



Catálogo de puntos de vista Punto de vista de desarrollo



USAC - Análisis y diseño de sistemas 2 - 1er semestre 2016
Ing. Ricardo Morales

Descripción

Definición

- Describe la arquitectura que soporta el proceso de desarrollo de software

Temas de interés

- Organización de módulos
- Procesamiento común
- Estandarización de diseño
- Estandarización de pruebas
- Instrumentación
- Organización de codelines



Descripción (II)

Modelos

- Modelo de estructura de módulos
- Modelos de diseño común
- Modelos codeline

Problemas

- Mucho detalle
- Descripción sobrecargada
- Enfoque desigual
- Falta de enfoque en el desarrollador



Descripción (III)

Stakeholders

- Ingenieros de producción, desarrolladores y probadores

Aplicabilidad

- Todos los sistemas con un desarrollo de software significativo



Descripción (IV)

- ▶ Este punto de vista es relevante casi para todos proyectos de sistemas de información grandes porque todos ellos tienen algún elemento de desarrollo de software
- ▶ La importancia de esta vista depende de la complejidad del sistema a ser construido, la experiencia de los desarrolladores, madurez de tecnologías usadas y familiaridad del equipo con dichas metodologías
- ▶ Se enfoca en temas de interés que son significativos para la arquitectura



Temas de interés (concerns)

▶ Organización de módulos

- ▶ El organizar el código en unidades mayores llamadas módulos, que contienen código relacionado, ayuda a administrar dependencias y ayuda a los desarrolladores a entenderlas y trabajar en ellas sin afectar otros módulos de formas no esperadas

▶ Procesamiento común

- ▶ Esta vista asegura que áreas de procesamiento común sean identificadas y especificadas claramente, agrupando y aislando dichas áreas en módulos separados

▶ Estandarización de diseño

- ▶ Estandarizar los aspectos clave del diseño provee beneficios críticos para la mantenibilidad, confiabilidad y cohesión técnica del sistema



Temas de interés (II)

- ▶ Estandarización de pruebas

- ▶ Las estandarización de los enfoques de pruebas, tecnologías y convenciones ayuda a hacer pruebas de forma consistente y acelera todo el proceso, incluye herramientas, infraestructura, datos y automatización

- ▶ Instrumentación

- ▶ Es la práctica de insertar código especial para loguear información acerca de la ejecución de pasos, estado del sistema, uso de recursos y otros para apoyar el monitoreo y depuración

- ▶ Organización de codelines

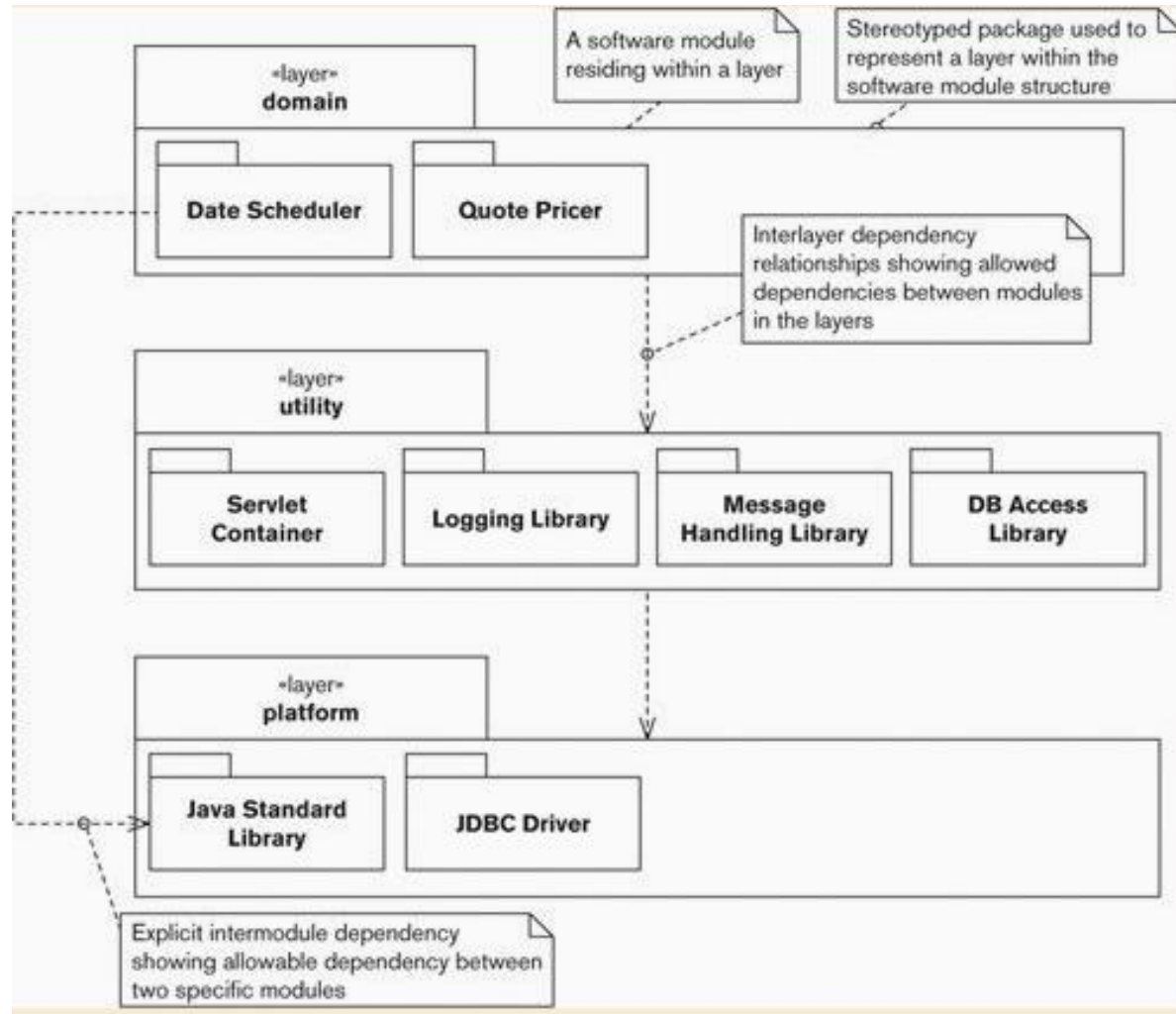


Modelos – modelo de estructura de módulos

- ▶ Define la organización del código fuente, en términos de módulos en los cuales se agrupa el código y las dependencias entre estos módulos
- ▶ También es común que imponga algún grado de organización de alto nivel a los módulos para evitar enumerar demasiadas dependencias individuales



Ejemplo



Actividades

Identificar y
clasificar
módulos

Identificar
dependencias
de módulos

Identificar
reglas de
capas



Modelos - modelo de diseño común

- ▶ Un modelo de diseño común tiene las siguientes piezas importantes
 - ▶ Una definición del proceso común requerido entre elementos, considerando:
 - ▶ Inicialización y recuperación
 - ▶ Terminación y reinicio
 - ▶ Logueo de mensajes e instrumentación
 - ▶ Internacionalización
 - ▶ Seguridad
 - ▶ Manejo de transacciones
 - ▶ Una definición del enfoque estándar de diseño
 - ▶ Una definición de que software común será usado y como debe ser usado
- ▶ Se modela a través de un documento que define estos aspectos



Actividades

Identificar
procesamiento
común

Definir el rol de
elementos
comunes

Identificar las
restricciones de
diseño
requeridas

Identificar y
definir patrones
de diseño



Modelos -Modelo codeline

- ▶ Normalmente captura los siguientes hechos
 - ▶ ¿Cómo se organiza el código en archivos fuentes?
 - ▶ ¿Cómo se agrupan los archivos en módulos?
 - ▶ ¿Qué estructura de directorios se usará para los archivos?
 - ▶ ¿Cómo se hará el build y pruebas automáticos?
 - ▶ ¿Qué tipo y alcance de pruebas se deben correr regularmente y como se corren?
 - ▶ ¿Cómo se liberan los binarios en un ambiente (idealmente con un proceso automatizado)?
 - ▶ ¿Cómo se controlan los fuentes usando administración de configuración para un desarrollo concurrente?
 - ▶ ¿Qué herramientas automatizadas se usarán para el build, pruebas y liberación, y cómo funcionan en conjunto?
 - ▶ Se describe con texto y diagramas
-



Actividades

Diseñar
estructura de
código fuente

Definir la
administración
de la
configuración

Diseñar enfoque
de build,
integración y
pruebas

Definir proceso
de release

