



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2019 00168

(22) Data de depozit: 18/03/2019

(41) Data publicării cererii:
29/11/2019 BOPI nr. 11/2019

(71) Solicitant:
• IONICĂ ANDREEA CRISTINA,
STR.1 DECEMBRIE 1918, BL.74, SC.3,
AP.68, PETROȘANI, HD, RO;
• LEBA MONICA, ALEEA TRANDAFIRILOR
BL.4, SC.5, AP.51, PETROȘANI, HD, RO;
• DOVLEAC RALUCA- ANAMARIA,
STR. 1 DECEMBRIE 1918, BL.105, AP.63,
PETROȘANI, HD, RO

(72) Inventatori:
• IONICĂ ANDREEA CRISTINA,
STR.1 DECEMBRIE 1918, BL.74, SC.3,
AP.68, PETROȘANI, HD, RO;
• LEBA MONICA, ALEEA TRANDAFIRILOR
BL.4, SC.5, AP.51, PETROȘANI, HD, RO;
• DOVLEAC RALUCA - ANAMARIA,
STR. 1 DECEMBRIE 1918, BL.105, AP.63,
PETROȘANI, HD, RO

(54) PROCEDEU DE PLANIFICARE BAZAT PE UN ALGORITM
DE DIVIZARE OPTIMĂ A SARCINILOR DE LUCRU

(57) Rezumat:

Invenția se referă la un procedeu de planificare a sarcinilor de lucru, bazat pe un algoritm de determinare a ordinii de efectuare a sarcinilor de lucru pentru diverse tipuri de proiecte, ținând seama de modul în care sarcinile de lucru se influențează reciproc și de prioritățile stabilite în conformitate cu cerințele clienților. Procedeu conform invenției cuprinde: preluarea (4) de la clienți a necesităților, determinarea (5) cerințelor clienților, determinarea (6) sarcinilor de lucru aferente cerințelor clienților, stabilirea (7) interdependențelor dintre sarcinile de lucru și, cu ajutorul unui algoritm (3, 8), prioritizarea sarcinilor de lucru pe baza unui indicator care cuantifică gradul în care au fost satisfăcute cerințele clienților.

Revendicări: 2
Figuri: 3

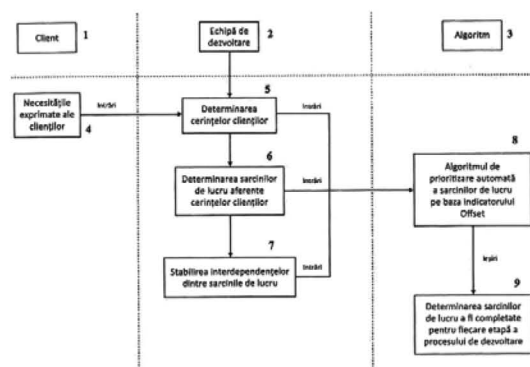


Fig. 1



PROCEDEU DE PLANIFICARE BAZAT PE UN ALGORITM DE DIVIZARE OPTIMĂ A SARCINILOR DE LUCRU

Invenția se referă la procedeul de a deduce în mod optim și automat ordinea de efectuare a sarcinilor de lucru pentru diverse tipuri de proiecte, ținând seama de modul în care sarcinile de lucru se influențează reciproc și de prioritățile stabilite în conformitate cu cerințele clienților.

Necesitatea procedeeului derivă din numărul ridicat de companii sau afaceri care există sau își desfășoară activitatea prin intermediul proiectelor din ce în ce mai complexe, proiecte care implică sute sau chiar mii de sarcini de lucru, de cele mai multe ori nerepetitive. Pentru buna gestiune a proiectelor și a sarcinilor de lucru aferente acestora, la momentul actual sunt utilizate platforme online de management de proiect care permit planificarea și monitorizarea sarcinilor de lucru și a gradului de îndeplinire a acestora. Deși aceste platforme dispun de multiple funcționalități și furnizează o baterie de instrumente folosite managementului de proiect, niciuna dintre aceste platforme nu furnizează o modalitate automată de divizare a sarcinilor de lucru, ci doar permit introducerea unor sarcini de lucru și stabilirea anumitor parametrii pentru acestea, precum: alocarea sarcinii de lucru unei anumite persoane, stabilirea unui termen limită pentru îndeplinirea sarcinii de lucru, stabilirea gradului de importanță sau de dificultate, ș.a.m.d.

În ceea ce privește modalitățile curente de divizare și prioritizare a sarcinilor de lucru, există mai multe abordări utilizate în practică și recunoscute în literatură, precum: Votul cumulativ, Analiza MoSCoW, Matricea Eisenhower sau punând accentul pe puterea de decizie a celui mai bine plătit om din companie (HiPPO - Highest-paid person's opinion).

O altă variantă de prioritizare a sarcinilor de lucru se bazează pe identificarea valorii care se dorește a fi adusă, astfel:

- Prioritizarea sarcinilor de lucru după valoarea adusă clientului;
- Prioritizarea sarcinilor de lucru după valoarea adusă afacerii;
- Alegerea sarcinilor de lucru care pot fi îndeplinite imediat în detrimentul celor mai complexe;
- Alegerea celor mai riscante sarcini de lucru pentru început;
- Alegerea sarcinilor de lucru calculând costurile ridicate de neîndeplinirea acestora;
- Alegerea sarcinilor de lucru prin identificarea interdependențelor dintre acestea și urmarea firului logic de dezvoltare a produsului;

- Alegerea sarcinilor de lucru care contribuie cel mai mult la atingerea țelului stabilit pentru etapa de dezvoltare curentă.

În contextul metodelor de priorizare și divizare eficientă a sarcinilor de lucru, algoritmul procedurii propus combină trei variante de priorizare a sarcinilor de lucru, și anume: alegerea sarcinilor de lucru în funcție de interdependențele dintre acestea, alegerea celor care contribuie cel mai mult la atingerea obiectivului stabilit pentru etapa de dezvoltare curentă și priorizarea sarcinilor de lucru după valoarea adusă clientului.

Algoritmul urmărește obținerea unui rezultat optim și obiectiv în ceea ce privește selectarea sarcinilor de lucru și împărțirea acestora în cadrul perioadelor de dezvoltare numite și etape de dezvoltare. Pe lângă aceasta, algoritmul reduce perioadele de timp afectate planificării sarcinilor de lucru incluse în fiecare etapă necesară și, mai mult decât atât, furnizează o soluție obiectivă, lipsită de subiectivismul membrilor echipei dezvoltatoare și a intereselor acestora.

O altă problemă rezolvată de către algoritm este furnizarea unui instrument care să poată cuantifica gradul de îndeplinire a cerințelor clienților la un moment dat, ținând cont de schimbarea rapidă a acestora. În acest sens, soluția vizată urmărește integrarea unei abordări iterative de dezvoltare a produselor solicitate de către clienți. Soluția are la bază o abordare de desfășurare a procesului de producție bazată pe ciclul de îmbunătățire continuă PDCA (Plan, Do, Check, Act) pentru a ajuta la organizarea și desfășurarea activităților de management.

Comparativ cu metodele tradiționale de dezvoltare a produselor în care cerințele clienților sunt identificate, produsele cerute sunt dezvoltate iar mai apoi livrate clientului la sfârșitul perioadei de dezvoltare. În cadrul abordării propuse, după identificarea cerințelor clienților, echipa dezvoltatoare a produsului identifică cele mai importante aspecte ale produselor dorite și le dezvoltă pe acestea, livrând clientului funcționalități reale ale produsului cerut, la intervale de timp prestabilite. Acest lucru ajută echipa dezvoltatoare să își adapteze produsele conform cerințelor clienților, chiar și în cazul în care aceștia din urmă solicită modificări semnificative privind forma sau funcționalitatea produsului dorit. Această metodă permite echipei de dezvoltare să se asigure de faptul că produsul dezvoltat este cel dorit de către client în fiecare etapă a procesului de dezvoltare. În continuare se prezintă modul de realizare a invenției cu referire la următoarele figuri:

Figura 1. Schema bloc

Figura 2. Schema de principiu

Figura 3. Schema logică

Figura 1 indică cele trei părți implicate în procesul de selectare și distribuire a sarcinilor de lucru în etape de dezvoltare.

Astfel [1] reprezintă clientul actual sau potențial al produselor și/sau serviciilor care sunt sau urmează a fi dezvoltate, și este persoana care formulează Necesitățile exprimate [4] referitoare la acestea. Necesitățile exprimate pot fi deduse în mai multe modalități, printre care: discuții directe cu clientul despre produsul sau serviciul dorit, studii de piață, sondaje de opinie, ș.a.m.d. Odată formulate, acestea sunt aduse la cunoștința Echipei de dezvoltare [2], formată atât din personalul direct productiv, de execuție, cât și din echipa managerială, iar pe baza lor, echipa de dezvoltare deduce Cerințele clienților [5] și le notează sub o formă care să evidențieze atât ceea ce dorește să primească clientul, cât și modul în care intenționează să utilizeze produsul și funcționalitatea așteptată de la acesta, cu scopul de a înțelege mai bine necesitățile clientului și felul în care pot fi satisfăcute acestea.

După notarea tuturor cerințelor clienților [5], echipa de dezvoltare [2] determină sarcinile de lucru [6] necesare a fi completate pentru a satisface aceste cerințe, și stabilește interdependențele dintre sarcinile de lucru [7] cu rolul de a furniza intrări pentru Algoritm [3]. Algoritm de prioritizare automată a sarcinilor de lucru [8] în etape de dezvoltare utilizează toate datele de intrare pentru Determinarea sarcinilor de lucru necesare a fi completate pentru fiecare etapă a procesului de dezvoltare [9], schema de principiu a acestuia fiind prezentată în Figura 2, iar schema logică a acestuia fiind prezentată în Figura 3.

Algoritm are la bază ideea divizării sarcinilor de lucru în etape de dezvoltare într-un mod care să permită atât maximizarea satisfacției clienților, cât și maximizarea funcționalității obținute la sfârșitul fiecărei etape de dezvoltare, satisfacție măsurată cu ajutorul indicatorului Offset [18]. Calculul indicatorului are la bază mecanismul specific metodei QFD (Quality Function Deployment) din cadrul managementului calității. Indicatorul Offset se calculează cu ajutorul următoarei formule:

$$offset = \sum_{j=1}^m \sum_{i=1}^n \frac{RUT(i,j) \cdot US(i)}{\sum_{i=1}^n US(i)}$$

Unde:

m – reprezintă numărul de sarcini de lucru, înregistrate în cadrul segmentului Sarcini de lucru [12]

n – reprezintă numărul de cerințe ale clienților, înregistrate în cadrul segmentului Cerințele clienților [10]

RUT – reprezintă valorile din Matricea Relațiilor [17], matrice în care sunt înregistrate valorile funcționale efective obținute prin îndeplinirea anumitor sarcini de lucru preselectate.

US – gradul de importanță a cerințelor clienților [11].

Valorile din cadrul Matricei Relațiilor [17] sunt calculate cu ajutorul următoarei formule:

$$RUT(i, j) = \begin{cases} 0, & \text{if } \prod_{k=1}^m (AT(k) + 1 - TT(k, j)) = 0 \\ \frac{IUT(i, j) \cdot T(j) \cdot AT(j)}{\sum_{r=1}^m IUT(1, r) \cdot T(r)}, & \text{else} \end{cases}$$

Unde:

AT – reprezintă stadiul sarcinii de lucru [15], acesta putând lua valoarea de 0 (pentru o sarcină de lucru nerealizată) sau 1 (pentru o sarcină de lucru realizată)

TT – reprezintă Matricea Corelațiilor [14], în cadrul căreia sunt evidențiate relațiile de interdependență dintre sarcinile de lucru

IUT – reprezintă Matricea Influențelor [16], matrice care reflectă ponderea pe care o are o sarcină de lucru în cadrul unei cerințe a clientului

T – reprezintă Gradul de dificultate al sarcinilor de lucru [13]

Calcularea indicatorul Offset [18] necesită parcurgerea următoarelor etape: Identificarea și notarea cerințelor clienților [10] sub o formă care să încorporeze funcționalitatea de bază urmărită de către client; Identificarea gradului de importanță [11] aferent fiecărei cerințe a clientului, cu scopul de a identifica cerințele care necesită satisfacerea imediată și a celor care pot fi satisfăcute ulterior. Pe baza cerințelor identificate sunt mai apoi deduse sarcinile de lucru [12] necesare a fi completate, gradul de dificultate al acestora [13] și relațiile de interdependență dintre acestea [14].

Apoi se verifică îndeplinirea sarcinilor de lucru [15] înregistrând sarcinile completate cu cifra 1 iar cele necompletate cu 0 și se stabilește măsura în care cerințele cuprind sarcinile de lucru, în formă procentuală, în cadrul Matricei Influențelor IUT [16]. Pe baza informațiilor de intrare înregistrate se pot calcula valorile din cadrul Matricei Relațiilor [17] iar ulterior poate fi calculat indicatorul Offset [18].

Schema logică prezentată în Figura 3 indică modul în care sarcinile de lucru sunt selectate pentru a fi distribuite în etape de dezvoltare a produsului. Astfel, inițial se declară variabila *sprint_offset* [19] și vectorul TS [20], variabilă ce reprezintă valoarea indicatorului Offset pentru etapa de dezvoltare curentă [19] și sarcinile de lucru selectate în etapa curentă [20].

În continuare se poate observa bucla repetitivă [21] - [26] caracteristică unui algoritm de determinare a maximului dintr-un șir de valori, inițiată prin declararea iteratorului i [21] și a variabilelor max și $index$ care au rolul de a contoriza numărul maxim de sarcini de lucru inhibate de către o sarcină de lucru neinhibată, respectiv indexul acesteia; blocul condițional inițial [22] asigură execuția algoritmului de determinare a maximului luând în considerare exclusiv sarcinile de lucru neinhibate; următorul bloc condițional [23] acționează ca un comparator între numărul de sarcini de lucru inhibate de către sarcina de lucru specifică iterației curente și valoarea variabilei max , dirijând execuția astfel încât la întâmpinarea unei valori mai mari sau egale cu cea a variabilei max se declanșează blocul de asignare [24] prin care valoarea variabilei max este suprascrisă de către numărul de sarcini de lucru inhibate de către sarcina de lucru curentă iar variabila $index$ înregistrează valoarea curentă a iteratorului i și anume indexul efectiv al sarcinii de lucru corespunzătoare; prin incrementarea iteratorului i [25] și interogarea condiției de oprire a buclei repetitive [26] se asigură execuția acesteia în intervalul $[1, m]$, unde m reprezintă numărul de sarcini de lucru existente.

După determinarea sarcinii de lucru neinhibată care inhibă cele mai multe alte sarcini de lucru, se recalculează valoarea indicatorului $Offset$ obținut prin îndeplinirea sarcinii de lucru respective. Prin compararea acestei valori cu cea a indicatorului $DeltaOffset$, se stabilește condiția de oprire [27] a buclei repetitive exterioare care stă la baza algoritmului recursiv de distribuire a sarcinilor de lucru pe etape de dezvoltare. Indicatorul $DeltaOffset$ arată valoarea reală a indicatorului $Offset$ din perioada de dezvoltare curentă, prin calcularea diferenței dintre valoarea indicatorului $Offset$ de la sfârșitul perioadei de dezvoltare și valoarea acestuia la începutul perioadei de dezvoltare.

Atâta timp cât valoarea cumulată reprezentând numărul maxim de sarcini de lucru inhibate de către sarcinile de lucru identificate anterior nu depășește valoarea indicatorului $DeltaOffset$, algoritmul înregistrează indexul sarcinii de lucru identificate în vectorul TS [28], după care invalidează rândul aferent sarcinii de lucru în copia auxiliară a Matricii Corelațiilor $TTaux$ prin suprascrierea fiecărui element cu valoarea 0, considerând astfel că sarcina de lucru selectată a fost completată și nu mai inhibă alte sarcini de lucru.

Ca o etapă premergătoare a iterației următoare, se recalculează vectorul TI [29], care conține sarcinile rămase nealocate iterațiilor, pentru determinarea noii liste de sarcini de lucru neinhibate (dintre care, din nou, se va selecta cea care inhibă cele mai multe alte sarcini de lucru). De asemenea, este necesară invalidarea poziției indexului curent a vectorului TI pentru a evita evaluarea acesteia în cadrul execuțiilor următoare ale algoritmului de determinare a maximului de sarcini de lucru inhibate. Algoritmul va rula până atunci când valoarea cumulată a



valorii variabilei `sprint_offset` va depăși valoarea indicatorului `DeltaOffset`, moment în care execuția ia sfârșit [30] și este posibilă inspectarea sarcinilor de lucru repartizate etapei de lucru curente prin analizarea pozițiilor vectorului `TS` care prezintă valoarea 1.

Similar metodei PDCA, la sfârșitul fiecărei etape de dezvoltare se verifică dacă rezultatul obținut este conform cu cerințele clienților iar pe baza acestui răspuns se aleg următorii pași.

REVENDICĂRI

1. Indicatorul Offset și procedeul de calcul al acestuia. Indicatorul Offset arată, în formă procentuală, cât de mult din cerințele clienților a fost obținut până la un moment dat, luând în considerare nu doar numărul de sarcini finalizate, ci și ponderea fiecăreia în cerințele clienților și gradul de dificultate al sarcinilor de lucru.
2. Algoritmul de prioritizare automată a sarcinilor de lucru în baza valorii indicatorului Offset. Algoritmul permite prioritizarea și divizarea automată a sarcinilor de lucru în etape de dezvoltare pe baza valorii indicatorului Offset, luând astfel în calcul scenariul care satisface cel mai bine cerințele clienților.

A handwritten signature in black ink, appearing to read "M. P. ...".

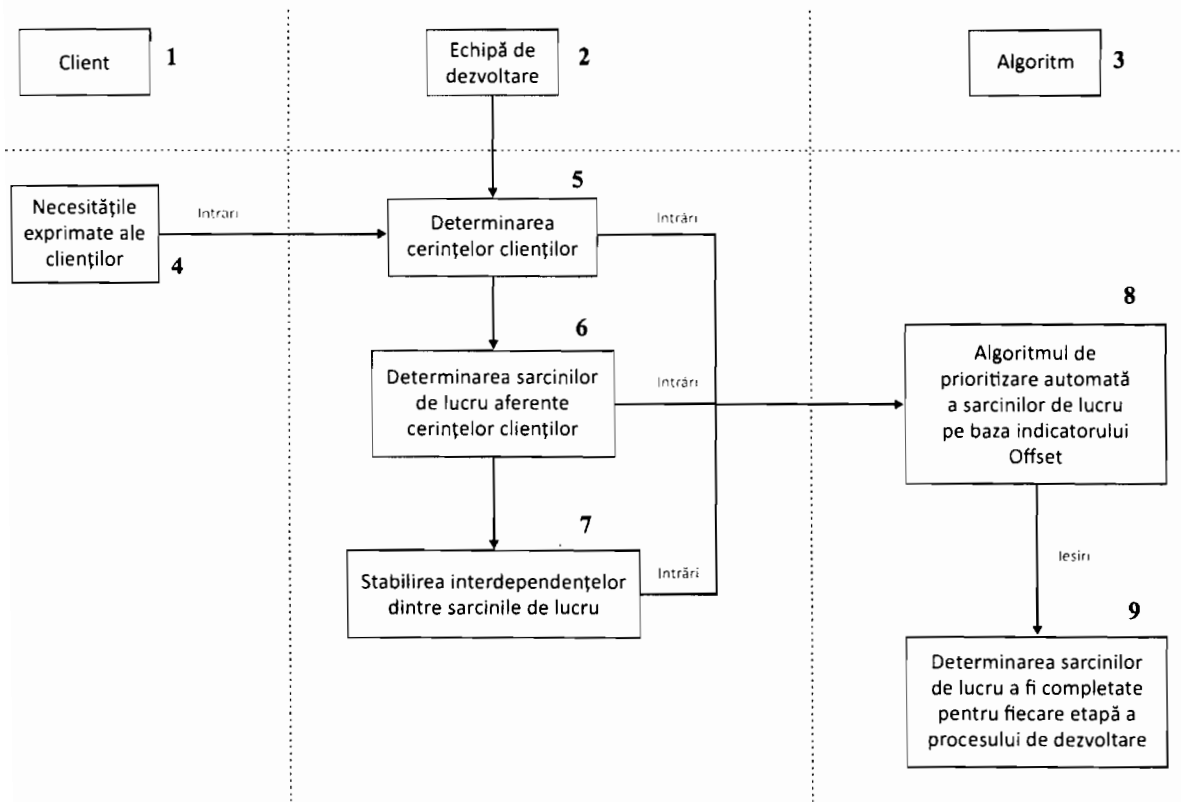


Figura 1. Schema bloc

Handwritten signature

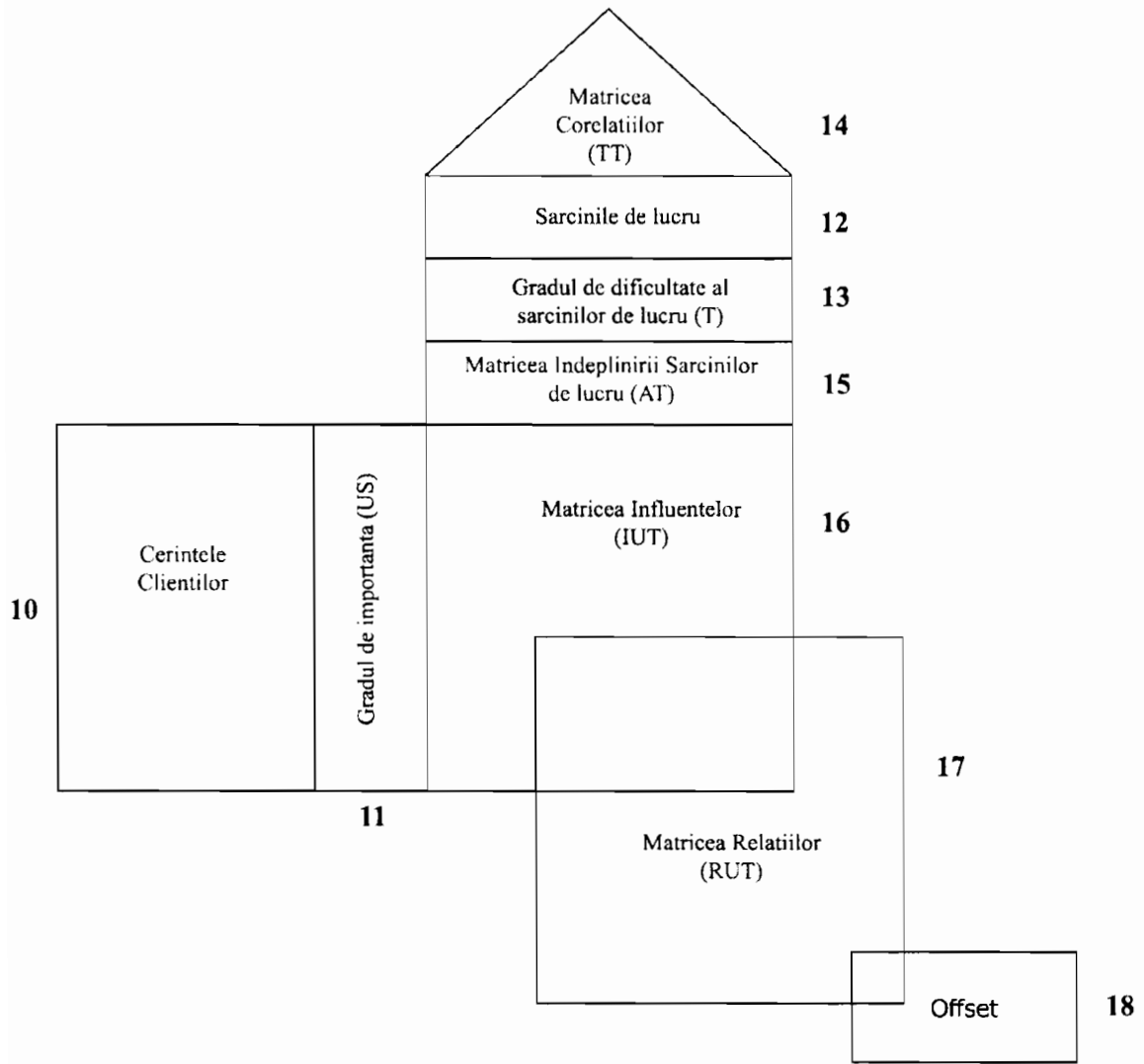


Figura 2. Schema de principiu

Handwritten signature

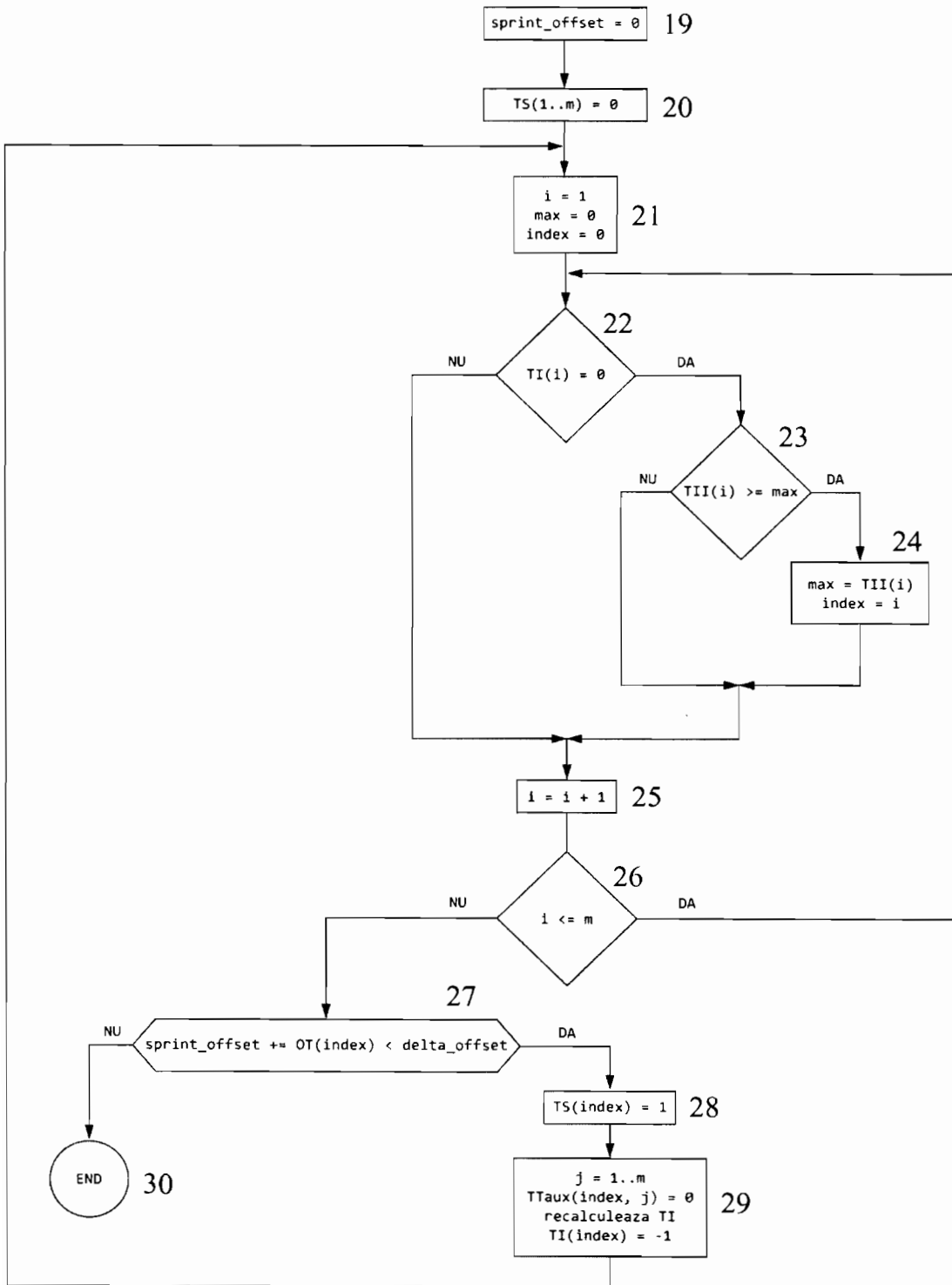


Figura 3. Schema logică

Handwritten signature