Ciclul de dezvoltare a unei aplicații software. Cerinte functionale si nonfunctionale. Studii de caz.

I. Obiective

- Se urmareste intelegerea modului in care pot fi formulate cerintele functionale si nonfunctionale pentru descrierea unei aplicatii software.
- Se urmareste prezentarea studiilor de caz ce vor fi folosite în cadrul orelor de aplicații practice.

II. Breviar teoretic

Vezi note de curs (Capitol 1)

Etapele mai importante ale unui proiect se numesc **faze**. Insiruirea fazelor de la inceputul pana la finalizarea proiectului constituie **ciclul de viata al proiectului**.

Pentru dezvoltarea unei aplicatii software se poate folosi unul din urmatoarele modele:

- Cascada - cu fazele Cerinte, Design, Implementare, Testare parcurse in ordine;
- Prototip - in primele faze este dezvoltat un prototip care experimenteaza anumite alternative si apoi se dezvolta aplicatia propriu-zisa;
- Incremental – aplicatia este livrata pe incremente, fiecare increment adaugand anumite functionalitati;
- Spirala – aplicatia este dezvoltata pornind de la un schelet de baza pe care se dezvolta componentele.

Detalii privind aceste modele de dezvoltare sunt oferite in curs (Capitol 1, subcapitol “Modele de dezvoltare a aplicatiilor software”), impreuna cu indicatii privind oportunitatea folosirii lor intr-un anume context.

Indiferent de tipul de model de dezvoltare ales, pentru buna derulare a proiectului este important ca cerintele privind rezultatul proiectului sa fie foarte clar formulate, sa ofere o descriere completa si corecta a ceea ce trebuie dezvoltat pe proiect. Aceste cerinte trebuie intelese de toti cei implicati in proiect (indiferent de pregatire), de aceea documentele nu trebuie sa includa detalii tehnice inutile, ci mai mult sa descrie intr-un limbaj accesibil ceea ce trebuie realizat.
Documentele trebuie irerahizate de la general la detaliu, pentru a permite urmarirea specificatiilor de o maniera simpla, pana la nivelul de detaliere dorit, in functie de tipul implicarii in proiect (manager, client, dezvoltator, etc).

O varianta de lucru ce poate fi folosita pentru cazul aplicatiilor IT (software/hardware) poate fi include:

1. *descrierea generala*, succinta a functionalitatilor aplicatiei (plecand de la principalii actori si actiunile permise ale acestora);
2. *detalierea cerintelor functionale* in documentul Use-case-uri – acest document va descrie pentru fiecare use-case fluxul de baza si fluxurile alternative, starea initiala/finala, cerintele impuse, posibile extensii;
3. *precizarea cerintelor nefunctionale* in documentul de Specificatii suplimentare
4. *dictionar de termeni* – acest dictionar poate elimina confuziile din prezentare si poate ajuta la scurtarea prezentarilor.

Sectiunea III prezinta un exemplu de completare a acestor documente. Desigur, nu e vorba de un standard unanim acceptat. Si alte variante - care raspund solicitarilor de claritate, simplitate in parcurgere, descriere completa - pot fi considerate.

In mod evident, aceste descrieri sunt fi actualizate (completate, modificate) pe durata derularii proiectului. De exemplu, unele specificatii pot fi detaliate dupa ce se decide design– ul. Mai mult, la modelul de dezvoltare de tip prototip se poate chiar renunta la unele cerinte sau unele cerinte pot fi modificate.
Documentele realizate ar trebui sa evidentieze si istoria acestor modificari, daca ele intervin.

III. Exemple

Caz 1: Aplicatie client-server pentru înscriere la cursuri şi înregistrare-vizualizare note

D1. Glosar/dicţionar de termeni

Curs – disciplină oferită de universitate
Ofertă de curs - variantă pentru un curs (ofertele unui curs au activ. didact. în zile/ore diferite, cu profesori diferiţi)
Catalog de cursuri – lista de oferte de cursuri
Facultate – totalitate profesori
Grupă – studenţii înscrişi la o ofertă de curs
Orar – ofertele selectate de un student/profesor pe un semestru
Nota – rezultatul evaluării unui student la un curs
Raport note – notele obţinute de un student într-un semestru
Fişă - notele obţinute de un student în toate semestrele

D2. Descriere generală/"problem statement"

Aplicaţia va înlocui software-ul existent şi va asigura portarea datelor din vechea bază de date
Funcţionalităţi cerute:
Studenţii:
Se pot înregistra la oferte de cursuri.
În catalogul de cursuri vor găsi la fiecare disciplină informaţii despre profesori, cunoştinţe/discipline impuse, credite alocate, zi-ora laborator/curs/seminar…
În prima perioada de înscriere, fiecare student alege mai multe oferte decât strict necesar (ofertele vor fi tratate în ordinea în care apar în lista studentului), aplicaţia nu acceptă suprapuneri de orar sau selectarea unor oferte cu discipline impuse nepromovate

- studenţii vor fi acceptaţi la o ofertă în ordinea mediilor obţinute în semestrele anterioare
- o ofertă este validată dacă un număr minim de studenti s-au încris

- numărul maxim permis de studenți permis pentru o ofertă este impus

La sârşitul primei perioade de înscriere sunt anunțați ce oferte au fost validate şi, dacă nu au acoperit 60 de credite, care este a doua perioada de înregistrare în care se pot înscrie la ofertele rămase libere.

Pot vizualiza orarul, notele.

Profesorii: Se pot înscrie la o ofertă de curs, pot vizualiza grupele lor, orarul lor, pot modifica –vizualiza notele

Secretara: Actualizează informațiile despre profesori, studenți, cursuri, setează perioadele de înregistrare, evaluare, etc

D3. Specificații suplimentare (neincluse la usecase):

Funcționalitate: multiuser
Compatibilitate: interfețe Windows
Fiabilitate, timpi de utilizare: 24h, 7 zile/sapt, perioade de pauza < 10%
Performanțe: 2000 useri simultan, o vizualizare<2sec
Securitate: doar studentul îsi poate schimba ofertele, doar profesorul poate modifica notele, etc
Restricții: transfer date din baza de date veche

D4. Usecases – descriere generală

Login
Menținere info profesori
Menținere info studenți
Închidere perioadă înregistrare
Selectarea cursului de către un profesor
Notare studenți de către un profesor
Selectarea ofertelor de curs de către un student
Vizualizare note de către un student, etc

4. 1. Login

Descriere: utilizatorii primesc sau nu drept de acces în aplicație
Flux de bază:
Actorul introduce nume şi parola.
Sistemul validează parola şi permite accesul actorului
Flux alternative:
Dacă nume/parolă sunt greşite:
Sistemul afişează un mesaj de eroare
Actorul alege dacă se întoarce la fluxul de bază sau iese
Stare inițială: ecran pentru login afişat
Stare finală: succes – actor logat, se afişează ecran cu menu aplicație
insucces: stare neschimbată, se afişează ecran login sau se iese
Cerințe speciale: codificare parole + nume
Extensii posibile: nu

Caz 2: Program pentru identificarea unui proces neliniar (ex: sistem termic - cuptor).

Caz 3: Proiectarea unui regulator PID software pentru comanda unui sistem de recirculare a apei.

Caz 4: Program pentru prelucrarea unei imagini alb-negru şi detectarea prezentei unui obiect cunoscut in respectiva imagine.

Caz 5: Program pentru gestiunea marfurilor intr-o magazie a unei firme de productie si control semi-automat al stocurilor.

IV. Mod de lucru:

1. Pentru aplicatia III. 1 se va completa descrierea pentru alte doua use case –uri.
2. Pentru una din aplicatiile III. 2, 3, 4, 5 se va schița descrierea conform formularelor propuse la aplicația III.1.
3. Indicați pentru aplicația aleasa cum ar putea fi dezvoltată aceasta pe model de tip prototip si incremental – vezi notele de curs.