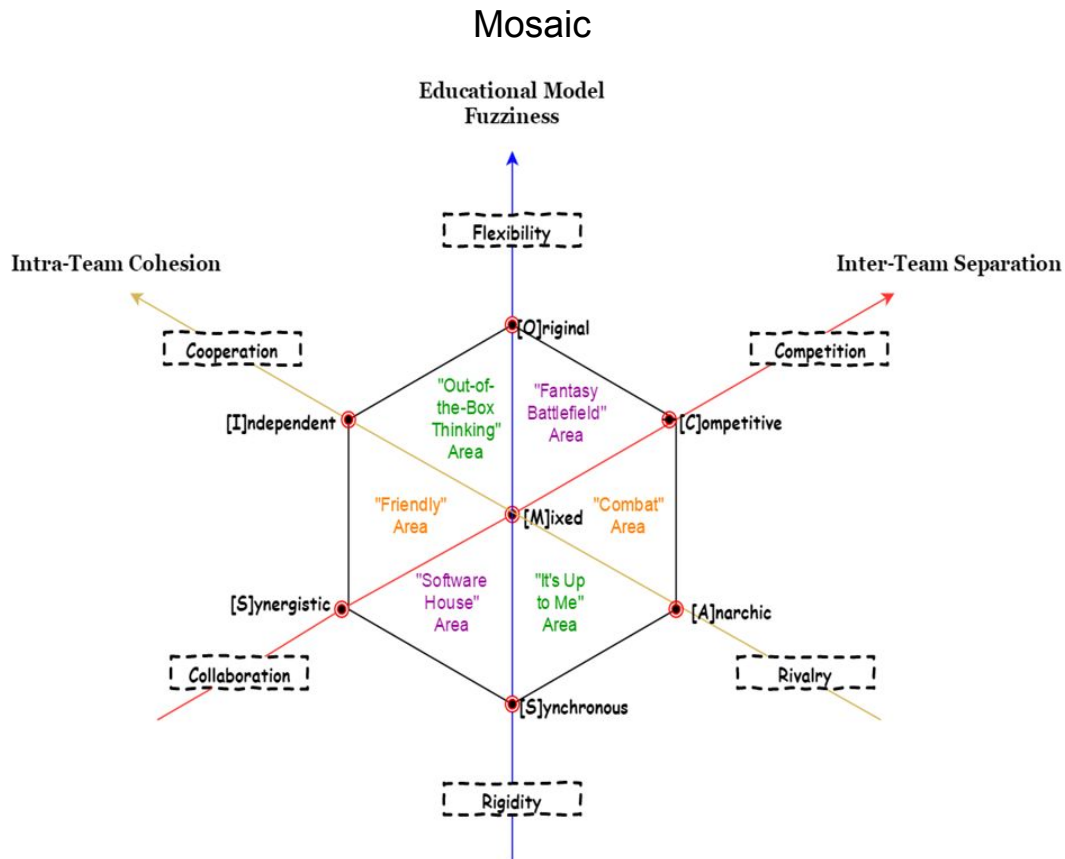


# Cursul 0

## Introducere



### Abordarea Independentă ("Independent")

- Solicitare Proiect: Scopul și conținutul documentului; Descrierea sumară a produsului software; Tehnologii și tooluri de dezvoltare; Estimări resurse necesare; Prezentare calendar livrări; Motivare echipă. etc.

### Abordarea Concurențială ("Competitive")

- Realizarea unui sistem de simulare a competițiilor între mașini de Formula 1.

### Abordarea Colaborativă ("Synergistic")

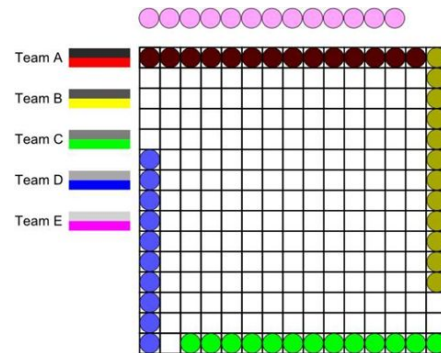
- Sistem Automat de Analiză și Extragere a Informației din Documente

### Abordarea Mixtă ("Mixed")

- Construirea unui sistem de prelucrare de imagini bazat pe o strategie de votare: fiecare echipa era împartită în patru celule: 3 celule care ofereau trei algoritmi diferiți de conversie și o celulă care combina cele trei rezultate într-unul mai bun decât oricare din cele trei rezultate inițiale.

### Abordarea Anarhica ("Anarchy")

- Scopul jocului este ca un pion sa ajunga pe partea opusa a tablei, moment in care echipa din care face parte pionul (cea la care a fost mutat) castiga un punct (creste nota echipei respective). Toti pionii care fac parte din echipa castigatoare mai castiga un punct si la nota individuala iar jocul se reia, cu jucatorii realocati in echipe diferite.



### Abordarea Artistica ("Originality")

- Scopul proiectului este de a obtine o imagine cu o valoare "artistica" de un cat mai ridicat nivel de exprimare utilizand doar generare procedurala, geometrie fractalica, si algoritmica simpla.

### Abordarea Sincrona ("Synchronous")

- Primul proiect este in saptamanile 3-6, iar al doilea in saptamanile 7-10

## Cursul 1

## Introducere

### Cuprins:

1. Dimensiunea unui proiect software
2. Planificarea proiectului
3. Execuția proiectului
4. Închiderea proiectului
5. Procesul de dezvoltare
6. Particularitățile proiectelor software

### 1. Dimensiunea unui proiect software

Un proiect software are două dimensiuni principale:

- Ingineria proiectului (se ocupă cu dezvoltarea efectivă a proiectului si se concentrează pe aspecte precum design, cod, testare)
- Managementul proiectului -> planificarea și controlul activităților de inginerie în scopul atingerii obiectivelor proiectului (costuri, timpi de execuție, calitate)

### Proiecte mici:

- Echipe formate dintr-un număr redus de persoane, durată de câteva săptămâni

- Metode informale de management și dezvoltare: email-uri, câteva termene limită, comunicare verbală

Proiecte mari:

- Echipa mari; durată câteva luni
- Taskuri efectuate cu atenție, planificate și urmărite pas cu pas; metode bine cunoscute
- Fiecare produs intermediar este documentat riguros și verificat

Procesul de management al proiectului are trei etape principale: planificarea, executia, inchiderea proiectului.

## **2. Planificarea proiectului**

- Activități administrative și de pornire
- Planificarea și orarul proiectului
  - Definirea obiectivelor proiectului
  - Estimarea costurilor și a efortului
  - Definirea unui plan de masurare a proiectului
  - Identificarea riscurilor și a modului de evitare/recuperare
- Obținerea acordului de la managementul superior
- Definirea și revizuirea planului de management al configurațiilor
- Realizarea unei echipe și stabilirea responsabilităților fiecăruia

## **3. Execuția proiectului**

- Execuția proiectului după planul propus
- Monitorizarea conformității cu procesele definite
- Analiza defectelor și efectuarea de activități de prevenire a acestora
- Monitorizarea performanțelor la nivel de program și a progresului proiectului
- Efectuarea de review-uri la anumite etape critice și replanificarea unor etape dacă este necesar

## **4. Închiderea proiectului**

- Etapa are loc după ce clientul și-a dat acceptul pentru produsul final
- Se urmărește stabilirea unor concluzii ca urmare a experienței acumulate, pentru a îmbunătăți procesele folosite în viitor

## **5. Procesul de dezvoltare**

Principiile fundamentale în MPS:

- Procesul de dezvoltare bazat pe arhitectură
  - Componentele arhitecturale - înțelese foarte bine înainte de a lua în considerare amănunțele de detaliu
  - Gradul de refacere/abandon a unor componente – ar trebui să scadă sau să rămână constant în timpul desfășurării unui proiect
- Modul de dezvoltare iterativ
  - Framework de planificare cât mai dinamic
  - Rezolvarea problemelor critice foarte devreme => Dezvoltare mai predictibilă & mai puține surprize => Management al riscului mult mai bun

- Principalele riscuri confruntate cat mai devreme.
  - La fel ca la modul de dezvoltare iterativ???
- Dezvoltarea bazată pe componente
  - Complexitatea dezvoltării de software ~ numărul de elemente generate de către membrii echipei
  - Diminuarea numărului acestora și a complexității procesului de management
- Plan de management al schimbărilor
  - Dinamica dezvoltării iterative => fluxurile de lucru concurente ale diferitelor echipe de dezvoltare care folosesc aceleași componente
  - Necesită linii de referință controlate foarte riguros
- Model de evaluare bazat pe demonstrații
  - Integrarea apare foarte devreme în viața unui proiect și se continuă pe parcursul întregului proces de dezvoltare.
  - Rezultatele intermediare sunt elemente esențiale, deoarece sunt tangibile și obiective
- Evaluare obiectivă a calității și corectă a progresului
  - Indicatorii de progres și calitate derivă direct din componentele dezvoltate și conferă informații importate în legatura cu trendul proiectului și gradul de corelare al produsului cu cerințele inițiale
- Notății bazate pe modele
  - Utilizarea unor notații ingineresti în faza de design va conduce la un control mai bun al complexității, evaluări intermediare mai obiective și mai corecte, precum și analize ce pot fi automatizate
- Procesul de dezvoltare configurabil și scalabil economic
  - Experiența, metodele, uneltele și tehnicile trebuie folosite împreună pentru a lărgi segmentul de piață țintă => o întoarcere a investiției mult mai mare
- Versiunile intermediare având nivele de detaliu din ce în ce mai mari

## 6. Particularitățile proiectelor software

- Invizibilitate - spre deosebire de un pod sau un drum care sunt construite și progresul este vizibil imediat, în cazul unui produs software progresul nu este evident foarte repede
- Complexitate - Produsele software sunt unele dintre produsele cu cea mai mare complexitate per euro/dolar/lei investiți
- Flexibilitate - Ușurința cu care un produs software poate fi modificat este unul dintre cele mai importante atu-uri ale acestui tip de proiecte

# Cursul 2

## Vedere de ansamblu asupra managementului de proiect

Cuprins:

1. Definiții
2. PMI – Organizarea Proiectului
3. Procese de Management al Proiectelor
4. Noțiuni de etică profesională

### 1. Definiții

- Operațiune – activitate continuă și repetitivă
- Program – grup de proiecte gestionate într-un mod coordonat
- Faza a unui proiect – o colecție de activități interconectate din punct de vedere logic, care de obicei conduc la realizarea unui produs intermediar/final important
- Tehnica Delphi – tehnica de previziune folosită pentru a aduna informații despre evenimentele viitoare din viața proiectului; se bazează pe părerile unor experți
- Managementul prin obiective – sistem de conducere managerială care definește responsabilitățile unui manager pe baza obiectivelor urmărite de către organizație
- Ciclul de viață al unui proiect – o serie de faze ale unui proiect, care de obicei urmează una după alta, alea căror nume și număr depinde de cerințele tuturor organizațiilor implicate în proiect

### 2. PMI – Organizarea Proiectului

PMI – Project Management Institute

Structuri organizatorice (din punctul de vedere al nivelului de autoritate al managerului de proiect):

- Funcțională:
  - Organizarea se bazează pe diferite arii de expertiză (marketing, producție etc.)
  - Fiecare angajat are un superior bine determinat
  - Managerul de proiect are putere scăzută; responsabilitatea revine managerilor funcționali
- Proiectizată (“Projectized”)
  - Întreaga organizare se bazează pe proiecte
  - Managerul de proiect deține controlul total asupra proiectelor
  - Fiecare angajat este repartizat la un proiect și răspunde direct managerului de proiect
- Matriceală
  - Slabă
    - Deciziile sunt luate în principal de managerul funcțional
    - Un manager de proiect poate îndeplini unul din următoarele două roluri:
      - Mesager: coordonează comunicarea din echipa; nu poate lua decizii
      - Coordonator: are puterea de a lua anumite decizii;
  - Puternică: deciziile sunt luate în principal de managerul de proiect
  - Echilibrată

### 3. Procese de Management al Proiectelor

Procesele se impart in:

- Procese de management - asigură execuția eficientă a proiectului
- Procese orientate pe produs - specifică și determină crearea produsului final

Grupuri de procese:

- Procese de inițializare
  - Reprezintă faza în care se obțin autorizațiile necesare începerii unui nou proiect sau a unei noi faze în cadrul unui proiect
- Procese de planificare
  - Grupul de procese ce țin de formularea obiectivelor proiectului
  - În această etapă se creează planul de proiect – se stabilește în detaliu modul în care vor fi atinse obiectivele propuse
- Procese de execuție
  - Managerul de proiect are responsabilitatea de a coordona personalul și resursele disponibile astfel încât să respecte planul de proiect stabilit
- Procese de monitorizare și control
  - Grup de procese responsabile cu măsurarea și analiza performanțelor proiectului
- Procese de încheiere
  - Grupul de procese necesare pentru a termina formal un proiect
  - Produsul este livrat și acceptat de către beneficiar, iar proiectul se încheie

### 4. Noțiuni de etică profesională

Responsabilitățile față de profesie

- Să fie sincer tot timpul, indiferent de situație
- Să nu ascundă eventuale conflictele de interese
- Să respecte legislația în vigoare
- Să respecte drepturile intelectuale ale altora

Responsabilitățile față de clienți și față de public

- Să își păstreze integritatea din punct de vedere profesional
- Să respecte caracterul confidențial al datelor personale
- Să evite primirea unor cadouri/compensații în situațiile în care acest lucru este inadecvat
- Să se asigure că niciun conflict de interese nu mijlocește interesele clientului sau judecata profesională

# Cursul 3

## Faza de planificare

Cuprins:

1. Planul de Proiect
2. Planificarea pas cu pas
3. Metode de dezvoltare
4. Prototipuri
5. Metode ale generației a 4-a
6. WBS
7. RBS

### 1. Planul de Proiect

Structura:

- Sumarul proiectului
- Secțiunea de planificare
  - Modul de execuție al diferitelor proceduri de planificare
  - Modul de dezvoltare ce va fi folosit, estimarea timpilor de execuție etc.
- Secțiunea de urmărire (tracking)
  - Măsurătorile ce vor fi făcute în timpul proiectului
- Secțiunea destinată echipei
  - Structura și membrii echipei, rolurile lor

#### Informații esențiale în planul de proiect:

	Obiectivele proiectului
	Procesul de dezvoltare folosit
	Modul de management
	Estimarea efortului
	Punctele de control intermediar (Milestones)
	Planul de management al riscului
	Controlul calității
	Planul de urmărire și verificare al proiectului
	Organizarea echipei
	Modul de rezolvare a eventualelor conflicte în cadrul echipe și/sau cu clientul

### 2. Planificarea pas cu pas

Pasul 0 – Alegerea proiectului

- În această etapă au loc activități ce duc la luarea unei decizii în legatură cu proiectele ce vor fi începute – această decizie poate fi luată individual sau poate să facă parte dintr-o strategie pe termen lung a companiei

Pasul 1 – Identificarea domeniului și a obiectivelor proiectului

- Modificarea obiectivelor în lumina analizei asupra persoanelor interesate în proiect
- Stabilirea metodelor de comunicare cu toate părțile interesate

Pasul 2 – Identificarea infrastructurii

- Identificarea modului în care va fi organizată echipa de dezvoltare
- Deciziile strategice sunt de obicei documentate fie într-un plan de strategie business, fie într-un plan tehnologic dezvoltat pe baza planului business

Pasul 3 – Analiza caracteristicilor

- Identificarea celor mai importante riscuri

- Analiza modului de implementare, selectarea ciclului de viață folosit pentru dezvoltare
- Revizuirea estimărilor asupra resurselor

Pasul 4 – Identificarea produselor și a activităților

- Documentarea eventualelor probleme ale produsului
- Modificarea rețelei de activități, luând în considerare nevoia pentru etape intermediare și puncte de verificare

Pasul 5 – Estimări ale efortului pentru fiecare activitate

- Efectuarea de estimări folosind o abordare de jos în sus
- Estimări de personal, timp, resurse

Pasul 6 – Analiza riscurilor

- Identificarea și cuantificarea riscurilor datorate activităților
- Ajustarea planurilor și a estimărilor astfel încât să ia în considerare riscurile identificate anterior

Pasul 7 – Alocarea resurselor

- Identificarea și alocarea resurselor
- Revizuirea planurilor și a estimărilor, astfel încât să ia în considerare constrângerile datorate resurselor

Pasul 8 – Revizuirea/Publicarea planului

Pasul 9 – Execuția planului

Pasul 10 – Nivele inferioare de planificare

### 3. Metode de dezvoltare

Metode structurate (inclusiv metodele OO)

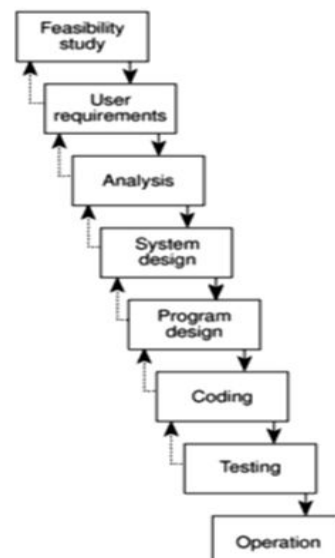
- O mulțime de pași și reguli care generează prodiagramele de flux, de date, etc.
- De cele mai multe ori sunt mult mai consumatoare de timp decât metodele intuitive, acest lucru ducând și la o creștere a costurilor proiectului
- Avantaje: sistemul este mult mai puțin sensibil la erori și mult mai ușor de întreținut la sfârșit
- Recomandate în cazul proiectelor mari, care implică mulți dezvoltatori și mulți utilizatori

Metode de dezvoltare rapidă

- Workshop-uri de trei-cinci zile în care dezvoltatorii lucrează intensiv împreună cu clienții pentru a identifica și pentru a cădea de acord asupra cerințelor business ale proiectului
- Time-box – întinderea fiecărei etape a proiectului este constrânsă de un deadline predeterminat, foarte scurt și inflexibil
- Cerințele ce nu pot fi satisfăcute într-un anumit time-box, sunt mutate în etapele următoare

Modelul în cascadă

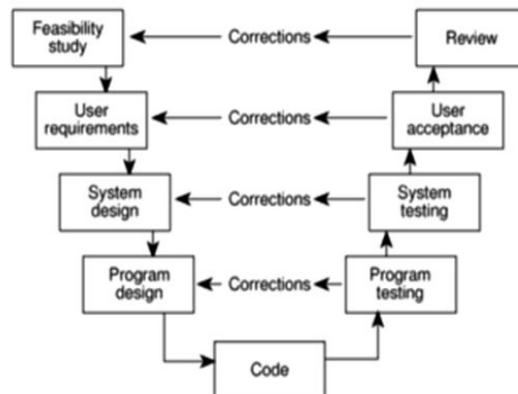
- Considerat metoda “clasică” de dezvoltare a sistemelor
- Permite controlul eficient al proiectelor și estimarea foarte exactă a timpilor de execuție





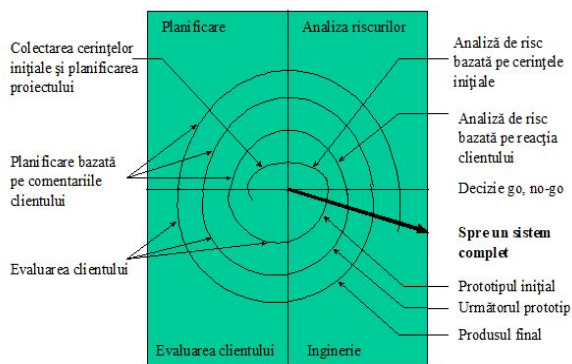
### Modelul procesului în V

- Extinde activitățile de testare din modelul în cascadă
- Fiecare pas are un proces de validare corespunzător.
- În cazul în care apar defecte, procesul de validare întoarce dezvoltarea la pasul de dezvoltare corespunzător; toți pașii următori trebuie să refacă.
- Ideal, acest tip de feed-back ar trebui să apară numai în cazul unei discrepante mari între specificațiile unei anumite activități și ceea ce a fost de fapt implementat



### Modelul în spirală

- Poate fi considerat ca o altă vedere a modelului în cascadă
- Un mai mare grad de detaliu este necesar la fiecare etapă a proiectului, acest fapt justificând și un mai mare grad de încredere în probabilitatea de succes a proiectului
- Acest model poate fi văzut ca o spirală în care sistemul dezvoltat este văzut din ce în ce mai în detaliu la fiecare rotație
- Un proces de evaluare a etapei precedente are loc înaintea începerii unei noi iterații
- Dezvoltare iterativă
  - Bazată pe ideea de ciclu de producție
  - Procesul de dezvoltare cuprinde mai multe cicluri de producție
- Dezvoltare incrementală
  - Fiecare ciclu are o complexitate (un nivel de detaliere) mai mare decât precedentul
- Activitățile unui ciclu de producție
  - (1) Planificare
    - Stabilirea obiectivelor, alternativelor de rezolvare și a restricțiilor pentru ciclul curent
  - (2) Analiza riscurilor
    - Analizează alternativele de rezolvare și restricțiile din (1)
    - Identifică factorii de risc
    - Decizia GO/NO GO (continuă/renunță)
  - (3) Inginerie - începutul unui ciclu nou
    - Dezvoltarea produsului pe următorul nivel de detaliere
    - Se pot folosi
      - Modelul clasic
      - Prototipizarea - pentru clarificarea unor cerințe
  - (4) Evaluarea clientului



- Avantaje
  - Abordare evoluționistă
  - Ajută la înțelegerea riscurilor și la identificarea modalităților de ținere sub control a acestora
  - Prototipizarea este folosită ca mecanism de reducere a riscurilor
  - Ciclul clasic de viață este încorporat într-un cadru iterativ, care reflectă mai bine lumea reală
- Dezavantaje
  - Analiza riscurilor este o activitate critică
  - Atenție acordată riscurilor tehnice în toate etapele proiectului

#### 4. Prototipuri

Tipuri de prototipuri:

- Throw-away
  - Folosit doar pentru a testa unele idei; se renunță la el în momentul în care începe dezvoltarea sistemului operațional
- Evoluționar
  - Este dezvoltat și modificat în continuu până în momentul în care poate deveni un sistem operațional
- Incremental
  - Sistemul operațional este dezvoltat și implementat în etape mici; feed-back-ul de la etapele anterioare este folosit și influențează dezvoltarea etapelor următoare

Activități:

- (1) Colectarea cerințelor
  - Dezvoltatorul și utilizatorul stabilesc obiectivele generale, cerințele cunoscute, domeniile în care cerințele vor fi definite ulterior
- (2) Producerea rapidă a unui proiect
  - Se reprezintă acele elemente care sunt percepute de utilizator
    - Formatul datelor de intrare
    - Formatul rezultatelor
- (3) Construirea prototipului
- (4) Reevaluarea prototipului de către utilizator
- (5) Rafinarea prototipului
- (6) Realizarea produsului final

Activitățile (3) - (5) se repetă până când sunt satisfăcute toate cerințele clientului

Avantajele utilizării prototipurilor:

- Comunicarea este îmbunătățită:
- Când nu există un sistem care poate fi imitat, clienții pot testa diferite prototipuri pentru a își da seama care dintre ele le este cel mai util
- Necesarul de documentație este redus datorită faptului că prototipul poate fi examinat în practică
- Costurile de întreținere sunt reduse; dacă clientul nu cere multe schimbări ale prototipului, este foarte probabil ca acesta să nu ceară nici multe schimbări ale produsului final
- Clienții pot fi mult mai implicați în deciziile legate de design-ul final al sistemului

## 5. Metode ale generației a 4-a

### 4GL - Fourth Generation Languages

#### Instrumente CASE Computer-Aided Software Engineering

- Specificarea cerințelor se face folosind limbaje de specificare apropiate de limbajul natural SAU folosind notații matematice (algebrice)
- Sprijin pentru modelare, inclusiv teste de consistență și validitate
- Traducerea automată a specificațiilor în cod sursă (forward engineering), trecându-se prin nivele de
  - Analiză - modele de analiză
  - Proiectare - modele de proiectare

#### Activități:

- (1) colectarea cerințelor
  - Ideal: clientul descrie cerințele folosind limbajul de specificare al instrumentului CASE
  - În realitate: dialog între client și specialistul în specificarea cerințelor
- (2) Proiectare
  - Include elaborarea de modele pentru analiză și proiectare
  - Este nevoie de o reprezentare a modelelor care să permită generarea automată de cod
- (3) Implementare folosind generarea automată de cod sursă
- (4) Testarea

#### Avantaje

- Productivitate ridicată
- Întreținere ușoară a programelor DACĂ
- Cerințele sunt formulate corect
- Activitatea de proiectare este bine structurată

#### Dezavantaje

- Curba de învățare a folosirii instrumentelor este lungă
- Codul generat nu este întotdeauna și eficient
- Costuri de achiziție/întreținere foarte mari
- Probleme de migrare, comunicare cu alte instrumente similare
- Problemă deschisă: întreținerea sistemelor mari

## 6. WBS – Work Breakdown Structures

- Liste de task-uri detaliate
- O decompoziție a muncii necesare dezvoltării unui proiect în bucați din ce în ce mai mici până la nivelul la care există suportul pentru o urmărire detaliată a progresului la care se afla proiectul
- Fiecare pas elementar din decompoziție va avea
  - un cost
  - o estimare muncii individuală
- Eforturile aferente și costurile activităților de la nivelele superioare sunt calculate pur și simplu prin însumarea eforturilor și costurilor activităților din care sunt compuse
- În momentul în care lista de task-uri, împreună cu estimările aferente, este aprobată, estimările de cost devin bugetul proiectului
- Exemple: pentru proiectele ce durează între 4 și 12 luni, activitățile trebuie descompuse în general până când nivelul cel mai de jos reprezintă aproximativ 1 sau 2 săptămâni de muncă; pentru proiectele mai lungi se poate ajunge și până la 4 și chiar 8 săptămâni

## 7. RBS - Resource Breakdown Structures

- Liste de resurse detaliate
- Sunt similare listelor de task-uri detaliate, dar se referă la organizație, echipa de dezvoltatori și personalul implicat în realizarea produsului
- Relația cea mai importantă ce este documentată prin intermediul RBS este aceea a autorității: cine răspunde în fața cui și cine spune cui ce să facă
- Nu este importantă poziția în cadrul organizației a fiecărui individ în parte (se urmărește documentarea faptului că angajatul X răspunde în fața managerului Y, indiferent dacă X este secretară, unul dintre programatori sau un alt manager)

# Cursul 4

## Managementul bugetului

Cuprins:

1. Estimarea Costurilor
2. Tipuri de estimări
3. Riscuri
4. Clasificarea costurilor
5. Optimizarea costurilor
6. Bugetul proiectului
7. Managementul costurilor
8. Subproces PCM

### 1. Estimarea Costurilor

Estimarea costurilor unui proiect reprezintă în mare bugetul aceluia proiect

### 2. Tipuri de estimări

Estimarea de bază

- Costurile pe activități și posturi de execuție
- Precizie de 10%
- Riscurile sunt evaluate => un buget al obiectivelor care va fundamenta decizia de investiție în proiect

Estimarea detaliată

- Precizie de 5%
- Permite evaluarea unei oferte de prestație care se identifică cu oferta clientului
- Este riguroasă, analitică; consumă timp și bani

Estimări exacte

- Estimare corectă a costurilor <= luate în considerare următoarele elemente:
  - Domeniul produsului - cerințele și caracteristicile produsului software
  - Domeniul proiectului - cerințele proiectului, constrangerile, modalitățile de control, etc.
  - Diverse presupuneri - disponibilitatea tuturor resurselor necesare, întârzieri, etc.
  - Constrangeri: de timp, referitoare la resurse, de buget, ale mediului de dezvoltare, etc.

### 3. Riscuri

- Depășirea bugetului prevăzut: activități suplimentare
- Stagnarea activităților: probleme în aprovizionare, subcontractări neonorate la timp
- Neîndeplinirea obiectivului propus la nivelul performanțelor angajate
- Repartiția necorespunzătoare a bugetelor alocate
- Necorelarea bugetelor acestora cu termenele sau problemele tehnologice
- Estimarea necorespunzătoare a volumului și distribuției bugetului necesar

### 4. Clasificarea costurilor

Criterii de clasificare:

- Câmpul de aplicație al costurilor
  - Costuri pentru funcția economică : cercetare, aprovizionare, producție, vânzare, etc.
  - Costuri în funcție de zona de exploatare: costul pentru locul de muncă, de uzină etc.
  - Costuri pe activitatea de exploatare : produse, familii de produse
  - Costuri pe centre de responsabilitate: studii de piață, dezvoltare, industrializare
  - Alte costuri (pe departamente, pe clienți, etc.)
- Conținutul costurilor
  - Costul manoperei directe: valoarea dată direct personal
  - Costul manoperei indirecte: valoare prin care se recompensează personalul auxiliar
  - Costul managementului proiectului
  - Costul componentelor necesare pentru realizarea proiectului (de la prototip până la produs final)
  - Cost primar - constituit ca suma tuturor costurilor directe (manoperă și materiale) utilizate pentru realizarea proiectului
  - Costuri de amortisment al echipamentelor utilizate în proiect
  - Costul de achiziție a echipamentelor necesare proiectului
  - Costuri variabile (proporțional cu volumul producției): salarizarea directă, materiile prime, materialele, energia
  - Costuri indirecte: costuri de regie, sumele prevăzute pentru administrarea proiectului, asigurarea dotărilor, întreținere, iluminat, etc.
  - Costuri fixe (costuri care rămân neschimbate și se efectuează indiferent de volumul producției realizate): chirii, taxe, asigurări
  - Administrație
  - Costuri de contract - costuri care regroupează o serie de angajamente efectuate în cadrul proiectului de o serie de factori externi.
- Momentul de calcul al costurilor
  - Costul constatat, un cost istoric și real calculat a posteriori
  - Costul prestabilit, calculat apriori, înainte de a demara activitățile, un cost previzionat, care susține decizia.

## 5. Optimizarea costurilor

Optimizarea costurilor pe relația client/furnizor, comportă patru etape de negociere:

- Compromisul nevoie/funcțiune:
  - utilizatorul își exprimă dorințele în termeni de performanțe, termene, cost de achiziție (și utilizare)
  - furnizorul formulează o propunere cu cel mai bun cost global
- Compromisul funcțiunii de serviciu/concepția produsului:
  - proiectantul caută soluții tehnice conceptuale mai puțin costisitoare pentru performanțele dorite de utilizator
- Compromisul proiectant-executant/furnizori:
  - se obține din partea furnizorilor: subcontractori, materiale, echipamente
  - prețul cel mai competitiv posibil
- Compromisul concepția produsului/realizare:
  - executantul se străduiește să optimizeze costurile de producție prin planificarea sarcinilor și gestionarea economică a resurselor

## 6. Bugetul proiectului

Bugetul proiectului reprezintă suma pusă la dispoziția echipei de proiect pentru a realiza conform exigențelor caietului de sarcini respectând calitatea și termenele impuse

Bugetul inițial conține:

- Sarcinile, costurile și eventualele rezerve
- Liniile de bugetare
- Modul în care se urmăresc costurile
- Abaterile tehnice permise și suporturile bugetare suplimentare aferente acestora

## 7. Managementul costurilor de proiect MCP

MCP va include

- procese adiționale
- tehnici numeroase de management general cum ar fi contul de profit și pierderi, elementele de fezabilitate, rata profitului, rata dobânzii, perioada de amortizare

MCP ar trebui să considere de asemenea și informațiile care vin din partea participanților (stakeholders)

- De exemplu, costul unui articol poate fi măsurat în diferite momente, când este sesizat ca necesar în cadrul proiectului, când este comandat furnizorului, când este livrat, când este exploatat și în funcțiune sau/și când este trecut în evidențele contabile

## 8. Subproces MCP

Principalele subproces ale MCP

- Planificarea resurselor (Resource Planning)
  - Determinind care resurse (oameni, echipament, materiale) și ce cantități în parte ar trebui utilizate pentru a operaționaliza activitățile de proiect
- Estimarea Costului (Cost Estimating)

- dezvoltarea unei aproximații (estimări) a costului resurselor necesare finalizării activităților proiectului
- Bugetarea Costului (Cost Budgeting)
  - Alocarea unei estimări de cost generale a activităților de muncă depuse individual
- Controlul Costului (Cost Control)
  - Controlul tuturor schimbărilor bugetului de proiect

## Cursul 5

# Managementul dezvoltării

Cuprins:

1. Fazele ciclului de viață
2. Componente
3. Fundamentele proiectării
4. Pași proiectării
5. Principiile proiectării
6. Etapele proiectării

### 1. Fazele ciclului de viață

- Definiția
  - Începe când este formulată problema de rezolvat
  - accentul pe CE face programul (CE informație se prelucrează, CE funcții sau performanțe trebuie să aibă sistemul, CE interfețe cu alte sisteme etc.)
- Dezvoltarea
  - pune accentul pe CUM trebuie realizată aplicația
  - se definesc structurile de date și arhitectura programului
  - detalii de implementare a procedurilor și datelor
  - testare
- Exploatarea
  - Instalare, exploatare, întreținere

### 2. Componente

- metode
  - informații despre cum se construiește aplicația
  - metode pentru:
    - planificarea și estimarea proiectului
    - analiza de sistem și analiza cerințelor
    - proiectarea structurilor de date, arhitecturii programului și a algoritmilor
    - coding, testare și întreținere
- instrumente
  - oferă sprijin automat și semiautomat pentru metode

- specifice pentru fiecare clasă de metode
- instrumente integrate (CASE)
- procedee
  - liantul ce unește metodele și instrumentele
  - definesc
    - secvența în care se aplică metodele
    - documentele (documentații, rapoarte, formulare) necesare
    - verificările pentru asigurarea calității
    - punctele de verificare (milestones) pentru evaluarea progreselor realizate

### 3. Fundamentele proiectării

Scopul proiectării este producerea specificațiilor de proiectare, formate din

- (i) proiectul de arhitectură a sistemului
- (ii) modelele logice și fizice de date
- (iii) specificațiile de proiectare a procedurilor
- (iv) proiectul interfețelor

### 4. Pașii proiectării

- Selectia
  - Obiective
    - căutarea și identificarea soluțiilor alternative (manuale și informatice) pentru sistemul studiat (țintă)
    - evaluarea fezabilității fiecărei soluții alternative
  - Activități
    - identificarea soluțiilor posibile
    - consultarea utilizatorilor, managerilor, personalului tehnic
    - start: specificarea cerințelor
    - analiza fezabilității fiecărei variante (soluții)
    - stabilirea soluției alese
    - conducerea la decizia: cumpără ȘI/SAU dezvoltă
- Achiziția
  - Obiectivele achiziției:
    - căutarea și identificarea produselor specifice care pot ajuta soluția recomandată pentru sistemul țintă
    - solicitarea, evaluarea și clasificarea propunerilor (ofertelor) furnizorilor
    - selectarea și recomandarea celei mai bune oferte
    - stabilirea cerințelor pentru integrarea produselor ce se vor achiziționa în soluție
  - Activitățile achiziției:
    - (i) stabilirea criteriilor tehnice
    - (ii) solicitarea de oferte
    - (iii) validarea ofertelor
    - (iv) evaluarea ofertelor
    - (v) stabilirea ofertei câștigătoare
    - (vi) stabilirea cerințelor de integrare a produselor achiziționate în soluția propusă
- Proiectarea propriu-zisă



- Schiță a proiectului general pentru sistemul țintă
  - proiect de ansamblu (proiect preliminar)
- Activități
  - (1) Proiectarea arhitecturii programului
  - (2) Analiza și distribuirea datelor (proiectarea logică a datelor)
  - (3) Proiectarea logică a prelucrărilor

## 5. Principiile proiectării

- Abstractizarea
  - Abstractizarea funcțională (procedurală) - se realizează prin
    - specificare: nume, parametri, pre- și postcondiții
    - parametrizare: clase de probleme
  - Abstractizarea datelor - caracteristici:
    - încapsularea: reprezentarea și operațiile sunt puse împreună (clase)
    - ascunderea informației: accesul la reprezentare se face numai prin intermediul operațiilor (metode get/set)
  - Abstractizarea controlului
- Ascunderea informației
- Descompunerea
  - Instrumente folosite:
    - diagrame ierarhice
    - rețele de procese (DFD – “Data Flow Diagram” numai cu procese și fluxuri de date)
  - Avantaje
    - gestionarea complexității
    - implementare și testare separată a subsistemelor
    - activități paralele, muncă în echipă
- Modularizarea
  - Permite proiectantului să
    - (i) descompună un sistem în unități funcționale
    - (ii) impună o ordine ierarhică a folosirii acestora
    - (iii) implementeze abstractizarea datelor
    - (iv) dezvolte subsisteme independente
  - Gradul de independență a unui modul
    - cuplarea (măsoară interdependența relativă)
    - coeziunea (măsoară puterea funcțională)

## 6. Etapele proiectării

### (1) structurarea sistemului

- descompunerea sistemului în subsisteme
- identificarea interfețelor dintre subsisteme - fluxuri de date

### (2) modelarea controlului

- arhitectura centralizată - un subsistem central care asigură controlul execuției tuturor celorlalte subsisteme
- sistemele dirijate de evenimente - subsistemele răspund la evenimente generate în exteriorul lor

(3) descompunerea în module

- descompunerea se face la nivelul fiecărui subsistem
  - diagramele de blocuri de arhitectură (conțin subsistemele și fluxurile de date)
  - modelele de control

(4) optimizarea proiectului

- (1) la proiectarea arhitecturii - se descompune sistemul în module și se specifică fiecare modul
- (2) la proiectarea de detaliu - se elaborează specificațiile de programare pentru fiecare modul
- (3) la implementare - se implementează (codificare + testare) modulele proiectate
- (4) la testarea de sistem - se măsoară performanțele sistemului și se detectează locurile înguste (servicii și module critice)
- (5) se reconfigurează și se recombina modulele critice și se reia de la (3), până când se obțin rezultatele așteptate

(5) revizuirea proiectului

- obiectul revizuirii
  - (a) caracteristicile funcționale
  - (b) atributele de performanță
  - (c) interfețele cu mediul extern
  - (d) dialogurile cu utilizatorul
  - (e) formatul rapoartelor
  - (f) condițiile ce generează excepții și gestiunea acestora

## **Cursul 6**

# **Managementul estimarilor**

Cuprins

1. Estimari
2. Metodologii de estimare
3. Estimarea efortului
4. Evaluarea costurilor – COCOMO
5. Estimarea iterativa
6. Analiza financiara a proiectelor

## 1. Estimari

- Realizate, utilizate si modificate in timpul etapelor de
  - Planificare strategica
  - Studiu de fezabilitate si/sau SOW (Statement of Work)
  - Propuneri
  - Evaluarea dezvoltatorului sau a sub-contractantilor
  - Planificarea proiectului (iterativa)
- Procesul de estimare
  - Estimarea dimensiunii produsului
  - Estimarea efortului necesar (oameni-luni)
  - Estimarea programului proiectului

## 2. Metodologii de estimare

- Estimari “top-down”
  - Avantaje
    - Usor de calculat
    - Foarte eficiente la inceput (ex. Estimarea initiala a costurilor)
  - Dezavantaje
    - Modele discutabile
    - Risc de potrivire redusă
    - Precizie redusă – nu iau in calcul detaliile proiectului
  - Exemple: Analogia, Judecata Expertului si Metodele algoritmice
- Estimari “bottom-up”
  - Genereaza WBSs (Work-Breakdown Structures)
  - Avantaje
    - Foarte eficiente in cazul activitatilor care sunt foarte bine intelese
  - Dezavantaje
    - Anumite activitati nu sunt cunoscute intotdeauna
    - Consumatoare de timp

## 3. Estimarea efortului

- Tabele de orar
  - Convertirea estimarilor de dimensiune in estimari de effort
  - Folosesc date istorice
- Combinare între estimările de dimensiune și de effort
- Programarea bazată pe angajament
  - Un dezvoltator isi ia un angajament pe baza unei estimari proprii

## 4. Evaluarea costurilor – COCOMO

- COCOMO – CONstructive COSt Model
  - cel mai bine documentat și transparent model de evaluare a costurilor
- Influența a 15 factori de cost în determinarea efortului de dezvoltare a entităților software

## 5. Estimarea iterativa

- Estimările sunt rafinate gradual
- La fiecare etapa de planificare – cea mai buna estimare posibila
- Estimările sunt revizuite iterativ in timp ce planurile sunt re-ajustate
- Planurile si deciziile sunt revizuite in functie de noile estimari
- Pastrarea unui echilibru: prea multe revizuri vs. prea putine

## 6. Analiza financiara a proiectelor

Exista trei metode principale de determinare a valorii financiare a unui proiect:

- Analiza NPV (Net Present Value) - Valoarea Actualizata a Investitiei
  - NPV reprezinta valoarea neta actualizata- acea valoare prezenta a beneficiilor obtinute printr-o investitie, dupa ce s-a tinut cont de orizontul de timp specific proiectului pentru care se calculeaza si luand in considerare valoarea in timp a investitiei.
  - Daca valoarea NPV este negativa, ea indica cu certitudine faptul ca proiectul nu ar mai trebui realizat. Daca insa este pozitiva, atunci nu mai ofera indicii referitoare la ceea ce ar trebui facut
  - O valoare pozitiva nu poate fi comparata cu alt proiect decat daca orizontul de timp si dimensiunile investitiei sunt aceleasi.
- Randamentul investitiei (Return Of Investment)
  - Printre factorii secundari trebuie luati in considerare urmatoarii:
    - Costurile- cu cat costa mai mult o anumita rutina de activitate, cu atat mai mare va fi beneficiul dedus din automatizare sau din suportul tehnologic specific
    - Cunoasterea- cu cat este mai mare potentialul de reutilizare a informatiei in sistem, cu atat este mai mare si ROI
    - Colaborarea- comunicarea intre angajati este costisitoare, astfel ca, cu cat va fi mai extinsa componenta de colaborare, cu atat va fi mai mare ROI potential
  - Daca ROI este mai mic decat este nevoie pentru a initia proiectul, este foarte posibil sa poata fi corectat astfel:
    - Modificarea calendarului costurilor: schimbarea costurilor din anul initial prin distribuirea investitiilor de instruire si consultant a pe parcursul celorlalti ani
    - Negocierea preturilor: o mica scadere procentuala a preturilor poate determina o crestere dramatica a ROI, in functie de dimensiunile proiectului
    - Cresterea graduala a costurilor cu angajatii pentru instruire si alte scopuri, pe masura ce utilizarea tehnologiei devine mai eficienta si determina cresterea ROI
- Perioada de amortizare
  - Perioada de amortizare este intervalul de timp necesar obtinerii beneficiilor pentru a stinge costul initial al proiectului
  - Acesta este un indicator cheie al riscului - intr-un mediu tehnologic in schimbare
  - Majoritatea companiilor doresc proiecte IT care sa aiba o perioada de amortizare relativ scurta

# Cursul 7

## Controlul proiectelor

### Cuprins

1. Acțiuni pentru controlul proiectului
2. Monitorizarea costurilor
3. Analiza riscurilor
4. Identificarea riscurilor

### 1. Acțiuni pentru controlul proiectului

- Anticiparea (acțiune proactivă)
  - buclă de control cu feedback ↔ control feed-forward
  - contracararea din timp a perturbațiilor din proiect
- Modificările de plan (acțiune reactivă)
  - cererea clientului
  - greșeli în planurile inițiale
  - dificultăți neprevăzute în planul inițial

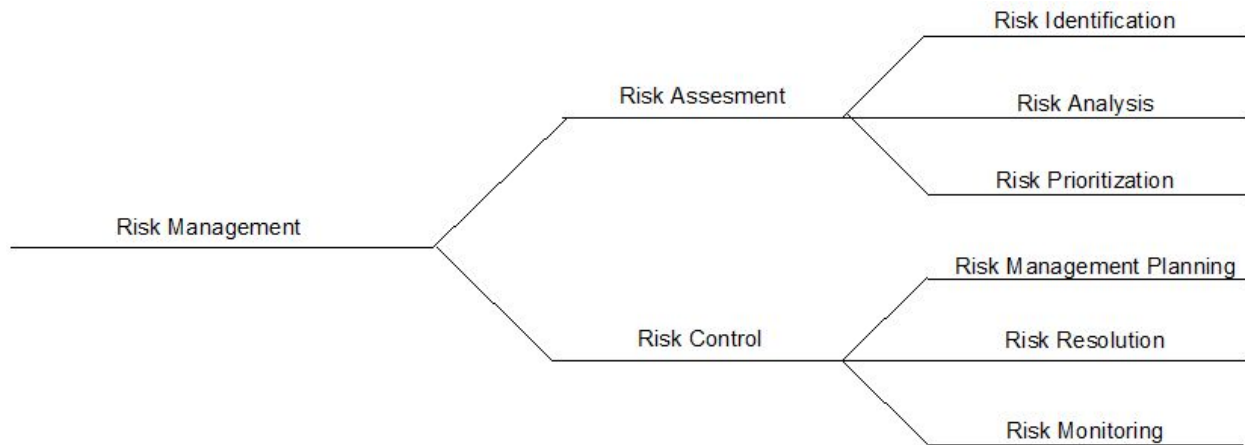
### 2. Monitorizarea costurilor

- Faza de concepție – cel mai potrivit moment
- Proiectul avansează → influența costurilor scade
- În fazele finale, în mod practic, lipsă de influență a costurilor

### 3. Analiza riscurilor

- În faza de elaborare a proiectului analiza de risc se realizează cu titlu preventiv asupra mijloacelor și ipotezelor de orientare a proiectului
  - riscul de concurență (bariere de intrare, concurența, etc.)
  - riscul de piață (conjunctura comercială, obiceiuri)
  - riscuri comerciale (fabricația produsului, la termen, raportul cost / calitate)
  - riscuri tehnologice (norme, competențe)
- În curs de execuție, prin mijloace tehnice de pilotaj se detectează și se măsoară abaterile realizând o analiză aposteori a cauzelor evenimentului într-o viziune corectivă ce va putea fi utilizată pentru proiectele viitoare
  - Conform metodei AMDEC (Analiza modurilor de defectare și efectele acestora) o percepție rațională a riscului se realizează după următoarea tipologie:
    - detectarea slăbiciunilor înainte ca ele să se producă poate fi mai mult sau mai puțin precisă și mai mult sau mai puțin tardivă, după caz. O procedură controlată, într-un context organizațional favorabil, poate da rezultate bune. Concret o informație bună circulă la un moment favorabil către un actor responsabil
    - diagnosticul cauzei – există metode statistice care pot furniza date care pot decide măsuri asiguratorii contra erorilor de diagnostic

- analiza prognostică a efectului – acest tip de analiză intervine când efectul încă nu este evident sau va fi realizat în termen lung care nu poate fi precizat



#### 4. Identificarea riscurilor

- Experiența cu proiecte similare în trecut
- Stabilirea profilului riscurilor
- Brainstorming în cadrul echipei
- Analiza planurilor inițiale – incertitudinile și presupunerile inițiale reprezintă de cele mai multe ori riscuri

## Cursul 8

### Managementul contractelor și achizițiilor

#### Cuprins

1. Reguli generale
2. Tipuri de contractari
3. Procesul de achiziție
4. Finalizarea contractului

#### 1. Reguli generale

- Contractele reclame derularea unor formalități contractuale
- Toate cerințele/specificațiile de produs și de proiect management trebuie menționate specific
- Dacă nu se menționează limpede în contract, modificările se fac numai când apar în discuție
- Orice schimbare trebuie întreprinsă prin acordul semnat de părți și conform unui derulator
- Modificările contractuale trebuie consemnate în scris
- Contractele trebuie să respecte cadrul legal

- Contractele trebuie să aibă rolul de a diminua din riscul de proiect
- Majoritatea guvernelor susțin contractările prin rezolvarea disputelor în fața unor curți speciale

## 2. Tipuri de contractări:

- Contractarea centralizată
  - Firma dispune de departament de contractări și relații comerciale
  - Avantaje
    - Creșterea experienței în contractări
    - Feedback membrilor departamentului
    - Standardizarea practicilor companiei
    - Profesioniștii în contractare au drumul batătorit în ascensiunea profesională
  - Dezavantaje
    - Supraîncărcarea angajaților departamentului
    - Dificultatea asigurării unei forme de asistență la nevoie
- Contractarea descentralizată
  - Procesul achizițiilor este delegat unui manager însărcinat cu contractările
  - Avantaje:
    - Accesul facil la cunoașterea tehnicilor contractuale
    - Solicita mai multă concentrare pe experiența contractuală
  - Dezavantaje:
    - Dificultate în menținerea experienței contractuale la un nivel înalt
    - Standardizare redusă a practicilor contractuale de la un proiect la altul
    - Tendințe de imprecizie în definirea carierei legate de profesia în contractare

## 3. Procesul de achiziție

Există 6 activități consecutive de achiziție

- Planul de achiziții
- Planul de Contractări
- Solicitări de Ofertă către furnizori prin Cererile de Ofertă adresate acestora
- Selectarea furnizorilor
- Administrarea contractelor
  - Atentie la:
    - analiza performanței furnizorului (buyer conducted performance review)
    - administrarea petițiilor (claims administration)
    - sistemul de menținerea evidențelor (records management system)
    - interpretarea contractului
- Finalizarea contractelor de achiziții

## 4. Finalizarea contractului

Finalizarea contractului presupune:

- Verificarea produsului - Verificarea produsului constă în controlul tuturor operațiunilor și elementelor care trebuie finalizate corect și satisfactor.
- Finalizarea financiară a contractului - Realizarea plăților finale și menținerea evidențelor
- Actualizarea evidențelor în sistemul de management al evidențelor

- Raportarea Finala a activitatii ca urmare a inchiderii contractului-Analiza si evidenta asupra gradului de performanta decurs si a eficientei activitatilor ca urmare a finalizarii contractului
- Arhivarea contractului - Organizarea documentelor contractului dupa sistemul dosarelor, mentinerea unui sistem de evidenta si inregistrare, folosirea unui sistem de codare, cu evidente istorice si elemente anexa utile in cazul nefinalizarii contractului sau a unor intimpinari de natura juridica si financiara
- Auditul de Achizitie - Este o trecere in revista structurata a procesului de achizitie. E posibil ca chiar furnizorul sa fie implicat in realizarea unor astfel de auditari
- Lectii invatate - vor reflecta modul de lucru cu furnizorul

## **Cursul 9**

### **Managementul oamenilor**

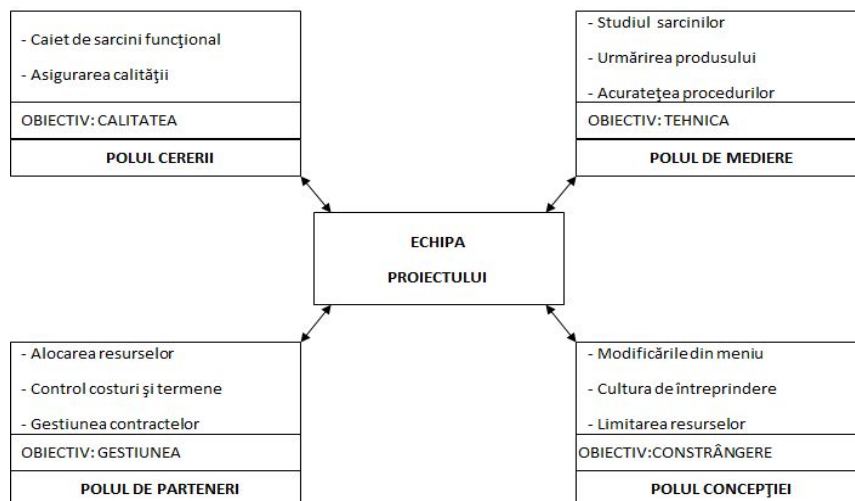
Cuprins:

1. Echipa de lucru a proiectului
2. Programatorul ca membru al unei echipe
3. Structura echipei proiectului
4. Conducerea proiectului
5. Cultura organizațională
6. Planificarea detaliată a activităților pentru un șef de proiect

#### **1. Echipa de lucru a proiectului**

- Polul cererii – regroupează clienții și utilizatorii potențiali ce pot fi consultați într-o manieră informațională de exemplu cu ocazia unui târg de prezentare sau într-o metodă formală, într-un cadru oferit de panoul de utilizatori
- Polul de mediere – are contribuții în sprijinirea inovației : consilii regionale, organisme interesate în dezvoltarea tehnologiei, etc. În general aceste organisme reușesc să promoveze diferite inovații materiale și competențe științifice, punând la dispoziție suporturi de proiecte care pot transforma o idee în produs comercial
- Polul de concepție – format din diferite centre de competență susceptibile a furniza sfaturi specializate asupra fezabilității tehnice a ideii
- Polul de parteneriat – sunt cuprinși ansamble de furnizori sau subantreprenori care participă la realizarea unor subansambluri ale proiectului





Caracteristici principale ale echipei de lucru în managementul proiectelor sunt:

- mărime: conține maxim 10, optim 6 – 7 persoane
- regulă de joc în care trebuie stabilite obiectivele și misiunile, calendarul, condițiile de buget, riscurile, nivelul de calitate, durata de viață a echipei
- puternică motivație la nivelul tuturor membrilor
- un sistem de comunicare eficient
- solidaritate reală bazată pe ascultare, întraajutorare, cooperare, utilizarea unui limbaj comun
- creativitate stimulată pentru cercetare și găsire de soluții rapide și eficiente

## 2. Programatorul ca membru al unei echipe

Tipuri de personalități de grup:

(I) orientată pe lucru - motivată de munca pe care o face

(II) orientată pe interacțiune - motivată de prezența și acțiunile colaboratorilor

(III) auto-orientată - motivată de dorința de succes personal

Personalități predominante în echipă:

- tipul I: orientată pe lucru
  - EȘEC echipa se sparge într-o sumă de indivizi, fiecare cu idei proprii asupra modului de realizare a proiectului
- tipul II: orientată pe interacțiune
  - SUCCESUL se obține prin participarea tuturor membrilor echipei
- tipul III: (III) auto-orientată
  - primează succesul personal nu succesul echipei

Recomandări:

- personalități de toate tipurile, în special I și II
- tipul III să aibă puțini reprezentanți
- șeful echipei să aibă o personalitate de tipul I

Stiluri de conducere a echipei

- autoritar
- democratic

### 3. Structura echipei proiectului

Structurarea proiectului

- stabilește câte echipe participă la realizarea acestuia
- alternative
  - o singură echipă
    - echipa este responsabilă cu produsul soft pe tot ciclul de viață a acestuia
    - avantaje
      - tranziție ușoară între activități
    - dezavantaje
      - nevoie de specializare mai largă a membrilor echipei
      - lucrează concomitent la mai multe proiecte
  - structură funcțională - fiecare activitate este realizată de o altă echipă
    - echipe diferite pentru: planificare, analiză, proiectare, implementare, testare de sistem, asigurarea calității, întreținere, documentare, etc.
    - în practică, se recomandă trei echipe (analiză, proiectare + implementare, testare + întreținere) și grupuri de sprijin specializate (documentare, instalare, instruire)
    - avantaje
      - specializare mai îngustă a personalului
      - atenție acordată activităților auxiliare (documentare, instruirea utilizatorilor, asistență tehnică)
    - dezavantaje
      - comunicarea între echipe
  - structură matriceală - fiecare activitate are o echipă de conducere și o echipă de specialiști
    - similară organizării matriciale a serviciului informatic
    - fiecare activitate are:
      - o echipă de conducere proprie
      - o echipă de specialiști proprie
      - o persoană face parte concomitent din mai multe echipe
    - avantaje
      - concentrarea competenței în echipe mici și stabile
      - flexibilitate în asigurarea de resurse umane pentru proiecte
      - dă o deosebită flexibilitate întreprinderii
      - stimulează cooperarea interdisciplinară
      - implică și provoacă angajații în activități inovatoare
    - dezavantaje
      - o persoană are doi șefi: unul pe linie tehnică, unul pe linie ierarhică
      - încurajează lupta pentru putere
      - poate conduce la discuții inutile în argumentarea acțiunilor
      - solicită pentru conducători mari aptitudini interpersonale
      - de multe ori dublează efortul intelectual
      - afectează moral personalul care este continuu rearanjat în alte grupe de lucru

Organizarea internă a echipei - de la descentralizat spre centralizat

- echipe informale (nestructurate)
  - Altruistă
    - echipa stabilește prin consens obiectivele de atins și deciziile de urmat
    - conducerea este asigurată prin rotație
  - Democratică
    - la fel cu echipa altruistă cu deosebirea că șeful nu se rotește
  - Avantaje
    - toți membrii participă la luarea deciziilor
    - membrii învață unii de la alții
    - satisfacții profesionale sporite
  - Sezavantaje
    - volum mare de comunicare
    - cerință: compatibilitate ridicată între membri
    - slabă responsabilitate: a greșit echipa (CINE?), nu individul
    - dezastru: echipe fără experiență și formate numai din incompetenți
- echipe structurate
  - echipa ierarhică (impartita in subechipe)
    - Avantaje: limitează numărul de canale de comunicare (la echipe mari)
    - Dezavantaje: de regulă devin șefi de subechipe specialiștii cei mai buni
  - echipa programatorului șef
    - Avantaje
      - deciziile se iau centralizat
      - canale de comunicare puține - productivitate sporită
      - dublarea productivității față de echipele nestructurate
    - Dezavantaje
      - programatorul-șef este de neînlocuit
      - programatorul șef culege toate laudele
      - este greu de evidențiat contribuția unui membru de rând

#### 4. Conducerea proiectului

Se disting patru opțiuni de configurații:

- proiect cu facilitator – un personaj detașat care se ocupă de un proiect antrenând în discuții și acțiuni minore definite
- proiecte cu coordonator – coordonatorul proiectului nu are autoritate ierarhică asupra membrilor grupului. El are doar o autoritate funcțională
- proiecte structurate în matrice – matricea are două logici : logica de funcțiuni și logica de proiecte.
- structuri ad-hoc

#### 5. Cultura organizațională

Cultura organizațională cuprinde un ansamblu de valori morale, principii, norme, simboluri și rituri care-și transmit și imprimă mesajul asupra comportamentului angajaților, cu implicații directe asupra eficienței, eficacității și imaginii întreprinderii.

Cultura organizațională este determinată de trei categorii de factori:

- factori centrali, sunt determinați de resursele umane ale organizației și ei sunt reprezentați de:
  - profilul și personalitatea conducătorilor

- riturile și simbolurile specifice instituției
- comunicația
- factori manageriali care sunt generați de metodele și instrumentele de lucru manageriale:
  - strategia
  - structurile și procedurile
  - sistemele de conducere
- factori de mediu, generați de starea organizației și a mediului exterior

## **6. Planificarea detaliată a activităților pentru un șef de proiect**

Pasul 1 – Identifică sarcinile

Pasul 2 – Pune sarcinile într-o ordine logică

Pasul 3 – Studiază implicațiile

Pasul 4 – Evaluarea resurselor cerute

Pasul 5 – Identifică ierarhia proiectului

Pasul 6 – Clarifică cine poate să adopte decizii

Pasul 7 – Monitorizează și controlează

Pasul 8 – Respectă anumite reguli de grup

# **Cursul 10**

## **Fazele finale ale proiectelor**

Cuprins:

1. Fazele finale ale proiectelor
2. Criteriile de baza pentru evaluarea membrilor echipei
3. Factori de succes a proiectelor

### **1. Fazele finale ale proiectelor:**

- Confirmarea indeplinirii tuturor cerintelor de proiect
- Verificarea finalizarii fiecarei faze de proiect
- Obținerea formală a aprobării de la beneficiar
- Evidența finalizărilor timpurii
- Plati finale
- Documenarea experienței/lecții învățate
- Actualizări de date
- Asigurarea finalizării proceselor
- Actualizare proceduri de proiect
- Adăugarea noilor abilități dobândite în baza de cunoștințe
- Auditarea achizițiilor
- Elaborarea procedurii de finalizare

- Finalizarea contractuala si administrativa
- Analiza factorilor de succes si eficienta
- Intocmirea si distribuirea raportului final
- Arhivarea generala a documentelor de proiect
- Evaluarea satisfatiei clientului
- Finalizarea operatiunilor si lucrari intretinere
- Disponibilizarea resurselor
- Celebrarea

## **2. Criteriile de baza pentru evaluarea membrilor echipei**

- Calitatea muncii depuse
- Costul
- Timpul consumat
- Creativitatea
- Performanta de natura administrativa
- Abilitatea de a lucra in echipa
- Atitudinea
- Abilitatile de comunicare
- Abilitatea tehnica
- Recomandarile de imbunatatire
- Consistenta in respectarea termenelor

## **3. Factori de succes a proiectelor**

In cadrul managementului de proiect exista numai 3 factori de succes:

- Predarea la timp a lucrarii - Predarea la timp se face conform agendei stabilite preliminar
- Incadrarea in buget - Proiectul se va incadra in estimarile de buget previzionate
- Predarea la nivel de calitate ridicata

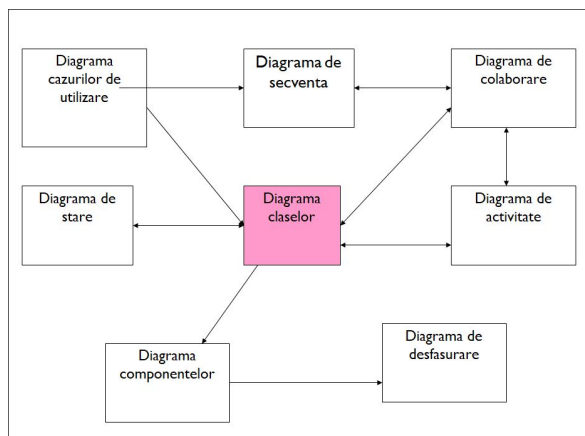
# Cursurile 11 & 12

## UML

### Cuprins

1. UML - Unified Modeling Language
2. Clase de obiecte
3. Mostenirea
4. Relații de asociere
5. Actorul
6. Diagrama de secvența
7. Diagrama de stări
8. Diagrama de colaborare
9. Diagrama de activități
10. Diagrama de pachete
11. Diagrame de componente
12. RUP

### 1. UML - Unified Modeling Language

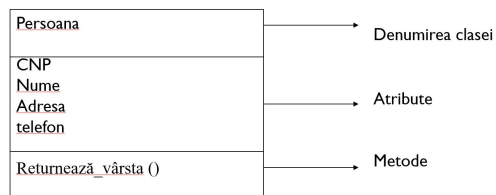


### 2. Clase de obiecte

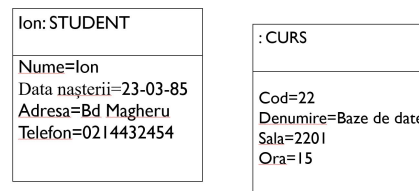
- un set de obiecte cu structură și comportament similar.
- obiectele sunt instanțe ale claselor

Exista trei niveluri de vizibilitate în UML:

- public (+) - are acces orice alt clasificator;
- protected (#) - are acces orice descendent;
- private (-) - numai clasificatorul însuși poate folosi această caracteristică.

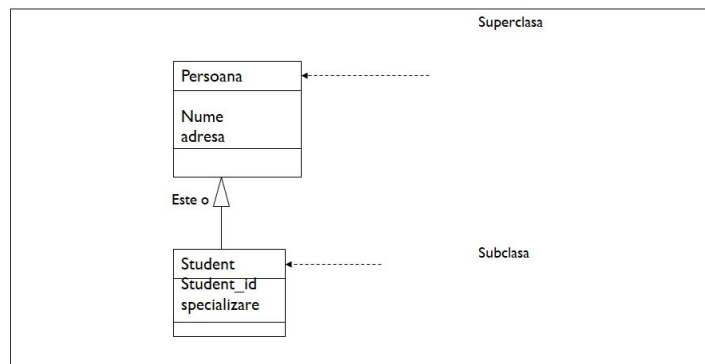


Clasa



Obiecte

### 3. Mostenirea



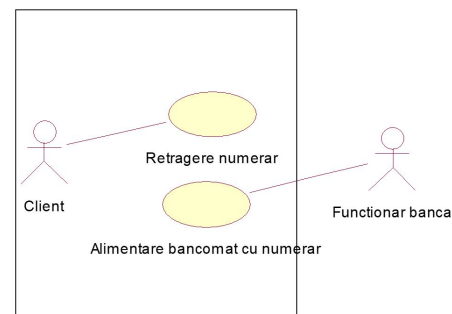
### 4. Relații de asociere

- Asocierea = corespondența abstractă între două clase
- Legătura = corespondența abstractă între două instanțe ale claselor
- Asocierea = abstractizare a legăturilor

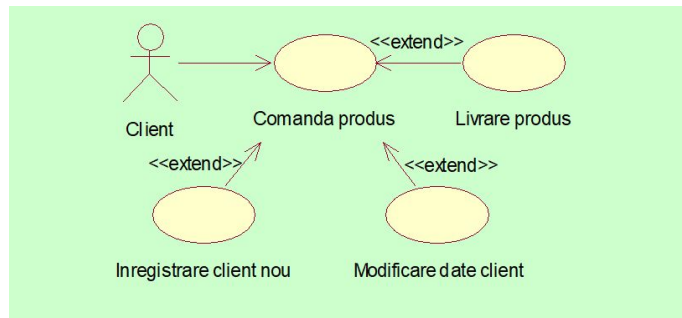
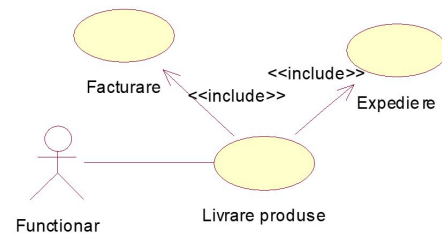
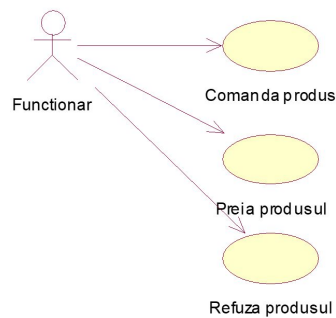
ASOCIERE	
ASOCIERE DIRECȚIONALĂ	
GENERALIZARE	
REALIZARE	
DEPENDENȚĂ	

### 5. Actorul

- Entitate exterioară sistemului informatic (SI) care beneficiază de servicii, rol jucat de o persoană/sistem care interacționează cu SI
- Persoană, dispozitiv fizic, alt sistem
- O persoană-mai multe roluri; un rol – mai multe persoane



- CINE- dorește/e interesat de informațiile din sistem
- CINE modifică date în SI
- CINE interacționează cu sistemul



## 6. Diagrama de secvență

- Înțelegerea ordinii evenimentelor pentru a parcurge întregul scenariu vizualizare a intercomunicării claselor
- Fiecare eveniment are ca rezultat un mesaj trimis unui obiect cu perspectiva că acel obiect va realiza o operație
- Recomandate pentru realizarea de specificații în timp real și pentru scenarii complexe



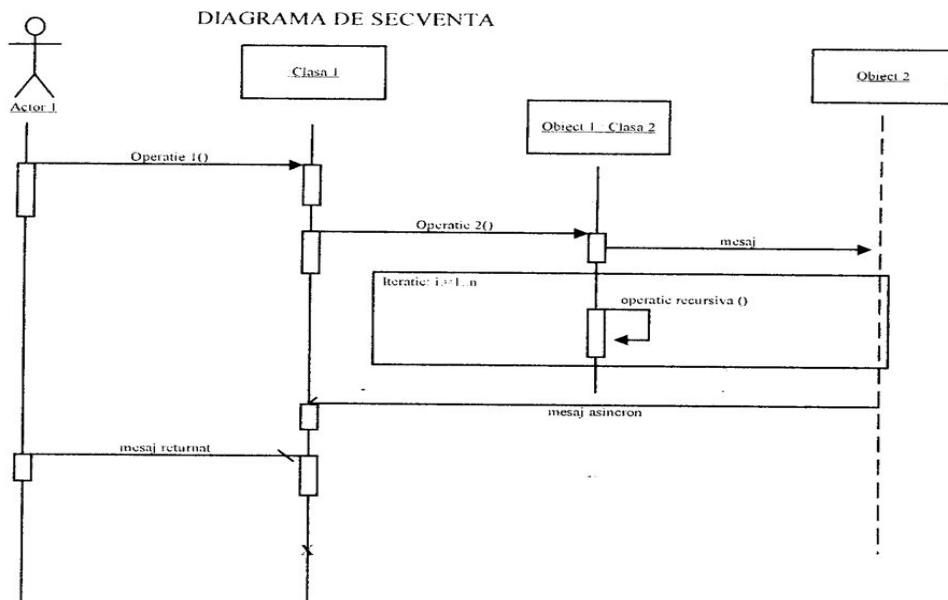


Figura 7.21. Diagrama de secvență (Sequence Diagram)

## 7. Diagrama de stari

- Comportament dinamic al obiectelor unei clase
- constă din stări, acțiuni, activități și tranziții
- o stare = „o condiție sau o situație din momentul existenței unui obiect care satisface în acel moment anumite condiții, efectuează anumite activități sau așteaptă anumite evenimente”.

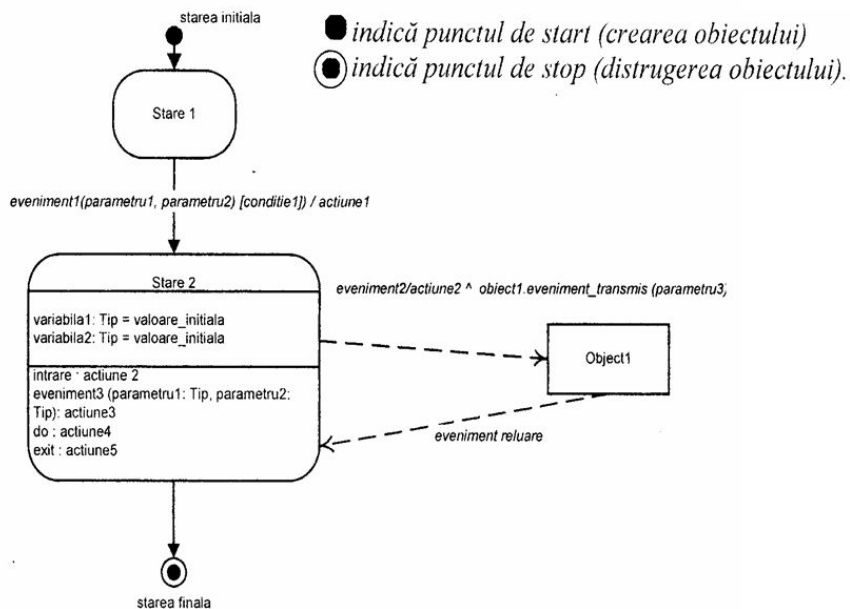
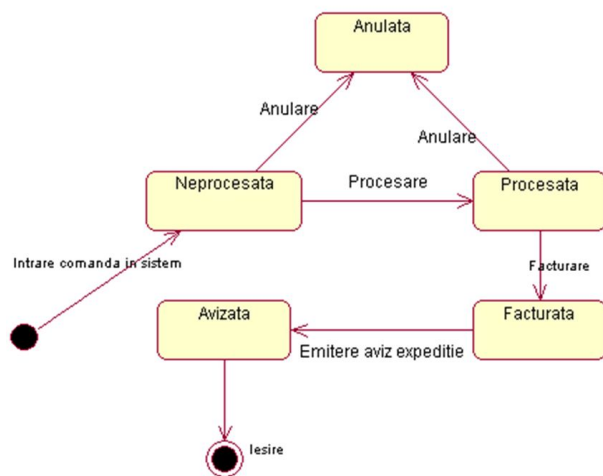


Figura 7.22. Diagrama de stare (Statechart Diagram)



## 8. Diagrama de colaborare

- descrie o examinare non-secvențială a modului în care interacționează obiectul.
- arată modul în care obiectele colaborează în cadrul unui singur scenariu al cazurilor de utilizare similar cu diagrama de secvență

### DIAGRAMA DE COLABORARE

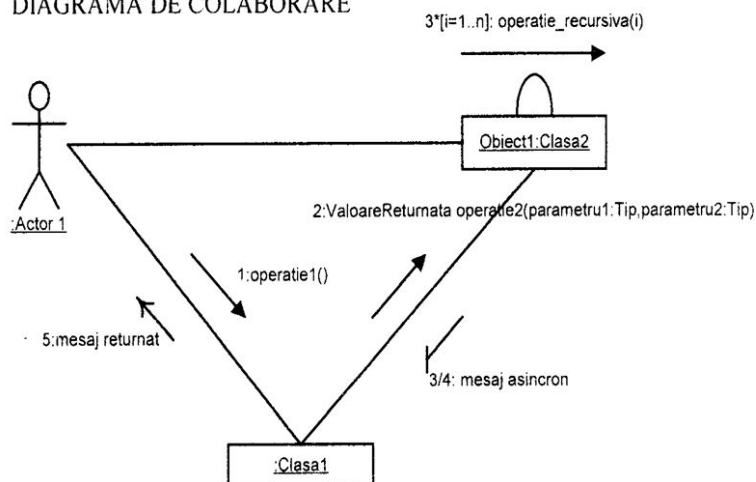


Figura 7.23. Diagrama de colaborare (Collaboration Diagram)

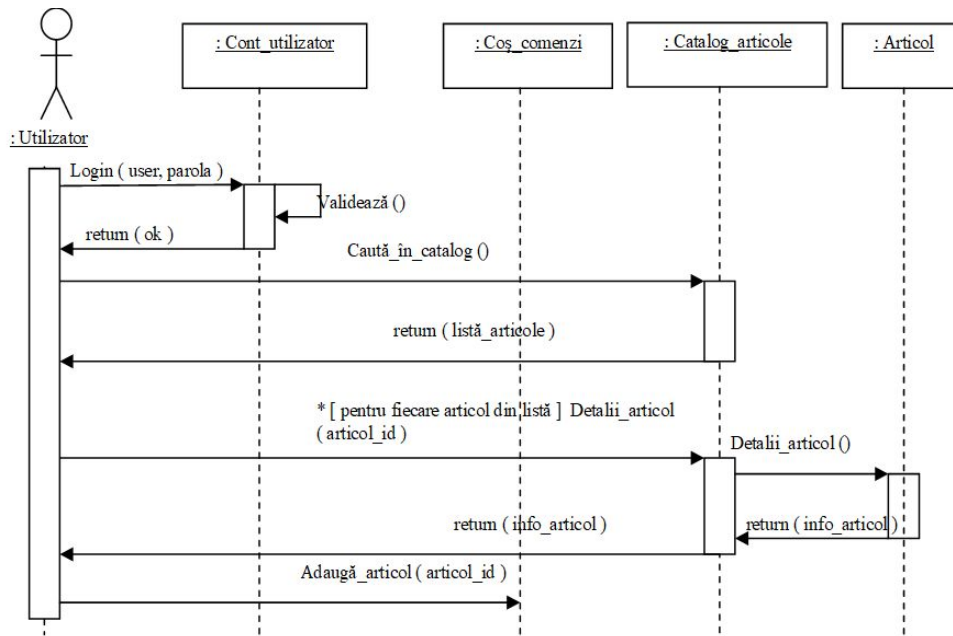


Figura 7.34. Reprezentarea unui scenariu cu ajutorul diagramei de secvență

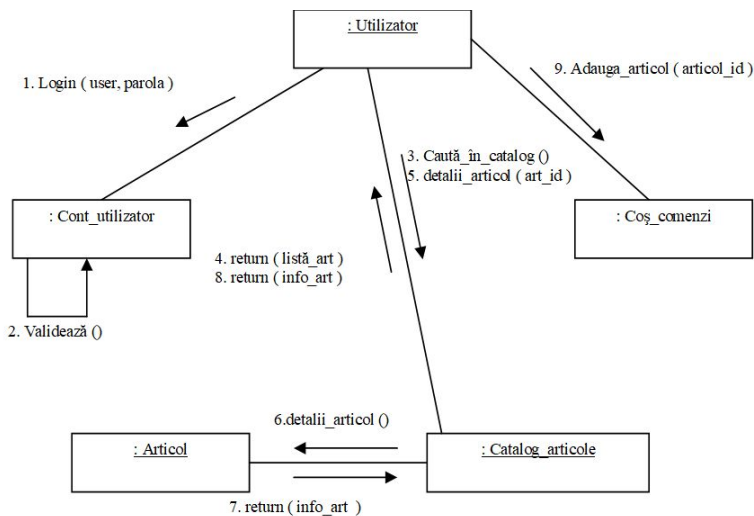


Figura 7.35. Reprezentarea unui scenariu cu ajutorul diagramei de colaborare

## 9. Diagrama de activitati

- o variantă a diagramei de stare, cu un scop puțin diferit, acela de a evidenția acțiuni și rezultate ale acestor acțiuni
- permite mai buna înțelegere a detaliilor din cadrul unei operații a unei clase
- ilustrează stările sub-acțiunilor și sub-tranzițiile.

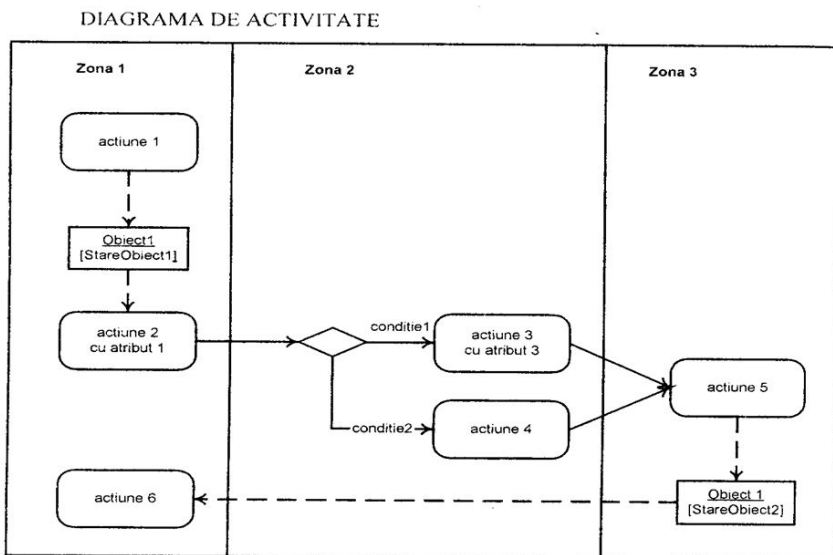


Figura 7.24. Diagrama de activitate (Activity Diagram)

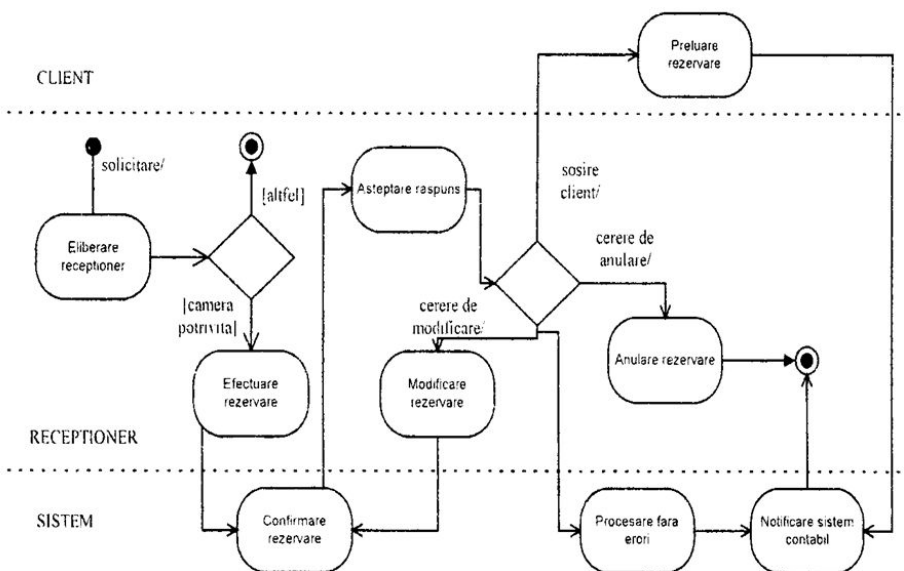


Figura 7.37. Diagrama de activitate pentru recepția unui hotel

## 10. Diagrama de pachete

- grupare a elementelor din cadrul diagramelor, numite pachete.
- pot fi ambalate alte pachete, clase, cazuri de utilizare, colaborări etc.
- Un element de modelare aparține unui singur pachet, dar alte pachete pot consulta acest element.
- Dacă se arată explicit conținutul pachetului, atunci numele pachetului se trece pe etichetă.

## DIAGRAMA PACHETELOR

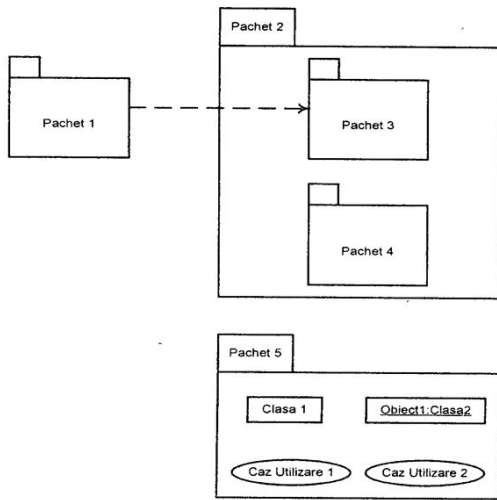
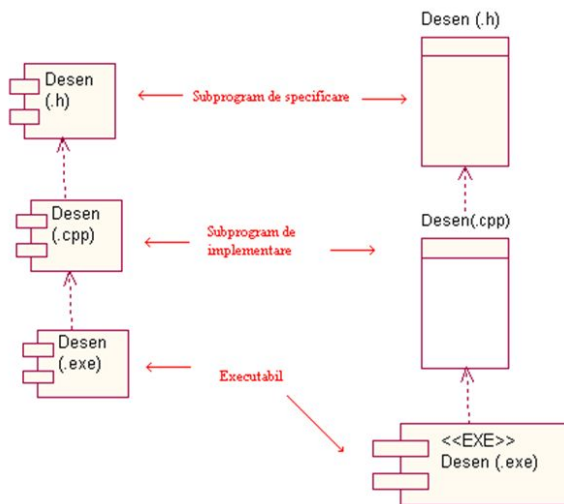
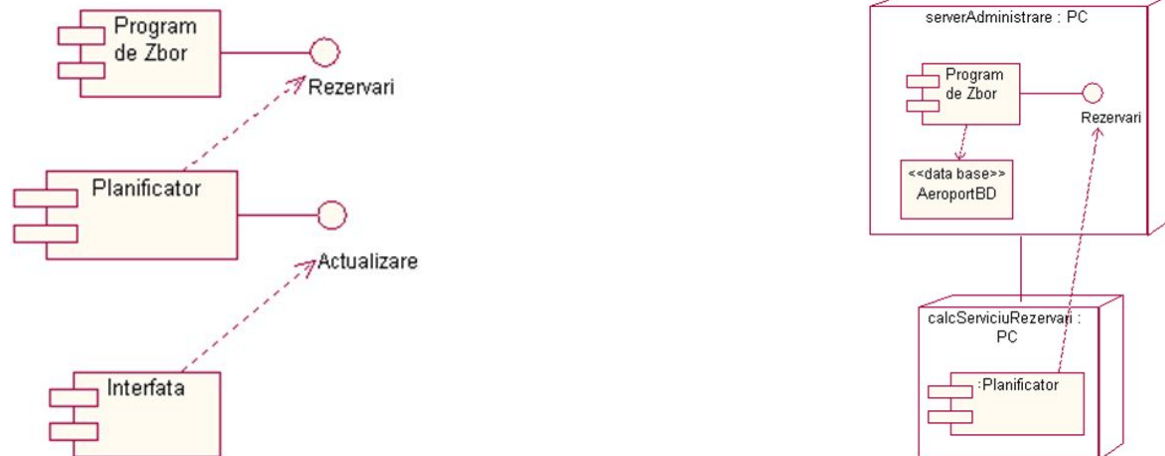


Figura 7.27. Diagrama pachet (Package Diagram)

## 11. Diagrame de componente

- prezintă dependențele existente între diverse componente software
- O componenta este un modul soft (cod sursa, cod binar, dll, executabil etc) cu o interfata bine definita.





## 12. RUP - Rational Unified Process

- Fazele descrise de UP sunt:
  - Explorarea inițială
    - definirea scopului proiectului și la pregătirea mediului de dezvoltare
  - Elaborarea
    - capturarea și organizarea cerințelor,
    - rafinarea viziunii asupra sistemului,
    - definirea și validarea arhitecturii și
    - elaborarea planului pentru faza de construcție.
  - construcția
    - Fiecare iterație - trei activități de baza:
      - managementul resurselor și controlul procesului,
      - dezvoltarea și testarea componentelor și
      - evaluarea la sfârșitul iterației.
  - Tranziția
    - finalizarea documentației
    - testarea produsului la client
    - modificări minore dictate de client
    - lansarea sau instalarea produsului final
- Fiecare faza are asociat un rezultat final
- Fazele reprezintă perioada de timp scursă între două rezultate finale
- La sfârșitul fiecărei faze este efectuată o analiză