Programación Orientada a Objetos

Tema 2:
Fundamentos de la programación orientada a objetos

Tema 2-1: El desarrollo de software: Evolución hacia la Orientación a Objetos

Contenidos

- Tema 2-1: El desarrollo de software: Evolución hacia la Orientación a Objetos

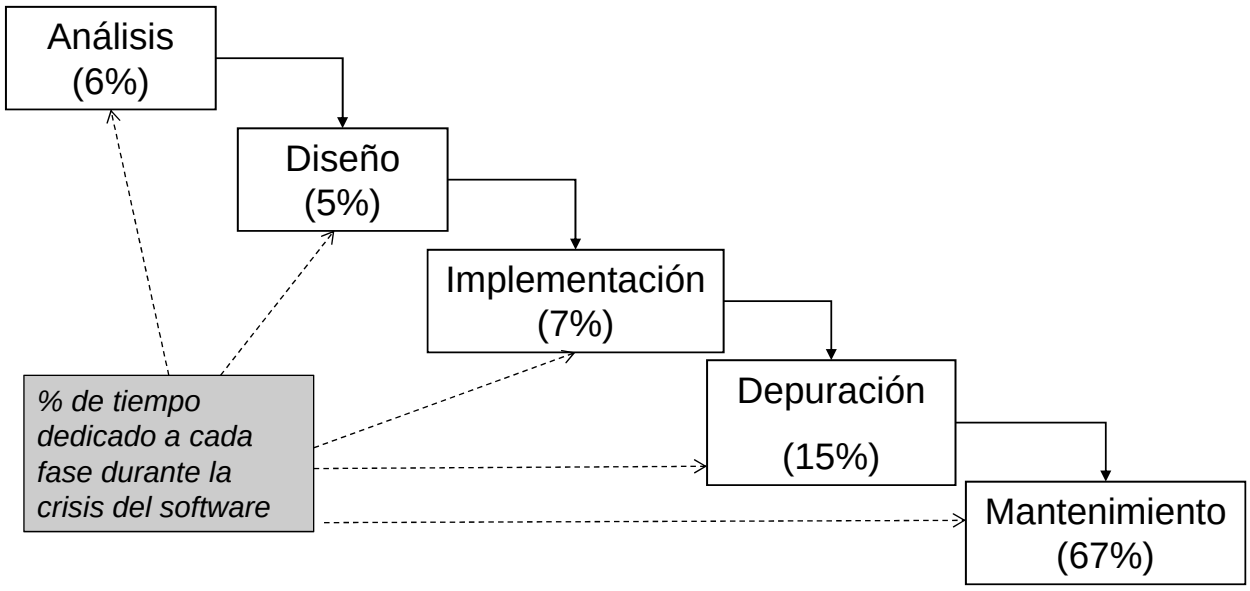
1. La crisis del software
2. La calidad del software
3. La complejidad inherente al software

La crisis del software

- A finales de los 60 se acuñó el término *crisis del software:*
  - El software era caro, poco fiable y escaso.
  - Metodologías y técnicas estructuradas no resuelven el problema.
  - Crecimiento de la complejidad de los problemas a representar.
  - Mayor problema: Mantenimiento del software.

La crisis del software

- Ciclo de vida del software (clásico):

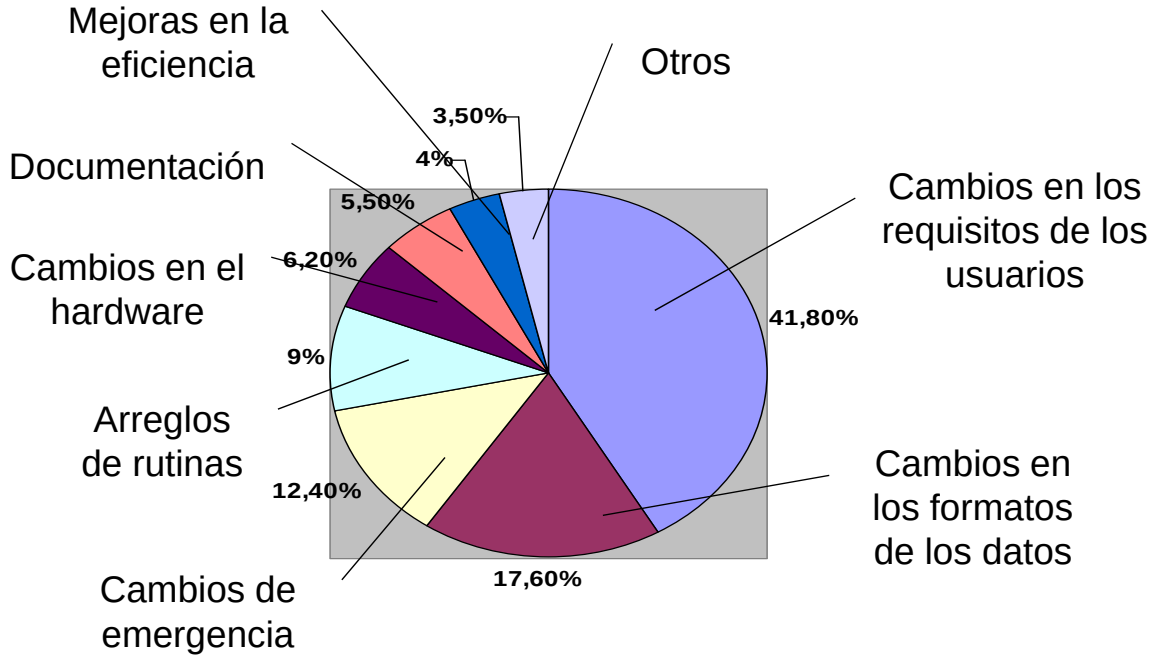


La crisis del software

- Mantenimiento del software:
  - Mantenimiento es lo que sucede después de que se ha distribuido un producto de software.
  - Se le dedicaba aprox. el 70 % del coste del software.
  - ¿Qué significa “mantenimiento” en software?
    - Parte noble: MODIFICACIÓN
      adaptación a los cambios
    - Parte no noble: DEPURACIÓN
      quitar errores

La crisis del software

- **Mantenimiento del software:**

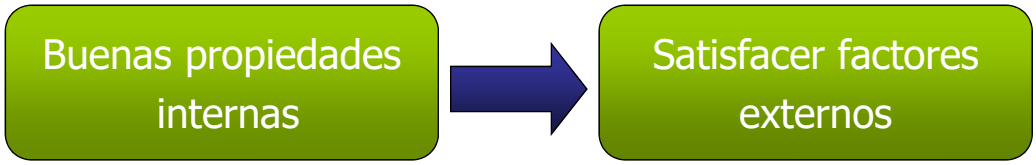


La crisis del software

- **Las consecuencias son Sistemas:**
  - ♦ Que no cumplen los requisitos iniciales.
  - ♦ Entregados fuera de plazo.
  - ♦ Sobrepasando ampliamente los presupuestos iniciales.
  - ♦ Con poca satisfacción por parte del usuario.

La calidad del software

- **Factores Externos**
  - Pueden ser detectados por los usuarios
  - Calidad externa es la que realmente preocupa
- **Factores Internos**
  - Sólo los perciben los diseñadores e implementadores
  - Medio de conseguir la calidad externa
- **OBJETIVO**



La POO es un conjunto de técnicas para obtener ***calidad interna*** como medio para obtener ***calidad externa*** (Reutilización y Extensibilidad)

La calidad del software

- **Factores Externos**

- Corrección
- Economía
- Portabilidad
- **Extensibilidad**
- **Reutilización**
- Facilidad de verificación
- Compatibilidad
- Eficiencia
- Robustez
- Integridad
- Facilidad de uso
- Funcionalidad
- Facilidad de reparación
- Oportunidad

- **Factores Internos**

- **Modularidad**
- Legibilidad

Universidad de Alcalá
Departamento Ciencias de la Computación

La calidad del software

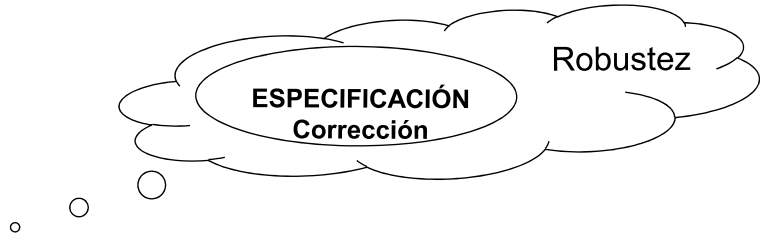
Corrección:

Es la capacidad de los productos software de realizar con exactitud su tarea, tal y como es definida en la **especificación**.

- Hay que definir los requisitos de manera precisa.

Robustez:

Es la capacidad de los productos software de reaccionar adecuadamente ante **situaciones excepcionales**.



Tienen que ver con el comportamiento (casos previstos o no)

Extensibilidad:

Es la facilidad de **adaptación** de los productos software a los **cambios** en la especificación.

- Cambios son frecuentes puesto que en la base de todo software hay algún fenómeno humano.
- Dificultad de adaptación proporcional al tamaño del sistema.
- Principios esenciales para facilitar la extensibilidad:
  - **Simplicidad** de la arquitectura del software.
  - **Descentralización**: módulos autónomos.

Universidad de Alcalá
La calidad del software
Departamento Ciencias de la Computación

Reutilización:

Es la capacidad de un producto software de ser utilizado en la **construcción de diferentes aplicaciones**.

- No reinventar soluciones para problemas ya resueltos.
- Se escribe menos software, luego se puede dedicar más tiempo a mejorar otros factores (fiabilidad).

Compatibilidad:

Es la facilidad de **combinar** unos elementos software con otros.

- Los sistemas necesitan interactuar con otros.
- Convenciones estándar de comunicación inter-módulos.

La calidad del software

Eficiencia:
Es la capacidad de un sistema software de requerir la **menor cantidad** posible de recursos hardware.

- Factor importante para la utilización.
- Debemos conjugar eficiencia con los otros objetivos.

Portabilidad:
Es la facilidad de **transferir** productos software a diferentes plataformas (entornos hw y sw).

Universidad de Alcalá
Departamento Ciencias de la Computación

La calidad del software

Facilidad de uso:
Es la facilidad con la que personas con diferentes niveles de experiencia pueden **aprender a usar** los productos software y aplicarlos a resolver problemas. También incluye la facilidad de instalación, operación y supervisión.

Funcionalidad:
Conjunto de posibilidades ofrecido por un sistema.

- Evitar añadir propiedades de forma incontrolada.
- Un buen producto software debe estar basado en un pequeño número de grandes ideas.
- Mantener constante el nivel de calidad.

Oportunidad:

Es la capacidad de un sistema software de ser **lanzado** cuando los usuarios lo desean, o antes.

- **Otros factores:**
  - **Economía**:
    Debe completarse con el presupuesto asignado.
  - **Integridad:**
    Protección del sistema contra modificaciones y accesos no autorizados.
  - **Facilidad para reparaciones** (de defectos)
  - **Facilidades de verificación**:
    Datos de prueba y procedimientos para detectar fallos.

Universidad de Alcalá
La calidad del software
Departamento Ciencias de la Computación

Consecuencias de estos criterios:

- Necesidad de una **BUENA DOCUMENTACIÓN**:
  - externa (usuarios) → facilidad de uso
  - interna (desarrolladores) → extensibilidad
  - interfaz del módulo → extensibilidad y reutilización

- Factores que pueden entrar en **CONFLICTO:**
  - integridad ↔ facilidad de uso
  - economía ↔ funcionalidad
  - eficiencia ↔ portabilidad
  - ajustarse a la especificación ↔ reutilización

La complejidad inherente al software

- La complejidad del dominio del problema:
  - Difícil interacción entre los usuarios de un sistema y sus desarrolladores.
  - Diferentes perspectivas sobre la naturaleza del problema.
  - Los requisitos de un sistema informático cambian durante su desarrollo.
- La dificultad de gestionar el proceso de desarrollo:
  - La construcción de un SI requiere la escritura de grandes cantidades de nuevo software y la reutilización de software existente.
  - Construcción de módulos independientes.
  - Comunicación entre los miembros del equipo de desarrollo.
  - Organización en diferentes fases o etapas.
  - Sujeto a continuos cambios.

SOLUCIÓN: POO

- Los SOO ofrecen rendimiento, flexibilidad y funcionalidad para implementaciones prácticas.
- Mediante POO se expanden las posibilidades de desarrollo de SI debido a la falta de restricciones en los tipos de datos. Estructuras de datos heterogéneas.
- La POO permite la reutilización de componentes software.