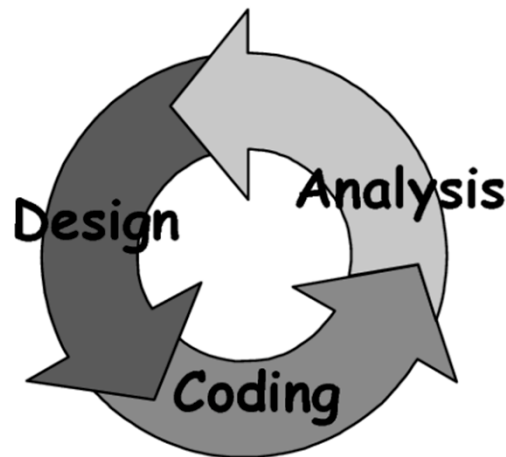


M. Caramihai, © 2022

**PROGRAMAREA
ORIENTATA
OBIECT**

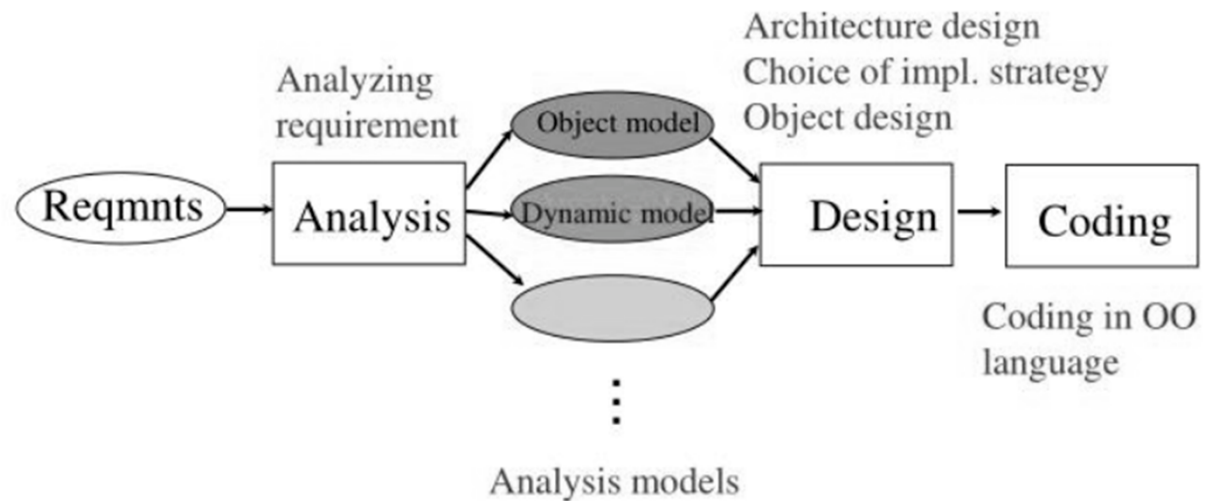
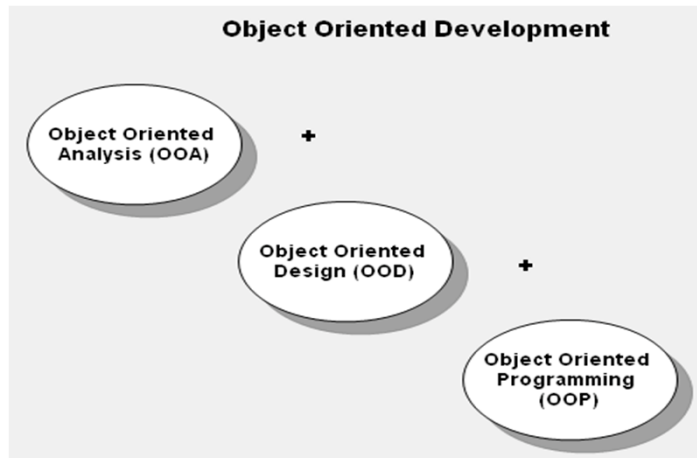
Analiza, proiectarea si programarea orientata obiect



Sistem informatic

- Acea parte a sistemului **informational** a unei organizatii in care stocarea / prelucrarea datelor se realizeaza prin mijloace informatice
- El este format din aplicatia / produsul informatic si elementele folosite pentru exploatarea sa de catre utilizatori
- Ciclul de viata al SI: defineste perioada de timp dintre "nasterea" (punerea in functiune) si "moarte" (inlocuire)

Dezvoltarea aplicatiilor OO



Analiza si proiectarea OO

- **Analiza:**

Intelegerea, gasirea si descrierea conceptelor in domeniul problemei

- **Proiectarea:**

Structurarea solutiei soft pornind de la etapa de analiza pana la nivel de implementare



Analiza *vs.* proiectare

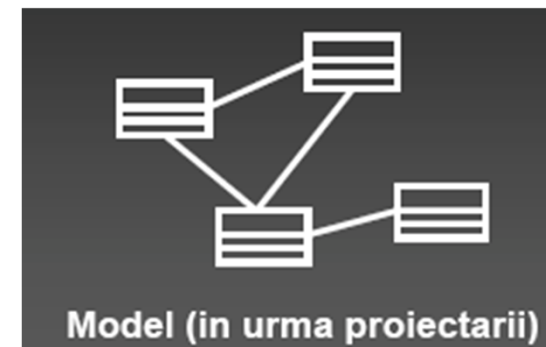
Analiza

- Evolutie
- Structura sistem
- Cerinte functionale
- Model (in mic)

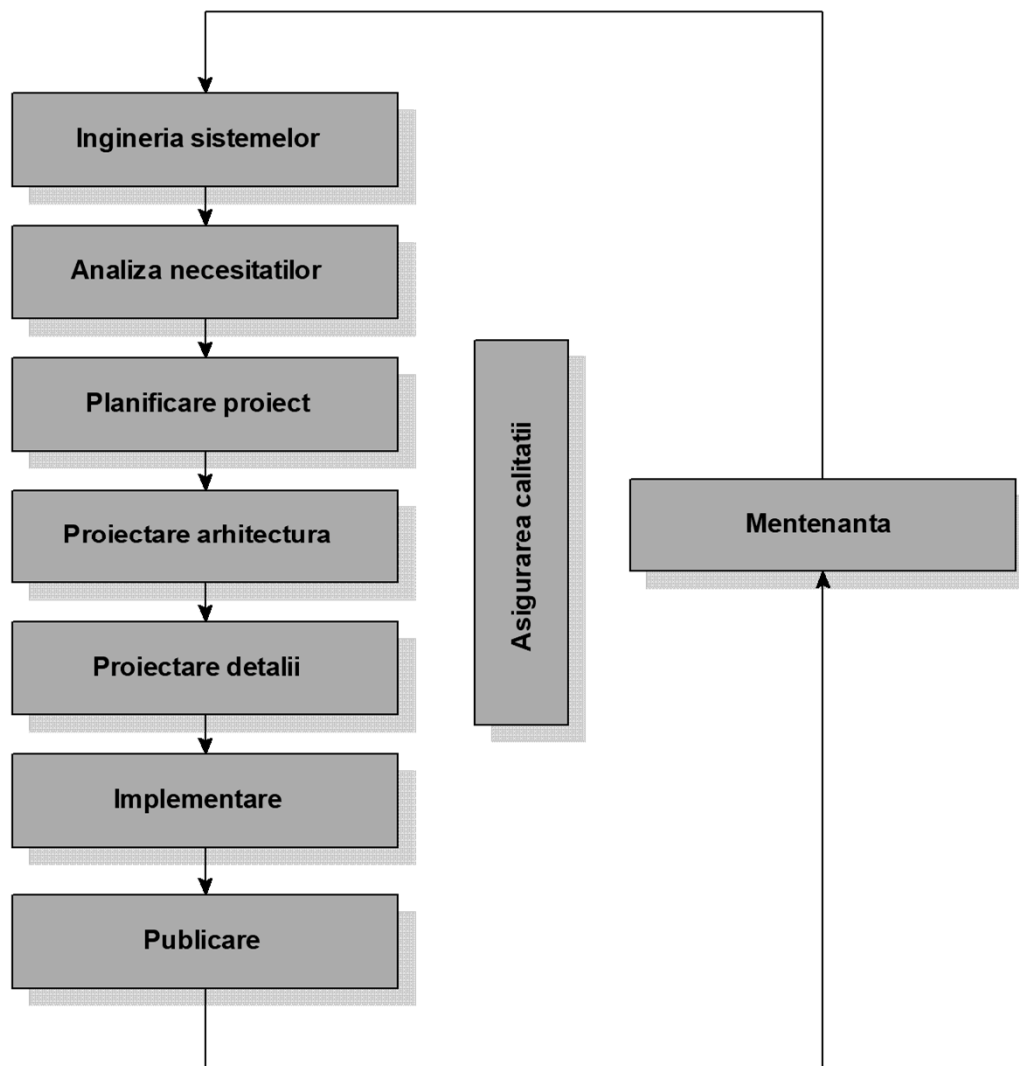


Proiectare

- Operatii si attribute
- Performanta
- Usor de programat
- Ciclul de viata al obiectelor
- Cerinte non-functionale
- Model (in mare)

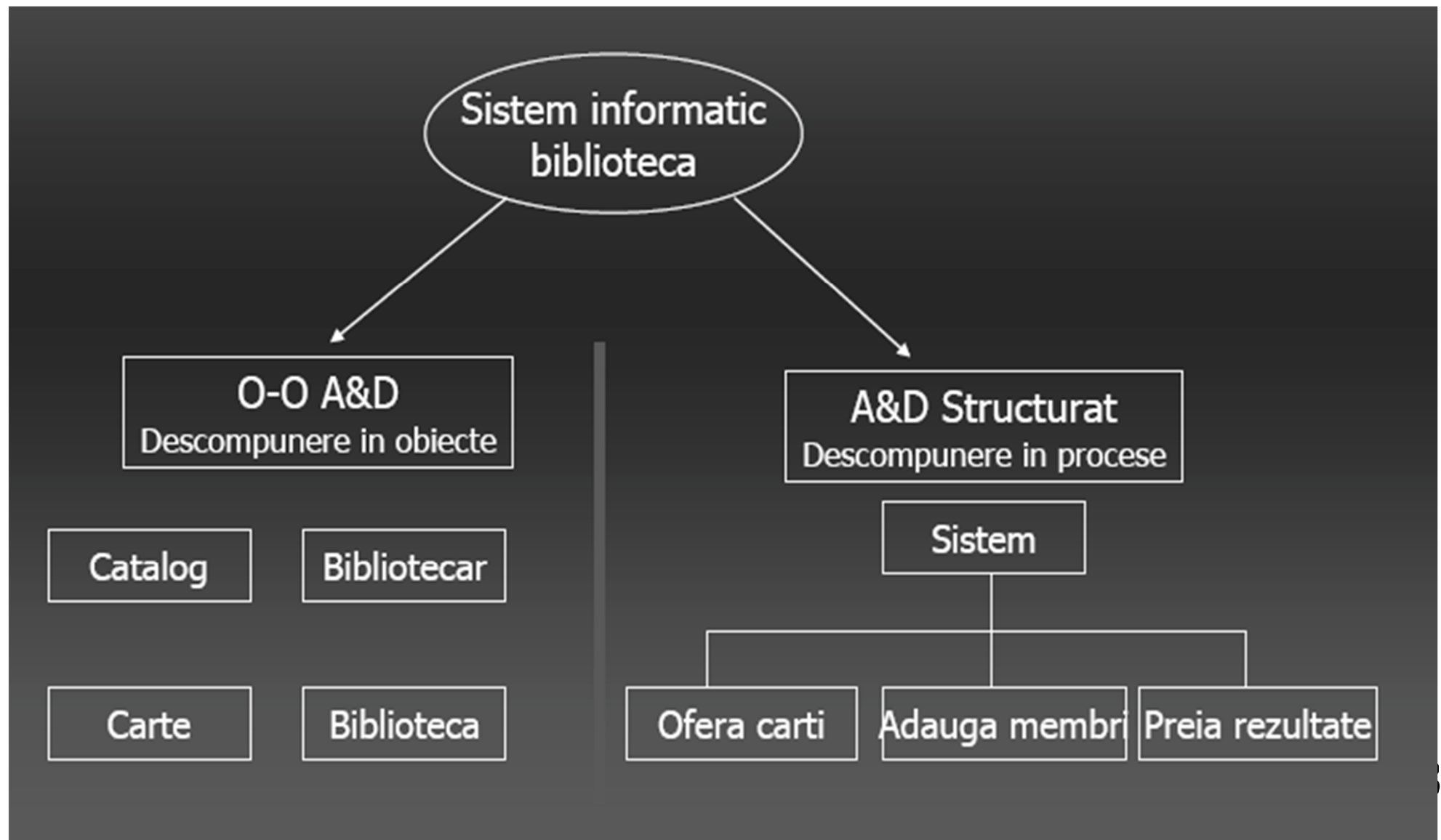


Ciclul de viata al *soft*-ului



● Unde se regaseste abordarea OO ?

Obiecte *vs.* functii

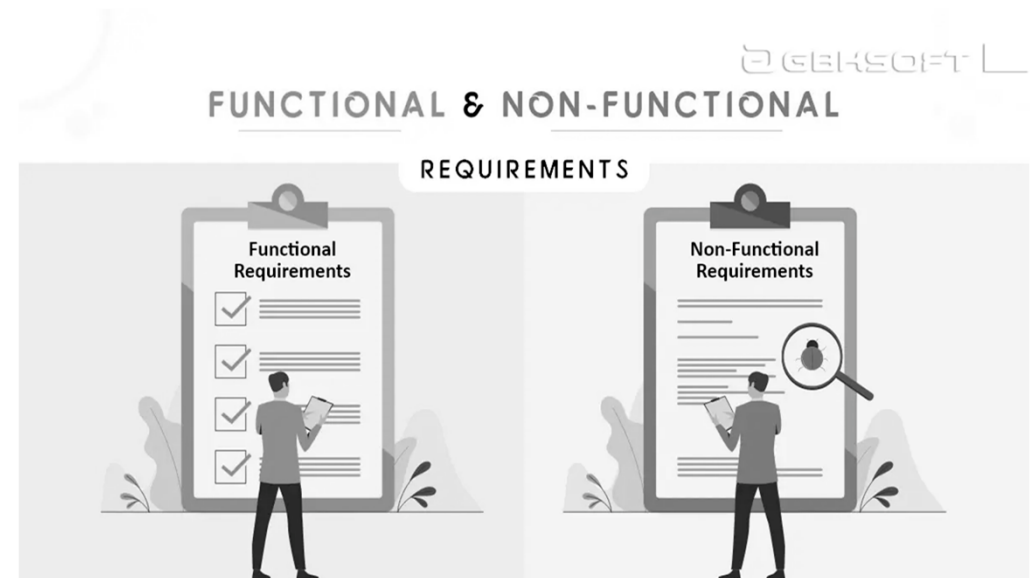


Cerintele dezvoltarii SI

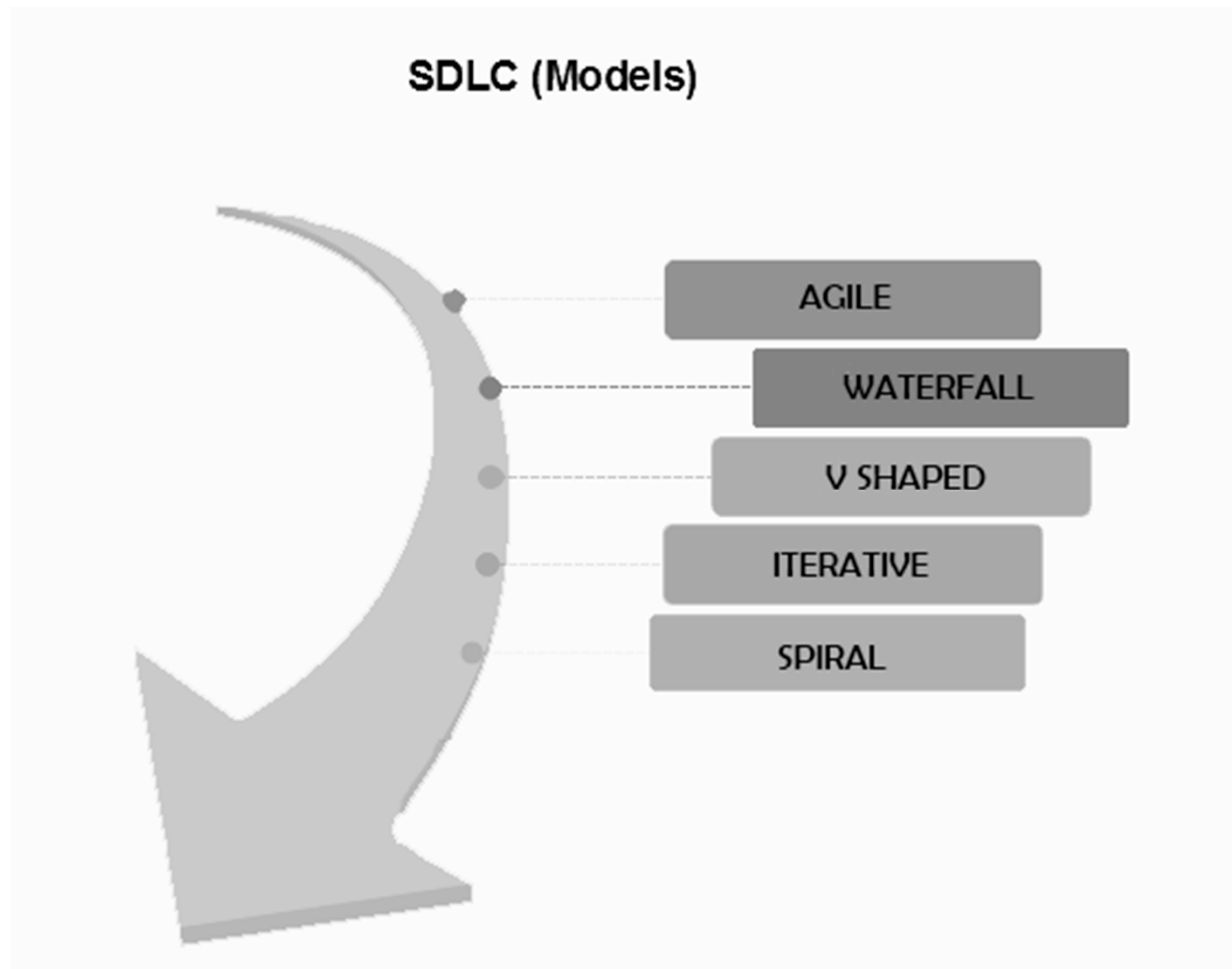
- Functionale

- Nefunctionale:

- Securitate / protectie
- Flexibilitate
- Robustete



Modele de dezvoltare (1)



Linear

Cascada

Agile

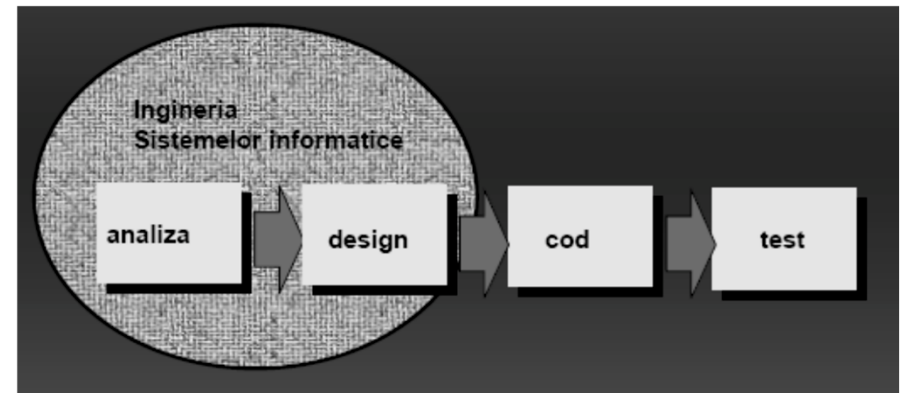
Spirala

V-model

Modele de dezvoltare (2)

● Modelul linear

1. Proiectele reale urmează rareori fluxul secvențial pe care îl propune modelul.
2. Este adesea dificil pentru client să precizeze toate cerințele în mod explicit. Modelul linear are dificultăți în adaptarea incertitudinii naturale care există la începutul multor proiecte.
3. Clientul trebuie să aibă răbdare. O versiune de lucru a programului (programelor) nu va fi disponibilă până târziu, pe parcursul implementării proiectului. O gafă majoră, dacă nu este detectată până când programul de lucru este revizuit, poate fi dezastruoasă.



Modele de dezvoltare (3)

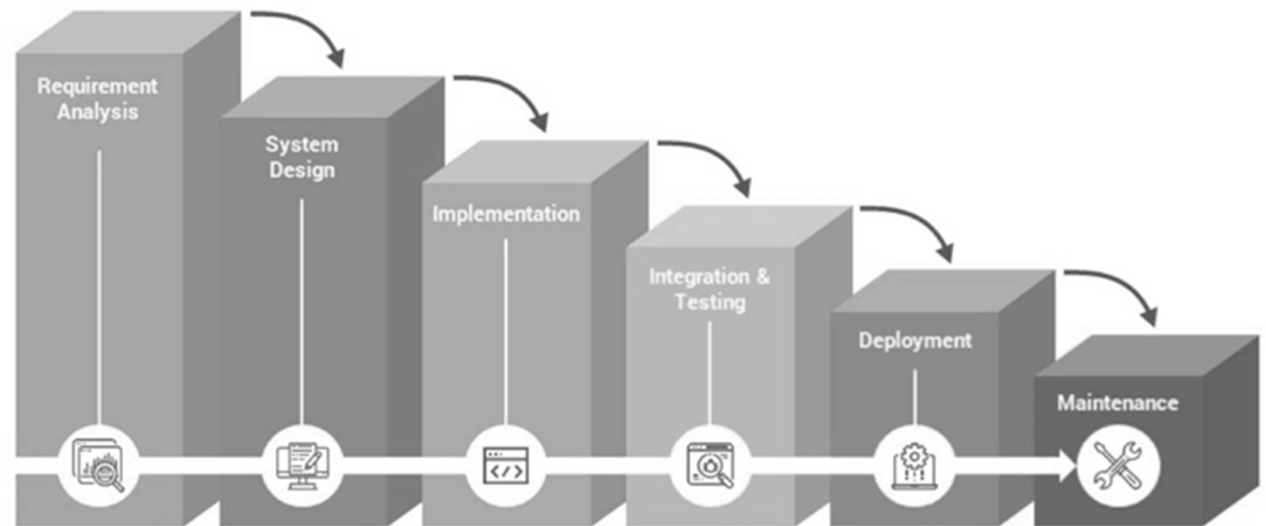
Modelul linear – *dezavantaje*:

- Controlul se modifica pe parcursul procesului
- Necesita ca toate constrangerile sa fie corect specificate inca de la inceputul procesului
- Versiunea functionala a unui program este livrata la sfasitul ciclului; erori in faze de inceput pot fi catastrofale.

Modele de dezvoltare (4)

Modelul cascada

ilustreaza procesul de dezvoltare software intr'un flux linear / secvential. Fiecare faza trebuie finalizata inainte ca urmatoarea faza sa poata incepe si nu exista suprapuneri intre faze.



Toate cerintele posibile sunt inregistrate in documentele privind cerintele produsului. Fiecare faza corespunde unei activitati si trebuie sa se termine la o anumita data prin producerea anumitor documente sau programe.

Modele de dezvoltare (5)

Cand se foloseste?

- ☐ Cerințele sunt foarte bine documentate, clare și fixe;
- ☐ Definiția produsului este stabilă;
- ☐ Tehnologia este înțeleasă și nu este dinamică;
- ☐ Nu există cerințe ambigue;
- ☐ Aplicația nu este complicată
- ☐ Când există un flux accentuat al personalului firmei;
- ☐ Resurse suficiente.



Modele de dezvoltare (6)

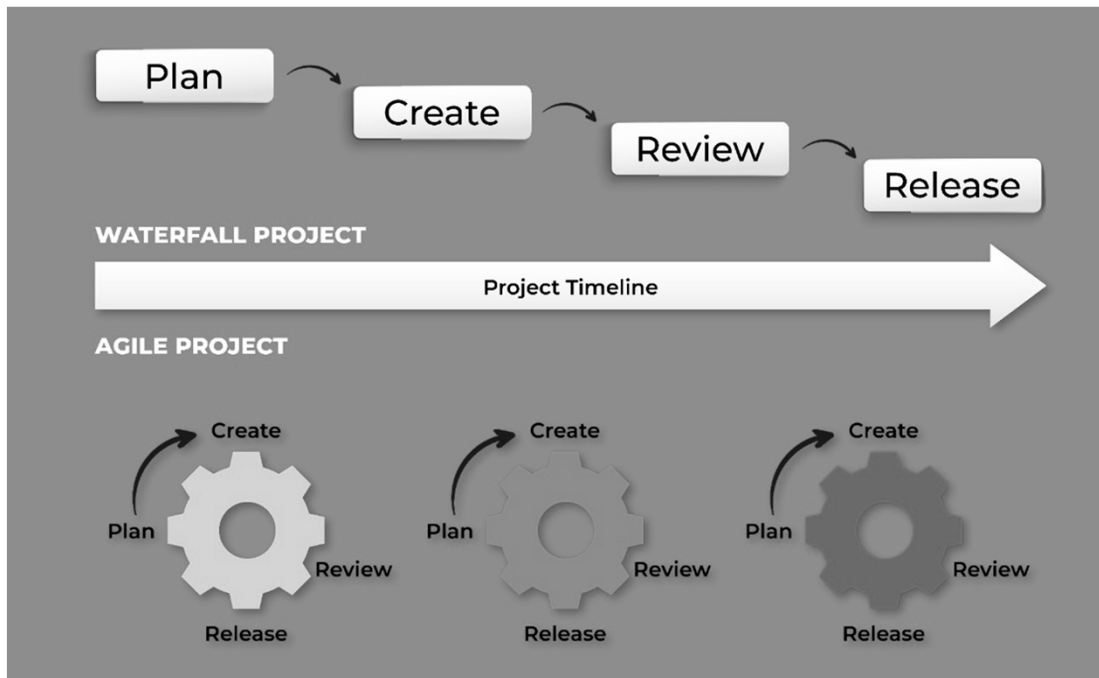
Modelul Agile se concentreaza pe adaptabilitatea procesului și satisfacția clienților prin livrarea rapidă a produselor software funcționale. Metodele Agile împart produsul în mici versiuni incrementale. Aceste versiuni sunt furnizate în iterații.



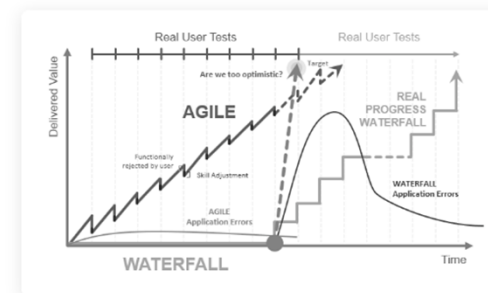
AGILE SOFTWARE DEVELOPMENT

Modele de dezvoltare (7)

Agile vs model cascada



AGILE VS WATERFALL



50%	AGILE Project challenge rate of 50%
53%	WATERFALL Project challenge rate of 53%

Slide 16

abc1

abc, 10/14/2022

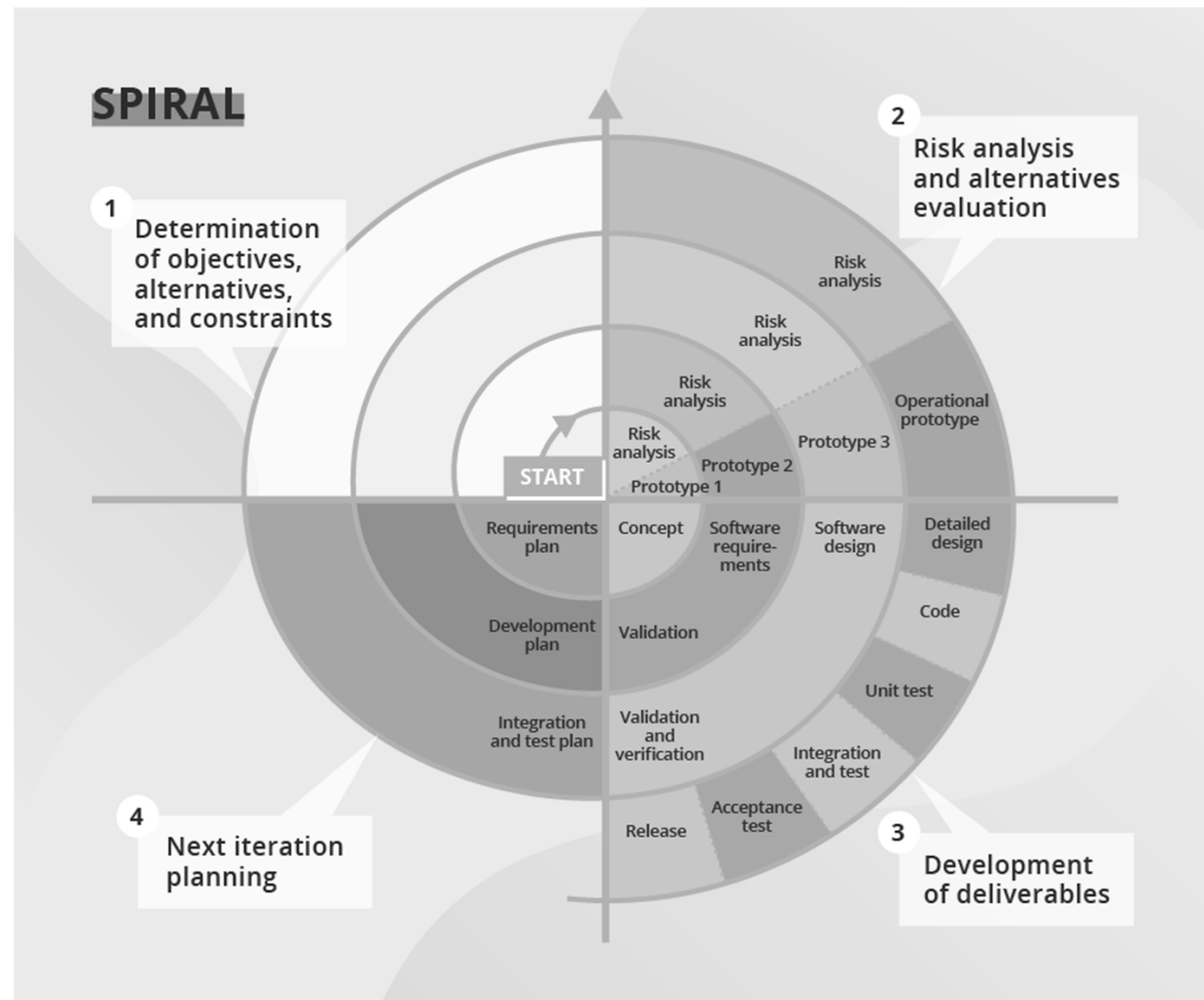
Modele de dezvoltare (8)

Un „model spiralat” este un tip de model iterativ în care iterațiile individuale din modelul spiralat iau forma unor mini-cascade.

Modelul în spirală oferă suport pentru gestionarea riscurilor.

Un risc este orice situație adversă care ar putea afecta realizarea cu succes a unui proiect.

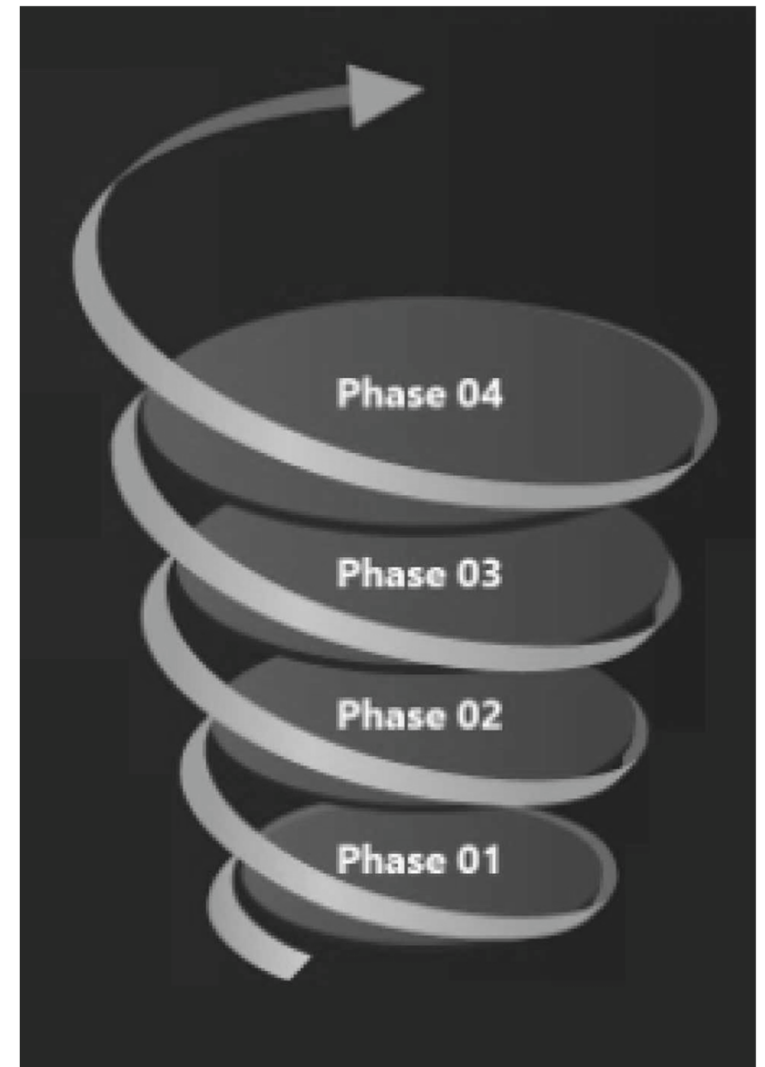
Cea mai importantă caracteristică a modelului spiralat o reprezintă gestionarea acestor riscuri necunoscute după începerea proiectului.



Modele de dezvoltare (9)

Cand se foloseste?

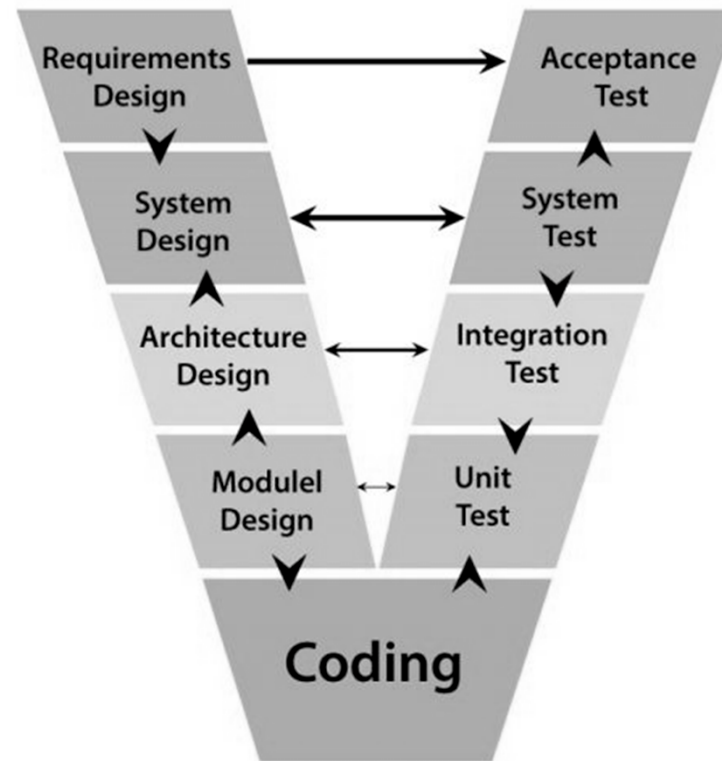
- ☐ Atunci când există o constrângere bugetară și evaluarea riscurilor este importantă.
- ☐ Pentru proiecte cu risc mediu spre mare.
- ☐ Angajamentul pe termen lung al proiectului din cauza potențialelor modificări ale priorităților economice pe măsură ce cerințele se schimbă în timp.
- ☐ Clientul nu este sigur de cerințele sale, ceea ce este de obicei cazul.
- ☐ Cerințele sunt complexe și necesită evaluare pentru a obține claritate.
- ☐ O nouă linie de produse ar trebui lansată în etape pentru a obține suficient feedback de la clienți.
- ☐ Se așteaptă modificări semnificative ale produsului în timpul ciclului de dezvoltare.



Modele de dezvoltare (10)

V-Model reprezintă un ciclu de viață de dezvoltare software care subliniază conceptul de „verificare și validare”. În fiecare etapă de dezvoltare a modelului V, va exista o fază de testare corespunzătoare care va valida un astfel de proces. Fazele de testare vor fi planificate în paralel cu dezvoltarea etapei pe care se presupune că vor fi testate

De asemenea, este considerat a fi o formă extinsă a modelului de cascadă, deoarece un pas nu poate fi făcut fără finalizarea unui proces anterior.



Modele de dezvoltare (11)

Costuri modificare

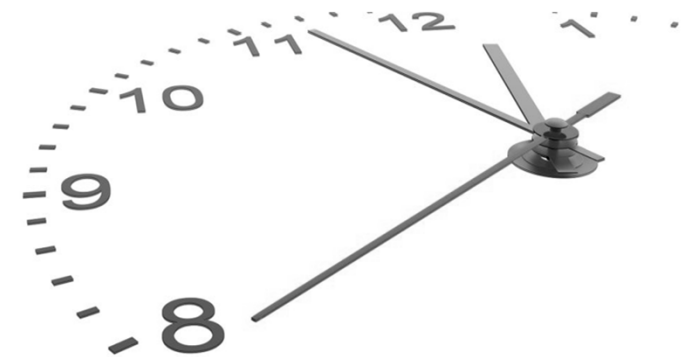
*Un sistem se modifica din ce
in ce mai greu odata cu
trecerea timpului*

Varsta sistemului

Cele 5 dimensiuni: scop, calitate, risc, timp, cost (1)



Scopul trebuie cât mai bine definit și detaliat astfel încât echipa să nu devieze de la acesta.



Timpul reprezintă resursa cea mai importantă, deoarece aceasta nu poate fi înlocuită. Orice deviere de la scop aduce timp irosit, iar acest lucru trebuie să fie acoperit în cadrul managementului de risc.

Cele 5 dimensiuni: scop, calitate, risc, timp, cost (2)



Calitatea reprezintă ceea ce clientul sau stakeholderii au nevoie astfel încât proiectul să fie considerat un succes. La bază, calitatea proiectului este reprezentată de posibilitatea utilizatorului de a folosi acel produs.



Managementul **riscului** reprezintă un proces continuu, proactiv și sistematic de identificare, evaluare și gestionare a riscurilor, realizat la nivelul fiecărei entități cu scopul de a furniza asigurări rezonabile în ceea ce privește atingerea obiectivelor.



Costul este o componentă primordială deoarece scopul principal al proiectului este generarea de profit.

Etapele dezvoltarii *soft*-ului (1)

- 1. Etapa de analiza** porneste de la specificarea problemei de rezolvat si are ca rezultat modelarea OO:
 - **Modelul obiectual:** pune in evidenta aspectele statice ale aplicatiei; sunt identificate conceptele cu care se lucreaza si li se asociaza clase si obiecte si relatiile aferente → diagrama de clase si diagrama de obiecte
 - **Modelul dinamic:** evidentiaza aspectele dinamice ale claselor / obiectelor care au un comportament dinamic semnificativ; sunt stabilite starile prin care poate trece un obiect, evenimentele ce determina tranzitia dintr'o stare in alta, etc.

Etapele dezvoltarii *soft*-ului (2)

- **Modelul functional:** descrie functionarea, i.e. proprietatile comportamentale ale obiectelor; sunt ilustrate relatiile dintre parametrii de I/O ale fiecărei fct, fara detalii privind implementarea algoritmului într'un limbaj de programare

2. Etapa de proiectare trece de la aspectele conceptuale la elementele legate de implementarea directa a produsului soft

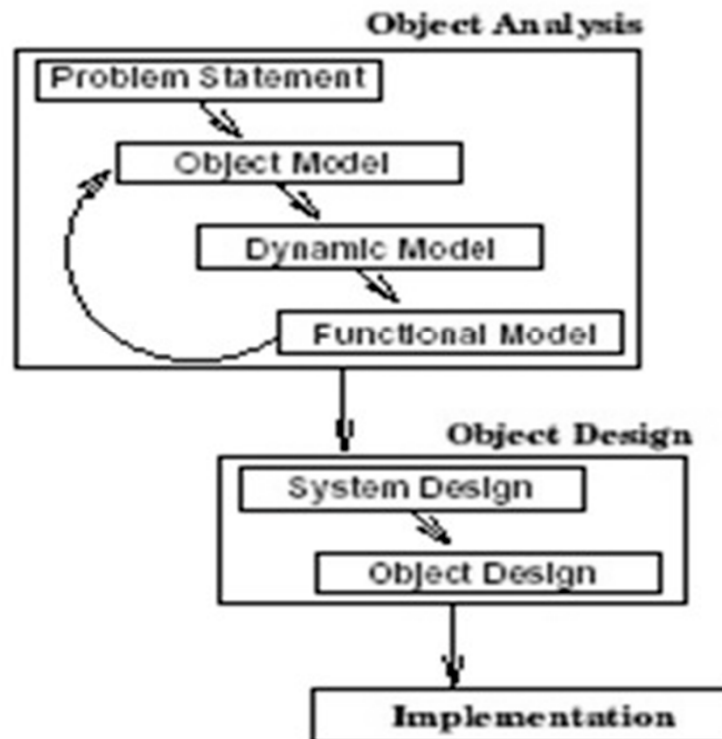
- **Proiectarea sistemului:** se iau decizii cu privire la arhitectura aplicatiei (resurse, platforma gazda) si modalitati de implementare (hard / soft)

Etapele dezvoltarii *soft*-ului (3)

- **Proiectarea obiectelor:** descrierea amanuntita a fiecărei clase. Pasi:
 - Structurarea diagramelor de interactiuni pentru fiecare scenariu
 - Structurarea detaliata a diagramelor de clase
- **Etapa de implementare** implica transpunerea proiectului intr'un limbaj de programare
- **Etapa de testare:**
 - Testarea modulelor *individuale*
 - Testarea *sistemului* (ca un tot unitar)

Etapele dezvoltarii *soft*-ului (4)

✈ OMT Methodology



Elaborarea Modelului Obiectual – MO (1)

- Pentru construirea MO se identifica intai clasele si asociatiile; se adauga attributele si se utilizeaza mostenirea
- Etape:
 - Se identifica clasele / obiectele; reguli:
 - ☐ Evitarea claselor redundante
 - ☐ Eliminarea claselor irelevante
 - ☐ Declararea ca attribute a proprietatilor celor mai relevante ale obiectelor
 - ☐ Se includ in clase operatiile ce sunt aplicate obiectelor
 - Pregatirea unui dictionar de date, i.e. descrierea precisa a fiecarui obiect, asociatii, atribut si operatii

Elaborarea Modelului Obiectual – MO (2)

- Identificarea relatiilor de asociere (orice dependenta intre doua / mai multe clase sau o referire de la o clasa la alta); reguli:
 - ➔ Se elimina asocierile irelevante
 - ➔ Se elimina asocierile redundante
 - ➔ Daca in procesul de analiza se elimina o clasa, asocierile corespunzatoare se elimina / se reformuleaza in functie de alte clase
 - ➔ Majoritatea asocierilor intre 3 / mai multe clase pot fi descompuse in asocieri binare
- Identificarea atributelor/operatiilor
- Organizarea claselor prin mostenire:
 - ➔ De jos in sus: generalizare in superclasa
 - ➔ De sus in jos: prin rafinarea claselor existente

Elaborarea Modelului Obiectual – MO (3)

- Iterarea modelului
- Gruparea claselor in module (o multime de clase ce reprezinta o submultime logica a intregului model)

Definitii:

- **Diagrama de clase:** reflecta instatieri posibile ale datelor
- **Diagrama de obiecte:** descrie modul in care inter-relacioneaza un set de obiecte
- **Relatia de agregare:** daca asociatia exprima o relatie de tip "parte-intreg"

Elaborarea Modelului Dinamic – MD (1)

Are la baza 2 concepte:

● **Stare:**

- Caracterizeaza setul de valori asociate obiectului
- Este o abstractizare a valorilor atributelor si legaturilor unui obiect
- Reprezinta raspunsul unui obiect la un eveniment (extern)
- Corespunde intervalului dintre 2 evenimente

Elaborarea Modelului Dinamic – MD (2)

● **Eveniment:**

- Caracterizeaza stimulii externi ce actioneaza asupra obiectelor.
- Reprezinta o transmitere unidirectionala de informatie (de la un obiect la altul).
- Special: erori
- **Concurenta:** doua sau mai multe evenimente necorelate cauzal
- **Scenariu:** secventa de evenimente ce apare intr'un caz particular de evolutie a sistemului.

Elaborarea Modelului Dinamic – MD (3)

- Operatii:
 - *Activitati*: Sunt asociate starilor si necesita timp pentru a fi efectuate.
 - *Actiuni*: Operatii instantanee asociate evenimentelor.
- MD al unei clase este mostenit de subclasele sale; acestea mostenesc atat starile cat si tranzitiile

Elaborarea Modelului Functional – MF

- Descrie modul in care se efectueaza calculele in interiorul unui sistem, fara a tine cont de secvente, decizii sau structuri
- **Proces:** transforma valorile datelor; rezultatele unui proces depind de comportamentul sistemului, specificat prin MD
- **Flux de date:** conecteaza iesirea unui proces cu intrarea altui proces

Ciclul de dezvoltare software

