**Principios de diseño**

Diseño:

Los diseños han de ocultar o diferir los detalles de implementación ν Las abstracciones permiten comprender la esencia de los subsistemas sin tener que conocer detalles innecesarios ν Las decisiones de diseño susceptibles de cambio deben ocultarse detrás de interfaces abstractas ν Los módulos se han de diseñar de forma que la información interna del módulo sea inaccesible a otros módulos que no la necesitan ν Una solución modular implica niveles de abstracción

Refinamiento sucesivo:

Estrategia de diseño descendente ν El diseño se refina con una jerarquía de detalles creciente.

Concepto muy ligado a la abstracción.

El refinamiento es un concepto complementario a la abstracción ν Es el procedimiento por el que se va pasando de los niveles superiores de abstracción a los niveles inferiores, es decir, la manera en que se va añadiendo información de un nivel a otro [Wirth, 1971]

Ocultar la información:

Los módulos se caracterizan por las decisiones de diseño que (cada uno) oculta al resto [Parnas, 1972]. Los módulos deberán especificarse y diseñarse de manera que la información (procedimientos y datos) que está dentro de un módulo sea inaccesible a otros módulos que no necesiten esa información. El ocultamiento de la información es un buen medio para conseguir

Modularidad:

Es una partición lógica del diseño software que permite al software complejo ser manejable para propósitos de implementación y mantenimiento. Es el atributo individual del software que permite a un programa ser intelectualmente manejable [Myers, 1978]. Es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y débilmente acoplados [Booch, 1994]. Propuesta de descomposición funcional del sistema

Cohesión:

La cohesión es la medida de la relación funcional de los elementos de un módulo. Aquí un elemento hace referencia a cualquier componente del módulo, instrucción o grupo de instrucciones, definiciones de datos, procedimientos… que lleva a cabo algún trabajo o define algún dato. Estudia la medida de la relación que existe entre los elementos de un mismo módulo. Organización de los elementos de forma que los que tengan una mayor relación para realizar una tarea, pertenezcan al mismo módulo y los elementos no relacionados se encuentren en módulos separados. Un subsistema o módulo tiene un alto grado de cohesión si mantiene “unidas” cosas que están relacionadas entre ellas y mantiene fuera el resto

Acoplamiento:

El acoplamiento es una medida de la interconexión entre los módulos de una estructura de software [Pressman, 2006]. El acoplamiento depende de la complejidad de la interconexión entre los módulos, el punto donde se realiza una entrada o referencia a un módulo y los datos que se pasan a través de la interfaz.

Arquitectura de software:

La arquitectura del software alude a la estructura global del software y las formas en que esa estructura proporciona integridad conceptual a un sistema [Shaw y Garlan, 1995]. La arquitectura del software es la estructura lógica y física de un sistema, forjada por todas las decisiones de diseño estratégicas y tácticas aplicadas durante el desarrollo [Booch, 1994]. Una arquitectura software es la descripción de los subsistemas y componentes de un sistema software y de las relaciones entre ellos. Los subsistemas y componentes se especifican habitualmente desde diferentes puntos de vista para mostrar las propiedades funcionales y no funcionales relevantes de un sistema software. La arquitectura software de un sistema es un “artefacto”, es el resultado de la actividad de diseño del sistema [Buschmann et al., 1996]

Separación de preocupaciones:

La estructura de un programa debe partirse horizontal y verticalmente. La partición horizontal define ramas separadas de la jerarquía modular para cada función principal del programa. Módulos de control. Enfoque entrada/proceso/salida. Beneficios de la partición horizontal. Proporciona software más fácil de probar. Lleva a un software más fácil de mantener. Propaga menos efectos secundarios. Proporciona software más fácil de ampliar. Puntos en contra de la partición horizontal. Aumenta la comunicación entre módulos, pudiendo complicar el control global del flujo del programa