



Resumen del Curso

Proceso Software: Conceptos, Estándares, Modelos, Arquitecturas y Herramientas

Francisco Ruiz

Universidad de Castilla-La Mancha

Escuela Superior de Informática

Índice (i)

- Objetivos
- Justificación
- Procesos de negocio.
- Proceso software (PS).
- Tecnología de proceso software.
- Proceso vs modelo de proceso (MP).
- Elementos de un MP.
- Niveles y vistas de un MP.
- Evolución de un PS - Metaproceso.
- Cuestiones.

Índice (ii)

- Lenguajes de Modelado de PS (LMP).
 - Requerimientos en LMP.
 - Propiedades de un LMP.
 - Tecnologías útiles.
 - Taxonomía.
 - Ejemplo: MVP-L.
- Entornos de Ingeniería del Software.
 - Conceptos básicos.
 - Modelo de servicios: ISO 15940.
 - Arquitectura PSEE.
- Lecturas

Objetivos

- Ingeniería del Software- Problemática actual:
 - El desarrollo y mantenimiento de software es un trabajo altamente complejo.
 - Los proyectos software son difíciles de gestionar.
 - La tecnología de **Proceso Software** (PS) intenta simplificar la gestión de proyectos software.
- Este tema introduce algunos de los principales aspectos de dicha tecnología:
 - Conceptos, Modelos, Arquitecturas y Herramientas.

Justificación (i)

- La importancia que la tecnología de PS tiene dentro de la Ingeniería del Software (IS) se comprueba viendo su aparición en el **SWEBOK** (*Software Engineering Body of Knowledge*):
 - 3 de las 10 áreas de conocimiento que forman la IS se refieren a esta tecnología.
 - también utiliza 4 de las 7 disciplinas relacionadas.
- En suma, se trata de incidir más en los aspectos ingenieriles.

Justificación (ii)

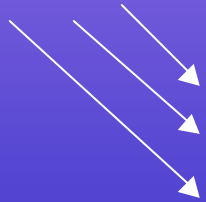
- SWEBOK es un proyecto conjunto de **IEEE-CS** y **ACM**.
- Está siendo debatido en el Subcomité JTC1/SC7 de ingeniería del software de ISO.
- Los objetivos principales de SWEBOK son cinco:
 - *Promover una visión consistente del mundo de la IS.*
 - *Clarificar el papel –y delimitar las fronteras- de la IS con respecto a otras disciplinas asociadas: ciencia de la computación, gestión de proyectos, ingeniería de computadores, y matemáticas.*
 - *Caracterizar los contenidos de la disciplina.*
 - *Proveer acceso a los contenidos del cuerpo de conocimientos.*
 - *Proveer las bases para desarrollar planes de estudios o materiales para certificaciones individuales.*

Justificación (iii)

- SWEBOK considera que la IS está formada por 10 áreas de conocimiento, y que tiene 7 disciplinas relacionadas:

<i>Áreas de Conocimiento</i>	<i>Disciplinas Relacionadas</i>
<u>Requisitos Software</u>	Ciencias Cognitivas y Factores Humanos
<u>Diseño de Software</u>	Ingeniería de Computadores
<u>Construcción de Software</u>	Ciencia de la Computación
<u>Prueba del Software</u>	Gestión y Ciencia de la Gestión
<u>Mantenimiento del Software</u>	Matemáticas
<u>Gestión Configuración Software</u>	Gestión de Proyectos
<u>Gestión de la IS</u>	Ingeniería de Sistemas
<u>Proceso de IS</u>	
<u>Herramientas y Métodos en IS</u>	
<u>Calidad del Software</u>	

PS



Proceso de negocio: definición

- Definición (Sharp, 2001):

Colección de tareas de trabajo interrelacionadas, iniciadas en respuesta a un evento, que permiten alcanzar un resultado específico para el cliente del proceso.

- Es decir, un proceso de negocio (PN) es un proceso para entregar un resultado a un cliente.

Proceso de negocio: características

- Otras características son:
 - Medible: debe poderse medir el PN en la forma que interese a los participantes (*stakeholders*).
 - Automatizable: las tareas pueden ser manuales, semiautomáticas y automáticas.
 - Niveles: se puede definir a distintos niveles de detalle (hitos, flujos de trabajo, etc.).
 - Para un cliente interno o externo.
 - Están ocultos dentro de la organización.
 - Se tiende a definir PN demasiado pequeños.

Proceso Software: definición

- Un proceso software (PS) es

Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software.

(Fugetta, 2000)

Proceso Software: apoyos

- Un PS aprovecha diversas contribuciones y conceptos:
 - **Tecnología de desarrollo de software:** herramientas, infraestructuras y entornos.
 - **Métodos y Técnicas de desarrollo de software:** cómo usar la tecnología.
 - **Comportamiento Organizacional:** la ciencia de las organizaciones y la personas.
 - **Marketing y economía:** cómo cualquier otro producto, el software debe dirigirse a clientes reales y, por tanto, está sujeto a las reglas del mercado.

Proceso Software: vs PN

- Por tanto, debemos prestar atención a la compleja interrelación que se produce en un PS entre los diversos factores organizacionales, culturales, tecnológicos y económicos.
- =>
- Un PS es un PN realizado por una organización para desarrollar y mantener un producto software.

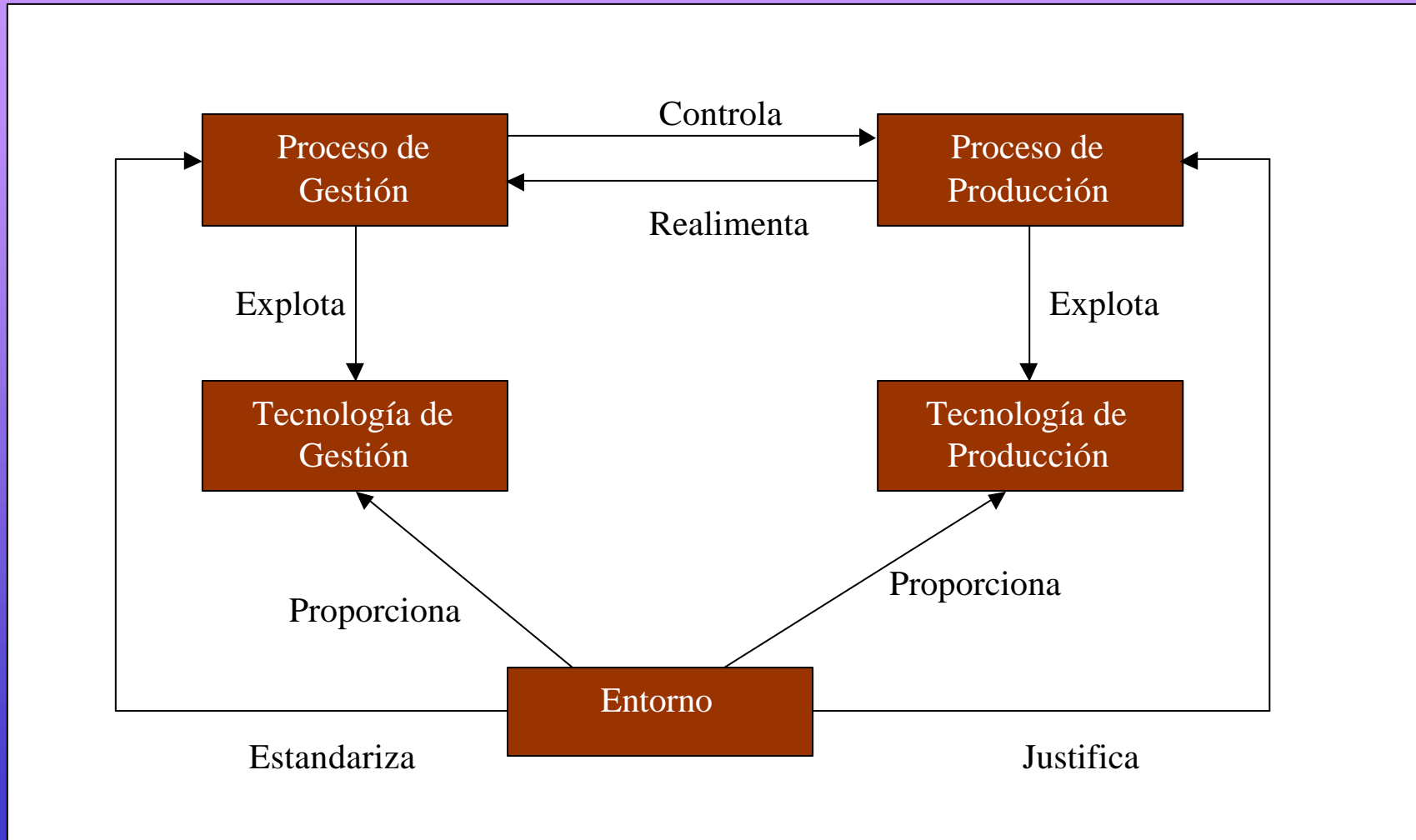
Procesos Software: naturaleza (i)

- Son complejos:
- No son procesos de producción:
 - Dirigidos por excepciones,
 - Muy determinados por circunstancias impredecibles,
 - Cada uno con sus peculiaridades.
- No son procesos de ingeniería “*pura*”:
 - Desconocemos las abstracciones adecuadas,
 - Dependen demasiado de demasiada gente,
 - Diseño y producción no están claramente separados,
 - Presupuestos, calendarios, calidad no pueden ser planificados de forma fiable.

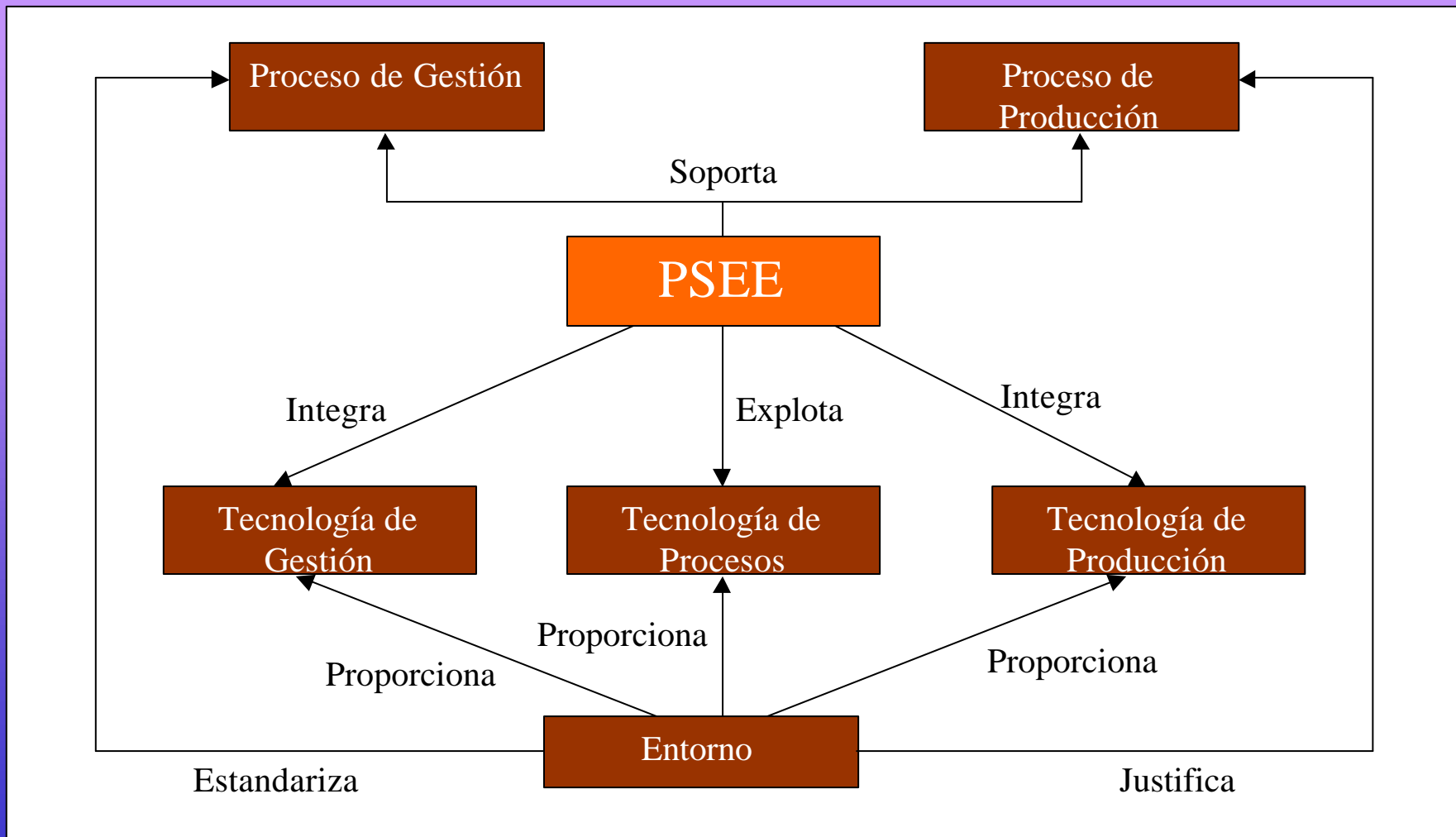
Procesos Software: naturaleza (ii)

- No son (*completamente*) procesos creativos:
 - Algunas partes pueden ser descritas en detalle,
 - Algunos procedimientos han sido impuestos.
- Están basados en descubrimientos que dependen de la comunicación, coordinación y cooperación dentro de marcos de trabajo predefinidos:
 - Los entregables generan nuevos requerimientos,
 - Los costes del cambio del software no suelen reconocerse,
 - El éxito depende de la implicación del usuario y de la coordinación de muchos roles (ventas, desarrollo técnico, cliente, etc.).

Tecnología de PS: fabricación vs PS



Tecnología de PS: impacto



Tecnología de PS: objetivo

Dominar la complejidad inherente al PS mediante una comprensión profunda del proceso en sí mismo y mediante un soporte automatizado por medio de un PSEE

Tecnología de PS: comienzos

- Riddle and William, 1986: El PS es la secuencia de actividades realizadas durante la creación y evolución de un sistema software.
- Dowson, 1985: PS es una colección de actividades relacionadas, vistas como un proceso coherente y razonado, envueltas en la producción de un sistema software.
- Wileden, 1985: Un lenguaje adecuado para expresar modelos de PS recibe el nombre de metamodelo de PS.

Proceso vs Modelo de Proceso (MP) (i)

- Los procesos de diferentes proyectos tienden a seguir patrones comunes.
- Es necesario intentar capturar estos aspectos comunes en una **representación del proceso**, la cuál describe estas características comunes y fomenta la homogeneidad.
- El estudio de los procesos de producción de software ha llevado al desarrollo de varios *Ciclos de Vida* del software que pueden ser empleados en la IS: en Cascada, Evolutivo y en Espiral.
 - Estos modelos del Ciclo de Vida ayudan a comprender mejor el PS, y a determinar el orden de actividades globales envueltas en la producción de software.

Proceso vs MP (ii)

- Pero los Ciclos de Vida no dan importancia a procesos que son cruciales para el éxito de proyectos software:
 - Gestión de la Configuración.
 - Identificar medidas y métricas, de acuerdo al aseguramiento de calidad requerido y políticas de control.
 - Gestión de Proyectos.
 - Mantenimiento (control de cambios, corrección de errores, ...).
 - Reutilización de elementos software.

Proceso vs MP (iii)

- El objetivo final de la tecnología de PS es lograr que la representación de un proceso pueda ser usada para conducir los actuales procesos de desarrollo y mantenimiento del software.
- Con tal fin, surgen varios conceptos:
 - **PSEE**: *Process-sensitive Software Engineering Environment*.
 - **Modelo de Procesos (MP)**: representación abstracta de una familia de procesos expresada en una adecuada **notación de modelado de procesos (formalismo)**.

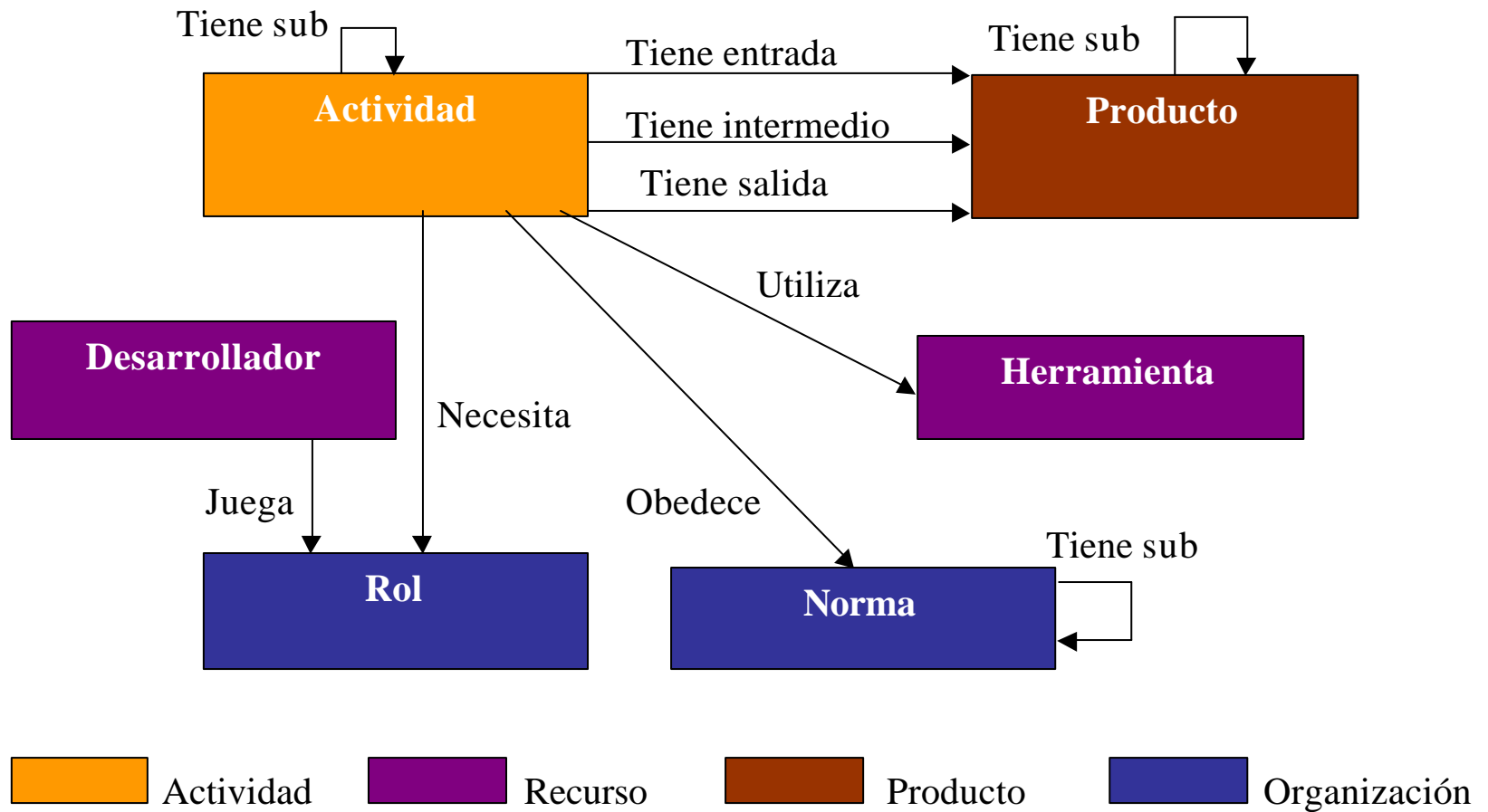
Proceso vs MP (iv)

- La disponibilidad de un MP (*computerizado*) proporciona capacidades para:
 - completar procesos: soporte directo a los desarrolladores (vía control de su trabajo, su coordinación con otros, etc.);
 - automatización: invocación automática de herramientas no interactivas);
 - dirección: soporte indirecto, como información del estado actual del proceso, el significado de los puntos de decisión, etc.
 - eficiencia: un MP preciso es primordial en el aumento de la efectividad, ya que proporciona una base no ambigua para la comunicación entre los procesos.

Proceso vs MP (v)

- Los MP juegan un rol esencial en la supervisión, simulación, validación, verificación y mejora de procesos:
 - supervisión: permiten una clara comprensión de lo que puede ser observado y por qué;
 - simulación: el comportamiento de un proceso puede ser estudiado al menor coste sin desarrolladores reales o herramientas.
 - validación: la simulación y supervisión, junto con la inspección de modelos, permiten que las propiedades de los modelos sean determinadas.
 - verificación: son necesarios para probar formalmente las propiedades de interés del proceso.
 - mejora: todas las capacidades anteriores son necesarias para la mejora de procesos.

Elementos de un MP



Niveles de un MP (i)

- Los procesos pueden ser representados con niveles incrementales de detalle, capturando sub-procesos cada vez más pequeños, correspondientes a asuntos cada vez más detallados:
 - Ciclo de Vida;
 - MP genérico;
 - MP detallado (*customized*);
 - Reificable (*enactable*);
 - Reificado (*enacting*).

Niveles de un MP (ii)

- Además, se distinguen dos dominios en cada nivel:
 - **Realización** (*performance*) del proceso:
 - concierne al acto de participar en un proceso, personas y herramientas.
 - **Reificación** (*enactment*) del proceso:
 - relacionado con el acto de conducir automáticamente el proceso, es decir, interpretar con más detalle el MP.
- Dualidad de pertenencia a los dominios:
 - Las herramientas y las personas, al ser condicionadas por el entorno, también pertenecen al dominio de reificación.
 - *Mientras que un proceso se realiza en el mundo real, es reificado en el mundo abstracto del modelo computerizado.*
 - Puesto que los computadores son parte del mundo real, se dice que la reificación es parte de la realización.

Vistas de un MP (i)

- Expresan un punto de interés particular en vez del MP completo (similar a vistas en BD):
 - Sub-modelos (en modelado *bottom-up*).
 - Modelos parciales (en modelado *top-down*).
- Las más habituales son:
 - De Actividades,
 - De Productos,
 - De Recursos, y
 - De Roles.
- Nos son disjuntas: una vista no puede ser definida sin usar conceptos de otras.

Evolución de un PS (i)

- Se deben considerar dos aspectos en un entorno de producción de software:
 - Un **proceso en desarrollo**, P, es decir el proceso de producción del mundo real incluyendo actores humanos y herramientas que acompañan a todas las actividades dirigidas al desarrollo y mantenimiento de un producto software, y
 - Un **modelo de procesos**, MP, que es una representación del mundo real, y captura el estado actual de las actividades para guiar, hacer cumplir o automatizar partes del proceso de producción o de mantenimiento.

Evolución de un PS (ii)

- Idealmente, P y MP deben estar perfectamente alineados: el estado interno de MP debe ser una fiel representación del actual estado de los asuntos en el mundo real.

⇒ P es una instancia de PM

- Pero cualquier PS en el mundo real es un proceso creativo y dinámico que abarca a mucha gente, y no puede ser reducido a la programación de autómatas.

⇒ El MP debe coordinar actividades y gestionar el flujo de información, y

⇒ MP debe adaptarse a cualquier evolución del proceso P en el mundo real.

Evolución de un PS (iii)

- Hay diversas razones por las que puede cambiar un PS:
 - puede ser erróneo;
 - ciertos pasos importantes no están previstos;
 - el MP puede ser genérico y necesita ser detallado para obtener resultados específicos;
 - las presunciones sobre las cuales se construyó el MP ya no son válidas;
 - las dinámicas políticas, humanas y tecnológicas pueden inducir a su cambio.
- Como resultado, el proceso P del mundo real y el MP deben evolucionar de forma conjunta y coherente.

Metaproceto

- Conclusión: *la evolución de un PS es en sí misma un proceso completo.*
- Este proceso de alto nivel es llamado **meta-proceso**:
 - El gestor del proyecto necesitará implicar los servicios del modelador para desarrollar un MP aumentado, validar el nuevo modelo, y decidir cuando comenzar a realizarlo.
 - Este metaproceto incluye:
 - Los pasos para cambiar los procesos del mundo real, y
 - Los pasos para introducir cambios en el MP.
 - Su principal función es asegurar que P y MP permanecen consistentes.

Cuestiones (i)

- Surgen una serie de preguntas que debemos plantearnos e intentar responderlas:
 - ¿Cómo construimos un MP?
 - ¿Cómo lo reificamos?
 - ¿Qué características debe tener el formalismo a utilizar?
 - ¿Cómo se puede cambiar y mejorar un PS y su MP asociado?
 - ¿Qué arquitectura debe tener un PSEE?
 - ¿Qué papel desempeñan las personas?
 -

Lenguajes de Modelado de PS

- Un lenguaje de modelado de PS (LMP) expresa los procesos software (PS) en forma de modelos de procesos software (MP).
- Todos los elementos de proceso (actividades, roles, recursos, etc.) deben poderse describir.
- Los elementos del metaproceso (evolución del proceso) también se deben poder expresar.
- Un LMP puede ser:
 - Formal: tiene sintaxis y semántica formales.
 - Semi-formal: tiene notación formal (normalmente gráfica) pero no tiene semántica formal.
 - Informal: sin sintaxis y semánticas formales (leng. natural).

Modelo de PS

- Representación de las actividades del mundo real de un PS.
- Un MP es desarrollado, analizado, refinado, transformado y reificado dentro del metaproceso.
- Por tanto, un MP debe:
 - Modelar adecuadamente el proceso del mundo real, y
 - Satisfacer los requisitos específicos de cada fase del metaproceso.
- El LMP debe adaptarse a estas necesidades del MP.

Requerimientos en LMP (i)

- Hay 6 elementos de proceso (primarios) que debe poder modelar un LMP:
 - Actividades.
 - Productos.
 - Roles.
 - Personas.
 - Herramientas.
 - Soporte para la evolución (al menos del MP):
 - a nivel *técnico* (p.e., mediante reflexión o interpretación), y
 - a nivel *conceptual* (mediante un metamodelo asociado).

Requerimientos en LMP (ii)

- Los **elementos del metaproceso** (secundarios) son:
 - Proyecto/Organización.
 - Organizaciones consisten de humanos relacionados con otros humanos y elementos.
 - Un proyecto es una estructura temporal de la organización montada para poder alcanzar un objetivo específico.
 - Contexto de Trabajo.
 - Formado por espacios de trabajo.
 - Cada espacio de trabajo contiene y controla artefactos para un (sub)proceso.
 - Estos artefactos suelen ficheros en un repositorio.
 - Vista de usuario.
 - Interfaz general para ayudar al usuario a comprender el MP y guiarlo durante su reificación.
 - Hay un modelo interno (cómo trabaja) y otro externo (cómo se hace).

Requerimientos en LMP (iii)

- **Elementos del metaproceso (cont.):**
 - Modelo de Cooperación.
 - Permitir modos de cooperación secuencial y paralelo.
 - Incluir protocolos de comunicación de objetos.
 - Coordinación de acciones (ordenación y sincronización).
 - Modelo de Versionado/Transacciones.
 - Modelo de Calidad/Rendimiento.
 - Modelo de Calidad del Producto, incluye objetivos de calidad del producto y métricas asociadas.
 - Modelo de Rendimiento del Proceso, expresa el cumplimiento con respecto a tiempos, costes, roles, etc.

Requerimientos en LMP (iv)

- Un LMP se usa de diferente forma por diferentes roles durante las diferentes fases del metaproceso; por tanto, en cada fase interesan unas características diferentes en el LMP.

Ejemplos:

- En *Especificación de requisitos del proceso*: orientado al modelado conceptual, intuitivo y con notación fácil para los usuarios no técnicos (gráfica).
- En *Implementación del Proceso*: debe permitir el suficiente detalle para que el MP sea reificado; por tanto, el LMP debe ser ejecutable (formal).

Propiedades de un LMP (i)

- Formalidad.
- Expresividad.
- Comprensibilidad.
- Abstracción y modularidad.
- Ejecutabilidad.
- Analizabilidad.
- Soporte de evolución.
- Múltiples vistas.

Tecnologías útiles

- *Gestión de proyectos*: diagramas de barras, redes de actividades.
- *Lenguajes de especificación formal*: redes de Petri (SLANG).
- *Notaciones informales de diseño*: OO (E3), UML.
 - Estáticas: ER.
 - De comportamiento: diagramas de transición de estados.
 - Funcionales: diagramas de flujo de datos.
- *Lenguajes de programación*: ADA (APPL/A).
- *Lenguajes de bases de datos*: BD activas (ADELE).
- *Herramientas CASE y mecanismos de integración*.
- *Flujos de Trabajo y Trabajo en Grupo*.

Taxonomía (i)

- Según el elemento del proceso en el que se centran:
 - Producto: EPOS, Adele.
 - Actividades: MARVEL, MERLIN, SLANG.
 - Proyecto: MS-Project.
 - Roles: PWI.
- Otra clasificación alternativa:
 - Funcionales: centrados en describir las actividades.
 - De Comportamiento: centrados en cuando y como se realizan las actividades.
 - Organizacionales: centrados en el “cuando” y “por quién”.
 - Informacionales: centrados en los artefactos y procesos y sus asociaciones.
- Otra clasificación es según la fase del metaproceso a la que se orientan:
 - elicitación, análisis, diseño, implementación, reificación, y evaluación.

Taxonomía (ii)

- **Formalismos utilizados en LMP:**
 - Lenguajes de Programación: APPL/A, JIL.
 - Reglas: Marvel, Oz, Atlantis.
 - Orientación a Objetos: E3, EPOS/Spell.
 - Grafos y gramáticas: Hakoniwa.
 - Redes de Petri: SLANG/SPADE.

Ejemplo: MVP-L

- **Multi View Process Modeling Language**
- Desarrollado en las universidades de Maryland y Kaiserslautern (Rombach, Marsh, Lott, Bröckers, Verlage).
- **Objetivo principal:**
 - Modelado descriptivo de grandes procesos del mundo real para comprender, analizar, guiar y mejorar los proyectos de desarrollo de software.
- MVP/L 2.0 Language Report: `\tema3\MVPLv2.pdf`

MVP-L: objetivos

- Construir MP descriptivos
- Instanciarlos en planes de proyecto prescriptivos.
- Analizar los planes de proyectos.
- Orientar en la reificación de los proyectos.
- Paquetes de MP para reutilización.
- Documentar el histórico de reificaciones.
- Descomponer la descripción de los procesos en diferentes vistas.

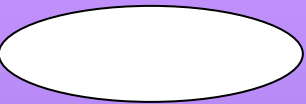
MVP-L: constructores (i)

- **Modelos** (Tipos) para la descripción de
 - Modelos de proceso: describen procesos, flujos de información, y descomposición del trabajo.
 - Modelos de producto: atributos de los productos creados durante la ejecución.
 - Modelos de recurso: herramientas y personas.
 - Los modelos pueden ser adaptados al contexto actual durante su instanciación.
- **Atributos**
 - Son globales o relacionados con uno de los 3 modelos.
 - Sus valores corresponden a medidas de datos y estados.

MVP-L: constructores (ii)

- Objeto Ejecutable: plan del proyecto.
- Relaciones entre objetos:
 - Flujo de producto (consume, produce, ..)
 - Flujo de control (criterios de entrada/salida de procesos, invariantes)
 - Descomposición/Agregación (refinamiento)
 - Ocultamiento de información (interfaz de modelo vs cuerpo de modelo)

MVP-L: representación gráfica



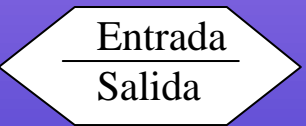
Producto



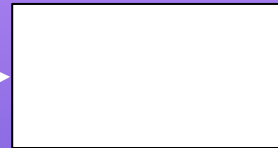
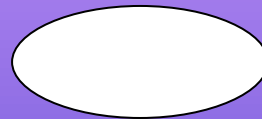
Proceso



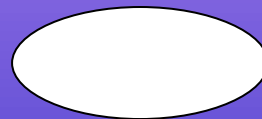
Recurso



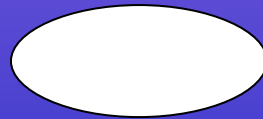
Entrada
Salida
Criterio de
Entrada/Salida



consume

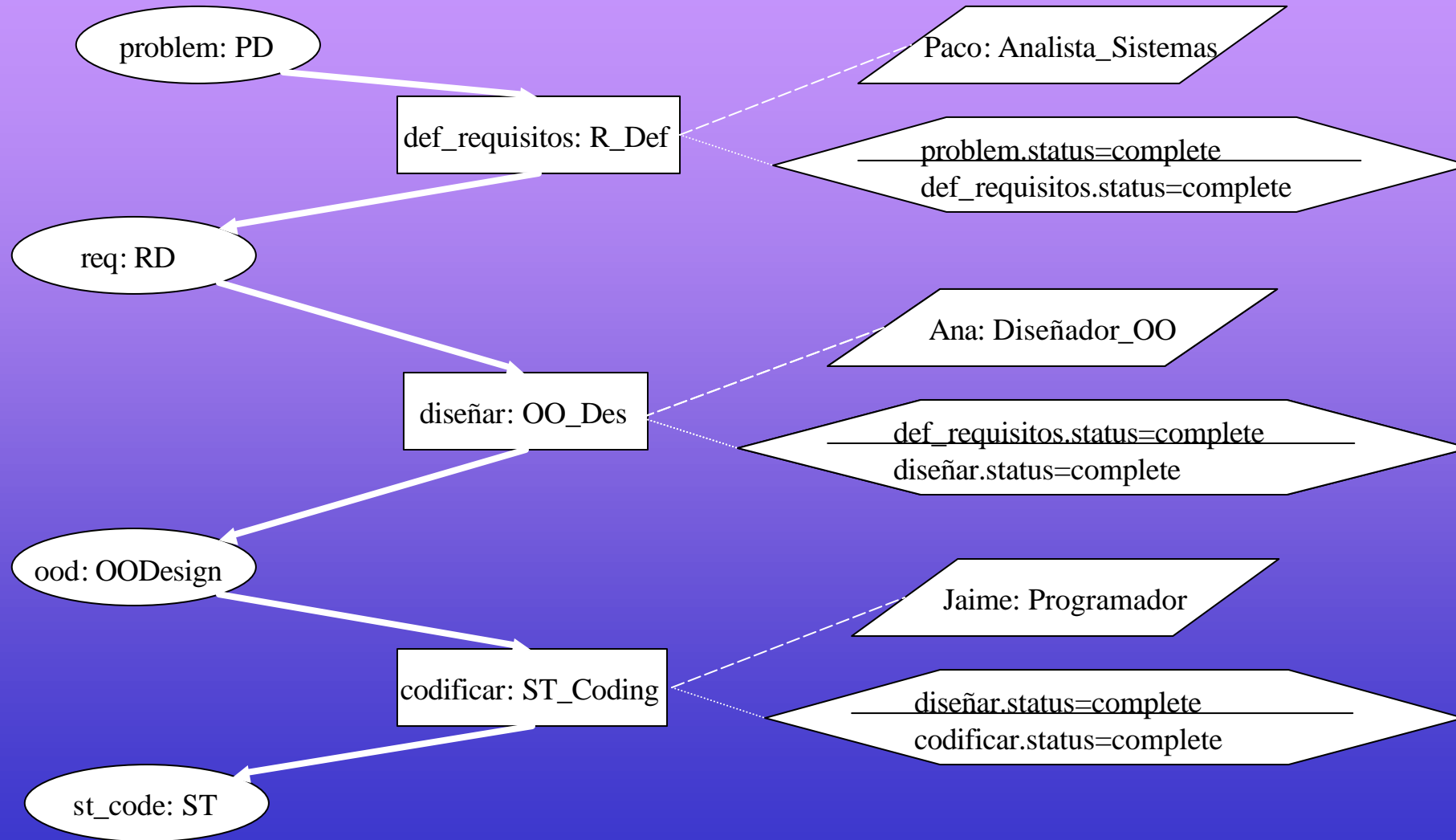


produce



consume_produce

MVP-L: ejemplo gráfico



MVP-L: ejemplo textual

```
project_plan
```

```
  imports
```

```
    product_model    PD, RD, OODesign, ST;
```

```
    process_model    R_Def, OO_Des, ST_Coding;
```

```
    resource_model    Analista_Sistemas, Diseñador_OO, Programador
```

```
  objects
```

```
    problem: PD;      def_requisitos: R_Def;
```

```
    req: RD;          diseñar: OO_Des;
```

```
    ood: OODesign;    codificar: ST_Coding;
```

```
    st_code: ST;
```

```
  object_relations
```

```
    def_requisitos(i1=>problem, o1=>req, r1=>Paco);
```

```
    diseñar(i1=>req, o1=>ood, r1=>Ana);
```

```
    codificar(i1=>ood, o1=>st_code, r1=>Jaime);
```

```
end project_plan
```

MVP-L: modelos e interfaces de procesos

- Modelo de Proceso:

```
process_model OO_Des
    process_interface
    .....
end process_interface

process_body
..... end
process_body

process_resources
.....
end process_resources

end process_model
```

- Interfaz de Proceso:

```
process_interface
    imports
        product_model          PD, OODesign;
        process_attribute_model Process_effort;
    exports
        effort: Process_effort;
    product_flow
        consume i1: PD;
        produce o1: OODesign;
        consume_produce
    context
        entry_exit_criteria
            local_entry_criteria (i1.status=complete)
            global_entry_criteria
            local_exit_criteria (o1.status=complete)
            global_exit_criteria
    end process_interface
```

MVP-L: cuerpo de un proceso

```
process_body
  refinement
    imports
      product_model    ClassDesign;
      process_model    UMLDesigning;
    objects
      c1, c2: ClassDesign;
      cDes1, cDes2: UMLDesigning;
    object_relations    # describiendo una agregación
      (cDes1 & cDes2)
    interface_refinement
      o1 = (c1 + c2)
    interface_relations
      cDes1(i1=>i1, o1=>c1)
      cDes2(i1=>i1, o1=>c2)
    attribute_mapping
      effort := cDes1.effort + cDes2.effort
      status:
        complete <-> (cDes1.status=complete) and (cDes1.status=complete);
        incomplete <-> (cDes1.status=incomplete) or (cDes1.status=incomplete);
  end process_body
```

MVP-L: recursos de proceso

```
process_resources
```

```
  personnel_assignments
```

```
    imports
```

```
      resource_model OO_Designer;
```

```
    objects
```

```
      d: OO_Designer;
```

```
  tool_assignments
```

```
    imports
```

```
      resource_model UML_Editor;
```

```
    objects
```

```
      rational: UML_Editor;
```

```
end process_resources
```

EIS: Conceptos básicos

- Entorno de Ingeniería del Software (EIS).
 - “*Colección de herramientas que proporcionan un soporte automático, parcial o total, a las actividades de ingeniería del software*”.
 - Un EIS da soporte a actividades humanas mediante una serie de **servicios** que describen las capacidades del entorno.
 - Mediante la automatización de actividades, de forma parcial o total, un EIS aporta beneficios a una organización:
 - reducción de costes (mayor productividad),
 - mejora en la gestión, y
 - mayor calidad en el producto final.

Modelo de Servicios: ISO 15940

- ISO 15940: *Information Technology - Software Engineering Environment Services*.
 - Provee un modelo de referencia: descripción de todos los servicios que soportan a los procesos del ciclo de vida del software (según ISO 12207).
 - Cada descripción de un servicio incluye: concepto, operaciones básicas, y automatización.
 - En enseñanza y entrenamiento de Ingeniería del Software:
 - Utiliza una base comúnmente acordada, de conceptos y definiciones, para la presentación de EIS.
 - Permite enseñar Ingeniería del Software basándose en un completo abanico de servicios.

Servicios EIS: categorías

- Los servicios de un EIS se clasifican en categorías que reflejan la amplitud de las actividades de Ingeniería del Software:
 - Ingeniería Técnica (*compilación*)
 - Gestión Técnica (*gestión de configuraciones*)
 - Gestión del Proyecto (*análisis de riesgos*)
 - Gestión del Proceso (*mejora de procesos*)
 - Soporte (*publicación*)
 - Globales (*gestión de objetos*)

Servicios EIS: Ingeniería Técnica

- Soportan actividades relacionadas con la especificación, diseño, implementación, prueba y mantenimiento de software:

Ing. de requisitos software

Diseño software

Simulación y modelado software

Verificación de software

Generación de software basado en componentes

Generación de código fuente

Compilación

Análisis estático de software

Depuración

Prueba de software

Integración de componentes

Ingeniería inversa de software

Reingeniería de software

Trazabilidad de software

Pruebas de cualificación de software

Prototipado software

Documentación de usuario

Servicios EIS: Gestión Técnica

- Soporte a actividades mixtas comunes a ingenieros y gestores:
 - Gestión de configuraciones.
 - Gestión de cambios.
 - Gestión del repositorio EIS.
 - Reutilización.
 - Colección y análisis de métricas.
 - Aseguramiento de calidad.
 - Auditoría.

Servicios EIS: Gestión del Proyecto

- Soporte a actividades relacionadas con la planificación y ejecución de un proyecto software:
 - Planificación.
 - Estimación.
 - Análisis de riesgos.
 - Seguimiento.
 - Evaluación.

Servicios EIS: Gestión del Proceso

- Ayudan a los proyectos a alcanzar disciplina, control y comprensión clara de sus procesos y actividades:
 - Definición de procesos.
 - Biblioteca de procesos.
 - Iniciación de procesos.
 - Utilización de procesos en proyectos.
 - Supervisión de procesos.
 - Mejora de procesos.
 - Documentación de procesos.

Servicios EIS: de Soporte

- Usados por todos los usuarios. Asociados con procesar y distribuir datos en formato manejable por personas.
 - Soporte global.
 - Publicación.
 - Soporte al trabajo en grupo.
 - Soporte a la comunicación de usuarios.
 - Administración del EIS.
 - Cumplimiento de políticas.

Servicios EIS: Globales

- Ayudan a que la infraestructura del EIS de soporte a las aplicaciones y herramientas.
 - Gestión de la infraestructura del EIS.
 - Comunicación inter-proceso.
 - Gestión de objetos.

Servicios de Gestión del Proceso (i)

- Definición de procesos:
 - Provee para el establecimiento de los procesos organizacionales, cubriendo el ciclo de vida del software a través de la adaptación y particularización un conjunto de clases de procesos de referencia de alto nivel.
 - Operaciones básicas:
 - Analizar los requisitos de proceso, incluyendo los específicos del dominio y los específicos de la aplicación.
 - Instanciar, componer, descomponer, particularizar y modularizar definiciones de proceso.
 - Simular, modelar y validar definiciones de proceso.
 - Automatización:
 - Todo lo anterior.

Servicios de Gestión del Proceso (ii)

- Biblioteca de procesos:
 - Soporta la reutilización de capacidades de procesos en base a activos de proceso (*assets*). Un activo puede oscilar desde la definición de una actividad simple hasta un ciclo de vida completo. Activos pueden ser objetos versionados.
 - Operaciones básicas:
 - Crear, modificar y eliminar activos de proceso.
 - Certificar, medir y administrar activos de proceso.
 - Automatización:
 - Almacenamiento y versionado de activos de proceso.
 - Procesamiento de informes de estado.

Servicios de Gestión del Proceso (iii)

- **Iniciación de procesos:**
 - Soporta la asignación de un modelo de ciclo de vida (metamodelo), un conjunto de procesos, y el EIS para satisfacer los requisitos y restricciones de un proyecto particular.
 - Operaciones básicas:
 - Revisar criterios y restricciones de un proyecto y seleccionar modelo de ciclo de vida.
 - Definir interrelaciones y particularizar procesos y actividades.
 - Automatización:
 - Definición de interrelaciones y particularización de procesos y actividades.

Servicios de Gestión del Proceso (iv)

- Utilización de procesos en proyectos:
 - Capacidades para ayudar a utilizar procesos dentro de un proyecto (p.e., asignación de usuarios, facilidades navegacionales, etc.).
 - Operaciones básicas:
 - Ayudar sobre el proceso y facilitar orientación para miembros del equipo del proyecto.
 - Consultar e informar sobre utilización y estado de procesos.
 - Especificar, recolectar y reportar sobre métricas de procesos.
 - Simular interactivamente definiciones de proceso, y gestionar representaciones de alto nivel.
 - Automatización:
 - Consulta y reporte de utilización y estado de procesos.

Servicios de Gestión del Proceso (v)

- Supervisión de procesos:
 - Soporta la observación, detección, registro y traza de actividades de procesos (dentro de proyectos).
 - Operaciones básicas:
 - Establecer condiciones y criterios de supervisión.
 - Observar la evolución en el estado de la reificación de procesos.
 - Detectar la ocurrencia de eventos de proceso específicos.
 - Registrar la ocurrencia de eventos de proceso específicos.
 - Automatización:
 - Detección y registro de la supervisión.
 - Presentación de datos de supervisión, incluidos gráficos.
 - Distribución de datos de supervisión.

Servicios de Gestión del Proceso (vi)

- Mejora de procesos:
 - Soporta la evaluación, medición y modificación de los procesos organizacionales y de proyectos específicos, y de los ciclos de vida de proyectos.
 - Operaciones básicas:
 - Definir objetivos de eficiencia.
 - Identificar mediciones, relacionados con los objetivos.
 - Establecer valores límites para la consecución de objetivos.
 - Evaluar la capacidad de los procesos.
 - Preparar informes de evaluación que comparan los datos actuales con los buscados.
 - Planificar las evaluaciones.
 - Automatización:
 - Recolección de datos de medidas.
 - Preparación de informes de evaluación.

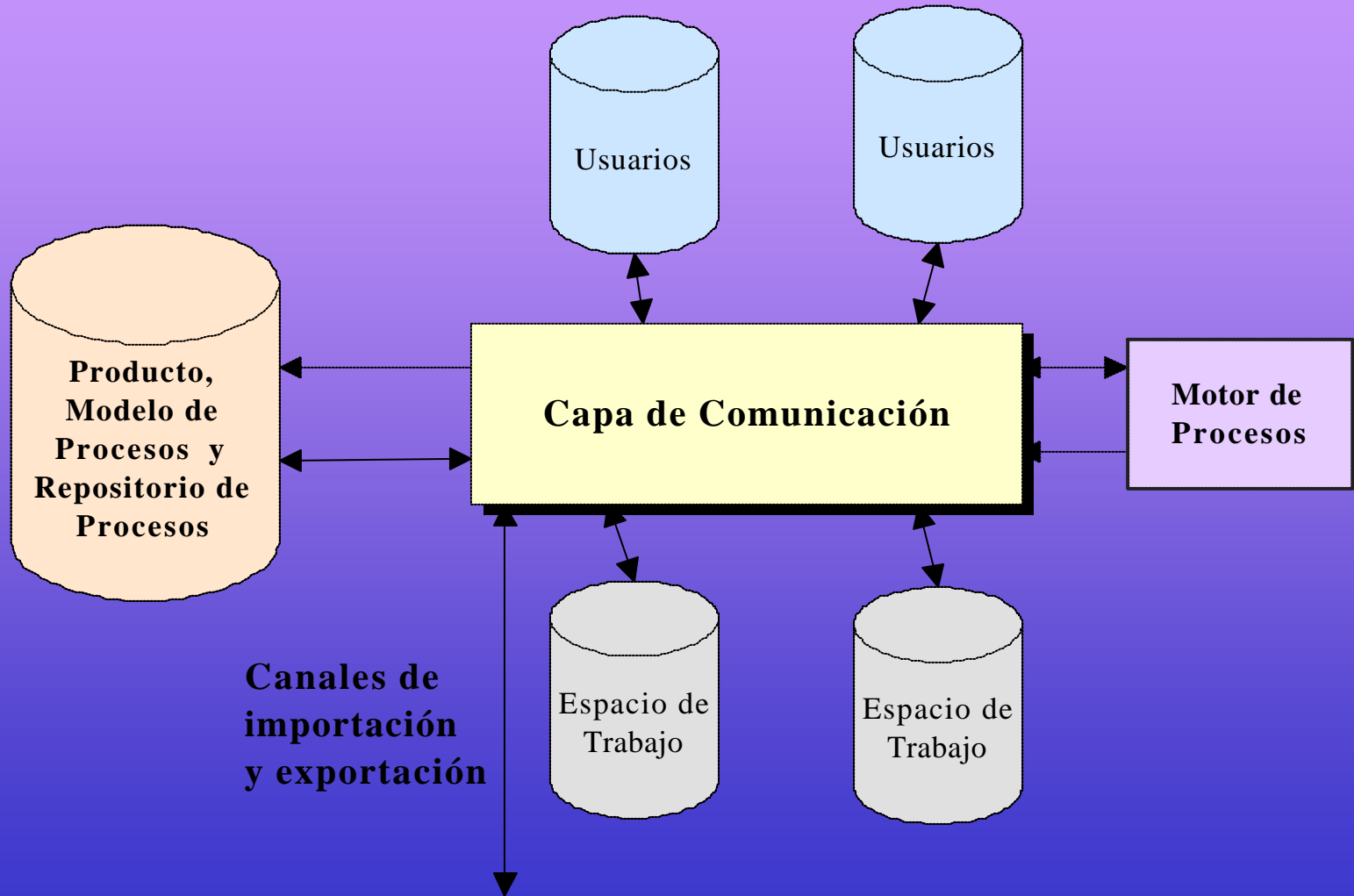
Servicios de Gestión del Proceso (vii)

- Documentación de procesos:
 - Da soporte para la documentación del proceso a todos los demás servicios.
 - Operaciones básicas:
 - Identificar los requisitos de documentación.
 - Diseñar y desarrollar los documentos.
 - Producir y editar documentos.
 - Distribuir los documentos.
 - Mantener dichos documentos.
 - Automatización:
 - Diseño, producción y edición de la documentación.
 - Distribución y mantenimiento de la documentación.

Arquitectura PSEE (i)

- *Personal-sensitive Software Engineering Environment.*
 - EIS centrados en procesos.
 - EIS basados en la Tecnología de Proceso Software:
 - Soporte computerizado al proceso, es decir,
 - Disponibilidad de un MP, y
 - Medios adecuados para definirlo, modificarlo, analizarlo y reificarlo.

Arquitectura PSEE (ii)



Arquitectura PSEE (iii)

- **Servicios básicos** requeridos:

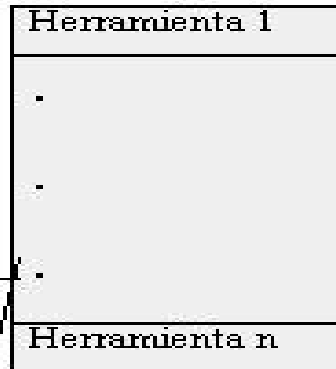
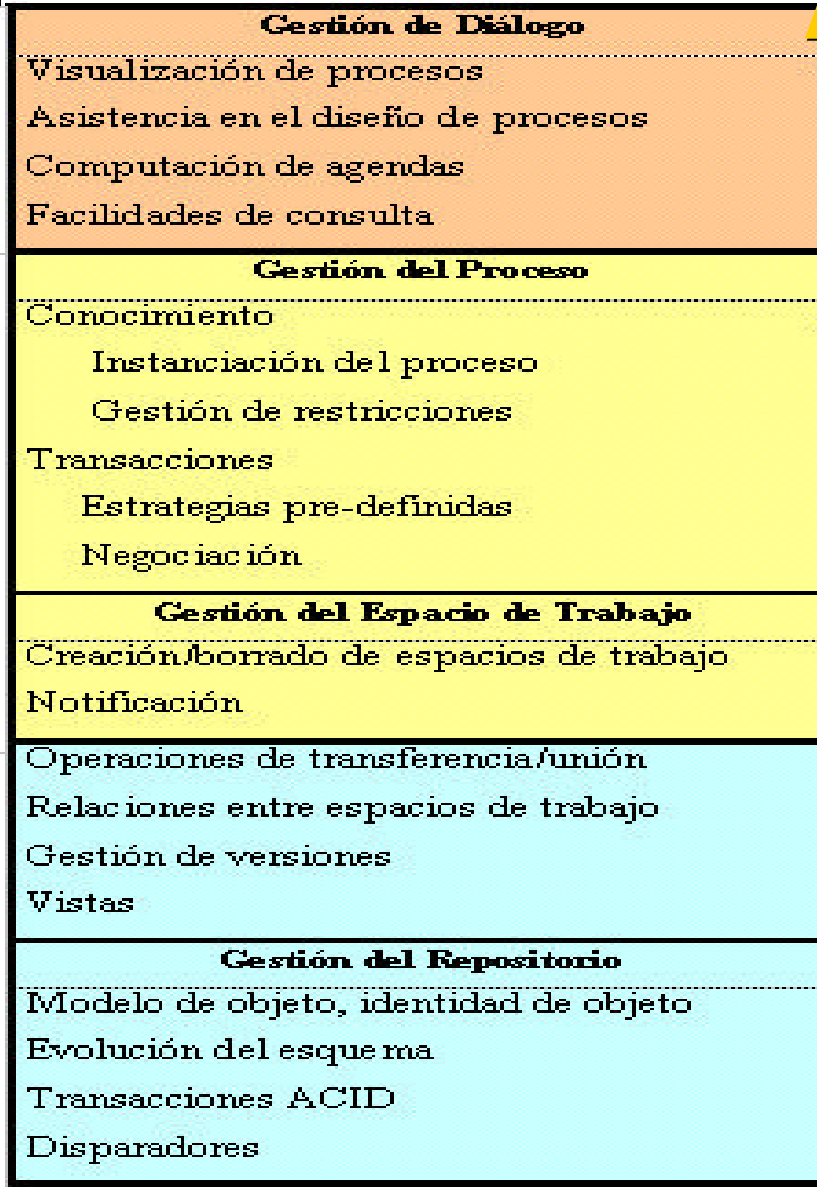
- Una gestión del diálogo para dar a los usuarios información sobre los procesos y permitirles llevar a cabo actividades.
- Una gestión del proceso cuya tarea es ejecutar un MP particular y coordinar las actividades concurrentes de múltiples usuarios.
- Una gestión del espacio de trabajo personal para cada usuario en cada uno de sus roles. Incluye todos los objetos software a los que tiene que acceder cada usuario con cada rol.
- Un gestor del repositorio PSEE para almacenar de forma persistente los objetos software y sus correspondientes relaciones, y poder acceder a ellos eficientemente.

Arquitectura PSEE (iv)

Interface

Motor de procesos

Repositorios



Servidor

Herramientas de Desarrollo Software

PSEE: gestión del diálogo

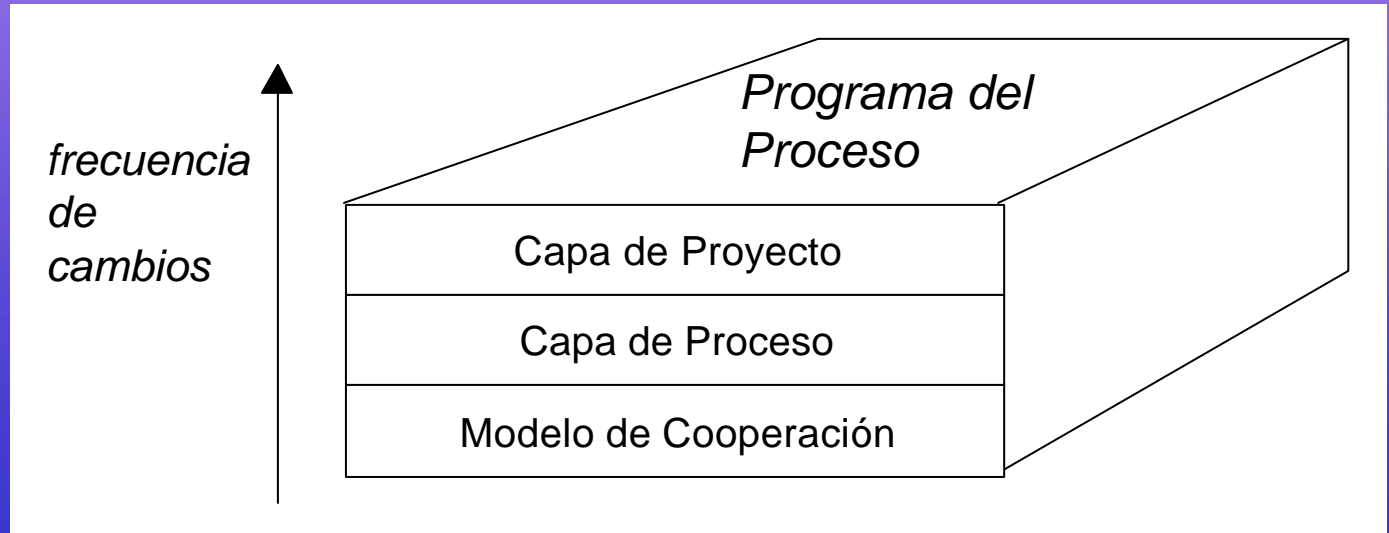
- Encapsula la **interfaz de usuario** de un PSEE.
- Incluye:
 - Visualización de procesos.
 - Asistencia en el diseño de procesos.
 - Computación de agendas.
 - Facilidades de consulta.
- Habitualmente los usuarios de un PSEE pueden interactuar con diferentes roles:
 - desarrolladores de software,
 - gestores de proyecto, e
 - ingenieros de procesos.
- Hay un patrón de interacción diferente para cada rol.

PSEE: gestión del proceso (i)

- Coordina las diferentes actividades de los múltiples actores involucrados en un proyecto software.
- Computa el espacio de trabajo específico de cada usuario involucrado en un proyecto de desarrollo software, reflejando el estado actual del proyecto.
- Incluye:
 - Conocimiento del proceso:
 - LMP,
 - Instanciación del proceso
 - Gestión de restricciones
 - Transacciones:
 - Estrategias predefinidas
 - Negociación

PSEE: gestión del proceso (ii)

- Puede haber un motor de procesos central o varios distribuidos, ejecutando una descripción formal del PS, llamada *Programa del Proceso Software* (PPS).
 - El PPS debe poder ser modificado durante la ejecución del proceso.
 - En función de su estabilidad, un PPS se puede dividir en 3 capas:



PSEE: gestión del espacio de trabajo

- Las dos motivaciones básicas que subyacen en la capa de gestión del espacio de trabajo son la **abstracción** y el **aislamiento**.
- Los espacios de trabajo permiten a los usuarios concentrarse en sus tareas específicas abstrayendo información irrelevante de otras partes del proyecto.
- Incluye:
 - Creación/borrado de espacios de trabajo
 - Notificación
 - Operaciones de transferencia/unión
 - Relaciones entre espacios de trabajo
 - Gestión de versiones
 - Vistas

PSEE: gestión del repositorio (i)

- Este servicio es responsable de mantener la consistencia y disponibilidad de la información que necesitan los otros componentes del PSEE.
- A esta información deben poder acceder al mismo tiempo diferentes usuarios del PSEE (conurrencia).
- Incluye:
 - Modelo de objetos
 - Identidad de objetos
 - Evolución del esquema
 - Transacciones ACID
 - Disparadores

PSEE: gestión del repositorio (ii)

- Un proyecto de desarrollo software habitualmente genera muchas formas diferentes de datos:
 - datos de **productos**: código fuente, datos de gestión de la configuración, documentación, ejecutables, juegos de pruebas, resultados de pruebas, simulaciones
 - datos del **proceso**: definición explícita de un MP, información del estado de la reificación de un proceso, datos para análisis y evolución del proceso, datos históricos, datos de gestión del proyecto,
 - datos **organizacionales**: información sobre propietarios de componentes del proyecto, roles y responsabilidades, datos de gestión de los recursos,

PSEE: gestión del repositorio (iii)

- **Características deseables:**
 - *soporte multiusuario,*
 - *eficiencia,*
 - *persistencia e integridad,*
 - *distribución,*
 - *heterogeneidad,*
 - *Evolución (de los datos y de los metadatos),*
 - *versionado y gestión de configuración,*
 - *gestión de transacciones flexible (ACID),*
 - *facilidades de consulta “ad-hoc”.*

PSEE: gestión de la comunicación

- Los usuarios prefieren los PSEE con una arquitectura abierta que soporte diferentes niveles de integración. Esto es debido a que:
 - el soporte a proyectos tiene que manejar distintos tipos de proyectos
 - debe ser adaptable a las necesidades cambiantes de un proyecto
 - debe ser adaptable a los nuevos avances tecnológicos
 - debe ser posible añadir nuevas herramientas
- Una buena técnica para integrar las diferentes partes es usar un entorno de comunicación.
- Incluye:
 - Adaptadores de protocolos
 - Notificaciones
 - Peticiones Síncronas/asíncronas

Lecturas

- **Bröckers**, A., Lott, C.M., Rombach, H.D. y Verlage, M. (1995): MVP-L Language Report Version 2. Universidad de Kaiserslautern (Alemania). Capítulos 1-4.
- **Derniame**, J.C., Kaba, B.A., & Wastell, D. (1999): “Software Process: Principles, Methodology and Technology”. LNCS 1500, Springer-Verlag. Capítulos 1, 3, 4 (4.1-4.4) y 5 (5.1).
- **Fuggetta**, A. (2000): Software Process: A Roadmap. *International Conference on Software Engineering*.
- **ISO 15940** (2000): ISO Information technology — Software Engineering Environment Services. Working draft v 5.
- **Sharp**, A., McDermott, P. (2001): “ Workflow Modeling. Tools for Process Improvement and Application Development”. Artech-House. Capítulo 4.
- **SWEBOK**. Software Engineering Coordinating Committee (IEEE-CS y ACM). Guide to the Software Engineering Body of Knowledge (trial versión 0.95). Disponible en <http://www.swebok.org>. Capítulos 1, 8, 9 y 10.