



# INGENIERÍA DE SOFTWARE III

UNIDAD I – INTRODUCCIÓN A LA INGENIERÍA DE  
SOFTWARE

Prof. MS. Luis Gilberto Salinas – Prof. Ing. Julia Talavera

## OBJETIVOS

- ❑ Proporcionar una visión general de los productos, procesos y modelos de desarrollo contemplados en la ingeniería del software

## CONTENIDO

- ☐ La Ingeniería de Software.
- ☐ Productos Software.
- ☐ Costos del Software.
- ☐ Proceso de Software.
- ☐ Modelos de Desarrollo de Software.
- ☐ Retos de la Ingeniería de Software.
- ☐ Demanda de Ingenieros de Software

## ¿Qué es la Ingeniería de Software?

- ❑ Establecimiento y uso de principios con caracteres de ingeniería apropiados para obtener, eficientemente, software confiable, que opere eficaz y eficientemente en máquinas reales
- ❑ Establecimiento y uso de métodos, herramientas y procedimientos, orientados a obtener software económico que sea fiable y funcione de mane Establecimiento y uso de principios con caracteres de ingeniería apropiados para obtener, eficientemente, software confiable, que opere eficaz y eficientemente en máquinas reales
- ❑ Este concepto se acuñó en 1968, en Conferencia de la OTAN en Alemania, con la intención de que mediante el uso de filosofías y paradigmas de disciplinas ingenieriles establecidas se resolviera la denominada crisis del software.
- ❑ Establecimiento y uso de métodos, herramientas y procedimientos, orientados a obtener software económico que sea fiable y funcione de manera eficiente.
- ❑ 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.

## Ingeniería de Software y Ciencias de la Computación - Diferencias

- ☐ Las ciencias de la computación conciernen a la teoría y fundamentos de cualquier sistema de cómputo, sea de hardware o de software.
- ☐ La Ingeniería de software concierne sólo al desarrollo de sistemas o productos de software.
- ☐ Podemos decir que la Ingeniería de Software es una nueva disciplina, con un nivel de madurez aún bajo, comparándola con la Ingeniería Civil, Eléctrica o Electrónica.

## Ingeniería de Software e Ingeniería de Sistemas - Diferencias

- ❑ La Ingeniería de Sistemas tiene en cuenta a todos los elementos del desarrollo de sistemas basados en cómputo, que incluyen hardware, software, bases de datos, recursos humanos. La Ingeniería del Software forma parte de este proceso.

## ¿Qué es el Software?

Sommerville:

“Programas de cómputo y su documentación asociada”

Pressman:

“Elemento del sistema de información que es lógico, con características considerablemente distintas a las del hardware”

## Productos de Software

- ❑ **Productos genéricos:** Productos que son producidos por una organización para ser comercializados.
- ❑ **Productos hechos a medida:** Sistemas que son desarrollados bajo pedido a un desarrollador específico.

La mayor parte del gasto del software se realiza en productos genéricos, pero existe un mayor esfuerzo en el desarrollo de los sistemas hechos a medida.



## Características de Productos Software

- ☐ Mantenibles.

El software debe evolucionar y seguir cumpliendo con sus especificaciones.

- ☐ Confiabilidad.

El software no debe causar daños 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 y documentación adecuada.

## Importancia de las Características

- ❑ 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 predominar. 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.

## Clasificación de Productos Software

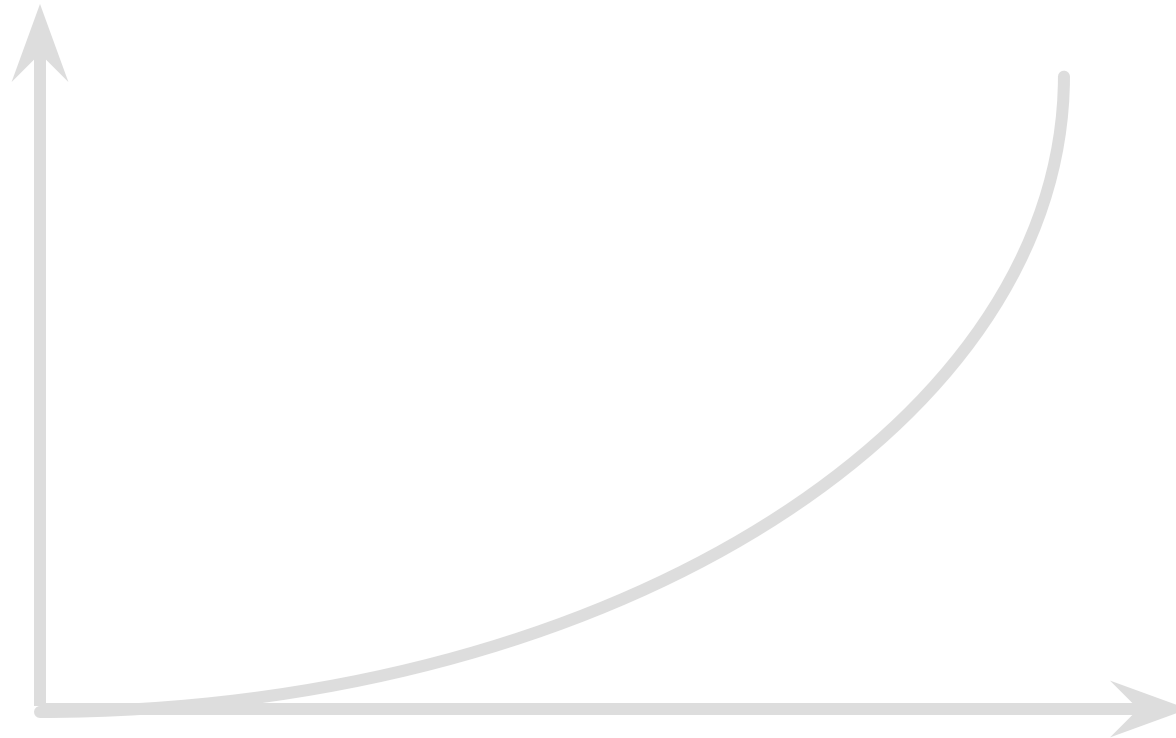
- ☐ Por su estructura:
  - ✓ Funcionales.
  - ✓ Orientados a objetos.
  - ✓ Orientados a listas.
  - ✓ Orientados a componentes.
- ☐ Por su función:
  - ✓ Programas o Sistemas de Usuario
  - ✓ Herramientas de Software.
  - ✓ Librerías.
  - ✓ Sistemas de uso genérico: Compiladores, S.O's, Procesadores de Texto, etc.
  - ✓ Bases de Datos.
  - ✓ Sistemas basados en Web.
- ☐ Por su plataforma de cómputo:
  - ✓ Sistemas embebidos o empotrados.
  - ✓ Sistemas de cómputo distribuido
  - ✓ Sistemas de tiempo real.
  - ✓ Sistemas basados en Chips.
  - ✓ Sistemas de cómputo ubicuos.

## Costos del Software

- ❑ Los costos del software a menudo dominan al costo del sistema. El costo del software en una PC es a menudo mas caro que la PC.
- ❑ Generalmente, 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 costos del software.

## Costos de Eficiencia

**Costos**



**Eficiencia**

## Proceso de Software

- ❑ Conjunto estructurado de actividades requeridas para desarrollar un sistema de software.
  - ✓ Especificación – que debe hacer el software y cuáles son sus especificaciones de desarrollo.
  - ✓ Desarrollo – producción 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

- ☐ Especificación - establecer los requerimientos y restricciones del sistema.
- ☐ Diseño - Producir un modelo del sistema.
- ☐ Codificación - 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 ?

- ☐ Confiable

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

Qué tan rápido puede producirse el sistema ?



## Problemas en el Proceso

Normalmente...

- ☐ Las especificaciones son incompletas o anómalas.
- ☐ No existe una distinción precisa entre la especificación, el diseño y la codificación.
- ☐ Solo hasta que el sistema se ha producido se puede probar.
- ☐ El software no siempre se puede remplazar 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.

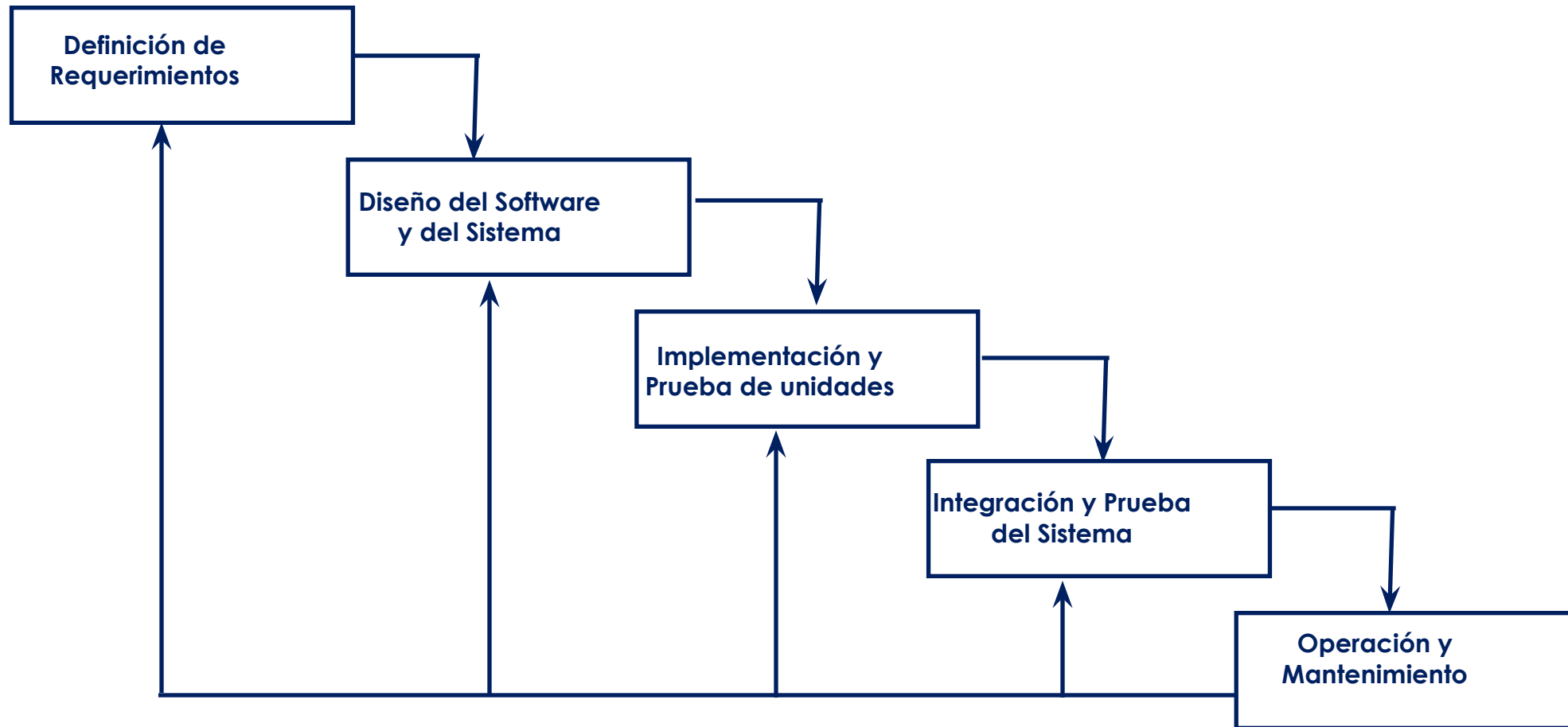
- ☐ Transformación Formal

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 en Cascada



## Las Fases

- ☐ 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 este modelo reside, en la dificultad de hacer cambios entre etapas.

## 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.

## Desarrollo Evolutivo



## Problemas y Aplicaciones

### ☐ Problemas

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

### ☐ Aplicabilidad

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



## Prototipado

- ☐ Prototipado evolutivo

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 desechable

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.

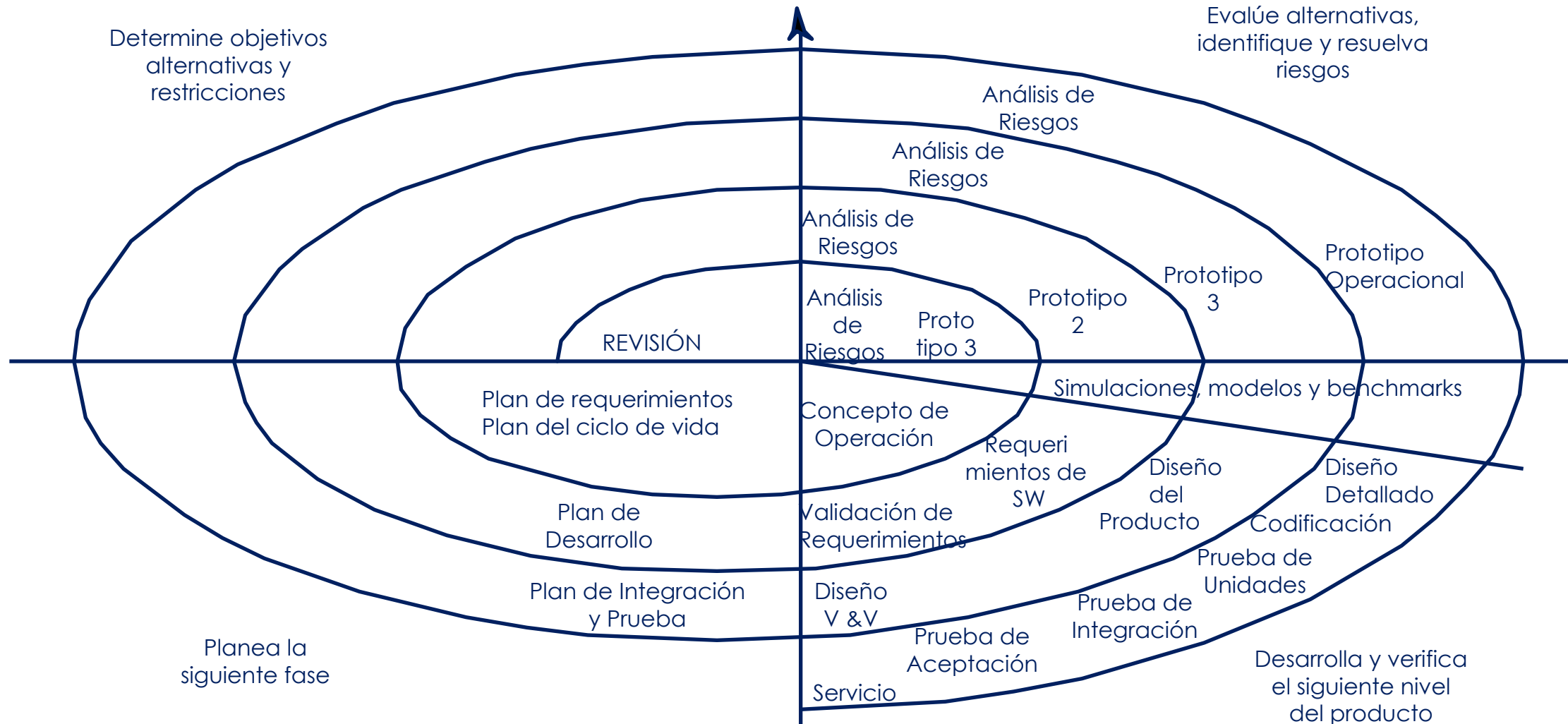
### ☐ Evolutivo

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

## Modelo de Procesos Híbridos

- ☐ Los sistemas grandes están compuestos 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 Procesos en Espiral



## Fases del Modelo en 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

## Ventajas del Modelo 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 del Modelo 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.



## ¿Qué modelo utilizar?

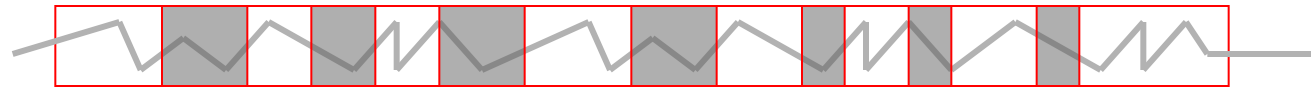
- ☐ Para sistemas bien comprendidos se utiliza el Modelo de Cascada. La fase de análisis de riesgos es relativamente fácil.
- ☐ Con requerimientos estables y sistemas de seguridad críticos, se utiliza modelos formales.
- ☐ Con especificaciones incompletas, se utiliza el modelo de prototipado.
- ☐ Pueden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

## 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.

## Visibilidad de Procesos

Visibilidad de  
un proyecto ideal



Visibilidad de  
un proyecto típico

Progreso del  
proyecto

La visibilidad del progreso cambia según la clase de proyecto.  
El desarrollo eficiente ofrece mas visibilidad que el desarrollo típico

## Visibilidad de Procesos

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 distintas plataformas 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

## Demanda de Ingenieros de Software

- ❑ En las últimas décadas, la tecnología de información ha crecido dramáticamente, con un importante impacto tanto en la economía global como en la vida diaria.
- ❑ El poder de cómputo se ha incrementado rápidamente mientras que el costo del hardware y las comunicaciones ha caído, haciendo mas económica la implementación de sistemas de software.
- ❑ Los dispositivos mecánicos en automóviles, aviones, y plantas de energía están reemplazándose por componentes de software porque el software es mas adaptable, puede proveer mayor funcionalidad, y puede ser actualizado mas fácilmente para satisfacer necesidades futuras.
- ❑ El software es utilizado en dispositivos médicos, sistemas de transporte y sistemas financieros para automatizar tareas críticas y repetitivas. Los científicos e investigadores de negocios utilizan software para filtrar información en almacenes de datos y para identificar hechos importantes y tendencias. Los bancos, compañías de seguros, telecomunicaciones y otras industrias de servicios utilizan software para automatizar y personalizar los servicios que ofrecen a sus clientes.
- ❑ Como las aplicaciones de software se han tornado mas complejas, ha surgido una demanda de ingenieros de software que tengan el conocimiento y la experiencia para desarrollar sistemas de software 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 existen estatutos para ninguno de estos aspectos:
  - ✓ Desarrollo de sistemas militares.
  - ✓ Piratería.

## Aspectos Éticos

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



## Bibliografía Consultada

- ❑ SOMMERVILLE, Ian. Ingeniería de Software 6 y 7º Edición. Addison Wesley. 2002-2005
- ❑ PRESSMAN, Roger 5º y 6º,7º Edición. Ingeniería del Software. Un enfoque práctico. McGraw-Hill. 2002-2005-2010
- ❑ McCONNELL, Steve. Desarrollo y gestión de proyectos informáticos. McGraw-Hill. 1997
- ❑ PIATTINI, Mario. Calidad de Sistemas Informáticos. Alfaomega - Ra-Ma. 2007



MUCHAS GRACIAS