



UNIVERSITÀ
DEGLI STUDI
DI PALERMO

Riutilizzo del Software

Ingegneria del Software

Argomenti

- ~ Panoramica sul riutilizzo
- ~ Framework applicativi
- ~ Linee di prodotto software
- ~ Riutilizzo dei sistemi applicativi

Riutilizzo del Software

- ~ Nella maggior parte delle discipline ingegneristiche, i sistemi sono progettati componendo componenti esistenti che sono stati usati in altri sistemi.
- ~ L'ingegneria del software è stata più focalizzata sullo sviluppo from-scratch
 - ora è riconosciuto che per ottenere un software migliore, più rapidamente e a costi inferiori, abbiamo bisogno di un processo di progettazione che si basa sul riutilizzo sistematico del software.
- ~ C'è stato un importante passaggio allo sviluppo basato sul **riuso** negli ultimi 10 anni.
 - perché?
 - richiesta di riduzione costi di produzione, manutenzione, miglioramento della qualità

Software Engineering basata sul riuso

~ Riutilizzo del sistema

- Sistemi interi, che possono includere diversi programmi applicativi, possono essere riutilizzati.

~ Riutilizzo delle applicazioni

- Un'applicazione può essere riutilizzata incorporandola senza modifiche in altre o sviluppando famiglie di applicazioni.

~ Riutilizzo dei componenti

- I componenti di un'applicazione, dai sottosistemi ai singoli oggetti, possono essere riutilizzati.

~ Riutilizzo di oggetti e funzioni

- Componenti software su piccola scala che implementano un singolo oggetto o funzione ben definita possono essere riutilizzati.

Software Engineering basata sul riuso

- ~ Sistemi e componenti a volte sono molto specifici
- ~ Diventa costoso riadattarli a nuove situazioni
- ~ Invece di riutilizzare codice si possono riutilizzare le idee
- ~ Riutilizzo concettuale
 - riutilizzare un modo di operare o un algoritmo

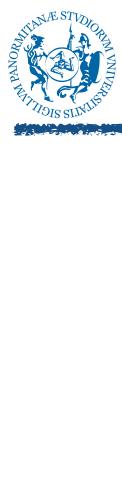
Benefici del riutilizzo

Benefit	Explanation
Sviluppo accelerato	L'introduzione di un sistema sul mercato il prima possibile è spesso più importante dei costi complessivi di sviluppo. Il riutilizzo del software può accelerare la produzione del sistema, perché i tempi di sviluppo e di validazione possono essere ridotti.
Uso efficace degli specialisti	Invece di fare sempre lo stesso lavoro, gli specialisti delle applicazioni possono sviluppare software riutilizzabile che racchiude le loro conoscenze.
Maggiore affidabilità	Il software riutilizzato, che è stato provato e testato in sistemi funzionanti, dovrebbe essere più affidabile del software nuovo. I suoi difetti di progettazione e implementazione dovrebbero essere stati individuati e risolti.



Benefici del riutilizzo

Benefit	Explanation
Riduzione dei costi di sviluppo	I costi di sviluppo sono proporzionali alle dimensioni del software da sviluppare. Riutilizzare il software significa scrivere meno righe di codice.
Rischio di processo ridotto	Il costo del software esistente è già noto, mentre i costi di sviluppo sono sempre una questione di giudizio. Questo è un fattore importante per la gestione dei progetti, perché riduce il margine di errore nella stima dei costi del progetto. Ciò è particolarmente vero quando vengono riutilizzati componenti software relativamente grandi, come i sottosistemi.
Conformità agli standards	Alcuni standard, come quelli dell'interfaccia utente, possono essere implementati come un insieme di componenti riutilizzabili. Ad esempio, se i menu di un'interfaccia utente sono implementati con componenti riutilizzabili, tutte le applicazioni presentano agli utenti gli stessi formati di menu. L'uso di interfacce utente standard migliora l'affidabilità, perché gli utenti commettono meno errori quando si presentano con un'interfaccia familiare.



Problemi con il riuso

Problem	Explanation
Creazione, manutenzione e utilizzo di una libreria di componenti	Popolare una libreria di componenti riutilizzabili e garantire che gli sviluppatori di software possano utilizzarla può essere costoso. I processi di sviluppo devono essere adattati per garantire l'utilizzo della libreria.
Individuare, comprendere e adattare componenti riutilizzabili	I componenti software devono essere individuati in una libreria, compresi e, talvolta, adattati per funzionare in un nuovo ambiente. Gli ingegneri devono essere ragionevolmente sicuri di trovare un componente nella libreria prima di includere una ricerca di componenti come parte del loro normale processo di sviluppo.
Aumento dei costi di manutenzione	Se il codice sorgente di un sistema o di un componente software riutilizzato non è disponibile, i costi di manutenzione possono essere più elevati perché gli elementi riutilizzati del sistema possono diventare sempre più incompatibili con le modifiche del sistema.

Problemi con riuso

Problem	Explanation
Mancanza di supporto agli strumenti	Alcuni strumenti software non supportano lo sviluppo con riuso. Può essere difficile o impossibile integrare questi strumenti con un sistema di librerie di componenti. Il processo software assunto da questi strumenti potrebbe non tenere conto del riuso. Questo è particolarmente vero per gli strumenti che supportano l'ingegneria dei sistemi embedded, meno per gli strumenti di sviluppo orientati agli oggetti.
Sindrome da "non inventato qui"	Alcuni ingegneri del software preferiscono riscrivere i componenti perché ritengono di poterli migliorare. Ciò ha a che fare in parte con la fiducia e in parte con il fatto che la scrittura di software originale è considerata più impegnativa del riutilizzo di software altrui.

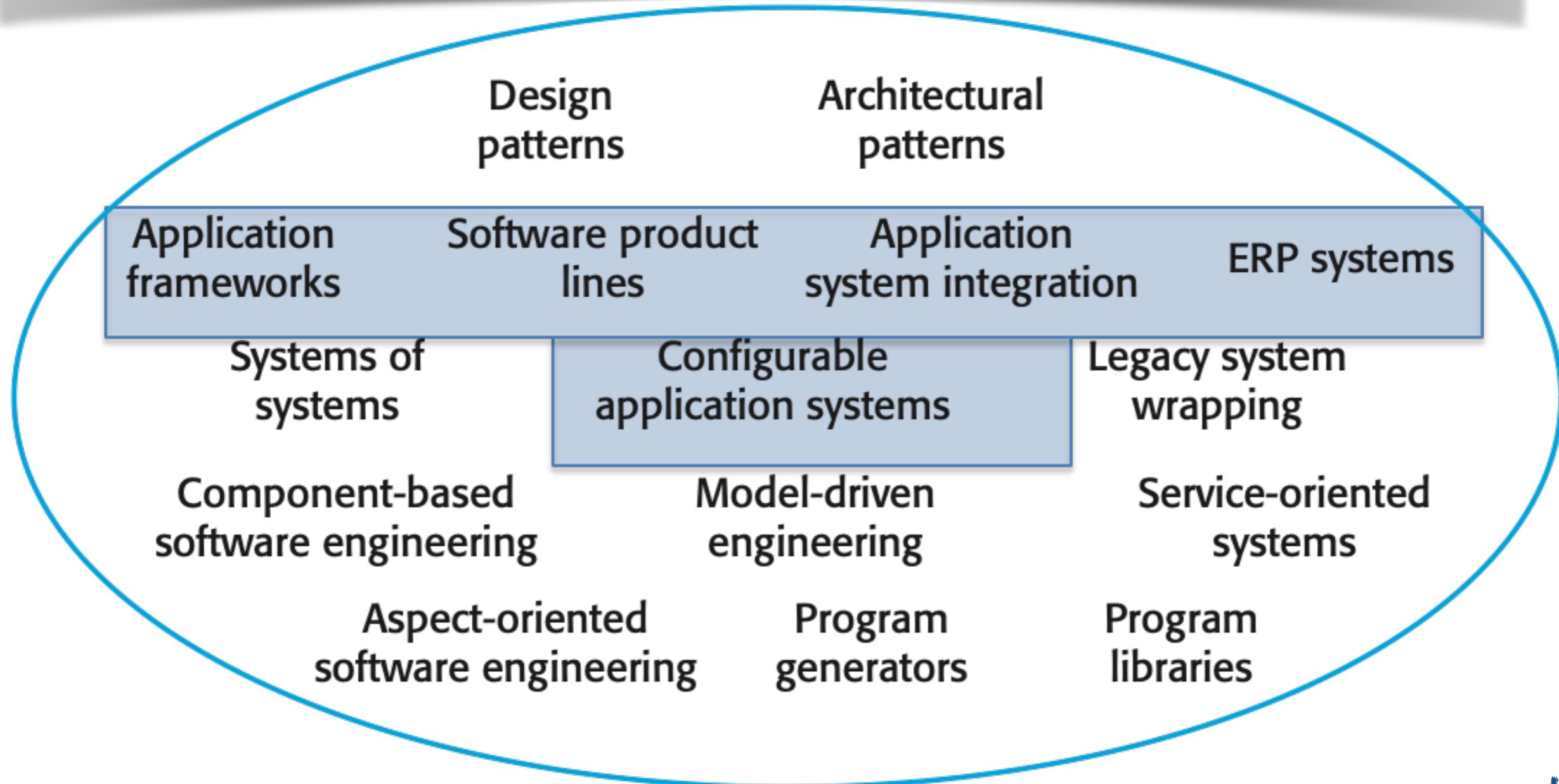


Panoramica sul riutilizzo

Panoramica sul riutilizzo

- ~ Negli ultimi vent'anni molte tecniche per il riutilizzo del software
 - sistemi dello stesso dominio applicativo sono simili
- ~ molti approcci diversi al riutilizzo che possono essere usati.
- ~ Il riutilizzo è possibile a vari livelli, da semplici funzioni a sistemi applicativi completi.
- ~ Gli standard dei componenti riutilizzabili ne facilitano il riutilizzo

Diversi metodi per supportare il riutilizzo



~ Descritti nella slide seguente

Metodi che supportano il riutilizzo del software

Approach	Description
Application frameworks	Collections of abstract and concrete classes are adapted and extended to create application systems.
Application system integration	Two or more application systems are integrated to provide extended functionality
Architectural patterns	Standard software architectures that support common types of application system are used as the basis of applications. Described in Chapters 6, 11 and 17.
Aspect-oriented software development	Shared components are woven into an application at different places when the program is compiled. Described in web chapter 31.
Component-based software engineering	Systems are developed by integrating components (collections of objects) that conform to component-model standards. Described in Chapter 16.

Metodi che supportano il riutilizzo del software

Approach	Description
Program generators	A generator system embeds knowledge of a type of application and is used to generate systems in that domain from a user-supplied system model.
Program libraries	Class and function libraries that implement commonly used abstractions are available for reuse.
Service-oriented systems	Systems are developed by linking shared services, which may be externally provided. Described in Chapter 18.
Software product lines	An application type is generalized around a common architecture so that it can be adapted for different customers.
Systems of systems	Two or more distributed systems are integrated to create a new system. Described in Chapter 20.

Metodi che supportano il riutilizzo del software

~ Domande chiave:

- qual'è la tecnica più adatta da utilizzare in un particolare situazione?
- ~ Tutto dipende dai requisiti del sistema che si sta sviluppando, dalle tecnologie e dai componenti riutilizzabili di cui si dispone
- ~ ...ed anche dalla esperienza delle persone nel team di sviluppo

Fattori chiave per il riutilizzo

- ~ Tempi di sviluppo del software
 - se a seconda dei tempi si può decidere se riutilizzare un software completo o solo componenti
- ~ La durata prevista del software.
 - vari fattori: in un sistema che deve durare a lungo ci si concentra sulla manutenibilità, se i requisiti sono mutevoli durante lo sviluppo e non si ha accesso al codice dei componenti è meglio riusare componenti open source
- ~ Il background, le competenze e l'esperienza del team di sviluppo.
 - le tecnologie di riutilizzo sono complesse -> tempo per capire e sviluppare -> concentrarsi in aree dove il team ha esperienza.
- ~ La criticità del software e suoi requisiti non funzionali.
 - per sistemi critici che devono essere certificati si dovrebbe creare un caso di fidatezza e questo è difficile se non si ha accesso al codice
- ~ Il dominio dell'applicazione.
 - in molti domini i prodotti possono essere riutilizzati con apposite configurazioni
- ~ La piattaforma di esecuzione del software.
 - alcuni modelli di componenti come .NET sono specifici per piattaforme Microsoft e non possono essere riutilizzati per sistemi progettati per altre piattaforme



Frameworks applicativi

Definizione di framework

- ~ "...un insieme integrato di artefatti software (come classi, oggetti e componenti) che collaborano per fornire un'architettura riutilizzabile per una famiglia di applicazioni correlate."
- ~ questo concetto è venuto fuori dalla realizzazione che gli oggetti non possono essere riutilizzati sempre
 - spesso sono troppo accurati e specifici per una singola applicazione
 - oggi sappiamo che il riutilizzo orientato agli oggetti è supportato meglio in un processo di sviluppo orientato agli oggetti tramite astrazioni più generiche dette framework
 - un framework quindi è una struttura generica che viene estesa per creare un sottosistema più specifico

Framework applicativi

- ~ I framework sono entità moderatamente grandi che possono essere riutilizzate. Sono a metà strada tra il riutilizzo di sistemi e componenti.
- ~ I framework sono un progetto di sotto-sistema composto da una **collezione di classi astratte e concrete** e le interfacce tra di loro.
 - Il sottosistema è implementato aggiungendo componenti per riempire parti del progetto e istanziando le classi astratte nel framework.
 - Forniscono il supporto a generiche funzionalità
- ~ I framework supportano il riutilizzo dei progetti in quanto forniscono lo scheletro dell'architettura di un'applicazione
- ~ Per esempio il **framework di un'interfaccia utente** fornirà il supporto alla gestione degli eventi dell'interfaccia e includerà una serie di strumenti per costruire nuove voci

Web application framework

- ~ I framework più diffusi sono quelli utilizzati per realizzare applicazioni web (Web Application Framework o WAF)
- ~ Supportano la costruzione di siti web dinamici come front-end per applicazioni web.
- ~ I WAF sono ora disponibili per tutti i linguaggi di programmazione web comunemente usati, ad esempio Java, Python, Ruby, ecc.
- ~ Il modello di interazione è basato sul modello composito Model-View-Controller.

Web application framework – caratteristiche

- ~ I framework sono spesso implementazioni di schemi di progettazione (design pattern)
 - Per esempio una framework MVC include i pattern Observer, Strategy, Composite ed altri descritti nel Gamma.

Web application framework – caratteristiche

~ Sicurezza

- I WAF possono includere classi per aiutare a implementare l'autenticazione dell'utente (login) e l'accesso.

~ Pagine web dinamiche

- Sono fornite classi per aiutare a definire modelli di pagine web e a popolarle dinamicamente dal database del sistema.

~ Supporto al database

- Il framework può fornire classi che forniscono un'interfaccia astratta a diversi database.

~ Gestione delle sessioni

- Le classi per creare e gestire le sessioni (una serie di interazioni con il sistema da parte di un utente) sono solitamente parte di un WAF.

~ Interazione con l'utente

- La maggior parte dei framework web ora fornisce il supporto AJAX (Holdener, 2008), che permette di creare pagine web più interattive.

Estendere un framework

- ~ I framework sono generici e vengono estesi per creare un'applicazione o un sottosistema più specifico. Forniscono uno scheletro di architettura per il sistema.
- ~ Estendere il framework comporta:
 - Aggiungere classi concrete che ereditano operazioni dalle classi astratte del framework;
 - Aggiungere metodi che sono chiamati in risposta ad eventi che sono riconosciuti dal framework (callback)
- ~ Il problema con i framework è la loro complessità, il che significa che ci vuole molto tempo per usarli efficacemente.

Estendere un framework

- ~ Gli oggetti del framework e non quelli dell'applicazione sono responsabili del controllo del sistema
 - ↗ inversione di controllo
- ~ Al verificarsi di alcuni eventi nell'interfaccia utente e nel framework del database
 - ↗ gli oggetti del framework chiamano i metodi hook che sono collegati alle funzionalità create per l'utente.
- ~ Queste funzionalità definiscono come l'applicazione deve rispondere agli eventi.
- ~ Per esempio un framework avrà un metodo che gestisce i clic del mouse dall'ambiente operativo, questo metodo è l'hook, che può essere configurato per chiamare i metodi appropriati dell'applicazione che gestiscono i clic del mouse.
- ~ Il problema con i framework è la loro complessità, il che significa che ci vuole molto tempo per usarli efficacemente.

Classi di Framework

- ~ Sono state analizzate e definite tre classi di framework:
 - ~ Framework dell'infrastruttura del sistema
 - ~ Supportano lo sviluppo di infrastrutture di sistema come comunicazioni, interfacce utente e compilatori.
 - ~ Framework per l'integrazione di middleware
 - ~ Standard e classi che supportano la comunicazione dei componenti e lo scambio di informazioni tra componenti (esempio .NET o Enterprise Java Beans)
 - ~ Framework per applicazioni aziendali
 - ~ Supportano lo sviluppo di specifici tipi di applicazioni come le telecomunicazioni o i sistemi finanziari.
 - ~ non molto utilizzati, sostituiti dalle linee di prodotto

Linee di prodotti software

- ~ Quando una società deve supportare un certo numero di sistemi simili, non identici, uno degli approcci più efficaci al **riutilizzo** e creare una **linea di prodotti software**
- ~ Le linee di prodotti software o famiglie di applicazioni sono applicazioni con funzionalità generiche che possono essere adattate e configurate per l'uso in un contesto specifico.
- ~ Una linea di prodotti software è un insieme di applicazioni con un'architettura comune e componenti condivise, con ogni applicazione specializzata per riflettere requisiti diversi.
- ~ L'adattamento può coinvolgere:
 - Configurazione dei componenti e del sistema;
 - Aggiunta di nuovi componenti al sistema;
 - Selezione da una libreria di componenti esistenti;
 - Modifica dei componenti per soddisfare nuovi requisiti.

Sistema di base per una linea di prodotti

- ~ Progettare una linea di prodotti richiede l'identificazione delle funzionalità comuni nelle istanze dei prodotti
 - Lo sviluppo di una applicazione di base d riutilizzare
- ~ una applicazione di base include i seguenti elementi

Specialized application components

Configurable application
components

Core
components

Sistema di base per una linea di prodotti

- ~ Componenti di base che forniscono supporto all'infrastruttura. Questi di solito non vengono modificati quando si sviluppa una nuova istanza della linea di prodotti.
- ~ Componenti configurabili che possono essere modificati e configurati per specializzarli in una nuova applicazione. A volte, è possibile riconfigurare questi componenti senza cambiare il loro codice usando un linguaggio di configurazione dei componenti incorporato.
- ~ Componenti specializzati, specifici del dominio, alcuni o tutti i quali possono essere sostituiti quando viene creata una nuova istanza di una linea di prodotti.

Framework applicativi e linee di prodotti

- ~ I framework applicativi e le linee dei prodotti software hanno molto in comune
- ~ Entrambi supportano un'architettura e componenti comuni
- ~ Richiedono un nuovo processo di sviluppo per creare una versione specifica di un sistema
- ~ Differenze principali ->

Framework applicativi e linee di prodotti

- ~ I framework applicativi si basano su caratteristiche **orientate agli oggetti** come il polimorfismo per implementare le estensioni. Le linee di prodotto non devono necessariamente essere orientate agli oggetti (ad es. software embedded per un telefono cellulare)
- ~ I framework applicativi si concentrano sul fornire **supporto tecnico** piuttosto che specifico del dominio. Le linee di prodotto incorporano informazioni sul dominio e sulla piattaforma.
- ~ Le linee di prodotto spesso controllano applicazioni per apparecchiature.
- ~ Le linee di prodotti software sono costituite da una famiglia di applicazioni, di solito di proprietà della stessa organizzazione.
- ~ I framework applicativi di solito sono orientati al software non includono componenti che interagiscono con l'hardware

Architetture per linee di prodotti

- ~ L'architettura di una linea di prodotti software spesso rispecchia un generico stile o schema architetturale specifico di alcune applicazioni.
- ~ Per esempio una linea di prodotti progettata per gestire l'invio di veicoli per servizi di emergenza
 - gli operatori ricevono le telefonate di segnalazione degli incidenti
 - ~ trovano il veicolo appropriato al tipo di incidente
 - ~ inviano il veicolo sul luogo dell'incidente
 - gli sviluppatori di tale sistema potrebbero creare versioni di questo sistema per polizia vigili del fuoco e ospedali

Architetture per linee di prodotti

- ~ Il sistema di invio dei veicoli di emergenza è un esempio di generica architettura di gestione e allocazione di risorse
- ~ Questi sistemi usano un database di risorse disponibili
- ~ Includono i componenti per implementare le politiche di allocazione delle risorse che sono state decise dalla società che usa il sistema
- ~ Gli utenti interagiscono con il sistema di gestione delle risorse per richiedere e ottenere le risorse e per porre domande sulle risorse e la loro disponibilità

Architetture per linee di prodotti

- ~ Le architetture devono essere strutturate in modo tale da separare i diversi sottosistemi e permetterne la modifica.
- ~ L'architettura dovrebbe anche separare le entità e le loro descrizioni e i livelli superiori del sistema accedono alle entità attraverso le descrizioni piuttosto che direttamente.

Architettura per un sistema di resource allocation

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

Architettura per un veicolo di spedizioni

Interaction

Operator interface

Comms system
interface

I/O management

Operator
authentication

Map and route
planner

Report
generator

Query
manager

Resource management

Vehicle status
manager

Incident
logger

Vehicle
despatcher

Equipment
manager

Vehicle
locator

Database management

Equipment
database

Transaction management

Incident log

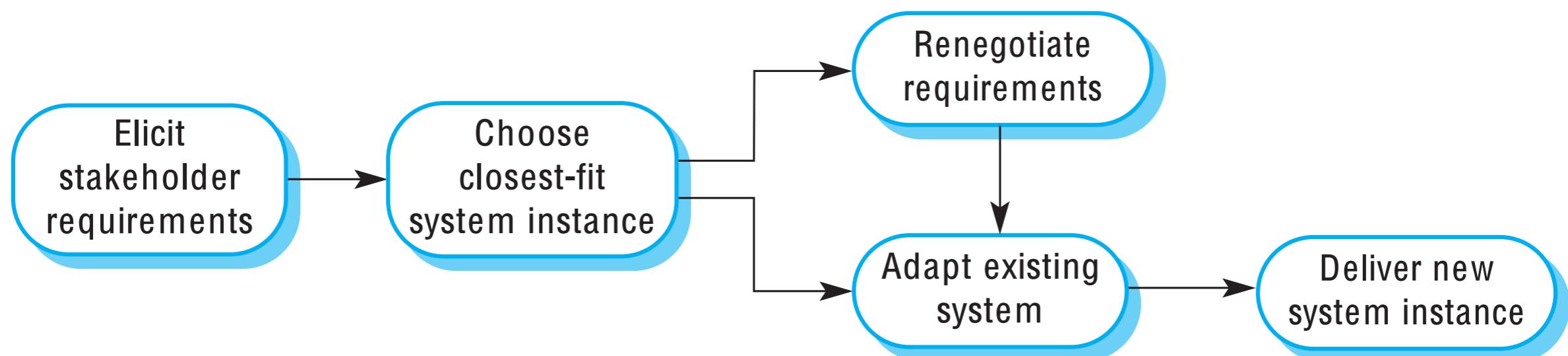
Vehicle database

Map database

Sviluppo di una istanza del prodotto

- ~ Per creare una nuova istanza di ogni linea di prodotti si devono modificare i singoli componenti
- ~ Si possono sviluppare vari tipi di specializzazione per una linea di prodotti software:
 - ~ Specializzazione della piattaforma
 - Diverse versioni dell'applicazione sono sviluppate per diverse piattaforme.
 - ~ Specializzazione dell'ambiente
 - Diverse versioni dell'applicazione sono create per gestire diversi ambienti operativi, ad esempio diversi tipi di apparecchiature di comunicazione.
 - ~ Specializzazione funzionale
 - Vengono create diverse versioni dell'applicazione per clienti con diversi requisiti.
 - ~ Specializzazione di processo
 - Vengono create diverse versioni dell'applicazione per supportare diversi processi aziendali.

Sviluppo di un'istanza del prodotto



Sviluppo di un'istanza del prodotto

- ~ Acquisire i requisiti delle parti interessate
 - & usare un membro della famiglia esistente come prototipo
- ~ Scegliere il membro della famiglia più adatto
 - & Trovare il membro della famiglia che meglio soddisfa i requisiti
- ~ Rinegoziare i requisiti
 - & Adattare i requisiti come necessario alle capacità del software
- ~ Adattare il sistema esistente
 - & Sviluppare nuovi moduli e apportare modifiche al membro della famiglia
- ~ Consegnare il nuovo membro della famiglia
 - & Documentare le caratteristiche chiave per un ulteriore sviluppo del membro

Sviluppo di un'istanza del prodotto

- ~ Quando si crea un nuovo membro di una linea di prodotti occorre trovare un compromesso tra il maggior riutilizzo possibile dell'applicazione generica e il soddisfacimento delle richieste specifiche degli stakeholder
- ~ Più i requisiti sono dettagliati, più è difficile che i componenti esistenti li soddisfino
- ~ Se gli stakeholder sono flessibili e richiedono poche modifiche da apportare al sistema si può consegnare il sistema più velocemente
- ~ Le linee di prodotti sono progettate per essere riconfigurabili
 - Aggiunta o rimozione di componenti dal sistema
 - definizione dei parametri e dei vincoli per i componenti del sistema
 - inclusione di conoscenze dei processi aziendali

Configurazione di una product line

- ~ Due fasi diverse per la configurazione:
 - s. Configurazione a design time
 - ~ L'organizzazione che sta sviluppando il software modifica un nucleo comune della linea di prodotti sviluppando, selezionando o adattando componenti per creare un nuovo sistema per un cliente.
 - s. Configurazione a deployment time
 - ~ Un sistema generico è progettato per essere configurato da un cliente o da consulenti che lavorano con il cliente. La conoscenza dei requisiti specifici del cliente e dell'ambiente operativo del sistema è incorporata nei dati di configurazione che sono usati dal sistema generico.



Riutilizzo di sistemi applicativi

Riutilizzo di sistemi applicativi

- ~ Un sistema applicativo è un sistema software che può essere adattato per diversi clienti senza cambiare il codice sorgente del sistema.
- ~ I sistemi applicativi hanno **caratteristiche generiche** e quindi possono essere usati/riutilizzati in diversi ambienti (mercato generale e non singoli clienti)
 - sono detti COTS - Commercial Off-The-Shelf System
- ~ I sistemi applicativi vengono adattati usando **meccanismi di configurazione** interni che permettono di adattare la funzionalità del sistema alle esigenze specifiche del cliente.
 - Per esempio, in un sistema di registrazione dei pazienti in un ospedale, possono essere definiti moduli di input e report di output separati per diversi tipi di pazienti.
 - si possono anche produrre plug-in per incrementare/customizzare le funzionalità

Riutilizzo di sistemi applicativi - vantaggi

- ~ Come con altri tipi di riutilizzo sw, può essere possibile una consegna più rapida di un sistema affidabile.
- ~ È possibile vedere quali funzionalità sono fornite dalle applicazioni e quindi è più facile giudicare se esse sono adatte o meno.
- ~ Alcuni rischi di sviluppo sono evitati usando il software esistente. Tuttavia, questo approccio ha i suoi rischi, come si dirà sotto.
- ~ Le aziende possono concentrarsi sulla loro attività principale senza dover dedicare molte risorse allo sviluppo di sistemi IT.
- ~ Con l'evoluzione delle piattaforme operative, gli aggiornamenti tecnologici possono essere semplificati perché sono responsabilità del fornitore del prodotto COTS piuttosto che del cliente.

Riutilizzo di sistemi applicativi - problemi

- ~ I requisiti di solito devono essere adattati per riflettere la funzionalità e il modo di operare del prodotto COTS.
- ~ Il prodotto COTS può essere basato su presupposti che sono praticamente impossibili da cambiare.
- ~ Scegliere il giusto sistema COTS per un'impresa può essere un processo difficile, specialmente perché molti prodotti COTS non sono ben documentati.
- ~ Ci può essere una mancanza di esperienza locale per supportare lo sviluppo dei sistemi.
- ~ Il fornitore del prodotto COTS controlla il supporto e l'evoluzione del sistema. Potrebbe uscire dal mercato o apportare modifiche che causano problemi ai clienti.

Sistemi applicativi configurabili

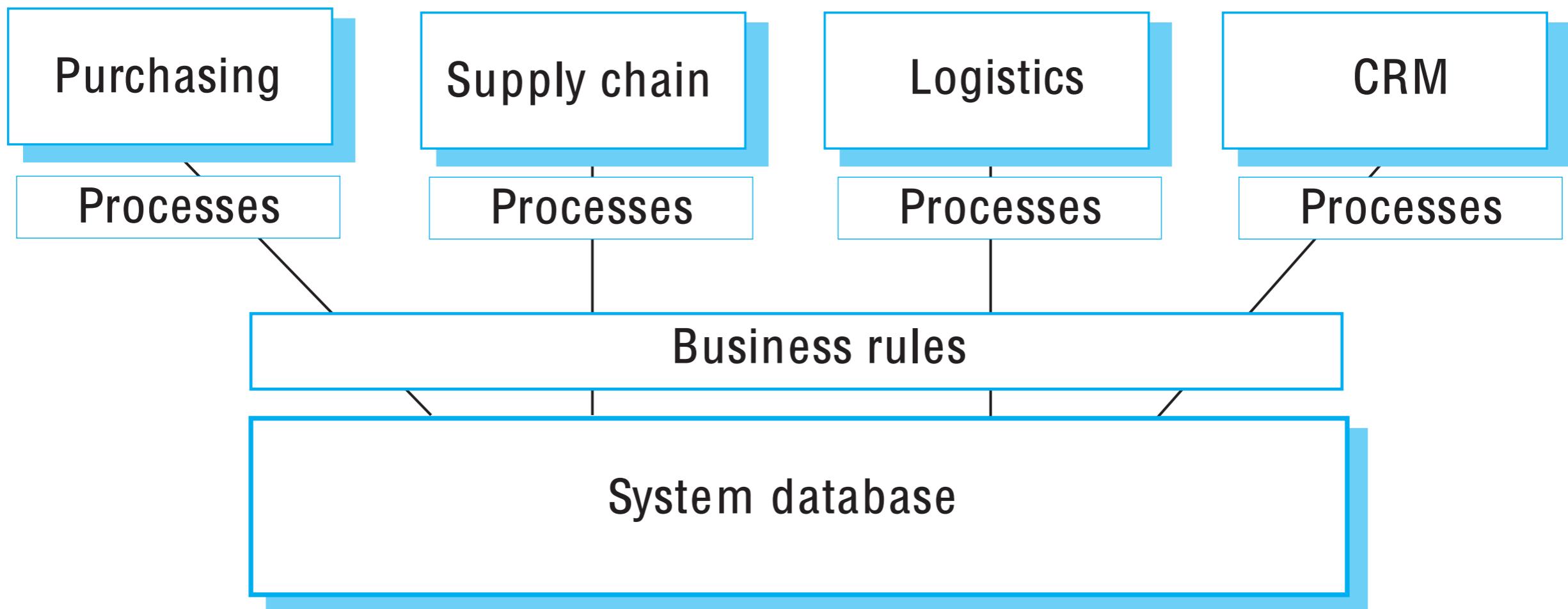
- ~ I sistemi applicativi configurabili sono sistemi applicativi generici che possono essere progettati per supportare un particolare tipo di business, un'attività commerciale o, a volte, un'impresa commerciale completa.
- ~ Per esempio, un sistema applicativo prodotto per i dentisti che può gestire gli appuntamenti, promemoria, records, il remind dei pazienti, ecc.
- ~ I sistemi specifici del dominio, come i sistemi per supportare una funzione aziendale (per esempio la gestione dei documenti) forniscono funzionalità che probabilmente saranno richieste da una serie di potenziali utenti.



Sistemi ERP

- ~ Un sistema ERP (Enterprise Resource Planning) è un sistema generico che supporta processi di business comuni come ordini e fatturazione, produzione, ecc.
- ~ Questi sono molto usati nelle grandi aziende - rappresentano probabilmente la forma più comune di riutilizzo del software.
- ~ Il nucleo generico viene adattato includendo moduli e incorporando la conoscenza dei processi e delle regole di business.

Architettura di un sistema ERP



Architettura di un sistema ERP

- ~ un certo numero di moduli per supportare diverse funzioni aziendali.
- ~ un insieme definito di processi di business, associati ad ogni modulo, che si riferiscono alle attività in quel modulo.
- ~ un database comune che mantiene le informazioni su tutte le funzioni di business correlate.
- ~ un insieme di regole di business che si applicano a tutti i dati nel database.

Configurazione di un ERP

- ~ I sistemi ERP richiedono una precisa configurazione:
 - & Selezionare le funzionalità richieste dal sistema.
 - & Stabilire un modello di dati che definisce come i dati dell'organizzazione saranno strutturati nel database del sistema.
 - & Definire le regole di business che si applicano a quei dati.
 - & Definire le interazioni previste con i sistemi esterni.
 - & Progettare i moduli di input e i rapporti di output generati dal sistema.
 - & Progettare nuovi processi di business che siano conformi al modello di processo sottostante supportato dal sistema.
 - & Impostare i parametri che definiscono come il sistema è distribuito sulla sua piattaforma sottostante.
- ~ Fase finale: test del nuovo sistema

Sistemi applicativi integrati

- ~ I sistemi applicativi integrati sono applicazioni che includono due o più prodotti del sistema applicativo e/o sistemi applicativi legacy.
- ~ Si può usare questo approccio quando non c'è un singolo sistema applicativo che soddisfi tutte le tue esigenze o quando vuoi integrare un nuovo sistema applicativo con i sistemi che già usi.

Sistemi integrati - scelte di progettazione

- ~ Quali sistemi applicativi individuali offrono la funzionalità più appropriata?
 - Tipicamente, ci saranno diversi prodotti di sistemi applicativi disponibili, che possono essere combinati in modi diversi.
- ~ Come verranno scambiati i dati?
 - Prodotti diversi normalmente usano strutture e formati di dati unici. Dovete scrivere adattatori che convertano da una rappresentazione all'altra.
- ~ Quali caratteristiche di un prodotto saranno effettivamente utilizzate?
 - I singoli sistemi applicativi possono includere più funzionalità di quelle necessarie e le funzionalità possono essere duplicate in prodotti diversi.

Key Points

- ~ Esistono molti modi diversi per riutilizzare il software. Si va dal riutilizzo di classi e metodi nelle librerie al riutilizzo di sistemi applicativi completi.
- ~ I vantaggi del riutilizzo del software sono: costi inferiori, sviluppo più rapido del software e minori rischi. L'affidabilità del sistema aumenta. Gli specialisti possono essere utilizzati in modo più efficace concentrando le loro competenze sulla progettazione di componenti riutilizzabili.
- ~ I framework applicativi sono collezioni di oggetti concreti e astratti progettati per essere riutilizzati attraverso la specializzazione e l'aggiunta di nuovi oggetti. Di solito incorporano buone pratiche di progettazione attraverso i design pattern.



Key points

- ~ Le linee di prodotti software sono applicazioni correlate sviluppate a partire da una o più applicazioni di base. Un sistema generico viene adattato e specializzato per soddisfare requisiti specifici di funzionalità, piattaforma di destinazione o configurazione operativa.
- ~ Il riutilizzo dei sistemi applicativi riguarda il riutilizzo di sistemi su larga scala, non disponibili sul mercato. Questi forniscono molte funzionalità e il loro riutilizzo può ridurre radicalmente i costi e i tempi di sviluppo. I sistemi possono essere sviluppati configurando un singolo sistema applicativo generico o integrando due o più sistemi applicativi.
- ~ I problemi potenziali del riutilizzo dei sistemi applicativi includono la mancanza di controllo sulle funzionalità e sulle prestazioni, la mancanza di controllo sull'evoluzione del sistema, la necessità di supporto da parte di fornitori esterni e le difficoltà nel garantire l'interoperabilità dei sistemi.