

В.А. Федорієнко

Національний університет оборони України ім. І. Черняхівського, Київ

## ВПЛИВ ЕФЕКТУ ЧАСОВИХ ПОКАЗНИКІВ ЗАВАНТАЖЕНОСТІ НА ОПТИМІЗАЦІЮ СТВОРЕННЯ СПЕЦІАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ERP СИСТЕМИ

*Проведений аналіз контролю працездатності по часовим показникам ERP системи та розкриті основні з них. Запропоновано новий структурний підхід ієрархічної побудови взаємозв'язків часових показників завантаженості системи. Визначені суб'єкти створення та модифікації спеціального програмного забезпечення. Досліджено характер зміни значень конкретних часових показників при діяльності програмістів різної кваліфікаційної класності. Запропоновано, прийняти деякі часові показники з граничними межами в якості ефекту працездатності для побудови оптимізаційної залежності класності суб'єктів створення та модифікації спеціального програмного забезпечення в умовах ресурсних обмежень.*

**Ключові слова:** часові показники завантаженості, класність програмістів, оптимізаційна модель, ієрархія часових показників.

### Вступ

**Постановка проблеми.** Загально відомо, що будь-який програмно-технічний комплекс потребує уваги, щодо підтримки його у працездатному стані. Виробники технологій систем класу ERP (Enterprise Resource Planning) пропонують власне стандартне програмне забезпечення із максимальним наближенням інструментарію та програмних модулів до функціональних потреб використання. Враховуючи специфіку області використання, рівень приведення та підтримки відповідності національних, галузевих, внутрішніх вимог нормативно-правових актів визначає складність проекту. При виконанні різних проектів інформатизації за будь-якої методологією WaterFlow, Cascade, Agile, EP тощо [1–3], що описується різними стандартами управління проектами (ISO 9000, PM Book, ГОСТ 34 та інші) передбачають етапи розробки, впровадження та супроводження. В ході даних робіт здійснюється створення та модифікація програмного коду спеціального програмного забезпечення (СПЗ).

В ERP-системах існує ряд напрямків по проведенню робіт щодо підтримки системи у працездатному стані. Одним із напрямів його контролю є моніторинг системного навантаження, ключовим індикатором якого є часовий фактор. Робота по підтримці працездатності розпочинається з аналізу завантаженості системи на предмет виявлення проблемних областей, після чого отримані дані підлягають поглибленому аналізу для одержання більш детальної інформації [4]. Однією з причин ускладнення працездатності можуть бути дії програмістів при створенні та модифікації програмного коду СПЗ. Часові показники є досить точними індикаторами проблем в системі і гарно відображають рівень пра-

цездатності програм, як системних, так, і програм-розробників. У наслідок діяльності програмістів різних кваліфікаційних класів часовий параметр може досить вдало відобразити їх вплив на ефективність працездатності системи, це підтвердили результати виконання проектів на програмній платформі компанії SAP AG.

Аналіз завантаженості системи за часовими показниками може використовуватися для визначення пріоритетів при усуненні проблем із продуктивністю системи, а також дає можливість ідентифікувати, так звані, “вузькі місця” проходження сигналу, щоб потім швидко усунути такі проблеми, наприклад, шляхом перерозподілу системних ресурсів (додаванням буферної пам'яті, збільшенням/зменшенням кількості процесів тощо) чи відновленням СПЗ шляхом програмування. Для точної локалізації “вузьких місць” необхідно розглянути наявні часові показники, уточнити їх ієрархічну залежність та взаємозв'язок із кроками транзакції по системним компонентам (мережі, серверів застосувань, бази даних).

Тому, існує протиріччя між вимогою до моніторингу, як процедурою швидкого контролю за продуктивністю системи, для подальшої її оптимізації, та можливостями щодо оцінки діяльності програмістів різних кваліфікаційних класів щодо створення та модифікації СПЗ. Тому дослідження питань оптимізації залежності кваліфікації програмістів від часових показників системи в умовах ресурсних обмежень є актуальним питанням.

**Аналіз останніх досліджень і публікацій.** Дослідженню питань контролю продуктивності програмно-технічних комплексів присвячені роботи [5–6]. У [4] зазначається, що оптимізацію продуктивності системи SAP можна вважати задачею фундамен-

тальної важливості і при цьому збільшення часу реакції системи призводить до зростання затрат. Питанням підбору програмістів за кваліфікаційними вимогами розглядалися в [7–8]. У роботі [9] описана модель, яка дозволяє визначити розрахунок зусиль та обсяг робіт на підтримку програмного забезпечення без врахування кваліфікаційних вимог програмістів.

**Мета статті** – аналіз впливу ефекту часових показників завантаженості на оптимізацію процесу створення спеціального програмного забезпечення ERP системи.

## Виклад основного матеріалу

У кожному програмно-технічному комплексі, є вузли в системі, що призводять до ускладнення продуктивності і спричиняють появу “вузких місць”. Вони негативно впливають на пропускну здатність і час відгуку. Такі проблеми продуктивності можуть завдати досить негативний вплив на доступність розроблених програм SAP-систем і тому повинні бути усунуті у найкоротший термін. Одночасно, характер модифікації та створення СПЗ частково впливає на зміну часових показників. Розглянемо приклад роботи підготовленої розробником програми СПЗ на програмно-технічних комплексах SAP ERP в Міністерстві оборони України та в компанії “Fozzy Group”.

Монітор робочого навантаження (workload monitor, транзакція ST03N), як один із найважливіших компонентів управління продуктивністю, є інтегрований засіб SAP, що відображає статистичні дані завантаженості системи, корисний при аналізі найбільш важких для системи транзакцій та звітів, за конкретний період часу [10]. Найбільш важливими параметрами монітору завантаженості, є число транзакційних кроків та середній час відповіді. Через даний монітор відображається навантаження, як загальне робоче, так і розділене за типами задач. Кожному типу задач відповідають показники із певними значеннями часу відповіді.

До типів задач для контролю (моніторингу) системного навантаження відносять наступні процеси виконання: комунікаційних функцій ALE (Application Link Enabling); автоматичної обробки запитів – AUTOABAP; автоматизованого завдання програми обробки – AUTOTN; фонові обробки даних – BACKGROUND; регулярної синхронізації буферу локальних таблиць – BUFFER SYNC; очищення буферної таблиці бази даних – DDLOG CLEANUP; затримки виклику програми обробки – DEL. THCALL; діалогового режиму (он-лайн) – DIALOG; передачі запитів із ICM (Internet Communication Manager) у відповідності інтернет-протоколів (HTTP, HTTPS, NNTP, SMTP, FTP) – HTTP; функцій віддаленого виклику RFC (Remote

Function Call) – RFC; задачі дистанційної програми обробки виклику – RPCTH; друку – SPOOL.

Слід зазначити, що часові показники і кроки транзакції взаємопов’язані і подані у системі по кожному конкретному типу задачі. Наприклад, при існуючому типі діалогового режиму, крок транзакції відповідає зміні екрану користувача; для процесів оновлення та друку – крок транзакції відповідає обробці одного завдання; фоновий процес при обробці одного фонового завдання може виконати один або декілька транзакційних кроків, або кроків транзакції діалогового процесу.

Визначивши часові показники та порівнявши їх між собою можна проаналізувати важливість програм, які займають найбільшу частину часу бази даних, процесорного часу та часу відгуку інших основних компонентів.

Для аналізу завантаженості системи використовують наступні часові показники: середній час відповіді (resp tm); час очікування диспетчером (disp tm); час завантаження контексту (rollin tm); час вивантаження контексту (rollout tm); час завантаження програм (load tm); час бази даних (db tm); час очікування контексту (rollwt tm); час GUI (GUI tm); час блокування (enq tm); час процедур бази даних (DBproc tm); час обробки (proc tm); процесорний час (cpu tm); час мережі (network tm); час користувача (user tm); час виконання циклів обміну (roundtrips tm); часу простоювання (lost tm).

**Середній час відповіді** на крок транзакції для діалогових завдань (Average response Time/Dialog Step) розглядається у системі SAP, як критерій значення для оцінки продуктивності.

Монітор завантаженості дозволяє також виконувати структурний аналіз середнього часу відповіді, шляхом накопичення таких статистичних даних, як протяжність відповіді бази даних, процесорний час тощо [4].

**Час очікування диспетчером** (disp tm) визначається з моменту прийняття диспетчером запиту на обробку, після чого диспетчер намагається знайти вільний робочий процес необхідного типу (діалоговий, відновлення тощо). За умови, якщо робочий процес знайдений – передає йому запит.

Якщо всі робочі процеси необхідного типу зайняті, запит переміщується у чергу диспетчера. Як тільки необхідний робочий процес стає вільним, диспетчер відразу передає запит, що очікував, із черги диспетчера. Отже, час очікування диспетчера (Average Wait Time per Dialog Step) – це час, який був затрачений запитом в очікуванні вільного робочого процесу.

Крім того, існує багато інших не значних часових величин, пов’язаних із процесом очікування, якими можна знехтувати для ієрархічної побудови залежності часових показників.

Протягом кроків транзакції, сервером застосування створюються, а потім зберігаються такі дані, як змінні, внутрішні таблиці, списки екранів тощо. На початку обробки кроку транзакції робочим процесом контекст користувача повинен стати доступним цьому процесу. Ця процедура називається **завантаженням контексту** (rollin tm). Аналогічно, по завершенню обробки кроку транзакції, робочий процес повинен зберегти контекст користувача – виконати **вивантаження контексту** (rollout tm). Середній час вивантаження контексту не включений у середній час відповіді на крок транзакції.

Всі АВАР програми (АВАР класична мова програмування для платформи SAP), які необхідні для виконання кроку транзакції та, які не доступні у буферах серверу застосувань, повинні бути завантажені і, при необхідності, згенеровані за **час завантаження програм** (load tm). Час виконання запиту даних або зміни даних у базі даних системи SAP називається **часом бази даних** (db tm, DB Time). Час бази даних вимірюється з початку відправлення запиту до бази даних і до моменту, коли необхідні дані повністю отримані сервером застосувань. Зменшення пропускної здатності мережі, між сервером застосувань і базою даних ведуть до збільшення часу бази даних. Перед зверненням до бази даних інтерфейс бази даних робочого процесу перевіряє наявність необхідних даних у буферах серверу застосувань. Якщо буфери містять необхідні дані, то ці дані вибираються із буферів. Використання буферів даних серверів застосувань може значно (у сотні разів) збільшити продуктивність виборки даних [4].

**Час очікування контексту** (rollwt tm, Average Roll Wait Time) виникає при виконанні RFC-запитів або, інакше кажучи, при виникненні взаємодій між програмними компонентами, і містить у собі **час виконання циклів обміну** (roundtrips tm), – між сервером застосувань і елементами керування клієнтським програмним забезпеченням. Час, витрачений на взаємодію між сервером застосування і клієнтським програмним забезпеченням (затримки сигналу в мережі, формування екрану на клієнтському місці), включений до аналізу даних монітору завантаження, називається **часом GUI** (GUI tm). У цілому, якщо в ході виконання кроку транзакції не виникало RFC-взаємодій із зовнішніми системами, то час GUI буде приблизно дорівнювати часу очікування контексту:

$$\text{GUI tm} \approx \text{rollwt tm.} \quad (1)$$

**Час блокування** (enq tm) – час затрачений робочим процесом на запити операцій блокування.

**Час процедур бази даних** (DBproc tm) – час виконання процедур збереження у базі даних.

**Час обробки** (proc tm) – визначальний показник [11], який є складовою часу відповіді та розра-

ховується, як (2) різниця середнього часу відповіді та суми значень часових показників диспетчеру, завантаження програм, завантаження та очікування контексту на сервері застосувань, часу процедур блокування, часу бази даних та часу процедур бази даних:

$$\begin{aligned} \text{proc tm} = & \text{resp tm} - \text{disp tm} - \text{load tm} - \text{db tm} - \\ & - \text{rollin tm} - \text{rollwt tm} - \text{enq tm} - \text{DBproc tm.} \end{aligned} \quad (2)$$

Час обробки показує, як довго виконувалися прикладні АВАР-програми.

Усі значення часових показників, зазначених вище, відносяться до дій, що виконуються робочим процесом системи SAP. Витрачений на ці дії час розраховується незалежно від того, робочий процес був активним, чи перебував у стані очікування.

Наприкінці кроку транзакції робочий процес за допомогою операційної системи визначає, скільки **процесорного часу** (cpu tm) витрачено на виконання своєї роботи.

Процесорний час не є незалежною величиною – він входить до перерахованих вище показників (час завантаження програм, контекст користувача, час обробки).

**Час мережі** (network tm) становить суму часу доставки першого запиту від ПЗ користувача до системи SAP на початку кроку транзакції й часу доставки фінального пакету даних від системи SAP до клієнтського екрану користувача на заключному етапі виконання кроку транзакції. Час мережі не включений у час відповіді.

Оскільки час відповіді для користувача або **час користувача** (user tm) у системі SAP є величиною свідомо меншою, ніж те, що спостерігає реальний користувач, то час відповіді для користувача дорівнює сумі часу відповіді системи й часу мережі:

$$\text{user tm} = \text{resp tm} + \text{network tm.} \quad (3)$$

У результаті дослідження основних часових показників при завантаженості системи у процесі виконання діалогового режиму ієрархічна структура часових показників приймає такий вигляд, як на рис. 1.

Із ієрархії слідує, що час користувача (клієнта), є основним часовим показником. Суб'єкти, що безпосередньо впливають своїми діями на завантаженість та продуктивність системи – користувачі, адміністратори, програмісти (розробники), тестувальники.

Зосередимо увагу на програмістах.

Враховуючи характер програм, що створюються та модифікуються спеціальним програмним забезпеченням, програми, що є непрацездатними впливають у першу чергу на resp tm та load tm.

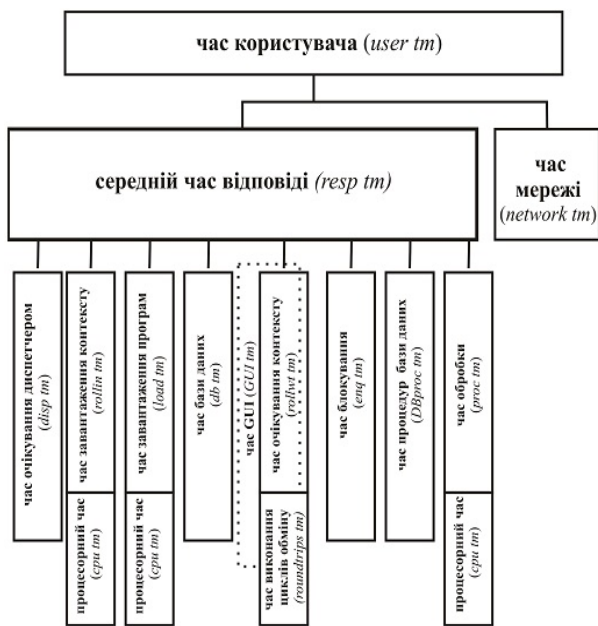


Рис. 1. Ієрархічна побудова взаємозв'язків часових показників

У роботі [12] була знайдена детермінована математична функціональна залежність діяльності програміста при виправленні програм та розкрита її природа:

$$y(x) = 9,112e^{-0,1043x} + \frac{4,218}{1 + e^{-\frac{2}{63,371}(x-\Delta x)}}, \quad (4)$$

де  $\Delta x$  – параметр регресії логістичної функції співпадіння точки симетрії з віссю ординат. За умов апроксимації  $\Delta x \approx 50,5$ .

Якісний контроль ефективності діяльності програмістів досягається на етапі впровадження та супроводження проекту можна контролювати за рахунок зміни супутнього фактору – часових показників працездатності системи.

Для прикладу, на рис. 2 представлена функціональна залежність виправлення помилок у спеціальному програмному забезпеченні програмістами різного рівня (класу).

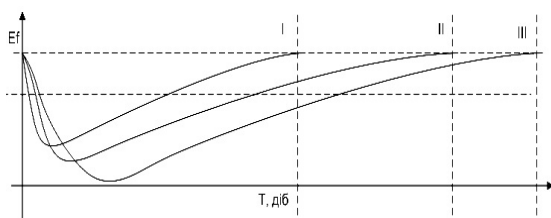


Рис. 2. Різниця в часі між рівною кількістю програмістів різного класу по досягненню встановленого ефекту при мінімальній складності СПЗ

Даний поділ математично обґрунтовується параметром інтенсивності внесення помилок при виправленні програм, 0,1, 0,3 та 0,7, відповідно для I, II та III класів [8]. Оптимізаційна модель має ресурсні обмеження на час, вартість, якість та кількість програмістів в проектній команді.

1. *Ресурсне обмеження: час.* Кількість відновлених програм програмістами прямо-пропорційно залежить від значень часових показників ERP системи (5) і показує працездатність проекту (рис. 2).

На рис. 2 рівень ефективності,  $Ef$  дорівнює часу завантаженості програми:

$$Ef = load \cdot tm. \quad (5)$$

При досягненні встановленого рівня працездатності, програми переходять у нормальний режим функціонування. Спробуємо представити часовий проміжок у вигляді кошторису прямих витрат на оплату праці програмістів, що то за даними [13] становить більше 50% від вартості усього проекту інформатизації. Тобто, потребує особливої уваги щодо оптимального залучення програмістів. Формула (6) показує перехід з від ресурсного обмеження *Час* до *Вартість*:

$$W = 3П \cdot t + УВ \cdot t + f(t), \quad (6)$$

де  $W$  – вартість проекту в грошах,  
 $3П$  – заробітна плата програміста,  
 $t$  – час виконання проекту,  
 $УВ$  – утрачена вигода,  
 $f(t) = const$  – постійні додаткові виплати.

2. *Ресурсне обмеження: вартість.* На рис. 3 відображено накладення графіків кривих оптимізаційної моделі (рис. 2) з обмеженням на час та вартість (оплата праці програміста).

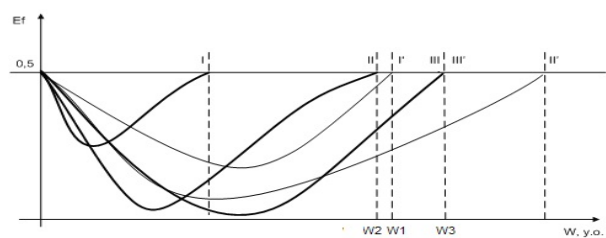


Рис. 3. Графік порівняння вибору типу програміста за грошовим обмеженням на оплату праці із часовими обмеженнями проекту

Нехай, вісь абсцис показує вартість.

Допускаємо, що рівень досягнення працездатності програм, може бути встановленим на рівні не нижче 0,5 – умовний рівень мінімальної готовності СПЗ.

Якщо вважати різницю у заробітній платі між програмістами I, II та III класів (співпадає з поділом на Junior, Medium та Senior), то виходячи з кон'юнктури ринку праці [13–14], заробітна плата програміста I класу буде становити наближену ве-

личину коштів, що на 1/3 вище, ніж для програміста II класу, та на 2/3 вище – III класу.

Тому, на графіку можна побачити, деяку особливість.

Через різницю в оплаті праці програмісти різних класів у часовому вимірі (потовщені криві на рис. 3), мають порядок досягнення ефекту відмінний, ніж у вартісному перерахунку (6) (потоншені криві, рис.3). Помітно, з рис. 3, що програмісти II класу стають менш вигідніші в утриманні, ніж III класу. Це пояснюється наявністю нелінійної залежності між ними. З рис. 3, програмісти досягають ефект (наприклад, значення часового показника *load tm*) в точках I', II' та III'. Причому, для III класу часові та вартісні криві збігаються. Їх можна інтерпретувати, як значення *припинення фінансування* (стоп фінансування), відповідно, в значеннях W1, W2, W3 для I, II та III класів програмістів.

Відомо, що витрати включають основні складові: заробітну плату (ЗП) та утрачену вигоду (Ув).

При додаванні значення утраченої вигоди, Ув, (рис. 4) до кривої програмістів, крива програміста зміститься праворуч, що говорить про збільшення витрат на фінансування.

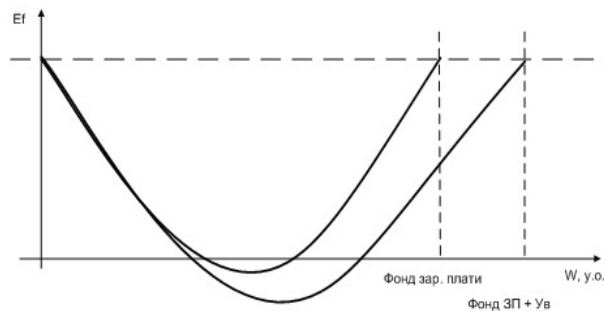


Рис. 4. Графік залежності росту витрат від коефіцієнту утраченої вигоди, Ув

На рис. 5 показані три варіанти кривих розподілу для множини значень отриманих при досягненні ефекту програмістами різних класів виходячи з різних правил оплати праці (див. рис. 3).

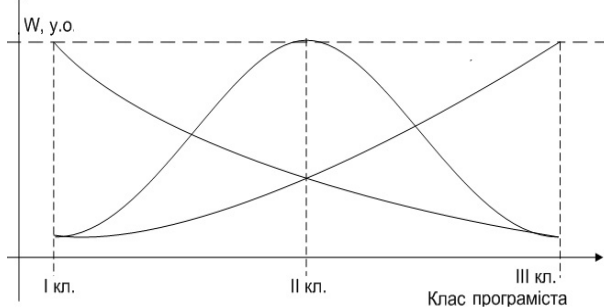


Рис. 5. Графік залежності класифікації програмістів різних класів (підкласів) від витрат (з допущенням наявності перехідних класів)

Із попередніх умов, роблячи висновок з графіку (рис. 5) можна припустити, що найкраща оптимізація буде розташовані по краях, а найгірша – в піку. Плавність кривих вказує на допущення існування деяких перехідних класів програмістів. Це може бути обумовленим різними підходами та правилами щодо кількісної оцінки діяльності програмістів (метрика) та оцінки якості його продукту – програм СПЗ.

Графік (рис. 6) показує варіанти швидкості досягнення ефекту різної кількості програмістів одного класу (див. рис. 3).

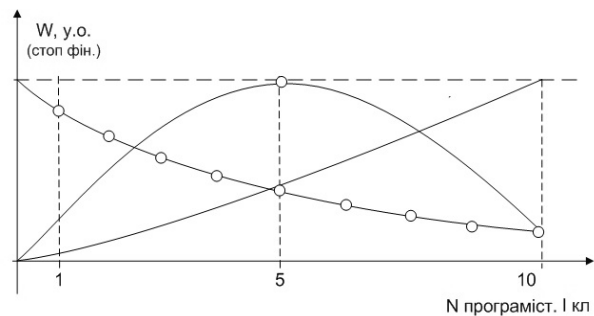


Рис. 6. Графік можливих залежностей числа програмістів I класу від точки припинення фінансування

На осі ординат відкладені значення припинення фінансування залучення програмістів одного класу W-стоп. Крива, що спадає вниз (з кільцями) є наочним прикладом неочікуваного результату: вигідніше залучати програмістів I класу (дорожчих), ніж II та III класів (дешевших), для швидкого відновлення працездатності СПЗ. Є застереження, що дана умова виконається, лише при одночасному повному (без простою) залученні програмістів I класу. Інші дві криві відображають різні варіанти для різних класів.

3. *Ресурсне обмеження: якість та кількість програмістів.* Припустимо, що є задане число програмістів проектної команди, нам необхідно визначити їх співвідношення за класом. Нехай їх загальна кількість відома і рівна 10:

$$N = 10$$

$$N = N_I + N_{II} + N_{III} \tag{7}$$

$$N_{III} = 10 - N_I - N_{II},$$

де N – загальне число програмістів (трьох класів),  $N_I, N_{II}, N_{III}$  – для I, II, III класів. Тоді, на рис. 7, відкладемо відрізок із між максимально допустимими значеннями по осях (0, 10) та (10, 0). Із цього слідує, що площині утвореного прямокутного трикутника (10, 0, 10) належить область визначених даних для програміста II класу (7). Причому, площа із замкненими фігурами, в середині графіка (рис. 7), відображає границі оптимального підбору команди із

програмістів трьох не тільки виходячи із чисельності, але і з вартості (межі замкнених фігур нанесені по значеннях вартості  $W$  щодо досягнення встановленого ефекту для програмістів  $I'$ ,  $II'$  та  $III'$  (рис. 3).

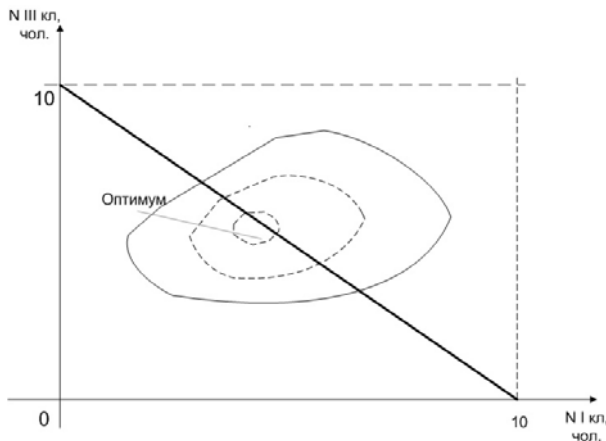


Рис. 7. Графік оптимізації складу проектної команди за умови мінімізації фінансування проекту за класами

Припускаємо, що в проектній команді наявні обмеження на кількість програмістів за класами: 8 програмістів III класу і 4 програмісти I класу. Додавши третю вісь – величину припинення фінансування ( $W$ , стоп фін.) отримуємо попередній графік у тривимірному просторі (рис. 8).

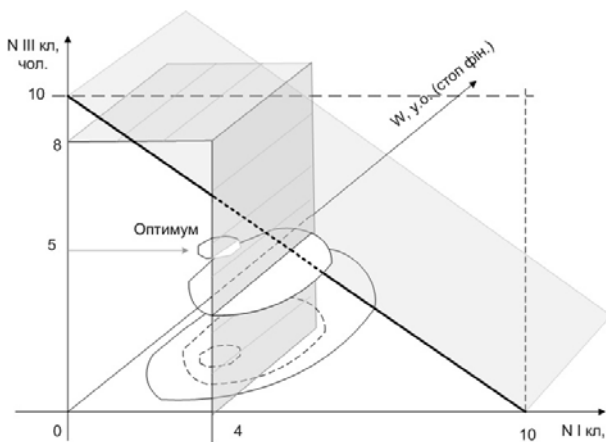


Рис. 8. Графік оптимізації складу проектної команди за класами з урахуванням обмежень

Межі встановленої кількості програмістів представлені трьома площинами, а допустимі області витрат – трьома. Із графіку розглянутого прикладу випливає, що оптимум витрат проектної команди досягається при наборі програмістів: I кл. = 3 чол., II кл. = 2 чол., I кл. = 5чол.

В якості показника ефекту працездатності СПЗ в ході його створення модифікації були використані часові показники із визначеними граничними значеннями (табл. 1). Зазначений вид проблемних значень часових показників завантаженості системи вказує на наявність у системі “вузького місця” при-

чиною якого може стати програма з помилками, як наслідок діяльності програміста.

Таблиця 1

Критеріальні значення часових показників процесу надходження запиту

Назва часового показнику	Гранична межа середнього часу	Вид проблеми
Час очікування диспетчером	10% <i>resp</i> $t_m$ ; 50 мс	Загальна проблема продуктивності
Час завантаження/вивантаження контексту користувача	20 мс	Заповнення буферу, розширена пам'ять або проблеми із процесором
Час завантаження програм	50 мс	Заповнений програмний буфер або проблеми із процесором
Час бази даних	40% від ( <i>resp</i> $t_m$ - <i>disp</i> $t_m$ ); 200-600 мс	Проблеми із базою даних, проблеми мережі чи проблеми із процесором
Час очікування контексту	200 мс	Проблеми взаємодій із ПЗ користувача чи взаємодією із зовнішніми компонентами
Час GUI	200 мс	Проблеми взаємодії із ПЗ користувача
Час блокування	5 мс	Проблеми процесу блокування, мережеві проблеми
Час обробки	$2 * \text{cpu } t_m$	Проблеми із процесором чи проблеми взаємодії із компонентами системи
Час обробки одного запиту бази даних	5 мс	Проблеми із базою даних

Наприклад, зростання значення часу бази даних більше, ніж на 40 відсотків від різниці середнього часу відповіді та часу очікування диспетчером, свідчить про наявність проблем із базою даних, мережею чи процесором (табл. 1).

При таких проблемах гранична межа встановлена на значенні в 200 мс (в окремих випадках – 600 мс).

Суттєве перевищення граничної межі середнього часу значень отриманих з монітору завантаження за часовими показниками (табл.1), вказує на

наявність певного виду проблем і є основою для прийняття рішення щодо їх усунення.

Існує два різних джерела статистичних даних часових показників, перше – у результаті вимірів робочих процесів у системі SAP та друге – в операційній системі (процесорний час). **Час простоювання** (lost tm) – порівняння даних цих двох джерел інформації, – це різниця (8) середнього часу відповіді, загального часу затримки, протягом яких робочий процес СПЗ не потребував процесорного часу (час очікування диспетчером, час бази даних, час блокування, час очікування контексту, час процедур бази даних) та витраченого процесорного часу:

$$\begin{aligned} \text{lost tm} = & \text{resp tm} - (\text{disp tm} + \\ & \text{db tm} + \text{rollwt tm} + \text{enq tm} + \\ & + \text{DBproc tm}) - \text{cpu tm}. \end{aligned} \quad (9)$$

Виходячи із вище зазначеного, найбільш ймовірними причинами значного збільшення часу затримки можуть бути:

– проблема з процесором – недостатня продуктивність процесора, – як наслідок, виникнення конкуренції процесів, їх призупинка планувальником операційної системи;

– виникнення очікування у робочому процесі системи SAP. Кожного разу, коли робочий процес системи переходить у стан “зупинений”,

час обробки росте без збільшення процесорного часу.

Дані затримки, також можуть бути спричинені діяльністю програмістів.

В залежності від особливостей програми та класу програмістів значення часових показників буде змінюватися.

## Висновки

Отже, у результаті проведеного аналізу наявних часових показників була запропонована нова ієрархічна побудова та досліджені взаємозв'язки часових показників із кроками транзакції по системних компонентах.

Контроль за часовими показниками може використовуватись, як початкова процедура при визначенні працездатності спеціального програмного забезпечення.

Була вперше розроблена модель підвищення ефективності підбору проектної команди роботи програмістів різних кваліфікаційних класів з урахуванням часових показників працездатності спеціального програмного забезпечення.

Приведені грубі моделі оптимізації роботи програмістів при відхиленнях виправленні помилок програми для досягнення початкового ефекту.

## Список літератури

1. Войтенко О.С. Когнітивні моделі та інформаційні технології управління проектами та програмами: автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.22 “Управління проектами та програмами” / Войтенко Олександр Степанович – К., 2008. – 22 с.
2. Asmar L. Framework for the agile development of innovative Product-Service-Systems for existing physical rehabilitation systems / L. Asmar, M. Rabe // *Procedia Manufacturing*. – 2018. – № 24. – С. 147-152.
3. IEEE Standard Glossary of Software Engineering Terminology [Електронний ресурс]: IEEE Std 610.12-1990. – NY: The Institute of Electrical and Electronics Engineers, 1990. – 84 p. – Standards Coordinating Committee of the Computer Society of the IEEE. – Available at: <http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/610-12-1990.pdf>.
4. Хагеман С. SAP R/3. Системное администрирование / С. Хагеман, Л. Вилл. – М.: Лори, 2013. – 480 с.
5. Response-Time Analysis for Mixed Criticality Systems: materials of international Symposium // 32nd Real-Time Systems Symposium, Vienna, November 29–December 2, 2011, Vienna: IEEE, 2011. – P. 116-134. Available at: <https://ieeexplore.ieee.org/abstract/document/6121424/metrics>.
6. ABAP SQL Monitor Implementation Guide and Best Practices [Електронний ресурс] // SAP AG. – 2016. – Режим доступу: <https://www.sap.com/documents/2013/10/92b57ae6-527c-0010-82c7-eda71af511fa.html>.
7. Теоретичні підходи для розрахунку штату програмістів, необхідних для підтримки єдиного інформаційного середовища / Ю.А. Кіріпнічков, В.А. Федорієнко, О.В. Головченко [та ін.] // Збірник наукових праць ЦВСД НУОУ ім. І. Черняхівського. – 2014. – № 3(52). – С. 133-139.
8. Sijin Joseph. Programmer Competency Matrix [Electronic resource] / Sijin Joseph. – Boston: Perspectives on Software, 2014. – Available at: <http://sijinjoseph.com/programmer-competency-matrix/>.
9. Nguyen V. Improved Size and Effort Estimation Models for Software Maintenance [Electronic resource]: dis. ... PhD (Computer Science) / V. Nguyen. – Los Angeles, 2010. – 183 p. – Available at: [http://csse.usc.edu/csse/TECHRPTS/PhD\\_Dissertations/files/Nguyen\\_Dissertation.pdf](http://csse.usc.edu/csse/TECHRPTS/PhD_Dissertations/files/Nguyen_Dissertation.pdf)
10. Андерсон Д. SAP за 24 часа / Д. Андерсон, Д. Ларокка; пер. с англ. В. М. Котовский. – Днепропетровск: Баланс Бизнес Букс, 2007. – 400 с.
11. Definition of SAP response time/CPU time, Note 8963 [Electronic resource] // SAP AG. – 2016. – Available at: <https://service.sap.com/notes>.
12. Регресивний аналіз пошуку функції залежності кількості працюючих програм єдиного інформаційного середовища при визначеній класності програмістів / В.Л. Шевченко, В.А. Федорієнко, Ю.А. Кіріпнічков, О.В. Головченко // Збірник наукових праць ЦВСД НУО України І. Черняхівського. – 2015. – № 3(55). – С. 6-12.
13. Зарплати розробників [Електронний ресурс] // DOU.ua. – 2018. – Режим доступу: <https://jobs.dou.ua/salaries/>.

14. Професії IT-сфери: Технік-програміст [Електронний ресурс] . – Хмельницький: ХРЦЗ, 2013. – Режим доступу: [http://www.dcz.gov.ua/khm/control/uk/publish/printable\\_article; jsessionid =3DA9 DC8E8CF5F368554D447A878E8845?art\\_id=275188](http://www.dcz.gov.ua/khm/control/uk/publish/printable_article; jsessionid =3DA9 DC8E8CF5F368554D447A878E8845?art_id=275188).

## References

1. Voytenko, O.S. (2008), “*Kohnityvni modeli ta informatsiyi tekhnolohiyi upravlinnya proektamy ta prohramamy*” [Upravlinnya proektamy ta prohramamy] Kyiv, 22 p.
2. Asmar, L., Rabe, M., Vodchyts, O.H., Low, Y., Jingye, Y., Kühn, A. and Dumitrescu, R. (2018), Framework for the agile development of innovative Product-Service-Systems for existing physical rehabilitation systems, *Procedia Manufacturing*, Vol. 24, pp. 147-152.
3. IEEE Standard Glossary of Software Engineering Terminology (1990), *IEEE Std 610.12-1990*. – 84 p. – available at: <http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/610-12-1990.pdf>.
4. Hagermann, S. and Will, L. (2013), *SAP R/3 System Administration*, Lori, Moscow, 480 p.
5. Baruah, S.K., Burns, A. and Davis, R.I. (2011), Response-Time Analysis for Mixed Criticality Systems, *IEEE 32nd Real-Time Systems Symposium*, Vienna, November 29–December 2, pp. 116-134, available at: <https://ieeexplore.ieee.org/abstract/document/6121424/metrics>.
6. SAP AG (2016), *ABAP SQL Monitor Implementation Guide and Best Practices*, available at: [www.sap.com/documents/2013/10/92b57ae6-527c-0010-82c7-eda71af511fa.html](http://www.sap.com/documents/2013/10/92b57ae6-527c-0010-82c7-eda71af511fa.html) (accessed 7 November 2018).
7. Kirpichnikov, Y.A., Fedoriienko, V.A., Golovchenko, O.V. and Koshlan, O.A. (2014), Teoretychni pidkhody dlya rozrakhunku shtatu prohramistiv, neobkhidnykh dlya pidtrymky yedynoho informatsiynoho seredovyscha, [Theoretical approaches to the calculation of the state of programmers necessary to support a single information environment], *Zbirnyk naukovykh prats' TSVSD NUOU im. I. Chernyakhovskoho*, No. 3, pp. 133-139.
8. Joseph, S. (2014), *Perspectives on Software*, Programmer Competency Matrix, Boston, available at: <http://sijinjoseph.com/programmer-competency-matrix/> (accessed 1 December 2018).
9. Nguyen, V. (2010), *Improved Size and Effort Estimation Models for Software Maintenance: dissertation*, Los Angeles, 183 p., available at: [www.ascusc.org/jcmc/vol5/issue2/](http://www.ascusc.org/jcmc/vol5/issue2/).
10. Anderson, G.W. and Larocca, D. (2007), “*SAP za 24 chasa*” [SAP in 24 Hours], Balance Business Books, Dnipropetrovsk, 400 p.
11. SAP AG, SAP corporate website (2014), *Definition of SAP response time/CPU time, Note 8963*, available at: <https://service.sap.com/notes>, (accessed 11 November 2014).
12. Shevchenko, V.L., Fedoriienko, V.A., Kirpichnikov, Y.A. and Golovchenko, O.V. (2015), “Rehresyvnnyy analiz poshuku funktsiyi zalezhnosti kilkosty pratsyuyuchykh prohram yedynoho informatsiynoho seredovyscha pry vyznacheni klassnosti prohramistiv” [Regressive analysis of the search function for the dependence of the number of working programs of a single information environment with a certain class of programmers], *Zbirnyk naukovykh prats' TSVSD NUOU im. I. Chernyakhovskoho*, No. 3, pp. 6-12.
13. Analitical Portal DOU.ua. (2018), “*Zarplaty razrabotchikov*” [Developer salaries], available at: <https://jobs.dou.ua/salaries/> (accessed 4 October 2018).
14. Khmel'nyts'kyu rehional'nyy tsentr zaynyatosti (2013), “*Profesiyyi IT-sfery: Tekhnik-prohramist*” [Professions of IT sphere: Technician-programmer] Khmelnytskyu, available at: [www.dcz.gov.ua/khm/control/uk/publish/printable\\_article; jsessionid =3DA9 DC8E8CF5F368554D447A878E8845?art\\_id=275188](http://www.dcz.gov.ua/khm/control/uk/publish/printable_article; jsessionid =3DA9 DC8E8CF5F368554D447A878E8845?art_id=275188), (accessed 3 October 2018).

Надійшла до редколегії 14.11.2018

Схвалена до друку 11.12.2018

### Відомості про автора:

#### Федорієнко Віталій Анатолійович

старший науковий співробітник  
Національного університету оборони України  
ім. І. Черняхівського,  
Київ, Україна  
<https://orcid.org/0000-0002-0921-3390>

### Information about the author:

#### Vitalii Fedoriienko

Senior Research  
of Ivan Cherniakhovsky  
National Defence University of Ukraine,  
Kyiv, Ukraine  
<https://orcid.org/0000-0002-0921-3390>

## ВЛИЯНИЕ ЭФФЕКТА ВРЕМЕННЫХ ПОКАЗАТЕЛЕЙ ЗАГРУЖЕННОСТИ НА ОПТИМИЗАЦИЮ СОЗДАНИЯ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ERP СИСТЕМЫ

В.А. Федориенко

Проведен анализ контроля работоспособности по временным показателям ERP системы и раскрыты основные из них. Предложен новый структурный подход иерархического построения взаимосвязей временных показателей загрузки системы. Определены субъекты создания и модификации специального программного обеспечения. Исследован характер изменения значений конкретных временных показателей при деятельности программистов разной квалификационной классности. Предложено, принять некоторые временные показатели с предельными границами в качестве эффекта работоспособности для построения оптимизационной зависимости классности субъектов создания и модификации специального программного обеспечения в условиях ресурсных ограничений.

**Ключевые слова:** временные показатели загрузки, классность программистов, оптимизационная модель, иерархия временных показателей.



---

**INFLUENCE OF THE EFFECT OF LOADING TIME INDICATORS ON OPTIMIZATION OF PROGRAMMING DEVELOPMENT OF ERP SOFTWARE**

V. Fedoriienko

*It is common knowledge that any software and hardware complex requires attention to maintain its working capacity. The ERP system manufacturers (Enterprise Resource Planing) offer their own standard software with maximum approximation of tools and software modules to use functional needs. Considering attention at the specifics of the using area, the level of reaching, and maintaining compliance with national, sectoral, and internal requirements acts determine the complexity of the project. In ERP systems, there are a number of aproaches for conducting work to maintain the system in working condition. One of the aproachs of control ERP systems is monitoring of system load, the key indicator of which is the time factor. There is a contradiction between the requirement for monitoring, as a procedure for rapid control of system performance. Further its optimization with the allocation of the hierarchy will also allow to influence the evaluation of the activity of programmers of different qualification classes. Qualitative control of the effectiveness of programmers is achieved at the stage of implementation and support of the project. The efficiency of the system may be controlled by changing the additional factor - the performance time of the system. The qualification division of programmers is mathematically substantiated by the parameter of the intensity of making mistakes in their programs, 0,1, 0,3 and 0,7 respectively for I, II and III classes [8]. The optimization model has resource limitations on time, cost (wage), and quality, and the number of programmers in the project team. As indicator of the efficiency of software, time values were used with the specified limit values during the ime of its creation and modification. These type of problem shows time indicators of system loading. There is a "bottleneck" in the system that may cause the problem in the developed program. This may be a consequence of the programmer's activity. As indicators of the effect of SDR performance during its creation of modifications, time indicators were used with the specified limit values (Table 1). The above-mentioned kind of troublesome value of the time values of the system's load indicates that there is a "bottleneck" in the system that may lead to a program with errors, as a consequence of the programmer's activity. The presence of a hierarchical structure allows a person responsible for the efficiency of the system in time to make a managerial decision after a quick assessment of the system. This allow us to fully monitor and optimize system performance during project management. Thus, the methodology for making a decision on selecting a design team was improved. An optimal solution for developing and modifying software is based on the limitation of the number and quality of programmers, wages and time.*

**Keywords:** *temporal indices of load, quality levels of programmers, optimization model, hierarchy rhythm of time indices.*