

SISTEMUL DE CUANTIFICARE A UNUI TASK

Explicarea mecanismului de calcul pentru primele 30 de zile si pentru perioada ulterioara

Autor: Prodan Alexandru Mihai

In randurile de fata sintetizez, intr-o forma clara si usor de urmarit, modul in care este cuantificat un task in aplicatia AXIS ADHD, pe baza formulelor definite in lucrarea sursa. Obiectivul nu este reformularea intregii lucrari, ci explicarea stricta a mecanismului prin care o sarcina primeste un punctaj final, ce variabile intra in calcul si de ce formula se modifica dupa primele 30 de zile de utilizare.

Modelul porneste de la o observatie simpla: doua sarcini care, la prima vedere, par similare pot avea in realitate greutate foarte diferite. Unele cer efort mare, dar aduc un castig functional redus; altele sunt scurte, insa au un efect important asupra organizarii, echilibrului emotional sau continuitatii rutinei. Din acest motiv, punctajul final nu este construit arbitrar, ci rezulta din intalnirea dintre dificultatea reala a task-ului, impactul sau emotional, profilul initial al utilizatorului, comportamentul observat in timp si o variatie controlata care mentine dinamica sistemului.

Acelasi tip de activitate poate produce scoruri diferite in functie de stadiul de utilizare al aplicatiei. In prima faza, sistemul compenseaza dificultatile initiale prin folosirea profilului rezultat din testul de configurare. In faza a doua, accentul nu mai cade pe ceea ce declara profilul initial, ci pe datele recente despre somn, miscare si ritm constant de lucru. Aceasta trecere de la un model bazat pe profil la unul bazat pe comportament este elementul central care face sistemul adaptiv.

In randurile urmatoare explic, in ordine logica, toate componentele formulei: E, I, T, Vi, G, P, r, bonusurile si penalizarile. Pe langa mecanismul propriu-zis, analiza include o comparatie directa cu hedonic calculus al lui Jeremy Bentham, precum si exemple numerice concrete care arata de ce modelul propus este fezabil, mai apropiat de realitate si semnificativ mai aplicabil decat modelele abstracte clasice de evaluare a utilitatii.

Am urmarit o formulare cat mai clara si cat mai fireasca. Punctajul unui task nu apare aici ca un simplu artificiu de gamificare, ci ca expresia unei relatii mai ample dintre efort, valoare functionala, context personal si consecventa comportamentala. Privit in ansamblu, sistemul nu incearca doar sa numere actiuni, ci sa surprinda felul in care o sarcina capata valoare in viata de zi cu zi si poate fi integrata coerent intr-o aplicatie functionala.

1. ARHITECTURA GENERALA A CUANTIFICARII

Cuantificarea unui task porneste de la ideea ca orice sarcina trebuie evaluata pe mai multe planuri simultan. O activitate poate fi dificila, dar putin valoroasa emotional; poate fi simpla, dar extrem de utila pentru stabilizarea rutinei; sau poate sa para usoara in teorie, dar sa fie foarte greu de initiat pentru un utilizator aflat intr-o perioada de dezechilibru. Din acest motiv, modelul analizat nu atribuie puncte numai pe baza duratei sau a numarului de subtask-uri, ci combina o evaluare obiectiva cu una functionala si comportamentala.

In termeni operationali, procesul urmeaza sase pasi. Primul pas este definirea task-ului: utilizatorul introduce denumirea, poate adauga subtask-uri, poate selecta un nivel de dificultate si, optional, un timp estimat si un calendar. Al doilea pas este calculul gradului de efort E. Al treilea pas este calculul impactului emotional I. Al patrulea pas este obtinerea indicelui combinat $T = E + I$, care devine nucleul functional al recompensarii. Al cincilea pas adauga factorul contextual: Vi in primele 30 de zile sau, ulterior, indicatorii G si P. Al saselea pas aplica randomizarea si eventualele bonusuri sau penalizari.

Schema de baza: task -> E -> I -> T -> componenta contextuala -> randomizare -> bonusuri/penalizari -> R

Sistemul separa ceea ce tine de valoarea intrinseca a task-ului de ceea ce tine de starea generala a utilizatorului. E si I descriu sarcina in sine. Vi, G si P descriu contextul in care acea sarcina este realizata. Randomizarea r nu schimba logica matematica, ci doar introduce o variatie controlata pentru a evita predictibilitatea absoluta. Bonusurile si penalizarile intervin abia la final, ca ajustari conditionate de comportamente specifice, de exemplu furnizarea dovezii foto sau repetarea excesiva a aceluiasi tip de task.

Prin urmare, scorul final R nu trebuie citit ca o masura simpla de dificultate. El trebuie interpretat ca valoarea functionala completa a task-ului in contextul curent al utilizatorului. Aceasta distinctie este esentiala: doua task-uri aparent similare pot primi punctaje diferite pentru ca sunt realizate in stadii diferite ale adaptarii utilizatorului, cu niveluri diferite de somn, miscare, consecventa sau vulnerabilitate initiala.

Aplicatia afiseaza recompensa inainte de confirmarea finala a task-ului. Din punct de vedere al experientei de utilizare, acest pas este util deoarece ii arata imediat utilizatorului daca sarcina este configurata realist si daca merita acceptata in forma respectiva. Cu alte cuvinte, cuantificarea nu este doar un calcul intern al sistemului, ci si un instrument de negociere si clarificare a sarcinii inainte de inceperea ei.

2. COMPONENTA E - GRADUL DE EFORT (PARTEA I)

Prima componenta majora a cuantificarii este gradul de efort, notat cu E. Aceasta componenta incearca sa masoare dificultatea task-ului dintr-o perspectiva combinata: ce simte utilizatorul despre sarcina, ce estimeaza sistemul despre ea si ce presupune concret activitatea in termeni de timp, complexitate si consum de resurse. Formula este construita pentru a evita dependenta exclusiva de autoevaluare sau, invers, dependenta exclusiva de clasificarea automata facuta de AI.

Indicatorii care intra in E

Indicator	Nota	Regula de evaluare
Dificultatea perceputa	dificultate_utilizator	Utilizatorul alege: usor = 1, moderat = 2, dificil = 3.
Dificultatea estimata de sistem	dificultate_estimata	AI-ul estimeaza obiectiv sarcina: usor = 0, moderat = 1, dificil = 2.
Timpul alocat	timp_alocat	Sub 10 minute = 1; intre 10 si 30 minute = 3; peste 30 minute = 5.

Dificultatea perceputa este importanta fiindca, pentru utilizator, rezistenta la initiere poate fi mai relevanta decat dificultatea tehnica a activitatii. De exemplu, raspunsul la un e-mail poate parea banal pentru un observator extern, dar poate fi trait ca fiind dificil din cauza anxietatii, evitarii sau supraincercarii executive. In formula ta, aceasta perceptie are valoare numerica si nu este ignorata.

Dificultatea estimata de sistem are rol de corectie. Ea tempereaza eventualele subevaluari sau supraevaluari ale utilizatorului. Daca acesta marcheaza un task ca fiind foarte usor, dar sistemul il recunoaste drept o sarcina complexa, formula nu il lasa sa ramana artificial la un nivel minimal. Invers, daca utilizatorul tinde sa considere orice activitate drept dificila, componenta estimata de sistem reduce distorsiunea.

Timpul alocat este tratat separat fiindca durata influenteaza direct costul de oportunitate si pragul motivational. O sarcina de 5 minute nu poate fi punctata identic cu una de 45 de minute doar pentru ca ambele sunt etichetate drept moderate. In plus, in lucrare se prevede ca, daca utilizatorul introduce o durata estimata, sistemul poate face o medie intre estimarea AI si cea oferita manual, astfel incat punctajul sa ramana plauzibil.

3. COMPONENTA E - GRADUL DE EFORT (PARTEA A II-A)

Pe langa dificultatea perceputa, dificultatea estimata si durata, formula E mai include complexitatea si consumul de energie. Aceste doua variabile permit sistemului sa diferentieze intre o sarcina scurta dar fragmentata in multe etape si o sarcina liniara, precum si intre un task cognitiv usor si unul care produce o oboseala mentala sau fizica semnificativa.

Indicator	Valori	Explicatie
Complexitatea	mica = 1; medie = 3; mare = 6	Se bazeaza pe numarul de subtask-uri: 1-2, 3-5, respectiv peste 6.
Energia consumata	scazut = 1; moderat = 3; intens = 5	Este estimata de AI in functie de efortul mental sau fizic cerut de activitate.

$$E = \text{dificultate_utilizator} + \text{dificultate_estimata} + \text{timp_alocat} + \text{complexitate} + \text{energie}$$

$$\text{Intervalul teoretic al lui E: } E \in [4, 21]$$

Valoarea minima este 4 deoarece chiar si un task foarte simplu are un cost minim: utilizatorul il considera usor, sistemul il recunoaste ca usor, durata este sub 10 minute, complexitatea este mica si energia este scazuta. Valoarea maxima este 21 si apare atunci cand toate componentele se afla la nivelul superior. Acest interval este util pentru normalizarea ulterioara, dar mai ales pentru compararea task-urilor intre ele.

Exemplu concret: se considera un task formulat ca redacteaza si trimite proiectul final. Utilizatorul il percepe ca dificil = 3. Sistemul il estimeaza ca moderat spre dificil = 2. Timpul depaseste 30 de minute = 5. Exista patru subtask-uri, deci complexitatea este medie = 3. Energia mentala estimata este moderata = 3. In acest caz, $E = 3 + 2 + 5 + 3 + 3 = 16$.

Din punct de vedere practic, E nu spune inca ce merita task-ul pentru utilizator. El spune doar cat de solicitanta este sarcina in sine. Un task poate avea E mare, dar impact emotional redus. De aceea, formula nu se opreste aici. Valoarea de efort trebuie completata de o a doua componenta, care masoara cat de valoros sau daunator este acel task pentru starea psihologica si functionala a utilizatorului.

4. COMPONENTA I - IMPACTUL EMOTIONAL

A doua componenta a sistemului este impactul emotional, notat cu I. Ea apare pentru ca nu toate task-urile trebuie judecate numai dupa efort. Unele activitati sunt dificile, dar produc o imbunatatire semnificativa a starii de bine sau a stabilitatii pe termen lung. Altele sunt usoare si chiar placute pe moment, dar pot fi daunatoare sau lipsite de valoare functionala. Prin introducerea lui I, modelul analizat recompenseaza nu doar munca depusa, ci si efectul real al sarcinii asupra vietii utilizatorului.

Indicator	Scala	Explicatie
Impact emotional imediat (IEI)	negativ = -2; inexistent = 0; redus = 4; mediu = 7; ridicat = 13	Reflecta starea de bine imediata dupa finalizarea task-ului.
Impact emotional pe termen lung (IET)	inexistent = -2; relevant = 8; semnificativ = 18	Reflecta persistenta beneficiului dupa 12 h, 12-72 h sau peste 72 h.

$$I = IEI + IET$$

Intervalul teoretic al lui I: $I \in [-4, 31]$

Un exemplu util pentru intelegere este comparatia dintre doua activitati. Daca utilizatorul finalizeaza o sesiune scurta de organizare a biroului, impactul imediat poate fi mediu, iar efectul pe termen lung relevant. Daca, in schimb, utilizatorul bifeaza o activitate impulsiva sau nociva, placerea de moment poate fi chiar absenta sau negativa, iar efectul pe termen lung poate fi clar nefavorabil. Formula permite astfel ca unele task-uri sa primeasca o valoare I foarte mica sau chiar negativa.

Sa luam un exemplu pozitiv: dupa finalizarea unui task de curatarea completa a spatiului de lucru, sistemul poate evalua impactul imediat la 7 si impactul pe termen lung la 8, rezultand $I = 15$. In cazul unui task mic, dar cu efect durabil, cum ar fi programare consult medical, I poate deveni chiar mai important decat E, deoarece valoarea functionala a actiunii depaseste efortul brut depus pentru executare.

Prin introducerea lui I, sistemul evita o eroare frecventa a multor aplicatii de productivitate: aceea de a trata toate activitatile ca pe simple unitati de lucru. Modelul analizat recunoaste ca anumite sarcini au valoare terapeutica, reglatoare sau organizatoare disproportionat de mare fata de durata lor efectiva. Aceasta este una dintre ideile cele mai puternice ale mecanismului de cuantificare.

5. INDICELE COMBINAT T

Dupa calcularea lui E si I, sistemul construiesc un indicator unic, notat cu T, care uneste intr-o singura valoare greutatea obiectiva a sarcinii si valoarea sa psihologica. Aceasta alegere simplifica formula finala si face posibil un sistem de recompensare scalabil. In loc sa trateze separat fiecare indicator in formula finala, modelul foloseste un indice principal care concentreaza esenta task-ului.

$$T = E + I$$

Intervalul teoretic al lui T: $T \in [0, 52]$

Lucrarea atribuie lui T cea mai mare pondere din scorul final: 55%. Motivul este logic. T inglobeaza atat efortul, cat si impactul emotional, adica exact ceea ce descrie valoarea functionala a sarcinii. Daca aceasta componenta ar avea o pondere prea mica, scorul final ar fi dominat de context si nu de task. Daca ar avea o pondere prea mare, sistemul ar deveni rigid si ar ignora starea reala a utilizatorului. Valoarea de 55% pastreaza echilibrul dintre obiectul evaluat si contextul in care este evaluat.

$$PT = 0,55 \times (T / 52)$$

Normalizarea prin impartire la 52 este necesara deoarece T poate avea amplitudini mari. Fara aceasta operatiune, o sarcina cu valoare ridicata a lui T ar domina integral scorul final si ar face celelalte componente irelevante. Dupa normalizare, PT devine o contributie procentuala controlata, cu valori intre 0 si 0,55.

Daca este reluat exemplul anterior, in care $E = 16$ si $I = 15$, rezulta $T = 31$. Componenta procentuala devine $PT = 0,55 \times (31 / 52) =$ aproximativ 0,3279. Aceasta cifra nu este inca scorul final afisat utilizatorului, ci doar partea cea mai importanta din el. Restul scorului depinde de etapa de utilizare a aplicatiei: in primele 30 de zile se adauga profilul initial V_i , iar ulterior se adauga indicatorii dinamici G si P, plus randomizarea.

Din punct de vedere conceptual, T reprezinta raspunsul la intrebarea: cat de valoroasa este aceasta sarcina in sine. Toate celelalte componente raspund la alta intrebare: cat de mult ar trebui ajustata recompensa in functie de persoana si de starea ei actuala. Aceasta separatie face modelul foarte usor de explicat si de implementat software.

6. PROFILUL INITIAL V SI ROLUL SAU IN PRIMELE 30 DE ZILE

In prima luna de utilizare, formula se bazeaza pe un vector initial V rezultat din profilarea utilizatorului. Vectorul are trei componente: V1 pentru impulsivitatea attentionala, V2 pentru impulsivitatea motorie si V3 pentru nivelul de planificare sau organizare. Fiecare componenta poate lua valorile 0, 1 sau 2, corespunzatoare nivelurilor scazut, moderat si ridicat.

$$V_i = V_1 + V_2 + V_3$$

$$V_i \in [0, 6] \text{ si } PVi = 0,35 \times (V_i / 6)$$

Semnificatia acestei componente este dubla. Mai intai, V_i intra direct in formula finala si contribuie cu 35% la punctaj in primele 30 de zile. In al doilea rand, valorile individuale ale lui V1, V2 si V3 configureaza modul in care aplicatia structureaza task-urile. Prin urmare, vectorul initial nu este doar un multiplicator matematic, ci si o cheie de personalizare a mediului digital.

Componenta	Valoare mare inseamna	Efecte principale in aplicatie
V1 - atentie	neatentie predominanta	Sarcini mai scurte, notificari mai dese, bonusuri pentru task-uri scurte si dinamice.
V2 - motor	agitatie predominanta	Task-uri fizice sau dinamice prioritizate, limbaj mai activ, bonus posibil pentru task-uri fizice.
V3 - planificare	dezorganizare ridicata	Fragmentare crescuta, mod asistat activ, feedback frecvent si bonus pentru structura clara.

Aceasta alegere este foarte importanta pentru prima etapa. Utilizatorul nu are inca istoric real de utilizare, iar aplicatia nu poate sti daca somnul, activitatea fizica sau ritmul de finalizare sunt stabile. In lipsa acestor date, profilul initial functioneaza ca o forma de compensare si adaptare. Cu cat nevoile initiale sunt mai accentuate, cu atat recompensa poate fi calibrata mai favorabil, pentru a creste sansa de aderenta la sistem.

Mai simplu spus, V_i raspunde la intrebarea: de cat sprijin motivational are nevoie utilizatorul in faza de inceput. De aceea, dupa 30 de zile, aceasta componenta este inlocuita. Nu pentru ca ar fi gresita, ci pentru ca devine tot mai putin relevanta comparativ cu datele reale produse de comportamentul zilnic.

7. FORMULA FINALA IN PRIMELE 30 DE ZILE

In prima etapa, scorul final al unui task este format din trei componente principale: valoarea intrinseca a task-ului exprimata prin PT, profilul initial al utilizatorului exprimat prin PVi si randomizarea controlata exprimata prin Pr. Formula oficiala din lucrare este urmatoarea:

$$R = [10 \times (PT + PVi + Pr)] \times \text{alte bonusuri, daca exista}$$

Aici, PT poate contribui cu maximum 55%, PVi cu maximum 35%, iar Pr cu maximum 10%. Factorul de scalare cu 10 transforma un rezultat de tip 0,64 intr-un scor usor de citit, de exemplu 6,4 puncte. El nu modifica raporturile matematice dintre componente, ci doar face punctajul mai intuitiv pentru utilizator.

Exemplu numeric: se considera un task cu $E = 16$ si $I = 15$. Atunci $T = 31$, iar $PT = 0,55 \times (31 / 52) = 0,3279$. Pentru profilul initial luam $V1 = 2$, $V2 = 1$, $V3 = 2$, deci $Vi = 5$. Rezulta $PVi = 0,35 \times (5 / 6) = 0,2917$. Daca randomizarea produce $r = 4$, atunci $Pr = 0,10 \times (4 / 12) = 0,0333$. Scorul de baza devine $R = 10 \times (0,3279 + 0,2917 + 0,0333) = 6,53$ puncte.

Daca acelasi task este introdus de un utilizator cu profil initial mai putin vulnerabil, de exemplu $Vi = 2$, atunci PVi devine 0,1167 si scorul de baza scade la aproximativ 4,78 puncte. Observatia importanta este ca sarcina nu s-a schimbat; s-a schimbat doar necesarul de sprijin motivational al utilizatorului. Aceasta este exact functia primei etape: sa faca sistemul suficient de incurajator pentru cei care, la inceput, pornesc de la un nivel mai mare de dificultate.

Daca exista un bonus de +10%, de exemplu pentru o fotografie valida sau pentru o structura clara a task-ului, interpretarea operationala naturala este aplicarea unui factor de 1,10 asupra scorului de baza. In exemplul de mai sus, $6,53 \times 1,10 = 7,18$ puncte. Lucrarea exprima aceste ajustari ca bonusuri procentuale aplicate lui R, ceea ce lasa loc unei implementari simple si transparente.

Rezulta astfel un sistem in care prima luna nu urmareste doar masurarea performantei, ci si sustinerea tranzitiei catre utilizarea regulata a aplicatiei. Cu alte cuvinte, formula initiala este deliberat mai protectiva si mai motivationala decat formula ulterioara.

8. DE CE SE SCHIMBA FORMULA DUPA 30 DE ZILE

Dupa primele 30 de zile, lucrarea renunta la V_i ca factor direct de scor si il inlocuieste cu doi indicatori dinamici: G si P. Logica este foarte buna. Profilul initial descrie o stare de pornire, nu comportamentul real pe termen mediu. Daca aplicatia ar continua sa recompenseze mereu pe baza aceluiasi profil fix, ea ar risca sa pastreze aceeaasi configuratie chiar si atunci cand utilizatorul si-a schimbat deja rutina sau, dimpotriva, cand aceasta s-a deteriorat.

Etapa a doua muta accentul de la vulnerabilitatea initiala la comportamentul efectiv. Aplicatia incepe sa intrebe, implicit, nu care era configuratia initiala, ci care este configuratia comportamentala curenta. Aceasta schimbare este cruciala pentru orice sistem adaptiv matur. Ea transforma punctajul dintr-un instrument de onboarding intr-un instrument de reglare fina a comportamentului.

Noul model pastreaza PT drept nucleu stabil al valorii task-ului, dar in locul lui PV_i introduce: G, indicatorul somn plus miscare, cu pondere de 25%, si P, indicatorul progresului lunar, cu pondere de 10%. Randomizarea ramane activa cu 10%. In acest fel, suma ponderilor ramane coerenta si scorul final continua sa reflecte atat sarcina, cat si starea curenta a utilizatorului.

Tranzitia de principiu: profil fix (V_i) -> indicatori dinamici (G si P)

Schimbarea are si o valoare motivationala. In prima etapa, utilizatorul simte ca sistemul il sprijina. In etapa a doua, utilizatorul incepe sa simta ca stilul sau de viata are consecinte directe asupra recompensei. Daca doarme mai bine, se misca mai mult si mentine un ritm constant de finalizare, punctajele cresc. Daca rutina se degradeaza, scorurile devin mai modeste. Sistemul incepe astfel sa invete comportamental, nu doar sa compenseze vulnerabilitatea initiala.

Din perspectiva implementarii, aceasta trecere este una dintre cele mai valoroase caracteristici ale modelului analizat. Ea arata ca aplicatia nu ramane blocata intr-o fotografie initiala a utilizatorului, ci foloseste date noi pentru a actualiza permanent recompensa. In termeni practici, cuantificarea devine mai corecta si mai rezistenta la schimbari de context.

9. INDICATORUL G - SOMN SI MISCARE

Primul indicator introdus dupa 30 de zile este G. Acesta cuantifica doua dimensiuni ale starii generale: calitatea odihnei si nivelul de activitate fizica din ultimele 24 de ore. Ideea de baza este simpla: un utilizator care doarme suficient si se misca mai mult are sanse mai mari sa initieze si sa finalizeze sarcini cu eficienta. Prin urmare, sistemul nu se limiteaza la evaluarea task-ului, ci ia in calcul si terenul biologic pe care acel task este executat.

Componenta	Regula de punctaj	Valori
Numar de pasi	sub 1000 = 0; intre 1000 si 3000 = 2; peste 3001 = 3	maxim 3 puncte
Ore de somn	sub 7 ore = 0; intre 7 si 9 ore = 2; peste 9 ore = 3	maxim 3 puncte

$$G = \text{somn} + \text{numar_de_pasi}$$

$$G \in [0, 6] \text{ si } PG = 0,25 \times (G / 6)$$

Lucrarea propune o metoda pragmatica pentru estimarea somnului chiar si in absenta unui dispozitiv dedicat. Utilizatorul isi declara ora de culcare, iar aplicatia urmareste activitatea telefonului. Daca activitatea dupa acea ora este minima, sistemul presupune ca somnul s-a desfasurat conform planului. Daca telefonul ramane activ, momentul de adormire este mutat catre prima perioada de inactivitate suficient de lunga. Aceasta solutie nu este una clinica, dar este suficienta pentru un indicator comportamental util in aplicatie.

Exemplu: daca utilizatorul a dormit 7,5 ore si a facut 3.200 de pasi, componenta somn primeste 2 puncte, iar componenta pasi primeste 3 puncte. Rezulta $G = 5$. Contributia la scor devine $PG = 0,25 \times (5 / 6) = 0,2083$. Aceasta este o contributie substantiala, ceea ce arata cat de importanta devine rutina biologica in etapa a doua.

Mesajul implicit al formulei este foarte valoros: recompensa pentru task nu mai depinde numai de sarcina si de profilul psihologic, ci si de igiena functionala zilnica. Astfel, cuantificarea devine un instrument de feedback indirect despre stilul de viata, fara a transforma aplicatia intr-un simplu monitor de sanatate.

10. INDICATORUL P - PROGRESUL LUNAR

Al doilea indicator din etapa a doua este P, conceput pentru a masura consecventa pe termen mediu. Spre deosebire de G, care raspunde la intrebarea cum este utilizatorul in ultimele 24 de ore, indicatorul P raspunde la intrebarea cat de constant a fost utilizatorul in ultima luna. Aceasta diferenta este importanta deoarece performanta unei singure zile poate fi accidentala, in timp ce media task-urilor finalizate pe zi ofera o imagine mai stabila.

Media task-urilor finalizate pe zi	Valoarea lui P
0-8 task-uri / zi	1
9-14 task-uri / zi	3
15-30 task-uri / zi	5
peste 31 task-uri / zi	7

$$PP = 0,10 \times (P / 7)$$

Ponderarea de numai 10% arata ca progresul lunar nu trebuie sa eclipseze valoarea task-ului curent. El functioneaza ca un indicator de stabilitate, nu ca determinant principal. Aceasta decizie este corecta: un utilizator nu trebuie sa fie pedepsit excesiv intr-o zi buna doar pentru ca a avut o luna slaba, dar nici sa primeasca aceeasi recompensa daca lucreaza haotic si fara continuitate.

Exemplu: daca utilizatorul finalizeaza in medie 12 task-uri pe zi in ultimele 30 de zile, valoarea lui P este 3. Atunci $PP = 0,10 \times (3 / 7) = 0,0429$. Daca media urca la 22 task-uri pe zi, P devine 5 si PP urca la 0,0714. Diferenta nu este enorma, dar este suficienta pentru a arata ca ritmul sustinut este recompensat.

Indicatorul P rezolva o problema frecventa in aplicatiile de productivitate: tentatia de a produce explozii scurte de activitate urmate de abandon. Prin legarea unei parti din scor de media lunara, sistemul descurajeaza performanta teatrala si favorizeaza rutina reala. In cazul utilizatorilor cu ADHD, aceasta alegere este deosebit de relevanta deoarece stabilitatea, nu intensitatea izolata, este de regula cea mai greu de mentinut.

Din perspectiva tehnica, P este usor de calculat: totalul task-urilor validate in ultimele 30 de zile se imparte la numarul de zile analizate, apoi media este mapata in una dintre cele patru trepte. Avantajul acestei abordari este simplitatea. Ea este suficient de clara pentru utilizator si suficient de robusta pentru dezvoltare.

11. RANDOMIZAREA, BONUSURILE SI PENALIZARILE

Pe langa componentele deterministe, sistemul include un factor de randomizare notat cu r . Rolul sau nu este de a transforma scorul intr-un joc de noroc, ci de a preveni predictibilitatea absoluta. Daca utilizatorul ar sti ca doua task-uri similare produc intotdeauna exact acelasi scor, valoarea motivationala a sistemului s-ar reduce in timp. De aceea, modelul introduce o variatie mica, dar controlata.

Distributia propusa in lucrare este urmatoarea:

- 1) valori intre 1 si 5 cu probabilitate 50%
- 2) valori intre 6 si 10 cu probabilitate 35%
- 3) valori intre 11 si 12 cu probabilitate 15%

$$Pr = 0,10 \times (r / 12)$$

Prin aceasta normalizare, randomizarea poate contribui cu maximum 10% la scor. Ea nu poate rasturna ierarhia dintre task-uri, dar poate produce variatii suficient de vizibile incat utilizatorul sa simta dinamismul sistemului. In cazul unui r mare, lucrarea propune chiar semnalarea unui tip de eveniment de tip Jackpot sau Critical Hit.

Bonusurile si penalizarile actioneaza dupa calcularea scorului de baza. Exemplele explicite din lucrare sunt foarte importante pentru cuantificarea practica: furnizarea dovezii foto aduce +10%; refuzul dovezii reduce recompensa la doar 10% din punctajul initial al task-ului; dupa 3 zile consecutive de activitate, a patra zi aduce +2% global timp de 12 ore; repetarea excesiva a aceluiasi task injumatatesteste scorul la a doua repetare consecutiva, iar la a treia repetare anuleaza punctele si adauga o penalizare de -5%.

Din punct de vedere operational, o implementare curata este tratarea acestor bonusuri ca multiplicatori aplicati lui R . De exemplu, pentru dovada foto, $R_{final} = R \times 1,10$. Pentru refuzul dovezii, $R_{final} = R \times 0,10$. Pentru penalizarea de -5%, se poate aplica un factor 0,95 dupa anularea punctajului de baza sau, daca se doreste o implementare mai stricta, se poate inregistra o penalizare separata in istoricul utilizatorului.

Aceasta combinatie dintre randomizare, bonus si penalizare produce un sistem viu, dar nu haotic. Utilizatorul simte ca exista surpriza, recompensa suplimentara pentru comportamente bune si cost pentru tentativele de exploatare mecanica a aplicatiei. Din punct de vedere motivational, aceasta parte este la fel de importanta ca formula de baza.

12. FORMULA FINALA DUPA PRIMELE 30 DE ZILE

Dupa stabilirea unei baze comportamentale, formula finala se rescrie prin inlocuirea lui PV_i cu PG si PP . Astfel, punctajul unui task depinde in continuare de valoarea lui intrinseca, dar este ajustat in functie de somn, miscare, progres lunar si randomizare. Formula oficiala devine:

$$R = [10 \times (PT + PG + PP + Pr)] \times \text{alte bonusuri, daca exista}$$

Ponderile se redistribuie astfel: $PT = 55\%$, $PG = 25\%$, $PP = 10\%$ si $Pr = 10\%$. Observam ca sistemul pastreaza acelasi total de 100% , dar muta accentul de pe profilul static pe comportamentul observabil. Aceasta schimbare este ceea ce transforma mecanismul intr-un model adaptiv real.

Exemplu complet: se foloseste acelasi task ca mai sus, cu $E = 16$, $I = 15$, $T = 31$ si $PT = 0,3279$. Se considera ca utilizatorul a dormit $7,5$ ore si a facut peste 3000 de pasi; atunci $G = 5$ si $PG = 0,25 \times (5 / 6) = 0,2083$. In ultimele 30 de zile media a fost de 12 task-uri pe zi, deci $P = 3$ si $PP = 0,10 \times (3 / 7) = 0,0429$. Daca randomizarea da $r = 4$, avem $Pr = 0,0333$.

Scorul de baza devine $R = 10 \times (0,3279 + 0,2083 + 0,0429 + 0,0333) =$ aproximativ $6,12$ puncte. Daca acelasi utilizator ajunge la $G = 6$ si $P = 5$, scorul urca la $10 \times (0,3279 + 0,2500 + 0,0714 + 0,0333) = 6,83$ puncte. Diferenta vine exclusiv din imbunatatirea rutinei si a consecventei, nu din schimbarea task-ului.

Aceasta formula are o valoare pedagogica foarte buna pentru utilizator. Ea transmite ideea ca recompensa pentru o sarcina nu este izolata de felul in care acesta traieste. Un task realizat in conditii de somn bun, miscare si disciplina constanta este valorizat mai mult decat acelasi task realizat intr-un context dezorganizat. Mesajul este subtil, dar extrem de puternic.

Important este si faptul ca formula ramane simpla. Desi integrarea pare sofisticata la nivel conceptual, implementarea software este directa: se calculeaza patru contributii normalizate, se aduna, se inmultesc cu 10 si apoi se aplica bonusurile. Simplitatea aceasta o face potrivita pentru productie.

13. EXEMPLE COMPARATIVE SI INTERPRETAREA REZULTATELOR

Pentru a vedea clar diferenta dintre cele doua etape, este util sa comparam acelasi task in doua momente distincte. Se considera sarcina pregateste dosarul pentru consultatia medicala. Ea are dificultate perceputa 2, dificultate estimata 1, timp 3, complexitate 3 si energie 3, deci $E = 12$. Impactul emotional imediat este 7, iar cel pe termen lung 18, deci $I = 25$. Rezulta $T = 37$ si $PT = 0,55 \times (37 / 52) = 0,3913$.

In prima etapa, pentru un utilizator cu $V1 = 2$, $V2 = 1$, $V3 = 2$, avem $V_i = 5$ si $PV_i = 0,2917$. Daca randomizarea da $r = 3$, atunci $Pr = 0,0250$. Scorul de baza este $10 \times (0,3913 + 0,2917 + 0,0250) = 7,08$ puncte. Daca se aplica bonusul de +10% pentru dovada foto, scorul urca la 7,79 puncte.

In etapa a doua, acelasi task este evaluat pe baza comportamentului actual. Daca utilizatorul a dormit sub 7 ore si a facut sub 1000 de pasi, $G = 0$, deci $PG = 0$. Daca media lunara este de 9 task-uri pe zi, atunci $P = 3$ si $PP = 0,0429$. Cu acelasi $r = 3$, scorul devine $10 \times (0,3913 + 0 + 0,0429 + 0,0250) = 4,59$ puncte. Diferenta este mare si arata clar cum lipsa odihnei si miscarii reduce recompensa, chiar daca task-ul are o valoare functionala ridicata.

Daca, in schimb, acelasi utilizator doarme 8 ore, face peste 3000 de pasi si are media lunara de 20 task-uri pe zi, atunci $G = 5$, $PG = 0,2083$, iar $P = 5$, $PP = 0,0714$. Scorul ajunge la $10 \times (0,3913 + 0,2083 + 0,0714 + 0,0250) = 6,96$ puncte. Observam ca sistemul nu pedepseste task-ul, ci reevalueaza recompensa in functie de contextul comportamental al utilizatorului.

Interpretarea corecta a scorului este urmatoarea: un punctaj mare nu inseamna doar ca task-ul este greu, ci ca este valoros, bine incadrat in rutina curenta si sustinut de un comportament favorabil. Un punctaj mai mic nu inseamna neaparat ca task-ul este lipsit de importanta, ci poate semnala un context biologic sau comportamental slab care reduce capacitatea de functionare. Acesta este motivul pentru care sistemul analizat poate fi util nu doar pentru recompensare, ci si pentru monitorizare indirecta.

14. SINTEZA MECANISMULUI DE CALCUL

Sintetizand, mecanismul de cuantificare functioneaza in doua faze clare. In faza initiala, formula recompenseaza valoarea task-ului si compenseaza vulnerabilitatea de pornire a utilizatorului. In faza ulterioara, formula recompenseaza valoarea task-ului, dar muta accentul pe somn, miscare si consecventa. Aceasta tranzitie transforma sistemul dintr-un mecanism de adaptare initiala intr-un mecanism de reglare continua.

Element	Primele 30 de zile	Dupa 30 de zile
Nucleul task-ului	PT = 55%	PT = 55%
Context principal	PVi = 35%	PG = 25% + PP = 10%
Variatie controlata	Pr = 10%	Pr = 10%
Sens dominant	sprijin motivational de inceput	responsabilizare prin comportament real

Logica minima de implementare poate fi scrisa astfel:

- 1) se citește task-ul și se calculează E;
- 2) se estimează IEI și IET, apoi se obține I;
- 3) se calculează $T = E + I$ și apoi PT;
- 4) dacă utilizatorul este în primele 30 de zile, se calculează PVi; altfel se calculează PG și PP;
- 5) se calculează Pr pe baza lui r;
- 6) se obține R prin formula specifică etapei și se aplică bonusurile sau penalizarile;
- 7) scorul final se afișează utilizatorului și se înregistrează în istoric.

Punctul forte al modelului analizat este echilibrul. Formula este suficient de simplă pentru a fi implementată ușor, dar suficient de bogată pentru a reflecta dificultatea reală a task-ului, impactul sau funcțional și starea curentă a utilizatorului. Mai mult, bonusurile și penalizarile previn abuzul și mențin caracterul motivant al sistemului.

15. ELEMENTE DISTINCTIVE ALE SISTEMULUI

Particularitatea principală a sistemului analizat nu constă doar în faptul că alocă puncte, ci în modul în care definește valoarea unui task. Majoritatea sistemelor digitale de productivitate folosesc fie reguli foarte simple, de tip un task egal un număr fix de puncte, fie logici aproximative, în care durata sau bifarea unei activități devin criterii aproape exclusive. Modelul propus se desprinde de această schemă simplificatoare și introduce o cuantificare stratificată, în care valoarea sarcinii rezultă din interacțiunea dintre efort, impact emoțional, profil psihologic, comportament recent și consistența pe termen mediu.

Inovația este vizibilă și la nivel de arhitectură temporală. Formula nu rămâne identică pe toată durata utilizării, ci se schimbă după primele 30 de zile. Aceasta înseamnă că sistemul recunoaște un fapt foarte important: datele relevante pentru un utilizator nou nu sunt aceleași cu datele relevante pentru un utilizator care a acumulat deja istoric comportamental. Prin urmare, algoritmul nu este static, ci făcut în timp.

Un al treilea element distinctiv este amestecul dintre date subiective și date obiective. Dificultatea percepută de utilizator nu este ignorată, dar nici nu este lăsată necontrolată. Ea este pusă în dialog cu estimarea făcută de sistem. La fel, impactul emoțional nu este tratat ca o impresie vagă, ci este tradus într-un indicator numeric cu efect direct asupra punctajului final. Astfel, modelul conservă complexitatea experienței umane fără să renunțe la rigoarea matematică.

Inovațiile menționate au și o consecință practică importantă: sistemul poate fi explicat, auditat și ajustat. Spre deosebire de multe mecanisme opace bazate exclusiv pe AI, modelul analizat are formule clare, intervale finite și ponderi explicite. Aceasta îl face nu doar mai interesant teoretic, ci și mult mai potrivit pentru implementare reală într-un produs digital care trebuie justificat în fața utilizatorilor, specialiștilor sau finanțatorilor.

Prin urmare, specificul sistemului vine din combinarea a patru proprietăți rare în același model: personalizare, adaptare în timp, cuantificare explicabilă și legătură directă cu comportamentul real al utilizatorului. Tocmai această combinație îl diferențiază de un simplu mecanism de gamificare.

16. PERSONALIZARE SI ADAPTARE IN TIMP

Un motiv esential pentru care sistemul merita analizat separat este faptul ca nu acorda puncte in acelasi fel pentru toti utilizatorii si nici pentru acelasi utilizator in orice moment. In multe aplicatii, recompensarea este uniforma: cine bifeaza activitatea primeste aceeasi valoare, indiferent daca executarea a fost usoara sau extrem de dificila. Modelul analizat respinge aceasta uniformizare si porneste de la ideea ca aceeasi activitate poate avea costuri psihologice radical diferite pentru doua persoane diferite.

Aceasta personalizare nu este una ornamentala, ci structurala. In primele 30 de zile, vectorul initial V modifica atat configurarea aplicatiei, cat si punctajul. Ulterior, sistemul se recalibreaza pe baza lui G si P . Cu alte cuvinte, personalizarea nu se reduce la recomandari de interfata sau la tonul mesajelor, ci intra direct in nucleul matematic al recompensei. Acest lucru este neobisnuit si foarte valoros in raport cu sistemele clasice de habit tracking.

Adaptarea in doua faze rezolva si o problema metodologica importanta. Daca sistemul s-ar baza numai pe profilul initial, ar risca sa ramana prizonierul unei fotografii de inceput. Daca s-ar baza numai pe date recente, ar ignora dificultatile reale ale unui utilizator nou, exact in momentul in care acesta are nevoie de cel mai mare sprijin. Prin impartirea utilizarii in doua etape, algoritmul rezolva simultan nevoia de sprijin initial si nevoia de acuratete ulterioara.

Relevant este si faptul ca personalizarea nu urmareste sa fie neutra axiologic. Sistemul analizat favorizeaza in mod deliberat comportamentele benefice pentru reglare: somn adecvat, miscare, consecventa, fragmentare inteligenta, confirmare valida a task-ului. Asadar, nu este doar un model descriptiv, ci unul orientat catre schimbare comportamentala functionala. Aceasta orientare controlata il apropie de un instrument de interventie, nu doar de unul de evidenta.

Mai simplu spus, modelul porneste de la doua adevaruri simultane: oamenii sunt diferiti, iar aceeasi persoana se modifica in timp. Un sistem care integreaza ambele realitati in formula de calcul este, in mod firesc, mai nuantat decat unul care trateaza comportamentul uman ca pe o succesiune de bifari standardizate.

17. JEREMY BENTHAM SI HEDONIC CALCULUS

Pentru a evidenta mai clar specificul sistemului analizat, este utila raportarea lui la una dintre cele mai cunoscute incercari istorice de cuantificare a valorii unei actiuni: hedonic calculus, numit si calculul felicific, asociat lui Jeremy Bentham. In forma sa clasica, acest model urma sa evalueze placerea sau durerea produsa de o actiune prin mai multi factori, printre care intensitatea, durata, certitudinea, proximitatea, fecunditatea, puritatea si extensiunea efectelor.

Hedonic calculus a fost gandit ca un instrument de evaluare morala in cadrul utilitarismului. Scopul lui era sa ofere un mod de comparare a actiunilor in functie de cantitatea de placere sau durere pe care o produc. Prin urmare, unitatea de analiza nu era task-ul cotidian al unei persoane concrete, ci consecinta morala mai larga a unei actiuni. Modelul lui Bentham avea o mare valoare filosofica, deoarece incerca sa aduca rigoare in evaluarea deciziilor umane, dar ramanea in mare masura abstract si dificil de operationalizat cu precizie in contexte individuale reale.

In comparatie cu modelul analizat, diferenta initiala este de scop. Bentham cauta un instrument pentru evaluarea utilitatii morale generale a actiunilor. Modelul analizat propune un mecanism de cuantificare functionala a sarcinilor individuale, destinat unei aplicatii care trebuie sa influenteze comportamentul de zi cu zi. Cu alte cuvinte, hedonic calculus este in primul rand normativ si filosofic, in timp ce sistemul analizat este operational, algoritmic si aplicat.

Mai exista o diferenta majora: la Bentham, unitatea centrala este placerea sau durerea, iar aceasta este evaluata in termeni relativ generali. In modelul analizat, placerea sau valoarea emotionala nu dispar, dar ele reprezinta doar o componenta a calculului. Impactul emotional I este pus alaturi de efort, profil, somn, miscare, progres si validare comportamentala. Astfel, sistemul analizat nu se reduce la hedonia resimtita, ci integreaza si constrangerile executive si functionale ale vietii reale.

Prin urmare, comparatia cu Bentham este foarte utila. Ea arata ca sistemul analizat apartine aceleiasi familii mari de idei care incearca sa cuantifice valoarea unei actiuni, dar muta aceasta ambitie din zona filosofiei morale generale in zona unei inginerii comportamentale aplicate, mult mai apropiata de realitatea utilizatorului concret.

18. COMPARATIA DIRECTA CU HEDONIC CALCULUS

Criteriau	Hedonic calculus - Bentham	Sistemul propus
Scop	Evaluarea utilitatii morale a actiunilor	Evaluarea functionala si motivationala a task-urilor cotidiene
Unitate de analiza	Placere / durere produsa de o actiune	Task concret introdus in aplicatie
Factori centrali	Intensitate, durata, certitudine, proximitate, fecunditate, puritate, extensiune	Efort, impact emotional, profil initial, somn, miscare, progres, randomizare
Tip de model	Normativ, filosofic, abstract	Algoritmice, operational, adaptiv
Personalizare	Foarte limitata	Ridicata, pe baza profilului si comportamentului real

Prima diferenta directa este ca modelul lui Bentham nu a fost creat pentru a recompensa sau ghida o persoana concreta in timp real, ci pentru a judeca valoarea unei actiuni din perspectiva consecintelor sale globale. Sistemul analizat este, in schimb, construit exact pentru acest tip de interactiune imediata. El produce scoruri care pot fi afisate instant, arhivate, comparate si transformate in recompense behaviorale.

A doua diferenta este legata de date. Hedonic calculus nu presupunea acces la semnale concrete precum somnul, numarul de pasi, frecventa task-urilor finalizate sau dovada foto. Sistemul analizat functioneaza pe baza unor variabile observabile si masurabile digital. Aceasta il face nu doar mai precis in contextul aplicatiei, ci si semnificativ mai verificabil.

A treia diferenta este legata de structura recompensei. In modelul lui Bentham, evaluarea nu era conceputa ca mecanism motivational integrat intr-un ciclu de feedback zilnic. In modelul analizat, punctajul rezultat este gandit pentru a sustine initierea, finalizarea si repetarea comportamentelor benefice. Altfel spus, Bentham masoara pentru a judeca, in timp ce sistemul analizat masoara pentru a regla, a stimula si a construi consecventa.

Aceasta comparatie arata clar ca sistemul analizat nu este o simpla reeditare moderna a calculului felicific, ci o transformare a ideii de cuantificare a valorii intr-un instrument mult mai specific, mai contextual si mai tehnic.

19. DE CE SISTEMUL PROPUȘ ESTE MAI APROPIAT DE REALITATE

Sistemul propus este mai apropiat de realitate deoarece pleacă de la modul în care oamenii funcționează efectiv, nu de la un model idealizat al deciziei. În viața cotidiană, valoarea unei acțiuni nu depinde doar de plăcerea produsă sau de utilitatea ei teoretică, ci și de cât de greu este să fie inițiată, de resursele disponibile în acel moment, de energia mentală, de somn, de obiceiurile anterioare și de capacitatea persoanei de a susține o rutină. Toate aceste elemente apar, într-o formă sau alta, în formula ta.

Un al doilea motiv este acela că sistemul analizat acceptă explicit variabilitatea umană. În realitate, aceeași persoană nu funcționează identic în fiecare zi. Uneori are energie, alteori nu; uneori doarme bine, alteori prost; uneori are ritm bun de lucru, alteori cade într-o perioadă de dezorganizare. Formula care folosește G și P după prima lună este mult mai aproape de această realitate oscilantă decât orice model fix, indiferent cât de elegant ar fi filosofic.

Sistemul este mai realist și pentru că admite tensiunea dintre efort și valoare emoțională. Există sarcini mici cu efect major și sarcini mari cu valoare redusă. Tocmai de aceea introducerea lui I lângă E este esențială. În practică, oamenii nu își organizează viața doar în funcție de cantitatea de muncă, ci și în funcție de ceea ce activitatea respectivă rezolvă sau stabilizează în plan psihologic și funcțional.

Mai mult, modelul analizat include mecanisme de control al exploatarei sistemului: reducerea punctelor la repetarea excesivă a aceluiași task, solicitarea foto-dovezii, bonusuri condiționate și randomizare controlată. Aceste elemente nu sunt accesorii. Ele îl apropie de realitate tocmai pentru că recunosc faptul că utilizatorii nu interacționează întotdeauna perfect sincer sau perfect constant cu o aplicație.

Din acest motiv, afirmația că sistemul este mai apropiat de realitate nu este doar retorică. Ea poate fi susținută concret: modelul include mai multe tipuri de date, acceptă variația temporală, recompensează consecvența și evită tratamentul mecanic al task-urilor. Toate acestea îl fac mai plauzibil și mai utilizabil decât un sistem abstract centrat exclusiv pe plăcere sau pe bifarea simplă a activităților.

20. FEZABILITATEA SI AMPLOAREA PRACTICA A SISTEMULUI

Fezabilitatea sistemului rezulta in primul rand din faptul ca fiecare componenta poate fi implementata cu instrumente deja disponibile sau realist de obtinut intr-o aplicatie moderna. Dificultatea perceputa este introdusa direct de utilizator. Dificultatea estimata, energia si impactul emotional pot fi inferate de un model AI sau de reguli euristice initiale. Timpul, complexitatea si subtask-urile sunt date structurale ale task-ului. Somnul si activitatea telefonului pot fi estimate prin senzori si evenimente de utilizare. Numarul de pasi poate fi preluat din platformele mobile. Progresul lunar se obtine direct din istoricul intern al aplicatiei.

Fezabilitatea este sustinuta si de simplitatea relativa a formulelor. Desi modelul include multe componente, fiecare dintre ele este usor de calculat. Exista intervale finite, normalizari simple si ponderi fixe. Nu este nevoie de modele matematice opace sau de infrastructuri imposibil de intretinut. Mai mult, sistemul poate fi lansat incremental: initial cu reguli mai simple pentru I sau energie, apoi rafinat pe baza datelor colectate.

Importanta practica a modelului vine din faptul ca muta cuantificarea task-urilor din zona superficiala a productivitatii in zona unei inginerii comportamentale sensibile la realitatea neurodivergentei. El nu se limiteaza la formula faci ceva si primesti puncte. In schimb, urmareste ce a fost facut, cat a costat, ce efect produce, in ce stare se afla utilizatorul, cat de constant a fost si cum poate sistemul sa sustina motivatia fara sa devina usor de manipulat. O asemenea profunzime este rar intalnita in aplicatiile curente.

Daca aceasta contributie este formulata cu prudenta academica, importanta ei poate fi sustinuta prin trei motive. Primul: sistemul uneste psihologia motivatiei cu arhitectura software si cuantificarea explicabila. Al doilea: trateaza utilizatorul ca pe un proces dinamic, nu ca pe un profil fix. Al treilea: transforma recompensa din simplu ornament vizual intr-un mecanism functional, capabil sa influenteze comportamentul in timp.

Prin urmare, fezabilitatea si amploarea practica nu se exclud. Dimpotriva, tocmai faptul ca modelul este in acelasi timp articulat conceptual si implementabil il face puternic. Exista multe idei teoretic impresionante care nu pot fi construite. Exista si multe produse care pot fi construite, dar sunt conceptual sarace. Sistemul analizat are sansa rara de a sta la intersectia dintre cele doua categorii.

21. EXEMPLE CONCRETE DE TASK-URI - PARTEA I

Exemplul 1: spalarea vaselor dupa masa

Pentru acest task se pot stabili urmatoarele valori: dificultate perceputa = 1, dificultate estimata = 1, timp = 1, complexitate = 1, energie = 1. Rezulta $E = 5$. Impactul emotional imediat poate fi 4, iar impactul pe termen lung 8, deci $I = 12$. Astfel, $T = 17$, iar $PT = 0,55 \times (17/52) = 0,1798$. In primele 30 de zile, pentru un utilizator cu $V_i = 4$, $PV_i = 0,2333$. Daca $r = 2$, $Pr = 0,0167$. Scorul de baza este $R = 10 \times (0,1798 + 0,2333 + 0,0167) = 4,30$ puncte. Task-ul primeste un scor moderat, desi este simplu, pentru ca produce un efect organizator real.

Exemplul 2: programarea unei consultatii medicale

Acest task poate parea scurt, dar functional este foarte valoros. Se considera urmatoarele valori: dificultate perceputa = 3, dificultate estimata = 1, timp = 1, complexitate = 1, energie = 2. Rezulta $E = 8$. Impactul imediat poate fi 7, iar impactul pe termen lung 18, deci $I = 25$. $T = 33$, $PT = 0,3490$. Cu $V_i = 5$ si $r = 3$, scorul devine $10 \times (0,3490 + 0,2917 + 0,0250) = 6,66$ puncte. Acest exemplu arata foarte clar de ce modelul analizat este superior unui sistem care puncteaza doar dupa durata: o activitate scurta poate avea o valoare functionala majora.

Exemplul 3: 20 de minute de invatare ghidata

Se considera urmatoarele valori: dificultate perceputa = 2, dificultate estimata = 2, timp = 3, complexitate = 3, energie = 3. $E = 13$. Impactul imediat poate fi 4, iar cel pe termen lung 8, deci $I = 12$. $T = 25$, $PT = 0,2644$. In etapa a doua, daca $G = 5$, $P = 5$ si $r = 4$, scorul devine $10 \times (0,2644 + 0,2083 + 0,0714 + 0,0333) = 5,77$ puncte. Daca task-ul este validat cu foto-dovada si primeste bonus de 10%, scorul final urca la aproximativ 6,35 puncte.

Cele trei exemple arata deja un principiu important: sistemul analizat nu recompensa doar munca grea, ci si actiunile mici cu efect mare, precum si sarcinile de durata medie care sprijina consecventa. Acesta este unul dintre motivele pentru care modelul poate functiona bine pentru utilizatori reali.

22. EXEMPLE CONCRETE DE TASK-URI - PARTEA A II-A

Exemplul 4: curatarea completa a camerei

Pentru o astfel de activitate se pot stabili urmatoarele valori: dificultate perceputa = 3, dificultate estimata = 2, timp = 5, complexitate = 6, energie = 5. $E = 21$, adica maximul teoretic. Impactul imediat poate fi 13, iar impactul pe termen lung 18, deci $I = 31$, maximul teoretic. Rezulta $T = 52$ si $PT = 0,55$. Daca utilizatorul este in primele 30 de zile si $V_i = 6$, iar $r = 5$, scorul de baza este $10 \times (0,55 + 0,35 + 0,0417) = 9,42$ puncte. Aceasta este o valoare mare si justificata, deoarece task-ul este simultan costisitor si extrem de util.

Exemplul 5: plimbare de 15 minute in aer liber

Pentru acest task se pot stabili urmatoarele valori: dificultate perceputa = 1, dificultate estimata = 1, timp = 3, complexitate = 1, energie = 2. $E = 8$. Impactul imediat poate fi 7, iar impactul pe termen lung 8, deci $I = 15$. $T = 23$, $PT = 0,2433$. In etapa a doua, daca utilizatorul tocmai are somn bun si activitate fizica buna, $G = 5$ si $P = 3$, iar $r = 2$, scorul devine $10 \times (0,2433 + 0,2083 + 0,0429 + 0,0167) = 5,11$ puncte. Activitatea este relativ scurta, dar sistemul o trateaza ca pe o interventie benefica de reglare, nu ca pe o simpla plimbare fara valoare.

Exemplul 6: trimiterea unui e-mail dificil

Un astfel de task poate avea dificultate perceputa = 3, dificultate estimata = 1, timp = 1, complexitate = 1, energie = 2. $E = 8$. Impactul imediat poate fi 7, iar impactul pe termen lung 8, deci $I = 15$. $T = 23$, $PT = 0,2433$. In prima etapa, cu $V_i = 5$ si $r = 1$, scorul devine $10 \times (0,2433 + 0,2917 + 0,0083) = 5,43$ puncte. Desi durata este mica, scorul este suficient de bun pentru a reflecta dificultatea executiva si eliberarea de tensiune pe care o produce finalizarea sarcinii.

Aceste exemple confirma faptul ca sistemul analizat distribuie recompensa intr-un mod sensibil la realitatea traita a utilizatorului. El nu cade in capcana de a supraevalua doar sarcinile lungi sau, invers, de a recompensa nediferentiat activitatile scurte. Valorile se schimba in functie de impactul real si de contextul de executie.

23. EXEMPLE CONCRETE DE TASK-URI - PARTEA A III-A SI CONCLUZIE

Exemplul 7: repetarea excesiva a aceluasi task

Se considera un task foarte mic de tip verifica telefonul si raspunde la un mesaj simplu, cu $E = 5$, $I = 4$ si $T = 9$. In mod normal, scorul ar putea fi decent pentru un task usor. Totusi, daca acelasi task este repetat consecutiv de doua ori doar pentru acumulare artificiala de puncte, regula de penalizare il injumatatesteste. La a treia repetare, utilizatorul nu mai primeste puncte si suporta o penalizare suplimentara. Acest exemplu arata de ce sistemul analizat este fezabil si robust: el nu doar recompenseaza, ci si filtreaza tentativele de exploatare.

Exemplul 8: task asistat cu fragmentare si succes consecutiv

Se considera un task de tip incepe proiectul de licenta, initial blocant pentru utilizator. Modul asistat il fragmenteaza in trei pasi: deschide fisierul, scrie titlul, redacteaza primul paragraf. Dificultatea perceputa scade, iar probabilitatea de initiere creste. Daca utilizatorul finalizeaza trei task-uri consecutive in modul asistat in decurs de o ora, al patrulea poate primi bonusul de +10% descris in lucrare. Aceasta situatie ilustreaza foarte bine una dintre fortele sistemului: algoritmul nu doar masoara succesul, ci creeaza si conditiile in care succesul devine mai probabil.

Per ansamblu, exemplele concrete confirma trei idei. Prima: sistemul este fezabil, deoarece poate traduce in scoruri numerice o gama larga de task-uri reale. A doua: sistemul este mai apropiat de realitate, deoarece diferentiaza intre activitati mici dar importante, sarcini lungi si dificile, activitati de reglare si comportamente repetitive fara valoare functionala. A treia: sistemul ofera o modalitate mai clara de a transforma o problema vaga - cum este masurata corect valoarea unui task pentru o persoana cu dificultati executive - intr-o structura operationala coerenta, adaptiva si implementabila.

Concluzia generala este ca modelul propus depaseste atat sistemele clasice de productivitate, cat si modelele filosofice abstracte de evaluare a valorii actiunilor. El este suficient de riguros pentru a fi sustinut teoretic, suficient de flexibil pentru a se adapta utilizatorului si suficient de concret pentru a fi integrat intr-un produs digital real. Tocmai aceasta combinatie explica forta practica a sistemului analizat.

24. VALIDARE EMPIRICA SI DIRECTII VIITOARE

Un argument suplimentar in favoarea sistemului este acela ca poate fi validat empiric in mod relativ clar. Fiind construit din variabile explicite si intervale finite, modelul permite testarea relatiei dintre scorul calculat si comportamentele observabile ale utilizatorului: rata de finalizare, consecventa pe 30 de zile, abandonul task-urilor, revenirea dupa perioade de blocaj, raspunsul la bonusuri sau sensibilitatea la degradarea somnului. Cu alte cuvinte, sistemul nu este doar un construct teoretic elegant, ci unul care poate fi pus la proba in conditii reale.

Validarea se poate face etapizat. Intr-o prima faza, se poate urmari daca scorurile mai mari corespund intr-adevar task-urilor pe care utilizatorii le percep drept mai importante si mai utile. Intr-o a doua faza, se poate analiza daca tranzitia de la formula initiala la formula dinamica imbunatateste aderenta si reduce abandonul aplicatiei. Intr-o a treia faza, se poate calibra ponderea componentelor pe baza datelor colectate, fara a renunta la arhitectura de baza.

Aceasta posibilitate de validare este importanta si in comparatia cu modele abstracte precum hedonic calculus. Un sistem care poate fi ajustat pe baza comportamentului real are o sansa mult mai mare de a deveni util si sustenabil pe termen lung. Mai mult, el poate evolua fara sa-si piarda coerenta, deoarece formulele de baza raman explicabile chiar si atunci cand unele praguri sau ponderi sunt rafinate.

Ca directie viitoare, sistemul poate integra si alti indicatori, de exemplu ritmul cardiac, nivelul perceput de stres, regularitatea ciclului circadian sau date provenite de la smartwatch. Totusi, chiar si fara aceste extensii, modelul actual are deja suficienta profunzime pentru a functiona convingator. Aceasta combinatie dintre robustete actuala si potential de dezvoltare ulterioara intareste ideea ca sistemul este fezabil, modern si relevant pentru cercetare aplicata.

In concluzie, privit ca baza teoretica pentru un produs real, modelul isi arata forta nu doar prin noutate, ci si prin capacitatea de a fi verificat, rafinat si sustinut prin date. Acesta ramane, probabil, cel mai puternic argument final in favoarea sistemului propus.