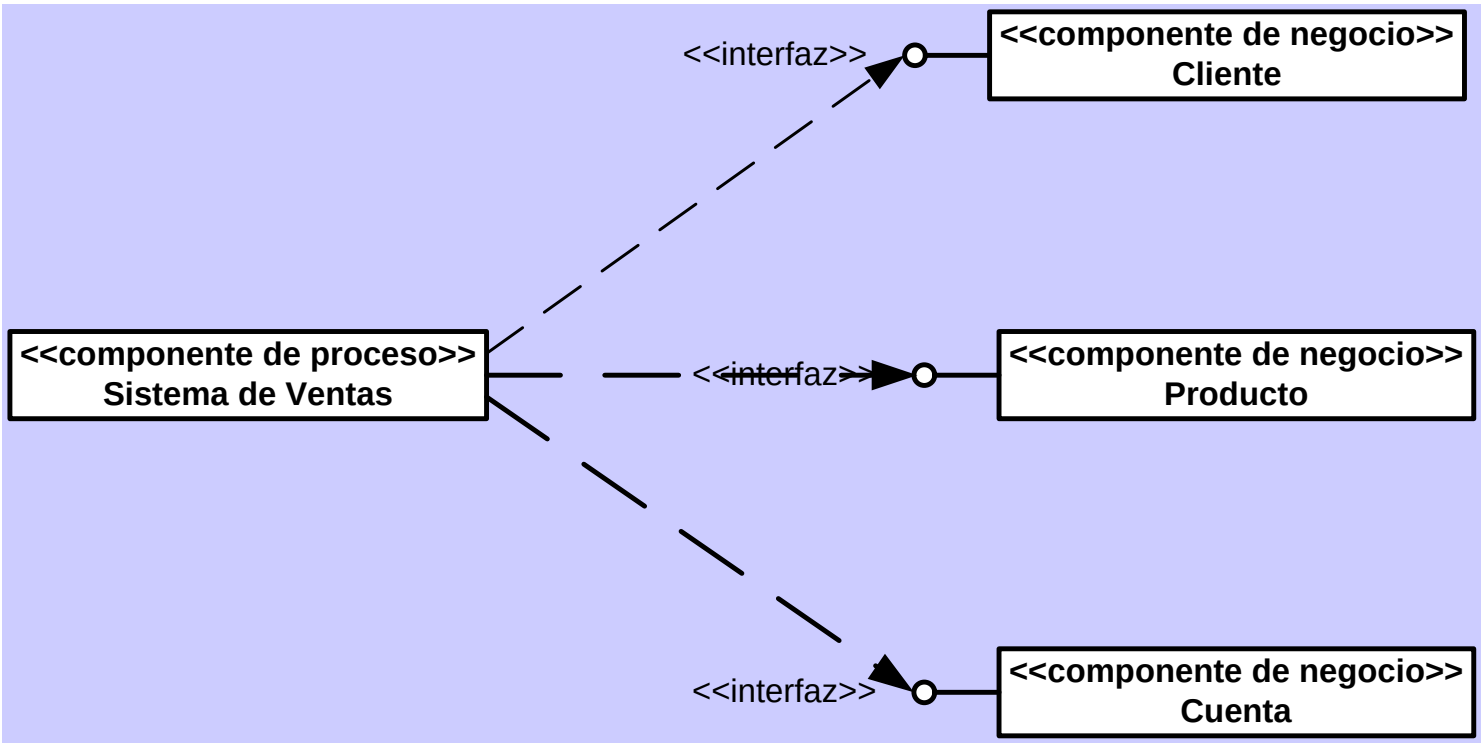
    (Sametinger, 1997)

Antecedentes

- Existen varias modalidades de reutilización utilizadas en empresas de software:
  - Individual
  - Oportunista
  - Gestionada:
    - Institucionalizada, sistemática, planificada, mejorada
- Tradicionalmente, la reutilización ha estado basada en oportunidad
  - Los componentes se almacenan en un repositorio a la espera de una oportunidad de reutilización

Antecedentes

- Desarrollo de Software Basado en Componentes
  - Las aplicaciones se crean mediante la integración de componentes nuevos, legados o de terceros (COTS)



Definiciones de Líneas de Productos de Software

- "...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)

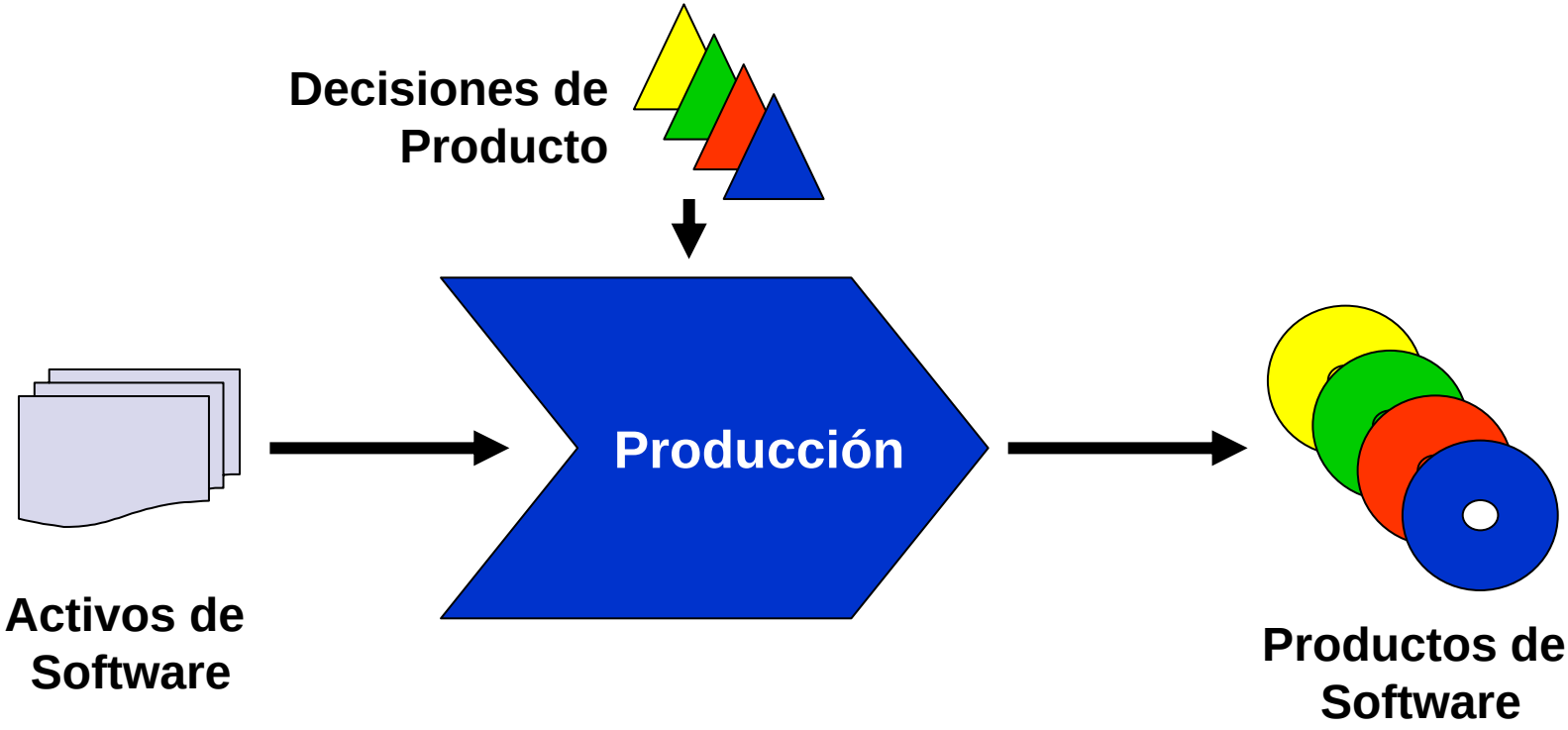
- "... es un conjunto de sistemas de software que comparten un conjunto común y gestionado de aspectos que satisfacen las necesidades específicas de un segmento de mercado o misión y que son desarrollados a partir de un conjunto común de activos fundamentales [de software] de una manera preescrita" (Clements and Northrop, 2002)

- "...consiste de una familia de sistemas de software que tienen una funcionalidad común y alguna funcionalidad variable" (Gomma, 2004)
  - La funcionalidad común descansa en el uso recurrente de un conjunto común de activos reutilizables (requisitos, diseños, componentes, servicios web, etc.)
  - Los activos son reutilizados por todos los miembros de la familia

Líneas de Productos de Software (LPS)

- Modelo Básico de una Línea de Productos de Software (LPS)

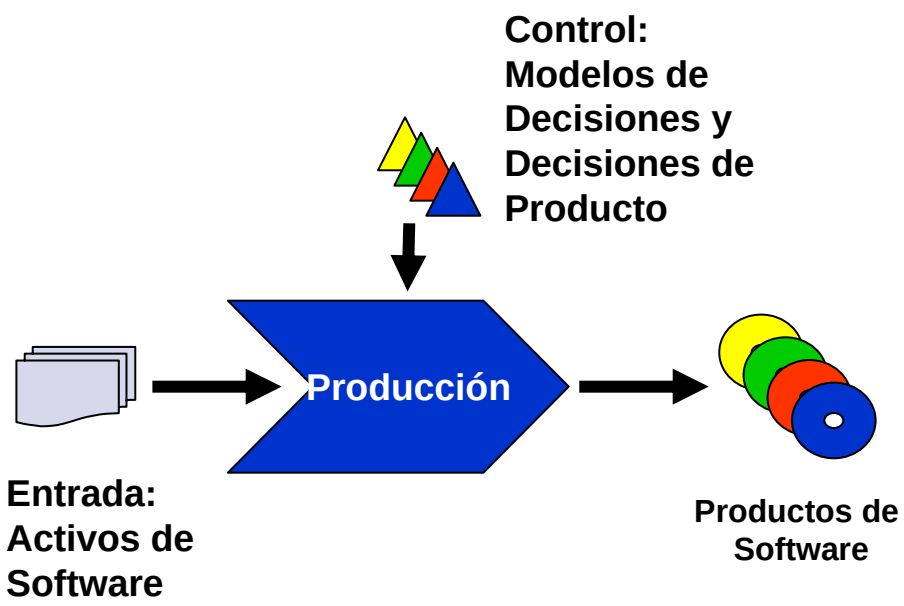
www.softwareproductlines.com



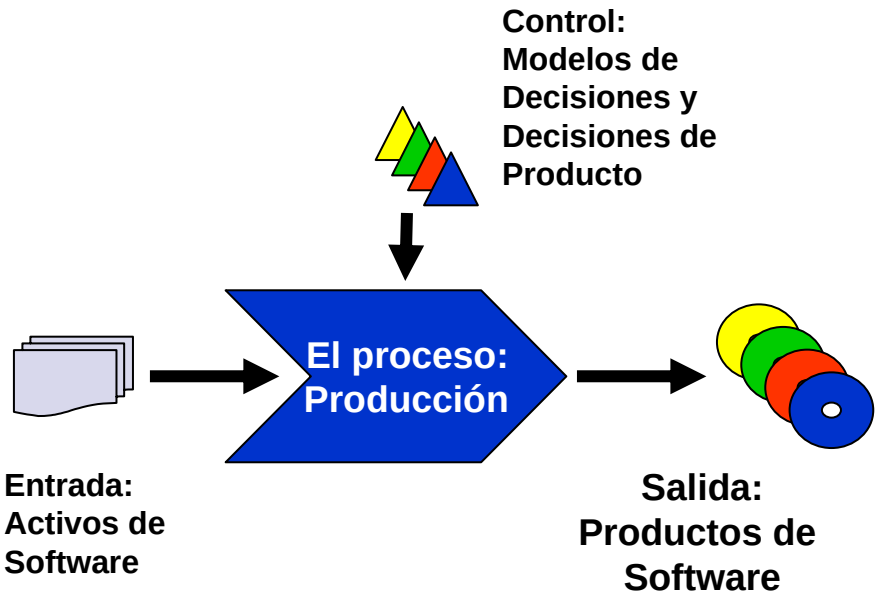
Modelo Básico de una Línea de Productos de Software

- **La entrada: Activos de Software**
  - Una colección de partes de software (requisitos, diseños, componentes, casos de prueba, etc.) que se configuran y componen de una manera prescrita para producir los productos de la línea

- **El control: Modelos de Decisión y Decisiones de Productos**
  - Los Modelos de Decisiones describen los aspectos variables y opcionales de los productos de la línea
  - Cada producto de la línea es definido por un conjunto de decisiones (decisiones del producto)



Modelo Básico de una Línea de Productos de Software



- **El proceso de producción**
  - Establece los mecanismos o pasos para componer y configurar productos a partir de los activos de entrada
  - Las decisiones del producto se usan para determinar que activos de entrada utilizar y como configurar los puntos de variación de esos activos
- **La salida: Productos de software**
  - Conjunto de todos los productos que pueden o son producidos por la línea de productos

Beneficios

- La entrega de productos de software de una manera
  - más rápida,
  - económica y
  - con una mejor calidad

- Las LPS producen mejoras en:
  - Tiempo de entrega del producto (*time to market*)
  - Costos de ingeniería
  - Tamaño del portafolio de productos
  - Reducción de las tasas de defectos
  - Calidad de los productos

Beneficios

- Beneficios tácticos y estratégicos (Krueger, 2006):
  - Beneficios tácticos de ingeniería:
    - Reducción en el tiempo promedio de creación y entrega de nuevos productos
    - Reducción en el número promedio de defectos por producto
    - Reducción en el esfuerzo promedio requerido para desarrollar y mantener los productos
    - Reducción en el costo promedio de producción de los productos
    - Incremento en el número total de productos que pueden ser efectivamente desplegados y mantenidos

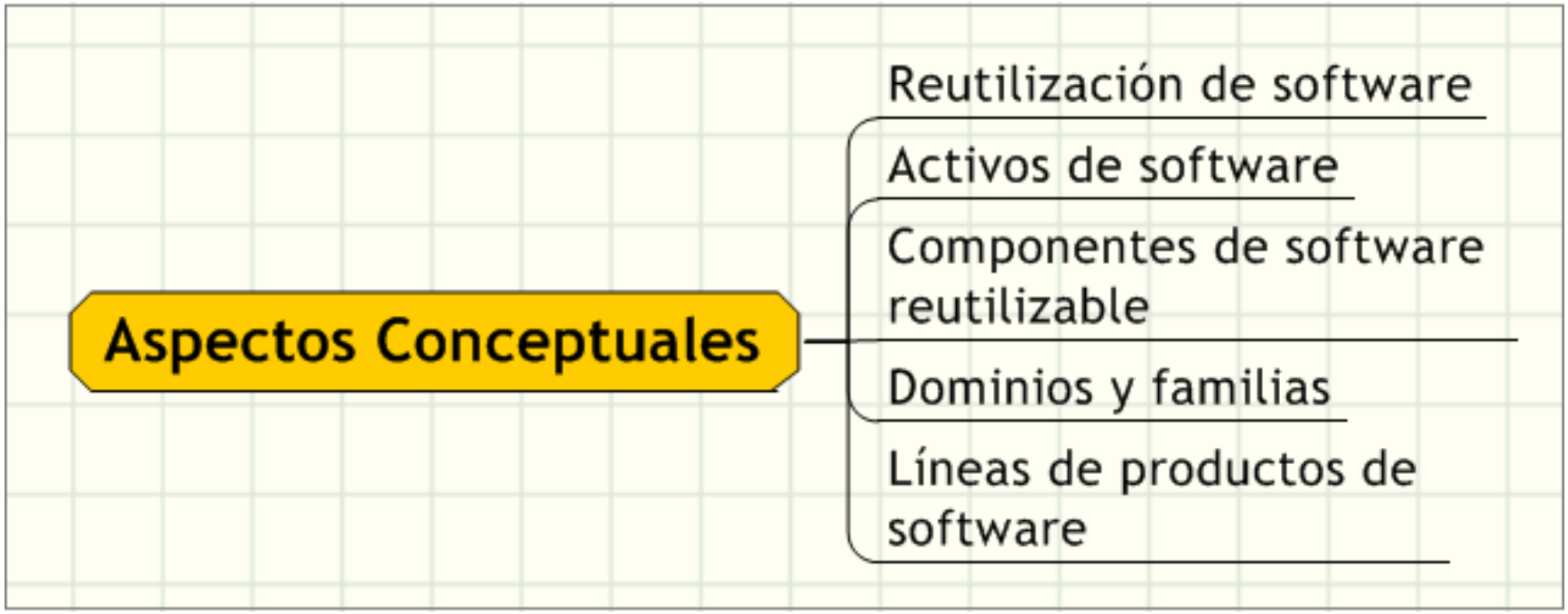
Beneficios

- Beneficios tácticos y estratégicos (cont.):
  - Beneficios estratégicos de negocios
    - Reducción en el tiempo de entrega (*time-to-market*) y el tiempo de retorno (*time-to-revenue*) de nuevos productos
    - Mejoras en el valor competitivo del producto
    - Márgenes mayores de ganancias
    - Mejor calidad de los productos
    - Mejoras en la reputación de la empresa
    - Mayor escalabilidad del modelo de negocios en términos de productos y mercados
    - Mayor agilidad para expandir el negocio a nuevos mercados
    - Reducción de riesgos en la entrega de productos
  - Algunas empresas han reportado mejoras que van en el rango de factores de 3 a 50 en los beneficios discutidos anteriormente

Aspectos fundamentales

- El paradigma de desarrollo de software LPS requiere que las empresas que lo adopten consideren:
  - **Aspectos conceptuales**
    - Conceptos en los que las LPS se fundamentan
  - **Aspectos tecnológicos**
    - Qué tecnologías son fundamentales para desarrollar y mantener activos y productos de software
  - **Aspectos metodológicos**
    - Cómo desarrollar y mantener los activos y productos de software
  - **Aspectos organizativos**
    - Cómo debe la empresa organizarse internamente
  - **Aspectos gerenciales**
    - Cómo gestionar los proyectos de desarrollo de activos y productos

Desarrollo de Software Basado en Línea de Productos



Evolución de la Reutilización de Software

Reutilización de software

- La reutilización de activos de software en LPS tiene varias características:
  - Es **estratégica**
    - Consolida lo común entre la línea de productos
    - Maneja estratégicamente la variación entre los productos de la línea
    - Elimina la duplicación de esfuerzos de ingeniería
  - Es **predictiva**
    - La reutilización de activos se da en uno o más productos sobre una línea bien definida
    - Se reutilizan arquitecturas de software, en lugar de reutilizar componentes de manera oportunista
  - Es **gestionada**
    - Es sistemática, planificada, institucionalizada y mejorada

Activos de software reutilizable

- Un **activo de software reutilizable** es un producto de software diseñado expresamente para ser utilizado múltiples veces en el desarrollo de diferentes sistemas o aplicaciones
- Un activo de software puede ser:
  - ✓ **Un componente de software**
  - ✓ **Una especificación de requisitos**
  - ✓ **Un modelo de negocios**
  - ✓ **Una especificación de diseño**
  - ✓ **Un algoritmo**
  - ✓ **Un patrón de diseño**
  - ✓ **Una arquitectura de dominio**
  - ✓ **Un esquema de base de datos**
  - ✓ **Una especificación de prueba**
  - ✓ **La documentación de un sistema**
  - ✓ **Un plan**

Componentes de software reutilizable

- Un **componente de software reutilizable** es
  - *"Una pieza [de software] funcional que es liberada independientemente [de otras] y que proporciona acceso a sus servicios a través de sus interfaces"* [Brown, 2000]

- Puede ser liberado (desplegado e *instanciado*) independientemente de otros:

- Ofrece servicios a través de sus interfaces
  - Para utilizar su funcionalidad se emplean sus interfaces

Componentes de software reutilizable (CSR)

- Definición del CBDi Forum [1999]:

  "Un componente es una pieza de software que describe y/o libera un conjunto de servicios que son usados sólo a través de interfaces bien definidas"

- Características esenciales de un CSR:
  - Identificable
  - Autocontenido
  - Rastreable a través de su ciclo de desarrollo
  - Reemplazable por otro componente
  - Accesible solamente a través de su interfaz
  - Inmutabilidad de sus servicios
  - Documentación de sus servicios
  - Mantenido sistemáticamente

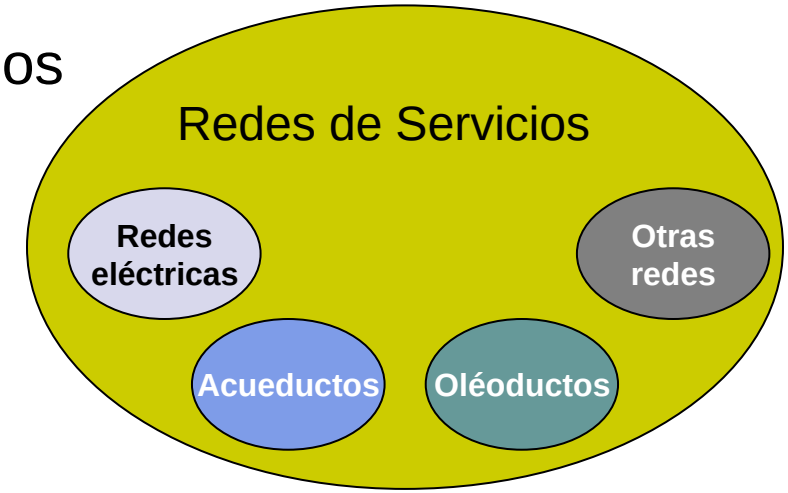
Componentes de software reutilizable (CSR)

Tipos de CSR

- Según su modificabilidad
  - Caja negra
  - Caja blanca
- Según su granularidad
  - Componentes de uso específico
  - Componentes de negocio
  - Marcos (*frameworks*)
  - Componentes de aplicación
- Según su fabricante
  - Componentes hechos en casa
  - COTS – Component Off The Shelf
- Según la tecnología usada
  - Componentes imperativos
    - Módulos, funciones
  - Componentes OO
    - Clases
  - Componentes distribuidos
    - Componentes CORBA
    - Componentes .NET
    - Componentes J2EE
    - Servicios web

Dominios y familias

- Un **dominio** es un área de aplicación de productos de software que:
  - están centradas en torno a un cuerpo de conocimientos
  - tienen una *economía de alcance* asociada
    - Ocurre cuando construir un activo y usarlo en múltiples productos ocasiona más beneficios que crear el activo para cada producto
  - Pueden dividirse en subdominios



Dominios y familias

- Una **familia de productos** de software es un conjunto de productos de software asociados a un dominio determinado
- Los miembros de la familia comparten aspectos comunes tales como:
  - un diseño arquitectónico común
  - un conjunto componentes reutilizables
  - capacidades y servicios comunes
  - tecnologías comunes

Líneas de productos de software (LPS)

- Una LPS es una familia de productos de software que:
  - tiene un conjunto de aspectos gestionados que son comunes a todos los miembros de la familia
  - los productos de la línea son desarrollados a partir de un conjunto de activos de software reutilizables

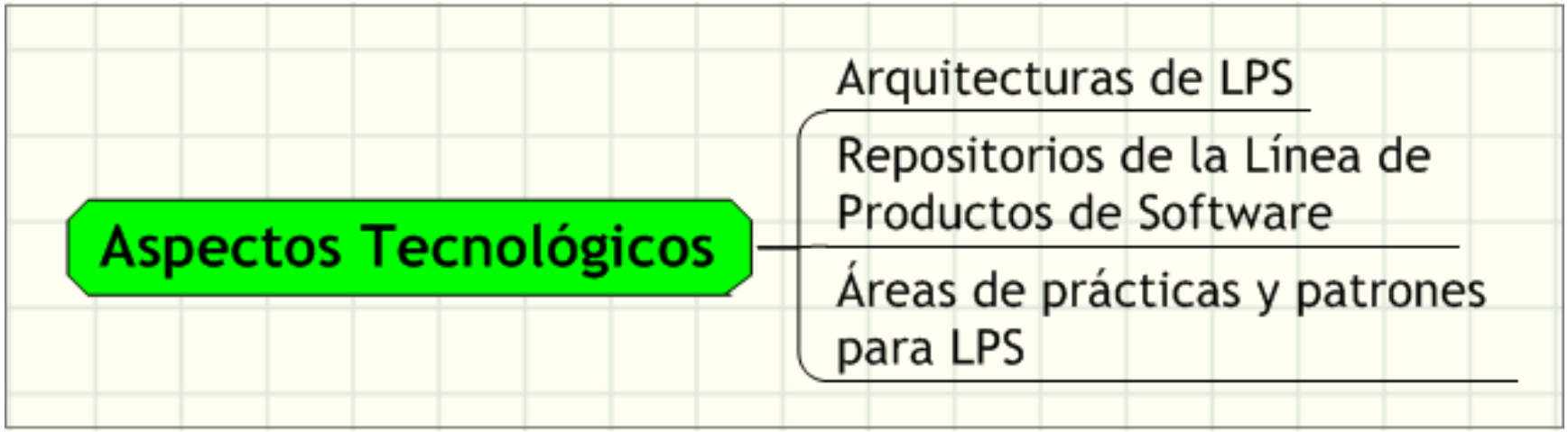
- Una familia de productos de software tiene:
  - **Aspectos comunes** que son compartidos por todos sus productos
  - **Aspectos variables** que establecen diferencias entre los productos

Líneas de productos de software (LPS)

- El objetivo principal de una LPS es:
  - “Reducir el tiempo, esfuerzo, costo y complejidad de crear y mantener los productos de la línea mediante:
    - La capitalización de los **aspectos comunes** de la línea de productos
      - A través de la consolidación y reutilización de los activos de entrada a la línea
    - El manejo de los **aspectos variables** de los productos de la línea
      - A través de los puntos de variación de los activos y los modelos de decisión”

(Krueger, 2006)

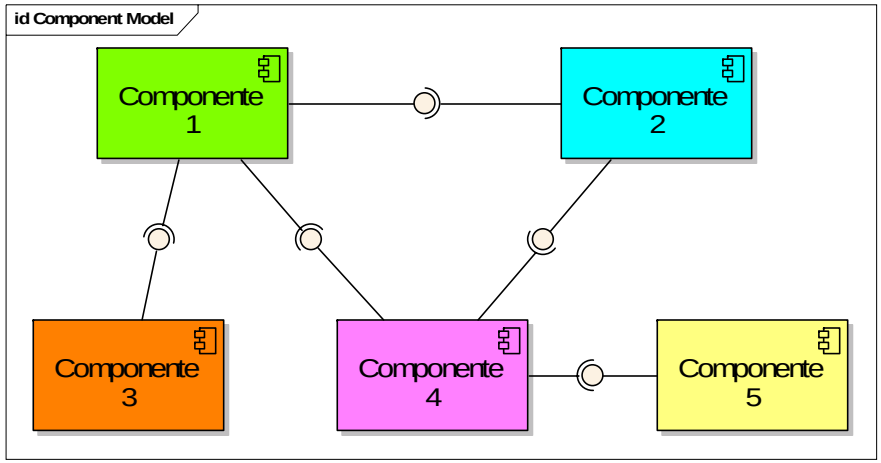
Desarrollo de Software Basado en Línea de Productos



Arquitecturas de LPS

- "Una arquitectura de software es la estructura o estructuras de un sistema que comprende los componentes del software, las propiedades visibles externamente de estos componentes, y las relaciones entre ellos"

  (Bass, 1998)



- Las propiedades externas de los componentes son:
  - sus interfaces (APIs) y
  - sus características
    - rendimiento, manipulación de errores, uso compartido de recursos, etc.

Arquitecturas de LPS

- La arquitectura de una LPS es una arquitectura de software genérica
  - Describe la estructura de toda la familia de productos y no solamente la de un producto particular
  - Captura los aspectos comunes y variables de una familia de productos de software
    - Los aspectos comunes de la arquitectura son capturados por los componentes de software que son comunes a toda la familia
    - Los aspectos variables de la arquitectura son capturados por los componentes de software que varían entre los miembros de la familia
  - También denominada arquitectura de dominio

- La arquitectura LPS debe ser instanciada cada vez que se desarrolla un producto de la línea

Arquitecturas de LPS

- **Una arquitectura LPS es instanciada a través de mecanismos de variabilidad:**
  - Herencia
    - Ej. Suplantación de un método heredado de una clase en un componente
  - Puntos de extensión
    - Ej. Se agrega nueva funcionalidad o comportamiento a un componente
  - Parametrización
    - El comportamiento de un componente puede ser parametrizado a tiempo de diseño y definido a tiempo de implementación
    - Ej. macros o templates
  - Configuración
    - Selección y "deselección" de los componentes de la arquitectura
  - Selección a tiempo de compilación
    - La implementación de una funcionalidad es seleccionada, entre varias posibles, al momento de la compilación del componente o de la aplicación

Repositorios LPS

- Las líneas de productos de software requieren almacenar sus activos de software en repositorios
- Un repositorio LPS es una base de datos especializada que:
  - almacena activos de software y
  - facilita la recuperación y el mantenimiento de los activos de software
- Su objetivo es asegurar la disponibilidad de activos para apoyar el desarrollo de productos de la LPS

Repositorios LPS

- El repositorio mantiene información relevante de cada activo usado en la LPS:
  - Especificación técnica del activo
  - Historia o registro de uso
  - Clasificación del activo
  - Documentación del activo

Repositorios LPS

Tipos de Repositorios LPS

- Según su alcance
  - Locales
    - Son desarrollados y reusados internamente por una organización o empresa
  - Globales o de uso comercial
    - Disponibles a terceros bajo adquisición o subscripción
    - Ejemplos: COTS, Servicios Web
- Según su aplicabilidad
  - De dominio específico
  - De dominio general
- Según su propósito
  - De reuso
    - Permiten el almacenamiento y recuperación de activos de software
  - De referencia
    - Facilitan la localización de activos en otros repositorios
    - Ejemplo: UDDIs

Áreas de prácticas y patrones para LPS

- La introducción del paradigma LPS en una empresa de software es un proceso complejo, gradual y lleno de dificultades

- Para obtener los beneficios que este paradigma ofrece, una empresa debe tomar en consideración diferentes factores:
  - tecnológicos,
  - metodológicos,
  - organizacionales y
  - gerenciales

- Clements y Northrop (2002) definen un conjunto de **áreas de prácticas** y **patrones**
  - Son esenciales considerar para asegurar el éxito de la implantación del paradigma LPS en una empresa

Áreas de Práctica LPS

- Un **área de práctica** es una colección de actividades que una empresa debe ejecutar y dominar para implantar exitosamente una LPS

- Estas áreas de práctica describen actividades que son normalmente recomendadas por el SEI para el desarrollo exitoso de software

- Guardan una correspondencia estrecha con las áreas de procesos definidas por el CMMI-SW

Áreas de Práctica LPS

- Tres tipos de áreas de prácticas LPS recomendadas por Clements y Northrop (2002):
  - Áreas de práctica de Ingeniería de Software
    - Ejemplos: Definición y evaluación de una arquitectura LPS
  - Áreas de práctica de Gestión Técnica
    - Ejemplo: Planificación de los proyectos de desarrollo de componentes ó de productos (aplicaciones)
  - Áreas de práctica de Gestión Organizacional
    - Ejemplo: Estructuración de la empresa

Patrones LPS

- Un **patrón** es una regla de tres partes, las cuales expresan una relación entre un contexto, un problema y una solución (Alexander, 1979)
- Los patrones LPS plantean soluciones a problemas recurrentes relacionados con las situaciones organizacionales de las LPS
- Las soluciones son planteadas en términos de las áreas de prácticas y sus relaciones
- Un ejemplo: El patrón "Que Construir"

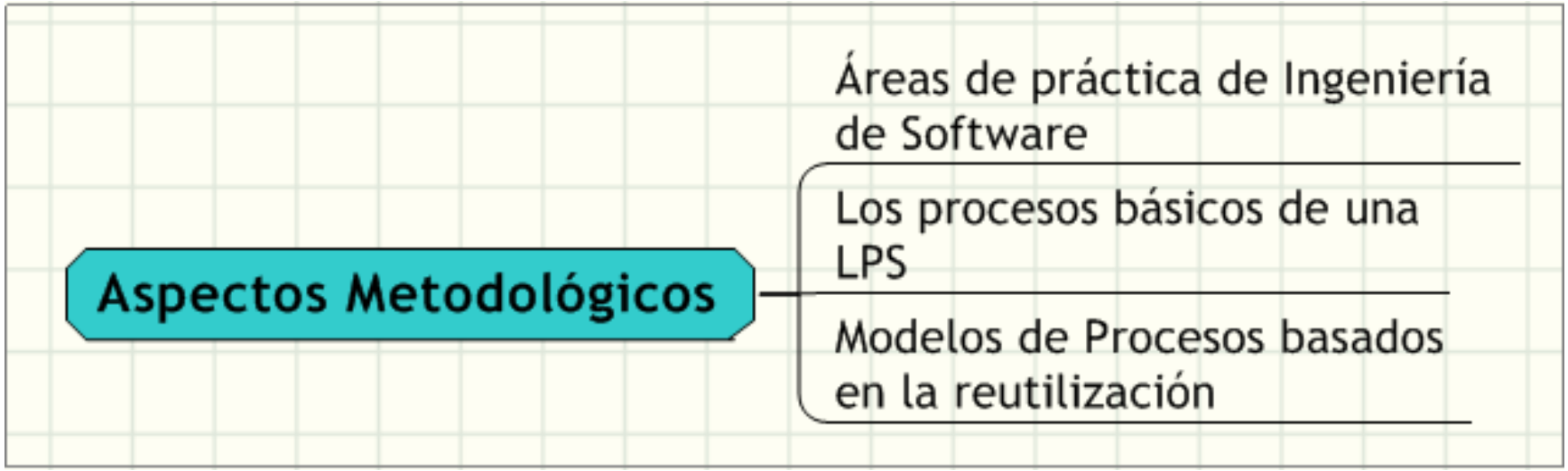
Un ejemplo: El patrón "Que Construir"

- **El Contexto:**
  - Una empresa ha decidido crear una línea de productos de software y conoce bien el dominio de aplicación de los productos
- **El Problema:**
  - Determinar que productos deberán ser incluidos en la línea de productos
- **La Solución:**
  - Para determinar que productos producir, se requiere información relacionada con:
    - El dominio de aplicación, la tecnología y el mercado
    - La justificación del negocio
    - El proceso para describir el conjunto de productos que serán incluidos en la línea de productos

Un ejemplo: El patrón "Que Construir"

- **Las áreas de práctica requeridas por la solución:**
  - Análisis del Mercado
    - Ayuda a entender el mercado que tendrá los productos de la línea: qué productos tienen mayor demanda, cuál es la competencia, cuál es el tamaño del mercado y cuales las oportunidades
  - Entendimiento de dominios relevantes
    - Proporciona un modelo del dominio, los requisitos del dominio y los aspectos comunes y variables a todos los sistemas (aplicaciones) que forman el dominio
  - Proyección tecnológica
    - Permite predecir que productos que productos pueden llegar a ser factibles en el futuro cercano
  - Construcción de un caso de negocios
    - Proporciona una justificación de la selección de productos y del enfoque se usará para construirlos
  - Definición del alcance (*scoping*)
    - Describe cuales productos serán incluidos en la línea de productos y cuales no

Desarrollo de Software Basado en Línea de Productos



Áreas de Práctica de Ingeniería de Software

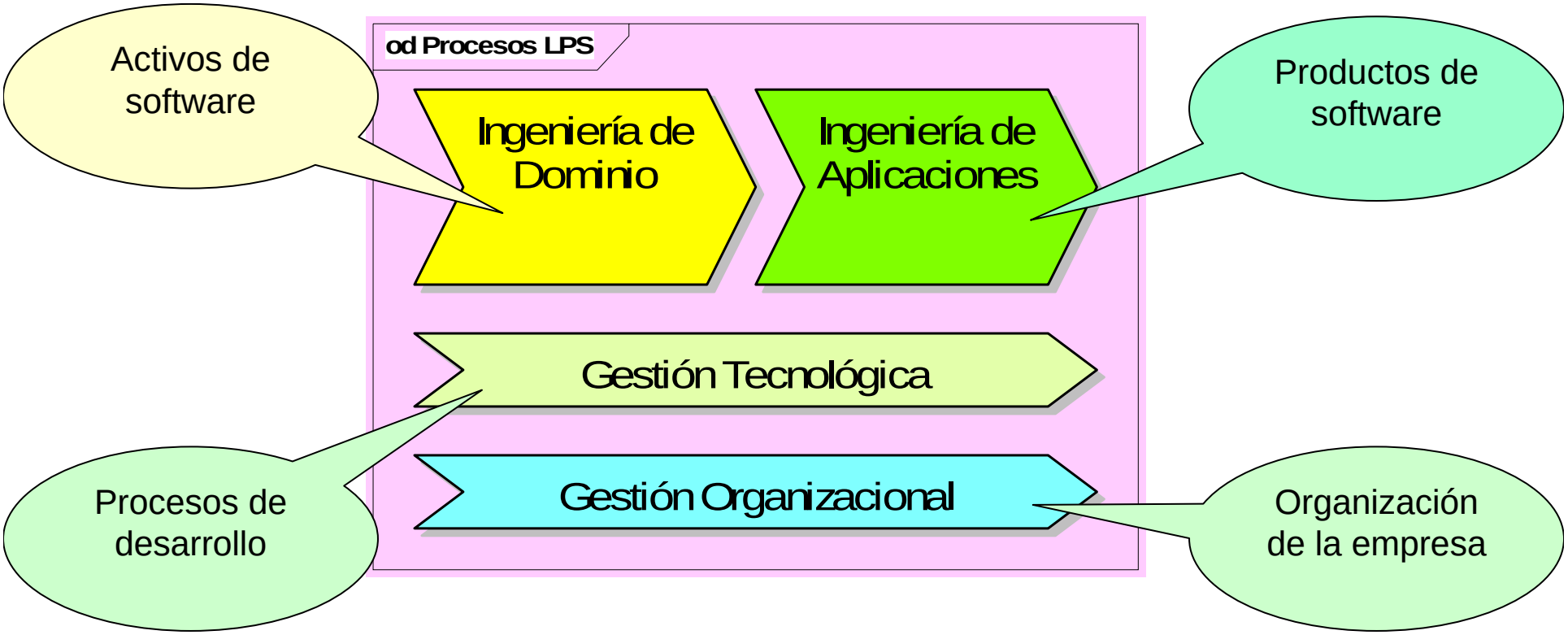
- Los aspectos metodológicos de las LPS involucran la aplicación de un conjunto de prácticas de ingeniería:

- Definición de la arquitectura LPS
- Evaluación de la arquitectura LPS
- Desarrollo de componentes
- Utilización de COTS
- Minería de activos existentes

- Ingeniería de Requisitos
- Integración de sistemas de software
- Pruebas
- Entendimiento de dominios relevantes

Los procesos básicos de una LPS

- Procesos de negocio de una LPS



Ingeniería de Dominio

- La Ingeniería de Dominio (ID) captura información y representa el conocimiento sobre un dominio determinado, con el fin de:
  - crear activos de software reutilizables en el desarrollo de cualquier nuevo producto de una LPS

- Productos de la ID:
  - Definiciones de dominios (descripciones del contexto)
  - Modelos del dominio
  - Modelos de requisitos del dominio
  - Modelos arquitectónicos (arquitecturas de dominio)
  - Ontologías del dominio
  - Lenguajes del dominio
  - Estándares del dominio

Ingeniería de Dominio

- Actividades principales de la Ingeniería de Dominio
  - Análisis de Aspectos:
    - Analiza la familia para determinar los requisitos que son comunes, opcionales y diferentes a todos sus miembros
  - Diseño de la Arquitectura LPS:
    - Produce una arquitectura de dominio la cual tiene:
      - Componentes comunes a todos los miembros de la familia
      - Componentes opcionales que son requeridos por algunos miembros
      - Componentes variantes de los cuales algunos miembros de la familia emplean distintas versiones
        - Tienen **puntos de variación** que permiten configurarlos
  - Implementación del Dominio
    - Consiste en la creación y almacenamiento de los activos de software que se emplearán para producir los productos de software

Ingeniería de Aplicaciones

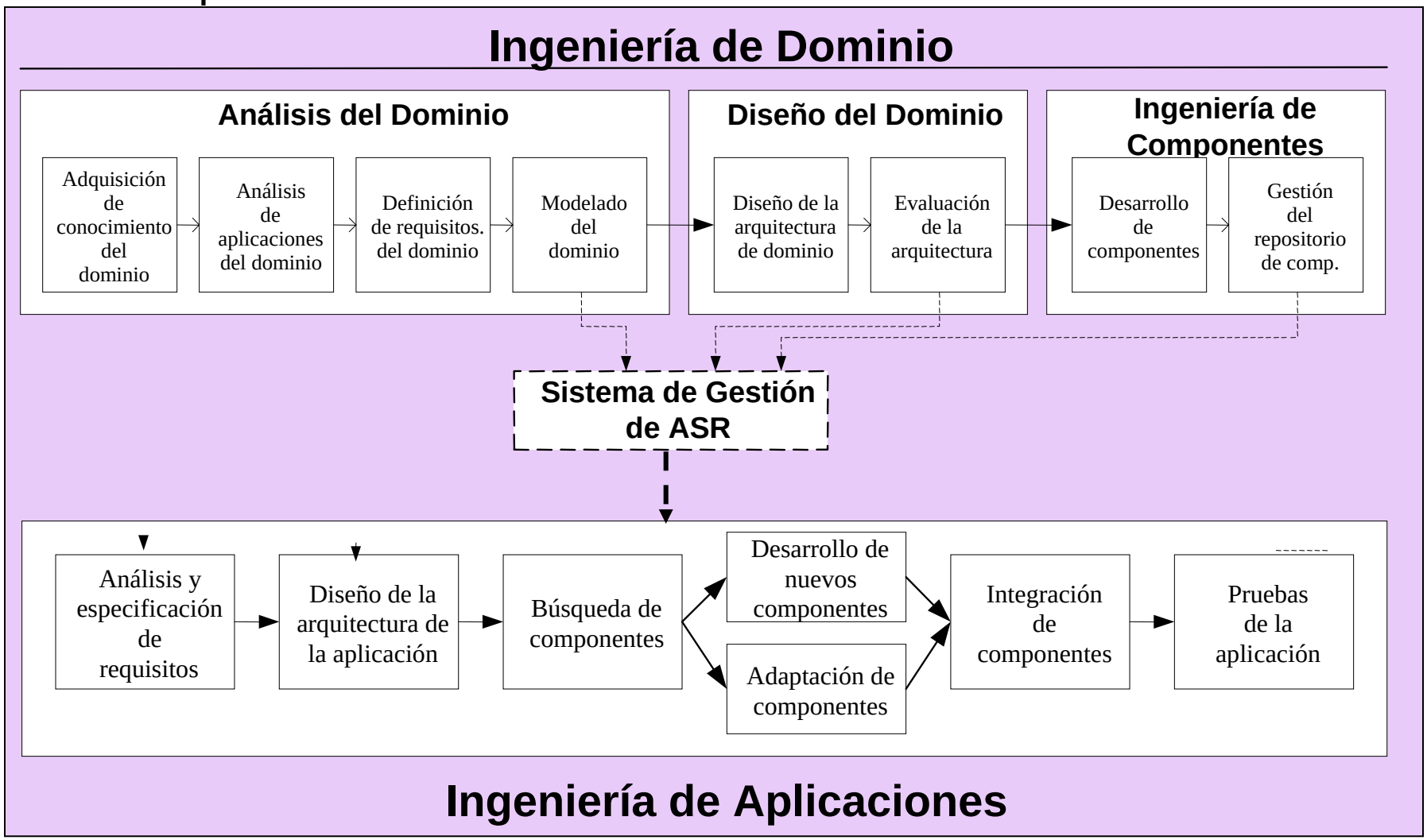
- La Ingeniería de Aplicaciones (IA) se encarga del desarrollo de los productos de la LPS a través de:
  - la reutilización de activos de software
  - planes de producción
- La arquitectura de dominio es empleada como un modelo de referencia para diseñar los productos de la LPS
- El repositorio LPS provee los activos requeridos durante el desarrollo de cada nuevo producto de la LPS

Modelos de procesos para LPS

- El Modelo TWIN
- El método WATCH
  - Modelo WATCH Component
  - Modelo WATCH App
- El modelo del Software Engineering Institute (SEI)
- El modelo ESPLEP
  - Evolutionary Software Product Lines Engineering Process

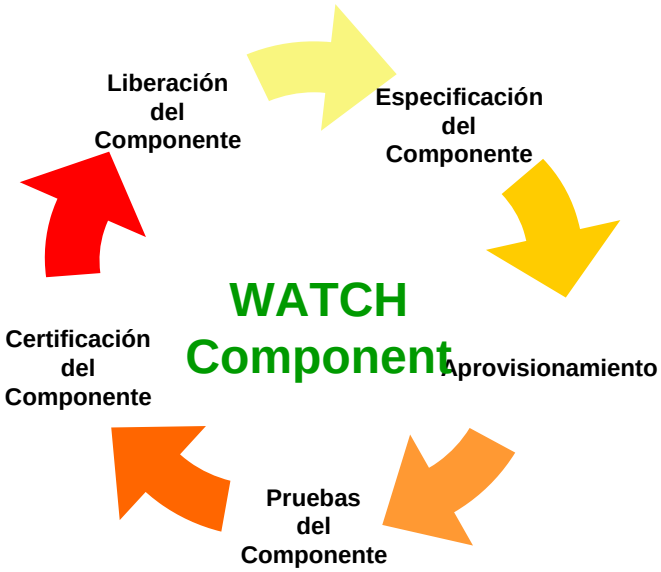
El Modelo TWIN extendido

- Modelo empleado en el Desarrollo de Software basado en Componentes



El Método WATCH

- Modelo propuesto en la Universidad de Los Andes (Venezuela) para el desarrollo de aplicaciones empresariales
- Consta de dos componentes metodológicos:

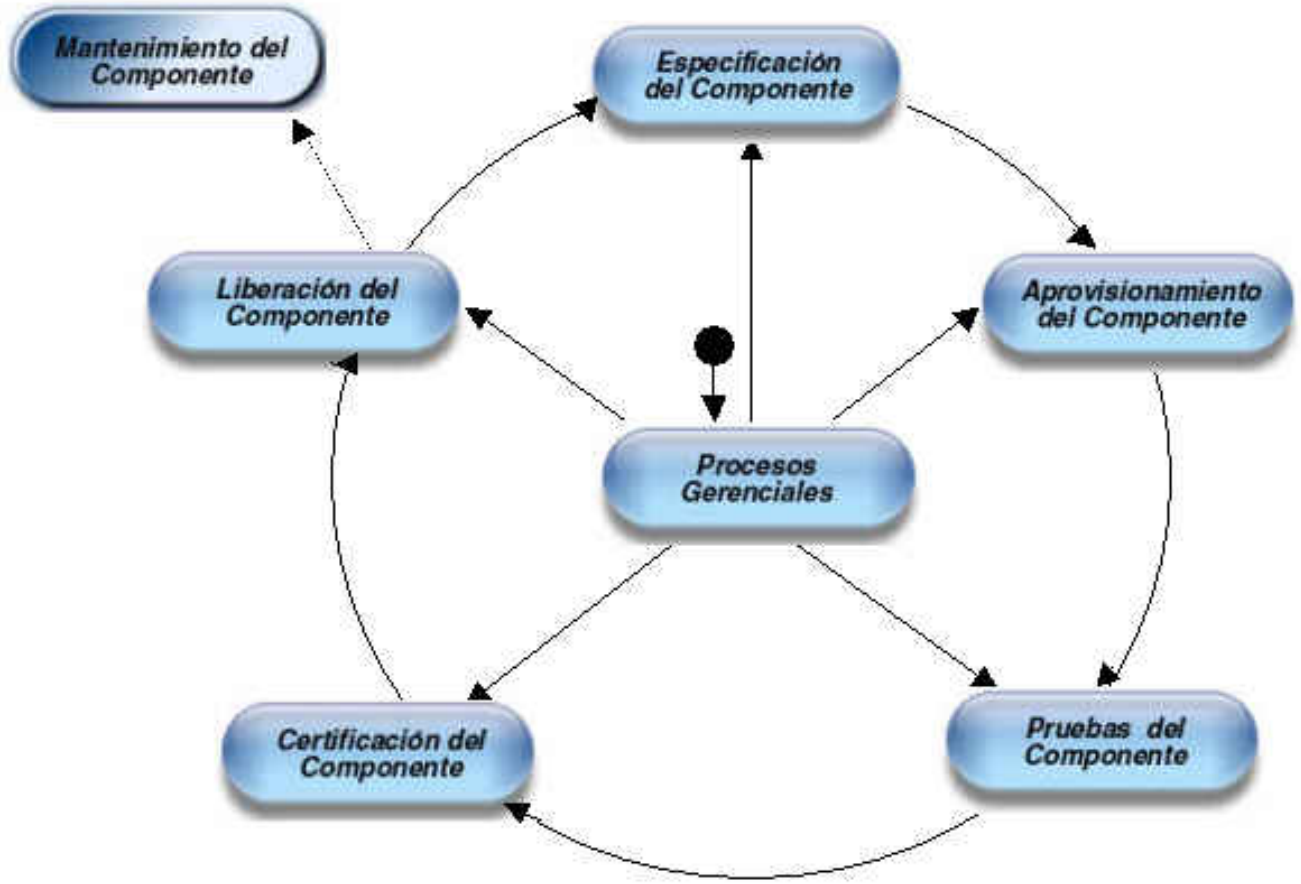


Ingeniería de Dominio: Desarrollo de Componentes

Ingeniería de Aplicaciones: Desarrollo de Aplicaciones Empresariales

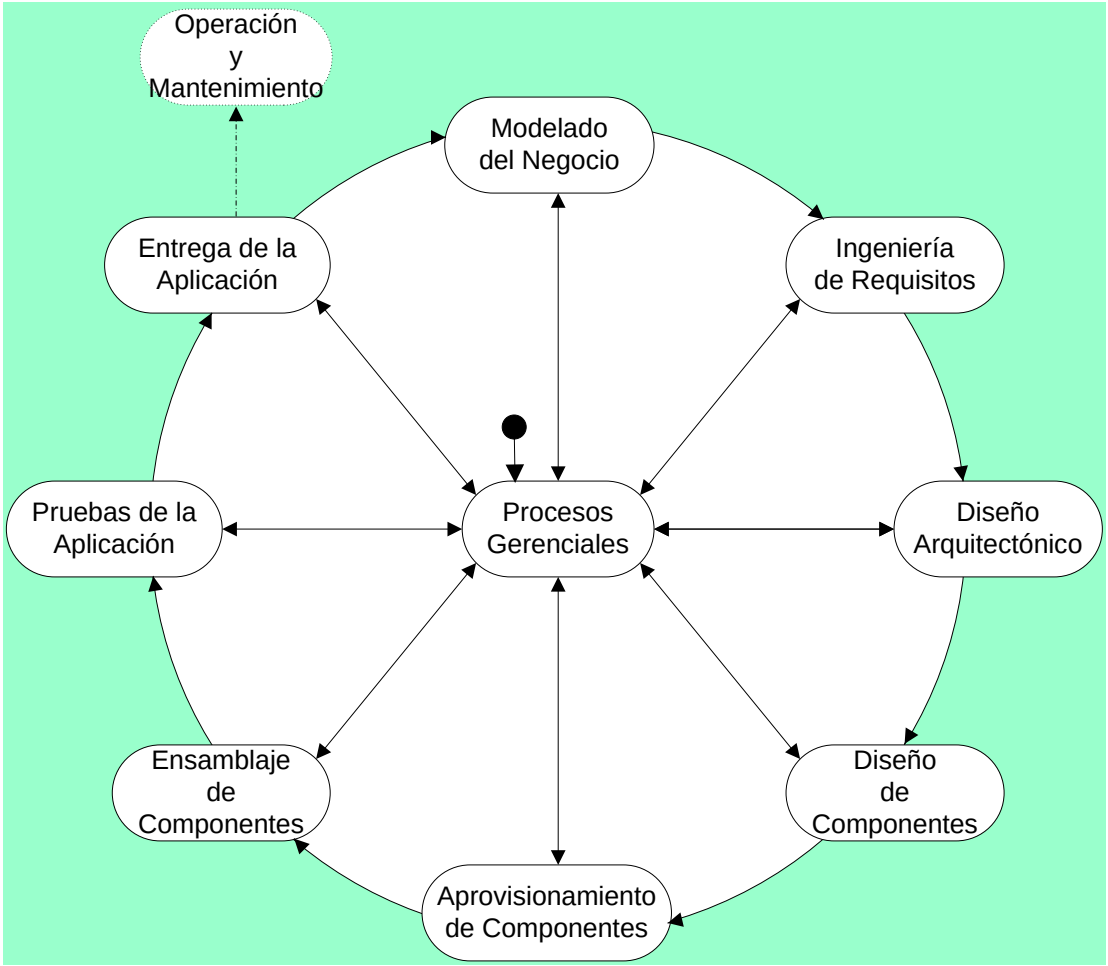
El Método WATCH-Component

- Modelo de procesos para el desarrollo de componentes de software reutilizables



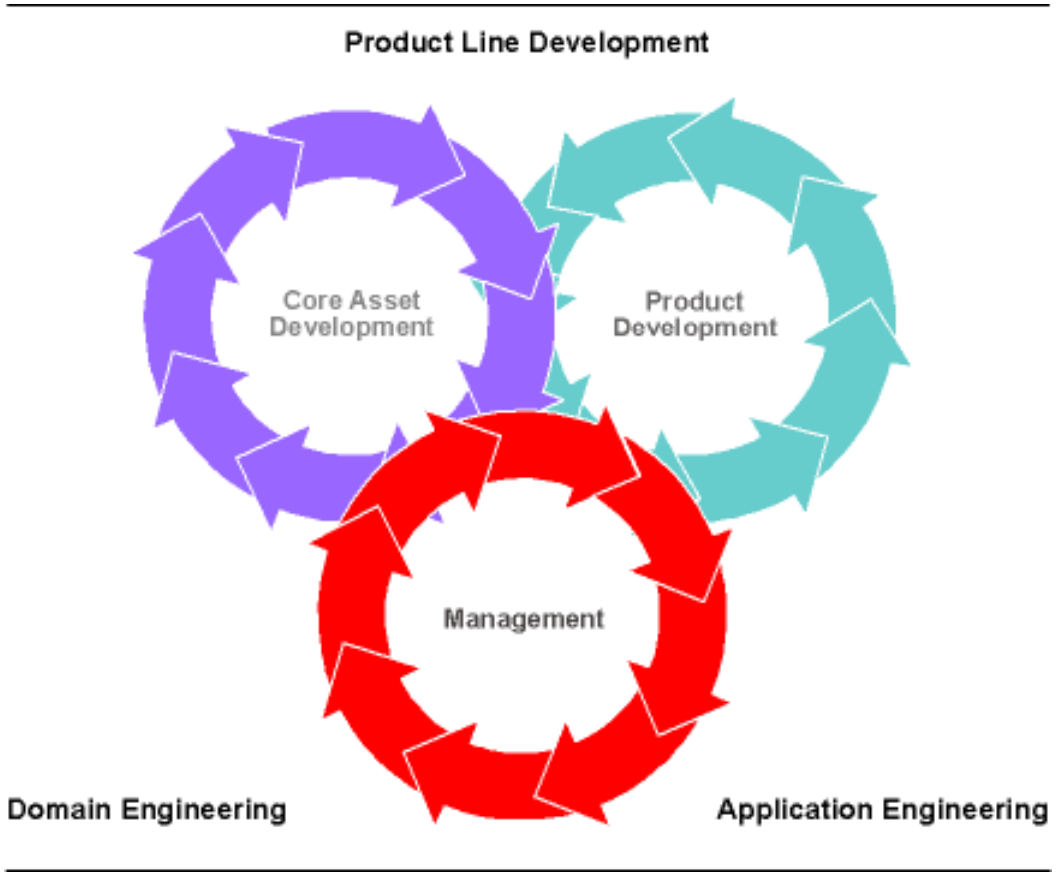
El método WATCH-Application

- Modelo de procesos para el desarrollo de aplicaciones empresariales (Montilva y Barrios, 2004)



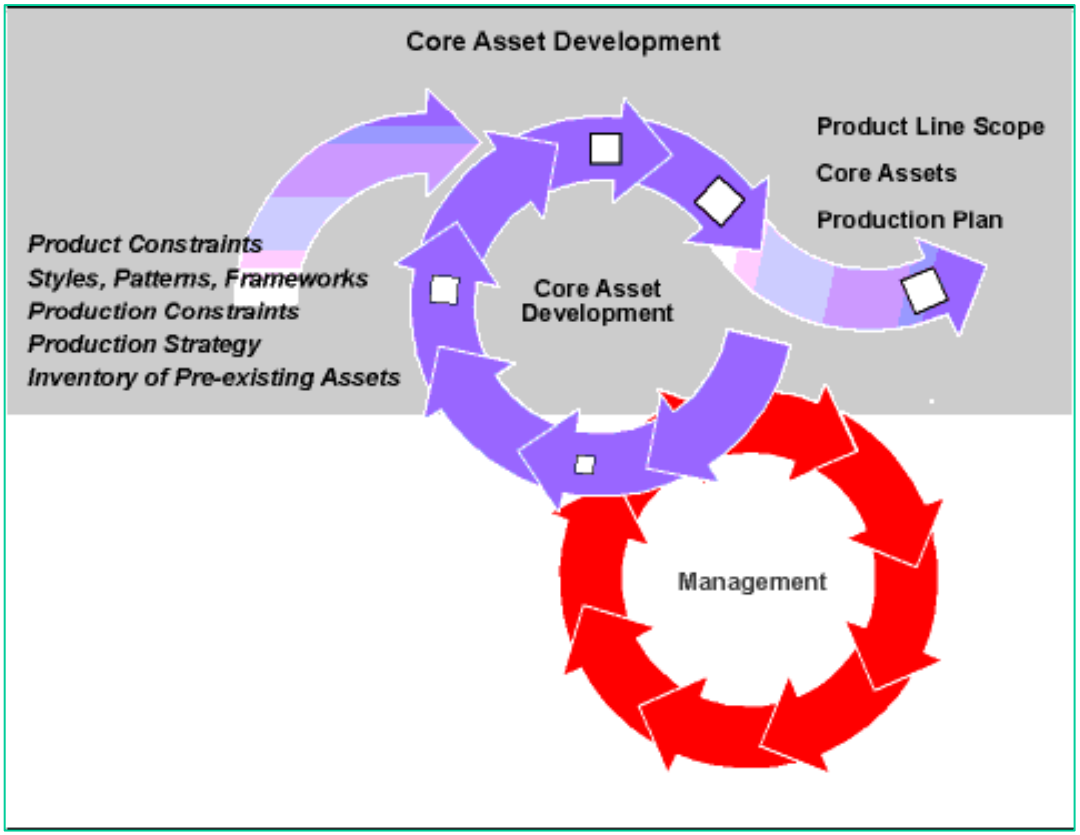
El modelo del SEI

- Modelo de procesos de LPS desarrollado en el Software Engineering Institute (SEI)
- Disponible en http://www.sei.cmu.edu/productlines/framework.html



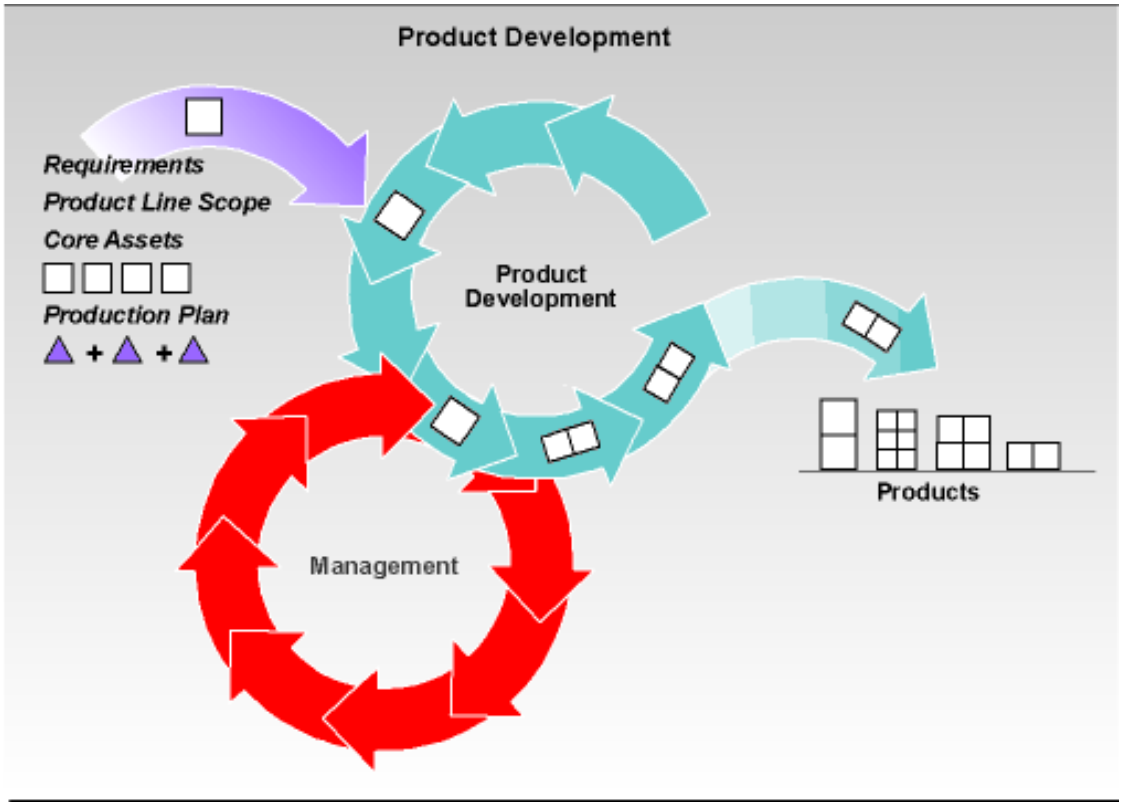
El modelo del SEI

- Desarrollo de Activos Fundamentales (Ingeniería de Dominio)
  - Objetivo:
    - Establecer la capacidad de producción para los productos mediante el desarrollo de activos de software reutilizables
  - Salidas:
    - Alcance de la línea
    - Activos
    - Plan de Producción



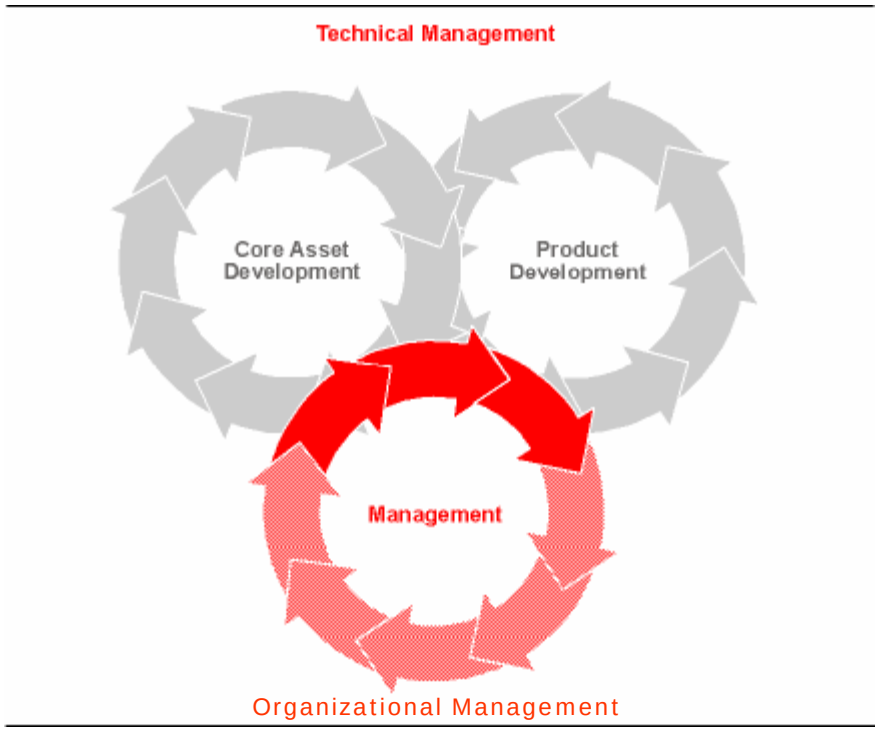
El modelo del SEI

- Desarrollo de Productos (Ingeniería de Aplicaciones)
  - Objetivo:
    - Elaborar los productos de la línea a partir del ensamblaje de activos fundamentales siguiendo el plan de producción
  - Salida:
    - Productos acabados de la línea



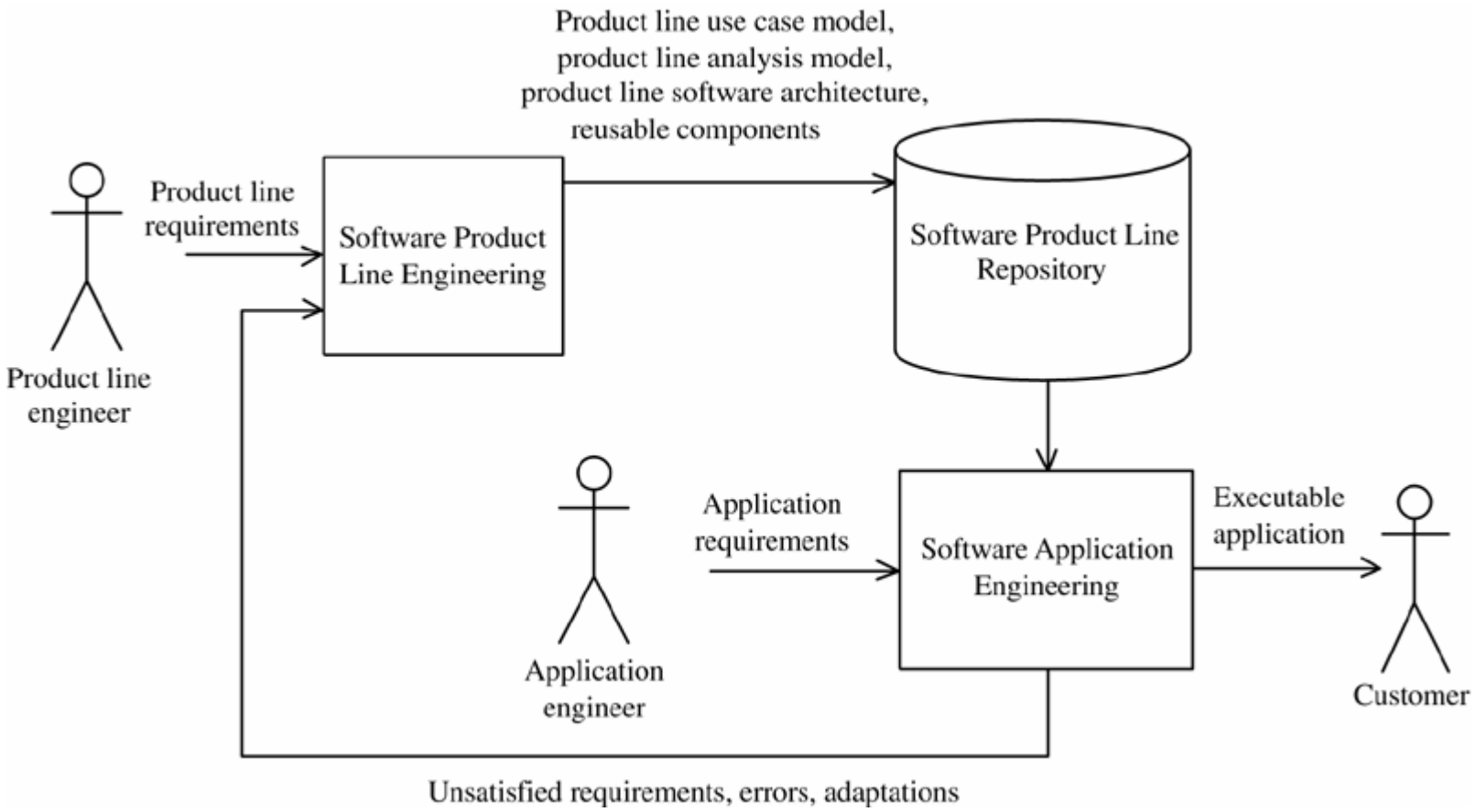
El modelo del SEI

- Gestión de la Línea de Productos (Management)
  - Objetivo:
    - Proporcionar los recursos, coordinar y supervisar el desarrollo de activos y productos
  - Dividida en:
    - Gestión técnica
      - Orientada a los grupos que desarrollan activos y productos
    - Gestión organizacional
      - Orientada a los aspectos organizacionales (estructura, relaciones, recursos, financiamiento, etc.)



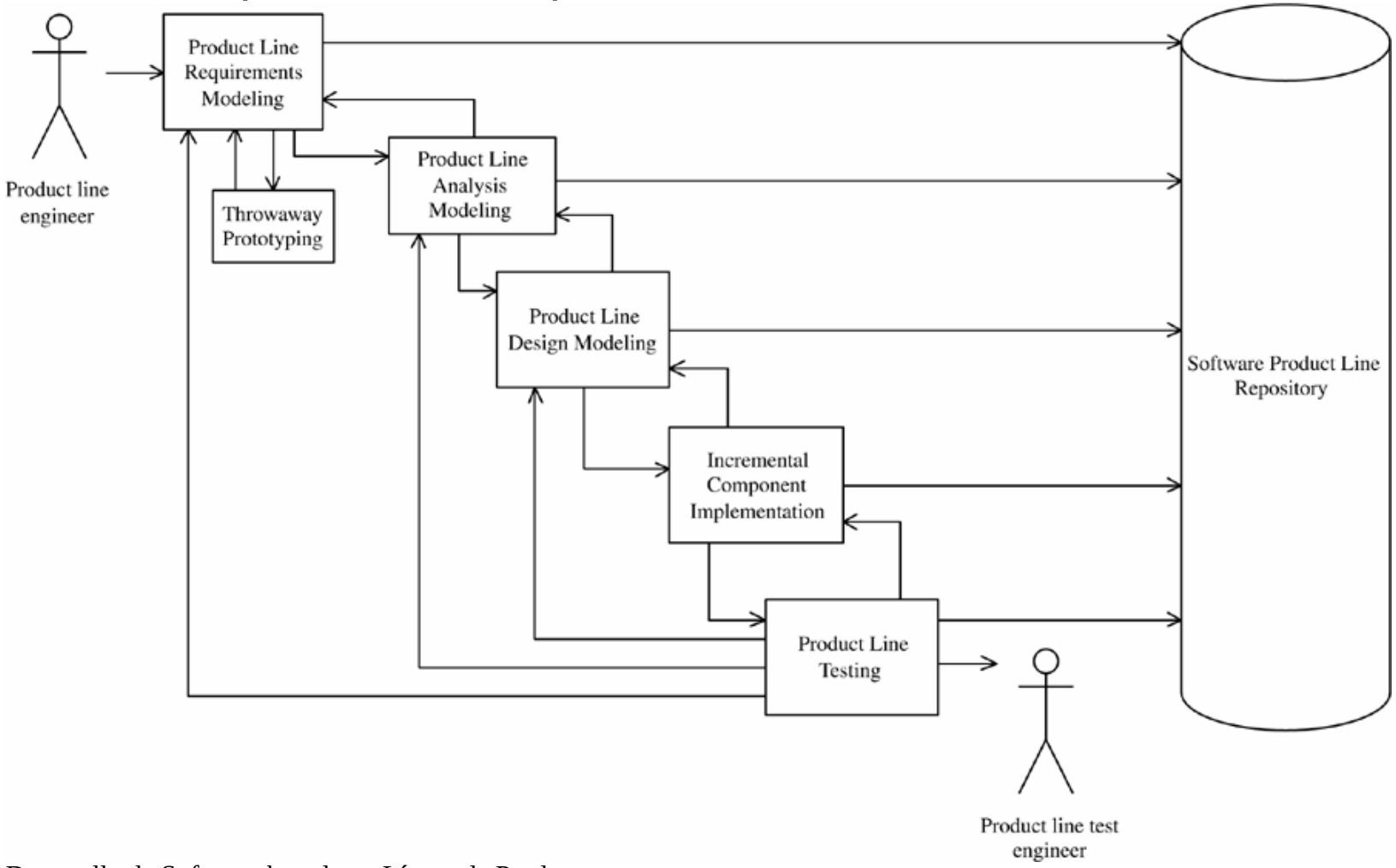
El modelo SPLEP

- SPLEP = Evolutionary Software Product Line Engineering Process (Gooma, 2004)



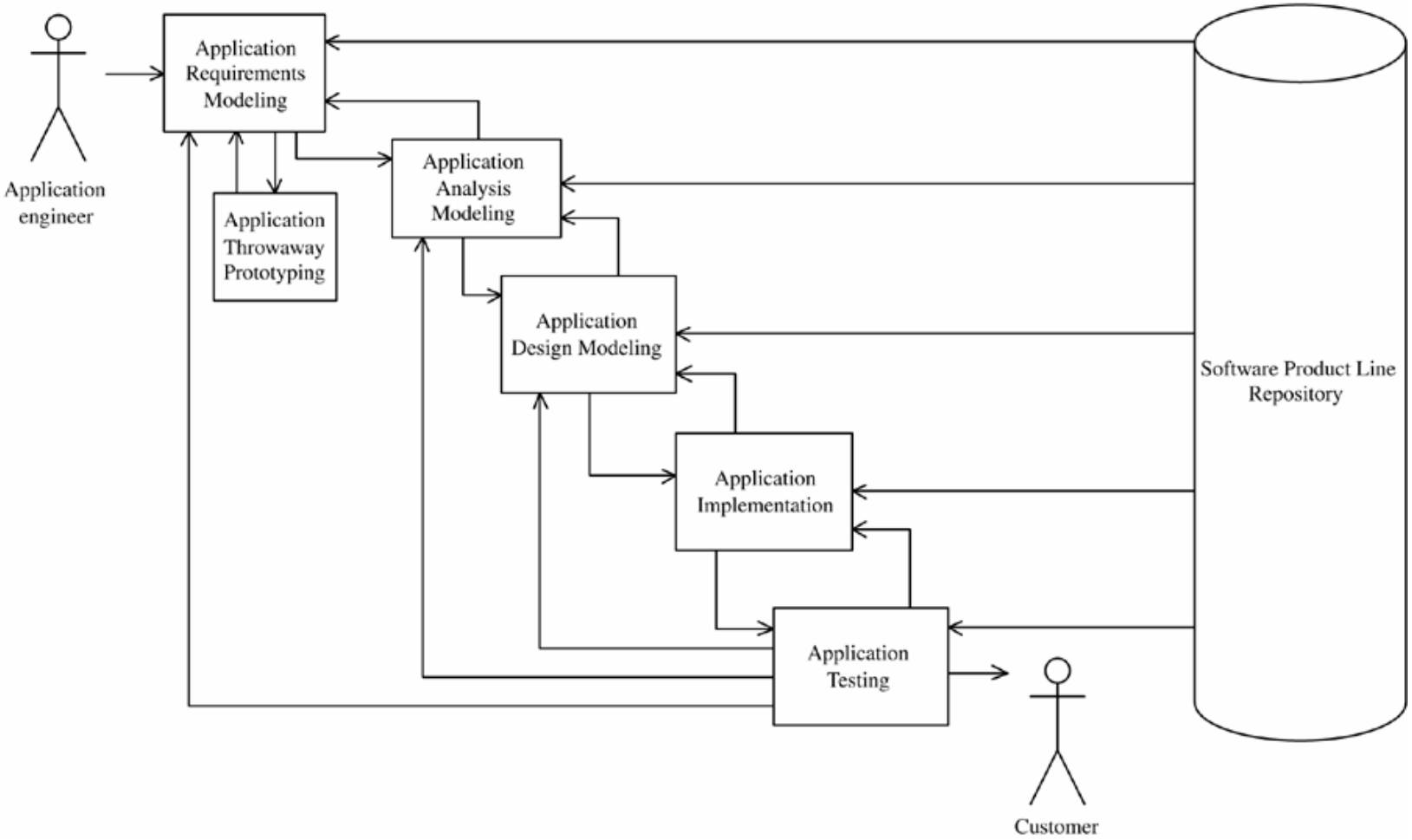
El modelo SPLEP

- Fases de la Ingeniería de Línea de Productos del método SPLEP (Gooma, 2004)

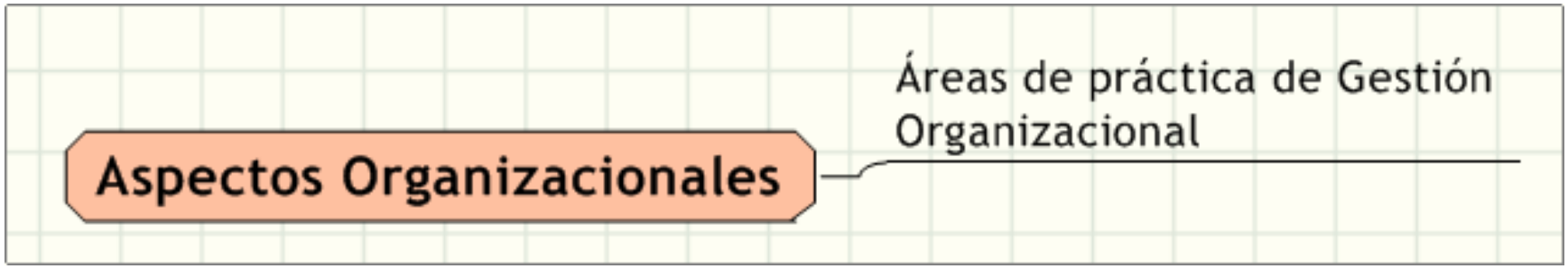


El modelo SPLEP

- Fases de la Ingeniería de Aplicaciones del método SPLEP (Gooma, 2004)



Desarrollo de Software Basado en Línea de Productos



Aspectos Organizacionales

- Están relacionados con:
  - la organización de la empresa y
  - las actividades que ella debe implantar para asegurar el aprovechamiento eficaz y eficiente del paradigma LPS

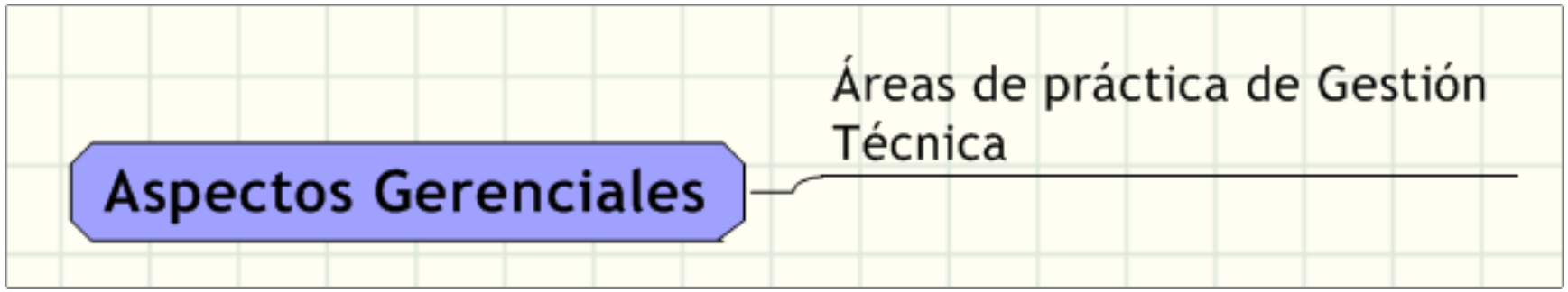
Áreas de práctica de Gestión Organizacional

- Los aspectos organizacionales de las LPS involucran la aplicación de un conjunto de prácticas de gestión:

- Construcción de casos de negocio
- Gestión de relaciones con los clientes
- Desarrollo de una estrategia de adquisición
- Análisis de mercados
- Operaciones

- Planificación organizacional
- Gestión de riesgos organizacionales
- Estructuración de la empresa
- Proyección de tecnologías
- Capacitación de personal

Desarrollo de Software Basado en Línea de Productos



Aspectos Gerenciales

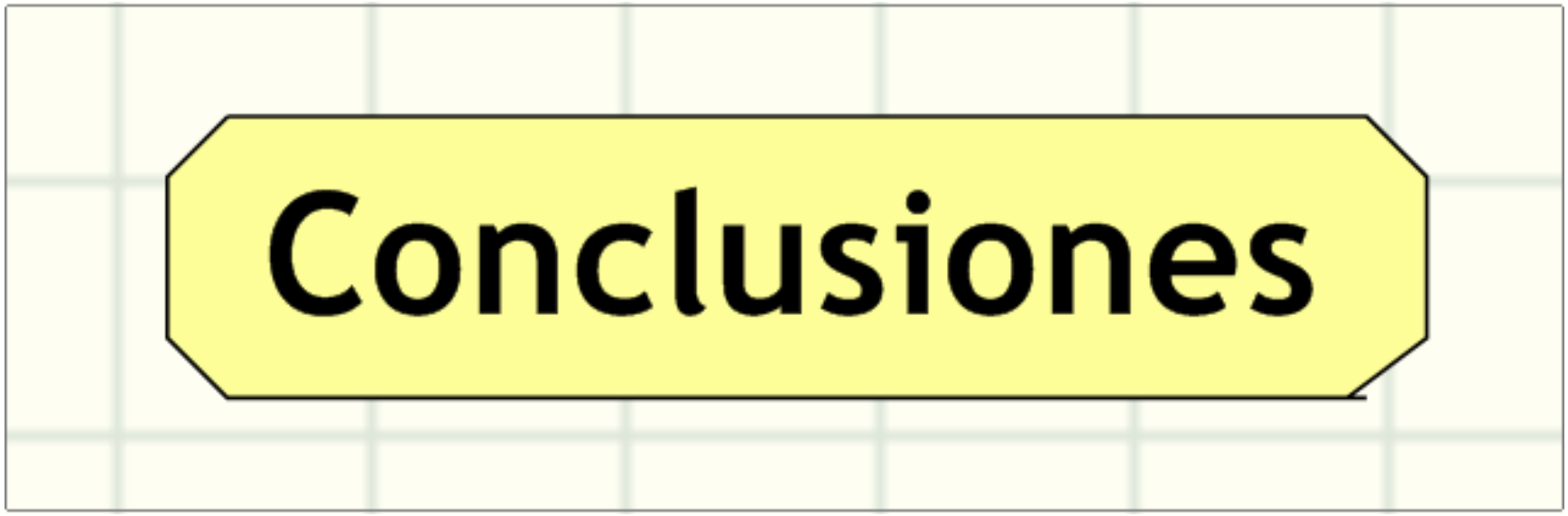
- Están relacionados con la aplicación de los procesos gerenciales en las actividades de Ingeniería de Dominio e Ingeniería de Aplicación de una LPS
  - Planificación de Proyectos
  - Organización de Grupos de Trabajo
    - Grupos de Soporte
      - Administración de Repositorios de Activos de Software
      - Grupos de Mantenimiento de Aplicaciones
    - Grupos de Desarrollo
      - Grupos de desarrollo de componentes
      - Grupos de desarrollo de aplicaciones
  - Dirección
  - Administración de recursos
  - Control

Áreas de práctica de Gestión Técnica

- Los aspectos gerenciales de las LPS involucran la aplicación de un conjunto de prácticas de gestión técnica:

- Gestión de la Configuración
- Recolección de datos, métricas y seguimiento
- Análisis de hacer/comprar/descubrir/encomendar (aprovisionamiento de activos)

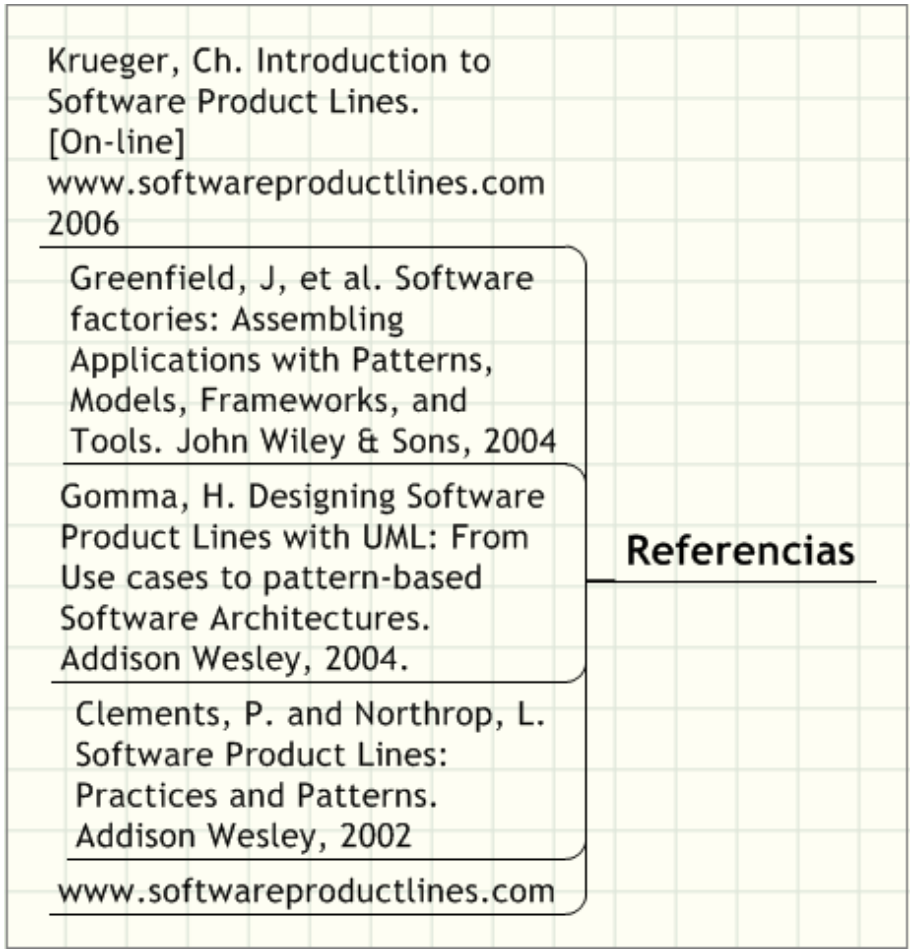
- Definición de procesos
- Alcance
- Planificación técnica
- Gestión de riesgos técnicos
- Soporte de herramientas


Conclusiones

Conclusiones

- Las Líneas de Productos de Software representan el estado del arte en Reutilización del Software
- La implantación del paradigma LPS en una empresa es un proceso complejo
- Para manejar esta complejidad se requiere considerar diferentes aspectos:
  - Conceptuales
  - Tecnológicos
  - Metodológicos
  - Organizacionales
  - Gerenciales

Desarrollo de Software Basado en Línea de Productos



Gracias por su atención

jonas@ula.ve
http://www.webdelprofesor.ula.ve/ingeniería/jonas