

Actividades del proceso

Héctor Antillanca E.
Depto. Ingeniería Informática
UNIVERSIDAD DE SANTIAGO DE CHILE

Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

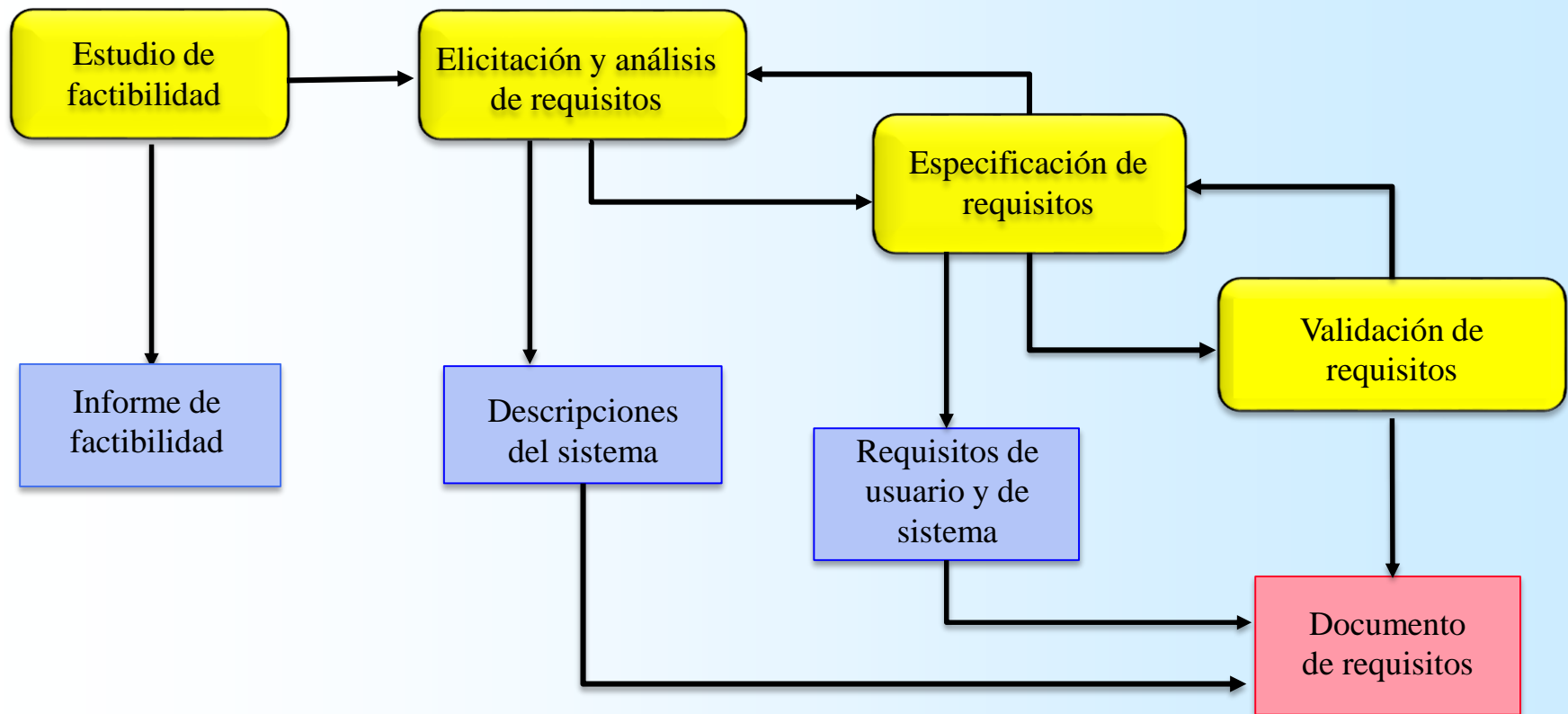
Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

Especificación del software

- ◆ Es el proceso de establecer qué servicios debe proveer el software y cuales son sus restricciones de desarrollo y de operación.
- ◆ *Proceso de ingeniería de requisitos*
 - Estudio de factibilidad
 - » Estudio del problema y una solución propuesta. Se decide continuar o no con desarrollo del proyecto.
 - Elicitación y análisis de requisitos
 - » Conocer lo que esperan los interesados (stakeholders) de una solución basada en software.
 - Especificación de requisitos
 - » Definir con detalle los requisitos del futuro sistema.
 - Validación de requisitos
 - » Comprobar la validez de los requisitos.

El proceso de ingeniería de requisitos



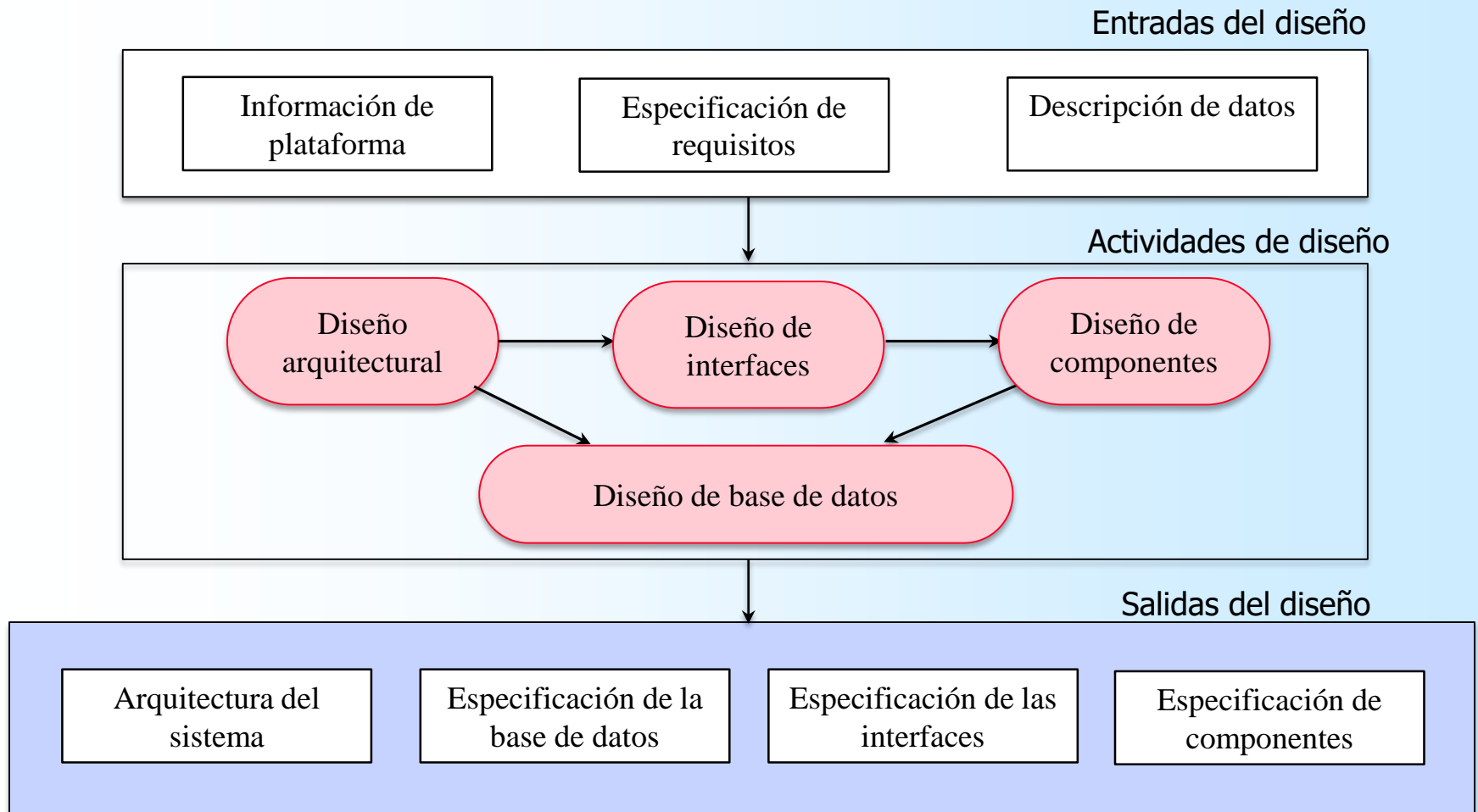
Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

Implementación y diseño de software

- ◆ Es el proceso de convertir la especificación del sistema en uno ejecutable.
- ◆ Diseño de software
 - Diseñar una estructura que realice la especificación
- ◆ Implementación
 - Traducir esta estructura en un programa ejecutable;
- ◆ Estas actividades están muy relacionadas y pueden estar combinadas.

Modelo general de un proceso de diseño



Actividades del diseño

- ◆ *Diseño arquitectura*, se identifica la estructura completa del sistema, los componentes principales (subsistemas y módulos), sus relaciones y cómo ellos están distribuidos.
- ◆ *Diseño de base de datos*, se diseña la estructura de datos del sistema y cómo serán representados en una base de datos.
- ◆ *Diseño de las interfaces*, se definen las interfaces entre los componentes.
- ◆ *Selección y diseño de componentes*, se buscan componentes reutilizables. Si no los hay, se diseña cómo deberían operar.

Implementación del sistema

- ◆ El software se implementa ya sea desarrollando un programa o programas o configurando un sistema de aplicación.
- ◆ Las actividades de diseño e implementación están entrelazadas para la mayoría de los tipos de sistemas de software.
- ◆ La programación es una actividad individual, no hay un proceso estándar.
- ◆ Depuración es la actividad de encontrar defectos y corregirlos.

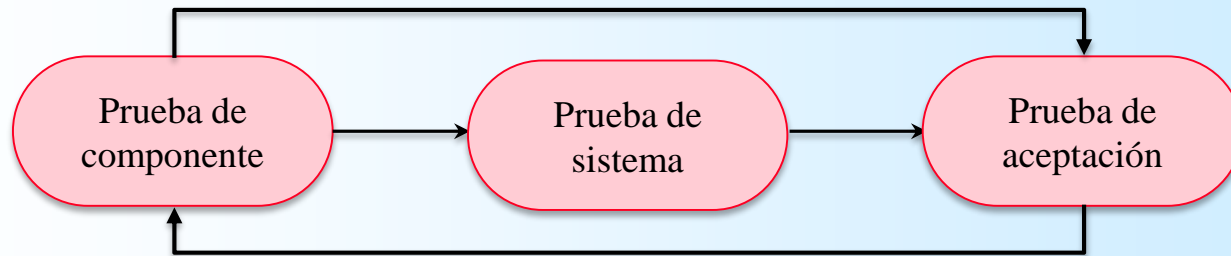
Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

Validación del software

- ◆ Verificación y validación (V & V): actividades para mostrar que un sistema está conforme con respecto a su especificación (verificado) y que satisface las necesidades del cliente (válido).
- ◆ Implica comprobar y revisar los procesos seguidos y probar el sistema.
- ◆ La prueba del sistema involucra ejecutar el sistema con casos de prueba derivados de la especificación de los datos reales a ser procesados por el sistema.

Etapa de Pruebas (testing)



Etapa de Pruebas

◆ Prueba de componentes

- Cada componente se prueba en forma separada.
- Los componentes pueden ser funciones, objetos o agrupaciones coherentes de estas entidades.

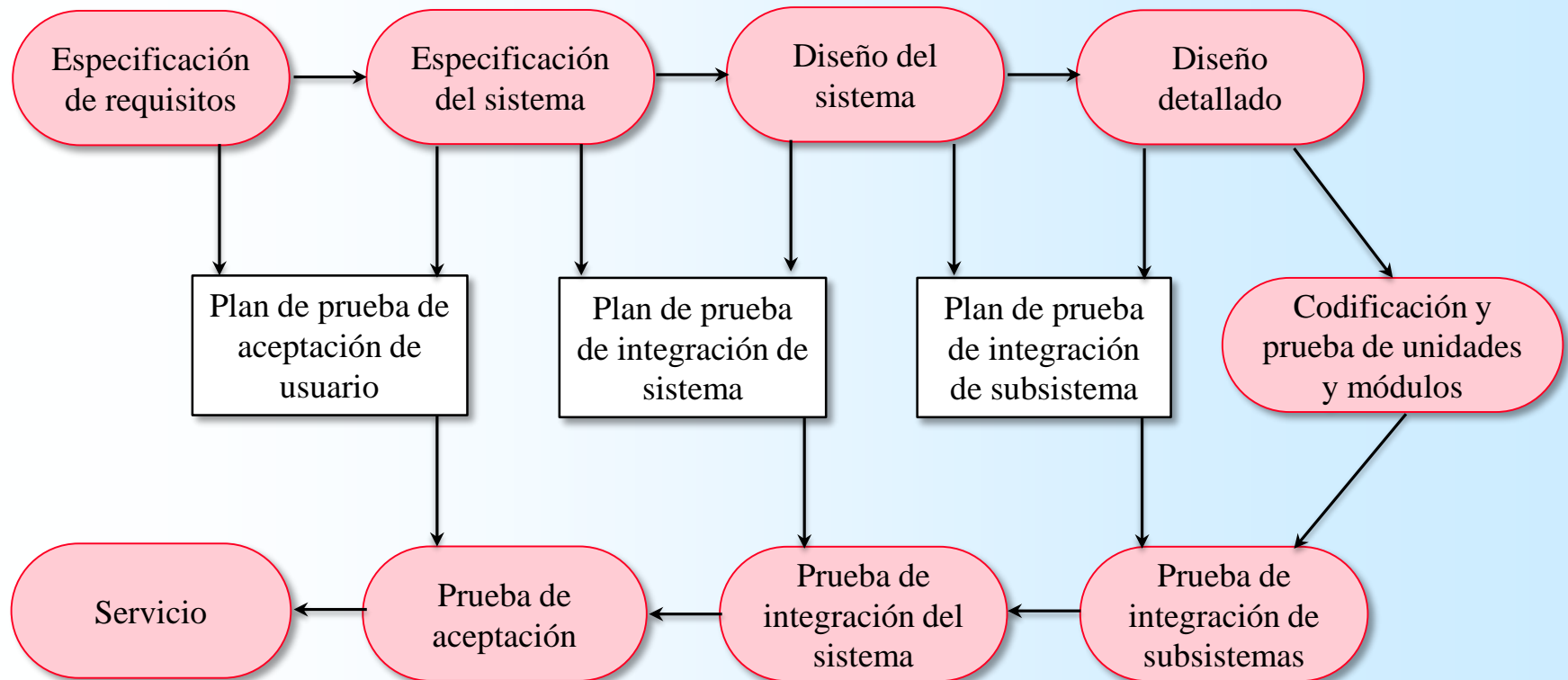
◆ Prueba del sistema

- Se prueba el sistema completo. Muy importante probar las propiedades emergentes.

◆ Prueba de cliente

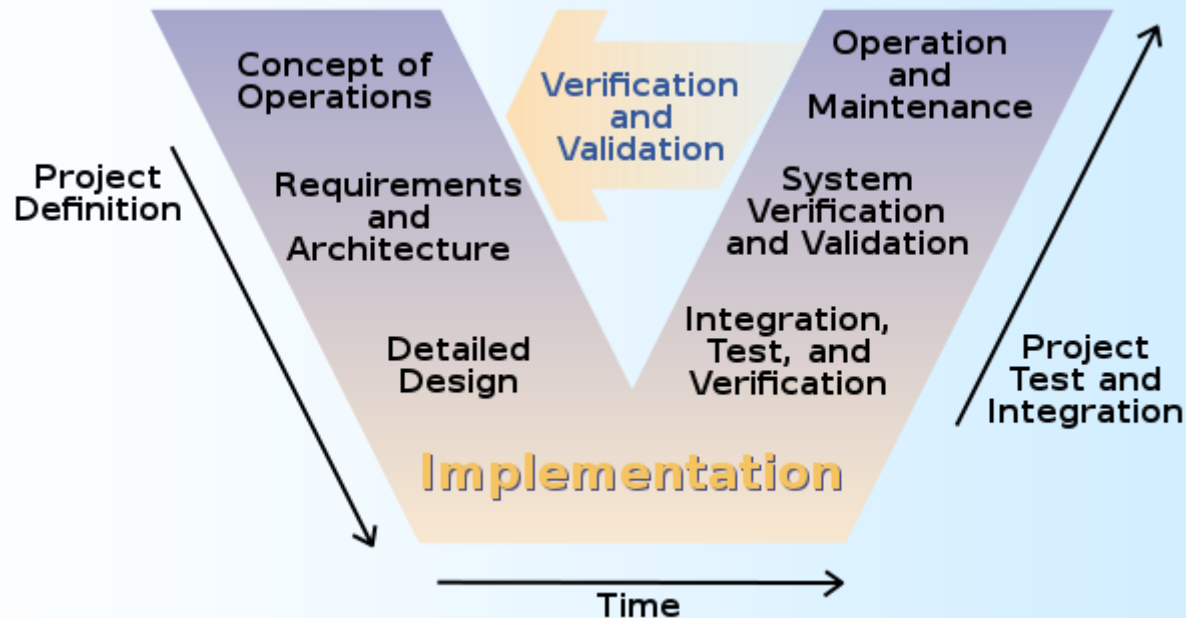
- El software se prueba con datos del cliente para comprobar que satisface las necesidades del cliente.

Fase de prueba en un proceso controlado por plan (modelo V)



Propuesto por la Administración Federal Alemana

Fase de prueba en un proceso controlado por plan (modelo V)



Pruebas de aceptación

◆ Pruebas “alfa”

- Son pruebas que se realizan a los sistemas hechos a la medida, para un cliente.
- La prueba termina cuando el cliente como el desarrollador están de acuerdo en que el sistema entregado es una implementación aceptable de los requisitos.

◆ Prueba “beta”

- Cuando el software es un producto para un mercado de potenciales clientes, el proceso de prueba es denominado prueba “beta”.
- Se entrega el sistema a algunos potenciales clientes que están de acuerdo con usar el sistema. Lo prueban y reportan retroalimentación,
- Con esta información el sistema se modifica y se libera, ya sea para hacer una nueva prueba “beta” o para su venta.

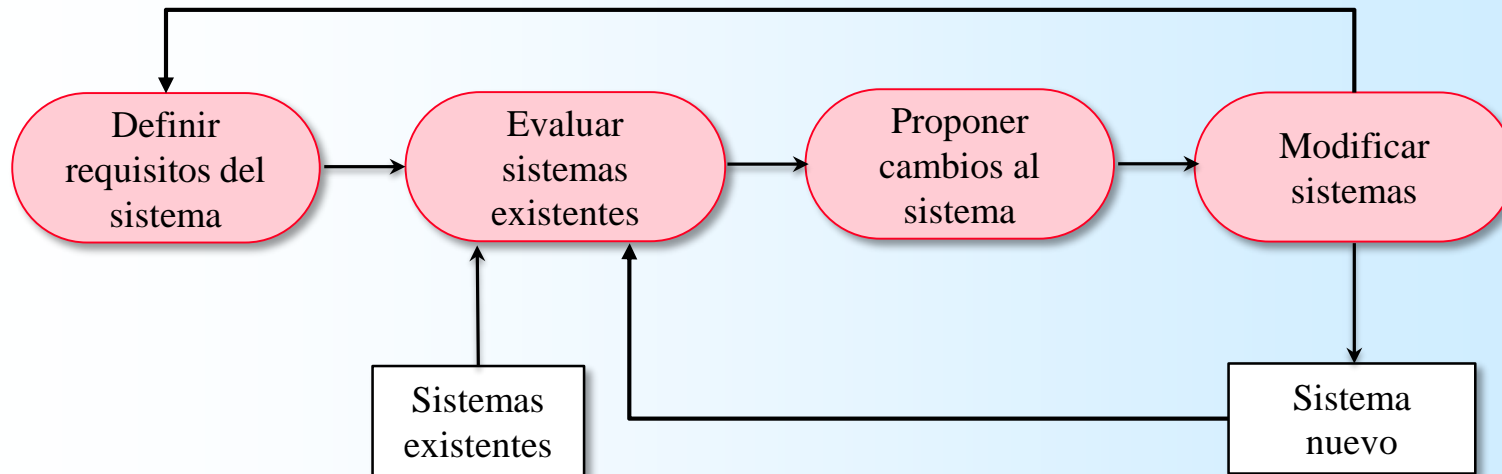
Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

Evolución del software

- ◆ El software es inherentemente flexible y puede cambiar.
- ◆ En la medida que cambian los requisitos del software, debido a los cambios de las circunstancias del negocio, el software que lo soporta también debe evolucionar y cambiar.
- ◆ La demarcación entre desarrollo y evolución (mantención) es menos relevante en la medida que los sistemas son cada vez menos completamente nuevos.

Evolución de sistema



Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

Lidiando con los cambios

- ◆ El cambio es inevitable en todos los sistemas de software grandes.
 - Los cambios del negocio lleva a nuevos requisitos del sistema
 - Nuevas tecnologías abren nuevas posibilidades de mejorar las implementaciones.
 - El cambio de la plataforma require tambien cambiar las aplicaciones.
- ◆ Los cambios llevan a re-hacer trabajo de modo que los costos del cambio incluye el trabajo adicional (esto es, re-analizar los requisitos) así como el costo de implementar nuevas funciones.

Como reducir los costos del rehacer

- ◆ *Anticipar el cambio.* Se requiere que el proceso de software incluya actividades que anticipen posibles cambios antes de que sea necesario rehacer trabajo significativo.
 - Por ejemplo, se puede desarrollar un sistema prototipo para mostrar algunas características del sistema clave para los clientes.
- ◆ *Tolerar el cambio.* Se diseña el proceso de manera que los cambios se puedan acomodar a un costo relativamente bajo.
 - Esto normalmente involucra alguna forma de desarrollo incremental. Los cambios que sean propuestos pueden ser implementados en incrementos que todavía no han sido desarrollados. Si esto es imposible, entonces solamente un incremento (una parte pequeña del sistema) puede ser alterado para incorporar el cambio.

Lidiando con los requisitos cambiantes

- ◆ *Prototipo del sistema.* Se desarrolla rápidamente una versión o parte del sistema para comprobar los requisitos del cliente y determinar la factibilidad de las decisiones del diseño. Este enfoque apoya la anticipación del cambio.
- ◆ *Entrega incremental.* Se entregan al cliente incrementos del sistema para comentarios y experimentación. Esto apoya evitar los cambios y tolerar los cambios.

Prototipado de software

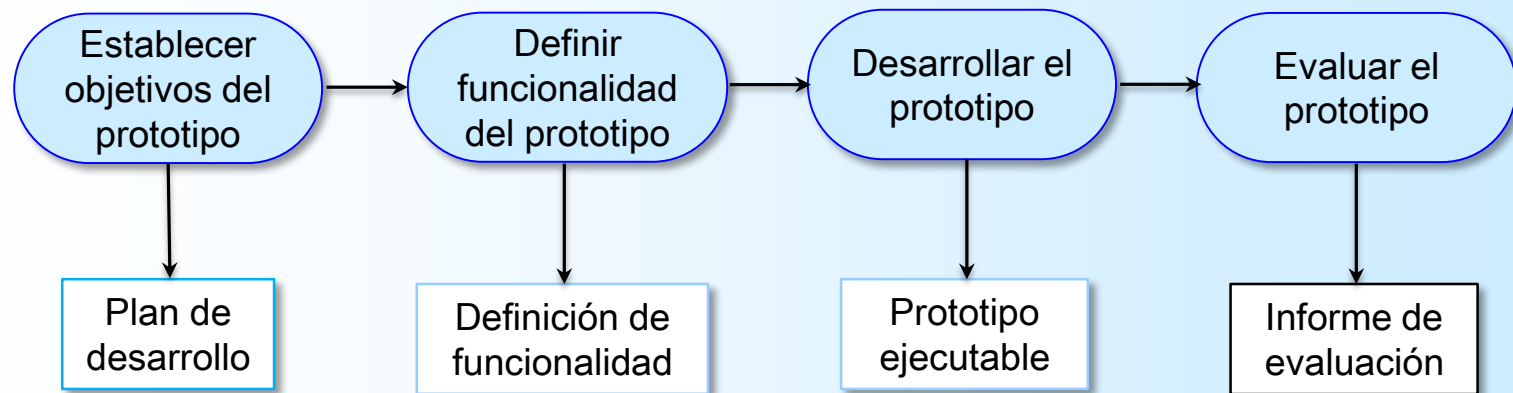
- ◆ Un prototipo es una versión inicial de un sistema y se construye para demostrar conceptos e intentar opciones de diseño.
- ◆ Un prototipo puede ser usado en:
 - El proceso de ingeniería de requisitos para ayudar con la elicitación y validación de requisitos;
 - En los procesos de diseño para explorar opciones y desarrollar un diseño de IU;
 - En el proceso de prueba para ejecutar pruebas back-to-back¹.

1: Back-to-back testing: dos o más componentes de diferentes versiones (destinados a responder con resultados similares), son probados con la misma serie de datos en diferentes condiciones.

Beneficios del prototipar

- ◆ Mejora la usabilidad del sistema.
- ◆ Mejora el conocimiento de las necesidades reales del usuario.
- ◆ Mejora la calidad del diseño.
- ◆ Mejora la mantenibilidad.
- ◆ Se reduce el esfuerzo de desarrollo.

Proceso de desarrollo de prototipo



Desarrollo de prototipo

- ◆ Puede estar basado en herramientas o lenguajes de prototipado rápido
- ◆ Puede significar dejar funcionalidad fuera
 - El prototipo debería enfocarse en áreas del producto no están bien entendidas;
 - Pueden no estar incluidas en el prototipo la comprobación y recuperación de errores;
 - Se enfoca más en los requisitos funcionales que en las condiciones o restricciones de operación tales como la confiabilidad y la seguridad.

Prototipos desechables

- ◆ Este tipo de prototipo debería ser descartado después que ha sido desarrollado dado que no es una buena base para un sistema de producción:
 - Normalmente es imposible sintonizarlo para que satisfaga las restricciones y condiciones de operación;
 - Los prototipos normalmente no tienen documentación;
 - La estructura del prototipo usualmente se degrada en la medida que van cambiando los requisitos;
 - El prototipo probablemente no satisface los estándares de calidad de la organización.

Entrega incremental

- ◆ Más que liberar un sistema como una simple entrega, el desarrollo y la entrega es dividida en incrementos con cada entrega siendo una parte de la funcionalidad requerida.
- ◆ Los requisitos son priorizados y los de más alta prioridad son incluidos en los primeros incrementos.
- ◆ Una vez que se inicia el desarrollo de un incremento, sus requisitos son congelados. Los requisitos de los próximos incrementos pueden seguir siendo analizados y evolucionar.

Desarrollo y entrega incremental

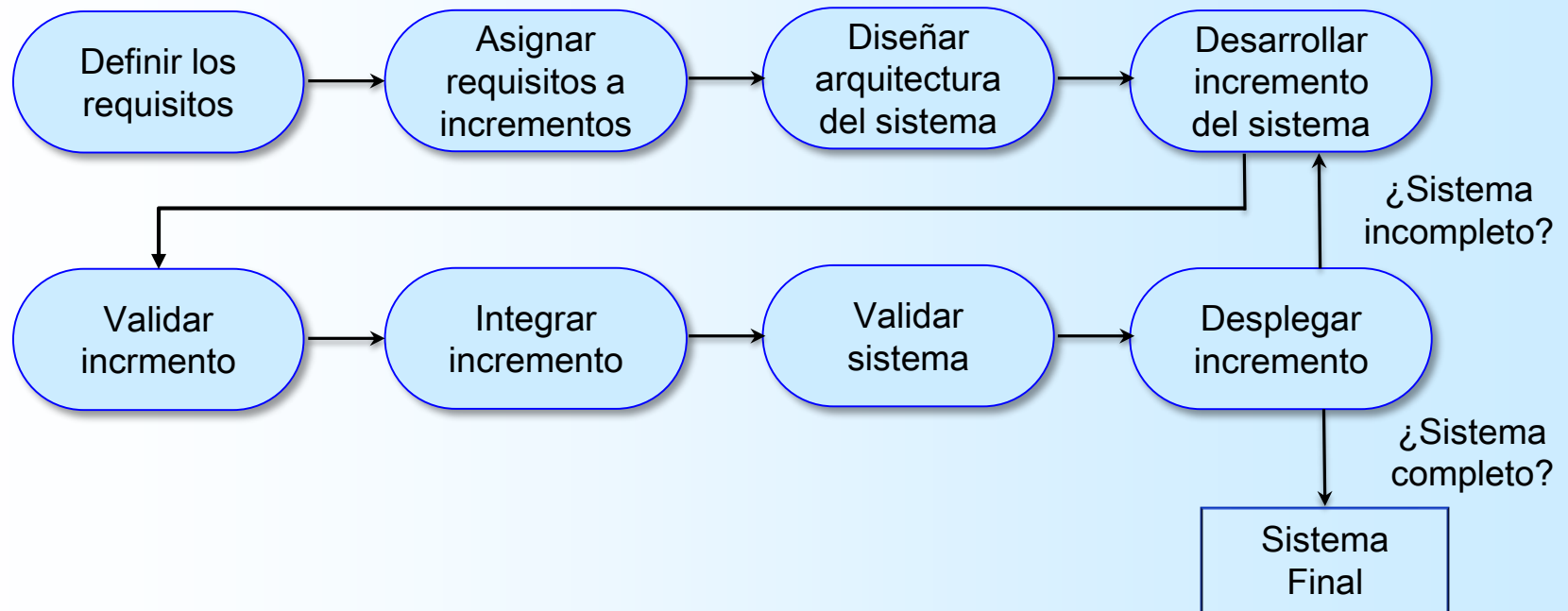
◆ Desarrollo incremental

- Se desarrolla el sistema en incrementos y se evalúa cada incremento antes de proceder a desarrollar el próximo;
- Este enfoque es normal en los métodos ágiles;
- La evaluación la hace un usuario/cliente cercano.

◆ Entrega incremental

- Se despliega un incremento para ser usado por usuarios finales;
- Se obtiene una evaluación más realista del uso práctico del software;
- Es difícil implementar sistemas de reemplazo cuando los incrementos tienen menos funcionalidad que el sistema que está siendo reemplazado.

Entrega incremental



Ventajas de la entrega incremental

- ◆ Con cada incremento se puede entregar valor al cliente al ofrecerle tempranamente funcionalidad útil del sistema.
- ◆ Cada incremento actúa como un prototipo que ayuda a elicitar requisitos de los futuros incrementos.
- ◆ Se reducen los riesgos de fracaso del proyecto completo.
- ◆ Los servicios del sistema de más alta prioridad tienden a ser más probados.

Problemas de la entrega incremental

- ◆ La mayoría de los sistemas requieren un conjunto básico de facilidades que usan diferentes partes del sistema.
 - Como los requisitos no están definidos en detalle hasta que un increment es implementado, puede ser difícil identificar facilidades comunes necesarias para todos los incrementos.
- ◆ La esencia del proceso interactivo es que la especificación es desarrollada en conjunto con el software.
 - Sin embargo, esto entra en conflicto con el modelo de adjudicación de los proyectos que tienen muchas organizaciones, donde la especificación completa del sistema es parte del contrato de desarrollo del sistema.

Temario

- ◆ Especificación de software
- ◆ Desarrollo de software
- ◆ Validación de software
- ◆ Evolución del software
- ◆ ¿Cómo enfrentar el cambio?
- ◆ Mejoramiento del proceso

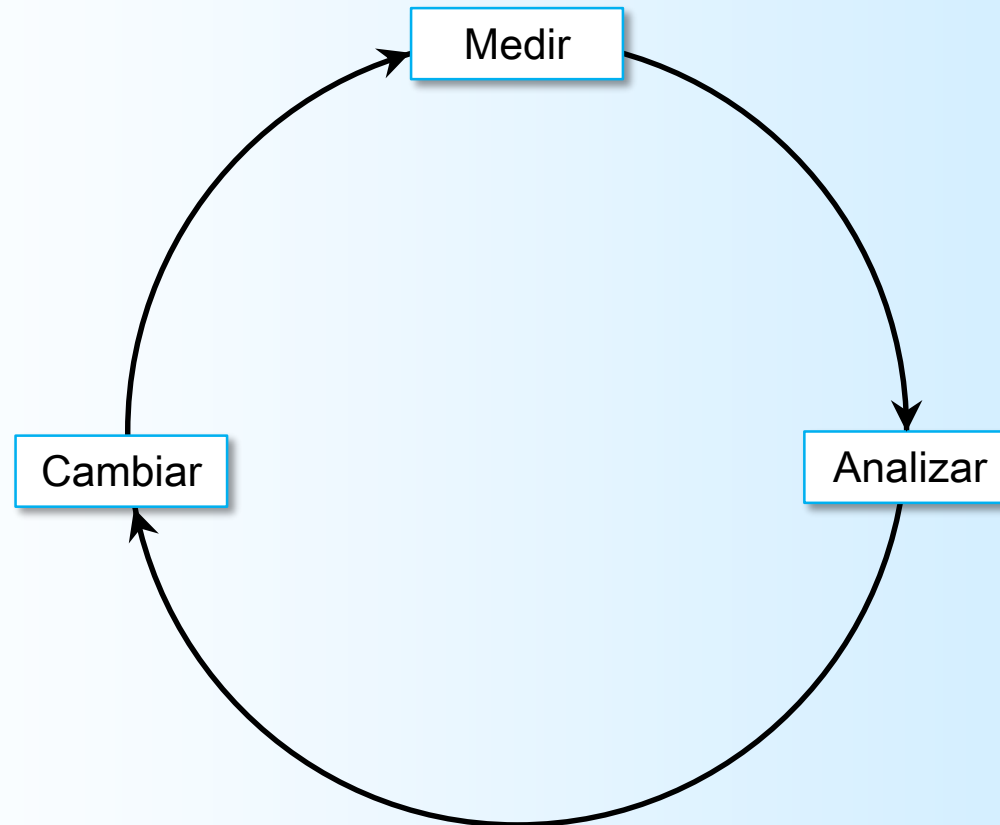
Mejoramiento del proceso

- ◆ Muchas compañías de software incluyen la actividad de mejoramiento del proceso como una manera de aumentar la calidad de su software, ya sea reduciendo costos o acelerando sus procesos de desarrollo.
- ◆ Mejorar el proceso de software significa comprender bien los procesos existentes y cambiar los procesos para aumentar la calidad del producto y/o reducir los costos y tiempos de desarrollo.

Enfoques para mejorar

- ◆ Aumentar la *madurez del proceso*, se enfoca en mejorar el proceso y la gestión del proyecto introduciendo buenas prácticas de ingeniería de software.
 - El nivel de madurez del proceso es una medida que refleja el grado de adopción de buenas prácticas técnicas y de gestión en la ejecución de los procesos de software de la organización.
- ◆ *Agilizar* el proceso, se enfoca en el desarrollo iterativo y en la reducción de la sobrecarga de trabajo en el proceso.
 - Las principales características de los métodos ágiles son la entrega rápida de funcionalidad y la capacidad de respuesta efectiva (“responsividad”) a las necesidades cambiantes del cliente.

Ciclo de mejoramiento del proceso



Actividades de mejoramiento del proceso

◆ *Medir el proceso*

- Usted mide uno o más atributos del proceso o del producto. Estas mediciones forman una base que le ayuda a decidir si las acciones de mejora del proceso han sido efectivas.

◆ *Analizar el proceso*

- El proceso actual es evaluado, se identifican las debilidades y cuellos de botella del proceso. Se desarrollan modelos de procesos (a veces llamados mapas del proceso) para describir los procesos.

◆ *Cambiar el proceso*

- Se proponen cambios al proceso para tratar debilidades identificadas. Se introducen los cambios y el ciclo resume los datos reunidos acerca de la efectividad de los cambios.

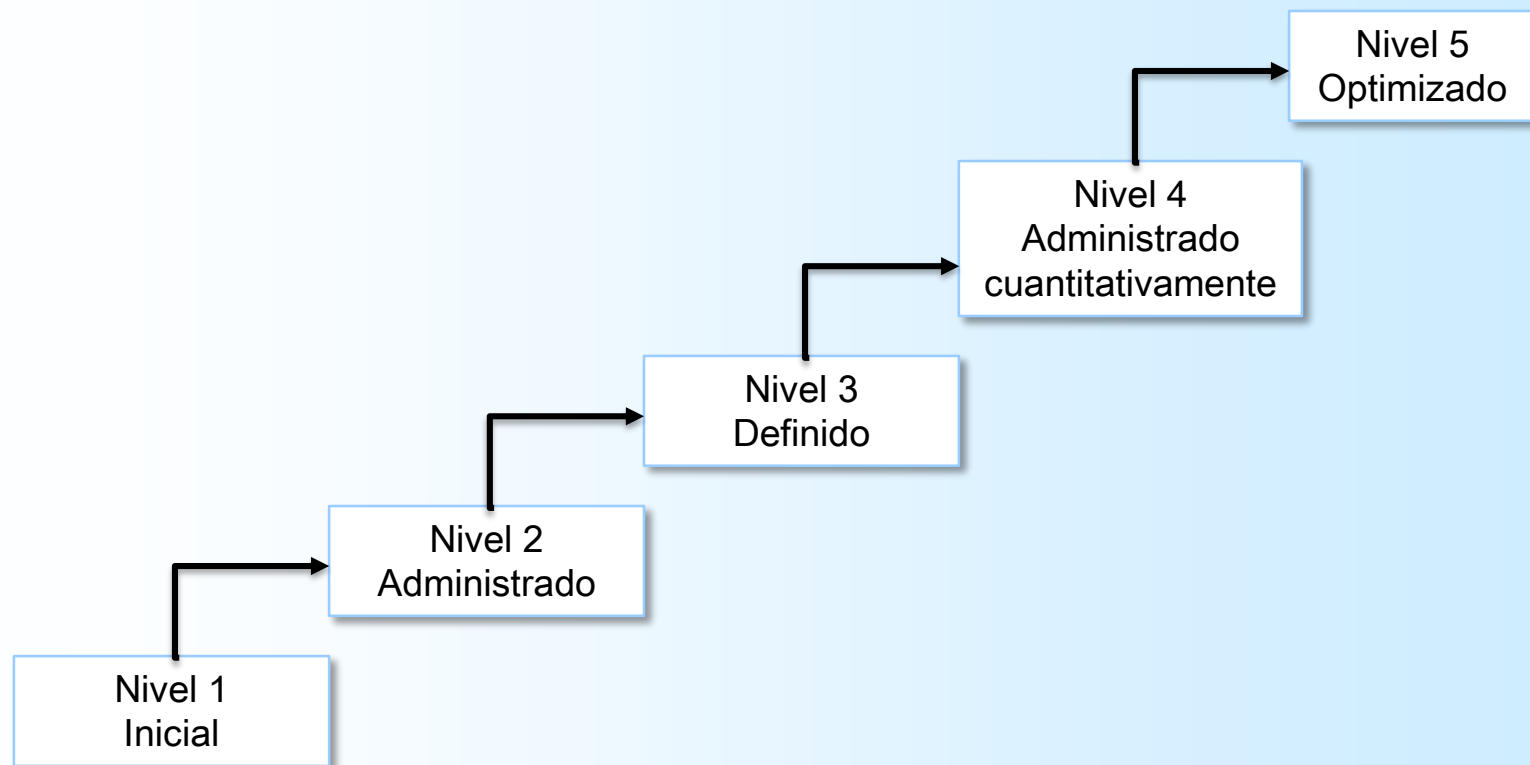
Medir el proceso

- ◆ Donde sea possible, coleccionar datos cuantitativos del proceso.
 - Sin embargo, en las organizaciones que no tienen claramente establecido estándares de proceso es muy difícil si usted no sabe qué medir. Un proceso debiera estar definido antes de hacer cualquier medición.
- ◆ Las mediciones deberían ser usadas para evaluar las mejoras.
 - Esto no significa que las mediciones debieran controlar las mejoras. Controlar las mejoras debiera ser un objetivo organizacional.

Métricas del proceso

- ◆ Tiempo que toman las actividades del proceso en ser completadas
 - Por ejemplo, tiempo calendario o esfuerzo para completar una actividad o proceso.
- ◆ Recursos requeridos por los procesos o actividades
 - Por ejemplo, esfuerzo total en días-persona.
- ◆ Número de ocurrencias de un evento particular
 - Por ejemplo, número de defectos descubiertos.

Niveles de madurez de capacidad



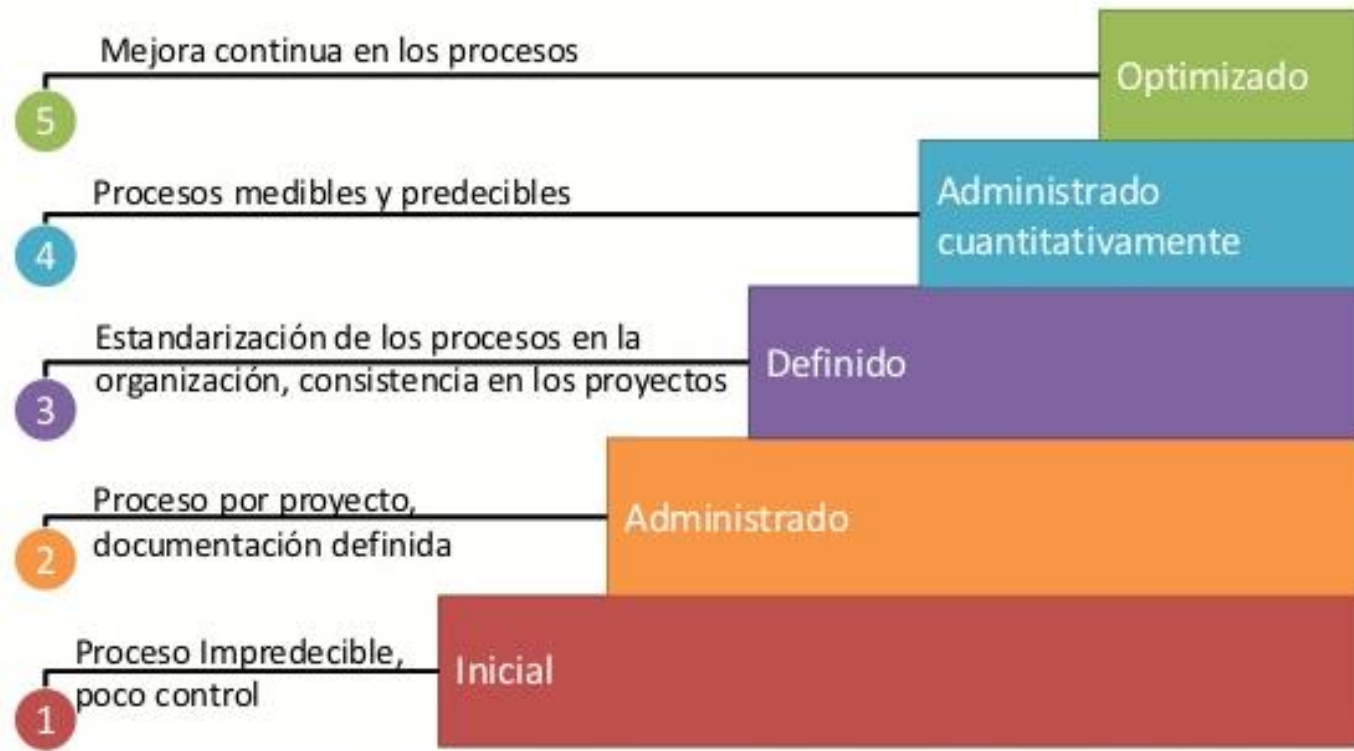
Modelo de madurez de capacidad del SEI

- ◆ Inicial
 - Esencialmente no controlado
- ◆ Repetible
 - Uso de procedimientos de gestión del producto bien definidos.
- ◆ Definido
 - Uso de procedimientos y estrategias de gestión del proceso bien definidas.
- ◆ Administrado
 - Uso de estrategias de gestión de calidad bien definidas.
- ◆ Optimizado
 - Uso de estrategias de mejoramiento del proceso bien definidas.

Modelo de madurez de capacidades (CMM)

- ◆ **Modelo de Madurez de Capacidades o CMM** (*Capability Maturity Model*): Modelo de evaluación de los procesos de una organización.
- ◆ Desarrollado inicialmente para los procesos relativos al desarrollo e implementación de software por la Universidad Carnegie-Mellon para el [Software Engineering Institute](#) (SEI).
- ◆ El SEI es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMM" es una marca registrada del SEI.
- ◆ CMMI: evolución de CMM. **Integración de varios modelos de madurez de capacidades** de una organización o **Capability Maturity Model Integration** (CMMI).

Niveles de madurez de CMMI



Puntos importantes

- ◆ Los procesos del software corresponden a las actividades involucradas en la producción de los sistemas de software.
- ◆ Los modelos de procesos generales describen la organización de los procesos de software.
 - Ejemplos de esos modelos generales incluyen el modelo cascada, el desarrollo incremental y el desarrollo basado en reutilización.
- ◆ La ingeniería de requisitos es el proceso de desarrollar una especificación de software.

Puntos importantes

- ◆ En los procesos de diseño e implementación se transforma la especificación de requisitos en un sistema de software ejecutable.
- ◆ La validación del software es el proceso de comprobar que el sistema está conforme a su especificación y satisface la necesidades reales de los usuarios del sistema.
- ◆ La evolución del software toma lugar cuando usted cambia los sistemas de software existentes para que satisfagan nuevos requisitos. El software debe evolucionar para seguir siendo útil.
- ◆ Los procesos deberían incluir actividades tales como desarrollo de prototipo y entrega incremental para enfrentar los cambios.

Puntos importantes

- ◆ Los procesos pueden ser estructurados para desarrollo y entrega iterativa de manera que los cambios puedan hacerse si alterar el sistema como un todo.
- ◆ Los principales enfoques para mejoramiento de procesos son los enfoques ágiles, comprometidos para reducir la sobrecarga de los procesos, y enfoques maduros basados en mayor gestión de proceso y el uso de buenas prácticas de ingeniería de software. The principal approaches to process improvement are agile
- ◆ El modelo de madurez de procesos del SEI identifica niveles de madurez que esencialmente corresponden al uso de buenas prácticas de ingeniería.