



ugr

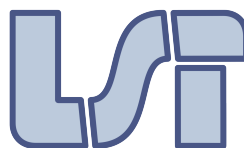
Universidad
de Granada

FUNDAMENTOS DE INGENIERÍA DEL SOFTWARE1
GRADO EN INGENIERÍA INFORMÁTICA

Resumen del temario

Autor

Carlos Sánchez Páez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2017-2018

Índice

1. Tema 1. Introducción a la Ingeniería del Software	2
1.1. El producto Software	2
1.1.1. Definición de Software	2
1.1.2. Tipos de software	2
1.1.3. Características principales	2
1.1.4. Proceso de producción	3
1.1.5. Problemas en el desarrollo	5
1.2. Concepto de Ingeniería del Software	6
1.2.1. Definiciones de la Ingeniería del Software	6
1.2.2. Terminología usada en Ingeniería del Software	6
1.3. Proceso de desarrollo del software	8
1.3.1. Concepto de proceso de desarrollo	8
1.3.2. Modelo general de proceso	9
1.3.3. Tipos de modelos de proceso	10
1.3.4. Proceso unificado	13
1.3.5. Desarrollo Ágil	15
2. Tema 2. Ingeniería de Requisitos	16
2.1. Introducción	16

Índice de figuras

1. Etapas del proceso de producción del software.	3
2. Esfuerzo invertido en las distintas etapas.	4
3. Esfuerzo invertido en las distintas etapas del mantenimiento.	4
4. Problemas en la comunicación.	5
5. Impacto del cambio en las distintas etapas.	5
6. Sistema basado en computadora.	6
7. Sistema Software.	7
8. Modelo.	7
9. Estructura del proceso.	9
10. Tipos de flujos de proceso.	9
11. Modelo en cascada.	10
12. Modelo incremental.	11
13. Modelo de prototipos	12
14. Modelo en espiral de Boehm	13
15. Proceso unificado.	13
16. Ejemplo de Proceso Unificado	14
17. Resultados del informe <i>CHAOS</i>	16

1. Tema 1. Introducción a la Ingeniería del Software

1.1. El producto Software

1.1.1. Definición de Software

El software se puede definir de varias formas:

- Programa o conjunto de programas de cómputo que incluye datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.
- Transformador de información, para lo que adquiere, gestiona, modifica, produce o transmite esa información.

1.1.2. Tipos de software

Podemos clasificar el software en varios tipos:

1. Por *campo de aplicación*:

- a) Software de **sistemas**.
- b) Software de **aplicaciones**.
- c) Software de **programación**.

2. Por *tipo de licencia*:

- a) Según derechos de autor
 - 1) Software de **código abierto**.
 - 2) Software de **código cerrado**.
 - 3) Software de **dominio público**.
- b) Según su destinatario
 - 1) Usuario final (software **hecho a medida**).
 - 2) Para distribución (software **genérico**).

1.1.3. Características principales

- 1. El software es un producto lógico:
 - a) No se fabrica, sino que se desarrolla.
 - b) No se estropea, sino que se deteriora.
- 2. Crea modelos de la realidad.
- 3. Está formado por múltiples piezas que deben encajar perfectamente.

1.1.4. Proceso de producción

El proceso de producción del software está formado por las siguientes etapas:

1. **Definición.** Debemos precisar lo que queremos desarrollar. Para ello debemos realizar varias tareas:
 - Ingeniería de Sistemas.
 - Ingeniería de Requisitos.
 - Planificación de proyectos.
2. **Construcción.** Hemos de determinar cómo desarrollaremos el software. Depende de:
 - Diseño del software.
 - Generación del código.
 - Prueba del software.
3. **Evolución.** Consiste en precisar las partes del software que cambiarán.
 - Corrección.
 - Adaptación.
 - Mejora.
 - Prevención.

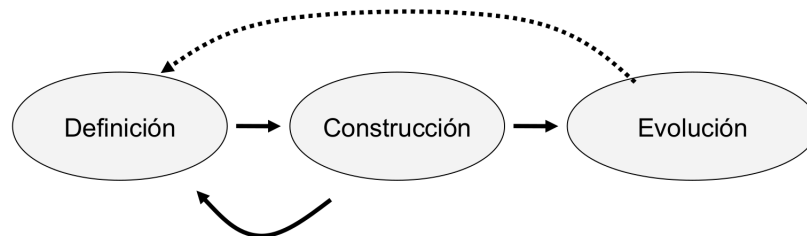


Figura 1: Etapas del proceso de producción del software.

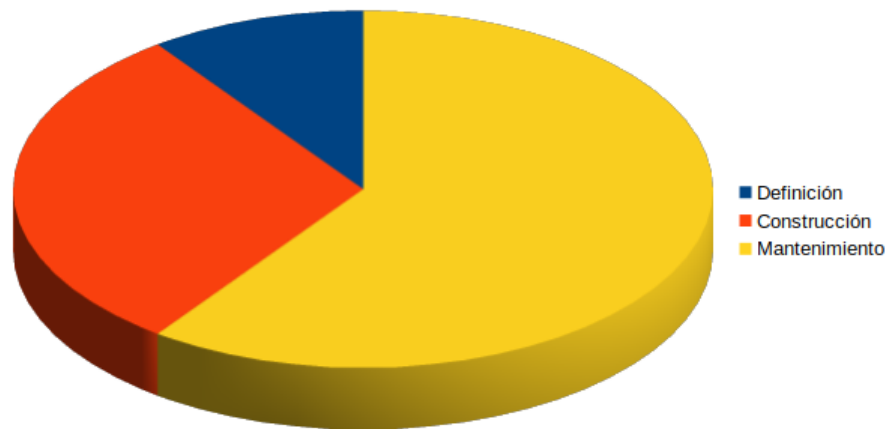


Figura 2: Esfuerzo invertido en las distintas etapas.

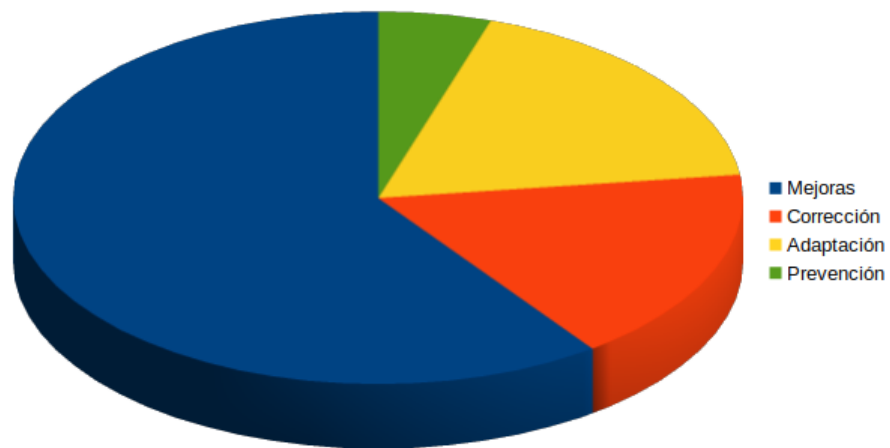


Figura 3: Esfuerzo invertido en las distintas etapas del mantenimiento.

1.1.5. Problemas en el desarrollo

1. Comunicación entre personas.

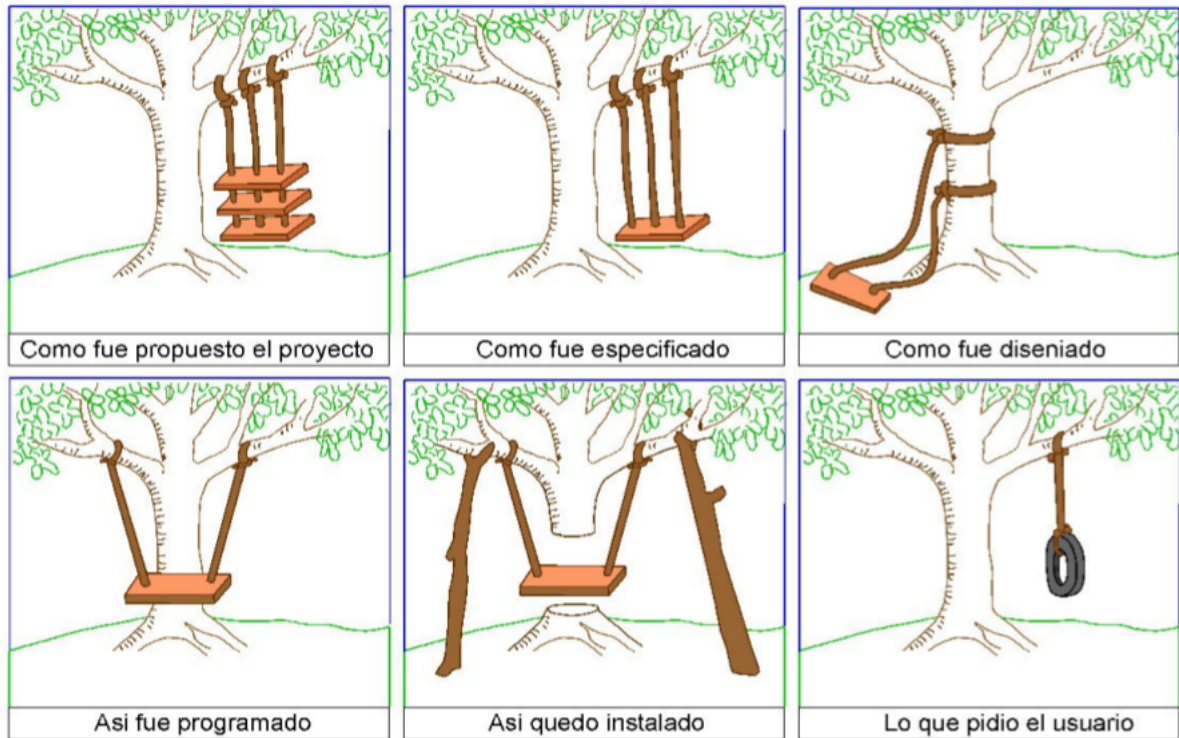


Figura 4: Problemas en la comunicación.

2. Incumplimiento de la **planificación**.

3. Incorporación de **cambios** en etapas avanzadas.

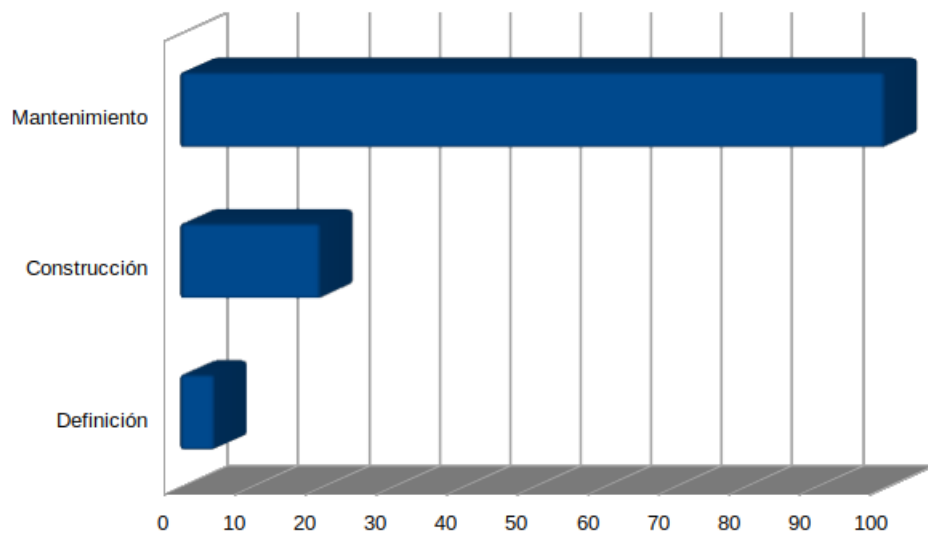


Figura 5: Impacto del cambio en las distintas etapas.

Desastres ocasionados por sistemas software La mayoría de ellos tienen como causa pruebas deficientes, mala documentación, diseños pobres o inexistentes, mal estudio del problema, etc.

1.2. Concepto de Ingeniería del Software

La *Ingeniería del Software* surgió por varias necesidades:

- Mal funcionamiento (calidad).
- Mantenimiento del software existente.
- Demanda creciente del nuevo software.
- Adaptación a las nuevas tecnologías.
- Incremento de la complejidad.

1.2.1. Definiciones de la Ingeniería del Software

1. Establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales.
2. 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.
3. Estudio de los principios y métodos para el desarrollo y mantenimiento de sistemas software.
4. Aplicación de un enfoque sistémico, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, aplicación de la ingeniería al software.
5. Conjunto de teorías, métodos e instrumentos (tecnológicos y organizativos) que permitan construir sistemas software con las características de calidad deseadas.
6. Disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación hasta el mantenimiento del sistema después de su puesta en operación.

1.2.2. 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.

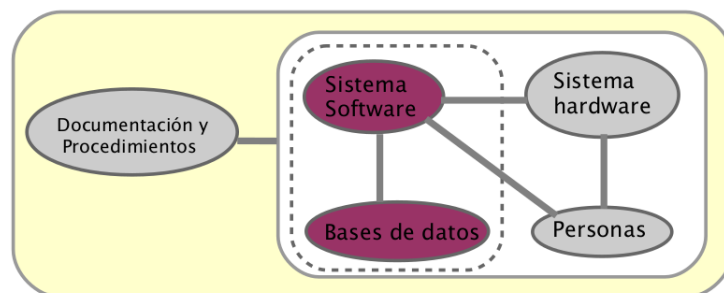


Figura 6: Sistema basado en computadora.

- **Sistema Software.** Conjunto de piezas o elementos software relacionados entre sí y organizados en subsistemas.

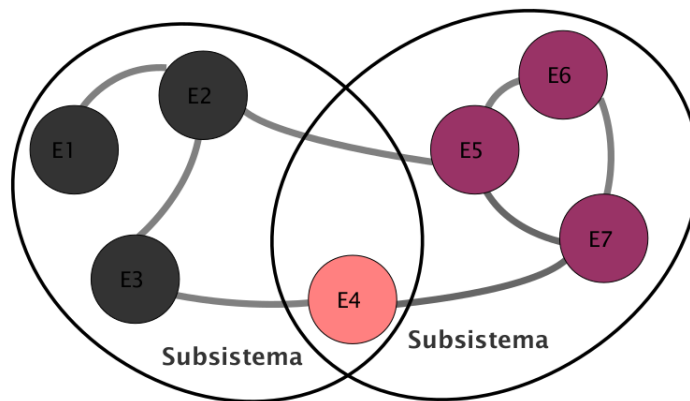


Figura 7: Sistema Software.

- **Modelo.** Representación de un problema en un determinado lenguaje. De un mismo problema se pueden construir muchos modelos.

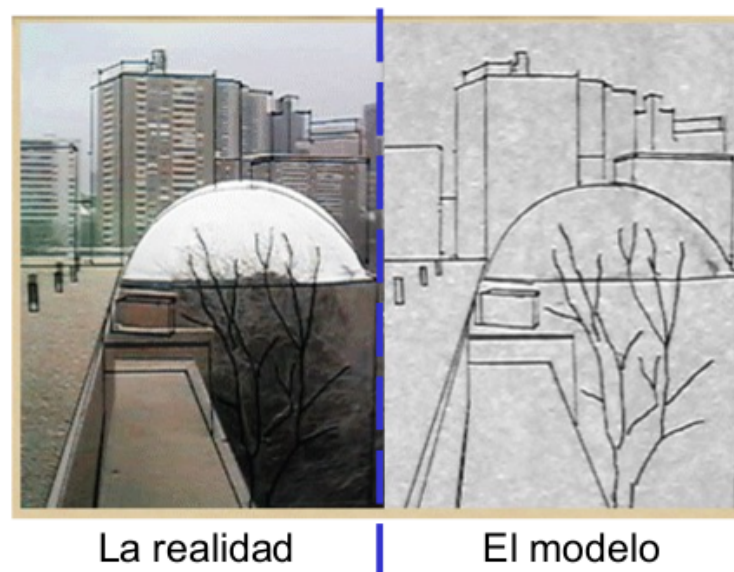


Figura 8: Modelo.

- **Principio.** Elementos que son adquiridos mediante el conocimiento. Determinan las características que debe poseer un modelo para ser una representación adecuada de un sistema.
- **Herramienta.** Instrumentos que permiten la representación de modelos.
- **Técnica.** Modo de utilización de las herramientas.
- **Heurísticas.** Conjunto de reglas empíricas que al ser aplicadas producen modelos que se adecuan a los principios. Ejemplo: *No usar materiales flexibles para representar la maqueta de un edificio.*

- **Proceso.** Estructura que debe establecerse para la obtención eficaz de un producto de Ingeniería.
- **Método.** Proporciona la experiencia técnica para elaborar el producto software. Se basa en principios fundamentales e incluye actividades de modelado.

1.3. Proceso de desarrollo del software

1.3.1. Concepto de proceso de desarrollo

- **Proceso de desarrollo del software.** Conjunto de *actividades, acciones y tareas* que se realizan cuando va a crearse un producto o sistema software.
- **Actividad.** Busca el logro de objetivos amplios e independientes del tipo de aplicación a desarrollar y su complejidad.
- **Acción.** Conjunto de tareas que elaboran un producto importante como resultado.
- **Tarea.** Objetivo pequeño y bien definido que produce un resultado tangible.

Las actividades que se realizan pueden ser de varios tipos:

1. **Estructurales.** Se dedican a obtener el producto:

- a) **Comunicación.** Colaboración con el cliente para entender los objetivos y requisitos del proyecto.
- b) **Planificación.** Definición del plan de proyecto en el que se describen los riesgos probables, recursos adquiridos y productos obtenidos a la vez que se programan las actividades, acciones y tareas.
- c) **Modelado.** Representación mediante modelos del sistema propuesto junto con la solución o soluciones apropiadas.
- d) **Construcción.** Generación de código y su prueba.
- e) **Despliegue.** Entrega al consumidor y evaluación por parte de éste, lo que sirve como retroalimentación para el equipo de desarrollo.

2. **Sombrilla.** Se aplican a lo largo de todo el proceso. Se dedican a:

- a) Seguimiento y control del proyecto.
- b) Administración del riesgo.
- c) Aseguramiento de la calidad.
- d) Revisiones técnicas.
- e) Mediciones de parámetros del proceso.
- f) Administración de la configuración.
- g) Administración de la reutilización.
- h) Preparación y producción del producto de trabajo.

1.3.2. Modelo general de proceso

- **Estructura del proceso.** Cada una de las actividades, acciones y tareas se encuadran dentro de una estructura que define su relación con el proceso y entre ellas.

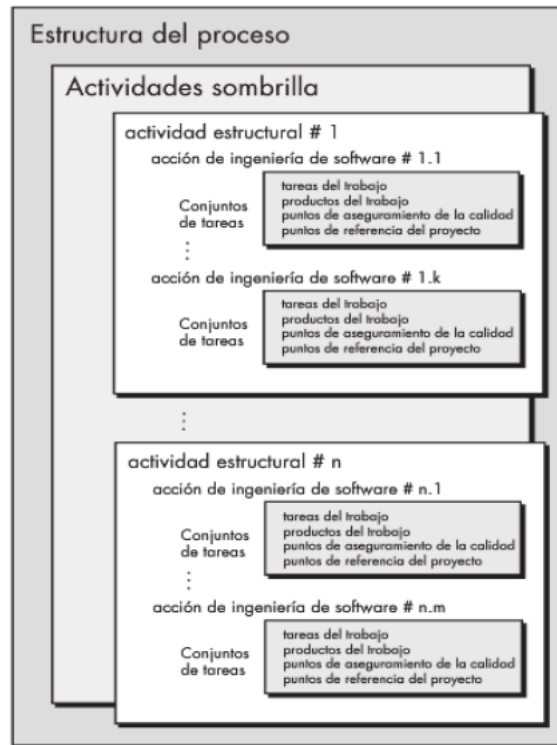
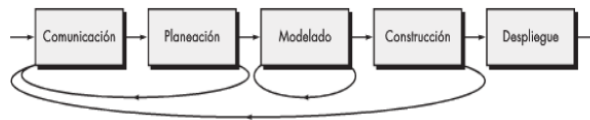


Figura 9: Estructura del proceso.

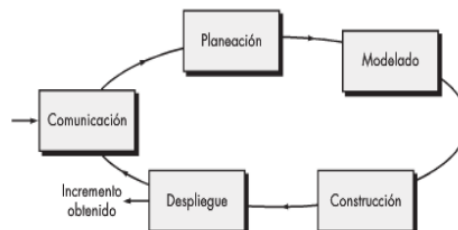
- **Flujo del proceso.** Describe la forma en la que se organizan las actividades estructurales, acciones y tareas en los procesos con respecto a la secuencia y el tiempo.



(a) Flujo de proceso lineal.



(b) Flujo de proceso iterativo.



(c) Flujo de proceso evolutivo.

Figura 10: Tipos de flujos de proceso.

■ **Acciones y tareas de las actividades estructurales.**

1. **Obtención de requisitos.** Obtención de información respecto a la acción que debe realizar el software.
2. **Estimación y planificación del proyecto.** Estimar el tiempo y los costes del desarrollo del software.
3. **Análisis de requisitos.** Documento en el que se especifica lo que debe hacer el sistema software.
4. **Diseño.** Búsqueda de la solución. Descripción de los componentes, sus relaciones y funciones que le dan solución al problema.
5. **Implementación.** Traducción del diseño a un lenguaje de programación entendible por una máquina.
6. **Prueba del software.** Revisión y validación del código que se va desarrollando.
7. **Evaluación y aceptación.** Evaluación del producto y aceptación por parte de los interesados en él.
8. **Entrega y asistencia.** Sistema pasa a operar y se ofrece asistencia para su correcto funcionamiento.

1.3.3. Tipos de modelos de proceso

- **Modelo en cascada.** Presenta una estructura secuencial y un flujo lineal. Sin embargo, los proyectos no suelen adecuarse a este modelo, es difícil expresar los requisitos a través de él al principio del proyecto y ofrece poca comunicación con el cliente, ya que hasta el final no hay un ejecutable que se pueda evaluar.

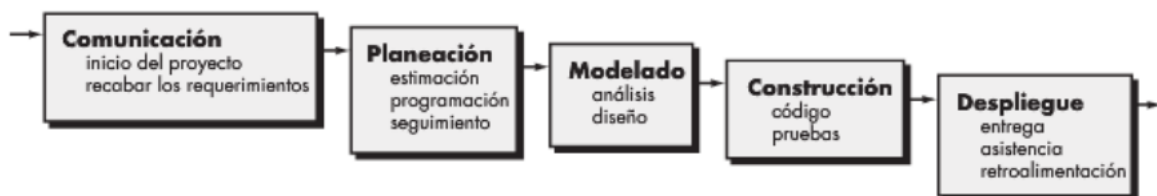


Figura 11: Modelo en cascada.

- **Modelo incremental.** Su estructura es secuencial mientras que el flujo de proceso es lineal y paralelo entre incrementos.

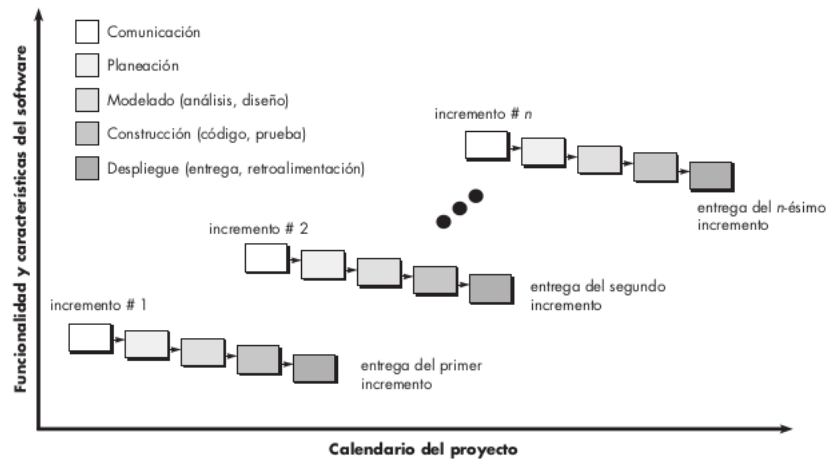


Figura 12: Modelo incremental.

- **Modelo evolutivo.** Es también iterativo. Nace como solución a varios factores, como un tiempo de entrega muy limitado o la necesidad de facilitar la incorporación de cambios. En cada iteración del proceso se obtiene un producto terminado y operativo. Sus características generales son:
 1. Afrontan los riesgos altos (técnicos, de requisitos, etc.) tan pronto como sea posible.
 2. Retroalimentación temprana por parte del cliente.
 3. Manejo de la complejidad (pasos cortos y sencillos).
 4. El conocimiento adquirido durante una iteración de la evolución se puede usar en el resto de iteraciones.
 5. Involucra continuamente al usuario (evaluación, retroalimentación, afinamiento y refinamiento de requisitos, etc.).

Hay dos tipos fundamentales de modelos evolutivos:

1. **Modelo de prototipos.** Un prototipo es una representación limitada de un producto que se utiliza para probar su diseño y comprender mejor el problema y sus posibles soluciones. Los prototipos pueden ser *evolutivos* (productos finales) o *desechables* (usados dentro de otros modelos de proceso).

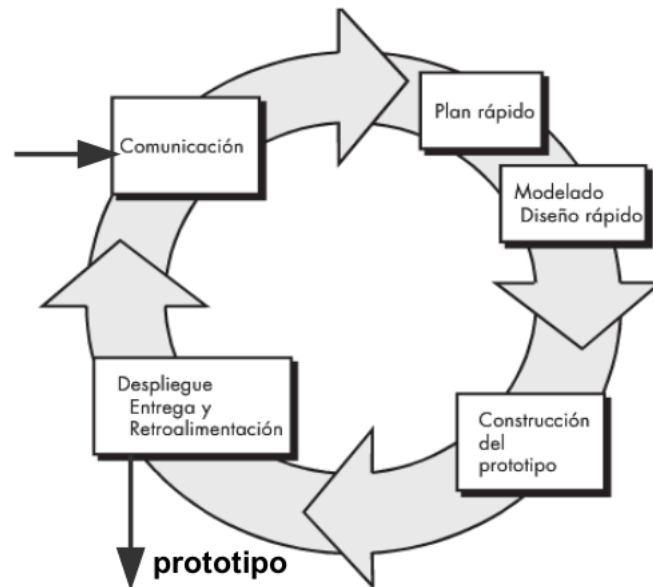


Figura 13: Modelo de prototipos

Este modelo se utiliza para:

1. Facilitar la obtención y validación de requisitos (*desechable*).
2. Estudios de viabilidad (*desechable*).
3. Propuestas de diseños alternativos (*desechable*).
4. En casos muy concretos como producto final (*evolutivo*).

Presentan todas las características de los modelos evolutivos, aunque se les añaden algunos inconvenientes:

- Crear falsas expectativas por parte del cliente (*desechable*).
 - Puede que el prototipo *desechable* se elabore con una metodología ineficiente y ésta se mantenga en el producto final.
- **Modelo en espiral de Boehm.** Además de las características de los procesos iterativos, incluye otras más:
1. Se centra en el análisis de riesgo, construyendo prototipos para su estudio.
 2. La espiral puede continuar una vez que se entregue el momento para llevar a cabo el mantenimiento.
 3. Es adecuado para el desarrollo de sistemas a gran escala.

Sus principales inconvenientes son que no es controlable y que requiere un equipo de desarrollo con gran experiencia en análisis de riesgo.

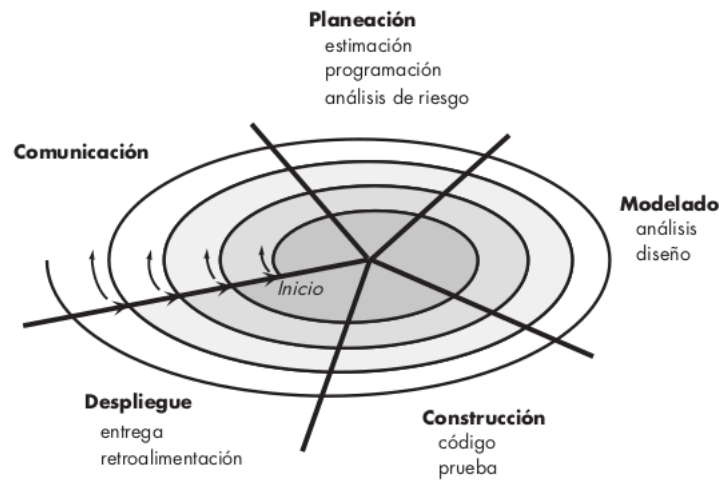


Figura 14: Modelo en espiral de Boehm

1.3.4. Proceso unificado

Es un modelo de proceso evolutivo y compuesto por cuatro fases:

- Inicio o concepción.
- Elaboración.
- Construcción.
- Transición.

Estas etapas se reparten entre las actividades estructurales como podemos ver en el diagrama:

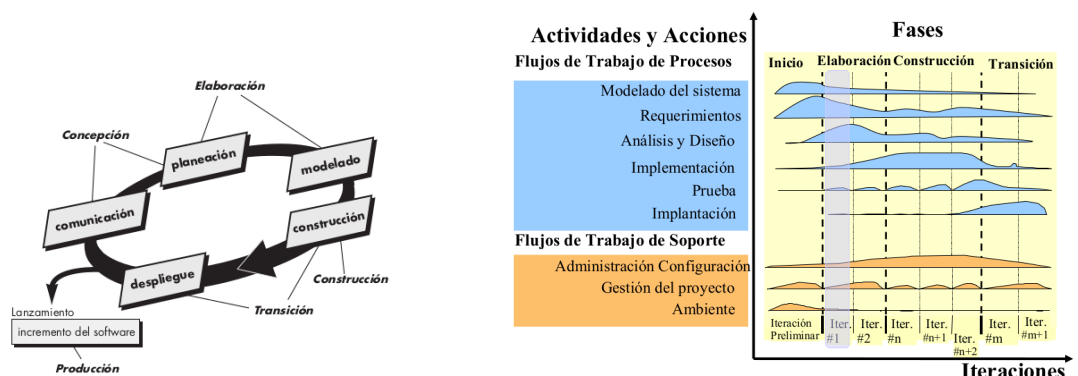


Figura 15: Proceso unificado.

Además de las características de los modelos de proceso evolutivos, el proceso unificado incorpora las siguientes:

1. Es un modelo de proceso adaptable a la complejidad y al tipo de sistema.
2. Está centrado en la arquitectura, mostrando y decidiendo los distintos aspectos arquitectónicos de un sistema software en etapas tempranas. De esta forma pueden servir de base a las posteriores.
3. Está dirigido por casos de uso, desarrollándose uno o varios en cada iteración. En iteraciones tempranas los casos de uso determinarán la arquitectura.

Acciones y tareas en cada fase

- **Inicio.** Agrupa actividades tanto de comunicación del cliente como de planificación. Se propone una arquitectura aproximada para el sistema y se estudian numerosos factores (viabilidad, alcance, riesgos, etc.).
- **Elaboración.** Incluye actividades de comunicación y modelado de la arquitectura básica sobre la que se asentará la fase de construcción.
- **Construcción.** Se completan los modelos de requisitos y se implementan los elementos necesarios para completar el sistema. Según se van terminando los elementos, son probados e integrados al producto final. También se realizan pruebas de aceptación por parte del usuario.
- **Transición.** Consiste en asegurarse de que el sistema cumple con los requisitos especificados (mediante pruebas por parte de los usuarios). Además, se genera el material necesario para lanzar el producto al mercado.

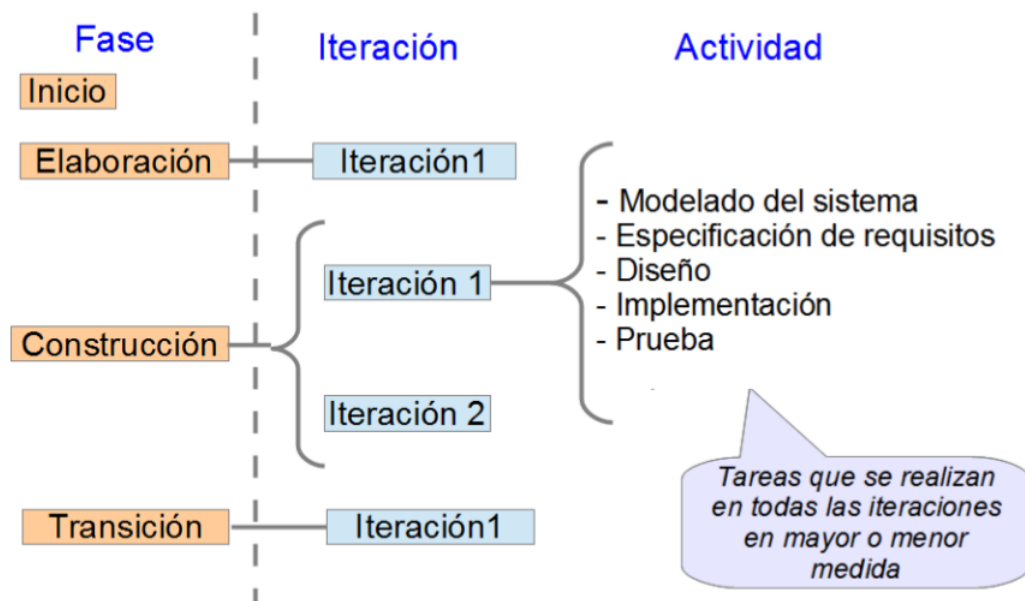


Figura 16: Ejemplo de Proceso Unificado

1.3.5. Desarrollo Ágil

En 2001, diecisiete expertos se preguntaron por qué muchos proyectos generaban menos valor del esperado, no se terminaban a tiempo, tenían problemas de calidad serios, etc. Tras ésto, elaboraron el *Manifiesto el Desarrollo Ágil de Software* con el que intentaban dar solución a estos problemas.

Características del Desarrollo Ágil

1. Es un proceso iterativo e incremental, por lo que es evolutivo.
2. Requiere entregas frecuentes y trabajo en equipo.
3. Establece autonomía en el equipo de desarrollo.
4. Exige revisiones y reuniones retrospectivas frecuentes.

Sus beneficios son que se mejora la productividad y que se manejan mejor los riesgos. Consta de varias técnicas:

- Scrum
- XP (*Extreme Programming*)
- Programación en parejas.
- TDD (*Test Driven Development*)

2. Tema 2. Ingeniería de Requisitos

2.1. Introducción

En 1995 se realizó el informe *CHAOS* sobre los resultados que se obtuvieron en diversos proyectos software:

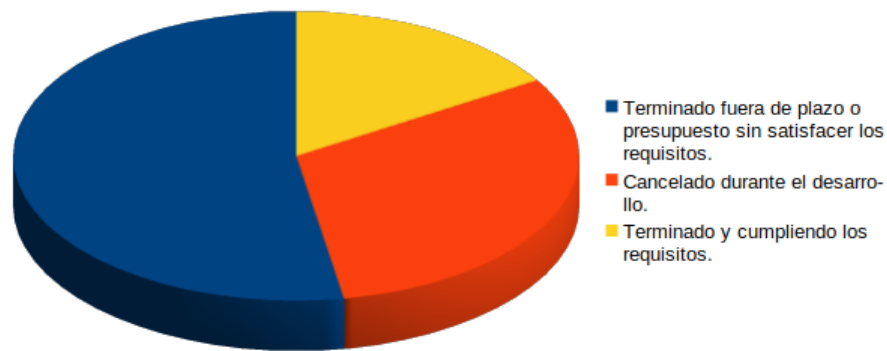


Figura 17: Resultados del informe *CHAOS*.

Los principales factores de fracaso son:

1. Falta de información por parte de los usuarios.
2. Especificación de requisitos incompleta.
3. Continuos cambios de los requisitos.
4. Pobres habilidades técnicas en la especificación de requisitos.

La **Ingeniería de Requisitos** cubre las tareas y proporciona los mecanismos adecuados para:

- Entender y analizar las necesidades del cliente.
- Evaluar la viabilidad de las necesidades.
- Negociar una solución razonable.
- Especificar la solución sin ambigüedades, confeccionando un documento que describa la solución acordada.
- Validar y analizar la especificación reflejada en el documento de especificación de requisitos, obteniendo el *modelo de análisis*.
- Administrar y desarrollar los requisitos a lo largo del proceso de desarrollo.

El proceso de construcción de una *especificación de requisitos* es iterativo. En él, partimos de las especificaciones iniciales incompletas, poco claras y ambiguas y llegamos a especificaciones finales **completas, claras, documentadas y validadas**.