Kapitola 15 – Opätovné použitie softvéru

Opätovné použitie softvéru

- Vo väčšine inžinierskych disciplín sú systémy navrhnuté skladaním existujúcich komponentov, ktoré boli použité v iných systémoch.
- Softvérové inžinierstvo sa viac zameriavalo na pôvodný vývoj, ale teraz sa uznáva, že na dosiahnutie lepšieho softvéru, rýchlejšie a pri nižších nákladoch, potrebujeme proces návrhu, ktorý je založený na systematickom opätovnom používaní softvéru .
- Za posledných 15 rokov došlo k veľkému prechodu na vývoj založený na opätovnom použití.

Softvérové inžinierstvo založené na opätovnom použití

- Opätovné použitie systému
  - Kompletné systémy, ktoré môžu obsahovať niekoľko aplikačných programov, môžu byť opätovne použité.
- Opätovné použitie aplikácie
  - Aplikáciu možno opätovne použiť buď jej začlenením bez zmeny do inej , alebo vývojom rodín aplikácií.
- Opätovné použitie komponentov
  - Komponenty aplikácie od podsystémov až po jednotlivé objekty možno opätovne použiť.
- Opätovné použitie objektu a funkcie
  - Softvérové komponenty malého rozsahu, ktoré implementujú jeden dobre definovaný objekt alebo funkciu, možno opätovne použiť.

Výhody opätovného použitia softvéru

Výhoda	Vysvetlenie
Zrýchlený vývoj	Uvedenie systému na trh čo najskôr je často dôležitejšie ako celkové náklady na vývoj. Opätovné použitie softvéru môže urýchliť produkciu systému, pretože sa môže skrátiť čas vývoja aj overovania .
Efektívne využitie špecialistov	Namiesto toho, aby robili tú istú prácu znova a znova, môžu aplikační špecialisti vyvíjať opätovne použiteľný softvér, ktorý zahŕňa ich znalosti.
Zvýšená spoľahlivosť	Opätovne použitý softvér, ktorý bol vyskúšaný a testovaný vo fungujúcich systémoch, by mal byť spoľahlivejší ako nový softvér. Mali by sa nájsť a opraviť chyby v jeho návrhu a implementácii.
Nižšie náklady na vývoj	Náklady na vývoj sú úmerné veľkosti vyvíjaného softvéru. Opätovné použitie softvéru znamená, že je potrebné napísať menej riadkov kódu.

Výhody opätovného použitia softvéru

Výhoda	Vysvetlenie
Znížené riziko procesu	Náklady na existujúci softvér sú už známe, zatiaľ čo náklady na vývoj sú vždy vecou posúdenia. Toto je dôležitý faktor pre riadenie projektu, pretože znižuje chybovosť v odhade nákladov na projekt. To platí najmä vtedy, keď sa opätovne používajú relatívne veľké softvérové komponenty, ako sú subsystémy.
Dodržiavanie noriem	Niektoré štandardy, ako napríklad štandardy používateľského rozhrania, môžu byť implementované ako súbor opakovane použiteľných komponentov. Napríklad, ak sú ponuky v používateľskom rozhraní implementované pomocou opakovane použiteľných komponentov, všetky aplikácie prezentujú používateľom rovnaké formáty ponúk. Používanie štandardných používateľských rozhraní zvyšuje spoľahlivosť, pretože používatelia robia menej chýb, keď im je k dispozícii známe rozhranie.

Problémy s opätovným použitím

Problém	Vysvetlenie
Vytváranie, údržba a používanie knižnice komponentov	Naplnenie opakovane použiteľnej knižnice komponentov a zabezpečenie toho, aby vývojári softvéru mohli používať túto knižnicu, môže byť drahé. Vývojové procesy sa musia prispôsobiť tak, aby sa zabezpečilo používanie knižnice.
Nájdenie, pochopenie a prispôsobenie opakovane použiteľných komponentov	Softvérové komponenty musia byť objavené v knižnici, pochopené a niekedy prispôsobené na prácu v novom prostredí. Inžinieri si musia byť dostatočne istí, že nájdu komponent v knižnici predtým, ako zahrnú vyhľadávanie komponentov ako súčasť svojho bežného vývojového procesu.
Zvýšené náklady na údržbu	Ak zdrojový kód opätovne použitého softvérového systému alebo komponentu nie je dostupný, náklady na údržbu môžu byť vyššie, pretože opätovne použité prvky systému sa môžu stať čoraz nekompatibilnejšie so systémovými zmenami.

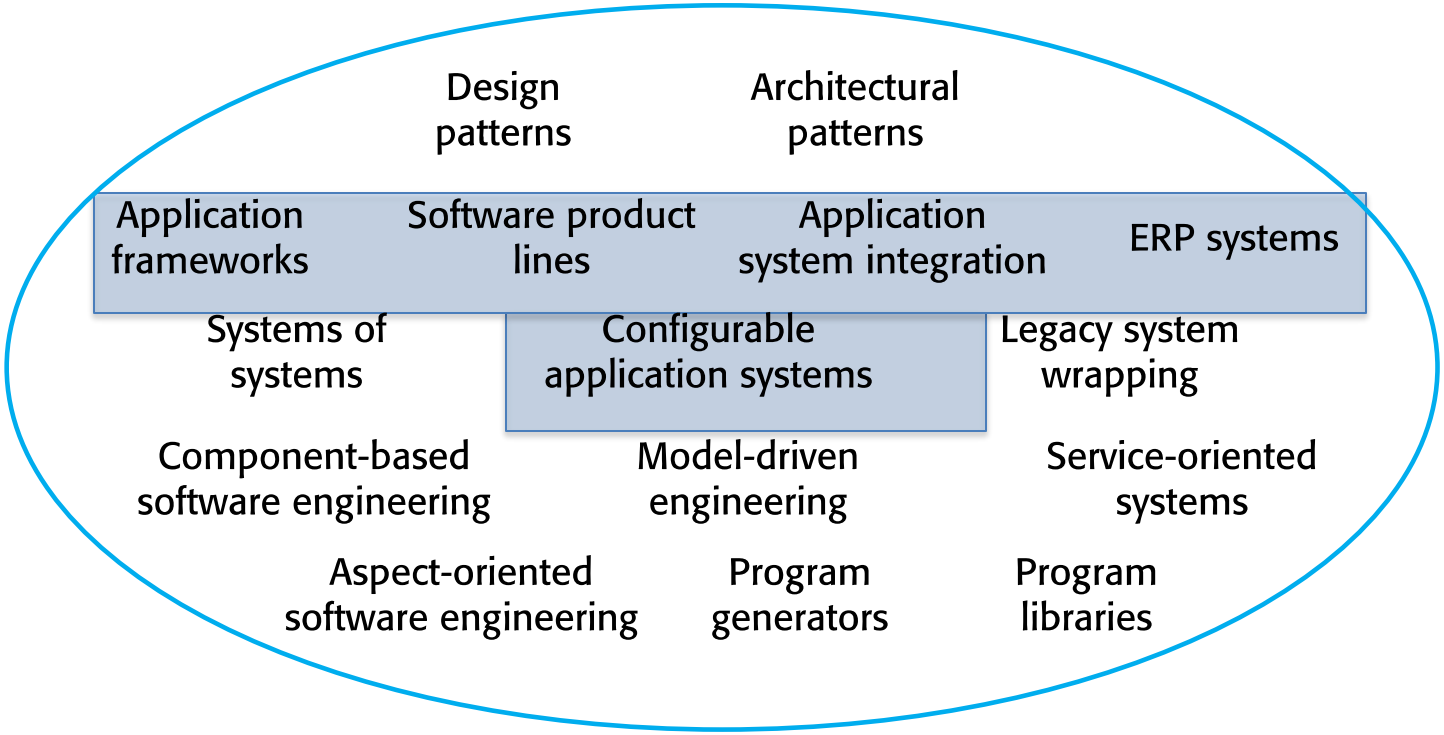
Problémy s opätovným použitím

Problém	Vysvetlenie
Nedostatok podpory a nástrojov	Niektoré softvérové nástroje nepodporujú vývoj s opätovným použitím. Môže byť ťažké alebo nemožné integrovať tieto nástroje do systému knižnice komponentov. Softvérový proces predpokladaný týmito nástrojmi nemusí brať do úvahy opätovné použitie. To platí najmä pre nástroje, ktoré podporujú inžinierstvo vstavaných systémov, menej už pre objektovo orientované vývojové nástroje.
Syndróm "tu nevynájdený"	Niektorí softvéroví inžinieri uprednostňujú prepisovanie komponentov, pretože veria, že ich môžu zlepšiť. Čiastočne to súvisí s dôverou a čiastočne so skutočnosťou, že písanie originálneho softvéru sa považuje za náročnejšie ako opätovné použitie softvéru iných ľudí.

Oblasti opätovného použitia

- Hoci opätovné použitie sa často jednoducho chápe ako opätovné použitie systémových komponentov, existuje mnoho rôznych prístupov k opätovnému použitiu, ktoré možno použiť.
- Opätovné použitie je možné na rôznych úrovniach od jednoduchých funkcií až po kompletné aplikačné systémy.
- Oblasť opätovného použitia pokrýva celý rad možných techník opätovného použitia.

Oblasti opätovného použitia



Oblasti, ktoré podporujú opätovné použitie SW

Oblasť	Popis
Aplikačné rámce	Kolekcie abstraktných a konkrétnych tried sú prispôsobené a rozšírené na vytváranie aplikačných systémov.
Integrácia aplikačného systému	Dva alebo viac aplikačných systémov sú integrované, aby poskytovali rozšírenú funkčnosť
Architektonické vzory	Ako základ aplikácií sa používajú štandardné softvérové architektúry, ktoré podporujú bežné typy aplikačných systémov .
Aspektovo orientovaný vývoj softvéru	Zdieľané komponenty sú pri kompilácii programu votkané do aplikácie na rôznych miestach. Popísané vo webovej kapitole 31.
Softvérové inžinierstvo založené na komponentoch	Systémy sa vyvíjajú integráciou komponentov (zbierok objektov), ktoré zodpovedajú štandardom modelu komponentov .

Oblasti, ktoré podporujú opätovné použitie SW

Oblasť	Popis
Konfigurovateľné aplikačné systémy	Systémy špecifické pre jednotlivé domény sú navrhnuté tak, aby ich bolo možné konfigurovať podľa potrieb konkrétnych zákazníkov systému.
Dizajnové vzory	Všeobecné abstrakcie, ktoré sa vyskytujú naprieč aplikáciami, sú reprezentované ako návrhové vzory zobrazujúce abstraktné a konkrétne objekty a interakcie .
ERP systémy	Pre organizáciu sú nakonfigurované rozsiahle systémy, ktoré zahŕňajú všeobecnú obchodnú funkčnosť a pravidlá.
Balenie starého systému	Staršie systémy sú „zabalené“ definovaním súboru rozhraní a poskytovaním prístupu k týmto starým systémom prostredníctvom týchto rozhraní.
Modelom riadené inžinierstvo	Softvér je reprezentovaný ako doménové modely a modely nezávislé na implementácii a z týchto modelov je generovaný kód .

Oblasti, ktoré podporujú opätovné použitie SW

Oblasť	Popis
Generátory programov	Systém generátora vkladá znalosti o type aplikácie a používa sa na generovanie systémov v tejto doméne z modelu systému dodaného používateľom.
Programové knižnice	Na opätovné použitie sú k dispozícii knižnice tried a funkcií, ktoré implementujú bežne používané abstrakcie.
Servisne orientované systémy	Systémy sú vyvíjané prepojením zdieľaných služieb, ktoré môžu byť poskytované externe .
Softvérové produktové rady	Typ aplikácie je zovšeobecnený okolo spoločnej architektúry, aby sa dal prispôsobiť rôznym zákazníkom.
Systémy systémov	Dva alebo viac distribuovaných systémov sú integrované, aby vytvorili nový systém .

Plánovacie faktory pre opätovne použite

- Plán vývoja softvéru.
- Očakávaná životnosť softvéru.
- Zázemie, zručnosti a skúsenosti vývojového tímu.
- Kritickosť softvéru a jeho nefunkčné požiadavky.
- Doména aplikácie.
- Platforma pre softvér.

Aplikačné rámce

Definícia rámca

- *Softvérový rámec (kostra) je...*

"... integrovaná sada softvérových artefaktov (ako sú triedy, objekty a komponenty), ktoré spolupracujú na poskytovaní opätovne použiteľnej architektúry pre rodinu súvisiacich aplikácií."

Aplikačné rámce

- Rámce sú stredne veľké entity, ktoré možno opätovne použiť. Sú niekde medzi opätovným použitím systému a komponentov.
- Rámce sú návrhom podsystému, ktorý pozostáva z kolekcie abstraktných a konkrétnych tried a rozhraní medzi nimi.
- Subsystém je implementovaný pridaním komponentov na vyplnenie častí dizajnu a vytvorením inštancií abstraktných tried v rámci .

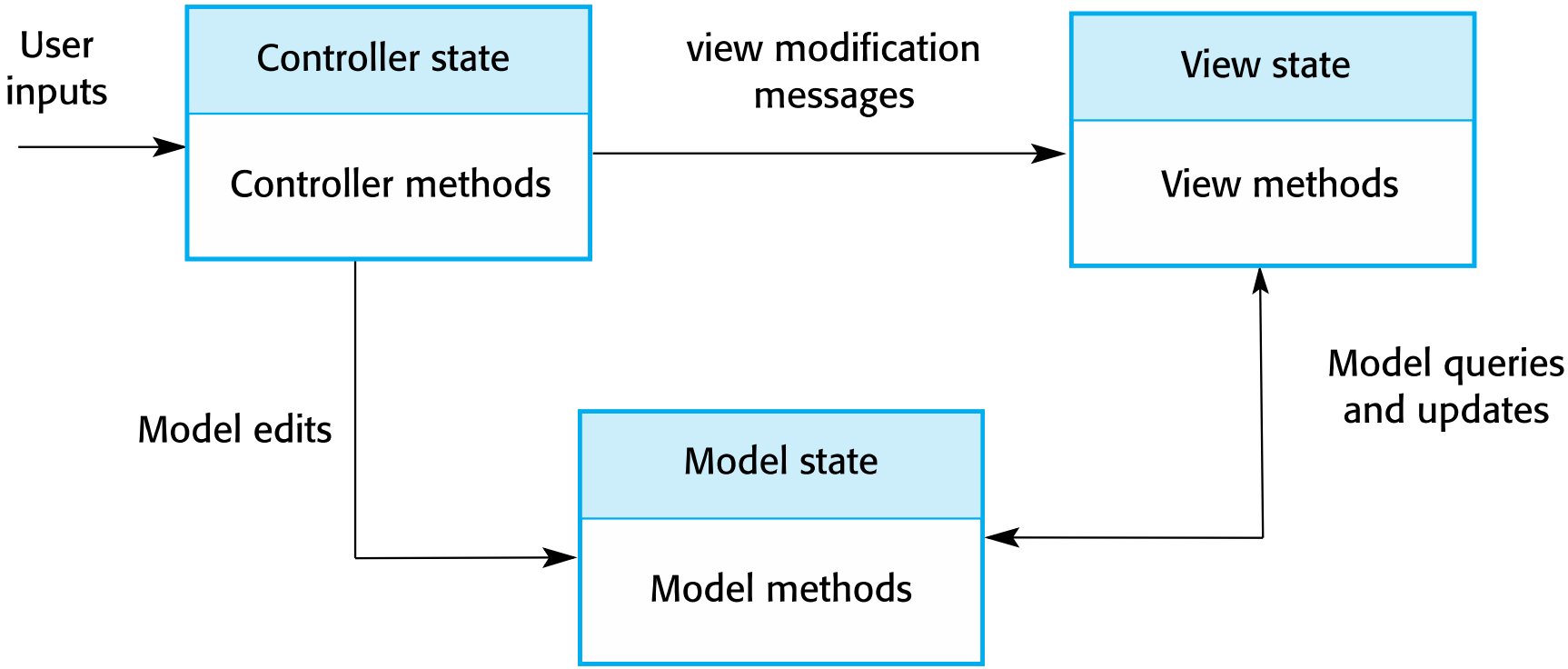
Rámce webových aplikácií (WAF)

- Podporte vytváranie dynamických webových stránok ako front-end pre webové aplikácie.
- WAF sú teraz dostupné pre všetky bežne používané webové programovacie jazyky, napr. Java, Python, Ruby atď.
- Interakčný model je založený na zloženom vzore Model-View-Controller.

Ovládač zobrazenia modelu

- Rámec systémovej infraštruktúry pre návrh GUI.
- Umožňuje viacero prezentácií objektu a samostatné interakcie s týmito prezentáciami.
- Rámec MVC zahŕňa vytvorenie množstva vzorov

Model -View-Controller



Funkcie WAF/Web Application Framework

- *Bezpečnosť*
  - WAF môžu obsahovať triedy, ktoré pomáhajú implementovať autentifikáciu používateľa (prihlásenie) a prístup.
- *Dynamické webové stránky*
  - Triedy vám pomôžu definovať šablóny webových stránok a dynamicky ich napĺňať zo systémovej databázy.
- *Podpora databázy*
  - Rámec môže poskytovať triedy, ktoré poskytujú abstraktné rozhranie pre rôzne databázy.
- *Správa relácií*
  - Triedy na vytváranie a správu relácií (množstvo interakcií používateľa so systémom) sú zvyčajne súčasťou WAF.
- *Interakcia používateľa*
  - Väčšina webových rámcov teraz poskytuje podporu AJAX (Holdener , 2008), ktorá umožňuje vytvárať interaktívnejšie webové stránky.

Rozšírenia rámca

- Rámce sú všeobecné a sú rozšírené tak, aby vytvorili špecifickejšiu aplikáciu alebo podsystém . Poskytujú kostru architektúry systému.
- Rozšírenie rámca zahŕňa
  - Pridanie konkrétnych tried , ktoré dedia operácie z abstraktných tried v rámci;
  - Pridanie metód , ktoré sa volajú v reakcii na udalosti, ktoré sú rozpoznané rámcom.
- Problémom rámcov je ich zložitosť, čo znamená, že ich efektívne používanie trvá dlho.

Triedy rámcov

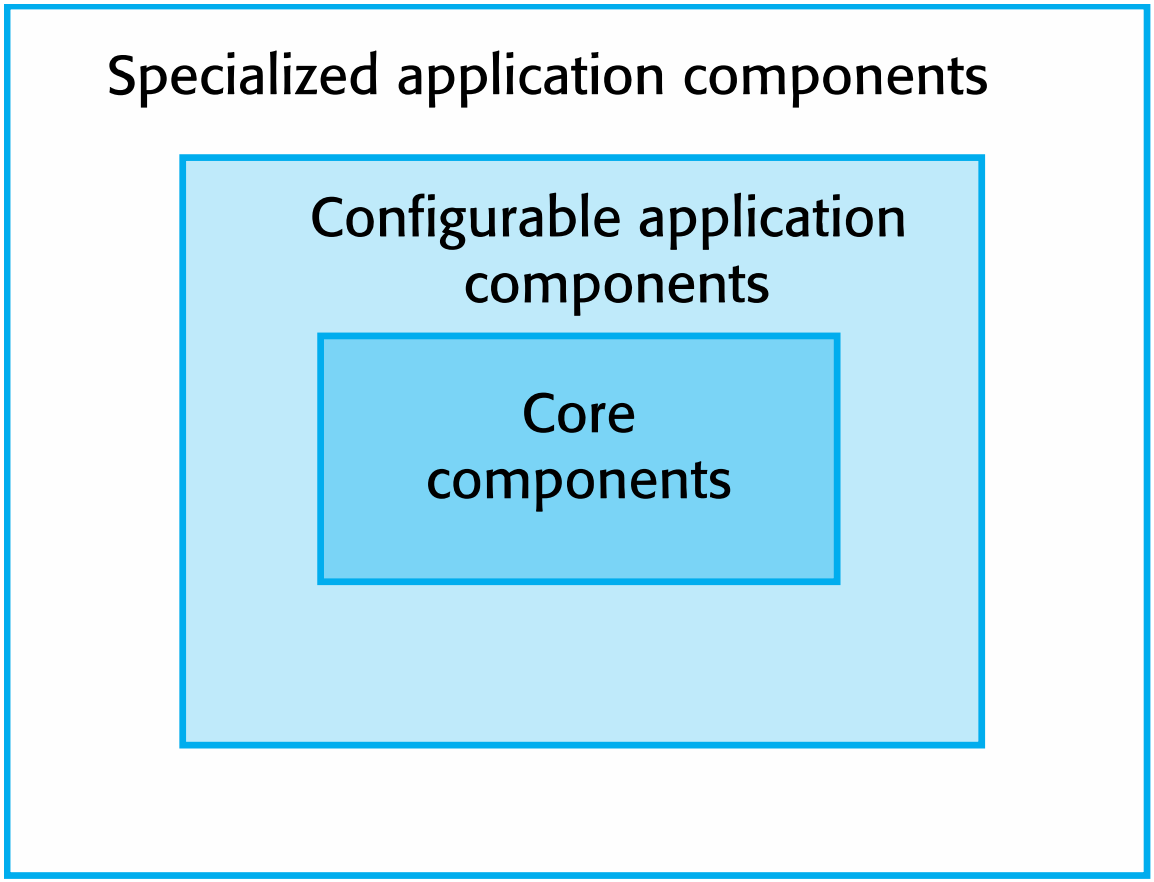
- Rámce systémovej infraštruktúry
  - Podporte vývoj systémových infraštruktúr, ako sú komunikácie, používateľské rozhrania a kompilátory.
- Integračné rámce middleware
  - Štandardy a triedy, ktoré podporujú komunikáciu komponentov a výmenu informácií.
- Podnikové aplikačné rámce
  - Podporte vývoj špecifických typov aplikácií, ako sú telekomunikačné alebo finančné systémy.

Softvérové produktové rady

Softvérové produktové rady

- Softvérové produktové rady alebo rodiny aplikácií sú aplikácie so všeobecnými funkciami, ktoré možno prispôsobiť a nakonfigurovať na použitie v špecifickom kontexte.
- Softvérový produktový rad je súbor aplikácií so spoločnou architektúrou a zdieľanými komponentmi, pričom každá aplikácia sa špecializuje na rôzne požiadavky.
- Adaptácia môže zahŕňať :
  - Konfigurácia komponentov a systému;
  - Pridávanie nových komponentov do systému;
  - Výber z knižnice existujúcich komponentov;
  - Úprava komponentov, aby vyhovovali novým požiadavkám.

Základné systémy pre rad softvérových produktov



Základné aplikácie

- Základné komponenty , ktoré poskytujú podporu infraštruktúry. Tieto sa zvyčajne pri vývoji novej inštancie produktového radu nemenia.
- Konfigurovateľné komponenty , ktoré možno upraviť a nakonfigurovať tak, aby sa špecializovali na novú aplikáciu. Niekedy je možné prekonfigurovať tieto komponenty bez zmeny ich kódu pomocou vstavaného konfiguračného jazyka komponentov.
- Špecializované komponenty špecifické pre doménu, z ktorých niektoré alebo všetky môžu byť nahradené, keď sa vytvorí nová inštancia produktového radu.

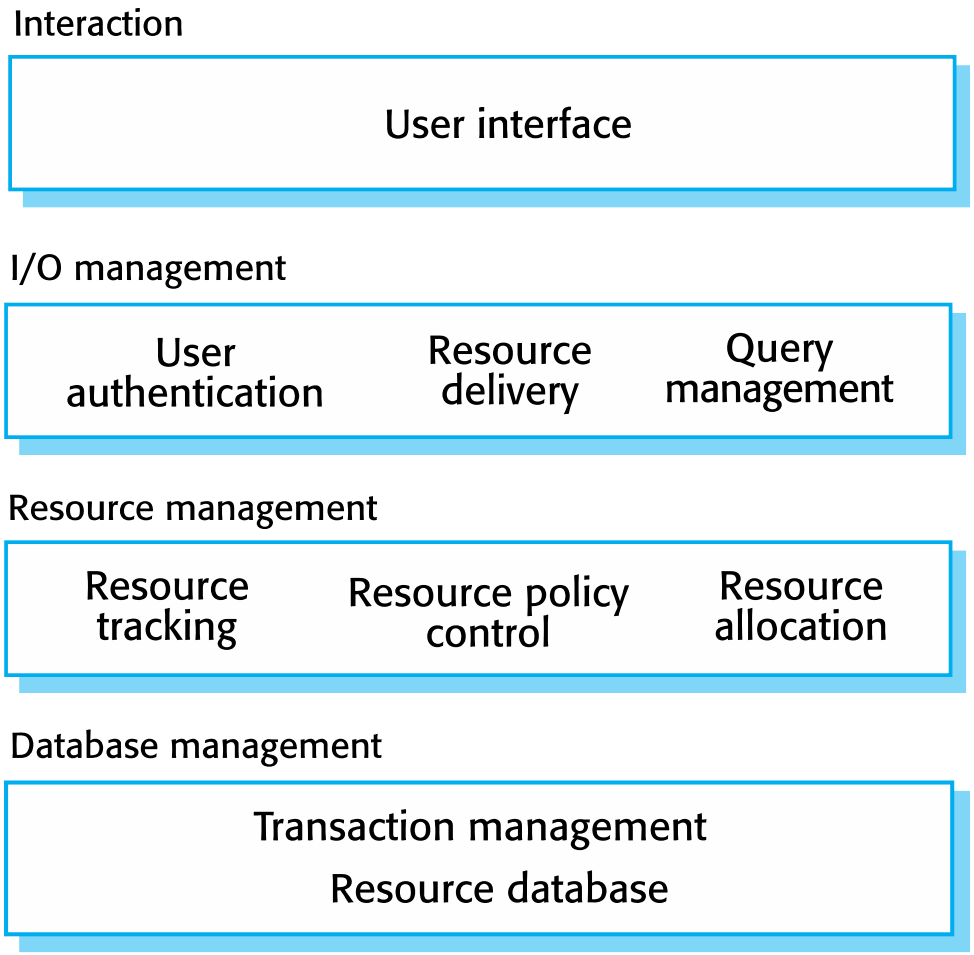
Aplikačné rámce a produktové rady

- Aplikačné rámce sa pri implementácii a rozšírení spoliehajú na objektovo orientované funkcie, ako je polymorfizmus. Produktové rady nemusia byť objektovo orientované (napr. vstavaný softvér pre mobilný telefón)
- Aplikačné rámce sa zameriavajú skôr na poskytovanie technickej podpory než na špecifickú doménu. Produktové rady obsahujú informácie o doméne a platforme.
- Produktové rady často riadia aplikácie zariadení.
- Softvérové produktové rady pozostávajú z rodiny aplikácií, ktoré zvyčajne vlastní tá istá organizácia.

Architektúry produktových radov

- Architektúry musia byť štruktúrované tak, aby oddeľovali rôzne podsystémy a umožňovali ich modifikáciu.
- Architektúra by tiež mala oddeľovať entity a ich popisy a vyššie úrovne v systémových entitách prístupu skôr prostredníctvom popisov ako priamo.

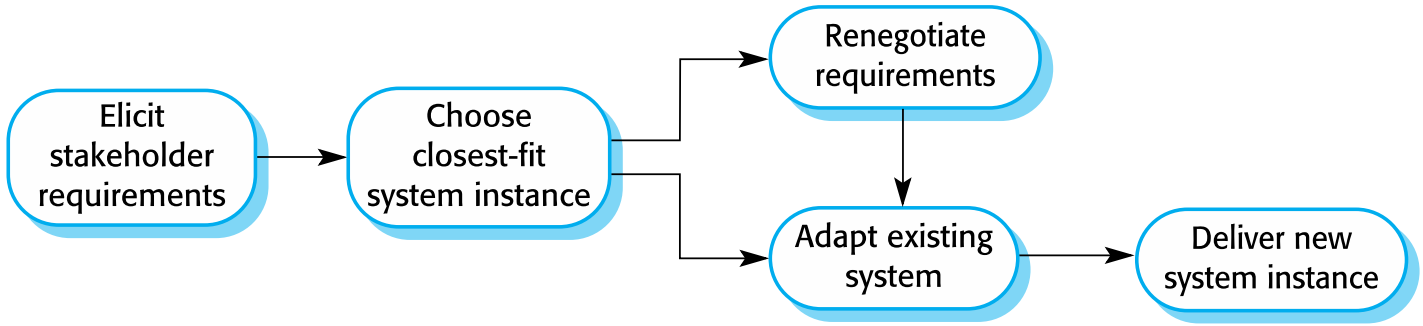
The architektúra systému prideľovania zdrojov



Špecializácia produktovej rady

- Platformizácia
  - Pre rôzne platformy sú vyvinuté rôzne verzie aplikácie.
- Špecializácia
  - Rôzne verzie aplikácie sú vytvorené tak, aby zvládali rôzne operačné prostredia, napr. rôzne typy komunikačných zariadení.
- Funkčná špecializácia
  - Pre zákazníkov s rôznymi požiadavkami sú vytvorené rôzne verzie aplikácie.
- Procesná špecializácia
  - Na podporu rôznych obchodných procesov sú vytvorené rôzne verzie aplikácie.

Vývoj inštancií produktu



Vývoj inštancií produktu

- Vyvolajte požiadavky zainteresovaných strán
  - Použite existujúceho člena rodiny ako prototyp
- Vyberte si najbližšieho člena rodiny
  - Nájdite člena rodiny, ktorý najlepšie spĺňa požiadavky
- Znova vyjednať požiadavky
  - Prispôsobte požiadavky podľa potreby schopnostiam softvéru
- Prispôsobte existujúci systém
  - Vyvíjajte nové moduly a vykonajte zmeny pre člena rodiny
- Dodajte nového člena rodiny
  - Zdokumentujte kľúčové vlastnosti pre ďalší rozvoj členov

Konfigurácia produktovej rady

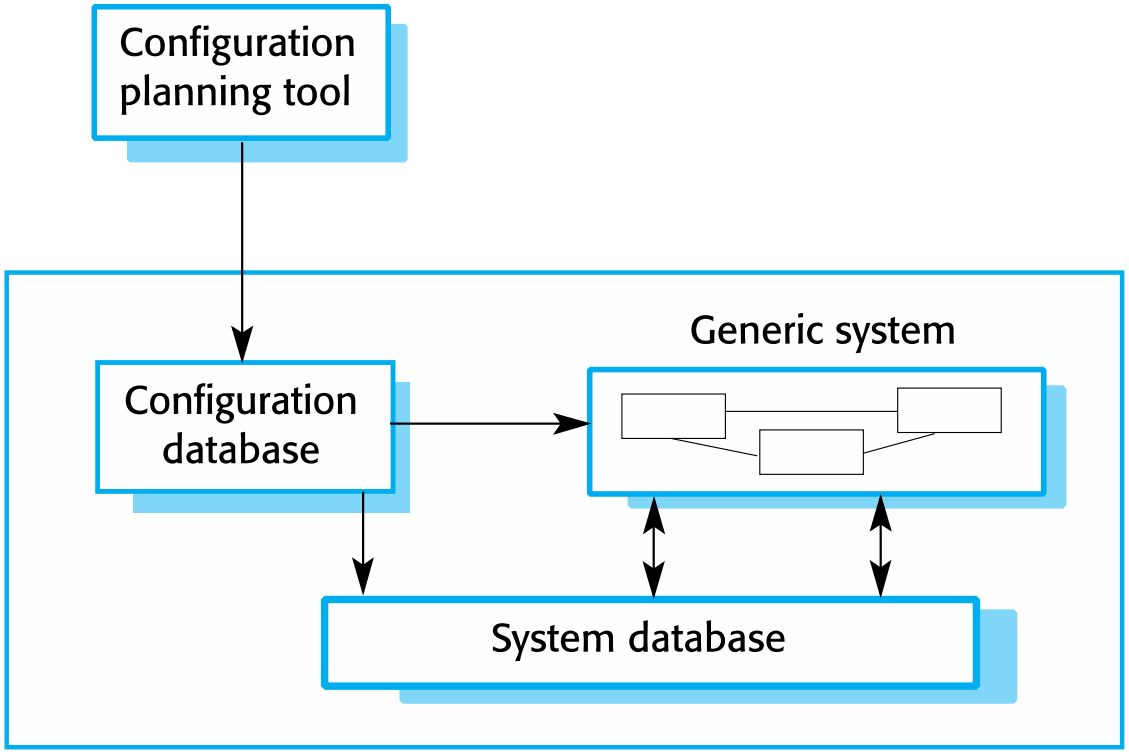
✧ Konfigurácia v čase vývoja

- Organizácia, ktorá vyvíja softvér, modifikuje spoločné jadro produktového radu vývojom, výberom alebo prispôsobením komponentov na vytvorenie nového systému pre zákazníka.

✧ Konfigurácia v čase nasadenia

- Všeobecný systém je navrhnutý na konfiguráciu zákazníkom alebo konzultantmi spolupracujúcimi so zákazníkom. Znalosť špecifických požiadaviek zákazníka a operačného prostredia systému je zakotvená v konfiguračných údajoch, ktoré používa generický systém.

Konfigurácia v čase nasadenia



Úrovne konfigurácie v čase nasadenia

- Výber komponentov , kde vyberáte moduly v systéme, ktoré poskytujú požadovanú funkcionalitu.
- Definícia pracovného toku a pravidiel , kde definujete pracovné toky (ako sa spracovávajú informácie, etapa po etape) a pravidlá validácie, ktoré by sa mali vzťahovať na informácie zadané užívateľmi alebo generované systémom.
- Definícia parametra , kde zadávate hodnoty špecifických systémových parametrov, ktoré odrážajú inštanciu aplikácie, ktorú vytvárate

Opätovné použitie aplikačného systému

Opätovné použitie aplikačného systému

- Produkt aplikačného systému je softvérový systém, ktorý je možné prispôsobiť rôznym zákazníkom bez zmeny zdrojového kódu systému.
- Aplikačné systémy majú všeobecné funkcie, a tak sa dajú použiť/znovu použiť v rôznych prostrediach.
- Produkty aplikačného systému sú prispôsobené pomocou vstavaných konfiguračných mechanizmov, ktoré umožňujú prispôsobiť funkčnosť systému špecifickým potrebám zákazníka.
  - Napríklad v nemocničnom systéme záznamov o pacientoch môžu byť pre rôzne typy pacientov definované samostatné vstupné formuláre a výstupné správy.

Výhody opätovného použitia aplikačného systému

- Rovnako ako pri iných typoch opätovného použitia môže byť možné rýchlejšie nasadenie spoľahlivého systému.
- Je možné vidieť, aké funkcie aplikácie poskytujú, a tak je ľahšie posúdiť, či sú vhodné alebo nie.
- Niektorým vývojovým rizikám sa dá vyhnúť používaním existujúceho softvéru. Tento prístup má však svoje riziká, o ktorých hovorím nižšie.
- Podniky sa môžu sústrediť na svoju hlavnú činnosť bez toho, aby museli venovať veľa zdrojov vývoju IT systémov.
- S vývojom operačných platforiem môžu byť aktualizácie technológií zjednodušené, pretože za ne zodpovedá skôr predajca produktov COTS než zákazník.

Problémy opätovného použitia aplikačného systému

- Požiadavky sa zvyčajne musia prispôsobiť tak, aby odrážali funkčnosť a spôsob prevádzky produktu COTS.
- Produkt COTS môže byť založený na predpokladoch, ktoré je prakticky nemožné zmeniť.
- Výber správneho systému COTS pre podnik môže byť náročný proces, najmä preto, že mnohé produkty COTS nie sú dobre zdokumentované.
- Na podporu rozvoja systémov môže byť nedostatok miestnych odborných znalostí.
- Predajca produktov COTS riadi podporu a vývoj systému.

Konfigurovateľné aplikačné systémy

- Konfigurovateľné aplikačné systémy sú generické aplikačné systémy , ktoré môžu byť navrhnuté tak, aby podporovali konkrétny typ podnikania, obchodnú činnosť alebo niekedy aj celý podnik.
  - Napríklad môže byť vytvorený aplikačný systém pre zubných lekárov, ktorý spracuje schôdzky, zubné záznamy, odvolanie pacientov atď.
- Systémy špecifické pre jednotlivé domény , ako napríklad systémy na podporu obchodnej funkcie (napr. správa dokumentov), poskytujú funkcie, ktoré bude pravdepodobne vyžadovať celý rad potenciálnych používateľov.

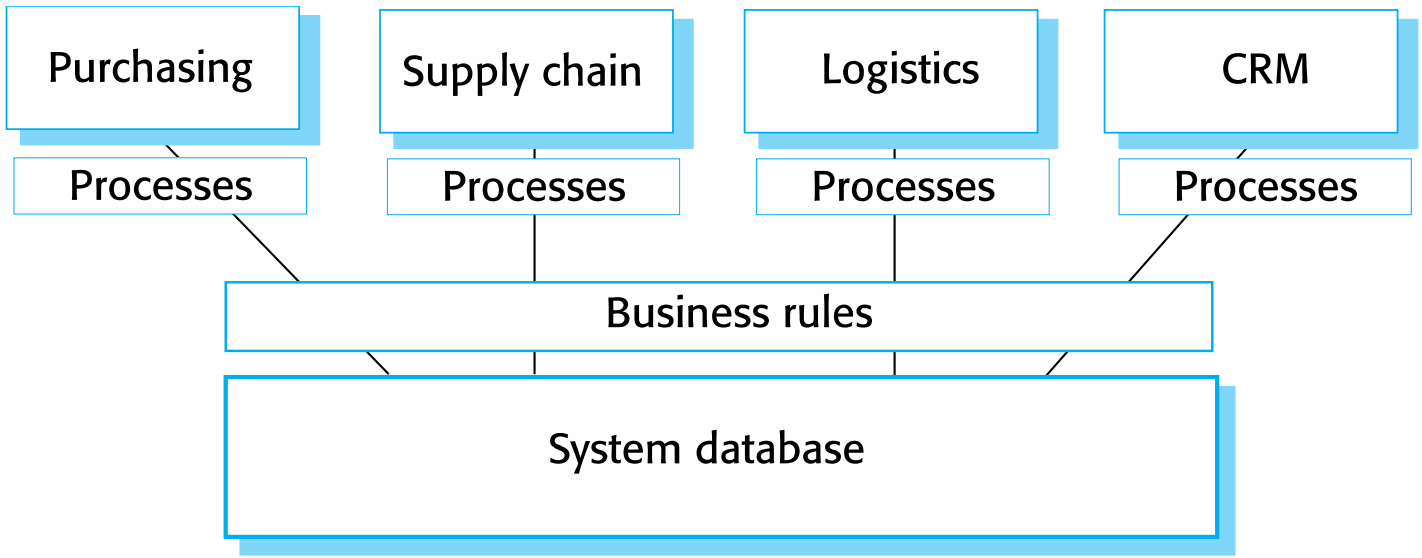
COTS - riešenie a COTS - integrované systémy

Konfigurovateľné aplikačné systémy	Integrácia aplikačného systému
Jediný produkt, ktorý poskytuje funkcie požadované zákazníkom	Je integrovaných niekoľko heterogénnych systémových produktov, ktoré poskytujú prispôsobenú funkčnosť
Založené na všeobecnom riešení a štandardizovaných procesoch	Pre zákaznícke procesy môžu byť vyvinuté flexibilné riešenia
Vývoj sa zameriava na konfiguráciu systému	Vývoj je zameraný na systémovú integráciu
Predajca systému je zodpovedný za údržbu	Vlastník systému je zodpovedný za údržbu
Dodávateľ systému poskytuje platformu pre systém	Vlastník systému poskytuje platformu pre systém

ERP systémy

- Systém Enterprise Resource Planning (ERP) je všeobecný systém, ktorý podporuje bežné obchodné procesy, ako je objednávanie a fakturácia, výroba atď.
- Tie sú vo veľkých spoločnostiach veľmi využívané – predstavujú asi najbežnejšiu formu opätovného použitia softvéru.
- Všeobecné jadro je prispôsobené zahrnutím modulov a zahrnutím znalostí o obchodných procesoch a pravidlách.

Architektúra ERP systému



Architektúra ERP

- Množstvo modulov na podporu rôznych obchodných funkcií.
- Definovaná množina obchodných procesov spojených s každým modulom, ktoré súvisia s aktivitami v tomto module.
- Spoločná databáza, ktorá uchováva informácie o všetkých súvisiacich obchodných funkciách.
- Súbor obchodných pravidiel, ktoré sa vzťahujú na všetky údaje v databáze.

Konfigurácia ERP

- Výber požadovanej funkcionality zo systému.
- Vytvorenie dátového modelu, ktorý definuje, ako budú dáta organizácie štruktúrované v systémovej databáze.
- Definovanie obchodných pravidiel, ktoré sa vzťahujú na tieto údaje.
- Definovanie očakávaných interakcií s externými systémami.
- Navrhovanie vstupných formulárov a výstupných zostáv generovaných systémom.
- Navrhovanie nových obchodných procesov, ktoré sú v súlade so základným procesným modelom podporovaným systémom.
- Nastavenie parametrov, ktoré definujú, ako je systém nasadený na základnej platforme.

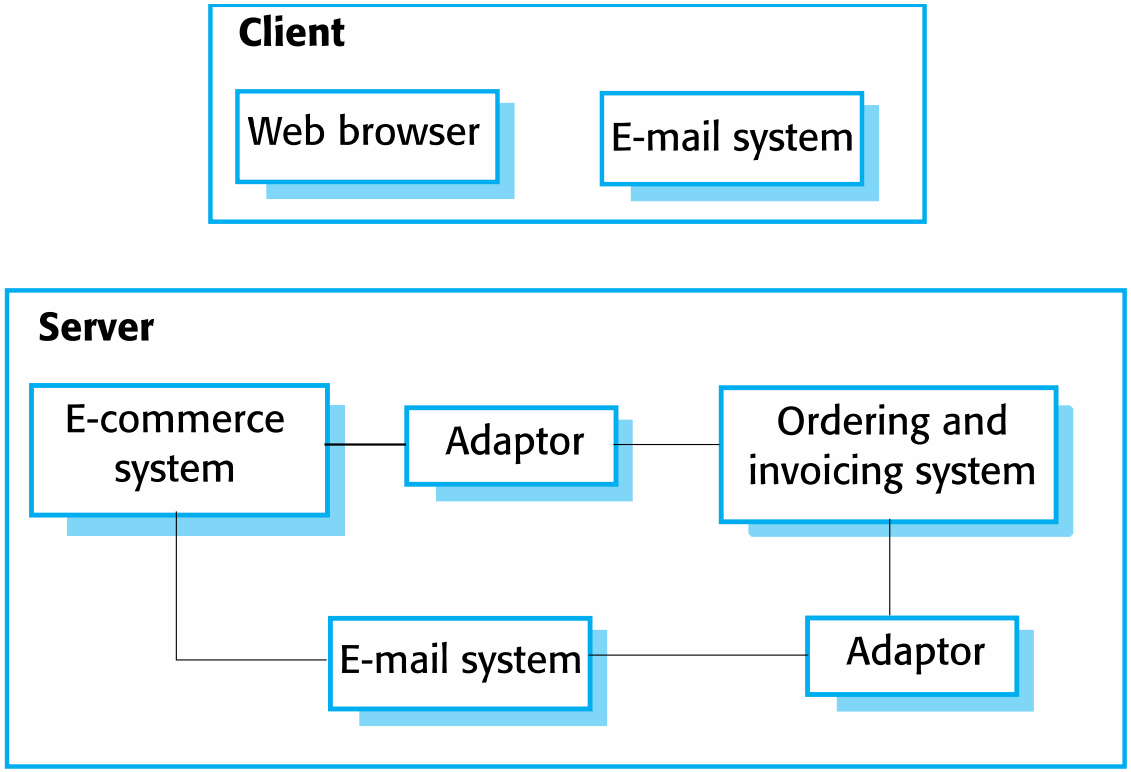
Integrované aplikačné systémy

- Integrované aplikačné systémy sú aplikácie, ktoré zahŕňajú dva alebo viac produktov aplikačného systému a/alebo staršie aplikačné systémy.
- Tento prístup môžete použiť, ak neexistuje jediný aplikačný systém, ktorý by vyhovoval všetkým vašim potrebám, alebo ak chcete integrovať nový aplikačný systém so systémami, ktoré už používate.

Voľby dizajnu

- Ktoré jednotlivé aplikačné systémy ponúkajú najvhodnejšiu funkčnosť?
  - Typicky bude k dispozícii niekoľko produktov aplikačného systému, ktoré možno rôznymi spôsobmi kombinovať.
- Ako sa budú vymieňať údaje?
  - Rôzne produkty bežne používajú jedinečné dátové štruktúry a formáty. Musíte napísať adaptéry, ktoré konvertujú z jednej reprezentácie na druhú.
- Aké vlastnosti produktu budú skutočne použité?
  - Jednotlivé aplikačné systémy môžu obsahovať viac funkcií, ako potrebujete, a funkcie môžu byť duplicitné v rôznych produktoch.

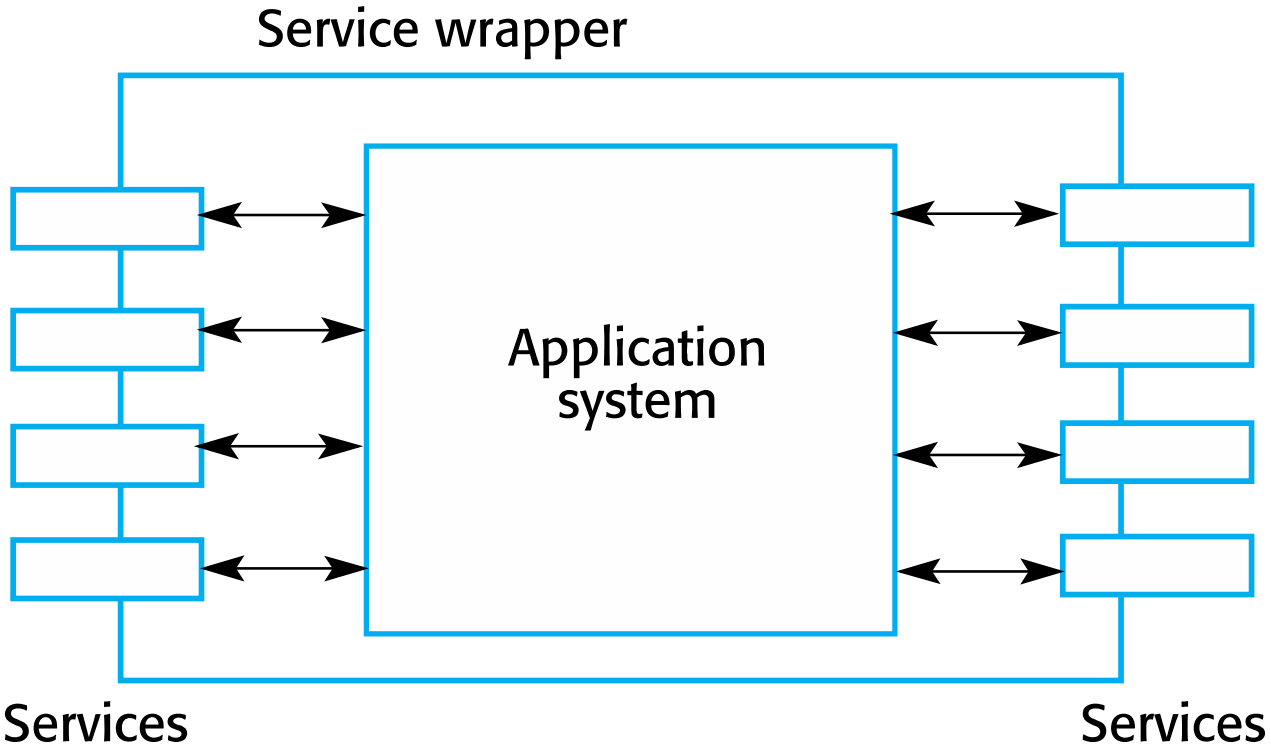
Integrovaný systém obstarávania



Servisne orientované rozhrania

- Integrácia aplikačného systému sa môže zjednodušiť, ak sa použije prístup orientovaný na služby.
- Servisne orientovaný prístup znamená umožnenie prístupu k funkcionalite aplikačného systému prostredníctvom štandardného servisného rozhrania so službou pre každú samostatnú jednotku funkcionality.
- Niektoré aplikácie môžu ponúkať servisné rozhranie, ale niekedy musí toto servisné rozhranie implementovať systémový integrátor. Musíte naprogramovať obal, ktorý skryje aplikáciu a poskytne externe viditeľné služby.

Balenie aplikácie



Problémy s integráciou aplikačného systému

- Nedostatok kontroly nad funkčnosťou a výkonom
  - Aplikačné systémy môžu byť menej účinné, ako sa zdá
- Problémy s interoperabilitou aplikačného systému
  - Rôzne aplikačné systémy môžu vytvárať rôzne predpoklady, čo znamená, že integrácia je náročná
- Žiadna kontrola nad vývojom systému
  - Vývojári kontrolujú predajcovia aplikačných systémov, nie používatelia systému
- Podpora od predajcov systému
  - Dodávatelia aplikačných systémov nemusia poskytovať podporu počas životnosti produktu