

Software Engineering (Ingineria Sistemelor Soft)

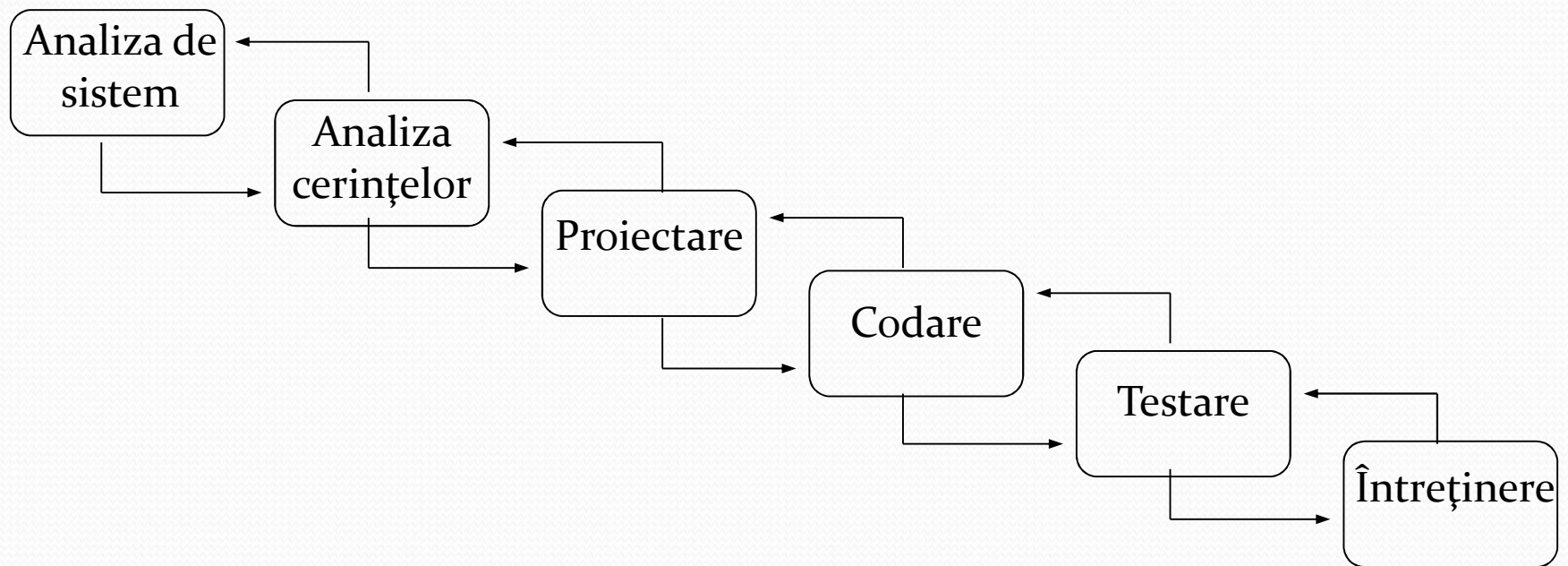
Curs 2

Etapele ciclului de viață al aplicațiilor

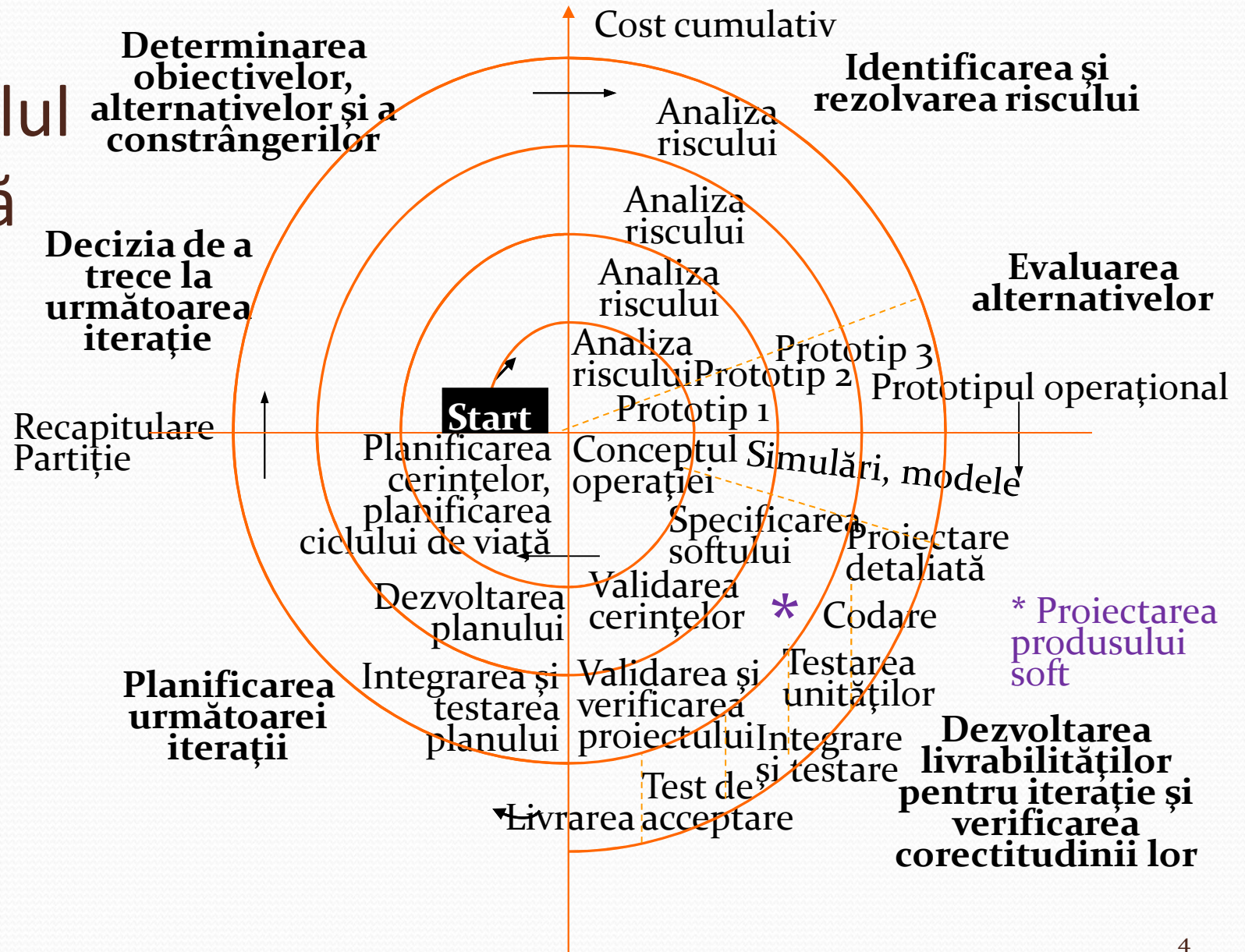
Lector dr. Pop Andreea-Diana

Paradigme de dezvoltare a softului

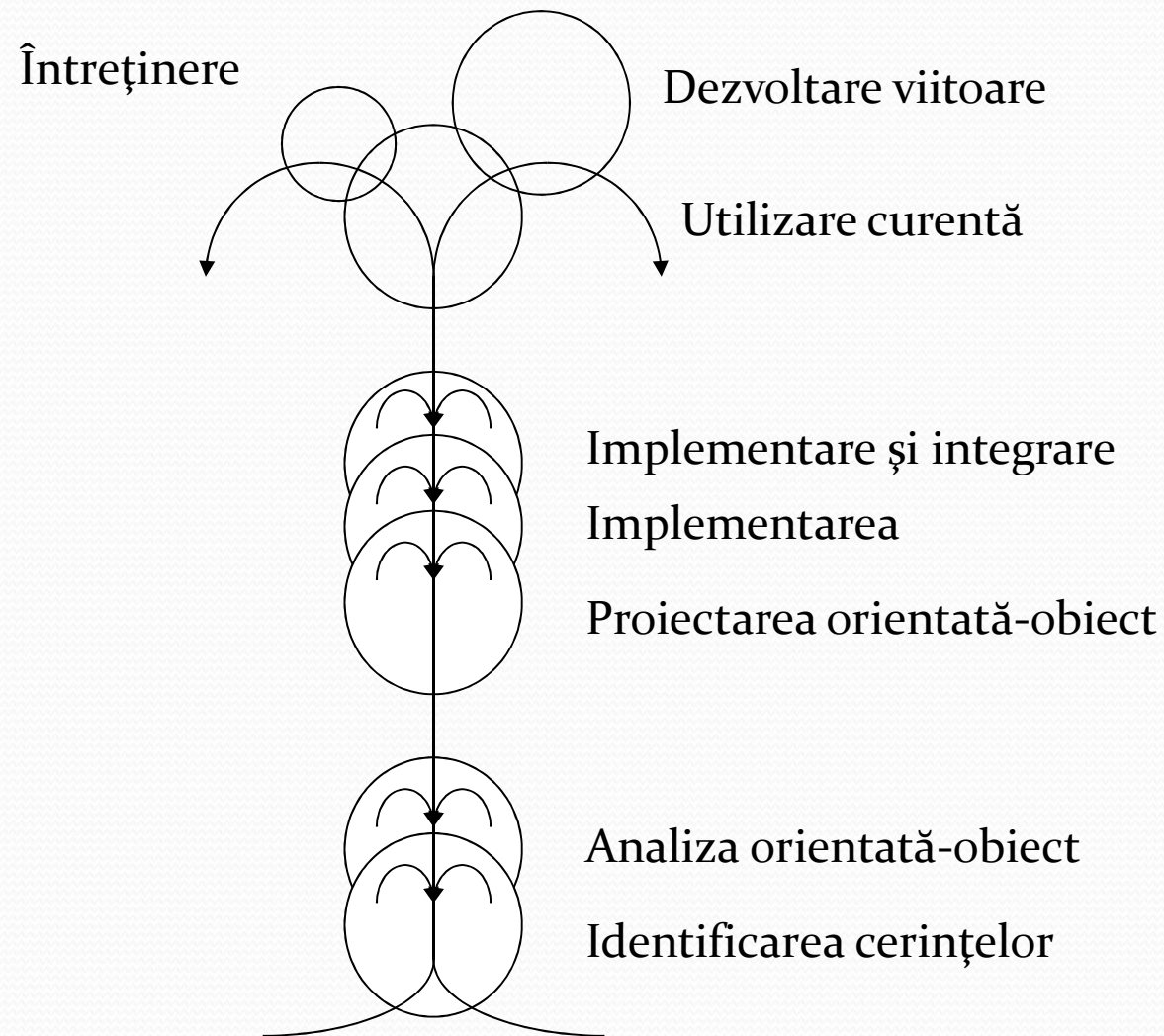
Modelul clasic de dezvoltare a softului



Modelul spirală



Modelul fântână

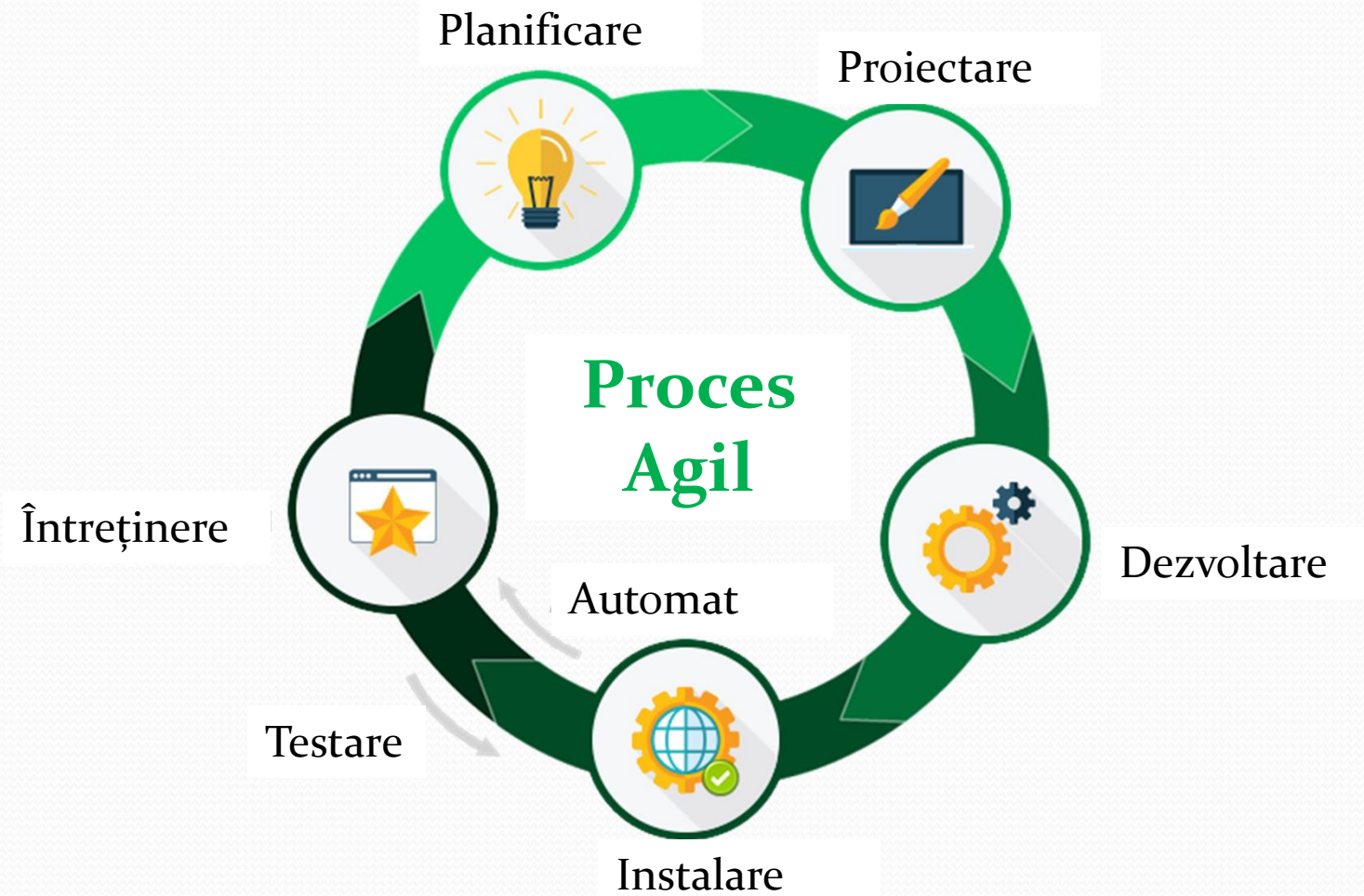




Prototipizare evolutivă

- Conceptul inițial
- Proiectarea și implementarea prototipului inițial
- Rafinarea prototipului până e acceptabil
- Completarea și rafinarea prototipului

Programare agilă





Etapele ciclului de viață al aplicațiilor

Indiferent ce model al ciclului de viață s-ar alege pentru dezvoltarea de soft, majoritatea au ca și etape:

- Planificarea;
- Analiza și Specificarea cerințelor;
- Proiectarea;
- Implementarea;
- Integrarea;
- Utilizarea și Întreținerea.



Planificarea

Ține cont de faptul că programul este întotdeauna softul parte a unui sistem mai mare, care include elemente și resurse de ordin:

- organizațional,
- uman,
- material (sisteme de calcul și de comunicație) și
- programe.

Aici se stabilesc cerințele pentru toate elementele sistemului și se alocă resursele pentru dezvoltarea programului.



Analiza și Specificarea cerințelor

- Se referă numai la aplicație.
- Ea cuprinde înțelegerea domeniului problemei, stabilirea funcțiilor, performanțelor și interfețelor.
- Cerințele pentru întregul sistem și pentru aplicație sunt documentate și sunt discutate împreună cu clientul și cu utilizatorul.



Specificarea cerințelor – parte a etapei de Analiză a cerințelor

Specificarea cerințelor este ultima fază a procesului de analiză a cerințelor, după colectarea cerințelor și analiza efectivă a lor. Rezultatul analizei, este producerea unui document numit Specificarea cerințelor softului. Acest document trebuie să satisfacă două cerințe contradictorii: pe de o parte documentul trebuie să fie clar și pe înțelesul clientului, pe de altă parte, trebuie să fie complet și detaliat. El va reprezenta criteriul de acceptare al produsului final.

Iată ce trebuie să conțină un astfel de document, după Pressman:

Documentul de specificare a cerințelor

Pagina de titlu

Identificarea proiectului și echipei,
identificarea documentului, data

Cuprins

Precizează paginile la care se găsește
fiecare capitol și secțiune

I. Introducere

A. Reprezentarea fluxului informațional

1. Fluxul de date
2. Fluxul de control

B. Reprezentarea conținutului informației

C. Descrierea interfeței sistemului

II. Descrierea informației

A. Descompunerea funcțiilor

B. Descrierea funcțiilor

1. Descrierea narativă a prelucrărilor
2. Restricții/limitări

3. Cerințe de performanță

4. Restricții de proiectare

5. Diagrame de sprijin

C. Descrierea controlului

1. Specificarea controlului

2. Restricții de proiectare

III. Descrierea comportamentului

A. Stări ale sistemului

B. Evenimente și acțiuni

IV. Criterii de validare

A. Margini de performanță

B. Clase de teste

C. Răspunsul așteptat de la soft

D. Considerații speciale

V. Bibliografie

VI. Anexe



Tipuri de specificații

Există trei mari tipuri de specificații:

- Specificații informale
- Specificații semiformale
- Specificații formale



Specificații informale

Constau în pagini scrise în limbaj natural, dar, din păcate, limbajul natural lasă loc diferitelor interpretări și a formulărilor ambigue și din această cauză, se preferă alte metode de specificare mai clară.



Specificații semiformale

Sunt cele obținute folosind:

- Tehnici grafice orientate pe acțiuni:

cum ar fi cele ale lui DeMarco [De Marco 1978], Gane și Sarsen [Gane și Sarsen 1979], și respectiv a lui Yourdon și Constantine [Yourdon și Constantine 1979] (diagrame de flux de date)

- Tehnici grafice orientate pe date:

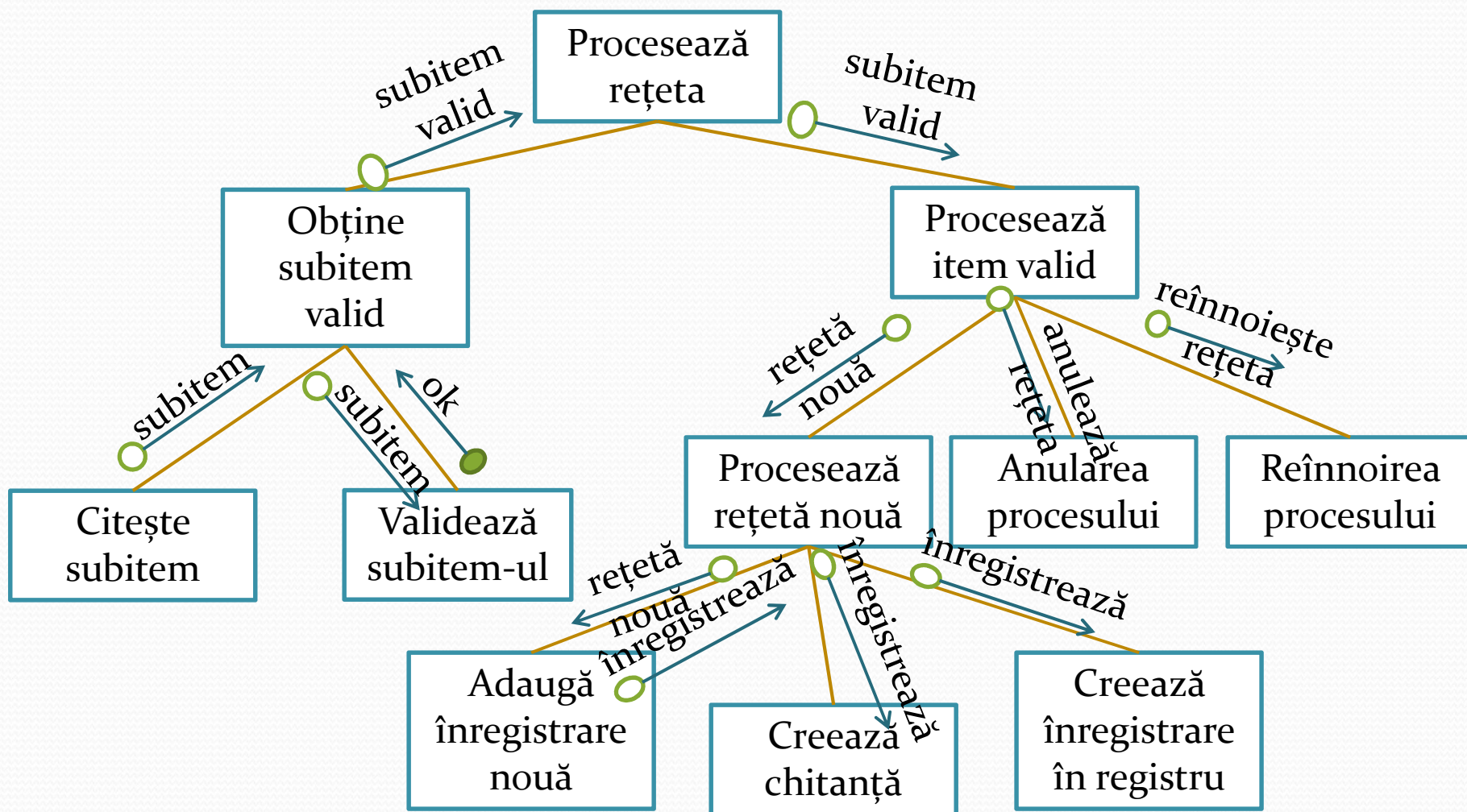
o astfel de tehnică a fost introdusă în 1976 de către [Chen 1976]: modelul entitate – relație de analiză și specificare a cerințelor

- Alte tehnici semiformale de specificare a cerințelor:

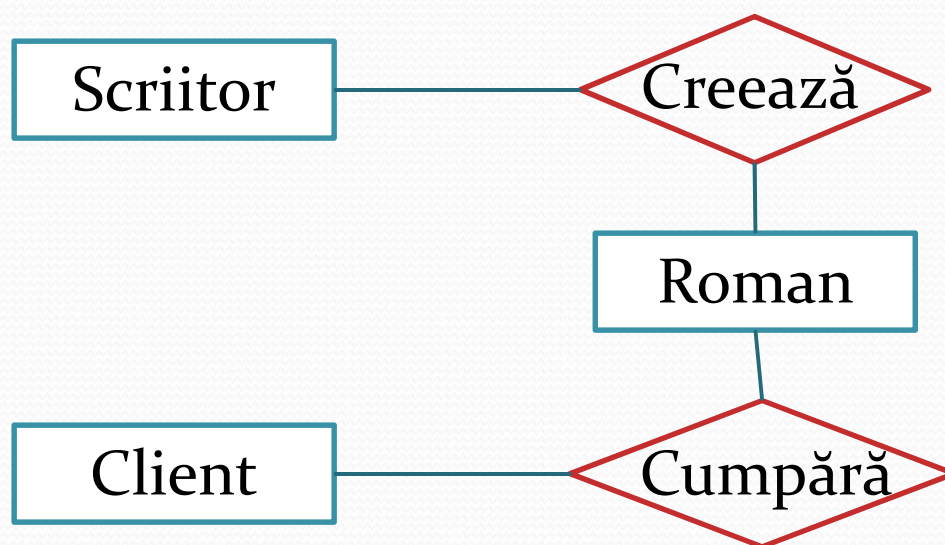
PLS/PLA (Problem Statement Language/Problem Statement Analyzer [Teichroew și Hershey 1977]); SADT (Structured Analysis Design Tehnique, [Ross 1985]); SREM (the Software Requirements Engineering Method, [Alford 1985])

Exemplu de specificație semiformală

Diagramă de structură



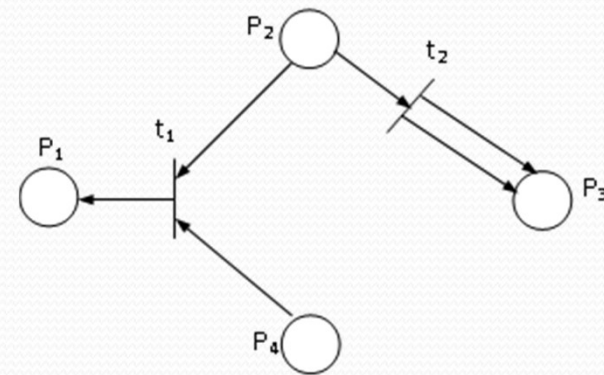
Model Entitate-Relație



Specificații formale

Sunt cele obținute folosind:

- Mașini cu stări finite
- Rețele Petri
- Limbajul Z
- Anna [Luckham și von Henke 1985], Gist [Balzer 1985], Vienna definition method (VDM) [Jones 1986], Limbajul CSP (Communicating Sequential Process [Hoare 1985])
- Ex.: $y: [0 \leq x \leq 9, y^2 == x \wedge y \geq 0]$





Proiectarea

- Este un proces format din mai mulți pași care pune accentul pe patru componente distincte ale programului:
 - arhitectura,
 - structurile de date,
 - detaliile procedurale și
 - descrierea interfeței.
- Proiectarea traduce cerințele într-o reprezentare a programului care poate fi analizată din punctul de vedere al calității înainte ca să se înceapă codificarea.
- Specificațiile de proiectare sunt parte integrantă a configurației soft.



Implementarea

- Este procesul de traducere a specificațiilor de proiectare într-o formă înțeleasă de mașină.
- Dacă proiectarea este detaliată suficient de mult, atunci codificarea se poate automatiza.
- Implementarea înseamnă: realizarea specificațiilor de programare, codificarea, conversia.
- Tot aici se face și testarea unitară a procedurilor și modulelor implementate.
- Ea trebuie să confirme că toate instrucțiunile, procedurile și modulele respectă specificațiile de proiectare.



Testarea

- Pune accentul pe aspectele interne ale softului.
- Ea trebuie să confirme că funcțiile prevăzute sunt îndeplinite: la date de intrare predeterminate se obțin rezultate ce se pot verifica că sunt corecte.
- În această activitate se realizează testarea de integrare (programul se assemblează din componentele sale implementate și testate deja), urmată de testarea de acceptare, la care participă și clientul, utilizatorul.



Integrarea

- Are loc integrarea modulelor și se realizează testarea de integrare.
- După acest pas, aplicația trebuie să funcționeze ca un tot unitar.



Utilizarea și Întreținerea

- Este etapa cea mai îndelungată din ciclul de viață a unui produs soft.
- Întreținerea poate fi însoțită uneori de o "aducere la zi" a proiectului (adăugarea de funcționalități noi, schimbarea tehnologiilor, repararea unor erori care nu au fost identificate până acum).
- În momentul în care întreținerea sa este prea costisitoare, se impune renunțarea la aplicația curentă și crearea uneia noi.