

# Ghid de achiziții software pentru instituțiile publice

Versiune 1.0



asociația patronală a industriei de  
software și servicii





# Ghid de achiziții software pentru instituțiile publice

Versiune 1.0

Versiunea actualizată a ghidului precum și alte instrumente și materiale relevante pot fi descărcate de la adresa:  
[www.anis.ro/programe/ghidachizitii](http://www.anis.ro/programe/ghidachizitii)



# CONȚINUT

<b>1. Introducere</b>	7
<b>2. Principii</b>	9
Independență	9
Autonomie și continuitate	9
Definirea problemelor și obiectivelor	9
Definirea cerințelor	10
Formulări clare și naturale	10
Cerințe juste – principiul necesității	10
Inclusiv vs. exclusiv	10
Evaluare obiectivă și transparentă	10
Implicarea pe parcursul proiectului	11
<b>3. Caietul de sarcini</b>	12
Descrierea instituției și a activității	12
Problemele și nevoile curente	12
Scopul și obiectivele proiectului	13
Rolurile utilizatorilor	13
Cerințe funcționale	14
Cerințe nefuncționale	17
Cerințe de comunicare și organizare	27
Livrabilele proiectului	30
Procesul de acceptanță	35
Cerințe post implementare	38
Structura financiară	40
<b>4. Criterii de atribuire</b>	42
Cadrul legislativ	42
Metodologie de evaluare	43
Stabilirea criteriilor de evaluare	47
Criteriile de preț și cost	47
Punctarea ofertelor în funcție de cost sau preț	49
Criterii tehnice	50
<b>5. Concluzii</b>	55



# 1. INTRODUCERE

Scopul acestui ghid este să îndrume personalul implicat în procesul de achiziție a soluțiilor software, în redactarea documentației tehnice de atribuire. Ghidul este adresat în principal instituțiilor publice dar poate fi foarte bine utilizat și în cazul achizițiilor din domeniul privat. Recomandările din ghid sunt aplicabile atât achizițiilor de soluții software personalizate ("soluții custom" sau "soluții la comandă") cât și achizițiilor de produse software deja existente.

Sarcina de a crea un caiet de sarcini pentru o achiziție software poate părea la început extrem de complexă și copleșitoare, mai ales dacă persoana însărcinată nu are o pregătire tehnică de specialitate sau se află la primul astfel de demers.

Există de asemenea ideea preconcepută că un caiet de sarcini trebuie redactat într-un limbaj savant sau "de specialitate". Prin acest ghid dorim să demontăm această idee și să arătăm că orice specialist într-un domeniu public, care cunoaște problemele cu care se confruntă organizația sa, poate redacta un caiet de sarcini coerent, chiar și fără a avea o pregătire tehnică în domeniul software. Tocmai din acest motiv, în redactarea ghidului am încercat să evităm un limbaj foarte tehnic și, de asemenea, să explicăm într-un mod cât mai simplu noțiunile tehnice relevante. Din același motiv am evitat traducerea anumitor termeni tehnici în limba română acolo unde am considerat că termenul în engleză este mai relevant. Credem că un caiet de sarcini trebuie în principal să detalieze nevoile beneficiarului și problemele care trebuie rezolvate cu cât mai multe detalii și într-un limbaj cât mai simplu și clar. În aceste sens, ghidul propune o structură a caietului de sarcini și o serie de recomandări privind realizarea acestuia.

Exemplele oferite în ghid au strict rolul de a explica mai bine anumite concepte și nu trebuie preluate ad-litteram în documentația de atribuire.

## Soluție software

Termenul de "soluție software" folosit în acest ghid cuprinde toate componentele și serviciile necesare implementării unei soluții ce are la bază software. Aceste servicii și componente pot cuprinde: licențe pentru produse software de bază (sisteme de operare, sisteme de baze de date), licențe pentru produse software specializate (ERP, CRM, BPM, KB, HelpDesk), software existent disponibil comercial pe scară largă tip COTS (Commercial of the Shelf) echipamente hardware, servicii de analiză și arhitectură, servicii de consultanță în domeniul software, servicii de dezvoltare și testare, servicii de design grafic și multe altele, lista nefiind exhaustivă.

## Ce nu își propune acest ghid

Ghidul se referă strict la achiziții software și nu își propune să acopere partea de achiziții hardware, networking, servicii de infrastructură sau comunicații.

Ghidul adresează în principal latura tehnică a achizițiilor de software și nu își propune să adreseze aspectele legale ale achizițiilor precum condiții de eligibilitate, procedura achizițiilor sau contractarea. Aceste aspecte vor fi eventual elaborate într-o versiune viitoare a ghidului.

De asemenea, ghidul nu își propune să facă recomandări de tehnologie, arhitectură sau privind procesul care ar trebui urmărit pe perioada unei implementări.

Nu în ultimul rând, trebuie reiterat faptul că acest ghid nu este exhaustiv. Ghidul propune o structură a documentației tehnice de atribuire și un set de recomandări de la care se poate porni în elaborarea documentației. Documentația de atribuire trebuie însă adaptată de la caz la caz prin eliminarea unor cerințe care poate nu sunt aplicabile sau prin adăugarea de cerințe noi, specifice proiectului.

## Despre ghid

Ghidul a fost realizat la inițiativa Guvernului României de către Asociația Patronală a Industriei de Software și Servicii (ANIS). Grupul de lucru care a participat la elaborarea ghidului a fost format din: Mihai Matei (Essensys Software), Alex Lăpușan (Zitec), Dan Gârlașu (Oracle) cu

suportul și coordonarea doamnei Valerica Dragomir (ANIS) și Alex Timis (ANIS).

Prezentul document reprezintă prima versiune a acestui ghid. Ca în orice demers inițial există, cu siguranță, aspecte care pot fi îmbunătățite, precum și posibile erori sau inadvertențe care pot fi îndreptate. Există deschiderea și disponibilitatea autorilor de a îmbunătăți conținutul în versiunile viitoare pe baza sugestiilor primite de la utilizatorii ghidului.

Încurajăm utilizatorii acestui ghid să trimită comentariile și sugestiile de îmbunătățire folosind adresele de contact de mai jos. Este de asemenea foarte util să înțelegem și alte probleme de care vă loviți în achiziția sau derularea proiectelor software în organizațiile dumneavoastră. Identificarea acestor probleme poate genera noi inițiative pentru crearea unor instrumente de suport în activitatea dumneavoastră.

Adresa de contact pentru sugestii și observații: **ghidachizitii@anis.ro**



## 2. PRINCIPII

În cele ce urmează enumerăm câteva principii pe care le considerăm vitale în derularea achizițiilor publice, pe lângă cele specificate de legislația în vigoare: nediscriminarea furnizorilor, tratamentul egal al furnizorilor, respectarea criteriului proporționalității etc.

Suntem convinși că urmărirea acestor principii de **transparență și independență**, va conduce la achiziții eficiente și proiecte de succes.

### 2.1. Independență

Principiul de bază al acestui ghid este independența față de influențele potențialilor furnizori.

În realizarea unui caiet de sarcini trebuie să vă asigurați independența față de orice influență externă și să puneți pe primul loc interesul instituției pentru care lucrați. În mod evident, orice ingerință externă poate deturna sau dirija procesul de achiziție în detrimentul intereselor instituției dumneavoastră.

Scopul acestui ghid este tocmai acela de a vă oferi suportul practic necesar pentru a elabora caiete de sarcini, eliminând astfel orice dependență externă.

### 2.2. Autonomie și continuitate

Independența trebuie asigurată atât în faza de pregătire a caietului de sarcini și a procedurii de atribuire cât și ulterior încheierii relațiilor contractuale cu furnizorul selectat. Este critic, ca încă din faza de pregătire a documentației de atribuire, să asigurați autonomia instituției dumneavoastră, precum și continuitatea soluției independent de furnizorul inițial, ulterior încheierii relațiilor contractuale cu acesta.

În acest sens, documentația de atribuire trebuie să conțină prevederi care să asigure continuitatea soluției și să permită instituției dumneavoastră operarea de modificări, respectiv:

- Transferul codului sursă (în cazul soluțiilor personalizate) cu posibilitatea modificării independent de furnizor;
- Transferul tuturor livrabilelor realizate în cadrul proiectului în formă prelucrabilă/editabilă;
- Proprietatea datelor și/sau posibilitatea extragerii acestora în mod automat, fără a fi necesară intervenția furnizorului (export date)

Opțiunile pe care le aveți la dispoziție în acest sens sunt descrise pe larg în capitolul 3.6 Cerințe nefuncționale – Legal și licențiere.

### 2.3. Definirea problemelor și obiectivelor

Un factor foarte important în succesul oricărui proiect este comunicarea clară a problemelor cu care vă confrunțați în prezent și a obiectivelor și rezultatelor pe care proiectul trebuie să le rezolve. Din acest motiv, în cadrul caietului de sarcini trebuie să comunicați cât mai clar atât problemele curente, cât și obiectivele proiectului.

Identificarea clară a problemelor cu care vă confrunțați, precum și o listă clară de obiective, vă va ghida în procesul de documentare a cerințelor proiectului.

Fără aceste informații, un furnizor nu va putea înțelege problemele pe care trebuie să le rezolve și, în consecință, soluția pe care o va propune nu va fi una optimă.

## 2.4. Definirea cerințelor

Soluția software pe care o solicitați va trebui să respecte o serie de cerințe. Numărul și complexitatea acestor cerințe influențează direct costul și durata de implementare a proiectului.

Din acest motiv, este foarte important ca în caietul de sarcini să identificați clar toate cerințele pe care soluția trebuie să le respecte. Omiterea unor cerințe va cauza subestimarea proiectului de către furnizori și va cauza probleme ulterior.

Este de asemenea foarte important ca cerințele să fie exprimate cât mai clar și în detaliu, astfel încât potențialii furnizori să poată estima cât mai corect complexitatea acestora. Toate detaliile pe care le puteți furniza vor ajuta furnizorii să propună soluții optime pentru problemele dumneavoastră.

## 2.5. Formulări clare și naturale

Ideea că limbajul folosit într-un caiet de sarcini trebuie să fie unul savant sau "de specialitate" este o idee preconcepțată care nu face decât să creeze o barieră artificială de comunicare între dumneavoastră și potențialii furnizori.

Recomandarea noastră este ca în elaborarea caietului de sarcini să folosiți un limbaj cât mai natural și ușor de înțeles. De asemenea, recomandăm să definiți termenii de specialitate specifici domeniului dumneavoastră acolo unde este cazul. Formularea criptică a cerințelor va cauza ineficiențe, neînțelegeri și întrebări de clarificare necesare.

## 2.6. Cerințe juste – principiul necesității

Pe lângă enumerarea tuturor cerințelor și detalierea acestora, este la fel de important ca cerințele pe care soluția trebuie să le respecte să fie juste, respectiv să servească strict scopului proiectului și rezolvării problemelor cu care vă confrunțați fără alte adăugări necesare.

Din acest punct de vedere este recomandat să evitați în special cerințele artificiale sau ascunse care ar putea crea constrângeri și limitări artificiale sau chiar ar putea elimina din competiție unii furnizori.

Evitați de asemenea cerințele suplimentare sau conexe care nu sunt neapărat necesare proiectului (fenomenul de "placare cu aur" / "gold plating"). Acestea vor crea costuri suplimentare și vor îngreuna implementarea proiectului. În acest sens, este recomandat să faceți distincția între cerințe necesare (must have) și cerințe opționale (nice to have).

## 2.7. Inclusiv vs. exclusiv

În elaborarea caietului de sarcini, recomandăm o abordare cât mai inclusivă în privința furnizorilor. Cu siguranță există criterii juste de eligibilitate și tehnice care pot fi folosite pentru a filtra furnizorii, însă recomandăm ca aceste criterii să fie folosite cu măsură pentru a nu restricționa artificial participarea furnizorilor la competiție. Un număr mai mare de furnizori vă va oferi o plajă mai largă de opțiuni, precum și o șansă mai mare de a contracta soluția optimă.

În acest sens, este important să evitați limitarea artificială a participării prin introducerea de criterii tehnice sau de eligibilitate care pot fi satisfăcute de un singur furnizor. Orice criteriu limitativ trebuie să aibă o justificare în obiectivele proiectului.

## 2.8. Evaluare obiectivă și transparentă

În mod evident, evaluarea obiectivă și transparentă a ofertelor este vitală pentru un proces de achiziții eficient. În acest sens, recomandăm ca metodologia de evaluare să fie clar definită

în caietul de sarcini, asigurând astfel transparența procesului de evaluare. De asemenea, este recomandat să detaliați criteriile și subcriteriile de evaluare ce vor fi folosite pentru desemnarea furnizorului câștigător.

## 2.9. Implicarea pe parcursul proiectului

Credem că implicarea beneficiarului pe parcursul unui proiect, și nu doar în faza incipientă și în faza finală de acceptanță, este vitală pentru succesul proiectului.

În acest sens, este importantă alocarea resurselor umane necesare pe parcursul proiectului. Din acest punct de vedere, recomandăm alocarea unui responsabil unic de proiect care să aibă autoritatea necesară pentru a lua decizii în cadrul proiectului.

Recomandăm de asemenea un proces recurent de revizuire a progresului (spre exemplu, cu recurență lunară). În cadrul acestui proces se poate urmări progresul proiectului, se pot evalua rezultatele parțiale și, cel mai important, se poate acționa din timp în cazul în care proiectul deviază de la traseu. În cazul în care beneficiarul este implicat doar în faza de acceptanță finală, există riscul ca eventualele probleme apărute pe parcurs să fie descoperite doar la finalul proiectului cu consecințe mult mai mari.

## 3. CAIETUL DE SARCINI

În secțiunile următoare descriem capitolele pe care le recomandăm în cadrul unui caiet de sarcini. Pentru fiecare capitol există recomandări punctuale pentru elaborarea conținutului. Această listă de capitole reprezintă o recomandare minimă și nu este exhaustivă. Evident, lista poate fi completată cu alte capitole în funcție de necesitățile dumneavoastră.

### 3.1 Descrierea instituției și a activității

#### *“Care este activitatea instituției?”*

Este foarte probabil ca furnizorii de software interesați de proiectul dumneavoastră să nu cunoască foarte multe detalii despre instituția dumneavoastră și activitățile acesteia. Există deci o barieră artificială de comunicare inițială care trebuie însă depășită cât mai repede și eficient, iar un prim pas în eliminarea acestei bariere este chiar caietul de sarcini.

O primă întrebare pe care orice furnizor o are este: “Care este activitatea beneficiarului?”. Astfel, o introducere practică și utilă în caietul de sarcini este descrierea succintă a instituției dumneavoastră, a scopului acesteia și a activităților principale pe care le desfășurați. În această secțiune este indicat să detaliați activitățile relevante în cadrul proiectului care vor fi afectate de noua soluție software.

În cazul în care există o pagină de prezentare a activității instituției sau documentație relevantă în acest sens, este util ca aceasta să fie menționată sau chiar anexată caietului de sarcini.

Descrierea instituției și a activității este de asemenea un loc bun pentru introducerea termenilor specifici instituției dumneavoastră care ar putea să nu fie foarte familiari. În cazul în care terminologia folosită este foarte complexă, recomandăm crearea unui capitol separat de terminologie însă termenii importanți pot fi explicați încă din introducere.

### 3.2 Problemele și nevoile curente

#### *“Care sunt problemele de care ne lovim în prezent?”*

Un capitol esențial în fișa tehnică este descrierea problemelor cu care se confruntă instituția dumneavoastră și care au condus la necesitatea implementării unei soluții software. Identificarea și descrierea acestor probleme și nevoi este foarte importantă pentru definirea cerințelor care vor rezolva aceste probleme. De asemenea potențialii furnizori trebuie să înțeleagă problemele cu care vă confrunțați pentru a putea propune o soluție optimă.

Pentru identificarea acestor probleme, o abordare foarte practică este implicarea colegilor dumneavoastră în identificarea și detalierea acestora. Este foarte posibil ca majoritatea problemelor pe care colegii dumneavoastră le sesizează să fie formulate în limbajul tehnic specific al instituției în care lucrați. Este important să filtrați, să puneți în contextul descrierii activității (vezi capitolul anterior) și să “traduceți” aceste probleme pe înțelesul oricărei persoane care nu cunoaște detaliile activității instituției. Alte tehnici de identificare a problemelor sunt descrise în capitolul “Cerințe funcționale”.

Reluăm aici ideea importanței limbajului natural și clar, precum și explicitarea termenilor tehnici. În cazul în care în activitatea dumneavoastră se folosesc foarte mulți termeni tehnici, este utilă definirea unui capitol de Terminologie sau Glosar în cadrul caietului de sarcini.

**IMPORTANT:** Identificarea problemelor cu care se confruntă instituția dumneavoastră vă este la îndemână. Aveți toate cunoștințele și instrumentele de care aveți nevoie, dumneavoastră și colegii dumneavoastră puteți identifica aceste probleme mai bine decât orice furnizor.

**IMPORTANT:** Identificarea și detalierea problemelor cu care vă confrunțați este un pas esențial

care nu trebuie omis din caietul de sarcini. De cele mai multe ori identificarea corectă a unei probleme conduce la identificarea rapidă a unei soluții.

### 3.3 Scopul și obiectivele proiectului

#### “Unde vrem să ajungem?”

Pentru a determina obiectivele proiectului trebuie să vă imaginați situația în care vă doriți să ajungeți după implementarea proiectului. Întrebarea la care trebuie să răspundeți este: “Care este situația în care vrem să ajungem după ce acest proiect este implementat?”. Răspunsurile la această întrebare reprezintă obiectivele proiectului.

Definirea unor obiective clare va ajuta foarte mult și va ghida de asemenea și în realizarea cerințelor proiectului. Evident, un proiect poate avea mai multe obiective, care pentru a fi relevante trebuie să respecte principiile SMART, respectiv:

#### **S – Specific**

Obiectivele trebuie să fie concrete/specifice.

#### **M – Măsurabil**

Măsurarea realizării obiectivului trebuie să fie simplă și evidentă în urma unei observării obiective (vs subiectivă și interpretabilă)

#### **A – Realizabil / Achievable**

Obiectivele trebuie să fie realizabile în mod realist și nu imposibil de atins.

#### **R – Relevant**

Obiectivul trebuie să fie relevant pentru beneficiar.

#### **T- Temporal**

Trebuie să fie indicat un termen clar pentru realizarea obiectivului. Spre exemplu: “la finalul proiectului”, “într-un an de la implementare” etc.

**Exemplu bun:** “Ne propunem ca prin acest proiect, în 2 ani de la implementare, 60% din populația județului Prahova să își consulte online situația taxelor și impozitelor, reducând astfel prezența la ghișee cu 40%”.

**Contra-exemplu:** “Ne propunem ca prin acest proiect să îmbunătățim percepția și experiența cetățeanului în relația cu instituțiile statului”.

### 3.4 Rolurile utilizatorilor

#### “Cine va utiliza sistemul?”

Pentru o bună înțelegere a cerințelor, este importantă înțelegerea utilizatorilor care vor interacționa cu sistemul. Astfel caietul de sarcini trebuie să cuprindă o secțiune în care sunt enumerate rolurile care vor accesa sistemul. De asemenea, pentru fiecare rol, trebuie descrise responsabilitățile în cadrul sistemului și în cadrul instituției.

Pentru identificarea acestor roluri este recomandat să examinați procesele de lucru din cadrul instituției care sunt relevante pentru sistem.

Exemple tipice de roluri sunt: “Secretariat”, “Funcționar cu publicul”, “Operator financiar”, “Șef departament X”, “Management”, “Administrator de sistem” etc.

Pe lângă rolurile de utilizatori care vor utiliza sistemul, este util să enumerați și să descrieți și alte roluri (actori) care nu vor utiliza direct sistemul dar sunt relevante pentru înțelegerea

soluției. Un exemplu de astfel de rol ar putea fi “Autoritate de reglementare X” sau “Auditor extern”, respectiv roluri care poate beneficiază de rapoarte generate de sistem chiar dacă nu accesează direct sistemul. Aceste roluri trebuie tratate distinct și trebuie marcate ca nefiind utilizatori direcți ai sistemului.

## 3.5 Cerințe funcționale

### “Ce trebuie să facă sistemul?”

Cerințele funcționale reprezintă facilitățile sau funcțiile pe care sistemul trebuie să le realizeze. Cu alte cuvinte, ceea ce trebuie să facă sistemul.

Formularea corectă și completă a cerințelor este vitală deoarece un procent important de defecte și/sau de erori apărute într-un proiect software se datorează erorilor din analiza inițială a nevoilor beneficiarului. Cu cât o eroare se propagă mai adânc fără a fi remediată, cu atât remediarea ei va costa mai scump, astfel:

- \$1 în timpul definirii cerințelor;
- \$5 în faza de concepție a aplicației;
- \$10 în etapa de codare a aplicației;
- \$20 în timpul testelor modulelor software;
- \$200 după livrarea sistemului.

Desigur, valorile de mai sus sunt orientative însă ordinul de mărime este relevant, astfel erorile neremediate apărute în definirea problemelor și cerințelor inițiale au cel mai mare impact. Examinarea problemelor identificate este un bun punct de plecare pentru determinarea cerințelor funcționale. De cele mai multe ori, identificarea problemei conduce la identificarea rapidă a unei soluții optime de rezolvare, soluție care devine evidentă atunci când problema este cunoscută. Astfel, cerințele funcționale constituie răspunsul la problemele identificate anterior.

O altă modalitate recomandată de identificare a cerințelor este consultarea colegilor dumneavoastră în privința funcționalităților de care ar avea nevoie pentru a-și desfășura activitatea mai eficient. Ca și în cazul problemelor, aceste cerințe trebuie filtrate și structurate.

Câteva tehnici pentru identificarea cerințelor sunt:

- Examinarea problemelor identificate;
- Grupuri de lucru (brainstorming);
- Interviuri cu colegii dumneavoastră;
- Examinarea elementelor și documentelor existente (fluxuri de lucru, formulare, rapoarte, legislație etc);
- Examinarea sistemelor informatice existente sau similare;
- Observarea activității colegilor (shadowing);
- Utilizarea de chestionare online;
- Revizuirea cerințelor existente pentru a determina cerințe indirecte.

În cele ce urmează enumerăm câteva tipologii de cerințe funcționale ce pot fi folosite ca punct de plecare în enumerarea cerințelor dumneavoastră. Având în vedere diversitatea cerințelor funcționale, nu avem pretenția că această listă ar fi completă.

### Clasificarea cerințelor în funcție de importanță

Atât cerințele funcționale cât și nefuncționale, pot fi clasificate în funcție de importanță folosind metodologia MoSCoW:

- Cerințe critice (MUST have) – sunt cerințe fără de care sistemul nu își îndeplinește scopul;
- Cerințe importante (SHOULD have) – cerințe importante dar care nu sunt strict necesare;

- Cerințe posibile (COULD have) – cerințe posibile sau opționale (Nice to have) care nu sunt neapărat necesare și sunt mai puțin importante;
- Cerințe eliminate (WON'T have) – cerințe fără relevanță care vor fi eliminate din scopul proiectului.

Clasificarea cerințelor vă va ajuta în stabilirea cerințelor care sunt incluse în caietul de sarcini. În mod evident toate cerințele critice (must) trebuie incluse în documentația de atribuire. De asemenea, în funcție de bugetul și timpul disponibil, puteți include și o parte din cerințele importante (should) sau posibile (could).

Cerințele importante (should) și posibile (could) ar putea fi introduse în documentație ca cerințe opționale suplimentare. În cazul în care criteriile de evaluare implică și rezolvarea acestor cerințe pentru punctaj suplimentar, este foarte important să clarificați acest lucru în caietul de sarcini.

Includerea cerințelor eliminate (WON'T have) în documentația de atribuire poate fi o modalitate foarte practică de a elimina anumite presupuneri ale furnizorilor, presupuneri care ar putea genera costuri inutile. În cazul în care o parte din aceste cerințe ar putea deveni cerințe într-o versiune ulterioară, este utilă menționarea lor, astfel încât furnizorii, pe cât posibil, să proiecteze o arhitectură de soluție care să acomodeze aceste cerințe mai ușor.

### Evidența generală de informații

O bună parte din cerințele funcționale pot fi încadrate în categoria evidenței generale de informații, respectiv capacitatea sistemului de a stoca anumite tipuri de informații. În astfel de cazuri, cerința de bază este că utilizatorii trebuie să poată adăuga, modifica și șterge diferite tipuri de înregistrări.

**IMPORTANT:** În astfel de cazuri este importantă detalierea atributelor înregistrării (câmpurile). De asemenea, este utilă identificarea relațiilor cu alte entități (spre exemplu, un contribuabil este asociat unei unități teritoriale). Enumerarea acestor atribute și relații poate conduce la identificarea unor cerințe ascunse. Spre exemplu, identificarea atributului "Județ" necesită, cel mai probabil, definirea unui nomenclator de județe.

Alte funcționalități comune ar putea fi:

- Căutare după diverse criterii cu specificarea acestora;
- Ordonarea listelor după unul sau mai multe câmpuri;
- Validări speciale (ex: interogarea altor sisteme);
- Exportul mai multor înregistrări;
- Printarea unei înregistrări;
- Posibilitatea de a importa una sau mai multe înregistrări;
- Gruparea informațiilor după diverse criterii;
- Versionarea modificărilor (păstrarea versiunilor anterioare);
- Auditarea modificărilor;
- Arhivarea înregistrărilor;
- Accesul securizat la înregistrări (permisiuni la nivel de înregistrare).

Un exemplu de astfel de cerință ar putea fi: "Sistemul trebuie să permită evidența contribuabililor. Utilizatorii trebuie să poată adăuga, modifica și șterge înregistrările contribuabililor. Ștergerea unei înregistrări presupune mutarea în arhivă de unde poate fi restaurată samd". Iată că în acest caz, simpla formulare a unei cerințe a condus la o cerință ascunsă, respectiv aceea de arhivare.

### Nomenclatoare

O categorie specială de înregistrări sunt înregistrările de tip nomenclator. Nomenclatoarele sunt simple liste de înregistrări, cu o dinamică mult mai mică a modificărilor. Exemple de nomenclatoare: județe, țări, orașe, zile libere legal, unități teritoriale etc.

Identificarea nomenclatoarelor în cadrul caietului de sarcini este foarte utilă chiar dacă pe parcursul proiectului este probabilă și descoperirea altor nomenclatoare neidentificate inițial.

## Fluxuri de lucru

Documentarea fluxurilor de lucru sub formă de diagrame de flux este o modalitate foarte utilă de a descrie cerințele funcționale. În cazul în care alegeți să includeți astfel de diagrame în caietul de sarcini, este important să marcați corespunzător pe diagramă pașii din flux care trebuie realizați în cadrul sistemului și pașii care se realizează în afara sistemului.

## Integrări cu alte sisteme

O altă categorie de cerințe este reprezentată de integrarea cu alte sisteme software, integrare care poate lua foarte multe forme.

Integrarea la nivel de date poate presupune un transfer de date cu un alt sistem și poate fi unidirecțională sau bidirecțională. Integrarea unidirecțională presupune transferul de date dinspre sistemul A către sistemul B sau viceversa, în timp ce integrarea bidirecțională presupune transferul de date în ambele direcții.

Integrare de API (interfață programatică, Application Programming Interface) presupune utilizarea interfețelor programatice oferite de un alt sistem. Un exemplu ar fi utilizarea unui API de transmitere de SMS-uri sau integrarea cu serviciile online oferite de Registrul Comerțului sau Ministerul Justiției.

Există, de asemenea, posibilitatea integrării la nivel de componente sau chiar la nivel de interfață utilizator. În astfel de cazuri un sistem A folosește componente sau chiar fragmente de interfață utilizator puse la dispoziție de un sistem B. Un exemplu de integrare la nivel de interfață ar fi integrarea unui formular de plată online oferit de un procesator de plăți.

## Activități automate

O parte din funcțiile unui sistem reprezintă activități automate. Exemple de astfel de activități ar fi backup-ul automat al bazei de date sau procesări de date recurente (ex: închideri de zi, lună etc.).

Recomandarea noastră este să automatizați în cel mai mare grad posibil operațiunile, reducând astfel încărcarea echipei ce va utiliza proiectul rezultat și reducând posibilele erori de operare. Anumite procese trebuie să fie automatizate în mod obligatoriu (de exemplu, efectuarea de copii de siguranță a datelor), deoarece în lipsa automatizării acestora este foarte posibil ca operațiunea să nu aibă loc sau să aibă loc cu o frecvență aleatoare, anulând scopul funcționalității.

## Notificarea utilizatorilor

În multe cazuri există cerințe funcționale privind notificarea utilizatorilor în privința evenimentelor relevante. În cazul notificărilor este utilă enumerarea tipurilor de notificare și specificarea canalelor pe care notificările vor fi transmise (Ex: mail, sms, în aplicație etc.).

Exemple de notificări ar putea fi: notificarea transmisă pe mail la alocarea unei sarcini, finalizarea generării unui raport care durează mult, rezultatul backup-ului bazei de date transmis administratorilor de sistem etc.

## Raportare

Majoritatea sistemelor informatice pun la dispoziția utilizatorilor un set de rapoarte predefinite care pot fi generate pe baza datelor stocate în sistem. Conținutul unui raport poate varia de la o listă simplă de înregistrări, până la rapoarte complexe conținând indicatori de performanță și elemente vizuale complexe (grafice, hărți etc). Rapoartele pot fi de asemenea parametrizate. Spre exemplu, un raport care generează lista contribuabililor cu datorii mai mari de X, având domiciliul în orașul Y, este un raport cu parametrii X și Y care pot fi modificați de utilizator.

În cazul în care cerințele de raportare sunt numeroase, este recomandată alocarea unui capitol dedicat în documentația de atribuire. În cadrul acestui capitol enumerați rapoartele dorite menționând denumirea fiecărui raport, conținutul pe scurt și parametrii doriți. În cazul în care cerințele de raportare nu pot fi stabilite, este necesar să menționați cel puțin numărul de rapoarte dorite și complexitatea acestora astfel încât furnizorii să poată estima în mod corect bugetul (ex: 20 de rapoarte de complexitate mică, 10 rapoarte de complexitate medie și 5 rapoarte de complexitate mare).



În cazul în care rapoarte similare sunt deja generate pe alte căi (exemplu: excel sau folosind alte sisteme) este foarte util să anexați caietului de sarcini aceste exemple.

O cerință uzuală pe care o puteți solicita este capabilitatea de printare și export a rezultatelor în diferite formate uzuale (pdf, html, word, excel etc.).

Pe lângă rapoartele predefinite, unele sisteme oferă posibilitatea utilizatorilor de a-și crea propriile rapoarte. În cazul în care solicitați această funcționalitate, nu uitați să o menționați în cadrul cerințelor de raportare. Această facilitate poate necesita cunoștințe tehnice și ar putea fi accesibilă doar personalului tehnic sau poate fi accesibilă și utilizatorilor obișnuiți.

În cazul proiectelor cu nevoi de raportare complexe și variabile în timp, realizarea acestora în mod programatic ar putea presupune costuri considerabile. În astfel de situații recomandăm analizarea posibilității de a utiliza platforme dedicate de raportare și procesare a datelor, incluzând în proiect doar activitățile necesare integrării cu o astfel de platformă și, eventual, costurile de licențiere și instruire. Într-un astfel de scenariu configurarea rapoartelor ar putea fi realizată cu resurse interne.

## 3.6 Cerințe nefuncționale

### *“Cum trebuie să se comporte sistemul?”, “Ce alte caracteristici sunt importante?”*

Spre deosebire de cerințele funcționale care exprimă ceea ce trebuie să facă sistemul, cerințele nefuncționale exprimă cerințe despre cum trebuie să funcționeze sistemul, respectiv cerințe care nu țin neapărat de funcționalitate. Dacă identificarea cerințelor funcționale este relativ naturală și intuitivă, cerințele nefuncționale sunt caracteristici ale sistemului care nu sunt la fel de evidente. Din acest motiv considerăm că este utilă o listă a posibilelor cerințe nefuncționale ce pot influența un sistem informatic.

În cele ce urmează propunem o astfel de listă cu cele mai comune cerințe nefuncționale pe care o puteți adapta nevoilor dumneavoastră. Aceste cerințe trebuie incluse și detaliate în caietul de sarcini sau eliminate dacă nu sunt aplicabile în cazul dumneavoastră. Lista propusă nu este exhaustivă, astfel puteți adăuga în caietul de sarcini și alte cerințe care sunt relevante pentru dumneavoastră.

**IMPORTANT:** Întotdeauna cerințele suplimentare presupun și costuri suplimentare. Acest lucru este evident în cazul cerințelor funcționale deoarece sistemul “va face mai multe” deci este evident ca va costa mai mult. În cazul anumitor cerințe nefuncționale, costurile pot crește exponențial chiar dacă acest lucru nu este la fel de evident. Astfel, ca și în cazul cerințelor funcționale, este recomandat să solicitați doar acele cerințe nefuncționale strict necesare, eliminând cerințele opționale (nice to have).

Spre exemplu, o interfață utilizator care este disponibilă doar în limba română este mai puțin costisitoare decât o interfață disponibilă în mai multe limbi. Pentru o interfață disponibilă în mai multe limbi sunt necesare următoarele intervenții: dezvoltarea de cod/logică pentru localizare, traducerea interfeței în celelalte limbi, testarea fiecărei interfețe, localizarea manualelor de utilizare și, eventual, chiar instruirea în fiecare limbă. Mai mult, în cazul în care și conținutul bazei de date trebuie localizat, impactul este mult mai mare și va presupune un efort considerabil de administrare inclusiv pentru beneficiar (ex: localizarea știrilor pentru un site). Astfel, dacă o interfață în limba română este suficientă, este mult mai eficient să nu solicitați o interfață disponibilă în mai multe limbi. Același lucru este valabil pentru majoritatea cerințelor nefuncționale.

### 3.6.1 Accesibilitate

Accesibilitatea se referă la capacitatea unui sistem de a fi utilizat de persoane cu dizabilități

(ex: lipsa totală sau parțială a vederii, lipsa auzului etc.). În cazul în care sistemul trebuie să fie accesibil persoanelor cu dizabilități, este important să specificați aceste cerințe în caietul de sarcini.

NOTĂ: Majoritatea sistemelor de operare oferă funcții predefinite pentru persoanele cu dizabilități (magnifier, text-to-speech etc.). În cazul în care, pe lângă aceste funcții, sunt necesare funcții speciale dedicate persoanelor cu dizabilități, acestea trebuie specificate în detaliu.

### 3.6.2 Audit

Cerințele de auditare se referă la capacitatea sistemului de a înregistra toate operațiile efectuate de către utilizatori, de către alte sisteme sau operații automate. În scopul trasabilității, registrul de audit (audit log) este disponibil ulterior pentru consultare, spre exemplu în cazul unui audit extern.

În cazul în care sistemul dumneavoastră necesită astfel de capabilități, este important să le menționați în caietul de sarcini. Este de asemenea recomandat să menționați explicit operațiile care necesită auditare și detaliile ce trebuie auditate. Evident, în cazul în care doriți auditarea tuturor operațiilor, costurile de implementare și exploatare vor crește.

### 3.6.3 Backup

Procesul de backup este procesul prin care datele unui sistem sunt salvate (backup) pentru a fi ulterior restaurate (restore) în caz de nevoie (pierderea datelor, dezastru etc.). Existența unei proceduri de backup este obligatorie pentru orice sistem informatic. Cerințele de backup nu ar trebui să lipsească din nici un caiet de sarcini.

Cerințele de backup trebuie să cuprindă frecvența cu care este necesară realizarea de backup-uri, tipul backup-ului, mediul de backup (online, offline) șamd.

Recomandăm programarea și includerea în serviciile de mentenanță a unor exerciții periodice prin care să verificați că procedura de back-up și restore funcționează corect.

### 3.6.4 Capacitate și volum de date

O secțiune care nu trebuie să lipsească din caietele de sarcini este cea legată de capacitatea și volumele de date pe care sistemul trebuie să le suporte în prezent și viitor. Aceste informații pot fi exprimate în valori curente și valori anuale aditive. Spre exemplu, numărul curent de utilizatori este 5000 și ne așteptăm la adăugarea unui număr maxim de 200 de utilizatori anual. Creșterile anuale pot fi exprimate și procentual.

Acuratețea acestor valori este importantă pentru proiectarea și dimensionarea corectă a soluției. Este astfel recomandat să evitați exagerarea volumelor deoarece va conduce la creșterea nejustificată a costurilor.

Exemple uzuale de date volumetrice:

- Număr total de utilizatori;
- Număr de utilizatori activi simultan;
- Număr de înregistrări preferabil de fiecare tip relevant (ex: nr facturi, nr de documente etc.);
- Număr de documente;
- Dimensiune stocare documente.

### 3.6.5 Conformitate și Certificări

Soluțiile software utilizate în anumite domenii de activitate ar putea necesita certificări și autorizații speciale pe care soluția trebuie să le respecte. De asemenea, pentru anumite soluții ar putea exista standarde sau legislație care trebuie respectate. În cazul în care există astfel de cerințe, acestea trebuie să fie detaliate și justificate în caietul de sarcini.

**IMPORTANT:** Includerea în caietul de sarcini a unor cerințe nejustificate de certificare și conformitate pot restrânge lista furnizorilor eligibili și pot genera costuri nejustificate. Recomandăm ca includerea unor astfel de cerințe eliminatorii să se facă cu prudență și în cazurile în care acestea sunt chiar necesare.

### 3.6.6 Compatibilitate și interoperabilitate

În cazul în care soluția furnizorului trebuie să fie compatibilă cu alte soluții sau tehnologii, aceste cerințe trebuie documentate și detaliate. Un aspect pe care este indicat să îl luați în considerare este structura parcului IT disponibil în cadrul instituției (calculatoare desktop, monitoare etc.). Această evaluare poate conduce la anumite cerințe de compatibilitate (de ex: sisteme de operare vechi, rezoluții ecran reduse).

Exemple:

- Compatibilitatea cu anumite sisteme de operare și versiuni;
- Anumite browsere internet;
- Capacități hardware (ex: rezoluții de ecran);
- Sisteme de baze de date;
- Standarde tehnologice;
- Etc.

### 3.6.7 Disponibilitate (availability)

Cerințele de disponibilitate exprimă perioada în care soluția trebuie să fie funcțională și accesibilă utilizatorilor.

Spre exemplu, o aplicație de registratură mai puțin critică ar putea fi disponibilă doar în timpul săptămânii, în intervalul orar 8.00-18.00, interval în care sunt acceptate întreruperi de maxim 1 oră. O soluție de dispecerat pentru situații de urgență trebuie, cel mai probabil, să fie disponibilă 24 de ore din 24, 7 zile din 7 posibile cu întreruperi de maxim 1 ora/luna pentru mentenanța tehnică în anumite intervale orare.

Cerințele de disponibilitate se pot exprima sub forma intervalului orar în care sistemul trebuie să fie disponibil și a duratei acceptabile de indisponibilitate în intervalul de disponibilitate. Alternativ, se poate exprima sub forma raportului între timpul de disponibilitate (uptime) împărțit la suma timpului de disponibilitate adunat cu timpul de indisponibilitate, respectiv timpul total.

Spre exemplu, o soluție disponibilă permanent pe perioada a treizeci de zile cu o întrerupere de o oră are o disponibilitate de  $30 \text{ zile} \times 24 \text{ ore} - 1 \text{ oră} / 720 = 719 / 720 = 99.86\%$ , respectiv timp funcționare / (timp funcționare + timp nefuncționare) sau mai simplu timp funcționare/timp total.

**IMPORTANT:** Trebuie subliniat faptul că un grad foarte mare de disponibilitate presupune și costuri considerabile (în general pentru infrastructura redundantă dar și pentru o arhitectură software care să suporte o înaltă disponibilitate). Exagerarea nevoilor de disponibilitate, a intervalelor orare critice precum și a disponibilității echipei de suport tehnic din partea furnizorului va duce la o creștere nejustificată a costurilor.

### 3.6.8 Documentație

Documentația este o parte importantă a oricărei soluții software. În caietul de sarcini trebuie definite documentele care trebuie livrate de către furnizor. O listă posibilă a acestor documente este definită în secțiunea "Livrabilele proiectului".

### 3.6.9 Extensibilitate

Extensibilitatea unei soluții reprezintă gradul și ușurința cu care aceasta poate fi extinsă cu noi funcționalități.

În cazul în care există extensii viitoare ale soluției, este foarte util ca acestea să fie comunicate astfel încât furnizorul să definească o arhitectură care să suporte extensia. De asemenea, alte cerințe și așteptări în privința extensibilității trebuie solicitate explicit în caietul de sarcini. Exemple: posibilitatea de a adăuga noi rapoarte, posibilitatea de a adăuga noi roluri de securitate etc.

Pentru a asigura în practică extensibilitatea, este important să vă asigurați contractual de disponibilitatea echipei furnizorului precum și să încercați să prevedeați costurile pe care astfel de extinderi le implică în viitor. Acest lucru se traduce prin solicitarea unor garanții de timp de intervenție și de păstrarea condițiilor comerciale pentru o anumită perioadă de timp (ex. Costurile de manoperă sau licențiere nu se vor modifica pentru o anumită perioadă de timp).

### 3.6.10 Garanție

Garanția unei soluții protejează beneficiarul de impactul eventualelor defecte care sunt descoperite ulterior acceptanței soluției, eventual în mediul de producție. Garanția trebuie să acopere atât cerințele funcționale, cât și cerințele nefuncționale, iar defectele identificate trebuie remediate pe cheltuiala furnizorului.

Spre exemplu, o soluție care se dovedește a funcționa neperformant în condițiile de volum de date și încărcare stabilite în caietul de sarcini sau specificațiile tehnice, este o soluție defectă care trebuie remediată de furnizor pe cheltuială proprie. În cazul în care încărcarea și volumul de date sunt mai mari decât cele specificate, discutăm despre o soluție care necesită modernizare, costul urmând să fie suportat de beneficiar.

Iată încă o dată de ce formularea clară a cerințelor este importantă. În derularea garanției este importantă definirea cât mai explicită a noțiunii de defect, respectiv un comportament al soluției diferit de cel stabilit de caietul de sarcini sau specificațiile funcționale. Caietul de sarcini și, ulterior, specificațiile tehnice sunt referință pentru determinarea defectelor.

În general, perioada de garanție începe să curgă începând cu data instalării în producție. Cu toate acestea, în cazul în care instalarea în producție a soluției este amânată din cauze care nu țin de furnizor, garanția trebuie să înceapă să curgă de la o altă dată limită (ex: cel târziu X zile de la finalizarea cu succes a acceptanței). O cerință poate fi: "Garanția furnizorului va începe să curgă de la data instalării în producție sau în cazul amânării cu mai mult de o lună a instalării din cauze care nu țin de furnizor, la un interval de o lună de la acceptarea soluției".

Majoritatea defectelor sunt descoperite în primele câteva luni de funcționare ale unei soluții. Recomandăm astfel ca perioada de garanție minim solicitată să fie de 6 luni pentru soluții de complexitate mică-medie și cel puțin 12 luni pentru soluții de complexitate mai mare. Evident, perioada solicitată poate fi mai mare și este chiar indicat dacă specificul proiectului o cere. Trebuie însă subliniat faptul că și această cerință poate presupune costuri suplimentare.

Cerințele privind garanția trebuie solicitate și detaliate în documentația de atribuire. Durata garanției, termenul de la care începe să curgă perioada de garanție, condițiile de acoperire, precum și serviciile pe care furnizorul trebuie să le ofere în perioada de garanție, sunt detalii care trebuie cât mai clar solicitate în documentația de atribuire și, ulterior, preluate în contractul

cu furnizorul.

### 3.6.11 Instalare

Modul în care o soluție este instalată în mediul de producție poate fi un aspect extrem de important în anumite medii. Spre exemplu, timpul de instalare sau upgrade ar putea fi foarte constrâns în cazul unui sistem de dispecerat pentru situații de urgență.

În cazul în care există cerințe speciale privind instalarea, acestea trebuie bine documentate în documentația de atribuire. Exemple: timpul maxim de instalare în cazul în care există constrângeri de timp, ferestrele de timp disponibile (ex: noaptea, doar în weekend etc.), tipul instalării (automat, manual), resursele umane disponibile pentru instalare și nivelul tehnic al acestora.

### 3.6.12 Interoperabilitate

Interoperabilitatea se referă la capacitatea unui sistem de a schimba informații cu alt sistem. Interoperabilitatea presupune capacitatea de a interpreta automat informații furnizate de alt sistem și, de asemenea, de a genera informații care pot fi interpretate automat de alte sisteme. În general, interoperabilitatea se referă la folosirea de standarde open universal recunoscute. Exemple de standarde open sunt: HTML, PDF, XML etc.

În cazul în care există cerințe speciale de interoperabilitate pe care sistemul trebuie să le respecte, acestea trebuie detaliate în caietul de sarcini.

### 3.6.13 Infrastructura hardware și software existentă

O secțiune foarte importantă a cerințelor nefuncționale este secțiunea în care este descrisă infrastructura hardware și software existentă cu care noul sistem trebuie să se integreze. Având în vedere importanța descrierii, este recomandat ca aceasta să fie un capitol distinct al caietului de sarcini.

Descrierea infrastructurii poate să cuprindă următoarele aspecte:

- Topologia logică a rețelei locale și inter-locație dacă există;
- Identificarea soluțiilor folosite în instituție, relevante pentru proiect (Active Directory, mail server, soluții de colaborare etc.);
- Conectivitate internet dacă este relevantă pentru proiect (caracteristici conexiuni);
- Tehnologii de dezvoltare software utilizate (limbaje de programare, baze de date etc.);
- Servere eventual disponibile pentru noul sistem (configurație hardware și software);
- Licențe software disponibile dacă sunt relevante (sistem de operare, bază de date);
- Alte sisteme software relevante în cadrul proiectului;
- Descrierea stațiilor client (hardware minim, sistem de operare, browser, software etc.);
- Număr de stații și servere;
- Alte echipamente și software relevante în cadrul proiectului;
- Orice alte informații legate de hardware sau software ce ar putea fi relevante pentru proiect.

În descrierea infrastructurii existente este foarte utilă utilizarea diagramelor.

### 3.6.14 Localizare și globalizare

Internaționalizarea presupune capacitatea aplicației de a funcționa în alte culturi (țări). Cerințele de internaționalizare trebuie documentate explicit în documentația de atribuire.

Exemple de astfel de cerințe:

- disponibilitatea interfeței în mai multe limbi;
- disponibilitatea documentației în mai multe limbi;
- suport multi valută;
- posibilitatea de a schimba formatul folosit pentru numere și date calendaristice;
- funcționarea corectă în zone diferite de timp.

### 3.6.15 Legal și licențiere

De multe ori aspectele legale sunt ignorate din caietele de sarcini cu consecințe nefaste pentru beneficiar. Omiterea cerințelor în această privință poate cauza o relație de captivitate față de furnizorul soluției ceea ce este evident de evitat.

Pentru a preveni astfel de situații este foarte important să specificați în caietul de sarcini cerințele legate de licențiere și alte aspecte legale.

#### Codul sursă și modificări ulterioare

În cazul soluțiilor personalizate (la cheie), este foarte important să clarificați aspectele legate de proprietatea intelectuală și accesul la codul sursă al soluției.

**IMPORTANT:** Pentru a evita o relație de dependență, trebuie să vă asigurați că veți putea modifica codul sursă independent de furnizorul inițial, veți putea transfera codul sursă către un alt furnizor sau contracta un alt furnizor pentru modificări sau servicii de mentenanță.

Acest lucru se poate realiza uzual prin următoarele mecanisme:

- licențiere nelimitată a tuturor livrabilelor cu transfer al codului sursă și a livrabilelor cu posibilitatea modificării ulterioare de către beneficiar fără restricții (recomandat);
- transferul proprietății intelectuale asupra tuturor livrabilelor către instituția dumneavoastră.

Din punct de vedere practic ambele opțiuni vă vor oferi posibilitatea modificării ulterioare a soluției fără a depinde de furnizorul inițial.

Opțiunea licențierii fără limitări a soluției și transferul codului sursă și a tuturor livrabilelor în formă prelucrabilă/editabilă, va permite instituției dumneavoastră modificarea sub orice formă a soluției fără a depinde de furnizor. De asemenea această opțiune nu presupune constrângeri suplimentare pentru furnizor. Acesta va putea reutiliza la rândul său rezultate din proiect în alte proiecte (ex: componente reutilizabile).

Opțiunea de transfer a proprietății intelectuale presupune constrângeri suplimentare pentru furnizor și costuri posibil mai mari. În acest caz furnizorul nu va putea reutiliza componente sau fragmente din proiecte anterioare (ex: componente proprii, framework, etc.) deoarece nu va putea realiza integral transferul de proprietate intelectuală. De asemenea furnizorul nu va putea refolosi rezultate din proiect în alte proiecte, acestea devenind integral proprietatea dumneavoastră. Această opțiune este mai puțin flexibilă dar poate fi necesară în anumite proiecte (ex: siguranță națională).

#### Licențiere

În cazul unei achiziții de produse software deja existente este foarte complicat ca un furnizor să transfere proprietatea intelectuală către beneficiar sau să fie de acord cu modificarea codului sursă. Însă este esențial să clarificați tipul de licențiere dorit și politica de upgrade și suport. În acest caz trebuie să clarificați dacă licența pe care o doriți este una perpetuă sau una temporară (tip abonament). În cazul licențierii sub formă de abonament trebuie să clarificați situația în care veți finaliza contractul cu furnizorul respectiv. În privința upgrade-urilor (versiuni noi) trebuie să definiți cerințele legate de perioada pentru care sunt solicitate upgrade-uri și dacă acestea trebuie oferite contra cost sau nu. De asemenea, perioada în care furnizorul va oferi suport pentru produsul oferit este importantă (sub forma de suport tehnic sau update-uri), precum

și condițiile în care suportul este oferit (perioadă, costuri etc). În mod evident, unele cerințe pot genera costuri suplimentare pe care trebuie să le bugetați.

### Proprietatea datelor

Indiferent de tipul soluției, soluție personalizată sau produs software, aspectul proprietății datelor stocate în sistem nu trebuie neglijat. În acest sens trebuie să vă asigurați că datele rămân proprietatea instituției dumneavoastră în orice situație. De asemenea, trebuie să vă asigurați că furnizorul oferă o modalitate tehnică prin care datele pot fi extrase în orice moment din sistemul existent într-o formă prelucrabilă (export într-un format standard, acces la baza de date etc.).

### Componente dezvoltate de terți

Un alt aspect important este legat de utilizarea de componente software dezvoltate de terți. Astfel de componente accelerează procesul de dezvoltare și sunt uzual folosite de dezvoltatori. Este important ca furnizorul să asigure licențierea corectă a acestor componente.

### Încetarea activității furnizorului și transfer

Există situații în care furnizorul își încetează activitatea sau situații în care relația contractuală cu acesta nu poate continua. Este important să stabiliți cerințele privind o astfel de situație. Contractual, furnizorul i se poate solicita să realizeze transferul tuturor livrabilelor proiectului către beneficiar sau către un alt furnizor.

### Legislație aplicabilă

În cazul în care activitatea instituției dumneavoastră este guvernată de anumite legi care au un impact asupra sistemului, este necesară menționarea legilor aplicabile în această secțiune.

**IMPORTANT:** Aceste cerințe legale trebuie reluate sub formă de clauze contractuale și în contractul dintre instituție și furnizor.

## 3.6.16 Migrare

Există frecvent situații în care o soluție software nouă este concepută să înlocuiască o soluție anterioară. Cel mai probabil, în astfel de situații, soluția existentă a fost folosită de-a lungul timpului pentru a colecta date și informații. Nu uitați să solicitați în caietul de sarcini migrarea datelor în cazul în care informațiile din sistemul anterior trebuie migrate în noul sistem, în multe situații această activitate nu este trivială.

Într-un astfel de caz este important să anexați caietului de sarcini cât mai multe informații despre sistemul anterior și, în special, documentația bazei de date dacă există sau, cel puțin, structura acesteia. Fără aceste informații furnizorii nu pot face o evaluare corectă a complexității și efortului necesar migrării.

Majoritatea provocărilor în cazul unei migrării provin din posibile inconsistențe sau erori în datele existente (ex: date lipsă sau invalide) precum și din dificultatea mapării datelor vechi pe noua structură a bazei de date. Din acest motiv, dacă situația o permite, este recomandat să transmiteți furnizorilor o mostră de baze de date actuală, eventual anonimată, astfel încât aceștia să poată înțelege complexitatea problemei pe care o au de rezolvat.

## 3.6.17 Monitorizare

Un aspect tehnic important al unei soluții software este capacitatea acesteia de a fi monitorizată din punct de vedere tehnic. O cerință minimă care nu ar trebui să lipsească este înregistrarea erorilor întâmpinate (error log). Registrul de erori poate fi ulterior folosit pentru investigarea și rezolvarea erorilor folosite.

O altă cerință ar putea fi alertarea automată a administratorilor de sistem atunci când sunt întâmpinate condiții anormale de funcționare. În cazul unor aplicații critice astfel de funcționalități ar putea fi absolut necesare, în timp ce pentru aplicații non-critice astfel de

cerințe pot fi omise.

Existența unor indicatori de performanță ai sistemului și a rapoartelor de sistem ar putea fi de asemenea cerințe dacă astfel de funcționalități sunt necesare. Pe lângă aceste sugestii puteți formula și alte cerințe concrete privind monitorizarea sistemului.

### 3.6.18 Performanță

Performanța unui sistem este foarte importantă pentru funcționarea eficientă a acestuia. Funcționarea neperformantă (lentă) a unor funcții cheie poate conduce la blocarea sistemului și la imposibilitatea utilizării acestuia. Astfel formularea cerințelor de performanță în caietul de sarcini este foarte importantă.

Trebuie subliniat faptul că performanța unui sistem este influențată de foarte mulți factori, printre care numărul de utilizatori activi, volumul de date, distribuția datelor, caracteristicile hardware, conexiunea la rețea (locală sau internet), gradul de încărcare al conexiunilor, arhitectura sistemului, structura bazei de date și multe altele. O parte dintre acești factori sunt cu siguranță cunoscuți (volum de utilizatori) sau pot fi influențați de furnizor (ex: arhitectura), însă alți factori nu pot fi controlați de furnizor sau chiar nu pot fi cunoscuți la momentul realizării caietului de sarcini.

Din acest motiv este foarte important ca în formularea cerințelor de performanță să se stabilească și condițiile în care cerințele respective trebuie îndeplinite. Cu alte cuvinte, trebuie detaliate volumele de date preconizate, număr de utilizatori activi simultan, caracteristicile hardware, detaliile de infrastructură samd.

Cerințele de performanță pot fi formulate sub formă de timp optim și timp maxim în care anumite operații trebuie să poată fi realizate. În acest sens, este util să identificați cerințele funcționale cheie ale sistemului și să asociați acestora cerințe de performanță.

Un exemplu de cerință de performanță ar putea fi: "Regăsirea unei înregistrări de contribuabil folosind CNP-ul trebuie să dureze sub X secunde în mod uzual și maxim Y secunde în condiții de încărcare maximă a sistemului".

### 3.6.19 Recuperare în caz de dezastru (Disaster recovery)

Un dezastru este un eveniment care poate afecta funcționalitatea sistemului. Un dezastru poate fi defectarea dispozitivelor de stocare (hard disk sau SSD), defectarea plăcii de bază a unui server dar și dezastre naturale cum ar fi incendii, cutremure, inundații etc.

Indiferent de infrastructura de care dispune instituția dumneavoastră, este vital să aveți la dispoziție un plan de restaurare a sistemului în caz de dezastru. Acest plan are rolul de a repune în funcțiune un sistem în cel mai scurt timp posibil în cazul apariției unui dezastru. În panica ulterioară apariției unui dezastru un astfel de plan se va dovedi extrem de valoros.

Unul din parametrii cheie ai acestui plan care trebuie specificat este timpul maxim de restaurare. Cu cât acest timp este mai mic, cu atât investiția necesară este mai mare. Pentru a asigura un timp de restaurare foarte scurt este posibil ca anumite echipamente redundante să fie necesare (ex: servere).

Este de asemenea foarte important ca acest plan să fie actualizat și testat periodic pentru a asigura validitatea și aplicabilitatea acestuia în caz de nevoie.

### 3.6.20 Scalabilitate

Scalabilitatea unui sistem reprezintă capacitatea acestuia de a gestiona volume mai mari de activitate și date prin îmbunătățirea caracteristicilor hardware (scalare verticală) sau prin



adăugarea de echipamente hardware adiționale (scalare orizontală).

În mod ideal, răspunsul unui sistem la scalare ar trebui să fie liniar. Cu alte cuvinte, dublarea caracteristicilor hardware sau adăugarea unui server nou ar trebui să permită unui sistem acomodarea unui volum dublu de activitate. În mod evident, acest răspuns liniar este imposibil de obținut în practică.

Scalabilitatea unui sistem este o caracteristică foarte importantă atunci când volumele gestionate sunt mari (număr mare de utilizatori, volum mare de date etc). Pentru aceste situații trebuie să formulați cerințe de scalabilitate.

În general, scalabilitatea verticală nu presupune neapărat intervenții din partea dezvoltatorilor. Hardware-ul poate fi îmbunătățit și soluția, cel mai probabil, va funcționa corect fără a fi necesare modificări. Scalabilitatea orizontală are însă un impact important în arhitectura sistemului. Astfel, este important să menționați această cerință în caietul de sarcini în cazul în care sistemul trebuie să suporte scalabilitate orizontală.

### 3.6.21 Securitate și confidențialitate

Un alt set de cerințe care nu ar trebui să lipsească din niciun caiet de sarcini sunt cele legate de securitate. Domeniul securității este unul foarte vast și sensibil. În acest ghid am încercat să structurăm câteva cerințe minime legate de securitate fără a încerca să acoperim toate scenariile posibile.

**Autentificarea** reprezintă procesul prin care un actor este identificat (de obicei actorul este un utilizator dar poate fi vorba și de un alt sistem informatic). Cerințele de securitate trebuie să clarifice mecanismul de autentificare dorit. Câteva tipuri uzuale de autentificare sunt:

- Pe bază de utilizator și parolă;
- Integrat active directory;
- Cu certificate electronice;
- Cu dispozitive tip digipass;
- Cu coduri transmise via SMS (2 factor authentication);
- Folosind standard de tip OpenAuth sau OpenID.

**Autorizarea** este procesul prin care unui actor îi este permis accesul la resursele sistemului (funcționalități, date, rapoarte etc.). În majoritatea sistemelor prin procesul de autorizare se poate controla accesul la funcțiile sistemului. Spre exemplu: "Doar rolul financiar are acces la rapoartele contabile". În anumite situații este necesară și controlarea accesului la date. Spre exemplu: "Utilizatorii dintr-o unitate teritorială vor putea accesa doar înregistrările contribuabililor din teritoriul respectiv", sau "Un utilizator cu rolul de front-office va putea accesa doar înregistrările proprii".

Pentru a detalia cerințele de autorizare este utilă folosirea unei matrici unde funcționalitățile sunt aranjate pe linii, coloanele reprezintă rolurile din aplicație iar intersecția reprezintă tipul de acces permis/nepermis. O matrice similară poate fi folosită și pentru accesul la date/înregistrări.

Recomandăm ca cel puțin autorizarea la nivelul funcțiilor aplicației să fie configurabilă și să permită modificări ulterioare, respectiv permisiunile rolurilor și utilizatorilor să fie configurabile.

#### Protocoale de comunicații

Un aspect important legat de securitate unui sistem îl reprezintă protocoalele de comunicații folosite pentru comunicarea între diversele componente (client-server, server-server). Este o bună practică ca orice comunicație între diversele componente ale sistemului să fie criptată (ex: SSL) dacă există riscul ca respectiva comunicație să fie expusă. Spre exemplu, în cazul unui sistem disponibil online (inclusiv website de prezentare), este obligatorie o conexiune securizată între browser și server (respectiv folosirea HTTPS).

Procedura de actualizare a certificatelor SSL este un aspect care nu trebuie omis. Aceste certificate au o durată de valabilitate și un cost aferent, de aceea este important să aveți foarte bine organizat acest proces de achiziție și prelungire, pentru a evita situația în care sistemul să nu fie operațional la expirarea lor.

#### **Criptare și semnătură electronică**

În situația în care sistemul gestionează date sensibile și este necesar un grad ridicat de securitate puteți solicita criptarea informațiilor sensibile. Implementarea semnăturii electronice poate fi de asemenea necesară dacă anumite acțiuni ale utilizatorilor trebuie să fie opozabile legal. În cazul în care sistemul necesită criptarea anumitor informații sau semnătură electronică menționați aceste cerințe explicit în caietul de sarcini.

Cerințele de securitate pot avea un impact considerabil asupra costurilor și arhitecturii sistemului. Ca și în celelalte cazuri, cerințele de securitate trebuie să fie juste și adecvate scopului proiectului.

### **3.6.22 Usurința în utilizare (Usability)**

Ușurința în utilizare (Usability) a unui sistem se referă la ușurința cu care utilizatori pot utiliza și învăța respectivul sistem. Chiar dacă ușurința în utilizare, performanța și calitatea unui sistem ar trebui să fie caracteristici implicite ale oricărui sistem, este util să menționați aceste cerințe în caietul de sarcini în special dacă există cerințe speciale.

Chiar dacă poate părea lipsită de importanță, ușurința în utilizare se poate dovedi esențială dacă utilizatorii acestuia au un grad mai scăzut de pregătire tehnică sau sunt copleșiți cu alte activități ale instituției. În astfel de situații un sistem greu este încă un blocaj în activitatea de zi cu zi care într-un final conduce la ineficiența instituției.

O caracteristică a unui sistem utilizabil este **intuitivitatea și familiaritatea** interfeței utilizator. Cu alte cuvinte un utilizator ar trebui să poată folosi un sistem informatic nou cu un minim de instruire sau chiar fără instruire. Utilizarea unei interfețe familiare și consistente contribuie foarte mult la ușurința în utilizare a sistemului.

Utilizabilitatea unui sistem este de asemenea caracterizată de **ușurința în utilizare** și de **eficiența utilizării**. Ușurința și eficiența pot fi măsurate prin numărul de interacțiuni (mouse click, butoane apășate, câmpuri completate șamd) pe care un utilizator trebuie să le facă pentru a duce la îndeplinire o anumită operație. În anumite cazuri minimizarea numărului de interacțiuni poate face diferența între un sistem de succes și unul care nu își atinge scopul.

În determinarea cerințelor de ușurință în utilizare este utilă definirea și caracterizarea audienței căreia sistemul se adresează. Această caracterizare poate face parte din caietul de sarcini, separat sau ca descriere a rolurilor, și va fi cu siguranță utilă furnizorilor în proiectarea soluției.

În special în sistemele cu volume mari de operații sau unde viteza de lucru este importantă (spre exemplu aplicații de ghișeu public) pot exista cerințe speciale de eficiență pe care le puteți solicita în caietul de sarcini. Spre exemplu "operația de completare a formularului X trebuie să poată fi realizată de un utilizator în maximum 5 minute folosind doar tastatura, presupunând că datele necesare sunt disponibile și viteza acestuia de tastare este satisfăcătoare".

În cazul sistemelor online disponibile publicului, ușurința în utilizare este de asemenea foarte importantă. Dacă pentru angajații instituției există posibilitatea instruirii, este foarte puțin probabil ca utilizatorii externi (publicul) să aloce timp unei instruirii. În astfel de cazuri sistemul trebuie să fie intuitiv și familiar pentru a fi eficient și a-și atinge scopul.

Pentru astfel de cazuri este utilă identificarea funcționalităților cheie pentru care viteza și eficiența sunt importante. Aceste cerințe trebuie detaliate și solicitate explicit în caietul de sarcini.

### 3.6.23 Tehnologie

În situația în care din diverse motive soluția solicitată trebuie să folosească anumite tehnologii sau există constrângeri în privința tehnologiilor ce pot fi folosite la implementare, este necesar ca astfel de constrângeri să fie documentate în caietul de sarcini.

Ca o excepție, o limitare tehnică acceptabilă ar fi să solicitați ca noua soluție să fie compatibilă cu soluțiile software existente, sau chiar să folosească același limbaj de programare sau un sistem de bază de date deja licențiat (dacă gradul de uzură morală și tehnică a soluțiilor existente o permit). Acest lucru va garanta un cost redus de exploatare pentru toate aplicațiile beneficiarului, prin reducerea nivelului de complexitate a întregului portofoliu de aplicații.

**IMPORTANT:** introducerea unor constrângeri tehnologice poate limita drastic concurența. Dacă astfel de constrângeri sunt necesare există acestea trebuie justificate și susținute. Este foarte posibil ca introducerea unor constrângeri tehnice să cauzeze contestarea caietului de sarcini și posibil chiar anularea unei competiții.

## 3.7 Cerințe de comunicare și organizare

**“Care este structura echipelor de proiect?”, “Care sunt responsabilitățile?”, “Cum vom comunica?”**

Succesul unui proiect software depinde în mare măsură de resursele umane implicate în proiect și de modul în care acestea reușesc să comunice eficient pentru a depăși dificultăți și blocaje inerente în cadrul oricărui proiect.

### Structura echipelor de proiect

În privința organizării este recomandată definirea cerințelor minime privind structura echipelor implicate în proiect atât din partea furnizorului cât și din partea beneficiarului.

În calitate de beneficiar trebuie să vă asigurați că alocați proiectului suficiente resurse umane interne, cu competențe adecvate scopului proiectului precum și un buget de timp/efort alocat proiectului. Chiar dacă structura echipei beneficiarului nu constituie o cerință propriu-zisă a caietului de sarcini, definirea structurii acesteia este necesară încă din prima etapă a proiectului.

În mod similar trebuie să vă asigurați că viitorul furnizor va aloca o echipă de proiect adecvată și competentă. Structura acestei echipe poate constitui o cerință a caietului de sarcini la care furnizorul trebuie să răspundă prin oferta sa.

Structura fiecărei echipe (beneficiar/furnizor) se realizează prin definirea rolurilor existente în cadrul echipei. Pentru fiecare rol este necesară cel puțin definirea următoarelor atribute:

- Denumirea rolului;
- Listă de responsabilități în cadrul proiectului;
- Listă de competențe pe care trebuie să le dețină;
- Cerințe și constrângeri (experiență, certificări, samd.)

**IMPORTANT:** modul în care sunt formulate cerințele privind echipa furnizorului poate limita considerabil numărul de furnizori. Recomandarea noastră este să definiți strict cerințele necesare proiectului evitând introducerea de cerințe vag relevante sau exotice (spre exemplu certificări tehnice rare).

În funcție de specificul proiectului, un rol poate fi îndeplinit de una sau mai multe persoane (ex: dezvoltator sau tester). De asemenea, o persoană poate cumula unul sau mai multe roluri. În cazul în care există constrângeri în privința rolurilor, este necesară definirea acestora.

În cadrul fiecărei echipe, recomandăm definirea rolului de coordonator unic de proiect care trebuie asigurat de o singură persoană. Acest rol coordonează echipa proprie și are experiența, capacitatea și autoritatea de a lua decizii în cadrul proiectului.

În măsura în care complexitatea proiectului o permite, această persoană poate fi și punctul de contact unic din partea autorității contractante sau a furnizorului.

În mod evident, nu putem recomanda configurația echipei potrivita pentru orice proiect și nu ne propunem acest lucru. Am încercat însă în cele ce urmează să identificăm câteva roluri des întâlnite în cadrul proiectelor software.

Exemple de roluri posibile în echipa beneficiarului:

- **Coordonator proiect**  
Rolul este descris mai sus. În cazul proiectelor de dimensiune și complexitate mică, acesta poate fi și unicul punct de contact în relația cu furnizorul. În cazurile în care proiectul permite, această abordare este foarte recomandată din rațiuni de eficiență.
- **Responsabil IT**  
În majoritatea cazurilor este necesară alocarea unui responsabil IT care cunoaște detaliile infrastructurii IT&C a beneficiarului. Această persoană poate clarifica detaliile tehnice și, de asemenea, poate fi implicată în procesul de acceptanță și implementare a sistemului.
- **Specialist domeniu / responsabil formulare cerințe (ex: Specialist comunicare publică)**  
În formularea cerințelor funcționale și ulterior în detalierea acestora, este necesară implicarea specialiștilor în domeniile respective de activitate. Aceștia sunt persoane din organizația beneficiarului care cunosc în detaliu procesele de lucru, legislația și problemele cu care se confruntă organizația. Inițial rolul acestora este de formulare a problemelor și cerințelor funcționale. Ulterior, în cadrul proiectului, aceștia vor fi implicați în detalierea cerințelor funcționale, în validarea diverselor livrabile (specificații, machetă șamd) precum și în procesul de acceptanță.
- **Project manager**  
În general acest rol va fi probabil îndeplinit de coordonatorul proiectului. În cadrul proiectelor foarte complexe este utilă implicarea unuia sau mai multor specialiști în project management din partea beneficiarului. Acest rol asigură planificarea activităților în echipa beneficiarului precum și coordonarea și validarea activităților cu responsabilul din partea furnizorului.
- **Utilizatori / viitori utilizatori**  
Implicarea utilizatorilor unui sistem existent sau a viitorilor utilizatori, reprezintă o modalitate foarte practică de a identifica problemele cu care aceștia se confruntă, precum și pentru a valida cerințele unui viitor sistem.
- **Dezvoltatori și alte roluri tehnice**  
În cazul în care beneficiarul dispune de echipă proprie de dezvoltare, poate fi utilă implicarea acestor specialiști în cadrul proiectului. Această implicare poate fi foarte relevantă în cazurile în care proiectul prevede integrări cu alte aplicații utilizate de beneficiar, precum și pentru preluarea sarcinilor tehnice ulterior încheierii proiectului (ex: extindere, mentenanță etc.).

Echipa de roluri în echipa furnizorului

- **Coordonator proiect**  
Este recomandat ca Furnizorul să desemneze un responsabil unic de coordonarea proiectului. Ca și în cazul beneficiarului, pentru proiecte de complexitate mică și eventual medie, coordonatorul proiectului poate prelua și alte roluri în cadrul echipei și poate fi unicul punct de contact pentru echipa beneficiarului. Este esențial ca această persoană să aibă putere de decizie în compania furnizorului, astfel încât să poată soluționa toate situațiile apărute în derularea proiectului.
- **Project manager**  
Rol responsabil de planificarea activităților furnizorului.
- **Coordonator echipă tehnică**  
Coordonează echipa tehnică a furnizorului. În funcție de complexitatea proiectului poate îndeplini și alte roluri tehnice în cadrul proiectului.
- **Arhitect soluții software**  
Responsabil de definirea arhitecturii software a soluției livrate. Pentru proiecte de complexitate foarte mare poate fi necesar un arhitect software dedicat.

- **Arhitect baze de date**  
Responsabil de proiectarea bazelor de date.
- **Analist software**  
Responsabil de discuțiile de analiză cu specialiștii beneficiarului, de elaborarea specificațiilor, realizarea machetelor funcționale, derularea testelor de acceptanță etc. Rolul analistului este unul cheie în derularea oricărui proiect software. Obiectivul principal al analistului este să se asigure că problemele și cerințele beneficiarului sunt adresate corespunzător în soluția software propusă de furnizor. Analistul are, de asemenea, rolul de a intermedia comunicarea dintre echipa tehnică a furnizorului și echipa de specialiști a beneficiarului și de a clarifica eventualele inconsistențe și blocaje.
- **Arhitect UI/UX**  
Rol responsabil de definirea principiilor și a arhitecturii interfeței utilizator (UI). Rolul de arhitectul UI este de a asigura o experiență utilizator (UX) cât mai intuitivă, consistentă și eficientă. În cadrul proiectelor cu cerințe deosebite în privința interfeței utilizator (ex: aplicații ghișeu sau aplicații online dedicate publicului) este importantă definirea acestui rol.
- **Designer grafic UI / designer web**  
Responsabil de aspectul grafic al interfeței utilizator.
- **Dezvoltator software / Consultant implementare**  
Evident, din echipa furnizorului nu pot lipsi dezvoltatorii software în cazul soluțiilor personalizate sau consultanții de implementare în cazul implementării de produse software existente (ex: ERP, CRM, DM etc.). În funcție de cerințele proiectului, specializările dezvoltatorilor pot fi foarte diverse. Câteva posibile exemple de specializări sunt: Web, Front-End, Baze de date, Full stack, BI, mobile, embedded. De asemenea, dezvoltatorii pot fi specializații în anumite tehnologii sau limbaje.
- **Tester / coordonator echipă testare**  
Asigurarea de către furnizor a unei echipe de testare separată de echipa de dezvoltare este o practică recomandată.
- **Administrator de sistem / infrastructură**  
Sistemele informatice utilizate, atât în procesul de dezvoltare cât și după punerea în producție a soluțiilor, necesită configurare, actualizări ale diverselor pachete de software precum și intervenții în perioada de mentenanță.

Echipa de conducere

- **Sponsori executivi**  
Atât din partea beneficiarului cât și din partea furnizorului este recomandată identificarea unui Sponsor executiv. Aceste persoane fac parte din echipele de conducere ale beneficiarului și furnizorului și asigură o interfață permanentă de comunicare la nivel înalt.

**Notă:** în conformitate cu art 179, lit g) din legea 98/2016, în relație cu această poziție, se pot stabili criteriile de calificare pentru personalul de conducere al instituției, în măsura în care astfel de criterii sunt justificate și aceste persoane nu participă în echipa de implementare conform Art. 32 alin (5) din HG 395/2016.

### Comunicare

Asigurarea unei comunicări eficiente între echipa beneficiarului și a furnizorului este un aspect cheie în succesul oricărui proiect software. De cele mai multe ori între cele două echipe există inițial o barieră de comunicare cauzată în principal de zonele de specialitate diferite. Eliminarea acestei bariere trebuie să fie o prioritate pentru ambele echipe, iar primii pași în această direcție constau exact în elaborarea unui caiet de sarcini pe cât de clar posibil.

### Instrumente de comunicare structurată

Instrumentele de comunicare și colaborare folosite pot ușura și structura comunicarea pe parcursul proiectului. Utilizarea mijloacelor clasice de comunicare este evidentă implicită (telefon, email, întâlniri). Pe lângă acestea, recomandăm însă și utilizarea unor instrumente de colaborare moderne și structurate, spre exemplu portaluri de colaborare disponibile online, aplicații cloud pentru editare colaborativă (ex: Google Apps, Microsoft Office 365, suita Atlassian etc.), aplicații de comunicare online (soluții de tip forum sau de mesagerie) sau aplicații de planificare a task-urilor. Multe astfel de instrumente sunt gratuite sau presupun costuri rezonabile de utilizare și

pot avea un impact foarte mare în buna desfășurare a unui proiect.

Utilizarea unor astfel de instrumente poate asigura accesul imediat și transparent al echipelor la toate detaliile legate de proiect (documente, detalii de contact, activități, calendar, situații blocante șamd). În caietul de sarcini puteți prevedea cerința ca astfel de instrumente să fie utilizate sau eventual chiar puse la dispoziție de către furnizor.

### Identificarea și rezoluția blocajelor (Issue management)

Blocajele sunt inerente în cadrul oricărui proiect și pot fi cauzate de deficiente de comunicare, de lipsa de resurse (inclusiv timp), de întârzieri, de neclaritatea sau instabilitatea unor cerințe și multe alte cauze. Un prim pas în rezolvarea blocajelor este identificarea acestora.

Recomandăm ca în orice proiect să fie definit un mecanism simplu și accesibil prin care orice membru al celor două echipe poate semnaliza apariția unui blocaj în proiect care are potențialul de a afecta întregul proiect. Acest mecanism poate presupune simpla comunicare a unui blocaj către coordonatorii de proiect via email sau adăugarea blocajului într-un sistem dedicat (ex: o listă dedicată într-un portal online sau un document tip spreadsheet).

Evidența centralizată a blocajelor active și adresarea acestora în cadrul întâlnirilor de proiect, este o practică pe care o recomandăm în orice proiect. Eliminarea cât mai rapidă a acestor blocaje este un obiectiv principal al coordonatorilor de proiect.

### Raportarea progresului (Status reporting)

Pe parcursul proiectului este foarte recomandat să urmăriți progresul înregistrat pentru a evita eventualele surprize și întârzieri la finalul proiectului. Prin urmare, indiferent de procesul de dezvoltare folosit, este recomandat să prevedeați încă din caietul de sarcini cerințe minimale privind raportarea progresului, întâlnirile de stadiu (status) și formatul acestora, precum și eventualele documente pe care furnizorul trebuie să le pregătească (rapoarte de stadiu/status).

Recomandăm ca frecvența întâlnirilor să fie între 2 și 4 săptămâni, cu participarea obligatorie a coordonatorilor de proiect, precum și a rolurilor cheie din fiecare echipă.

Agenda acestor întâlniri trebuie să cuprindă cel puțin următoarele:

- Măsurarea progresului și identificarea eventualelor întârzieri;
- Eliminarea blocajelor apărute în cazul în care nu au fost adresate imediat;
- Clarificarea eventualelor nelămuriri funcționale/tehnice;
- Validarea livrabilelor intermediare și rezolvarea eventualelor inconsistențe;
- Demonstrarea funcționalităților implementate (chiar și parțiale) și rezolvarea eventualelor diferențe de așteptări.

## 3.8 Livrabilele proiectului

### “Care sunt rezultatele concrete ale proiectului?”

Livrabilele proiectului reprezintă toate elementele pe care furnizorul se angajează să le livreze beneficiarului în cadrul contractului. Pe lângă soluția software funcțională care este evident livrabilul principal, un proiect software cuprinde și alte elemente care sunt necesare în derularea unui proiect și, ulterior, instalării în mediul de producție. În cazul proiectelor de complexitate medie și mare, urmărirea unui proces structurat de dezvoltare și elaborarea documentației de proiect necesare poate face diferența între un proiect de succes și unul eșuat.

Caietul de sarcini trebuie să conțină toate cerințele în privința livrabilelor pe care furnizorul trebuie să le furnizeze în cadrul proiectului. Livrabilele intermediare sunt cel puțin la fel de importante ca soluția propriu-zisă astfel încât solicitarea acestora nu trebuie lăsată la voia întâmplării presupunând că livrarea acestora este implicită.

Recomandăm ca lista livrabilelor solicitate să fie descrisă într-un capitol separat al

documentației de atribuire chiar dacă livrabilele pot fi referințe și în alte secțiuni ale acesteia. De asemenea, este de dorit ca pentru fiecare livrabil să fie detaliate cerințele minime de calitate pe care trebuie să le respecte, precum și condițiile de acceptanță. Recomandăm, de asemenea, ca și livrabilele intermediare să facă obiectul unor acceptanțe intermediare.

Lista propusă mai jos conține o serie de livrabile comune întâlnite în cadrul proiectelor software. Ca și în celelalte cazuri, trebuie privită ca un punct de plecare în elaborarea caietului de sarcini și adaptată la cerințele dumneavoastră prin eliminarea sau adăugare de noi elemente.

### Cod sursă

Codul sursă poate fi transferat la finalul proiectului în cazul în care contractul prevede că beneficiarul va deține proprietatea intelectuală, precum și în cazul în care acordul de licențiere prevede și posibilitatea modificării codului sursă de către beneficiar. Se pot prevedea, de asemenea, și livrări intermediare.

**IMPORTANT:** Pentru a evita o relație de dependență, trebuie să vă asigurați că veți putea modifica codul sursă independent de furnizorul inițial, transfera codul sursă către un alt furnizor sau contracta un alt furnizor pentru modificări sau servicii de mentenanță. Opțiunile legale pe care le aveți la dispoziție sunt descrise în detaliu în capitolul 3.6 Cerințe nefuncționale, secțiunea "Legal și licențiere".

În astfel de cazuri puteți specifica limba în care trebuie scris codul (adică limba folosită pentru comentariile din cod, denumirea variabilelor, metodelor etc). De asemenea, puteți solicita ca codul sursă să respecte anumite reguli și standarde de calitate.

Exemple de reguli de scriere a codului (coding guidelines):

- Java – <http://www.oracle.com/technetwork/java/index-135089.html>
- C# – <https://msdn.microsoft.com/en-us/library/ff926074.aspx>
- PHP – <http://www.php-fig.org/psr/psr-1/> / <http://www.php-fig.org/psr/psr-2/>

### Kituri de instalare / executabile

Unul din livrabilele proiectului poate fi kitul de instalare al soluției sau fișierele executabile compilate. În cazul în care furnizorul va livra și un kit de instalare, acesta trebuie supus procesului de acceptanță. În cazul în care un kit de instalare nu este solicitat, este obligatoriu însă ca lista de livrabile să includă un document ce descrie pașii necesari pentru instalarea soluției.

### Documentație de utilizare

Chiar dacă sistemul software este intuitiv și ușor de utilizat, este indicat ca orice sistem să dispună de un manual de utilizare care să fie accesibil atât în interiorul aplicației cât și online (web) și offline (pdf, chm, etc.). Este important ca manualul de utilizare să acopere și funcționalitatea complexă a aplicației care poate nu este la fel de intuitivă.

### Documentație de administrare și operare

Documentația de administrare și operare cuprinde instrucțiuni adresate administratorilor de sistem care pot cuprinde:

- Procedura de instalare a soluției;
- Monitorizarea parametrilor de funcționare a aplicației;
- Administrarea utilizatorilor și a drepturilor;
- Configurarea parametrilor tehnici;
- Taskuri de mentenanță recurente (ex: curățare loguri);
- Raportarea și investigarea erorilor de funcționare;
- Modalități rapide de punere în funcțiune;
- etc.

Orice modificare viitoare a soluției trebuie să fie însoțită de actualizarea acestor documente.

### Strategia de backup

Documentul descrie strategia de backup a sistemului și cuprinde:



- Elementele care trebuie salvate;
- Frecvența salvărilor și tipul acestora;
- Responsabilitățile pentru backup;
- Locația unde sunt păstrate salvările;
- Perioada de păstrare a salvărilor anterioare;
- Etc.

### Procedura de recuperare în caz de dezastru

Procedura de recuperare în caz de dezastru conține pașii exacti care trebuie urmăriți pentru restaurarea sistemului în cazul apariției unui dezastru. Este important ca acest document să fie testat și actualizat corespunzător pentru a fi imediat disponibil într-o situație de urgență. Recomandăm ca procedura de recuperare să fie disponibilă și ca document separat (chiar dacă poate face parte și din manualul de administrare).

### Procedura de instalare

Procedura de instalare descrie toți pașii necesari instalării și configurării sistemului. În mod uzual, procedura de instalare trebuie să detalieze cerințele preliminare instalării (prerequisites), pașii de instalare și configurare, precum și un set minimal de teste prin care se poate verifica funcționarea corectă a aplicației.

### Viziune și scop (vision & scope)

Documentul de viziune și scop descrie scopul și obiectivele sistemului și este relevant ca prim document ce trebuie parcurs de o persoană nou introdusă în proiect. Documentul trebuie să ofere o imagine de ansamblu asupra obiectivelor proiectului și a rațiunilor acestuia. În mare, acesta poate să conțină descrierea instituției, problemele actuale și obiectivele proiectului așa cum au fost formulate în documentația de atribuire.

### Specificație funcțională și tehnică

Specificația funcțională și tehnică poate fi un document sau un set de documente în care cerințele din caietul de sarcini sunt clarificate, detaliate și structurate sub forma de funcții sau caracteristici tehnice.

### Document de arhitectura și design

Documentul de arhitectura și design este în principal un document tehnic în care sunt documentate deciziile tehnice majore luate în implementarea soluției.

Documentul poate să cuprindă următoarele:

- Tehnologiile folosite;
- Componentele logice;
- Componentele fizice ale soluției;
- Modul în care componentele sunt conectate și comunică;
- Protocoalele de comunicație utilizate;
- Componente terțe folosite dacă este cazul;
- Eventuale paternuri utilizate;
- etc.

### Documentația bazelor de date

În cazul în care soluția software presupune și utilizarea uneia sau a mai multor baze de date, unul din livrabilele furnizate trebuie să fie documentația bazei/bazelor de date. În cazul soluțiilor proprietare sau open-source, această documentație nu este neapărat relevantă, însă orice extensie a soluțiilor standard trebuie documentată inclusiv la nivelul bazelor de date.

Indiferent însă de tipul soluției adoptate (proprietară, open-source sau personalizată) trebuie să vă asigurați că veți avea proprietatea datelor stocate și posibilitatea migrării acestora în altă soluție.

Documentația solicitată poate să cuprindă următoarele informații:



- Diagrama bazei de date;
- Descrierea tabelor;
- Semnificația câmpurilor (cel puțin a celor neintuitive);
- Descrierea relațiilor dintre tabele;
- Denormalizări în cazul în care există;
- Securitate (modul în care se realizează autentificarea și autorizarea la nivelul bazei dar și alte detalii importante de securizare precum criptarea bazei de date, firewall pentru baza de date sau parte de audit);
- Strategia de indexare;
- Constrângeri;
- Elemente de programabilitate dacă este cazul (proceduri stocate, funcții, view-uri etc.);
- Etc.

### Arhitectura Interfeței utilizator (UI)

Arhitectura interfeței utilizator detaliază structura interfeței utilizator și regulile care guvernează interfața și poate cuprinde următoarele:

- Tipurile de ecrane și structura fiecăruia;
- Principiile de prezentare a informației;
- Structura meniurilor;
- Adaptabilitate la diverse dimensiuni de ecran dacă este cazul;
- Mecanismele de navigare între ecrane;
- Modul de prezentare a erorilor de validare;
- Modul de prezentare a mesajelor și alertelor;
- Comportamentul controalelor de input;
- Elemente reutilizabile de interfață;
- Persistarea preferințelor de interfață dacă este cazul;
- etc.

Existența unui document de arhitectură UI asigură dezvoltarea unei interfețe utilizator consistente.

### Macheta interfeței utilizator (UI)

Macheta interfeței utilizator (sau prototipul UI) este un livrabil cheie în cadrul unui proiect care implică și o interfața utilizator. Macheta poate fi un document sau chiar o aplicație în care doar interfața utilizator este funcțională fără ca logica aplicației să fie implementată. Scopul machetei este să ilustreze vizual modul în care cerințele funcționale vor fi implementate practic.

Acest livrabil este foarte important pentru că este foarte ușor de înțeles de către viitorii beneficiari ai sistemului care pot vizualiza și interacționa cu viitoarea aplicație. Astfel, eventualele erori sau neînțelegeri ale cerințelor pot fi corectate încă de la începutul proiectului cu costuri mult mai mici. Macheta poate fi folosită ca referință și de către dezvoltatori pe parcursul implementării.

Macheta este realizată în paralel cu specificația funcțională și trebuie livrată și acceptată la începutul proiectului înainte de a începe dezvoltarea efectivă. O machetă corect realizată va corespunde în foarte mare măsură cu aplicația finală.

Pentru ca o machetă să fie relevantă în cadrul proiectului, aceasta trebuie să fie interactivă, respectiv să permită viitorilor utilizatori să interacționeze prin navigare, completarea formularelor etc. În cazul în care macheta este compusă doar din ecrane statice beneficiile aduse vor fi mai mici.

Machetele UI pot fi împărțite în trei categorii:

- Fidelitate scăzută (low fidelity) – schițe desenate de mână sau folosind programe de desen;
- Fidelitate medie – acestea pot fi realizate folosind aplicații specializate pentru machetare sau pot chiar să fie programate. Machetele de fidelitate medie ilustrează structura interfeței și toate detaliile de interfață fără însă a prinde și design-ul grafic final al soluției;

- Fidelitate mare (Hi fi) – machetele de fidelitate mare cuprind și detaliile grafice și alte efecte ale interfeței și evident presupun un efort mare de implementare.

Machetele de fidelitate medie oferă un compromis foarte bun între costurile relativ scăzute de realizare și beneficiile pe care le aduc și sunt astfel recomandate în majoritatea proiectelor.

Recomandăm, de asemenea, ca machetele UI să fie realizate folosind instrumente dedicate pentru machetare care oferă productivitate și permit iterarea rapidă (Balsamiq, Axure, Justinimd, Moqups etc.).

**IMPORTANT:** În cazul în care macheta UI este dezvoltată prin scriere de cod (de exemplu: o aplicație web sau desktop), acest cod nu trebuie folosit în cadrul proiectului efectiv. Motivul este că în dezvoltarea machetei sunt acceptabile compromisuri privind calitatea codului pentru a câștiga productivitate, compromisuri care însă nu pot fi acceptate pentru soluția finală.

### Design grafic interfață utilizator (UI)

Designul grafic ilustrează modul în care interfața utilizator va arăta după implementare și cuprinde schema de culori folosită, fonturi, spațierea elementelor etc. În general, designul grafic presupune livrarea unor propuneri grafice sub formă de imagini statice (jpeg, pdf) cu ecrane relevante din aplicație.

În cazul în care instituția beneficiarului trebuie să respecte anumite criterii unitare de identitate sau identitatea grafică a fost stabilită anterior, este important să comunicați toate aceste cerințe prin intermediul caietului de sarcini.

### Plan de testare

Planul de testare documentează strategia de testare a aplicației și poate include:

- Metodele de testare;
- Mediul de testare;
- Descrierea rolurilor în echipa de testare;
- Responsabilități în cadrul echipei de testare;
- Lista de funcționalități testate sau excluse de la testare;
- Criteriile de succes și eșec ale testelor;
- Modul de clasificare al defectelor (severitate, frecvență, prioritate etc.);
- Calendarul testelor.

Se recomandă ca planul de testare să aibă anexată și lista testelor folosite (cazuri de test), precum și detaliile acestora (pași de testare, rezultate așteptate etc.).

### Plan de teste de acceptanță

Planul de teste de acceptanță cuprinde testele pe care sistemul trebuie să le treacă pentru a fi acceptat. De asemenea, planul de acceptanță detaliază criteriile care trebuie îndeplinite pentru ca soluția să fie acceptată.

În general, planul de acceptanță este elaborat de către furnizor la începutul proiectului, însă validarea conținutului este prerogativul beneficiarului.

### Aplicații conexe

Pe lângă soluția software, există situații în care sunt dezvoltate și alte aplicații conexe. Acestea ar putea fi diverse instrumente/aplicații de administrare sau configurare, aplicații utilizate pentru mentenanță, scripturi de migrare a datelor sau alte scripturi utilitare. În cazul în care astfel de aplicații fac parte din scopul proiectului, este utilă solicitarea acestora în caietul de sarcini.

### Hardware

În mod evident, în cazul în care prin caietul de sarcini este solicitat și hardware, acesta trebuie menționat în lista de livrabile.

## Cloud

În situația în care necesarul hardware, de stocare sau de procesare variază puternic în timpul exploatării aplicației sau doriți soluții cu un înalt grad de disponibilitate, este recomandat să luați în calcul și soluțiile oferite de furnizorii de cloud-computing. O astfel de soluție vă poate oferi flexibilitate în modificarea infrastructurii sau metode de a reduce costurile în momentele în care aplicația dumneavoastră nu este încărcată (de exemplu: pe timpul nopții dacă aplicația are o încărcare mare doar în timpul zilei).

În cazul în care soluția presupune sau permite utilizarea platformelor de cloud-computing (Microsoft Azure, Amazon AWS, IBM Cloud, Oracle Cloud, Digital Ocean, etc.), acest lucru trebuie prevăzut în caietul de sarcini. Costurile de utilizare ale acestor platforme variază în funcție de utilizare, astfel estimarea costurilor în faza de ofertare poate fi foarte dificilă. Din acest motiv recomandăm să solicitați în mod explicit detalierea structurii acestor costuri pe întreaga durată a proiectului. În cazul în care aceste costuri vor fi suportate de beneficiar, este indicată facturarea sau refacturarea transparentă de către furnizor (respectiv justificarea detaliată a consumului). În cazul în care aceste costuri vor fi suportate de furnizor, având în vedere dificultatea estimării acestora, este recomandată alocarea unui limite de buget (lunară/ anuală) până la care furnizorul va suporta aceste costuri, eventual cu reportarea surplusului neconsumat.

## Servicii conexe

Apar și situații în care sunt solicitate furnizorului și alte servicii conexe care nu sunt neapărat implicite. Astfel de exemple ar putea fi servicii de instalare și punere în funcțiune hardware, servicii de configurare de infrastructură (hardware, comunicații, software de bază), teste de securitate, alte servicii de consultanță etc. Toate aceste servicii trebuie solicitate explicit în caietul de sarcini.

## 3.9 Procesul de acceptanță

### “Cum vom verifica și valida rezultatele proiectului?”

Procesul de acceptanță este procesul prin care **toate** livrabilele proiectului sunt validate și recepționate de către beneficiar ca fiind conforme cu cerințele formulate și cu obligațiile asumate de furnizor. Astfel, procesul de acceptanță nu se referă doar la acceptarea finală a soluției software (acceptanța finală), ci și la acceptanțe intermediare, care sunt derulate pe parcursul proiectului, pentru validarea livrabilelor intermediare sau conexe proiectului (ex: specificații, machetă, hardware etc).

Caietul de sarcini este referința principală și fundația oricărui proiect. Un caiet de sarcini cu cerințe clare și detaliate va facilita procesul de acceptanță, în timp ce un caiet de sarcini vag sau succint poate conduce chiar la imposibilitatea refuzării unor livrabile de slabă calitate. Din acest motiv este important să specificați în caietul de sarcini criteriile de acceptare sau refuzare, precum și criteriile de calitate și gradul de detaliere necesar pentru celelalte livrabile ale proiectului, altele decât soluția propriu-zisă.

Cu excepția proiectelor care au încă din faza de licitație pregătită o specificație tehnică detaliată anexată caietului de sarcini (care poate este rezultatul unui proiect de consultanță separat), de cele mai multe ori proiectele software încep cu o fază de analiză detaliată a cerințelor. Scopul analizei este clarificarea și detalierea cerințelor formulate în caietul de sarcini și are ca rezultat principal specificația tehnică și funcțională, macheta funcțională și, în funcție de proiect, și alte livrabile (design grafic, arhitectură, design-ul bazelor de date etc.).

În cadrul fazei de analiză este foarte posibil ca anumite cerințe formulate inițial în caietul de sarcini să se transforme prin detaliere, alte cerințe este posibil să fie eliminate și la fel de posibil este ca noi cerințe să fie identificate, care nu au fost observate inițial. Într-o anumită măsură, astfel de modificări sunt normale și sunt rezultatul negocierii unui compromis constructiv între beneficiar și furnizor. Într-o astfel de negociere caietul de sarcini devine un document de referință. Și, în această situație, un caiet de sarcini clar și detaliat va ajuta la rezolvarea rapidă a blocajelor, în timp ce un caiet de sarcini vag poate genera blocaje insurmontabile.

Este recomandat ca specificațiile tehnice și funcționale, macheta funcțională (dacă există), precum și celelalte livrabile care presupun obligații pe care furnizorul și le asumă în privința soluției software, să fie supuse unui proces de acceptanță intermediară și să devină partea a contractului (prin anexarea sau prin menționarea efectelor acestor livrabile în contract). În acest fel, specificațiile tehnice pot fi folosite ca referință în procesul de acceptanță. De asemenea, în cazul în care caietul de sarcini nu era suficient de clar, faza de analiză este o oportunitate pentru a îndrepta situația.

### Acceptanța finală și teste de acceptanță

În procesul de acceptanță finală, specificațiile tehnice devin documentul de referință pentru validarea conformității soluției software. Cu alte cuvinte, soluția software finală trebuie să respecte cerințele tehnice și funcționale definite în specificație.

Pentru derularea procesului de acceptanță este recomandată utilizarea unui plan de acceptanță. Acest document este realizat și agreat în timpul sau imediat după faza de analiză și cuprinde testele pe care sistemul trebuie să le treacă cu succes pentru a fi considerat conform. Chiar dacă planul de teste de acceptanță este realizat de furnizor, conținutul acestuia trebuie validat și agreat de beneficiar. În această privință trebuie să vă asigurați că, cel puțin funcționalitățile cheie ale sistemului precum și cerințele nefuncționale, sunt acoperite de testele prevăzute în planul de acceptanță.

Pe lângă testarea funcționalității (cerințele funcționale), este foarte recomandat să verificați și îndeplinirea cerințelor tehnice/nefuncționale, în special dacă aceste cerințe sunt importante pentru proiect (performanță, securitate, disponibilitate, redundanță etc.).

De asemenea procesul de acceptanță trebuie să valideze și celelalte livrabile ale proiectului în cazul în care acestea nu au făcut obiectul acceptanțelor intermediare.

**IMPORTANT:** testarea anumitor cerințe nefuncționale poate presupune derularea de teste specializate care pot presupune un efort considerabil din partea furnizorului. În astfel de situații trebuie ca aceste cerințe să fie cuprinse în caietul de sarcini. Exemple: îndeplinirea unor cerințe speciale de securitate poate presupune teste de penetrare sau un audit de securitate, îndeplinirea cerințelor de performanță poate presupune teste de performanță dedicate (load testing) etc.

În situația în care un plan de teste de acceptanță nu este definit, este indicat să acoperiți prin testele de acceptanță toate cerințele soluției.

**IMPORTANT:** Prin procesul de acceptanță pe care îl definiți trebuie să vă asigurați că acceptanța nu se rezumă doar la testele incluse în planul de acceptanță și că veți avea libertatea de a derula și alte teste proprii care pot cauza inclusiv refuzul sau observații ale procesului de acceptanță.

Caietul de sarcini trebuie să detalieze modul în care se va desfășura procesul de acceptanță. Printre detaliile care trebuie menționate se numără următoarele:

### Durata alocată testelor de acceptanță

Cu cât sistemul este mai complex, cu atât durata rezervată pentru acceptanță trebuie să fie mai mare.

### Mediul/infrastructura de testare

Mediul tehnic (servere, instanțe cloud etc) pentru derularea testelor de acceptanță poate fi chiar cel pe care soluția urmează să funcționeze. Este însă mult mai indicat ca un mediu tehnic dedicat, cât mai similar (preferabil identic), să fie utilizat pentru testele de acceptanță și, ulterior, pentru alte teste (denumit mediu staging, de test sau pre-producție).

### Responsabilitatea testării.

Testele de acceptanță trebuie derulate de personalul beneficiarului cu eventuala participare a furnizorului.

### Clasificarea defectelor

Pe parcursul testelor de acceptanță vor fi descoperite defecte care nu au fost identificate de furnizor în testele proprii. Clasificarea acestor defecte este importantă pentru stabilirea criteriilor de acceptanță. O clasificare uzuală a defectelor este:

O clasificare uzuală a defectelor este:

- *Defecte majore.* Defecte care blochează parțial sau integral o funcționalitate importantă a aplicației fără a exista o cale de ocolire (workaround);
- *Defecte medii.* Defecte care blochează parțial o funcționalitate pentru care există o cale de ocolire (workaround) sau defecte care blochează o funcționalitate non-critică;
- *Defecte minore.* Defecte care nu împiedică funcționarea sistemului dar pot genera un efort suplimentar pentru utilizatori pentru ocolirea lor;
- *Defecte cosmetice.* Defecte ale interfeței utilizator, greșeli de ortografie, mici probleme de compatibilitate etc.

### Rezultatul procesului de acceptanță

O practică întâlnită este ca rezultatul procesului de acceptanță să se încheie cu unul din următoarele rezultate:

- Acceptat;
- Acceptat cu observații minore;
- Acceptat cu rezerve;
- Refuzat.

Pentru fiecare caz trebuie stabilite criteriile clare care se referă în general la numărul și tipul defectelor obținute și, eventual, la timpul în care acestea pot fi rezolvate. Spre exemplu, pentru acceptarea fără observații, criteriul aplicabil ar putea fi derularea acceptanței fără identificarea vreunui defect (situație foarte puțin probabilă), în timp ce acceptat cu observații minore ar presupune identificare numai a unor defecte mici sau cosmetice care pot fi remediate într-un termen scurt bine specificat (ex: 1-2 săptămâni).

Consecințele rezultatelor acceptanței trebuie documentate în documentația de atribuire. Spre exemplu, acceptarea proiectului cu observații sau rezerve ar putea presupune ca furnizorul să remedieze defectele observate într-un termen prestabilit în documentația de atribuire.

Refuzul este situația ale cărei consecințe trebuie definite foarte bine în caietul de sarcini deoarece poate presupune consecințe legale și financiare importante. Refuzul poate presupune rezilierea imediată a contractului și eventuale penalități pentru furnizor. De asemenea, refuzul poate presupune reluarea testelor de acceptanță după scurgerea unei perioade prestabilite în care furnizorul poate remedia defectele soluției. Este recomandat ca impactul legal și financiar (ex: blocarea plăților, penalizări, reziliere) al rezultatelor acceptanței să fie detaliat în documentația de atribuire și reluat în contractul cu furnizorul.

### Procesul verbal de acceptanță

Procesul de acceptanță are ca rezultat încheierea unui proces verbal de acceptanță formal. Acest document trebuie să conțină observațiile beneficiarului, defectele identificate și eventualele observații ale furnizorului. În cazul în care este necesar, documentul trebuie să conțină și soluțiile necesare pentru defectele identificate dacă aceste soluții nu sunt evidente.

Procesul verbal trebuie să conțină și termenul agreed până la care defectele vor fi remediate sau de la care va fi reluat procesul de acceptanță în cazul refuzului. De asemenea, în cazul în care punerea în funcțiune a sistemului este amânată din cauze care țin exclusiv de furnizor (cum ar fi remedierea unor defecte identificate în procesul de acceptanță), garanția oferită de furnizor trebuie prelungită corespunzător.

## 3.10 Cerințe post implementare

### “Ce alte activități vor fi necesare pe durata de viață a soluției?”

Serviciile post implementare sunt serviciile derulate ulterior instalării în producție a soluției software sau ulterior a acceptanței finale.

Ciclul de viață al unei soluții software nu se încheie în momentul în care soluția este instalată în producție. Din acest motiv este foarte important ca autoritatea contractantă să prevadă activitățile care vor fi necesare pe toată durata de viață a soluției. Aceste activități pot fi asigurate atât de echipa internă a autorității contractante, cât și de furnizorul soluției și, în cele mai multe cazuri, este recomandată o abordare mixtă.

În cazul în care serviciile post implementare trebuie asigurate parțial sau integral, de către furnizor, acest aspect trebuie specificat clar în documentația de atribuire. De asemenea autoritatea contractantă trebuie să prevadă un buget pentru aceste servicii. Costurile pentru aceste servicii pot lua forma unui abonament lunar sau anual.

### 3.10.1 Servicii de configurare și actualizare

Aceste servicii asigură funcționarea în parametri normali a soluției software și pot să acopere partea software (configurări și actualizări ale versiunilor unor componente utilizate) dar și partea de infrastructură (arhitectura de rețea, configurări și actualizări ale softwareului aferent infrastructurii informatice, modernizări ale infrastructurii etc).

### 3.10.2 Suport tehnic

Suportul tehnic presupune servicii de asistență tehnică oferite utilizatorilor și beneficiarului pentru ca soluția să poată fi folosită în condiții normale. Suportul tehnic răspunde la întrebări de bază ale utilizatorilor (ex: "cum să fac .."), rezolvă probleme tehnice mai avansate (ex: configurări, parametrizări) și, în același timp identifică eventuale defecte ale soluției.

Serviciile de suport tehnic pot fi accesibile prin telefon, mail sau, cel mai indicat, prin soluții online de tip help-desk integrate în aplicație. Recomandăm să solicitați ca acest serviciu să fie disponibil în timpul orelor dumneavoastră uzuale de activitate și să specificați timpii maximi de răspuns doriți.

Suportul tehnic poate fi încadrat în 5 niveluri:

**Nivel 0** – suportul tehnic de care beneficiază utilizatorii fără a contacta un serviciu dedicat. Acesta poate fi sub formă de manuale de utilizare, tutoriale, sisteme tip knowledgebase, forum-uri, liste de probleme cunoscute și modalități de ocolire etc. Chiar dacă acest tip de suport nu reprezintă un serviciu propriu-zis, este foarte recomandat să introduceți astfel de materiale pentru a reduce considerabil costurile de suport tehnic.

**Nivel 1** – reprezintă prima linie de suport adresată utilizatorilor finali ai soluției. În general aici sunt rezolvate problemele de bază de utilizare a soluției. De asemenea, la acest nivel, sunt identificate eventualele incidente care sunt escaladate la nivelurile superioare de suport (ex: defecte).

În cazul în care instituția dumneavoastră poate asigura serviciile de suport de nivel 1 adresat utilizatorilor finali, este o variantă foarte recomandată care poate reduce semnificativ costurile de suport.

**Nivel 2** – nivel la care sunt adresate probleme tehnice mai avansate (ex: probleme de configurare și parametrizare a soluției, confirmarea defectelor) care însă nu implică neapărat intervenții la nivelul codului sursă. Acest nivel de suport este asigurat în general de persoane care cunosc bine soluția și, preferabil, dețin un nivel bun de cunoștințe tehnice. Și în acest

caz, pentru reducerea costurilor, atunci când este posibil, nivelul 2 poate fi asigurat de către personalul beneficiarului (ex: departamentul IT).

**Nivel 3** – reprezintă cel mai avansat nivel de suport tehnic. În general aceste servicii sunt asigurate de echipa tehnică de implementare a furnizorului. La acest nivel sunt reproduse și rezolvate eventualele defecte identificate la nivelul 2.

**Nivel 4** – la ultimul nivel sunt serviciile de suport asigurate de furnizorii hardware și software de bază. Acest nivel asigură rezolvarea unor defecte identificate în infrastructura hardware/software care depășesc capacitatea furnizorului de rezolvare (ex: defecte ale sistemelor de bază de date, ale browserelor internet sau ale sistemelor de operare).

### 3.10.3 Garanție

Serviciile asigurate în perioada de garanție fac parte, de asemenea, dintre serviciile post implementare. În perioada de garanție furnizorul va repara pe cheltuială proprie eventualele defecte identificate de suportul tehnic (level 2-3).

### 3.10.4 Servicii de intervenție

Acestea sunt serviciile prin care furnizorul garantează timpi de intervenție pentru identificarea și rezolvarea de probleme ce nu permit utilizarea în parametri normali a soluției. Este important ca aceste servicii să acopere perioadele de utilizare intensă ale soluției. Din rațiuni financiare, nu este recomandată achiziționarea de pachete de tip 24x7 dacă soluția nu este utilizată permanent.

### 3.10.5 Servicii de monitorizare și prevenție

Aceste servicii presupun monitorizarea recurentă a funcționării soluției cu scopul evitării unor probleme de funcționare în viitor. Acestea pot include monitorizarea log-urilor soluției, monitorizarea parametrilor de funcționare (utilizare procesor, spațiu alocat, memorie utilizată etc), optimizarea indecșilor folosiți de bazele de date, executarea periodică a diverselor proceduri critice (ex: disaster recovery) sau monitorizarea performanței aplicației.

### 3.10.6 Instruire

Instruirea personalului beneficiarului se va face atât la începerea exploatării soluției software, cât și periodic sau la apariția de personal nou în echipa beneficiarului. Nevoile de instruire diferă de la beneficiar la beneficiar, de la proiect la proiect, însă ignorarea lor va reduce dramatic șansele de succes ale soluției. Recomandăm lectura unui studiu realizat în 2011 (Cushing Anderson, IDC, "Impact of Training on Project Success") ce a arătat că proiectele ce au alocat peste 6% din buget pentru instruire au avut o rată de succes mult mai mare față de cele ce au alocat sub 3%.

### 3.10.7 Servicii de modificare/extensie

Autoritatea contractantă poate să solicite furnizorului alocarea unui buget de efort lunar sau anual care să acopere mici ajustări și modificări ale soluției. Acestea pot fi de obicei incluse sub formă de abonament periodic (lunar sau anual), și asigură disponibilitatea specialiștilor furnizorului, în limita orelor incluse în abonament, pentru implementarea de mici ajustări sau modificări ale soluției, suficient de mici cât să nu justifice un proces nou de achiziție (ajustări pentru adaptare la modificările legislative, mici modificări în fluxurile instituției etc). În lipsa contractării acestor servicii, furnizorul nu va fi obligat să rezolve astfel de cerințe în timp util pentru dumneavoastră.



### 3.10.8 Parametrii de performanță ai serviciilor (SLA – Service Level Agreement)

În formularea cerințelor privind serviciile post implementare este importantă definirea parametrilor de performanță ai acestor servicii pe care furnizorul trebuie să îi respecte. Câțiva parametri relevanți sunt enumerați în cele ce urmează.

**Disponibilitate suport tehnic** – perioada în care serviciile de suport tehnic sunt disponibile sau, mai exact, în care echipa de suport tehnic este disponibilă (ex: 24 x 7 – disponibilitate permanentă 24 de ore în 7 zile, 8x5 disponibilitate în timpul programului normal de lucru săptămânal 8 ore/zi etc.).

**Timp de răspuns** – timpul de răspuns la incidentele de suport tehnic sesizate. Acesta poate să varieze în funcție de gravitatea incidentelor. Timpul de răspuns nu asigură neapărat și o soluție a incidentului reclamat.

**Timp de soluționare temporară (workaround)** – timpul în care furnizorul trebuie să asigure o soluție temporară incidentului reclamat. Și acest timp poate varia în funcție de gravitatea incidentului sau defectului.

**Timp de soluționare permanentă** – timpul în care furnizorul trebuie să asigure o soluție permanentă/finală incidentului reclamat.

**Timp de prezența on-site** – timpul în care furnizorul trebuie să asigure prezența specialiștilor proprii on-site pentru incidente critice.

**Numar de incidente on-site** – numărul de solicitări de prezență on-site la care furnizorul trebuie să răspundă lunar sau anual.

**Timp de alocare** – timpul în care furnizorul trebuie să aloce specialiști pentru solicitările de modificare ale beneficiarului dacă astfel de servicii au fost contractate.

**Disponibilitate de efort garantată** – efortul pe care furnizorul trebuie să îl asigure pentru cerințele de modificare ale beneficiarului dacă astfel de servicii au fost contractate.

## 3.11 Structura financiară

### *“Cât va costa proiectul?”, “Cum vor fi eșalonate plățile?”, “Există costuri ascunse?”*

Structura financiară prevăzută în caietul de sarcini trebuie să cuprindă planul financiar al proiectului. În principal acesta cuprinde structura plăților ce urmează să fie efectuate și condițiile în care aceste plăți vor fi făcute.

Recomandăm ca planul de plăți să fie corelat cu livrabilele importante ce urmează să fie livrate de furnizor și să corespundă efortului depus pentru realizarea acestora. Evident, livrabilele respective vor fi supuse unui proces de acceptanță intermediar iar plățile vor fi condiționate de succesul procesului de acceptanță.

În structura financiară a proiectului trebuie să țineți cont și de faptul că firmele active în industria software au costuri mari de operare recurente, în special cu personalul (costuri salariale lunare), care nu pot fi susținute pe termen lung fără finanțare. Evident, un furnizor aflat în dificultate financiară nu este o situație de dorit. Astfel, în structurarea plăților, este recomandat să găsiți un echilibru corect între interesele instituției dumneavoastră și presiunea financiară pe care o va suporta furnizorul. Acest aspect este cu atât mai important pentru proiectele care cuprind hardware sau licențe care pot avea costuri considerabile pe care furnizorul nu le poate finanța pe perioade lungi.

O recomandare fermă în privința structurii financiare este dificil de oferit însă, în funcție de



complexitatea și durata proiectului, sugerăm ca plățile programate să nu fie mai frecvente decât o plată lunar, și nu mai rare decât o plată la trei luni, excluzând eventualul avans sau plata aferentă livrării finale. Un număr foarte mare de plăți poate presupune și un efort suplimentar considerabil pentru validarea livrărilor, în timp ce un număr prea mic de plăți poate presupune riscuri privind progresul și rezultatul final al proiectului, precum și o posibilă presiune financiară pe furnizor.

În cazul în care un avans este prevăzut contractual, acesta poate să acopere eventual parțial efortul inițial al furnizorului până la prima plată programată, cu respectarea prevederilor legale cu privire la efectuarea de plăți în avans din fonduri publice. Dacă proiectul presupune și achiziții de hardware și licențe software, avansul poate fi mai mare pentru a acoperi cel puțin parțial costurile furnizorului.

Pentru plățile ulterioare acceptanței finale este recomandat să rezervați un procent relevant din valoarea proiectului. De asemenea, pentru garanția de bună execuție, trebuie rezervat un procent din valoarea proiectului. În multe cazuri garanția de bună execuție este constituită de furnizor la începutul proiectului (ex: scrisoare de garanție, transfer) și restituită furnizorului la finalul perioadei de garanție. În cazul în care garanția este asigurată pe o perioadă mare, există posibilitatea restituirii parțiale pe parcurs.

### **Alte costuri**

Orice soluție software presupune și costuri viitoare pentru mentenanță, operare, dezvoltări ulterioare, upgrade-uri. Pe cât posibil trebuie să vă asigurați că aceste costuri pot fi ținute sub control și nu generează o situație de captivitate pentru instituția dumneavoastră.

În cazul în care contractul presupune servicii pe care furnizorul trebuie să le ofere în viitor, care eventual nu sunt prinse în contractul inițial (ex: dezvoltări ulterioare, mentenanță) este indicat să agreeți costul acestor servicii, sub forma de tarif orar/zilnic, precum și condițiile în care acest preț poate varia (ex: indexare anuală cu indexul inflației sau cu un procent maxim).

În cazul soluțiilor software bazate pe produse existente, upgrade-urile de software (versiuni minore, majore) pot presupune costuri suplimentare și servicii conexe (actualizare, testare, adaptare configurări etc). Caietul de sarcini trebuie să prevadă dacă aceste versiuni noi trebuie asigurate gratuit de furnizor în prețul inițial al contractului pentru o anumită perioadă sau vor fi furnizate contra cost. În cazul variantei contra cost, aceste costurile trebuie să fie predictibile, în timp ce prețul serviciilor conexe trebuie agreeat în prealabil (tarif orar/zilnic pe tip de resurse).

### **Estimarea costului total**

În cadrul procedurilor de achiziție în care criteriul de atribuire este cel mai bun raport calitate-cost, propunerea financiară a furnizorului trebuie să includă și informațiile necesare evaluării de cost sau chiar o evaluare de cost realizată de furnizor ce va trebui validată de autoritatea contractantă.

## 4. CRITERII DE ATRIBUIRE

Legislația în vigoare prevede ca și criteriile aplicabile în cadrul achizițiilor publice următoarele:

- Criterii de calificare și selecție;
- Criterii de atribuire

În aceasta primă versiune, ghidul își propune să abordeze strict problematica criteriilor de atribuire, fără a acoperi criteriile de calificare și criteriile de selecție. Această temă ar putea fi eventual abordată în cadrul unei versiuni ulterioare a ghidului.

### 4.1 Cadrul legislativ

În vederea stabilirii criteriilor de atribuire trebuie aplicate prevederile Legii nr. 98/2016 privind contractele de achiziții publice precum și a normelor metodologice aprobate prin HG. 395/2016.

Chiar dacă în acest ghid încercăm să rezumăm cadrul legal, este necesar ca la stabilirea criteriilor de atribuire **să respectați următoarele prevederi legislative:**

- Capitolul IV, Secțiunea a 7-a – Criterii de atribuire, Art. 187-192 din Legea nr. 98/2016
- Capitolul al II-lea, Secțiunea a 4-a, Paragraful 3 – Stabilirea criteriului de atribuire, Art. 32-34 din HG nr. 395/2016

Legea nr. 98/2016 este disponibilă pe site-ul ANAP la adresa:

[http://anap.gov.ro/web/wp-content/uploads/2016/05/L98\\_2016.pdf](http://anap.gov.ro/web/wp-content/uploads/2016/05/L98_2016.pdf)

Hotărârea Guvernului nr 395/2016 este de asemenea disponibilă pe site-ul ANAP la adresa:

[http://anap.gov.ro/web/wp-content/uploads/2016/06/HG395\\_16.pdf](http://anap.gov.ro/web/wp-content/uploads/2016/06/HG395_16.pdf)

Conform art 187 alin (3) din lege, autoritatea contractantă are dreptul de a aplica unul din următoarele criterii de atribuire:

- a. Prețul cel mai scăzut;
- b. Costul cel mai scăzut;
- c. Cel mai bun raport calitate-preț;
- d. Cel mai bun raport calitate-cost.

Conform art. 187 alin (8) **autoritatea contractantă nu va utiliza costul cel mai scăzut/prețul cel mai scăzut** drept criteriu de atribuire în cazul anumitor categorii de contracte de achiziție publică/acorduri-cadru de lucrări sau de servicii care au ca obiect servicii intelectuale și care presupun activități cu nivel de complexitate ridicat.

Mai mult decât atât, în conformitate cu normele metodologice – HG nr. 395/2016 art. 32, alin. (6), **singurele criterii aplicabile în cazul proiectelor de complexitate ridicată sunt “Cel mai bun raport calitate-preț” sau “Cel mai bun raport calitate-cost”**

Având în vedere că implementarea unei soluții software presupune în mod evident activități cu nivel de complexitate ridicat, singurele criterii aplicabile legal și pe care le recomandăm pentru achiziția de soluții software sunt “Cel mai bun raport calitate-preț” sau “Cel mai bun raport calitate-cost”.

Este important de subliniat că prevederile legale menționate mai sus se referă strict la servicii intelectuale.

În cazul în care achiziția dumneavoastră presupune și achiziții de hardware sau produse software comercializate de mai mulți agenți economici (ex: sisteme de operare, sisteme de baze de date sau alte produse software ce pot fi achiziționate separat), achiziția poate fi separată în loturi cu criterii diferite de atribuire sau în achiziții distincte, iar în situația în care achiziția privește

produse cu un grad ridicat de standardizare, atunci se poate utiliza și criteriului de atribuire "prețul cel mai scăzut" / "costul cel mai scăzut".

Practica separării în loturi este reglementată de Art. 141 din Legea nr. 98/2016.

## 4.2 Metodologie de evaluare

În conformitate cu art 190 din Legea nr. 98/2016, autoritatea contractantă stabilește factorii de evaluare și ponderea acestora în evaluarea finală. De asemenea, conform Art. 32 alin.(2) din HG nr.395/2016, factorii de evaluare precum și algoritmul de punctare trebuie precizate în mod clar și detaliat în cadrul documentației de atribuire.

În cazul în care stabilirea unor ponderi nu este posibilă din motive obiective, autoritatea contractantă are opțiunea de a indica factorii de evaluare în ordinea importanței (conform art 190 (3) din Legea nr 98/2016).

Pentru cazurile în care se pot stabili ponderi obiective, în funcție de numărul de criterii pe care doriți să le folosiți pentru atribuire, propunem două metodologii de evaluare:

- Evaluare cu un nivel de criterii, aplicabilă pentru un număr limitat de criterii (sub 10);
- Evaluare cu două niveluri de criterii, aplicabilă pentru achiziții complexe unde sunt folosite multe criterii de evaluare.

Diferența dintre cele două metode este strict organizatorică, din punct de vedere matematic cele două metode sunt perfect echivalente.

### 4.2.1 Evaluare cu un nivel de criterii

Metoda presupune alegerea unui set de criterii direct evaluabile și stabilirea unor ponderi procentuale pentru fiecare. Suma ponderilor asociate criteriilor trebuie să fie 100%.

Grila de evaluare prevăzută în caietul de sarcini ar putea fi structurată astfel (acesta este doar un exemplu de calcul pentru a ilustra structura grilei și modul de evaluare):

<b>Criteriu</b>	<b>Pondere</b>
Prețul achiziției	25%
Calitatea tehnică a arhitecturii propuse	10%
Calitatea și detalierea soluției în privința cerințelor funcționale	30%
Calitatea și detalierea soluției în privința cerințelor nefuncționale	30%
Experiența personalului alocat	5%

Rezultatul evaluării folosind grila de mai sus ar fi structurat astfel:

Criteriu	Pondere	Evaluare Furnizor 1	Punctaj ponderat	Evaluare Furnizor 2	Punctaj ponderat
Prețul achiziției	25%	7	1.75	10	2.5
Calitatea tehnică a arhitecturii propuse	10%	5	0.5	7	0.7
Calitatea și detalierea soluției în privința cerințelor funcționale	30%	8	2.4	10	3
Calitatea și detalierea soluției în privința cerințelor nefuncționale	30%	6	1.8	9	2.7
Experiența personalului alocat	5%	4	0.2	7	0.35
<b>Punctaj total</b>			<b>6.65</b>		<b>9.25</b>

Punctajul ponderat pentru fiecare criteriu este rezultatul înmulțirii dintre punctajul evaluat (spre exemplu pe o scară de la 1 la 10, dar se poate folosi foarte bine orice scară) și ponderea stabilită prin grilă. Punctajul total este suma punctajului ponderat.

#### 4.2.2 Evaluare cu două niveluri de criterii

În cazul soluțiilor complexe în care veți folosi pentru atribuire un număr mare de criterii, recomandăm separarea acestora pe două niveluri:

- Categoriile/criteriile principale cu ponderi procentuale asociate care influențează punctajul final;
- Subcategoriile asociate criteriilor principale cu ponderi asociate în cadrul categoriei care afectează punctajul categoriei.

Suma ponderilor asociate categoriilor principale trebuie să fie 100%. De asemenea, pentru fiecare criteriu principal, suma ponderilor subcategoriilor trebuie să fie 100%.

Avantajul acestei metode constă în faptul că permite definirea unor categorii principale asigurând astfel o structură mai bună a grilei de evaluare. Evident, această structură poate fi convertită la o structură cu un nivel sau invers, printr-o simplă conversie aritmetică, neafectând în vreun fel rezultatul final.

Exemplu de structură a grilei cu două niveluri.

<b>Criteriu</b>	<b>Pondere generală</b>	<b>Pondere în cadrul categoriei</b>
<b>Cost/Preț</b>	<b>25%</b>	
<b>Calitatea propunerii</b>	<b>35%</b>	
Gradul de detaliere		<b>10%</b>
Calitatea tehnică a arhitecturii		<b>10%</b>
Calitatea și detalierea soluției în privința cerințelor funcționale		<b>40%</b>
Calitatea și detalierea soluției în privința cerințelor nefuncționale		<b>40%</b>
<b>Calitatea planului de proiect</b>	<b>10%</b>	
Gradul de detaliere		<b>10%</b>
Corelarea cerințelor și livrabilelor		<b>70%</b>
Realismul planului propus		<b>20%</b>
<b>Procesul de dezvoltare</b>	<b>10%</b>	
Subcriteriu		<b>100%</b>
<b>Personalul alocat</b>	<b>10%</b>	
Subcriteriu		<b>100%</b>
<b>Servicii post implementare</b>	<b>10%</b>	
Subcriteriu		<b>100%</b>
<b>Personalul alocat</b>	<b>10%</b>	
Subcriteriu		<b>100%</b>
<b>Servicii post implementare</b>	<b>10%</b>	
Subcriteriu		<b>100%</b>

Mecanismul de calcul este similar cu cel pentru evaluarea cu un singur nivel, cu diferența că are doi pași:

- Calcularea punctajului pentru fiecare criteriu principal folosind ponderile subcriteriilor;
- Calculare punctajului total folosind ponderile asociate criteriilor principale.

Aplicând grila de mai sus, un exemplu de evaluare ar fi urmatorul:

Criteriu	Pondere generală	Pondere în cadrul categoriei	Evaluare	Punctaj ponderat subcriterii	Punctaj ponderat criterii principale
<b>Cost/Preț</b>	<b>25%</b>		<b>6</b>	<b>6</b>	<b>1.50</b>
<b>Calitatea propunerii</b>	<b>35%</b>			<b>7.60</b>	<b>2.66</b>
Gradul de detaliere		<b>10%</b>	<b>5</b>	<b>0.50</b>	
Calitatea tehnică a arhitecturii		<b>10%</b>	<b>7</b>	<b>0.70</b>	
Calitatea și detalierea soluției în privința cerințelor funcționale		<b>40%</b>	<b>10</b>	<b>4.00</b>	
Calitatea și detalierea soluției în privința cerințelor nefuncționale		<b>40%</b>	<b>6</b>	<b>2.40</b>	
<b>Calitatea planului de proiect</b>	<b>10%</b>			<b>7.60</b>	<b>0.76</b>
Gradul de detaliere		<b>10%</b>	<b>10</b>	<b>1.00</b>	
Corelarea cerințelor și livrabilelor		<b>70%</b>	<b>8</b>	<b>5.60</b>	
Realismul planului propus		<b>20%</b>	<b>5</b>	<b>1.00</b>	
<b>Procesul de dezvoltare</b>	<b>10%</b>			<b>6.00</b>	<b>0.6</b>
Subcriteriu		<b>100%</b>	<b>6</b>	<b>6.00</b>	
<b>Personalul alocat</b>	<b>10%</b>			<b>10.00</b>	<b>1</b>
Subcriteriu		<b>100%</b>	<b>10</b>	<b>10.00</b>	
<b>Servicii post implementare</b>	<b>10%</b>			<b>8.00</b>	<b>0.8</b>
Subcriteriu		<b>100%</b>	<b>8</b>	<b>8.00</b>	
<b>PUNCTAJ TOTAL</b>					<b>7.32</b>

Culorile sunt folosite în tabel pentru a arăta grupurile de date corelate.

## 4.3 Stabilirea criteriilor de evaluare

Criteriile de atribuire pe care le veți alege pentru evaluare țin foarte mult de specificul și complexitatea proiectului. În acest ghid nu ne propunem să stabilim o metodologie standard de evaluare, însă propunem un set de criterii pe care le puteți adapta cerințelor dumneavoastră specifice eventual adăugând criteriile proprii specifice proiectului.

În stabilirea criteriilor de atribuire trebuie ținut cont de prevederile legale, astfel criteriile utilizate trebuie să respecte următoarele condiții:

- Să fie alese în **legătură directă cu natura și obiectul contractului**, fără adăugarea unor criterii artificiale (a se avea în vedere art.188 alin.(1) din Legea nr.98/2016 și Art. 32 alin.(6) lit. a) din HG nr.395/2016);
- Să **reflecte avantajele** pe care autoritatea contractantă le obține;
- Se pot referi la **procesul folosit** pe perioada proiectului, cât și ulterior potrivit art 188 alin. (2) din Legea nr.98/2016;
- Să fie cât mai **obiective**, astfel încât autoritatea contractantă să nu aibă "o libertate de apreciere nelimitată", potrivit art. 189 alin.(1) din Legea nr.98/2016;
- Să asigure o **concurență reală** între ofertanți, potrivit art. 189 alin.(2) din Legea nr.98/2016.

Pentru a asigura un grad de obiectivitate cât mai mare este important ca fiecare criteriu utilizat să fie cât mai bine detaliat iar modul de evaluare/punzare al fiecărui criteriu să fie cât mai clar explicat în documentația de atribuire.

În funcție de specificul și complexitatea obiectului contractului ce urmează a fi atribuit, autoritatea contractantă alege dintre factorii de evaluare enumerați mai jos pe aceia pe care îi consideră relevanți pentru maximizarea beneficiilor obținute în urma derulării procedurii de atribuire, raportat la necesitățile sale. Pentru fiecare factor de evaluare ce urmează a fi utilizat, autoritatea contractantă precizează în documentația de atribuire metoda de acordare a punctajului și algoritmul de calcul necesar aplicării criteriului de atribuire ales.

Factorii de evaluare menționați mai jos sunt:

- fie de natură "cantitativă", fapt ce presupune că punctajul se acordă în urma aplicării unei formule aritmetice/matematică (de ex. în cazul în care se punctează o specificație tehnică cum ar fi viteza de răspuns a sistemului, sau viteza procesorului, soluția tehnică cea mai rapidă va primi punctaj maxim, iar celelalte soluții vor fi punctate proporțional în funcție de nivelul parametrului în cauză prezentat);
- fie de natură "calitativă", situație în care punctajul se acordă în funcție de aprecierea obiectivă a comisiei de evaluare, în corelație cu aspectele descrise în enumerarea de mai sus, departajarea realizându-se pe clase de punctaj în funcție claritatea/acuratețea, coerența, relevanța și altele asemenea criterii aferente propunerilor tehnice prezentate de ofertanți.

## 4.4 Criteriile de preț și cost

Pentru diferențierea ofertelor în funcție de cheltuielile pe care le implică achiziția de soluții software complexe, autoritățile contractante au dreptul de a utiliza ca și criterii de atribuire, cel mai bun raport calitate-preț, sau cel mai bun raport calitate-cost.

Criteriile "prețul cel mai scăzut" și "costul cel mai scăzut" sunt aplicabile doar pentru achiziția produselor cu un grad ridicat de standardizare (ex: echipamente hardware uzuale, software standard tip COTS).

În situația utilizării criteriului de atribuire cel mai bun raport calitate-preț sau cel mai bun raport calitate-cost, în alocarea ponderii pentru criteriul de preț/cost trebuie să țineți cont de faptul că, **în cazul contractelor ce presupun dezvoltare/proiectare sisteme informatice complexe, ponderea alocată factorului nu poate fi mai mare de 40%** (art. 32 alin. (6) din HG nr.395/2016).

În cazul criteriului "prețul cel mai scăzut" (care poate fi utilizat doar pentru achiziția produselor cu grad ridicat de standardizare), sau în situația criteriului de atribuire "cel mai bun raport calitate-preț", prețul se referă strict la oferta financiară a furnizorului și este ușor aplicabil.

În cazul criteriilor de atribuire "costul cel mai scăzut" și "cel mai bun raport calitate-cost", costul presupune însă o evaluare mai complexă, dar poate fi un criteriu mult mai relevant pe termen lung. Costul pe durata de viață a achiziției presupune determinarea costurilor totale pe care autoritatea le va suporta pe durata de operare/utilizare a soluției, suplimentar față de prețul de achiziției (respectiv determinarea TCO – Total Cost of Ownership).

Pentru evaluarea costului trebuie definită, în primul rând, durata de viață dorită a soluției. Pentru o analiză relevantă a costului recomandăm ca această perioadă să fie între 3 ani și 5 ani, fiind desigur posibile și niveluri mai mari, în funcție de specificul proiectului.

Având în vedere dinamica tehnologică, o perioadă medie de viață de 5 ani pentru o soluție software este rezonabilă. După o astfel de perioadă, presupunând că nu suferă transformări sau upgrade-uri, soluția intră într-o perioadă de îmbătrânire. Recomandăm ca orice soluție software mai veche de 5 ani să fie evaluată tehnologic pentru a identifica riscurile posibile (ex: incompatibilități cu sistemele actuale, tehnologie ieșită din uz, lipsa suportului din partea furnizorului etc.).

Pentru utilizarea criteriului de atribuire costul cel mai scăzut trebuie să definiți în mod clar în documentația de atribuire elementele de cost care trebuie avute în vedere de furnizor în elaborarea soluției. Astfel, conform art. 33 alin. (3) și alin. (4) din HG nr. 395/2016, autoritatea contractantă trebuie să includă în documentația de atribuire cel puțin următoarele informații:

- Condițiile, mediul și intensitatea de utilizare;
- Durata de utilizare anticipată;
- Durata de utilizare pentru calcularea costurilor;
- Elementele de cost evaluate;
- Rata de actualizare pentru calculul financiar;
- Modalitatea efectivă de realizare a calculului costului pe durata de viață;
- Condițiile contractuale de monitorizare a respectării elementelor de cost precum și consecințele nerespectării acestora.

Pentru evaluarea costului total, legea prevede determinarea costurilor anuale pentru fiecare an de utilizare pentru fiecare element de cost, în conformitate cu prevederile art. 33 alin. (2) – (4) din HG nr. 395/2016.

În cazul utilizării criteriului costul cel mai scăzut, este recomandată includerea în documentația de atribuire a unui "template" (foaie de calcul / spreadsheet) pe care furnizorii să îl poată utiliza pentru estimarea costului total. Această foaie de calcul poate, eventual, să fie solicitată ca parte a propunerii financiare și validată de autoritatea contractantă în cadrul procesului de achiziție.

#### 4.4.1 Elemente de cost pentru stabilirea costului total

Lista de mai jos este un punct de pornire pentru identificarea elementelor de cost pe care le puteți include în evaluarea costului total.

##### Actualizări software

În special în cazul soluțiilor bazate pe produse software existente, costul actualizărilor este foarte relevant.

##### Licențe software necesare

Costul pentru achiziția licențelor software necesare funcționării soluției și, eventual, costurile de actualizare sunt de asemenea importante. În această categorie intră sistemele de operare comerciale, sistemele de baze de date, soluții software de birou (exemplu: Office, Acrobat) și orice alte licențe ar putea fi necesare funcționării soluției.



### Costuri de administrare și operare

Orice soluție software presupune un efort din partea beneficiarului pentru administrare și operarea acesteia (administrare tehnică, monitorizare, suport tehnic de nivel 1, backup, administrarea incidentelor). Aceste costuri pot varia în funcție de soluția propusă.

### Costuri de extensie și modificare

Majoritatea soluțiilor suferă modificări ulterior implementării primei versiuni. În cazul în care aceste modificări vor fi realizate de același furnizor, este relevant costul estimat al acestora. Având în vedere că estimarea viitoarelor modificări este practic imposibilă pentru estimarea acestor costuri, se poate utiliza un efort estimat de ore anual înmulțit cu un tarif orar specificat de furnizor.

### Hardware servere

Cerințele hardware ale soluției propuse ar putea presupune investiții noi în hardware în momentul implementării inițiale sau ulterior, pe durata de viață a soluției.

### Hardware client

Echipamentele hardware tip client, respectiv sisteme desktop, laptop, tablete sau mobile ar putea presupune de asemenea costuri în funcție de caracteristicile tehnice ale soluțiilor propuse.

### Costuri de gazduire (cloud)

Costurile de găzduire sau de utilizare a serviciilor tip cloud în cazul soluțiilor cu componente cloud.

### Costuri de networking și comunicații

Costuri cu infrastructura de rețea și comunicații. Spre exemplu costul conexiunilor la internet, costuri de telefonie, costuri de mesagerie (ex: notificări SMS) etc.

### Instruire

Costurile de instruire suportate de instituție pe durata de viață. Pot fi evaluate inclusiv costuri de logistică (închiriere spații de instruire, transport etc) precum și costuri pentru pregătirea materialelor de instruire.

### Efort propriu

Costul aferent efortului depus de personalul autorității contractante pentru implementarea proiectului, în cazul în care acesta poate varia în funcție de soluția oferită de furnizor sau de procesul de implementare propus.

### Riscuri

Impactul financiar al materializării unor riscuri ar putea fi cuantificat. În acest sens trebuie identificate riscurile posibile pe durata de viață a soluției. Spre exemplu: riscuri legate de personal (relevante în cazul utilizării unor tehnologiilor exotice sau învechite), riscuri tehnologice (ex: finalizarea ciclului de viață), riscuri legale (ex: falimentul furnizorului), riscuri de securitate (ex: compromiterea datelor personale) etc. Disciplina de administrare a riscurilor (risk management) este una foarte complexă pe care nu avem pretenția că o putem acoperi în acest ghid.

## 4.5 Punctarea ofertelor în funcție de cost sau preț

Punctarea ofertelor în funcție de cost sau preț este evident relevantă doar în situația utilizării criteriilor "cel mai bun raport calitate-preț" și "cel mai bun raport calitate-cost", pentru celelalte criterii fiind suficientă ordonarea ofertelor.

Pentru punctarea ofertanților în funcție de preț sau cost se poate folosi o metodă simplă de transformare a prețului în punctaj. Punctajul pentru factorul de evaluare "prețul ofertei" sau "costul ofertei" se acordă astfel:

- a. pentru cel mai scăzut dintre prețurile/costurile ofertelor se acordă punctajul maxim alocat factorului de evaluare respectiv (ex: 10 puncte)

b. pentru celelalte oferte punctajul se acordă astfel:

**Punctaj = (preț minim/preț oferta) x punctaj maxim alocat.**

#### Exemplu de calcul:

Presupune cazul în care punctarea ofertelor se va face într-un interval de punctaj de la 1 la 10 iar ofertele sunt în intervalul de la 1.000 lei (cea mai mica ofertă) la 20.000 lei (cea mai mare ofertă).

Oferta minimă de preț (cu valoarea de 1.000 lei) va avea punctajul maxim, respectiv 10 puncte. Celelalte oferte vor fi punctate astfel:

- Oferta de 2.000 lei va avea punctajul =  $1.000/2.000 \times 10 = 5$  puncte
- Oferta de 5.000 lei va avea punctajul =  $1.000/5.000 \times 10 = 2$  puncte
- Oferta de 10.000 lei va avea punctajul =  $1.000/10.000 \times 10 = 1$  punct
- Oferta de 20.000 lei va avea punctajul =  $1.000/20.000 \times 10 = 0,5$  puncte
- S.a.m.d.

## 4.6 Criterii tehnice

Criteriile tehnice pot fi avute în vedere doar în cazul utilizării criteriilor de atribuire "cel mai bun raport calitate-preț" sau "cel mai bun raport calitate-cost".

În cele ce urmează propunem o serie de criterii pe care le puteți adapta situației dumneavoastră și care vor fi utilizate pe lângă componenta/componentele de preț/cost. Acestea sunt grupate în următoarele categorii:

- Calitatea propunerii tehnice;
- Personalul alocat;
- Calitatea și relevanța planului de proiect;
- Procesul de dezvoltare;
- Servicii post implementare.

### 4.6.1 Calitatea propunerii tehnice

#### Gradul de detaliere al propunerii

Propunerile tehnice ale furnizorilor trebuie să adreseze în detaliu modul de îndeplinire, în procesul de implementare a viitorului contract, toate cerințele formulate în documentația de atribuire. O propunere tehnică detaliată arată interesul potențialului furnizor precum și înțelegerea cerințelor formulate.

#### Calitatea tehnică a soluției propuse

Utilizând acest criteriu, puteți evalua relevanța arhitecturală a soluției propuse, modularitatea arhitecturii, tehnologiile folosite, adaptarea la infrastructura tehnică a beneficiarului, corespondența cerințelor funcționale cu elementele soluției (sistem, subsisteme, module), gradul în care tehnologia propusă răspunde cerințelor nefuncționale etc.

#### Înțelegerea și acoperirea cerințelor funcționale

Propunerile furnizorilor trebuie să răspundă în detaliu tuturor cerințelor funcționale formulate. Pe lângă acoperirea integrală a cerințelor, răspunsul furnizorilor trebuie să arate și înțelegerea cerințelor, a nivelului de complexitate și să ofere soluții valide și eficiente acestor cerințe.

#### Înțelegerea și acoperirea cerințelor nefuncționale

Furnizorii trebuie de asemenea să răspundă integral și detaliat cerințelor nefuncționale formulate. Soluțiile propuse pentru a adresa unele cerințele nefuncționale, pot avea uneori un impact profund în arhitectura soluției sau pot influența alte cerințe (funcționale sau nefuncționale). În astfel de cazuri trebuie validat că impactul a fost corect evaluat de furnizor.

### **Eficientizări tehnice și de proces**

Criteriul urmărește încurajarea furnizorilor să propună soluții eficiente din punct de vedere tehnic cum ar fi reducerea necesarului de hardware și software, arhitecturi extensibile, simple și modulare. De asemenea furnizorii ar putea propune și eficientizări de proces cu beneficii pentru autoritatea contractantă.

### **Caracteristici estetice și ușurința în utilizare (usability)**

Aspectul grafic al soluțiilor propuse poate fi un criteriu de evaluare, în special în cazul soluțiilor cu componente adresate publicului (ex: soluții online). ușurința în utilizare este relevantă pentru orice interfață utilizator și astfel poate fi folosită în cadrul criteriilor de atribuire.

### **Angajamente și cerințe suplimentare**

Criteriul urmărește încurajarea și punctarea suplimentară a furnizorilor care își asumă angajamente suplimentare față de cerințele minime prevăzute în caietul de sarcini. Conform Art 32 alin (9) lit b): "cuantumul valoric al avantajelor de natură financiară pe care ofertanții le pot oferi prin asumarea unor angajamente suplimentare în raport cu cerințele minime prevăzute în caietul de sarcini sau documentul descriptiv", aceste angajamente suplimentare trebuie evaluate valoric.

## **4.6.2 Personalul alocat**

Este important de remarcat că gradul de pregătire a personalului alocat pentru implementarea contractului (denumit și personal cheie) este un criteriu care poate fi folosit doar în cadrul criteriilor de atribuire și nu poate fi folosit ca și criteriu de calificare și selecție (Art. 32 alin (4) din HG nr. 395/2016).

### **Experiența relevantă**

Criteriu care vizează experiența personalului cheie în realizarea unor activități de tipul și natura celor ce urmează a fi îndeplinite în viitorul contract în cadrul unor proiecte similare.

### **Pregătire tehnică**

Criteriu care vizează pregătirea personalului alocat în utilizarea tehnologiilor relevante în proiect.

### **Structura echipei de proiect**

Evaluarea structurii echipei de proiect propuse de furnizor, respectiv dimensiunea adecvată a echipei, rolurile definite în cadrul echipei, definirea unor responsabilități clare pentru fiecare rol, acoperirea specializărilor necesare în proiect, corelarea cu planul de proiect și efortul/costul propus.

### **Certificări tehnice**

Certificările tehnice deținute de o persoană pot fi un indicator al competențelor acesteia, însă lipsa unor certificări nu presupune implicit lipsa competențelor. Astfel, utilizarea certificărilor tehnice în cadrul criteriilor de atribuire trebuie făcută cu justă măsură și prudență. Introducerea unor criterii de punctaj artificiale poate influența considerabil concurența între furnizori, în special în cazul punctajului suplimentar acordat pentru certificări tehnice rare, cumul de certificări sau certificări irelevante în proiect.

În cazul în care veți folosi ca și criteriu de atribuire certificările tehnice, recomandăm ca punctajul acordat acestui criteriu să nu fie determinant în economia generală a punctajului.

## **4.6.3 Calitatea planului de proiect**

### **Gradul de detaliere al planului**

Un plan bine realizat și realist este un prim pas necesar pentru orice proiect reușit. Astfel, detalierea planului de realizare propus de furnizor este un criteriu de evaluare relevant.

### **Corelarea cerințelor și a livrabilelor solicitate cu planul propus**

Criteriul urmărește gradul de corelare dintre planul propus și cerințele și livrabilele solicitate în proiect. Planul de proiect trebuie să clarifice termenii la care vor fi livrate diferitele livrabile solicitate precum și fazele/activitățile în care vor fi implementate cerințele funcționale.

### **Acoperirea tuturor activităților relevante**

Criteriul urmărește gradul în care planul de proiect acoperă toate activitățile necesare pentru finalizarea proiectului precum și ponderea acestora în cadrul proiectului. Activitățile includ: analiză, testare, proiect management, dezvoltare, instruire, acceptanță, elaborare documentații etc.

### **Corelarea planului cu prețul proiectului**

Măsura în care planul de proiect este corelat cu bugetul proiectului. În măsura în care planul de proiect include și informații despre efortul estimat (ceea ce este recomandat) acestea pot fi folosite pentru a determina acuratețea planului și a estimării.

### **Realismul planului propus**

Măsura în care planul propus este realist. Criteriul ar putea urmări subestimarea sau supraestimarea considerabilă a anumitor activități din proiect.

### **Durata proiectului sau termenul de livrare**

Pentru criteriul durată se poate folosi un algoritm identic cu cel propus pentru punctarea ofertelor în funcție de cost sau preț, însă în loc de bugetul proiectului se va folosi numărul de zile de la startul proiectului până la acceptanță sau punerea în producție.

## **4.6.4 Procesul de dezvoltare**

Procesul utilizat de furnizorul soluției și adaptarea procesului la complexitatea proiectului și cerințele beneficiarului sunt aspecte vitale pentru succesul proiectului.

### **Claritatea procesului**

Criteriul vizează claritatea cu care furnizorul definește procesul de dezvoltare/implementare urmărit, definirea fazelor proiectului, modul în care diversele livrabile sunt realizate și transformate în cadrul procesului, responsabilitățile rolurilor în fiecare fază, șabloanele de documentație utilizate.

### **Relevanța procesului**

Criteriul urmărește relevanța procesului propus în contextul proiectului și adaptarea procesului propus la complexitatea proiectului și cerințele de proces ale autorității contractate. În cazul soluțiilor de complexitate mică-medie, un proces mai curând informal poate fi mai eficient, în timp ce soluții de complexitate mare necesită un proces mai formal.

### **Instrumente de proces și colaborare**

Criteriul evaluează instrumentele folosite de furnizor pe parcursul proiectului. Acestea pot include: soluții de colaborare online, instrumente de planificare (proiect management), aplicații de prototipare UI, soluții de organizare a cerințelor, aplicații de diagramare, soluții de evidență a defectelor și incidentelor, aplicații de testare, aplicații de testare a performanței, aplicații pentru elaborarea manualelor de utilizare etc.

### **Project management**

Criteriul vizează modul în care furnizorul va asigura pe întreaga perioadă a proiectului disciplina de proiect management, precum și identificarea clară a responsabilităților și activităților pe care rolul sau echipa de proiect management urmează să le realizeze. Printre aceste responsabilități se numără: monitorizarea progresului, identificarea și comunicarea eventualelor întârzieri, adresarea eventualelor blocaje și neclarități precum și administrarea riscurilor.

### **Raportarea progresului**

Monitorizarea și raportare progresului este o atribuție a rolului sau echipei de proiect management însă, având în vedere importanța acestei activități, o puteți lua în considerare ca un criteriu de evaluare distinct. Acest criteriu urmărește frecvența rapoartelor de stadiu, structura și conținutul acestora, demo-uri intermediare ale soluției și comunicarea întârzierilor.

#### **Administrarea situațiilor blocante (Issue Management)**

O altă atribuție foarte importantă a rolului de proiect management este administrarea și adresarea situațiilor blocante inerente în orice proiect. Un criteriu distinct în acest sens trebuie să vizeze modul în care furnizorul va identifica, administra, adresa și eventual escalada blocajele apărute în proiect.

#### **Administrarea cerințelor de modificare (Change Management)**

În mod natural pe parcursul oricărui proiect apar modificări ale cerințelor inițiale (eliminare, adăugare, modificare, simplificarea sau creșterea în complexitate). Un potențial criteriu de evaluare, relevant în special pentru proiecte de complexitate mare, este procesul prin care furnizorul va identifica, administra, negocia și încorporează sau exclude din scop astfel de modificări.

#### **Testare**

Criteriul urmărește evaluarea procesului de testare, respectiv structura echipei de testare și adecvarea acesteia la nevoile proiectului, tipurile de teste realizate (unitare, funcționale, de regresie, automate, de utilizabilitate, de compatibilitate, de performanță, de securitate, distructive, de acceptanță etc), planificarea testelor, structura și evidența scenariilor de test, evidența defectelor, prioritizarea defectelor sau instrumentele folosite pentru evidența defectelor, instrumente folosite pentru testare.

#### **Dezvoltare**

Criteriul urmărește evaluarea procesului de dezvoltare utilizat de furnizor, respectiv utilizarea obligatorie de soluții de source control, politica de checkin-uri, reguli pentru scriere a comentariilor, politica privind review-urile de cod, utilizarea de standarde de scriere a codului, utilizarea de instrumente automate de analiză a codului, utilizarea de instrumente de productivitate și eliminare a erorilor în scrierea codului (refactoring, code smell etc.), utilizarea standardelor deschise, unde este posibil, sau utilizarea de instrumente pentru monitorizarea performanței în timpul dezvoltării.

### **4.6.5 Servicii post implementare**

#### **Instruire**

Criteriul poate evalua tipurile de instruire oferite de furnizor (utilizatori finali, administrare și operare, dezvoltare etc), resurse disponibile pentru instruire (manuale, cursuri, tutoriale text, tutoriale video) precum și suportul logistic oferit de furnizor pentru instruire.

#### **Garanție**

Criteriul va evalua perioada de garanție oferită de furnizor, eventual suplimentară față de garanția solicitată în documentația de atribuire, defectele acoperite de garanție, limitele garanției sau responsabilitatea furnizorului în cazul defectelor.

#### **Servicii de mentenanță și suport**

Criteriul urmărește evaluarea serviciilor de mentenanță și suport tehnic oferite de furnizor, programul de lucru și disponibilitatea serviciilor, disponibilitatea unei persoane sau echipe dedicate pentru suport, instrumentele puse la dispoziție pentru evidența incidentelor sau defectelor, nivelul de suport (nivel 1 – probleme de bază, nivel 2 – suport tehnic mai avansat, nivel 3 – suport tehnic al echipei de dezvoltare, nivel 4 – escaladare la furnizorii de tehnologie), disponibilitatea furnizorului pentru prezența on-site pentru anumite tipuri de incidente, tehnologii de suport la distanță, disponibilitatea geografică a furnizorului în cazul în care este relevantă, monitorizare regulată preventivă și frecvența monitorizării, precum și evaluarea parametrilor de performanță ai serviciilor de mentenanță și suport la care se angajează furnizorul prin contract (SLA – Service Level Agreement).

### **Actualizări**

Acest criteriu este în special relevant în cazul soluțiilor bazate pe produse software existente eventual personalizate. În astfel de situații actualizarea în timp a produsului de bază este foarte importantă. Criteriul va evalua tipul de actualizări oferite gratuit de furnizor (versiuni minore, majore, fix-uri) și perioada pe care acestea sunt oferite.

## 5. CONCLUZII

Aceasta primă versiune a ghidului acoperă, prin recomandări practice, două componente ale procesului de achiziție a soluțiilor software:

- Realizarea documentației tehnice de atribuire;
- Procesul de evaluare folosind criteriile de atribuire "cel mai bun raport calitate-preț" și "cel mai bun raport calitate-cost".

Urmărind structura și recomandările din capitolul "3. Caietul de sarcini" veți putea elabora documentația tehnică structurând cerințele proiectului în cerințe funcționale și cerințe nefuncționale.

În capitolul "4. Criterii de atribuire" este explicat cadrul legal aplicabil, sunt propuse două metodologii posibile pentru evaluare și este de asemenea propus un set posibil de criterii de atribuire cu explicațiile aferente.

Având în vedere problematica foarte complexă abordată, ghidul nu și-a propus să impună un standard de realizare a documentației de atribuire sau un standard de evaluare, însă este un foarte bun punct de plecare în aceste demersuri.

Din acest motiv trebuie să insistăm pe necesitatea **adaptării recomandărilor din ghid la cerințele instituției** dumneavoastră, fără a prelua ad-litteram anumite recomandări sau exemple din ghid. Este foarte posibil ca anumite recomandări să nu fie necesare sau aplicabile în cazul dumneavoastră, la fel cum este posibil ca anumite cerințe pe care le aveți să nu fie acoperite de ghid.

În aceste concluzii vrem de asemenea să insistăm asupra **principiului necesității** în elaborarea documentației de atribuire, respectiv în formularea de cerințe cât mai clare care sunt necesare proiectului, evitând cerințe vagi, artificiale sau cerințe opționale. Recomandăm în special o atenție sporită în formularea cerințelor nefuncționale a căror supradimensionare față de necesitățile reale poate conduce la costuri exponențial mai mari.

În speranța că acest ghid va fi util în activitatea dumneavoastră, vă urăm succes în realizarea documentației și în derularea proiectelor dumneavoastră.

### Sugestii de îmbunătățire

Acest ghid se află la prima versiune și există intenția clară de a fi dezvoltat în continuare. Ca în orice demers inițial, această versiune poate să conțină omisiuni, greșeli sau inadvertențe care vor fi corectate în versiunile ulterioare.

În acest sens vă încurajăm să transmiteți sugestiile dumneavoastră folosind metodele de comunicare indicate mai jos. Pe lângă aceste sugestii suntem foarte interesați să înțelegem provocările cu care vă confrunțați în elaborarea documentațiilor de atribuire precum și în derularea achizițiilor ce privesc soluții software. Înțelegerea acestor provocări ne va ajuta în dezvoltarea ulterioară a acestui ghid dar și în elaborarea de instrumente conexe ghidului care să vă ajute în procesul de achiziție.

Adresa de contact pentru sugestii și observații: [ghidachizitii@anis.ro](mailto:ghidachizitii@anis.ro).





Versiunea actualizată a ghidului precum și alte instrumente și materiale relevante pot fi descărcate de la adresa:  
[www.anis.ro/programe/ghidachizitii](http://www.anis.ro/programe/ghidachizitii)

