PROIECT la
Ingineria programării

Cuprins:

Obiectivele proiectului: dobândirea de abilităţi pentru

- ***munca în echipă în scopul realizării proiectelor informatice***
- ***parcurgerea etapelor de realizare a proiectelor software***
- ***comunicare eficientă între membrii echipei şi cu clienţii***
- ***buna prezentare a rezultatelor muncii***
- ***organizarea (managementul) activităţii***

Proiect – partea 1: SPECIFICAŢII

Obiective:

- (auto)**organizarea** activităţii în cadrul echipei
- **Analiza** temei tehnice
- Scrierea **specificaţiilor**

Lecturi:

1. Organizarea activităţii după metoda *echipei programatorului-şef*

Activitatea propriu-zisă de proiectare/realizare de produse program este efectuată în cadrul unor colective mici, care se "fac şi se desfac" în funcţie de proiectele care apar.

Se spune că echipa software de proiectare de proiectare, ideală, este de 5 (până la 7): prea puţini pentru a se forma "bisericuţe" (grupuri cu interese proprii şi pe bază de afinităţi) şi suficienţi pentru proiecte mai ample. De multe ori însă, echipa trebuie să fie mai mare (dar nu mai mult de 12-15, altfel activitatea practic nu mai poate fi coordonată).

In cele mai multe situaţii, echipa este organizată conform ***metodei echipei programatorului şef*** (în funcție de companie și specificul organizării, acesta se poate numi ***arhitect-șef, team leader*** etc.).

- activitatea este condusă de un proiectant cu experienţă, neapărat cel mai bun profesionist din grup (altfel îi vor fi contestate prerogativele de "şef"). Se spune că aplicaţia poate fi realizată integral de programatorul şef, dar ar dura prea mult;

- programatorul şef este secondat continuu de următorul în ierarhia valorică de un "ajutor" care trebuie să cunoască şi el proiectul în detaliu pentru a fi în măsură să preia conducerea grupului în caz de forţă majoră;

- în grup va exista un/o "secretar/ă" care trebuie să degreveze cât mai mult pe ceilalţi membri ai echipei, de activităţile de rutină: elaborare de devize, acte contabile, o parte a documentaţiei tehnice, interfaţa cu alte compartimente ale firmei, o parte a interfeţei cu clientul, etc. Acesta este un individ ordonat, care este dispus să preia fără dificultăţi sau adversităţi rolul de “sistem informaţional” (sau “nervos”) al echipei, prin aceea că strânge/distribuie informaţie, pregăteşte agenda şedinţelor, pune “cap la cap” documentaţiile, bate lumea la cap în legătură cu termenele etc.

- programatorul şef va prelua părţile cele mai dificile şi mai delicate din aplicaţie: drivere, mecanisme de acces, interacţiuni în timp real, etc.

- programatorul şef va distribui sarcinile în cadrul echipei, neapărat în funcţie de posibilităţile fiecăruia. Atenţie ! Plata trebuie făcută în funcţie de munca depusă de fiecare.

- conducătorul echipei trebuie să ştie să catalogheze eforturile echipei şi să închege o activitate desfăşurată de mari individualişti cum sunt programatorii buni: cu fiecare va trebui să vorbească "pe limba lui";

- neapărat, programatorul şef va ţine evidenţa şi va controla individual terminarea modulelor de program aferente fiecăruia; va determina accelerarea activităţii la programatorii rămaşi în urmă;

- fiecare modul va trebui testat conform uneia din cele două strategii prezentate la capitolul 4 (pentru echipă mare sau echipă mică); conducătorul echipei va fi cel care va verifica buna funcţionare a modulelor, cu mediile de testare scrise de proiectanţi;

- susţinerea avizărilor proiectului o va realiza tot "programatorul şef"; de modul cum ştie să susţină proiectul depinde de multe ori venitul lui şi al echipei sale;

Este clar că "programatorul şef" este un "om-orchestră”. Bun profesionist (cel mai bun din echipă !), bun organizator, abil psiholog, răbdător şi chiar charismatic... este aproape un personaj ideal, o ficţiune !

Oricum şi voi trebuie să vă străduiţi spre acest personaj ideal !

2. Organizarea activităţii într-o firmă de software

Câte case, atâtea obiceiuri, se spune. Firmele care realizează proiecte IT au şi ele diverse organizări. Să luăm ca studiu de caz o firmă ipotetică (dar stilul e real !). Aici, activitatea se desfăşoară pe colective (15-30 oameni). Câteva colective alcătuiesc o secţie, care are preocupări dintr-un domeniu mai restrâns (s. ex. secţii de sisteme SCADA, telematică, comenzi numerice etc.). Activitatea secţiilor este coordonată de un director de cercetare. In cadrul institutului există mai multe secţii de proiectare şi poate şi o secţie de microproducţie (aici se pot realiza unicate sau serii mici de echipamente, pentru aplicaţii dedicate).

La îndemâna proiectanţilor stau: o bibliotecă cu serviciu propriu de întocmire a sintezelor informaţionale, servicii auxiliare (legătorie, copiere, desen etc.), acces la bibliotecile de programe ale institutului etc.

Avizările sunt realizate de consiliul tehnic (un fel de sfat al înţelepţilor care hotăreşte dacă proiectul este bun sau nu). Consilierul care se ocupă de un proiect are rolul de a filtra şi a evita eventualele paralelisme în activitatea colectivelor, de a ajusta soluţiile tehnice şi de a impune tehnici mai productive de proiectare şi eventuala refolosire a modulelor de program din bibliotecile de programe ale institutului; şeful de proiect are acces la consilierul desemnat pentru proiectul lui, ori de câte îl solicită.

Un institut de proiectare are o mare capacitate de a realiza proiecte însemnate, deoarece concentrează specialişti, experienţă, conexiuni informative, unele servicii pe care o firmă mică nu şi le poate permite, unele mijloace tehnice de asemenea inaccesibile unei firme mici. De asemenea marca ("faima", credibilitatea) firmei are importanţa ei. Insă o firmă mică este mai flexibilă, mai uşor de condus, poate câteodată supravieţui mai uşor şi adapta mai bine la cerinţele pieţei, are regia mai mică iar "leneşii" sunt mai uşor de depistat.

3. Conţinutul specificaţiilor

Specificaţiile urmăresc descrierea interfeţelor (canalelor de comunicaţie) dintre aplicaţia software (văzută ca o cutie neagră) şi mediu. Prin mediu se înţelege mulţimea tuturor sistemelor cu care interacţionează aplicaţia. Categorii de astfel de sisteme pot fi:

• *utilizatorul – sau utilizatorii (de aici rezultând necesitatea descrierii interfeţei cu utilizatorul);*
sistemele cu care programul interacţionează (în sensul din teoria sistemelor): procese industriale, procese din alte domenii (sisteme complexe de programe, reţele de calculatoare, sisteme bancare, sisteme biologice, sisteme de baze de date etc.), alte aplicaţii sau sisteme de calcul cu care programul interacţionează, asimilabile acestor sisteme (în particular chiar şi pacienţii din aplicaţiile de informatică medicală);

• *bazele de date, fişierele cu care lucrează aplicaţia (ele pot fi privite ca făcând parte din mediu întrucât sunt stocate pe suporturi de memorie externă şi deci conţinutul lor trebuie de asemenea specificat);*

• *perifericele calculatorului pe care rulează programul (consolă, imprimantă etc. care trebuie privite, de asemenea, ca făcând parte din mediu întrucât sunt externe aplicaţiei şi deci conţinutul lor informaţional trebuie de asemenea specificat).*

Practic orice „intră” sau „iese” din aplicaţie trebuie descris în cadrul specificaţiilor, cu explicitarea conţinutului şi formei de prezentare, astfel încât proiectantul, clientul şi utilizatorul final al aplicaţiei să poată să cunoască funcţionarea aplicaţiei, atât la nivel de funcţiuni cât şi la nivel formal (aspectul interfeţelor). Descrierea trebuie realizată astfel încât:

• pe tot parcursul proiectării să se poată urmări în permanenţă, ca obiectiv esenţial, concordanţa dintre specificaţii şi ceea ce se obţine;

• să se poată valida produsul obţinut prin verificarea în cele mai mici detalii a concordanţei dintre acesta şi specificaţii;

• clientul şi utilizatorul final să poată să înţeleagă funcţionarea şi să cerceteze aspectul interfeţelor şi interconectării cu alte sisteme înainte de a începe proiectarea propriu-zisă, pentru a valida modelul dezirabil care să se constituie ulterior în obiectiv pentru proiectare

• proiectanţii să poată să continue modelarea şi proiectarea sistemului dorit, prin împărţirea în subsisteme, astfel încât intrările şi ieşirile să fie definite exact, plecând de la cele ale aplicaţiei văzute ca o cutie neagră;
• proiectanţii să poată să abordeze în paralel şi proiectarea altor sisteme cu care aplicaţia interacţionează, în condiţiile în care "canalele de comunicaţie" sunt definite riguros.

Specificaţiile vor conţine:
- o **foaie de titlu** conţinând semnăturile autorizate
- **colectivul** de elaborare
- **cuprinsul**
- **Memoriu tehnic** şi **anexe** (tot ce nu se poate include firesc în text trebuie pus în anexă, spre exemplu documente externe, tabele mari, schiţe voluminoase etc.)

Memoriul tehnic conţine:

Denumirea proiectului (din tema tehnică – de la client)

Prefață (o frază gen "Prezentul proiect a demarat în data de ... prin prezentarea unei Teme tehnice (v. Anexa 1) la sediul companiei noastre, de către XXX, reprezentantul YYY care cu aceasta a solicitat demararea realizării unui NNNN."); ulterior, fiecare versiune nouă adaugă informații la aceasta.

Nume de cod (denumire comercială) Nu e de loc uşor să găseşti o denumire potrivită pentru un produs ! Trebuie ceva atractiv şi sugestiv !

Introducere: Aici se desccrie necesitatea acestui sistem. Se descriu pe scurt obiective (scopului) pentru care este conceput sistemul precum şi modul de funcţionare pentru îndeplinirea acelor obiective, performanţele vizate.De asemenea se descrie pe scurt interacțiunea cu alte sisteme, precum și modul cum se integrează în asnamblul activității și obiectivelor strategice ale organizației clientului.

Glosar de termeni: aici se definesc termenii tehnici utilizați în document. Nu trebuie să presupunem că cititorul documentului (care poate fi inclusiv reprezentant al clentulu) are expertiză în domeniu.

Definirea cerințelor utilizator: aici se descriu serviciile furnizate utilizatorului de către sistem.

Cerințe funcționale

Funcţiile sistemului se stabilesc pe baza obiectivelor şi se sistematizează de regulă sub forma de listă în care se descriu succint acţiunile pe care le realizează sistemul. Funcţiile trebuie să fie net delimitate unele de altele. Ele trebuie încadrate în categorii bine definite.

Exemplu:
comunicarea cu nivelul ierarhic inferior, pentru un sistem de supraveghere de proces, arhivarea/readucerea datelor etc.

De regulă, se pot identifica în cadrul unei aplicaţii uzuale 4-8 funcţii. Funcţiile trebuie prezentate distinct. Fiecare funcţie trebuie descrisă prin 1-3 fraze. În cadrul descrierii unei funcţii, trebuie identificat prin accentuarea în text („bold") un grup de cuvinte (o sintagmă) care să ofere o caracterizare sintetică a funcţiei respective.
În cadrul descrierii funcţiilor trebuie să fie transpuse sintetic toate cerinţele exprimate în cadrul *Temei de proiectare*. Astfel, dacă sistemul trebuie să reacţioneze într-un anumit mod la evenimente externe, atunci această reacţie trebuie să fie explicată clar şi univoc la funcţia căreia ea îi corespunde. Toate cerinţele generate de serviciile dorite de client trebuie să se regăsească sub o formă sau alta în cadrul prezentării funcţiilor.

Cerințe nefuncționale

Acestea nu sunt legate direct de serviciile furnizate utilizatorilor de către sistem (spre exemplu cerințe de siguranță în furncționare, timp de răspuns, spațiu de stocare necesar, perrformanță, disponibilitate, restricții ale dispozitivelor de intrare-ieșire, restriciții referitoare la reprezentările de date utilizate în interfețele cu alte sisteme etc.).
Câteodată cerințele non-funcționale afectează arhitectura sistemului și cerințele funcționale.
Se adaugă aici și complianța cu standardele (dacă e cazul, dacă utilizatorul solicită aceasta):

Dacă e cazul (dacă există o cerință în acest sens – iar în cadrul proiectelor noaste e cazul) se adaugă un paragraf de **Analiză de risc.** Acesta este destinat evaluării aspectelor de siguranţă în funcţionare dorite (dacă e cazul). O aplicaţie desktop obişnuită (pentru calcule ştiinţifice sau simulare spre exemplu) nu este considerată a fi din categoria care trebuie să aibă o siguranţă deosebită în funcţionare, în schimb una de conducere de proces sau un sistem bancar, da. Se va face, dacă e cazul, o analiză în funcţionare degradată, care urmăreşte consecinţele căderii unor componente ale sistemului: aceste căderi vor fi ierarhizate în ordinea crescătoare a consecinţelor asupra bunei funcţionări a sistemului; vor fi prezentate aceste consecinţe, modul de reacţie dezirabil al operatorului şi eventual vor fi propuse măsuri ce pot fi luate chiar prin proiectarea programului, astfel ca aplicaţia să minimizeze consecinţele.

Arhitectura sistemului

Paragraful este destinat prezentării unei scheme care să ilustreze calculatorul sau reţeaua pe care rulează aplicaţia, periferia care este folosită, precum şi interacţiunea acestui calculator (acestor calculatoare) sau a aplicaţiei cu sisteme din exteriorul aplicaţiei – utilizatori, alte aplicaţii cu care interacţionează, alte sisteme de calcul, procese cu care interacţionează etc. Se voir preciza și funcțiile pe componentele modulelor sistemului. Componentele refolosite din alte aplicații vor fi de asemenea evidențiate.

Specificații ale cerințelor de sistem

Aici trebuie descrise în detaliu cerințele funcționale și nefuncționale și de asemenea se definesc interfețele cu alte sisteme. Aici pot fi prezentate modele de sistem care descriu relațiile dintre componentele sistemului și dintre acestea și mediu.

Cazuri de utilizare (*Use Case*)

Prezintă cazurile de utilizare, aşa cum au fost acestea prezentate la cursul de POO din anul II, respectiv vor fi prezentate mai aplu la cursul de IP de anul acesta, elaborate pentru scenariile de utilizare, respectiv capabilităţile sistemului descris. Se pot folosi editoare specializate pentru lucrul cu UML (spre exemplu ***StarUML*** care poate fi descărcat gratuit spre exemplu de la http://sourceforge.net/projects/staruml/).
Fiecare scenariu de regulă acoperă un număr restâns de posibile interacțiuni. Un scenariu pornește cu o schiță a interacțiunii, la care se adaugă succesiv detalii până la conturarea descrierii complete a interacțiunii. Astfel, un scenariu va include:

1. O descriere a ceea ce sistemul și utilizatorii se așteaptă la startul scenariului.
2. O descriere a fluxului normal de evenimente în cadrul scenariului.
3. O descriere a ceea ce se poate întâmpla anormal li cum sunt tratate aceste situații.
4. Informații despre alte activități care pot să se desfășoare smultan.
5. O descriere a stării sistemului la terminarea scenariului.

Scenariile sunt reprezentate grafic prin relațiile (simbol grafic: *săgeată*) dintre Actori (simbol grafic: *omuleț schematic*) și Cazuri de utilizare (simbol grafic: *oval*). Pentru fiecare caz de utilizare se adaugă o descriere care conține:

Actorii implicați
Descriere a interacțiunii, pas cu pas (adică a *scenariului*)

Datele procesate
Stimuli (intrări) spre exemplu comenzi utilizator
Răspunsul la stimuli
Comentarii

Diagrame de secvență (*Sequence Diagram*)
Acestea dau o bună imagine a derulării scenariilor, pe baza reprezentării UML respective, care va implica Actorii și subsistemele din cadrul sistemului.

Exemplu:
Pentru un sistem ierarhic de conducere de proces, trebuie prezentată pe larg structura informaţiei vehiculate în ambele sensuri între nivele ierarhice, semnificaţiile pentru fiecare componentă, la nivelul de detaliere minim necesar, semnificaţiile comenzilor etc., astfel încât să se poată ulterior realiza proiectarea simultană a ambelor componente software pe baza referenţialului comun al acestor specificaţii.

Modele de stare
Acestea sunt reprezentări ale evoluției stării, bazate pre reprezentări UML de stare. În multe din proiectele abordate, reprezentarea este foarte utilă pentru proiectare. Sigur, se realizează astfel de modele chiar și parțiale, dacă sunt cu adevărat relevante și de folos. Spre exemplu, la sistemul de acces o bună descriere este realizată de o astfel de diagramă.

Interfețe cu alte sisteme
Dacă sistemul interacţionează cu alte sisteme pentru fiecare canal de comunicaţie identificat trebuie explicate:
• modul de comunicare (suportul fizic)
• protocolul de comunicare folosit (dacă este un standard, se menţionează acest lucru şi dacă nu este foarte cunoscut, se prezintă succint principalele aspecte)
• structurarea şi semnificaţia informaţiei vehiculate
Se descriu (dacă e cazul) scimburile de informații cu alte sisteme, prin *Diagrame de secvență* (*Sequence Diagram*) UML sau alte forme de reprezentare.

Evoluția sistemului
Aici se descriu principalele ipoteze de funcționare a sistemului, care ar putea să aibă consecințe în viitor asupra evoluției sistemului. Astfel se analizează eventualele schimbări generate de evoluția hardware sau schimbarea cerințelor utilizatorului.
Spre exemplu, dacă comunicarea cu senzorii se realizează conform unui standard IoT, atunci se poate pe bună dreptate presupune că schimbarea senzorilor e mult ușurată.

De asemenea trebuie efectuată o **analiză SWOT** a utilizării aplicaţiei (de altfel, o parte a celor de mai sus pot fi eventual slăbiciuini sau amenințări în analiza SWOT).

Planificarea lucrărilor
Se va propune un model de ciclu de viaţă pentru dezvoltarea programului, cu justificarea alegerii şi un grafic de eşalonare a lucrărilor, conform acestui model (o diagramă GANTT). Pentru realizarea diagramei se recomandă, dacă se cunoaşte, utilizarea *Microsoft Project*.
Fiecare etapă va cuprinde denumirea, termenul, resursele necesare (proiectanţi, sisteme de calcul, software, altele).
Acest paragraf este parte a efortului de management al proiectului, nu e de fapt parte integrantă a specificațiilor, de aceea căteodată este cuprins în Specificații, câteodată nu, în funcție de specificul relațiilor cu clientul.

Anexe
Acestea furnizează informații detaliate, specifice, despre aplicația care va fi dezvoltată.

Interfaţa cu utilizatorul

Paragraful este destinat definirii tuturor aspectelor care privesc interfaţa cu utilizatorul (aspectul ecranului în toate situaţiile posibile, meniuri, submeniuri, cu acţiuni preconizate la fiecare comandă, ecrane de dialog, definirea acţiunilor pentru toate elementele de comandă, listele de mesaje ale sistemului, aspectul generic al graficelor, rapoartelor, schemelor, listelor, modalităţi de interacţiune cu utilizatorul, specifice etc.).
Se vor realiza desene care să ilustreze o descriere completă a interfeţelor în sensul raţiunilor prezentate în preambul. Cel mai bine este ca descrirea să conţină şi o arborescenţă a dialogurilor posibile, de unde apoi pot fi prezentate rând pe rând toate interfeţele şi dialogurile.

Structuri de baze de date şi fişiere

Trebuie definite la nivel logic structurile de baze de date şi fişiere (adică la nivelul la care sunt specificate spre exemplu câmpuri ale unui tabel, dar cu referire doar la conţinut şi nu la mărimea câmpului). Legăturile între tabele urmează a fi definite doar la proiectarea bazelor de date. De fapt aici se pune problema descrierii bazelor de date, aşa cum rezultă direct din cerinţe şi discuţii, fără a intreprinde cel mai mic efort de proiectare, spre exemplu echipa încă nu se gândeşte la normalizare sau la indecşi.

Tipărirea la imprimantă

Se prezintă generic rapoartele (eventual exemple) şi lista eventualelor mesaje care pot fi tipărite de sistem la imprimantă.
Atunci când e cazul se vor adăuga paragrafe distincte pentru alte periferice.

Evident, toate cele de mai sus sunt necesare în măsura în care sunt aplicabile. Spre exemplu, pentru un sistem încorporat nu există interfeţe utilizator, deci nu e cazul definirii acestora.

E de subliniat că documentaţia tehnică (deci inclusiv Specificaţiile) trebuie să aibă un aspect "tehnic" (deci exprimări exacte şi neredundante, fără înfloriturі dar corecte gramatical şi lingvistic, deci atenţie la exprimare şi greşeli de ortografie, deoarece o companie care produce documentaţii agramate îşi pierde repede bunul renume, dacă l-o fi avut vreodată). Fiecare companie îşi defineşte propriul stil de întocmire a documentaţiei, incluzând aici un şablon, o anumită structură, header-e şi footer-e unitare, fonturi şi reguli de tehnoredactare unitare etc. Se recomandă fonturi simple (s. ex. Arial) şi evitarea risipei de spaţiu (nu se pun spre exemplu capturi mari de ecran, care oricum la această etapă nici nu ar fi de unde să fie !; nu se lasă prea mult loc între titluri, paragrafe etc.)

De aceea vă veţi strădui să propuneţi propriile voastre abordări dar conform celor de mai sus. Cele realizate vor trebui să arate a documentaţii tehnice de firmă serioasă !

Proiect – partea a 2-a : PROIECTARE

Obiective:

- Stabilirea structurii generale (arhitecturii) a aplicaţiei
- Conceperea şi scrierea proiectului

Lecturi:

Proiectarea

Obiectivul primordial al proiectării e să se ajungă la o descriere care să permită pe de o parte delimitarea exactă a muncii în cadrul echipei şi pe de altă parte continuarea cu acţiunea de codificare (scriere de cod-sursă) în cadrul căreia fiecare proiectant să ştie exact ce să facă pentru a scrie codul ca activitate de rutină. Practic, două sunt rezultatele majote ale proiectării: o arhitectură (în care componentele se combină) respectiv o colecţie de descrieri pentru fiecare componentă arhitecturală în parte. Aici la fel de importantă ca şi descrierea componentelor în sine (la nivel de intrări, prelucrăru, ieşiri) este descrierea relaţiilor (a comunicării) dintre acestea. De aici încolo, fiecare programator (adică cel care scrie cod-sursă) poate prelua realizarea unei componente, pe baza descrierii sale şi a relaţiilor acesteia cu componentele "din jur".

Documentaţia de proiectare va conţine următoarele paragrafe:

1. Arhitectura programului
Aici se prezintă o descriere succintă a programului, pe baza unei scheme generale ("arhitectura programului"), cu descrierea componentelor şi a interacţiunilor dintre acestea. Vor fi evidenţiate aspectele de tehnologie folosite, inclusiv mediul de dezvoltare, modelele arhitecturale ("client-server", "three-tier" etc.), principiile generale de funcţionare, vor fi date detalii despre sistemul de operare, dacă e cazul (eventual ca şi paragraf separat). Dacă aplicaţia este pe bază pe dialog, se va face o schemă a formelor aplicaţiei şi a arborescenţei de parcurgere a tuturor dialogurilor.
Se vor utiliza – dacă e aplicabil – reprezentări UML specifice (diagrame de clase, diagrame de stare etc.)

2. Descrierea componentelor (modulelor)
În funcţie de tehnologia folosită, vor fi descrise individual toate componentele de program folosite, pentru fiecare fiind prezentate intrările, prelucrările şi ieşirile şi evidenţiate toate interacţiunile cu alte module. Se poate propune eventual un şablon de descriere care să fie folosit pentru descrierea unitară a tuturor modulelor.
Se vor utiliza – dacă e aplicabil – reprezentări UML specifice (diagrame de clase, diagrame de secvenţe, etc.). Se pot utiliza, pentru mai buna înţelegere a funcţionării, inclusiv scheme logice sau diagrame de stare.

3. Descrierea comunicării între module
În funcţie de tehnologia folosită, vor fi descrise individual toate canalele de comunicaţie dintre componentele de program folosite, pentru fiecare fiind prezentate modul de comunicare, lista de parametri, semnificaţiile acestora, restricţii etc. Se vor utiliza tabele sau descrieri conform unui şablon de descriere unitară a tuturor mesajelor dintre componente și/su module, de asemenea *Diagrame de secvență* UML (*Sequence Diagrams*) editate cu un editor specializat.

4. Structuri de baze de date şi fişiere

Se prezintă pe larg structurile, prin dezvoltarea informaţiilor de la specificaţii. Vor fi prezentate toate informaţiile aferente unui câmp, inclusiv cele deja existente (adică semnificaţia), tip, lungime, restricţii, eventual drepturi de acces. Vor fi definite legăturile, indecşii, cheile primare, se va realiza normalizarea etc., conform celor învățate la disciplina *Baze de date*. De asemenea se va realiza o schemă generală care să cuprindă toate tabelele, cu legături între ele.

Documentaţia de proiectare trebuie să aibă o formă asemănătoare celei de specificaţii (din motive de respectare a principiului uniformităţii), de aceea va fi similară acesteia până la nivelul schemei-bloc inclusiv (cu actualizările şi completările de rigoare) după care următoarele capitole vor fi conform celor de mai sus.

Proiect – partea a 3-a: IMPLEMENTARE

Obiective:

- Scrierea documentaţiei pentru testele de sistem
- Realizarea prototipului funcţional

Lecturi:

Documentaţia corespunzătoare testelor de sistem (dar pe acelaşi principiu se documen-tează şi testele de integrare şi cele de acceptanţă, vezi cursul) este transpunerea în viaţă al principiului ingineriei programării numit **conformabilitate**. Acesta cere ca informaţiile cerute de de verificarea corectitudinii programelor să fie formulate explicit şi să fie disponibile. Rezultă că în cadrul activităţii dintr-o firmă cu specific de proiectare software trebuie elaborare cerinţe specifice de testare, formulate explicit de regulă sub forma unui document.
După ce procesul de testare-depanare este considerat încheiat, membrii echipei trebuie să dovedească unor terţi (propriii şefi, propriul compartiment de calitate, reprezentanţii clienţilor) că produsul realizat “corespunde”, adică îndeplineşte funcţiunile stabilite iniţial prin specificaţii, inclusiv sub aspectul prezentării exterioare a acestora. Pentru ca acest proces de demonstrare să decurgă corect, documentul elaborat special pentru a transpune în practică principiul conformabilităţii trebuie să conţină toate informaţiile pentru ca un terţ:

- să înţeleagă ce anume trebuie să testeze (deci să aibă toate informaţiile pentru a avea acces la specificaţii)
- să ştie ce anume se foloseşte pentru demonstrarea capabilităţilor (mijloace tehnice)
- să ştie care sunt funcţionalităţile care urmează a fi testate, cu o granularitate suficient de fină
- să ştie cum trebuie să testeze, deci ce intrări sau acţiuni trebuie să introducă sau exercite asupra sistemului în vederea testării
- să aibă acces la informaţiile propriu-zise de confirmare (de aici principiul “conformabilităţii”) pentru a putea verifica adecvarea reacţiei sistemului la aşteptări; astfel, practic trebuie ca reacţia la intrarea sau acţiunea exercitată de utilizator, sistemul să răspundă în modul prevăzut.

Desigur, structura documentului pentru testare este în funcţie de cultura internă a organizaţiei. De aceea, putem da doar un exemplu, iar situaţiile din realitate sunt variaţiuni specifice ale exemplului prezentat. Documentul respectiv va trebui oricum să conţină secţiuni care să furnizeze toate informaţiile ce decurg din cele de mai sus. Cel mai adesea, respectivul document este împărţit în secţiuni ce corespund în linii mari alineatelor din enumerarea de mai sus. Câteodată, pentru ultimele 3 alineate, se întocmeşte un tabel, în care pe prima coloană se trec funcţionalităţile testate, pe următoarea, modul de invocare a acestora iar pe ultima, este descris “ce anume trebuie să se întâmple”.

Un scenariu clasic al demonstraţiei se derulează astfel:

- terţul care trebuie să confirme că sistemul funcţionează conform cerinţelor solicită documentaţia şi o studiază
- apoi, solicită mijloacele tehnice precizate în documentaţie (sisteme de calcul, echipamente, reţelistică, software etc.) în configuraţiile prezentate
- apoi stabileşte condiţiile de testare, folosind mijloacele tehnice puse la dispoziţie şi documentaţia de testare
- apoi, trece la baleierea, linie cu linie, a tabelului de de teste de acceptanţă sau a documentului respectiv, punct cu punct, verificând funcţionalităţile astfel descrise

- dacă constată nepotriviri dintre reacţiile reale ale sistemului şi cele prevăzute în documentaţie, va consemna aceasta
- în final, întocmeşte un proces-verbal prin care confirmă că sistemul funcţionează conform cerinţelor sau dimpotrivă, furnizează o listă de neconcordanţe.

Desigur, varianta din urmă este extrem de neplăcută pentru echipa de proiectare, care de altfel a întocmit şi documentul pe baza căruia a fost efectuată testarea ! Am fi tentaţi să credem că echipa poate "trişa", respectiv, scrie mai puţin decât e cazul în lista de teste. Dar oricum terţul va consulta şi specificaţiile şi trebuie să aibă competenţa pentru a decela situaţiile de acest gen. În plus, nici nu are interesul să fie "îngăduitor", fie că e reprezentant al clientului, fie că e de la compartimentul de calitate al propriei noastre firme. Pe de altă parte rareori documentaţia de testare e suficient de completă pentru a putea testa numai cu ea în faţă. De cele mai multe ori, în documentaţia în cauză se fac măcar trimiteri la documentaţia de specificaţii (spre exemplu, e puţin probabil ca în documentaţia de testare să fie descrise toate interfeţele, aşa cum se face la specificaţii), cu indicarea exactă a paginii, figurii etc.

Conform celor de mai sus, o posibilă structură a unei documentaţii de acest gen este:

- copertă asemănătoare cu cele din etapele precedente, doar denumirea diferind; aceasta poate fi "Caiet de sarcini pentru teste de acceptanţă/ sistem/ integrare" sau Lista testelor de acceptanţă/ sistem/ integrare" sau pur şi simplu "Teste de acceptanţă/ sistem/ integrare"; diferenţa dintre teste va fi cea cunoscută de la curs:
  - la teste sistem se folosesc metode de simulare a procesului (eventual chiar standuri, adică mici ansambluri/ echipamente ad-hoc, realizate special pentru necesităţile testării
  - la teste de integrare procesul va fi cel real
  - la testele de acceptanţă, pe copertă trebuie neapărat să figureze repezentantul legal al clientului (la celelalte, în special la testele sistem, nu e necesară prezenţa clientului)
- colectiv de realizare
- cuprins
- memoriu tehnic, care să cuprindă secţiuni alcătuite aproximativ după schema scenariului de testare descris mai sus:
  - Secţiunea 1: Generalităţi
    1.1. Denumire
    1.2. Istoric
    1.3. Cod:
    1.4. Scurtă descriere
    1.5. Structura (arhitectura) sistemului (atenţie, nu a programelor !)
    1.6. Funcţiile sistemului
  - Secţiunea 2: Mijloace de verificare

    Vor fi descrise toate mijloacele tehnice necesare demonstrării funcţionalităţilor, atât echipamente şi sisteme de calcul, inclusiv precizarea configuraţiilor, cât şi eventuale standuri de testare. De asemenea, vor fi enumerate aplicaţiile necesare, sistemele de operare, sistemele de gestiune de baze de date etc.

    Spre exemplu: Pentru realizarea testelor conform documentului de faţă, sunt necesare următoarele mijloace tehnice de verificare:

    2.1. Sistem de calcul PC cu min. 512 Mb RAM, 2 Gb HDD, min 2.2 GHz, sistem de operare QNX vers. 3
    2.2 Stand de simulare conform Anexei 2.
    2.3. Sistem de gestiune a bazelor de date Microsoft SQL Server 2005.

instalat pe sistem de calcul PC cu min 1 Gb RAM, 40 Gb HDD, min. 2.2 GHz, sistem de operare Windows XP

2.4. Structura de interconexiuni în reţea Ethernet conform figurii următoare

etc.

- Secţiunea 3: Condiţii de funcţionare
  Aici, vor fi descrise, punct cu punct, toate funcţionalităţile care vor fi testate.

  Spre exemplu:

  3.1. Sistemul va fi capabil să semnalizeze evenimentele de tip avarii prin mesaje specifice generate pentru fiecare avarie, afişate în fereastra de evenimente a aplicaţiei de monitorizare (vezi fig. 22, pag. 16, din Specificaţii)

  3.2. Sistemul va fi capabil să permită efectuare de telecomenzi prin selectarea pe schema sinoptică a obiectului dorit (vezi figurile 31-34, pag. 22, din Specificaţii)

  etc.
- Secţiunea 4: Condiţii de verificare
  Aici, vor fi reluate, punct cu punct, condiţiile de funcţionare de la secţiunea 3 şi va fi explicat cum anume se poate demonstra concret corectitudinea funcţionalităţii respective.

  Spre exemplu:
  Aplicaţia de monitorizare-telecomandă se lansează prin click pe shortcut corespunzător pe Desktop-ul sistemului de la 2.1. În continuare:

  4.1. Funcţionalitatea de la pct. 3.1 poate fi verificată prin aceea că utilizând simulatorul de proces de la pct. 2.2, prin apăsarea butoanelor 1-4 de pe simulator, în fereastra de evenimente indicată vor avare mesajele 5-9 din tabelul din Anexa 5 din Specificaţii, pag. 39); înâ'rzierea maximă admisibilă dintre acţionarea butoanelor şi apariţia mesajelor indicate este de 4,5 secunde.

  4.2 Funcţionalitatea de la pct. 3.2 poate fi verificată prin aceea că se expdiază telecomenzi prin click pe schema sinoptică din fig.33, pag.22 din Specificaţii, pe obiectele numite IC492, IN484 sau SN421; ca urmare, pe panoul simulatorului de proces menţionat la 2.2, se vor aprinde sau stinge led-urile numerotate cu 1-3, în funcţie de starea anterioară. Întârzierea maximă permisă este de 1,5 secunde.

  etc.

Proiect – ASPECTE FINALE

1. Se vor preda şi prezenta:
   - ***Specificaţii***
   - ***Documentaţia de Proiectare***
   - ***Teste de Acceptanţă*** (Caietul de Sarcini)
   - ***Planul de Marketing***
   - ***Dosarul de Management al Proiectului***
   - ***Prezentarea ppt***
   - ***Prototipul funcţional***

2. Conţinutul ***Dosarului de Management al Proiectului***:
   - Coperta (cu semnături de la Programatorul-şef, ajutor, secretar)
   - Cuprins
   - Echipa – calificare, atribuţii în cadrul echipei
   - Diagrama Gantt (preferabil elaborată cu *Microsoft Project*)
   - Tabel de sarcini (taskuri) în cadrul proiectului (preferabil elaborat cu *Microsoft Project*)
   - Lista documentelor întocmite
   - Pontaj (contribuţia efectivă la realizarea proiectului - date şi ore –pentru fiecare din echipă)
   - Proces-verbal de avizare a proiectului (cu acordarea punctajului- - exemplu în anexă)
   - Proces-verbal cu atribuirea punctajului în cadrul echipei (exemplu în anexă)

3. Conţinutul ***Planului de Marketing***:
   - Coperta
   - Colectivul de realizare
   - Cuprins
   - Calculul de stabilire a costurilor
   - Segmentul-ţintă
   - Comparaţie cu produsele concurente (performanţe, preţuri)
   - Propunere de preţ cu justificare
   - Estimarea volumului total de vînzări
   - Preliminarea vînzărilor pe 2 ani
   - Proiectarea campaniei de marketing
     i. metode
     ii. conţinut
     iii. mod de derulare (ce acţiuni, cînd - eventual cu ce periodicitate, cu ce materiale promoţionale etc.)
   - Materiale promoţionale
   - Analiza riscurilor şi a modului de minimizare a acestora
   - Concluzii referitoare la returnarea investiţiei (aici se vor lua în calcul şi costurile de marketing precum şi riscurile)

4. Conţinutul **prezentării** ppt:(< 30 secunde / slide, 10 minute)
   - Slide 1: titlul proiectului, echipa
   - Slide 2 – obiectivele :
     a. Didactice
        i. antrenament pentru munca în echipă
        ii. dezvoltarea abilităţilor de comunicare
        iii. însuşirea conţinutului şi practicii etapelor de realizare a aplicaţiilor software

b. profesionale
    i. realizarea…, specificînd etapele de proiectare, modul de încadrare într-un proiect mai mare precum şi punctul de pornire
    ii. îmbunătăţirea dialogului cu clientul prin realizarea unui prototip funcţional
    iii. pregătirea produsului pentru vînzare prin identificarea unei strategii de marketing

- Slide 3: Arhitectura generală (desen), cu specificarea părţii care revine echipei
- Slide 4: Funcţiile sistemului
- Slide 5: Tehnologia propusă (nu pentru prototipul funcţional ci pentru aplicaţia finală) cu justificare (inclusiv sistem de operare şi medii de dezvoltare)
- Slide 6: Arhitectura aplicaţiei
- Slide 7: Structuri de baze de date (schema relaţională)
- Slide 8: Comunicarea în sistem (structuri şi protocoale de comunicare)
- Slide 9: …Interfeţe utilizator (prezentare de la general la particular)
- Slide … - Soluţii tehnice (cîteva, considerate mai interesante, inclusiv fragmente mici de cod-sursă)
- Slide ... Prezentarea prototipului funcţional (2-3 slide-uri, cu sublinierea posibilităţilor de demonstrare a funcţionalităţilor)
- Slide . Prezentarea standului de testare propus şi a strategiei de testare
- Slide . Diagrama Gantt
- Slide … Strategia de Marketing, diagrame, preţuri, riscuri etc.
- Slide . : Concluzii:
    i. Ce s-a realizat
    ii. De ce e mai bun decît alte produse similare
    iii. Strategii de viitor
- Ultimul slide: Vă mulţumim pentru atenţie

Recomandări:

- *culori sobre, lipsite de stridenţă (eventual pastel)*
- *max. 2 culori de fundal*
- *fonturi mari, vizibile, simple*
- *max. 3 mărimi de fonturi*
- *max. 2 culori de fonturi (una pentru accentuare)*
- *texte scurte, mai ales enumerări, fără dezvoltări*
- *multe imagini, diagrame, scheme etc.*
- *animaţie dar nu sacadată (s. ex. nu literă cu literă sau care durează prea mult)*
- *comentariul va fi bogat şi diferit de textele afişate, pe care le va completa*
- *dicţie limpede, trebuie vorbit tare, repede, cu privirea spre auditoriu*
- *răspunsul la întrebări la obiect, scurt, rapid, fără pauză lungă de gîndire, nu trebuie iniţiate polemici*

5. ***Şedinţa de avizare*** va fi în _____ 2020, ora _____, conform programului afişat cu o săptămână înainte.

Prof. dr. ing. Vasile Stoicu-Tivadar

Timişoara, 01.02.2020

Antet firma de software
…

Proces-Verbal
de avizare a proiectului
„….”

Încheiat azi _____2020 la sediul ….(firma de software) cu ocazia şedinţei de prezentare şi avizare a proiectului « …. », realizat de echipa de proiectare condusă de programatorul-şef ……..... şi avînd un efectiv total de …….. proiectanţi.

Întrunită în şedinţă de analiză şi avizare a proiectului menţionat, comisia de avizare a hotărît că proiectul a fost admis / respins cu un punctaj de
…….. puncte.

Comisia de avizare:

Nr. crt.	Poziţia în comisie	Funcţia	Nume, prenume	Semnătura
1	Preşedinte		Vasile Stoicu-Tivadar	
2	Membru		Norbert Gal-Nădășan	
3	Membru		Mihaela Vida-Crișan	
4	Membru		Oana-Sorina Chirila	
5	Membru		Stelian Nicola	

Antet firma de software
…

Proces-Verbal
de atribuire a punctajului în cadrul echipei, pentru proiectul
„….”

Încheiat azi _____2020 la sediul ….(firma de software) cu ocazia şedinţei de prezentare şi avizare a proiectului « …. », realizat de echipa de proiectare condusă de programatorul-şef ……..…….. şi avînd un efectiv total de …….. proiectanţi.

Întrunită în şedinţă de atribuire a punctajului, echipa de proiectare a hotărît că punctajul total acordat de către comisia de avizare, de ..…… puncte, a fost împărţit între membrii echipei după cum urmează:

Nr. crt.	Funcţia în cadrul echipei	Nume, prenume	Punctajul	Semnătura
1				
2				
3				
.				
.				
.				

(Obs.: punctajul total trebuie să fie egal cu suma punctajelor din tabelul de mai sus).

Proiect la *IP anul 3 AIA* An univ. 2019-2020 / semestrul 2	Titular: *Stoicu-Tivadar Vasile*

Fişa de avizare pentru proiectele de *Ingineria Programării*

Nr. crt.	Proiect	Nume, prenume Progra-mator-şef	Punctaj							Total	Obs
			Docu-mentaţie	Aplicaţie	Ansamblu	Conţinut	Prezen-tare	Marke-ting	Răspuns la întrebări		
			[0..2]	[0..2]	[0..2]	[0..2]	[0..0.5]	[0..1]	[0..0.5]		
1											
2											
...					...						...
12											

Membru Comisia de Avizare: _____

Data: _____
Semnătura: _____