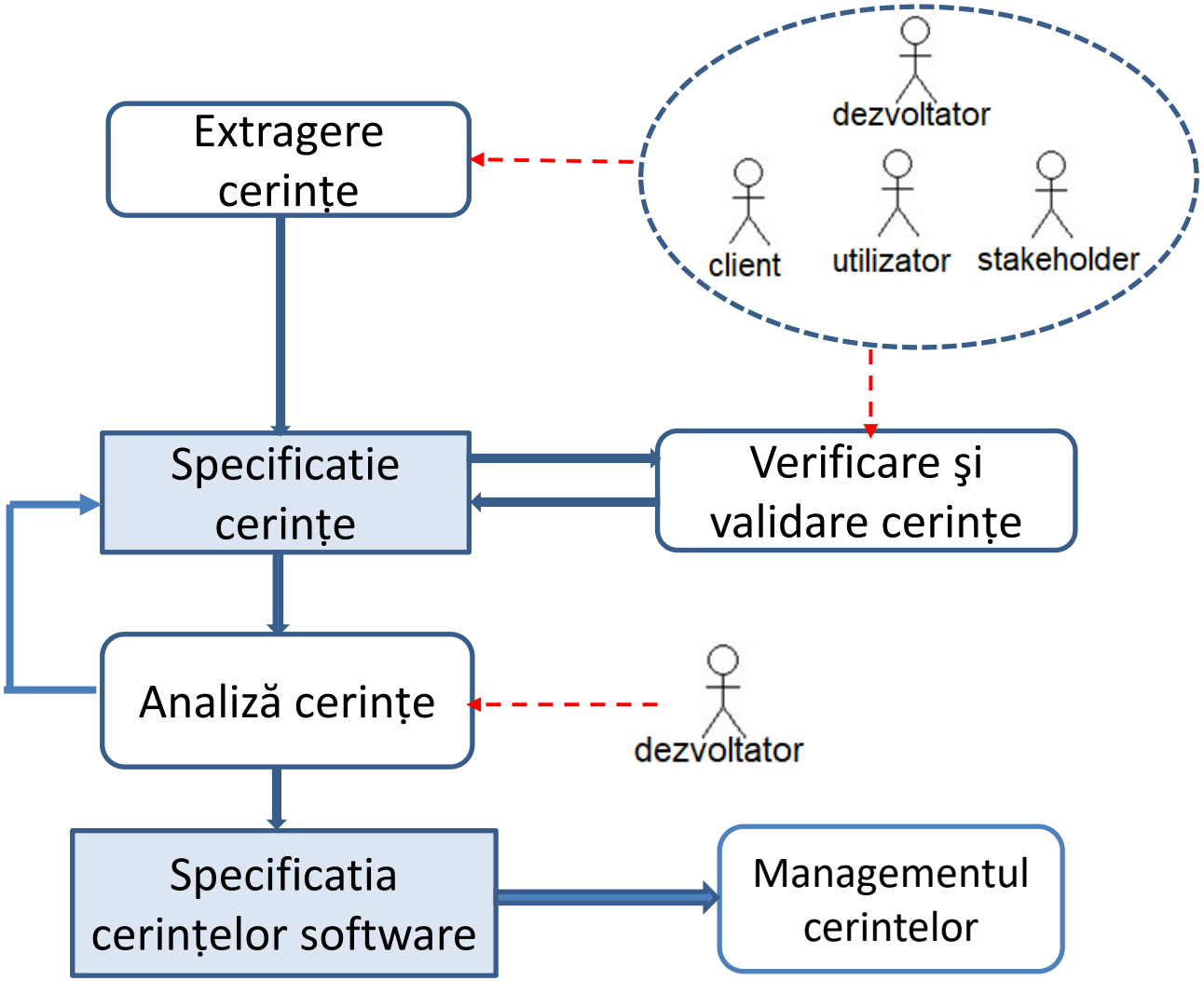
Procesul cerințelor în dezvoltarea software

Prof. univ. dr. ing. Florica Moldoveanu

Curs *Ingineria programelor* – UPB, Automatică şi Calculatoare
2020-2021

Procesul cerintelor



Verificarea si validarea cerintelor

- Verificarile sunt efectuate pe tot parcursul perioadei de specificare a cerintelor (poate fi intreaga durata de dezvoltare)
- Prin **validare** se urmareste sa se stabileasca împreuna cu toate partile interesate in proiect că:
  - cerintele sunt **complete, consistente, neambigue si corecte;**
  - au fost **bine întelese şi în acelaşi fel** de catre toate partile interesate în proiect;
  - satisfac standardele impuse.
- Modul uzual de verificare si validare consta in **revizia documentului de specificare** de catre un grup din care fac parte reprezentanti ai dezvoltatorului, clientul şi reprezentanti ai utilizatorilor.

Analiza cerintelor (1)

- Este responsabilitatea dezvoltatorului.
- **Participanti**: ingineri de sistem, ingineri software, analisti, manageri, persoane responsabile de asigurarea calitatii.
- **Scop:**
  - **Descoperirea erorilor** în cerintele formulate: cerintele sunt actualizate apoi revazute împreuna cu clientul si celelalte parti participante la extragerea cerintelor.
  - **Rezolvarea conflictelor dintre cerintele** diferitelor tipuri de utilizatori sau dintre cerinte functionale si ne-functionale.
  - **Clasificarea cerintelor dupa diferite criterii**:
    - Operationale, de calitate, cerinte de produs/proces, etc.
    - Importanța, prioritatea intr-o dezvoltare iterativa
    - Stabilitatea (modificarea lor pe parcursul dezvoltarii sau chiar si dupa livrare)
  - **Structurarea si formalizarea cerintelor extrase** – permite descoperirea erorilor in cerintele formulate, identificarea si rezolvarea problemelor dificile la începutul procesului de dezvoltare.

Analiza cerintelor (2)

- **In etapa de analiza cerintele sunt structurate si formalizate in modelul de analiza.**
- **Modelul de analiza - model conceptual al sistemului. Cuprinde:**
  - **Modelul obiect**, alcatuit din **clase, diagrame de clase si diagrame de obiecte**.

- **Obiectele** din modelul de analiza sunt entitati din domeniul aplicatiei (obiecte conceptuale).

Exemple: cărți, utilizatori, abonati in aplicatia SGCB;

studenti, profesori, cataloage, cursuri – intr-o aplicatie de planificare a studentilor la cursuri

- **Clasele** din modelul de analiza sunt clase conceptuale.

- In etapa de proiectare se decide cum vor fi implementate clasele: clase intr-un lb. POO sau tabele ale unei baze de date relationale.

- **O diagrama de clase conceptuala** poate fi schema conceptuala a unei baze de date

Analiza cerintelor (3)

- **Modelul dinamic**, alcatuit din **diagrame de interactiune si diagrame de stari**

- **O diagrama de interactiune** (de secventa sau de colaborare) reprezinta **interactiunile** dintre un actor şi sistem sau interactiuni între obiecte conceptuale, **în timpul unui singur scenariu**.

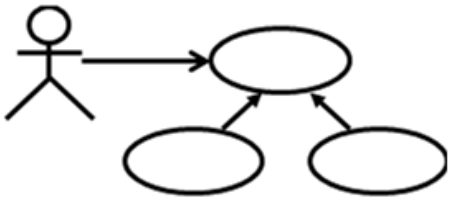
- Interactiunile sunt reprezentate prin mesaje schimbate intre participanti

- **O diagrama de stari** reprezinta **comportamentul in timp** al obiectelor dintr-o clasa de obiecte sau comportamentul in timp al intregului sistem.

- Comportamentul in timp este redat prin stări, tranzitii intre stari si evenimente care determina tranzitiile.

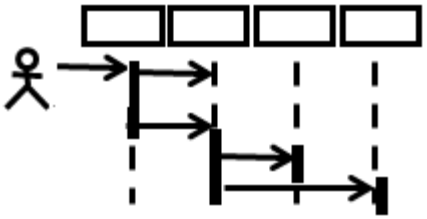
Analiza cerintelor (4)

Specificaţia funcţionala

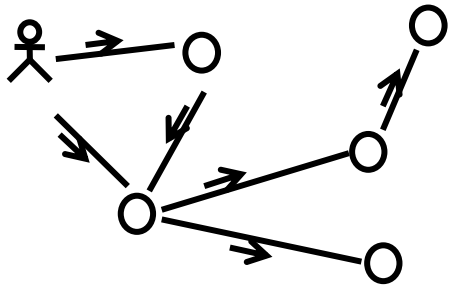


Modelul cazurilor de utilizare

Obiecte conceptuale

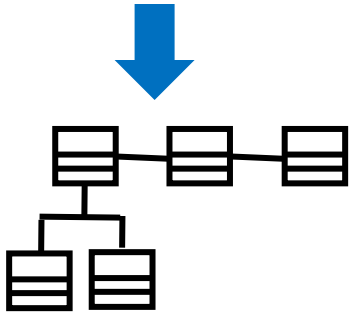


Diagrame de secventa care modeleaza scenarii



Diagrame de colaborare (de comunicare, in UML 2.x)

→ In analiza orientata obiect, se trece de la descrierea functionala a cerintelor structurata în raport cu actorii la o structurare obiect, pornind de la scenarii si cazuri de utilizare.



Diagrame de clase conceptuale

Managementul cerintelor

Scop: Managementul schimbarilor cerintelor pe parcursul dezvoltarii

- Pot aparea cerinte noi
- Se pot schimba cele existente →cerintele trebuie sa fie trasabile
- Prioritatile cerintelor se pot schimba pe parcursul dezvoltarii
- Mediul de operare si cel tehnic se pot modifica pe parcusul dezvoltarii

Managementul cerintelor:

- Include activitati de verificare si validare
- Necesita instrumente software care sa asiste procesul cerintelor

Cat timp sa se consume cu definirea cerintelor?

- Timpul consumat **depinde de modelul de dezvoltare** (scurt in modelele agile).
- Aproximativ 20-25% din timpul total al proiectului trebuie alocat extragerii, analizei si specificarii cerintelor.
- Documentatia poate fi minima daca produsul va fi utilizat pentru o perioada de timp scurta sau este dedicat unui numar mic de utilizatori.
- Se aloca mai mult timp cerintelor complicate.
- Nu trebuie supra-documentate functii usor de înteles de multe persoane.
- Documentul cerintelor trebuie actualizat la orice modificare sau adaugare de cerinte:

  → managementul schimbarilor, asistat de instrumente speciale.

Calitatile si avantajele unei bune specificatii a cerintelor software(1)

Standardul IEEE 830 (IEEE recommended practice for software requirements specifications) descrie continutul, calitatile si avantajele unei bune specificatii a cerintelor:

Calitatile - **specificatiile cerintelor trebuie sa fie:**

- **Corecte -** sa reprezinte cu acuratete sistemul de care clientul are nevoie si pe care dezvoltatorul intentioneaza sa-l construiasca.
- **Neambigue** – fiecare cerinta definita trebuie sa aibă o singura interpretare
- **Complete** – toate aspectele sistemului, functionale si nefunctionale, inclusiv situatiile exceptionale, trebuie sa fie descrise in documentul cerintelor.
- **Consistente** – sa nu existe contradictii între diferite cerinte specificate.
- **Clasificate** dupa importanţă şi stabilitate.

Calitatile si avantajele unei bune specificatii a cerintelor software(2)

- **Verificabile**. Trebuie evitate cerinte ca : ”sistemul va furniza un raspuns rapid”, “sistemul nu va cadea niciodata”, etc.
  - O cerinta este verificabila daca se poate demonstra prin teste ale sistemului ca este implementata.
- **Realizabile** – sa poata fi implementate.
- **Trasabile**: usor de urmarit in diferite artefacte ale procesului de dezvoltare.
  - Este o calitate critica pentru testarea sistemului si pentru evaluarea efectului schimbarilor in cerinte.

Calitatile si avantajele unei bune specificatii a cerintelor software(3)

Avantajele

- Documentul cerintelor software contine o descriere completa a produsului

  → sta la baza contractului dintre client si dezvoltator.

- Reduce efortul de dezvoltare: revizia documentului cerintelor poate evidentia omisiuni, inconsistente şi neînţelegeri care pot fi corectate mai usor în aceasta etapa decat mai tarziu.
- Sta la baza estimarii costurilor si a planificarii procesului de dezvoltare.
- Permite planificarea testelor de sistem.
- Uşureaza transferul produsului la noi utilizatori sau pe platforme noi.
- Serveste ca baza pentru viitoarele îmbunatatiri sau modificari ale produsului.