Anexo: Diseño con reutilización – Diseño basado en Componentes

01-Las unidades de software que se reutilizan son de tamaños totalmente diferentes; describa y ejemplifique cada una de ellas.

Unidades de software que se reutilizan:

* Re utilización de  sistemas de aplicaciones: Puede  ser  reutilizada  incorporándolos sin ningún cambio en otros sistemas, configurando la aplicación para diferentes clientes.
* Reutilización  de  componentes: Varía  en  tamaño  desde  subsistemas  hasta  objetos simples.
* Re utilización de objetos y funciones: Pueden reutilizarse componentes software que implementan una única función.
* Una forma complementaria de reutilización es la reutilización de conceptos, en la que en lugar de reutilizar un componente,  la entidad reutilizada es más abstracta y se diseña para  ser configurada y adaptada a una variedad de situaciones. Puede incluirse tanto en patrones de diseño, productos de sistemas configurables y generadores de programas.

02-¿Cuáles son los beneficios de la reutilización del software?

Cada uno de los miembros de la organización obtiene beneficios de la reutilización de software, la tabla siguiente se resumen cada uno de los beneficios obtenidos por los colaboradores

|  |  |
| --- | --- |
| **Ventajas de Reutilización de Software para Colaboradores Individuales** | |
| **Colaborador** | **Beneficios** |
| Director General | • Ganancias de productividad grandes  • Tiempo en el mercado enormemente mejorado  • Crecimiento sostenido y presencia de mercado  • Capacidad de capturar un mercado económicamente atractivo |
| Director de Desarrollo | • Uso eficiente de personal  • Capacidad de explorar nuevos mercados, nueva tecnología, y/o nuevos productos  • Fondo de personal Fluido |
| Líder de Proyecto | • Previsibilidad aumentada  • Papeles bien establecidos y responsabilidades  • Producción eficiente  • Riesgos más controlados |
| Programador | • Moral más alta  • Mayor satisfacción de trabajo  • Puede concentrarse en aspectos realmente únicos de productos  • Integración de software más fácil  • Menos tardanzas en el calendario  • Mayor movilidad dentro de la organización  • Más comerciable  • Tienen el tiempo para aprender la nueva tecnología  • Son la parte de un equipo que construye productos con un registro de calidad establecido y reputación |
| Arquitecto o Desarrollador de Activos Base | • Mayor desafío  • Su trabajo tiene más impacto  • Prestigio dentro de la organización  • Tan comerciable como la línea de productos |
| Cliente | • Productos de calidad más altos  • Fecha de entrega previsible  • Coste previsible  • Gastos conocidos para requerimientos único  • Bien Probados materiales de entrenamiento y documentación  • Gastos de mantenimiento compartidos  • Potencial para participar en el grupo de un usuario |
| Usuario Final | • Menos defectos  • Mejor formación de materiales y documentación  • Una red de otros usuarios |

03- ¿Cuáles son los problemas con la reutilización?

Problemas para la Reutilización:

Existe un cierto número de dificultades para la reutilización del software. Con objeto de desarrollar una estrategia de reutilización efectiva, los administradores deben de entender cuales son estos problemas.

1. La reutilización del software no es en absoluto un asunto «de la máxima prioridad» para muchas firmas de software.
2. Aun cuando existe un número creciente de fabricantes de software que venden en la actualidad herramientas o componentes que proporcionan una asistencia directa para la reutilización del software, la mayoría de los desarrolladores de software no la utilizan.
3. La cantidad de entrenamiento disponible para ayudar a los ingenieros del software y a los administradores a comprender y a explicar la reutilización es relativamente escasa.
4. Hay muchos especialistas de software que siguen creyendo que la reutilización es «más problema que beneficio».
5. Hay muchas compañías que siguen apoyando las metodologías de desarrollo del software que no facilitan la reutilización, a la vez que desaniman a otras que parecen impulsar la reutilización del software.
6. Hay pocas compañías que proporcionen incentivos para producir componentes reutilizables de programas. Los administradores del proyecto hacen lo posible por realizar el trabajo con componentes de programa que sean tan específicos como sea posible

04- ¿Cuáles son los requerimientos críticos para el diseño y desarrollo de software con reutilización?

Los factores clave que deberían considerarse a la hora de planificar la reutilización son:

1. La agenda de desarrollo del software. Si el software tiene que desarrollarse rápidamente, debería intentarse reutilizar sistemas comerciales en vez de componentes individuales. Éstos son activos reutilizables de grano grueso. Si bien el cumplimiento de los requerimientos puede no ser perfecto, esta aproximación minimiza la cantidad de desarrollo requerido.

2. Vida esperada del software. Si se está desarrollando un sistema de larga vida, habría

que centrarse en la mantenibilidad del sistema. En esas circunstancias, no solamente

debería pensarse en las posibilidades inmediatas de la reutitización, sino también en

las implicaciones a largo plazo. Se tendrá que adaptar el sistema a nuevos requerimientos, lo cual probablemente signifique hacer cambios a los componentes y cómo éstos son utilizados. Si no se tiene acceso al código fuente, probablemente debería evitarse utilizar componentes y sistemas de proveedores externos; no se puede estar seguro de que estos proveedores serán capaces de continuar soportando el software reutilizado.

3. Los conocimientos, habilidades y experiencia del grupo de desarrollo. Todas las tecnologías de reutilización son bastante complejas y se necesita bastante tiempo para comprenderlas y usarlas de forma efectiva. Sin embargo, si el grupo de desarrollo posee habilidades en un área particular, probablemente habría que centrarse en ella.

4. La criticidad del software y sus requerimientos no funcionales. Para un sistema crítico que tiene que ser certificado por un regulador externo, se tiene que crear un caso de confiabilidad para el sistema (tratado en el Capítulo 24). Esto es difícil si no se tiene acceso al código fuente del software. Si el software tiene requerimientos de rendimiento estrictos, puede ser imposible utilizar estrategias tales como reutilización a través de generadores de programas. Estos sistemas tienden a generar código relativamente ineficiente.

5. El dominio de las aplicaciones. En algunos dominios de aplicaciones como los sistemas de información médica y de fabricación, hay varios productos genéricos que pueden reutilizarse para configurarlos a una situación particular. Si se está trabajando en tales dominios, siempre deberían considerarse éstos como una opción.

6. La plataforma sobre la que el sistema se va a ejecutar. Algunos modelos de componentes, como COM/Active X, son plataformas específicas de Microsoft. Si se está

desarrollando sobre una plataforma como éstas, esta aproximación puede ser la más

adecuada. De igual forma, los sistemas de aplicaciones genéricos pueden ser plataformas específicas y solamente es posible reutilizarlos si el sistema se diseña para la misma plataforma.

05- Al visualizar un componente como un proveedor de servicios, enfatizamos en dos características fundamentales: código fuente e interfaz; explique cada una de ellas.

Código Fuente: El código fuente de un programa informático (o software) es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

El código fuente de un programa está escrito por un programador en algún lenguaje de programación, pero en este primer estado no es directamente ejecutable por la computadora, sino que debe ser traducido a otro lenguaje (el lenguaje máquina o código objeto) que sí pueda ser ejecutado por el hardware de la computadora. Para esta traducción se usan los llamados compiladores, ensambladores, intérpretes y otros sistemas de traducción.

El término código fuente también se usa para hacer referencia al código fuente de otros elementos del software, como por ejemplo el código fuente de una página web que está escrito en el lenguaje de marcado HTML o en Javascript u otros lenguajes de programación web y que es posteriormente ejecutado por el navegador web para visualizar dicha página cuando es visitada.

El área de la informática que se dedica a la creación de programas y, por tanto a la creación de su código fuente, es la programación.

Interfaz: La interfaz gráfica de usuario, conocida también como GUI (del inglés graphical user interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

06- Ejemplifique cada uno de los niveles de abstracción de los componentes

la reutilización orientada a objetos está mejor soportada en un proceso de desarrollo orientado a objetos a través de abstracciones denominadas marcos de trabajo.

Existen tres tipos de Marco de Trabajos:

**1. Marcos de trabajo de infraestructura de sistemas.** Estos marcos de trabajo soportan 4 desarrollos de infraestructuras de sistemas tales como comunicaciones, interfaces de usuario y compiladores (Schmidt, 1997).

**2. Marcos de trabajo para la integración de middleware.** Consisten en un conjunto de estándares y clases de objetos asociados que soportan la comunicación de componentes y el intercambio de información. Ejemplos de este tipo de marcos son COF BA, COM+ de Microsoft y Enterprise Java Beans. Estos marcos proporcionan se

porte para modelos de componentes estandarizados, tal y como se expone en (

Capítulo 19.

**3. Marcos de trabajo de aplicaciones empresariales**. Se refieren a dominios de aplícaciones específicos tales como telecomunicaciones o sistemas financieros (Baumer < al, 1997). Estos marcos de trabajo encapsulan el conocimiento del dominio de la aplicación y soportan el desarrollo de aplicaciones para los usuarios finales.

07- A través de un ejemplo explique que es un Marco de Trabajo:

Uno de los marcos de trabajo más conocido y ampliamente usado para el diseño de GU

es el marco Modelo-Vista-Controlador (MVC) . El marco de trabajo MVC fue

propuesto originalmente en la década de los 80 como una aproximación al diseño de GU que permitió múltiples presentaciones de un objeto y estilos independientes de interacciónes con cada una de estas presentaciones. El marco MVC soporta la presentación de los datos de diferentes formas e interacciones independientes con cada una de estas presentaciones. Cuando los datos se modifican a través de una de las presentaciones, el resto de las presentaciones son actualizadas.

08- Analice ventajas y desventajas de utilizar cots. La reutilización de productos COTS está relacionada con la reutilización a gran escala de tos sistemas comerciales.

Éstos proporcionan bastantes ventajas, y su reutilización puede reducir los costes y tiempo de desarrollo de forma radical.

• Los problemas potenciales con la utilización basada en COTS Incluyen la ausencia de control sobre la funcionalidad y el rendimiento y ausencia de control sobre la evolución de los sistemas, la necesidad de soporte de vendedores externos y dificultades para asegurar que los sistemas pueden interoperar.

09- ¿Cuáles piensa deberían ser las características para que un componente sea reutilizable?

Pensamos que para que un componente sea reutilizable debe tener las siguientes caracteristicas:

• Eliminar los métodos específicos de la aplicación.

• Cambiar los nombres para hacerlos más generales.

• Añadir métodos para proporcionar una cobertura funcional más completa.

• Hacer que el manejo de excepciones sea consistente para todos los métodos.

• Añadir una interfaz de «configuración» para permitir que el componente se adapte a diferentes situaciones de uso.

• Integrar los componentes requeridos para incrementar la independencia.

10- Realice una conclusión personal del tema abordado.

CONCLUSIONES FINALES:

En este anexo se ha desarrollado una gran gama de aspectos de la reutilización del software empezando por el campo, seguidamente una larga descripción, historia y tipos de patrones que se usan a la hora de reutilizar así como las técnicas que se usan, tipos y reutilización basada en generadores y aplicaciones, en el aspecto de los patrones hay que destacar que son totalmente fundamentales para la reutilización del software en el diseño orientado a objetos porque nos documentan situaciones de cualquier tipo de diseño, hablando de generadores de programas los conceptos reutilizables están incluidos en un sistema generador, en cuanto a la reutilización de productos COTS se utilizan a grandes escalas en empresas bastante grandes y en la actualidad están resaltada por la programación de interfaces (APIs) que esta muy vinculada a la reutilización de productos COTS, permite reducir costes además de tiempo en el desarrollo y darle mayor funcionalidad aunque no todo es beneficioso y como todo tiene unos aspectos en el que surgen problemas y es la falta de control en el rendimiento y su funcionalidad y muestran una desconfianza o dificultad para asegurar que los sistemas puedan interoperar pero además de estos aspectos es una buena solución sobre todo en empresas de alto nivel a la hora d desarrollar sistemas con reutilización del software.