

05BM - Fundamentos de Ingeniería del Software.

**Bloque 3.- Modelos del Proceso Sw.
Tema 6. El proceso Software. Paradigmas de Ciclo de Vida.**

Departamento de Informática y Sistemas

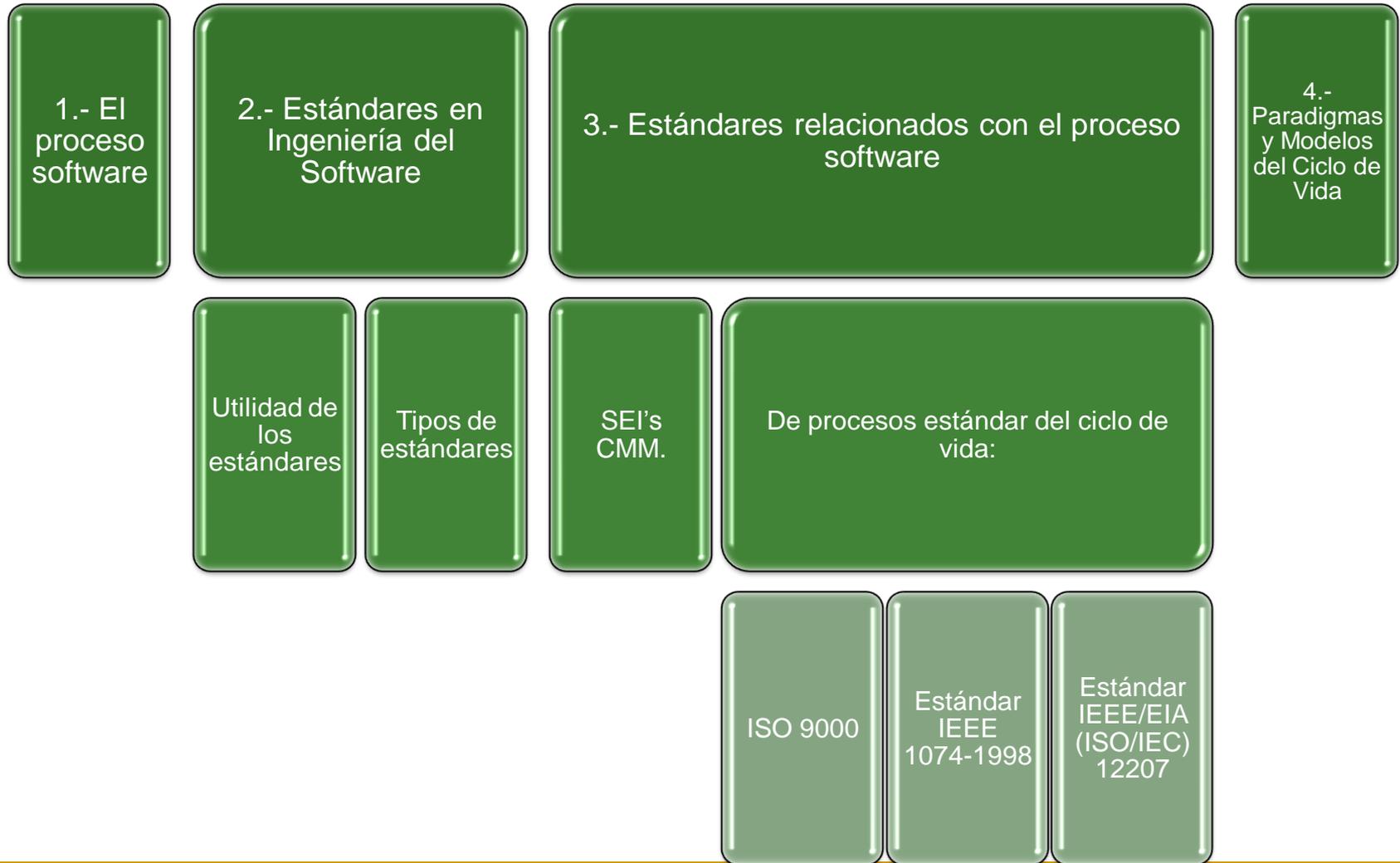
Facultad de Informática

Campus Universitario de Espinardo – Murcia

Profesor: Juan Antonio López Quesada

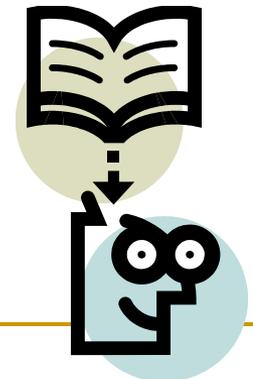


Índice de contenidos



Bibliografía

- (Hilera et al. 97) José R. Hilera, José A. Gutiérrez, J. Javier Martínez. “Estándares en la Ingeniería del Software”. Novática. Nov./dic. 1997. Número 130.
- (Piattini et al. 96) Cap. 3. Aptdos. 3.1 y 3.2.
- (Esteban Piattini 95) J. L. Esteban, M. Piattini. “Procesos del ciclo de vida del software”. Novática, Nov./dic. 1995.
- International Standard ISO/IEC 12207. “Information technology-Software life cycle processes”. 1995.
- IEEE Standards Collection. Software Engineering. 1998 Edition. Published by the Institute of Electrical and Electronics Engineers, Inc.
- (Pressman 2002) pp.16-18 y pp.146-147.
- (Sommerville 2002) pp. 8-9.



1.- El Proceso Software.

- Es un conjunto de actividades y resultados asociados que producen un *producto de software*.
- Es uno de los componentes de un *método de desarrollo de software*.
- Existen 4 actividades fundamentales de proceso, comunes para todos los procesos de software:



2.- Estándares en la Ingeniería del Sw.

- **Estándar:** Conjunto de criterios aprobados, documentados y disponibles para determinar la adecuación de una acción (*estándar de proceso*) o de un objeto (*estándar de producto*).
- **Guía:** Conjunto de criterios bien definidos y documentados que encaminan una actividad o tarea.
⇒ *es más flexible que un estándar*

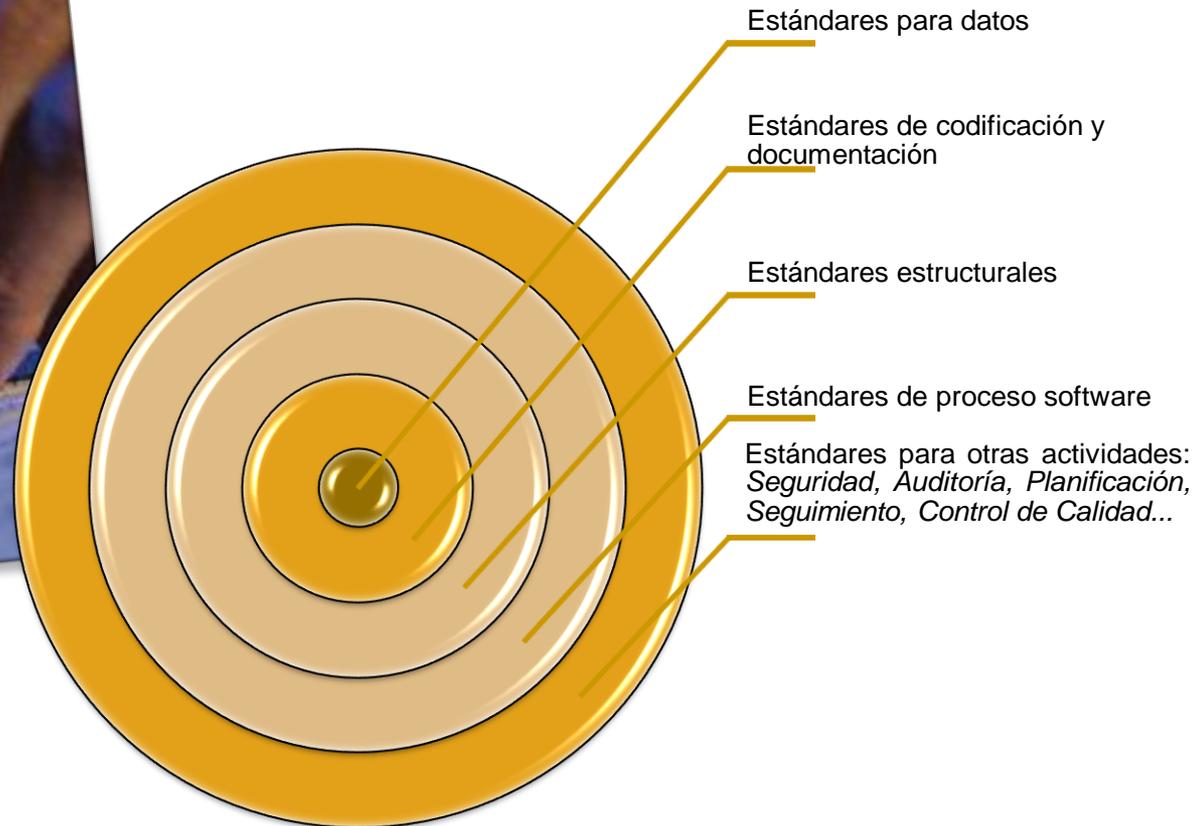
2.- Estándares en la Ingeniería del Sw.

2.1.- Utilidad de los estándares.

- Según Sommerville, los estándares son útiles porque:
 - Agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software.
 - Engloban los “conocimientos” que son patrimonio de una organización.
 - Proporcionan un marco para implementar procedimientos de aseguramiento de la calidad.
 - Proporcionan continuidad entre el trabajo de distintas personas.

2.- Estándares en la Ingeniería del Sw.

2.2.- Tipos de estándares.



3.- Estándares relacionados con el Proceso Software: *SEI's CMM (Software Engineering Institute - Modelo de Capacidad y Madurez)*

- ❑ Proporciona una medida de la eficacia global de las prácticas de ingeniería del sw de una compañía y establece para ello, cinco niveles de madurez del proceso.
- ❑ Los cinco niveles se obtienen como consecuencia de evaluar las respuesta del cuestionario de evaluación basado en el CMM (Capability Maturity Model). Los resultados se filtran en un único grado numérico que proporciona una indicación de la madurez del proceso en la organización.



3.- Estándares relacionados con el proceso software.

SEI's CMM (Capability Maturity Model)



**Carnegie Mellon
Software Engineering Institute**

- A partir de noviembre de 1986 el SEI, a requerimiento del Gobierno Federal de los Estados Unidos de América, desarrolló una primera definición de un modelo de madurez de procesos en el desarrollo de software, que se publicó en septiembre de 1987. Este trabajo evolucionó al modelo CMM o SW-CMM (CMM for Software), cuya última versión (v1.1) se publicó en febrero de 1993.
- Este modelo establece un conjunto de prácticas o procesos clave agrupadas en Areas Clave de Proceso (KPA - Key Process Area). Para cada área de proceso define un conjunto de buenas prácticas que habrán de ser:
 - Definidas en un procedimiento documentado.
 - Provistas (la organización) de los medios y formación necesarios.
 - Ejecutadas de un modo sistemático, universal y uniforme(institucionalizadas).
 - Medidas.
 - Verificadas.
- A su vez estas áreas de proceso se agrupan en cinco "niveles de madurez", de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez.

3.- Estándares relacionados con el proceso software.

SEI's CMM (Capability Maturity Model)



Carnegie Mellon
Software Engineering Institute

■ Los niveles son:

- ❑ **1.- Inicial.** Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobre costes. El resultado de los proyectos es impredecible.
- ❑ **2.- Repetible.** En este nivel las organizaciones disponen de unas prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente.
- ❑ **3.- Definido.** Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión por pares (peer reviews).
- ❑ **4.- Gestionado.** Se caracteriza por que las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.
- ❑ **5.- Optimizado.** La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

3.- Estándares relacionados con el proceso software.

SEI's CMM (Capability Maturity Model)

Bibliografía

- ❑ **Gonzalo Cuevas Agustín:** *Una Guía del CMM. Para Comprender el Modelo de Madurez de Capacidad del Software.* Traducción del Inglés "A Guide to the CMM" de Kenneth M. Dymond. 1998.
- ❑ **Mary Beth Chrissis:** *Libro con la descripción de las Areas de Procesos del Modelo CMMI.* "CMMI : Guidelines for Process Integration and Product Improvement de SEI.

Enlaces externos

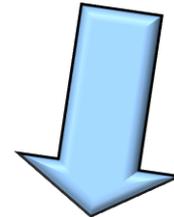
- ❑ [SEI - Software Engineering Institute](http://www.sei.cmu.edu/) - <http://www.sei.cmu.edu/>
- ❑ [ESI - European Software Institute](http://www.esi.es/) - <http://www.esi.es/>
- ❑ [System Security Engineering CMM](http://www.sse-cmm.org/) - <http://www.sse-cmm.org/>
- ❑ [Finkelstein's Capability Immaturity Model paper \(PDF file\)](http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/immaturity.pdf) - <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/immaturity.pdf>
- ❑ [Capt. Tom Schorsch's Capability Immaturity Model study](http://www.stsc.hill.af.mil/crosstalk/1996/11/xt96d11h.asp) - <http://www.stsc.hill.af.mil/crosstalk/1996/11/xt96d11h.asp>
- ❑ [Introducción a los niveles CMM CMMI](http://www.ingenierosoftware.com/calidad/cmm-cmmi.php) - <http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>
- ❑ http://www.cii-murcia.es/informas/ene05/articulos/Modelo_de_Madurez_de_la_Capacidad_del_Software.html

3.- Estándares relacionados con el proceso software.

3.2.- *Proceso Estándar del Ciclo de Vida.*

Multitud de estándares, métodos, técnicas, y entornos para desarrollar y gestionar software.

Software usado en multitud de sistemas diferentes: militar, finanzas, medicina, etc.



Dificultades para gestionar la producción de software, integrando productos y servicios

3.- Estándares relacionados con el proceso software.

3.2.- *Proceso Estándar del Ciclo de Vida.*

- Necesario conseguir un **marco común** para “hablar el mismo lenguaje” en el desarrollo y gestión de software
- **Objetivo:** Definir los procesos de desarrollo y mantenimiento del software, y de gestión del mismo, de forma genérica y abstracta
- Marco común \Rightarrow ***Estándares del ciclo de vida***

3.- Estándares relacionados con el proceso software.

3.2.- *Proceso Estándar del Ciclo de Vida.*



- Familia ISO 9000 ⇒ CALIDAD
 - IEEE 1074-1998 - *IEEE Standard for Developing Software Life Cycle Processes*
 - ISO/IEC 12207:1995 (E) *Information technology – Software life cycle processes* (posteriormente adoptado por IEEE/EIA)
-
- IEEE – *Institute of Electrical and Electronics Engineers.*
 - ISO – *International Organization for Standardization.*
 - IEC – *International Electrotechnical Commission.*

3.- Estándares relacionados con el proceso software.



3.2.- Proceso Estándar del Ciclo de Vida.

3.2.1.- ISO 9000.

- Familia de estándares para la gestión de la calidad de cualquier proceso de producción.
- La organización debe tener un sistema de calidad que supervise todas las fases de la producción y entrega del producto:
 - Audita los proyectos para asegurar que se cumplen los controles de calidad.
 - Mejora la calidad del propio sistema de calidad.
 - Proporciona entradas al grupo de desarrollo (como nuevas notaciones, procedimientos, estándares).
 - Produce informes para la dirección.
- Para cada proyecto se define un plan de calidad.

3.- Estándares relacionados con el proceso software.



3.2.- Proceso Estándar del Ciclo de Vida.

3.2.1.- ISO 9000.

- *ISO 9001. Quality Systems - Model for Quality Assurance in Design, Development, Production, Installation and Servicing.*
- *ISO 9000-3. Guidelines for Application of ISO 9001 to the Development, Supply and Maintenance of Software*
 - *Contiene directrices que implementa ISO 9001 para el desarrollador de software*
- *ISO 9004-2. Quality Management and Quality Systems Elements - Part 2.*
 - *Contiene guías para proporcionar servicios de software, como por ejemplo el soporte de usuario.*

Estándar de calidad: ISO 9000 para la producción de sw (Pressman 2002) p.146

3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.1.- ISO 9000



Servicio/Producto

■ 1. ESTRUCTURA PRINCIPAL

- Introducción
- Planificación de Sistemas de Información (Proceso PSI)
- Estudio de Viabilidad del Sistema (Proceso EVS)
- Análisis del Sistema de Información (Proceso ASI)
- Diseño del Sistema de Información (Proceso DSI)
- Construcción del Sistema de Información (Proceso CSI)
- Implantación y Aceptación del Sistema (Proceso IAS)
- Mantenimiento del Sistema de Información (Proceso MSI)

■ 2. INTERFACES

- Aseguramiento de la Calidad
- Seguridad
- Gestión de Configuración
- Gestión de Proyectos



3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.1.- ISO 9000



Servicio/Producto

- El objetivo de la interfaz de Aseguramiento de la Calidad de MÉTRICA Versión 3 es proporcionar un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos. Si en la organización ya existe un sistema de calidad, dichos planes deberán ser coherentes con el mismo, completándolo en los aspectos no contemplados relativos a normas particulares del cliente, usuario o sistema concreto.
- La calidad se define como “**grado en que un conjunto de características inherentes cumple con unos requisitos**” [ISO 9000:2000]. El Aseguramiento de la
- Calidad pretende dar confianza en que el producto reúne las características necesarias para satisfacer todos los requisitos del Sistema de Información.

Estándar de calidad: Interface de Calidad pag 1 , <http://www.csi.map.es/csi/metrica3/calidad.pdf>

3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.1.- ISO 9000



Servicio/Producto

- Una vez tomada la decisión de llevar a cabo un plan de aseguramiento de calidad en las alternativas propuestas, se define el contenido de dicho plan, de acuerdo a los estándares de calidad, si existen en la organización, sino se recomienda acudir a los estándares *UNE-EN-ISO 9001:2000 Sistemas de Gestión de la Calidad – Requisitos* y *UNE-EN-ISO 9000:2000 Sistemas de Gestión de la Calidad – Fundamentos y vocabulario*. El plan de aseguramiento de calidad debe cubrir todas las necesidades establecidas de modo que, aquellas normas impuestas por los usuarios o clientes que difieran de las existentes en el sistema de calidad, deben quedar también reflejadas en el plan.

Estándar de calidad: Interface de Calidad pag 2, <http://www.csi.map.es/csi/metrica3/calidad.pdf>

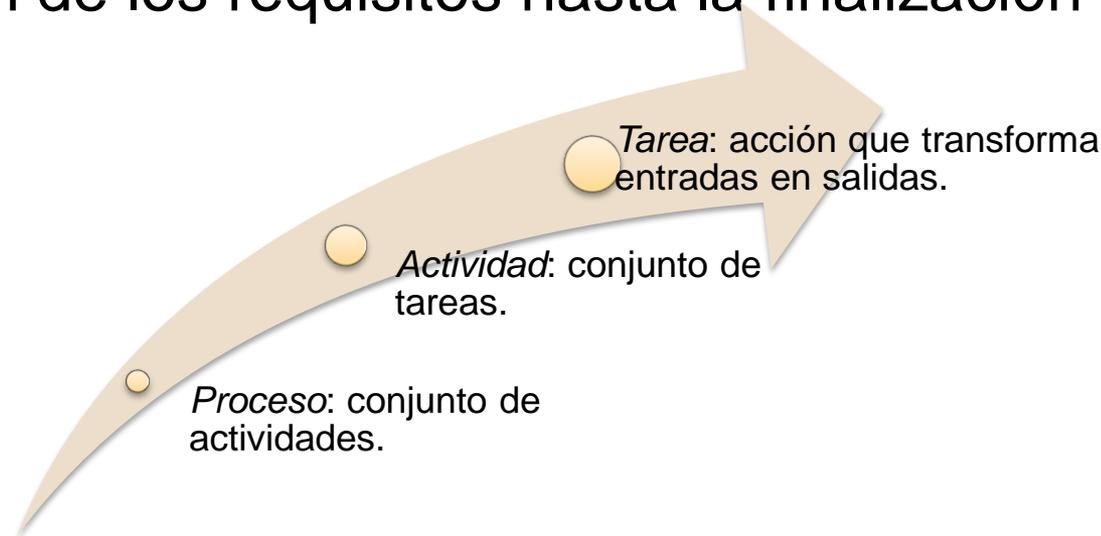
3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.2.- Estándar IEEE/EIA (ISO/IEC) 12207



- “Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”



IEEE/EIA (ISO/IEC) 12207. Information technology – Software life cycle processes.

3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.2.- Estándar IEEE/EIA (ISO/IEC) 12207



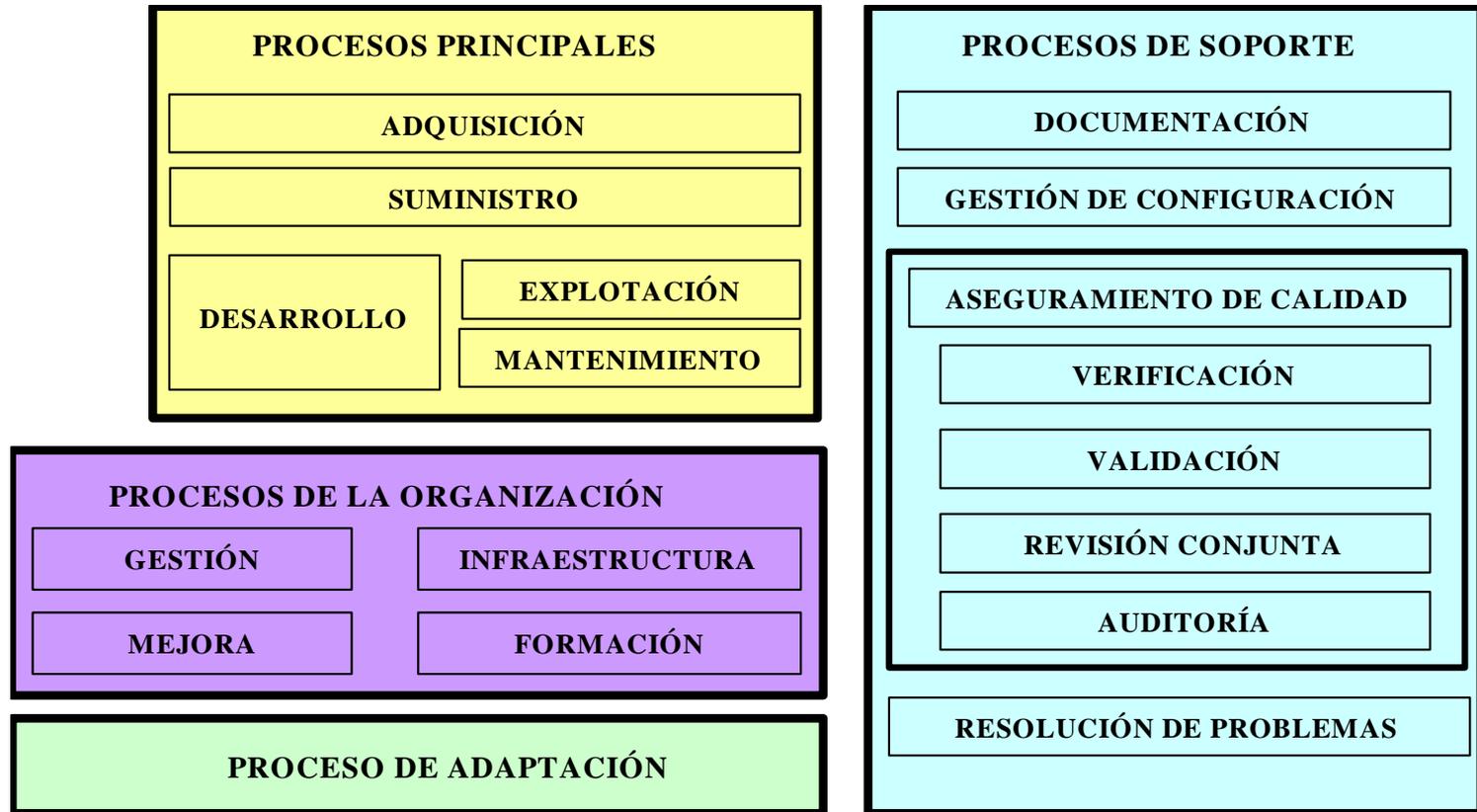
- Indica los procesos, actividades y tareas que se necesitan durante la adquisición de
 - un sistema que contiene software,
 - un producto software autónomo,
 - un servicio software,
- y durante el suministro, desarrollo, operación y mantenimiento de productos software.

IEEE/EIA (ISO/IEC) 12207. Information technology – Software life cycle processes.

3.- Estándares relacionados con el proceso software.

3.2.- Proceso Estándar del Ciclo de Vida.

3.2.2.- Estándar IEEE/EIA (ISO/IEC) 12207



IEEE/EIA (ISO/IEC) 12207. Information technology – Software life cycle processes.

4.- Paradigmas y Modelos del Ciclo de Vida

■ Ciclo de vida.

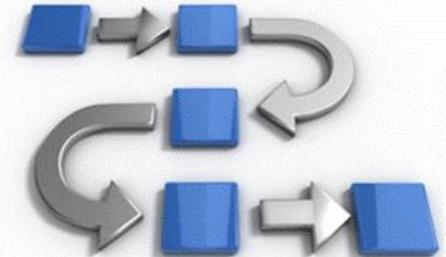
- Sucesión de etapas por las que atraviesa un producto software a lo largo de su existencia (i.e. durante su desarrollo y explotación).

“Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software”

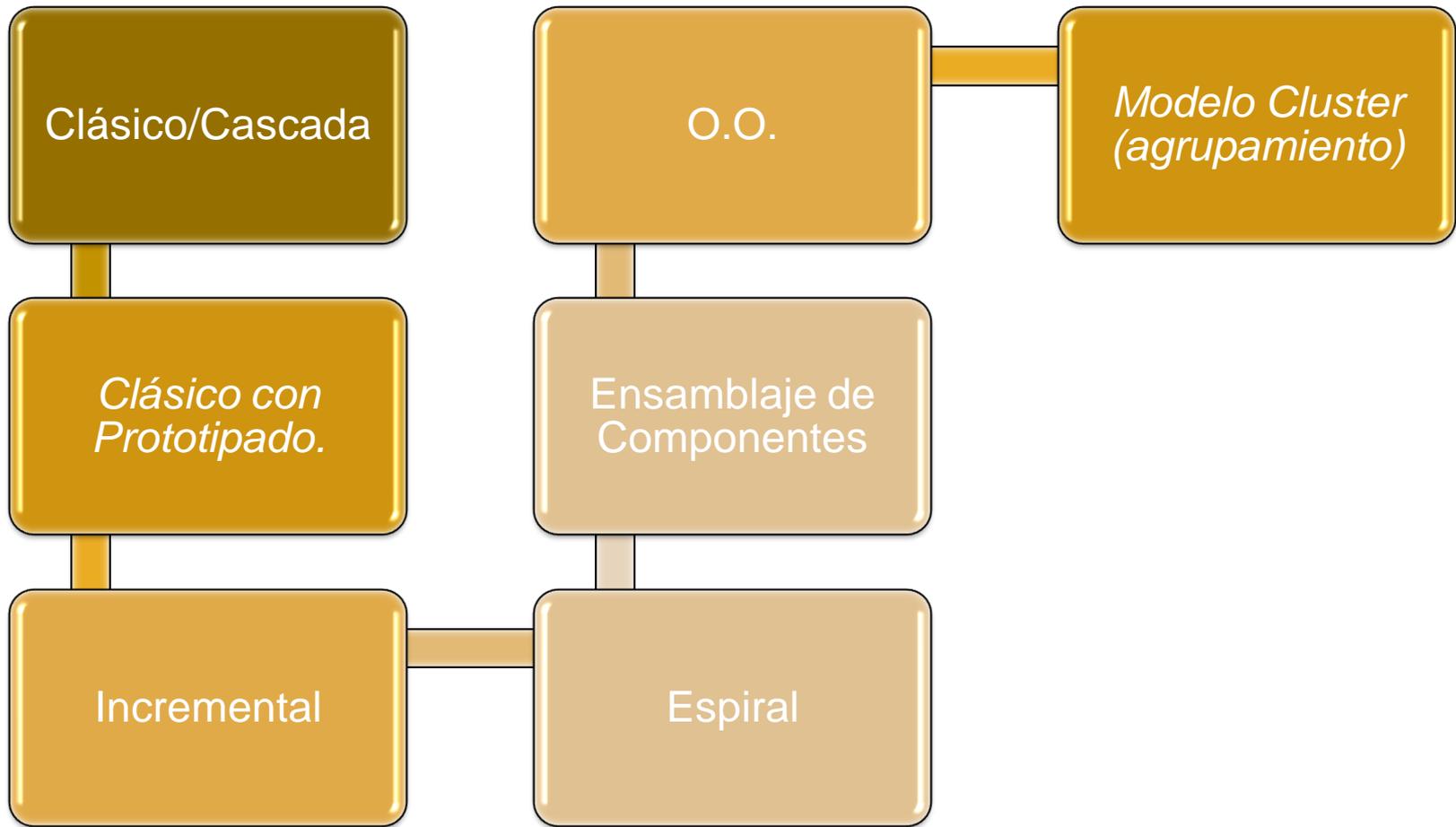
IEEE 1074

“Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”

ISO 12207-1



4.- Paradigmas y Modelos del Ciclo de Vida



4.- Paradigmas y Modelos del Ciclo de Vida

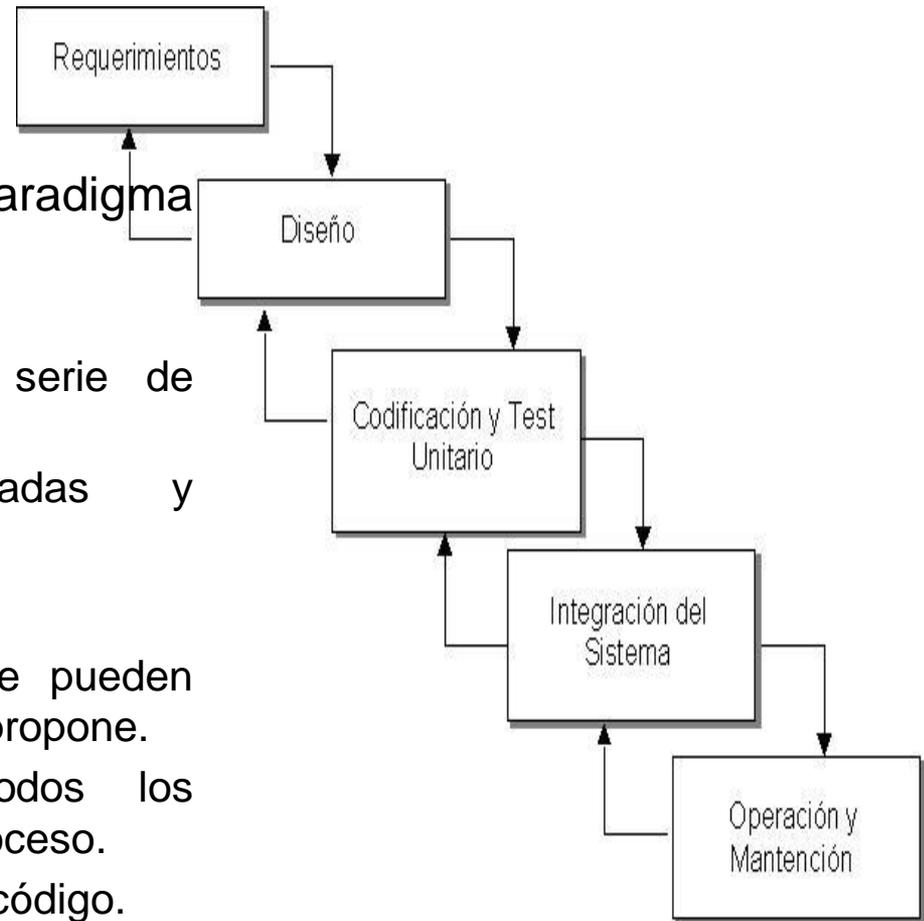
■ Paradigma clásico:

□ Paradigma “en cascada” o paradigma “orientado a fases”

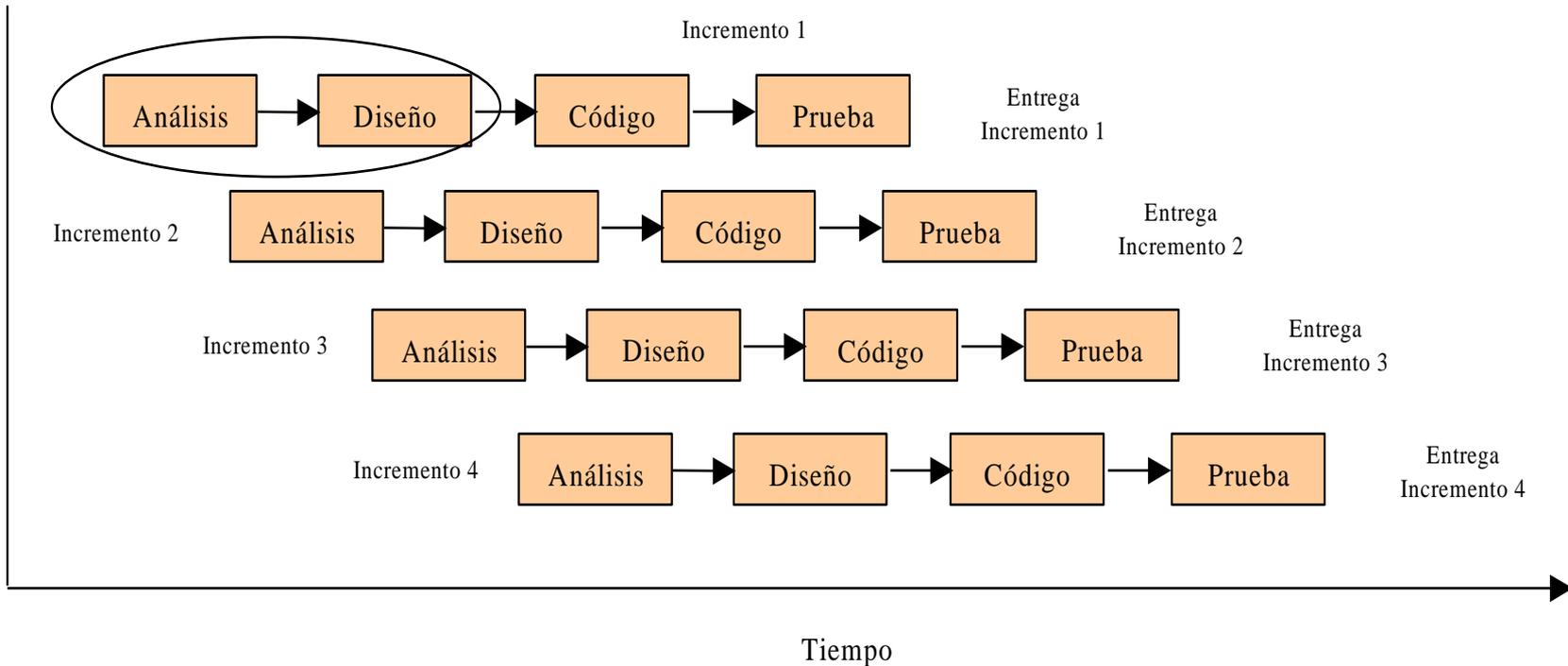
- Primer modelo empleado (1970).
- Ejecución secuencial de una serie de fases.
- Cada fase genera entradas y documentación para la siguiente.

■ Críticas al modelo de Cascada:

- Los proyectos reales raramente pueden seguir el flujo secuencial que se propone.
- Dificultad para establecer todos los requerimientos al principio del proceso.
- El mantenimiento recae sobre el código.
- Se tarda mucho tiempo en pasar por todo el ciclo (hasta que no termina una fase no empieza la siguiente).



4.- Paradigmas y Modelos del Ciclo de Vida

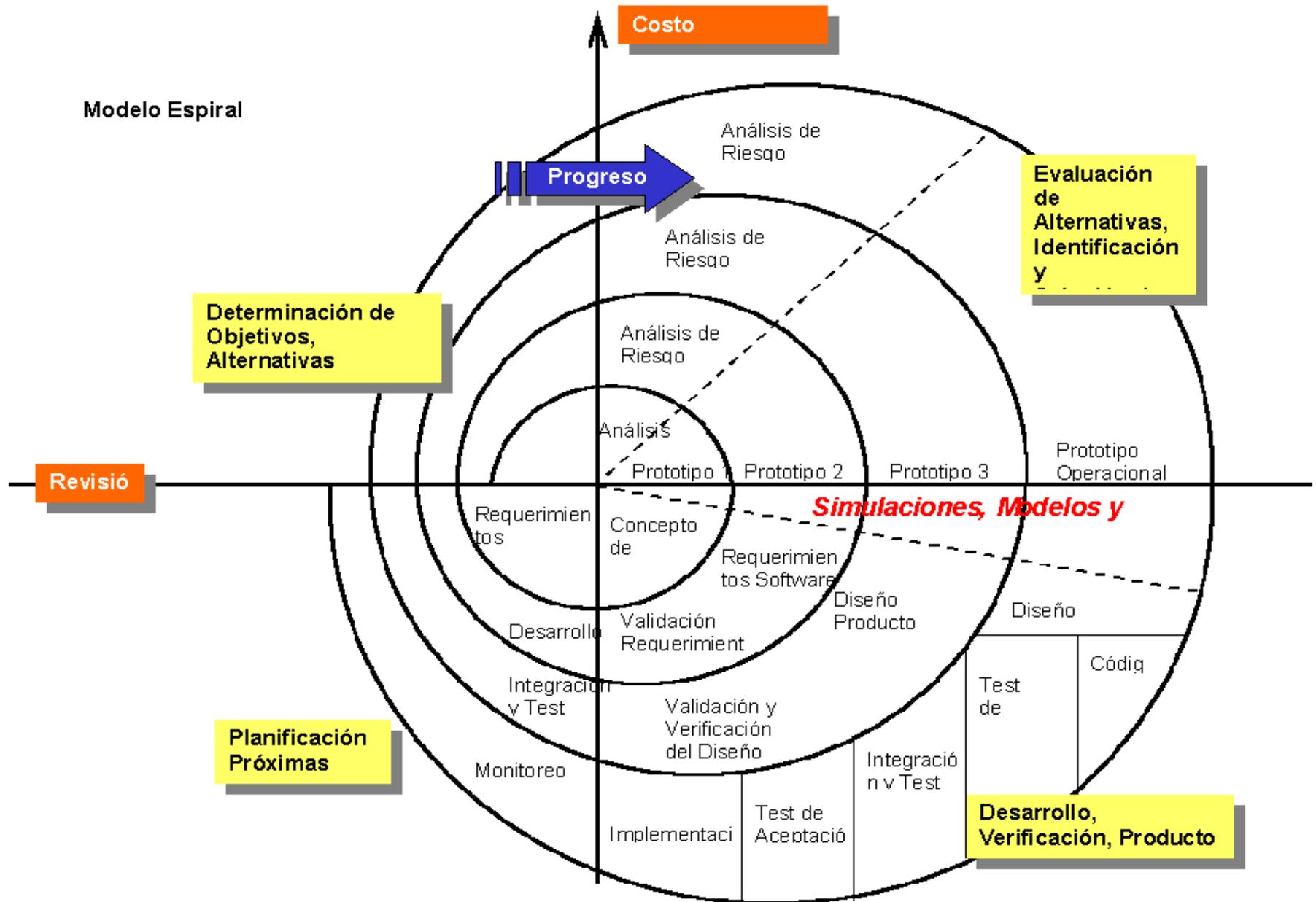


Modelo Incremental: El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.

4.- Paradigmas y Modelos del Ciclo de Vida

- ❑ Por ejemplo, el software de tratamiento de textos desarrollando con el paradigma incremental podría extraer funciones de gestión de archivos básicos y de producción de documentos en el primer incremento; funciones de edición más sofisticadas y de producción de documentos en el segundo incremento; corrección ortográfica y gramatical en tercero; y una función avanzada de esquema de pagina en el cuarto. Se deberían tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos.
- ❑ Cuando se utiliza un modelo incremental el primer incremento es un producto esencial (núcleo del sistema). Este proceso se repite hasta que se elabore el producto completo.
- ❑ El desarrollo incremental es particularmente útil cuando la dotación de personal no esta disponible para una implementación completa en la fecha limitada que se haya establecido para el proyecto.

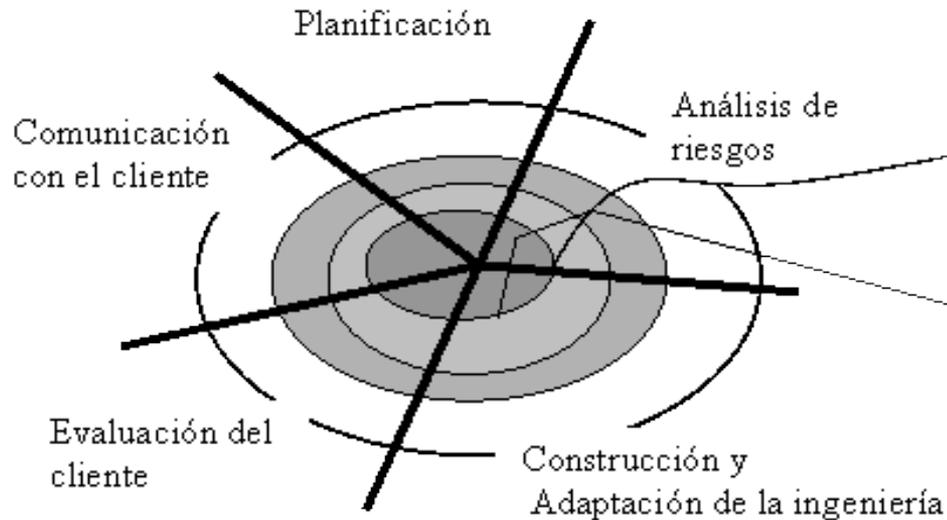
4.- Paradigmas y Modelos del Ciclo de Vida



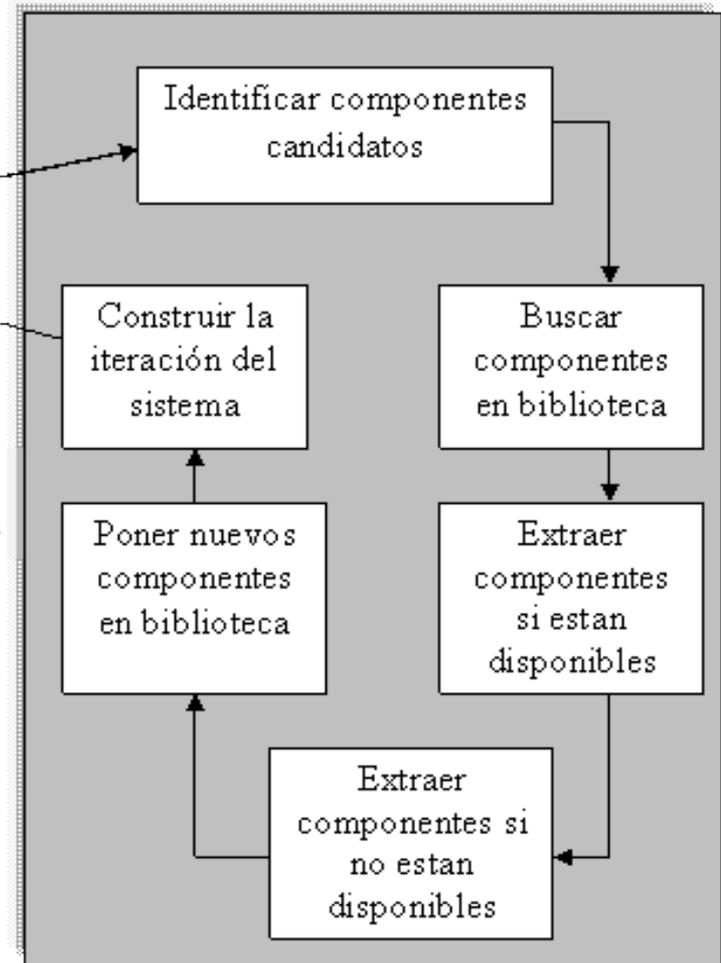
4.- Paradigmas y Modelos del Ciclo de Vida

- El modelo de desarrollo en espirales actualmente uno de los más conocidos y fue propuesto por Boehm. El ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra.
- Cada ciclo de desarrollo se divide en cuatro fases:
 - *Definición de objetivos: Se definen los objetivos. Se definen las restricciones del proceso y del producto. Se realiza un diseño detallado del plan administrativo. Se identifican los riesgos y se elaboran estrategias alternativas dependiendo de estos.*
 - *Evaluación y reducción de riesgos: Se realiza un análisis detallado de cada riesgo identificado. Pueden desarrollarse prototipos para disminuir el riesgo de requisitos dudosos. Se llevan a cabo los pasos para reducir los riesgos.*
 - *Desarrollo y validación: Se escoge el modelo de desarrollo después de la evaluación del riesgo. El modelo que se utilizará (cascada, sistemas formales, evolutivo, etc.) depende del riesgo identificado para esa fase.*
 - *Planificación: Se determina si continuar con otro ciclo. Se planea la siguiente fase del proyecto.*
- **Este modelo a diferencia de los otros toma en consideración explícitamente el riesgo, esta es una actividad importante en la administración del proyecto.**
- El ciclo de vida inicia con la definición de los objetivos. De acuerdo a las restricciones se determinan distintas alternativas. Se identifican los riesgos al sopesar los objetivos contra las alternativas. Se evalúan los riesgos con actividades como análisis detallado, simulación, prototipos, etc. Se desarrolla un poco el sistema. Se planifica la siguiente fase.

4.- Paradigmas y Modelos del Ciclo de Vida

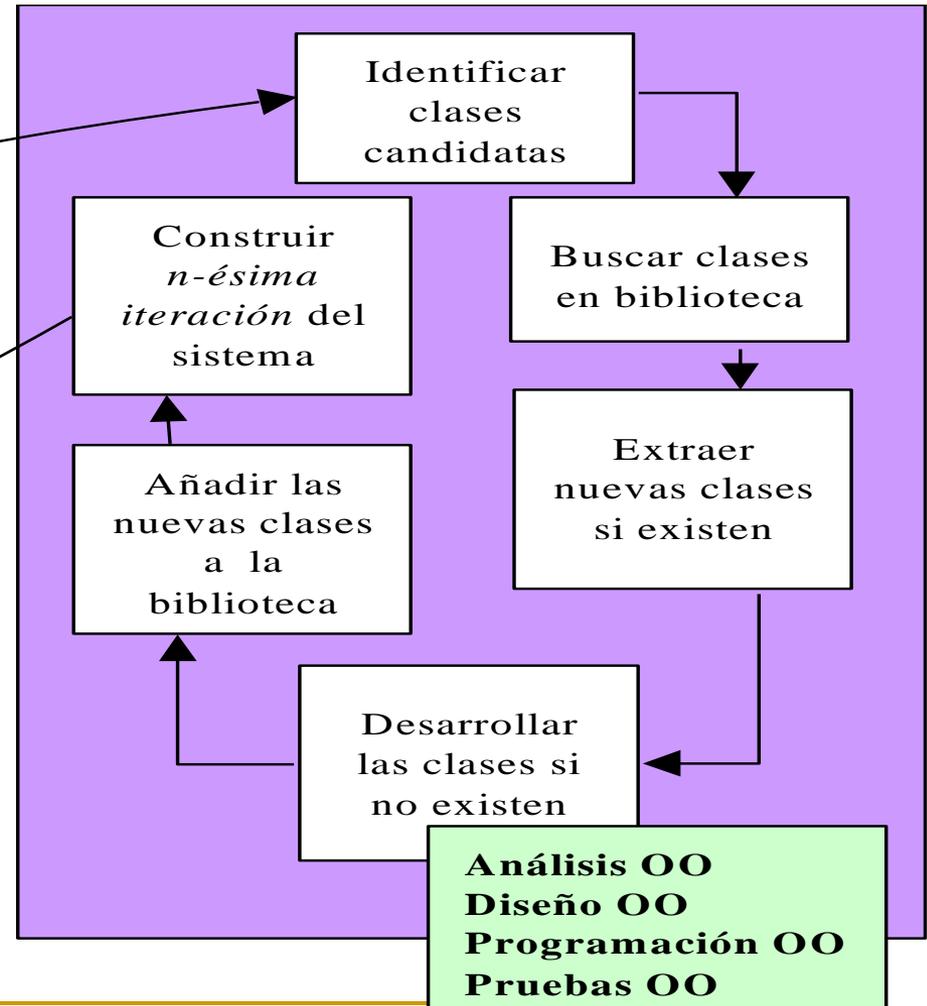
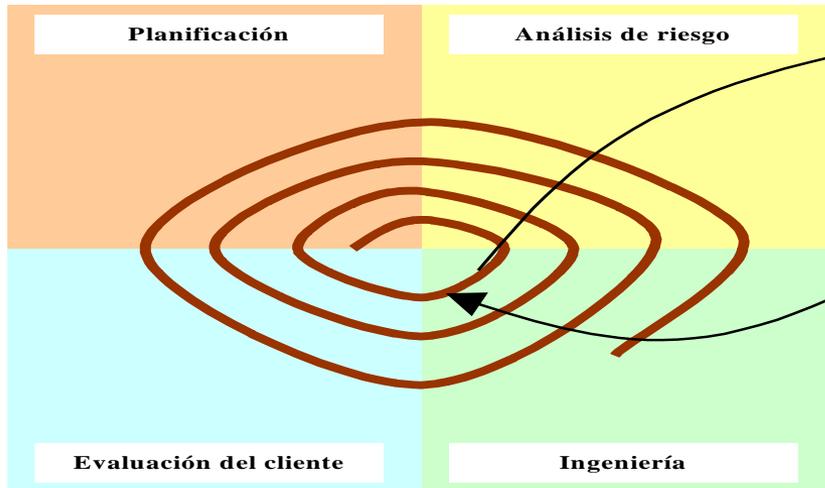


Modelo de Ensamblaje de Componentes



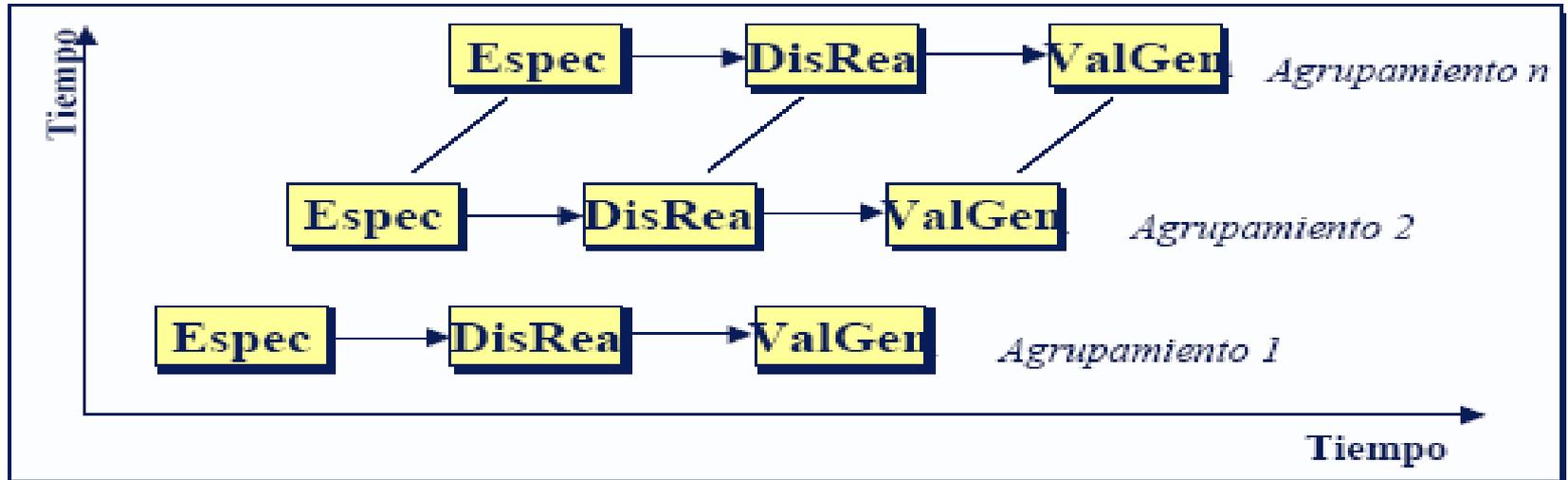
Incorpora muchas de las características del Modelo Espiral. Es evolutivo por naturaleza y exige un enfoque interactivo para la creación del software. Sin embargo, el modelo ensamblador de componentes configura aplicaciones desde componentes (reutilización). Esto se debe gracias a que, si se diseñan y se implementan adecuadamente, los componentes son reutilizables por las diferentes aplicaciones. **En primer lugar se identifica los componentes candidatos. Si estos componentes han sido creados por programas anteriores se almacenan en una biblioteca. Se determina cuáles de ellos ya existen a fin de reutilizarlos.** En caso de que no exista se desarrollan. Este proceso se inicia en el estado de Análisis de Riesgos del Espiral y se inserta en el estado de Construcción de Ingeniería.

4.- Paradigmas y Modelos del Ciclo de Vida



Ciclo de vida OO.

4.- Paradigmas y Modelos del Ciclo de Vida



Modelo Cluster (agrupamiento)

Propuesto por **Bertrand Meyer** [Meyer, 1990]

- ❑ *Cluster: Unidad organizativa básica. Es un grupo de clases relacionadas o, recursivamente, clusters relacionados.*
- ❑ El miniciclo de vida que gobierna el desarrollo de un *cluster* está formado por Especificación, Diseño, Implementación, Verificación/Validación y Generalización