# *MODELE DE DEZVOLTARE SOFTWARE -2*

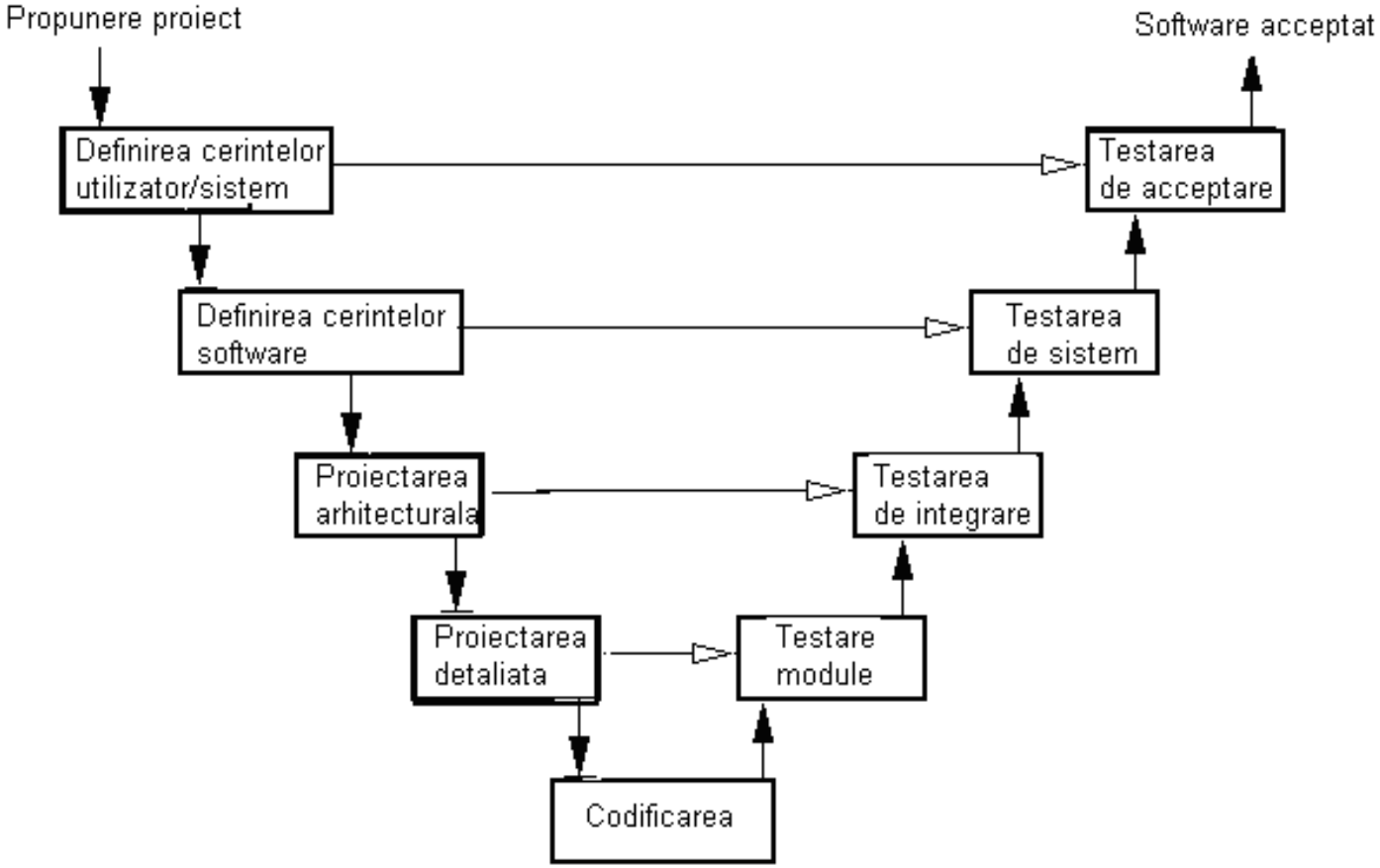
*Prof. univ. dr. ing. Florica Moldoveanu*

*Ingineria programelor*

UPB, Automatică şi Calculatoare

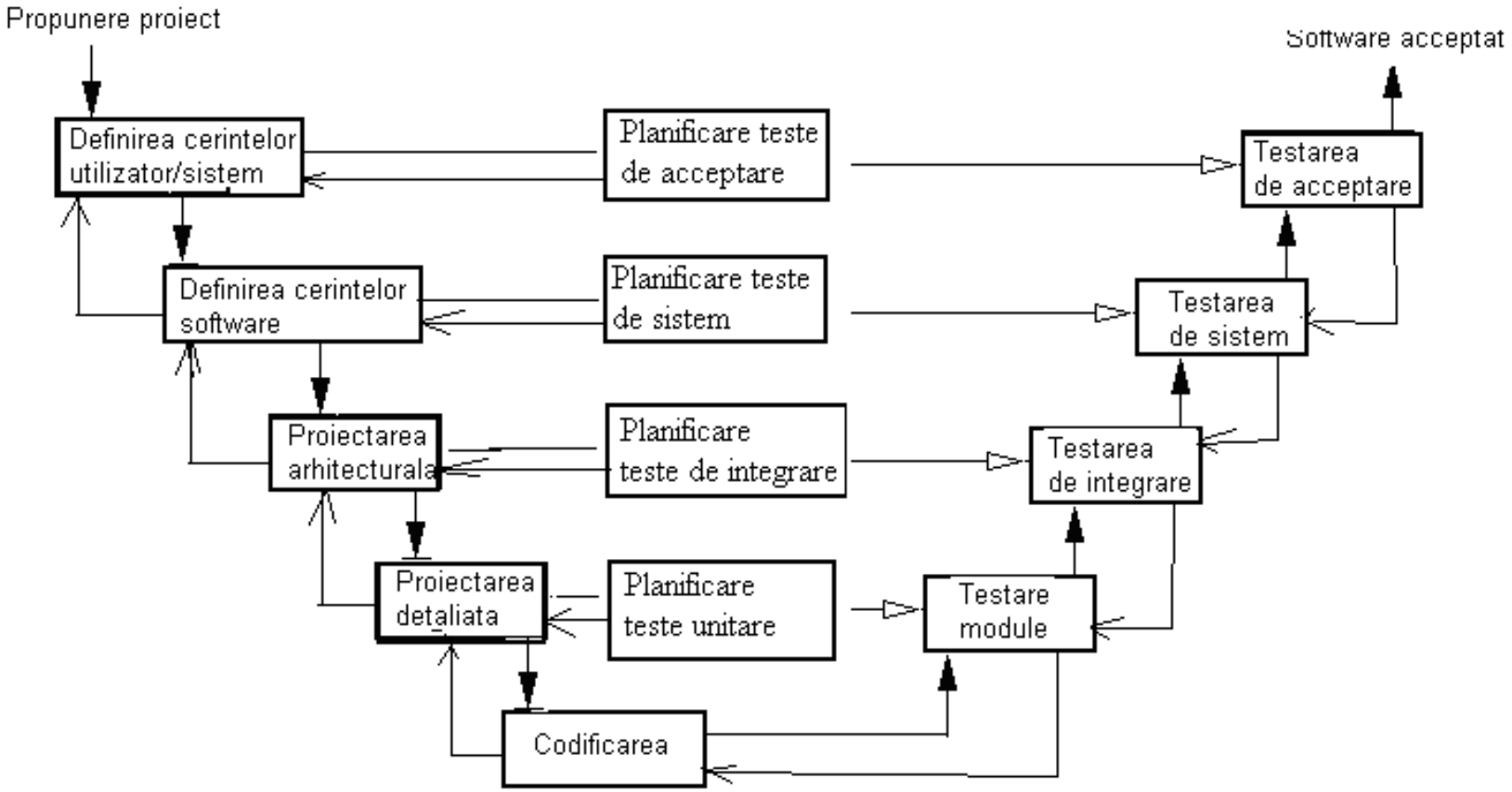
2020-2021

# Modelul “In V” (V model)



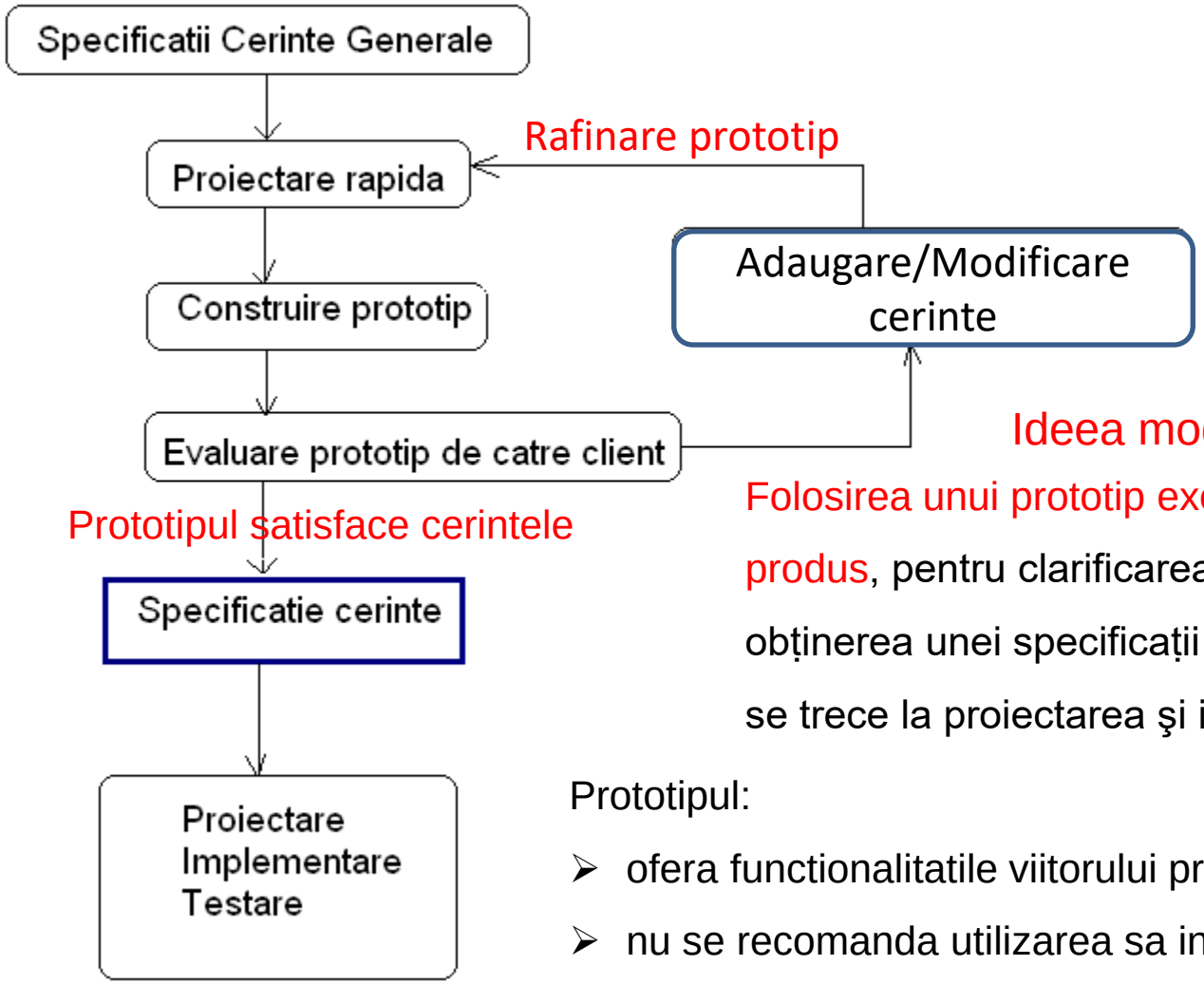

- Este o varianta a modelului cascada, care include elaborarea planurilor de test in fazele de specificare si proiectare.
- Săgeţile orizontale evidenţiază corelarea dintre etapele de creare a artefactelor software (documente, cod) şi cele de verificare.

## MODELUL "IN V" detaliat




Avantaj fata de modelul cascada: şanse de succes mai mari datorita elaborarii planurilor de test în primele etape ale procesului de dezvoltare.

# Dezvoltarea pe baza de prototip



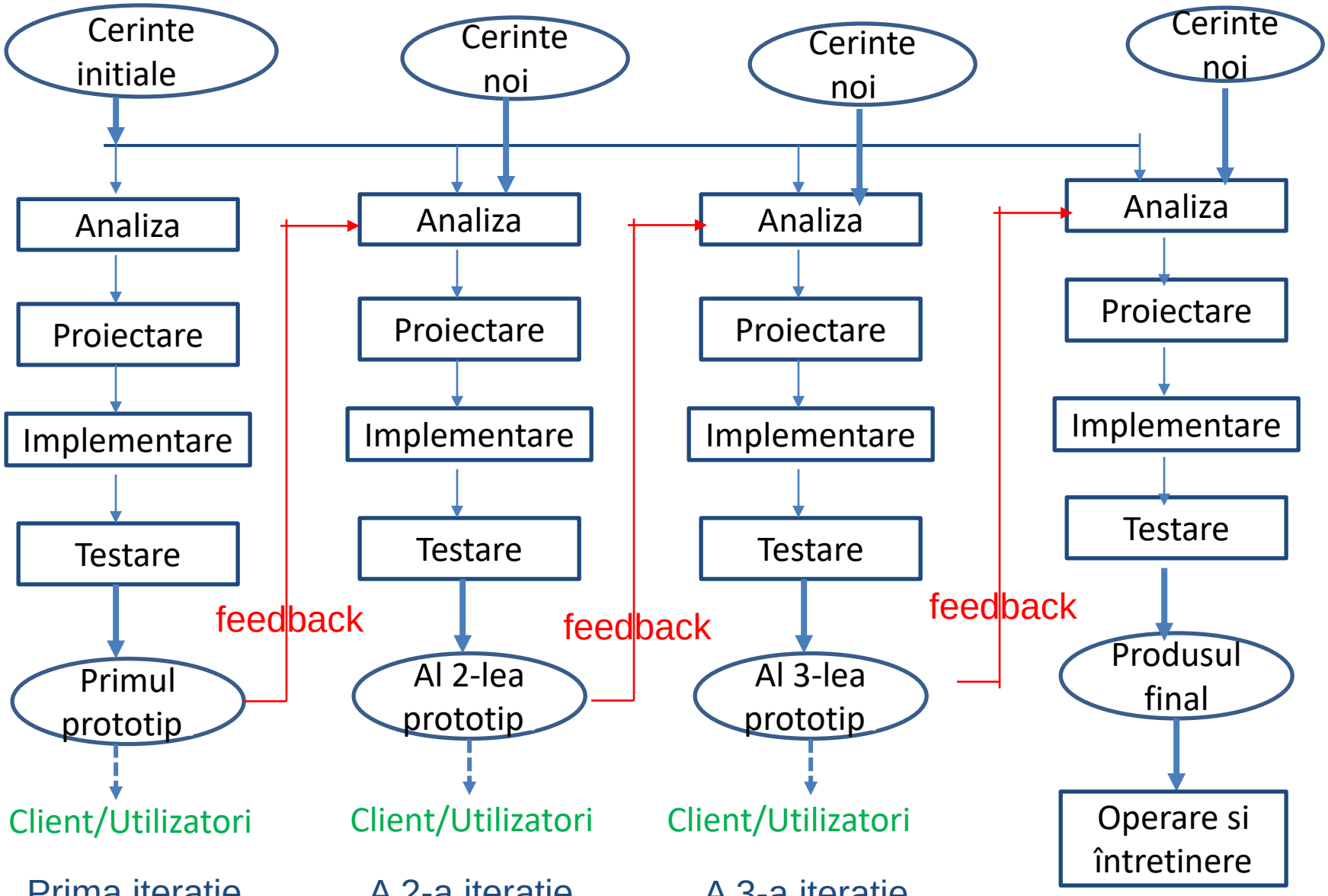

Ideea modelului:

Folosirea unui prototip executabil al viitorului produs, pentru clarificarea cerintelor utilizator şi obținerea unei specificații a cerințelor înainte de a se trece la proiectarea şi implementarea produsului

Prototipul:

- ofera functionalitatile viitorului produs si interfata utilizator
- nu se recomanda utilizarea sa in dezvoltarea produsului final, chiar daca este implementat in acelasi limbaj

# Modelul Iterativ si Incremental (1)



# Modelul Iterativ si Incremental (2)

## Caracteristici generale

- Produsul software este dezvoltat in mai  multe iteratii.
- Fiecare iteratie include un “ciclu de dezvoltare cascada” si se incheie cu un produs executabil care poate fi livrat clientului.
- In fiecare iteratie se adauga noi functionalitati (cerinte) la ultimul prototip:
  - Cerinte prevazute initial
  - Cerinte noi – aparute pe parcursul iteratiei anterioare
- Analiza cerințelor într-o iterație include feedback-ul la utilizarea prototipului anterior.

# Modelul Iterativ si Incremental (3)

**Iteratiile**

❖ Scopul fiecarei iteratii este de a produce un produs executabil prin care se poate demonstra partilor interesate în proiect o parte dintre functionalitatile viitorului produs.

❖ Durata unei iteratii depinde de modelul de dezvoltare şi tipul de proiect; poate fi de cateva saptamani, una sau mai multe luni.

❖ Cu cat o iteratie este mai scurta cu atat mai repede se obtine feedback-ul partilor interesate.

❖Pentru prima iteratie se alege un subset de cerinte care corespund cazurilor de utilizare principale ale produsului (pun in evidenta aspectele cheie ale produsului).

❖ In fiecare iteratie se implementeaza un subset de cerinte prevazute initial dar si cerinte noi (neprevazute initial), tinand cont si de feedback-ul partilor interesate in proiect: client, utilizatori finali, alții.

❖ Obiectivele unei iteratii se stabilesc pe baza evaluarii iteratiilor precedente.

# Modelul Iterativ si Incremental (4)

**Iteratiile (cont)**

❖ In fiecare iteratie se creaza un nou prototip prin modificarea prototipului anterior → proiectare, codificare, testare.

➢ Arhitectura produsului software trebuie sa fie flexibila la schimbari ( extensibila, sa nu presupuna modificarea sa la fiecare noua iteratie)

❖ Analiza intr-o iteratie trebuie sa includa evaluarea evolutiei produsului: evolutia numarului de defecte descoperite in fiecare nou prototip, a complexitatii codului si a arhitecturii, a efortului de actualizare.

➢ Metricile sunt suportul pentru determinarea eficientei procesului si a calitatii produsului: sunt importante nu numai valorile lor absolute ci si evolutia lor in timp.

❖Documentatia este construita treptat, in timpul fiecarei iteratii.

# Modelul Iterativ si Incremental (5)

**Dezvoltarea iterativă şi incremetală poate fi:**

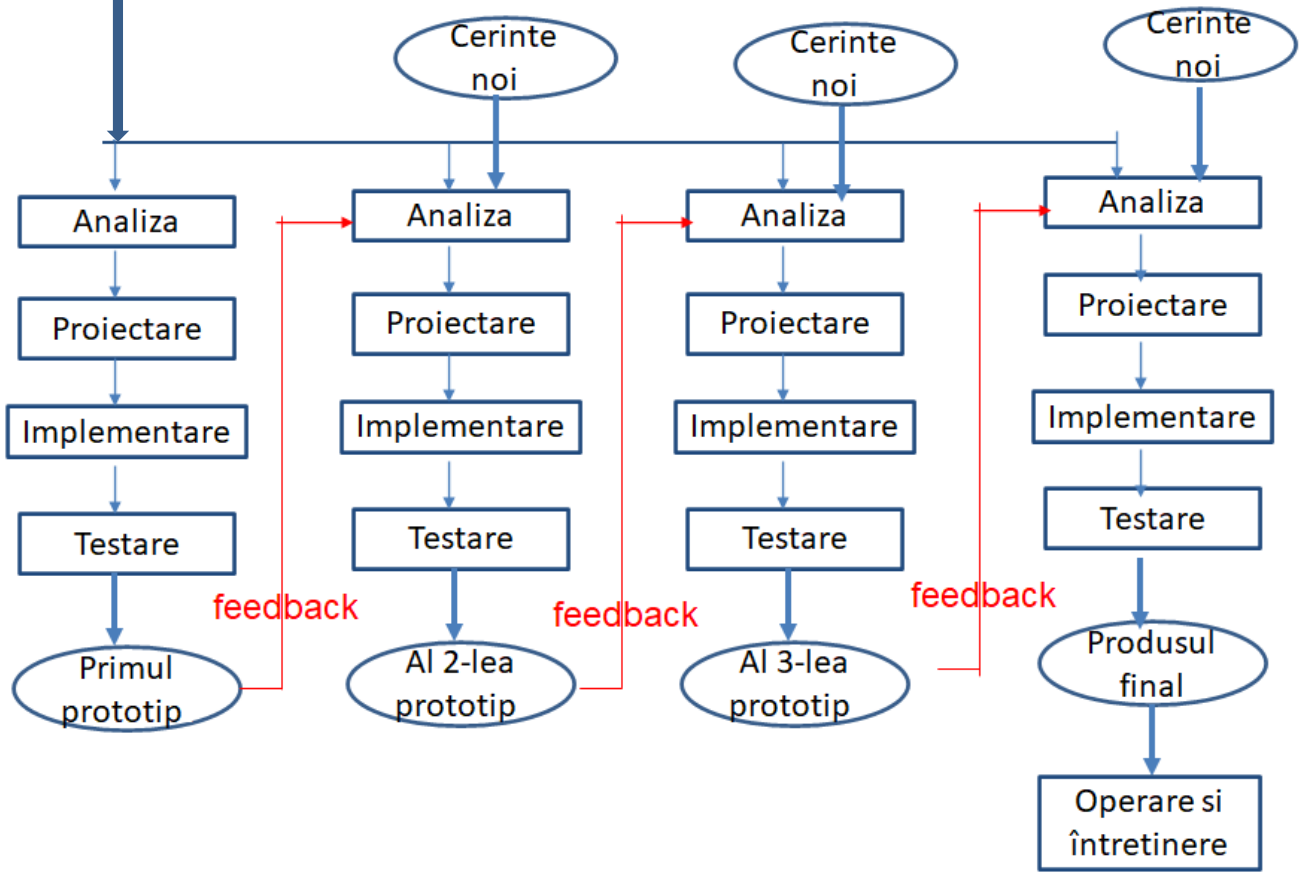
- Proces planificat – Exemplu: RUP (Rational Unified Process)

  Inainte de începerea iterațiilor se efectueaza o analiza de nivel inalt:

  - Se analizeaza: scopul sistemului, obiectivele de nivel înalt, viitorii utilizatori, principalele functionalitati si constrangeri de operare
  - Se definesc cerintele de nivel inalt ale sistemului
  - Se planifica iteratiile si se aloca cerintele pe iteratii
  - Se defineste o arhitectura software de nivel inalt
  - Durata iteratiilor nu este fixa: saptamani – luni, in functie de activitatile planificate
- Proces "agil" – Exemple: SCRUM, FDD (Feature-Driven Development), XP, ş.a.
  - Nu se pune accentul pe planificarea iteratiilor, cerintele sunt colectate pe parcurs
  - Iteratiile sunt foarte scurte (1-2 saptamani), indiferent de proiect
  - Se pune accentul pe feedbackul clientului/utilizatorilor
  - Clientul este implicat in procesul de dezvoltare (stabilirea obiectivelor urmatoarei iteratii, prioritizarea cerintelor)
  - Documentaţie minimală

# Dezvoltarea iterativa – process planificat (1)

**Analiza de nivel inalt**
-Se defineste o "viziune" asupra sistemului
-Se analizeaza "domeniul problemei"/mediul de operare
-Se definesc cerintele de nivel inalt ale sistemului
-Se planifica iteratiile si se aloca cerintele pe iteratii
-Se defineste o arhitectura software de nivel inalt



# Dezvoltarea iterativa – process planificat (2)

**Analiza de nivel inalt**

❖ **Se defineste o "viziune" asupra sistemului**: scopul său, obiectivele de nivel înalt, cine îl va utiliza, principalele functionalitati si constrangeri de operare.

❖ In unele cazuri, **se analizeaza "domeniul problemei"** (business modeling). Scopul:

-de a se intelege structura si dinamica organizatiei tinta (in care va fi implementat sistemul), problemele din organizatia tinta si imbunatatirile potentiale;

-de a se asigura o aceeasi intelegere a organizatiei tinta de catre client, utilizatorii finali si dezvoltatori;

-de a se deriva cerintele de sistem in scopul sprijinirii organizatiei tinta

❖ **Se definesc cazurile de utilizare ale viitorului sistem si actorii** (cei care vor interactiona cu sistemul). Cazurile de utilizare exprima cerintele functionale de nivel inalt ale sistemului.

❖ **Se dezvolta un plan initial al proiectului:** se impart cerintele in subseturi care vor fi implementate in diferite iteratii, se prioritizeaza cerintele.

❖ **Se defineste o arhitectura de nivel inalt a produsului software.**

# Rational Unified Process (RUP)

❑ Rational Unified Process este un **proces de inginerie software**, dezvoltat de corporatia Rational Software (devenita divizie IBM din 2003) si integrat cu suita de instrumente de dezvoltare oferita de IBM.

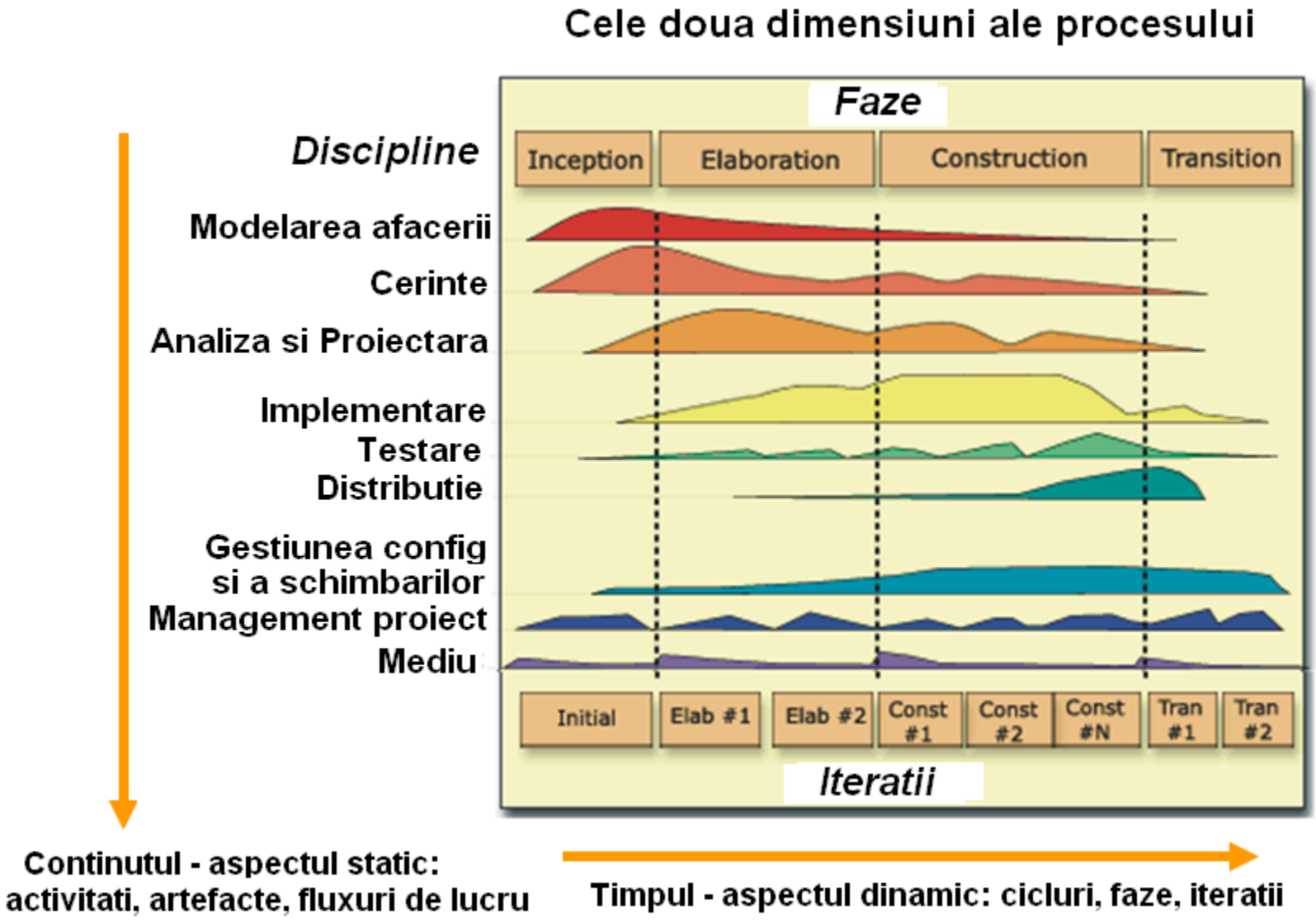
❑ **Furnizeaza o abordare disciplinata** de asignare a sarcinilor si responsabilitatilor intr-o organizatie care dezvolta software.

❑ RUP este o **platforma proces** care poate fi adaptata si extinsa pentru a satisface necesitatile organizatiei care o adopta.

❑ **Proces de dezvoltare iterativă, dirijat de urmatoarele principii**

- Diminuarea timpurie a riscurilor:
  - Primele iteratii trebuie sa adreseze riscurile cele mai mari
  - Riscul trebuie evaluat pe tot parcursul dezvoltarii: riscul se schimba in timp
  - Lista riscurilor actualizată este intrare pentru dezvoltarea planului iteratiei urmatoare
- Definirea timpurie a arhitecturii: in primele iteratii, testata prin prototipuri executabile
- Utilizarea de metrici obiective pentru evolutia procesului si a produsului dezvoltat

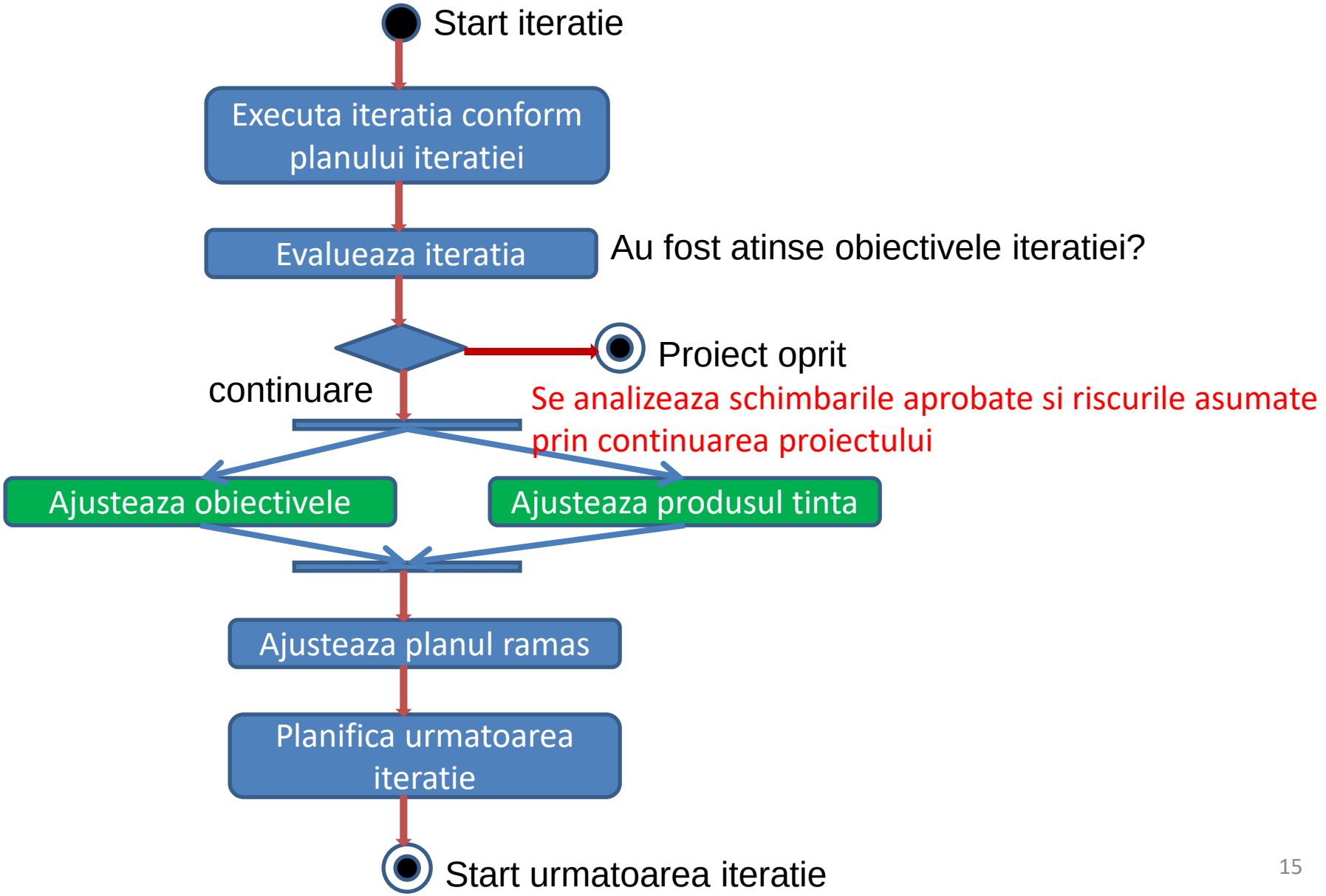
# Dezvoltarea iterativa in RUP(1)



# Dezvoltarea iterativa in RUP(2)

- **Procesul de dezvoltare a unui produs software consta din mai multe cicluri de dezvoltare (iteratii)**, la sfarsitul fiecarui ciclu obtinandu-se un **produs executabil** care poate fi distribuit utilizatorilor. El satisface anumite cerinte specificate.
- **Un ciclu de dezvoltare este divizat in 4 faze consecutive**:
  - Faza de inceput (Inception) –se dezvolta o viziune asupra produsului final si o descriere a afacerii
  - Elaborarea (Elaboration) - se specifica cerintele produsului si se proiecteaza arhitectura
  - Construirea (Construction) - se construieste produsul adaugand toate componentele la arhitectura
  - Tranzitia (Transition) - produsul software este transferat comunitatii de utilizatori
- **Fiecare faza se incheie cu un jalon (milestone)** - se verifica satisfacerea anumitor criterii si se decide: intreruperea proiectului, refacerea fazei, trecerea la faza urmatoare
- **Fiecare faza se poate desfasura in mai multe iteratii**

# Desfasurarea unei iteratii in RUP



# Modelul Iterativ si Incremental - Concluzii

**Avantaje:**

• In fiecare iteratie se obtine un produs executabil, care satisface o parte din cerintele utilizator. Opus modelului cascada in care un produs executabil este disponibil la sfarsitul procesului de dezvoltare.

• Prototipurile pot fi livrate clientului/utilizatorilor:

→ feedback-ul partilor interesate in proiect (utilizatori, client, s.a) este distribuit pe intreg parcursul dezvoltarii.

• Flexibil la schimbarea cerintelor: cerintele noi sau modificate pot fi incoporate in urmatorul prototip.

• Depanarea si testarea mai usor de efectuat intr-o iteratie – se testeaza cerintele adaugate

• Produsul este mai fiabil decat intr-o dezvoltare in cascada: aspectele cele mai importante ale sistemului sunt cel mai mult testate.

• Riscurile de eşuare a proiectului sunt reduse.

**Dezavantaje (riscuri) – in special intr-un proces neplanificat**

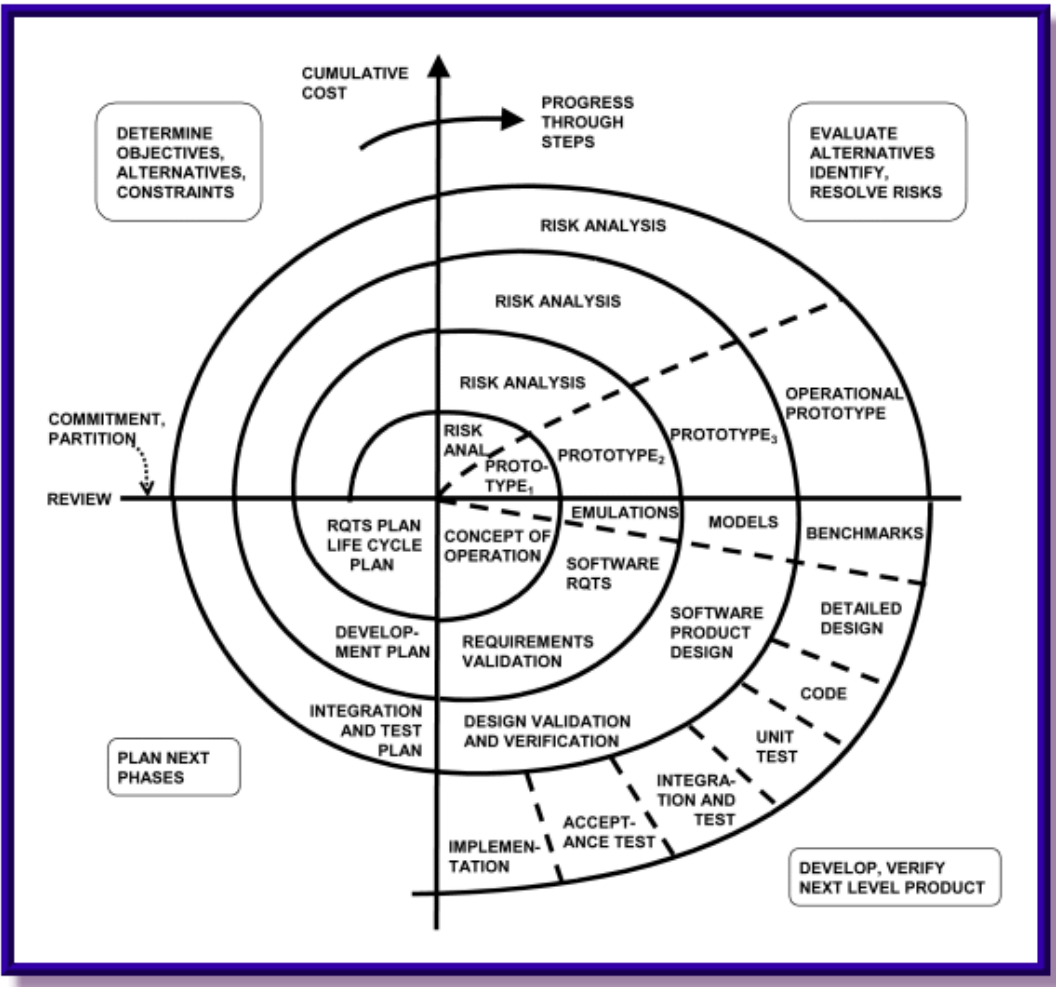
• Arhitectura initiala a produsului software poate fi degradata → testare dificila, intretinere dificila

• Abordarea incrementala se poate transforma usor intr-una ‘construieste si repara’ (build and fix)→ afecteaza calitatea produsului final: numarul de defecte ramase este mare, intretinerea dificila, s.a.

# Modelul in Spirala

-Model focalizat pe analiza riscurilor.
-Combina ideea prototiparii cu dezvoltarea in cascada si dezvoltarea iterativa
-Adecvat proiectelor mari si critice.

Boehm, 1986: http://en.wikipedia.org/wiki/Spiral_model




- Fiecare iteratie (ciclu de dezvoltare) consta in repetarea a 4 faze si se termina cu un prototip al viitorului produs:

1. **Determinarea obiectivelor,** a constrangerilor, generarea alternativelor. Se colecteaza si studiaza cerintele, prin comunicare continua cu clientul.
2. **Evaluarea alternativelor**: analiza riscurilor, construirea de prototipuri.
3. **Dezvoltarea si verificarea urmatorului nivel al produsului:**

1) cerinte utilizator; 2) cerinte software;

3) proiectarea; 4) implementarea, integrarea, testarea

4. Evaluarea versiunii curente a produsului.

**Planificarea urmatorului ciclu.**

La sfarsitul unei iteratii, prototipul obtinut este evaluat de client. Se decide daca se va efectua o noua iteratie.

# Lecturi recomandate


1. https://acodez.in/12-best-software-development-methodologies-pros-cons/
2. The seductive and dangerous V Model: http://www.clarotesting.com/page11.htm
3. Modern Methods of Software Development: https://task.gda.pl/files/quart/TQ2015/04/tq419v-c.pdf
4. Rational Unified Process: A Best Practices Approach: http://www.eecg.toronto.edu/~jacobsen/courses/ece1770/slides/rup.pdf
5. Kruchten, P. ,The Rational Unified Process: An Introduction**,** Third Edition, Addison-Wesley, Pearson Education, Inc., NJ, 2004
6. https://www.geeksforgeeks.org/software-engineering-spiral-model/