

Productivitatea muncii în sistemul social complex al dezvoltării de software

Ec. Otilia PÂRLOG, mat. Adrian PÂRLOG, ing. Gheorghe NOȘCA
Ministerul Apărării Naționale

Ec. Petrișor OPREA

Catedra de Informatică Economică, A.S.E., București

Cercetarea productivității muncii în sisteme sociale complexe reprezintă un domeniu nou în industria software. Sistemul social analizat prezintă anumite deficiențe, care sunt prezentate în lucrare. Se realizează, de asemenea, formalizarea productivității în acest sistem.

Cuvinte cheie: *productivitatea muncii, microsistem social, simulare.*

1. Introducere

Începând cu anii '70, organizațiile producătoare de software au investit într-o mulțime de inovații tehnologice, cum ar fi limbajele de generația a patra, produsele CASE, analiza și programarea orientate pe obiect, software complet reutilizabil, care sunt destinate să mărească productivitatea în acest domeniu. Cheltuielile pe uneltele de dezvoltare a aplicațiilor software au fost în mod constant estimate să aibă o creștere de 19% anual conform [5], transformându-le în unul din segmentele fierbinți ale industriei de software. Însă, spre deziluzia atât a managerilor de proiecte software cât și a oamenilor de afaceri care le utilizează, nivelul productivității în dezvoltarea de software a rămas în urma productivității realizate în multe discipline socio-economice ajunse la maturitate. Stagnarea productivității este deseori provocată de spirala costurilor, timpul excesiv pentru vindere și neînțelegerile ivite cu utilizatorii, atât pe timpul proiectării cât și al exploatării produselor software. De aceea nu este de mirare că într-o eră a consolidării creșterii nivelurilor de productivitate, productivitatea software să se situeze în vârful listei de obiective a managerilor în activitatea de dezvoltare de software.

2. Mecanismul dezvoltării de software - microsistem social complex

Explicațiile convenționale învinuiesc stagnarea productivității în industria de dezvoltare de software prin slaba calitate a conducerii efective. Se poate demonstra că există mai

multe cauze sistemice, comensurabile într-un mod intuitiv, pentru paradoxul productivității. Explicația acestor cauze constă în aceea că potențialul de productivitate a uneltelor de inginerie software ar putea fi irosit, nu datorită greșelilor organizatorice pentru a institui practicile de conducere corespunzătoare, ci pentru că mecanismul de dezvoltare a software-ului este un sistem social complex care conduce la situații cu consecințe diferite de cele estimate. O astfel de concluzie este perfect valabilă și atunci când se are în vedere creșterea productivității înregistrate ca urmare a aplicării tehnologiei CASE.

O retrospectivă asupra literaturii de specialitate sugerează că mii de proiecte de dimensiuni medii și mari se reîntorc la tehnologia CASE, pe baza intuiției că aceasta este calea de urmat pentru îmbunătățirea productivității. Cheltuielile anuale în această direcție sunt raportate a fi în jur de 100 milioane USD și sunt într-o creștere rapidă. La prima vedere, imaginea contribuției CASE pare eterogenă. Pe de o parte, literatura abundă cu căi extravagante pentru creșterea potențialului productiv al CASE-ului, iar pe de altă parte, există o mulțime de rapoarte referitoare la eșecuri majore.

Totuși, o privire din interior relevă o separare interesantă. Studiile de laborator stabilesc că folosirea proiectelor pilot relativ mici conduce la raportarea unor creșteri spectaculoase ale productivității, cu 100%-600%. Când impactul productivității tehnologiei CASE este evaluat în condiții reale, se înregistrează creșteri mult mai modeste ale productivității (15%-30%) sau deloc.

Variația dimensiunilor proiectelor, măsurate în features (caracteristici) [7]

Activități	Unități de măsură	Proiect mic: 5 caracteristici		Proiect mare: 80 caracteristici	
		Fără instrumente	Cu instrumente	Fără instrumente	Cu instrumente
Captarea de noi cerințe	minute	636	162	2976	432
Rezolvarea cerințelor noi	minute	156	303	34958	67861
Rezolvarea problemei	minute	120	35	7804	2291
Rezolvare caracteristici	minute	487	188	7792	3016
Dezvoltarea planului de realizare	minute	180	68	2880	1088
Efortul total	minute	1579	757	56411	74688
Productivitatea		7.6	15.8	3.4	2.6
Impactul productivității			+107.9%		-23.5%

Productivitatea este calculată cu formula:

$$W = \frac{\text{Număr_caracteristici}}{\text{Efort_total (minute)}} * \frac{2400 \text{ minute}}{1 \text{ săptămână}} \quad (1)$$

Studiind tabelul, se observă că pentru proiecte mici, de maxim 5 caracteristici,

valoarea productivității se dublează, plecând de la 7.6 în cazul neutilizării de tools-uri și ajungând la 15.8 după folosirea acestora. Pentru probleme de dimensiuni mari (80 features) productivitatea va scădea de la 3.4 la 2.6, odată cu creșterea efortului total de la 56411 la 74688.

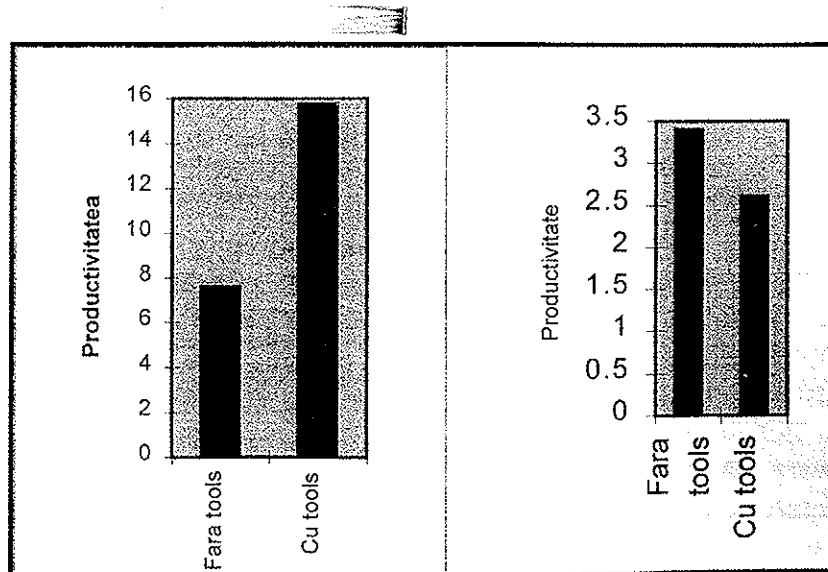


Fig. 1. Productivitatea în cazul proiectelor mici și a proiectelor mari

Este de așteptat să se concluzioneze că aceste diferențe dintre rezultatele studiilor de laborator referitoare la potențialul de productivitate a tehnologiei CASE și rezultate-

le obținute în sistemele reale se datorează conducerii defectuoase. Cercetările întreprinse în sistemele dinamice sugerează faptul că principala cauză pentru paradoxul

productivității derivă din complexitatea sistemelor sociale.

3. Comportarea sistemelor sociale

Cercetările din ultimele decenii relevă faptul că sistemele sociale aparțin clasei sistemelor cu feedback neliniare cu legături multiple, aceeași clasă care înglobează sistemele tehnologice cele mai complicate, cum ar fi rafinăriile chimice, rețelele de comunicații și pilotarea automată. Inginerii au apreciat că structura complexă de tip feedback este adesea o sursă a împrăstierii cercetării și că această singură judecată intuitivă este inoperantă pentru estimarea modului în care sistemele se vor modifica în timp, chiar cu o bună cunoaștere a structurii interne a lor. În sistemele ingineresti descrise se folosesc pentru anticiparea stării sistemului metodele avansate ale analizei dinamice și modelarea pe calculator. Prin contrast, în descrierea și conducerea sistemelor sociale, managerii și politicienii continuă să se bazeze pe intuiție și observație, fără nici o analiză dinamică adecvată pentru a preveni apariția consecințelor nedorite. Nu e de mirare că sistemele sociale par diabolice deoarece făcând un lucru evident, cel mai adesea nu se obține ieșirea așteptată. Se poate vedea aceasta în multe programe guvernamentale la care se obțin rezultate ineficiente sau, mai rău, opusul la ceea ce s-a intenționat.

4. "Microuniversul" dezvoltării de software

În ultimele trei decenii, progresele înregistrate în cercetarea sistemelor dinamice au demonstrat atât fezabilitatea cât și utilitatea construcției modelelor pe calculator pentru a studia sistemele sociale complexe. Prin analogie cu modelul avionului de luptă al inginerului aeronautic dintr-un simulator aerodinamic și modelul vaporului proiectantului naval dintr-un simulator hidraulic, un "microunivers" este un sistem dinamic

simulat, o replică pe calculator a modului de structurare a realității. Folosind această unealtă, managerii, ca și inginerii, au la dispoziție un laborator în care să simuleze starea sistemului și pot conduce experimentele înaintea implementării de noi programe în realitate.

Un microunivers care să simuleze dezvoltarea de software este un instrument extrem de valoros, deoarece cercetarea pe un microcolectiv real este deficitară, putând conduce la costuri considerabile de timp sau bani, fără să se poată experimenta un număr suficient de variante, astfel încât rezultatele cercetării să fie relevante și să se indice cu claritate strategia de urmat. Când experimentarea cu un sistem real este imposibil de a fi realizată intervine simularea, în principal în acele cazuri în care aceasta reprezintă singura cale pentru studierea modului de lucru al sistemelor complexe.

5. Definierea productivității microuniversului

Să considerăm un subsistem de dezvoltare de software (figura 2). Acesta reprezintă modulul central asupra căruia își exercită influența alte trei module: subsistemul conducerii resurselor umane, subsistemul de planificare și subsistemul de control.

Structura productivității (figura 3) este bazată pe modelul productivității grupului, care stabilește că:

$$\text{Productivitatea efectivă} = \text{Productivitatea potențială} - \text{Pierderile datorate proceselor defectuoase} \quad (2)$$

unde *productivitatea potențială* este nivelul productivității atins când un individ sau un grup realizează cea mai bună utilizare a resurselor disponibile. Ea este funcție de categoria lucrării abordate (tipul proiectului, complexitatea produsului, limbajul de programare) și resursele echipei (uneltele software, facilitățile hardware).

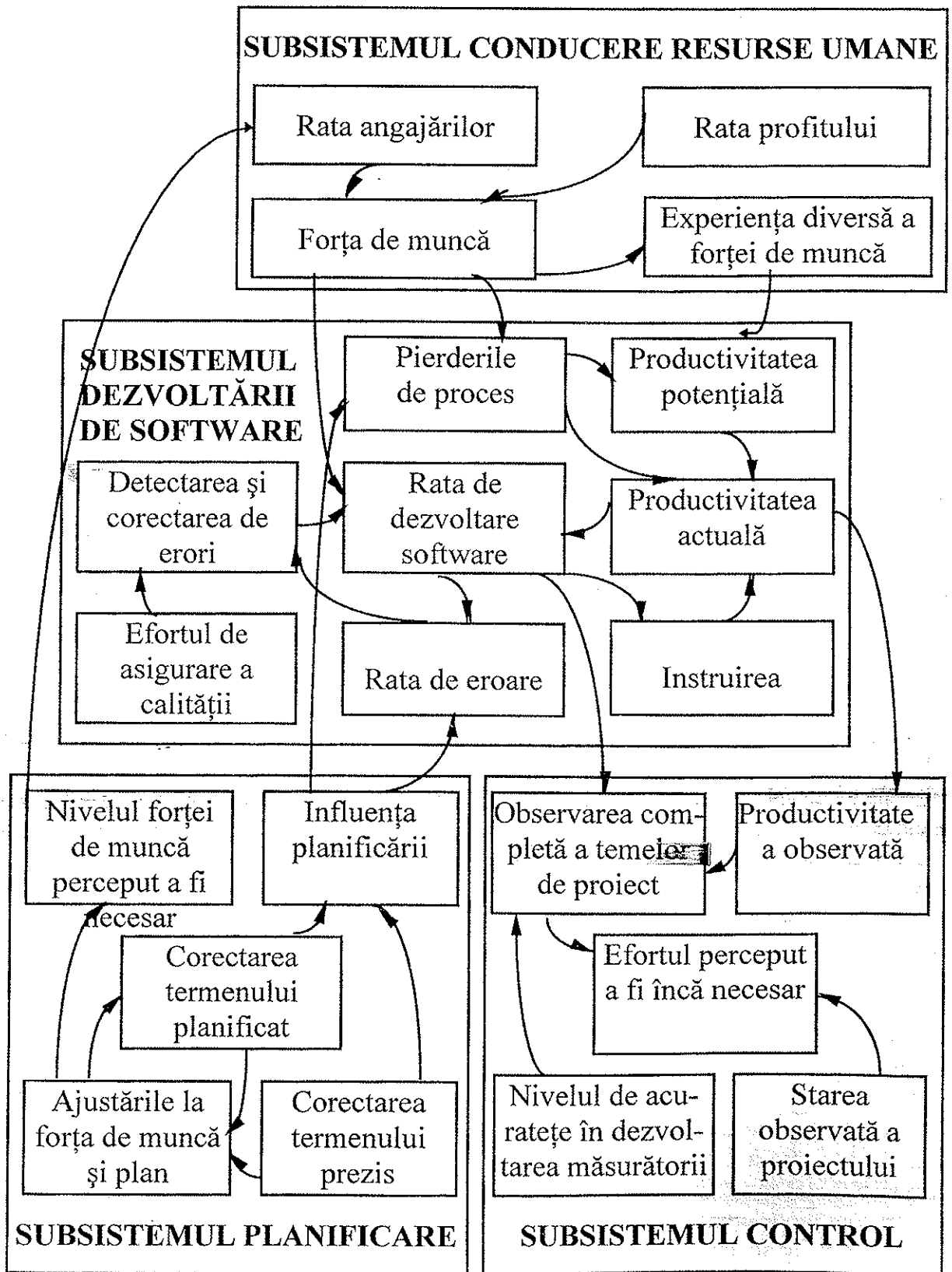


Fig. 2. Structura simplificată a microuniversului de dezvoltare de software.
Cele 4 subsisteme au mai mult de 100 de legături cauzale.

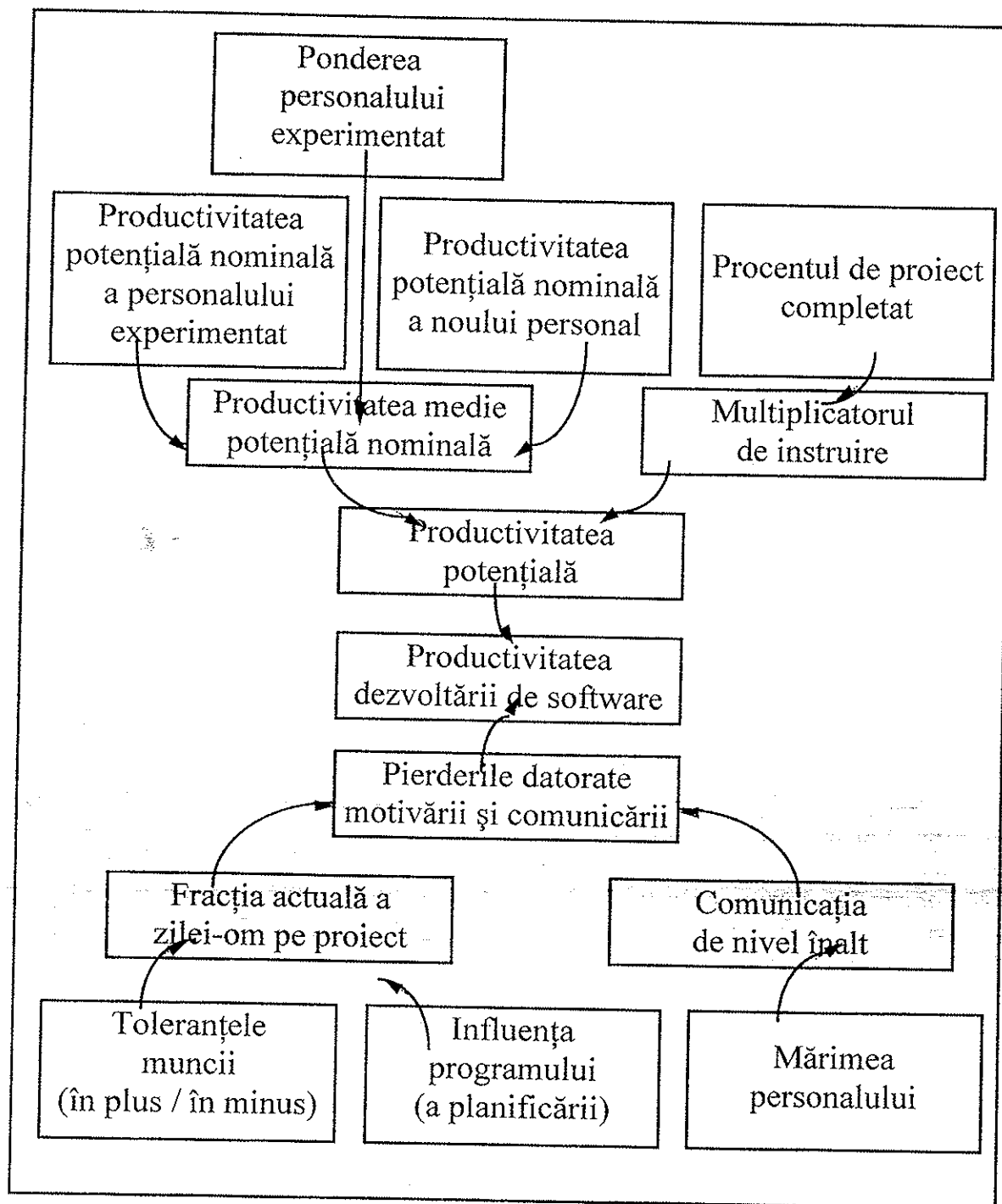


Fig. 3. Structura productivității microuniversului de dezvoltare de software.

Productivitatea potențială nominală medie a echipei este media ponderată a două productivități potențiale nominale, una pentru conducerea experimentată și una pentru conducerea nou apărută, corespunzătoare personalului cu experiență în abordarea lucrării și celui adăugat ca urmare a corecțiilor aduse prin intermediul subsistemului de planificare a forței de muncă implicate. Productivitatea potențială nominală medie are influențe puternice asupra ciclului de viață a proiectului. Microuni-

versul atrage efectele instruirii personalului. Pe măsura progresării în abordarea temei, echipa ajunge să cunoască mai bine lucrarea și productivitatea crește.

Productivitatea efectivă rareori egalează productivitatea potențială datorită pierderilor înregistrate în procesele defectuoase. În microunivers, pierderile sunt un multiplicator care în mod efectiv diminuează fracția productivității medii a unei zi-persoană. De exemplu, dacă pierderile medii sunt în jurul a 4 ore-persoană, iar ziua-persoană pentru

timpul complet al unui angajat este de 8 ore, valoarea multiplicatorului este de 0,5 [6].

6. Factori de diminuare a productivității software

Procesele defectuoase referite în ecuația productivității (2) sunt în mod obișnuit datorate factorilor motivați dinamici și comunicației de nivel înalt. Mulți factori de motivație, cum ar fi posibilitatea de promovare, nivelul responsabilității și salariul, nu sunt de natură dinamică. Ei rămân constanți pe durata vieții proiectului și, în mod implicit, pot fi încorporați în valoarea determinată pentru productivitatea potențială.

Un exemplu al unui factor de motivație dinamic este influența planificării. Ea este destul de importantă, trăgând fracția efectivă a productivității individuale în sus sau în jos, dacă influența este pozitivă, respectiv negativă. Astfel, când membrii unui proiect înțeleg că sunt în urma programului planificat, pot fi doritori să muncească mai intens (totuși nu într-un mod nedefinit) pentru a readuce proiectul în program. Pe de altă parte, când ei percep că există excese în planificarea termenelor sau în bugetul alocat, tind să absoarbă părți ale acestor excese prin plată îmbunătățită și/sau diminuarea intensității muncii. Dincolo de aceste limite, excesele sunt translatate în mod normal în reduceri ale bugetului alocat sau ale termenelor proiectului.

Pierderile adiționale la productivitate sunt generate de comunicațiile de nivel înalt. Când o echipă dezvoltă un produs software, fiecare individ trebuie să consume o parte din timp comunicând cu alți membri ai echipei pentru a împărtăși experiența, a explica deciziile, a răspunde la întrebările despre modulele de proiect pe care le elaborează și așa mai departe. Aceste comunicații sunt esențiale pentru finalizarea proiectului, fiecare cale de comunicație creată translatându-se într-un efort adițional și, în con-

secință, timp suplimentar. În microuniversul creat, astfel de comunicații reprezintă pierderi în productivitatea potențială a unei persoane.

Concluzii

Pentru o corectă planificare și conducere a dezvoltării de software se poate recurge la folosirea unui sistem dinamic minimizat pentru a simula tendința productivității pe termen lung într-un mecanism - proiect ipotetic condus științific. Minimizarea permite echipei proiectantului să examineze cauzele posibile una câte una, prin experimentarea controlată, și să identifice cauzele care duc la diminuarea productivității.

Bibliografie

1. L. Fried - "Team Size and Productivity in Systems Development", J. Information Systems Management, Summer, 1991.
2. I. Ivan, P. Oprea - "Particularitati ale evaluării productivității muncii în activitatea de programare", Revista Româna de Statistica, nr. 11, 1996
3. I. Ivan, A. Balog, S. Coman - "Aspecte ale productivității muncii în activitatea de programare", Buletinul Român de Informatică, nr. 6, 1981
4. P. Oprea - "Modeling labour productivity in software development activity", Proceedings of student's research in Economic Informatics, Bucharest 1997
5. O. Pârlog, Gh. Noșca - "Software Quality Cost Structure", The Proceedings of the 3rd International Symposium of Economic Informatics, Bucharest, Mai 1997
6. G. Richardson - *Feedback Thought in Social Science and Systems Theory*, University of Pennsylvania Press, Philadelphia, 1991.
7. I. Steiner - *Group Process and Productivity*, Academic Press, New York, 1972.