Principios generales de alto nivel

1. Servicios diseñados teniendo en cuenta el objetivo, demanda y capacidad.
2. Diseñados "de fuera a dentro" no "de dentro afuera".
3. Servicios diseñados dentro del contexto del sistema.
4. Diseñados alrededor de una comprensión del valor y la eficiencia del flujo.
5. No tratará una clase especial de variación como si fuera una clase común.
6. Los servicios serán cocreados con los usuarios.
7. Los diseños de servicios serán prototipados.
8. Los servicios serán diseñados, construidos y desplegados de forma incremental e iterativa.
9. Los servicios serán diseñados y entregados de forma colaborativa.

Principios del proceso de diseño (reducir/minimizar)

1. Minimizar actividades que no aportan valor al cliente.
2. Estructurar trabajo en torno a procesos, no funciones, productos o ubicación.
3. Minimizar la fragmentación del trabajo. Un solo individuo responsable.
4. Complejidad de los procesos será minimizada.
5. Multiples versiones de un proceso serán ~~minimizar~~ desarrolladas cuando sea necesario.
6. Variaciones en el proceso serán minimizadas para maximizar fiabilidad y predictibilidad.
7. Procesamiento local será maximizado.
8. Evitar descomposición excesiva de los procesos.
9. Reducir al mínimo las interrupciones y retrasos innecesarios del proceso.
10. Se reducirá al mínimo la necesidad de conciliación.
11. Se reducirá al mínimo la necesidad de controles.
12. Se reducirá ~~necesidad~~ al mínimo la necesidad de inspección.
13. Solo mediremos lo que importe.

Principios de diseño de los datos

1. Normalizar datos en organización y socios.
2. Datos transferibles y reutilizables.
3. Evitar introducción de datos, utilizando métodos de búsqueda, selección y confirmación de datos.

Principios de diseño de la organización

1. Grupos de trabajo por procesos y competencias.
2. Personal capacitado para tomar decisiones.
3. Ubicación del trabajo: eficiencia y eficacia.

Principios de diseño de la tecnología

1. Es facilitador, no impulsor.
2. Se "introducirá" en el diseño, no se "empujará".
3. El diseño tecnológico es flexible y ágil.

Principios de SOA

1. Contrato de Servicio normalizado
   Acuerdos de comunicación normalizados
2. Autonomía de referencia de los servicios
   Solo son conscientes de la existencia de otros.
3. Transparencia en la ubicación de los servicios
   Acoplamiento débil. Pueden ser llamados desde cualquier punto de la red.
4. Abstracción de servicios.
   Cajas negras. Lógica interna oculta.

[Abstracción, reutilización, composición, encapsulación, normalización]

5. Autonomía de los servicios.
   Son independientes y controlan la funcionalidad que encapsulan (diseño y ejecución)
6. Servicios sin estado.
   Devuelven el valor solicitado o emiten una excepción. Minimizan uso de recursos.
7. Granularidad de los servicios.
   Tamaño y alcance adecuado. Funcionalidad es relevante.
8. Normalización de Servicios
   Descomposición o consolidación para minimizar redundancia.
9. Capacidad de composición de servicios.
   Servicios pueden utilizarse para componer otros servicios.
10. Descubrimiento de servicios.
    Metadatos comunicativos que permiten descubrirlos e interpretarlos correctamente.
11. Reutilización de Servicios.
    Lógica dividida en varios servicios para favorecer la reutilización del código.
12. Encapsulación de servicios.
    Servicios "Legacy" no planteados inicialmente bajo SOA pueden encapsularse.

Características de un buen diseño

1. Verificación
2. Reusabilidad
3. Satisfacción del cliente
4. Efectividad/Eficacia
5. Mantenibilidad

Estilos arquitectónicos comunes

1. Pipes&filtros
2. Orientado a objetos
3. Invocación implícita
4. Pas capes
5. Cliente-Servidor
6. Repositorio (y pizarra)
7. Máquina virtual

SOA es una arquitectura independiente de la tecnología

Organización del diseño de software

1. Estilos arquitectónicos
   Definen componentes y conectores. Conjunto de restricciones sobre como pueden ser utilizados.
   Reutilización de diseño y código. Interoperabilidad.
2. Patrones arquitectónicos
   Estrategias de implementación de componentes y conectores. Nivel abstracción.
3. Patrones de diseño.
   Soluciones estándar a problemas de implementación creados a nivel de programación.
4. Idioms
   Implementan aspectos particulares de componentes característicos y relaciones. A nivel de lenguaje

Patrones SOA Básicos

1. Servicios agnósticos
   Funcionalidades comunes a varios problemas de negocio.
   Reutilización y facilidad de composición.

2. Declaración de servicios agnósticos
   Estos servicios deberán declarar explícitamente que son agnósticos.
   Reusabilidad en otros proyectos. Composición

3. Transacciones en servicios atómicos
   Implementan operaciones reusables o son aptos de participar en composiciones transaccionales. Statelessness.
   En componente de capa superior.

4. Enterprise Service Bus
   Gestión de mensajes: transformación, enrutamiento...
   Componente de capa superior. Bajo acoplamiento, interoperabilidad XML. Flexibilidad. Latencias. Fault point

5. Service Façade
   Servicio intermedio entre el contrato que firma el consumidor y el servicio.
   Elimina/Reduce el acoplamiento. Reduce el impacto de un cambio. Varios Façade.

6. Gestión de autenticación.
   Autentica consumidores que acceden a un servicio.
   Uso de tokens.
   Facilidad de composición

7. Autenticación de mensajes de origen
   Uso de certificados para la autenticación de clientes y sus mensajes

8. Análisis de mensajes
   Servicio de análisis de mensajes para evitar mensajes malintencionados

9. Service Callbacks
   Consumidores consultan de forma asíncrona. Servicio notifica al consumidor. Libera recursos para tareas de larga duración.

10. Varios contratos de servicio
    Un servicio soporte varios contratos a la vez. Mientras soporte a versiones antiguas o facilita la readaptación.

Principios Integración de Servicios

1. Orquestación
   Diferentes servicios de grano fino existentes para conseguir un servicio compuesto.

2. Transformación
   Formatos de datos (ESB)

3. Transporte
   Protocolos de negociación y transporte.

4. Mediación
   Múltiples interfaces. Versiones y/o canales

5. Consistencia no funcional
   Seguridad, rendimiento, monitorización

Clasificación del mantenimiento

1. Adaptativo
   Necesitante cambios en el entorno del software.

2. Perfectivo
   Implementar o cambiar requerimientos referentes a mejoras funcionales del software.

3. Correctivo
   Diagnosticar y corregir errores

4. Preventivo
   Aumentar la capacidad de mantenimiento del software.

Errores, defecto, fallo

Un error del programador introduce un defecto y este defecto produce un fallo.

Error → Resultado incorrecto. Acción humana que lleva a cabo un resultado incorrecto

Defecto → Definición de datos incorrecta. Procesamiento incorrecto en el programa

Fallo → Incapacidad del sistema para realizar las funciones requeridas

Verificar → Construir el producto correctamente

Validar → Construir el producto correcto

Software fácil de probar: (7)

- Operativo
- Observable
- Controlable
- Descomponible
- Simple
- Estable
- Comprensible

Mantenimiento

- Adaptativo: modificas el sistema para hacer frente a cambios en el ambiente del software.
- Perfectivo: implementes nuevos, o cambia requerimientos de usuario referentes a mejoras funcionales del software.
- Correctivo: diagnosticas y corriges errores
- Preventivo: aumentas la capacidad de mantenimiento del software o fiabilidad para evitar problemas futuros.

SIAM - ITIL

SIAM es una adaptación de ITIL centrada en la gestión de Servicios Proporcionados gobernanza, garantía y gestión coherente.

Ciclo de pruebas

Código

→ Pruebas de unidad
- Caja Blanca
  - Camino Básico
    - Grafo de flujo
    - Complejidad ciclomática
  - Estructura Control
    - Condición
    - Flujo
    - Bucle
- Caja Negra
  - Clases de equivalencia
  - Valores límite
  - Aleatorio. Conjetura errores

→ Pruebas de integración
- Ascendente
- Descendente
- Regresión
- Humo

→ Pruebas de sistema
- Prueba de recuperación
- Prueba de seguridad
- Prueba de resistencia (stress)
- Prueba de rendimiento

→ Pruebas de validación
- Revisión de la configuración
- Prueba Alfa
- Prueba Beta

→ Depuración