

Introducción a la ingeniería de software

Ingeniería de Software 1

Temario

- Información de la materia
- Problemática del desarrollo de software
- Calidad e ingeniería de software
- Proceso y ciclo de vida
- Áreas de la ingeniería de software

Información de la materia

Información de la materia

- Objetivo
 - Brindar los conceptos teóricos y prácticos que permitan **comprender y ejecutar** los distintos **procesos** que involucra el **desarrollo del software**.

Temario de la materia

- Introducción a la ingeniería de software
- Proceso y ciclo de vida
- Requerimientos
- Test Driven Development (TDD)
- Gestión de proyectos
- Prueba
- Mediciones
- Estimación
- Aseguramiento de la calidad (SQA)
- Gestión de la configuración (SCM)
- Modelos metodológicos

Información de la materia

- Bibliografía principal
 - Sommerville. Ingeniería de Software 7 ed.
 - Pressman. Software Engineering: A Practitioner's Approach.
 - Pfleeger. Software Engineering – Theory and Practice.

Información de la materia

- Material
 - Las presentaciones y los ejercicios utilizados en clase estarán disponibles en el web de la materia.
 - <http://aulas.ort.edu.uy> (Moodle)
 - Las presentaciones son una guía durante la clase, no son un material de estudio completo.
 - Para profundizar, utilizar la bibliografía citada.

Información de la materia

- Evaluación
 - Obligatorio 1: 20 puntos
 - Obligatorio 2: 30 puntos
 - Parcial: 40 puntos
 - Actuación en clase: 10 puntos
- Aprobación: mínimo 70 puntos
- Exoneración: mínimo 86 puntos

Información de la materia

- Forma de trabajo
 - Clases teórico / prácticas, 3 hs. semanales
 - Consultas a los docentes
 - Devolución del obligatorio

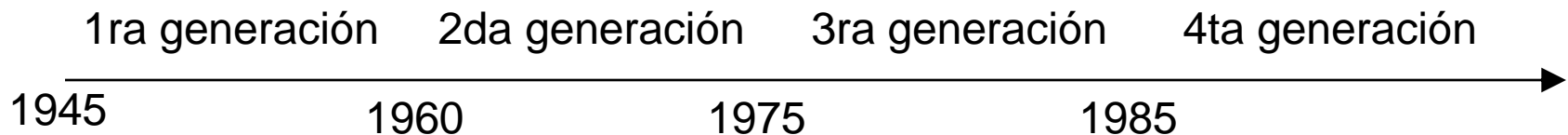
Problemática del desarrollo de software

Software

- Definición Software [Pressman]
 1. *Instrucciones* (programas de computadoras) que cuando se ejecutan *proporcionan la función y rendimiento* deseados.
 2. *Estructura de datos* que permiten a los programas manipular adecuadamente la información.
 3. *Documentos* que describen la operación y uso de los programas.

Evolución del software

- Primera generación
 - Válvulas
 - Procesos por lotes
 - Tarjetas perforadas
- Segunda generación
 - Transistor
 - Sistemas tiempo real
 - Multiprocesamiento
 - Base de datos
- Tercera generación
 - Circuitos integrados
 - Procesamiento distribuido
 - Aplicaciones de uso general
 - Interfaces interactivas
- Cuarta generación
 - Microprocesador
 - OOT y otras tecnologías
 - Computación en paralelo
 - Redes/Internet



Características del software

- **El hardware se fabrica, ensamblando componentes.**

El hardware se diseña a partir de ciertos componentes preestablecidos y reutilizables. A partir de su diseño se fabrica ensamblando dichos componentes previamente fabricados.

- ¿ Cómo le parece que se construye el software ?
- ¿ Qué impacto tiene en el esfuerzo necesario para su construcción relativa al hardware ?

Características del software

- **El hardware se desgasta y se reemplaza.**

El hardware tiene una probabilidad de exhibir fallos al principio de su vida, luego tiende a estabilizarse hasta que empieza a desgastarse y la probabilidad de fallos se incrementa nuevamente.

- ¿ Qué curva de tasa de fallos le parece que tendría el software ?
- ¿Cuál le parece que son las fuentes de cambio del software ?
- ¿Cuál sería el costo de introducir cambios en el software ?

Crisis del software

- Costos inversos de hardware vs software
- Aumento en la complejidad del software
 - Mayor tamaño, costo y tiempos de desarrollo
- Procesos inmaduros
 - Pobres técnicas de desarrollo y gestión
 - Falta de capacitación
- Sociedad dependiente de IT
 - Software elemento estratégico
 - Contextos de negocios cambiantes
 - Demanda creciente de clientes y usuarios



Aflicción crónica

Estadísticas

- Estudio de Standish Group (1994)
 - 365 empresas y 8380 aplicaciones
- Resultados
 - Exitosos 16.2 %
 - Terminados fuera de planificación 52.7 %
 - Costo promedio 189 %
 - Tiempo promedio 222 %
 - Alcance promedio 61 %
 - Cancelados 31.1%

Estadísticas

- Causas
 - Poco involucramiento del usuario 12.8 %
 - Requerimientos incompletos 12.3 %
 - Cambio de requerimientos 11.8 %
 - Falta de compromiso de directivos 7.5 %
 - Desconocimiento de las tecnologías 7.0 %
 - Falta de recursos 6.4 %
 - Expectativas poco realistas 5.9 %
 - Objetivos poco claros 5.3%
 - Plazos de tiempo poco realistas 4.3%
 - Nuevas tecnologías 3.7 %
 - Otros 23.0 %

Table 1**Standish project benchmarks over the years**

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

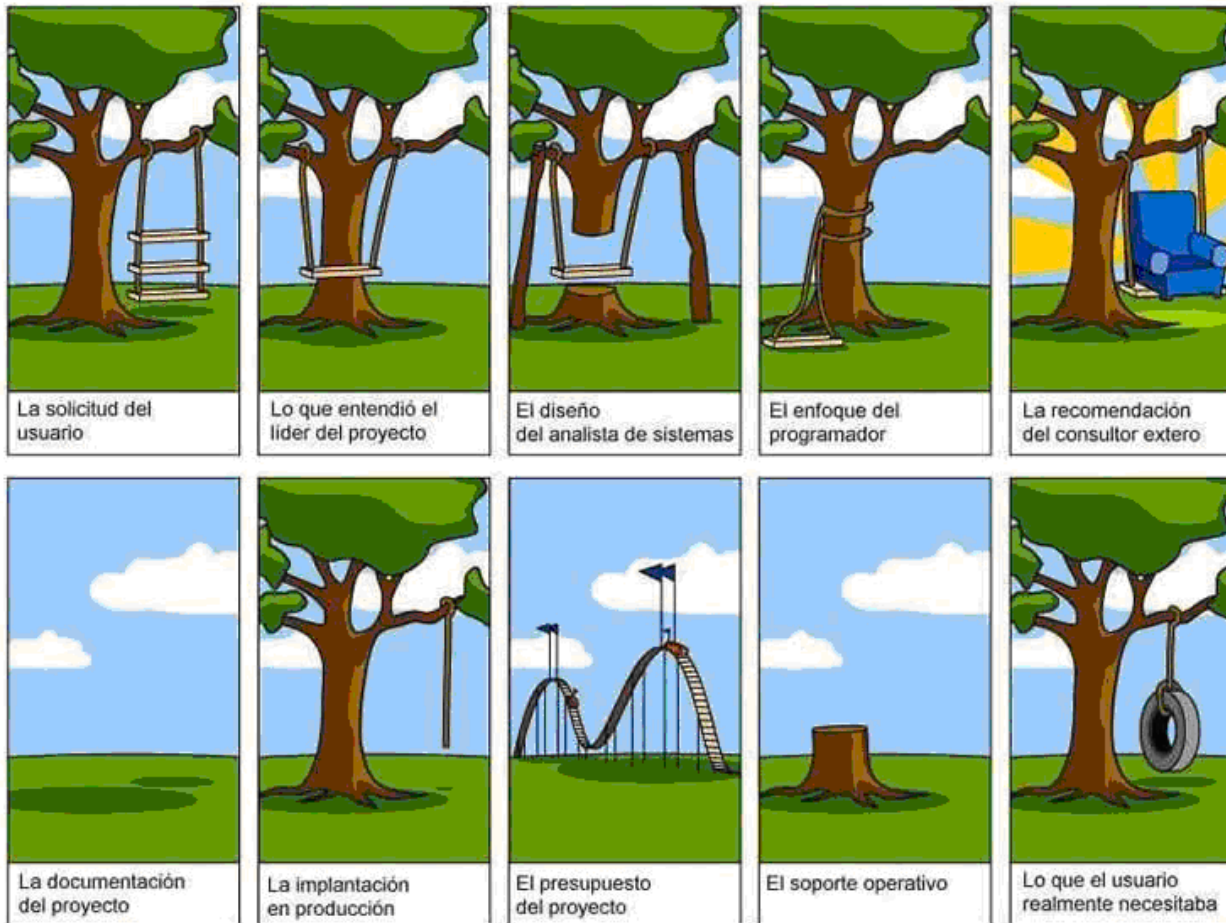
Successful - Completado en tiempo y presupuesto, con el alcance especificado

Challenged – Completado y operativo con presupuesto excedido, fuera del tiempo estimado y con menos funcionalidad que las originalmente especificadas

Failed – El proyecto se canceló en algún momento durante su desarrollo

Vrije J. Laurenz Eveleens and Chris Verhoef, "The Rise and Fall of the Chaos Report Figures", IEEE Software January/February 2010

Visiones de un proyecto



Posible solución

- Profesionalizar el desarrollo de software.
- Transformar el desarrollo de arte a ingeniería.
- Sistematizar actividades posibilitando espacio para la creatividad.
- **Aplicar ingeniería de software.**

Calidad e ingeniería de software

Calidad e ingeniería de software

- Definición de Ingeniería de Software
 - (1) La aplicación de un **enfoque sistemático, disciplinado y cuantificable** al **desarrollo, operación y mantenimiento** de software; es decir, la aplicación de ingeniería al software.
 - (2) El estudio de los enfoques como en (1).
[Estándar IEEE 610.12]

Calidad e ingeniería de software

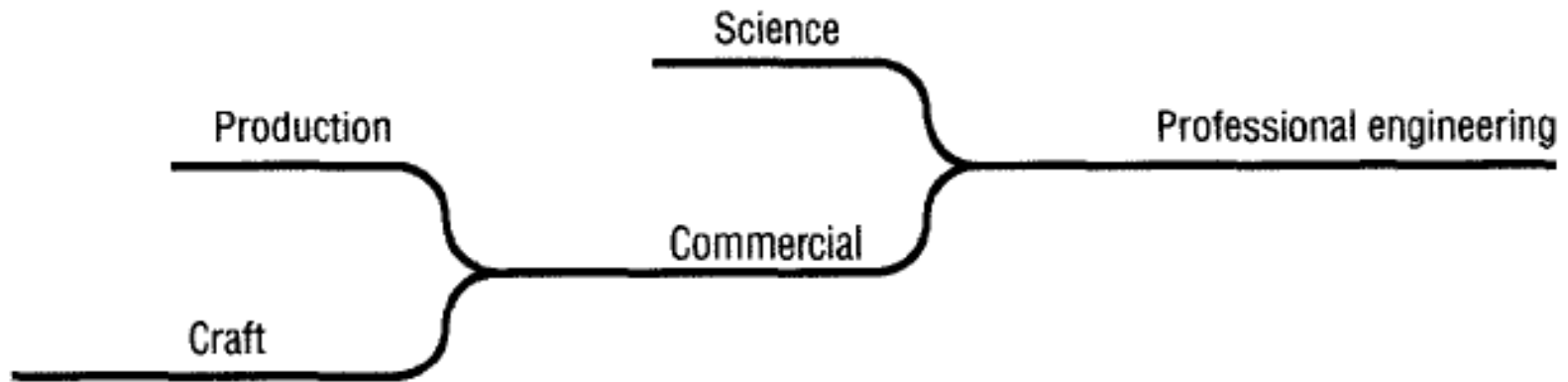
- Definición de Ingeniería de Software
 - **Disciplina tecnológica y gerencial** abocada a la producción y mantenimiento sistemático de productos de software que son desarrollados y modificados en **tiempo** y bajo estimaciones de **costos**. [Fairley]

Calidad e ingeniería de software

- Definición de Ingeniería [Mary Shaw]
 - Crear soluciones eficientes (económicas)
[y eficaces (satisfactorias para consumo)]
 - Para problemas prácticos
 - Aplicando conocimientos científico
 - Para construir artefactos
 - Al servicio de la humanidad

Evolución de Ingeniería

(Mary Shaw)

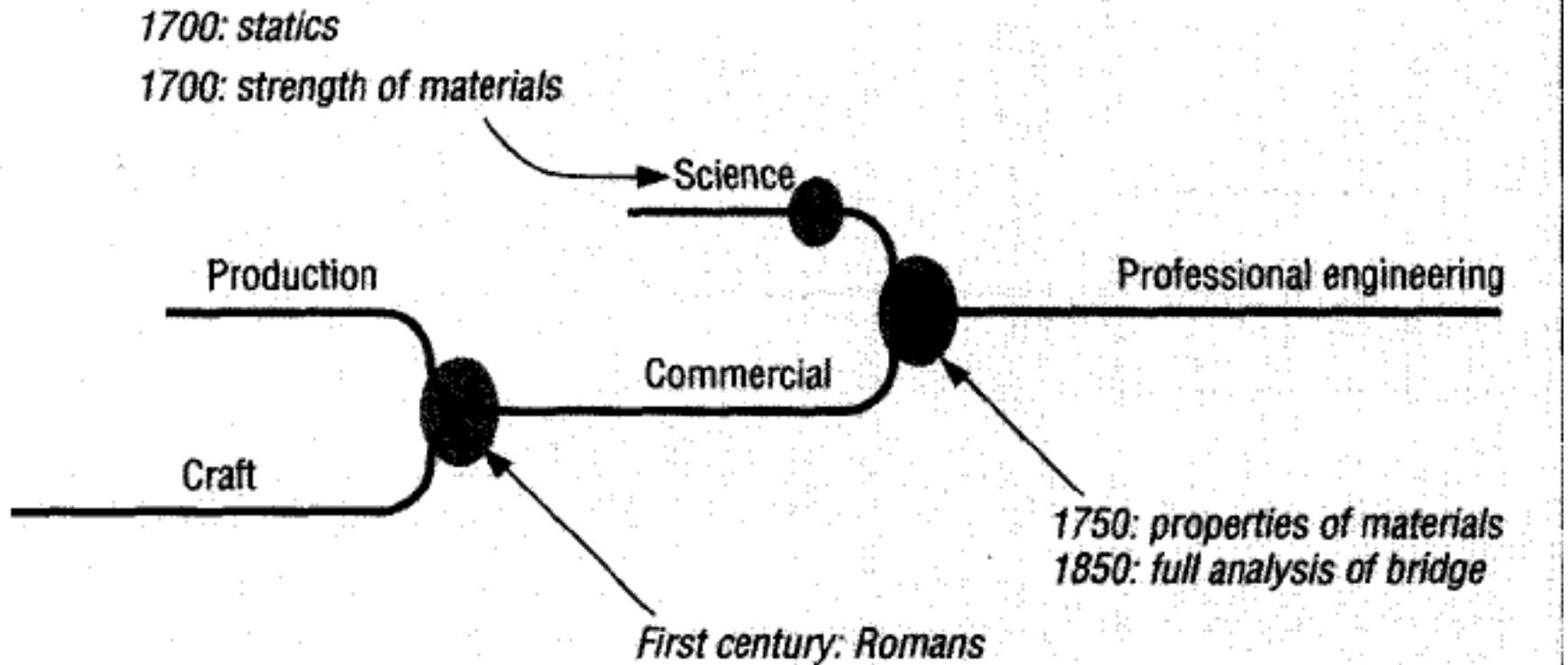


- Virtuosos
- Intuición
- Manufactura para uso

- Artesanos entrenados
- Procedimientos
- Manufactura para vender

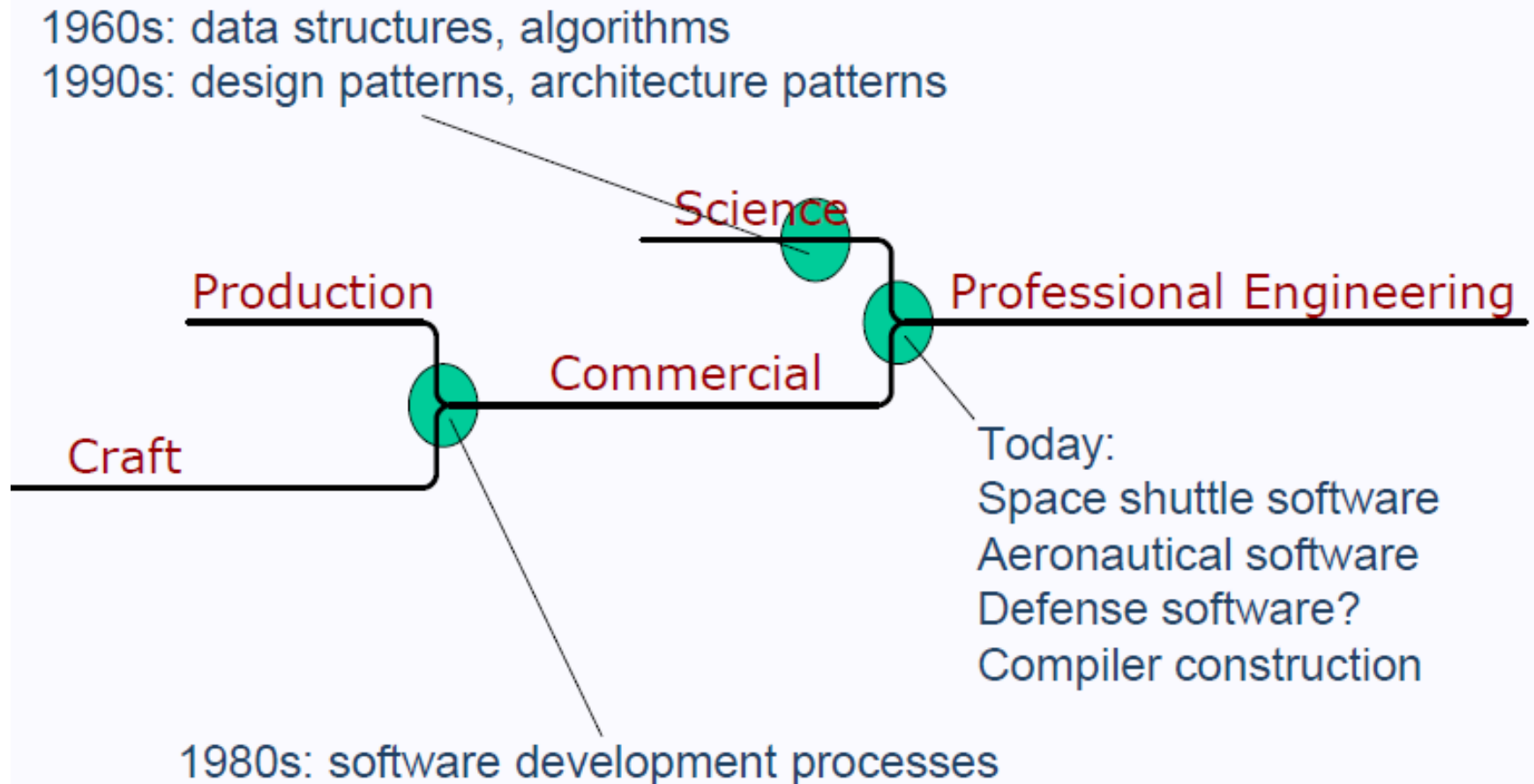
- Profesionales
- Teoría establecida
- Segmentación de mercados por producto

Evolución de Ingeniería Civil

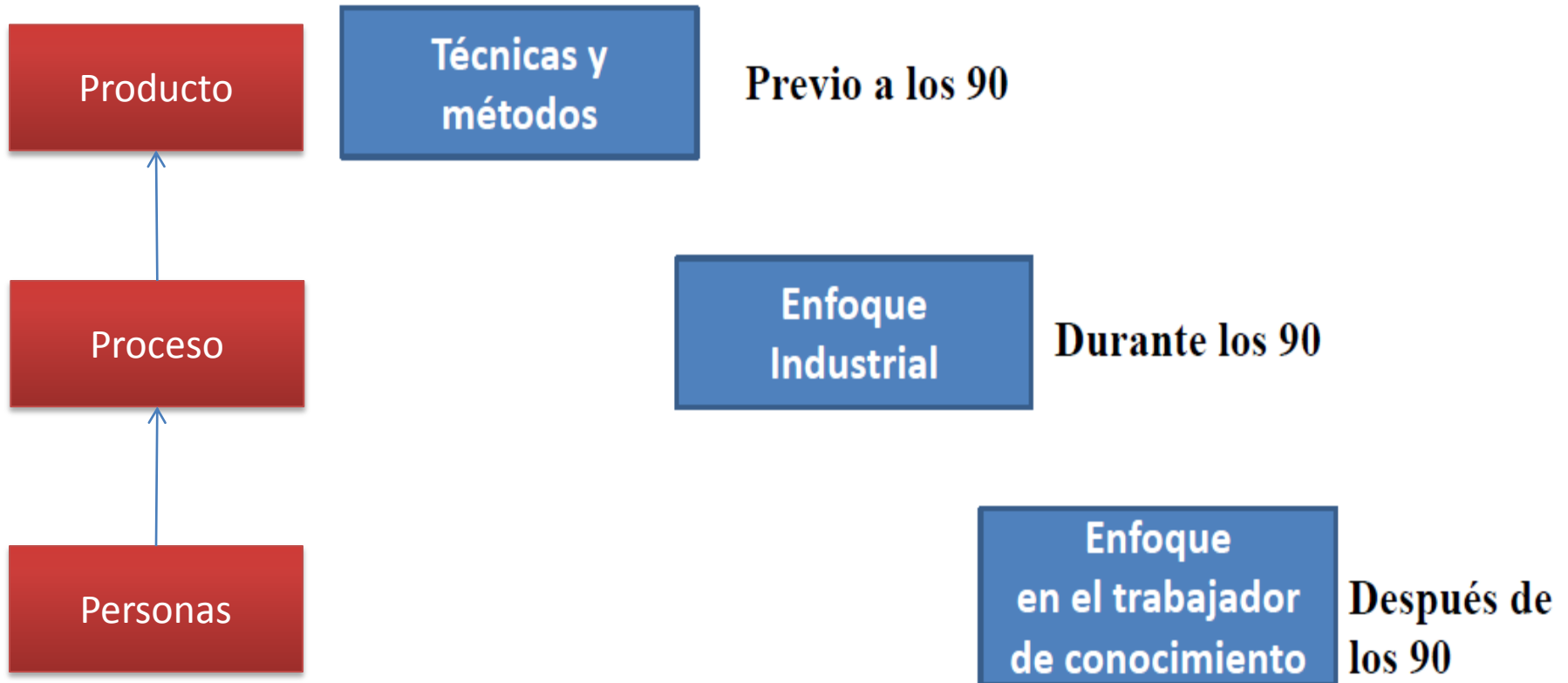


Donde se encuentra evolucion Ing. De Software?

Evolución de IS



Evolución de IS



Calidad e ingeniería de software

- Definición de Ingeniería de Software
 - Aplicación disciplinada de **principios de ingeniería, ciencia y matemática, métodos y herramientas** a la producción **económica** de software. [Humphrey]

Calidad e ingeniería de software

- Definición de calidad [Juran 1999]
 - (a) Calidad es el conjunto de ***características*** de un producto que ***satisfacen las necesidades de los clientes*** y, en consecuencia, hacen satisfactorio el producto
 - (b) Calidad consiste en ***no tener deficiencias*** en el producto o en el proceso

Calidad e ingeniería de software

- Calidad e ingeniería de software
 - La ingeniería de software, como disciplina tecnológica y gerencial, busca aplicar ciertas prácticas a la producción de software, con el objetivo de lograr la calidad.

Calidad e ingeniería de software

- La IS es tecnología multicapa:
 - Capa 1: enfoque de calidad.
 - Capa 2: procesos.
 - Capa 3: métodos.
 - Capa 4: herramientas.

Calidad e ingeniería de software

- Enfoque de calidad.
 - Es la forma de gestionar la calidad para mejorar los procesos, ej: ISO, CMM, etc.
- Procesos.
 - Delimitan marco de trabajo.
 - Definen pasos a dar.
 - Consideran las áreas clave para definirlo, ejecutarlo y mejorarlo.
- Métodos.
 - Indican como construir técnicamente el sw.
 - Existen distintos métodos para cada actividad.
 - Diseño, programación, gestión, etc.
- Herramientas.
 - Automatizan los métodos.
 - CASEs, gestores, workflows, etc.

Proceso y ciclo de vida

Proceso

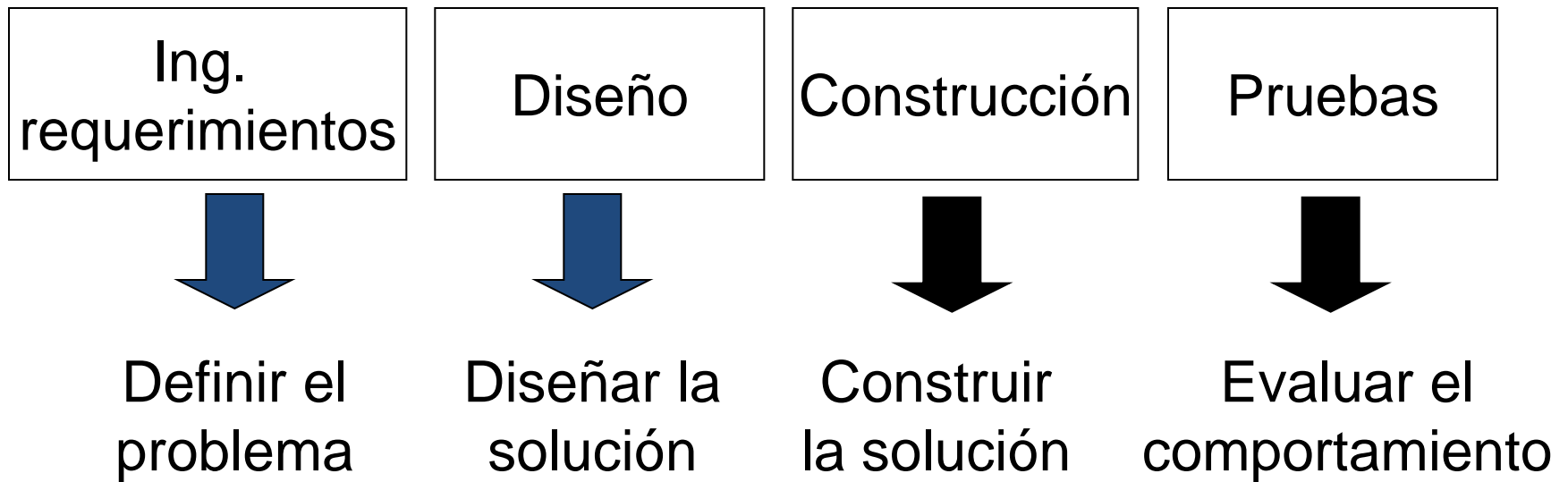
- Proceso
 - Una serie de pasos que involucran actividades, restricciones y recursos con el fin de producir un resultado esperado de algún tipo. [Pflegger]
- Proceso de Software
 - Define las **actividades, prácticas, técnicas, roles, herramientas** necesarias para transformar los requerimientos del usuario en un producto de software. [Jacobson]

Proceso y ciclo de vida

- Proceso de software
 - En general las actividades del proceso de software se dividen en dos grandes grupos:
 - Procesos de ingeniería
 - Comprenden actividades inherentes a la construcción del producto.
 - Procesos de apoyo
 - Comprenden actividades inherentes a la gestión de los procesos de construcción del producto.

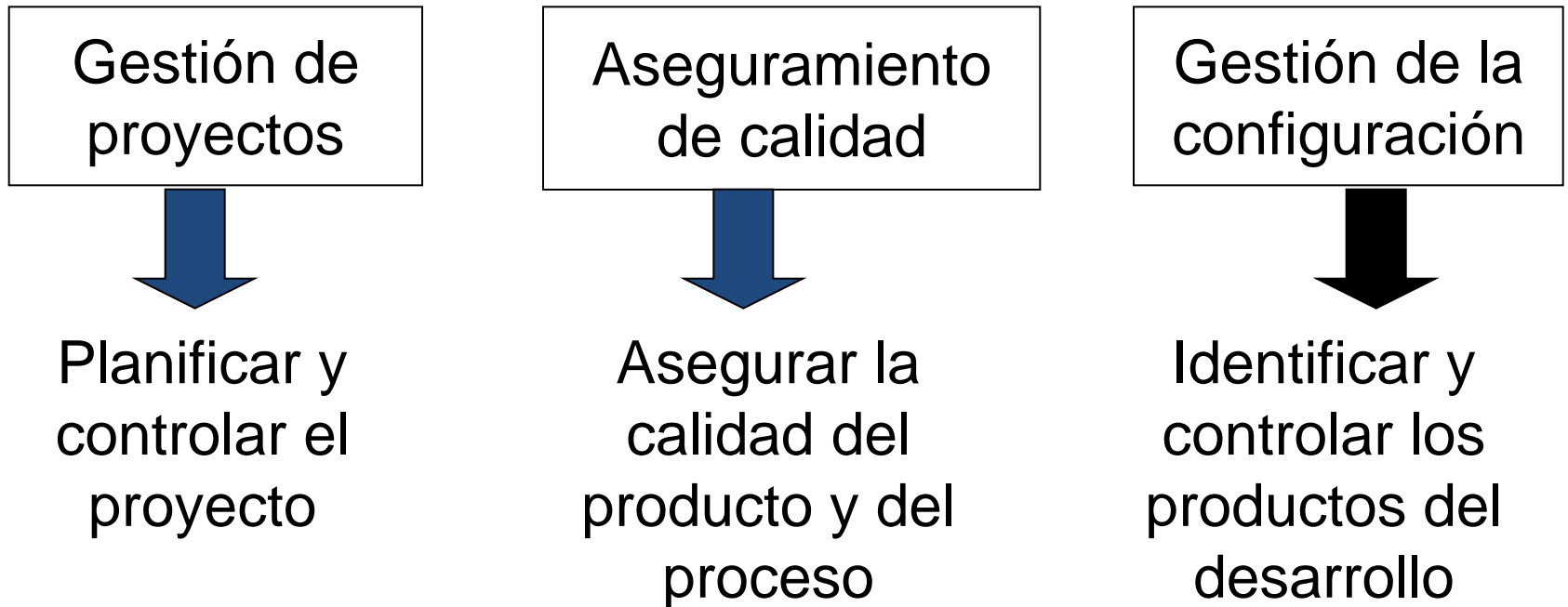
Proceso y ciclo de vida

Procesos o actividades típicas de ingeniería

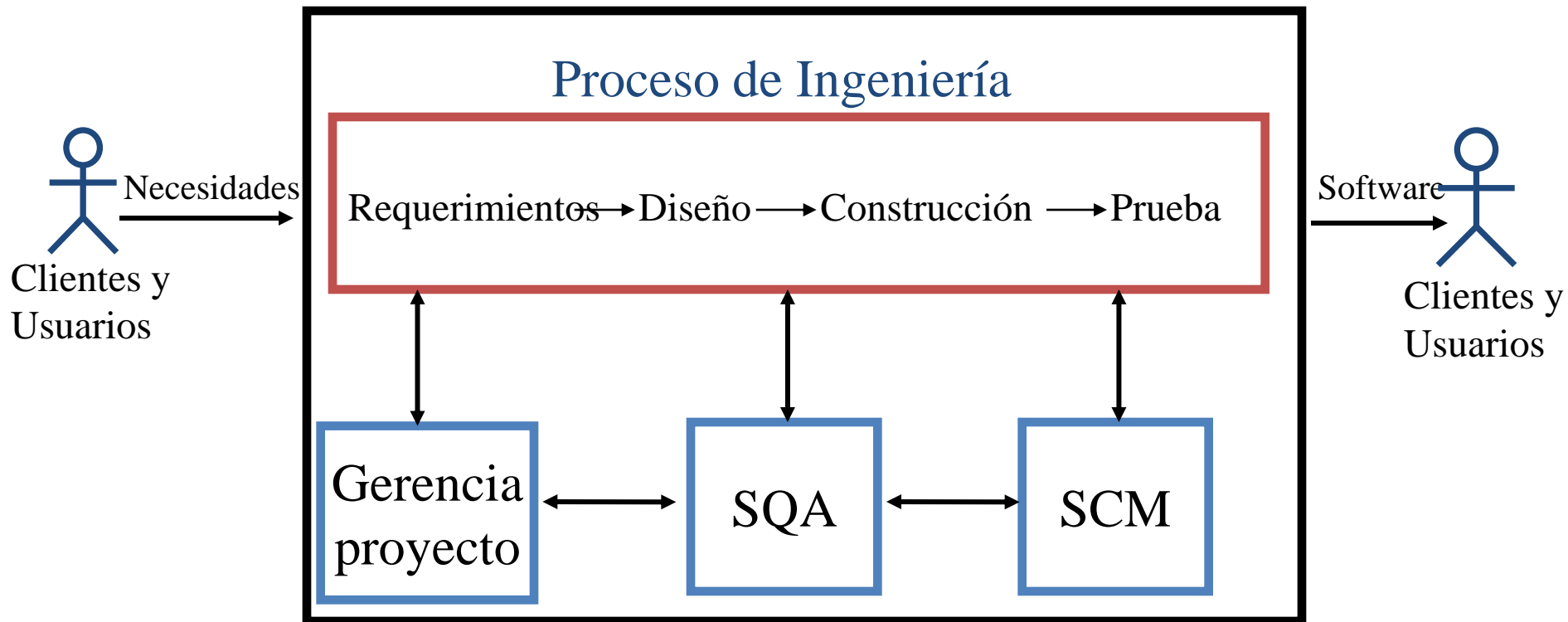


Proceso y ciclo de vida

Procesos o actividades típicas de apoyo



Proceso y ciclo de vida



Proceso y ciclo de vida

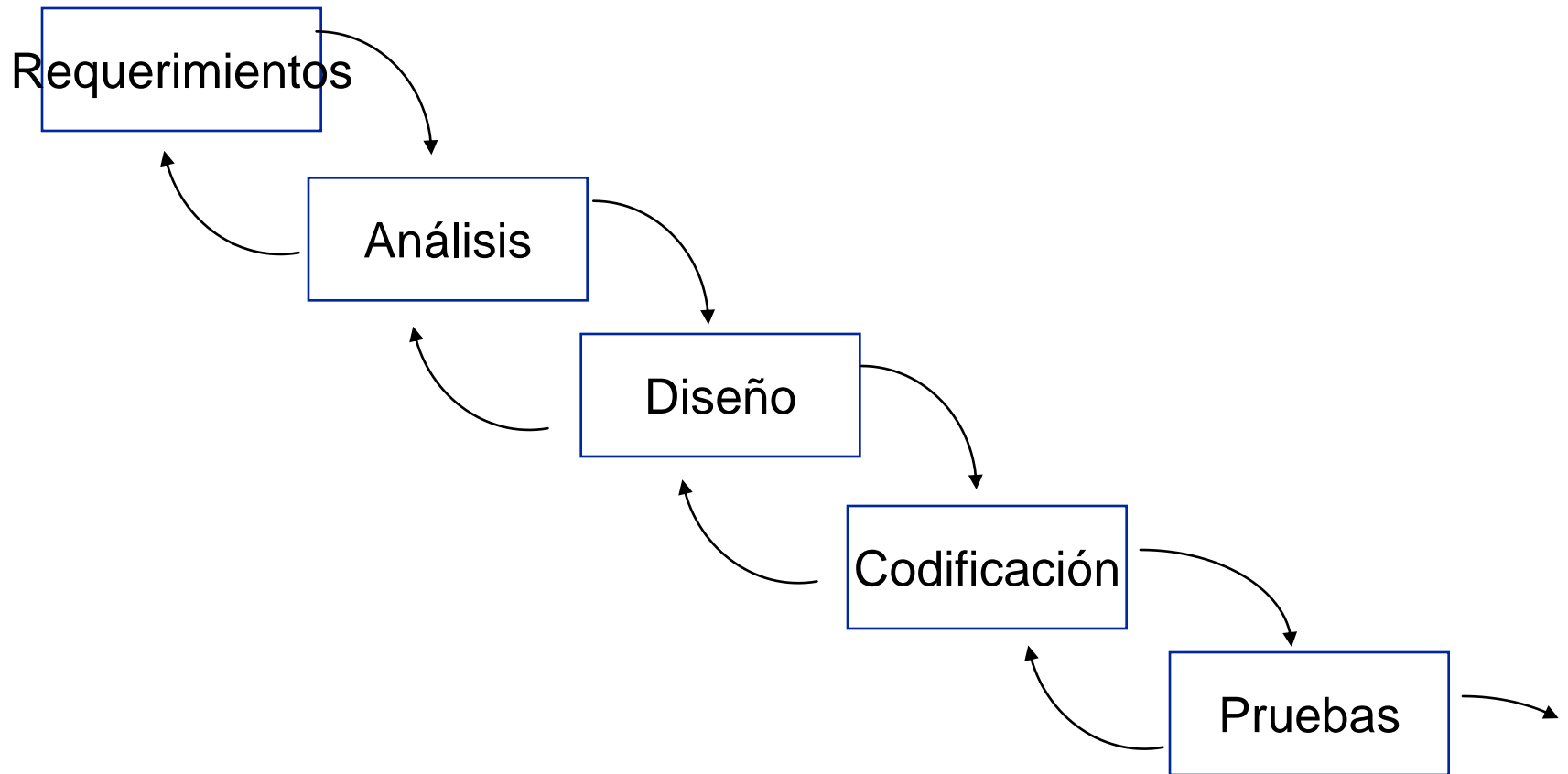
- Un proceso debe definir QUE hacer, QUIEN debe hacerlo, CUANDO y COMO hacerlo para alcanzar una meta.
- Componentes de un proceso
 - Marco común de trabajo, una filosofía de calidad.
 - Tareas (análisis, diseño, prueba, etc.) y actividades de apoyo (mediciones, revisiones, versionado, etc.)
 - Roles y responsabilidades
 - Métodos técnicos
 - Ciclo de vida del desarrollo

Proceso y ciclo de vida

- Ciclo de vida
 - Es un modelo de referencia de alto nivel de las actividades necesarias para el desarrollo de software.
 - Un modelo de ciclo de vida:
 - Describe las principales fases y actividades del desarrollo.
 - Define el orden de las fases.
 - Sirve para “saber donde estamos” durante un proyecto.

Proceso y ciclo de vida

- Ciclo de vida en cascada



Proceso y ciclo de vida

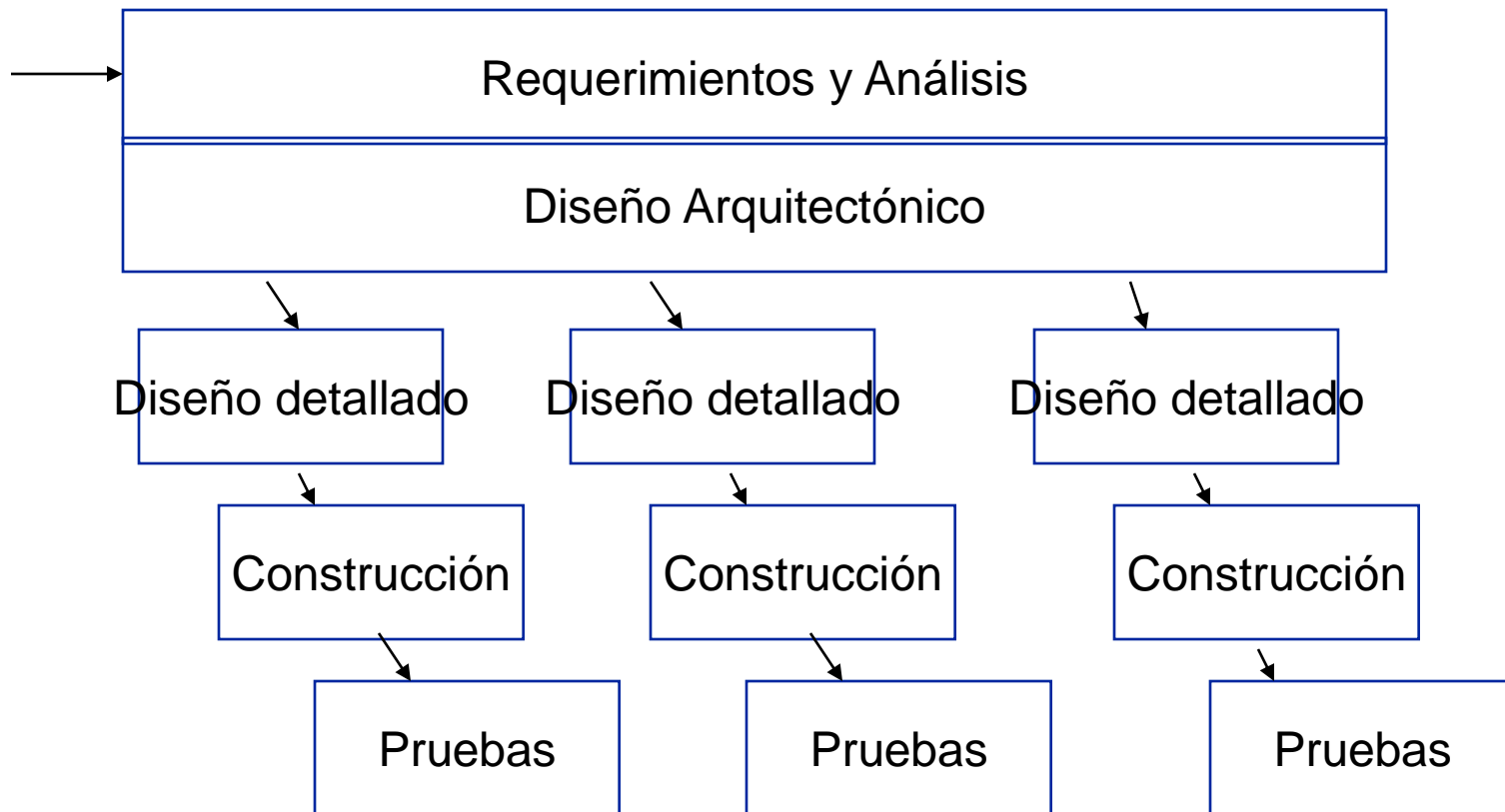
- Ciclo de vida en cascada
 - El proyecto progresa a través de una secuencia ordenada de fases.
 - Revisión al final de cada fase para determinar si se pasa a la siguiente.
 - El software como producto tangible se obtiene al final del ciclo.

Proceso y ciclo de vida

- Ciclo de vida en cascada
 - Buena performance en proyectos con especificaciones estables y bien comprendidas. (Por ejemplo: proyectos de mantenimiento)
 - No es flexible
 - Los Requerimientos detectados tardíamente suelen ser muy costosos
 - Se dificulta la vuelta a una etapa anterior
 - Feedback de defectos tardío. El costo de corrección puede ser enorme dependiendo del tipo de defecto encontrado

Proceso y ciclo de vida

- Ciclo de vida incremental

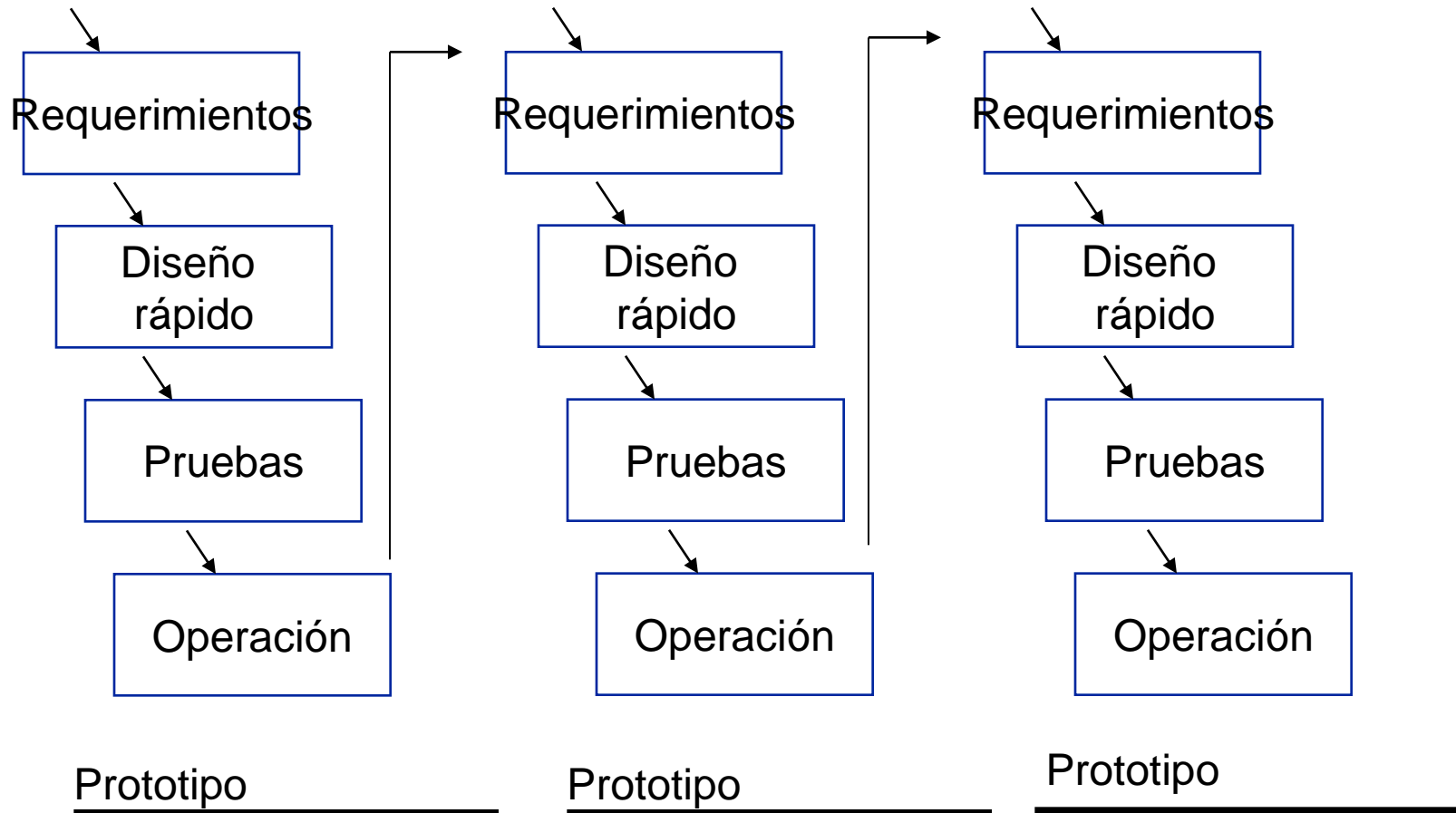


Proceso y ciclo de vida

- Ciclo de vida incremental
 - En la primera vuelta se hace casi todo el relevamiento y análisis.
 - Se itera sobre Diseño, Codificación y Pruebas
 - El cliente va viendo el avance.

Proceso y ciclo de vida

- Ciclo de vida evolutivo



Proceso y ciclo de vida

- Ciclo de vida evolutivo
 - Las iteraciones incluyen análisis.
 - Útil cuando
 - Requerimientos poco definidos
 - El dominio de la aplicación es poco conocido para el desarrollador y para el cliente
 - Produce resultados visibles en forma temprana
 - Puede ser difícil de determinar cuando el producto es aceptable

Proceso y ciclo de vida

- Beneficios de un proceso y ciclo de vida para el software:
 - Estandarizar
 - Entrenar y comunicar
 - Gestionar y realizar seguimiento de proyectos
 - Mejorar con la experiencia de cada proyecto
- No existe un proceso o ciclo de vida estándar para el desarrollo de software

Áreas de la ingeniería de software

Áreas de la ingeniería de software

- Ingeniería de requerimientos de software
 - Definición o identificación de los servicios o capacidades que debe proveer el software para satisfacer las necesidades de los usuarios.
 - Definición del problema a resolver.

Áreas de la ingeniería de software

- Diseño de software
 - Diseño arquitectónico
 - Determinar la estructura general del sistema.
 - Asignar los requerimientos del software a los subsistemas.
 - Definir interfaces de los subsistemas.
 - Etc.

Áreas de la ingeniería de software

- Diseño de software
 - Diseño detallado
 - Descomponer el sistema en clases o módulos.
 - Seleccionar mecanismos y algoritmos.
 - Asignar responsabilidades a módulos.
 - Etc.
- Construcción del software
 - Codificar, validar y probar unitariamente el software.

Áreas de la ingeniería de software

- Pruebas del software
 - Integración de los distintos componentes del software.
 - Verificación dinámica del software contra la especificación del comportamiento esperado.
 - Existen distintos niveles de prueba y distintos objetivos de las pruebas

Áreas de la ingeniería de software

- Mantenimiento del software
 - Modificar el software una vez entregado para corregir fallas o incorporar a cambios.

Áreas de la ingeniería de software

- Gestión de proyectos de software.
 - Planificación y seguimiento de las actividades necesarias para la construcción y mantenimiento del software.
- Gestión de la configuración del software
 - Identificación y control del software y sus componentes durante su construcción, mantenimiento y uso.

Áreas de la ingeniería de software

- Calidad del software
 - Asegurar la calidad del producto y de los procesos durante el ciclo de vida del proyecto.
- Procesos de ingeniería del software
 - Definición, implantación y mejora del proceso de software.

Áreas de la ingeniería de software

- Herramientas y métodos.
 - Ambientes de desarrollo para software y métodos para la realización de las distintas actividades del desarrollo del software.

Bibliografía

- Capítulos 1 y 2, Ingeniería del Software, 4ta. Edición. R. S. Pressman, 1998
- Capítulos 1 y 2, Software Engineering. Shari Pfleeger, 2001.
- Software Engineering Body of knowledge. IEEE, 2003.
- Software Engineering 2 ed., vol. 1 y 2. R. Thayer, I. Sommerville, 2002.
- Object-Oriented Software Engineering. I. Jacobson, 1993.
- Agile Software Development. A. Cockburn, 2002.
- Material del curso ISO 9000 en Informática. Amalia Alvarez, 2001.