

Şabloane e-business

Șabloane

- Șablon
 - descrie o problemă cu repetări multiple în practică, într-un anumit context
 - descrie o soluționare ce a fost validată prin utilizări repetate în practică
- Reutilizare
 - clase și obiecte multiple, între care există relații și un comportament global ce pot fi reutilizate
- Șabloanele sunt folosite în implementare, dar și pentru activitățile de analiză și proiectare

Definiții

- Un **șablon orientat-obiect** reprezintă un model format dintr-un număr de clase ce lucrează împreună în rezolvarea unei probleme din domeniul tehnic sau economic.
- Un **șablon de proiectare** este un șablon orientat obiect ce descrie o soluție pentru rezolvarea unei probleme de proiectare.
- Un **șablon de analiză** este un șablon orientat obiect ce descrie o soluție pentru o problemă de business/analiză.

Fundamente teoretice (1)

- În dezvoltarea unei aplicații informatice se pot întâlni situații asemănătoare celor întâlnite anterior
 - indicată reutilizarea unor soluții deja aplicate și validate în practică
- De obicei există numeroase detalii care diferențiază problemele deja rezolvate de situațiile curente, asemănarea existând doar în punctele esențiale

Fundamente teoretice (2)

- Un șablon reprezintă abstractizarea unei probleme
 - este un model de soluționare pentru o problemă apărută într-un anumit context, verificat în practică
- Un șablon este descris prin:
 - **nume**
 - fiecare este identificat printr-un nume unic
 - **scop**
 - problema pe care trebuie să o rezolve un șablon
 - **soluționarea**
 - modul de soluționare propus de șablon pentru problemă
 - **participanți și colaboratori**
 - entitățile implicate în șablon
 - **consecințe și implementare**

Fundamente teoretice (3)

- Un șablon trebuie să se refere la o problemă întâlnită în mod repetat
 - Soluționarea trebuie să fie o rezolvare efectivă a problemei și nu numai formulări de principii sau considerații teoretice
 - Aplicarea unui șablon trebuie făcută numai în contextul pentru care a fost definit
-
- Șabloanele pot fi combinate în diverse moduri în vederea soluționării de probleme complexe

Motivele utilizării șabloanelor

- Reutilizarea soluțiilor
- Stabilirea unei terminologii comune
- O perspectivă de nivel înalt (de abstractizare) a problemei cât și a procesului de proiectare/analiză
- Prin reutilizarea soluțiilor se beneficiază și se învață din experiența altora
- Nu mai este necesară reinventarea de noi soluții pentru rezolvarea unor situații comune deja întâlnite în practică
- Comunicarea și lucrul în echipă necesită un vocabular și un punct de vedere comun asupra problemei de rezolvat, lucru permis de șabloanele de proiectare/analiză ce furnizează un punct de vedere comun în timpul etapelor de analiză și proiectare a unui soft

Avantaje ale șabloanelor (1)

- Îmbunătățirea comunicării în echipă și a studiului individual
 - Acest lucru se datorează membrilor tineri ai echipei, motivați de cei experimentați, să învețe lucruri noi și să aplice instrumentele puternice cunoscute de aceștia.
- Îmbunătățirea posibilității de modificare a codului program
 - Motivul este faptul că întotdeauna există factorul timp, hotărâtor în orice proces de dezvoltare de soluții software, iar aplicarea șabloanelor permite modificarea mai rapidă a produsului software în timp util, cu minim de efort.
- Șabloanele ilustrează principiile de bază ale abordării orientate-obiect
 - De obicei sunt utile în creșterea înțelegerii principiilor obiectuale de bază.

Avantaje ale șabloanelor (2)

- **Facilitarea procesului de proiectare și de analiză**
 - Oferă o alternativă față de ierarhiile laborioase de clase.
 - Aplicarea șabloanelor permite dezvoltarea de soluții complexe fără ierarhii vaste de clase.
- **Adoptarea de strategii îmbunătățite în proiectare și analiză**
 - Câteva strategii sugerate sunt: proiectarea interfețelor, practicarea compunerii față de ierarhia claselor, identificarea parametrilor ce variază în contextul problemei urmată de încapsularea acestora.
- **Îmbunătățirea productivității proceselor de dezvoltare software, creșterea consistenței între aplicații**
 - datorate reutilizării, cât și aplicării repetate.

Avantaje ale șabloanelor (3)

- Oferă mai mult decât o simplă reutilizare de cod sursă
 - datorat nivelului înalt de abstractizare al reutilizării, astfel putându-se aplica pe orice platformă sistem.
- Se pot combina în rezolvarea unor probleme mai dificile, complexe
- S-au bucurat de o largă acceptare, numărul lor a crescut exponențial
 - exprimă o realitate și furnizează suportul necesar reducerii efortului de dezvoltare de aplicații software mai mult decât orice carte, lucrare sau site Web de specialitate.

Dezavantaje ale șabloanelor

- **Necesită învățarea unui număr foarte mare de șabloane**
 - Implică multă muncă în cunoașterea acestora în vederea aplicării corecte.
- **Sindromul NIH** (not-invented-here)
 - Există neîncrederea în munca altora, de fapt pentru persoanele pentru care ceea ce nu este făcut personal nu reprezintă un lucru bun.
- **Nu includ cod sursă**, cu excepția șabloanelor de proiectare.
- **Șabloanele pot deveni un domeniu haotic/superficial**
 - Cu cât se va dovedi sau înțelege mai mult valoarea acestora tot mai multe persoane vor începe exploatarea lor în vederea creșterii vânzărilor și mai puțin în vederea utilității și aplicabilității.

Obiecte (1)

- În mod tradițional obiectele conțin date și metode
- Noua perspectivă a șabloanelor definește un obiect ca o entitate cu **responsabilități**
- Noua vedere a obiectelor:
 - permite focalizarea asupra a ceea ce pot face obiectele, nu numai asupra simplei lor modalități de implementare
 - permite o mai bună selecție și definire a obiectelor
 - definirea obiectelor este mai flexibilă
 - focalizarea pe ceea ce fac obiectele permite aplicarea moștenirii diferit din punct de vedere al comportamentului acestora
 - gândirea în termeni de responsabilități facilitează definirea interfeței publice a obiectelor
 - unui obiect cu responsabilități i se poate cere îmbunătățirea acestora, dar informația pentru care un obiect este responsabil nu este în interiorul acestuia, nu implică nimic din interiorul obiectului.

Obiecte (2)

- **Încapsularea** - modul de ascundere a datelor obiectelor
 - aceasta se poate aplica și pentru metode, subclase sau alte obiecte
- **Moștenirea** – exprimă relațiile de generalizare / specializare dintre clase
 - reutilizarea se obține, de obicei, prin crearea de clase, urmată de derivarea de noi clase din clasele de bază
- Din perspectiva șabloanelor moștenirea se aplică astfel: identificarea variațiilor și încapsulare lor
 - Primul pas constă în identificarea elementelor variabile în procesul de proiectare
 - metoda total opusă celei tradiționale în care se considera variabil ceea ce produce schimbare în proiectare față de ceea ce se poate schimba fără reproiectare
 - Urmează încapsularea acestora, pasul II
 - de fapt, multe șabloane folosesc încapsularea pentru a crea straturi de obiecte slab cuplate

Abordarea obiectuală din perspectiva șabloanelor

- Din perspectiva șabloanelor într-un proces de dezvoltare software se parcurg următorii pași:
 1. Identificarea șabloanelor corespunzătoare domeniului problemei de rezolvat.
 2. Pentru setul inițial de șabloane identificat se parcurg următorii pași de analiză:
 - a. se alege șablonul care furnizează, respectiv creează, cea mai mare parte din contextul celorlalte șabloane;
 - b. se aplică acest șablon la nivel conceptual global;
 - c. se identifică și alte șabloane suplimentare ce se vor adăuga la setul inițial de șabloane analizat;
 - d. se repetă analiza pentru noul set de șabloane identificat.
 3. Se adaugă detaliile necesare proiectării, se extind prin metode și clase.

Alegerea șablonului (2a)

- În alegerea șablonului se analizează câte o pereche de șabloane ce se pot aplica o dată
 - Întotdeauna un șablon existent într-un sistem stabilește relații cu celelalte existente, prin crearea unui context pentru acestea
- În etapa de analiză se începe cu identificarea relațiilor dintre șabloane, cât și a modul în care se stabilesc aceste relații
 - Acestea conduc la identificarea contextului în care se pot aplica șabloanele, dar și a celor ce creează context pentru alte șabloane.

Alegerea șablonului (2a)

- Un concept folosit în etapa de analiză este **șablonul seniormost** – acel șablon ce impune constrângeri asupra a ceea ce vor face celelalte șabloane.
 - Există un șablon care definește modul de comportament al altor șabloane?
 - Există două șabloane care se influențează reciproc?
 - Se poate defini un șablon fără altul, există un șablon ce are nevoie de un altul?

Metoda de realizare matriceală a sistemelor informatice din perspectiva șabloanelor

- Analysis Matrix - metodă de analiză a unei aplicații e-business
 - permite identificarea, coordonarea și regăsirea elementelor variabile în concepte obiectuale din perspectiva șabloanelor
- Pașii necesari acestei metode sunt:
 - Identificarea celor mai importante caracteristici, cerințe, pentru un caz de utilizare și organizarea acestora într-o matrice bidimensională.
 - Procesarea celorlalte cazuri de utilizare urmată de extinderea matricei, dacă este cazul. Fiecare caz se procesează independent de celelalte.
 - Extinderea metodei matriceale cu noi concepte.
 - Liniile matricei identifică regulile problemei.
 - Coloanele identifică cazurile speciale.
 - Identificarea șabloanelor.
 - Dezvoltarea unui model global pentru sistemul informatic.

Observații (1)

- Construind matricea bidimensională pentru celelalte cazuri din problemă (respectiv cazul 2, apoi cazul 3, ...) se identifică lipsurile și inconsistența în informații
- Beneficiarii aplicațiilor software ridică cel mai adesea probleme.
 - Ei își cunosc cel mai bine problema de rezolvat dar din păcate nu gândesc la nivel conceptual ci pe cazuri particulare, folosesc termenul de întotdeauna drept similar al termenului de obicei, sau niciodată față de rareori.
- Toate aceste lucruri duc la acele inconsistențe ce trebuie identificate încă din etapa de analiză.

Observații (2)

- Fiecare linie a matricei bidimensionale reprezintă un mod specific de implementare a conceptelor generale identificate din analiză.
- Fiecare coloană reprezintă un mod specific de implementare a cazurilor de utilizare identificate.
- Pentru fiecare linie sau coloană se identifică un șablon sau o clasă de obiecte, apoi se elaborează un model de proiectare a sistemului de realizat la nivel global, conceptual.

Observații (3)

- Aplicabilitatea acestei metode se justifică în special, în cazul problemelor complexe, cu multe cazuri specifice.
- Se analizează fiecare caz în parte, se identifică caracteristicile și comportamentul comun tuturor cazurilor, după care se construiește matricea de analiză care permite implementarea soluției software prin șabloane.
- Această metodă permite capturarea tuturor aspectelor particulare ale unei probleme.

Șabloane pentru aplicații de comerț electronic tip B2B

- Șabloanele de analiză (business pattern)
 - au un caracter conceptual și sunt utilizate pentru a crea modele adecvate și flexibile care să descrie comportamentul proceselor de afaceri studiate.
- Șabloanele de proiectare
 - propun structuri și relații care asigură rezolvarea, în contexte bine definite, a unor probleme de proiectare.

Șabloane de analiză

- **Șabloanele de analiză** descriu o soluție pentru probleme uzuale ce aparțin domeniului de afaceri/analiză a unei aplicații
 - se caracterizează prin mai multă specificitate decât cele de proiectare, descriind o parte din domeniul de afaceri studiat
- În această categorie se includ următoarele șabloane: Item-Item Description, Business Entity, Contact Point, Place și Shipping/Billing.

Exemple de șabloane de analiză

- **Item-Item Description**

- Este unul dintre cele mai uzuale și utile șabloane de analiză fiind descris drept colecția de obiecte ce împarte aceeași descriere dar cu instanțe diferite.

- **Business Entity**

- Uzual, orice organizație dezvoltă relații cu alți parteneri de afaceri, persoane fizice sau juridice. Un șablon tip Business Entity descrie relațiile dintre diferite tipuri de organizații și partenerii săi de afaceri.

Exemple de șabloane de analiză

- **Contact Point**

- Șablonul tip Contact Point descrie modul de interacțiune organizație-parteneri de afaceri.
- Subclasele acestui șablon trebuie să descrie cel puțin două lucruri: modul de transmitere de informații de la organizație la parteneri, cât și modul de catalogare al acestora (label info).

- **Shipping/Billing**

- În general, în orice aplicație de comerț electronic tip B2B este necesară livrarea către client a produselor/serviciilor sau informațiilor, cât și facturilor întocmite, atât fizic cât și electronic.
- Șablonul Shipping/Billing descrie tocmai modul de transmitere al acestora către client.

Exemple de șabloane de analiză

- Place

- Majoritatea organizațiilor implicate într-o afacere online au nevoie de urmărirea zonelor în care s-au dezvoltat procesele de afaceri.
- În cazul firmelor internaționale se impune și cunoașterea orelor, perioada vacanțelor, legile privind taxele, etc.
- Șablonul Place descrie modul de urmărire a tuturor acestor informații utile.

Șabloane de proiectare

- Șabloanele de proiectare descriu o soluție pentru problemele uzuale ce apar în etapa de proiectare a sistemelor informatice
- Cele mai întâlnite sunt Singleton, Proxy și State
- „Gang of Four” (GoF) - Design Patterns: Elements of Reusable Object-Oriented Software, Gamma, Helm, Johnson, and Vlissides.
- Ian Shalloway, James R. Trott, Pattern Oriented Design: Using Design Patterns From Analysis to Implementation, Addison-Wesley, 2000

Exemple de șabloane de proiectare

- Singleton

- Acest șablon este de tip creațional și asigură o singură instanțiere, în orice moment, pentru o clasă.
- Șablonul Singleton poate fi aplicat în cazul profilelor clienților unde este necesară menținerea informațiilor de bază ale acestora (nume, număr de telefon, etc.).

Exemple de șabloane de proiectare

- Proxy
 - Șablonul Proxy indică modul de reprezentare a unui obiect ce nu este momentan în memorie.
- Să presupunem un ecran de căutare de clienți în care utilizatorul poate defini un criteriu de căutare și obține lista clienților ce îndeplinesc respectivul criteriu.
 - Există vreo posibilitate de minimizare a volumului informațiilor trimise prin rețea?
 - Toate aceste se pot realiza printr-un șablon proxy. Aplicând acest șablon se reduce volumul informațiilor transmise prin rețea pentru că inițial se aduce doar câtă informație este necesară identificării fiecărui obiect client (nume), iar în momentul alegerii din sursă a unui client, obiectul ClientProxy aduce obiectul client din baza de date.

Exemple de șabloane de proiectare

- State
 - Șablonul State descrie modul de implementare al unui obiect ce-și poate schimba comportamentul la schimbarea stării lui interne.
 - Ideea este de a construi un obiect ca o colecție de mai multe obiecte stări, ce prezintă comportamente diferite cerute de obiect pentru fiecare stare în care poate fi.
 - Fiecare comportament ce este afectat de starea internă a obiectului este implementat în obiectele tip stare ca metodă.

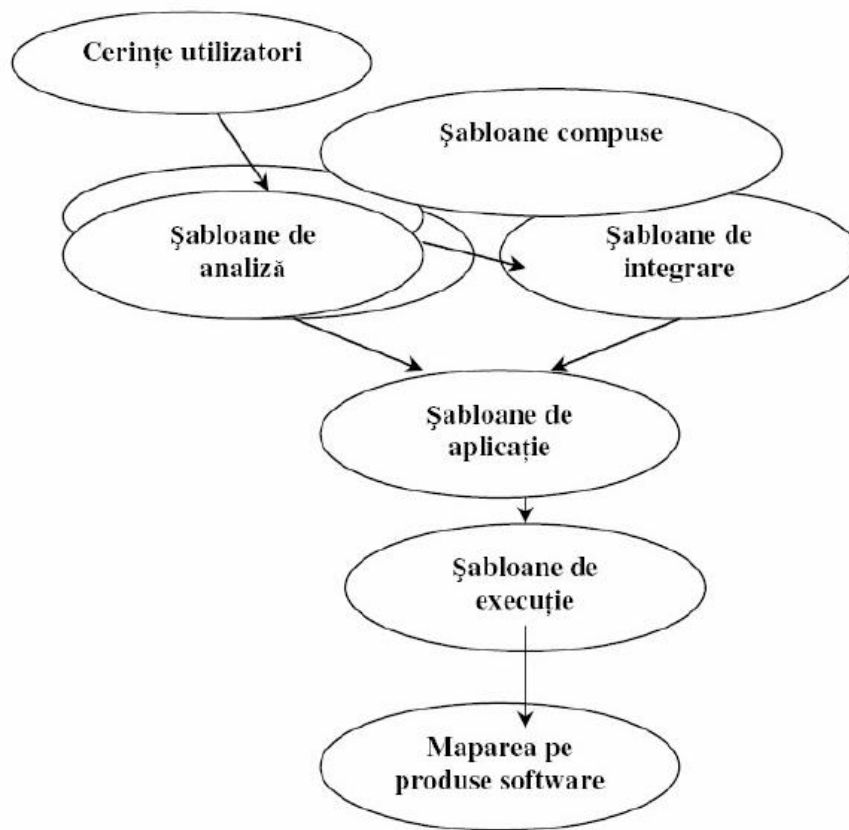
Aplicarea șabloanelor în soluții B2B

- Rolul dezvoltatorilor IT este de a evalua problema de afaceri și de a construi soluții software care să o rezolve
 - Necesită timp
 - Soluția: utilizarea și aplicarea experienței arhitecților IT din domeniu
- Pentru a putea captura experiența dobândită de alți experți IT trebuie creat un depozit de valori, de soluții deja dezvoltate în practică, pe baza cărora se pot dezvolta soluții noi
- Această tehnică a reutilizării implică câștig de timp, bani și efort iar într-un proces de dezvoltare soluții IT, facilitează obținerea unei soluții solide, funcționabile.

Șabloanele e-business IBM

- Aceste șabloane permit implementarea cu succes a soluțiilor B2B aplicând principiul reutilizării componentelor
- Abordarea din perspectiva șabloanelor se bazează pe un model structurat pe straturi de elemente ce pot fi exploatate și aplicate în orice metodologie de realizare de sisteme informatice
 - Fiecare strat este astfel structurat încât detaliile fiecăruia se construiesc pe baza stratului anterior.

Modelul structurat de șabloane pentru aplicații e-business



Șabloanele e-business IBM

- Modelul include:
 - **șabloane de analiză** (business patterns) - identifică interacțiunile dintre utilizatori, date și reguli de afaceri
 - **șabloane de integrare** (integration patterns) - realizează legătura dintre șabloanele de afaceri atunci când o soluție B2B nu poate fi descrisă printr-un singur șablon de afaceri, ajută la combinarea șabloanelor business pentru aplicații complexe
 - **șabloane compuse** (composite patterns) - reprezintă combinația dintre primele două straturi de șabloane
 - **șabloane de aplicație** (application patterns) - descrie, la un nivel conceptual, interacțiunile dintre aplicații (reguli de afaceri) și date din cadrul primelor două categorii de șabloane

Șabloanele e-business IBM

- Modelul include:
 - **șabloane de execuție** (runtime patterns) - descrie structura logică, de middleware, suportată de șabloanele tip aplicație, înfățișând nodurile majore, rolurile și interfețele dintre acestea
 - **implementări software** testate din punct de vedere practic (product mappings) - identifică implementările software testate pentru fiecare șablon tip runtime (execuție)
 - **linii directoare** (best-practice guidelines) ce ghidează proiectarea, dezvoltarea, cerințele non-funcționale, realizarea și managementul aplicațiilor dezvoltate.

Șabloane de analiză

- Descrie relațiile dintre utilizatori și organizație, regulile de afaceri cât și datele ce vor fi accesate.
- Există patru tipuri de șabloane de afaceri de bază:
 - **self-service** (user-to-business): utilizatorii interacționează direct cu organizația prin Internet sau Intranet
 - Ex: aplicații Web simple
 - **information aggregation** (user-to-data): utilizatorii pot extrage volume mari de informație
 - Ex: managementul cunoștințelor, inteligența afacerilor
 - **collaboration** (user-to-user): Internetul suportă colaborarea dintre utilizatori
 - Ex: email, chat, video conferințe
 - **extended enterprise** (business-to-business): aplicații ce fac legătura între multiplele procese de afaceri între organizații diferite
 - Ex: EDI, SCM.

Șabloane de integrare

- Permit combinarea mai multor șabloane de afaceri în vederea rezolvării unei probleme.
- Principalele șabloane de acest tip sunt:
 - **access integration**: permit integrarea unui număr de servicii printr-un punct comun de acces
 - Ex: portaluri
 - **application integration**: permite integrarea unui număr de aplicații și surse de date fără cererea directă a utilizatorilor, cum ar fi fluxuri manageriale, brokeri de mesaje.

Custom Design

- Șabloanele de afaceri și de integrare pot fi combinate pentru implementarea unei soluții e-business specifică, cum ar fi Custom Design



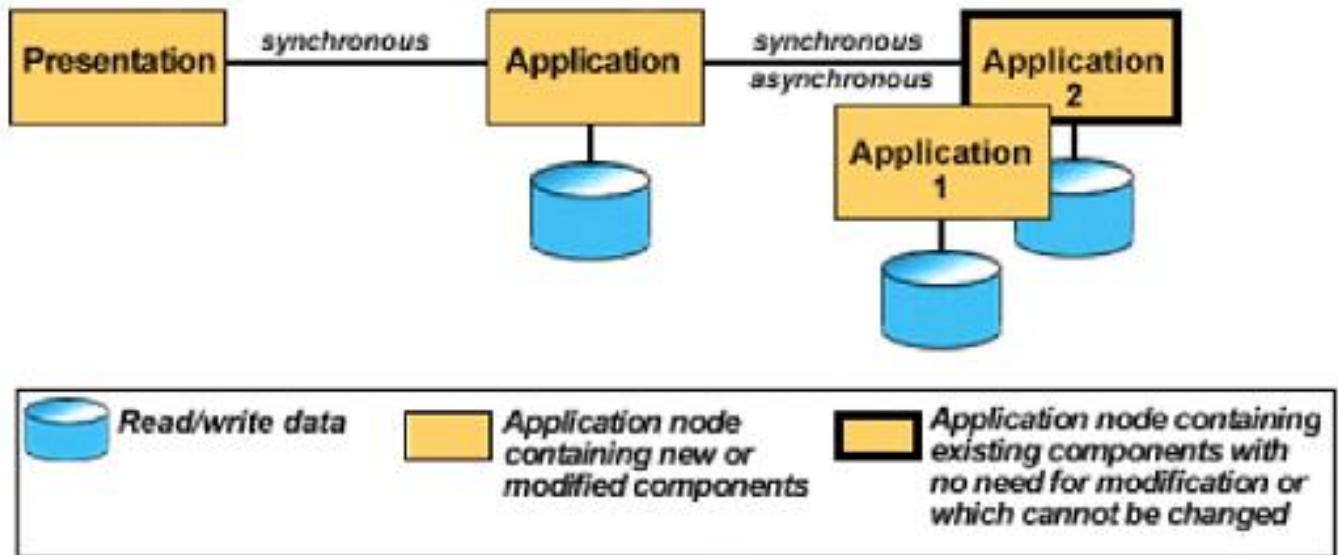
Șabloane de aplicație

- După identificarea șabloanelor business, se definesc componentele logice la nivel global și modul lor de interacțiune, ceea ce reprezintă un **șablon de aplicație**.
- De obicei un șablon de analiză are multiple șabloane de aplicație.
 - poate conține componentele logice ce descriu stratul de prezentare al aplicației (presentation tier), stratul regulilor de afaceri sau stratul de backend al aplicației.

Șabloane de aplicație

- Șablonul de aplicație împarte aplicația într-un număr de componente conceptuale de bază, identificând obiectivele acesteia.
- Ex. șablonul business – Self-Service
 - scopul este de a permite utilizatorilor accesul la informațiile din organizație de tip back-end.
 - în acest șablon componentele conceptuale sunt: aplicația de prezentare, aplicația ce descrie regulile de afaceri și aplicațiile existente în firmă (back-end) ce reprezintă stratul de persistență a datelor.

Self-Service::Directly Integrated Single Channel

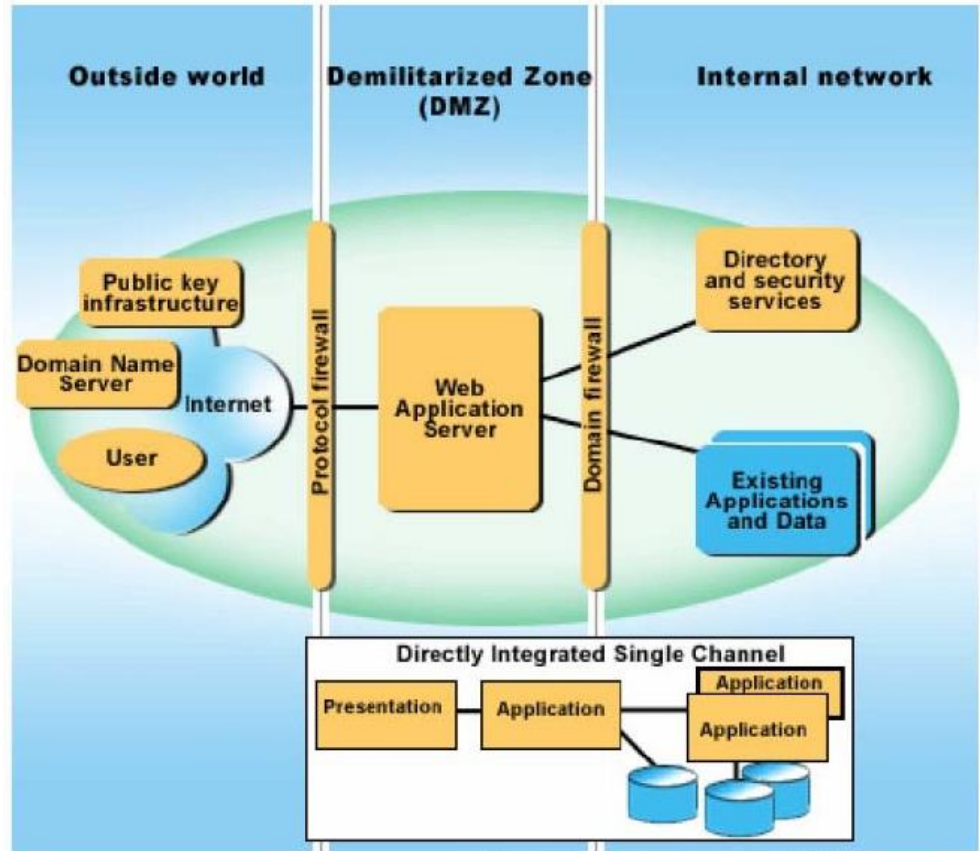


Șabloane de execuție

- În șabloanele de execuție (runtime), atenția se focalizează pe nodurile logice și pe locul de plasare în întreaga structură a rețelei
 - Șabloanele aplicație se dezvoltă prin adăugarea de noi funcții explicite, fiecare fiind asociată unui nod de execuție ce pot exista pe mașini diferite sau pe aceeași mașină
- Continuând exemplul anterior, se trece la identificarea șablonului de execuție potrivit pentru șablonul business și cel de aplicație ales

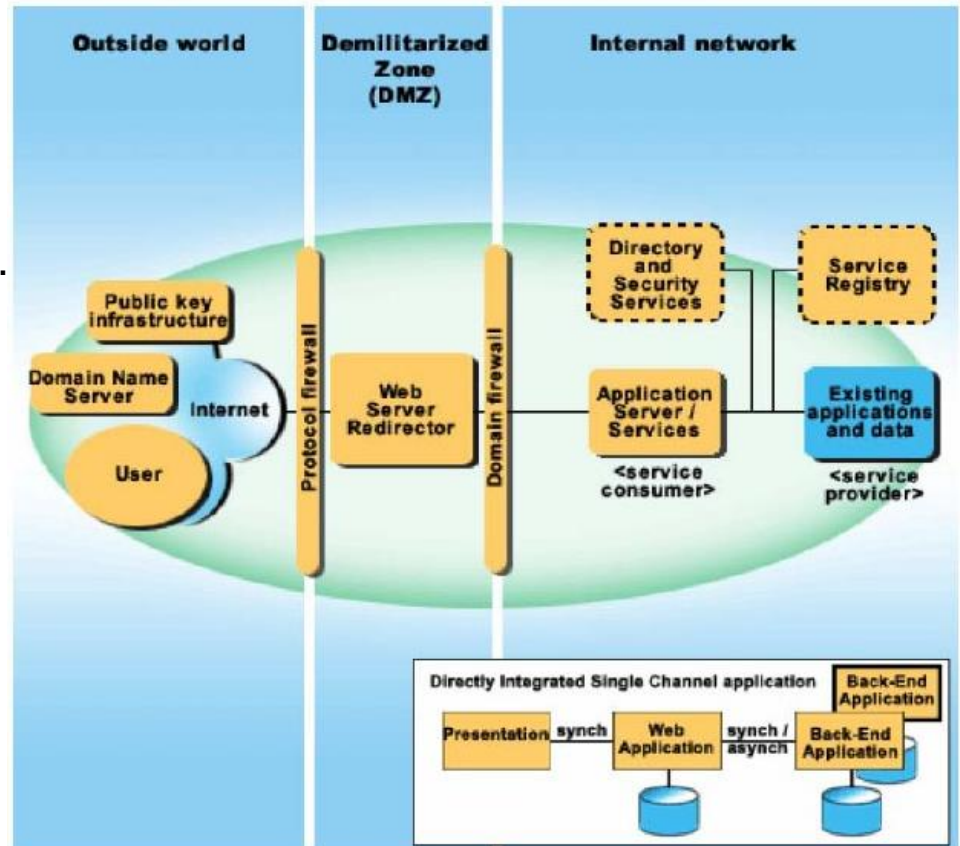
Șabloane de execuție

- Presupunând accesul utilizatorilor la datele organizației prin Internet, se impun și anumite măsuri de securitate.



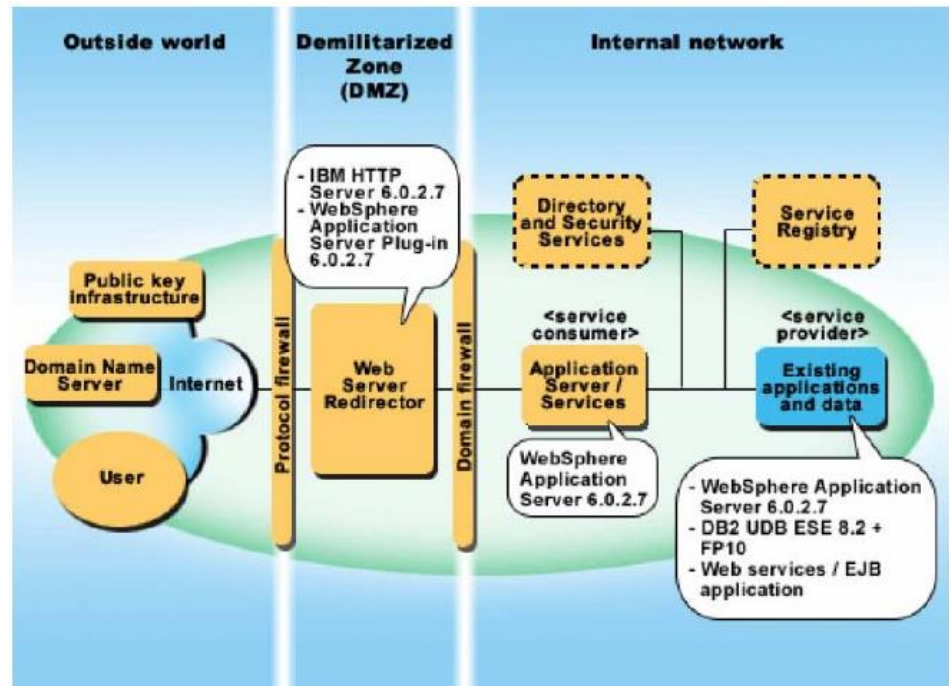
Șabloane de execuție

- O caracteristică a șablonului de execuție este plasarea serverului Web de aplicație între două firewall-uri.
- În cazul cererilor crescute de securitate există un șablon de execuție ce împarte serverul Web de aplicație în două noduri funcționale, separând funcțiile serverului HTTP de serverul de aplicație



Implementări software testate

- Ultimul pas în definirea structurii unei aplicații este corelarea produselor IT existente cu unul sau mai multe noduri logice de execuție.
- Fiecare mapare pe produse IT este orientată către o platformă software particulară funcție de preferințele beneficiarilor.



Sumar

- Șabloanele facilitează înțelegerea și analiza problemelor complexe, cât și divizarea lor în componente mai mici, mai ușor de administrat și de implementat.
- Prezentarea este doar o abordare succintă a tipurilor de șabloane ce pot fi aplicate în domeniul e-business.
- Teoria poate fi studiată mai pe larg pe site-ul IBM:
<http://ibm.com/developerworks/patterns/>