Nuevas Maneras de Pensar el Problema de Hacer Software

MDE, SOC y BPM

Francisco Ruiz

Universidad de Castilla-La Mancha (España)

http://alarcos.inf-cr.uclm.es

Uruguay, *Julio-2009*

- Se habla de “crisis del software” desde los años 70
  - Da la casualidad que justo los que nos dedicamos a esto somos los peores profesionales, los más “chapuceros”, en todos los países, o
  - Es que nos enfrentamos a un problema difícil, especial y distinto al que se enfrentaron antes otras ingenierías.
- A veces los éxitos se confunden con los fracasos
  - Si somos malos haciendo software, ¿por qué el software es cada vez más frecuente en la vida de cualquier persona y más importante para cualquier organización?
- En realidad, hemos seguido un proceso histórico muy interesante.
  - Para entender donde estamos y hacia donde vamos debemos comprender de donde venimos.

- A lo largo del tiempo hemos sido capaces de resolver una gran cantidad de dificultades, en un camino que siempre se ha caracterizado por:
  - Aprovechar el aumento de potencia y capacidad del hardware para “*hacer software más cerca de las personas y más lejos de las máquinas*”.

- Fuimos capaces de trabajar de manera lógica y no física.
  - Los enchufes en clavijas pasaron a ser 0's y 1's.
- Inventamos lenguajes y traductores para poder "representar" mejor los algoritmos como ideas.
  - El código máquina dejó de usarse para programar.
- Ideamos lenguajes "cercanos" al idioma natural o a los idiomas de las ciencias (matemáticas) para mejorar nuestra capacidad de expresar.
  - COBOL se pareció al inglés lo máximo posible.
  - PROLOG se basaba en la lógica matemática.
- Descubrimos que teníamos que organizar bien el flujo de ejecución del código.
  - Programación estructurada (PASCAL).

- Conforme el software se fue haciendo más complejo tuvimos que enfrentarnos a nuevos retos:
  - Si tenemos mucho código mejor separarlo en varias partes.
    - Programación modular (MODULA 3).
  - Necesitábamos poder manejar informaciones complejas de distinta naturaleza.
    - Tipos abstractos de datos
    - Sistemas de bases de datos.
  - En un software grande es un lío "organizar" las piezas de código. Necesitamos un criterio para decidir qué piezas tener y qué datos y código poner en cada una.
    - Orientación a objetos.

- Pero seguimos teniendo otros retos pendientes:
  - Si hemos ido subiendo de **nivel de abstracción** en los lenguajes de programación, ¿nos permite la tecnología actual dar otro salto más?.
    - Java es código fuente, y ¿UML no?.
    - ¿Existe alguna manera de construir software más rápida y con menos errores?.
  - La **integración** sigue siendo un problema difícil.
    - Integrar sistemas
    - Integrar tecnologías
  - Seguimos teniendo dificultades para **entender** bien a los clientes/usuarios.
    - Muchos proyectos técnicamente correctos fracasan (el software no sirve a los supuestos destinatarios o no lo usan).
  - El software es la **red**.
    - El concepto clásico cerrado de “aplicación” software está desapareciendo.

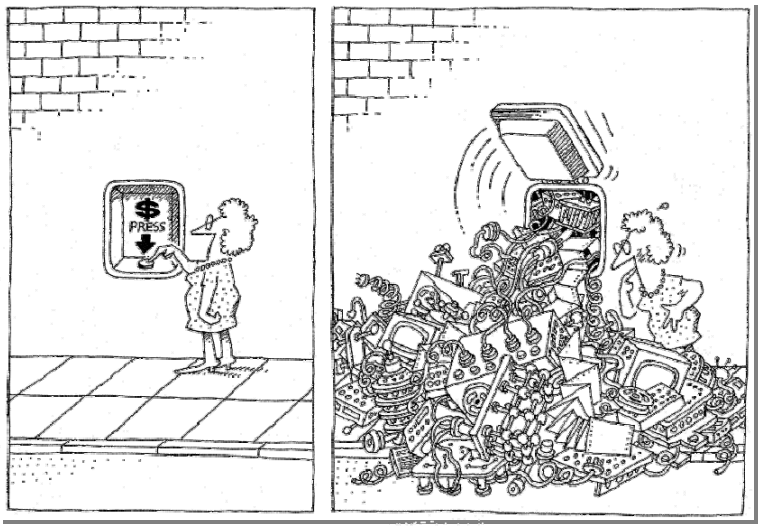
- Para enfrentar estos retos surgen algunos nuevos paradigmas y plataformas tecnológicas
  - que no son alternativos a los anteriores, sino complementarios
- Ingeniería Dirigida por Modelos (MDE)
  - MDA – Model-driven Architecture
- Orientación a Servicios (SOC)
  - SOA – Service-oriented Architecture
- Orientación a los Procesos de Negocio (BPM)
  - BPMS – Business Process Management Systems

- Booch, G. (2007).
  - The Promise, The Limits, The Beauty of Software.
  - Computer Science Teachers Association, ACM.
  - http://csta.acm.org/Resources/sub/Turing_Lecture.ppt

  - *Software development has been, is, and will remain fundamentally hard*.
  - *It is a tremendous privilege to be a software professional*
  - *It is also a tremendous responsibility*

El Problema del Desarrollo de Software
Naturaleza del Problema

- Booch, G. (2007).
  - The Promise, The Limits, The Beauty of Software.
  - Computer Science Teachers Association, ACM.
  - http://csta.acm.org/Resources/sub/Turing_Lecture.ppt



Ingenio (DRAE)

- Industria, maña y artificio de alguien para conseguir lo que desea.
- Máquina o artificio mecánico (ingenio de azúcar).

Ingeniería (DRAE)

- Estudio y aplicación, por especialistas, de las diversas ramas de la tecnología.

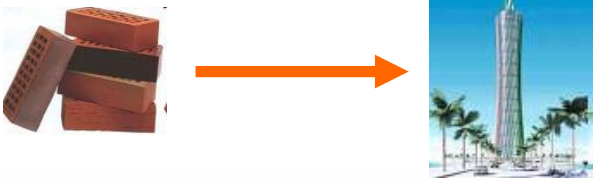
Ingeniero/a

- Persona que aplica los conocimientos de una o varias ramas de la ciencia para resolver cierto tipo de **necesidad** de la gente,
  - Mediante el diseño, construcción u operación de algún tipo de **artefacto o sistema**.

- Cualquier ingeniería se caracteriza porque:
  - Se necesitan conocimientos avanzados para diseñar y construir el tipo de sistemas que la caracteriza.
    - Diferencia entre técnico e ingeniero.
  - Existen dos “momentos”:
    - Primero, conocer el problema, y
    - Sólo después, podemos diseñar y construir la solución.
  - Para conseguir buenos resultados (en calidad, tiempo y costes) es necesario trabajar de forma organizada y sistemática.
  - La creatividad es necesaria (diseño), pero no es suficiente,
    - Diferencia entre artista e ingeniero.

El **sentido común** es muy importante.

- Ley del Mínimo Esfuerzo
  - Entre las opciones correctas elegir la más sencilla.
  - Reutilización
    - Del código, del resto de artefactos software y del conocimiento.
- No inventar la rueda
  - Emplear estándares.
- Aprender de la experiencia (nuestra o de otros).
  - Utilizar “buenas prácticas” y “lecciones aprendidas”.
- Zapatero a tus zapatos
  - No linealidad entre escala vs complejidad.



- La ingeniería existe porque las personas diseñan y construyen artefactos/sistemas cada vez más complejos.
- El mayor nivel de complejidad que el ser humano ha enfrentado a lo largo de su historia se encuentra en algunos de los sistemas software actuales (Windows Vista, Linux, MS Office), ..
- Un indicador de la complejidad de un sistema es el número de variables independientes que afectan al comportamiento del sistema.
  - En un sistema físico (automóvil) son decenas o cientos.
  - En un sistema software (Windows) pueden ser miles o decenas de miles.

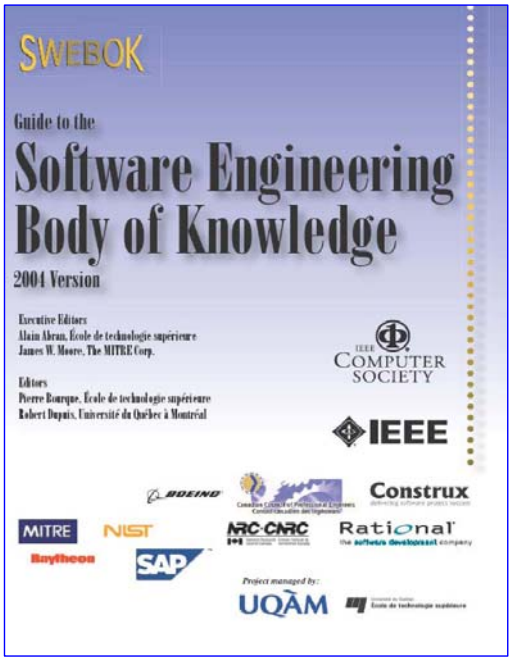
- ¿A qué se parece el software?
  - A un frigorífico (que se fabrica).
  - A un libro (que se idea y se escribe).
  - A una receta de cocina (que se inventa y se anota).
  - A un servicio de un abogado en un juicio (que nos ayuda con su conocimiento especializado).
- ¿Producto o Servicio?.
- Entonces, ¿la gente que hace software qué clase de habilidades y capacidades debe tener?
  - Arquitecto
  - Albañil
  - Jardinero
  - Artista

Definición

Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de los principios y hábitos de la ingeniería al software.

(IEEE,1993)

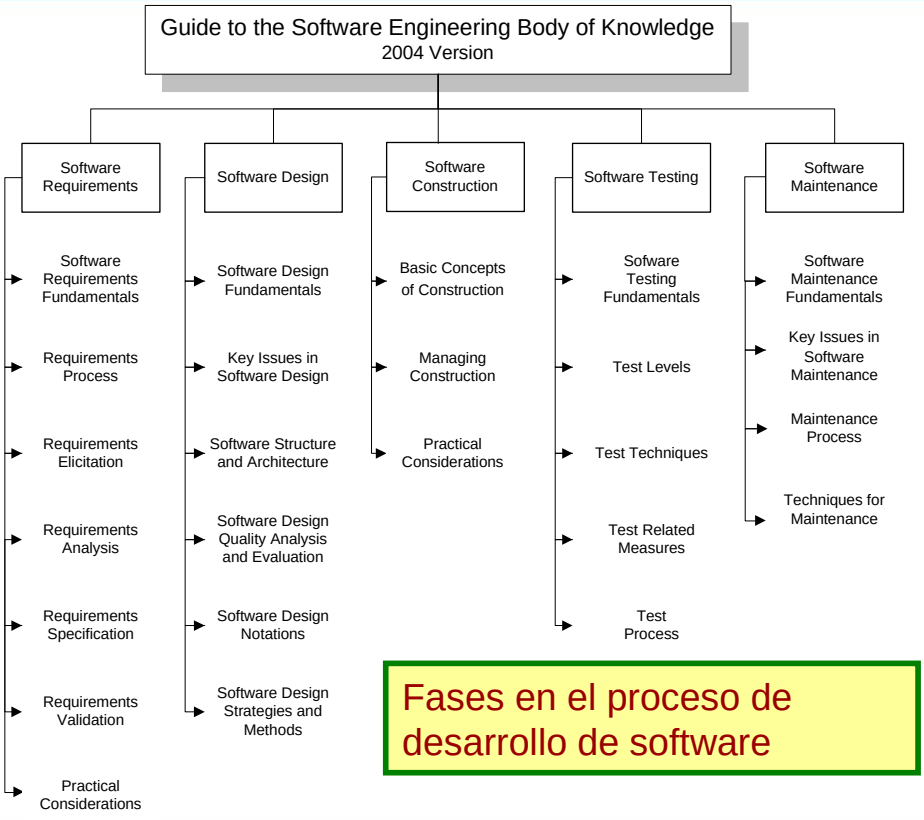
Software Engineering Body of Knowledge
http://www.swebok.org/



RESUMEN DE ÁREAS:

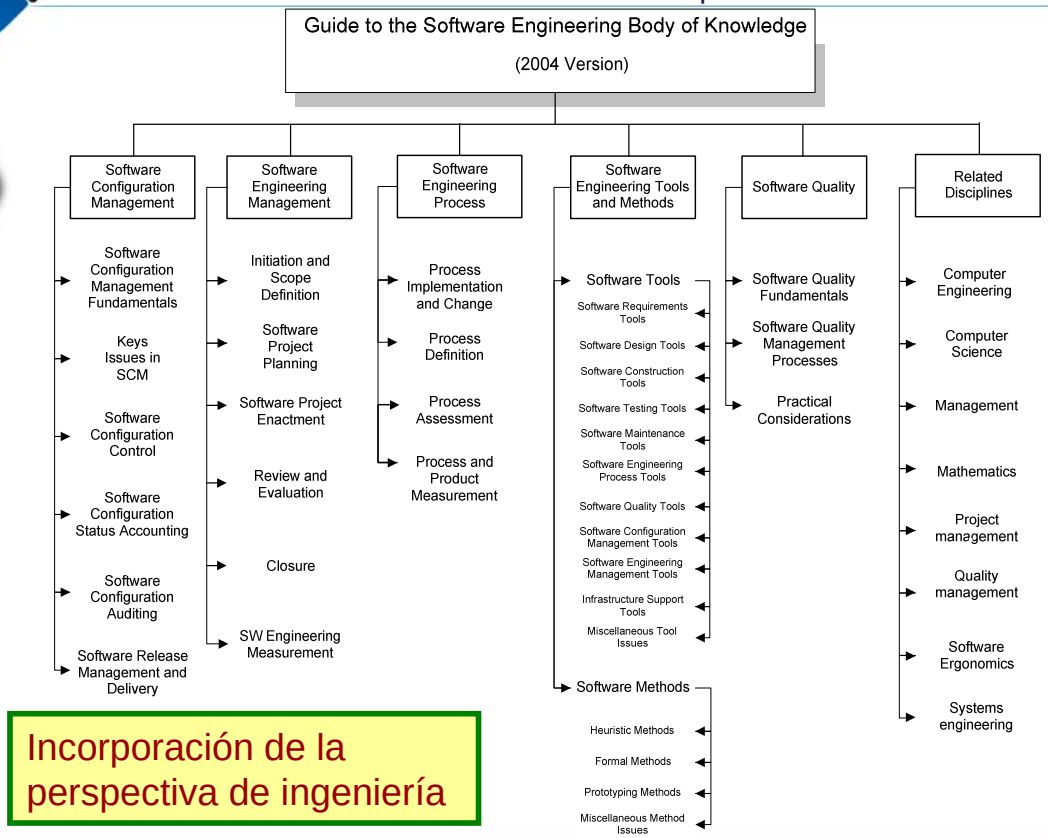
- Fases del Proceso de Desarrollo
  - Requisitos
  - Diseño
  - Construcción
  - Pruebas
  - Mantenimiento
- Perspectiva de Ingeniería
  - Gestión de la Configuración (gestión de productos)
  - Gestión de la Ingeniería (gestión de proyectos)
  - Proceso de Ingeniería (orientación a procesos)
  - Herramientas y Métodos (tecnología de soporte)
  - Calidad
- Disciplinas Relacionadas

Cuerpo de Conocimientos - SWEBOK



Fases en el proceso de desarrollo de software

Cuerpo de Conocimientos - SWEBOK



Incorporación de la perspectiva de ingeniería

Ingeniería del Software es uno de los 5 currículos de ACM:

- Ciencia de la Computación (*Computer Science*)
- Ingeniería de Computadores (*Computer Engineering*)
- **Ingeniería del Software (*Software Engineering*)**
- Sistemas de Información (*Information Systems*)
- Tecnología de la Información (*Information Technology*)

¿Por qué los distingue?

¿Es que Informática no es una profesión, y sí lo es Ingeniería del Software?

ACM Computing Curricula 2005. Overview Report

- Pesos asignados a los tópicos de Ingeniería del Software

Área de Conocimiento	CE		CS		IS		IT		SE	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Fundamentos de Programación	4	4	4	5	2	4	2	4	5	5
. . . . .										
Desarrollo de Sistemas de Información	0	2	0	2	5	5	1	3	2	4
Análisis de Requisitos Técnicos	2	5	2	4	2	4	3	5	3	5
Fundamentos de Ingeniería para Software	1	2	1	2	1	1	0	0	2	5
Economía de Ingeniería para Software	1	3	0	1	1	2	0	1	2	3
Modelado y Análisis de Software	1	3	2	3	3	3	1	3	4	5
Diseño de software	2	4	3	5	1	3	1	2	5	5
Verificación y Validación de Software	1	3	1	2	1	2	1	2	4	5
Mantenimiento del Software	1	3	1	1	1	2	1	2	2	4
Procesos Software	1	1	1	2	1	2	1	1	2	5
Calidad del Software	1	2	1	2	1	2	1	2	2	4
. . . . .										
Desarrollo Digital	0	2	0	1	1	2	3	5	0	1
. . . . .										

Papel Social del Informático (Dahlbom & Mathiassen, 1997):

	Construyo cosas	Ayudo a la gente	Cambio las cosas
Orientación	Máquina	Cultura	Poder
Actividad	Construcción	Evolución	Intervención
Papel	Ingeniero	Facilitador	Emancipador

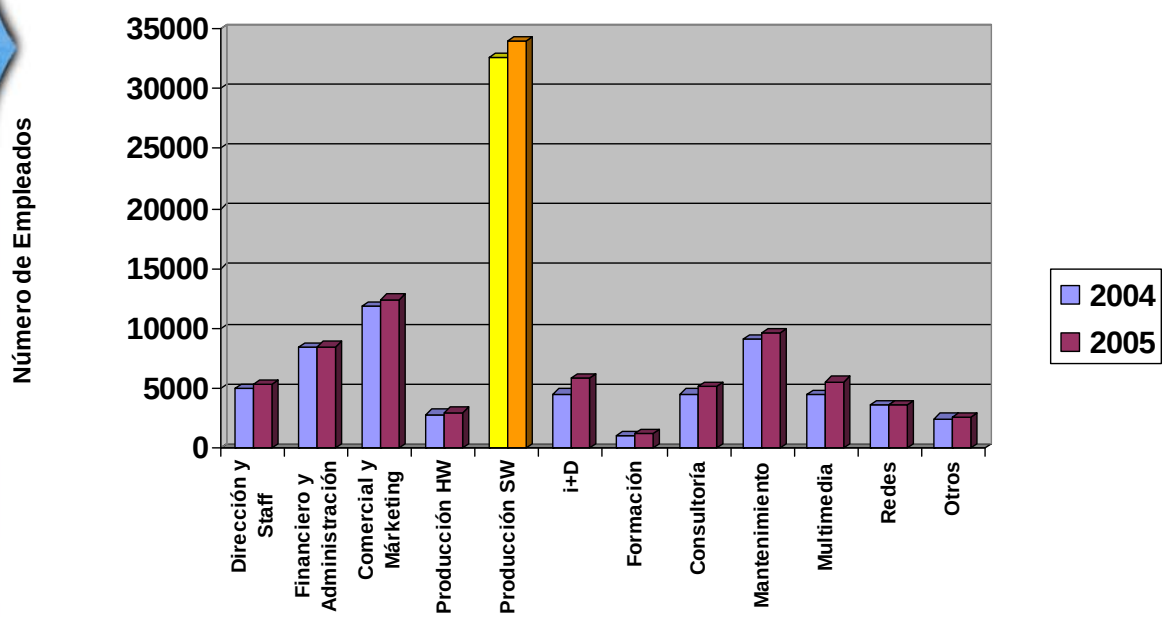
Esto implica tres tipos de roles diferentes dentro de la informática:

- los ingenieros de **desarrollo**: construyen artefactos software o hardware;
- el personal de **soporte**: ayuda a usar aquello que otros desarrollaron; y
- los **consultores**.

Distribución de los ocupados en perfiles TIC en la Unión Europea 15 (miles), *Career-Space.*

Ocupaciones (SOC90)	Total Puestos TIC	% Incr. 2000-2004
Analistas y Programadores	1.885	+6,1
Ingenieros de Software	1.306	+10,0
Administradores de Sistemas Informáticos	1.019	+4,1
Operadores Informáticos	696	-0,5
Consultores y Gestores	437	+3,7
Ingenieros de Diseño y Desarrollo TIC	399	+0,2
Ingenieros de Computadores	348	+6,5
Ingenieros Eléctricos	203	-0,5
Ingenieros Electrónicos	196	+3,0
Total TIC	6.489	+4,7
Total Empleo	166.696	+0,8

Telefonica (2007): Personal por áreas funcionales en el sector de TI en España).



La demanda de acceso a los estudios se ha reducido en los últimos años.

¿Porqué?

- Si las perspectivas laborales, a pesar de la crisis, son prometedoras.
- Si la sociedad está cada vez más informatizada.

- ¿Sólo ocurre con Informática o es un fenómeno más general?.
- ¿Sólo ocurre en España o es más global?.
- ¿Uruguay?

- Factores locales (España):
  - Imagen social devaluada del Ingeniero/a
  - ROI (retorno de inversión) bajo
    - Muchos consideran que el esfuerzo no merece la pena
    - ¿Culpa de las empresa?
- Factores globales (países desarrollados):
  - Cultura del no esfuerzo, de lo fácil.

Pero …, ¿existe además algún factor específico sólo de Informática?

- Peter J. Denning and Andrew McGettrick. CACM 48(11), nov-2005.
  - **Recentering Computer Science**
  - The recent decreases of enrollment in computer science programs signal a chasm between our historical emphasis on programming and the contemporary concerns of those choosing careers.

- Porque durante mucho tiempo hemos “vendido” una imagen social que asocia Informática a “programar” y la gente (los jóvenes y sus padres) se hace el siguiente razonamiento:
  - No tiene sentido estudiar una ingeniería, que encima es de las más difíciles, para luego trabajar de programador si con eso de la globalización los hindúes y otros programan como locos por cuatro euros.

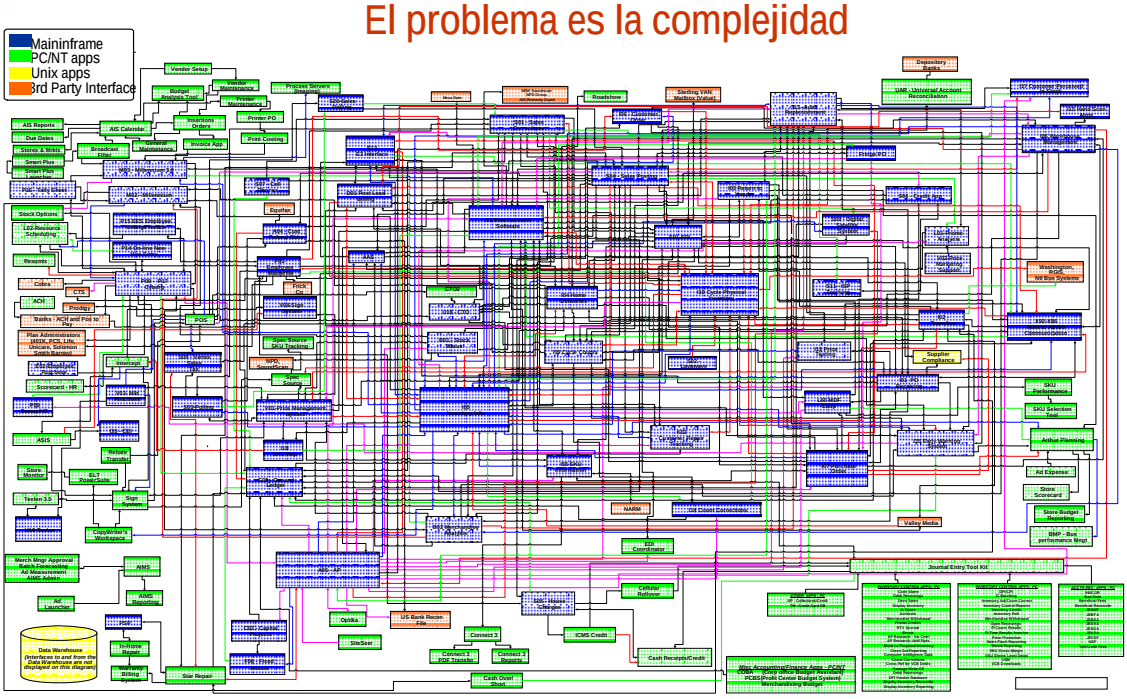
ES EL SENTIDO COMÚN DE NUESTRA GENTE

! HAGAMOS QUE NUESTROS TITULADOS SEAN MAS INGENIEROS DE SOFTWARE Y MENOS ALBAÑILES DE SOFTWARE !

- En estos últimos años, tres nuevas maneras de pensar (paradigmas) el problema de hacer software están teniendo un impacto importante en la academia y la industria:

- Ingeniería Dirigida por Modelos (MDE)
  - MDA – Model-driven Architecture
- Orientación a Servicios (SOC)
  - SOA – Service-oriented Architecture
- Orientación a los Procesos de Negocio (BPM)
  - BPMS – Business Process Management Systems

El problema es la complejidad



Diseño de una Aplicación Real (Retail)

Ensamblador

Prog. Estructurada ...

Prog. O. Objetos ...

Prog. O. Componentes ...

Prog. O. Aspectos ...

Prog. O. Servicios ...

Prog. O. Eventos

Prog. O. X???

Prog. O. Y???

Prog. O. Z???

- **Demasiado bajo nivel**
- **Poca expresividad**
- **Programas muy complejos**

Es preciso romper ese nudo “Gordiano”

La programación no debe ser el centro de atención. Hay que elevar NOTABLEMENTE el nivel de abstracción

¿Cómo se hace en otras ingenierías más maduras?

- Ingenierías civiles (caminos, canales, puertos, …)
- Arquitectura y construcción
- Ingeniería aeronáutica y del espacio
- …

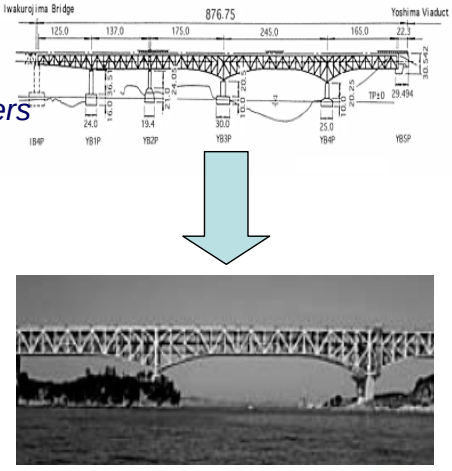
Las ingenierías tradicionales usan “modelos”

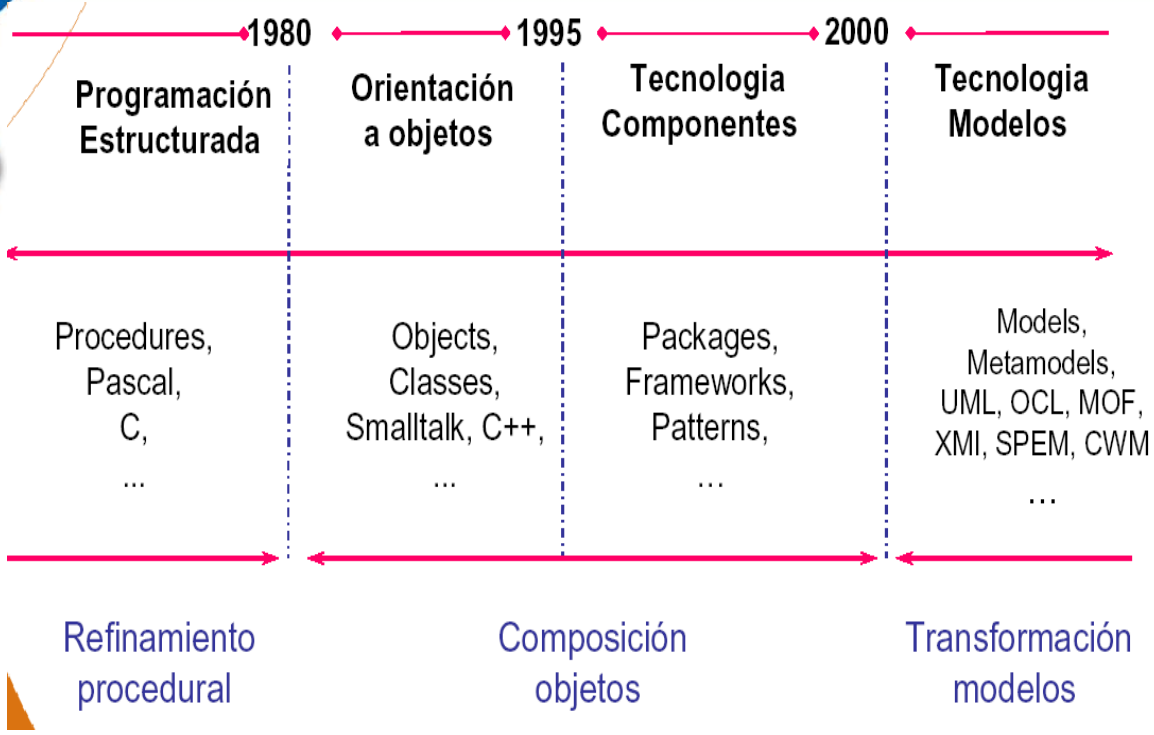
Tan antiguos como las Ingenierías (p.e. Vitruvius)

Los ingenieros tradicionales siempre construyen modelos antes de construir sus obras y artefactos

Los modelos sirven para:

- **Especificar el sistema**
  - Estructura, comportamiento,…
  - Comunicarse con los distintos *stakeholders*
- **Comprender el sistema (si ya existe)**
- **Razonar y validar el sistema**
  - Detectar errores y omisiones en el diseño
  - Prototipado (*ejecutar* el modelo)
  - Inferir y demostrar propiedades
- **Guiar la implementación**





[Selic, 2003]

Características de los Modelos

Abstractos

- Enfatizan ciertos aspectos, mientras que ocultan otros

Comprensibles

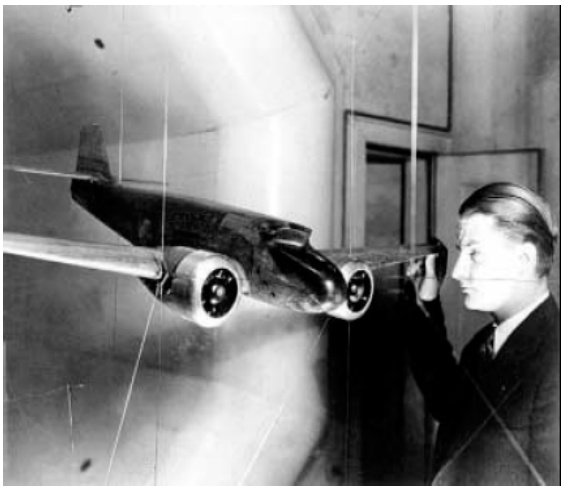
- Expresados en un lenguaje comprensible por por los usuarios y *stakeholders*

Precisos

- Fieles representaciones del objeto o sistema modelado

Predictivos

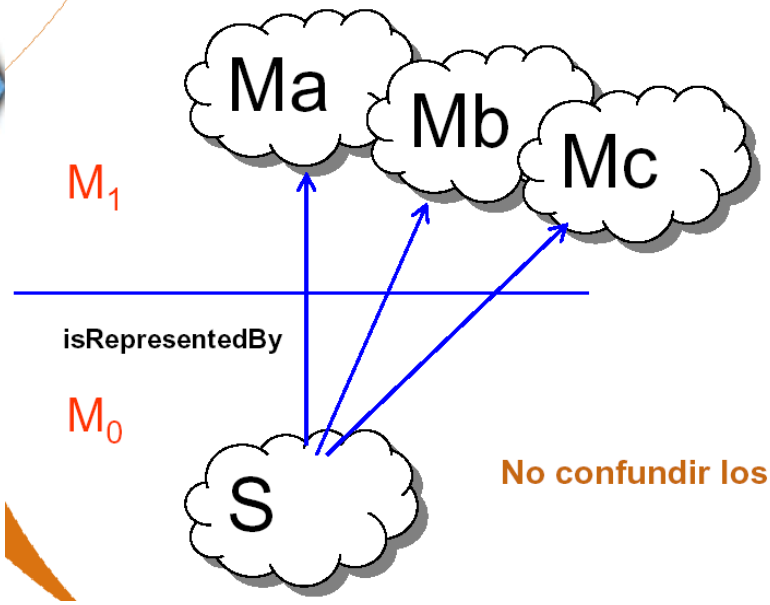
- Deben de poder ser usados para inferir conclusiones correctas



Baratos

- Mas fáciles y baratos de construir y estudiar que el propio sistema

Modelos



Un sistema S puede estar representado por **múltiples modelos**

Cada modelo ofrece una **visión parcial del sistema**

No confundir los modelos con el sistema

¿Qué es Model-driven Engineering (MDE)?

Un enfoque de ingeniería del software en donde las entidades de primer nivel son los modelos y las transformaciones de modelos.

- **frente a los programas y los compiladores, que constituyeron el paradigma análogo hace treinta años.**

Implica la generación (casi) automática de implementaciones a partir de modelos.

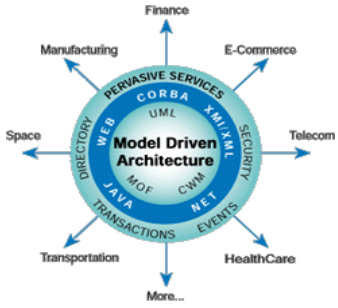
Son claves los lenguajes, tanto de modelado como de transformación de modelos. Los modelos son conformes a meta-modelos.

MDA es la propuesta para MDE que hace OMG, usando sus estándares:

- MOF, UML, OCL, XMI, QVT

MDA

Model Driven Architecture



Created by Business Analysts to Describe Business Processes

Computation-Independent Model (CIM)

CIM >> PIM Mapping

Created by Architect/ Designer to Describe Architecture

Platform-Independent Model (PIM)

PIM >> PSM Mapping

Platform-Specific Model (PSM)

Created by Developer or Tester to Implement Solution

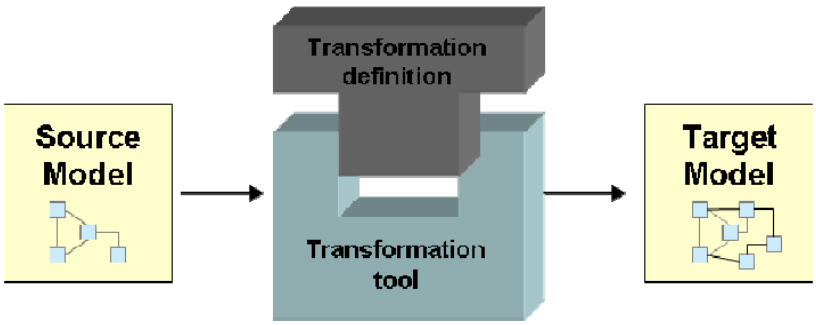
PSM >> Code Mapping

Code

Los **Modelos** son el artefacto central del desarrollo en lugar del **código** fuente

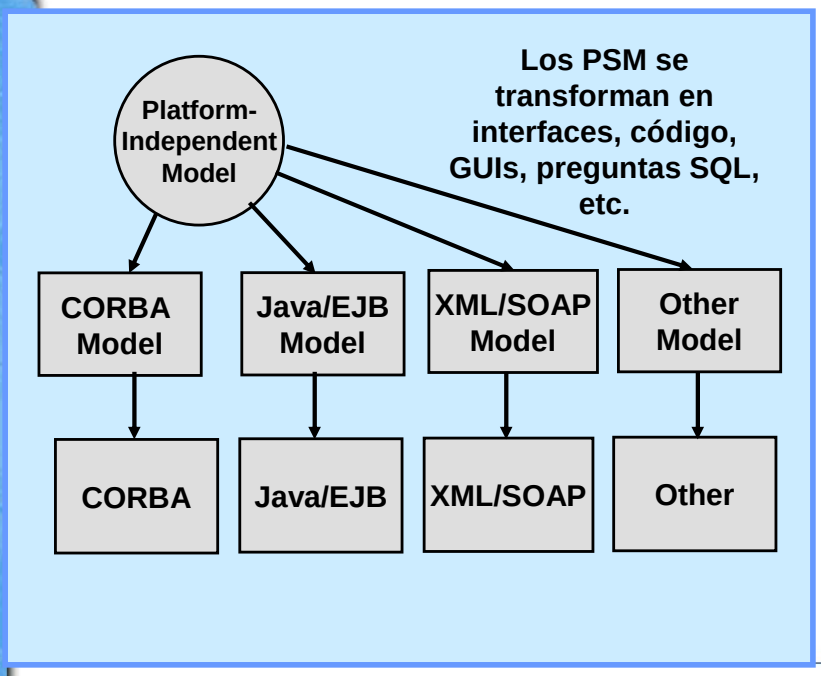
Transformaciones de modelos

- Generación automática de un modelo destino desde un modelo fuente, de acuerdo a una definición de transformación.
- Una definición de transformación es un conjunto de reglas de transformación.
- Una regla de transformación es una descripción de cómo uno o varios elementos del metamodelo fuente pueden ser transformados en uno o varios elementos del metamodelo destino.

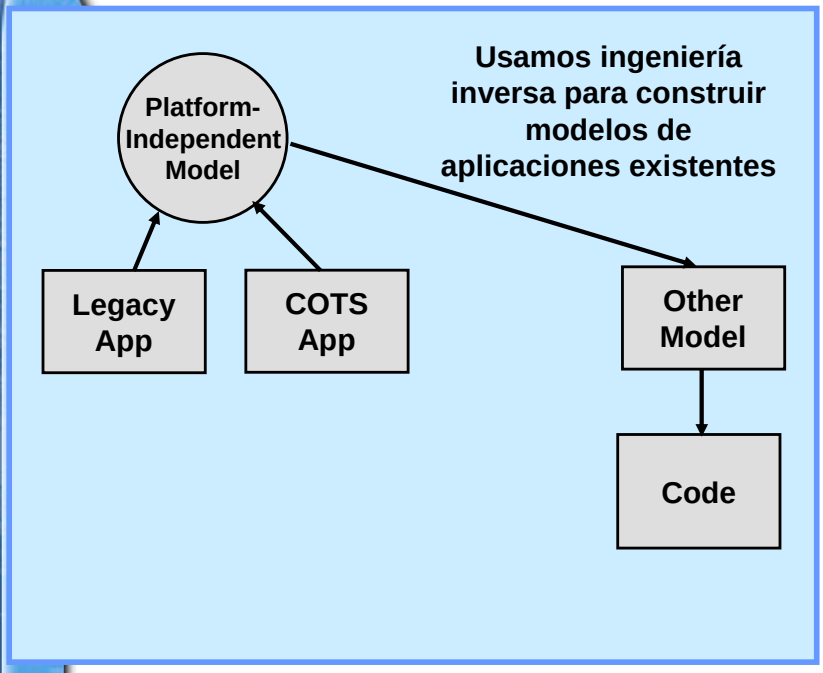


~~Write Once, Run Everywhere~~

Model Once, Generate Everywhere!



Es fácil contar con implementadores automáticos a partir de modelos específicos, pues son de bajo nivel



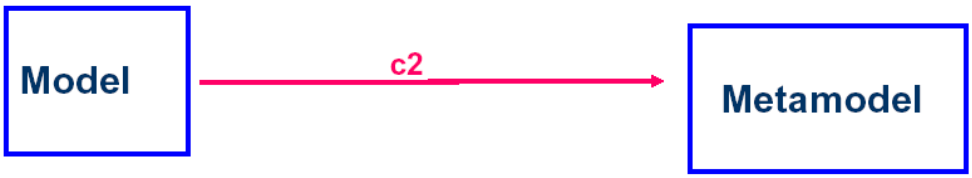
También es útil para:

(1) **Integración** en nuestra aplicación de COTS, sistemas de terceros y sistemas heredados.

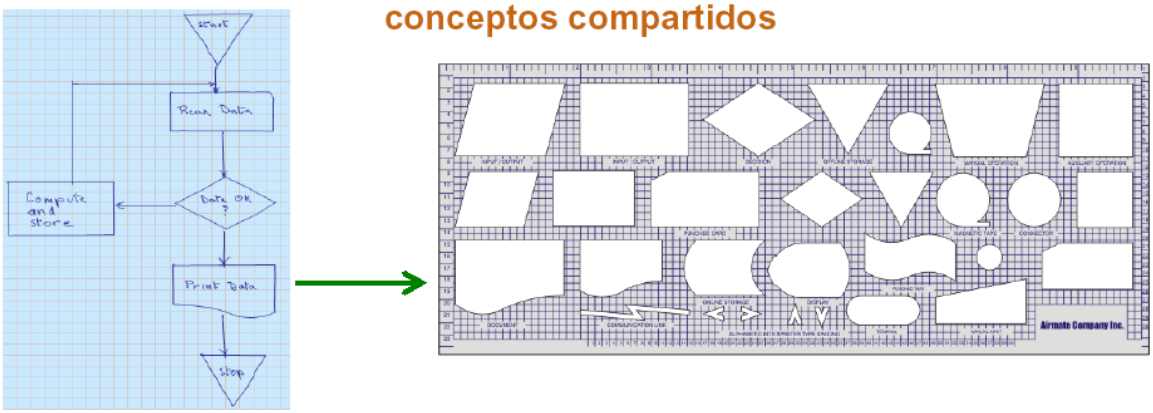
(2) **Architecture Driven Modernization**: modernización de sistemas actuales.

NASA, DoD, EDF, Banca

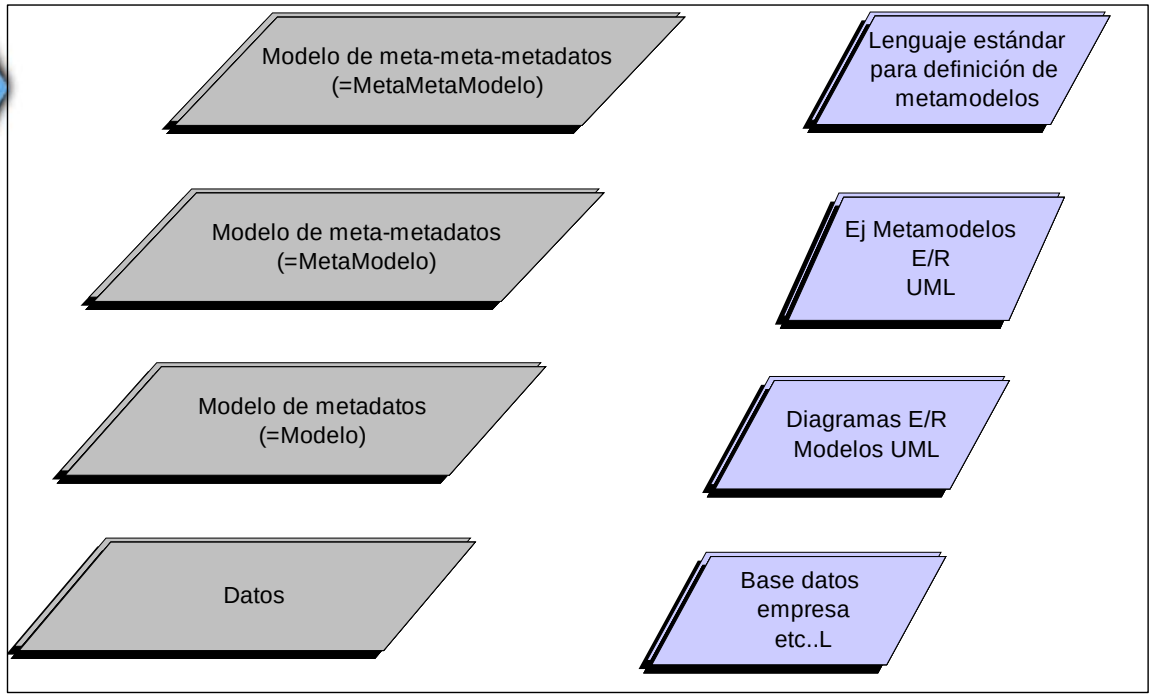
Metamodelado



El metamodelo es la especificación formal de conceptos compartidos



Metamodelado



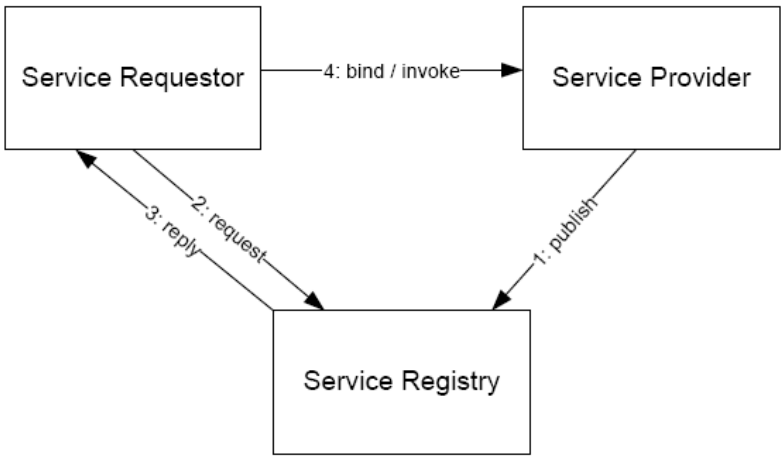
Meta-Object Facility (MOF)

La **Computación orientada a Servicios** (***Service-oriented Computing***) es una de las principales tendencias, tanto en la tecnología software como en la ingeniería de negocios.

La idea central es capturar funcionalidad relevante del negocio como un **servicio** y proveer información suficientemente detallada para que los clientes puedan usarlo.

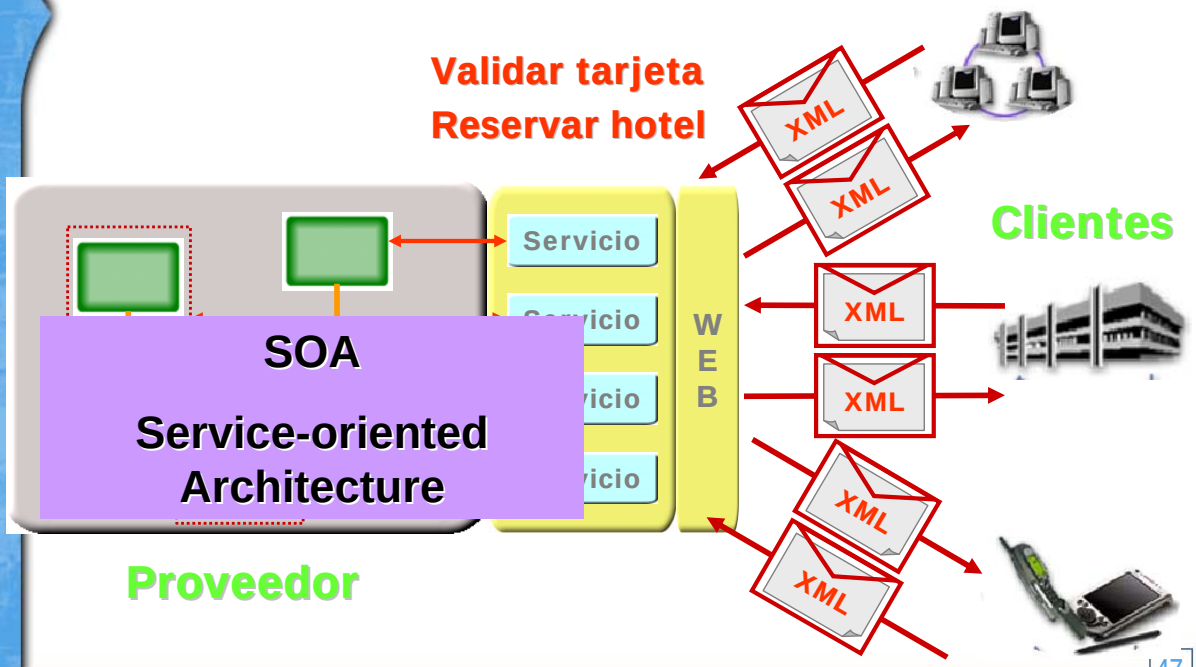
Servicio:

- Un servicio captura funcionalidad con valor para el negocio y la pone lista para ser usada.



Service-oriented Computing (SOC)

Nuevo enfoque de interacción entre sistemas mediante **servicios**



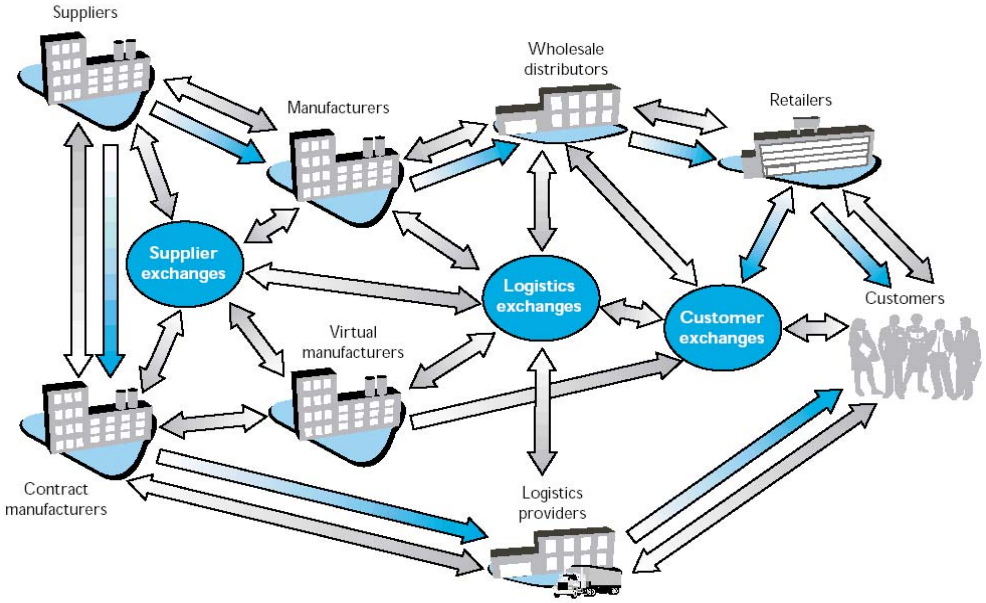
BPM – Business Process Management

Gestión Guiada por Procesos

- Es una nueva manera de abordar el problema de comunicación entre los clientes/usuarios de las TI y los técnicos
  - Antes
    - Gente de Negocio: procesos, roles, personas, …
    - Técnicos: sistemas, máquinas, datos, …
  - Ahora con BPM
    - Los técnicos hablan de los mismo.
    - La tecnología BPMS permite salvar la distancia con los sistemas, máquinas y aplicaciones que automatizan los PN.

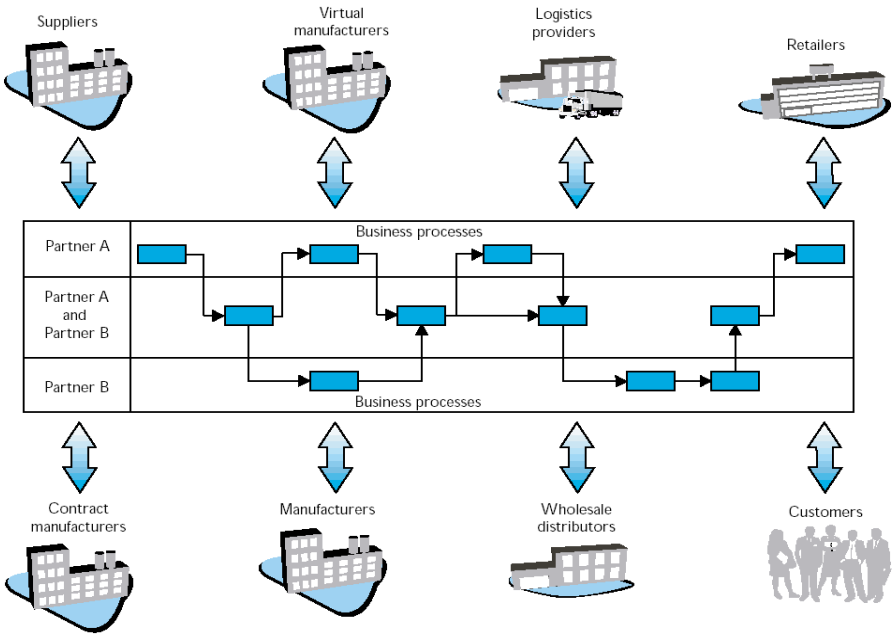
BPM – Business Process Management

Antes

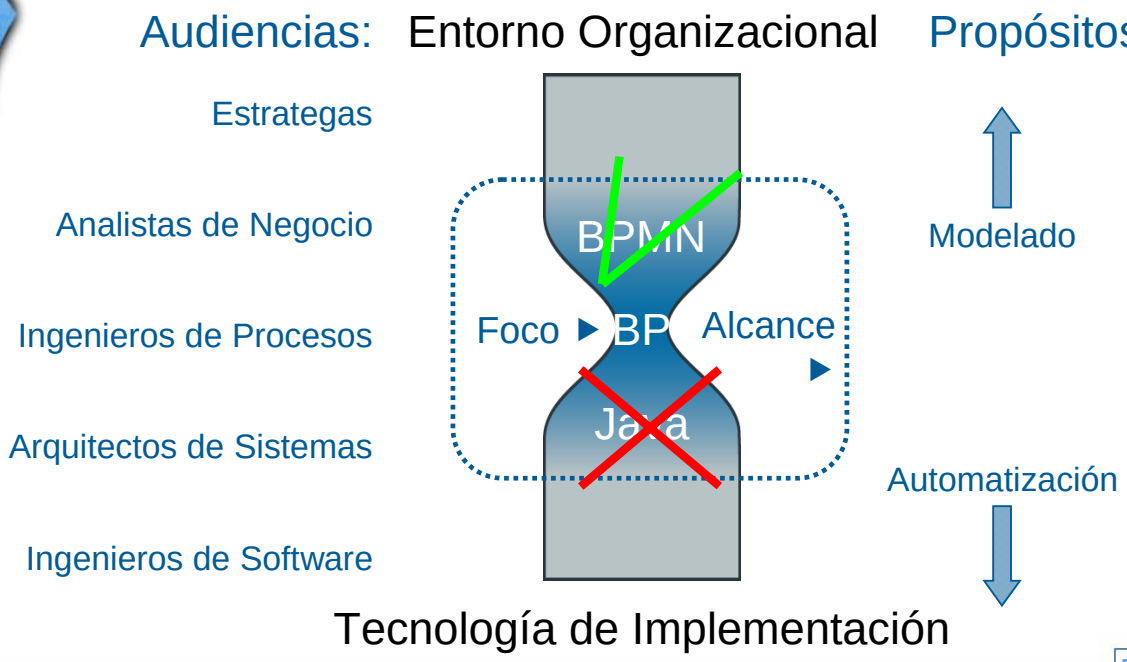


BPM – Business Process Management

Después



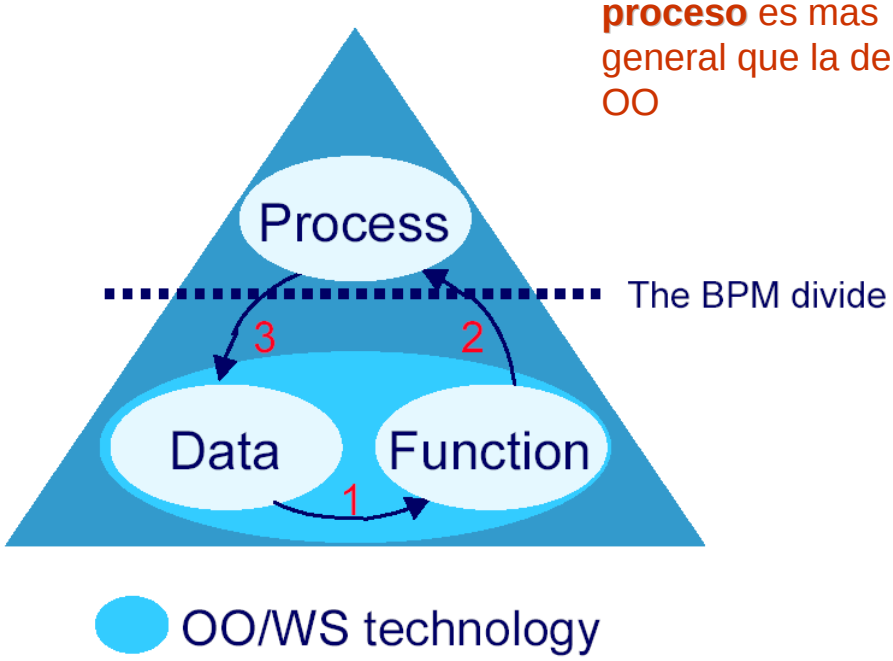
- Lenguajes pensados para la gente no técnica
  - Basados en conceptos de negocio, organizacionales.



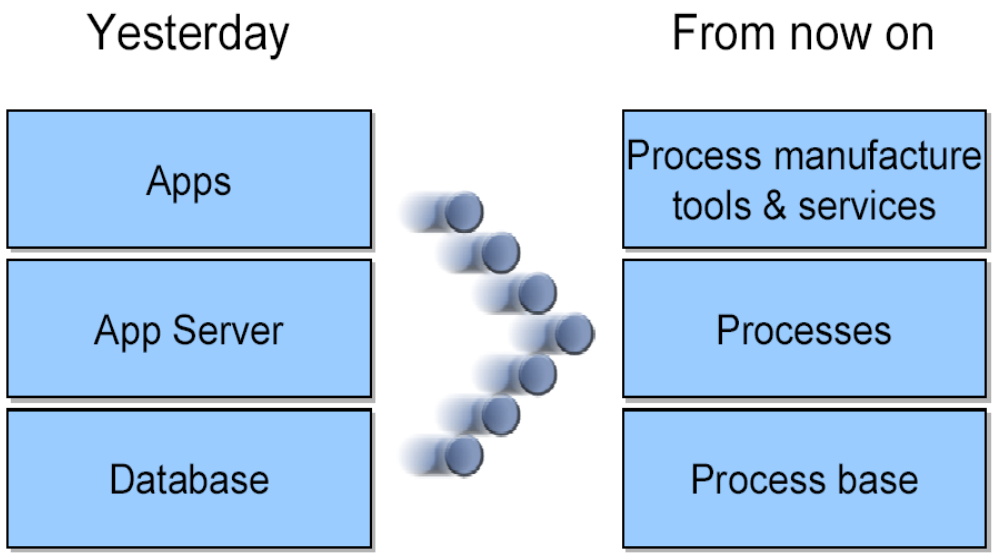
- DOMINIO DEL PROBLEMA
  - VS
- DOMINIO DE LA SOLUCIÓN

- En este paradigma el enfoque se centra en el dominio del **problema**.
  - Nuestra especialidad (ingenieros) es el dominio de la solución.
  - El dominio del problema pertenece a los usuarios.
  - En el dominio del problema debemos adaptarnos a los usuarios.
  - **REQUISITOS** DE ALTO NIVEL.

La perspectiva de **proceso** es mas general que la de OO

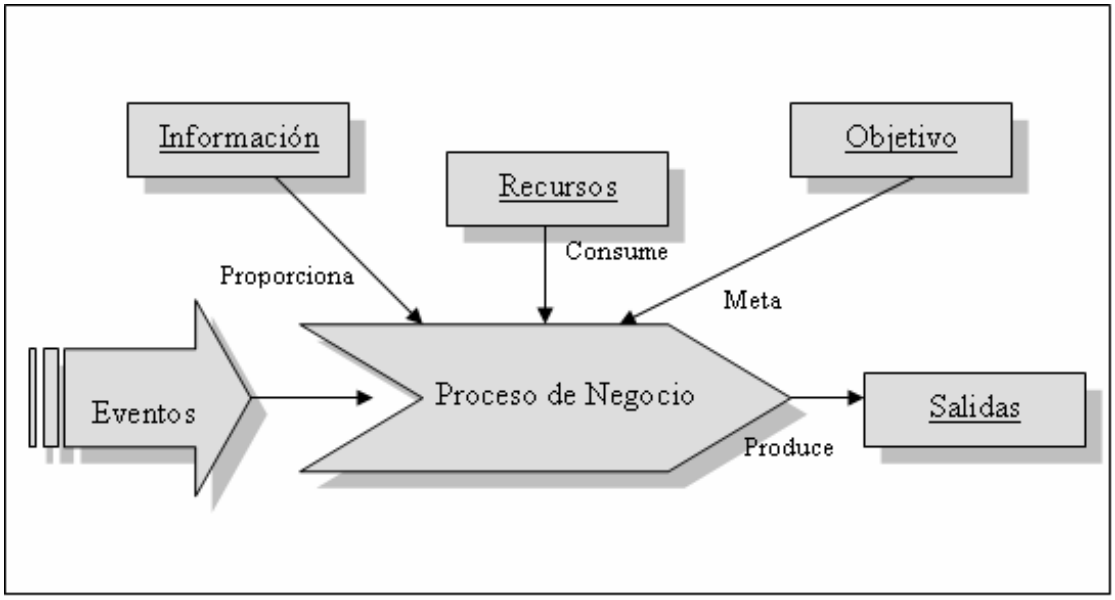


?



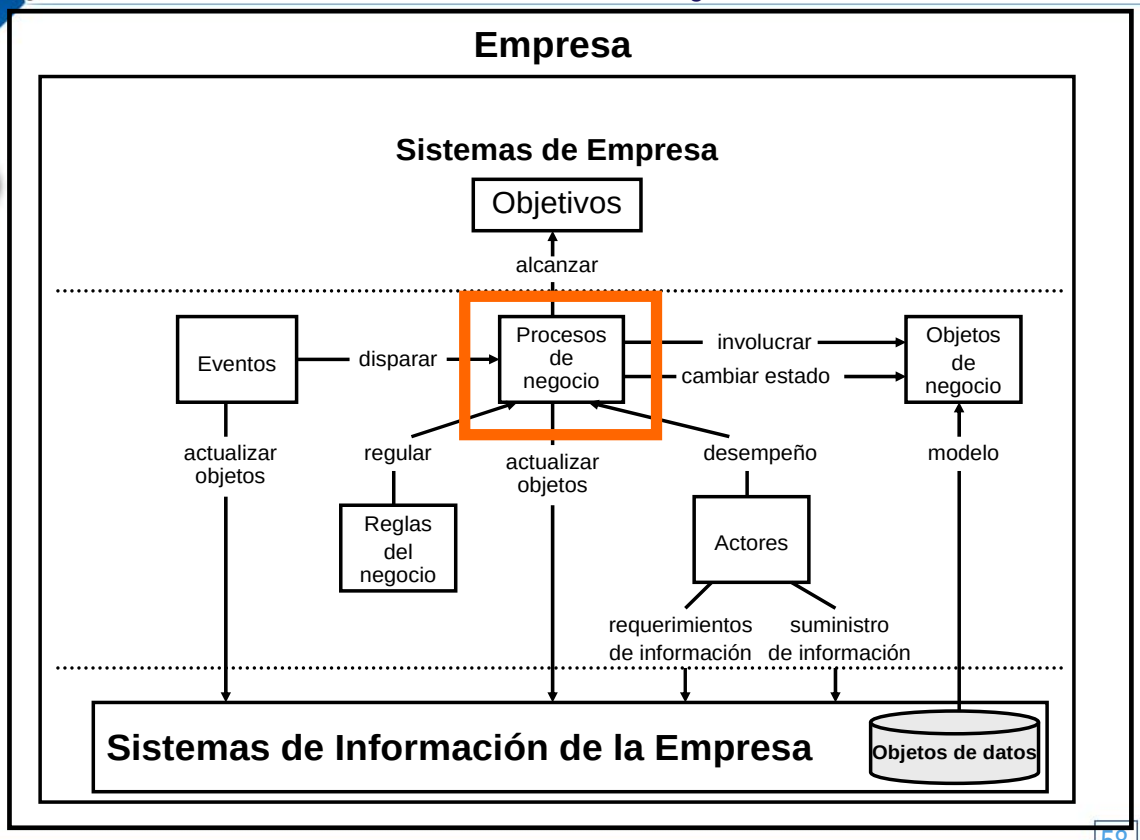
Un Proceso de Negocio (Business Process) es un conjunto de actividades que son realizadas en coordinación en entorno organizacional y técnico. (Weske, 2007)

- Estas actividades, en su conjunto, ayudan a alcanzar un determinado objetivo de negocio.
- Cada proceso de negocio es realizado (*enacted*) por una única organización, pero puede interactuar con procesos de otras organizaciones.



Características:

- Grandes y complejos
- Muy dinámicos
- Ampliamente distribuidos y particularizados
- Larga duración
  - una ejecución puede durar meses o años
- Automatizados
  - al menos en parte
- Dependientes de la inteligencia y juicio humanos
- Difíciles de hacer visibles



Tipos de procesos:	Industriales	de Información	de Negocio
Foco	**COSAS**	**DATOS**	**RELACIONES**
Propósito	Transformar y ensamblar materiales y componentes en otros componentes y productos finales, usando recursos	Procesar y transmitir datos estructurados y no estructurados, y conocimiento	Alcanzar las condiciones que satisfacen las necesidades de los participantes, clientes o usuarios
Característi-cas	Tradiciones de la ingeniería industrial	Tradiciones de la ingeniería informática	Basados en estructuras de comunicación y coordinación humanas encontradas en todos los lenguajes y culturas
Acciones	Ensamblar, Transformar, Transportar, Almacenar, Inspeccionar	Enviar, Invocar, Grabar, Recuperar, Consultar, Clasificar,	Solicitar, Prometer, Ofrecer, Rechazar, Proponer, Cancelar, Medir

Puntos de Vista

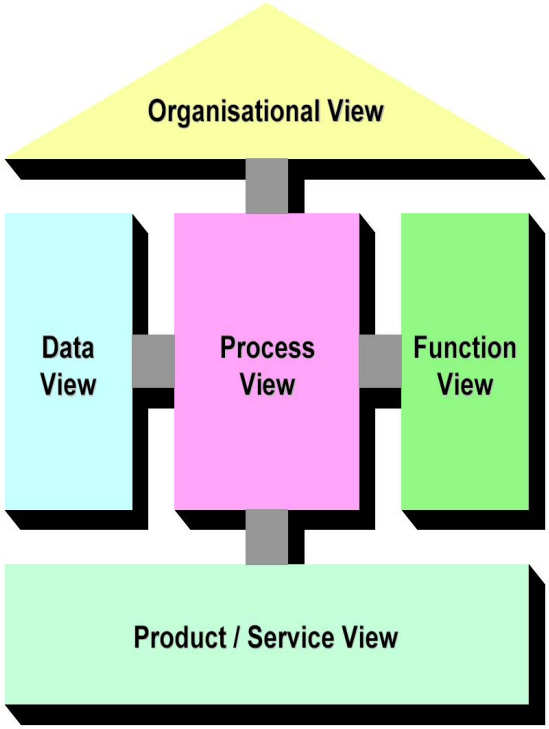
Datos
- ¿Qué información es importante? (ej: Paciente, Proveedor, Producto, ..)

Funciones
- ¿Qué funciones serán realizadas? (ej: Hacer plan de producción, procesar pedidos)

Organización
- ¿Qué unidades organizacionales existen? (ej: Compras, Almacén, Contabilidad)

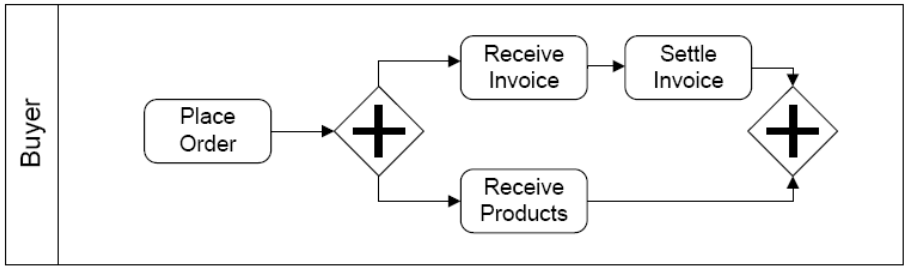
Procesos
- Interrelaciones entre datos, funciones y unidades organizacionales

Productos/Servicios
- ¿Cuáles son los productos/servicios importantes? (ej: historia clínica, diagnóstico, factura)

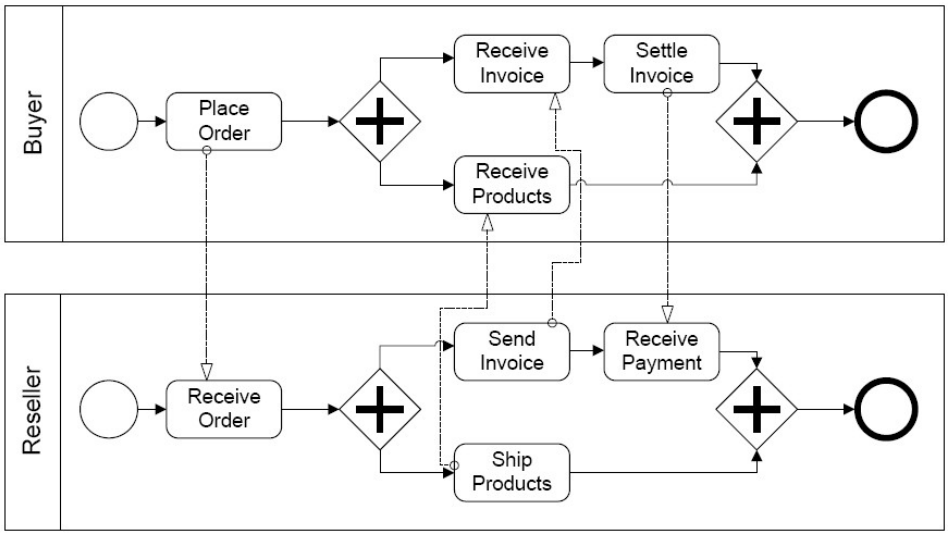


- Son una representación abstracta (gráfica) de los procesos de una organización, que muestran principalmente *cómo* y por *quién* son llevadas a cabo las *actividades* que generan valor para la organización.
- Muestran también:
  - Los actores involucrados en los procesos,
  - Cuáles son las actividades operativas distinguibles,
  - Que actividades son ejecutables y por quien,
  - Cuales son las entradas y salidas de actividades
  - Cuál es la secuencia de las actividades,
  - Los recursos consumidos, y
  - Los eventos que dirigen el proceso.

- La realización de las actividades de un PN necesita ser dirigida de forma similar a los instrumentos de una orquesta.



- Cuando un PN de una organización **interactúa** con otros PN de la misma o de otras organizaciones, es necesario establecer algún tipo de coordinación en la ejecución de los procesos, de igual forma que en un ballet los movimientos de los bailarines siguen todos una coreografía común.



BPM (Business Process Management) incluye conceptos, métodos y técnicas para dar soporte a

el diseño, administración, configuración, realización y análisis de procesos de negocio. (Weske, 2007)

- La base de todo es la representación explícita de los procesos con sus actividades y restricciones de ejecución.
- Una vez definido (modelado), un PN puede ser sujeto a análisis, mejora o realización.

¿Cómo vé la comunidad de Ingeniería del Software al BPM?

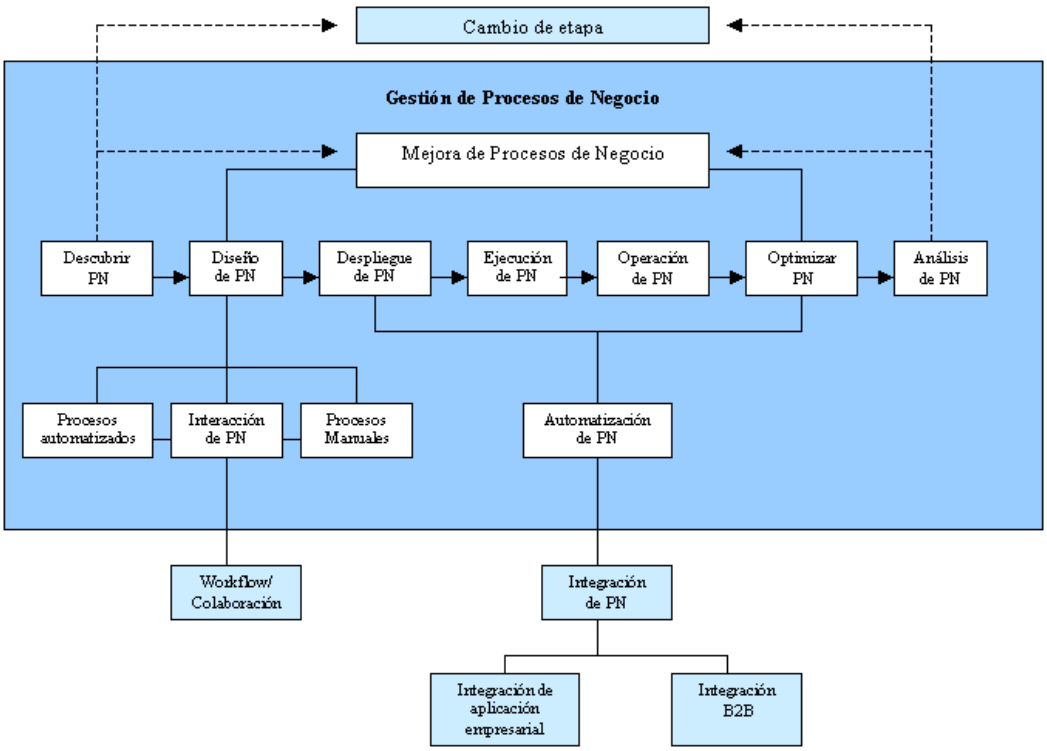
- Existe una confusión porque se habla de dos cosas diferentes:

1. Unos ven los procesos de negocio como la clave central del dominio del problema (cómo funcionan las organizaciones).

   Esta es la perspectiva del BPM.

2. Otros lo ven de manera más acotada: un medio para identificar los requisitos de un sistema software.

   Ej: En Proceso Unificado se propone usar procesos de negocio como técnica para el modelado del negocio.



- Descubrimiento
  - Hacer explícita la manera en que se hacen las cosas (frente a cómo se deberían hacer).
- Diseño
  - Modelar, simular y reestructurar el PN.
- Despliegue
  - Implantar un nuevo PN a todos los participantes (personas, sistemas, otros procesos).
- Ejecución
  - Asegurar que el nuevo PN es llevado a cabo por todos los participantes.
- Interacción
  - Permitir a las personas gestionar el interfaz entre procesos automáticos y manuales.

- Operación y Mantenimiento
  - Intervenir para resolver excepciones. Reasignar participantes.
- Optimización
  - Cambiar el PN para mejorarlo.
  - La **mejora** de procesos debe ser un esfuerzo **continuo**, en ciclos de diseño-despliegue-ejecución-operación-optimización.
- Análisis
  - Medir el rendimiento del PN e idear estrategias de mejora.
- Automatización
  - Se realiza durante las etapas de despliegue, ejecución, operación y optimización.

Business Process Management Systems

- Un **BPMS** es un sistema software genérico que permite coordinar la realización (ejecución) de procesos de negocio en base a representaciones de proceso explícitas (modelos). (Weske, 2007)

Business Process Management Systems

Sistemas TI destinados a ser el núcleo clave en la gestión de las organizaciones.

- Pretenden
  - Integrar sistemas
  - Automatizar actividades
  - Gestionar todas las fases del ciclo de vida de los PN
    - Entorno integrado.
    - Soporte a gestores, analistas de negocio, ingenieros de procesos, departamentos, empleados, implementadores TI.
  - Despliegue transparente (de la TI)
  - Proveer visibilidad y control

Similitud con los SGBD

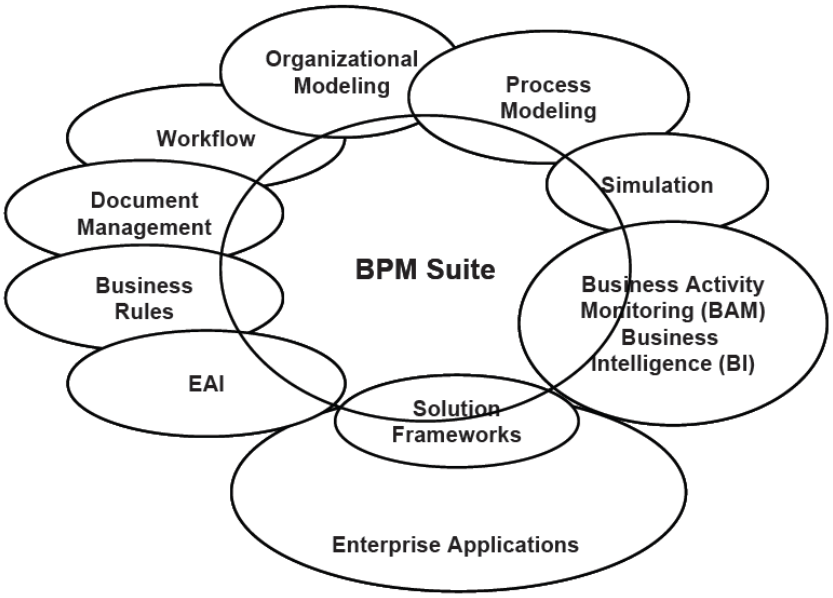
	RDBMS	BPMS
Language	Relational model/SQL User and IT interface Standards based Creates and manipulates data - tables & records Reports	BPML/BPQL User and IT interface Standards based Creates and manipulates process Reports
Engine	Executes SQL Provides logic eg stored procedures Responsible for integrity of data Creates data about the data Is a 'data state engine'	Executes BPML/BPQL Provides business logic Responsible for integrity of processes Creates data about the processes Is a 'process state engine'
Infrastructure	Maps to physical data infrastructure Interfaces to other data stores and RDBMS Retains data about the data - recovery, integrity, performance, monitoring, tuning	Maps to physical IT, business and organisational infrastructure. Retains data about the process - recovery, integrity, performance, monitoring, tuning
Tool-set	Initial focus on 'the engine' Niche players provided modelling, performance monitoring, operational tools	Initial focus on 'the engine' Existing process & enterprise modeling/simulation players provide modelling, performance monitoring & operational tools
Lifecycle	Early adopters solved niche, intractable problems Start-ups joined by major players Rationalisation of RDBMS vendors Became platform over 10-15 years	Early adopters solved niche, intractable problems Start-ups joined by major players (BPEL) Small number of strong BPMS vendors will emerge Will become platform over 10-15 years

PROCESOS

DATOS

- ¿Por qué ahora y no antes?
  - Porque hasta ahora la tecnología no había avanzado lo suficiente para integrar todas las capacidades necesarias.

- Un BPMS, también llamado BPM Suite, puede incluir múltiples partes.



BPMN

- Notación desarrollada inicialmente por BPMI (Business Process Management Initiative).
- Fusión con OMG (Object Management Group) en Junio de 2005

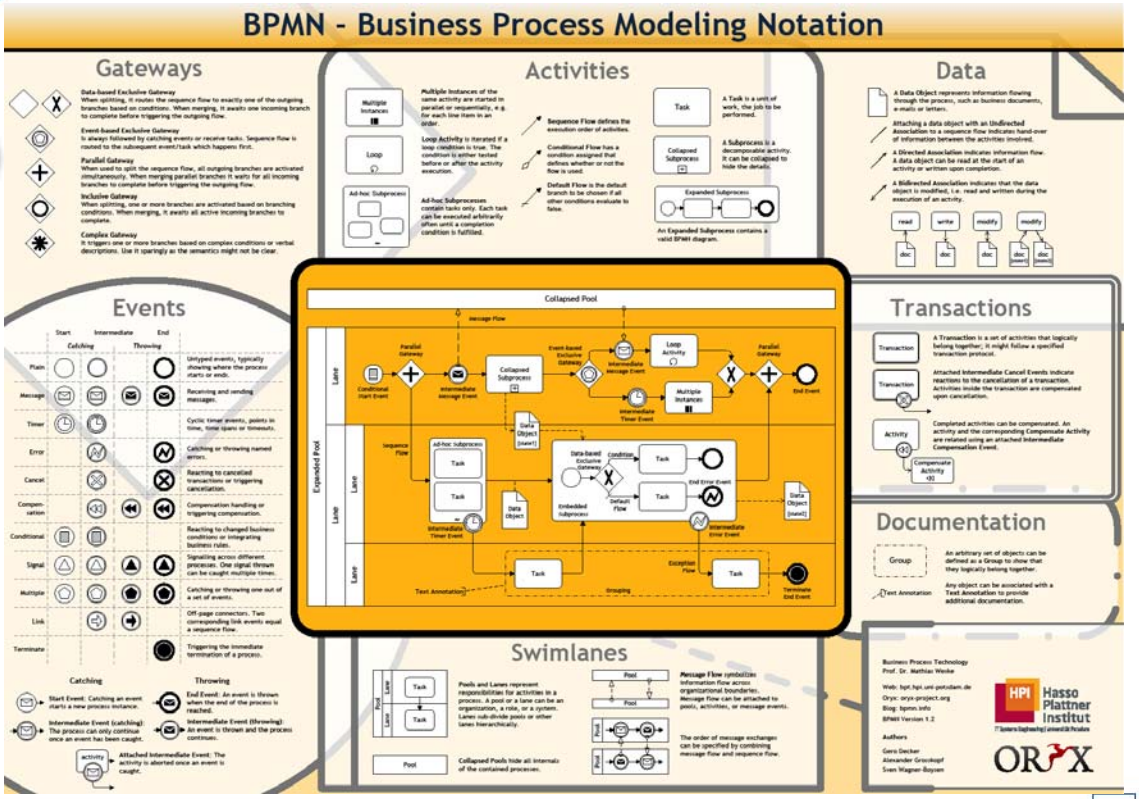
BPMI.org — Mundo de la Gestión

OMG OBJECT MANAGEMENT GROUP — Mundo de la Tecnología

BPMN Especificación aprobada (BPMI/OMG) – Feb-2006
Versión actual: 1.2 (enero-2009) *(en PDF)*

BPMN define un ***Diagrama de Procesos de Negocio*** que está basado en la técnica de diagramas de flujo y adaptado para crear modelos gráficos de las operaciones de los procesos de la organización.

Está compuesto de un conjunto de **elementos gráficos** que facilitan el desarrollo de un solo diagrama entendible tanto por audiencias de negocios (analistas de negocios) como por audiencias técnicas (arquitectos de sistemas e ingenieros software).



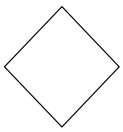
Elementos Principales de un Diagrama BPMN

Objetos de Flujo

Eventos

Actividades

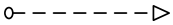
Nodos de Decisión/unión



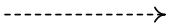
Objetos de Conexión

Flujos de Secuencia

Flujos de Mensaje



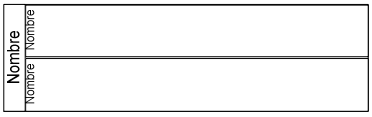
Asociación



Swimlane

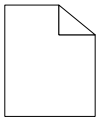
Pool (Entidad/Rol)

Lane (participante dentro de un pool)



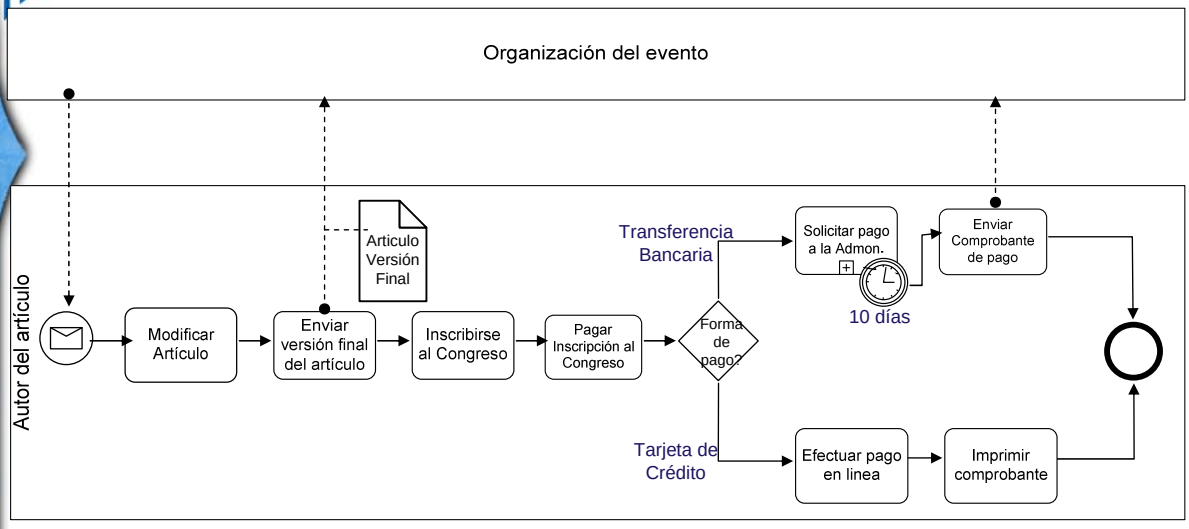
Artefactos

Objeto de datos

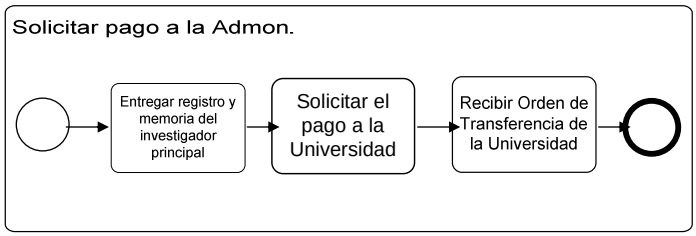


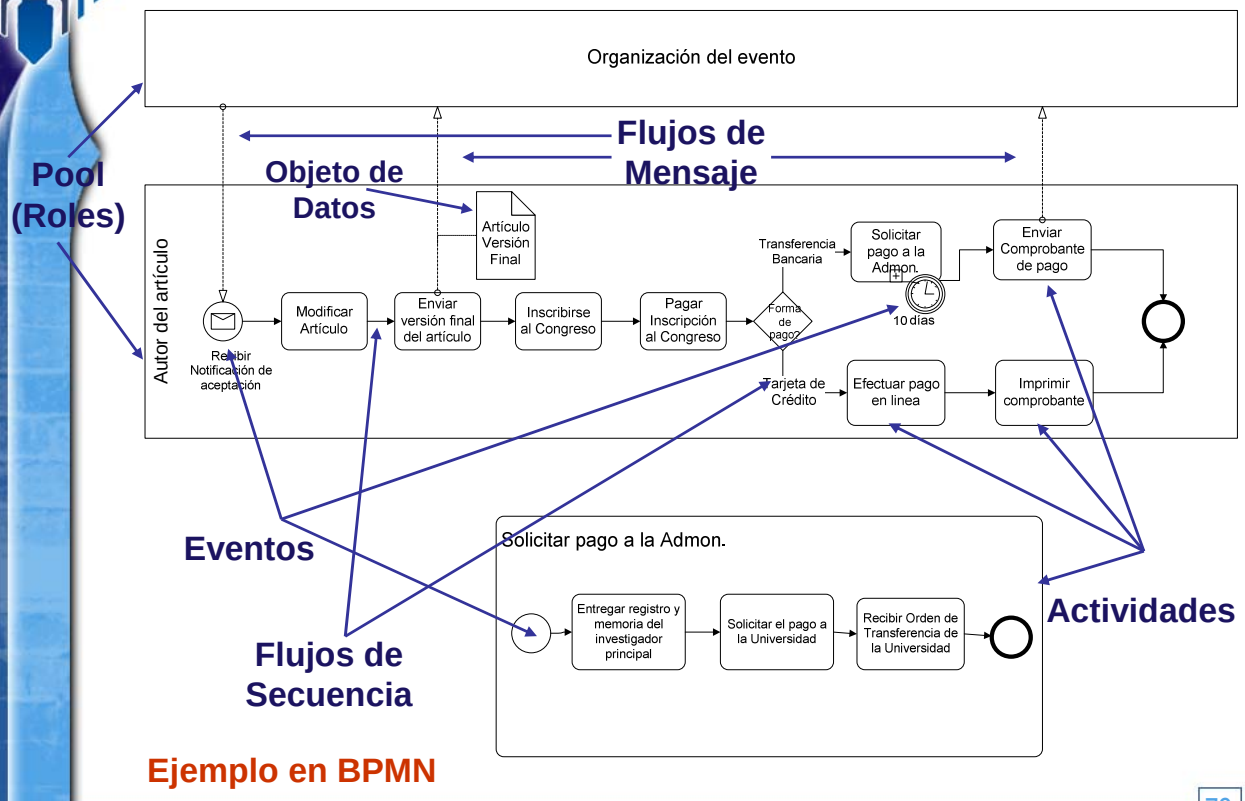
Anotación de texto

Grupo



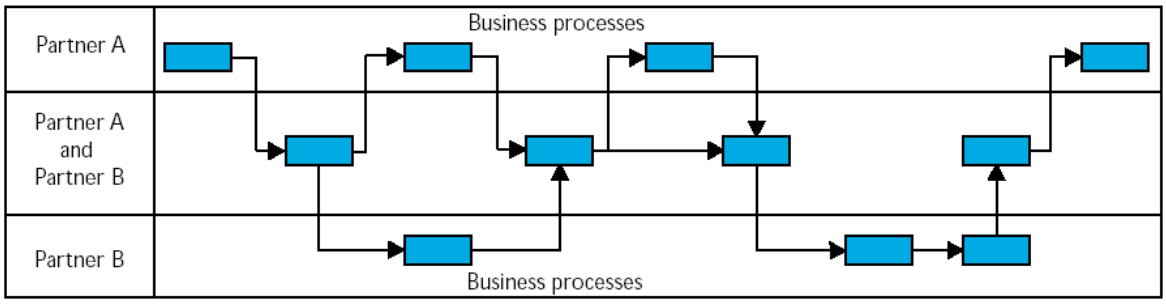
Ejemplo en BPMN



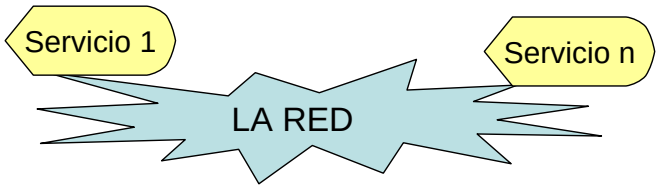


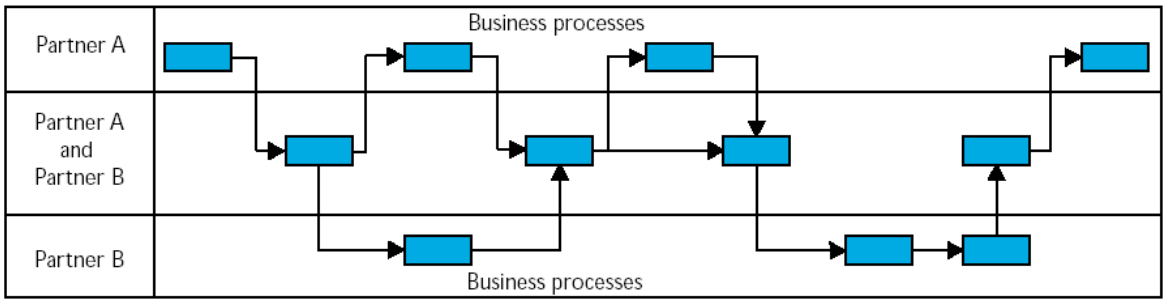
Ejemplo en BPMN

Integración BPM + SOC



Integración BPM + SOC

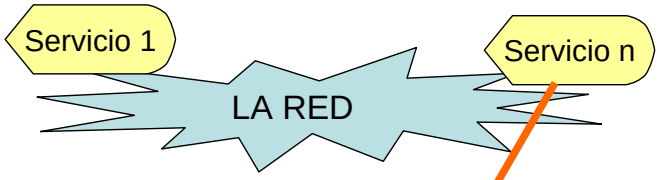


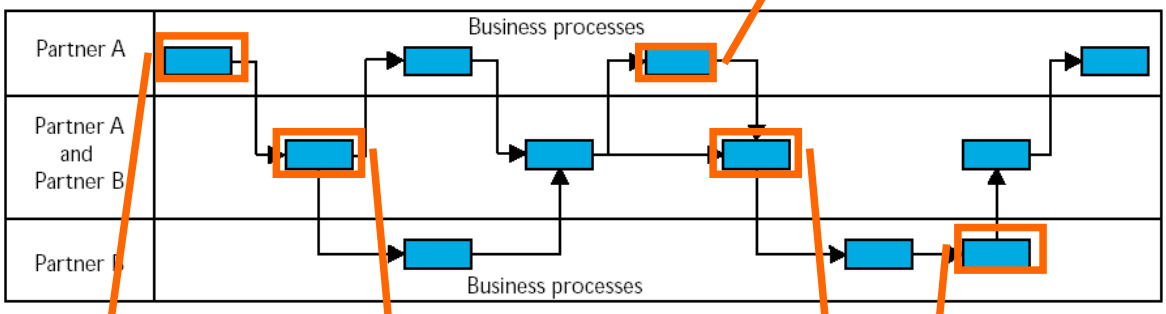


Sistema Legado 1 ... Sistema Legado n

Nuevos Paradigmas
Integración

Integración BPM + SOC

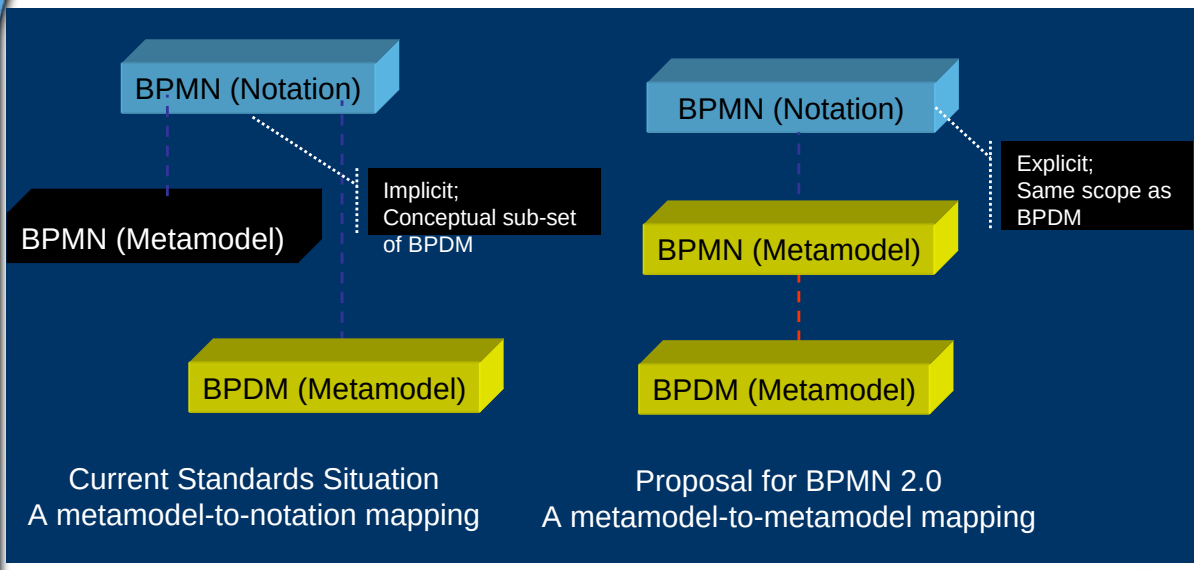




Sistema Legado 1 ... XML Sistema Legado n XML Sistema Nuevo

Integración BPM + MDE

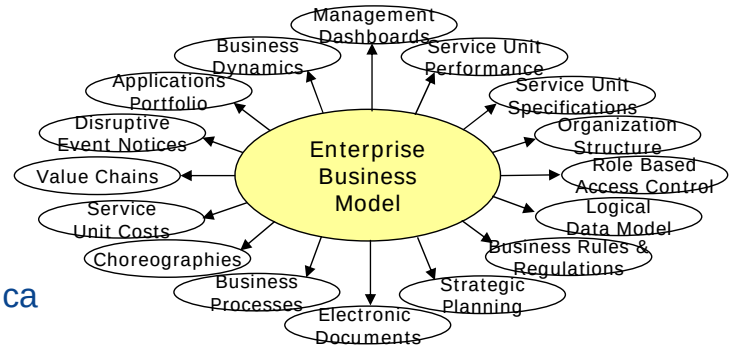
BPDM – Business Process Definition Metamodel (OMG)



La integración de BPM y MDE ha llevado al nuevo concepto de "**Model-driven Business**".

Convergencia de

- Servicios
- Procesos de negocio
- Organización
- Cadenas de valor
- Planificación estratégica
- …

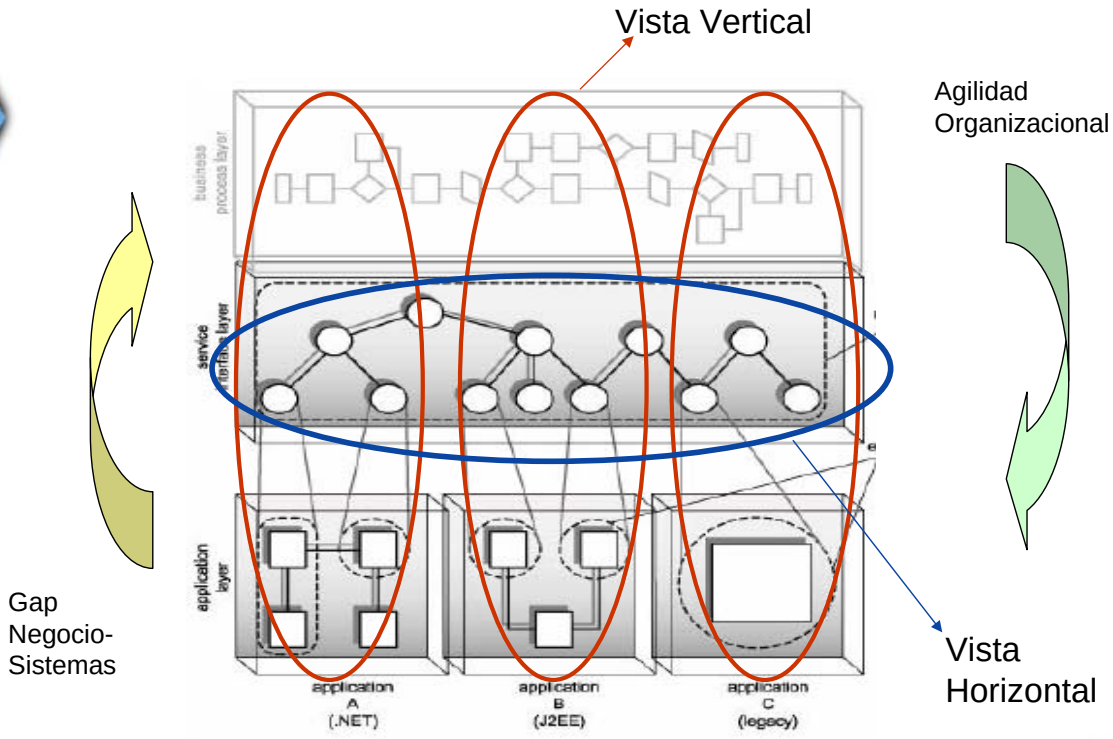


Tambien se habla de Model-based Management

Integración BPM + SOC + MDE ?

- Evolución previsible de la industria del software
  - Foco en el negocio (procesos de negocio)
  - El software como tecnología para proveer servicios
  - Los modelos son el principal tipo de artefacto

- Retos de una Organización actual
  - Agilidad para adaptarse a los cambios en el negocio adaptando y/o integrando sus procesos de negocio y tecnologías de información.
- Aspecto Clave
  - Separar la definición de los procesos de negocio de su implementación tecnológica.
  - Añadir una capa de **Servicios**



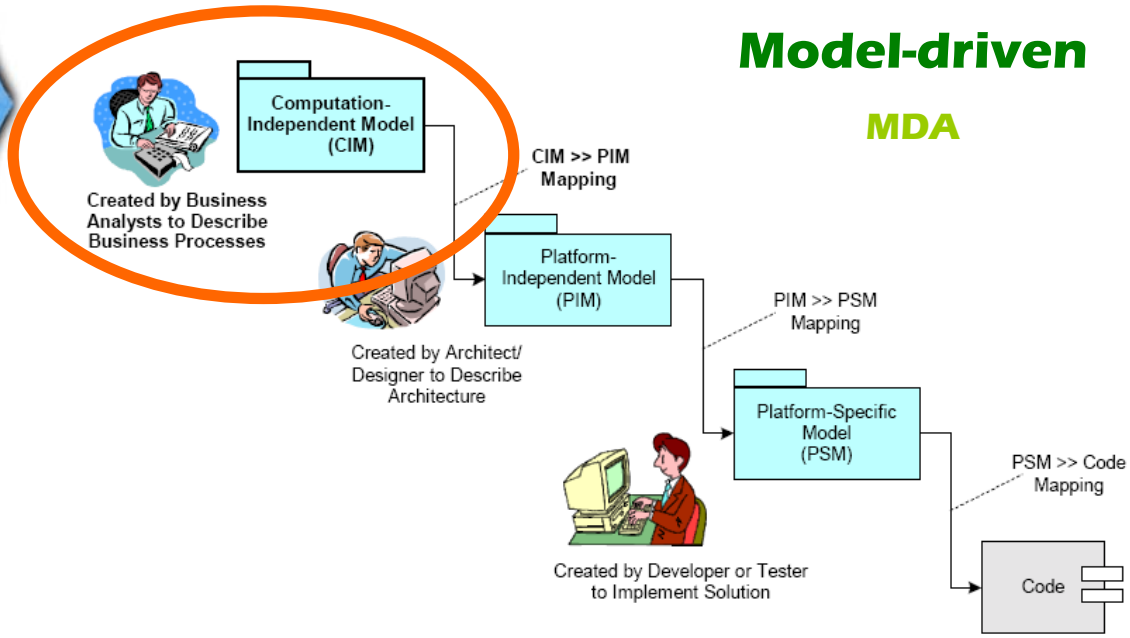
- Service Oriented Computing (SOC)
  - Desarrollar servicios basados en software, con interfaces bien definidas que permiten la interacción entre suministradores y proveedores de servicios para realizar procesos de negocio.
- Business Process Management (BPM)
  - Optimizar los procesos de negocio que satisfacen las necesidades de una organización mediante tecnología BPM, de forma que dichos procesos se implementan como secuencias de invocaciones de servicios (orquestación, coreografía).
- Model Driven Engineering (MDE)
  - Desarrollar sistemas software usando modelos como artefactos centrales. Modelos, metamodelos (modelos de modelos) y transformaciones entre ellos pueden ser aplicados para automatizar la derivación de servicios desde los procesos de negocio.

- ¿Es necesaria toda esta tecnología nueva?
  - ¿Será otra moda?.
  - ¿Otro lenguaje?; ¿es que no vale UML?.
- Si el problema es de otros (negocio/TI), ¿porqué me tengo que preocupar yo?.
- ¿Dónde se sitúa esta tecnología en relación con otras (bases de datos, etc.)?

- Ventajas de UML
  - Es un lenguaje conocido
  - Estándar
  - Fácil de aprender
- Desventajas de UML
  - No ha sido diseñado para modelar procesos de negocios
    - => No está orientado al dominio del problema
  - Implica un enfoque orientado a objetos
    - => Contradictorio con un enfoque “orientado al negocio”
  - Sólo lo conocen los expertos TI.
  - UML no tiene todavía una semántica formal.
    - BPMN sí (basada en el Π calculo).

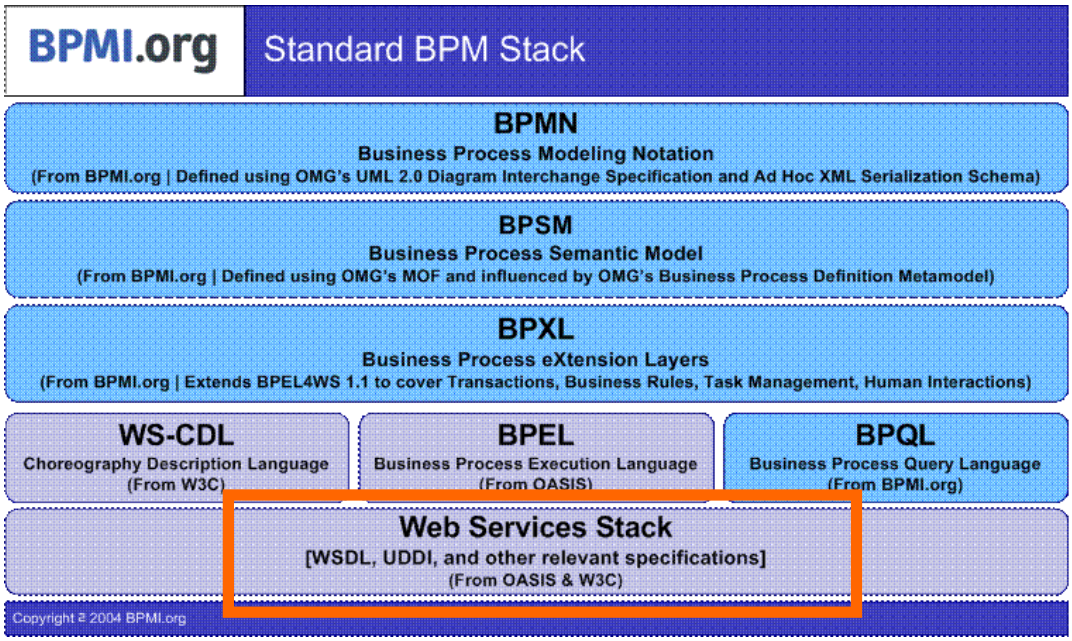
- Es problema del médico comprender lo que le pasa al paciente.
- Es problema del decorador de interiores comprender lo que le gusta a su cliente.

- Es problema del ingeniero informático comprender el dominio del problema
  - Sin ello, es imposible construir una solución realmente útil.
- Somos nosotros los que debemos acercarnos al usuario, y no al contrario.
  - Mirar la historia de la Informática.



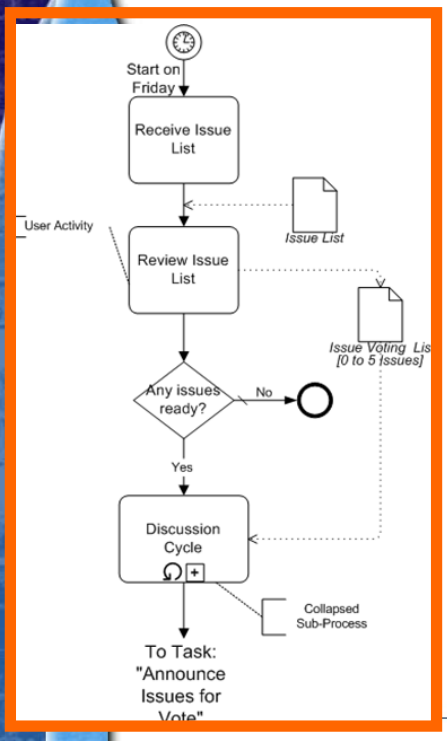
Service-oriented

Servicios Web



XML

XSD, XPDL, BPDM




```
<process name="EMailVotingProcess">
   <!-- The Process data is defined first-->
   <sequence>
      <receive partnerLink="Internal" portType="tns:processPort"
               operation="receiveIssueList" variable="processData"
               createInstance="Yes"/>
      <invoke name="ReviewIssueList" partnerLink="Internal"
              portType="tns:internalPort" operation="sendIssueList"
              inputVariable="processData" outputVariable="processData"/>
      <switch name="Anyissuesready">
         <!-- name="Yes" -->
         <case condition="bpws:getVariableProperty(ProcessData,NumIssues)>0">
            <invoke name="DiscussionCycle" partnerLink="Internal"
                    portType="tns:processPort" operation="callDiscussionCycle"
                    inputVariable="processData"/>
            <!- Other Activities not shown -->
         <!--name="No" -->
         </case>
         <otherwise>
            <empty/>
         </otherwise>
      </switch>
   </sequence>
</process>
```

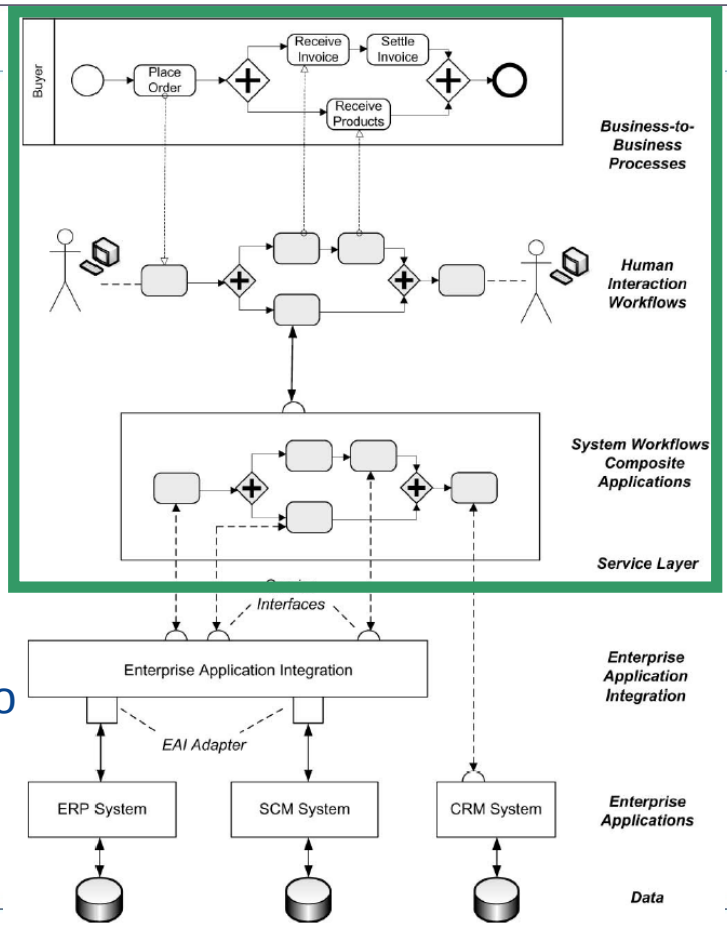
- Estos Paradigmas vienen para quedarse
  - Son un paso más en la historia de cómo nos enfrentamos al objetivo central de la **Informática**,

    "Resolver las necesidades de información de la gente mediante sistemas basados en computador"
- Abordan problemas que hasta ahora no se habían podido resolver de una manera adecuada.
  - Ni siquiera habíamos pensado en cómo resolverlos porque teníamos otros más cercanos.
    - Integración de sistemas
    - Complejidad del diseño y creación de los sistemas
  - ¿Ultimo paso en nuestro acercamiento a los usuarios?

    DEL BIT AL NEGOCIO/ORGANIZACIÓN

BPM

y su contexto



- BPM + SOC + MDE
  - Profesionalidad
  - Aprendizaje a lo largo de toda la vida
  - Conocimiento vs Expertez

La potencia sin control no sirve de nada

Evolución horizontal de la manera de abordar el problema de construir software
- Programación estructurada
- Programación OO
- Model-driven

Evolución vertical del problema de diseñar sistemas de calidad (que satisfagan las necesidades de los clientes/usuarios)
- Codificación
- Diseño
- Análisis
- Requisitos
- ¿Negocio?

Dilema
- Nuestro problema versus necesidades de la gente
- SISTEMA SOFTWARE vs SISTEMA DE INFORMACIÓN/NEGOCIO

- Hacer Software es un problema complejo y seguirá siéndolo.
- La Ingeniería del Software pretende resolverlo mediante la aplicación de maneras sistemáticas y metódicas de trabajar (igual que hicieron hace tiempo otras ingenierías).
- Es vital para el futuro (profesional, laboral y académico) de la Informática que se incida más en la perspectiva de ingeniería.
  - Más arquitecto, menos albañil.

- Hacer software es una actividad humana muy intensa en conocimiento.
- A lo largo de los últimos 50 años se han ido desarrollando diferentes maneras (paradigmas) de abordar el problema.
  - La programación estructurada, la modularidad, la orientación a objetos, etc., permitieron encontrar una buena manera de resolver algún aspecto del problema de hacer software.
- Recientemente se han desarrollado nuevos paradigmas que intentan ayudar a resolver algunos de los problemas y retos que todavía están por resolver o mejorar.

- La Ingeniería dirigida por Modelos (**MDE**) busca conseguir que el artefacto central no sea el código sino modelos de diversos tipos y niveles, de forma que se pueda pasar de unos modelos a otros mediante transformaciones automáticas, hasta llegar al código.
- La Orientación a Servicios (**SOC**) postula desarrollar y utilizar el software en base a la idea de servicio en vez de la tradicional de producto.
- La Gestión de Procesos de Negocio (**BPM**) pretende aportar una solución integrada a las organizaciones desde el punto vista de procesos (que es lo que una organización conoce) en lugar del foco en los Sistemas, tradicional en Informática.

- La **integración** de estos tres paradigmas crea unas sinergias que abren nuevas oportunidades y ventajas a las organizaciones que hacen software, así como una mejora global de la competitividad de cualquier tipo de empresa.

Francisco Ruiz

http://alarcos.inf-cr.uclm.es/per/fruiz/
francisco.ruizg@uclm.es