



Ministerul Educației și Cercetării al Republicii Moldova  
IP Colegiul “Iulia Hasdeu” din Cahul

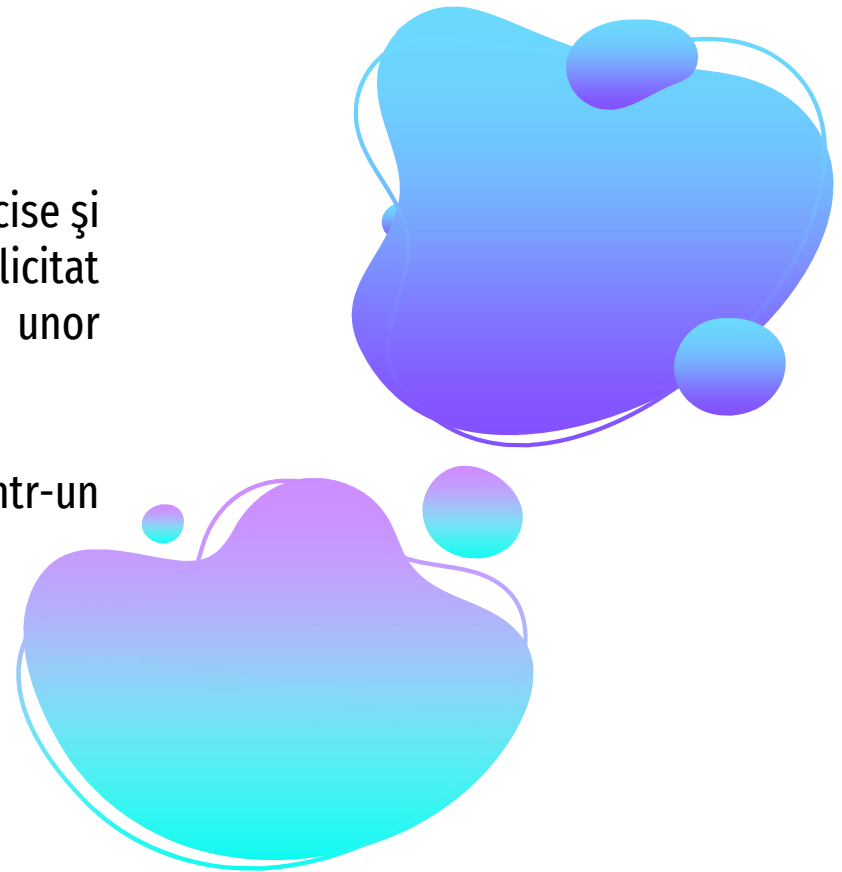
Laborator nr. 2 – Planificarea aplicațiilor Web

# **Specificarea cerințelor unei aplicații web**



## Scopul lucrării:

- Formarea deprinderilor de a formula întrebări concise și logice față de client în cea ce privește produsul solicitat și de a putea descrie aceste întrebări sub forma unor cerințe față de produs.
- Formarea abilităților de a aduna cerințele într-un document al cerințelor utilizatorului.



## Ce este o cerință ?

---

Poate varia de la o descriere abstractă de nivel înalt a unui serviciu sau a unei constrângeri a sistemului până la o specificație funcțională precizată în detaliu în termeni matematici.

## Cerința – noțiuni teoretice ?

---

Noțiunea de cerință este mai veche decât dezvoltarea programelor. De exemplu, o cerință pentru o locomotivă ar putea arăta astfel: pe o cale ferată uscată, locomotiva trebuie să fie capabilă să pornească un tren de 100 tone pe o pantă de maxim 5% cu o accelerație de cel puțin  $0,5 \text{ m/s}^2$ . De remarcat că această cerință spune ce vrea clientul. Nu spune nimic despre cum ar putea fi realizată. Nu se specifică tipul motorului (cu aburi, diesel, electric) sau materialul din care să se confecționeze roțile.



# Cerințe funcționale și non-funcționale

## **Cerințe funcționale (ce?)**

Precizarea serviciilor pe care trebuie să le ofere sistemul, cum trebuie să reacționeze sistemul la intrări particulare și cum trebuie să se comporte sistemul în situații particulare.

## **Cerințe non-funcționale (cum?)**

Constrângeri asupra serviciilor sau funcțiilor oferite de sistem, cum ar fi constrângeri de timp, constrângeri asupra procesului de dezvoltare, standarde, etc.

## **Cerințe de domeniu**

Cerințe care vin din partea domeniului de aplicație a sistemului și care reflectă caracteristicile aceluia domeniu.

## FUNCTIONAL REQUIREMENTS

**WHAT** THE SYSTEM SHOULD DO



PRODUCT FEATURES



USER REQUIREMENTS

## NON-FUNCTIONAL REQUIREMENTS

**HOW** THE SYSTEM SHOULD DO IT



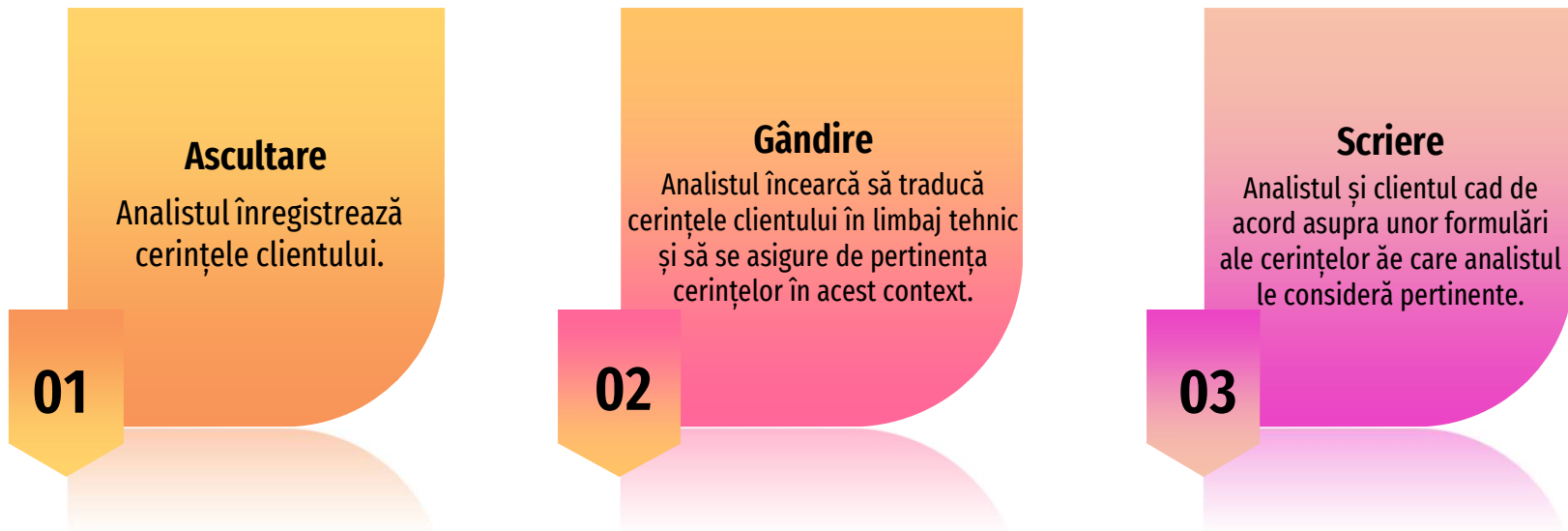
PRODUCT PROPERTIES



USER EXPECTATIONS

# Extragerea cerințelor

Poate varia de la o descriere abstractă de nivel înalt a unui serviciu sau a unei constrângeri a sistemului până la o specificație funcțională precizată în detaliu în termeni matematici.



## Extragerea cerințelor

În concluzie, analistul trebuie:

- să extragă și să clarifice cerințele clientului;
- să ajute la rezolvarea diferențelor de opinie între clienți și utilizatori;
- să sfătuiască clientul despre ce este tehnic posibil sau imposibil;
- să documenteze cerințele;
- să negocieze și să obțină o înțelegere cu clientul.





## Metode pentru identificarea cerințelor utilizatorilor

---

**Interviuri:** Interviurile trebuie astfel structurate încât să poată aborda toate aspectele implicate de sistemul software ce va trebui dezvoltat. Când există foarte mulți utilizatori, se va selecta un set reprezentativ dintre aceștia pentru a fi intervievați.

**Studiul sistemelor software existente deja:** De multe ori, noul sistem software este destinat înlocuirii altui sistem existent.

**Studiul de fezabilitate:** Studiul de fezabilitate reprezintă analiza și proiectarea principalelor caracteristici ale sistemului în scopul de a determina dacă implementarea este posibilă;

**Prototipul:** Un prototip este un model executabil al unor aspecte selectate din sistemul propus. Dacă cerințele sunt neclare sau incomplete, ar putea fi utilă dezvoltarea unui prototip, bazat pe un set de cerințe de probă, pentru a identifica cerințele reale ale utilizatorilor.

# Notiuni teoretice

Standardele ingineriei programării recomandă următoarele caracteristici ale stilului de prezentare a unui DCU:

1. DCU trebuie scris folosind un limbaj, vocabular și stil ușor de înțeles de către toți utilizatorii. DCU trebuie să fie clar, consistent, modificabil;
2. Un DCU este clar dacă orice cerință este neambiguă (are o singură interpretare) și este înțeleasă de toți participanții la proiect;
3. O cerință trebuie scrisă într-o singură propoziție iar justificările trebuie separate de aceasta.
4. Un DCU este consistent dacă nu există conflicte între cerințe. Folosirea mai multor termeni pentru a specifica același lucru este un exemplu de inconsistență;
5. Un DCU este modificabil dacă orice modificare a cerințelor poate fi documentată ușor, complet și consistent.



# Sarcina lucrării

Formularea întrebărilor logice față de client.

Elaborarea unui set de cerințe față de un produs soft.

Specificarea cerințelor clientului.

Documentarea cerințelor utilizatorului.

Scrierea unui raport despre îndeplinirea lucrării cu concluziile respective.

