

# INGENIERÍA DE APLICACIONES

---

## Modelos de Procesos de Software

Dra. María Luján Ganuza

mlg@cs.uns.edu.ar

DCIC - Depto. de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur, Bahía Blanca

2019



# Temario

## Modelos de Procesos de Software:

- El Modelo de Cascada
- Modelo de Desarrollo Incremental
- Ingeniería de software orientada a la reutilización

# Procesos de Software

- Un **proceso de software** es un conjunto de actividades relacionadas que conduce a la producción de un producto de software.
- Todos los procesos de software incluyen 4 **actividades** fundamentales para la ingeniería de software:
  1. Especificación
  2. Diseño e Implementación
  3. Validación
  4. Evolución

# Procesos de Software

- Las descripciones de los **procesos de software** incluyen:
  1. Las **actividades** que se llevan a cabo en estos procesos.
  2. Los **productos**, resultados de una actividad de proceso. (Por ejemplo, el resultado de la actividad de diseño arquitectónico puede ser una arquitectura de modelo del software).
  3. Los **roles**, que reflejan las responsabilidades de las personas involucradas en el proceso (administrador de proyecto, programador, etc.)
  4. Las **pre y post condiciones** de las actividades. (Pre condición para una actividad de diseño: que los requisitos estén completos. Post condición: que los modelos UML que describen la arquitectura hayan sido revisados)

# Modelos de Procesos de Software

- Un **modelo de proceso** de software es una representación simplificada de un proceso de software.
- Cada modelo de proceso representa un proceso desde una perspectiva particular, proporcionando solo información parcial sobre ese proceso.
- Por ejemplo, un **modelo de actividad de proceso** muestra las actividades y su secuencia pero puede no mostrar los roles de las personas involucradas en estas actividades.

# Modelos de Procesos de Software

- Modelos de Procesos de Software:
  - El modelo de Cascada.
  - Modelo de Desarrollo Incremental.
  - Ingeniería de software orientada a la reutilización.

# Modelos de Procesos de Software

- Modelos de Procesos de Software:
  - **El modelo de Cascada.** Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y los representa como separados fases de proceso tales como especificación de requisitos, diseño de software, implementación, prueba.
  - Modelo de Desarrollo Incremental.
  - Ingeniería de software orientada a la reutilización.

# Modelos de Procesos de Software

- Modelos de Procesos de Software:
  - El modelo de Cascada.
  - **Modelo de Desarrollo Incremental.** Intercala las actividades de especificación, desarrollo y validación El sistema se desarrolla como una serie de versiones (incrementos), con cada versión agregando funcionalidad a la versión anterior.
  - Ingeniería de software orientada a la reutilización.

# Modelos de Procesos de Software

- Modelos de Procesos de Software:
  - El modelo de Cascada.
  - Modelo de Desarrollo Incremental.
  - **Ingeniería de software orientada a la reutilización.** se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se centra en la integración de estos componentes en un sistema en lugar de desarrollar ellos desde cero.

# Modelos de Procesos de Software

## El Modelo de Cascada

- Fue el primer modelo publicado del proceso de desarrollo de software.
- Debido a la cascada de una fase a otra, este modelo se conoce como el 'modelo de cascada' o **ciclo de vida del software**.
- Es un ejemplo de un proceso impulsado por un plan: en principio, debe planificar y programar todas las actividades del proceso antes de comenzar a trabajar en ellas.

# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

1. Análisis y definición de los requisitos.
2. Diseño de sistema y software.
3. Implementación y pruebas unitarias.
4. Integración y prueba del sistema.
5. Operación y mantenimiento .

# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

- 1. Análisis y definición de los requisitos.**  
Los servicios del sistema, las limitaciones y los objetivos se establecen mediante consultas con los usuarios del sistema. Estos son luego definidos en detalle y servir como una especificación del sistema.
2. Diseño de sistema y software.
3. Implementación y pruebas unitarias.
4. Integración y prueba del sistema.
5. Operación y mantenimiento .

# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

1. Análisis y definición de los requisitos.
2. **Diseño de sistema y software.**  
El proceso de diseño de sistemas asigna los requisitos de hardware y/o software estableciendo. El diseño del software implica identificar y describir las abstracciones fundamentales del sistema de software y sus relaciones.
3. Implementación y pruebas unitarias.
4. Integración y prueba del sistema.
5. Operación y mantenimiento.

# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

1. Análisis y definición de los requisitos.
2. Diseño de sistema y software.
3. **Implementación y pruebas unitarias.**  
Se implementa el diseño del software como un conjunto de programas o unidades de programa. Las pruebas unitarias implican verificar que cada unidad cumple con sus especificaciones.
4. Integración y prueba del sistema.
5. Operación y mantenimiento.

# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

1. Análisis y definición de los requisitos.
2. Diseño de sistema y software.
3. Implementación y pruebas unitarias.
4. **Integración y prueba del sistema.**  
Las unidades de programas individuales o programas están integrados y probados como un sistema completo para garantizar los requisitos se han cumplido. Después de la prueba, el sistema de software se entrega a el cliente.
5. Operación y mantenimiento.

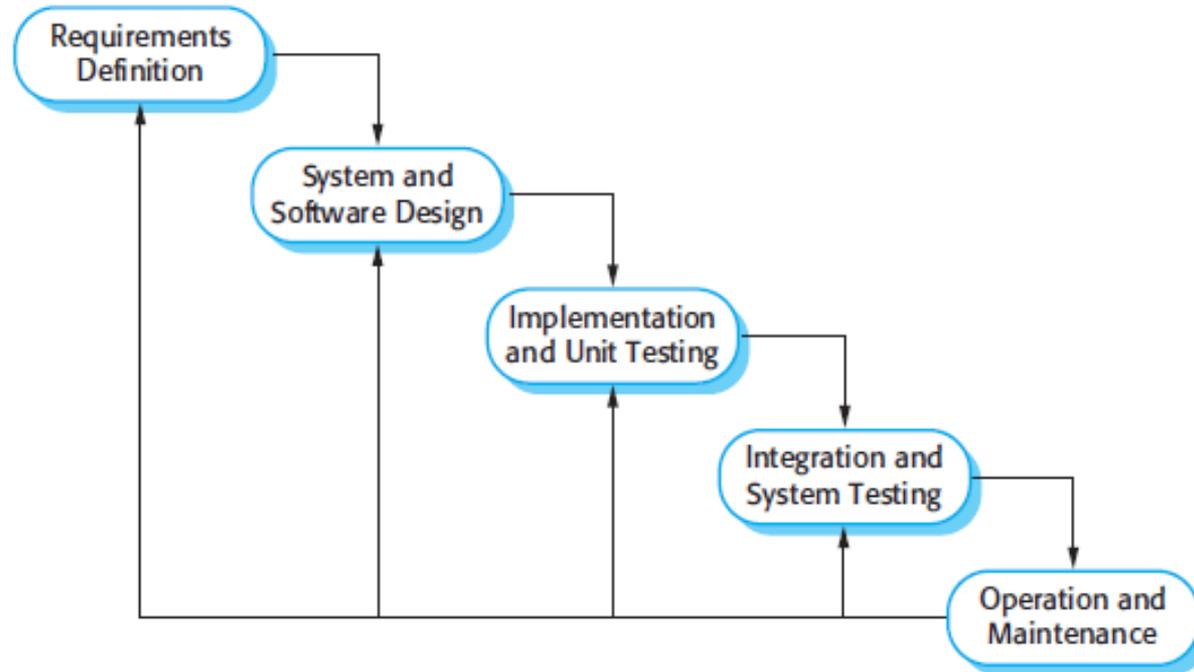
# Modelos de Procesos de Software

## **Etapas Principales** del Modelo de Cascada:

1. Análisis y definición de los requisitos.
2. Diseño de sistema y software.
3. Implementación y pruebas unitarias.
4. Integración y prueba del sistema.
5. **Operación y mantenimiento.**  
El sistema está instalado y en uso. El mantenimiento implica la corrección de errores que no se descubrieron en etapas anteriores y la mejora de la implementación de las y de los servicios del sistema a medida que se descubren nuevos requisitos.

# Modelos de Procesos de Software

## El Modelo de Cascada



# Modelos de Procesos de Software

## Modelo de Desarrollo Incremental

- El **desarrollo incremental** se basa en la idea de desarrollar una implementación inicial, exponiendo esto al comentario del usuario y evolucionándolo a través de varias versiones hasta que se ha desarrollado un sistema adecuado.
- Las actividades de especificación, desarrollo y validación están **intercaladas** en lugar de separadas.

# Modelos de Procesos de Software

## Modelo de Desarrollo Incremental

- El desarrollo incremental refleja la forma en que resolvemos los problemas.
- Este tipo de desarrollo, es más barato y más **propenso a cambios** en el software **a medida que se desarrolla**.
- Los primeros incrementos del sistema incluyen la funcionalidad más importante o requerida con más urgencia. Cada incremento o versión del sistema incorpora parte de la funcionalidad.

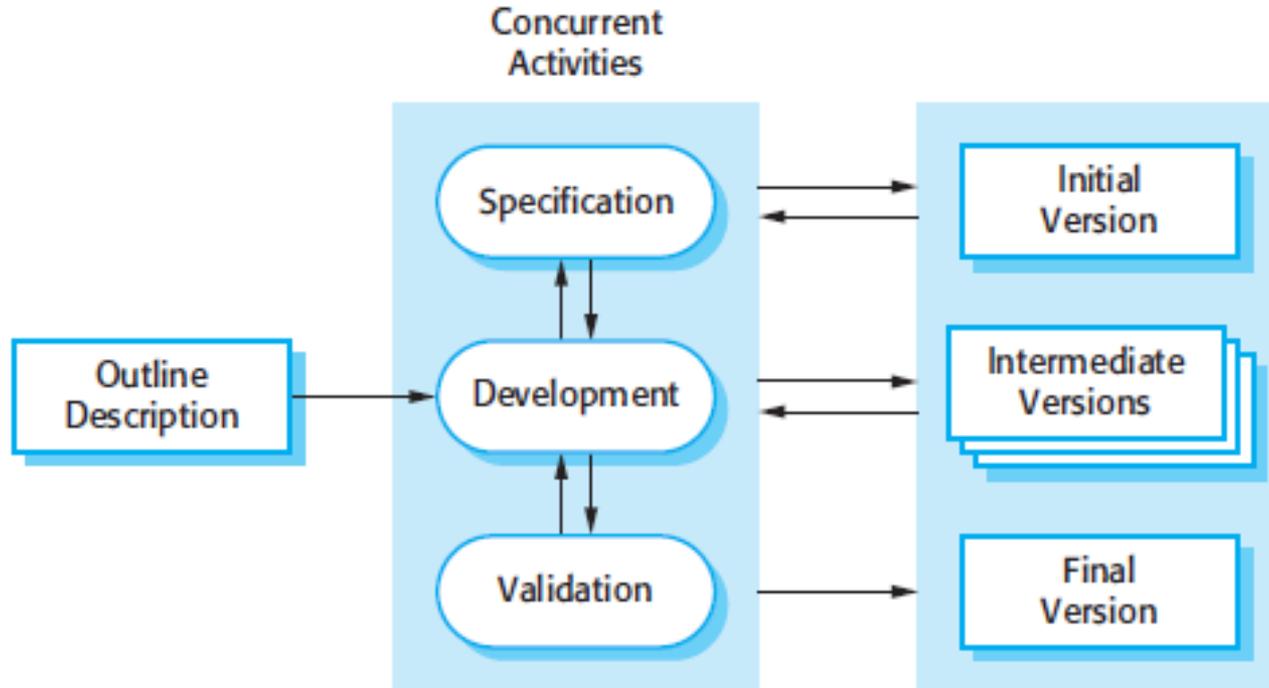
# Modelos de Procesos de Software

## Modelo de Desarrollo Incremental

- El desarrollo incremental refleja la forma en que resolvemos los problemas.
- Este tipo de desarrollo, es más barato y más **propenso a cambios** en el software **a medida que se desarrolla**.
- Los primeros incrementos del sistema incluyen la funcionalidad más importante o requerida con más urgencia. Cada incremento o versión del sistema incorpora parte de la funcionalidad.

# Modelos de Procesos de Software

## Modelo de Desarrollo Incremental



# Modelos de Procesos de Software

**Ventajas del** Modelo de Desarrollo Incremental con respecto al Modelo en Cascada:

- 1. Se reduce el costo de adaptación a los requisitos cambiantes del cliente.** La cantidad de análisis y documentación que debe rehacerse es mucho menor de lo que es requerido con el modelo de cascada.
- 2. Feedback de los clientes.** Los clientes pueden hacer devoluciones de demostraciones parciales del software y ver cuánto se ha avanzado.
- 3. Es posible una entrega más rápida de software útil para el cliente,** incluso si toda la funcionalidad no se ha incluido.

# Modelos de Procesos de Software

Desde una perspectiva de gestión, el enfoque incremental tiene dos **problemas**:

- **El proceso no es visible.** Los gerentes necesitan entregables regulares para medir Progreso. Si los sistemas se desarrollan rápidamente, no es rentable producir documentos que reflejan cada versión del sistema.
- **La estructura del sistema tiende a degradarse a medida que se agregan nuevos incrementos.** A menos que el tiempo y se gasta dinero en refactorizar para mejorar el software, el cambio regular tiende a corrompe su estructura. La incorporación de más cambios de software se vuelve cada vez más difícil y costoso

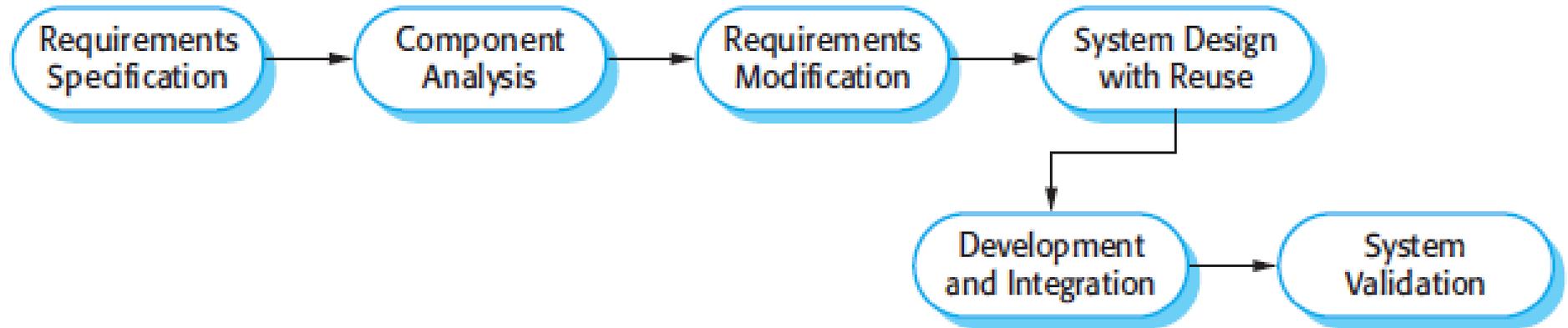
# Modelos de Procesos de Software

## Ingeniería de software orientada a la reutilización

- Los enfoques orientados a la reutilización se basan en una gran base de componentes de **software reutilizables** y en un marco integrador para la composición de estos componentes.
- A veces, estos componentes son sistemas en sí mismos (sistemas comerciales disponibles) que pueden proporcionar una funcionalidad específica.

# Modelos de Procesos de Software

## Ingeniería de software orientada a la reutilización



# Modelos de Procesos de Software

## Etapas específicas en el modelo de Ingeniería de software orientada a la reutilización

1. **Análisis de componentes** Dada la especificación de requisitos, se realiza una búsqueda de componentes para implementar esa especificación. Por lo general, no hay una coincidencia exacta y los componentes que pueden usarse solo brindan algunas de las funcionalidades requeridas.
2. Modificación de requisitos
3. Diseño del sistema con reutilización.
4. Desarrollo e integración

# Modelos de Procesos de Software

## Etapas específicas en el modelo de Ingeniería de software orientada a la reutilización

1. Análisis de componentes
2. **Modificación de requisitos** Se analizan los requisitos utilizando información sobre los componentes que se han descubierto. Si es posible, se modifican para reflejar los componentes disponibles. Si no es posible, se vuelve a la etapa anterior para buscar nuevas alternativas.
3. Diseño del sistema con reutilización
4. Desarrollo e integración

# Modelos de Procesos de Software

## Etapas específicas en el modelo de Ingeniería de software orientada a la reutilización

1. Análisis de componentes.
2. Modificación de requisitos.
3. **Diseño del sistema con reutilización.** Se diseña un nuevo framework del sistema o se reutiliza uno existente. Los diseñadores toman en cuenta el componentes que se reutilizan
4. Desarrollo e integración.

# Modelos de Procesos de Software

## Etapas específicas en el modelo de Ingeniería de software orientada a la reutilización

1. Análisis de componentes.
2. Modificación de requisitos.
3. Diseño del sistema con reutilización.
4. **Desarrollo e integración** El software que no puede obtenerse externamente es desarrollado, y los componentes reutilizados se integran al sistema.

# Modelos de Procesos de Software

## **Modelo de Ingeniería de software orientada a la reutilización**

- La ingeniería de software orientada a la reutilización tiene la ventaja obvia de reducir la cantidad de software a desarrollar y, por lo tanto, reducir costos y riesgos.
- Por lo general, también lleva a una entrega más rápida del software.
- Sin embargo, los compromisos de requisitos son inevitables y esto puede conducir a un sistema que no satisface las necesidades reales de los usuarios.
- Además, se pierde cierto control sobre la evolución del sistema a medida que las nuevas versiones de los componentes reutilizables no están bajo el control de la organización que los usa.

# Material Bibliográfico

- Ian Sommerville. 2010. *Software Engineering* (9th ed.). Addison-Wesley Publishing Company, USA.
- Cadle, J., & Yeates, D. (Eds.). 2004. *Project management for information systems*. Pearson education.
- Epstein, D., & Maltzman, R. 2013. *Project workflow management: a business process approach*. J. Ross Publishing.

# Recordemos que...

- Fecha límite de conformación de comisiones y definición de ideas del proyecto:

**Jueves 29/08**

- Presentación de la idea del proyecto:

**Martes 03/09**