

КОВАЛЕНКО Олена

Вінницький національний технічний університет

<https://orcid.org/0000-0003-2864-9058>

e-mail: ok@vntu.edu.ua

СТОРОЖУК Юрій

Вінницький національний технічний університет

<https://orcid.org/0009-0002-1066-1358>

e-mail: ystorozhuk@vntu.edu.ua

РОЗРОБКА СИСТЕМИ ТЕСТУВАННЯ ПРОДУКТИВНОСТІ КОРПОРАТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ МОДЕЛІ ЗБАЛАНСОВАНИХ ПОКАЗНИКІВ

В статті розглядається актуальна проблема розробки системи тестування продуктивності корпоративних інформаційних систем на основі моделі збалансованих показників. Така система активно використовується для вимірювання ефективності роботи підприємства, корпорації і може бути адаптована для оцінювання продуктивності програмного продукту. В статті виконано короткий огляд сучасних підходів до тестування продуктивності РІС, обґрунтування застосування МЗП, розробку концептуальної моделі системи тестування, опис практичної реалізації та формулювання висновків і перспектив подальших досліджень.

Запропонована система проєкцій продуктивності, ключові показники, приклади використання програмних продуктів. За результатами досліджень сформовані рекомендації для тестування продуктивності розподілених інформаційних систем за допомогою запропонованих проєкцій

Ключові слова: програмна інженерія, життєвий цикл ІТ-проєкти, дослідження продуктивності, оцінювання продуктивності, тестування, алгоритми роботи системи.

KOVALENKO Olena, STOROZUK Yuriy

Vinnitsia National Technical University

DEVELOPMENT OF A SYSTEM FOR TESTING THE PRODUCTIVITY OF CORPORATE INFORMATION SYSTEMS BASED ON THE BALANCED SCOREBOARD MODEL

The article considers the current problem of developing a system for testing the productivity of corporate information systems based on the balanced scorecard model. Such a system is actively used to measure the efficiency of an enterprise, corporation and can be adapted to assess the productivity of a software product. The article provides a brief overview of modern approaches to testing the productivity of RIS, justifies the use of the BSC, develops a conceptual model of the testing system, describes practical implementation and formulates conclusions and prospects for further research.

A system of productivity projections, key indicators, and examples of using software products is proposed. Based on the research results, recommendations for testing the performance of distributed information systems using the proposed projections were formulated. The article proposes a conceptual and practical model of the performance testing system for corporate distributed information systems based on the balanced scorecard model. The main scientific novelty is the integration of technical, business, process and innovation indicators into a single evaluation system, which allows to increase the objectivity and relevance of testing results. The practical significance of the developed system lies in the possibility of its adaptation to different types of corporate systems and their software, increasing the efficiency of testing and improving the quality of managerial decision-making.

Prospects for further research are related to the automation of building MRP models for various industries, the introduction of machine learning methods to predict productivity changes, the development of tools for deeper integration with DevOps processes, and the expansion of the scorecard taking into account the specifics of the latest architectures (serverless, edge computing).

Keywords: software engineering, IT project life cycle, performance research, performance assessment, testing, system operation algorithms.

Стаття надійшла до редакції / Received 02.08.2025

Прийнята до друку / Accepted 24.08.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Проблема забезпечення продуктивності корпоративних розподілених інформаційних систем набуває актуальності в умовах зростаючої складності архітектур, підвищення вимог до масштабованості, надійності та доступності сервісів [1]. Продуктивність визначає якість користувацького досвіду, ефективність бізнес-процесів і конкурентоспроможність організацій у цифровому середовищі. Як правило, продуктивність програмного забезпечення визначають за технічними характеристиками, але останнім часом став популярним підхід до комплексного визначення продуктивності програмного забезпечення інформаційних систем.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Тестування продуктивності є невід'ємною складовою життєвого циклу розвитку інформаційних систем та їх програмного забезпечення, однак існуючі підходи часто мають суттєві недоліки, серед яких – фрагментарність оцінювання, орієнтація лише на окремі технічні метрики, недостатня інтеграція з бізнес-цілями та відсутність системного підходу до аналізу результатів [2, 3]. Зазначені недоліки обумовлюють необхідність розробки нових методик, які дозволяють комплексно оцінювати продуктивність програмного забезпечення з урахуванням як технічних, так і організаційно-економічних аспектів. Одним із перспективних напрямів є використання моделі збалансованих показників (МЗП), яка зарекомендувала себе як ефективний інструмент стратегічного управління та моніторингу діяльності організацій [4], але недостатньо досліджена у сфері тестування продуктивності програмних продуктів.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є розробка концептуальної та практичної моделі системи тестування продуктивності корпоративних розподілених інформаційних систем на основі МЗП, що дозволяє підвищити об'єктивність, повноту та релевантність оцінювання.

Для досягнення цієї мети поставлено наступні завдання:

адаптація МЗП до задач тестування продуктивності програмного забезпечення; розробка концепції та практичної методики впровадження системи тестування; визначення ключових показників та принципів контролю.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Тестування продуктивності корпоративних розподілених систем має низку особливостей, що відрізняють його від аналогічних процесів у централізованих та монолітних системах. До таких особливостей належать: розподіленість компонентів, неоднорідність інфраструктури, динамічність навантаження, складність трасування запитів, взаємозалежність сервісів, взаємодія з бізнес-цілями [5]. Сучасні методи тестування продуктивності, такі як навантажувальне, стресове, об'ємне та тривале тестування, зазвичай фокусуються на окремих аспектах (наприклад, максимальна пропускна здатність, час відгуку, стійкість до пікових навантажень) і рідко враховують бізнес-вимоги, економічну доцільність чи стратегічні цілі організації [6].

Засоби тестування (JMeter, Gatling, Locust, k6 тощо) забезпечують широкі можливості для автоматизації та збору технічних метрик, але не дають цілісної картини впливу продуктивності на бізнес-результати [7, 8]. Відомі підходи до інтеграції тестування продуктивності з DevOps-практиками (CI/CD, observability) значно покращують оперативність і гнучкість контролю, однак не вирішують проблему комплексної оцінки ефективності системи з різних перспектив [8].

Моделі збалансованих показників (Balanced Scorecard, BSC) дозволяє враховувати взаємозв'язок між технічними, бізнесовими, процесними та інноваційними показниками і формувати єдину систему моніторингу та аналізу [4, 5]. Застосування МЗП у сфері тестування продуктивності PIC відкриває можливість інтегрувати традиційні метрики (latency, throughput, error rate) з показниками задоволеності користувачів, вартісними аспектами, рівнем автоматизації тестування та інноваційними ініціативами.

Формування моделі тестування продуктивності для корпоративних інформаційних систем (зокрема їх програмного забезпечення) на основі МЗП передбачає визначення ключових перспектив і показників оцінювання. Доцільно виділяти такі перспективи:

1. Технічна перспектива:

Основні метрики – час відгуку, пропускна здатність, рівень використання ресурсів, стійкість до відмов, масштабованість.

2. Користувацька перспектива:

Показники задоволеності користувачів, час виконання ключових бізнес-операцій, частота негативних відгуків, відсоток виконання SLA.

3. Процесна перспектива:

Рівень автоматизації тестування, швидкість реагування на інциденти, інтеграція з CI/CD, тривалість циклу тестування.

4. Інноваційна/розвиткова перспектива:

Впровадження нових підходів до тестування, використання ML/AI для генерації сценаріїв, рівень компетентності команди, кількість ініціатив з покращення продуктивності.

Взаємозв'язок між показниками забезпечується завдяки причинно-наслідковим зв'язкам між проєкціями. Наприклад, підвищення автоматизації тестування (процесна перспектива) сприяє зниженню часу виявлення проблем (технічна перспектива), що, у свою чергу, підвищує задоволеність користувачів і зменшує витрати (користувацька та бізнесова перспектива).

Методика розробки вимог до продуктивності розподілених інформаційних систем з урахуванням МЗП передбачає тісну співпрацю між технічними фахівцями, бізнес-аналітиками та кінцевими користувачами.

На першому етапі визначаються бізнес-цілі та ключові сценарії використання системи, які трансформуються у вимоги до продуктивності (наприклад, максимальний час відгуку для певної операції, допустимий рівень відмов, цільова пропускна здатність).

Далі відбувається деталізація вимог у вигляді параметрів тестування: кількість одночасних користувачів, типи навантаження, профілі трафіку, сценарії відмов. На основі МЗП формуються ключові показники, які відслідковуються під час тестування.

Наведемо приклад практичного сценарію:

- Вимога: Час відгуку для операції "оформлення замовлення" не повинен перевищувати 2 сек для 95% запитів.

- Показник МЗП: % запитів із часом відгуку < 2 сек (технічна перспектива), % позитивних відгуків користувачів (користувацька перспектива).

Фрагмент коду для автоматизації збору та візуалізації метрик (Prometheus + Grafana) представлено на рис. 1.

```
prometheus.yml
```yaml
scrape_configs:
 - job_name: 'app'
 static_configs:
 - targets: ['localhost:8080']
```

Рис. 1. Фрагмент коду для автоматизації збору та візуалізації метрик (Prometheus + Grafana)

Запит у Grafana (PromQL) представлено на рисунку 2.

```
1 histogram_quantile(0.95, sum(rate(http_request_duration_seconds_bucket[5m])) by (le))
```

Рис. 2. Запит у Grafana (PromQL)

Цей запит відображає 95-й перцентиль часу відгуку для HTTP-запитів.

Фрагмент коду автоматизованого тесту на Locust представлено на рис. 3

```
```python
1 from locust import HttpUser, task, between
2
3 class WebsiteUser(HttpUser):
4     wait_time = between(1, 5)
5
6     @task
7     def place_order(self):
8         self.client.post("/order", json={"item": "product1", "quantity": 1})
```

Рис. 3. Фрагмент коду автоматизованого тесту на Locust

Використання МЗП у дашбордах можна представити як створення окремих панелей для технічних, процесних, користувацьких та інноваційних показників (наприклад, час відгуку, кількість автотестів, відсоток SLA, кількість нових сценаріїв тестування).

Запропонована методика може бути реалізована за допомогою таких компонентів тестування:

1. Визначення ключових показників (KPI) для кожної перспективи МЗП

Технічна перспектива:

- Середній та 95-й перцентиль часу відгуку (response time, p95)

- Пропускна здатність системи (throughput, req/sec)

- Кількість помилок (error rate, %)
 - Час відновлення після збою (MTTR)
 - Рівень використання CPU/RAM/мережі
- Користувачка перспектива:
- Рівень задоволеності користувачів (User Satisfaction Score, опитування)
 - Відсоток виконання SLA
 - Середній час виконання ключових бізнес-операцій
- Процесна перспектива:
- Відсоток автоматизованих тестів у регресійному наборі
 - Середній час виконання тестового циклу (test cycle time)
 - Кількість інцидентів, виявлених на етапі тестування
- Інноваційна перспектива:
- Кількість нових сценаріїв тестування, створених за місяць
 - Впровадження ML/AI для генерації сценаріїв (наявність, % покриття)
 - Кількість навчальних заходів для команди

2. Автоматизований збір і візуалізація метрик здійснюється завдяки інтеграції JMeter з Prometheus та Grafana.

У JMeter використовується Backend Listener з Prometheus PushGateway.

Метрики надходять у Prometheus, а Grafana будує дашборд за ключовими показниками.

Розглянемо приклад автоматизованого запуску тестування (CI/CD) Jenkinsfile (Groovy) (Рис. 4).

```
pipeline {
    agent any
    stages {
        stage('Load Test') {
            steps {
                sh 'jmeter -n -t test_plan.jmx -l results.jtl'
            }
        }
        stage('Publish Results') {
            steps {
                publishHTML(target: [
                    allowMissing: false,
                    alwaysLinkToLastBuild: true,
                    keepAll: true,
                    reportDir: 'reports',
                    reportFiles: 'index.html',
                    reportName: 'JMeter Report'
                ])
            }
        }
        stage('Push Metrics') {
            steps {
                sh 'python push_metrics_to_prometheus.py'
            }
        }
    }
}
```

Рис. 4. Приклад автоматизованого запуску тестування (CI/CD) Jenkinsfile (Groovy)

Автоматизований аналіз SLA та формування звітності формується після виконання спеціального скрипту. Фрагмент Python-скрипту для аналізу результатів тесту представлено на рис. 5

```
import pandas as pd

results = pd.read_csv('results.csv')
sla_threshold = 2.0 # секунди
sla_met = (results['response_time'] < sla_threshold).mean() * 100
print(f"SLA виконано для {sla_met:.2f}% запитів")
```

Рис. 5. Фрагмент Python-скрипту

Приклад дашборду Grafana для МЗП може бути реалізовано за допомогою чотирьох панелей:

- “Технічні метрики”: Response Time p95, Throughput, Error Rate.
- “Користувачькі метрики”: User Satisfaction (з інтеграцією опитувань), SLA.
- “Процесні метрики”: % автоматизованих тестів, середній час CI/CD циклу.

- “Інновації”: Кількість нових сценаріїв, ML-покриття.

Оцінка та оптимізація системи здійснюється за такою логічною послідовною схемою:

- Якщо SLA не досягнуто — автоматичне створення задачі в Jira для аналізу вузьких місць.

- Регулярний перегляд цільових значень KPI на основі історичних даних та зворотного зв'язку від бізнесу.

- Впровадження А/В тестування для оцінки впливу змін на продуктивність.

Під час впровадження системи тестування продуктивності в комерційних корпоративних розподілених системах було виявлено, що збільшення автоматизації тестування на 30% дозволило скоротити середній час реагування на інциденти на 40%, а впровадження ML-алгоритмів для генерації сценаріїв дозволило виявити “пікові” проблеми, які не були зафіксовані стандартними статичними тестами.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У статті запропоновано концептуальну та практичну модель системи тестування продуктивності корпоративних розподілених інформаційних систем на основі моделі збалансованих показників. Основною науковою новизною є інтеграція технічних, бізнесових, процесних та інноваційних показників у єдину систему оцінювання, що дозволяє підвищити об'єктивність і релевантність результатів тестування. Практичне значення розробленої системи полягає у можливості її адаптації до різних типів корпоративних систем та їх програмного забезпечення, підвищенні ефективності тестування та покращенні якості прийняття управлінських рішень.

Перспективи подальших досліджень пов'язані з автоматизацією побудови моделей МЗП для різних галузей, впровадженням методів машинного навчання для прогнозування змін продуктивності, розробкою інструментів для глибшої інтеграції з DevOps-процесами та розширенням системи показників з урахуванням специфіки новітніх архітектур (serverless, edge computing).

Література

1. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd Edition. O'Reilly Media, 2021. 618 p.
2. Waseem Muhammad, Liang Peng Marquez, Gaston Di Salle Amleto. Testing Microservices Architecture-Based Applications / V/ Waseem, P. Liang, G. Amleto // A Systematic Mapping Study. DOI: 10.1109/APSEC51365.2020.00020
3. Pradeep Kumar Adaptive Workload Modeling using AI for Performance Testing of Cloud-Based Multitenant Enterprise Applications K. Praddep // International Journal of Innovative Research and Creative Technology 2024 Volume 10, Issue 1. P. 1-17.
4. Kaplan R.S., Norton D.P. The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, 1996. 322 p.
5. Довбня С.Б., Дрофа Є.А. Еволюція і напрями розвитку системи збалансованих показників. / С. Б. Довбня, Є.А. Дрофа // Науковий вісник Міжнародного гуманітарного університету. 2018. 34. С. 4-9.
6. Apache JMeter [Електронний ресурс]. – Режим доступу: <https://jmeter.apache.org/>
7. Locust Documentation [Електронний ресурс]. – Режим доступу: <https://docs.locust.io/>
8. Fowler M. Continuous Integration [Електронний ресурс]. – Режим доступу: <https://martinfowler.com/articles/continuousIntegration.html>

References

1. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd Edition. O'Reilly Media, 2021. 618 p.
2. Waseem Muhammad, Liang Peng Marquez, Gaston Di Salle Amleto. Testing Microservices Architecture-Based Applications. A Systematic Mapping Study. DOI: 10.1109/APSEC51365.2020.00020.
3. Pradeep Kumar Adaptive Workload Modeling using AI for Performance Testing of Cloud-Based Multitenant Enterprise Applications. International Journal of Innovative Research and Creative Technology 2024 Volume 10, Issue 1. P. 1-17.
4. Kaplan R.S., Norton D.P. The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, 1996. 322 p.
5. Dovbnya S.B., Drofa YE.A. «Evolution and development directions of the balanced scorecard system» Evolyutsiya i napryamy rozvytku systemy zbalansovanykh pokaznykiv. Scientific Bulletin of the International Humanitarian University.. 2018.
6. Apache JMeter [Electronic resource]. – Access mode: <https://jmeter.apache.org/>
7. Locust Documentation [Electronic resource]. – Access mode: <https://docs.locust.io/>
8. Fowler M. Continuous Integration [Electronic resource]. – Access mode: <https://docs.locust.io/>