

Ingeniería de Software

- ◆ Diseño, construcción y mantenimiento de sistemas de software grandes.

Diapositivas Traducidas por:

Dr. Pedro Mejía Alvarez.
CINVESTAV-IPN, México
Septiembre 2003.

Objetivos

- ◆ Definir la Ingeniería de Software y explicar su importancia.
- ◆ Discutir los conceptos de producto de software y proceso de software.
- ◆ Explicar la importancia de la visibilidad de los procesos.
- ◆ Introducir la noción de responsabilidad profesional.

Tópicos

- ◆ Productos de Software.
- ◆ El proceso de Software.
- ◆ El modelo de Espiral de Boehm.
- ◆ La visibilidad de los procesos.
- ◆ Responsabilidad profesional.

Ingeniería de Software

- ◆ Las economías de los países desarrollados dependen en gran parte del software.
- ◆ Mas y más sistemas son actualmente controlados por software.
- ◆ La Ingeniería de Software concierne a teorías, métodos y herramientas para el desarrollo profesional de software.
- ◆ El gasto en La Ingeniería de Software, representa un alto porcentaje del PIB de los países desarrollados.

Ingenieria de Software: Conceptos Basicos

- ◆ Que es la Ingenieria de Software ?
- ◆ Cual es la diferencia entre un programador y un Ingeniero de Software?
- ◆ Cual es la diferencia entre un Ingeniero de Software y un Ingeniero de Sistemas?
- ◆ Cual es la diferencia entre la Ingenieria de Software y la Computacion ?
- ◆ Que es el software ?
- ◆ Que es un proceso de software ?
- ◆ Que es un modelo de software?

Que es la Ingeniería de Software

- ◆ La Ingeniería de Software es una disciplina de la Ingeniería que concierne a todos los aspectos de la producción de software
- ◆ Los Ingenieros de Software adoptan un enfoque sistemático para llevar a cabo su trabajo y utilizan las herramientas y técnicas necesarias para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y recursos disponibles.

Diferencia entre Ingeniería de Software y Computación

- ◆ La computación concierne a la teoría y fundamentos de cualquier sistema de cómputo, sea de hardware o de software.
- ◆ La Ingeniería de software concierne solo al desarrollo de sistemas o productos de software
- ◆ La Ingeniería de Software todavía está lejos de ser una ciencia como lo son la Química, la Ingeniería Civil o la Electrónica.

Ingenieria de Sistemas e Ingenieria de Software

- ◆ La Ingeniería de Sistemas concierne a todos los aspectos del desarrollo de sistemas basados en cómputo, que incluyen hardware, software y el proceso de Ingeniería. La Ingeniería de Software es solo parte de este proceso.

Que es el Software ?

Programas de cómputo y su documentacion asociada

- ◆ Sistemas o Productos de software grandes y complejos.
- ◆ Que contiene el software.
- ◆ Que tipos de software hay ?
- ◆ Cual es el costo del software.
- ◆ Como se desarrolla el software ?
- ◆ Como saber si un software tiene calidad.

Productos de Software

- ◆ Productos genéricos.
 - Productos que son producidos por una organización para ser vendidos al mercado.
- ◆ Productos hechos a medida.
 - Sistemas que son desarrollados bajo pedido a un desarrollador específico.
- ◆ La mayor parte del gasto del software es en productos genéricos, pero hay más esfuerzo en el desarrollo de los sistemas hechos a medida.

Características de los Productos de Software

◆ Mantenibles.

- Debe ser posible que el software evolucione y que siga cumpliendo con sus especificaciones.

◆ Confiabilidad.

- El software no debe causar danos físicos o económicos en el caso de fallos.

◆ Eficiencia.

- El software no debe desperdiciar los recursos del sistema.

◆ Utilización adecuada.

- El software debe contar con una interfaz de usuario adecuada y su documentación.

Importancia de las características del producto

- ◆ La importancia relativa de las características depende en el tipo de producto y en el ambiente en el que será utilizado.
- ◆ En algunos casos, algunos atributos pueden dominar.
 - En sistemas de seguridad críticos de tiempo real, los atributos clave pueden ser la confiabilidad y la eficiencia.
- ◆ Los costos tienden a crecer exponencialmente si son requeridos altos niveles de alguna característica.

Que contiene el software ?

- ◆ El software contiene:
 - Líneas de código de algún lenguaje ?
 - Instrucciones de computadora.
 - Descripción de las estructuras de datos.
 - Algoritmos.
 - Procedimientos y funciones.
 - Componentes de software.

Que tipos de software hay ?

- ◆ Por su estructura:
 - Funcionales.
 - Orientados a objetos.
 - Orientados a listas.
 - Orientados a componentes.
- ◆ Por su funcion:
 - Programas o Sistemas de Usuario
 - Interfaces Hombre-Maquina.
 - Herramientas de Software.
 - Librerias.
 - Sistemas de uso generico: Compiladores, S.O's, Procesadores de Texto, etc.
 - Bases de Datos.
 - Sistemas basados en Web.

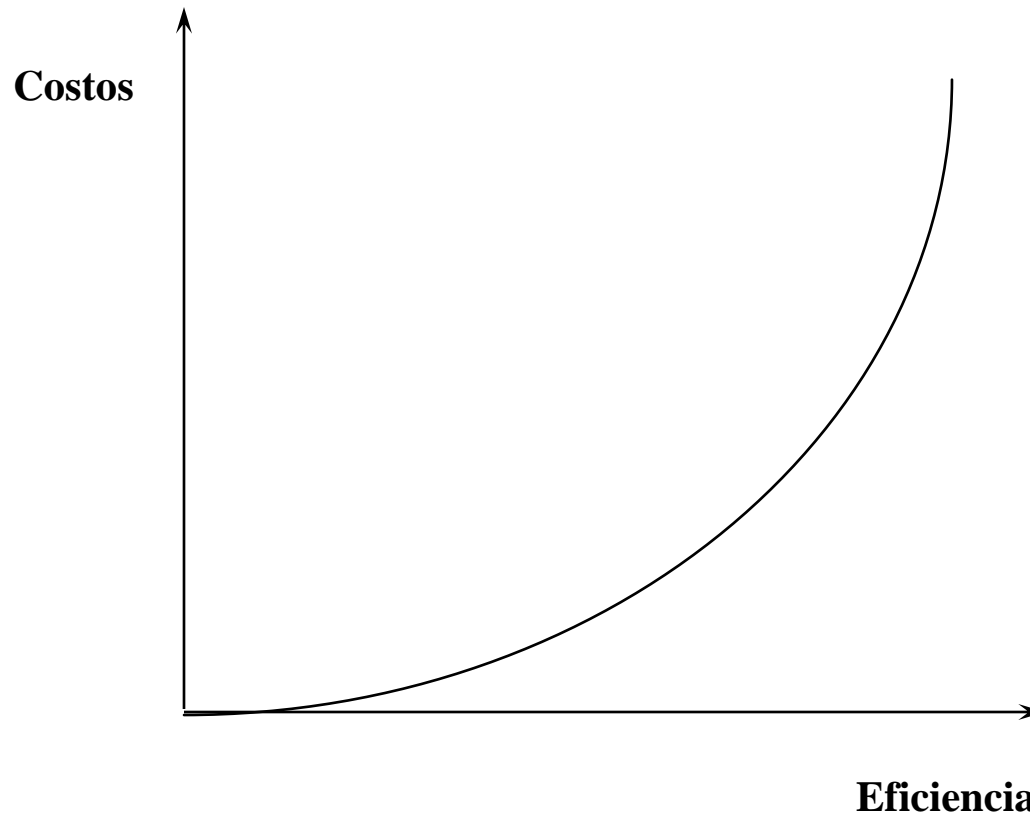
Que tipos de software hay ?

- ◆ Por su plataforma de computo:
 - Sistemas embebidos.
 - Sistemas de computo distribuido.
 - Sistemas de computo paralelo.
 - Sistemas de tiempo real.
 - Sistemas basados en Chips.
 - Wearable computing systems.
 - Sistemas de computo ubiguos.

Costos del Software

- ◆ Los costos del software a menudo dominan al costo del sistema. El costo del software en un PC es a menudo mas caro que la PC.
- ◆ Cuesta mas mantener el software que desarrollarlo. Para sistemas con una larga vida, este costo se multiplica.
- ◆ La Ingeniería de Software concierne a un desarrollo efectivo en cuanto a costes del software.

Costes de Eficiencia.



El Proceso de Software

- ◆ Conjunto estructurado de actividades requeridas para desarrollar un sistema de software.
 - Especificación- que debe hacer el software y cuales son sus especificaciones de desarrollo.
 - Desarrollo – produccion del sistema de software.
 - Validación – verificar que el software hace lo que el cliente pide.
 - Evolución – cambiar/adaptar el software a las demandas.
- ◆ Las actividades varían dependiendo de la organización y del tipo de sistema a desarrollarse.
- ◆ Debe estar explícitamente modelado si va a ser bien administrado.

Proceso Genérico de Software

- ◆ **Especificación** - establecer los requerimientos y restricciones del sistema
- ◆ **Diseño** - Producir un modelo en papel del sistema
- ◆ **Manufactura** - construir el sistema
- ◆ **Prueba** - verificar que el sistema cumpla con las especificaciones requeridas
- ◆ **Instalación** - entregar el sistema al usuario y asegurar su operacionalidad
- ◆ **Mantenimiento** - reparar fallos en el sistema cuando sea descubiertos

Características del proceso

◆ Entendible

- Se encuentra el proceso bien definido y es entendible ?.

◆ Visible

- El proceso es visible al exterior ?.

◆ Soportable

- Puede el proceso ser soportado por herramientas CASE ?.

◆ Aceptable

- El proceso es aceptado por aquellos involucrados en el ?.

Características del proceso

◆ Confiable

- Los errores del proceso son descubiertos antes de que se conviertan en errores del producto ?.

◆ Robusto

- Puede continuar el proceso a pesar de problemas inesperados ?.

◆ Mantenable

- Puede el proceso evolucionar para cumplir con los objetivos organizacionales ?.

◆ Rapidez

- Que tan rápido puede producirse el sistema ?.

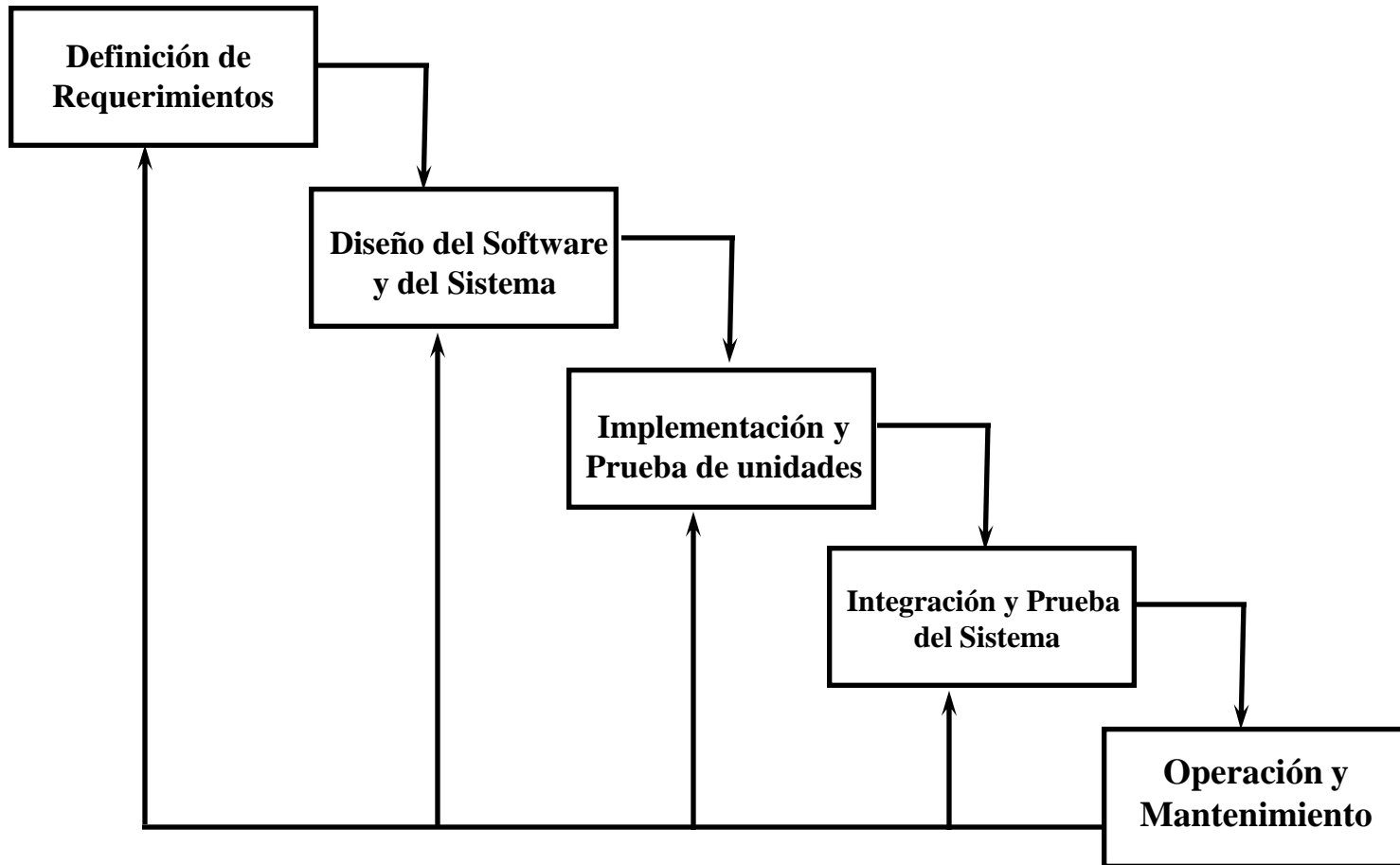
Problemas en el Modelo del Proceso

- ◆ Normalmente, las especificaciones son incompletas o anómalas
- ◆ No existe una distinción precisa entre la especificación, el diseño y la manufactura
- ◆ Solo hasta que el sistema se ha producido se puede probar
- ◆ El software no se puede reemplazar siempre durante el mantenimiento

Modelos de Desarrollo de Software

- ◆ Representación formal o simplificada de proceso de software.
- ◆ Modelos Genéricos:
 - **Modelo de Cascada**
 - ◆ Separar en distintas fases de especificación y desarrollo.
 - **Desarrollo Evolutivo**
 - ◆ La especificación y el desarrollo están intercalados.
 - **Prototipado**
 - ◆ Un modelo sirve de prototipo para la construcción del sistema final.
 - **Transformación Formal**
 - ◆ Un modelo matemático del sistema se transforma formalmente en la implementación.
 - **Desarrollo basado en Reutilización**
 - ◆ El sistema es ensamblado a partir de componentes existentes.

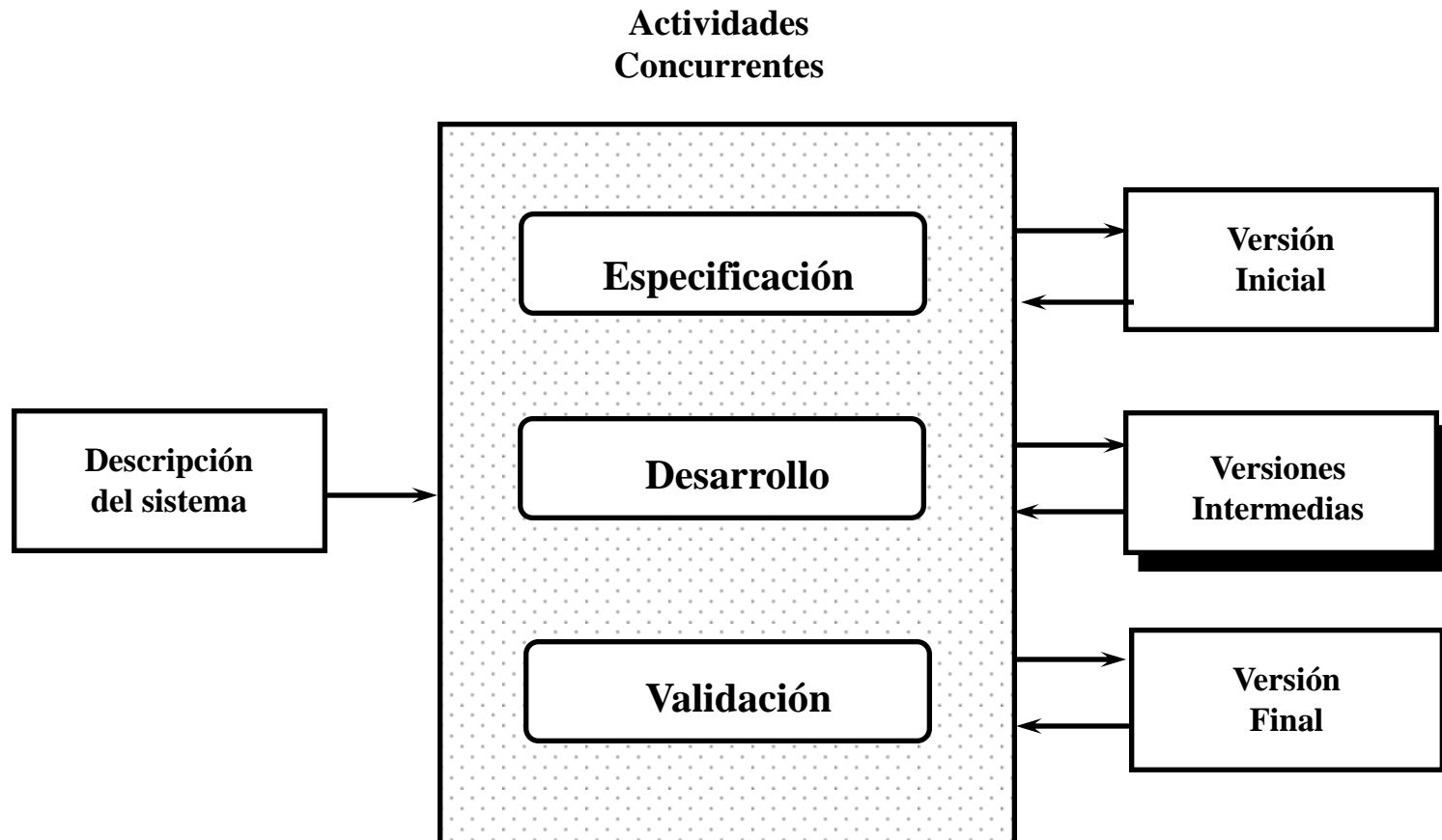
Modelo de Cascada (gráfica)



Fases del Modelo de Cascada

- ◆ Análisis de requerimientos y definición.
- ◆ Diseño del sistema y del software.
- ◆ Implementación y prueba de unidades
- ◆ Integración y prueba del sistema.
- ◆ Operación y mantenimiento.
- ◆ La dificultad en esta modelo reside, en la dificultad de hacer cambios entre etapas.

Desarrollo Evolutivo



Desarrollo Evolutivo

◆ Problemas

- Poca visibilidad en el proceso
- Los sistemas están pobremente especificados
- Se requieren habilidades especiales.

◆ Aplicabilidad

- Para sistemas interactivos pequeños o medianos.
- Para partes de sistemas grandes (p.ej. la interfaz de usuario).
- Para sistemas de corta vida.

Prototipado

◆ Prototipado exploratorio

- El objetivo es trabajar con clientes hasta evolucionar a un sistema final, a partir de una especificación inicial. Se debe comenzar con unas especificaciones bien entendidas.

◆ Prototipado de “throw-away”.

- El objetivo es entender los requerimientos del sistema. Se puede comenzar con especificaciones poco entendidas.

Problemas y Riesgos con los Modelos.

◆ Cascada.

- Alto riesgo en sistemas nuevos debido a problemas en las especificaciones y en el diseño.
- Bajo riesgo para desarrollos bien comprendidos utilizando tecnología conocida.

◆ Prototipado.

- Bajo riesgo para nuevas aplicaciones debido a que las especificaciones y el diseño se llevan a cabo paso a paso.
- Alto riesgo debido a falta de visibilidad

◆ Evolutivo.

- Alto riesgo debido a la necesidad de tecnología avanzada y habilidades del grupo desarrollador.

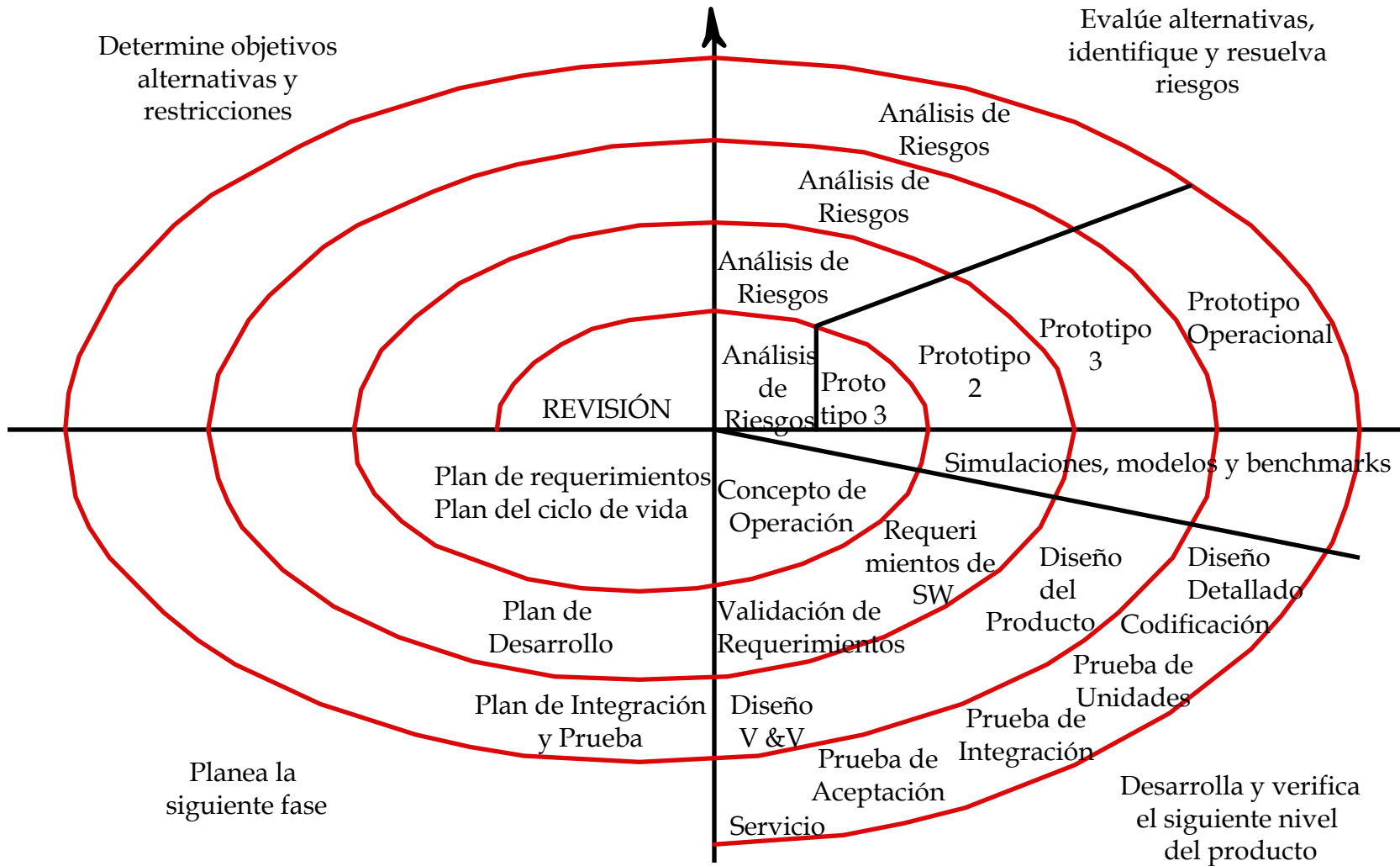
Manejo de Riesgos

- ◆ La tarea principal del administrador consiste en minimizar riesgos.
- ◆ El “riesgo” inherente en una actividad es se mide en base a la incertidumbre que presenta el resultado de esa actividad.
- ◆ Las actividades con alto riesgo causan sobre-costes en cuanto a planeación y costos
- ◆ El riesgo es proporcional al monto de la calidad de la información disponible. Cuanto menos información, mayor el riesgo.

Modelos de Procesos Híbridos

- ◆ Los sistemas grandes están hechos usualmente de varios subsistemas.
- ◆ No es necesario utilizar el mismo modelo de proceso para todos los subsistemas.
- ◆ El prototipado es recomendado cuando existen especificaciones de alto riesgo.
- ◆ El modelo de cascada es utilizado en desarrollos bien comprendidos.

Modelo de Proceso de Espiral



Fases del Modelo de Espiral

- ◆ Planteamiento de Objetivos
 - Se identifican los objetivos específicos para cada fase del proyecto.
- ◆ Identificación y reducción de riesgos.
 - Los riesgos clave se identifican y analizan, y la información sirve para minimizar los riesgos.
- ◆ Desarrollo y Validación.
 - Se elige un modelo apropiado para la siguiente fase del desarrollo.
- ◆ Planeación.
 - Se revisa el proyecto y se trazan planes para la siguiente ronda del espiral.

Plantilla para una ronda del espiral

- ◆ Objetivos.
- ◆ Restricciones.
- ◆ Alternativas.
- ◆ Riesgos.
- ◆ Resolución de riesgos.
- ◆ Resultados.
- ◆ Planes.
- ◆ Garantías (commitments).

Ventajas del Modelo de Espiral

- ◆ Centra su atención en la reutilización de componentes y eliminación de errores en información descubierta en fases iniciales.
- ◆ Los objetivos de calidad son el primer objetivo.
- ◆ Integra desarrollo con mantenimiento.
- ◆ Provee un marco de desarrollo de hardware/software.

Problemas con el Modelo de Espiral

- ◆ El desarrollo contractual especifica el modelo del proceso y los resultados a entregar por adelantado.
- ◆ Requiere de experiencia en la identificación de riesgos.
- ◆ Requiere refinamiento para uso generalizado.

Que modelo utilizar ?

- ◆ Para sistemas bien comprendidos utiliza el Modelo de Cascada. La fase de análisis de riesgos es relativamente fácil.
- ◆ Con requerimientos estables y sistemas de seguridad críticos, utiliza modelos formales.
- ◆ Con especificaciones incompletas, utiliza el modelo de prototipado.
- ◆ Pueden utilizarse modelos híbridos en distintas partes del desarrollo.

Visibilidad de Procesos

- ◆ Los sistemas de software son intangibles por lo que los administradores necesitan documentación para identificar el progreso en el desarrollo.
- ◆ Esto puede causar problemas..
 - El tiempo planeado para entrega de resultados puede no coincidir con el tiempo necesario para completar una actividad.
 - La necesidad de producir documentos restringe la iteración entre procesos.
 - .El tiempo para revisar y aprobar documentos es significativo.
- ◆ El modelo de cascada es aún el modelo basado en resultados mas utilizado.

Documentos del Modelo de Cascada

Actividad	Documentos Producidos
Análisis de Requerimientos	Documento de Requerimientos
Definición de Requerimientos	Documento de Requerimientos.
Especificación del Sistema.	Especificación Funcional, Plan de Pruebas de Aceptación.
Diseño Arquitectural	Especificación de la Arquitectura, y Plan de Pruebas del Sistema
Diseño de Interfaces	Especificación de la Interfaces y Plan de pruebas de Integración.
Diseño Detallado	Especificación del diseño y Plan de prueba de Unidades.
Codificación	Código de Programa
Prueba de Unidades	Reporte de prueba de unidades
Prueba de Módulos	Reporte de prueba de módulos
Prueba de Integración	Reporte de prueba de integración y Manual de usuario final
Prueba del Sistema	Reporte de prueba del sistema
Prueba de Aceptación	Sistema final mas la documentación.

Visibilidad del Modelo

Modelo de Proceso	Visibilidad del Proceso
Modelo de Cascada	Buena visibilidad, cada actividad produce un documento o resultado
Desarrollo Evolutivo	Visibilidad pobre, muy caro al producir documentos en cada iteración.
Modelos Formales	Buena visibilidad, en cada fase deben producirse documentos.
Desarrollo orientado a la reutilización	Visibilidad moderada. Importante contar con documentación de componentes reutilizables.
Modelo de Espiral	Buena visibilidad, cada segmento y cada anillo del espiral debe producir un documento.

Retos de la Ingeniería de Software

- ◆ Mantener y tratar con sistemas legados. Tratar con una mayor diversidad de sistemas con mayores demandas de computo, y menores tiempos de entrega
- ◆ **Sistemas Legados**
 - Sistemas antiguos que deben ser mantenidos y mejorados.
- ◆ **Heterogeneidad**
 - Sistemas que incluyen una mezcla de software y hardware.
- ◆ **Entrega**
 - Existe una presión incremental por una entrega a tiempo de los productos de software.
- ◆ **Formalidad.**
 - Existe una gran demanda de que exista formalidad en el proceso de desarrollo de software

Retos de la Ingeniería de Software

- ◆ Por que no podemos desarrollar sistemas de software con tecnicas formales como lo hacen los Ingenieros en Electronica, los Ing. Quimicos o los Ingenieros Civiles. ?

Demanda de Ingenieros de Software

En las últimas décadas, la tecnología de la información creció dramáticamente, con un significativo impacto en la economía global y en la vida diaria. La capacidad informática se ha incrementado rápidamente mientras que los costos de hardware han bajado, provocando este incremento que el uso de implementación de sistemas (Software) sean preferidos al hardware. Dispositivos mecánicos en automóviles, aviones, y poderosas plantas se han reemplazados por componentes de software, porque el software es más adaptable, da más funcionalidad, se actualiza fácilmente, para cambios en futuras necesidades.

Demanda de Ingenieros de Software

El Software es usado en dispositivos médicos, sistemas del transporte y sistemas financieros para automatizar tareas repetitivas pero críticas. Científicos e investigadores de negocios usan software para examinar cuidadosamente su data warehouse e identificar hechos, tendencias, banca, seguros, telecomunicaciones y otras industrias de servicio que usan software para automatizar y personalizar los servicios que ofrecen a sus clientes. El software de aplicación tiene un crecimiento más complejo, y surge la demanda de ingenieros de software quienes tienen el conocimiento y la experiencia para desarrollar sistemas de alta calidad.

Responsabilidad profesional

- ◆ Los Ingenieros de software no solo deben considerar aspectos técnicos. Deben tener una visión mas amplia, en lo ético, social y profesional.
- ◆ No existe estatutos para ninguno de estos aspectos.
 - Desarrollo de sistemas militares.
 - Piratería.
 - Que es mejor para la profesión de Ingeniero de Software.

Aspectos Éticos

- ◆ Confidencialidad.
- ◆ Competencia.
- ◆ Derechos de propiedad intelectual.
- ◆ Mal uso de la computadora.

Una Profesion Madura de la Ingenieria de Software

Consultar:

A Mature Profession of Software Engineering.

Gary Ford

Norman Gibbs

Technical Report: CMU/SEI-96-TR-004

www.sei.cmu.edu

Resumen

- ◆ La Ingeniería de software concierne a las teorías, métodos y herramientas para el desarrollo, administración y evolución de productos de software.
- ◆ Los productos de software consisten de programas y documentación. Los atributos de los productos son, mantenabilidad, dependabilidad, eficiencia y usabilidad.
- ◆ El proceso de software consiste en aquellas actividades involucradas en el desarrollo de software.

Resumen

- ◆ El modelo de cascada considera cada actividad del proceso como una actividad discreta.
- ◆ El modelo de desarrollo evolutivo considera actividades del proceso en forma concurrente.
- ◆ El modelo de espiral se basa en análisis de riesgos.
- ◆ La visibilidad del proceso involucra la creación de documentos o resultados de las actividades.
- ◆ Los Ingenieros de software deben tener responsabilidades éticas, sociales y profesionales.