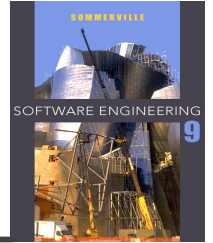


Capítulo 16- REUTILIZACIÓN DE SOFTWARE

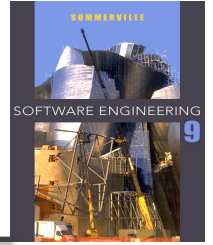
Lecture 1

Temas



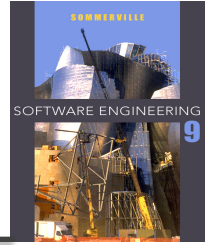
- ✧ Panorama de reutilización.
- ✧ Estructuras de aplicación.
- ✧ Líneas de productos de software.
- ✧ COTS (comercial fuera de la plataforma) reutilización de producto.

Reutilización de software



- ✧ En la mayoría de las disciplinas de la ingeniería, los sistemas han sido diseñados por la composición de los componentes existentes que han sido utilizados en otros sistemas.
- ✧ La ingeniería de software se ha centrado más en el desarrollo original, pero ahora se reconoce que para lograr un mejor software, más rápidamente y a menor costo, necesitamos un proceso de diseño que se basa en la reutilización sistemática del software.
- ✧ Se ha producido un cambio importante para el desarrollo basado en la reutilización en los últimos 10 años.

Ingeniería de software basada en la Reutilización



✧ *Reutilización del sistema de aplicación*

El conjunto de un sistema de aplicación puede volver a utilizarse, ya sea mediante su incorporación sin modificaciones en otros sistemas (reutilización COTS) o desarrollando familias de aplicaciones.

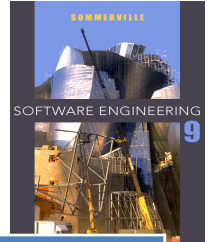
✧ Reutilización de componentes

Los componentes de una aplicación desde sub-sistemas hasta objetos simples pueden ser reutilizados. [Capítulo 17]

✧ Reutilización de Funciones y objetos

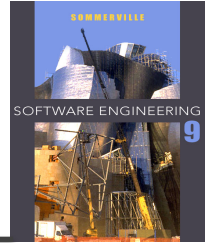
Los componentes de software que implementan un objeto bien definido o función pueden ser reutilizados.

Beneficios de reutilizar software



<i>Beneficios</i>	<i>Explicación</i>
Incrementa la dependencia	Software reutilizado, que ha sido probado y comprobado en los sistemas de trabajo, debería ser más confiable que el nuevo software. Sus fallas de diseño e implementación deberían haber sido encontrados y corregidos.
Reduce el riesgo de procesos	El costo del software existente ya es conocido, considerando que los costes de desarrollo son siempre una cuestión de criterio. Este es un factor importante para la gestión de proyectos, ya que reducen el margen de error en la estimación del costo del proyecto. Especialmente cierto cuando se vuelven a utilizar relativamente grandes componentes de software, tales como subsistemas.
Uso efectivo de especialistas	En vez de hacer el mismo trabajo una y otra vez, los especialistas de aplicaciones pueden desarrollar un software reutilizable que encapsula su conocimiento

Beneficios de reutilizar software

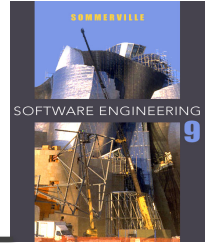


Beneficios	Explicación
Cumplimiento de estándares	Algunos estándares, como los estándares de interfaz de usuario, se pueden implementar como un conjunto de componentes reutilizables. Por ejemplo, si los menús en la interfaz de usuario se implementan utilizando componentes reutilizables, todas las aplicaciones presentan los mismos formatos de menú a los usuarios. El uso de las interfaces de usuario estándar mejora la fiabilidad ya que los usuarios cometen menos errores cuando se les presenta una interfaz familiar.
Desarrollo acelerado	Llevar al mercado el sistema lo más temprano posible a menudo es más importante que los costos generales de desarrollo. La reutilización del software del sistema puede acelerar la producción debido a que tanto el tiempo de desarrollo y de validación pueden ser reducidos.

Problemas con la reutilización

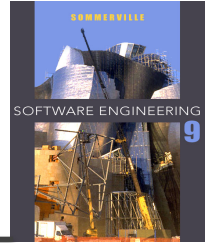
Problemas	Explicación
Incremento en costos de mantenimiento	Si el código fuente reutilizado de un sistema de software o componente no está disponible entonces los costos de mantenimiento pueden ser mayores ya que la reutilización de elementos del sistema puede volverse cada vez mas incompatible con cambios en el sistema.
Falta de herramientas de soporte	Algunas herramientas de software no soportan el desarrollo realizado con reutilización. Puede ser difícil o imposible de integrar estas herramientas con un componente del sistema. El proceso de software asumido por estas herramientas puede no utilizar reutilización como parte del desarrollo. Esto es particularmente cierto para sistemas de ingeniería con el soporte incluido, menos para herramientas de desarrollo orientado a objetos.
El síndrome de no fue inventado aquí	Algunos ingenieros de software prefieren reescribir componentes porque ellos creen que pueden mejorarlos. Esto en parte tiene que ver con la confianza y el hecho que escribir software original es visto más como un desafío que al reutilizar software desarrollado por otras personas.

Problems with reuse



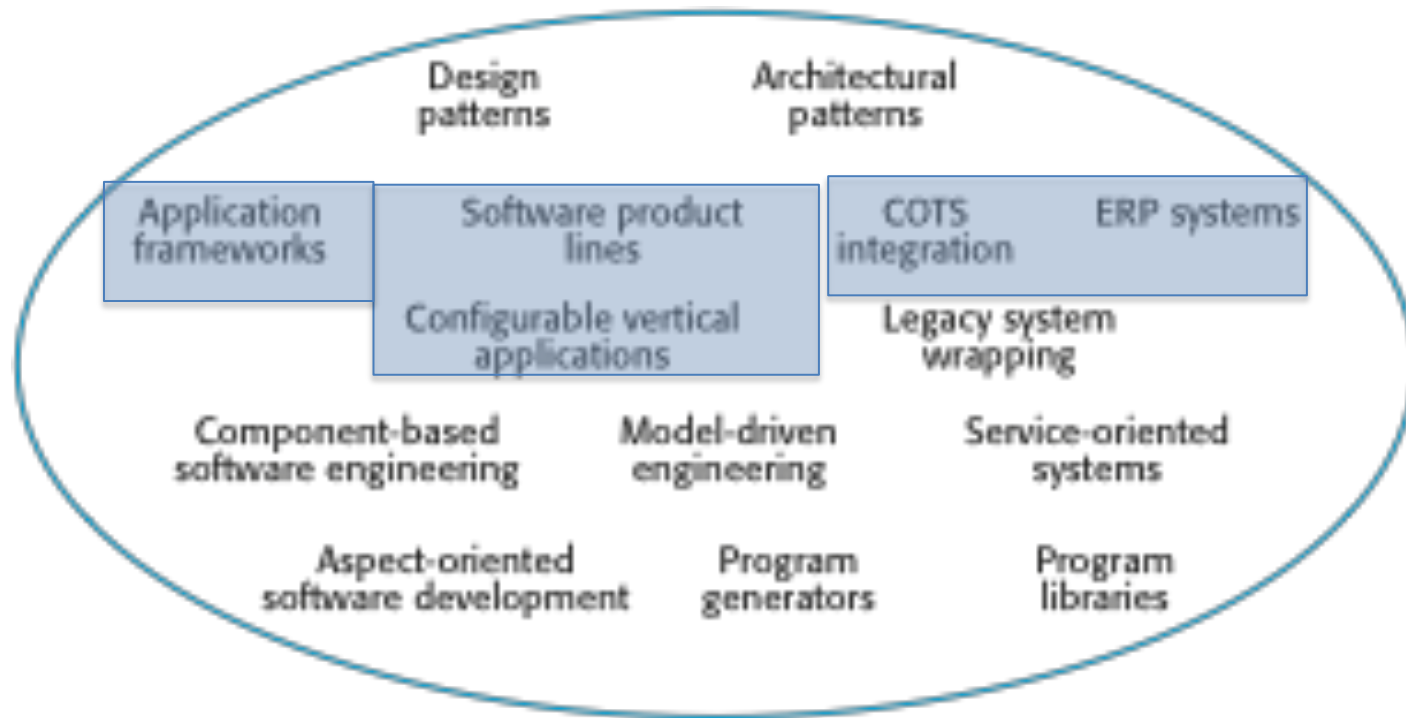
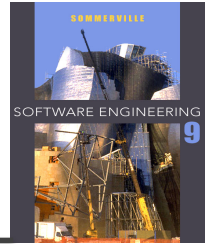
Problemas	Explicación
Crear, mantener y utilizar la información de los componentes.	Rellenar la biblioteca de componentes reutilizables y asegurar a los desarrolladores de software que pueden utilizar esta biblioteca puede ser costoso. Los procesos de desarrollo tienen que ser adaptados para asegurarse de que se utiliza la biblioteca.
Encontrar, comprender y adaptar componentes reutilizables.	Los componentes de software tienen que ser descubiertos en la biblioteca, entendidos y a veces, adaptados para trabajar en el nuevo entorno. Los ingenieros deben estar razonablemente seguros de encontrar el componente en la biblioteca antes incluyen la parte de búsqueda de componentes en el proceso de elaboración de normas.

Panorama de reutilización

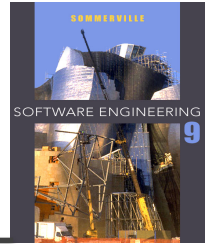


- ✧ Aunque la reutilización es muchas veces simplemente el pensamiento de reutilizar componentes de un sistema, hay muchos enfoques diferentes para reutilizar que podrían ser utilizados.
- ✧ Reutilizar es posible en un rango de niveles desde simples funciones hasta completar sistemas de aplicaciones.
- ✧ El panorama de reutilización cubre un rango de posibles técnicas de reutilización.

Panorama de Reutilización

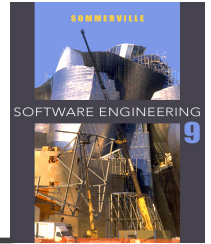


Enfoques que apoyan la reutilización de software



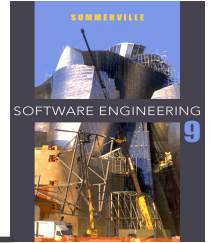
Enfoque	Descripción
Patrones arquitectónicos	Arquitecturas estándar de software que soportan tipos comunes de sistemas de aplicaciones son utilizadas como la base de aplicaciones. [Descritas en capítulo 6, 13 y 20]
Patrones de diseño	Abstracciones genéricas que ocurren a través de aplicaciones son representadas como patrones de diseño mostrando objetos e interacciones abstractas y concretas. [Descrito en capítulo 7]
Desarrollo basado en componentes	Los sistemas son desarrollados integrando componentes (colección de objetos) que se ajustan a los estándares de componentes en modelos. [Descrito en capítulo 17]
Estructuras de aplicación	Colección de clases abstractas y concretas son adaptadas y extendidas para crear sistemas de aplicación
Envoltura de sistemas heredados *	Los sistemas heredados (ver capítulo 9) son 'envueltas' o 'cubiertas' definiendo un grupo de interfaces y permitiendo el acceso a éstos sistemas heredados a través de éstas interfaces.

Enfoques que apoyan la reutilización de software

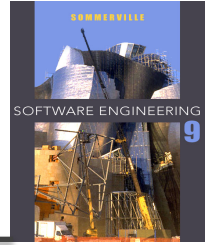


Enfoque	Descripción
Sistemas orientados al servicio	Los sistemas son desarrollados mediante la vinculación de los servicios compartidos, que pueden ser proporcionados externamente. [Descrito en el capítulo 19]
Líneas de producto de software	Un tipo de aplicación es generalizado alrededor de arquitecturas comunes por lo que pueden ser adaptadas para diferentes clientes.
COTS reutilización de productos	Los sistemas son desarrollados mediante la configuración e integración existente en sistemas de aplicaciones.
Sistemas ERP	Sistemas de gran escala que encapsulan funcionalidad, y reglas genéricas empresariales son configuradas por una organización.
Aplicaciones verticales configurables	Sistemas genéricos son diseñados de tal forma que puedan ser configurados a las necesidades específicas del sistema del cliente.

Enfoques que apoyan la reutilización de software



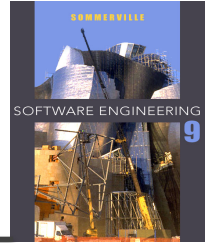
Enfoque	Descripción
Biblioteca de programas	Biblioteca de clases y funciones ,que comúnmente se emplean utilizando abstracciones , están disponibles para reutilizar.
Ingeniería dirigida por modelos	El software se representa como modelos de dominio y modelos independientes de aplicación el código se genera a partir de estos modelos. [Descrito en el capítulo 5.]
Generadores de programas	Un sistema generador contiene conocimiento de un tipo de aplicación y es utilizado para generar sistemas en ese dominio donde un usuario supe un modelo de sistema.
Desarrollo de software orientado a aspectos	Los componentes compartidos se enlazan en una aplicación en diferentes lugares cuando se compila el programa. [Descrito en el capítulo 21.]



Reutilización: factores de planificación

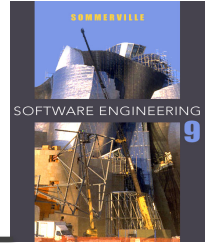
- ✧ El calendario de desarrollo para el software.
- ✧ La expectativa de vida del software.
- ✧ Los antecedentes, habilidades y experiencia del equipo de desarrollo.
- ✧ Lo crítico del software y sus requerimientos no funcionales.
- ✧ El dominio de la aplicación.
- ✧ La plataforma de ejecución para el software

Estructuras de aplicaciones



- ✧ Las estructuras son entidades moderadamente grandes que pueden ser reutilizadas. Están en algún lugar entre el sistema y el componente de reutilización.
- ✧ Las estructuras son el diseño de un subsistema hecho para la colección de clases abstractas y concretas y las interfaces entre ellas.
- ✧ El subsistema es implementado añadiendo componentes para llenar partes del diseño y creando instancias de las clases abstractas de la estructura.

Estructura de clases



✧ Marcos de infraestructura del sistema.

- Soporte de desarrollo de infraestructura de sistemas como comunicaciones, interfaces de usuario y compiladores.

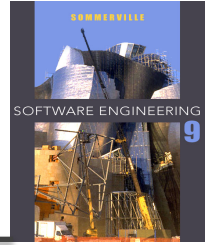
✧ Marco de integración middleware

- Los estándares y clases que soportan componentes de comunicación e intercambio de información.

✧ Marco de aplicación empresarial.

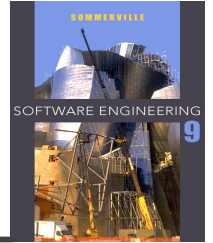
- Soporta el desarrollo de tipos específicos de aplicaciones como de telecomunicaciones o sistemas financieros.

Estructura de aplicaciones WEB



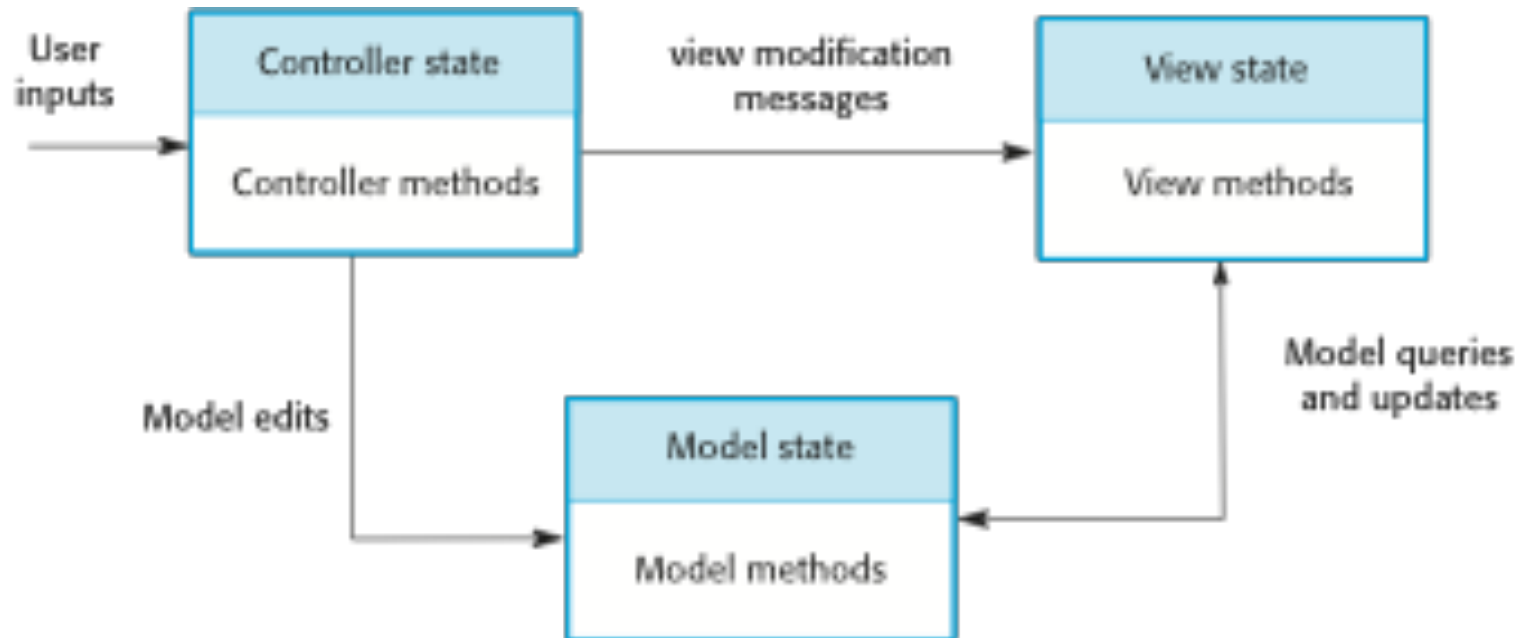
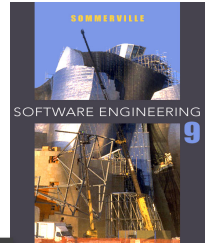
- ✧ Apoya la construcción de sitios web dinámicos como un front-end para aplicaciones web.
- ✧ WAFs están ahora disponibles para todos los usos comunes de lenguajes de programación web, ejemplos: Java, Python, Ruby, etc.
- ✧ Modelo de interacción se basa en el patrón compuesto Modelo-Vista-Controlador.

Modelo-Vista controlador

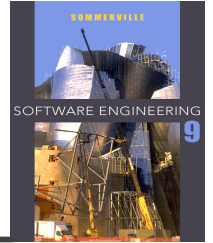


- ✧ Marco de infraestructura de sistema para diseño GUI.
- ✧ Permite múltiples presentaciones de un objeto y las interacciones individuales con estas presentaciones.
- ✧ Marco de trabajo MVC involucra la instalación de un número de patrones (como se explica en el Capítulo 7).

Patrón Modelo-Vista controlador

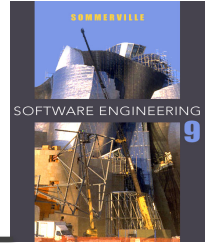


Características de WAF



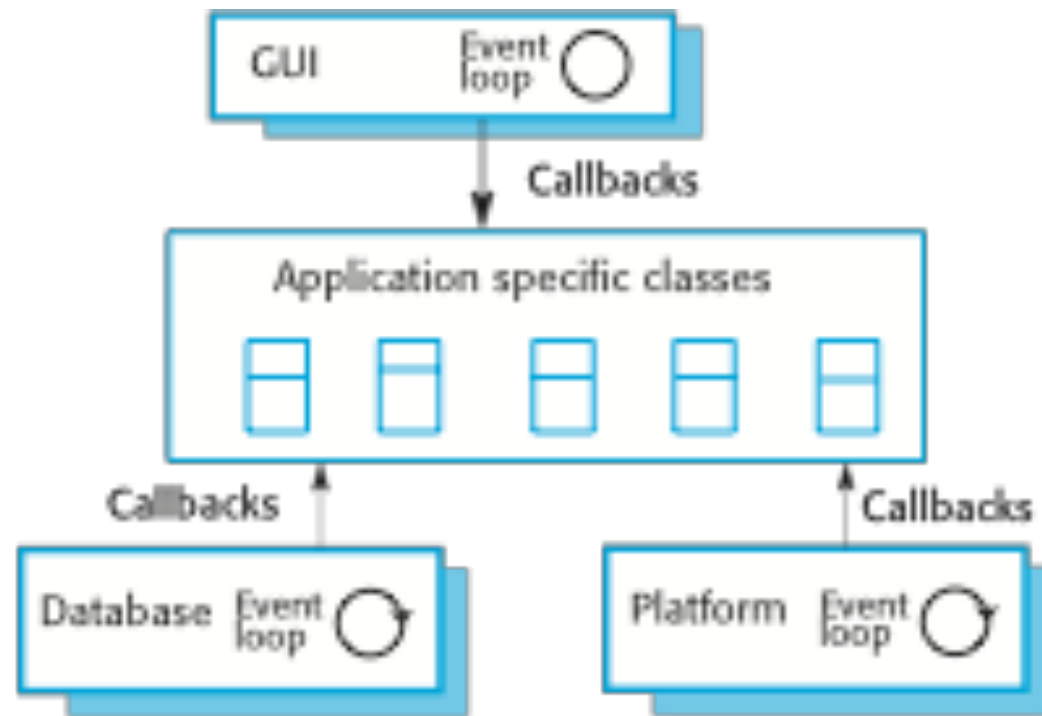
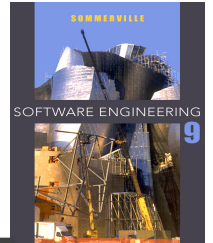
- ✧ Seguridad
 - WAFs puede incluir clases para ayudar a implementar la autenticación de usuario (login) y el acceso.
- ✧ Páginas web dinámicas
 - Las clases se ofrecen para ayudar a definir las plantillas de página web y para rellenar estas dinámicamente a partir de la base de datos del sistema.
- ✧ Soporte de base de datos
 - La estructura puede proporcionar clases que provean una interfaz abstracta de diferentes bases de datos.
- ✧ Gestión de sesiones
 - Las clases para crear y gestionar las sesiones (un número de interacciones con el sistema por un usuario) suelen ser parte de un WAF.
- ✧ La interacción del usuario
 - La mayoría de las estructuras web ofrecen ahora soporte AJAX (Holdener, 2008), lo que permite la creación de páginas web más interactivas.

Estructuras extensibles

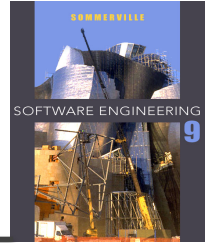


- ✧ Las estructuras son genéricas y extensibles para crear una aplicación o subsistema más específico. Ellos proveen un esqueleto arquitectónico para el sistema.
- ✧ La ampliación de la estructura incluye:
 - Adición de clases concretas que heredan las operaciones de las clases abstractas en la estructura.
 - Adición de métodos que se llaman en respuesta a eventos que son reconocidos por el marco.
- ✧ Problema con las estructuras es su complejidad lo que significa que se necesita mucho tiempo para usarlos con eficacia.

Inversión de control en Estructuras



Puntos clave

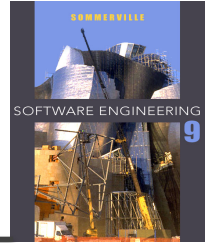


- ✧ Muchos nuevos sistemas de software empresariales son ahora desarrollados mediante la reutilización de conocimiento y código de sistemas ya implementados.
- ✧ Hay muchas diferentes formas de reutilizar software. Estos van desde la reutilización de clases y métodos en las bibliotecas a la reutilización de los sistemas de aplicación completos.
- ✧ Las ventajas de la reutilización de software son los costos más bajos, desarrollo de software más rápido y menores riesgos. Se incrementa la fiabilidad del sistema. Los especialistas pueden utilizarlos de manera más eficaz mediante la concentración de su experiencia en el diseño de componentes reutilizables.
- ✧ Estructuras de aplicación son conjuntos de objetos concretos y abstractos que se han diseñado para su reutilización a través de la especialización y la adición de nuevos objetos. Ellos suelen incorporar buenas prácticas de diseño a través de los patrones de diseño.

Capítulo 16- Reutilización de software

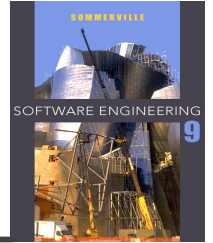
Lectura 2

Líneas de producto de software



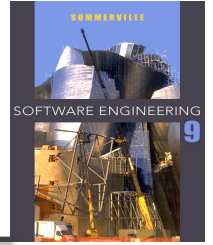
- ✧ Líneas de producto de software o familias de aplicaciones son aplicaciones con funcionalidades genéricas que pueden ser adaptadas y configuradas para su uso en un específico contexto.
- ✧ Una línea de producto de software es un grupo de aplicaciones con una arquitectura en común y componentes compartidos, con cada aplicación especializada para reflejar diferentes requerimientos.
- ✧ La adaptación puede incluir:
 - Componentes y sistemas de configuración.
 - Adición de nuevos componentes al sistema.
 - Selección de componentes existentes de la biblioteca.
 - Modificación de componentes para cumplir con los nuevos requerimientos.

Marcos de aplicación y líneas de productos



- ✧ Marcos de aplicación se basan en características orientadas a objetos como el polimorfismo para implementar extensiones. Las líneas de productos no necesitan ser (por ejemplo, software embebido para un teléfono móvil) orientadas a objetos
- ✧ Marcos de aplicación se centran en la prestación de asistencia técnica en lugar de apoyo específico del dominio. Las líneas de productos incrustan información del dominio y la plataforma.
- ✧ Las líneas de productos a menudo controlan las aplicaciones para equipos.
- ✧ Líneas de productos de software se componen de una familia de aplicaciones, por lo general pertenecen a la misma organización.

Especialización de Línea de producto



✧ Especialización de plataforma

- Las diferentes versiones de la aplicación se han desarrollado para diferentes plataformas.

✧ Especialización de medio ambiente

- Las diferentes versiones de la aplicación se crean para manejar diferentes entornos operativos por ejemplo diferentes tipos de equipos de comunicación.

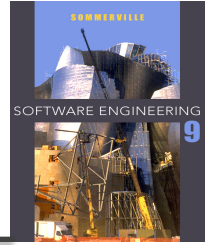
✧ Especialización funcional

- Las diferentes versiones de la aplicación se crean con diferentes necesidades.

✧ Especialización de proceso

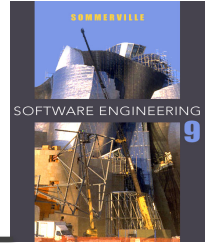
- Las diferentes versiones de la aplicación se crean para apoyar diferentes procesos de negocio.

Arquitectura de línea de producto



- ✧ Arquitecturas deben estructurarse de tal manera que separaren los diferentes sub-sistemas para que puedan ser modificados.
- ✧ La arquitectura también debe separar entidades y sus descripciones y los niveles más altos en las entidades de acceso al sistema a través de las descripciones en lugar de hacerlo directamente.

La arquitectura de un sistema de asignación de recursos



Interaction

User interface

I/O management

User
authentication

Resource
delivery

Query
management

Resource management

Resource
tracking

Resource policy
control

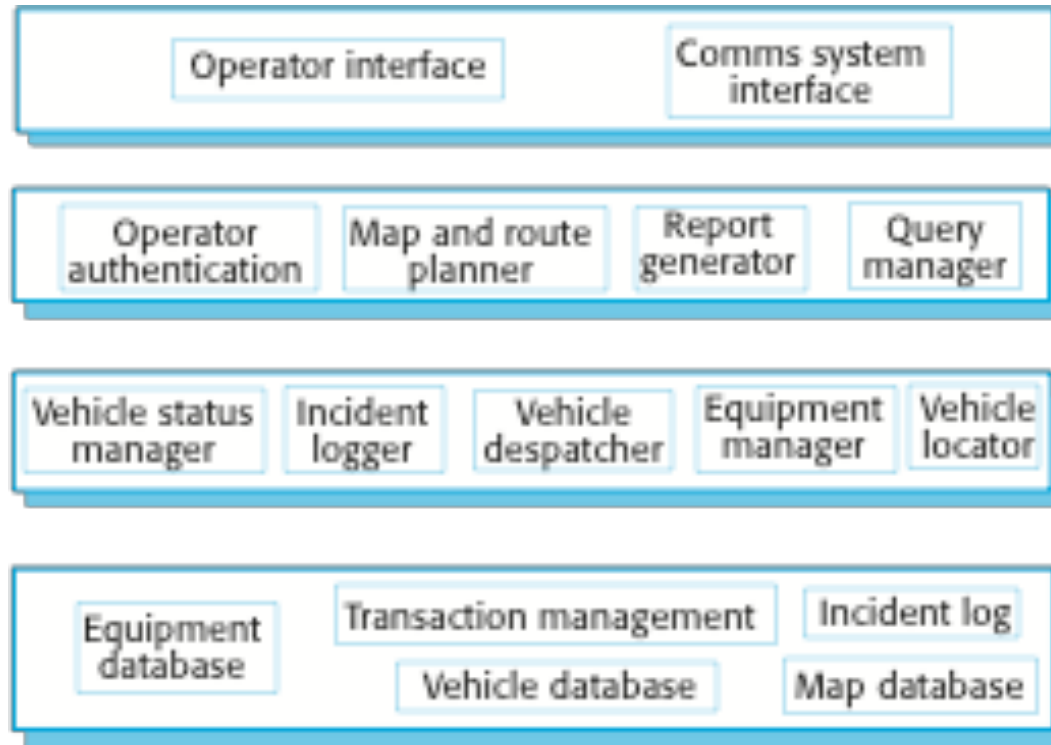
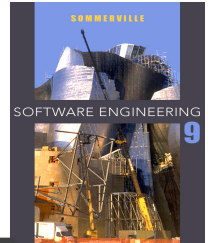
Resource
allocation

Database management

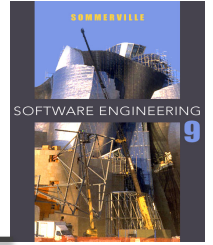
Transaction management

Resource database

La arquitectura de la línea de productos de un despachador de vehículos

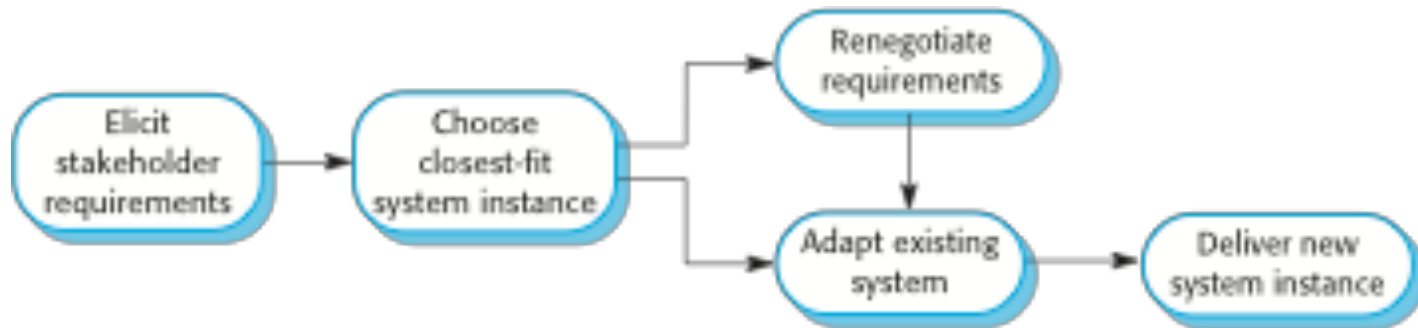
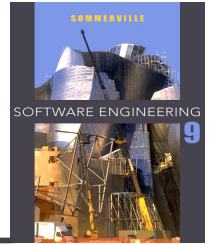


Despacho de vehículos

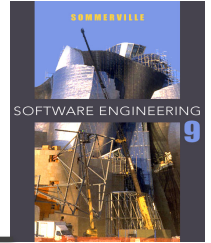


- ✧ Un sistema de gestión de recursos especializados, donde el objetivo es asignar recursos (vehículos) para manejar incidentes.
- ✧ Las adaptaciones incluyen:
 - A nivel de interfaz de usuario, hay componentes de visualización del operador y las comunicaciones;
 - A nivel de gestión de E / S, hay componentes que se encargan de la autenticación, generación de informes y la planificación de la ruta;
 - A nivel de gestión de recursos, hay componentes para la localización de vehículos y de expedición, la gestión del estado del vehículo y registro de incidentes;
 - La base de datos incluye las bases de datos de equipos, vehículos y mapas

Desarrollo instancia del producto

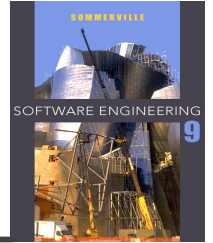


Desarrollo de instancia del producto



- ✧ Explicar requisitos de los interesados
 - Miembro familiar existente usado como prototipo
- ✧ Elija un miembro familiar cercano que ajuste
 - Encuentre el miembro de la familia que mejor cumpla con los requisitos.
- ✧ Renegociar los requisitos
 - Adaptar los requisitos que sean necesarios para las capacidades del software.
- ✧ Adaptar el sistema existente
 - Desarrollo de nuevos módulos y hacer modificaciones para miembros de la familia.
- ✧ Enviar nuevo miembro de la familia
 - Características clave del documento para un mayor desarrollo del miembro.

Configuración línea de producto



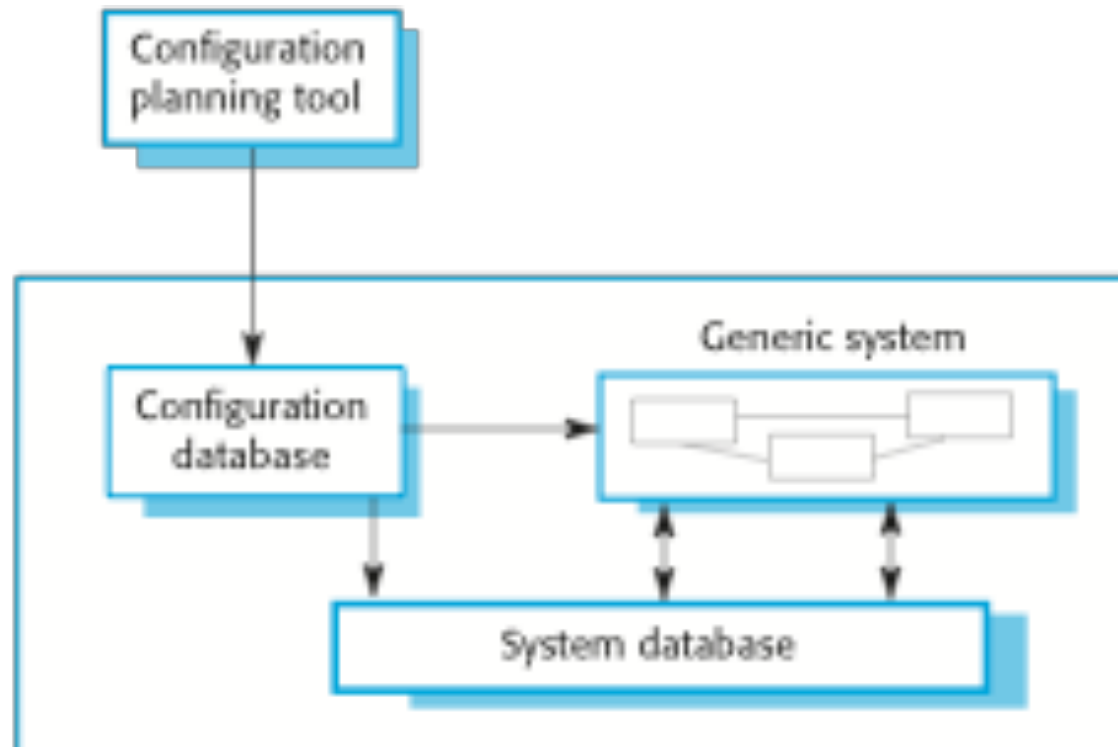
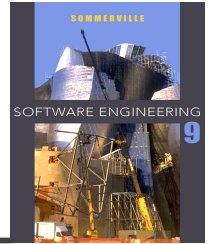
✧ Configuración en tiempo de diseño

- La línea de producto se adapta y cambia de acuerdo a los requerimientos de clientes en particular.

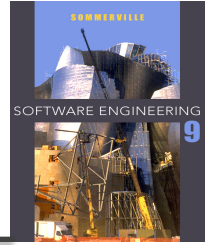
✧ Configuración en tiempo de despliegue

- La línea de productos se configura mediante la incorporación de conocimiento de los requisitos del cliente y los procesos de negocio. El código fuente del software en sí, no se cambia.

Configuración de tiempo de despliegue



Niveles de configuración de tiempo de despliegue

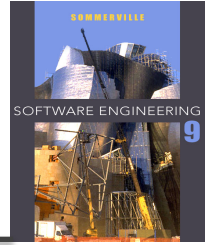


- ✧ Selección de componentes, dónde eliges los modulo en un sistema que provee la funcionalidad necesaria.
- ✧ Flujo de trabajo y definición de reglas, donde se definen los flujos de trabajo (cómo se procesa la información, etapa por etapa) y las reglas de validación que debe aplicarse a la información introducida por los usuarios o generada por el sistema.
- ✧ Definición de parámetros, donde se especifican los valores de los parámetros específicos del sistema que reflejen la instancia de la aplicación que está creando.

COTS reutilización de producto

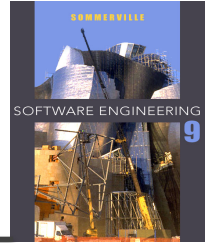
- ✧ Un producto de comercial fuera de la plataforma (COTS) es un sistema de software que puede ser adaptado para diferentes clientes sin cambiar el código fuente del sistema.
- ✧ Sistemas COTS tienen características genéricas y así se puede utilizar / reutilizar en diferentes entornos.
- ✧ Productos COTS se adaptan mediante el uso de mecanismos integrados de configuración que permiten que la funcionalidad del sistema se adapte a las necesidades específicas del cliente.
 - Por ejemplo, registro del paciente en un sistema hospitalario, formas de entrada separados y los informes de salida pueden ser definidos para diferentes tipos de pacientes.

Beneficios de reutilizar COTS



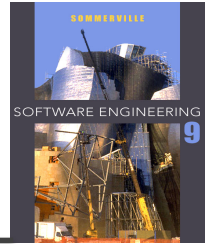
- ✧ Al igual que con otros tipos de reutilización, un despliegue más rápido de un sistema fiable puede ser posible.
- ✧ Es posible ver qué funcionalidad es proporcionada por las aplicaciones y lo que es más fácil juzgar si es o no probable que sean adecuados.
- ✧ Algunos riesgos de desarrollo se evitan mediante el uso de software existente. Sin embargo, este enfoque tiene sus propios riesgos, como se explica a continuación.
- ✧ Las empresas pueden centrarse en su actividad principal, sin tener que dedicar una gran cantidad de recursos para el desarrollo de sistemas de TI.
- ✧ Como las plataformas operativas evolucionan, actualizaciones tecnológicas pueden simplificarse ya que estos son la responsabilidad del proveedor del producto COTS y no el cliente.

Problemas de reutilizar COTS



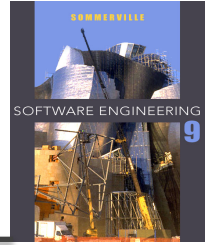
- ✧ Requisitos tienen generalmente que ser adaptados para reflejar la funcionalidad y el modo de funcionamiento del producto COTS.
- ✧ El producto COTS puede basarse en suposiciones que son prácticamente imposibles de cambiar.
- ✧ Elegir el sistema COTS adecuado para una empresa puede ser un proceso difícil, sobre todo porque muchos de los productos COTS no están bien documentados.
- ✧ Puede haber una falta de experiencia local para apoyar el desarrollo de sistemas.
- ✧ El control de los proveedores de productos de soporte del sistema COTS y evolución.

Solución COTS y sistemas integrados COTS



<i>Sistemas de solución COTS</i>	<i>Sistemas de integración COTS</i>
Producción simple que proporciona la funcionalidad requerida por el cliente.	Muchos sistemas de productos heterogéneos e integrados proporcionan funcionalidad personalizada.
Basado en soluciones genéricas y procesos estandarizados.	Soluciones flexibles pueden ser desarrolladas por procesos personalizados.
Desarrollo focalizado es en sistemas de configuración.	Desarrollo focalizado es un sistema de integración
Proveedor del sistema es responsable del mantenimiento	Propietario del sistema es responsable del mantenimiento.
Proveedor del sistema proporciona la plataforma para el sistema	Propietario del sistema proporciona la plataforma para el sistema

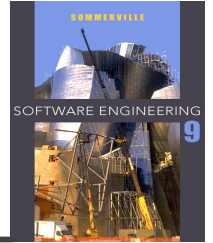
Sistemas de solución COTS



- ✧ Los sistemas de soluciones COTS son sistemas de aplicaciones genéricos que pueden ser diseñados para soportar un tipo particular de negocio, actividad empresarios o, a veces, un negocio entero de una empresa.
 - Por ejemplo, un sistema de solución COTS pueden ser producidos para dentistas que manejan citas dentales, historiales dentales, recuperación del paciente, etc.
- ✧ Sistemas de solución COTS de dominios específicos, tales como sistemas de apoyo a una función empresarial (por ejemplo, gestión de documentos) proporcionar la funcionalidad que es probable se requiera por una gama de usuarios potenciales.

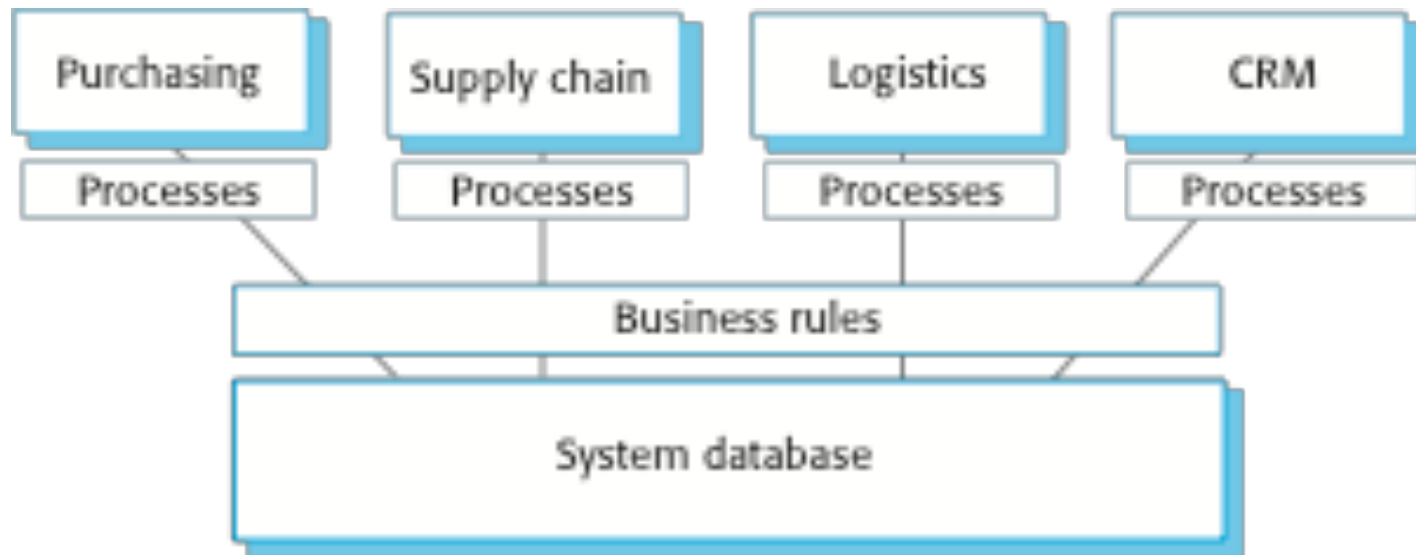
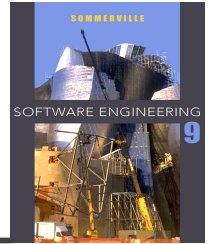


Sistemas ERP

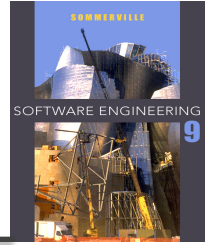


- ✧ Un sistema de planificación de recursos empresariales (ERP) es un sistema genérico que soporta los procesos de negocio comunes, tales como pedidos y facturación, fabricación, etc.
- ✧ Estos son muy ampliamente utilizados en grandes empresas - que representan probablemente la forma más común de la reutilización de software.
- ✧ El núcleo genérico se adapta mediante la inclusión de módulos e incorporando el conocimiento de los procesos de negocio y las reglas.

Arquitectura de un sistema ERP

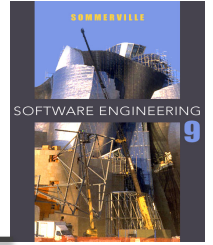


Arquitectura ERP



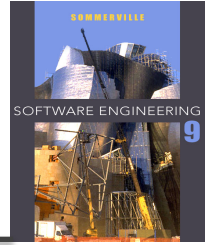
- ✧ Un número de módulos soportan diferentes funciones de negocio.
- ✧ Un conjunto definido de procesos de negocio, asociado con cada módulo, que se relacionan a las actividades en ese módulo.
- ✧ Una base de datos común que mantiene la información sobre todas las funciones comerciales relacionadas.
- ✧ Un conjunto de reglas de negocio que se aplican a todos los datos en la base de datos.

Configuración ERP



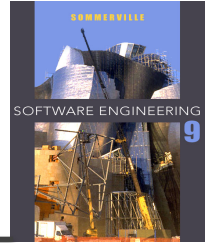
- ✧ Selección de la funcionalidad requerida por el sistema.
- ✧ El establecimiento de un modelo de datos que define cómo se estructurarán los datos de la organización en la base de datos del sistema.
- ✧ Definición de reglas de negocio que se aplican a esos datos.
- ✧ Definición las interacciones esperadas con sistemas externos.
- ✧ El diseño de los formularios de entrada y los informes de salida generados por el sistema.
- ✧ El diseño de nuevos procesos de negocio que se ajustan al modelo de proceso subyacente apoyado por el sistema.
- ✧ Ajuste de los parámetros que definen cómo se implementa el sistema en su plataforma subyacente.

Sistemas integrados COTS



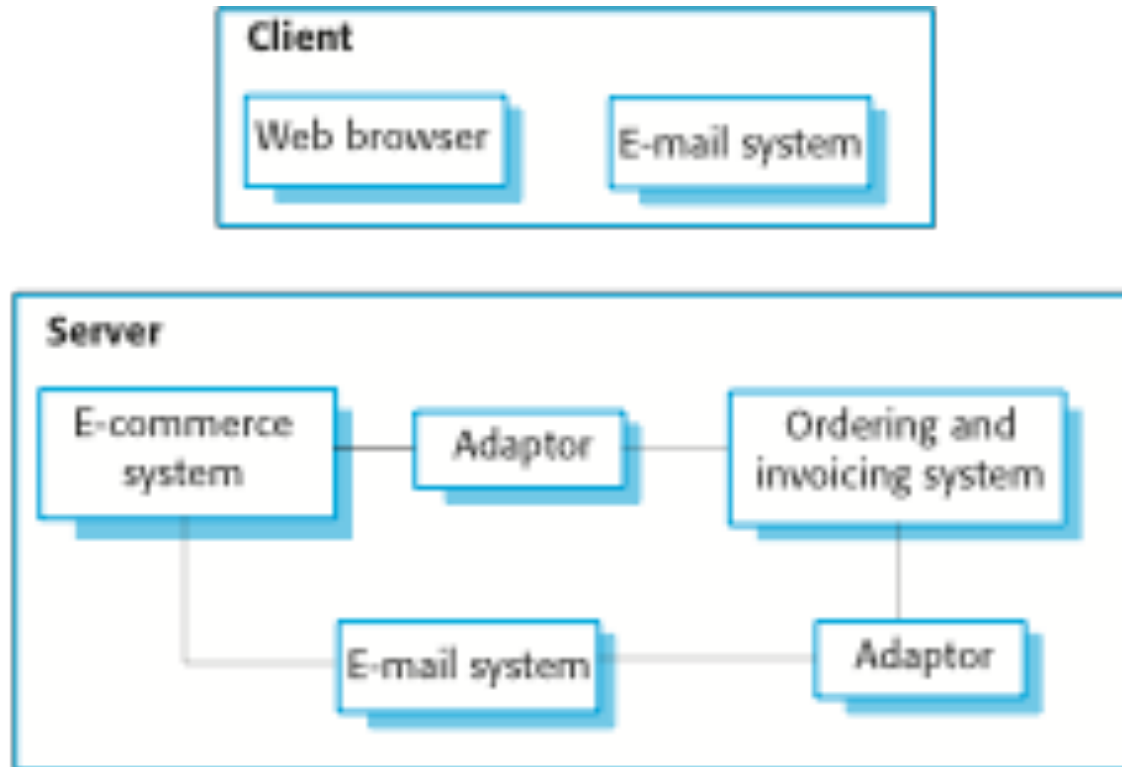
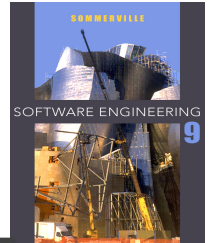
- ✧ Sistemas integrados-COTS son aplicaciones que incluyen dos o más productos COTS y / o sistemas de aplicaciones heredadas.
- ✧ Usted puede usar este método cuando no hay un sistema COTS única que cumple con todas sus necesidades de tiempo o cuándo desea de integrar un nuevo producto COTS con los sistemas que ya utiliza.

Decisiones de diseño

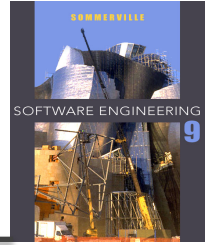


- ✧ ¿Qué productos COTS ofrecen la funcionalidad más adecuada?
- ✧ Típicamente, habrá varios productos COTS disponibles, que se pueden combinar de diferentes maneras.
- ✧ ¿Cómo se intercambiarán datos?
- ✧ Los diferentes productos que normalmente usan estructuras de datos únicas y formatos. Tienes que escribir adaptadores que convierten de una representación a otra.
- ✧ ¿Qué características de un producto puedan efectivamente utilizar?
- ✧ Productos COTS pueden incluir más funcionalidad que usted necesita y la funcionalidad se puede duplicar a través de diferentes productos.

Un sistema de adquisiciones COTS integrada

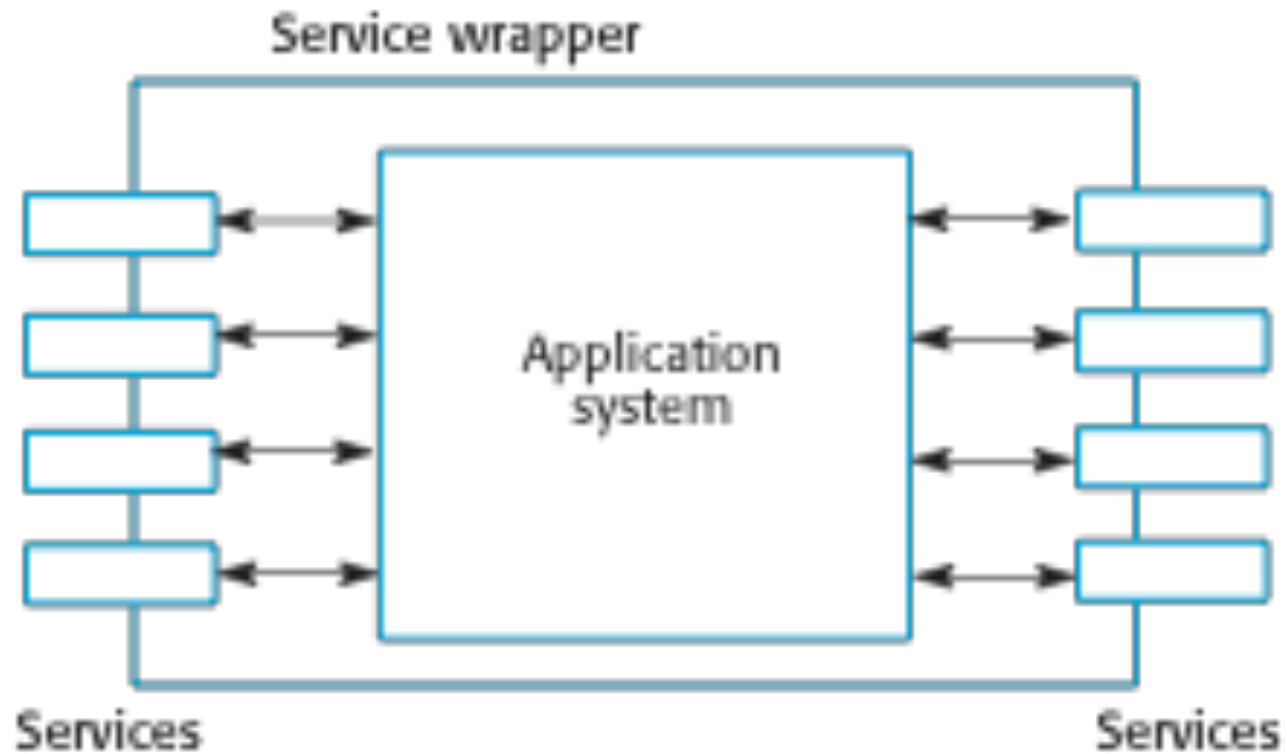
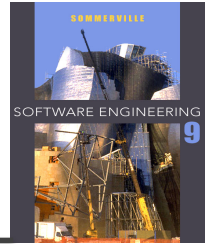


Interfaces de COTS orientadas a servicios

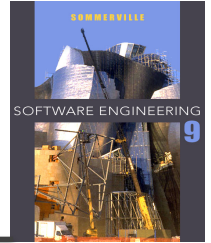


- ✧ Integración COTS pueden simplificarse si se utiliza un enfoque orientado al servicio.
- ✧ Un enfoque orientado a servicios significa permitir el acceso a la funcionalidad del sistema de aplicación a través de una interfaz de servicio estándar, con un servicio para cada unidad discreta de funcionalidad.
- ✧ Algunas aplicaciones pueden ofrecer una interfaz de servicio, pero, a veces, esta interfaz de servicio tiene que ser implementada por el integrador de sistemas. Usted tiene que programar un envoltorio que oculta la aplicación y proporciona servicios visibles externamente.

Cubierta de aplicación

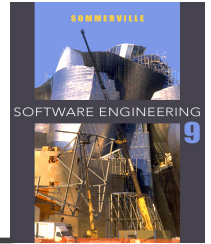


Problemas de integración de sistemas COTS



- ✧ La falta de control sobre la funcionalidad y el rendimiento
 - Sistemas COTS pueden ser menos eficaces de lo que parecen
- ✧ Problemas de interoperabilidad con el sistema COTS
 - Diferentes sistemas COTS pueden hacer diferentes supuestos que significa la integración es difícil
- ✧ No hay control sobre la evolución del sistema
 - Proveedores COTS no son usuarios de los sistemas de control de la evolución
- ✧ El apoyo de los proveedores COTS
 - Proveedores COTS no pueden ofrecer apoyo a lo largo de la vida útil del producto.

Puntos clave



- ✧ Líneas de productos de software son las aplicaciones relacionadas, que se desarrollan a partir de una base común. Este sistema genérico está adaptado para cumplir los requisitos específicos para la funcionalidad, la plataforma de destino o configuración operativa.
- ✧ Reutilización producto COTS se refiere a la reutilización de gran escala, sistemas fuera de biblioteca. Estos proporcionan una gran cantidad de funcionalidad y su reutilización puede reducir radicalmente los costos y el tiempo de desarrollo. Los sistemas pueden ser desarrollados mediante la configuración de un producto COTS único, genérico o mediante la integración de dos o más productos COTS.
- ✧ Sistemas de planificación de recursos empresariales son ejemplos de reutilización COTS a gran escala. Se crea una instancia de un sistema ERP mediante la configuración de un sistema genérico con información sobre los procesos y reglas de negocio del cliente.
- ✧ Los problemas potenciales con la reutilización basado en COTS incluyen la falta de control sobre la funcionalidad y el rendimiento, la falta de control sobre la evolución del sistema, la necesidad de apoyo de los vendedores y las dificultades para asegurar que los sistemas puedan interoperar.