

Ingeniería del Software I

2º

Tema 7 Evolución del software



- **Cambio del software es inevitable**
 - **Nuevos requerimientos** surgen cuando se utiliza el software.
 - **El entorno empresarial cambia.**
 - Los **errores** deben ser reparados.
 - Nuevas tecnologías y **sistemas** se agregan al sistema.
 - Siempre hay que mejorar el rendimiento y la confiabilidad
- Un problema clave para todas las organizaciones es **gestionar el cambio** en los sistemas de software existentes.



La importancia de la evolución

- Las organizaciones hacen grandes inversiones en **SUS sistemas de software - son activos críticos de negocio.**
- Para mantener el valor de estos activos, es necesario actualizar los sistemas.
- La mayor parte del **presupuesto** de software en las grandes empresas **se dedica al cambio y evolución de software existentes** en lugar de desarrollar nuevo software.

Modelo espiral del desarrollo y la evolución

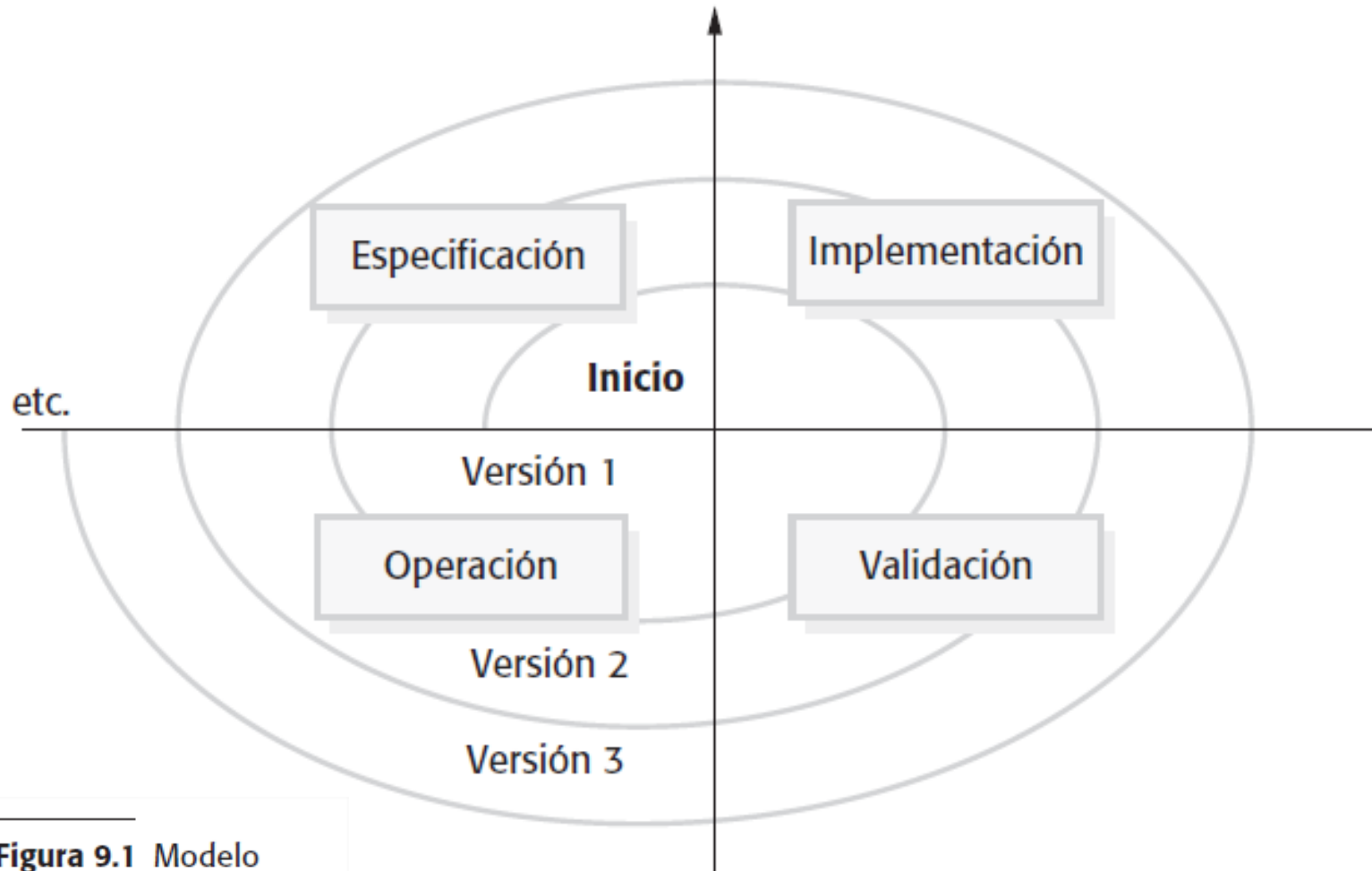
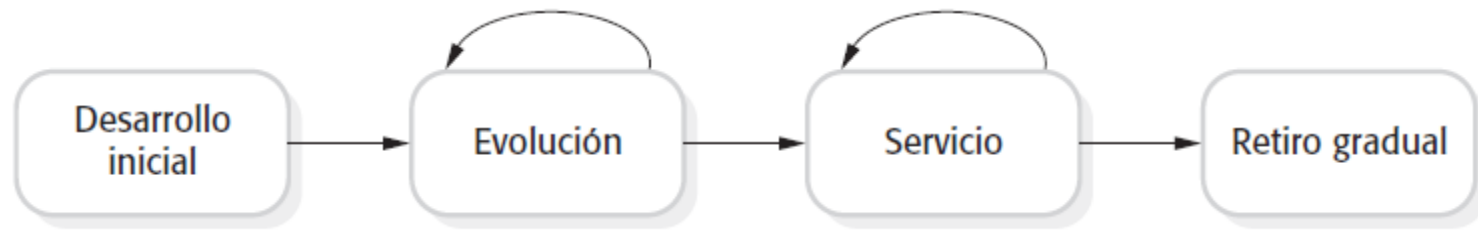


Figura 9.1 Modelo en espiral de desarrollo y evolución

Evolución y mantenimiento

Figura 9.2 Evolución
y servicio



- **Evolución**

- Es la etapa en el ciclo de vida del software en la cual el sistema está **en uso** y **evoluciona** cada vez que **nuevos requerimientos** se proponen y se implementan en el sistema.

- **Prestación de servicios**

- En esta etapa, el software **sigue siendo útil**, pero los únicos cambios necesarios para el mantenimiento **son correcciones** y evoluciones de la operación. **No se agrega ninguna funcionalidad nueva.**

- **Eliminación**

- El software es **operativo**, pero **no se realizan más cambios.**



- **Los procesos de evolución** del software dependen de:
 - El **tipo** de software que se está manteniendo.
 - Los **procesos** de desarrollo utilizados.
 - Las **habilidades** y la experiencia de las personas involucradas.
- Las **propuestas de cambio** son el motor de la **evolución** del sistema.
 - Las propuestas deben detallar los componentes relacionados con el cambio, permitiendo así estimar su coste e impacto.
- La **gestión del cambio (identificación del cambio)** y **evolución continua** se da durante **la vida completa del sistema**.



Identificación de cambios y proceso de evolución

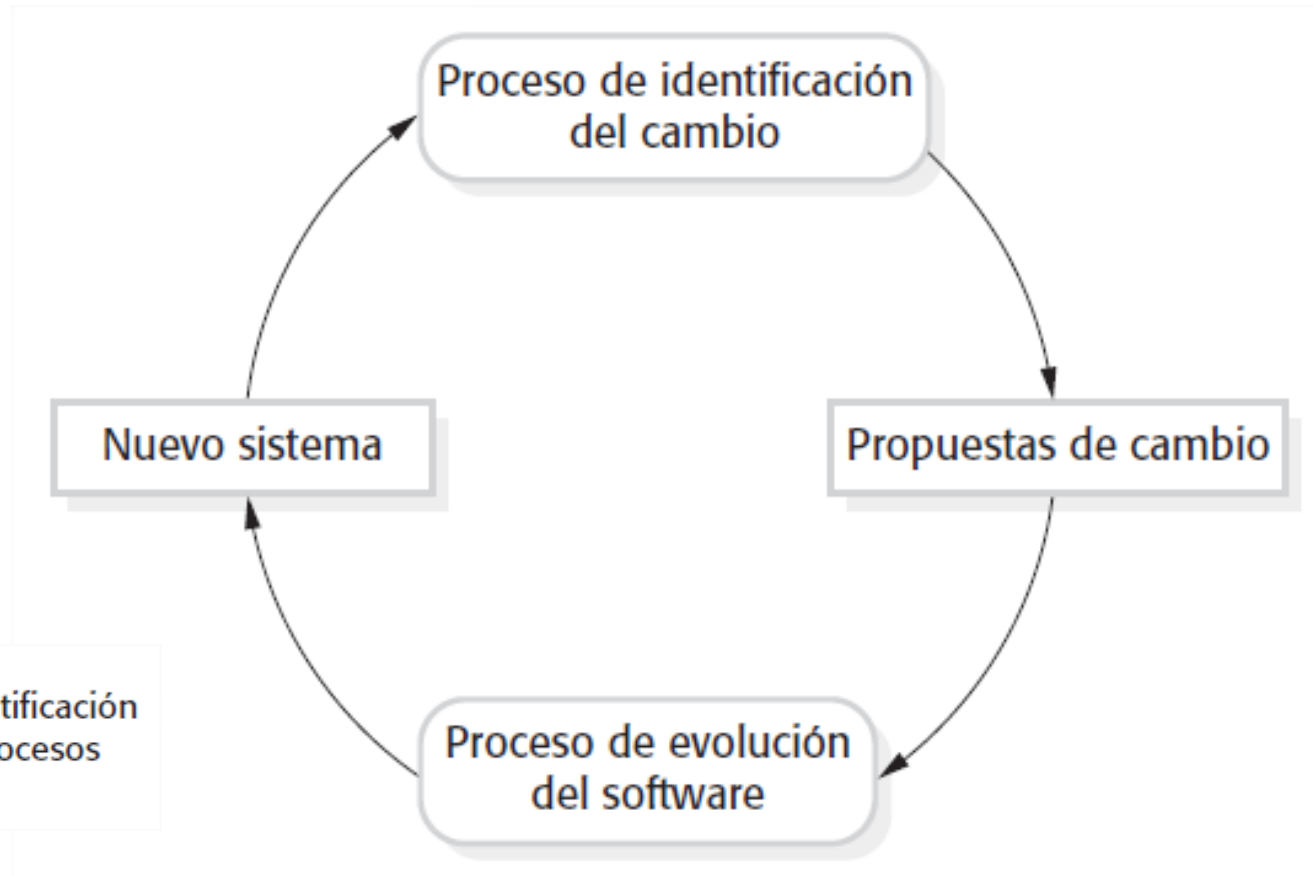


Figura 9.3 Identificación del cambio y procesos de evolución

El proceso de evolución del software

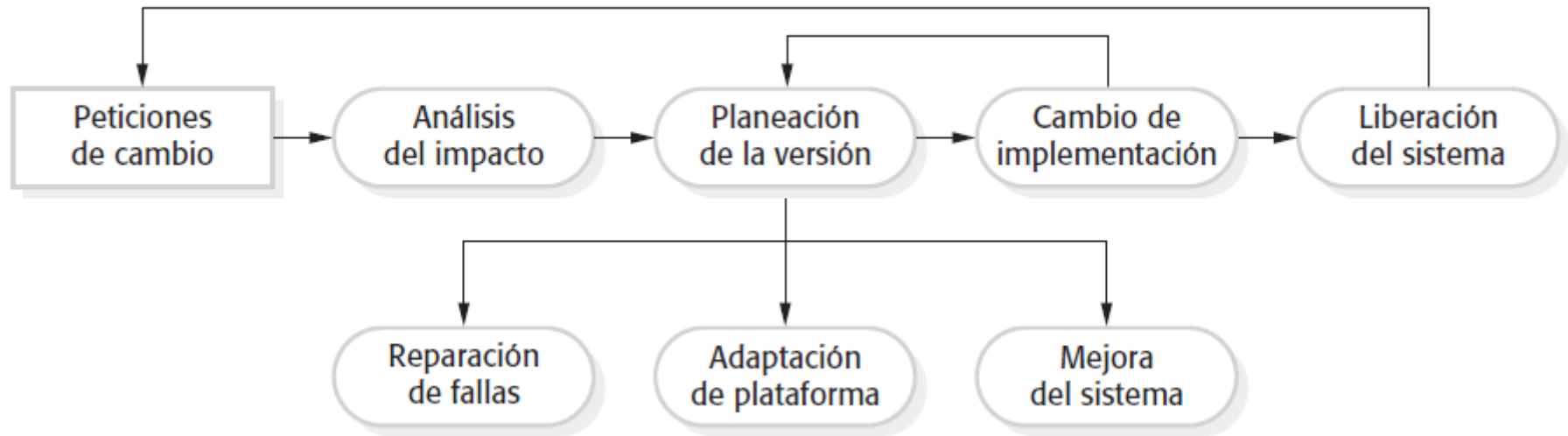


Figura 9.4 Proceso de evolución del software

Implementación de cambios

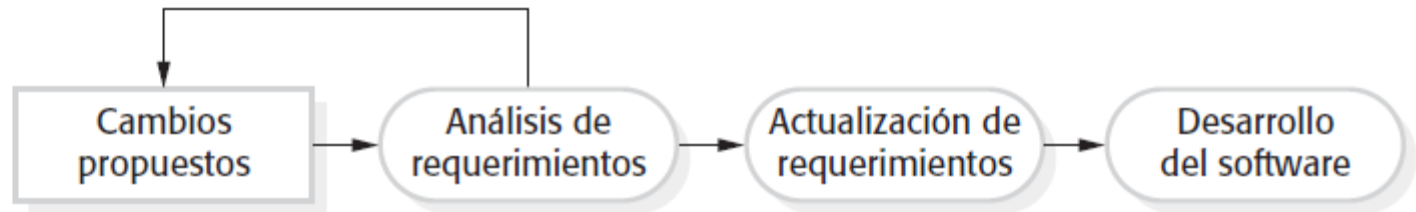


Figura 9.5
Implementación
del cambio

- En cada **iteración del proceso de desarrollo**, las revisiones del sistema se diseñan, se implantan y se aprueban.
- El **desarrollo de un cambio conlleva la comprensión del programa**, sobre todo si no se mantienen los mismos desarrolladores.
 - Hay que comprender cómo se estructura el programa, cómo ofrecer la funcionalidad y cómo afectará el cambio al programa actual.

Las solicitudes de cambio de urgencia

- **Cambios urgentes.** Se pueden ejecutar **sin pasar por todas las etapas del proceso** de ingeniería de software.
 - Un **fallo grave** debe ser acometido sin interrumpir el funcionamiento del programa.
 - Ojo: Los **cambios en el entorno del sistema** o actualización de un sistema operativo pueden tener consecuencias negativas.
 - Los **cambios de negocio** que requieren una **respuesta inmediata** por parte del equipo de desarrollo. Ejemplo: lanzamiento de una novedad de un producto.

- **Los requisitos del sistema pueden cambiar mientras el sistema se está desarrollando** porque el entorno está cambiando.
- Los sistemas están estrechamente vinculados con su entorno.
 - Cuando un sistema se instala en un entorno ha de adaptarse, y por lo tanto, **los requisitos del sistema pueden cambiar.**
- Los sistemas **deben ser modificados para ser útiles** en un entorno.

- Es la **modificación de un programa después de que se haya hecho la puesta en marcha**.
 - Suele suceder en los productos a medida.
 - Los productos genéricos deben evolucionar constantemente (nueva versión del producto)
- El mantenimiento normalmente **no implica grandes cambios** en la arquitectura del sistema.
 - Los cambios se implementan mediante la modificación de los componentes existentes y la agregación de nuevos componentes al sistema.

- **Mantenimiento para reparar errores. Correctivo.**
 - Cambio de un sistema para corregir las deficiencias que surgen.
- **Mantenimiento para agregar o modificar la funcionalidad del Sistema. Evolutivo.**
 - Modificar el sistema para satisfacer las nuevas necesidades.
- **Mantenimiento para adaptar el software a un entorno operativo diferente. Adaptativo.**
 - Cambio de un sistema para que opere en un entorno diferente (ordenador, sistema operativo, etc.)

- **La estabilidad del equipo**
 - Los costes de mantenimiento se reducen si trabaja el mismo equipo en desarrollo y en el mantenimiento.
- **Responsabilidad contractual**
 - Los desarrolladores de un sistema no suelen tener ninguna responsabilidad contractual de mantenimiento por lo que no hay motivación para realizar tareas relacionadas con el mantenimiento en el desarrollo del producto.
- **Habilidades del personal**
 - El personal de mantenimiento a veces no tienen la experiencia necesaria.
- **Tiempo de vida y estructura del programa**
 - A medida que los programas aumentan su tiempo de operación, su estructura se degrada y se vuelven más difíciles de entender y de cambiar.

- **Re-estructuración o re-escribir todo o parte de un sistema heredado sin cambiar su funcionalidad.**
- Aplicable cuando algunos subsistemas de un sistema más grande requieren un mantenimiento frecuente.
- La reingeniería involucra **añadir dedicación** para hacer que el programa sea **más fácil de mantener**. El sistema ha de ser **re-estructurado** y **re-documentado**.

- **Riesgo reducido**
 - Hay un alto riesgo en el desarrollo de software nuevo. Puede haber problemas de desarrollo, problemas de personal y problemas de especificación.
- **Coste reducido**
 - El coste de la reingeniería es a menudo mucho menos que los costes de desarrollo de un nuevo software.

- La **re-ingeniería** tiene lugar después de operar y mantener un sistema durante cierto tiempo y el **coste** de mantenimiento **ha aumentado**:
 - Utiliza **herramientas automatizadas** para rediseñar un sistema y crear un nuevo sistema que sea más sencillo de mantener.
- La **refactorización** es un **proceso continuo de mejora** en todo el proceso de desarrollo y evolución. Se tiene la intención de evitar la degradación de la estructura y el código del programa. De esta manera, se **evitan costes en la dedicación de un mantenimiento posterior**.

- La **evaluación** de valor de **negocio de un producto** debe tomar en cuenta diferentes puntos de vista:
 - Los usuarios finales del sistema
 - Los clientes (negocio)
 - Los gerentes
 - Los administradores de software
 - Altos directivos.
- Es necesario **entrevistar** a los diferentes interesados y recopilar los resultados.

¿Cómo evaluamos el valor de negocio del producto?

- **El uso del sistema**
 - Si los sistemas **se utilizan solamente de** vez en cuando o por un pequeño número de personas, estos pueden tener un **bajo valor** comercial.
- **Los procesos de negocio** que son soportados
 - Un sistema puede tener un bajo valor comercial si se impone el **uso de procesos de negocio ineficientes**.
- **La fiabilidad** del sistema
 - Si un **sistema no es confiable** y los problemas afectan directamente a los clientes, el valor del sistema baja su valor.
- **Las salidas** del sistema
 - Si la empresa depende de los **resultados del sistema y son buenos**, el valor del mismo aumenta.

¿Cómo evaluamos su calidad?

- Evaluación de **procesos de negocio**
 - ¿En qué medida el proceso de negocio apoya los objetivos actuales de la empresa?
- Evaluación del **entorno**
 - ¿Qué eficiente es el entorno del sistema y cuánto cuesta mantenerlo?
- Evaluación de la **solicitud**
 - ¿Qué calidad tiene el sistema de software?

- Utilizar un enfoque de perspectiva **orientada a procesos** y buscar respuestas de los actores del sistema:
 - ¿Existe un modelo de proceso definido y continúa hoy en día?
 - ¿Las diferentes partes de la organización utilizan diferentes procesos para la misma función?
 - ¿Cómo se ha adaptado el proceso?
 - ¿Cuáles son las relaciones con otros procesos de negocio y son éstos necesarios?
 - ¿El proceso cubre sus necesidades con la aplicación de software?
- Ejemplo: un sistema de solicitud de viajes puede tener un bajo valor comercial debido al uso generalizado de la solicitud realizada por la web.