

## El Proceso

Capítulo 2  
Roger Pressman,  
5ª Edición

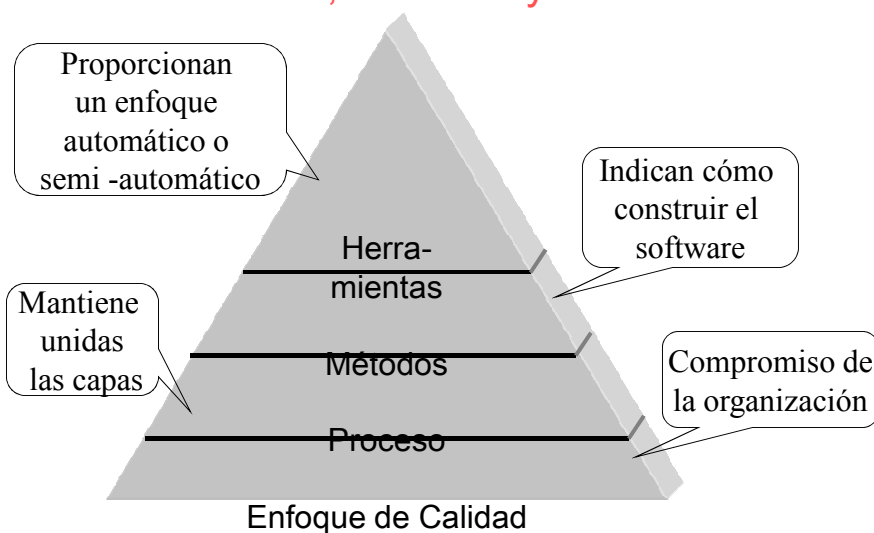
### El Proceso de Desarrollo de Software

- ¿Qué es?  
Marco de trabajo de tareas a realizar para desarrollar Software de alta calidad.
- ¿Es sinónimo de Ingeniería de Software?  
Define un enfoque para desarrollar software en forma ingenieril, pero la IS comprende, además, de Métodos y Herramientas.

## 2.1 Ingeniería de Software una tecnología estratificada

- Definiciones de Ingeniería de Software
  - (Fritz Bauer, 1969). Es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales.
  - (IEEE, 1993). Ingeniería de Software (IS) es:
    - 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de la IS.
    - 2) El estudio de los enfoques como en 1.

### 2.1.1 Proceso, métodos y herramientas



## 2.1.2 Una visión general de la IS

- Independientemente de la *entidad* a la que se va a aplicar la ingeniería, se deben resolver las siguientes preguntas:
  - ¿Cuál es el problema a resolver?
  - ¿Cuáles son las características de la entidad que se utilizan para resolver el problema?
  - ¿Cómo se realiza la entidad (y la solución)?
  - ¿Cómo se construirá la entidad?
  - ¿Qué enfoque se va a utilizar para evitar los errores actuales?
  - ¿Cómo se mejorará a la larga la entidad?

## 2.1.2 Una visión general de la IS

- La entidad que nos interesa es el software.
- El trabajo a realizar en la IS se puede dividir en tres fases genéricas.
  - Definición. Centrada en el qué.
  - Desarrollo. Centrada en el cómo.
  - Mantenimiento. Se centra en el cambio:
    - Corrección, adaptación, mejora y prevención.

## 2.1.2 Una visión general de la IS

- Para realizar bien las fases genéricas existen una serie de actividades protectoras, entre ellas están:
  - Seguimiento y control del proyecto
  - Revisiones Técnicas Formales
  - Garantía de Calidad del Software
  - Gestión de la Configuración del Software
  - Preparación y producción de documentos
  - Gestión de reutilización
  - Mediciones
  - Gestión de riesgos

## 2.2 El Proceso de Software



## 2.2 El Proceso de Software

- El Software Engineering Institute (SEI) ha desarrollado un modelo completo que se basa en un conjunto de funciones de IS que deberían estar presentes conforme las organizaciones adquieren madurez en sus procesos.
- Existen 5 niveles de madurez.

## 2.2 El Proceso de Software

- Los 5 niveles de madurez del CMM:
  - Nivel 1: Inicial. Pocos procesos el éxito depende del esfuerzo personal.
  - Nivel 2: Repetible. Se establecen procesos de gestión.
  - Nivel 3: Definido. Se usan estándares y documentación.
  - Nivel 4: Gestionado. Se recopilan medidas.
  - Nivel 5: Optimización. Se posibilita la mejora.

## 2.2 El Proceso de Software

- SEI ha definido ACP (Áreas Clave de Proceso) para cada nivel, mismas se identifican con las características siguientes:
  - Objetivos
  - Compromisos
  - Capacidades.
  - Actividades.
  - Métodos de supervisar la implantación.
  - Métodos para verificar la implantación.

## 2.2 El Proceso de Software

- Las ACP por nivel son:
  - Nivel 2: Repetible
    - Gestión de la configuración.
    - SQA
    - Gestión de subcontratación.
    - Seguimiento y supervisión del proyecto.
    - Planificación del proyecto.
    - Gestión de requisitos
  - Nivel 3: Definido.
    - Revisiones periódicas
    - Coordinación entre grupos
    - Ingeniería de Productos de Software
    - Gestión de integración del software
    - Programa de formación
    - Definición del proceso de la aorganización
    - Enfoque del proceso de la organización

## 2.2 El Proceso de Software

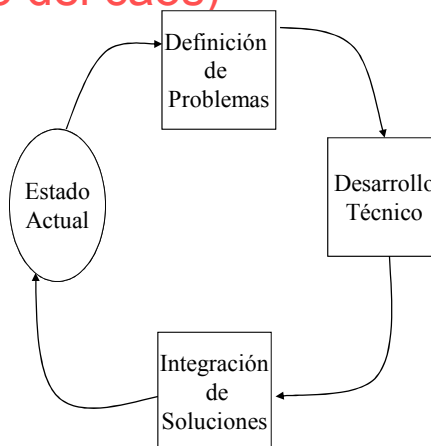
- Las ACP por nivel son:
  - Nivel 4: Gestionado.
    - Gestión de calidad de software
    - Gestión cuantitativa del proceso
  - Nivel 5: Optimización.
    - Gestión de cambios del proceso
    - Gestión de cambios de tecnología
    - Prevención de defectos

## 2.3 Modelos de Proceso del Software

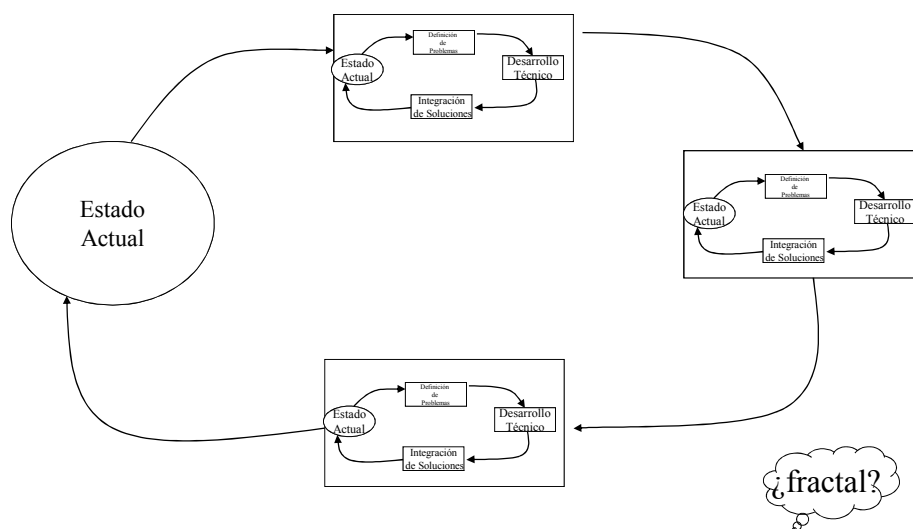
- Se debe escoger una estrategia de desarrollo, llamada:
  - modelo de proceso o
  - paradigma de IS
- Se selecciona de acuerdo a:
  - naturaleza del proyecto y aplicación
  - métodos y herramientas
  - controles y entregas requeridas

## Fases de un bucle de resolución de problemas (modelo del caos)

- Todo desarrollo de software se puede caracterizar con 4 etapas, como en la figura, que buscan:
  - entender el estado actual de sucesos
  - identificar el problema a resolver
  - aplicar tecnología para solucionarlo
  - presentar resultados integrados



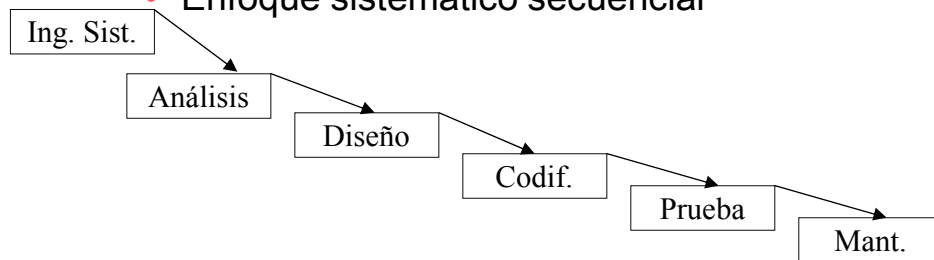
## Fases dentro de las fases del bucle de resolución de problemas





## 2.4 Modelo Lineal Secuencial (1)

- Ciclo de vida clásico, modelo en cascada
- más antiguo, más usado
- Enfoque sistemático secuencial

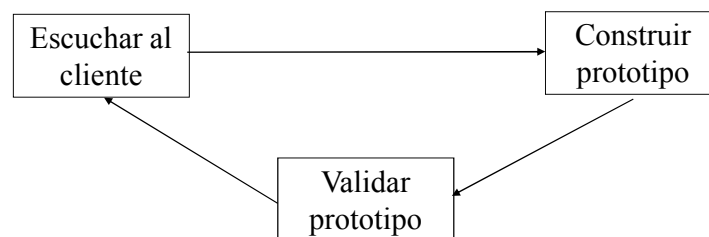


## 2.4 Modelo Lineal Secuencial (2)

- Críticas:
  - Proyectos reales raras veces se ajustan.
  - Raras veces cliente expone todos los req. de entrada.
  - Producto operativo al final => Paciencia (cliente) alta.
- Consejo:
  - Usar cuando todos los requerimientos que han sido establecidos claramente de entrada.

## 2.5 Modelo de Construcción de Prototipos (1)

- No están claros los reqs. de entrada
- Iterativo. ¿Hasta cuando se itera?
- Working prototype, desechar y empezar con desarrollo de sistema.

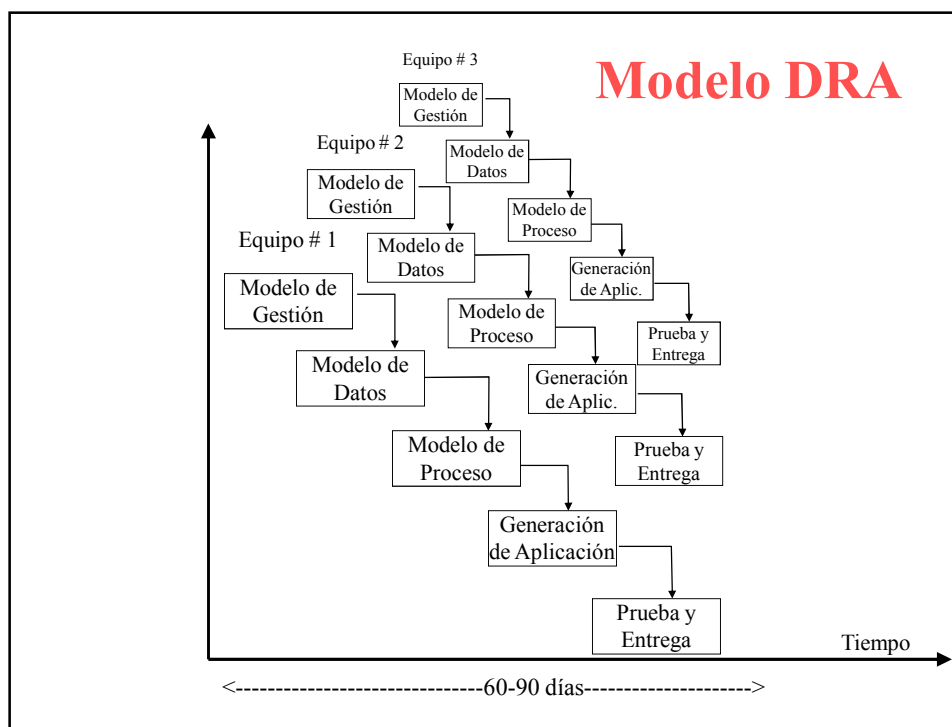


## 2.5 Modelo de Construcción de Prototipos (2)

- Críticas:
  - Cliente cree que es el sistema.
  - Peligro de familiarización con malas elecciones iniciales (quick and dirty).
- Consejo:
  - Usar cuando inicialmente no están claros los requerimientos.
  - Definir claramente de entrada las reglas de juego con el cliente.
  - No ceder a presión del cliente.

## 2.6 Modelo de Desarrollo Rápido de Aplicaciones (DRA) (1)

- Lineal secuencial con ciclo extremadamente corto.
- Candidatos: sistemas que se pueden modularizar => equipos de desarrollo paralelos.
- Basado en el uso de componentes y T4G.



## 2.6 Modelo de Desarrollo Rápido de Aplicaciones (DRA) (3)

- El enfoque DRA comprende las siguientes fases:
- Modelo de Gestión:
  - ¿Qué información conduce?
  - ¿Qué información se genera?
  - ¿Quién la genera?
  - ¿A dónde va?

## 2.6 Modelo de Desarrollo Rápido de Aplicaciones (DRA) (4)

- Modelo de Datos:  
**Identificación de Objetos y relaciones**
- Modelo de Procesos:  
**Descripciones de procesos de negocio para ABM de objetos de MD**

## 2.6 Modelo de Desarrollo Rápido de Aplicaciones (DRA) (5)

- Generación de Aplicaciones:  
**T4G + Reusabilidad de Componentes**
- Pruebas y Entregas:  
**Prueba de Componentes Nuevos e interfaces.**

## 2.6 Modelo de Desarrollo Rápido de Aplicaciones (DRA) (6)

- Críticas:
  - Proyectos grandes => gran número de personas.
  - Alto compromiso en tiempo.
  - No apto para todo tipo de sistema (baja reutilización de componentes).
  - Desaconsejable cuando existen riesgos tecnológicos altos o alta interoperatividad con programas ya existentes.

## 2.7 Modelos Evolutivos (1)

- Se adaptan más fácilmente a los cambios introducidos a lo largo del desarrollo.
- Iterativos
- En cada iteración se obtienen versiones más completas del Software.

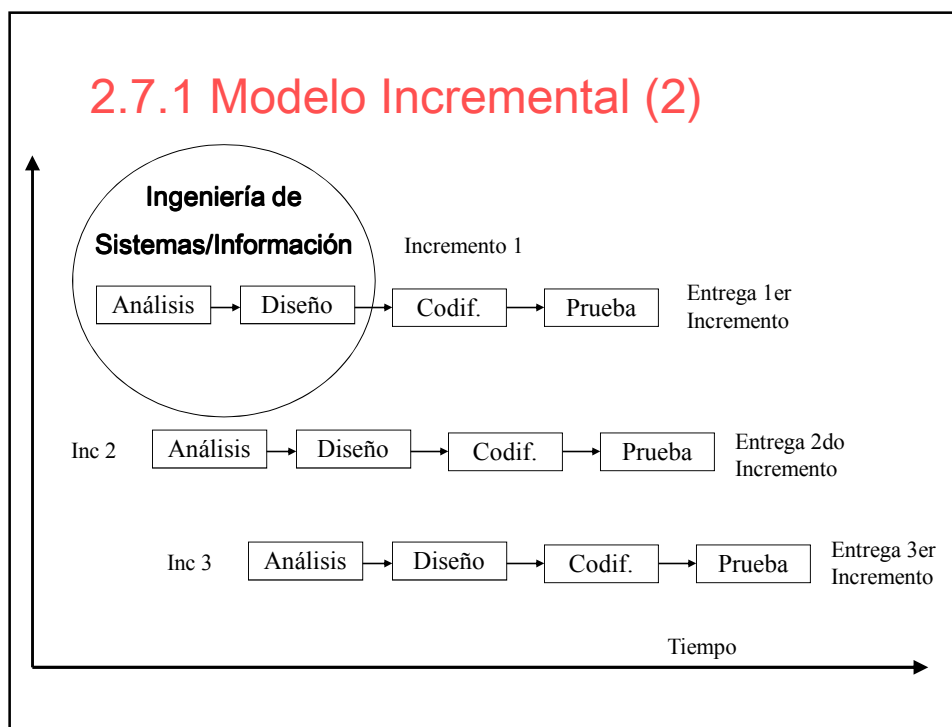
## 2.7 Modelos Evolutivos (2)

- Modelos Evolutivos:
  - Modelo Incremental
  - Modelo en Espiral
  - Modelo de Desarrollo Basado en Componentes
  - Modelo WINWIN
  - Modelo de Desarrollo Concurrente

## 2.7.1 Modelo Incremental (1)

- Iteración de Lineal Secuencial.
- Cada iteración devuelve un “Incremento” o versión operativa.
- Útil cuando no se está seguro de cumplir con plazos de tiempo o se tiene una fecha imposible de cambiar.

## 2.7.1 Modelo Incremental (2)



### 2.7.2 Modelo en Espiral (1)

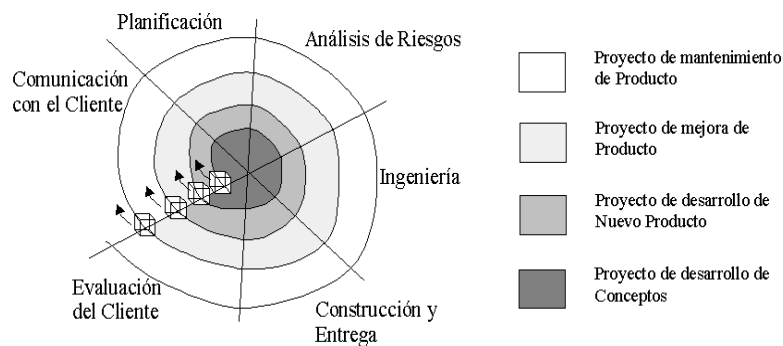
- Útil para proyectos grandes.
- Permite usar el prototipado en todas las etapas de la evolución para reducir el riesgo.
- Mantiene el enfoque sistemático de los pasos sugeridos por el lineal secuencial, pero lo incorpora dentro de un marco iterativo más real.

### 2.7.2 Modelo en Espiral (2)

- Críticas:
  - Difícil de convencer a los clientes de que es controlable.
  - Requiere mucha habilidad para el análisis de riesgos y de esta habilidad depende su éxito.
  - No ha sido utilizado tanto como el lineal secuencial o el de prototipos.



## 2.7.2 Modelo en Espiral (3)



## 2.7.2 Modelo en Espiral (4)

- Comunicación con el cliente:
  - Tarea requerida para la comunicación entre el desarrollador y el cliente.
- Planificación:
  - Definición de recursos, tiempos y otra información relacionada con el proyecto.

### 2.7.2 Modelo en Espiral (5)

- Análisis de Riesgos:
  - Tareas para evaluar riesgos técnicos y de gestión
- Ingeniería:
  - Tarea para crear una o más representaciones de la aplicación

### 2.7.2 Modelo en Espiral (6)

- Construcción y Acción:
  - Tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (Ej. Documentación y práctica).
- Evaluación del Cliente:
  - Tarea para obtener la reacción del cliente según la evaluación de la representación del software.

### 2.7.3 El Modelo en Espiral WINWIN (Victoria&Victoria) (1)

- Su objetivo es el mostrar los requisitos al cliente.
- El desarrollador simplemente pregunta al cliente lo que se necesita.
- El cliente por su parte proporciona detalles suficientes para continuar.

### 2.7.3 El Modelo en Espiral WINWIN Victoria&Victoria (2)

