

Proiectarea sistemelor informatice

Seminar 1 - Introducere



Dezvoltarea de software este un proces complex



Implică o multitudine de elemente
care trebuie armonizate

Ce armonizăm



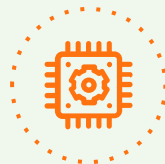
Resurse umane

Beneficiari, dezvoltatori,
specialiști în domeniu, utilizatori



Modele de afaceri

Producție, HR, contabilitate,
sănătate, învățământ



Cerințe tehnice

Software, hardware, rețele



Cerințe economice

Bugete, termene

Provocări

- + Comunicare inadecvată în cadrul echipei de dezvoltare
- + Planificare defectuoasă a procesului de dezvoltare software
- + Reticența beneficiarilor sau utilizatorilor
- + Testarea inadecvată a software-ului
- + Progresul rapid al tehnologiei
- + Modificarea cerințelor clienților
- + Limite de timp, resurse, infrastructură



Soluții



Creșterea nivelului de disciplină și rigurozitate



Etapizare, planificare, coordonare, stabilire rezultate (livrabile)



Facilitarea comunicării între membrii echipei de dezvoltare



Folosirea de modele



Utilizarea de standarde

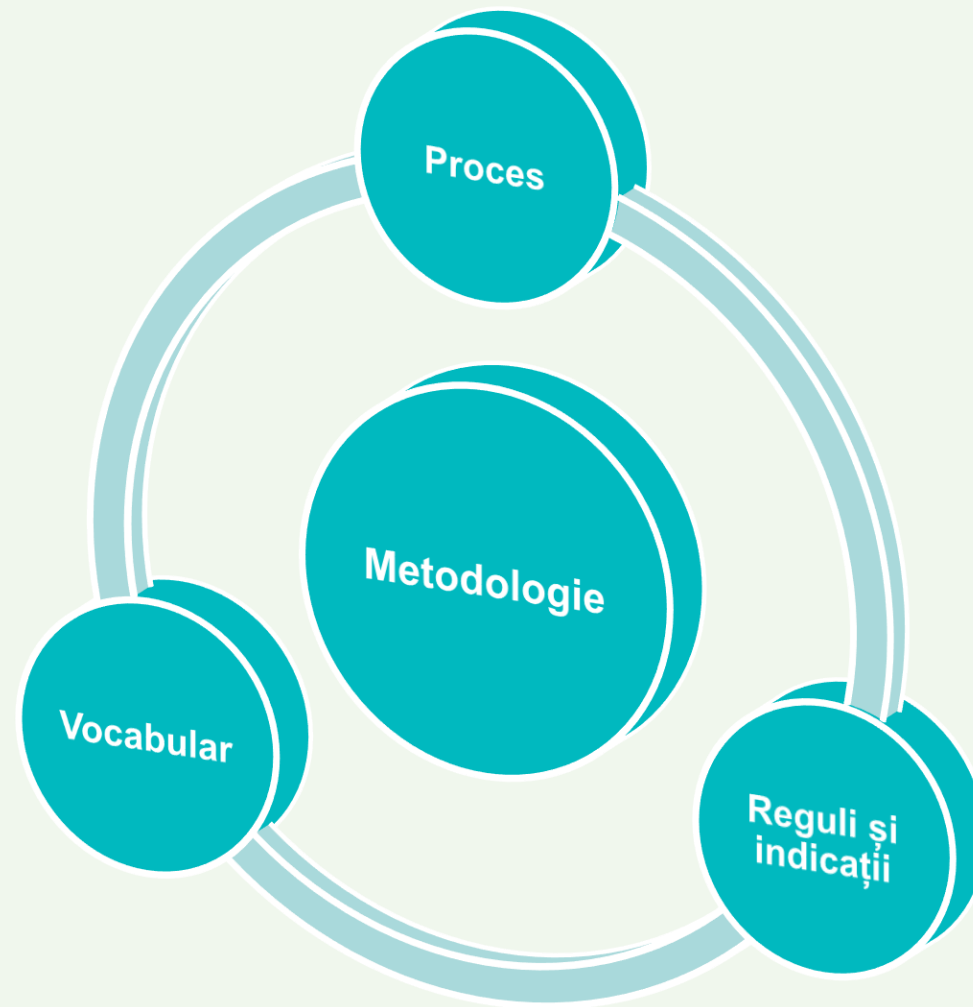
Ce sunt modelele?

- Simplificări sau abstractizări ale unor elemente reale sau care se proiectează.
- Evidențiază numai acele elemente care sunt importante pentru analist.
- Sunt specificate folosind notații grafice sau textuale precise, cu ajutorul unui anumit limbaj de simboluri.
- O colecție de imagini și text care are o anumită semnificație și intenționează să reprezinte ceva.

Ce sunt modelele?

- În dezvoltarea de software modelele pot fi de mai multe feluri:
 - modele de mediu
 - modele de domeniu
 - modele de specificare a sistemului
 - modele de proiectare a sistemului
- Modelele sunt valoroase deoarece:
 - sunt rapide
 - sunt intuitive
 - este mai simplu să modifice un model decât cod sursă

Metologii de dezvoltare software



Metologii de dezvoltare software



Proces: un set de activități care concură la atingerea obiectivelor urmărite



Vocabular: descrie procesul și rezultatele obținute în timpul aplicării acestuia



Reguli și indicații: definesc calitatea procesului și a rezultatelor



Roluri într-o echipă de dezvoltare

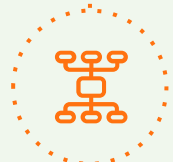
Metologii de dezvoltare software



Partea unei metodologii care poate fi standardizată este ***vocabularul (notația)***



UML (Unified Modeling Language) – notație comună care poate fi aplicată mai multor metodologii



Este foarte greu să se definească un singur proces potrivit pentru toate tipurile de proiecte



Exemple de alte notații standard

Exemple de metologii

- RUP (Rational Unified Process):
 - proces iterativ și incremental
 - livrarea de versiuni parțiale ale produsului la fiecare iterație
 - perioade scurte livrare și verificări frecvente
 - rezultate verificabile pentru clienți
 - proces exhaustiv; poate deveni greu de controlat
 - personalizarea RUP necesită un efort semnificativ
- OMT (Object Management Technique)
 - Precursor al UML
 - Utilizează conceptele orientării obiect

Exemple de metologii

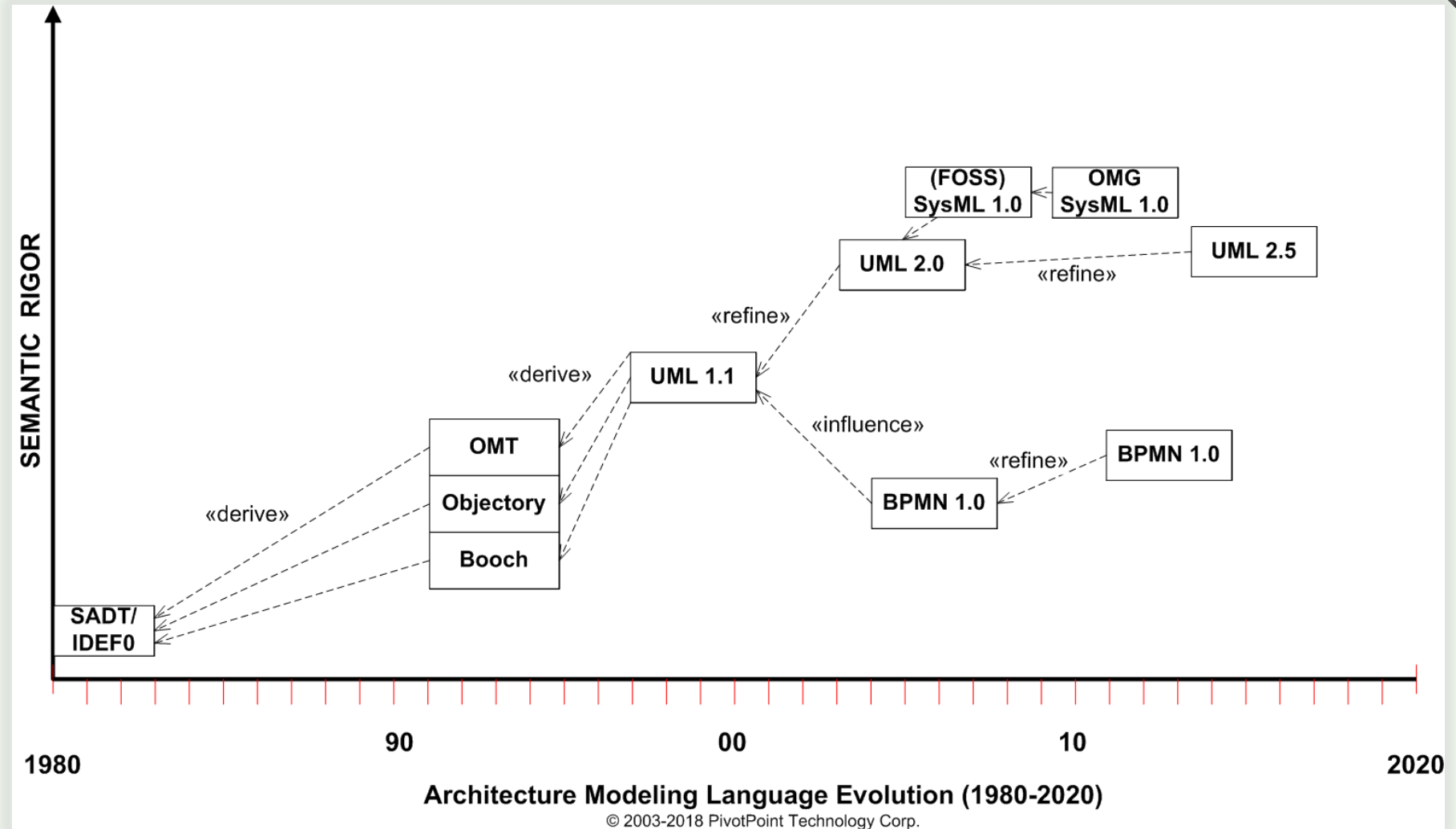
- XP (Extreme Programming)
 - metodologie agilă de dezvoltare
 - pune accentul pe codificare (standarde, principii)
 - încurajează ca programatorii să lucreze câte doi (“pair programming”)
 - numeroase sesiuni de discuții pe parcursul dezvoltării
 - fiecare iterație (1-4 săptămâni) are un rezultat funcțional
 - suport redus pentru modelare
 - relație strânsă între clienți și dezvoltatori
 - lipsa documentației de realizare

Ce este UML?

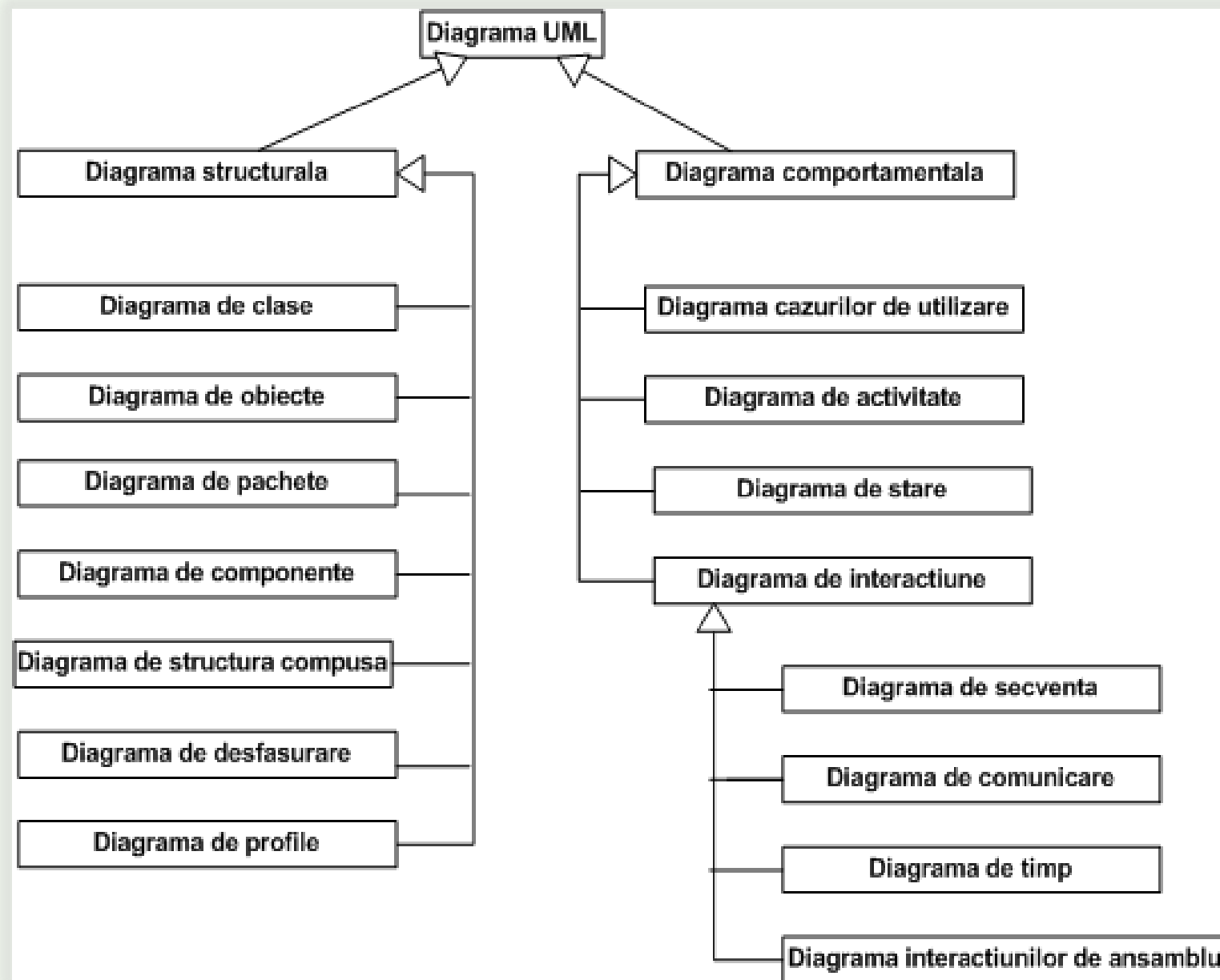
- ✓ **UML = Unified Modeling Language**
- ✓ **Limbaj de notații** pentru specificarea, construirea, vizualizarea și documentarea sistemelor software
- ✓ **Combină** cele mai bune practici în domeniul construirii diagramelor din ultimii 50 de ani
- ✓ **Standardizează** notațiile, dar nu stabilește modul în care acestea să fie folosite
- ✓ **Nu este o metodologie**, poate fi folosit ca vocabular pentru metodologii
- ✓ **Oferă flexibilitate** dezvoltatorilor, asigurând în același timp consistență
- ✓ Este dezvoltat și întreținut de **Object Management Group**



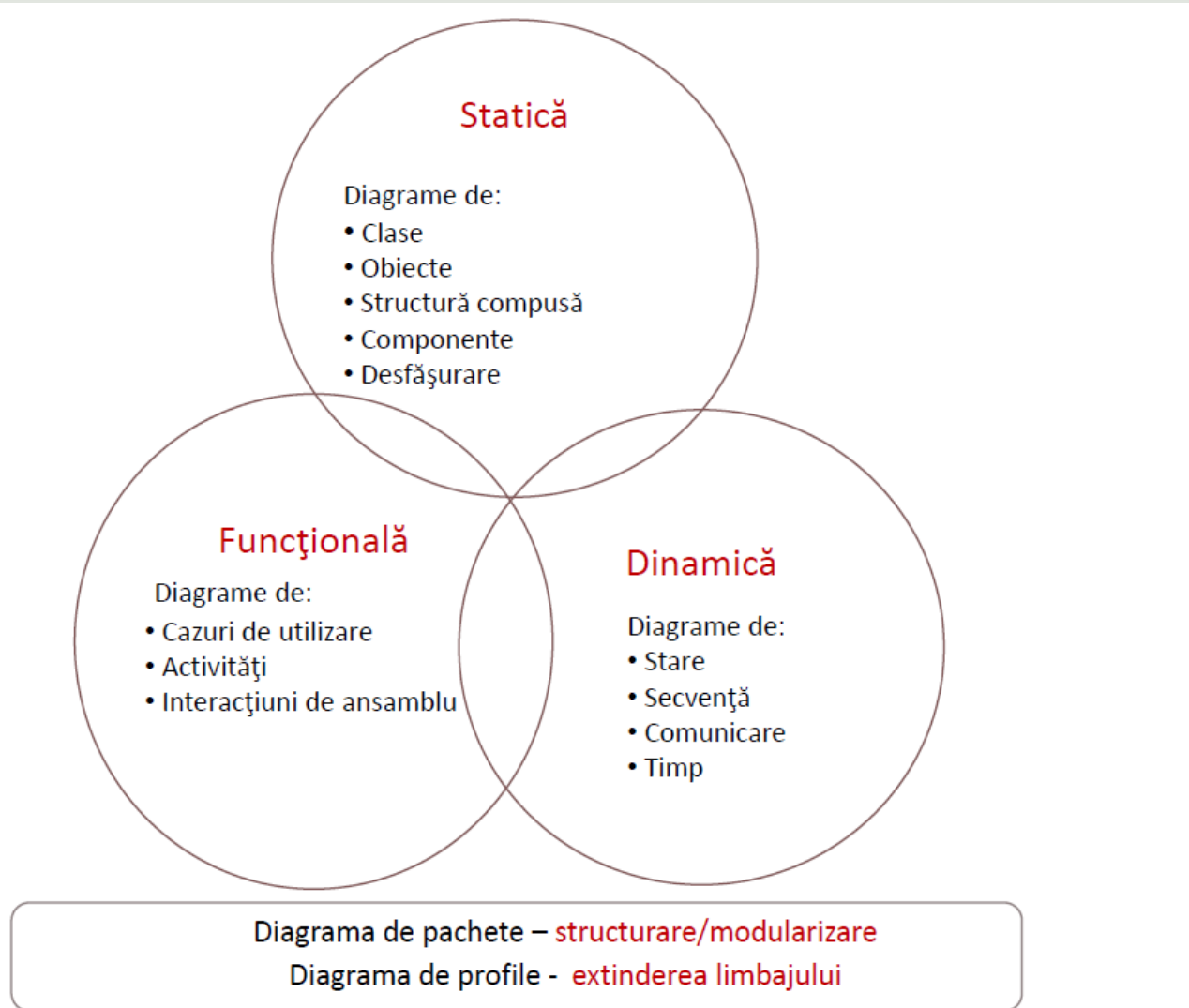
Istoria UML



Diagrame UML



Perspective asupra sistemului



Instrumente de tip CASE

- **CASE = Computer Aided Software Engineering**
- Necesitate:
 - lucrul cu modele vizuale poate fi o activitate dificilă și **consumatoare de timp**
 - nevoia unui suport informatic atunci când vrem să menținem **integritatea** modelelor
 - posibilitatea de a **genera cod**



Instrumente de tip CASE

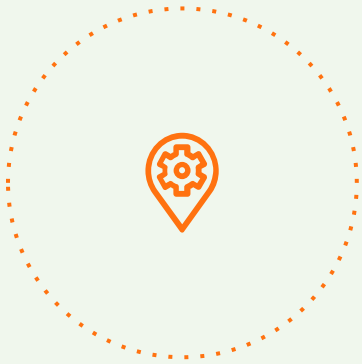
- ✓ Crearea diagramelor
- ✓ Gestiunea informațiilor despre diagrame
- ✓ Verificarea consistenței modelelor
- ✓ Crearea de legături între modele
- ✓ Urmărirea versiunilor modelelor
- ✓ Generarea de cod
- ✓ Inginerie directă și inversă



Cerințe seminar PSI

- Nota de la seminar: **50%** din nota finală
- Cerința de promovare: **minim nota 5** la seminar
- Nota de la seminar: **20%** activitate la seminar; **80%** evaluare proiect
- Evaluarea la seminar presupune realizarea unui proiect **individual**
- Proiectul implică **urmărirea etapelor** de specificare a cerințelor, analiză, proiectare și implementare a unui sistem informatic
- Se recomandă ca tema proiectului să coincidă cu **tema de licență**
- Punctarea proiectului este **condiționată** de încărcarea la timp și de prezentarea personală la seminar

Studiu individual



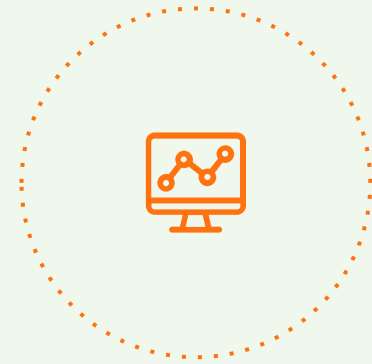
UML

Investigați limbajul UML folosind resurse Web. Prezentați în câteva paragrafe versiunea actuală a limbajului.
www.uml.org



OMG

Investigați scopul Object Management Group. Identificați și alte notații pe care le gestionează.
www.omg.org



CASE

Identificați un set de instrumente CASE suport pentru UML. Analizați și documentați suportul oferit pentru construirea diagramelor.