

Unidad 1 - Cuadro sinoptico

Ingenieria de Software Basada en Componentes vs No Basada en Componentes

Enfoque de desarrollo	Ventajas	Desventajas	Caracteristicas	Beneficios
A) Desarrollo de software basado en componentes (ISBC)	<ul style="list-style-type: none"> Reduce el tiempo de desarrollo al reutilizar componentes. Menos errores si los componentes ya estan probados. Facilita mantenimiento y escalabilidad por modulos. Permite integrar componentes comerciales (COTS/CYD). Aumenta la productividad del equipo. 	<ul style="list-style-type: none"> Dependencia de terceros (costos, cambios de versiones, caidas). Posibles incompatibilidades entre interfaces o tecnologias. Integracion puede ser compleja (adaptadores, configuracion). No siempre se ajusta 100% a lo requerido. Riesgos de licencias y seguridad en componentes externos. 	<ul style="list-style-type: none"> Reutilizacion de componentes internos o comerciales. Interfaces/contratos definidos (APIs, entradas/salidas). Bajo acoplamiento y alta cohesion. Arquitectura modular (capas/servicios). Enfoque: ensamblar e integrar mas que programar todo. 	<ul style="list-style-type: none"> Menor costo total de desarrollo y mantenimiento. Entrega mas rapida (time-to-market). Mayor calidad por uso de piezas validadas. Flexibilidad para reemplazar componentes. Mejor escalabilidad y evolucion del sistema.
B) Desarrollo de software no basado en componentes (tradicional / desde cero)	<ul style="list-style-type: none"> Control total del codigo y del disenio. Se adapta exactamente a los requisitos del sistema. Menos dependencia de proveedores externos. Optimizacion especifica para el negocio. Uniformidad tecnologica si se usa una sola plataforma. 	<ul style="list-style-type: none"> Mayor tiempo y costo de desarrollo. Mayor riesgo de errores al construir todo nuevo. Mantenimiento mas dificil si el sistema queda monolitico. Requiere mas personal/experiencia para modulos complejos. Menor reutilizacion: se repite trabajo en otros proyectos. 	<ul style="list-style-type: none"> Desarrollo completo desde cero. Alta personalizacion de logica y disenio. Mayor esfuerzo de pruebas y validacion. Puede ser monolitico o modular, pero sin reutilizacion real. Evolucion depende mas del equipo interno. 	<ul style="list-style-type: none"> Solucion hecha a medida (encaja con necesidades exactas). Independencia de licencias y proveedores. Mayor control de seguridad y rendimiento. Permite reglas de negocio unicas sin limitaciones de terceros. Propiedad total del software y su ciclo de vida.