

Introducción a la Ingeniería del Software

1.El producto software

2.El concepto de Ingeniería del Software

3.El proceso de desarrollo del software

EL PRODUCTO SOFTWARE

Naturaleza del software

Definición de software

Características del software

Tipos y dominios de aplicación del software

Proceso de producción

EL PRODUCTO SOFTWARE

Naturaleza del software

Producto

- Proporciona potencial de cómputo
- Es un transformador de información

Vehículo

para distribuir un producto

- Actúa como base para el control de la computadora
Sistemas Operativos
- Actúa como base para la comunicación de información
Redes
- Actúa como base para la creación y control de otros programas
Herramientas y ambientes de software

El software distribuye el producto más importante de nuestro tiempo

INFORMACIÓN

EL PRODUCTO SOFTWARE

Definición de software

¿Software = Programa de computadora?

Definición incompleta

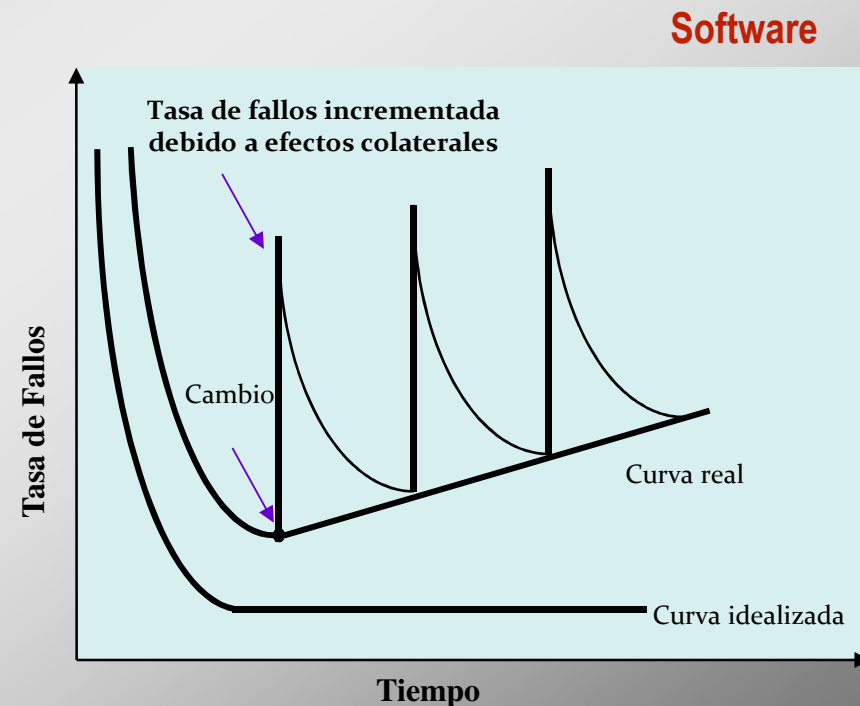
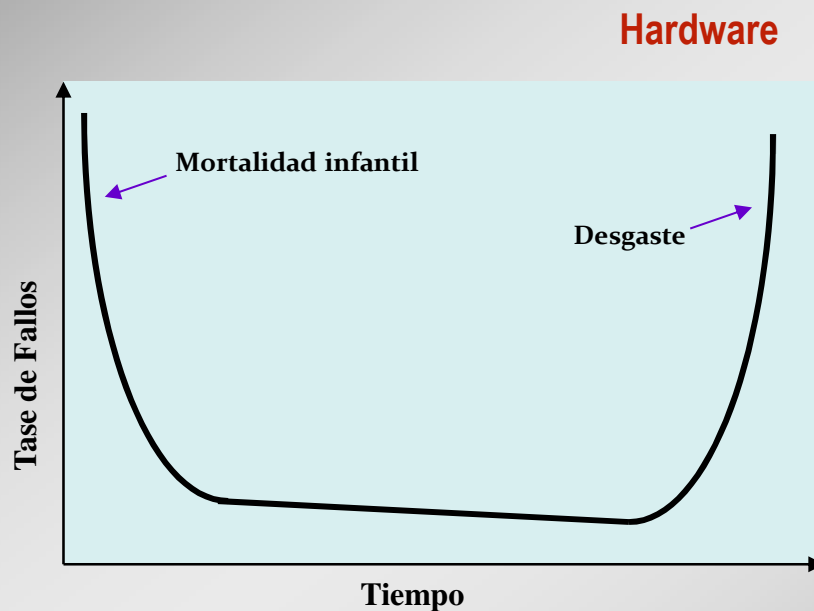
El software es:

1. Instrucciones (programas) que cuando se ejecutan proporcionan las funciones y características buscadas
2. Estructuras de datos que permiten a los programas manipular la información adecuadamente
3. Información en papel o en forma virtual (documentación) que describe la operación y uso de los programas

EL PRODUCTO SOFTWARE

Características del software

- El software no se fabrica en el sentido clásico
- El software no se “desgasta”, pero se deteriora



- La mayor parte del software se construye para uso individualizado

EL PRODUCTO SOFTWARE

Tipos y dominios de aplicación del software

+ Software genérico

Sistema autónomo producido por una organización de desarrollo y vendido en el mercado abierto a cualquier cliente que pueda comprarlo

+ Software a medida

Sistema desarrollado por una empresa especialmente para un cliente particular

Dominios de aplicación

+ Software de sistemas

Conjunto de programas que proporcionan servicio a otros programas

+ Software de aplicación

Programas que resuelven una necesidad específica de negocios

EL PRODUCTO SOFTWARE

- ✚ Software de ingeniería y ciencias

Implementa algoritmos "devoradores" de números

- ✚ Software empotrado

Reside dentro de un producto o sistema e implementa y controla características y funciones para el usuario final y para el sistema en sí

- ✚ Software de gestión

Proporciona una capacidad específica para uso de muchos consumidores diferentes

- ✚ Aplicaciones Web

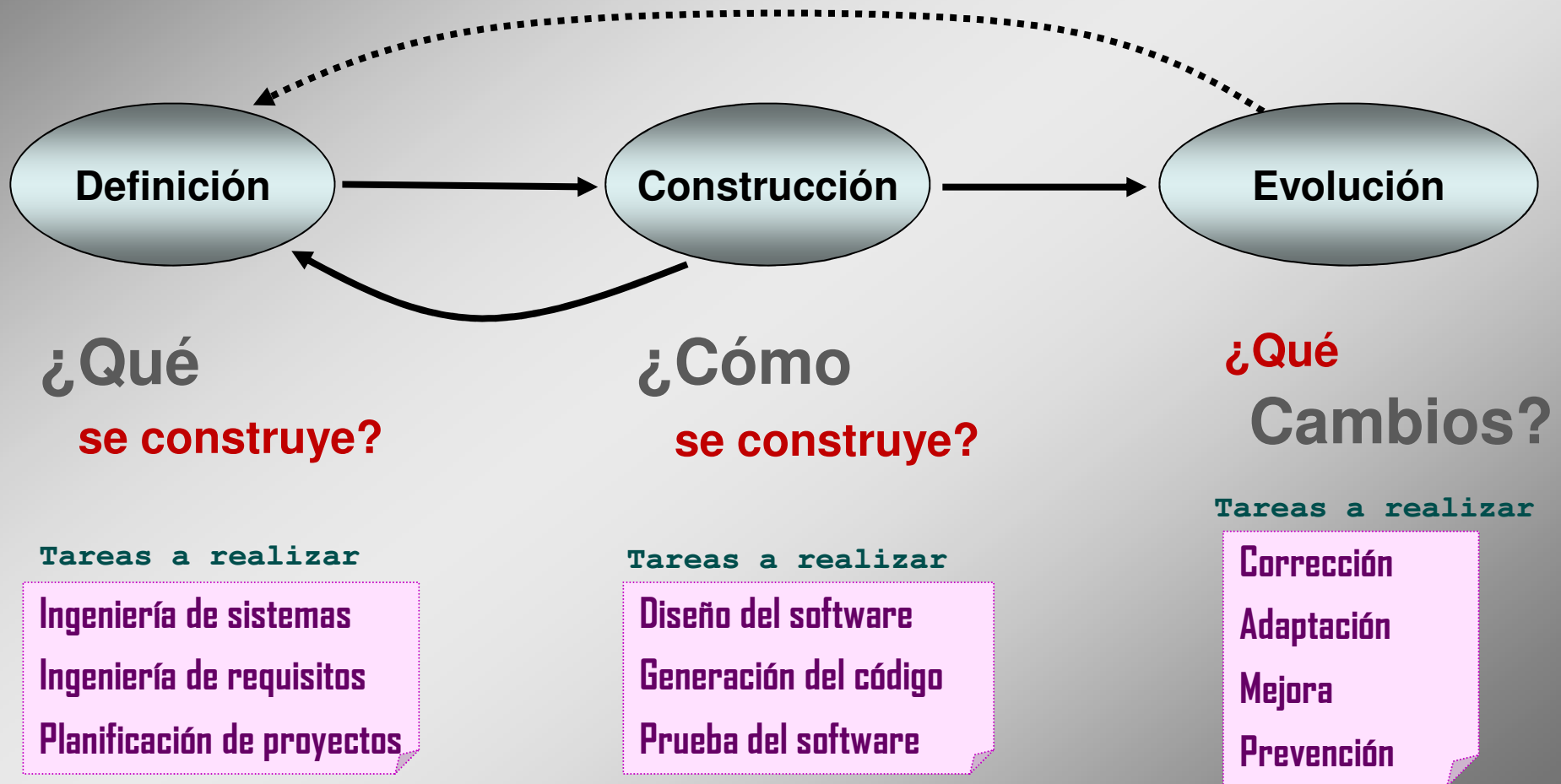
Software centrado en redes que agrupa una amplia gama de aplicaciones

- ✚ Software de inteligencia artificial

Implementa algoritmos no numéricos para resolver problemas complejos difíciles de tratar computacionalmente o con análisis directo

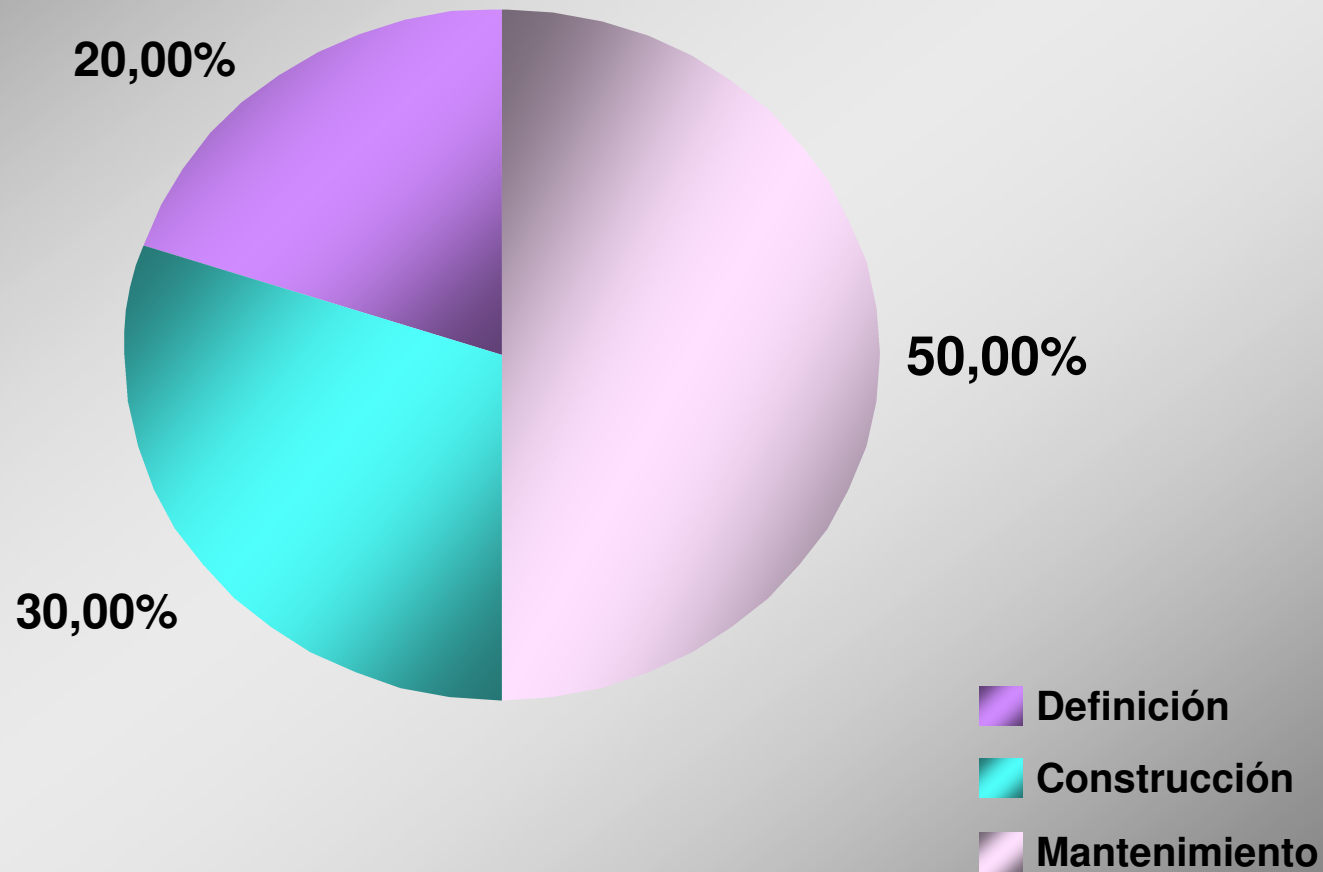
EL PRODUCTO SOFTWARE

Proceso de producción



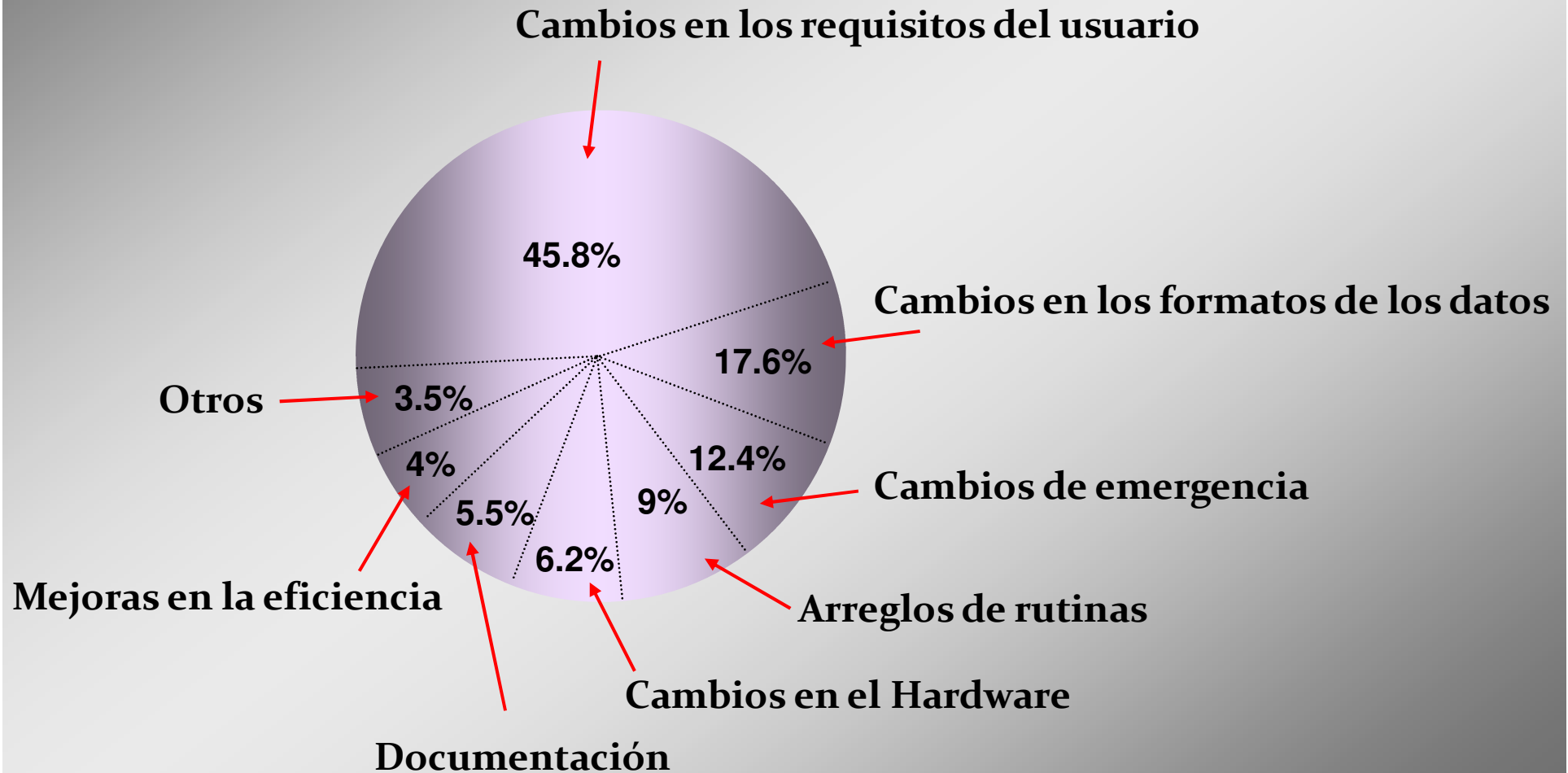
EL PRODUCTO SOFTWARE

Esfuerzo requerido por etapas



EL PRODUCTO SOFTWARE

Mantenimiento



EL PRODUCTO SOFTWARE

Problemas

- ✚ Comunicación entre personas

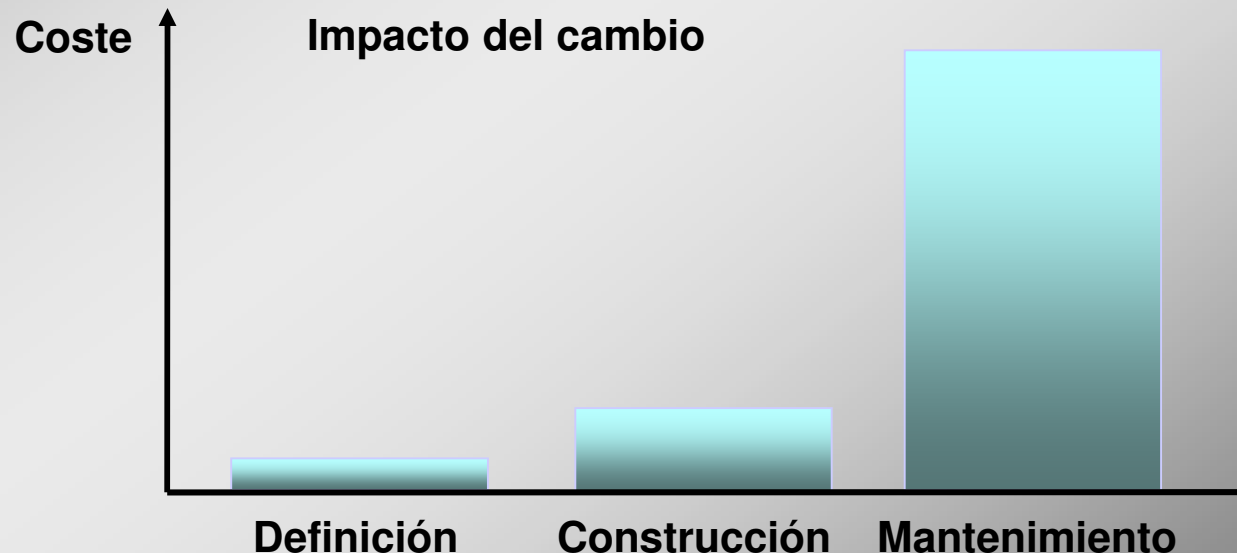
CLIENTES ↔ DESARROLLADORES



- ✚ Incumplimiento de la planificación

¿SOLUCIÓN: HORDA MONGOLIANA?

- ✚ Incorporar cambios en etapas avanzadas del proceso



EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

Definición de Ingeniería del Software

Terminología usada en ingeniería del software

EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

Ingeniería del software

Se propuso en 1968 para discutir

La crisis del software

Consecuencia del nuevo hardware

Software muy complejo
Grandes proyectos con años de retraso
Coste del software mucho más de lo previsto
Software poco fiable
Software difícil de mantener
Software de pobre ejecución

Se concluye

Se debe entender el problema antes de desarrollar una aplicación
El diseño es una actividad crucial
El software debe tener alta calidad
El software debe ser fácil de mantener

Debe hacerse

EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

Definición de Ingeniería del software

- ✚ “Establecimiento y uso de **principios fundamentales de la ingeniería** con objeto de desarrollar en forma económica software que sea **fiable** y que trabaje con eficiencia en máquinas reales” (Friz Bauer, 1972).
- ✚ “Aplicación **práctica** del conocimiento científico en el diseño y construcción de programas de computadora y la **documentación** asociada y requerida para el desarrollo, operación y **mantenimiento** del programa” (B. Bohem, 1976)
- ✚ “Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, aplicación de la ingeniería al software (**estándar** - IEEE, 1993)

EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

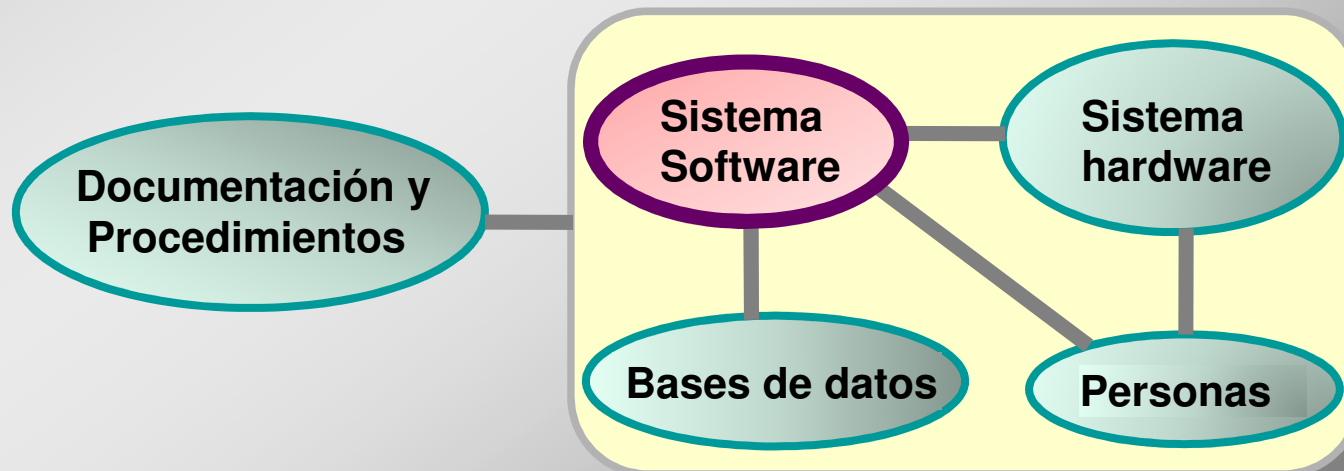
Terminología usada en ingeniería del software

✚ Sistema

Conjunto de elementos relacionados entre sí y con el medio, que forman una unidad o un todo organizativo

✚ Sistema basado en computadora

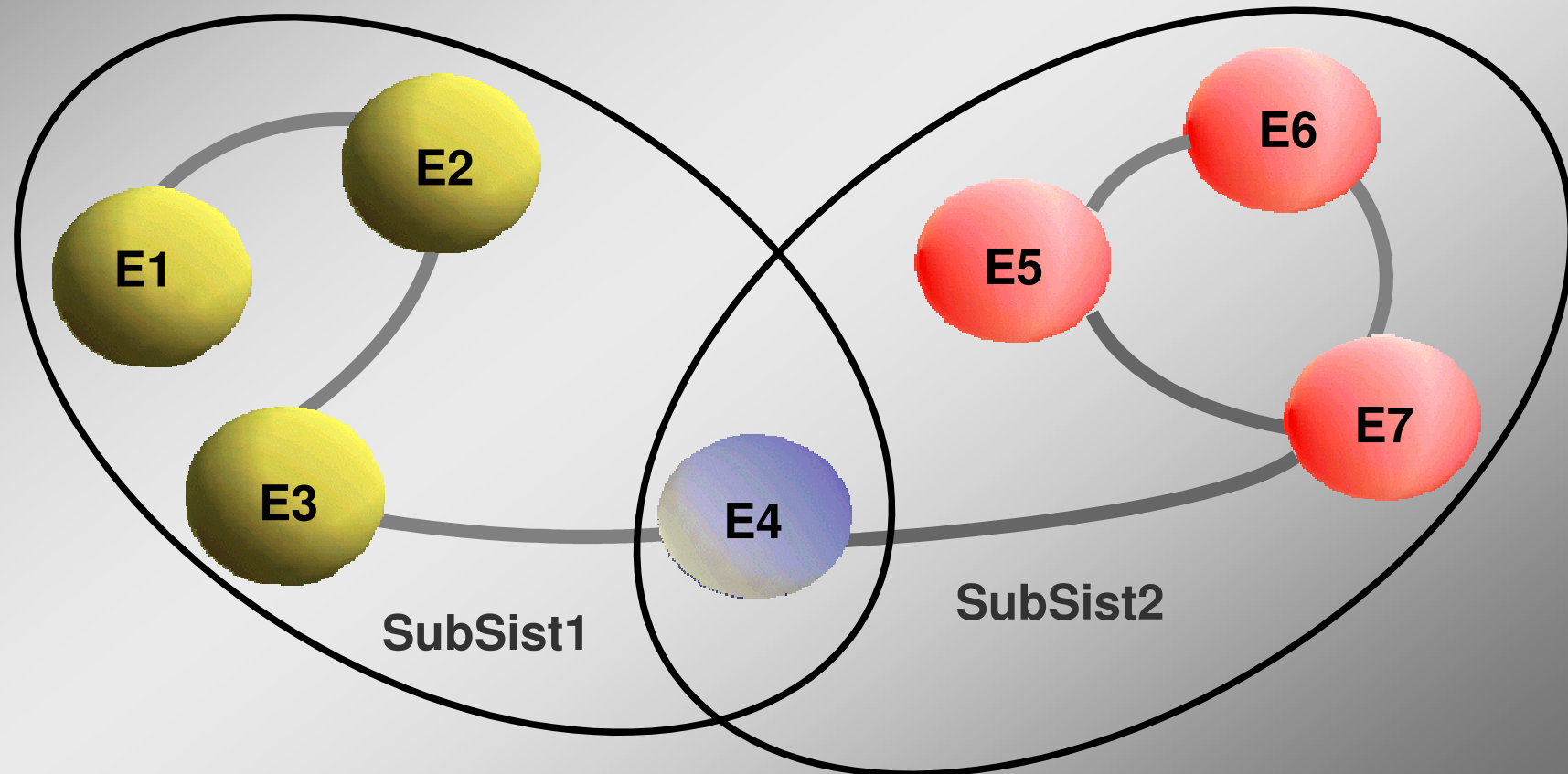
Conjunto o disposición de elementos organizados para cumplir una meta predefinida al procesar información



EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

✚ Sistema software

Conjunto de piezas o elementos software relacionados entre si y organizados en subsistemas



EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

✚ Modelos

Representación de un sistema en un determinado lenguaje: De un mismo sistema se pueden construir muchos modelos

✚ Principios

Elementos adquiridos mediante el conocimiento, que definen las características que debe poseer un modelo para ser una representación adecuada de un sistema

✚ Herramientas

Instrumentos que permiten la representación de modelos

✚ Técnicas

Modo de utilización de las herramientas

EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

🚦 Heurísticas

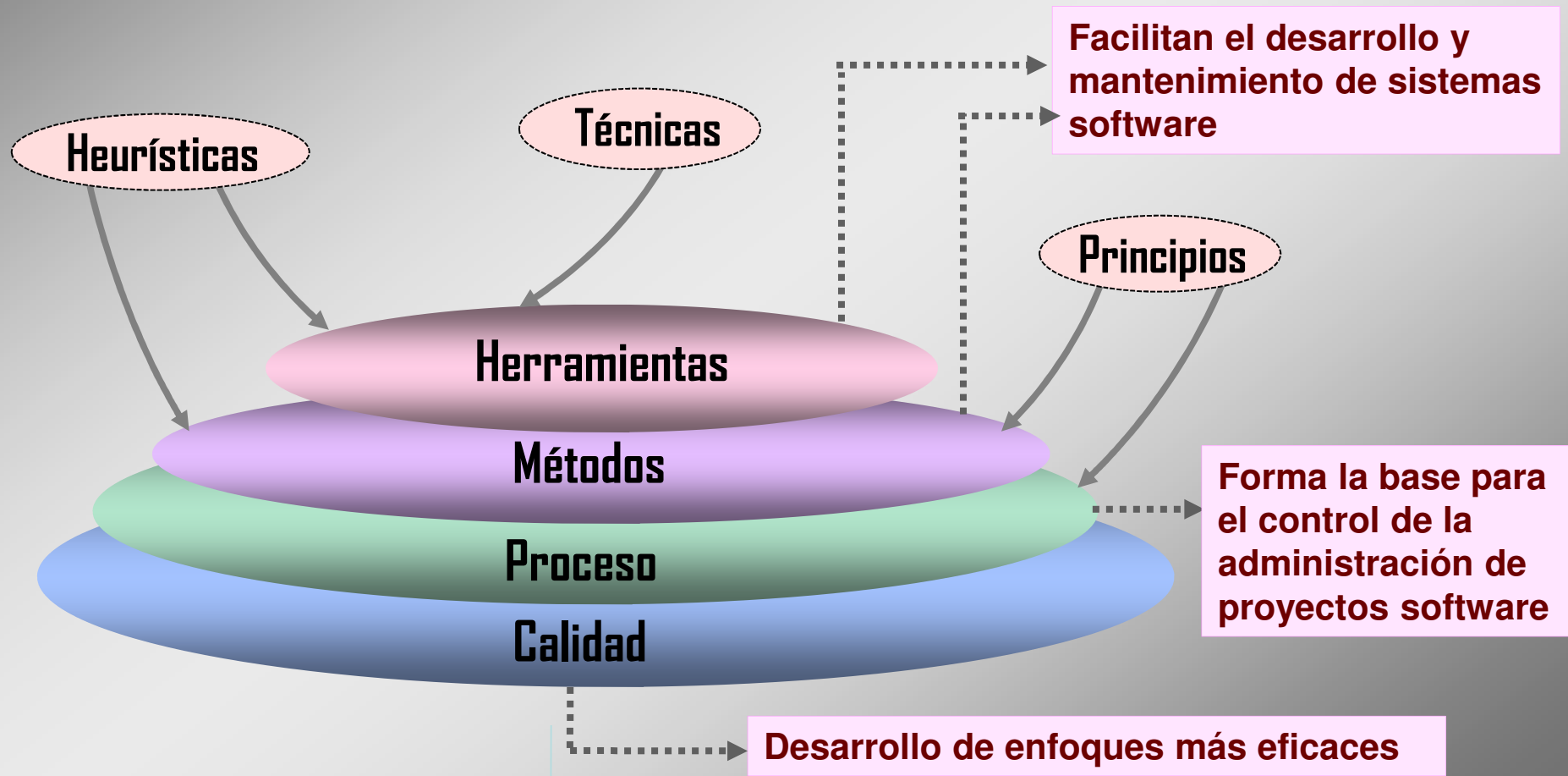
Conjunto de reglas empíricas, que al ser aplicadas producen modelos que se adecuan a los principios

🚦 Métodos

Secuencia de actividades para la obtención de un producto (modelo), que describen cómo usar las herramientas y las heurísticas

EL CONCEPTO DE INGENIERÍA DEL SOFTWARE

Resumen



EL PROCESO DE DESARROLLO DEL SOFTWARE

Modelos de ciclo de vida del software:

- El modelo en cascada

- Desarrollo incremental

- Ingeniería del software orientada a la reutilización

Modelos orientados al cambio

- Prototipo del sistema

- Entrega incremental

- Proceso unificado

- Métodos ágiles

EL PROCESO DE DESARROLLO DEL SOFTWARE

✚ Proceso de desarrollo o ciclo de vida del software

Estrategia que define la división y ubicación temporal de las etapas (o actividades) que se realizan durante el desarrollo del software

✚ Etapas

Se caracterizan por las tareas que se realizan y por el producto (documento, artefacto, ...) que se obtiene

✚ Modelos (o paradigmas) de ciclo de vida del software

Son representaciones abstractas (simplificadas) del proceso de desarrollo del software

Existen diversos modelos según las etapas que se consideren, posición relativa de las mismas y las tareas a realizar en ellas

EL PROCESO DE DESARROLLO DEL SOFTWARE

Etapas principales

Especificación de requisitos

Análisis del problema a resolver. Documento en el que se indica la funcionalidad del software y las restricciones sobre su operación

Diseño

Búsqueda de una solución. Descripción de los componentes, sus relaciones y funciones

Implementación

Traducción del diseño a un lenguaje de programación entendible por una máquina. El código

Validación

Revisiones de todo lo que se ha obtenido junto con la prueba del código

Mantenimiento y evolución

Reparación de fallos y adaptación a nuevos entornos

Planificación

Estimación del tiempo y de los costes de desarrollo

EL PROCESO DE DESARROLLO DEL SOFTWARE

Modelos de ciclo de vida del software

✚ El modelo en cascada (o ciclo de vida clásico)

Considera las actividades de especificación, desarrollo, validación y evolución, y las representa como fases separadas del proceso

✚ Desarrollo incremental

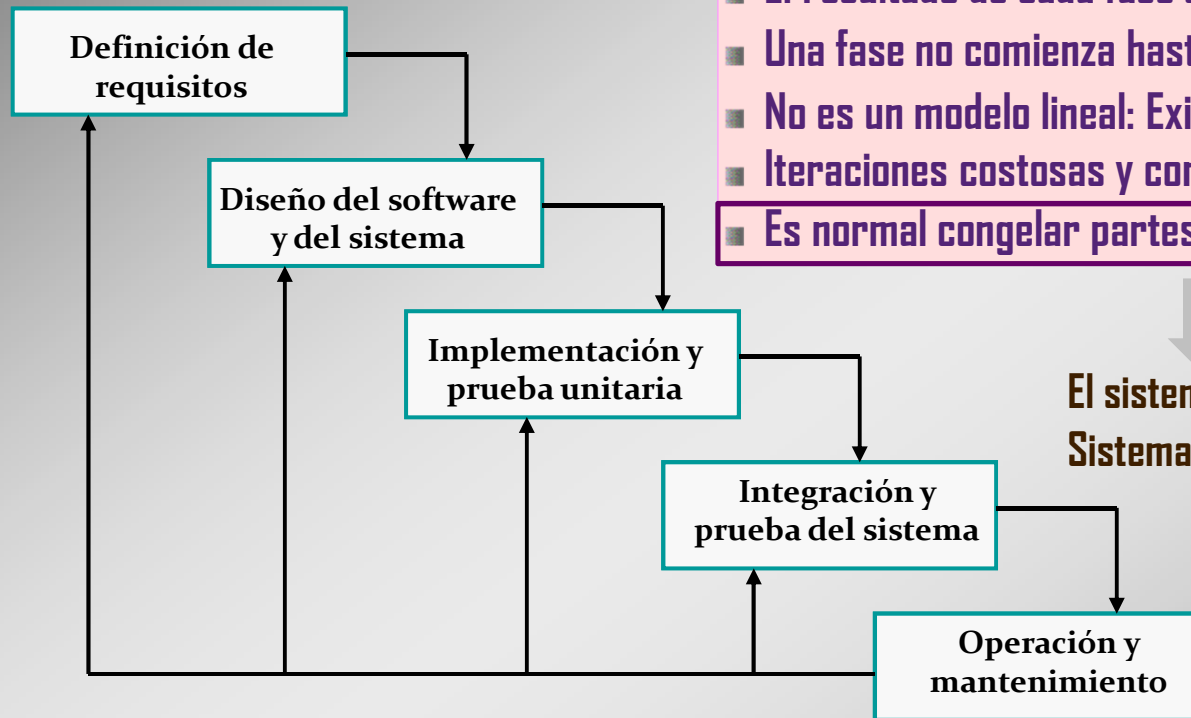
Intercala actividades de especificación, desarrollo y validación. El sistema se desarrolla como una serie de versiones (incrementos) y cada versión añade funcionalidad a la anterior

✚ Ingeniería del software orientada a la reutilización

Se basa en la existencia de componentes reusables. El proceso de desarrollo se centra en integrar esos componentes en vez de desarrollarlos desde el principio

EL PROCESO DE DESARROLLO DEL SOFTWARE

El modelo en cascada



Características

- El resultado de cada fase es uno o más documentos aprobados
- Una fase no comienza hasta que finaliza la anterior
- No es un modelo lineal: Existe retroalimentación de una fase a otra
- Iteraciones costosas y con rediseño significativo
- Es normal congelar partes del desarrollo

El sistema no hará lo que el usuario quiere
Sistemas mal estructurados

Ventaja

La producción de documentación en cada fase

Problema

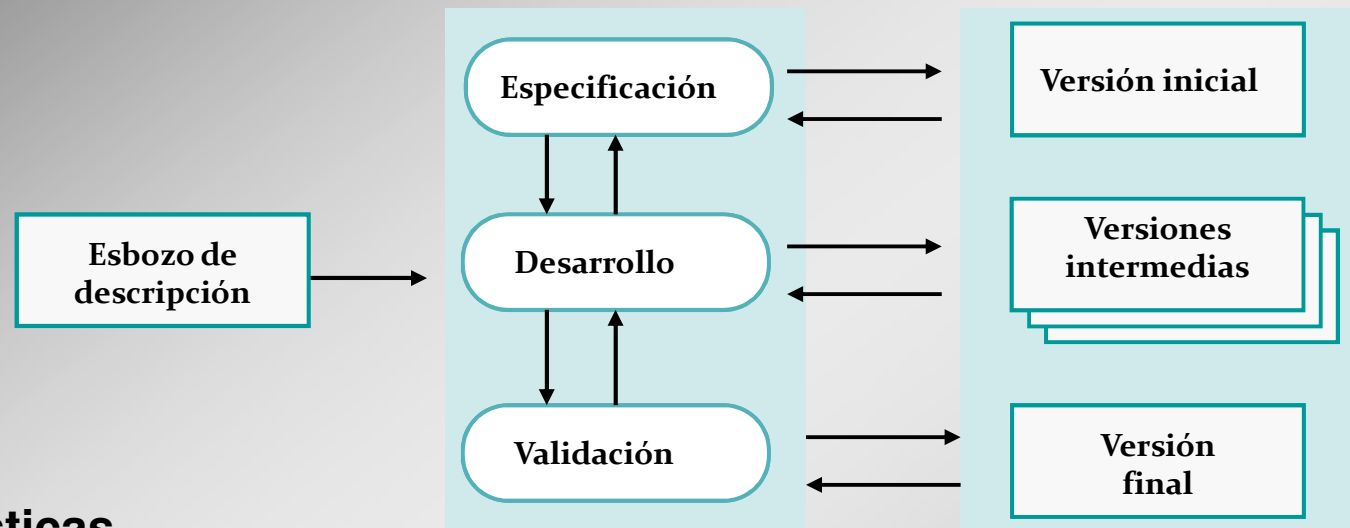
Inflexible división del proyecto en distintas etapas

Debe usarse

Cuando los requisitos sean bien comprendidos y sea poco probable que cambien durante el desarrollo del sistema

EL PROCESO DE DESARROLLO DEL SOFTWARE

Desarrollo incremental



Características

- Se desarrolla una implementación inicial y se refina por medio de muchas versiones
- La especificación, el desarrollo y la validación están intercaladas con rápida retroalimentación
- Rara vez se trabaja por adelantado una solución completa: Se avanza hacia ella y se retrocede si hay errores
- Resulta más barato y fácil realizar cambios en el software conforme se diseña
- Cada incremento incorpora algunas de las funciones que necesita el cliente

El cliente puede evaluar el desarrollo del sistema en una etapa relativamente temprana
Si el incremento actual no cumple lo requerido, sólo se cambia dicho incremento

EL PROCESO DE DESARROLLO DEL SOFTWARE

Ventajas

- ↘ Se reduce el costo de adaptar los requisitos cambiantes del cliente
- ↘ Es más sencillo obtener retroalimentación del cliente sobre el trabajo de desarrollo que se realizó
- ↘ Es más rápida la entrega e implementación de software útil al cliente

Problemas

- ↘ El proceso no es visible
- ↘ La estructura del sistema se degrada conforme se tienen nuevos incrementos



Son particularmente agudos

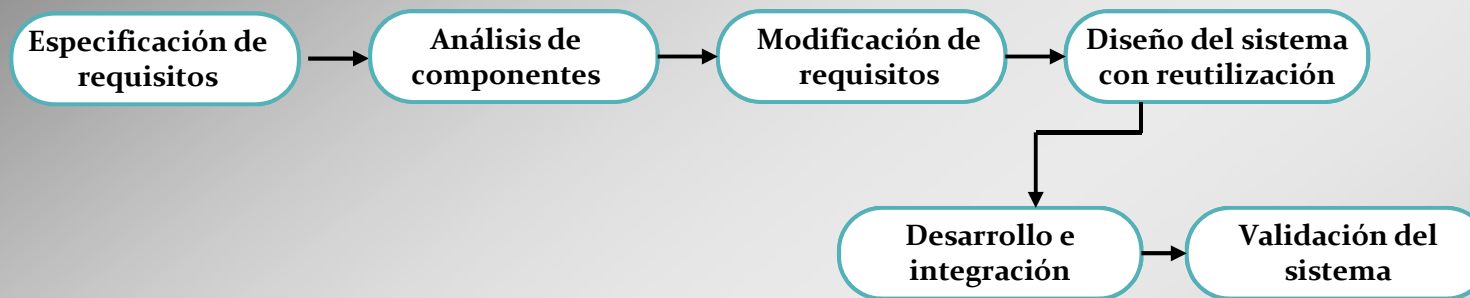
Para sistemas grandes, complejos y de larga duración, donde diversos equipos desarrollan diferentes partes del sistema

Es mejor que el modelo en cascada

Para la mayoría de los sistemas empresariales, de comercio electrónico y personales

EL PROCESO DE DESARROLLO DEL SOFTWARE

Ingeniería del software orientada a la reutilización



Análisis de componentes

Dada la especificación de requisitos, se buscan los componentes que implementan esa especificación

Modificación de requisitos

Se analizan los requisitos usando la información de los componentes encontrados

Diseño del sistema con reutilización

Se diseña el marco conceptual del sistema o se reutiliza un marco existente

Desarrollo e integración

Se desarrolla el software que no se halla podido adquirir y se integran los componentes para crear el nuevo sistema

EL PROCESO DE DESARROLLO DEL SOFTWARE

Ventajas

- ↘ Disminuye la cantidad de software que hay que desarrollar
- ↘ Disminuye los costes y los riesgos
- ↘ Conduce a entregas más rápidas

Problemas

- ↘ Se pierde control sobre la evolución del sistema cuando las versiones de los componentes no están bajo el control de la organización que los usa
- ↘ Los compromisos de requisitos son inevitables



Puede conducir

A un sistema que no cubra las necesidades reales de los usuarios

EL PROCESO DE DESARROLLO DEL SOFTWARE

Modelos orientados al cambio

Para reducir los costes del **rehacer**

- ✚ Evitar el cambio

En el proceso de desarrollo se incluyen actividades que anticipan posibles cambios antes de que se requieran

- ✚ Tolerancia al cambio

El proceso de desarrollo se diseña de manera que los cambios se ajusten con un coste relativamente bajo

Formas de enfrentar el **cambio**

- ✚ Prototipo del sistema (prototipo desechable)

Se desarrolla rápidamente una versión del sistema, o una parte, para comprobar los requisitos del cliente y la factibilidad de algunas decisiones de diseño

- ✚ Entrega incremental (prototipo evolutivo)

Los incrementos del sistema se entregan al cliente para su comentario y experimentación, lo que apoya tanto el hecho de evitar el cambio como de tolerarlo

EL PROCESO DE DESARROLLO DEL SOFTWARE

Prototipo del sistema (prototipo desechable)

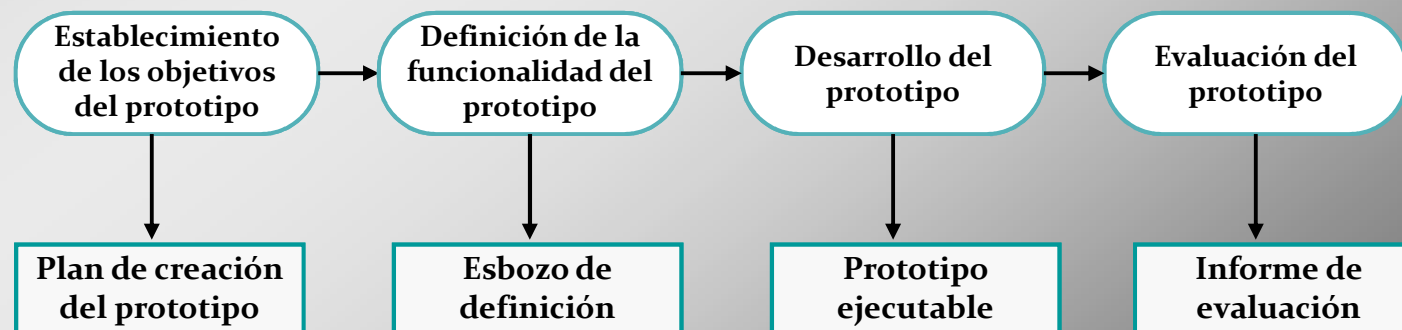
Prototipo

Versión inicial del software que se utiliza para demostrar los conceptos, probar las opciones de diseño y, de forma general, comprender mejor el problema y sus posibles soluciones

Se usa para contribuir a anticipar cambios en:

- ↗ El proceso de ingeniería de requisitos: Ayuda a la selección y validación de requisitos del sistema
- ↗ El proceso de diseño de sistemas: Ayuda a buscar soluciones específicas y al diseño de interfaces de usuario

Modelo para el desarrollo de prototipos



EL PROCESO DE DESARROLLO DEL SOFTWARE

Ventajas

- ↘ Permite obtener nuevas ideas para requisitos y descubrir fortalezas y debilidades del software
- ↘ Permite comprobar la factibilidad de un diseño propuesto
- ↘ Participación del usuario final en el desarrollo de interfaces gráficas

Problemas

- ↘ Que el prototipo no se utilice de la misma forma que el sistema final
- ↘ Que el tiempo de capacitación durante la evaluación del prototipo sea insuficiente
- ↘ Que se entreguen prototipos cuando existan demoras en la entrega de la versión final del software

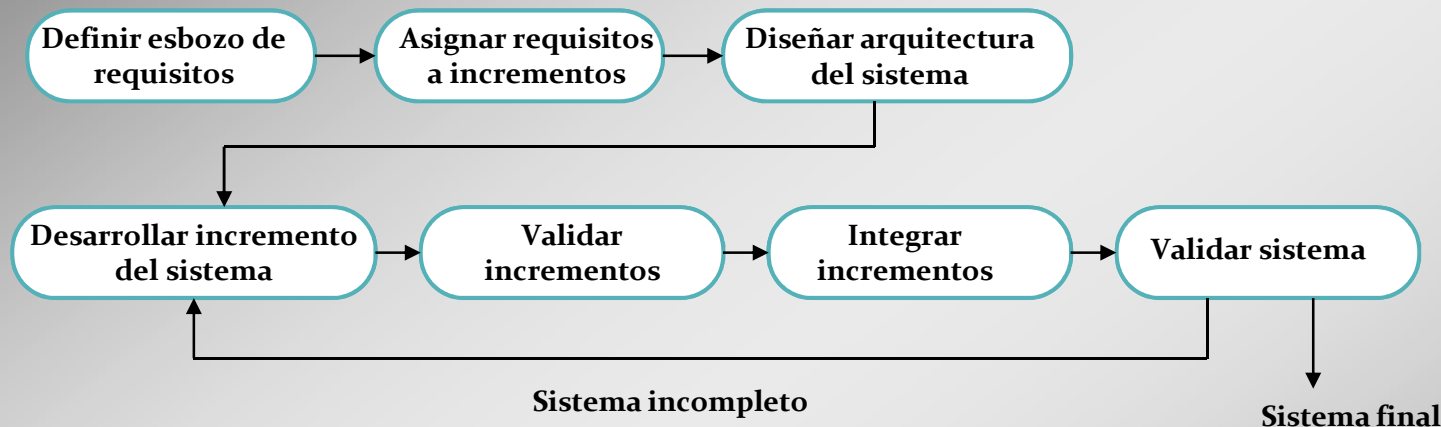


No es aconsejable

- ❖ Puede ser imposible corregir el prototipo para cubrir requisitos no funcionales
- ❖ El cambio rápido significa que el prototipo no está documentado
- ❖ Es posible que los cambios degraden la estructura del sistema
- ❖ Se hacen más flexibles los estándares de calidad de la organización

EL PROCESO DE DESARROLLO DEL SOFTWARE

Entrega incremental (prototipo evolutivo)



Ventajas

- Los clientes pueden usar los primeros incrementos para adquirir experiencia que informe sobre sus requisitos para posteriores incrementos del sistema
- Los clientes deben esperar hasta la entrega completa del sistema, antes de ganar valor del mismo
- Debe ser relativamente sencillo incorporar cambios al sistema
- Los servicios más importantes del sistema reciben mayores pruebas

EL PROCESO DE DESARROLLO DEL SOFTWARE

Problemas

- ⚡ La mayoría de los sistemas requieren de una serie de recursos que se usan para diferentes partes del sistema
- ⚡ El desarrollo iterativo resulta complicado cuando se diseña un sistema de reemplazo
- ⚡ En los procesos iterativos la especificación se desarrolla en conjunto con el software

No es el mejor enfoque para algunos tipos de sistemas

- ⚡ Sistemas muy grandes que son desarrollados por equipos que trabajan en diferentes ubicaciones
- ⚡ Sistemas embebidos en los que el software depende del desarrollo del hardware
- ⚡ Sistemas críticos en los que se analizan todos los requisitos para comprobar las interacciones que comprometen la seguridad y protección del sistema

EL PROCESO DE DESARROLLO DEL SOFTWARE

Proceso unificado

Es un modelo híbrido que apoya la creación de prototipos y la entrega incremental

Identifica cuatro fases en el proceso de desarrollo software

✚ Concepción

El objetivo es establecer un caso empresarial para el sistema: Se identifican las entidades externas que interactuarán con el sistema y se valora la aportación del sistema hacia la empresa

✚ Elaboración➡ Modelo de requisitos para el sistema

Los objetivos son desarrollar la comprensión del problema, establecer un marco conceptual arquitectónico para el sistema, diseñar el plan del proyecto e identificar los riesgos clave

✚ Construcción➡ Sistema software funcionando y la documentación relacionada

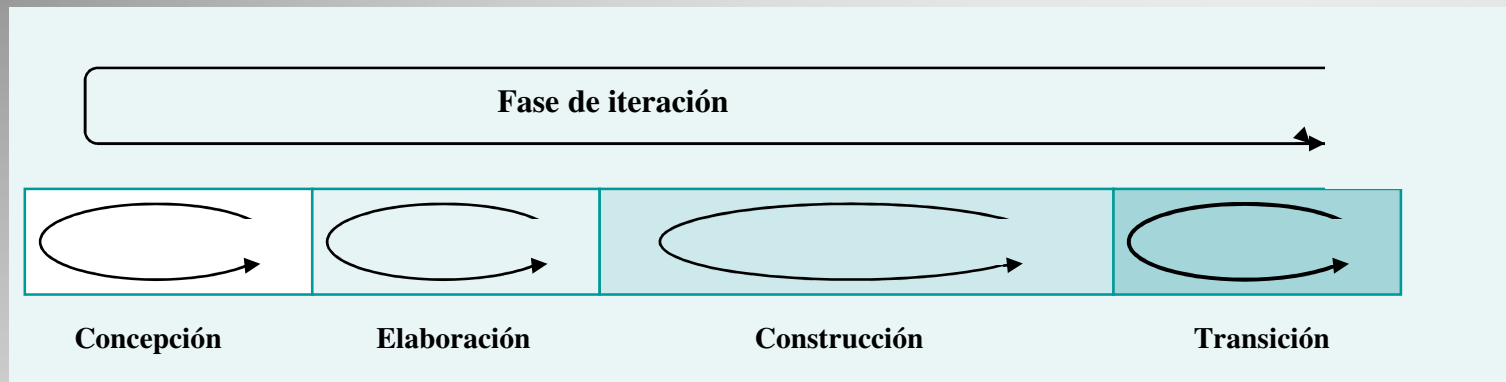
Incluye diseño, programación, integración de las partes del sistema que se han desarrollado en paralelo y pruebas del sistema

✚ Transición➡ Sistema software documentado que funcione en un entorno operacional

Se interesa por el cambio del sistema desde los desarrolladores hacia los usuarios y por ponerlo a funcionar en un ambiente real

EL PROCESO DE DESARROLLO DEL SOFTWARE

La iteración se apoya en dos formas



- Cada fase puede ser iterativa, con los resultados desarrollados incrementalmente
- Todo el conjunto de fases puede expresarse de manera incremental

EL PROCESO DE DESARROLLO DEL SOFTWARE

Se describe desde tres perspectivas

+ Dinámica

Muestra las fases del modelo a través del tiempo

+ Estática

Se enfoca en las actividades (flujos de trabajo) que tienen lugar durante el proceso de desarrollo

Flujos de trabajo de proceso

- Modelado del sistema
- Requisitos
- Análisis y diseño
- Implementación
- Pruebas
- Despliegue

Flujos de trabajo de apoyo

- Administración de la configuración y del cambio
- Administración del proyecto
- Entorno

+ Práctica

Describe las buenas prácticas de ingeniería del software en el desarrollo de sistemas

Prácticas que se recomiendan

- Desarrollo de software de manera iterativa
- Gestión de requisitos
- Uso de arquitecturas basadas en componentes
- Modelado visual del software
- Verificar la calidad del software
- Controlar los cambios

EL PROCESO DE DESARROLLO DEL SOFTWARE

Métodos ágiles

Conjunto de métodos que enfatizan el enfoque iterativo, la adaptabilidad del proceso y la colaboración, reduciendo al mínimo la documentación y los procedimientos

Características

- ◆ La medida del progreso es el software desarrollado y funcional
- ◆ La documentación es la del código junto con el diseño de dicho código
- ◆ No hay una estructura organizativa rígida en el equipo de desarrollo
- ◆ El proceso debe ser capaz de cambiar para adaptarse a las necesidades o requisitos nuevos que aparezcan en las iteraciones

¿Cuándo son apropiados?

Funcionan bien para equipos pequeños y son útiles cuando se necesita que el proceso se adapte rápidamente

EL PROCESO DE DESARROLLO DEL SOFTWARE

¿Cómo surgen?

Snowbird, Utah (USA), febrero 2001, 17 expertos se preguntaron

¿Por qué tantos proyectos de desarrollo de software no se terminan a tiempo, cuestan más de lo presupuestado originalmente, tienen problemas serios de calidad y generan menor valor del esperado?

Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Kent **Beck** Mike **Beedle**
Arie **van Bennekum** Alistar **Cockburn**
Ward **Cunningham** Martin **Fowler**
James **Grenning** Jim **Highsmith**
Andrew **Hunt** Ron **Jeffries**
Jon **Kern** Brian **Marick**
Robert C. **Martin** Steven **Mellor**
Ken **Schwaber** Jeff **Sutherland**
Dave **Thomas**