

METODOLOGII DE PROIECTARE A APLICAȚIILOR WEB

Elena Jitaru

jitaru@ici.ro

Institutul Național de Cercetare – Dezvoltare în Informatică, ICI, București

Rezumat: Lucrarea prezintă principalele metodologii pentru proiectarea aplicațiilor Web și propune a nouă metodologie dezvoltată în cadrul proiectului CEEEX Pro-Inclusiv „Metode și instrumente pentru crearea de interfețe multimodale, necesare dezvoltării sistemelor informatice inclusive, cu grad ridicat de accesibilitate”.

Cuvinte cheie: metodologie, proiectare aplicații Web, accesibilitate

Abstract: The paper presents the main methodologies for designing web applications and proposes a methodology developed in the CEEEX project Pro-Inclusiv „Methods and tools to create multimodal interfaces for the development of inclusive information systems, with high degree of accessibility”.

Keywords: methodology, designing Web applications, accessibility

1. Abordări metodologice

Ingineria Web este un domeniu nou al ingineriei software, care pune accent pe dezvoltarea sistemelor Web. Deși utilizarea unor metodologii de proiectare nu este încă o practică uzuală în domeniul ingineriei Web, necesitatea unor abordări sistematice ce trebuie adoptate atunci când se dezvoltă sisteme complexe este astăzi larg acceptată. Dezvoltarea aplicațiilor Web are câteva caracteristici care diferă de dezvoltarea altor tipuri de aplicații. Pe de o parte, în procesul de dezvoltare participă diferite tipuri de actori (stakeholders): analiști, utilizatori, designeri, experți în marketing, multimedia și securitate etc. Pe de altă parte, principalele caracteristici ale acestor sisteme sunt structura de navigare, interfața cu utilizatorul și capacitatea de personalizare. Structura necesită un ghid intuitiv pentru a evita ca utilizatorul să se piardă în spațiul de navigare. Proiectarea interfeței cu utilizatorul trebuie să ia adesea în calcul aspecte multimedia și de marketing.

În România, există foarte multă literatură de specialitate care abordează aspecte ale proiectării interfeței cu utilizatorul și interfeței grafice a siturilor Web, dar mai puțină despre metodologiile de proiectare a acestora. De aceea am considerat necesară o trecere în revistă a acestor abordări.

În ultimii ani, au fost propuse câteva abordări pentru dezvoltarea sistemelor Web. Aceste metode oferă elemente de modelare specifice pentru analiză și proiectare, multe dintre ele definesc o notație proprie utilizată pentru reprezentarea grafică a elementelor. Aproape toate propun procese specifice pentru a sprijini dezvoltarea sistematică sau semi-automată a aplicațiilor Web. Escalona și Koch (2004) arată că doar câteva dintre metodologiile Web existente (NDT – Navigational Development Techniques, OOHDMM – Object-Oriented Hypermedi Design Method, UWE – UML-based Web Engineering, W2000) încep ciclul de dezvoltare cu o analiză detaliată a cerințelor. Alte abordări analizate de autori propun tehnici clasice pentru analiza cerințelor Web sau ignora această fază a procesului de dezvoltare. Analiza cerințelor este considerată de toate abordările ingineriei software ca fiind un pas cheie în dezvoltarea unor sisteme software de succes. Efortul investit în analiza cerințelor reduce în mod considerabil deficiențele în fazele ulterioare ale dezvoltării. Specificarea cerințelor constă în realizarea unei descrieri a cerințelor utilizând diferite tehnici: de la descrierea textuală informală la specificația formală în limbaje precum limbajul Z. Escalona și Koch (2004) au dezvoltat un metamodel comun pentru reprezentarea conceptelor și relațiilor ingineriei cerințelor Web (Web requirements engineering (WebRE)).

Metodologiile de proiectare a aplicațiilor Web disponibile în literatura de specialitate și care vor fi prezentate pe scurt în paginile următoare pot fi clasificate conform părerii lui Distant et al. (2007) după cum pun accent pe “ce” se cere să facă aplicația (proiectare conceptuală în domeniul problemei) sau pe “cum” poate satisface aplicația cerințele și cum poate implementa ceea “ce” se cere să facă aplicația (proiectarea logică în domeniul soluțiilor). UWA- Ubiquitous Web Application, OOHDMM, OOWS - Object-Oriented Web Solution și OO-H - Object-Oriented Hypermedia Method se încadrează în prima categorie, în timp ce UWE și WebML - Web Modeling Language pot fi considerate ca fiind metodologii hibride, deoarece acoperă atât proiectarea conceptuală cât și proiectarea logică.

Metodologiile de proiectare conceptuală fac abstracție de detaliile de implementare și oferă o privire de ansamblu asupra sistemului, din punctul de vedere al utilizatorilor. Modelarea conceptuală este punctul de plecare pentru implementarea sistemelor complexe. Totuși, distanța mare între modelul conceptual al unei aplicații Web și implementarea sa face ca utilizarea proiectării conceptuale să fie insuficientă pentru dezvoltarea unei aplicații Web. Dacă nu există niveluri de proiectare intermediară pentru translatarea specificațiilor conceptuale în proiectarea implementării, atunci activitățile de implementare a unei aplicații Web pot să aibă loc independent de proiectarea conceptuală, ceea ce ar conduce la o risipire a efortului. Tendința unor cunoscute metodologii de proiectare este de a se dezvolta spre abordări integrate, care să acopere întreg ciclul de viață al aplicațiilor Web. Printre metodologiile care urmează această direcție se număra OO-H, OOWS, UWE și OOHDM.

WSDM - Web Site Design Method („WiSDoM”) este o abordare centrată pe utilizator care modelează aplicația pe baza cerințelor informaționale ale grupurilor de utilizatori. Această abordare permite realizarea de situri Web mai bine adaptate necesităților vizitatorilor și prin aceasta având un mai mare grad de utilizabilitate și de satisfacție (De Troyer, 1998). Procesul de dezvoltare este împărțit în patru faze:

1. *Modelarea audienței*, în care utilizatorii sunt clasificați și grupați în vederea studierii cerințelor sistemului potrivit fiecărui grup de utilizatori;
2. *Proiectarea conceptuală*, în care este proiectată o diagramă clasă pentru a reprezenta modelul static al sistemului și un model de navigare, pentru a reprezenta posibilitățile de navigare;
3. *Proiectarea implementării*, în care modelele proiectării conceptuale sunt translatare într-un limbaj abstract, ușor de înțeles de către calculator;
4. *Implementarea*, în care rezultatul proiectării implementării este scris într-un limbaj calculator specific.

Faza de modelare a utilizatorului își propune să identifice diferitele roluri ale utilizatorilor în cadrul a două sub-faze:

Clasificarea utilizatorilor, adică identificarea potențialilor utilizatori/vizitatori ai sitului 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.

Celelalte faze din procesul de dezvoltare WSDM se bazează pe clasificarea utilizatorilor realizată în prima fază. Se face o distincție clară între proiectarea conceptuală (care nu conține nici un detaliu de implementare) și proiectarea prezentării, care ia în considerație limbajul de implementare folosit. Această separare este similară distincției făcute în proiectarea bazelor de date între schema conceptuală și schema logică.

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

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.

În **W3DT - World Wide Web Design Technique** procesul de dezvoltare cuprinde șapte pași, ghidând proiectantul de la analiza cerințelor la implementarea sistemelor informatice bazate pe Web. Pentru prima fază este furnizată o colecție de metode pentru a determina cerințele utilizatorilor. Faza de proiectare este împărțită în: structurarea informațiilor, proiectarea navigării, proiectarea organizațională și proiectarea interfeței. Structurarea informațiilor și proiectarea navigării sunt înțelese ca un proces iterativ, în care un domeniu larg de cunoștințe trebuie organizat și structurat în obiecte informaționale, care apoi sunt legate și evaluate. În cadrul fazei de implementare, obiectele informaționale sunt implementate prin pagini HTML și scripturi, iar structurile de acces al modelului sunt implementate prin legături în cadrul sitului Web. În metamodelul W3DT, un sistem informatic bazat pe Web este compus din pagini, legături și machete. Sunt diferențiate două tipuri diferite de legături și patru tipuri diferite de pagini.

SOHDM - Scenario-based Object-Oriented Hypermedia Design Methodology a fost prima abordare care pune accent pe importanța unui proces care permite analiștilor să “capteze” și să definească

cerințele aplicațiilor. SOHDM are similarități cu metodologia OOHDM, dar propune o specificare a cerințelor bazată pe scenarii. Ciclul de viață SOHDM cuprinde șase etape:

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

Definirea cerințelor începe cu proiectarea unei așa-zise diagrame de context, similară diagramei fluxului de date (DFD) definită de Yourdon (1989). Pentru a construi o astfel de diagramă de context, analistul trebuie să identifice entitățile externe care comunică și 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. Scenariile sunt reprezentate grafic 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 analiză. Fazele RNA sunt:

- *Analiza mediului*, având obiectivul de a analiza caracteristicile audienței. Sunt identificați 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*, în care sunt identificate 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/ caracteristicile de navigare;
- *Analiza implementării*, în care este identificat modul în care modelele descrise în faza anterioară vor fi implementate în limbajul calculatorului.

RNA oferă doar câteva ghiduri de acțiune pentru fiecare fază. Nu sunt propuse concepte de modelare sau o notație proprie, dar RNA este una dintre metodologiile care au pus accent pe importanța specificării cerințelor în procesul de dezvoltare al aplicațiilor Web. RNA subliniază necesitatea separării analizei cerințelor conceptuale de analiza cerințelor de navigare.

HFPM: Hypermedia Flexible Process Modeling prezentat de Olsina (1998) 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. Astfel, HFPM 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. Astfel, pentru „modelarea cerințelor” sunt definite următoarele task-uri:

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

Faza de „Dezvoltare” are definite următoarele task-uri:

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

OOHDM – Object-Oriented Hypermedi 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 putem găsi „context de navigare”. Procesul OOHDM este împărțit în patru faze:

- *Modelul conceptual*, reprezentat ca un model clasă, este 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 anterioare.

OOHDM a fost extins ulterior cu o tehnică specială ce se ocupă de interacțiunea cu utilizatorul în faza de analiză a cerințelor. Tehnica folosită se numește User Interaction Diagram (UID) și este construită pentru interacțiunea specială a utilizatorului Web cu sistemul Web (Vilain et al., 2000).

Diagramele UID se bazează pe tehnica binecunoscută a cazurilor de utilizare. Cazurile de utilizare sunt folosite pentru a „capta” cerințele, dar sunt considerate în OOHDM destul de ambigue și insuficiente pentru definirea cerințelor aplicațiilor Web, în special în ceea ce privește interacțiunea dintre utilizator și sistem. De aceea această metodologie sugerează pentru specificarea cerințelor rafinarea cazurilor de utilizare construind diagrame UID, care sunt folosite pentru a modela grafic interacțiunea dintre utilizator și sistem fără a lua în considerație aspectele specifice ale interfeței.

UWE – UML-based Web Engineering este o metodologie pentru Web a ingineriei software orientată pe model care își propune să acopere întregul ciclu de viață al dezvoltării aplicațiilor Web, punând accent pe aplicațiile adaptive. UWE oferă: o notație bazată pe UML (Unified Modeling Language), 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, construind 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 implementarea cazului de utilizare. Aceasta din urmă poate fi descrisă textual sau modelată prin diagramele UML de activitate. UWE clasifică cerințele în două grupuri: funcționale și ne-funcționale. Cerințele funcționale sunt:

- Cerințe privind conținutul;
- Cerințe privind structura;
- Cerințe privind prezentarea;
- Cerințe privind adaptarea;
- Cerințe privind modelul utilizator.

UWE propune interviurile, chestionarele și listele de verificare (checklists) ca fiind tehnici potrivite pentru captarea cerințelor și cazurile de utilizare, scenariile și glosarele, pentru specificarea cerințelor. Pentru a le valida, UWE propune parcurgerile (walk-through), auditurile și prototipurile.

W2000 (Baresi et al., 2001) este o metodologie orientată obiect derivată din HDM, care urmează principiul SoC 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. Perrone et al. (2004) arată ca W2000 organizează proiectarea unei aplicații hipermedia complexe în jurul a patru dimensiuni:

- *Informația*, definind unitățile (entitățile) informaționale conceptuale de bază percepute de utilizator și diferitele structuri (colecții) de acces permițând utilizatorului să le acceseze pe baza necesităților sale;
- *Navigarea*, definind structurile de navigare în 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.

WebML - Web Modeling Language este un sistem de notații pentru specificarea vizuală a siturilor complexe la nivel conceptual (Ceri et al., 2000). Spre deosebire de modelarea datelor, care este o activitate consolidată, modelarea hipertext este o disciplină mai tânără căreia îi lipsește o bază organizată de concepte, notații și metode de proiectare.

Limbajul WebML oferă instrumente și notații elementare pentru modelarea hipertext. Ca bază se ia modelul Entitate-Relație ce folosește concepte simple și expresive pentru specificarea modelelor exprimate prin notații grafice intuitive. Modelul propus în WebML trebuie înțeles ca 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 sitului (engl. site views). Unitățile sunt fragmente atomice de conținut, ce oferă alternative de combinare dinamică a conținutului extras din entități și relații al schemei de date. Totodată, unitățile permit specificarea formelor de introducere a datelor de către utilizatori. Din unități sunt formate paginile, care sunt elementele de interfață oferite utilizatorilor. Paginile și unitățile sunt conectate într-o structură hipertext. Legăturile reprezintă esența modelării hipertext: ele exprimă posibilitatea de navigare și transmitere de parametri de la o unitate la alta, necesari pentru generarea conținutului unei pagini.

Specificarea sitului în WebML constă din 4 modele:

- *modelul structural*, care exprimă conținutul sitului sub formă de entități și relații;
- *modelul hipertext*, care descrie unul sau mai multe hipertexte care pot fi publicate pe sit. Fiecare hipertext diferit definește o așa-zisă „vedere a site-ului” (site view).

Descrierile vederilor sitului 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 face ca modelarea Web să fie naturală și pentru personalul non-tehnic. În același timp, WebML este un limbaj formal care poate fi utilizat eficient pentru generarea unei implementări a sitului într-un cadru arhitectural specific. WebML poate fi privit și ca un profil UML pentru dezvoltatorii de aplicații Web. Conceptele WebML pot fi în mod natural exprimate folosind diagramele de clasa UML și pot fi utilizate în conjuncție cu modelarea tradițională orientată obiect UML pentru a extinde analiza dirijată de model.

WebML a stat la baza programului WebRatio (www.webratio.com), al cărui succes comercial demonstrează importanța modelării conceptuale a siturilor web. Accentul în WebRatio se pune pe modelare și reutilizarea implementărilor precedente, care sunt generate în mod automat sau semiautomat din modele de nivel înalt. Valoarea abordării propuse nu constă în componentele individuale ale sistemului, ci în definiția unei carcase bine sistematizate în cadrul căreia elaborarea aplicațiilor Web poate fi organizată în concordanță cu principiile fundamentale ale ingineriei soft și toate sarcinile sunt îndeplinite conform conceptelor, notațiilor și metodelor adecvate.

NDT – Navigational Development Techniques este o tehnică utilizată pentru specificarea și analiza aspectelor privind navigarea în aplicațiile Web. După Escalona et al. (2004) NDT pune accent pe tehnici de elicitare și specificare selectate de NDT pentru captarea și definirea cerințelor. Fluxul de lucru al analizei cerințelor în NDT începe captând cerințele și studiind mediul prin aplicarea unor tehnici precum interviul, brainstorming-ul și JAD. 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 numit BNL (Bounden Natural Language). Prototipurile de vizualizare sunt utilizate pentru a reprezenta sistemul de navigare, vizualizarea datelor și interacțiunea utilizatorului. Întregul proces 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 (UWA Consortium, 2002) oferă o metodologie și un set de modele și instrumente pentru proiectarea conceptuală, centrată pe utilizator, a aplicațiilor Web ubicue. Comparată cu alte metodologii de proiectare Web, UWA se potrivește în special pentru proiectarea aplicațiilor care intenționează să fie ubicue (accesibile diferitelor tipuri de utilizatori, în diferite contexte de utilizare și cu diferite scopuri) și centrate pe utilizator.

Întreaga problemă a modelării este împărțită în următoarele aspecte 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, pentru a ajuta proiectantul să exploateze conceptele și să înțeleagă negocierea 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 metamodelle și notații și subliniază interdependența lor reciprocă și un mediu software unificat, bazat pe Rational Rose, care integrează instrumente specifice fiecărei activități de modelare.

2. Analiza abordărilor actuale

Aceste metodologii propun construirea unor modele diferite, dar care cuprind cel puțin un model conceptual, un model de navigare și un model de prezentare, chiar dacă acestea poartă și alte denumiri. Fiecare model este construit pe un set de elemente de modelare, cum sunt nodurile și legăturile pentru modelul de navigare, sau imaginea și ancora pentru modelul de prezentare. În plus, toate aceste metodologii definesc sau aleg o notație pentru construcțiile pe care le definesc.

Beneficiile modelării conceptuale rezultă din crearea stabilității: adesea conținutul se schimbă, dar structura generală a paginii rămâne neschimbată. Alteori, șablonul se schimbă, dar conținutul rămâne stabil. De aceea pentru modelarea conceptuală a sistemelor informatice bazate pe Web conținutul trebuie separat de structura de navigare, structura de navigare trebuie separată de machetă (de exemplu, structura generală a paginii), iar macheta trebuie separată de conținut.

Distante et al. (2007) propun un cadru de lucru general pentru dezvoltarea aplicațiilor Web. În ciuda diferențelor dintre diferite metodologii pentru aplicațiile de inginerie Web găsite în literatura de specialitate, poate fi identificat un set comun de caracteristici și pași de dezvoltare. Figura 3-1 reprezintă grafic aceste caracteristici comune. După Distante et al. (2007), orice metodologie de proiectare modelează aplicațiile Web pe trei niveluri de construcție (conținut, navigare și prezentare) și adresează un număr de aspecte mergând de la structură la comportament. Aspectele sunt ortogonale pe toate nivelurile de construcție. Structura se aplica atât la conținut (cum este structurat conținutul) cât și la navigare (structuri de tip hipertext) și prezentare (de exemplu, organizarea paginii). În mod similar, comportamentul (de exemplu regulile de business și operațiile de utilizare) pot fi asociate la conținut, navigare și prezentare.

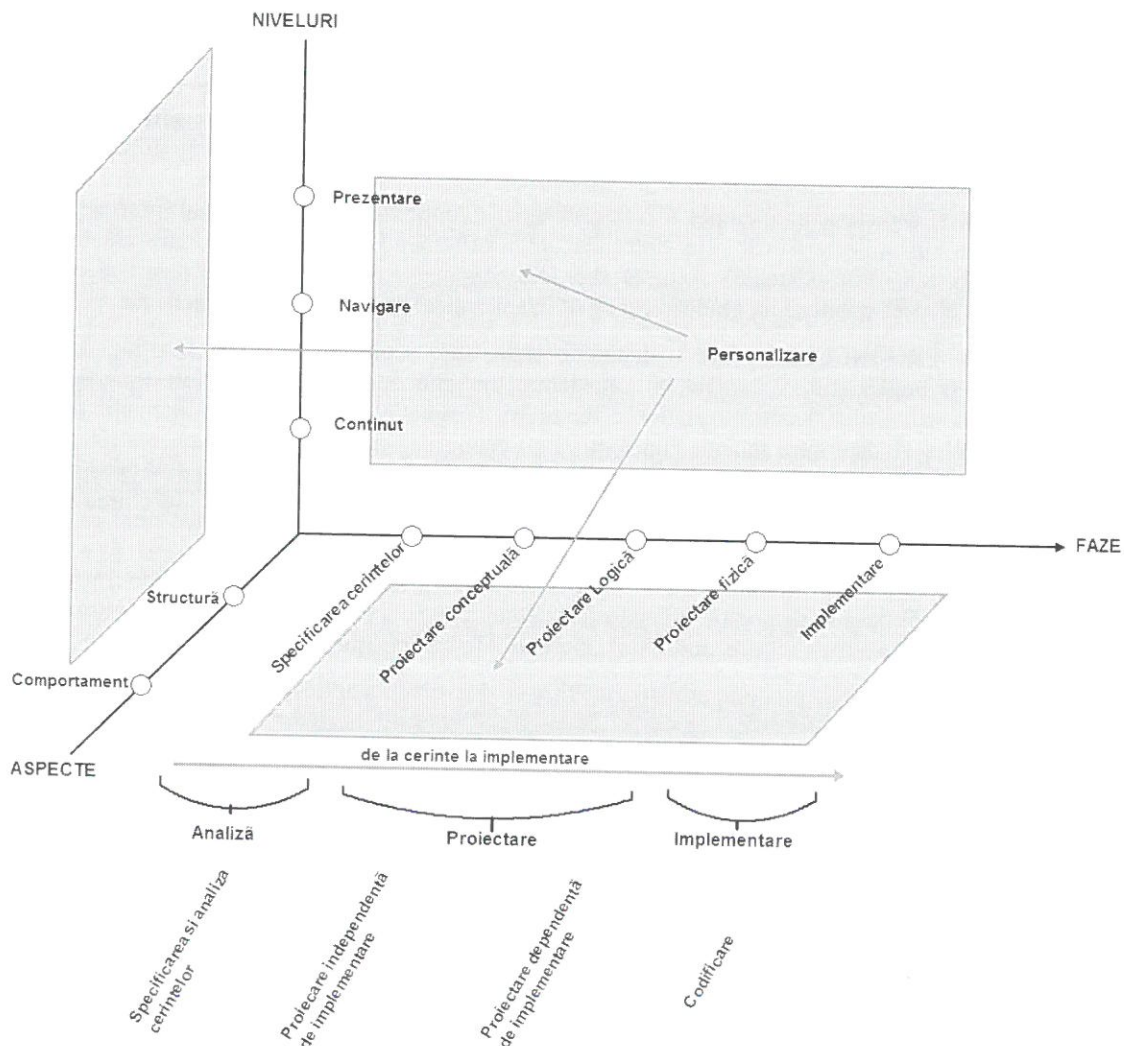


Figura 4-1. Cadru de lucru general pentru dezvoltarea aplicațiilor Web

După cum se vede din figura 4-1, procesul de dezvoltare începe cu specificarea și analiza cerințelor și trece printr-un număr de pași de proiectare spre faza de implementare. Pașii de proiectare includ:

- proiectarea conceptuală, cu accent pe descrierea domeniului problemei și ce se așteaptă să facă sistemul, independent de orice detaliu tehnologic;
- proiectarea logică, cu accent pe operarea sistemului, ascunzând detaliile de implementare specifice unei anumite platforme;
- proiectarea fizică, care adaptează modelul logic al aplicației pentru a obține specificațiile detaliate pentru implementare pe platforma aleasă.

3. Metodologia PRO-INCLUSIV de dezvoltare a aplicațiilor Web

3.1. Trăsături generale

Diferența principală între proiectele tradiționale de software și proiectele de dezvoltare Web este natura conținutului. Multe proiecte Web gestionează un conținut foarte bogat. Un sit Web poate fi o aplicație sau un sistem hipertext (pagini de conținut). Multe proiecte, printre care și PRO-INCLUSIV, sunt o combinație a celor două, în care caz trebuie avute în vedere ambele aspecte.

Metodologia de dezvoltare a aplicațiilor Web PRO-INCLUSIV este centrată pe utilizator, acesta fiind punctul de plecare al proiectării. Datele și informațiile disponibile sunt modelate din punctul de vedere al potențialilor utilizatori, aplicațiile având astfel un grad ridicat de utilizabilitate.

Abordarea centrată pe utilizator ia ca punct de plecare necesitățile și cerințele utilizatorilor unei aplicații Web, ținând cont de faptul că audiența unui sit poate fi compusă din diferite tipuri de vizitatori/utilizatori. Aceștia pot avea cerințe diferite (cerințe informaționale, cerințe funcționale sau cerințe de utilizabilitate). Utilizatorii unui sit pot fi clasificați în clase de utilizatori. Fiecare clasă are propriile cerințe și caracteristici care pot fi reflectate atât în conținutul sitului (nu toate informațiile sunt relevante pentru toți utilizatorii) cât și în “interfața” sitului (limbajul folosit, prezentarea, caracteristicile navigatoarelor/browserelor existente). Această abordare conduce la situri dezvoltate pentru necesitățile vizitatorilor și de aceea pot avea un grad ridicat de utilizabilitate/accesibilitate și de satisfacție.

Metodologia PRO-INCLUSIV face o distincție clară între proiectare (care nu conține detalii de implementare) și implementare, care ia în considerare limbajul de implementare folosit, gruparea în pagini, folosirea meniurilor, legăturile statice și dinamice. Distincția este similară cu distincția făcută în proiectarea bazelor de date între schema conceptuală și schema logică (de exemplu, schema relațională). Această distincție și-a dovedit din plin utilitatea în ultimii 15 ani, permițându-ne să propunem o metodologie pentru proiectarea siturilor Web care nu va fi afectată de diversitatea și învechirea rapidă a tehnologiilor Web, precum și de limitările curente privind implementarea.

Un alt avantaj important este acela că schema conceptuală a unui sit, realizată în timpul proiectării conceptuale, poate fi pusă la dispoziția utilizatorilor sitului, pentru a preveni căutarea unor informații ce nu sunt disponibile și pentru a reduce sindromul „pierderii” în hiperspațiu.

Metodologia Pro-Inclusiv propune următoarele faze de dezvoltare a aplicațiilor Web (figura 4-2):

- Specificarea cerințelor;
- Modelarea informațiilor;
- Proiectarea navigării;
- Proiectarea prezentării;
- Implementarea;
- Testarea și evaluarea;
- Întreținerea și dezvoltarea.

Fiecare dintre aceste faze conține mai mulți pași, unii obligatorii, alții necesari doar pentru aplicațiile foarte complexe.

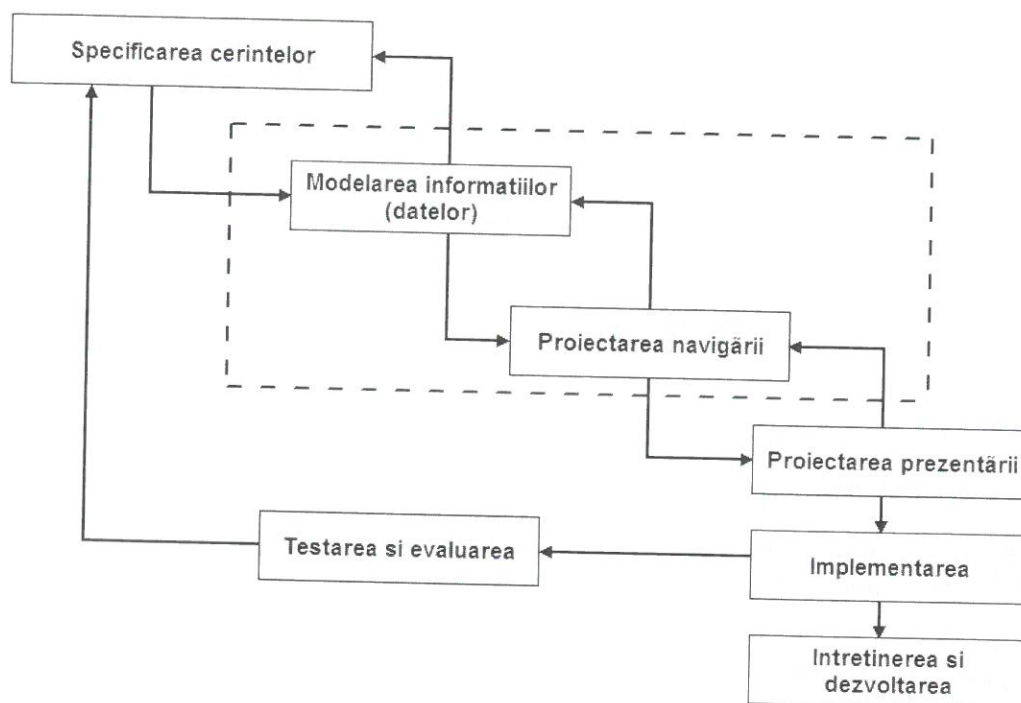


Figura 4-2. Procesul iterativ de dezvoltare a siturilor Web

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

3.2. Specificarea cerințelor (analiza)

În această fază se identifică și se clasifică audiența sitului Web. Sunt identificate cerințele informaționale pentru diferite clase de utilizatori (CE trebuie prezentat) precum și caracteristicile claselor de utilizatori (CUM trebuie prezentate informațiile). Clasele de utilizatori pot fi împărțite în profile. Clasele de utilizatori descriu cerințele informaționale, iar profilele descriu cerințele de utilizabilitate. O componentă cheie a fazei de analiză a proiectării centrate pe utilizator este analiza utilizatorului, care furnizează detalii despre cel ce utilizează produsul. Analiza utilizatorului identifică rolurile și definește caracteristicile utilizatorului, cum ar fi cunoștințele, experiența și abilitatea în folosirea unor produse similare, mediul, frecvența utilizării și în funcție de tipul produsului, echipamentul hardware, software și tehnologiile asistive.

Tehnicile recomandate pentru analiza cerințelor utilizatorilor sunt următoarele:

- chestionare;
- interviuri;
- observații (directe sau indirecte).

Importanța abordării accesibilității în primele faze ale proiectului

Accesibilitatea trebuie introdusă încă de la începutul proiectului. Atunci când accesibilitatea este abordată mai târziu în proiectarea produsului, schimbările de proiectare necesare pot fi foarte costisitoare. Abordarea accesibilității de timpuriu poate mări impactul pozitiv și micșora timpul și costul proiectării unor produse accesibile.

Un exemplu de implementare mai puțin eficientă a accesibilității într-o fază ulterioară este un site în care mai multe fotografii sunt stocate într-o bază de date și incluse în pagini web generate în mod dinamic. Dacă s-a luat în considerare accesibilitatea de la începutul proiectului, atunci în proiectarea inițială a bazei de date au fost incluse câmpuri pentru text alternativ și descrieri lungi, iar paginile web generate vor include textul alternativ și descrierile lungi, așa cum este recomandat. Dacă baza de date nu a fost dezvoltată cu aceste câmpuri, textul nu va fi introdus odată cu imaginile, iar codul ce generează

paginile dinamice nu va include descrierile. Va fi mult mai costisitor să se facă aceste schimbări mai târziu, în ciclul de dezvoltare al produsului.

Definirea rolurilor utilizatorului și a caracteristicilor sale este foarte importantă pentru proiectarea produsului, deoarece presupunerile proiectantului despre cine va folosi produsul pot să difere de ideile prezentate de utilizatori. De aceea este foarte important să se includă accesibilitatea încă din faza de analiză. Fără un proces formal de a-i lua în considerație pe ceilalți, se întâmplă să proiectezi pentru tine însuși. De aceea multe produse sunt proiectate pe baza preferințelor abilității și mediului proiectantului.

Proiectarea de produse care să țină cont și de necesitățile persoanelor cu dizabilități este încă relativ necunoscută. Ca urmare, tipurile de utilizatori care pot folosi produsele și situațiile în care produsele pot fi utilizate pot fi mai puțin inclusive decât dacă ar fi fost luate în considerație necesitățile persoanelor cu diferite abilități și dizabilități.

3.3. Modelarea informațiilor

Scopul acestei faze (cunoscută și sub numele de modelarea datelor) este de a modela cerințele informaționale exprimate în descrierile claselor de utilizatori. Aceasta se face construind (conceptual) modele ale obiectelor pentru diferite clase de utilizatori. Aceste modele se numesc Modele obiect utilizator (Mou).

Modelarea informațiilor urmează pașii de mai jos:

- Elaborarea cerințelor informaționale formale (de obicei sub forma unui text structurat, în limbaj natural);
- Împărțirea cerințelor informaționale formale în așa-numitele cerințe informaționale elementare;
- Pentru fiecare cerință informațională elementară se face un „chunk” (un chunk este cea mai înaltă modalitate de organizare a informației de care dispune un subiect la un moment dat). Cerința informațională elementară poate fi definită ca o cerință informațională care tratează un singur tip de obiect.

Luând în considerație accesibilitatea cu diferitele ei abordări și resurse încă din primii pași ai proiectării, aceasta va putea fi incorporată mai eficient. Dacă accesibilitatea este neglijată, modificările ulterioare pentru realizarea accesibilității vor fi foarte costisitoare.

Referința preeminentă relativă la accesibilitatea Web este „Web Content Accessibility Guidelines” (WCAG), versiunea 1.0, publicată de World Wide Web Consortium în 1999 (WCAG, 1999). WCAG este unul din cele 3 seturi de ghiduri/recomandări dezvoltate de Web Accessibility Initiative (WAI), celelalte două fiind User Agent Accessibility Guidelines (UAAG) și Authoring Tools. WCAG 1.0 (Ghiduri de accesibilitate a conținutului Web) explică autorilor cum să creeze un conținut Web accesibil. WCAG 1.0, dezvoltat de Grupul de lucru Web Content Guidelines a devenit Recomandare W3C în 5 mai 1999 și conține 14 ghiduri cu 65 puncte de control. WCAG 1.0 identifică trei niveluri de prioritate și trei niveluri de conformitate. În cadrul acestei faze a temei de cercetare a fost elaborat un set de “Recomandări de accesibilitate” (vezi paragraful 2.2.2).

În cazul aplicațiilor mai complexe, este necesară și o sub-fază de modelare funcțională (procesele și operațiile/serviciile care sunt puse la dispoziția utilizatorilor).

3.4. Proiectarea navigării

În această fază se construiește un Model conceptual al navigării. Acest model de navigare constă dintr-un număr de căi de navigare (navigation tracks), câte una pentru fiecare profil. O cale de navigare exprimă modul în care utilizatorul unui anumit profil poate naviga prin informațiile disponibile. Acest lucru se descrie în termeni de componente (noduri) și legături. Distingem componente informaționale, componente de navigare și componente externe.

Fiecare cale de navigare constă din trei niveluri. Nivelul cel mai de jos este nivelul context și este folosit pentru conectarea la diferite căi de navigare. Nivelul de mijloc, nivelul de navigare oferă diferite căi de accesare a informațiilor. Nivelul cel mai de jos este nivelul informație și conține informația prezentă prin intermediul componentelor informaționale.

Fazele de modelare a informațiilor (datelor) și proiectare a navigării fac parte din proiectarea conceptuală. Abordarea separată a fazelor de modelare a informațiilor (datelor) și proiectare a navigării mărește gradul de flexibilitate a aplicațiilor.

Proiectarea logică a bazei de date. Dacă pentru întreținerea informațiilor furnizate de un sit Web (sau a unei părți a acestora) se folosește o bază de date, atunci este necesară includerea acestei faze. Schema logică a bazei de date poate fi derivată din modelul informațiilor (modelul datelor).

3.5. Proiectarea prezentării

În această fază se proiectează aspectul sitului Web. Scopul principal este de a crea un aspect consistent, plăcut și eficient pentru proiectarea conceptuală realizată anterior. Există foarte multă literatură (inclusiv în limba română) relativă la acest aspect. Rezultatul implementării proiectării este un model de prezentare.

- Sub-fazele proiectării prezentării sunt următoarele:
- Proiectarea structurii sitului. În această sub-fază se va decide modul în care componentele (nodurile) definite în proiectarea navigării vor fi grupate în pagini. Vor fi definite paginile abstracte: fiecare pagină abstractă putându-se concretiza în multiple instanțieri de pagini concrete în faza de implementare.

Proiectarea machetei (layout) paginilor. Există trei seturi de concepte de modelare pentru a descrie macheta unei pagini. Fiecare set oferă un nivel diferit de abstractizare:

- *Concepte primitive de prezentare* – un set de baza de blocuri de construcție necesare proiectării unei machete a paginii. Tipurile de date primitive de prezentare sunt: audio, imagine, email, șiruri de caractere, întregi, video. Aceste primitive pot fi împărțite în *elemente de poziționare* și *elemente multimedia*;
- *Concepte de prezentare de nivel înalt*, care exprimă concepte de prezentare de tip liste, meniuri și secțiuni, logo-uri, banere etc.;
- *Concepte de tip șablon (template)*. Un șablon (template) poate fi definit ca o combinație a următoarelor elemente: subsol, antet, bară stângă, bară dreaptă, panou de conținut și zonă de editare. Șablonul (template) definește macheta generală a unei pagini. În această fază vor fi specificate șabloane (template) ce vor fi folosite pentru diferite tipuri de pagini. Pentru a specifica stilul sitului (fonturi, culori, grafice etc.) recomandăm utilizarea tehnicii *foilor de stil (CSS)*.

Proiectarea paginii. Proiectantul va descrie unde va fi poziționată informația pe o pagină (definită de modelele obiect utilizator) și cum va fi prezentată. Se va decide, de asemenea, unde vor fi prezentate legăturile (definite în proiectarea navigării). Șablonul (template) definit pentru fiecare pagină va fi luat ca punct de plecare.

La proiectarea paginilor trebuie să se țină cont de următorii factori:

Coerența, adică ușurința cu care vizitatorii își formează un model mental despre domeniul expus pe sit. Deosebim coerență locală, în contextul unei singure pagini și coerență globală, când vorbim de tot situl. Coerența globală este asigurată prin utilizarea unui număr redus (de obicei unul) de șabloane asociate unei entități.

Suprasolicitarea cognitivă. Există o limită a volumul de informație pe care o poate procesa mintea umană. O aplicație ce folosește un număr excesiv de fonturi, culori pe fiecare pagină necesită de la utilizator mai mult timp pentru adaptare. Pe lângă adaptarea la interfață există și problema consecvenței. Un sit web ce prezintă obiectele amplasate diferit de la o pagină la alta este mai puțin consecvent. Coerența și suprasolicitarea cognitivă se referă la interfața aplicației web.

Coeziunea informației într-un document. Informația din cadrul unei pagini trebuie să prezinte tot atâta coeziune ca și în lumea reală. De ex. dacă un document conține informații despre oferte de job-uri, atunci el este mai coeziv, deci mai bun, decât un document ce prezintă informații și despre oferte și despre cereri. Prin document înțelegem nu o pagină web, ci unitatea cu care se operează în cadrul unei aplicații.

Conectarea informației între documente. Documentele trebuie să fie independente unele de altele, adică să nu conțină informații ce se repetă. La proiectarea siturilor mari ce prezintă diverse tipuri de conținut trebuie aplicate metode de facilitare a procesului de adaptare la conținut.

3.6. Implementarea

În această fază are loc realizarea propriu-zisă a sitului folosind mediul de implementare ales. De exemplu, implementarea în HTML înseamnă că modelul proiectat trebuie convertit într-un set de fișiere care să conțină codul sursă HTML. Acest pas poate fi automatizat prin folosirea unor instrumente și medii disponibile pentru asistarea în implementarea HTML.

Pentru a îmbunătăți întreținerea, tot mai multe situri au informațiile de bază (sau o parte din ele) stocate într-o bază de date. Paginile sunt generate (de obicei, dar nu obligatoriu în timpul execuției) prin extragerea informațiilor necesare din baza de date. Informațiile pot fi extrase dintr-o bază de date existentă sau va fi implementată o nouă bază de date în acest scop.

La elaborarea aplicațiilor web sunt folosite diverse protocoale de comunicare, limbaje, sisteme soft și chiar sisteme de operare diferite, care deseori sunt greu de combinat. Proiectarea arhitecturii este un proces complex din cauza neomogenității tehnologiilor folosite. Obiectivele urmărite de proiectant sunt performanța, scalabilitatea, securitatea. Trebuie luate în considerație și restricțiile de timp, resurse umane disponibile, costul soluției elaborate și a softului utilizat la elaborarea acestuia etc.

În faza de modelare a informațiilor recomandăm folosirea ghidului WCAG, unul din cele 3 seturi de ghiduri/recomandări dezvoltate de Web Accessibility Initiative (WAI). TUAAG (Tehnicile pentru Ghidurile de Accesibilitate a agenților utilizator) oferă detalii de implementare a WCAG. Principalele tehnici ale WCAG 1.0 sunt:

- Tehnici de bază;
- Tehnici HTML;
- Tehnici CSS.

Aceste tehnici oferă un ghid de implementare, incluzând explicații, strategii și exemple detaliate. Documentele tehnice sunt organizate pe topici, de exemplu, Tehnicile HTML includ secțiuni privind formularele, imaginile, listele, legăturile, tabelele etc.

Recomandăm, de asemenea, folosirea ghidurilor tehnice WAI pentru tehnologii precum XML, CSS, HTML, Java, JavaScript, SVG, SMIL, PDF etc.

3.7. Testarea și evaluarea

Utilizarea metricilor constituie un bun mecanism pentru evaluarea unui sit Web. Calero et al. (2004) au realizat o clasificare a metricilor Web cu 385 metrici folosind modelul WQM, un model al calității Web tridimensional. Ca urmare a acestei clasificări, a rezultat ca tripletele “utilizabilitate, operare, prezentare” cu 149 metrici și “utilizabilitate, întreținere, prezentare” cu 93 de metrici sunt tripletele cu metrici bine definite. Cubul definit de Calero et al. (2004) este compus din acele aspecte de luat în considerație la evaluarea calității siturilor: caracteristici, procesele ciclului de viață și aspecte calitative, care pot fi considerate ortogonal (figura 4-3).

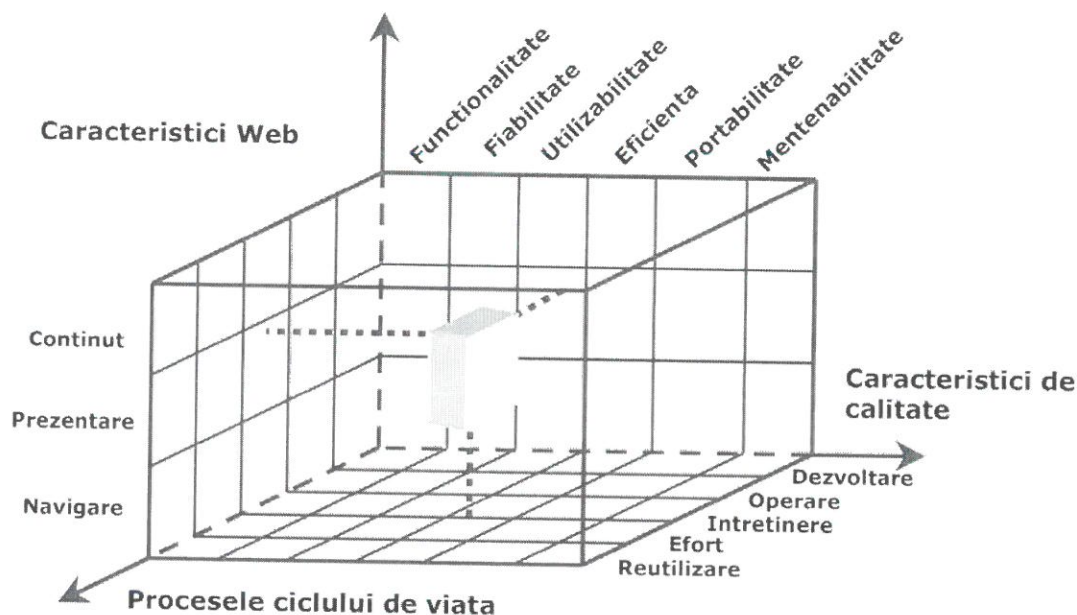


Figura 4-3. Distribuție metrică pe dimensiunile modelului

3.8. Abordări în evaluarea accesibilității

Schemă de certificare europeană privind accesibilitatea Web

Instituțiile europene de standardizare CEN/ISSS încearcă să obțină un agreement pentru o schemă de certificare europeană privind accesibilitatea Web. O astfel de marcă europeană de calitate pentru accesibilitatea Web ar face posibilă identificarea siturilor care respectă conținutul recomandărilor W3C WAI. Marca se bazează pe utilizarea unei metodologii de evaluare a accesibilității Web, UWEM (Unified Web Evaluation Methodology), în cadrul unei scheme europene de certificare. Necesitatea unor scheme de certificare a accesibilității se impune din faptul că mai multe State Membre au o legislație prin care accesibilitatea și evaluarea conformității sunt obligatorii.

Abordarea Dante

Majoritatea siturilor Web nu suportă accesibilitatea pentru utilizatorii cu dizabilități vizuale. De obicei, acești utilizatori folosesc aplicații de tip cititoare de ecran care citesc secvențial conținutul unei pagini Web, în audio. Din păcate, cititoarele de ecran nu sunt capabile să detecteze înțelesul diferitelor obiecte din pagină și astfel o serie de cunoștințe semantice implicite, transmise prin modul de prezentare a paginii, sunt pierdute. Yesilada et al. (2003) prezintă abordarea Dante, care poate să depășească aceste limite. Obiectele identificate sunt adnotate cu termeni din ontologia WafA (Web Authoring for Accessibility Ontology) pentru a explicita rolul acestora. Ontologia WafA își propune să încapsuleze cunoașterea extensivă pentru a explicita informațiile privind structura și navigarea paginilor Web. În esență, ontologia WafA definește concepte legate de modul în care obiectele sunt prezentate într-o pagină (proprietățile lor structurale) și cum sunt folosite pentru a realiza navigarea dorită. Această ontologie este folosită pentru a adnota manual pagina Web. Aceste adnotări explicitează cunoașterea implicită despre rolul obiectelor din pagină și permit cititoarelor de ecran să-și îmbunătățească în mod semnificativ capacitățile. Una dintre limitările majore ale acestei abordări este faptul că adnotările se fac manual. Cu toate acestea Dante este o abordare foarte utilă pentru a furniza accesibilitate utilizatorilor nevăzători. Odată codificată cunoașterea extensivă în ontologie, Dante transcodifică o pagină Web astfel încât obiectele vizuale din pagină să poată să-și îndeplinească rolul stabilit și în prezentările audio.

Transcodificarea HTML în VoiceXML

Shao et al. (2003) au definit o abordare bazată pe Transcodificarea HTML în VoiceXM. Ei au definit un limbaj de adnotare propriu, numit VXPL pentru a indica diferite structuri ale unei pagini Web.

Vocabularul propus este, totuși, mai puțin detaliat decât ontologia WAFa și nu prezintă aceeași înțelegere și analiza adâncă a paginilor Web ca aceasta.

3.9. Instrumente

Succesul W3C WAI în dezvoltarea de ghiduri de accesibilitate a condus la necesitatea unor instrumente pe care dezvoltatorii Web să le utilizeze pentru a verifica dacă resursele lor Web sunt conforme cu ghidurile WAI de accesibilitate. În prezent, este disponibilă o varietate de instrumente ca Bobby (Watchfire, n.d.), (Webaim, n.d.). Astfel de instrumente au limitele lor (de exemplu, ele pot determina dacă o imagine conține atributul ALT, dar nu și dacă textul explicativ oferă o descriere acceptabilă a imaginii). Totuși, acestea joacă un rol important dacă sunt utilizate împreună cu verificarea manuală a resurselor Web.

Deși aceste instrumente au fost utilizate de dezvoltatorii Web pentru a analiza problemele de accesibilitate pe propriile situri Web, ele pot fi utilizate pentru a verifica conformitatea cu ghidurile de accesibilitate pe siturile Web. Aceste instrumente de audit și, în unele cazuri, de reparare încearcă să măsoare conformitatea cu ghidurile WAI, considerându-le ca definitive. Prezentăm, mai jos, câteva astfel de instrumente.

1. **Serviciul de validare al W3C (HTML Validation Service and Toolkit)** testează conformitatea codului HTML cu specificațiile W3C. Serviciul este flexibil și permite validarea versiunilor curente și mai vechi de cod HTML cu condiția ca autorul să specifice versiunea la începutul documentului HTML. Serviciul de validare W3C verifică erorile de sintaxă și erorile de accesibilitate (de exemplu, elementele IMG trebuie însoțite, în mod obligatoriu, de atributul ALT);
2. **HERA** este un instrument de verificare a conformității paginilor Web cu WCAG 1.0 dezvoltat de Fundația SIDAR, Spania. Acest instrument realizează un set preliminar de teste pe pagină și identifică orice eroare care poate fi detectată automat, indicând și punctele de control (checkpoints) necesare pentru o verificare manuală. Hera asistă revizia manuală, furnizând instrucțiuni despre modul în care trebuie realizate testele și oferind două vizualizări ale paginii (normală și codul sursă HTML), având puse în evidență (prin culori și icon-uri) cele mai importante elemente de verificat. Există și posibilitatea de a genera un raport pentru tipărire sau descărcare în format XHTML, RDF/EARL sau PDF;
3. **Utilitarul Tidy** oferă, în plus, sfaturi despre cum se poate corecta codul pentru realizarea accesibilității. Tidy utilizează recomandările HTML 4.0 ca standard. Aplicațiile MS Windows pot fi configurate pentru a include și utilitarul Tidy;
4. **Serviciul de validare Bobby**, realizat de Center for Applied Special Technology (CAST), este un serviciu de validare a accesibilității care poate examina o singură pagină Web sau un sit complet. Bobby validează paginile Web pe baza recomandărilor inițiativei WAI a W3C și le testează pe mai multe sisteme de operare, browsere și agenți utilizatori; Bobby include un sistem de raportare complet, permițând specificarea parametrilor de validare. În plus, Bobby conține un menu pull-down, care permite specificarea extinderii analizei URL-ului. Se pot alege următoarele opțiuni:
 - extindere urmărire legături;
 - urmărire legături în domeniul URL-ului;
 - urmărire toate legăturile.

Rezolvând erorile de accesibilitate și respectând recomandările oferite de produsul Bobby se poate asigura o compatibilitate minimă cu accesibilitatea Web. Schimbările recomandate reduc incompatibilitatea cu recomandările W3C.

5. **Produsul Wave**. Wave este un program dezvoltat de "Institute on Disability", Pennsylvania, pentru verificarea accesibilității site-urilor Web. Programul Wave atașează icon-uri și text unei pagini pentru a ne ajuta să verificăm dacă este accesibilă.
6. **A-Prompt**. În Canada, Adoptive Technology Resource Centre al Universității din Toronto a dezvoltat un instrument de verificare a accesibilității Web numit A-Prompt. Acest software examinează paginile Web din punct de vedere al accesibilității, realizează automat reparații acolo unde este posibil și asistă proiectantul să facă reparații manuale, dacă este necesar.

Instrumentele automate pot stabili dacă un sit nu este accesibil sau, mai exact, dacă un sit respectă recomandările WAI care pot fi testate cu instrumente automate. Dar o astfel de abordare nu poate stabili dacă un sit este de fapt accesibil. Este imposibil pentru orice unealtă să facă o astfel de verificare automată și completă. Acest lucru poate fi stabilit numai printr-o testare manuală. Jacob Nielsen în Nielsen (2000) și Nielsen (2003) demonstrează că soluția cea mai bună pentru testarea utilizabilității (dar care poate fi aplicată și pentru accesibilitate) este testarea cu cinci utilizatori. Instrumentele de evaluare pot mări eficiența evaluării, salvând timp și efort; totuși, acestea nu pot înlocui evaluatorii umani. Instrumentele nu trebuie văzute ca un substitut al evaluării umane, ci ca un ajutor.

4. Concluzii

Metodologia PRO-INCLUSIV face o distincție clară între proiectare (care nu conține detalii de implementare) și implementare, care ia în considerare limbajul de implementare folosit, gruparea în pagini, folosirea meniurilor, legaturile statice și dinamice. Distincția este similară cu distincția făcută în proiectarea bazelor de date între schema conceptuală și schema logică (de exemplu, schema relațională). Metodologia Pro-Inclusiv propune următoarele faze de dezvoltare a aplicațiilor Web: Specificarea cerințelor; Modelarea informațiilor; Proiectarea navigării; Proiectarea prezentării; Implementarea; Testarea și evaluarea; Întreținerea și dezvoltarea.

Fiecare dintre aceste faze conține mai mulți pași, unii obligatorii, alții necesari doar pentru aplicațiile foarte complexe. Aplicarea metodologiei este exemplificată pe o aplicație web destinată persoanelor în vârstă și cu dizabilități. În cadrul acestei etape este prezentată structura funcțională (arhitectura) a aplicației, structura de date și soluția de implementare.

Bibliografie

1. **BARESI L., F. GARZOTTO, P. PAOLINI:** Extending UML for Modelling Web Applications. Annual Hawaii Int.Conf. on System Sciences, Miami, USA. January, 2001, pp. 1285 - 1294.
2. **CALERO, C., J. RUIZ, M. PIATTINI:** A Web Metrics Survey Using WQM. În: Koch, N., Fraternali, P., Wirsing, M. (Eds), Proc. of the 4th Int. Conf. on Web Engineering, Springer-Verlag, Heidelberg, 2004, ICWE 2004, LNCS 3140, pp. 147-60.
3. **CERLS., P. FRATERNALI, A. BONGIO:** Web Modeling Language (WebML): a Modeling Language for Designing Web Sites, the Ninth Int. World Wide Web Conf., Amsterdam, Netherlands, 15-19 May 2000 www.webml.org/webml/upload/ent5/1/www9.pdf
4. **DE TROYER, O., C. LEUNE:** WSDM: A User-Centered Design Method for Web Sites. Computer Networks and ISDN Systems. Proc. of the 7th Int. World Wide Web Conference, Elsevier, 1998, pp. 85 - 94.
5. **DISTANTE, D., P. PEDONE, G. ROSSI, G. CANFORA:** Model-Driven Development of Web Applications with UWA, MVC and JavaServer Faces. ICWE 2007, pp. 457-472.
6. **ESCALONA, M. J., N. KOCH:** Requirements Engineering for Web Applications: A Comparative Study. Journal on Web Engineering, Vol. 2, No. 3, February, 2004, pp. 193-212.
7. **ESCALONA, M. J., M. MEJÍAS, J. TORRES:** Developing systems with NDT & NDT-Tool. The 13th Int. Conf. on Information Systems Development: Methods and Tools, Theory and Practice. Vilna, Lituania, September 2004.
8. **OLSINA, L.:** Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy. The 1st Int. Workshop on Hypermedia Development, Hypertext, 1998, Pittsburg, USA.
9. **PERRONE, V., D. BOLCHINI:** Designing Communication Intensive Web Applications: A Case Study. Proc. of the VII Workshop on Requirements Engineering (WER 2004), 9-10 December, 2004, Tandil, Argentina.
10. **SHAOA, Z., R. CAPRA, M. PEREZ:** Transcoding HTML to VoiceXML. Proc. of 15th IEEE Int. Conf. on Tools with Artificial Intelligence, Sacramento, California, 2003.
11. **VILAIN, P., D. SCHWABE, C. SIECKENIUS DE SOUZA:** A Diagrammatic Tool for Representing User Interaction. Proc. UML'2000. LNCS 1939, 2000, pp. 133-147.
12. * *: UWA Consortium, Ubiquitous Web Applications. In: Proceedings of the eBusiness and eWork Conference 2002, (e2002: October 16-18 2002, Prague, Czech Republic) (2002)

13. **YOURDON, E.:** Modern Structured Analysis, Prentice-Hall, 1998.
14. * *: W3C (n.d.) HTML Testing - HTML Validation Service and Toolkit, <http://www.w3.org/MarkUp/html-test/>
15. * *: WAI (1999). *Web Content Accessibility Guidelines 1.0, W3C 1999*. W3C Web site: <http://www.w3.org/TR/WCAG10/>
16. * *: WAI (2004a). *Policies Relating to Web Accessibility*. W3C Web site: <http://www.w3.org/WAI/Policy/>
17. * *: WAI (2004b). *Web Content Accessibility Guidelines 2.0. W3C Working Draft 30 July 2004*. W3C Web site: <http://www.w3.org/TR/WCAG20/>
18. **SHAOA, Z., R. CAPRA, M. PEREZ:** Transcoding HTML to VoiceXML. Proc. of 15th IEEE Int. Conf. on Tools with Artificial Intelligence, Sacramento, California, 2003.
19. **YESILADA, Y., R. STEVENS, C. GOBLE:** A foundation for tool based mobility support for visual impaired web users. Proc. of the 12th Int. World Web Conf., 2003, pp. 422-430.