



UNIUNEA EUROPEANĂ



Instrumente Structurale  
2014-2020

# COMPETIT - Formare și calificare pentru competitivitatea întreprinderilor

COD SMIS: 128223

## **DESIGNER PAGINI WEB**



# Proiectarea aplicațiilor Web

## 1. Metodologii de realizare

- ▶ Modelarea proceselor economice reprezintă o condiție esențială în realizarea unor sisteme informatice complexe, care să realizeze cerințele funcționale dorite.
- ▶ Un model reprezintă un set de procese care cuprind diverse cerințe operaționale necesare în atingerea unor obiective ale procesului de afaceri. Astfel etapele de analiză și proiectare a arhitecturii unui sistem de comerț electronic sunt extrem de importante, deciziile corecte trebuie să fie luate încă din fazele inițiale ale proiectului.
- ▶ Totalitatea etapelor parcurse în procesul de dezvoltare a unei aplicații reprezintă ciclul de viață al acesteia.



Etapele acestui proces sunt (figura 1):

- ▶ **Culegerea specificațiilor** - definirea problemei, specificarea detaliată a funcționalităților pe care sistemul informatic trebuie să le îndeplinească;
- ▶ **Analiza cerințelor** - etapa în care se identifică caracteristicile esențiale care definesc o anumită soluție, pornind de la anumite cerințe;
- ▶ **Proiectarea** - adaugă elemente noi ale soluției construite în etapa de analiză, pe baza optimizării anumitor criterii;
- ▶ **Implementarea** - etapa în care se realizează un proiect executabil pornind de la soluția construită în faza de analiză;
- ▶ **Testarea** - etapa de verificare a echivalenței sistemului informatic implementat cu modelul creat în etapa de analiză [Avornicului2004].



UNIUNEA EUROPEANĂ



Instrumente Structurale  
2014-2020

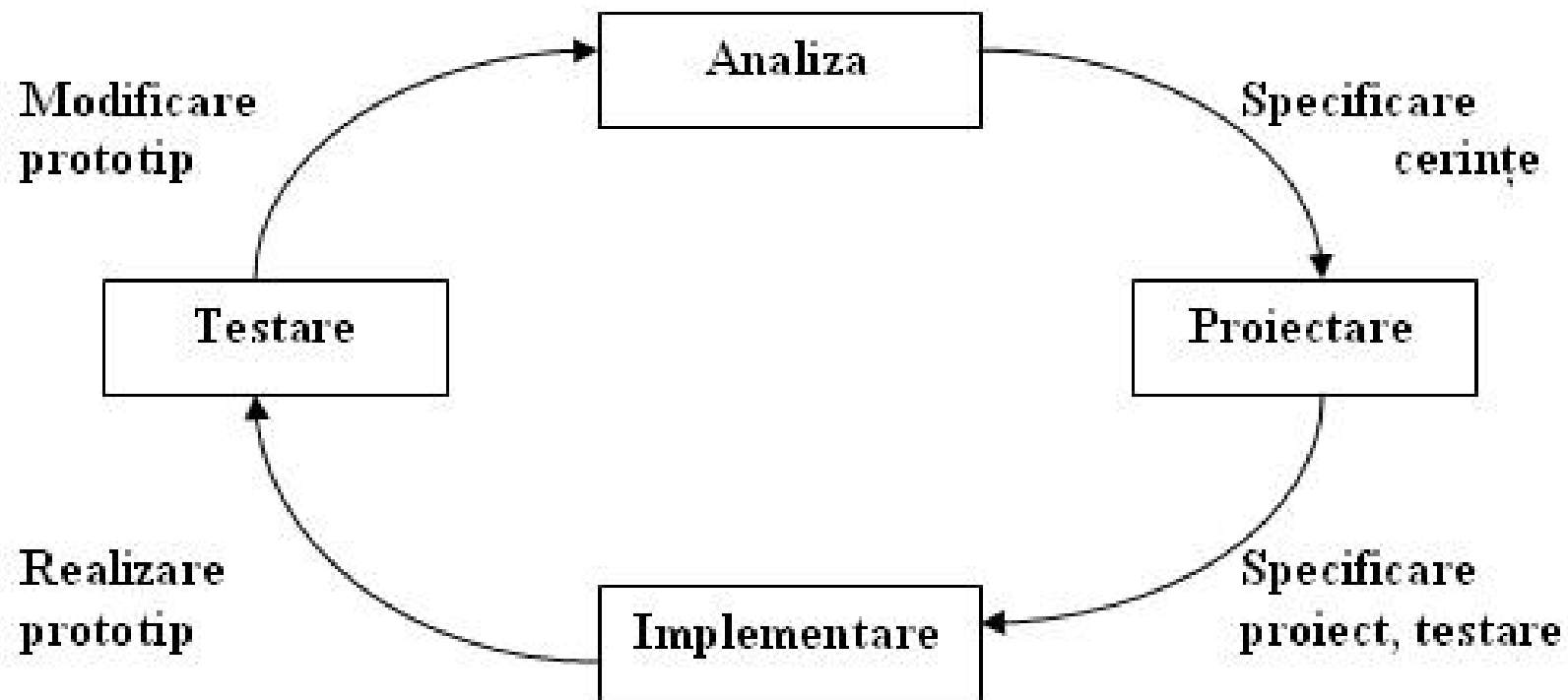


Figura.1. Etapele realizării unui sistem informatic<sup>1</sup>

1. Adaptare: Chaffey, D. - "E-Business and E-commerce Management", Second Edition, Ed. Prentice Hall, 2004, p. 431



UNIUNEA EUROPEANĂ



Instrumente Structurale  
2014-2020

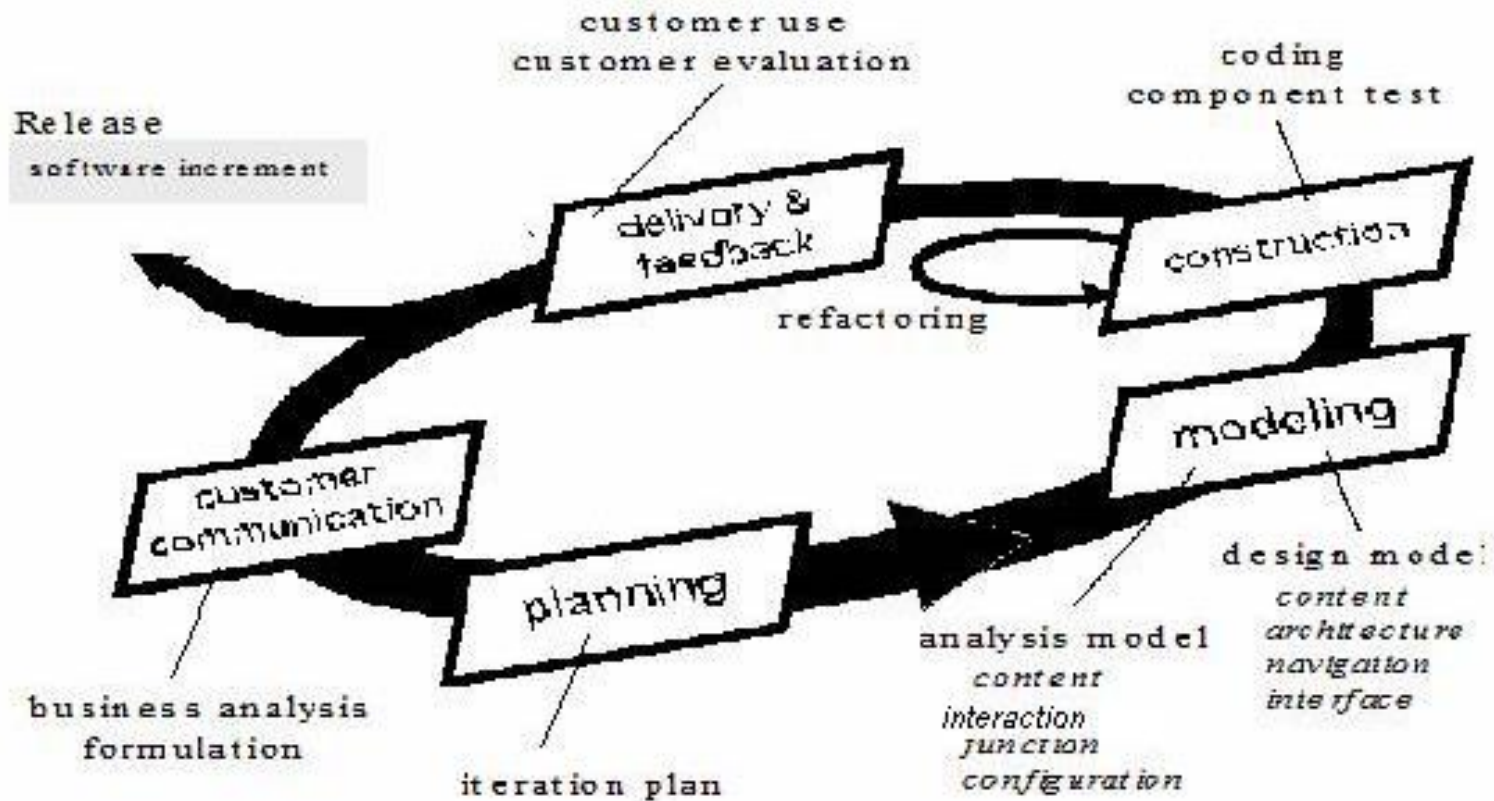


Figura. 2. WebE - după Roger S. Pressman, *Software Engineering, A Practitioner's Approach*, 6-th edition, 2005

Procesul de WebE (inginerie Web) are ca rol central dezvoltarea sistemelor Web. Soluția de dezvoltare incrementală este propusă de [Roger S. Pressman, 2005]



- ▶ Față de sistemele informatice clasice, dezvoltarea aplicațiilor Web are abordări specifice date de actorii participanți și caracteristici.
- ▶ Actorii participanți sunt diversificați: analiști, designeri, experți în marketing, multimedia și securitate și utilizatorii cărora le este destinată aplicația.
- ▶ Caracteristicile principalele ale sistemelor Web sunt: structura de navigare, interfața cu utilizatorul și capacitatea de personalizare.
- ▶ In proiectarea interfeței cu utilizatorul intervin aspecte multimedia, marketing, ergonomie și interacțiune om-calculator.

Din acest motiv modelul analitic pentru aplicații Web este descompus în:

- ❑ Modelul conținutului
- ❑ Modelul interacțiunilor
- ❑ Modelul funcțional
- ❑ Modelul configurației



### Metodologii Web existente:

- NDT - Navigational Development Techniques,
- OOHDM - Object-Oriented Hypermedia Design Method,
- UWE - UML-based Web Engineering,
- W2000
- W3DT - World Wide Web Design Technique
- HDM - Hypertext Design Model
- SOHDM - Scenario-based Object-Oriented Hypermedia Design Methodology
- RNA: Relationship-Navigational Analysis
- WSDM - Web Site Design Method ("WiSDoM")
- UWA- Ubiquitous Web Application
- WebML - Web Modeling Language



- ▶ Un rol important are analiza detaliată a cerințelor ca un punct cheie în dezvoltarea sistemelor software. Ingineria software demonstrează că analiza completă a cerințelor reduce problemele din fazele următoare.
- ▶ Specificarea cerințelor sau descrierea cerințelor este realizată prin diferite tehnici: *descrierea textuală informală, specificația formală (în limbaje Z)*. Un metamodel comun pentru reprezentarea conceptelor și relațiilor ingineriei cerințelor Web (Web requirements engineering (WebRE) este realizat de Escalona și Koch (2004).
- ▶ Unele metodologii Web propun tehnici clasice pentru analiza cerințelor, altele încep ciclul de dezvoltare cu o analiză detaliată a cerințelor: *NDT - Navigational Development Techniques, OOHDM - Object-Oriented Hypermedi Design Method, UWE - UML-based Web Engineering, W2000* [Escalona și Koch, 2004].





- ▶ Clasificarea metodelor de proiectare a aplicațiilor Web poate fi făcută în funcție de: *accentul pus pe proiectarea conceptuală (a problemei -ce)* sau pe cea *logică (a soluțiilor - cum)*.
- ▶ Proiectarea logică în domeniul soluțiilor este tratată de UWA- Ubiquitous Web Application, OOHDM, OOWS - Object-Oriented Web Solution și OO-H - Object-Oriented Hypermedia Method. UWE și WebML - Web Modeling Language sunt metodologii hibride, tratând atât proiectarea conceptuală cât și proiectarea logică [Distante et al. 2007].
- ▶ Tendința actuală este de a se dezvolta spre abordări integrate, care să acopere întreg ciclul de viață al aplicațiilor Web, cum ar fi: OO-H, OOWS, UWE și OOHDM.



**WSDM - Web Site Design Method (WiSDoM)** este o metodologie centrată pe utilizator, care modelează aplicația pe baza cerințelor informaționale ale grupurilor de utilizatori și are în vedere necesitățile vizitatorilor și ceea ce conduce la un grad mai mare de utilizabilitate și de satisfacție (De Troyer, 1998).

Procesul de dezvoltare este împărțit în patru faze:

- **Modelarea audienței**, când utilizatorii sunt clasificați și grupați în vederea studierii cerințelor sistemului potrivit fiecărui grup de utilizatori;
- **Proiectarea conceptuală**, unde este proiectată o diagramă clasă pentru a reprezenta modelul static al sistemului și un model de navigare, pentru a reprezenta posibilitățile de navigare;
- **Proiectarea implementării**, transpune modelele proiectării conceptuale într-un limbaj abstract, ușor de înțeles de către calculator;
- **Implementarea**, în care rezultatul proiectării implementării este scris într-un limbaj specific (php, asp, JavaScript, Ajax, etc).



În faza de modelare a audienței își propune să identifice diferitele roluri ale utilizatorilor în cadrul a două sub-faze:

- **Clasificarea utilizatorilor**, prin identificarea potențialilor utilizatori/vizitatori ai site-ului Web și clasificarea lor în funcție de interesele lor și de preferințele de navigare;
- **Descrierea grupurilor de utilizatori**, adică descrierea, pentru fiecare grup identificat în sub-faza anterioară, a cerințelor informaționale, funcționale și de securitate.

Clasificarea utilizatorilor realizată în prima fază stă la baza celorlalte faze din procesul de dezvoltare. WSDM face o distincție clară între proiectarea conceptuală (care nu conține nici un detaliu de implementare) și proiectarea prezentării, care se bazează pe limbajul de implementare. Această separare este similară cu distincția în proiectarea bazelor de date între schema conceptuală și schema logică.



**HDM (Hypertext Design Model)**: este un model conceptual pentru aplicații hipermedia, care separă modelarea informațiilor, modelarea navigării și modelarea prezentării. Obiectele de tip informație sunt conectate prin legături semantice, iar structurile interne ale obiectelor de tip informație sunt reprezentate prin legături structurale între componente.

**W2000** este o metodologie orientată obiect derivată din HDM, pe principiul SoC (Separation of Concerns) în timpul procesului de dezvoltare.

- W2000 extinde notația UML la modelarea aplicațiilor hipermedia.
- Analiza cerințelor în W2000 este împărțită în două subactivități: analiza cerințelor funcționale și analiza cerințelor de navigare. Fiecare actor identificat în faza de analiză a cerințelor are propriul model de cerințe funcționale și de navigare [Baresi et al., 2001].



W2000 organizează proiectarea unei aplicații hipermedia complexe în jurul a patru dimensiuni:

- *Informația*, definind entitățile informaționale conceptuale de bază percepute de utilizator și diferitele structuri de acces permițând utilizatorului să le acceseze pe baza necesităților sale;
- *Navigarea*, definind structurile de navigare principal în termenii nodurilor și legăturilor (links), permițând utilizatorului să le parcurgă;
- *Prezentarea*, definind structura paginii în termenii aspectelor de prezentare și elementelor grafice, precum și ai organizării și navigării în pagină;
- *Procesul de business*, definind procesele utilizatorului unei aplicații Web în termenii proceselor și operațiilor [Perrone et al. 2004].



**ARANEUS**: separă proiectarea bazei de date și proiectarea hipertext, dar ignoră problemele navigării în favoarea modelării informațiilor. Deși obiectele de tip informație sunt reprezentate ca rețele de componente, rolul central al proiectării bazei de date în ARANEUS impune numeroase restricții pentru obiectele nestructurate de tip informație și proiectarea navigării [Jitaru, 2010].

**W3DT - World Wide Web Design Technique** este procesul de dezvoltare cuprinde 7 pași:

1. Analiza cerințelor, unde este furnizată o colecție de metode pentru a determina cerințele utilizatorilor. Faza de proiectare este împărțită în:
2. structurarea informațiilor,
3. proiectarea navigării,
4. proiectarea organizațională
5. proiectarea interfeței.
6. Implementarea
7. Testarea



- ▶ Structurarea informațiilor și proiectarea navigării sunt abordate ca un proces iterativ, în care un domeniul de cunoștințe este organizat și structurat în obiecte informaționale, care apoi sunt legate și evaluate.
- ▶ În faza de implementare, obiectele informaționale sunt implementate prin pagini HTML și scripturi, iar structurile de acces al modelului sunt implementate prin legături în cadrul site-ului Web.
- ▶ În metamodelul W3DT, un sistem informatic bazat pe Web este compus din pagini, legături și machete și sunt diferențiate două tipuri diferite de legături și patru tipuri diferite de pagini.



**OOHDM - Object-Oriented Hypermedia Design Method:** propune dezvoltarea de modele separate: conceptual, de navigare și interfața abstractă pentru sistemele Web.

Modelul de navigare este construit cu o varietate de concepte, printre care **contextul de navigare**.

Procesul OOHDM este împărțit în patru faze:

- ❑ *Modelul conceptual* - reprezentat ca un model clasă, construit pentru a arăta aspectele statice ale sistemului;
- ❑ *Modelul de navigare* - constă dintr-o diagramă clasă de navigare (reprezentând posibilitățile statice ale navigării în sistem) și o diagramă de structură de navigare (incluzând structurile de acces și contextele de navigare);
- ❑ *Modelul interfeței abstracte* este dezvoltat folosind o tehnică specială numită ADV;
- ❑ *Implementarea*-constă în codul implementat și se bazează pe modelele dezvoltate.





- ▶ OOHDМ a fost extins ulterior cu o tehnică specială ce se ocupă de interacțiunea cu utilizatorul în faza de analiză a cerințelor numită **User Interaction Diagram (UID)** și construită pentru interacțiunea specială a utilizatorului Web cu sistemul Web.
- ▶ Diagramele UID se bazează pe tehnica USE\_CASE (cazurilor de utilizare). Deși USE\_CASE sunt folosite pentru a captarea cerințelor, OOHDМ le consideră ambigue și insuficiente pentru definirea cerințelor aplicațiilor Web, în special în ceea ce privește interacțiunea utilizator și sistem (Human Computer Interaction-HCI).
- ▶ Această metodologie sugerează rafinarea cazurilor de utilizare construind diagrame UID pentru specificarea cerințelor. Diagrame UID sunt folosite pentru a modela grafic interacțiunea dintre utilizator și sistem fără a lua în considerație aspectele specifice ale interfeței [Vilain et al., 2000].



**SOHDM - Scenario-based Object-Oriented Hypermedia Design Methodology** a fost considerată prima abordare cu accent pe importanța unui proces de analiză și care permite să definească cerințele aplicațiilor. SOHDM este similară cu metodologia OOHDM, cu mențiunea că propune o specificare a cerințelor bazată pe scenarii.

Ciclul de viață SOHDM cuprinde șase etape:

- ❑ *Analiza*, faza în care sunt descrise cerințele utilizând scenarii;
- ❑ *Realizarea modelului obiect*, care construiește o diagramă clasă pentru a prezenta structura statică a sistemului;
- ❑ *Proiectarea prezentării*, ce exprimă modul în care sistemul va fi prezentat utilizatorului;
- ❑ *Proiectarea navigării*, unde se dezvoltă un model de clasă de navigare, ce exprimă posibilitățile de navigare în sistem;
- ❑ *Realizarea implementării*, unde sunt dezvoltate paginile Web, interfața și baza de date;
- ❑ *Construirea sistemului*.



- ▶ Definierea cerințelor începe cu proiectarea unei *diagramme de context*, similară diagramei fluxului de date (DFD) definită de Yourdon (1989). Analistul identifică entitățile externe care interacționează cu aplicația și evenimentele care declanșează comunicarea între aceste entități și aplicație.
- ▶ Setul de evenimente este specificat ca un tabel ce arată entitățile ce participă la un eveniment. SOHDM propune asocierea unui scenariu la fiecare eveniment. Reprezentarea grafică a scenariilor este realizată folosind o notație proprietar numită *Scenario Activity Chart*.
- ▶ Un scenariu descrie procesul de interacțiune între utilizator și aplicație, atunci când un eveniment declanșează o activitate. Acesta specifică fluxul activității, obiectele implicate și tranzacția realizată.
- ▶ SOHDM propune un proces pentru a obține modelul conceptual al aplicației din aceste scenarii. Modelul conceptual propus este reprezentat printr-o diagramă clasă. Următorul pas în procesul de dezvoltare SOHDM este regruparea acestor clase cu obiectivul de a obține o diagramă clasă de navigare.



**RNA: Relationship-Navigational Analysis** este o metodologie care oferă o secvență de pași pentru a dezvolta aplicații Web, cu accent pe faza de analiză. Fazele RNA sunt:

- *Analiza mediului*, cu obiectivul de a analiza caracteristicile audienței: actorii aplicației, care apoi sunt clasificați în diferite grupuri în funcție de rolurile lor (similar cu faza de modelare a utilizatorilor din metodologia WSDM);
- *Analiza elementelor*, care identifică toate elementele de interes ale aplicației (documente, forme, informații, mock-ups etc.);
- *Analiza metacunoștințelor*, în care se construiește o schemă a aplicației. RNA își propune să identifice obiectivele, procesele și operațiile legate de aplicație și să descrie relațiile dintre aceste elemente;
- *Analiza navigării*, unde schema din faza anterioară este completată cu facilitățile și caracteristicile de navigare;
- *Analiza implementării*, în care este identificat modul în care modelele descrise în faza anterioară vor fi implementate în limbajul calculatorului.



**HFPM: Hypermedia Flexible Process Modeling** este o abordare ce include strategii de modelare a procesului descriptive, orientate pe analiză, și prescriptive.

HFPM include task-uri tehnice, de management, cognitive și participative, furnizează recomandări (ghiduri) pentru planificarea și conducerea unui proiect Web, acoperind întreg ciclul de viață a unui astfel de proiect software.

HFPM constă din 13 faze, pentru fiecare fiind definit un set de task-uri [Olsina, 1998].

*Modelarea cerințelor* definește următoarele task-uri

- ❑ Descrierea problemei;
- ❑ Descrierea cerințelor funcționale;
- ❑ Modelarea datelor;
- ❑ Modelarea interfeței cu utilizatorul;
- ❑ Descrierea cerințelor ne-funcționale.



*Faza de Dezvoltare* are definite task-urile:

- ❑ Modelarea cerințelor software;
- ❑ Modelarea conceptuală;
- ❑ Modelarea navigării;
- ❑ Modelarea interfețelor abstracte;
- ❑ Proiectarea modelului de lucru; Criterii cognitive;
- ❑ Integrare/modelare fizică;
- ❑ Documentare.



**UWE - UML-based Web Engineering** este o metodologie UML (Unified Modeling Language) pentru Web orientată pe model care acoperă întregul ciclu de viață al dezvoltării aplicațiilor Web, cu accent pe aplicațiile adaptive.

- ▶ UWE oferă: o notație bazată pe UML, o metodologie și un instrument pentru dezvoltarea sistematică a aplicațiilor Web.
- ▶ Proiectarea sistematică urmează principiul **SoC (Separation of Concerns)**, o caracteristică intrinsecă a domeniului Web, și construiește modele separate pentru *cerințe, conținut, navigare, prezentare, proces, adaptare și arhitectură*.
- ▶ UWE include o fază de inginerie a cerințelor specifice în care elicitarea, specificarea și validarea cerințelor sunt tratate ca activități separate ale procesului.



- ▶ Rezultatul final al captării cerințelor în UWE este un model al cazului de utilizare completat cu documentația care descrie utilizatorii aplicației, regulile de adaptare, interfețele și detaliile cazului de utilizare relevant pentru implementare. Documentația poate fi descrisă *textual* sau *modelată* prin diagramele de activitate UML.
- ▶ UWE clasifică cerințele în două grupuri: funcționale și ne-funcționale.
- ▶ Cerințele funcționale privesc:
  - conținutul;
  - structura;
  - prezentarea;
  - adaptarea;
  - modelul utilizator.





- ▶ UWE propune tehnice specifice pentru captarea cerințelor și cazurile de utilizare: *interviurile, chestionarele și listele de verificare (checklists)*, și pentru specificarea cerințelor: *scenariile și glosarele*.
- ▶ Validarea cerințelor în UWE este realizată prin:
  - parcurgeri (walk-through)
  - audituri
  - prototipuri.



**WebML - Web Modeling Language** este un sistem de notații pentru specificarea vizuală a siturilor complexe la nivel conceptual. [Ceri et al., 2000].

- ▶ Limbajul WebML oferă instrumente și notații elementare pentru modelarea hipertext.
- ▶ Baza este modelul Entitate-Relație care folosește concepte simple și expresive pentru specificarea modelelor exprimate prin notații grafice intuitive.
- ▶ Modelul WebML este o prelungire a modelului Entitate-Relație, care dă posibilitate programatorului să extindă schema de date a aplicației cu specificarea hipertextului utilizat pentru publicarea și manipularea datelor.
- ▶ Componentele principale ale WebML sunt: *paginile*, *unitățile* și *legăturile*, organizate în structuri modulare numite zone și vederi ale site-ului (site views).



- ▶ *Unitățile* sunt fragmente atomice de conținut, care oferă alternative de combinare dinamică a conținutului extras din entități și relații al schemei de date. Unitățile permit specificarea formelor de introducere a datelor de către utilizatori.
- ▶ *Paginile* sunt formate din unități, care sunt elementele de interfață oferite utilizatorilor.
- ▶ Paginile și unitățile sunt conectate într-o structură hipertext.
- ▶ *Legăturile* reprezintă baza modelării hipertext și oferă posibilitatea de navigare și transmitere de parametri de la o unitate la alta, pentru generarea conținutului unei pagini.



Descrierile vederilor site-ului constau din două submodele:

- ❑ *modelul compoziție*, care specifică ce pagini compun hipertextul și unitățile de conținut ale acestor pagini;
- ❑ *modelul de navigare*, exprimă legăturile stabilite între pagini și unități de conținut pentru a forma hipertextul;
- ❑ *modelul prezentării*, care exprimă aranjamentul și aspectul grafic al paginilor, independent de tehnologiile de ieșire și de limbajul folosit, prin intermediul unei sintaxe XML abstracte;
- ❑ *modelul de personalizare*, în care utilizatorii și grupurile de utilizatori sunt modelați explicit în schema structurii în forma unor entități predefinite, numite Utilizator și Grup.



Toate conceptele WebML sunt asociate cu o notație grafică, ceea ce îl face accesibil.

WebML poate fi asociat ca un profil UML pentru dezvoltatorii de aplicații Web.

Conceptele WebML pot fi exprimate folosind diagramele de clasa UML și pot fi utilizate în conjuncție cu modelarea orientată obiect UML pentru a extinde analiza dirijată de model.

WebML a stat la baza WebRatio ([www.webratio.com](http://www.webratio.com)). WebRatio pune baza pe modelarea și reutilizarea implementărilor precedente, care sunt generate automat sau semiautomat din modele de nivel înalt.



**NDT - Navigational Development Techniques** este o tehnică utilizată pentru specificarea și analiza aspectelor privind navigarea în aplicațiile Web. NDT pune accent pe tehnici de elicitare și specificare selectate de NDT pentru captarea și definirea cerințelor.

Fluxul de lucru al analizei cerințelor prin aplicarea unor tehnici ca interviul, brainstorming-ul și JAD [Escalona et al. 2004]. În pasul al doilea sunt captate și descrise obiectivele sistemului. Pe baza acestor obiective sunt identificate cerințele sistemului.

NDT le clasifică în: *cerințe de stocare a informației, cerințele actorilor, cerințe funcționale, cerințe de interacțiune, cerințe nefuncționale.*

- ▶ Cerințele de interacțiune sunt reprezentate prin fraze și prototipuri de vizualizare.
- ▶ Frazele arată cum sunt regăsite informațiile sistemului și sunt reprezentate printr- un limbaj special **BNL (Bounden Natural Language)**.
- ▶ Prototipurile de vizualizare sunt utilizate pentru a reprezenta sistemul de navigare, vizualizarea datelor și interacțiunea utilizatorului.
- ▶ Procesul de elicitare și specificare a obiectivelor și cerințelor propus de NDT se bazează pe șabloane (templates) și modele (patterns).



**UWA - Ubiquitous Web Applications design framework** oferă o metodologie și un set de modele și instrumente pentru proiectarea conceptuală, centrată pe utilizator, a aplicațiilor Web ubicue.

UWA este destinată pentru proiectarea aplicațiilor care intenționează să fie **ubicue** accesibile diferitelor tipuri de utilizatori, în diferite contexte de utilizare și cu diferite scopuri și HCI (centrate pe utilizator) [UWA Consortium, 2002].

Problema modelării este divizată în aspecte specifice ale proiectării :

- elicitarea cerințelor, pentru a defini ce aplicație trebuie realizată;
- proiectarea hipermedia, pentru a modela datele și modul în care pot fi parcurse (navigare) și prezentate și operațiile (serviciile) care sunt disponibile utilizatorului;





- proiectarea tranzacțională, pentru a modela comportamentul tranzacțional expus de aplicație;
  - proiectarea personalizată, pentru a specifica modul în care aplicația trebuie să se adapteze la context și în special la utilizator, tehnologie, canale de comunicație și locație.
- ▶ Fiecare activitate de modelare este definită în termenii unui *metamodel*, care captează setul de concepte și primitive relevante, o notație, bazată pe UML pentru a reprezenta conceptele, un set de recomandări (ghiduri) și euristici.
- ▶ Proiectantul exploatează conceptele și înțelege negocierile între diferite soluții de proiectare și un set de instrumente, pentru a realiza procesul de proiectare și a forța coerența și consistența proiectării.
- ▶ UWA furnizează un cadru de lucru unificat, care integrează diferite metamodele și notații în interdependență reciprocă și un mediu software unificat, bazat pe Rational Rose, cu instrumente specifice fiecărei activități de modelare.



## 2. Dezvoltarea aplicațiilor Web

### 2.1 Fazele de realizare a aplicațiilor Web

Pentru realizarea unei aplicații profesionale multimedia sau Web se parcurg 5 faze majore (figura 4):

- ❑ **Conceptia** (culegerea specificațiilor și analiza)
- ❑ **Realizarea propriu zisă** ( proiectarea, testarea, realizarea prototipului)
- ❑ **Publicarea și promovarea aplicației** (testarea prototipului)
- ❑ **Exploatarea și întreținerea aplicației** (modificarea și dezvoltarea prototipului)
- ❑ **Evoluția aplicației** ( dezvoltarea acesteia prin parcurgerea unui nou ciclu de analiză, proiectare, implementare, testare)

Modelul iterativ este adecvat aplicațiilor web deoarece un aspect esențial al acestor aplicații este timpul limitat și modificările frecvente efectuate pentru satisfacerea cerințelor.

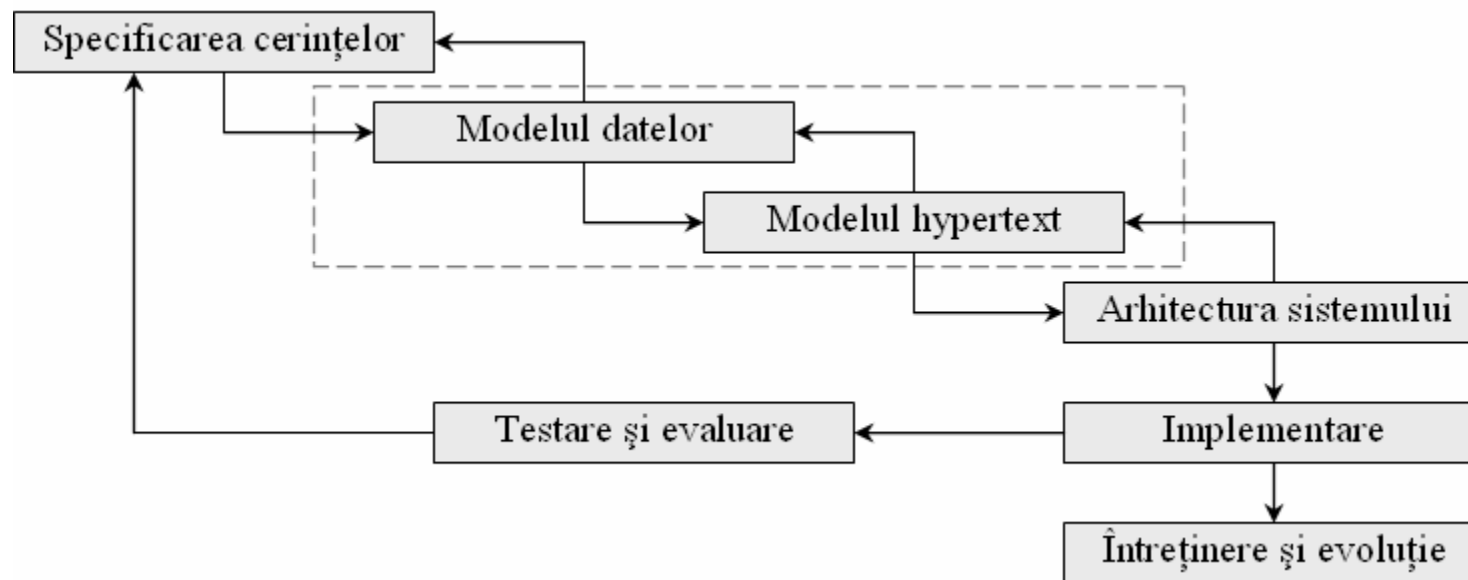


Figura 4. Procesul iterativ de dezvoltare a ste-ului [web](#)<sup>3</sup>

Sursa: Oleg Burlaca - "Sisteme de Management a Conținutului Web ", p.23



*Specificarea cerințelor:* este procesul în care analistul colectează și formalizează informația esențială despre domeniul viitoarei aplicații și a funcționalității așteptate. Rezultatul acestei faze este o specificare adresată proiectanților și clientului pentru validare, conform cerințelor înaintate de acest client. Aprobarea specificației de către client înseamnă că cerințele i-au fost satisfăcute și se poate trece la etapele de proiectare.

Modelele de date pentru Web se deosebesc de aplicațiile obișnuite deoarece în aplicațiile web prevalează operațiile de citire a datelor la etapa construirii dinamice a paginilor. Elaborarea schemei bazei de date cu satisfacerea formelor normale uneori nu este justificată practic.



- ▶ Modelarea hipertext are ca scop specificarea interfețelor aplicației web: structura ierarhică a paginilor site-ului, datelor utilizate pentru construirea paginilor, legăturilor dintre pagini.
- ▶ Modelarea datelor și hipertextului sunt strâns legate, totuși, abordările existente le examinează separat, motivând aceasta prin gradul de flexibilitate obținut. Prețul acestei flexibilități este timpul pentru ajustarea unui model la schimbarea celuilalt (de exemplu, compatibilitate, optimizare).
- ▶ Proiectarea arhitecturii constă în definirea elementelor hardware, de rețea și componentelor soft ce stau la baza aplicației ce oferă servicii utilizatorilor săi. Scopul acestei etape este identificarea unei combinații optime ale elementelor menționate pentru satisfacerea cerințelor înaintate din punct de vedere al performanței, securității, scalabilității, și costurilor.



UNIUNEA EUROPEANĂ



- ▶ Testarea și evaluarea sunt activități de verificare a conformanței aplicației implementate la cerințele înaintate. Menționăm testarea funcționalității, performanței și utilității (usability). Întreținere și evoluție reprezintă modificările efectuate după lansarea aplicației. Propunerile și cerințele față de o aplicație existentă sunt analizate și efectuate modificări în modelul date sau hipertext.



## A. Concepția aplicației

**Concepția** reprezintă faza în care firma care dorește realizarea unei aplicații multimedia își definește obiectivele aplicației și stabilește conținutul și sursele de lucru.

Aceasta presupune crearea aplicației, crearea conținutului, constituirea surselor de lucru, *planing*, estimarea timpului necesar realizării aplicației, repartiția sarcinilor în echipă și managementul fazei.

**Crearea aplicației** presupune stabilirea celor patru obiective specifice aplicațiilor multimedia: *concept, filosofie, inginerie și public*.

**Crearea conținutului** implică o concentrare a eforturilor echipei în realizarea elementelor media specifice publicațiilor electronice:

- ▶ *Texte* care presupun redactarea formatului numeric direct;
- ▶ *Fotografii* (realizate cu aparate fotografice clasice și numerice, camescope)
- ▶ *Ilustrații* care utilizează grafica tradițională sau vectorială, design, picturi etc;
- ▶ *Audiovizual* (achiziția sunetelor, înregistrarea și crearea sunetelor, filme, documentare etc).



- ▶ **Constituirea surselor de lucru** pornește de la cele *existente deja în proiect* și identifică sursele necesare a fi *create prin proiect*: achiziția de fotografii, audio, video (medii audiovizuale).

Pentru sursele existente cum sunt: texte în format numeric, ameliorarea de text, OCR etc, se continuă cu conversia numerică a imaginilor realizate, existente deja (fotografii, desene, ilustrații, diapozitive) și se finalizează cu achiziția de elemente audio, video.

- ▶ **Managementul fazei** reprezintă activitatea de coordonare a echipei, în care sarcina șefului de proiect este dată de identificarea echipamentelor, drepturi de utilizare, delimitarea elementelor media “brute” și a celor rezultate din prelucrări *planing*, estimarea timpului, repartitia sarcinilor în echipă.





Echipamentele necesare realizării fazei presupun: calculatoare, soft specializat, echipamente periferice dedicate graficii (scanere, camere video, magnetoscop etc.). Drepturile de utilizare sunt necesare pentru respectarea cadrului legislativ internațional și sunt stabilite la nivel software și al tipurilor media (drepturi de autor, copyright etc.)

- ▶ **Rezultatele fazei** sunt date atât de expertiza tehnică de sistem, estimarea timpului de dezvoltare cât și de arborescența, organizarea tipurilor media la nivel de structuri de directoare, conținut, tablou de bord (storyboard) și elemente de animație.



## **B. Realizarea propriu zisă a aplicației multimedia**

Pentru a realiza o aplicație multimedia se parcurg succesiv fazele descrise în figura 4. În această fază diferite task-uri sunt realizate paralel, sub coordonarea șefului de proiect.

Producția de aplicații multimedia solicită separarea task-urilor repetitive de către executanți, utilizarea unor scripturi pentru accelerarea lor, ceea ce conduce la economie de timp, eficacitate și productivitate.

Se recomandă concentrarea asupra unor sarcini bine precizate, ceea ce conduce la creșterea randamentului întregii echipe.



- ▶ **Tratarea și pregătirea surselor pentru integrare** este sarcina infograficianului și presupune:
  - Retușarea imaginilor, montaje, efecte speciale, optimizări;
  - Dimensionarea imaginilor la precizia constrânsă de proporțiile acestora (spre exemplu la video 4/3);
  - Export și compresie de imagine la suporturile solicitate de alte medii;
  - Crearea unei identități vizuale ca parte a personalizării site\_ului sau adaptarea celei existente)

Aceasta este etapa care nu permite estimarea timpului necesar deoarece are o evoluție permanentă, în funcție de fazele în care se află proiectul. În această fază se crează un **model de pagină** (*template*, tipuri predefinite, etc) corespunzătoare integrării bazelor de date și a altor elemente specifice site-urilor dinamice.



## Stabilirea elementelor de interfață, navigare, interactivitate

- ▶ **Ergonomia** site-ului este dată de modul în care este organizată parcurgerea acestuia și legăturile realizate în acest scop și presupune stabilirea elementelor de interfață, navigare, interactivitate: site-uri vrac; cadre cu cadre; Shockwave, Flash etc.; DHTML; Javascript; Php, alte limbaje de programare; personalizarea site-ului; trucuri (gimmicks) de atragere a vizitatorilor pe tehnologie pull; aspecte de confort ale utilizatorului
- ▶ **Realizarea animației** este sarcina infograficianului care utilizează în acest scop medii software orientate spre grafica vectorială: Flash, Shockwave alte medii software orientate spre animație.



**Dezvoltarea aplicației** este dependentă de complexitatea acesteia precum și de soluția tehnică de realizare și are ca obiective:

- Concepția bazelor de date (motoare de cautare, rezultate ale căutărilor, navigare)
- Integrarea altor servicii (medii multi-user, forum de discuții, chat, motoare de căutare, agenți, aplicații specifice etc)
- Programarea secvențelor specifice.



**Integrarea media și a bazelor de date** este faza în care se reunesc produsele specifice media, serviciile specifice și interfețele componente ale aplicației, fiind materializată prin:

- Montarea paginilor și a elementelor grafice, modele;
- Integrarea serviciilor specifice (programe, CGI, Java, forum etc);
- Integrarea interfețelor cu bazele de date (SQL, Php, file, maker, WEBobjects, Oracle etc).

Administrarea serverului este o sarcină permanentă a Webmasterului echipei de proiectare, care asigură funcționalitatea aplicației precum și dialogul permanent între membrii echipei, via Intranetul firmei.



Webmasterul este responsabil de asigurarea resurselor software și hardware necesare proiectării, testării și publicării aplicației prin:

- Deschiderea unui cont ftp;
- Gestiunea drepturilor utilizatorilor;
- Securitatea site-ului;
- Introducerea solicitărilor WEB specifice (alias, redirectare, DNS, realvideo, alte tehnologii);
- Instalarea pe server pentru buna funcționare a programelor elaborate (CGI, Java, Forum etc);
- Administrarea BD.



C. Finalizarea aplicației are urmărește **verificarea și testarea** acesteia respectiv **publicarea aplicației** fiind realizată de tester în colaborare cu întreaga echipă de proiectare și urmărește:

- Verificarea codurilor HTML și a diverselor aplicații intergrate;
- Testarea cu ajutorul diferitelor browsere și pe diverse mașini ale utilizatorului;
- Corecții de site care conduc la versiuni beta, demonstrativă și finală;
- Inregistrarea propriu zisă;
- Publicitate online și pe rețele sociale, promovare prin metode specifice Internetului.





**D. Exploatarea și întreținerea aplicației** este o sarcină permanentă a echipei de realizare.

Întreținerea sistemului este realizată de Webmaster și se referă la: utilizatori, servicii on-line, probleme tehnice, hotline, intervenții. Webmaster-ul gestionează pe întreg parcursul exploatarea feed-back al utilizatorilor, actualizarea, statisticile curente și asigură politica de menținere în funcțiune.

**Evoluția aplicației** are trei coordonate majore: comunicația cu vizitatorii site-ului, administrarea sistemului și elaborarea de versiuni noi ale aplicației. Elaborarea unor versiuni noi are loc începând de la faza de prototip și până la proiectul final diferite task-uri sunt realizate paralel, sub coordonarea șefului de proiect.



## **2.2 Echipa de realizare**

Echipa de elaborare a proiectului cuprinde câțiva specialiști care au competențe distincte și rol bine definit în realizarea acestui proiect.

**Seful proiectului:** coordonează dezvoltarea unei aplicații multimedia și asigură succesul artistic, tehnic și informatic, de la faza de concepție până la finalizarea concretă a acesteia.

### ***Sarcini:***

- Proiectează, coordonează și controlează procesul de producție în cadrul unui proiect multimedia
- Realizează structura site-ului în colaborare cu infograficianul
- Răspunde de evaluarea resurselor necesare pentru realizarea proiectului



- Organizează echipa necesară realizării proiectului multimedia
- Planifică și distribuie sarcinile în echipă
- Coordonează bugetul proiectului, calitatea și respectarea normelor legislative
- Elaborează documentația în toate fazele proiectului

**Webmaster:** responsabil de mentenanța produsului sau serviciilor multimedia.

***Sarcini:***

- Asigură și menține comunicația între membrii echipei furnizând resurse hardware și gestionând tranzacțiile, mesajele sau schimbul interactiv de informații între utilizatori
- Menține statisticile de frecvență a unui site, pentru a permite evaluarea eficacității sau rentabilității procedurilor utilizate, asigurând ameliorarea acestora

- Realizează integrarea componentelor în vederea testării site-ului precum și optimizarea acestuia pe platforme diferite: Windows 98, Windows NT, Unix, Microsoft Internet Explorer 2.x, 3.x, 4.0/5.0; Netscape Navigator 2.x, 3.x, 4.0
- Testează legăturile, controalele, cookies ONN/OFF, etc.
- Testează soluțiile dinamice pentru a furniza informații referitoare la diverse rezoluții, viteze ale modemurilor, etc.
- Asigură legătura cu furnizorii de acces Internet precum și referințe la motoarele de căutare
- Răspunde de mentenanța și securitatea site-ului
- Posedă cunoștințe referitoare la rețele de calculatoare, resursele sistem, protocoale de comunicație, sisteme de operare (Windows NT, UNIX etc)

**Webdesigner**: realizează elementele de interfață și navigare precum și cele de interactivitate cu vizitatorii site-ului.

**Sarcini:**

- Proiectează și realizează interfețe SQL / PHP
- Dezvoltă alte aplicații: scripturi, CGI proceduri C++, appleturi Java, servleți etc.

**Programator multimedia**: asigură realizarea tehnică și dezvoltarea informatică a unei aplicații multimedia pe baza analizei, dezvoltării și optimizării, folosind aplicații interactive cu funcționalități specifice.

**Sarcini:**

- Asamblează și integrează diferite medii componente și programează aplicații specifice
- Gestionează și manipulează elemente de interacțiune și realizează interfețele asociate CGI, appleturi etc.

- Elaborează aplicații on-line: Javascript, Visual Basic Script, Perl, ASP, PHP
- Elaborează aplicații off-line: Lingo pour Macromedia Director, ce servesc produselor off-line
- Asigură funcționalitatea globală a aplicației multimedia prin proiectarea modulelor client-side folosind C, C++, Java, .NET, SGBD specifice SQL pentru ORACLE, SQL Server, Access
- Posedă cunoștințe solide de : HTML, XML, SQL/PHP, Javascript, sisteme de operare și protocoale Visual C++ , MFC, API Win é, GUI Develeopment (GDI), COM, ActiveX, ATL, Shell Programming și Internet Programming SQL Server, HTML, FrontPageXpress, etc., modelare (forum, chat,mail), statistici
- Manipulează soluții de business intelligence ce permit analiza feedback-ului cu clienții, vizitatorii, mediul de afaceri.

**Infografician (infodesigner)**: specialist în grafica 2D, 3D care asigură retușarea imaginilor, fotografiilor și a altor elemente grafice, design sonor, video, montaj numeric, imagini de sinteză etc.

**Sarcini:**

- Realizează structura grafică pornind de la arhitectura site-ului
- Achiziționează imagini sursă 2D, prelucrează și crează imagini de sinteză
- Ameliorează paginile și pregătește ecranele
- Realizează elemente de animație
- Posedă cunoștințe de hardware multimedia (scanner, photo, video) și software multimedia de grafică vectorială (Illustrator, Corel, Photoshop), prelucrări de imagini (Quark Xpress, Flash, Director, 3D Studio Max, SoftImage, etc)

### 2.3 Analiza cerințelor

Pentru proiectarea unei aplicații Web interconectate sau chiar o pagină comercială, ce se dezvoltă spre un site complicat, abordarea într-o manieră sistematică este absolut necesară. Se poate utiliza una din metodologiile descrise anterior. Fără a poseda prea multe cunoștințe de proiectare a sistemelor informatice vom aborda metodologia **W3DT** în 7 pași.

**Analiza cerințelor**, unde este furnizată o colecție de metode pentru a determina cerințele utilizatorilor.

#### **Scopul**

- Ca și în cazul altor proiecte, cele mai bune rezultate se pot obține prin stabilirea exactă a ceea ce urmează să realizeze paginile Web.
- O parte a stabilirii scopului paginilor o reprezintă identificarea, cât mai precisă cu putință, a **auditoriului** căruia urmează să i se adreseze informația din pagini.



- Instrumentele de identificare a celor care vizitează o pagină Web sunt limitate și, până acum, nu există nici un standard acceptat pentru modul în care să se contorizeze numărul de utilizatori care asigură amortizarea investiției și obținerea unui profit, sau a numărului de oameni din audiența țintă care au recepționat mesajul transmis.
- Dacă se alocă inițial un interval de timp pentru realizarea unei estimări orientative a categoriilor de utilizatori cărora li se adresează site-ul, pe parcursul proiectării se poate urmări evidențierea în site a elementelor care ar trebui incluse ca să atragă acești utilizatori. Se pot analiza mai ușor ce legături externe trebuie încorporate în site și care să fie site-urile care să conțină legături spre el.

**Audiența țintă:** va juca un rol major în modul cum va arăta site-ul.

Există implicații majore *legate de design și determinate de audiență*:

- vârsta audienței impune anumite tipuri de fonturi (informația trebuie să fie cât mai lizibilă pentru vârstele înaintate),
- culorile trebuie să țină cont de tabuurile naționale (în unele țări negru este culoarea doliului, în alte țări este alb),
- culorile trebuie puse în acord cu tematica site-ului (un site academic va fi mai sobru, unul destinat jocurilor va fi mai vesel)
- exprimarea trebuie să fie adaptată (un *site informativ* va folosi pluraluri: **noi vă** informăm, **noi vă** sfătuim; un *site comercial* va folosi singularul persuasiv: **noi îți** arătăm, **îți** vindem, **te** sfătuim, o *pagină personală* va folosi singularul intim: **eu vă** informez, **eu** caut, **eu vă** sfătuiesc, un *site ce se adresează indivizilor* în probleme intime poate folosi adresarea singulară directă: **eu te** sfătuiesc).

Se stabilește dacă site-ul are nevoie de *versiuni redundante* în mai multe limbi, cu/fără prezentări multimedia, cu/fără structuri de tipul cadrelor și tabelor.

Pentru a atrage vizitatorii să revină, trebuie planificată o modificare frecventă a elementelor paginii, pentru a le menține interesul. Aceasta presupune un grad înalt de flexibilitate.

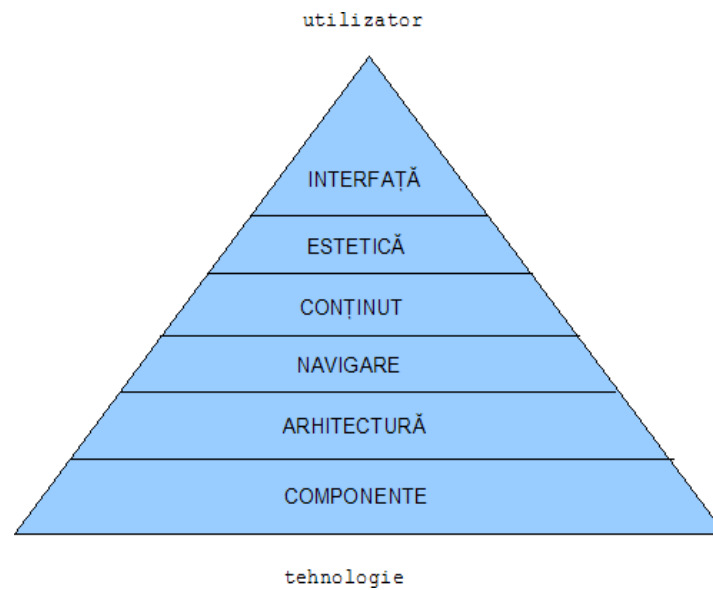
Un factor care trebuie luat în considerare în proiectarea paginilor Web, mai ales a celor realizate în cooperare, este întreținerea paginii (timp și cunoștințe de specialitate).

## 2.4 Etapele proiectării aplicațiilor Web

- ▶ Etapele proiectării pot fi ierarhizate într-o piramidă a cărei vârf este utilizatorul (beneficiarul aplicației) și a cărei performanțe depind de tehnologia utilizată.
- ▶ Proiectarea interfeței presupune structura și organizarea interfeței utilizator (UI) ceea ce implică: *layout-ul ecranului, modurile de interacțiune și mecanismele de navigare.*
- ▶ Proiectarea esteticii-(LOOK&FEEL) se focusează pe: scheme de culori, geometria layout-ului, font, dimensiune și amplasare text, utilizarea graficii, etc.
- ▶ Proiectarea conținutului are în vedere: layout, structură, schiță conținut, relațiile dintre obiectele conținut.
- ▶ Proiectarea navigării implică; fluxul navigării, legături între obiectele de conținut, pentru toate funcțiile aplicației etapele proiectării pot fi ierarhizate într-o piramidă a cărei vârf este utilizatorul (beneficiarul aplicației) și a cărei performanțe depind de tehnologia utilizată.

- ▶ Proiectarea arhitecturii se concentrează în două direcții: conținut ( prin structura hypermedia de ansamblu și la nivelul aplicației (prin componente și relații).
- ▶ Proiectarea componentelor presupune detalierea logicii de procesare necesară implementării componentelor funcționale, [Mandruță, 2011].

Figura 4 - Cristina Măndruță, Proiectarea sistemelor software.



## Proiectarea conținutului

- Schițarea conținutului materialului care urmează să fie inclus în pagina Web stabilește ce **materiale** trebuie colecționate, care trebuie create, și care trebuie rescrise pentru a fi compatibile cu site-ul on-line. O parte din informații poate fi conținută în legături către documente din alte site-uri, create anterior, consultând anterior limitările de licențiere sau **gradul de sharing** permis de proprietar.
- Se stabilesc care din informații vor fi numai de tip text, care din zonele de text vor fi derulante, care din porțiunile de text vor fi prezentate pe blocuri, astfel încât să încapă perfect într-o fereastră a programului de afișare.
- Se realizează împărțirea conținutului pe pagini ce urmează să fie puse în legătură. Pentru pagina principală se stabilește dacă este sau nu nevoie de un intro multimedia, și se împarte spațiul de afișare între elementele structurale: se aleg bannerele, titlurile paginilor, se stabilesc linkurile externe și interne, relative sau absolute.

- ▶ Organizarea de bază a paginilor succede schița globală a conținutului paginilor Web, cu legăturile externe dorite. Aceste pagini pot fi așezate într-o succesiune liniară, ca la o prezentare de diapozitive, trecerea de la o pagină la alta făcându-se cu ajutorul săgeților Next (înainte) și Back (înapoi), într-o structură arborescentă sau, dacă aceasta este prea rigidă, chiar într-una hibrid, în care să existe multiple legături între pagini, și pot fi așezate, desigur, într-o structură formată din combinarea celor de mai sus. Se poate construi această hartă pe o pagină de hârtie cu creionul, sau cu instrumente computerizate (NaviPress care permite vizualizarea interconexiunilor corespunzătoare unui set de pagini Web).
- ▶ Softurile de dezvoltare HTML/JavaScript, Adobe includ un gestionar de site (**site-map**), un sistem de gestiune ce asigură actualizarea automată a link-urilor la mutarea fișierelor, întreținerea hărții site-ului și a unor fișe de observații pentru fiecare fișier/link, un client FTP și un mecanism de întreținere a fiabilității prin generarea automată a unui raport cu erori de funcționare. În pagina principală se recomandă prezența hărții site-ului, pentru a ușura orientarea utilizatorilor în cadrul paginilor Web.



**Un site** constă, din punct de vedere al accesului, din una sau mai multe pagini care conțin text, grafică, animație, etc., legate într-o manieră coerentă, formând un tot unitar, în vederea descrierii unor informații sau creării unui efect constant [RUSU\_01].

Paginile Web sunt denumite generic documente, termenul de pagină sau carte fiind determinat de conținutul fișierului (.htm). Practic pagina este acel fișier care se regăsește și se formatează de către browser.

**Home page** sau pagina principală a prezentării conține de obicei un sumar, informații generale, despre celelalte pagini iar URL-ul acesteia permite celorlalți utilizatori consultarea site-urilor Web.

Structura prezentărilor este dependentă de conținutul acestora precum și de obiectivul specific al paginii Web (pagina personală, prezentare de firmă, aplicație comercială, comerț electronic, învățământ asistat, etc.).

Astfel, structura unei pagini Web poate fi încadrată în una din următoarele categorii de structuri definite:

- **Structură liniară:** pentru acest tip de organizare, legăturile se realizează de la o pagină la alta (înainte și înapoi). În acest caz se recomandă o legătură suplimentară cu Home page pentru a traversa rapid prezentarea respectivă. Acest tip de structură este utilizat în cazul materialelor documentare scurte, instrucțiuni pas cu pas, instruiri asistate de calculator.
- **Structura ierarhică:** permite rafinarea informației din prezentare de sus în jos prin detalierea succesivă în funcție de interesul manifestat de utilizator. În această manieră fiecare nivel are o interfață consistentă (sus, jos) și în același timp setul de posibilități de selecție este limitat la navigarea de bază. Structura ierarhică are avantajul traversării rapide a paginilor pornind de la nivelul superior până la nivelul cu cel mai mare grad de detaliere a informațiilor.

Dezavantajul acestei structuri apare atunci când există prea multe nivele, prea multe meniuri, și apare posibilitatea ca utilizatorul să uite de solicitarea inițială, efectul fiind cunoscut sub denumirea de "*voice mail syndrom*". Se recomandă limitarea la două sau trei nivele de ierarhizare, combinând, dacă este necesar, informațiile din paginile anterioare cu nodurile finale (frunzele).

- ***Structura liniară cu alternative***: Prin această structură se îmbunătățește (se micșorează) rigiditatea structurii liniare prin derivarea căii dintr-o cale principală, în ramuri care pornesc dintr-un punct. Aceste ramuri pot fi dezvoltate liniar în continuare sau pot fi reunite într-un punct al unei alte ramuri principale.

Această structură este cel mai popular mod de organizare a prezentărilor și reprezintă combinația dintre documente liniare și ierarhizate, deoarece deplasările de sus - jos, înainte și înapoi sunt mai facile decât traversările în cadrul ierarhiilor. Exemplul cel mai cunoscut pentru o astfel de structură sunt prezentările FAQ (Frequently Asked Questions).

- ▶ Indiferent care tip de structură este indicată pentru scopul și conținutul site-ului, în cazul unui site complex este indicată schițarea unei hărți sau a unei scheme bloc pentru paginile Web, în care pentru indicarea legăturilor se pot utiliza, de exemplu, linii cu săgeți.
- ▶ Aceste structuri pot fi realizate prin proiectarea unor legături consistente, necesare și suficiente, în multe situații elementele de legătură fiind realizate prin combinații de text și grafică sau prin butoane grafice interactive.

## Proiectarea interfeței

Dezvoltarea unei pagini WEB a unei firme sau a unei aplicații complexe (e-busienss, e-commerce, etc) trebuie să fie precedată de un **studiu de fezabilitate** și să aplice în primul rând **principiile proiectării** sistemelor informatice: principii de eficiență economică, ceea ce presupune evaluarea cheltuielilor necesare pentru conceperea, realizarea, implementarea, publicarea și întreținerea curentă a paginii WEB și compararea lor cu efectele economice directe și indirecte obținute de firma care-și face publicitate, managementul proiectului (surselor și a a echipei de dezvoltare) în fiecare fază de realizare, asigurarea calității soluțiilor abordate, prin aplicarea celor mai eficiente metode și tehnici de proiectare și specificarea unor caracteristici de calitate, care să fie controlate pe tot parcursul realizării paginii, adaptarea soluțiilor în concordanță cu resursele disponibile.

Caracteristicile esențiale ale unei interfețe Web sunt : *simplu de utilizat, ușor de învățat , simplu de navigat, intuitivă, consistentă, eficientă, fără erori, funcțională.*

## Principiile de proiectare a interfeței:

- *Anticiparea* următoarei acțiuni a utilizatorului și oferirea opțiunilor de navigare adecvate.
- *Comunicarea* sării oricărei activități inițiată de utilizator, starea (identitatea) utilizatorului și poziția în ierarhia conținutului.
- *Consistența* relativ la reprezentare controale, menu, pictograme, estetică.
- *Autonomia controlată* facilitarea navigării, cu impunerea convențiilor de navigare stabilite pentru aplicație.
- *Eficiența* din perspectiva utilizatorului.
- *Flexibilitatea* în raport cu diferite categorii de utilizatori; posibilitatea de a reveni din erori sau de pe căi de navigare alese greșit (UNDO).
- *Focalizarea* pe activitățile curente ale utilizatorului.
- *Respectarea Legii lui Fitt*: “timpul de obținere a unei ținte este funcție de distanța la ea și de dimensiunea țintei” referitoare la obiectul de interfață.

- ▶ *Obiecte de interfață* predefinite folosind biblioteca existentă.
- ▶ *Reducerea latenței* prin multitasking; feedback audio și video la lansarea operațiilor la care răspunsul nu este imediat; afișarea unui control animat al progresului (timer).
- ▶ *Ușor de învățat* cu design simplu și intuitiv; organizarea conținutului și funcționalității în categorii evidente pentru utilizator.
- ▶ *Metafore corespunzătoare* aplicației și utilizatorilor acesteia; imagini și concepte din experiența reală a utilizatorilor, augmentate cu facilități ajutătoare (completare formular tip cu posibilități de selecție din liste, modele de introducere a datelor predefinite).
- ▶ *Păstrarea integrității* prin salvare automată a tuturor datelor specificate de utilizator ( a formelor completate) în vederea recuperării la apariția unor erori.
- ▶ *Lizibilitate* la nivel de font-uri, dimensiuni și culori, ce trebuie să fie bine alese.
- ▶ *Urmărirea stării* prin memorarea succesiunii stărilor interacțiunii cu utilizatorul în vederea continuării acesteia la o reconectare ulterioară.
- ▶ *Navigare vizibilă* prin crearea iluziei că aplicația vine în întâmpinarea utilizatorului (tehnica pull), oferindu-i facilitățile de navigare necesare în contextul respectiv.

Principiile fundamentale legate de **Web design** au amprenta particularităților de proiectare:

- ▶ Indiferent de tipul de fișier care se prelucrează, trebuie urmărită dimensiunea sa minimală, prin aplicarea compresiilor în balanță cu rolul pe care îl va juca acel fișier în site: cele care au rol de informație pusă la dispoziție vor fi slab compresate și stocate în biblioteci sau arhive FTP; cele care au rol în design-ul paginii, vor fi compresate cât mai puternic; dacă stilistica adoptată o permite, se recomandă fundaluri monocrome și obiecte grafice alb-negru;
- ▶ Elementele trebuie să fie de așa natură așezate în pagină, încât indicatorul NREM (NotReadingEye Movement) să fie minim. NREM indică distanța pe care o parcurge privirea fără să citească nimic.



Hiperlegăturile trebuie să fie de patru tipuri:

1. Setul hiperlegăturilor structurale: dau o imagine privind structura site-ului și indică spre componentele sale (Register, FAQ, etc.); de regulă acestea se grupează pe o bară orizontală, imediat sub logo-ul site-ului;
2. Setul hiperlegăturilor de conținut: au rolul de cuprins al informației: (titluri de capitole, thumbnails etc).
3. Setul legăturilor externe, care indică în afara site-ului: de regulă sunt grupate într-o pagină la care se ajunge printr-un link structural, de genul “Other Links”, “Other Sources”;
4. Setul legăturilor interne, care sunt referințe încrucișate ce apar în conținutul informațional al site-ului; în unele browsere vor apare subliniat sau colorate cu albastru, motiv pentru care se recomandă evitarea sublinierilor și a culorii albastre pentru scris.

Designul reprezintă modul în care se armonizează elementele paginii (grafica, text, link-uri, elemente interactive, imaginile etc.) și se referă la relația între elementele paginii: locația, proporția, accentul pus pe fiecare element. Elementele designului: logo-ul și numele firmei, imaginea sau grafica, titlul și textul, alte ilustrații, sunt tratate atât la nivel de **structură a informațiilor** cât și a **structurii imaginilor**.

*Structura informațiilor* este centrată pe respectarea **regulii celor cinci degete**. Specialiștii afirmă că pentru a obține o eficiență maximă într-o înșiruire de tip listă e bine să existe cel mult 5 opțiuni deoarece 60% din vizitatori oricum vor citi numai pe primele 5, ignorându-le pe restul informațiilor. Imaginile animate care consumă resurse, ar trebui înlocuite cu imagini statice care să nu distragă atenția vizitatorului dar să fie atractive.



## **2.6 Proiectarea esteticii aplicației Web**

Estetica site-ului sau **LAYOUT** - ul este esențială în impactul cu utilizatorul. Ea începe cu pagina principală care este "cartea de vizită" și se întinde pe tot cuprinsul site-ului.

*Home* sau *Index* reprezintă locul de unde începe explorarea resurselor Web și se constituie în puncte de start pentru accesarea informațiilor specifice publicității sau companiei pe care o reprezintă. În continuare prezentăm câteva *trăsături caracteristice* necesare a paginii de intrare:[RADO- 96a]:

- ❑ pagina să fie caracteristică;
- ❑ să ofere cât mai multă informație fără a deveni "obeză"; încărcarea ei trebuie să dureze
- ❑ maxim câteva zeci de secunde;
- ❑ să se utilizeze grafică întrețesută astfel vizitatorul are o idee despre ce încarcă și poate hotărî dacă va continua sau nu;
- ❑ să se utilizeze tehnici asociate elementelor grafice audio/video, care permit rezervarea
- ❑ spațiului astfel ca textul să poată fi afișat primul.

Aceste principii generale pot fi extinse la nivelul întregii aplicații Web prin:

- Evitarea supraîncărcării paginii
- Respectarea unui raport 80% -20% între Conținut vs. Navigare și alte caracteristici
- Organizarea în sensul lizibilității maxime de la stânga sus la dreapta jos, conform priorității elementelor
- Gruparea elementelor (conținut, navigare, funcții) conform unor șabloane recunoscute de utilizator
- Evitarea necesității de derulare (scroll) pentru acces la totalitatea informațiilor unui obiect.
- Specificarea dimensiunilor elementelor layout-lui în valori relative la suprafața disponibilă.

Design-ul grafic sau LOOK&FEEL cum este deja cunoscut are câteva direcții specifice:

- layout
- scheme de culori
- fonturi, dimensiuni și stilori
- utilizare de alte media (audio, video, animație)
- alte elemente de estetică.

- ▶ Toate elementele de design prezintă **consistență stilistică** de la o pagină la alta; se recomandă crearea a cel puțin trei șabloane de structură, bazate pe aceleași culori, fonturi, structuri de frame-uri, tipuri de butoane.
- ▶ Elementele de design să fie puse în acord cu audiența, cu posibilele tabu-uri și probleme de accesibilitate; în acest sens fonturile vor fi mai mari dacă se adresează persoanelor în vârstă; exprimarea va fi conformă cu raportul față de audiență, așa cum s-a precizat la subcapitolul de studiu al audienței;
- ▶ Diferența codurilor de culoare între fundal și culoarea textului, bordurilor, etc., trebuie să fie cel puțin 40%; paleta de culori folosită să fie restricționată la Web-safe, dacă nu există obligativitatea de a afișa fotografii.

- ▶ Site-ul nu trebuie să se bazeze pe butoanele de navigare ale browserului, trebuie **să se pună la dispoziție soluții de navigare directă** între oricare două pagini ale site-ului.
- ▶ Se recomandă **să se evite scroll-ul orizontal**; e bine să se evite fontul clasic, TNR de 12 pe fond alb, care dă de multe ori impresia citirii unei foi de hârtie și nu dă impresia că avem de a face cu un site interactiv; pentru asta, se folosesc de obicei fonturi fără serifuri în titluri și fonturi cu serifuri în zonele mari cu text aglomerat, pentru creșterea lizibilității;
- ▶ **Design-ul să fie flexibil** pentru adăugarea în orice moment de noi informații, noi celule în tabele, noi câmpuri în formulare.

**Grafica** este un element de artă care are un rol dublu: de a transmite profismul prin calitatea graficii și de a sugera atmosfera, stilul și personalitatea paginii, “look and feel”.

Ea poate reprezenta doar o idee vizuală pe baza căreia culorile și textul, sloganul, imaginile vor fi folosite într-o anumită ordine sau chiar o animație complexă.

Ea trebuie să fie clară, originală și interesantă. Elementele caracteristice graficii statice sunt: *culorile, fundalul, mărimile imaginilor*, iar celor dinamice li se adaugă *numărul de cadre, timpul, facilități suplimentare de vizualizare*, etc.

La proiectarea unui site de success se realizează o combinație perfectă între design și conținut, acordând o atenție deosebită designului (grafica, sigla, culori, etc.). **Culorile vor influența dispoziția, starea sufletească și emoțiile vizitatorilor, și prezintă identitatea și imaginea brand-ului, ca element de marketing.**



### *Culori calde:*

- *roșu* este una dintre culorile care captează atenția, simbolizează pasiune, foc și dragoste, are ca efect creșterea tensiunii și transmite energie, simbolizând și război, violența, agresiune.
- *galben* reprezintă soarele și căldura, însemnând speranța, lumina și energia sau o stare de slăbiciune, boală, frică și lașitate.
- rozul este romantic, relaxant și feminin iar portocaliu vibrant și cald este asociat cu toamna, este mai puțin inflăcărat decât roșu și stimulează apetitul, fiind culoarea unor citrice, care simbolizează sanatate.

### *Culori reci:*

- *verdele* simbolizează sănătate și creștere (vegetație), reinnoire, regenerare și fertilitate deși poate însemna și lipsa de experiență.
- *albastru* este una dintre cele mai relaxante culori, asociat cu cerul și marea, o culoare sigură ce înseamnă inteligență, calm și încredere sau depresie și tristete.
- *purpuriu* ca și combinație între roșu și albastru este una dintre cele mai intrigante culori și reprezintă creativitate, spiritual și misterios iar purpuriul închis este asociat cu regalitatea și bogatia, nuanțele mai deschise fiind asociate cu romantismul și nostalgia.

### *Culori neutre:*

- *negru* poate reprezenta putere, eleganță, rafinement, dar și moarte în culturile occidentale, rău, furie și tristete.
- *alb* reprezintă curățenia, puritatea și spiritualitatea, viața și casatoria în culturile occidentale dar și moarte în culturile orientale.
- *gri* combinația dintre negru și alb este o culoare conservativă care simbolizează siguranță, maturitate și încredere, dar și tristete.
- *maro* este o culoare specială ce poate fi folosită ca neutră sau caldă, reprezentând credibilitate, stabilitate, inima, casa, pământul.

*Fundalul (background-ul)* unor site-uri pe Internet prezintă imagini de fond care fac aproape imposibilă citirea textului sau sunt complet inestetice.

Specialiștii arată că cel mai ușor se citește negru pe alb și de aceea se recomandă folosirea unor imagini de fond foarte pale fără artificii inutile.

Un site ce prezintă o galerie de artă va utiliza un fundal neutru pentru a pune în valoare tablourile, în schimb o agenție de turism va putea folosi un fundal adecvat care să atragă vizitatorii.

**Stil și personalitate** pot fi caracteristicile de bază a unei pagină web iar la crearea ei trebuie să concure toate elementele site-ului respectiv: grafica, fondul, textul, culoarea, elemente interactive, și nu un ultimul rând cuvintele, expresiile folosite în conținut, care vor reprezenta astfel primele elemente de **e-marketing** “at the first site” cu rol de cunoaștere și influențare a consumatorilor.

Recomandările legate de soluționarea acestor principii pot fi sintetizate astfel:

- maparea site-ului în fiecare pagină pentru orientarea vizitatorului în fiecare moment și să se ofere în fiecare pagină o legătură către home-page (prima pagină);
- claritatea și conciziunea documentelor;
- organizarea textului astfel încât cititorii să poată scana informațiile importante;
- conținutul paginilor să fie desine stătător;
- să nu se folosească pentru marcarea unor pasaje de text formatările rezervate titlurilor;
- informația să fie grupată atât semantic cât și vizual prin utilizarea de titluri/ subtitluri și separarea secțiunilor;

- să se păstreze spiritul publicațiilor în toate paginile;
- să se utilizeze legături descriptive și meniuri de legături pentru scanare rapidă, comentate și grupate în meniuri, pentru o parcurgere mai ușoară; o legătură este necesară numai dacă există un motiv întemeiat, evitând trimerile către documente redunante și fără multe legături la același site în aceeași pagină;
- în fiecare document să se furnizeze informații pentru contact și o referință către persoanele care pot da relații suplimentare;
- să se furnizeze versiuni text pentru documente liniare;
- să se utilizeze marcarea textului (*I*, **B**, U) și ancorele cu discernământ;
- să se evite redundanța în îndrumări;
- să se corecteze documentul atât din punct de vedere lexical cât și gramatical;
- să se folosească un design simplu, fără grafică ornamentală inutilă;
- să se evite stridența paginilor (text prea colorat, fundal țipător); acestea devin vulgare și obositoare.

În raport cu scopul documentului și structura aleasă, proiectantul va decide asupra structurii elementelor din pagină și a legăturilor între documente, deși putem puncta câteva posibilități generale de realizare a facilităților de bază:

- ▶ Utilizarea titlurilor descriptive, care furnizează relațiile cu celelalte pagini elaborate;
- ▶ Includerea unei legături înapoi (**Back**) pentru orice document care depinde de cel precedent;
- ▶ Evitarea formulărilor de tip: "Puteți studia această problemă cu...", sau "Avantajele acestei metode sunt:..." care pot fi regăsite cu ușurință în alte pagini, structurate și detaliate cu discernământ;
- ▶ Un paragraf plin de texte boldate și înclinate este greu de urmărit
- ▶ Se recomandă evitarea utilizării terminologiei browserelor deoarece fiecare browser are set diferit de meniuri ce gestionează documentele și metode diferite de deplasare înapoi-înainte, motiv pentru care legăturile similare se aranjează și se gestionează direct în document.

**Șabloanele de proiectare** specifice aplicațiilor Web destinate elementelor hypermedia sunt grupate în diverse categorii:

- ▶ Arhitecturale
- ▶ Construire componente
- ▶ Navigare
- ▶ Prezentare
- ▶ Comportament / interacțiune utilizator



Din multitudinea de soluții open-source amintim doar câteva adrese de șabloane:

- ▶ <http://www.designpattern.lu.unisi.ch> - repository cu șabloane
- ▶ <http://www.visi.com/~snowfall/InteractionPatterns.html>
- ▶ <http://www.welie.com/patterns>
- ▶ <http://www8.org/w8-papers/5b-hypertext-media/improving/improving.html>  
- navigare
- ▶ <http://www.anamorph.com/docs/patterns/index.html> - un limbaj de șabloane
- ▶ [http://www.mit.edu/~jtidwell/interaction\\_patterns.html](http://www.mit.edu/~jtidwell/interaction_patterns.html) - limbaj de șabloane specific Human Computer Interaction (HCI).

### 3. Proiectarea arhitecturală a unei e-afaceri

O afacere pe Internet (e-business) are o complexitate crescută și presupune activități specifice de proiectare pe care le vom detalia în capitolele următoare. Precizăm că în acest caz se stabilește:

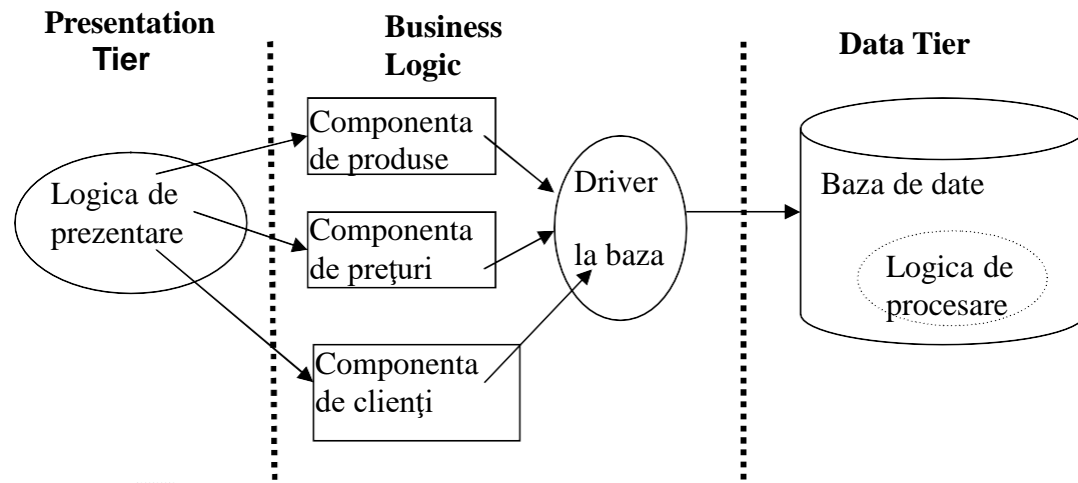
- ▶ **Arhitectura** structurată pe nivele a aplicației (two-tier, three-tier, multi-tier);
- ▶ **Tehnicile și tehnologiile** utilizate în realizarea aplicației (triada PHP, SQL server cu componente, Oracle și interfețe specifice, etc) Se aleg tehnologiile de implementare a aplicațiilor: software-ul server de gestiune a bazelor de date, utilitarele de monitorizare a acceselor, limbajele server-side și client-side:

- ▶ **Interfața cu clientul** reprezentată de datele ce se oferă utilizatorilor și datele care se prelevează prin intermediul formularelor. Informația nestructurată: textul și imaginile care vor apărea în paginile site-ului și care vor fi incluse la nivelul formatării; informația nestructurată este unidirecțională și corespunde cerinței de informare generală a site-ului;
- ▶ Soluția middleware adoptată;
- ▶ **Structurile de date** care se grupează în informație structurată: care va fi stocată într-o bază de date, pe server; respectiva bază de date este construită conceptual, prin specificarea atributelor și tipurilor acestora;
  - informația structurată este bidirecțională, bazele de date fiind actualizate în urma prelucrărilor datelor din formulare și citite prin acces concurențial;
  - datele citite trebuie integrate, de regulă prin tabele, în conținutul nestructurat;
- ▶ datele scrise nu sunt, de regulă, accesate concurențial: proprietarii site-urilor cu audiență publică preferă să nu ofere posibilitatea utilizatorilor să modifice date existente pe server, ci doar să le permită să adauge noi înregistrări, urmând ca gestiunea concurenței să fie arbitrată la nivel de server, uneori chiar de către persoane fizice.

- ▶ **Aplicațiile de prelucrare** sunt necesare preluării, manipulării și regăsirii acelor date, precum și *fluxul prin care datele vor ajunge de la utilizator la server*:
- ▶ Datele pot fi preluate din formular HTML sau din zonele de text ale unei prezentări Shockwave, Flash, pot fi validate sau transformate la nivel de client, de către un script JavaScript, PHP sau de către limbajul de action-scripting suportat de prezentarea Shockwave.
- ▶ Rezultatele cererilor pot să se întoarcă în pagină (*Hypermedia interactivă, design logic*) sau pot fi transferate prin HTTP, folosind metoda POST sau GET, pot fi preluate de server de către un script CGI, un servlet Java, apoi ele pot fi stocate în baza de date sau se pot returna răspunsuri sub formă de date structurate (preluate din baza de date) sau informație slab structurată (cod HTML/XML ca ieșire a scriptului server-side).

- ▶ Alegerea unei tehnologii server-side trebuie să țină cont de *scalabilitatea prelucrării*, și să asigure accesul concurențial. Un applet Java este un consumator de resurse dar servleturile Java sunt proiectate multithreading și asigură performanță. Pentru un site de complexitate medie limbajele interpretative Perl, PHP, ASP sau fișiere de comenzi Unix asigură funcționalitate și minimizează timp de răspuns.
- ▶ Dacă site-ul devine o aplicație Web, realizarea site-ului urmează o abordare *front-end*, pornind de la cerințele audienței, *middleware*-ul (programele agent care preiau cereri utilizator și gestionează pe baza lor datele) și care trebuie să determine *back-end*-ul (datele).
- ▶ Alegerea unei tehnologii server-side trebuie să țină cont de *scalabilitatea prelucrării*, și să asigure accesul concurențial. Un applet Java este un consumator de resurse dar servleturile Java sunt proiectate multithreading și asigură performanță. Pentru un site de complexitate medie limbajele interpretative Perl, PHP, ASP, Python, .NET asigură funcționalitate și minimizează timp de răspuns.

► Fig 5 - Arhitectura three tier a aplicatiilor

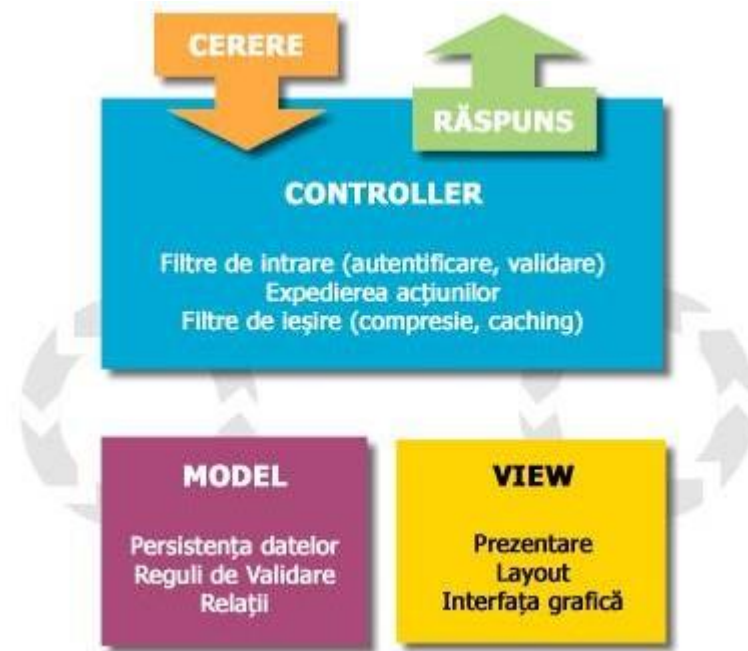


- ▶ Soluția 3-tier este ușor de implementat, deoarece server gateway poate fi creat ca aplicație Java standalone cu funcții client native suprascrise în clase Java. Acestea asigură comunicația ca server cu un applet Java în protocoale definite utilizator, la un capăt, în timp ce accesează serverul de baze de date la celălalt capăt, folosind protocoale native client/server. Scripturile pot fi elaborate în : Jscript, Php, Asp, Flash, Adobe Flex, etc.
- ▶ Pentru serviciile bazate pe web, pentru a avea cea mai largă utilizare a devenit din ce în ce mai important ca aceste servicii să fie accesibile unui număr cât mai mare de clienți, în acest sens s-a observat o proliferare a unor tipuri diferite de clienți pentru aplicațiile bazate pe web și odată cu dezvoltarea tranzacțiilor B2B, serverele care utilizează XML pot fi de asemenea clienți.

- ▶ Deseori este preferată arhitectura **model-view-controller (MVC)**. Această arhitectură izolează logica de funcționalitate (business logic) de interfața cu utilizatorul rezultând o aplicație caracterizată de ușurința modificării fie a aspectului vizual, fie a aspectului de funcționalitate, fără ca celălalt aspect să fie afectat.
- ▶ Modelul reprezintă informația (datele) și regulile folosite pentru a manipula datele, View reprezintă elementele de interfață cu utilizatorul, iar Controller implică comunicarea acțiunilor utilizatorului către Model.
- ▶ Arhitectura MVC este des întâlnită în aplicații web, în care View reprezintă pagina HTML, Controller-ul este codul care colectează date dinamice și generează conținutul din pagina HTML. Elementul Controller face legătura între Model și View prin analizarea, validarea și direcționarea cererilor provenite de la utilizator către logica business. El gestionează răspunsul primit de la Model și invocă elementele corespunzătoare ale elementului View pentru a reda conținutul semnificativ către utilizator



Fig 6. Arhitectura MVC



- ▶ Modelul reprezintă conținutul informațional (de obicei stocat în baze de date sau noduri XML) și logica business care transformă conținutul în funcție de acțiunile utilizatorilor.
- ▶ Avantajele acestui sistem arhitectural sunt reducerea complexității în design-ul arhitectural, creșterea flexibilității și a reutilizării aplicației.
- ▶ În proiectarea de detaliu a aplicației se pot folosi tehnologii precum Php și HTML pentru partea de server, Javascript, CSS și Ajax pentru partea de client (user interface) și MySQL pentru partea de date.
- ▶ Pentru a descrie promovarea aplicației se vor prezenta strategii și metode de internet marketing, cu exemple de campanii.
- ▶

#### **4. Testarea și înregistrarea site-ului**

Validarea automată a aplicațiilor presupune utilizarea unor unelte specializate, puse la dispoziție ca și soluții open source astfel:

- ❑ Validarea HTML/XHTML
- ❑ Validare CSS: <http://jigsaw.w3.org/css-validator/>
- ❑ Validare RSS <http://validator.w3.org/feed/>
- ❑ Link-uri “orfane” sau defecte: <http://validator.w3.org/checklink>

Cea mai des utilizată este oferta de validare Google: <https://validator.w3.org/>

Acest validator verifică validitatea de marcă a documentelor Web în HTML, XHTML, SMIL, MathML, etc. Dacă se validează conținut specific, cum ar fi RSS / Atom feed sau foi de stil CSS, conținutul MobileOK, sau pentru a găsi link-uri rupte, există alte validatoare și instrumentele specifice puse la dispoziție de Google.

O alternativă poate fi validator non- DTD-based. Opțiuni puse la dispoziția utilizatorilor:

- ❑ Validate by URI
- ❑ Validate by File Upload
- ❑ Validate by Direct Input

Verificarea erorilor care apar în timpul proiectării paginilor web poate fi realizată printr-un serviciu de validare HTML cu un Document Type Definition (**DOCTYPE**) ca primă linie pentru pagina HTML .

Pentru a valida documentele în XHTML trebuie să adăugăm la începutul documentului tagul DOCTYPE cu specificațiile corespunzătoare tipului de document (Strict, Transitional sau Frameset). Un exemplu de DOCTYPE :

```
!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd";
```

Aceasta specifică sintaxa (structura gramaticală) a paginii Web și se obține o listă cu mai multe erori de sintaxă HTML/XHTML sau XML care trebuie analizate și remediate.

Servicii de validare HTML/XHTML sau XML și testarea paginilor web verifică:

- structura documentelor;
- erorile de sintaxă;
- optimizează imaginile;
- legaturile interne și externe;

Pentru a putea publica site-urile Web este necesar transferul fișierelor pe serverul Web denumit *upload*. Această operație se realizează cu servicii FTP care permit:

- *upload* al fișierelor unui site la o gazdă web;
- Înlocuirea unui fișier sau o imagine;
- *download* se descarcă un fișier de pe un alt calculator;
- accesul unei anumite persoane pentru a încarca/descărca un fișier din site-ul propriu.

- ▶ Pentru conectarea la un server FTP este necesar *un program client* FTP instalat pe calculatorul sursă și un *URL* sau *adresa IP* a gazdei FTP obținută la înscriere.
- ▶ Odată conectat prin FTP, se pot primi sau transmite fișiere iar pentru a obține anumite fișiere/foldere de pe serverul gazdă se poate alege opțiunea ***download*** respectiv ***upload*** care asigură transferul dintr-un/într-un anumit director pe/de pe calculatorul personal.
- ▶ Prin introducerea loginului și parolei obținute la înregistrarea la calculatorul gazdă, se obține acces la un cont care dă dreptul la accesarea unui anume director pentru a transfera fișiere.
- ▶ Conectarea cu loginul anonymous se folosește numai pentru download de fișiere, datorită riscurilor de securitate care pot apare.

Pentru încărcarea pe server se urmează pașii:

- se pun toate fișierele corespunzătoare paginilor siteului care urmează să fie încărcat, într-o singură mapă (sau într-un singur director) pe hard-disk-ul corespunzător serverului respectiv;
- în cadrul directorului, se denumește fișierul corespunzător paginii gazdă cu numele *index.html* - acesta este fișierul care este încărcat pentru pagina gazdă de către majoritatea sistemelor software ale serverelor Web; pentru unele servere, acest fișier va trebui să poarte numele *default.html*;
- dacă se utilizează serverul unui provider: trebuie aflate eventualele convenții de denumire a fișierelor, care urmează a fi respectate iar fișierele corespunzătoare paginilor Web se transferă prin FTP, protocolul de transfer de fișiere Zmodem sau prin intermediul unui alt protocol de transfer electronic de fișiere.

Sunt puse la dispoziție câteva posibilități pentru publicarea sitului Web:

- ▶ Internet Service Provider (ISP) - ex: UPC, Romtelecom, RCS&RDS
- ▶ Gazdă gratuită pe Internet (*free host*) - ex: 50.ro
- ▶ Propriul server Web, instalat sub forma unei aplicații pe calculatorul personal - ex: serverul *Apache* , WampServer (care poate fi descărcat gratuit de la adresa <http://www.wampserver.com/en/> ) sau EasyPHP ([www.easyphp.org](http://www.easyphp.org))\

Fiecare beneficiar optează pentru una sau alta funcție de obiectivele aplicației Web, posibilitățile de întreținere cu personal propriu sau specializat și nu în ultimul rând bugetul alocat pentru această afacere.



## **5. Intreținerea și actualizarea site-ului**

Pentru ca site-ul creat să aibă succes și să atraga noi vizitatori, trebuie întreținut și actualizat periodic. Pe baza statisticilor s-a constatat ca o pagina web neactualizată este interesantă după o perioadă medie de 60 de zile. De aceea, pentru a menține interesul vizitatorilor, un site web nu trebuie să fie static. Dacă nu sunt oferite informații noi sau resurse utile vizitatorilor, ei nu vor avea nici un motiv să mai revină iar site-ul își va pierde popularitatea.

Intreținerea și actualizarea unui site web se realizează cu diferite scopuri:

- îmbunătățirea proiectării și machetării;
- actualizarea sau adăugarea unor informații suplimentare;
- realizarea unor versiuni multi-lingve;
- ca soluție a feedback-ului, pentru a răspunde la comentariilor, cerințelor sau observațiilor vizitatorilor;
- pentru a corecta erorile și problemele funcționale apărute în timpul proiectării.

**Întreținerea** eficientă a site-ului, se poate realiza pe baza unui plan de întreținere, etapizat în timp, care poate să conțină:

- frecvența de actualizare;
- îmbunătățirile sau modificările aduse;
- informații și resurse de actualitate;
- comentariile sau recomandările vizitatorilor;
- poziția în clasamentul motoarelor de căutare;
- îmbunătățirea configurației hardware și resursele software suplimentare.

**Actualizarea** site-ului după ce a devenit operational este o sarcină obligatorie ce presupune:

- îmbunătățirea unor caracteristici tehnico-funcționale;
- adăugarea de noi funcționalități;
- actualizarea conținutului informațional;

- gestionarea formelor de feedback pentru vizitatori;
- legături și resurse utile, sfaturi sau informații recente;
- adăugarea de funcționalități noi: blog;
- prezența în rețelele sociale *Facebook, Twitter, LinkedIn, YouTube*
- extinderea spre noi dimensiuni ale afacerii.





















































