



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



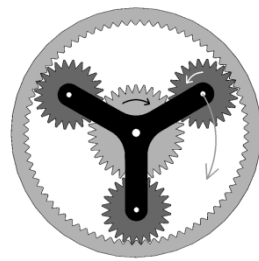
# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Managementul Proiectelor Software

### Principiile proiectării

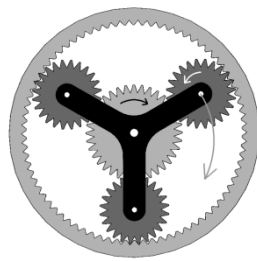
# Pașii proiectării: Selecția

- **Obiective**
  - căutarea și identificarea soluțiilor alternative (manuale și informatice) pentru sistemul studiat (țintă)
  - evaluarea fezabilității fiecărei soluții alternative
  - recomandarea celei mai bune dintre ele
- **Activități**
  - identificarea soluțiilor posibile
  - consultarea utilizatorilor, managerilor, personalului tehnic
  - start: specificarea cerințelor
  - diferă prin
    - gradul de automatizare a prelucrărilor
    - instrumentele soft folosite
    - arhitecturile hard-soft



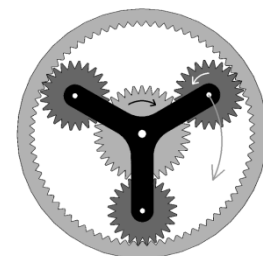
## Selecția (2)

- analiza fezabilității fiecărei variante (soluții)
  - tipuri de fezabilitate - ponderi
    - tehnică
    - operațională
    - economică
    - de derulare
  - indicator de fezabilitate: sumă punctaj \* pondere
    - scală de notare 0 - 100 (punctaj) pentru fiecare tip de fezabilitate
    - pondere asociată tipului de fezabilitate



## Selecția (3)

- stabilirea soluției alese
  - propunerea de sistem (“system proposal”)
    - planul proiectului
    - estimări de dimensiune
    - soluții identificate
    - analiza fezabilității
  - conducerea ia decizia
    - cumpără ȘI/SAU
    - dezvoltă



# Achiziția

- Cumpărare componente recomandate

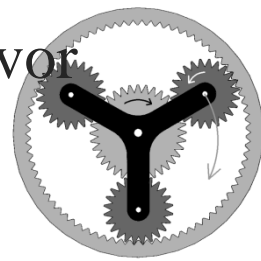
Hard

Soft

Servicii (consultanță, instruire)

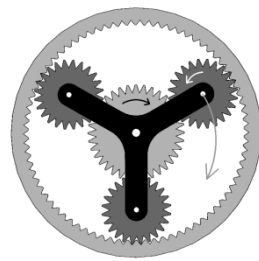
## Obiectivele achiziției:

- 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



# Achiziția (2)

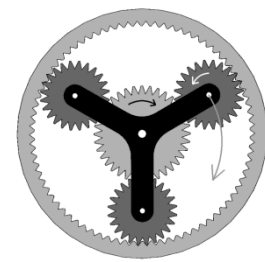
- activitățile achiziției:
  - (i) stabilirea criteriilor tehnice
  - (ii) solicitarea de oferte
  - (iii) validarea ofertelor
  - (iv) evaluarea ofertelor
  - (v) stabilirea ofertei câștigătoare
  - (vi) stabilirea cerințelor de integrare a produselor achiziționate în soluția propusă



# Proiectarea propriu-zisă

## Proiectarea generală

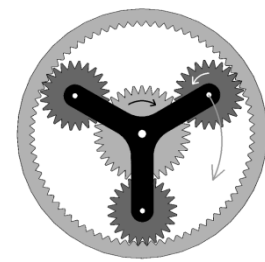
- Schiță a proiectului general pentru sistemul țintă
  - proiect de ansamblu (proiect preliminar)
- activități
  - (1) Proiectarea arhitecturii programului
  - (2) Analiza și distribuirea datelor (proiectarea logică a datelor)
  - (3) Proiectarea logică a prelucrărilor



# Proiectarea propriu zisă (2)

## Proiectarea de detaliu

- nivelului proiectării fizice
- activități
  - (4) Proiectarea fizică a datelor (fișiere și/sau BD)
  - (5) Proiectarea intrărilor și ieșirilor
  - (6) Proiectarea interfețelor on-line





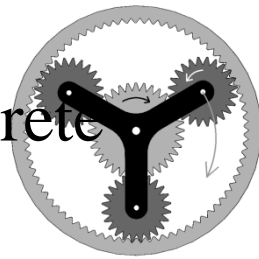
# Principiile proiectării

*Almost perfect is not good enough.*

Program care funcționează vs. Program bun

- factori de calitate
- program bun
  - conformitate cu documentul de specificare a cerințelor
  - Întreținere facilă a proiectului
  - cod sursă + **documente și alte resurse**
- Concepte și principii de proiectare
  - abstractizarea
  - ascunderea informației
  - descompunerea
  - modularizarea

Tehnici: concepte și principii aplicate în situații concrete



# Abstractizarea

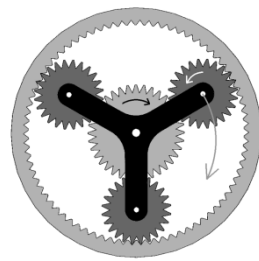
*Abstraction is everybody's zero but nobody's nought.*

neglijarea detaliilor pentru a surprinde esența  
dezvoltarea – succesiune de abstractizări

- nivelul conceptual: modele conceptuale, generale
- nivelul logic: modele logice
- nivelul fizic: modele fizice

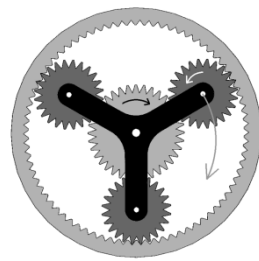
mecanisme de abstractizare

- abstractizarea funcțională (procedurală)
- abstractizarea datelor
- abstractizarea controlului



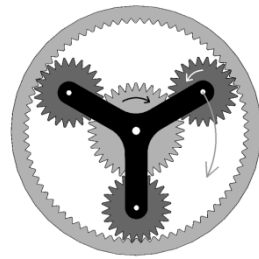
# Abstractizarea procedurală

- prelucrările efectuate de o procedură se concentrează în definiția acesteia (“interfața” procedurii)
- se realizează prin
  - specificare: nume, parametri, pre- și postcondiții
  - parametrizare: clase de probleme
- contract între cel care implementează procedura și cel care o folosește



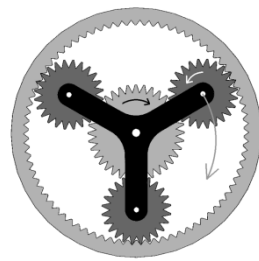
# Abstractizarea datelor

- Neglijarea tipului + accentul pe comportamentul (operațiile) acestuia
- Caracteristici
  - încapsularea: reprezentarea și operațiile sunt puse împreună (clase)
    - ascunderea informației: accesul la reprezentare se face numai prin intermediul operațiilor (metode get/set)
- specificarea TDA (tip de date abstract) conține
  - elementele ce formează reprezentarea TDA
  - structura - relațiile dintre elemente
  - domeniul - mulțimea valorilor valide
  - invariantul TDA
  - specificarea operațiilor



# Abstractizarea controlului

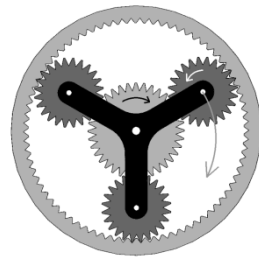
- evenimente
  - Interacțiunea cu utilizatorul
  - Aplicația, alte aplicații, sistemul de operare
  - TOE (task, object, event)
- excepții - erori de execuție
  - Declanșare, propagare, capturare, prelucrare
  - Throw, try, catch



# Descompunerea

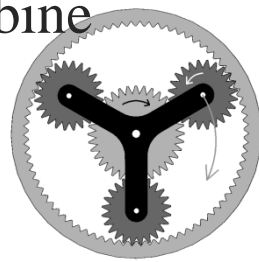
*Divide et impera.*

- subsisteme mai mici, mai ușor de înțeles și de gestionat
- exprimarea structurii sistemului prin
  - arhitectura soft
  - ierarhia controlului
- 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

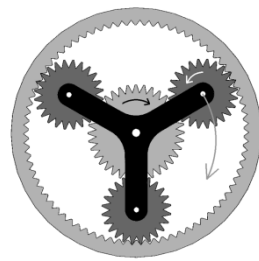
- Modul: o parte a unui program, separată logic de celelalte părți ale acestuia
- Program: colecție de abstractizări (module), fiecare gestionând un aspect particular al problemei de rezolvat
- Modularizarea permite proiectantului să
  - (i) descompună un sistem în unități funcționale
  - (ii) impună o ordine ierarhică a folosirii acestora
  - (iii) implementeze abstractizarea datelor
  - (iv) dezvolte subsisteme independente
- Sistem modular
  - colecție de abstractizări, fiecare reprezentând un subsistem bine definit
  - fiecare abstractizare se descompune în funcții



# Modularizarea (2)

## Caracteristicile unui modul

- sintactic
  - conține declarații de subprograme, variabile, tipuri de date, constante
  - compilare separată
  - implementează abstractizări procedurale sau TDA
- semantic
  - interfață
  - implementare
- nivel logic
  - stare (privată)
  - operații (publice)
  - client - server
- nivel fizic
  - variabile locale
  - proceduri și funcții publice





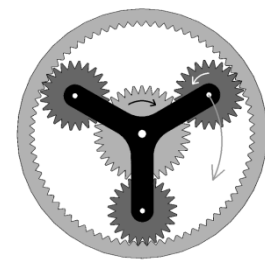
# Cerințele modularizării

## Gradul de independență a unui modul

- cuplarea (măsoară interdependența relativă)
- coeziunea (măsoară puterea funcțională)

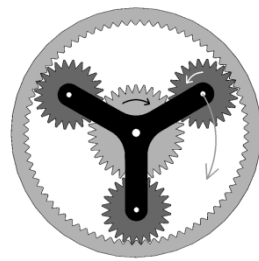
## Cuplarea

- tăria cuplării este caracterizată de
  - complexitatea interfețelor
  - tipul de conectare
  - tipul de comunicare între module
- cuplare slabă - module bine proiectate
  - module independente
  - cuplare prin date - parametri transmiși la apelul subprogramelor



## Cerințele modularizării (2)

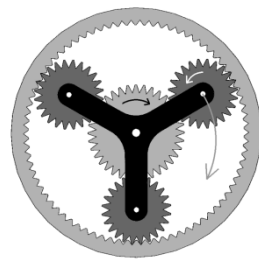
- cuplare de nivel mediu
  - cuplarea controlului:
    - indicator (flag) în client
    - rutină dispecer în client - apeluri în funcție de flag
- cuplare strânsă - de nivel înalt
  - cuplare externă - depinde de platforma gazdă
    - apeluri de servicii sistem, formate de conversie, protocoale de comunicație
  - cuplare folosind zone comune de memorie
    - două module care folosesc aceeași zonă de memorie
    - fiecare poate modifica valorile din zona comună fără ca să știe celălalt
  - cuplare de conținut - cod spaghetti
    - un modul poate modifica datele, controlul sau chiar instrucțiunile celuilalt



# Cerințele modularizării (3)

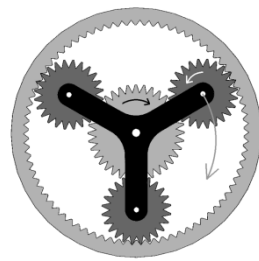
## Coeziunea

- înaltă - modul bine proiectat
  - Informațională
    - modulul implementează un TDA
  - funcțională
    - modulul implementează toate operațiile necesare unei funcții majore a aplicației
  - secvențială
    - parametrii de ieșire ai unui serviciu sunt parametri de intrare în alt serviciu
- medie
  - de comunicare
    - toate serviciile au aceiași parametri de intrare și de ieșire
  - procedurală
    - toate serviciile trebuie executate într-o anumită ordine



# Principiile proiectării

- joasă - modul slab proiectat
  - temporală
    - unele componente trebuie executate într-un moment bine definit al execuției
    - codul de inițializare
  - asociere logică
    - grupare după funcționalitate
      - generare de rapoarte
      - intrări-ieșiri
      - funcții matematice
  - incidentală
    - servicii independente



# Principii de proiectare a modulelor

- ▣ principiul interfețelor puține
  - număr minim de legături cu alte module
- ▣ principiul interfețelor mici
  - debit minim de informație schimbată cu alte module
- ▣ principiul interfețelor publice
  - serviciile din interfața unui modul sunt disponibile oricărui client, fără discriminare
- ▣ principiul ascunderii informației
  - datele locale ale modulului sunt accesibile numai prin intermediul serviciilor acestuia
  - toate detaliile de implementare a serviciilor sunt ascunse modulului client

