

Міністерство освіти і науки України

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ  
УНІВЕРСИТЕТ

Північно-Східний науковий центр  
Транспортної академії України

## **ВІСНИК**

ХАРКІВСЬКОГО НАЦІОНАЛЬНОГО  
АВТОМОБІЛЬНО-ДОРОЖНЬОГО УНІВЕРСИТЕТУ

---

Збірник наукових праць

Випуск 96

## **BULLETIN**

of  
KHARKIV NATIONAL  
AUTOMOBILE AND HIGHWAY UNIVERSITY

---

Collection of Scientific Works

Issue 96

Харків  
ХНАДУ  
2022

## РЕДАКЦІЙНА КОЛЕГІЯ

Головний редактор  
Заст. гол. редактора  
Помічник гол. редактора  
Відповідальний секретар

**Батракова Анжеліка Геннадіївна**  
**Гурко Олександр Геннадійович**  
**Бережна Катерина Вікторівна**  
**Даченко Віта Василівна**

докт. техн. наук, професор, проректор з науково-педагогічної роботи, ХНАДУ  
докт. техн. наук, професор, професор кафедри автоматизації та комп'ютерно-інтегрованих технологій, ХНАДУ  
канд. техн. наук, доцент, доцент кафедри мостів, конструкцій і будівельної механіки ім. В.О. Російського, ХНАДУ  
канд. хім. наук, доцент, доцент кафедри технології дорожньо-будівельних матеріалів і хімії ім. проф. М.І. Волкова, ХНАДУ

## Члени редколегії:

**Богомолов Віктор Олександрович**  
**Клец Дмитро Михайлович**  
**Батиґін Юрій Вікторович**  
**Бугаський Сергій Олександрович**  
**Бондаренко Володимир Васильович**  
**Возначенко Володимир Олексійович**  
**Волков Володимир Петрович**  
**Воронков Олександр Іванович**  
**Внукова Наталія Володимирівна**  
**Глушкова Діана Борисівна**  
**Горбачов Петро Федорович**  
**Золотарьов Віктор Олександрович**  
**Жданюк Валерій Кузьмич**  
**Кириченко Ігор Георгійович**  
**Клименко Валерій Іванович**  
**Кожушко Віталій Петрович**  
**Колодяжний Володимир Максимович**  
**Левтеров Андрій Іванович**  
**Подригаєло Михайло Абович**  
**Політурк Олександр Васильович**  
**Прулицька Людмила Анатоліївна**  
**Нікітіна Тетяна Борисівна**  
**Саснюк Наталія Віталіївна**  
**Солодов Валерій Григорович**  
**Толмачов Сергій Миколайович**  
**Вельчико Геннадій Вікторович**  
**Golovashchenko Sergey Fedorovich**  
**Hartmut Beckedahl**  
**Igors Tipans**  
**Ireneusz Zbicinski**  
**Laurinavicius Alfredas**  
**Mikhailenko Vladimir**  
**Prentkovskis Olegas**  
**Puodziukas Virgaudas**  
**Radovskiy Boris S.**  
**Roland Lachmayer**  
**Telatyev Bagdat Burkhanbauly**  
**Tyrsa Vira**  
**Sidi Mochammed Senouci**  
**Tropina Albina Albertovna**  
**Vrublevsky Aleksandr Nikolaevich**

докт. техн. наук, професор, ректор, ХНАДУ  
докт. техн. наук, професор, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри фізики, ХНАДУ  
канд. техн. наук, доцент, декан дорожньо-будівельного факультету, ХНАДУ  
канд. пед. наук, доцент, професор кафедри філософії та педагогіки професійної підготовки, ХНАДУ  
докт. техн. наук, доцент, професор кафедри транспортних технологій, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри технічної експлуатації і сервісу автомобілів ім. М.Я. Говорущенка, ХНАДУ  
докт. техн. наук, професор, професор кафедри двигунів внутрішнього згорання, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри екології, ХНАДУ  
докт. техн. наук, доцент, завідувач кафедри технології металів і матеріалознавства ім. О.М. Петриченка, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри транспортних систем і логістики, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри технології дорожньо-будівельних матеріалів і хімії ім. М.І. Волкова, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри будівництва і експлуатації автомобільних доріг ім. О.К. Бірulia, ХНАДУ  
докт. техн. наук, професор, декан механічного факультету, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри автомобілів ім. А.Б. Гредескула, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри мостів, ХНАДУ  
докт. ф-м. наук, професор, професор кафедри інформатики і прикладної математики, ХНАДУ  
канд. техн. наук, професор, завідувач кафедри інформатики і прикладної математики, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри технології машинобудування і ремонту машин, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри метрології і безпеки життєдіяльності, ХНАДУ  
канд. іст. наук, доцент, вчений секретар, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри природничих і гуманітарних дисциплін, ХНАДУ  
докт. пед. наук, професор, завідувач кафедри іноземних мов, ХНАДУ  
докт. техн. наук, професор, завідувач кафедри теоретичної механіки і гідравліки, ХНАДУ  
докт. техн. наук, професор, професор технології дорожньо-будівельних матеріалів і хімії ім. М.І. Волкова, ХНАДУ  
канд. техн. наук, головний конструктор СП «Кредо-Діалог», Мінськ  
Ph.D., Technical Leader Manufacturing Research Department, Ford Research & Advanced Engineering Research and Innovation Center, USA  
Dr.-Ing, Deputy Rector for International Academic Cooperation and Studies, Riga Technical University, Institute of Mechanics, Latvia  
Ph.D., prof., Department of heat and mass transfer, Technical University of Lodz, Poland  
Dr., prof., vice-rector, Vilnius Gediminas Technical University, Lithuania  
S. Dr., prof., Pusan National University, Plasma Research Center, Busan, South Korea  
Dr., prof., Dean of faculty of Transport engineering, Vilnius Gediminas Technical University, Lithuania  
Dr., assoc. prof., head of department of Roads, Vilnius Gediminas Technical University, Lithuania  
prof. consultant, Radnat Consulting, Irvine, United States  
Ph.D., Dr.-Ing, Leibniz Universität, Hannover, Institute of product development, Germany  
Dr., prof., Kazakh Academy of Transport and Communication, Kazakhstan  
Dr., prof., Baja California Autonomous University, Mexico  
Ph.D., prof., Institut Supérieur Automobile et Transport, France  
Dr., prof., Texas A and M University, Department of Aerospace Engineering, United States  
prof., D. Sc., University of Warmia and Mazury in Olsztyn, Poland.

## EDITORIAL BOARD

Editor-in-Chief  
Vice Editor-in-Chief  
Assistant of Editor-in-Chief  
Executive Secretary:

**Batrakova Angelika Gennadiivna**  
**Gurko Alexander Gennadiyovich**  
**Berezina Kateryna**  
**Datsenko Vita**

DSci, Prof., Vice-rector for Research and Education, KhNAHU  
DSci, Assoc. Prof., Professor of the Department of Automation and Computer-Integrated Technologies, KhNAHU  
PhD, Assoc. Prof. of Department of Bridges, Constructions and Building Mechanics, KhNAHU  
PhD, Assoc. Prof. of the Department of Technologies of Road-Building Materials and Chemistry, KhNAHU

## Editorial Board Members:

**Bogomolov Viktor Oлександрovych**  
**Klets Dmytro Mykhailovych**  
**Batygin Yuriy Viktorovych**  
**Bondarenko Volodymyr Vasylovych**  
**Bugayevskiy Segey Oлександрovych**  
**Golovashchenko Sergey Fedorovich**  
**Hartmut Beckedahl**  
**Hlushkova Diana Borysivna**  
**Horbachov Petro Fedorovych**  
**Igors Tipans**  
**Ireneusz Zbicinski**  
**Klimenko Valerii Ivanovych**  
**Kozhushko Vitalii Petrovich**  
**Kozhdyzhny Volodymyr Maksymovych**  
**Kirichenko Igor Heorhiiovych**  
**Laurinavicius Alfredas**  
**Levtarov Andrii Ivanovich**  
**Mikhailenko Vladimir S.**  
**Nikitina Tatyana Borysivna**  
**Podrygalo Mykhailo Abovych**  
**Poliaturs Oлександр Vasylovych**  
**Prentkovskis Olegas**  
**Pruylutska Liudmyla**  
**Puodziukas Virgaudas**  
**Radovskiy Boris S.**  
**Roland Lachmayer**  
**Saeniko Nataliia Vitaliivna**  
**Sidi Mochammed Senouci**  
**Solodov Valery Gryhorovych**  
**Telatyev Bagdat Burkhanbauly**  
**Tolmachov Serhii Mykolaiovych**  
**Tropina Albina Albertovna**  
**Tyrsa Vira**  
**Vodovychenko Volodymyr Oлександрovych**  
**Velychko Hennadii Viktorovych**  
**Vnuukova Nataliia Volodymyrivna**  
**Volkov Volodymyr Petrovych**  
**Voronkov Oлександр Ivanovych**  
**Vrublevsky Aleksandr Nikolaevich**  
**Zhdanyuk Valerii Kuzmiovych**  
**Zolotarev Viktor Oлександрovych**

DSci, Prof., Rector of the KhNAHU;  
DSci, Prof., KhNAHU;  
DSci, Prof., Head of Physics Department, KhNAHU;  
PhD, Assoc. Prof., Professor of the Department of Philosophy and Pedagogy of Professional Training, KhNAHU;  
PhD, Assoc. Prof., Dean of Road-Construction Faculty, KhNAHU;  
PhD, Technical Leader Manufacturing Research Department, Ford Research & Advanced Engineering Research and Innovation Center, USA;  
Dr.-Ing, Prof., Bergische Universität Wuppertal, Institute of Pavement Construction and Pavement Design, Germany;  
DSci, Assoc. Prof., Head of the Department of Metal Technology and Materials Science named after O. M. Petrychenko, KhNAHU;  
DSci, Prof., Head of the Department of Transport Systems and Logistics, KhNAHU;  
Dr., Deputy Rector for International Academic Cooperation and Studies, Riga Technical University, Institute of Mechanics, Latvia;  
PhD, Prof., Department of Heat and Mass Transfer, Technical University of Lodz, Poland;  
PhD, Prof., Head of the Department of Motor Vehicles named after A. B. Hredeskul, KhNAHU;  
DSci, Prof., Head of Bridges, Constructions, Construction mechanisms Department, KhNAHU;  
DSci, Prof., Professor of the Department of Informatics and Applied Mathematics, KhNAHU;  
DSci, Prof., Dean of Mechanical Faculty, KhNAHU;  
Dr., Prof., Vice-Rector, Vilnius Gediminas Technical University, Lithuania;  
PhD, Prof., Head of the Department of Informatics and Applied Mathematics, KhNAHU;  
Dr., Prof., Pusan National University, Plasma Research Center, Busan, South Korea;  
DSci, Prof., Head of the Department of Natural Sciences and Humanities, KhNAHU;  
DSci, Prof., Head of the Department of Mechanical Engineering Technologies and Motor Vehicle Repair, KhNAHU;  
DSci, Prof., Head of the Department of Metrology and Safety of Human Vital Activities, KhNAHU;  
Dr., Prof., Dean of Faculty of Transport Engineering, Vilnius Gediminas Technical University, Lithuania;  
Ph.D., Assoc. Prof., Academic Secretary, KhNAHU;  
Ph.D., Assoc. Prof., Head of the Department of Roads, Vilnius Gediminas Technical University, Lithuania;  
Prof. Consultant, Radnat Consulting, Irvine, United States;  
PhD, Dr.-Ing, Leibniz Universität, Hannover, Institute of Product Development, Germany;  
DSci, Prof., Head of the Department of Foreign Languages, KhNAHU;  
PhD, Prof., Institut Supérieur Automobile et Transport, France;  
DSci, Prof., Head of the Department of Theoretical Mechanics and Hydraulics, KhNAHU;  
Dr., Prof., Kazakh Academy of Transport and Communication, Kazakhstan;  
DSci, Prof., Professor of the Department of Technology of Road-Construction Materials and Chemistry named after M. I. Volkov, KhNAHU;  
Dr., Prof., Texas A and M University, Department of Aerospace Engineering, United States;  
PhD, Prof., Baja California Autonomous University, Mexico  
PhD, Assoc. Prof., Prof. of Transport Technologies Department, KhNAHU;  
PhD, Chief designer of «Credo-Dialogue», Minsk;  
DSci, Prof., Professor of the Department of Ecology, KhNAHU;  
DSci, Prof., Head of the Department of Motor Vehicle Repair and Maintenance named after M. Ya. Hovorushchenko, KhNAHU;  
DSci, Prof., Professor of the Department of Internal Combustion Engines, KhNAHU;  
Prof., DSci, University of Warmia and Mazury in Olsztyn, Poland;  
DSci, Prof., Head of the Department of Road Construction and Maintenance named after O. K. Birulia, KhNAHU;  
DSci, Prof., Head of the Department of Technology of Road-Construction Materials and Chemistry named after M. I. Volkov, KhNAHU.

## «Вісник ХНАДУ»

- Входить до затвердженого ДАК Переліку наукових професійних видань України, в яких можуть бути опубліковані результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата наук.
- Реферується українським реферативним журналом «Джерело» і входить до загальнодержавної реферативної бази даних «Україніка наукова» (Україна).
- Включений до міжнародних наукометричних баз даних:
  - Index Copernicus (Польща);
  - Scholar Google.

## КОМП'ЮТЕРНІ НАУКИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 629.341

DOI: 10.30977/BUL.2219-5548.2022.96.0.7

ІНФОРМАЦІЙНА СИСТЕМА СПОСТЕРЕЖЕННЯ ЗА ПОВЕДІНКОЮ  
СТУДЕНТІВ ПІД ЧАС ПРОВЕДЕННЯ Е-ТЕСТІВАксак Н. Г.<sup>1</sup>, Татарников А. О.<sup>2</sup><sup>1</sup>Харківський національний економічний університет ім. С. Кузнеця<sup>2</sup>Харківський національний університет радіоелектроніки

**Анотація.** У статті запропоновано клієнт-серверну систему спостереження та аналізу поведінки студентів під час проведення е-тестування, яка дозволяє максимально ефективно використовувати ресурси шляхом розроблення та реалізації методів оброблення та аналізу зібраних за час спостереження даних із використанням можливостей комп'ютерного зору. Пропонується комплекс взаємопов'язаних блоків моніторингу та стеження, що містить такі основні можливості: відстеження активних URL, отримання переліку мережних підключень, стеження в режимі реального часу будь-яких змін параметрів (ширини та висоти) активного вікна, створення знімків робочої ділянки екрана комп'ютера та фотознімків за допомогою вебкамери, автоматичне зчитування вмісту буфера обміну та ведення логу натиснутих клавіш, розпізнавання обличчя студента та стеження за його положенням щодо вебкамери. У разі виявлення порушень отримані повідомлення записуються та відправляються на сервер.

**Ключові слова:** електронне тестування, система моніторингу, розпізнавання обличчя, клієнт-серверна архітектура.

**Вступ**

У зв'язку з переведенням навчального процесу на принципово новий рівень істотно зростає роль самостійної роботи студентів, яка стала основною формою отримання знань. Тому все більше стає актуальним комп'ютерне тестування як засіб контролю успішності засвоєння студентами навчальних матеріалів. Це дозволяє прискорити процес контролю та забезпечує його об'єктивність. Однак процес тестування є вразливим до можливих зловживань із боку студентів – використання допоміжних засобів, допомога сторонніх тощо.

**Аналіз наявних рішень**

Для забезпечення контролю процесу проведення комп'ютерного тестування найчастіше використовуються клієнт-серверні архітектури, серед яких можна визначити однорівневі, дворівневі та триврівневі архітектури, що належать до інструментів цифрового прокторингу. Деякі інструменти можуть отримувати доступ до віддаленого комп'ютера користувача, створювати знімки екрана, забезпечувати одночасну роботу з декількома пристроями в режимі реального часу тощо.

Так, наприклад, програмний застосунок *Екзабус* призначений для аналізу та контро-

лю поведінки користувачів за допомогою AI-алгоритмів. Перед початком тестування учень має пройти спеціальну процедуру ідентифікації за фотографією, і тільки після цього може перейти до тестування [1].

Програма *ExamCookie* відстежує комп'ютерну активність студентів під час іспиту [2]. Програма гарантує, що студенти не можуть використовувати сторонні засоби під час іспиту. Основним недоліком цієї програми є те, що учень знає про моніторинг процесу тестування та може втручатися в роботу самої системи, наприклад ввести дані іншої особи.

Система прокторингу *ProctorEdu* забезпечує онлайн-спостереження, протоколювання та оцінювання поведінки користувачів під час проходження значних онлайн-заходів [3]. Ця система може бути підключена до необхідної платформи тестування (ПК, телефон, планшет), як у автоматичному режимі, так і з безпосередньою участю проктора (представника Центру тестування, який адмініструє проведення іспиту в конкретній аудиторії закладу вищої освіти відповідно до встановленого порядку проведення іспиту).

*Exam Monitor* – це ІТ-система, розроблена для цифрового іспиту, щоб відбити бажання від шахрайства [4]. До недоліків цієї програми належать: обмежений функціонал; відсу-

тність можливості контролю в режимі реального часу; необхідність постійного вивантаження інформації для подальшого ознайомлення з результатами спостережень.

*Proctortrack* – гібридна модель, яка в режимі реального часу об'єднує віддалених людей-прокторів із розширеним штучним інтелектом. Вона дозволяє втручатися у випадку підозрілої поведінки, шахрайства чи допомоги студентів. Результати тестування цілісності аналізуються за допомогою ШІ [5]. Основним недоліком *Proctortrack* є те, що для роботи цього ПЗ необхідно придбати ліцензію.

*Microsoft Remote Desktop* (MRD) – програма для віддаленого доступу та керування комп'ютером Windows. Клієнти існують практично для всіх версій Windows (а також із Linux, Mac OS X, iOS, Android) [6]. Недоліками є обов'язкова наявність на підключеному комп'ютері Windows Pro і вище та неможливість підключення в одній локальній мережі пристроїв до віддаленого робочого стола без додаткових налаштувань.

*Remote Utilities* – програмне забезпечення для віддаленого робочого стола, яке дозволяє користувачеві віддалено керувати іншим комп'ютером через власний протокол і бачити робочий стіл віддаленого комп'ютера, керувати його клавіатурою та мишею [7].

*UltraVNC* – це потужна, проста у використанні та безкоштовна програма для віддаленого доступу до ПК, що може відображати екран іншого комп'ютера (через інтернет або мережу) на власному екрані [8] за допомогою протоколу Remote Frame Buffer (RFB). Основним же недоліком цієї програми є її складність у роботі та налаштуванні.

*AeroAdmin* – це програма, що забезпечує віддалений доступ до комп'ютера через інтернет або в локальній мережі та працює за принципом «все в одному». *AeroAdmin* дозволяє віддалено керувати комп'ютером через інтернет, зокрема неконтрольований доступ, паралельні сесії, передачу файлів та можливість підключатися до необмеженої кількості віддалених комп'ютерів [9]. Незалежно від того, чи підключений користувачький комп'ютер або серверний, користувач може встановлювати обмеження та права доступу: тільки перегляд, захоплення клавіатури та миші, файловий менеджер, синхронізація буфера обміну. Зі свого боку адміністратор може додатково встановлювати права доступу для кожного учасника робочої сесії.

*DameWare* – метод віддаленого контролю

за комп'ютером користувача, який використовується як засіб технічної підтримки користувачів [10]. Серед недоліків можна виділити обмежений базовий функціонал у процесі використання безкоштовної версії.

*Chrome Remote Desktop* (CRD) є програмним забезпеченням для віддаленого підключення до комп'ютера користувача за допомогою браузера Chrome або пристроїв Chromebook [11]. Основним недоліком *Chrome Remote Desktop* є те, що його робота не можлива без встановлення та використання веббраузера Google Chrome.

*AnyDesk* – програма для віддаленого робочого стола із закритим вихідним кодом, що поширюється AnyDesk Software GmbH [12]. Програмне забезпечення забезпечує незалежний від платформи віддалений доступ до персональних комп'ютерів та інших пристроїв, на яких запущено хост-програму. До ключових недоліків програми *AnyDesk* можна віднести обмежений функціонал та збільшене навантаження на викладача через необхідність постійного спостереження за великою кількістю комп'ютерів.

Отже, кожна з розглянутих систем має свої переваги й такі недоліки:

- у процесі тестування студент знає про процес моніторингу, система не має можливості підтримки процесу контролю в режимі реального часу;
- обмежений для використання функціонал під час використання безкоштовної ліцензії;
- відсутність для студентів можливості ознайомлення із зібраними в процесі спостереження результатами після закінчення самого тестування;
- платна ліцензія.

### **Мета та постановка завдання**

Метою є створення клієнт-серверної системи аналізу поведінки студентів під час проведення е-тестування на основі розпізнавання обличчя.

Для досягнення поставленої мети необхідно розробити модель процесу спостереження за поведінкою студентів (ССПС) під час проведення е-тестів, її складових компонентів та їхню взаємодію.

### **Модель процесу спостереження за поведінкою студентів під час проведення е-тестів**

Процес спостереження за поведінкою студентів під час проведення е-тестів може бути поданий трирівневою структурою, що опису-

ється моделлю (1), яка містить такі компоненти: клієнтська частина, блок синхронізації із сервером та серверна частина (рис. 1).

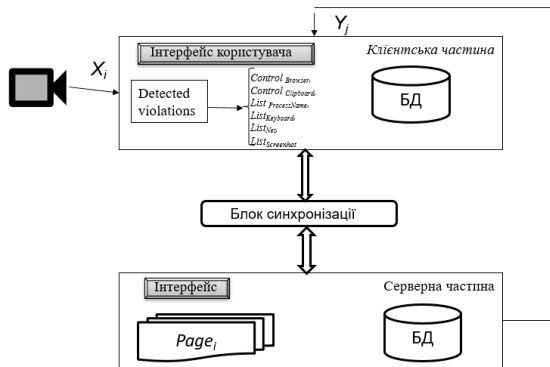


Рис. 1. Модель процесу спостереження за поведінкою студентів під час проведення е-тестів

Модель процесу спостереження за поведінкою студентів під час проведення е-тестів виражається як перетворення вхідних значень  $X$  у вихідні величини  $Y$ :

$$Z \subset X \times Y, \quad (1)$$

де  $X = \{X_j(i)\}$ , ( $i = \overline{1, N}$ ,  $N$  – кількість користувачів, що проходять тестування) – вхідні дані з комп'ютера користувача:  $X_1$  – вміст буфера обміну;  $X_2$  – активні URL у різних типах браузерів;  $X_3$  – список мережних процесів;  $X_4$  – скриншоти всієї ділянки екрана комп'ютера;  $X_5$  – натиснуті клавіші на клавіатурі;  $X_6$  – розміри вікна;  $X_7$  – список активних процесів;  $X_8$  – знімки з вебкамери;  $Y$  – виявлені порушення [13].

Для вихідної величини  $Y$  побудована множина завдань, розв'язання яких належить множині  $D_z = \{D_{user}, D_{sin}, D_{server}\}$ : тут  $D_{user} = \{Cont_{Browser}, Cont_{Clipboard}, List_{ProcessName}, List_{Keyboard}, List_{Net}, List_{Screenshot}\}$ , де  $Cont_{Browser}$  – завдання визначення активного URL у різних типах браузерів;  $Cont_{Clipboard}$  – завдання відстеження змін вмісту буфера обміну;  $List_{ProcessName}$  – завдання отримання списку активних процесів на комп'ютері користувача;  $List_{Keyboard}$  – завдання оброблення та логування натиснутих користувачем клавіш на клавіатурі;  $List_{Net}$  – завдання отримання списку мережних процесів;  $List_{Screenshot}$  – завдання створення скриншотів всієї ділянки екрана комп'ютера та знімків за допомогою вебкамери [14] за умови зміни активного

вікна програми або його розмірів;  $D_{sin}$  – завдання синхронізації клієнтської та серверної частини;  $D_{server} = \{Page_k\}$ ,  $k = 1, 2, 3$ , де  $Page_1$  – завдання створення сторінки, де відображається інформація про всі дії, які відбуваються на клієнтських машинах;  $Page_2$  – завдання створення сторінки роботи з текстовими даними, що дозволяє ознайомитися з текстовими результатами, які були зібрані за час спостереження;  $Page_3$  – завдання створення сторінки роботи із зображеннями, яке дозволяє користувачеві ознайомитися з фотознімками та скриншотами, що були зроблені під час тестування.

Відображення  $T: X \rightarrow Y$  дозволяє для кожного  $X(i)$  знайти таке  $Y_m \in Y$  ( $m = \overline{1, Q}$ ,  $Q$  – кількість порушень), що є розв'язкою завдання  $D_z$ .

Значення  $Y_m \in Y$  використовуються для прийняття рішення про порушення та вироблення подальшої тактики поведінки.

Користувацький інтерфейс є програмою, де на вхід подається зчитаний із сервера набір інструкцій, за допомогою яких активуються необхідні для спостереження функції. Зчитана інформація – це функції детектування та пошуку порушень. Функція детектування порівнює зчитані результати з попередніми та еталонними. Вихідним результатом є згрупований масив, що містить: ідентифікатор користувача, повідомлення про порушення, таблиці з інформацією про порушення на комп'ютері студента та на сервері. Процес прийняття рішення про порушення є «порівнянням» отриманих проміжних результатів із попередніми та з еталонними значеннями, що були передані із сервера перед початком тестування. На сервері після отримання повідомлення можна ознайомитися із самим порушенням, для цього реалізований спеціальний web-інтерфейс. Та в разі необхідності проктор може відправити повідомлення з попередженням конкретному студенту, або якщо потрібно змінити налаштування клієнтських застосунків.

Для оцінювання якості процесу функціонування запропонованої системи використовується сукупність критеріїв оцінки ефективності  $K = \{k_1, k_2\}$ , що дозволяє визначити реакцію детектування порушень.

Першим критерієм є час детектування порушень  $k_1: \tau = \min(\tau^{Br}, \tau^{Cl}, \tau^{Pr}, \tau^{key}, \tau^{Net}, \tau^{Scr})$ , де  $\tau^{Br}$  – час визначення активного URL у різних типах браузерів;  $\tau^{Cl}$  – час відстеження

змін вмісту буфера обміну;  $\tau^{Pr}$  – час отримання списку активних процесів на комп'ютері користувача;  $\tau^{key}$  – час оброблення та логування натиснутих користувачем клавіш на клавіатурі;  $\tau^{Net}$  – час отримання списку мережних процесів;  $\tau^{Scr}$  – час створення скриншотів усієї ділянки екрана комп'ютера та знімків за допомогою вебкамери за умови зміни активного вікна програми або його розмірів.

Співвідношення множини реально створених сторінок  $Page_k$  ( $i=1,2,3$ ) визначається такими характеристиками:  $\Psi_k = \{\rho_k, e_k\}$ , де

$$\rho_k = \frac{a}{a+c} \quad \text{– коефіцієнт повноти, що}$$

характеризує інформацію про дії на клієнтській машині, яка відображена на побудованих сторінках  $Page_k$  до загальної

кількості дій;  $e_k = \frac{b}{a+b}$  – коефіцієнт шуму,

що характеризує частку наданої інформації без порушень;  $a$  – кількість зафіксованих порушень;  $b$  – кількість поведінки без порушень;  $c$  – кількість загальних порушень.

Для ефективного функціонування пропонуваної моделі необхідно, щоб система відповідала таким вимогам:

$$k_2 : \forall (D_j \in D_z) \left[ (\tau < \tau^{\max}) \& (\rho_k \rightarrow 1) \& \right. \\ \left. \& (e_k \rightarrow 0) \right] \Rightarrow Y, \quad (2)$$

де  $D_j$  – підзавдання загального завдання  $D_z$ ;  $\tau^{\max}$  – максимально допустимий час вирішення завдання.

### Моделювання системи спостереження за поведінкою студентів під час е-тестування

Серверна та клієнтська частина програми для моніторингу тестування студентів призначена для використання на ПК під управлінням ОС Windows 10 (32/64-розрядні версії). В ОС має бути встановлений Connector/NET 8.0.20 або вище, та NET Framework версії 4.5 або вище.

Розроблюваний програмний застосунок має змогу працювати в повністю автономному режимі, за винятком ситуацій, коли необхідно виконати зміну порту підключення до серверного застосунку [15]. Також ПЗ має змогу віддаленого контролю за всіма підключеними користувачами та дозволяє відправляти конкретним користувачам повідомлення про порушення.

Програмне рішення, створене в Visual Studio, складається з десяти проєктів (рис. 2).

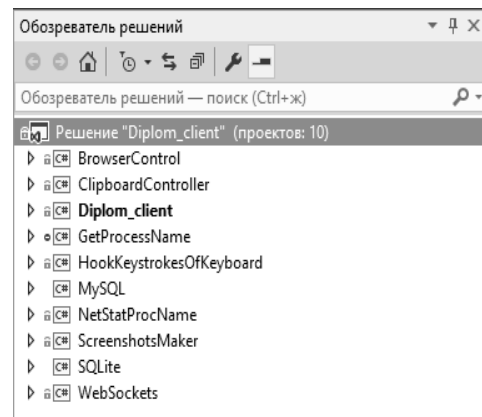


Рис. 2. Файлова структура проєкту

У випадку з використанням декількох проєктів поділ вихідного коду реалізується за допомогою поділу функціоналу програмного рішення на окремі складники, що реалізуються в різних за призначенням проєктах. Це дозволяє розбити функціонал клієнтського застосунку на окремі функціональні блоки. Такий підхід спрощує, прискорює роботу над кожним окремим елементом та дозволяє спростити процес оптимізації кожної окремої частини й усього проєкту загалом.

Файлова структура серверної частини відповідає стандартній структурі застосунків, що розробляються з використанням можливостей CMS WordPress. Файлова структура вихідного коду вебзастосунку складається з великої кількості стандартних файлів самої CMS та файлів розробленої «теми», в якій описані функціональні можливості програми, зокрема інтерфейс користувача, логіка доступу та оброблення даних, робота з користувачами тощо.

Для роботи CMS WordPress використовує трирівневу архітектуру MVC, тому що це найбільш широко використовуваний та надійний шаблон проєктування. Ця архітектура не накладає жодних обмежень на спосіб реалізації інтерфейсу користувача та забезпечує поділ відповідальності між функціональними блоками.

Вебзастосунок (серверна частина програми) містить три основні сторінки, з якими може працювати користувач.

На основній сторінці відображається інформація про всі дії, що відбуваються на клієнтських машинах, та є змога надсилати повідомлення клієнтам у процесі тестування. Сторінка роботи з текстовими даними дозво-

ляє ознайомитися з текстовими результатами, що були зібрані за час спостереження. Сторінка роботи із зображеннями дозволяє користувачеві ознайомитися з фотознімками та скріншотами, що були зроблені під час тестування.

Для роботи з різними браузерами використовуються класи Chrome, Firefox, Opera, Microsoft Edge, кожен з яких реалізує функціонал отримання активного URL.

Для початку детектування необхідно спочатку становити на комп'ютері користувача приховану програму «клієнт». Перед початком тестування із сервера відправляється команда про увімкнення спостереження.

Користувач не може взаємодіяти з інтерфейсом клієнтського застосунку, і тому не знає що за ним спостерігають. У цей час із сервера на клієнтські застосунки відправляється команда, «стеження за натиснутими клавішами».

Через 5 секунд на сервер починають надходити повідомлення про детектування порушень із боку студентів. Проктор може бачити ідентифікатор користувача, час детектування та тип порушення. На рис. 3 зображено вікно детектування порушень у режимі реального часу. Система надіслала повідомлення, що користувач у процесі тестування користується клавіатурою, що є недопустимим.

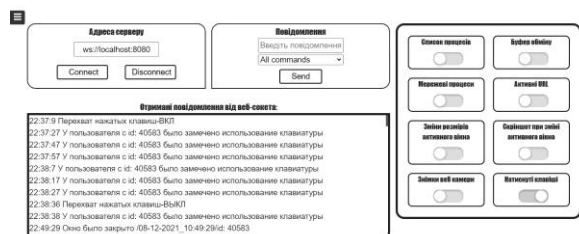


Рис. 3. Детектування порушень у режимі реального часу

Для того щоб відправити попередження конкретному студенту, проктору необхідно ввести ідентифікатор користувача й повідомлення в поле блоку «Повідомлення» та натиснути кнопку «Send». Після чого студент отримає попередження (рис. 4).

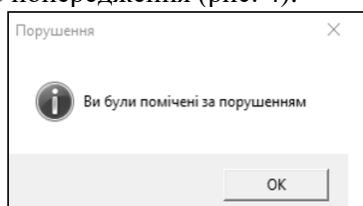


Рис. 4. Попередження про порушення

Після отримання такого повідомлення можна продовжити спостереження або завершити тестування.

Тестування програми проводилося на двох різних комп'ютерах із різними версіями Visual Studio, а саме в Visual Studio 2017, Visual Studio 2015 та Visual Studio 2010, де була перевірена основна функціональність програми й здійснена перевірка на сумісність з іншими версіями компілятора Visual Studio.

Для моніторингу за користувачами застосунків автоматично підключається до віддаленого сервера.

Під час проведення експериментів був проведений моніторинг 100 студентів із використанням запропонованої системи та декількома популярними аналогами, а саме AnyDesk, CRD, MRD. Були зафіксовані 12 порушень: (1) створення нового вікна браузера; (2) відкриття нової вкладки браузера; (3) зміна розмірів головного вікна; (4) увімкнення сторонніх утиліт; (5) підключення через віддалений робочий стіл; (6) пошук відповіді за запитом в інтернеті; (7) використання іншого монітора; (8) використання телефона; (9) допомога сторонньої особи; (10) поворот голови від екрана; (11) переключення між різними програмами або браузерами; (12) вимкнення системи моніторингу. Однак не всі системи змогли детектувати порушення (3), (4), (5), (8), (9), (10). Результати детектування порушень подано в табл. 1.

Таблиця 1 – Порівняльний аналіз детектування найскладніших порушень

Порушення	Кількість порушень (детектованих / загальна)			
	CRD	MRD	AnyDesk	ССПС
(3)	0/19	0/19	0/19	<b>19/19</b>
(4)	0/3	0/3	0/3	<b>2/3</b>
(5)	0/2	0/2	0/2	<b>1/2</b>
(8)	0/9	0/9	0/9	<b>9/9</b>
(9)	0/5	0/5	0/5	<b>5/5</b>
(10)	0/11	0/11	0/11	<b>10/11</b>

На відміну від аналогів, розроблена система спостереження за поведінкою студентів може виявляти значно більшу кількість порушень. Крім того, для оцінювання ефективності роботи запропонованої системи ССПС використовується сукупність критеріїв, що дозволяють визначити реакцію детектування

порушень. Основним критерієм, завдяки якому досягається високий рівень точності детектування, є час детектування порушень. На відміну від аналогів, у розробленій системі є можливість налаштування часових інтервалів для функцій детектування.

### Висновки

Запропоновано модель процесу спостереження за поведінкою студентів на основі розпізнавання обличчя, яка дозволяє в режимі реального часу відстежувати порушення під час проведення е-тестування. Для реалізації системи моніторингу на основі запропонованої моделі використана мова програмування C# на платформі .NET (клієнтська частина) та CMS WordPress (серверна частина). Інтерфейс користувача та організація взаємодії з ним реалізується за допомогою основних можливостей HTML, CSS, JavaScript, PHP та C#.

Описано принципи роботи розробленої системи та зроблений порівняльний аналіз з наявними аналогами.

### Література

1. Экзамус. URL: <https://ru.examus.net/> (дата звернення: 14.12.2021).
2. ExamCookie. URL: <https://www.exam-cookie.dk> (дата звернення: 15.12.2021).
3. ProctorEdu. URL: <https://proctored.ru/> (дата звернення: 15.12.2021).
4. Exam Monitor. URL: <https://sdu.exammonitor.dk/> (дата звернення: 16.12.2021).
5. Proctortrack. URL: <https://www.proctortrack.com/proctorlive-ai/> (дата звернення: 16.12.2021).
6. Microsoft Remote Desktop. URL: <https://www.microsoft.com/en-us/p/microsoft-remote-desktop/9wzdncrfj3ps> (дата звернення: 17.12.2021).
7. Remote Utilities. URL: <https://www.remotetools.com/> (дата звернення: 17.12.2021).
8. UltraVNC. URL: <https://uvnc.com/> (дата звернення: 18.12.2021).
9. AeroAdmin. URL: <https://www.aeroadmin.com/ru/> (дата звернення: 18.12.2021).
10. DameWare. URL: <https://www.dameware.com/> (дата звернення: 19.12.2021).
11. Chrome Remote Desktop. URL: <https://remotedesktop.google.com/> (дата звернення: 19.12.2021).
12. AnyDesk. URL: <https://anydesk.com/> (дата звернення: 20.12.2021).
13. Cloud-fog-dew Architecture for Personalized Service-oriented Systems / N. Axak, D. Rosinskiy, O. Barkovska, I. Novoseltsev // The 9th IEEE International Conference on Dependable

Systems, Services and Technologies, DESSERT'2018, Kyiv, Ukraine, 2018. P. 80–84.

14. Татарников А.О., Аксак Н.Г. Дослідження методів обробки та класифікації емоцій людини при розробці систем комп'ютерного зору. *Теорія і практика сучасної науки: Матеріали II міжнародної науково-теоретичної конференції* (м. Краків, 12 листопада 2021). Краків, 2021. С. 91–92.
15. Іващенко Г.С., Татарников А.О. Організація взаємодії та синхронізації даних між sqlite і mysql. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: Матеріали десятої міжнародної науково-технічної конференції* (м. Харків, 9–10 квітня 2020). Харків, 2020. С. 70.

### References

1. Examus. URL: <https://ru.examus.net/> (accessed: 14.12.2021).
2. ExamCookie. URL: <https://www.exam-cookie.dk> (accessed: 15.12.2021).
3. ProctorEdu. URL: <https://proctored.ru/> (accessed: 15.12.2021).
4. Exam Monitor. URL: <https://sdu.exammonitor.dk/> (accessed: 16.12.2021).
5. Proctortrack. URL: <https://www.proctortrack.com/proctorlive-ai/> (accessed: 16.12.2021).
6. Microsoft Remote Desktop. URL: <https://www.microsoft.com/en-us/p/microsoft-remote-desktop/9wzdncrfj3ps> (accessed: 17.12.2021).
7. Remote Utilities. URL: <https://www.remotetools.com/> (accessed: 17.12.2021).
8. UltraVNC. URL: <https://uvnc.com/> (accessed: 18.12.2021).
9. AeroAdmin. URL: <https://www.aeroadmin.com/ru/> (accessed: 18.12.2021).
10. DameWare. URL: <https://www.dameware.com/> (accessed: 19.12.2021).
11. Chrome Remote Desktop. URL: <https://remotedesktop.google.com/> (accessed: 19.12.2021).
12. AnyDesk. URL: <https://anydesk.com/> (accessed: 20.12.2021).
13. Cloud-fog-dew Architecture for Personalized Service-oriented Systems / N. Axak, D. Rosinskiy, O. Barkovska, I. Novoseltsev // The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2018, Kyiv, Ukraine, 2018. P. 80–84.
14. Tatarnykov A.O., Axak N.G. The research of methods of processing and classification of the human emotions in the development of the computer vision systems. *Theory and practice of modern science: Proceedings of the II International Scientific and Theoretical Conference* (Krakow, November 12 2021). Krakow, 2021. P. 91–92 [in Ukrainian].



15. Ivashchenko H.S., Tatarnykov A.O. Organization of interaction and synchronization of data between sqlite and mysql. *Modern directions of development of information and communication technologies and means of management*.: Proceedings of the tenth international scientific and technical conference (Kharkiv, April 9–10 2020). Kharkiv, 2020. С. 70. [in Ukrainian].

**Аксак Наталія Георгіївна**, д.т.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, nataliia.axak@nure.ua, тел. +38 050-142-18-80,

**Татарников Андрій Олександрович**, магістрант, Харківський національний університет радіоелектроніки, andrii.tatarnykov@nure.ua, тел. +38 050-766-18-85.

#### **The system of monitoring students' behavior during e-tests**

**Abstract. Problem.** In connection with the transfer of the educational process to a fundamentally new level, the role of independent work of students, which has become the main form of the knowledge, is growing significantly. Therefore, the computer testing is becoming increasingly important as a means of monitoring the students' learning materials. These speeds up the control process and ensures its objectivity. However, the testing process is vulnerable to possible students' abuse – use of aids, outside help, etc. **Goal.** The goal is developing a client-server system for analyzing students' behavior during e-testing. **Methodology.** Solution of the set tasks involved application of the set-theoretical approach in order to build up the model of an observation process for students' behavior and their component organization

as well as their interaction, image processing methods. **Results.** A model of the monitoring process of the students' behavior is proposed, which allows real-time monitoring of violations during e-testing based on face recognition. The principle of operation of the developed system also described comparative analysis with existing analogues. **Originality.** A monitoring system is proposed, which includes the following features: tracking the active URLs, obtaining the list of network connections, real-time tracking of any changes in parameters (width and height) of the active window, creation of screenshots of the working area of the computer screen and photographs using a web camera, automatic reading of the contents of the clipboard and keeping a log of pressed keys, taking screenshots of the working area of the computer screen and photographs using a web camera, recognizing the student's face and tracking their position regarding the webcam. If violations are detected, the received messages are recorded and sent to the server. **Practical value.** The use of the proposed monitoring system for students' behavior will allow to remotely identify a larger number of violations that were not previously detected.

**Key words:** electronic testing, monitoring system, recognition.

**Axak Natalia**, professor, Doct. of Science, Department of information systems, Simon Kuznets Kharkiv National University of Economics, nataliia.axak@nure.ua tel. +38 050-142-18-80, **Tatarnykov Andrii**, undergraduate, Kharkiv National University of Radioelectronics, tel. 38 050-766-18-85, andrii.tatarnykov@nure.ua.

---

## ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ МОНІТОРИНГУ ТА АНАЛІЗУ СТАНУ IP-МЕРЕЖІ НА ОСНОВІ ВИКОРИСТАННЯ ПРОТОКОЛУ SNMP

Голубничий Д. Ю., Коцюба В. П.

Харківський національний економічний університет ім. С. Кузнеця

***Анотація.** Проведений аналіз технологій контролю, діагностики та моніторингу обладнання комп'ютерних мереж. Обґрунтовано, що найбільш ефективним і надійним засобом, що дозволяє виконувати завдання з управління IP-мережними пристроями, є протокол SNMP. Наведені архітектура й компоненти SNMP та алгоритми практичного налаштування SNMP на комп'ютері з ОС Windows та Linux.*

***Ключові слова:** системи моніторингу, управління обладнанням, протокол мережного моніторингу, налаштування SNMP, MIB.*

### Вступ

Сучасні комп'ютерні мережі дозволяють передавати потоковий трафік (передачу даних, відео, голосу тощо) будуються на основі застосування інфокомунікаційних технологій, які об'єднали в собі системи електров'язку, передачі даних, автоматизовані системи управління, засоби комп'ютерних мереж, оброблення даних, інформаційно-вимірвальні системи, бази даних та ін.

Підтримувати надійне функціонування та взаємодію глобальних та локальних (відомчих) комп'ютерних мереж, забезпечення виконання визначених вимог щодо надійності й ефективності їхньої роботи та складових елементів можливе за рахунок застосування систем і засобів контролю, керування, моніторингу та аналізу стану IP-мереж [1].

### Аналіз публікацій

Системи моніторингу здійснюють управління комп'ютерною мережею в автоматизованому режимі, а рішення, що базуються на основі підготовленої інформації системою моніторингу про роботу обладнання (хостів, комутаторів, маршрутизаторів, серверів, модемів тощо) приймає адміністратор мережі.

Перевірку підключеного мережного обладнання та його доступність в IP-мережі можна здійснити за допомогою утиліти ping, яка використовується із стандартними скриптами. Для моніторингу мережі відправляються ICMP-запити й зворотно присилаються ICMP-відповіді, що устаткування доступне й підключене в єдину IP-мережу [2].

Програмне забезпечення Ping infoview є невеликою утилітою для опитування вузлів за іменами або IP-адресами з можливістю встановлення кількості та інтервалів пінгування. Вона є аналогом стандартної консольної програми ping.exe.

Звичайно, такий спосіб простий, надійний і часто застосовується тими, хто займається налагодженням цифрового обладнання IP-мереж та мережними адміністраторами, але якщо потрібно стежити за значеннями конкретних параметрів мережного обладнання, відповідністю параметрів і помилок, необхідно використовувати більш складні та надійні системи контролю.

На сучасному етапі для аналізу й діагностики комп'ютерних мереж застосовуються такі технології, програми та протоколи:

1. Вбудовані системи діагностики й управління (Embedded systems) – система управління та моніторингу обладнання локальних IP-мереж System Center Operations Manager (SCOM), яка забезпечує збір даних, їхній аналіз та зберігання на сервері баз даних Microsoft SQL Server. Ці системи виконуються у вигляді програмно-апаратних модулів, що встановлюються в комунікаційне обладнання, а також у вигляді програмних модулів, вбудованих в операційні системи Windows, UNIX та Linux.

Однак у цієї системи є низка недоліків, а саме система моніторингу охоплює багато загальних показників системи, але не придатна для стеження за визначеними параметрами, специфічність й складність під час налаштування [2, 3].

2. Система моніторингу Zabbix використовується для комплексного моніторингу мережного обладнання, серверів та сервісів. Складається із сервера моніторингу (ядра), що виконує періодичне опитування та отримання даних, обробляє їх, аналізує й здійснює запуск скриптів для розсилання повідомлень. Zabbix-проксі є надійним рішенням для централізованого віддаленого моніторингу.

нгу місць, філій, мереж, що не мають локальних адміністраторів й автоматичне виявлення за діапазоном IP-адрес.

Система моніторингу Zabbix має специфічні недоліки, такі як громіздкість сервісу, необхідність встановлення програмного забезпечення на все обладнання мережі тощо [3].

3. Система моніторингу Nagios – програма для моніторингу систем і мереж, працює також і під Sun Solaris, FreeBSD, AIX і HP-UX. За допомогою цієї програми доступний моніторинг стану хостів (завантаження процесора, використання диска, системні логи тощо), підтримка віддаленого моніторингу через шифровані тунелі SSH або SSL, можливість побудови карт мереж виявлення проблем відразу після їхнього виникнення та моніторинг безпеки системи [2, 3].

4. Система моніторингу Cacti дозволяє збирати статичні дані за певні часові інтервали й відтворювати їх у графічному вигляді за допомогою RRDtool-утиліти, написана в інфраструктурі Apache-PHP-MySQL, дозволяє налаштовувати збір і відображення даних моніторингу на основі вебінтерфейсу й заздалегідь складеного набору графіків даних за останній день, тиждень, місяць і рік. Є можливість довільно задавати часовий проміжок, за який буде згенерований графік.

Недоліками системи Cacti є швидке наростання кількості однотипних налаштувань у разі значної кількості середовищ і серверів, а також обмежена продуктивність деяких JMX рішень для Cacti.

5. Аналізатори протоколів (Protocol analyzers) є програмними або апаратно-програмними системами. Програмне забезпечення аналізатора складається з ядра, що підтримує роботу мережного адаптера та програмного забезпечення, який декодує протоколи канального рівня й протоколи верхніх рівнів, таких як IP, TCP, ftp, telnet, HTTP тощо. Суттєвим недоліком аналізаторів є те, що вони обмежуються лише функціями моніторингу й аналізу трафіку в мережах.

6. Одним із найбільш зручних, надійних, простих і швидких засобів, який дозволяє виконувати практично весь спектр завдань управління мережними пристроями, є протокол для управління пристроями в IP-мережах Simple Network Management Protocol (SNMP).

SNMP визначений Інженерною радою інтернету (IETF) як компонент стеку TCP/IP.

Основною концепцією протоколу є те, що вся необхідна для управління пристроєм ін-

формація зберігається на самому пристрої – будь то сервер, модем або маршрутизатор – у так званій адміністративній базі даних. Ці змінні можуть відтворювати такі параметри, як кількість пакетів оброблених пристроєм, стан його інтерфейсів, час й особливості функціонування пристрою тощо [3, 4].

За своєю сутністю моніторинг – це комплекс швидкого знаходження проблеми, оповіщення про неї адміністратора й діагностики, що дає повну й точну інформацію про несправну роботу IP-мережі. Необхідність вчасного прогнозування, запобігання поломок, оповіщення про них, зберігання інформації про стан обладнання в комп'ютерній мережі та оперативне усунення виявлених несправностей підтверджує актуальність цієї роботи.

#### Мета та постановка завдання

Метою роботи є аналіз принципів застосування технологій, програм і протоколів, що дозволяють керувати обладнанням і підтримувати надійну роботу комп'ютерної мережі.

Для досягнення поставленої мети необхідно розкрити особливості функціонування системи моніторингу мережного обладнання, побудованого на основі архітектур UDP/TCP та застосування мережних протоколів контролю й управління – SNMP відповідно до змісту рекомендації RFC 1155, 3584, 3411 тощо [4–7].

#### Виклад основного матеріалу

Усю необхідну інформацію протокол SNMP отримує з бази керуючої інформації Management Information Base (MIB). MIB є базою даних стандартизованої структури. База даних має деревоподібну структуру, а всі змінні класифіковані за тематикою. Кожне піддерево містить певну тематичну підгрупу змінних. Найбільш важливі компоненти, що відповідають за роботу мережних вузлів, об'єднані в підгрупі MIB-II. Існують два типи MIB: стандартні й фірмові. Стандартні MIB визначені комісією з діяльності інтернет Internet Activity Board (IAB), а фірмові – виробником пристрою.

У таблиці 1 наведено перелік найбільш поширених стандартів баз керуючої інформації. У базах даних, зазначених у таблиці 1, є багато змінних, які можуть бути корисні для діагностування мережі та мережного обладнання.

Наприклад, використовуючи MIB-II, можливо отримати відомості про загальну кількість пакетів, переданих мережним інтерфейсом, а за допомогою MIB-повторювача можна дізнатися інформацію про кількість колізій порту.

Таблиця 1 – Перелік основних стандартів баз керуючої інформації

База	Призначення
MIB-II	Задає значну кількість об'єктів, що можуть бути використані для управління мережними інтерфейсами
MIB-повторювача	Включена до підмножини MIB-II. Установлює об'єкти, які можна використовувати для керування повторювачем
MIB-мосту	Включена до підмножини MIB-II. Задає дані, які можна використовувати для керування мостом
RMON MIB	Вказує об'єкти даних, які можна використовувати для управління мережею загалом за допомогою протоколу Remote Network Monitoring (RMON)

У MIB кожен об'єкт має ім'я та тип. Ім'я об'єкта характеризує його становище в дереві MIB. У цьому разі ім'я дочірнього вузла містить ім'я батьківського вузла й задається цілим числом [6].

### Компоненти SNMP

Для найменування змінних бази MIB і однозначного визначення їхніх форматів використовується додаткова специфікація, що називається SMI – Структура управління інформацією. Наприклад, специфікація SMI включається як стандартне ім'я IpAddress і визначає його формат як рядок з 4 байтів. Інший приклад – ім'я Counter, для якого визначено формат у вигляді цілого числа в діапазоні від 0 до  $2^{32}$ .

Існує три компоненти SNMP, за допомогою яких він виконує свої основні завдання, наведені на рис. 1.

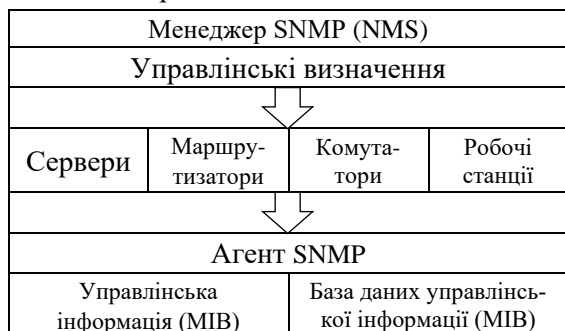


Рис. 1. Структура SNMP

1) Менеджер SNMP. Це централізована система вузлів на основі графічного інтерфейсу, яка використовується для моніторингу мережі. Її також називають Системою управління мережею Network management system (NMS). Він взаємодіє з двоспрямованим потоком інформації між вузлом NMS та елементами мережі. Елементами мережі можуть бути комутатори, комп'ютери (хости), маршрутизатори, сервери, IP-телефони, модеми, IP-відеокамери тощо.

2) Агент SNMP. Агент – це модуль програмного забезпечення для управління мережею, який встановлюється на мережному пристрої, такому як хост, сервер, маршрутизатор тощо. Агент підтримує базу даних про елементи мережі, які дистанційно керуються. Коли NMS запитує будь-яку інформацію, вона повертається разом із даними, що зберігалися в базі даних до NMS.

Якщо будь-яка пастка або помилка зустрічається агентом на керованому пристрої, він надсилає повідомлення про пастку менеджеру SNMP із зазначенням стану в реальному часі.

3) База даних управлінської інформації (MIB). Кожен з агентів SNMP підтримує інформаційну базу даних для керованих пристроїв, яка пояснює параметри пристроїв.

Менеджер SNMP використовує цю базу даних, щоб запитати в агента інформацію про конкретний пристрій для NMS. Отже, ця спільна інформація між агентом та менеджером відома як База даних MIB.

Структура MIB:

– це група інформації, що містить змінні, що містять значення, відповідні параметрам елемента мережі в його сховищах. Ці змінні відомі як керовані об'єкти та ідентифікуються ідентифікатором об'єкта (OID);

– MIB – це сукупність ідентифікаторів об'єктів у ієрархічному форматі, і кожен може ідентифікувати змінну, яку SNMP може встановити або прочитати;

– OID бувають двох видів – скалярні та таблицні. Скаляр повідомляє лише про один випадок події, що означає, що результат – лише один. Приклад: текст або номер;

– таблицний об'єкт – це таблиця, яка є пулом усіх пов'язаних OID і, отже, дає кілька результатів для одного значення об'єкта. Наприклад: для подвійного процесора центрального процесора це призведе до отримання двох значень.

Оскільки SNMP працює на прикладному рівні пакета протоколів TCP/IP, тому всі по-

відомлення SNMP будуть транспортуватися через протокол UDP (User Datagram Protocol). UDP-порт 161 використовується агентом SNMP для отримання запиту від менеджера. Однак менеджер може також надіслати запит на будь-який інший порт, який доступний крім цього.

Менеджер отримує відповідь у вигляді повідомлень, таких як повідомлення «Trap» та «Inform» на порт 162 UDP. NMS буде виконувати всі операції моніторингу та управління мережними пристроями / елементами та надавати основні дані, які використовуються для управління мережею [5, 6].

Агент SNMP, пов'язаний з кожним із керованих у мережі елементів, перекладає локальні дані MIB, такі як дані про продуктивність, інформацію про помилки, появу будь-якої події, у зручну форму для NMS. Для цього агент використовує Get-Requests, що доставляють дані до програмного забезпечення NMS. Мережні елементи, такі як маршрутизатори, комутатори, комп'ютери, модеми тощо, збирають і зберігають дані MIB, а за допомогою агента SNMP він робить їх доступними для сумісних із ними систем управління (рис. 2).

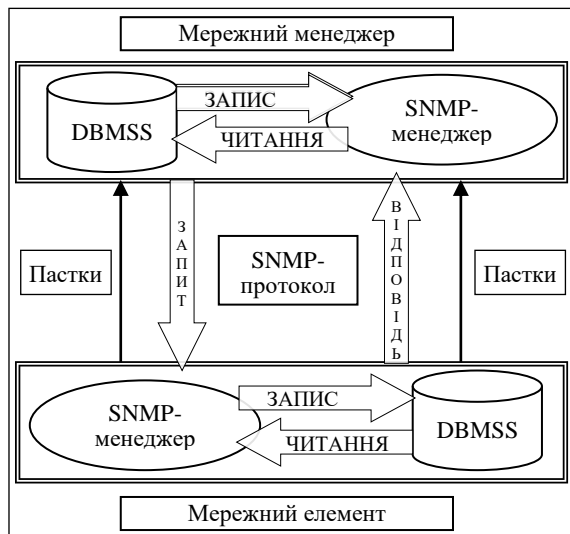


Рис. 1. Схема архітектури SNMP

Network Manager – це програмне забезпечення з відкритим кодом, таке як Solar winds та Cisco IOS. Для запуску SNMP менеджер мережі має встановити це програмне забезпечення на сервері. Основним завданням менеджера простого управління мережним протоколом є запит та отримання даних від агента для управління та моніторингу елементів мережі (рис. 2). Крім того, є можливість

редагування конфігурації, коли це потрібно відповідно до вимог мережі.

Іншим важливим завданням є отримання повідомлень «Trap» та «Inform» щодо несправностей і виникнення подій у мережі.

Команди SNMP.

Розгортаючи SNMP, елементами мережі керують за допомогою трьох команд: читання, запис та пастка:

- команда читання використовується NMS для моніторингу керованих мережних елементів, таких як маршрутизатори, комутатори тощо. Цю дію завершує NMS, вивчаючи різні змінні, що підтримуються елементами мережі;

- команда запису використовується NMS для управління мережними елементами. За допомогою цієї команди NMS може змінювати значення змінних, що зберігаються в керованих мережних елементах;

- команда пастка використовується керованими мережними елементами, щоб повідомляти про випадки й помилки в системі управління [3, 7].

Повідомлення запиту SNMP, які є PDU, передбачають такі операції, як отримати, GetNext та GetBulk:

- Отримати. Використовуючи це повідомлення, запит NMS на отримання понад однієї змінної з агента SNMP;

- GetNext. Ця операція дозволяє новій системі управління отримувати одну або більше наступних змінних з агента SNMP;

- GetBulk. Ця операція відповідає послідовній операції GetNext. За допомогою цього набору повідомлень запитів ми можемо отримати базу даних від агента.

Відповідь: він повертає змінну одиницю даних від агента до NMS у відповідь на PDU запит на отримання та встановлення (рис. 3).

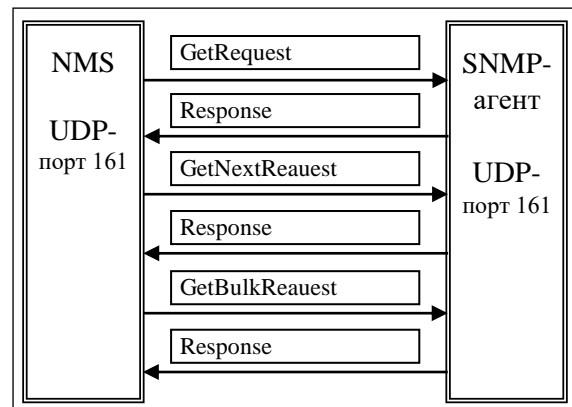


Рис. 2. Схема взаємодії за протоколом SNMP

Пастка. Ця команда ініціюється агентами SNMP. Коли подія відбувається, агент надсилає сигнал менеджеру SNMP для підтвердження події у формі цього PDU.

InformRequest. Його функція така сама, як і в команди «Trap». Він містить підтвердження отримання пакету від менеджера SNMP.

Коли в мережі відбувається подія, тоді SNMP «Trap» повідомляє про це менеджера SNMP.

Наприклад, перехід порту зі стану DOWN у стан UP у маршрутизаторі. Інформація SNMP – це також пастки SNMP, які є квитанцією про підтвердження від менеджера.

На рис. 4 показано зв'язок між керованими SNMP елементами мережі та менеджером надсилання пасток і повідомлень. Функціональність «Trap» та «Inform» відрізняється.

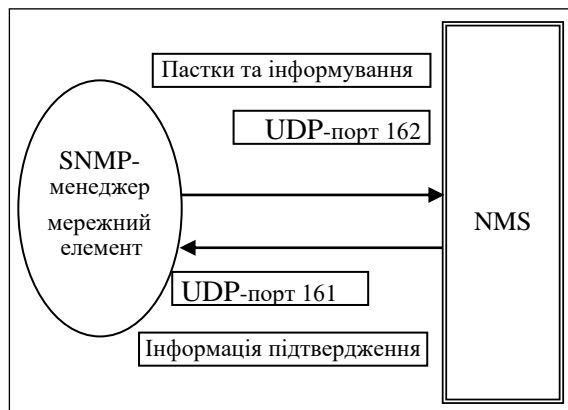


Рис. 3. SNMP-пастки

Повідомлення про захоплення SNMP надсилається лише один раз, а також відкидається, коли воно надсилається. Вони не зберігаються в пам'яті для отримання відповіді від менеджера. Повідомлення «Inform» надсилається знову й знову, поки не отримає відповідь від NMS або запит не закінчиться.

Якщо хост-пристрій не отримає відповіді від NMS, тоді він буде надсилати запит «Inform» кілька разів, поки не отримає необхідного результату. Отже, «Inform» використовує більше ресурсів пам'яті в мережі та мережних пристроях [8].

### Протокол SNMP v3

Уніфікований протокол мережного моніторингу SNMP версії 3 (SNMP v3) завдяки сучасній модульній архітектурі, удосконаленим протоколам управління, зокрема протокол прикладного рівня, схему баз даних і набір об'єктів даних, можливістю шифрувати

трафік, підтримці SNMP версії 1 та версії 2 й покращеному віддаленому налаштуванню широко застосовується для моніторингу та аналізу мережного устаткування, для знаходження й вирішення багатьох мережних проблем.

У SNMP v3 не застосовуються терміни «агент» і «менеджер», тепер використовується термін «сутності». Як і раніше, одна сутність знаходиться на керованому пристрої, а друга займається опитуванням застосунків.

У сутностей-агентів і сутностей-менеджерів є ядро, що виконує чотири основні функції (рис. 5):

- функції диспетчера;
- оброблення повідомлень;
- функції безпеки;
- контроль доступу.

Диспетчер – це система управління вхідним та вихідним трафіком. Для кожного вихідного блоку даних (PDU) він визначає тип необхідного оброблення (SNMP v1, SNMP v2, SNMP v3) та передає блок даних відповідного модуля в системі оброблення повідомлень.

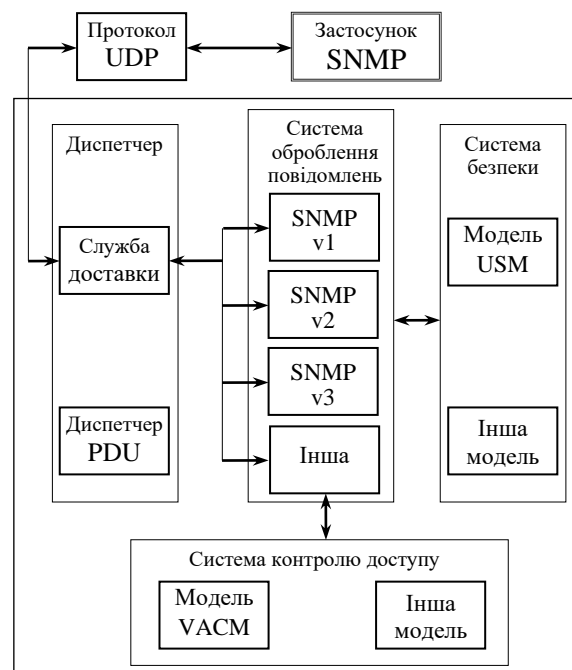


Рис. 4. Схема роботи ядра SNMP v3

Після того як система оброблення повідомлень поверне повідомлення, яке містить визначений блок даних, Диспетчер відправить його на рівень транспорту для подальшої передачі. Для вхідних повідомлень Диспетчер проводить зворотну операцію.

Система оброблення повідомлень отримує дані від Диспетчера PDU, додає до них відповідний заголовок та повертає їх назад Диспетчеру. Система безпеки відповідає за шифрування та автентифікацію.

Усі вихідні повідомлення перед відправленням спочатку передаються із системи оброблення повідомлень у систему безпеки, де шифруються поля в заголовку повідомлення, PDU, генерується код автентифікації та додається до заголовка повідомлення.

Система контролю доступу керує службами автентифікації контролю доступу до МІВ із вмісту PDU. Теоретично система контролю доступу може працювати з різними моделями контролю доступу, але в RFC 2275 описана тільки одна модель View-Based Access Control Model (VACM).

За допомогою цих команд та стандартної бази МІВ можна отримати найрізноманітнішу інформацію.

Наприклад: кількість прийнятих та надісланих пакетів по TCP, IP, UDP або ICMP. А ще можна дізнатися про кількість помилок, що були виявлені під час надсилання або отримання пакетів.

У процесі розроблення SNMP v3 достатньо уваги було приділено безпеці протоколу. Тепер стала підтримуватися модель, орієнтована на користувача User-Based Security Model, (USM) завдяки якій можна додавати модулі автентифікації та шифрування без зміни базової архітектури [7].

### **Практичне застосування протоколу SNMP. Налаштування SNMP у Windows. Налаштування даних агента SNMP**

Пуск → Панель управління → Адміністрування → Управління комп'ютером.

1. У дереві консолі необхідно розгорнути вузол «Служби та застосунки» й обрати пункт «Служби».

2. У ділянці праворуч двічі клацнути елемент «Служба SNMP».

3. Потім відкрити вкладку «Агент».

4. Ввести ім'я користувача або адміністратора комп'ютера в полі «Контакт», а потім ввести фізичне розташування комп'ютера або контакту в полі «Розташування». Ці коментарі обробляються як текст і є не обов'язковими.

5. У розділі «Служба» потрібно встановити прапорці поруч зі службами, що надаються комп'ютером, і натиснути кнопку «Так».

### **Налаштування співтовариства та пасток SNMP**

Пуск → Панель управління → Адміністрування → Управління комп'ютером.

1. У дереві консолі потрібно розгорнути вузол «Служби та застосунки» й обрати пункт «Служби».

2. У ділянці праворуч двічі клацнути елемент «Служба SNMP».

3. Відкрити вкладку «Трепінг».

4. У полі «Ім'я співтовариства» ввести ім'я співтовариства й натиснути кнопку «Додати в список».

5. У розділі «Адресати пасток» натиснути кнопку «Додати».

6. У полі «Host Name» ввести ім'я, IP-адресу вузла й натиснути кнопку «Додати». Ім'я вузла або його адреса з'явиться в списку призначення пасток.

7. Натиснути кнопку «Так» [7, 9].

### **Налаштування безпеки SNMP**

Пуск → Панель управління → Адміністрування → Управління комп'ютером.

1. У дереві консолі потрібно розгорнути вузол «Служби та застосунки» й обрати пункт «Служби».

2. У ділянці праворуч двічі клацнути елемент «Служба SNMP».

3. Відкрити вкладку «Безпека».

4. Установити прапорець «Пересилання пасток перевірки достовірності», якщо необхідно, щоб агент відправляв пастку в разі порушення перевірки достовірності.

5. У розділі «Прийнятні імена співтовариств» потрібно натиснути кнопку «Додати».

6. У полі «Права співтовариства» обрати дозволи, щоб вказати, як вузол оброблятиме запити SNMP від обраного співтовариства.

7. У полі «Ім'я співтовариства» ввести потрібне ім'я співтовариства з урахуванням реєстру, а потім натиснути кнопку «Додати».

8. Потім для того, щоб приймати запити SNMP від будь-якого вузла мережі, незалежно від його статусу, потрібно вибрати варіант «Приймати пакети SNMP з будь-якого вузла».

9. Щоб обмежити прийняття пакетів SNMP, потрібно натиснути «Приймати пакети SNMP із цих комп'ютерів», потім натиснути «Додати» й ввести в поле ім'я вузла, IP-адресу або IPX-адресу відповідного вузла. Натиснути «Додати», а потім кнопку «Так».

## Налаштування SNMP у Linux Налаштування SNMP у CentOS 7

Спочатку потрібно встановити останні оновлення за допомогою yum/dnf

```
yum update
```

Потім встановити SNMP

```
yum install net-snmp net-snmp-utils
```

та створити копію конфігураційного файлу

```
mv /etc/snmp/snmpd.conf  
/etc/snmp/snmpd.conf.orig
```

Тепер потрібно відредагувати налаштування агента

```
nano /etc/snmp/snmpd.conf
```

та додати рядки

```
community public syslocation  
MyLocation syscontact  
admin@example.com
```

Доцільніше вказувати дійсні назви про локацію та в email.

Потім необхідно додати сервіс в автозавантаження та перезапустити його

```
systemctl enable snmpd.service  
systemctl start snmpd
```

Як перевірити, що сервіс запущений:

```
systemctl status snmpd
```

Опитування агента за допомогою утиліти snmpwalk:

```
snmpwalk -v 2c -c public -O e 127.0.0.1
```

Опитування сервера локальною командою:

```
snmpwalk -v2c -c public localhost system
```

## Налаштування SNMP у Linux Debian

Насамперед потрібно встановити демона, клієнта та файли [9]

```
apt install snmpd snmp libsnmp-dev
```

Після встановлення переходимо до налаштування SNMP у Debian.

Файлом налаштування SNMP-агента за замовчуванням є /etc/snmp/snmpd.conf. Агент SNMP може бути запущений з налаштуваннями за замовчуванням. Проте для включення віддаленого моніторингу потрібно зробити декілька змін. Для цього необхідно створити резервну копію файлу

```
cp /etc/snmp/snmpd.conf  
/etc/snmp/snmpd.conf.orig
```

Тепер потрібно змінити директиву agentAddress. Її поточні налаштування дозволяють доступ тільки з локального комп'ютера. Для включення віддаленого моніторингу необхідно визначити IP-адресу інтерфейсу

```
vim /etc/snmp/snmpd.conf  
#####  
#####  
#  
# AGENT BEHAVIOUR  
#  
  
# Listen for connections from the  
local system only agentAddress udp:  
127.0.0.1:161,udp:192.168.43.62:161
```

Для налаштування автентифікації

```
directive community [source [OID]]
```

Rocommunity надає доступ тільки до читання, а rwcommunity – до читання / запису. У Access Control section потрібно помістити рядок

```
rocommunity S3CUrE 192.168.43.100.
```

Крім того, можна включити запит із локального хосту rocommunity S3CUrE localhost

```
rouser authOnlyUser  
rwuser authPrivUser priv  
rocommunity S3CUrE localhost  
rocommunity S3CUrE 192.168.43.100
```

Потім потрібно перезапустити SNMP

```
systemctl restart snmpd
```

Щоб додати сервіс в автозавантаження, необхідно ввести

```
systemctl enable snmpd
```

## Висновки

Наведений аналіз сучасних способів моніторингу та контролю мережного обладнання, показані переваги та недоліки основних технологій моніторингу та діагностування обладнання IP-мереж.

Розглянуті особливості функціонування, системи моніторингу мережного обладнання, побудованого на основі архітектури UDP/TCP із застосування мережного протоколу контролю та управління SNMP.

Показана структура та компоненти SNMP, наведена архітектура та схема роботи ядра SNMP v3 й детальний опис його роботи.

Наведені алгоритми практичного налаштування SNMP у Windows та Linux.

## Література

1. Методи наукових досліджень в телекомунікаціях: навч. посібник: у 2 т. / за ред. проф. В.В. Поповського. Харків: Компанія СМІТ, 2013. Т. 1. 390 с.
2. SNMP. URL: <https://znaimo.com.ua/>
3. Кордяк В., Дронюк І., Федевич О. Інформаційна технологія моніторингу та аналізу трафіку у комп'ютерних мережах. Національний університет «Львівська політехніка» 2015. URL: <http://ena.lp.edu.ua:8080/bitstream/ntb/31297/1/07-35-42.pdf>.



4. Взаємозв'язок між протоколами SNMP версії 1, версії 2 та версії 3 стандартної комп'ютерної мережі. URL: <https://datatracker.ietf.org/doc/rfc3584/>
5. Структура та ідентифікація керуючої інформації в мережах на основі стеку протоколів TCP/IP. URL: <https://www.rfc-editor.org/info/rfc1155>.
6. RFC 3411. MIB та архітектура SNMP. URL: <https://www.arc-it.net/html/archuse/archuse.html>.
7. Протокол управління SNMP. URL: <https://selectel.ru/blog/snmp/>
8. Обробка і відправлення повідомлень для SNMP. URL: <https://www.arc-it.net/html/resources/resources.html>.
9. Протоколи управління SNMP. URL: <https://selectel/blog/snmp/>

### References

1. Research methods in telecommunications. In 2 volumes. Volume 1: textbook manual / ed. prof. V.V. Popovsky. Harkiv: SMITH Company, 2013. 390 p.
2. SNMP. URL: <https://znaimo.com.ua/>
3. Kordyak V., Dronyuk I., Fedevich O. Information technology for monitoring and analysis of traffic in computer networks. Lviv Polytechnic National University, 2015. URL: <http://ena.lp.edu.ua:8080/bitstream/ntb/31297/1/07-35-42.pdf>.
4. Interconnection between SNMP versions 1, version 2 and version 3 of the standard computer network. URL: <https://datatracker.ietf.org/doc/rfc3584/>
5. Structure and identification of control information in networks based on the TCP / IP protocol stack. URL: <https://www.rfc-editor.org/info/rfc1155>.
6. RFC 3411. MIB and SNMP architecture. URL: <https://www.arc-it.net/html/archuse/archuse.html>.
7. SNMP management protocol. URL: <https://selectel.ru/blog/snmp/>
8. Processing and sending messages for SNMP. URL: <https://www.arc-it.net/html/resources/resources.html>.
9. SNMP Management Protocols. URL: <https://selectel/blog/snmp/>

**Голубничий Дмитро Юрійович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, [dmytro.holubnychyi@hneu.net](mailto:dmytro.holubnychyi@hneu.net), тел. +38 093-900-38-96,

**Коцюба Василь Петрович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, [vasyl.kotsyuba@gmail.com](mailto:vasyl.kotsyuba@gmail.com), тел. +38 067-573-27-89.

### Applying technologies of monitoring and condition analysis of IP-networks based on the use of the SNMP protocol

**Abstract. Problem.** Monitoring, security, condition analysis and control of telecommunications networks remains the most important part of system and network administration. Monitoring systems controls the network in an automated mode, and complex decisions based on the prepared information of the network monitoring system are made by the network administrator. **Goal.** The purpose of the work is to analyze the principles of applying technologies, programs and protocols that allow you to manage equipment and maintain reliable operation of the computer network. To consider features of functioning, the systems of monitoring of the network equipment are built on the basis of UDP/TCP architecture on application of the network protocol of control and management of SNMP. **Methodology.** Analytical methods of studying technologies of principles and approaches to network monitoring and management are used. The structure and components of SNMP are shown, the architecture and scheme of operation of SNMP v3 kernel and the detailed description of its work are resulted. **Results.** The analysis of modern methods of monitoring and control of network equipment is given, the advantages and disadvantages of the main technologies of monitoring and diagnosing the equipment of IP networks are shown. It is substantiated that the most effective and reliable tool that allows you to perform tasks on the management of IP network devices is the SNMP protocol. **Originality.** The monitoring system allows you to provide a set of solutions that maintain automatic monitoring of networks implemented on the basis of different technologies (data and speech, video), providing different services and built on equipment from different manufacturers. **Practical value.** The structure and components of SNMP are shown, the architecture and scheme of operation of SNMP v3 kernel and the detailed description of its work are resulted. Algorithms for practical SNMP configuration in Windows are presented, SNMP agent, trap and security data settings, configuring SNMP on Linux, SNMP settings in CentOS 7 and Debian 10. **Key words:** monitoring systems, equipment management, network monitoring protocol, SNMP settings, MIB.

**Dmitro Holubnychyi**, Ph.D., assoc. prof. kaf. information systems, Semyon Kuznets Kharkiv National University of Economics, tel. +38 38 093-900-38-96. [dmytro.holubnychyi@hneu.net](mailto:dmytro.holubnychyi@hneu.net),

**Vasyl Kotsyuba**, Ph.D., assoc. prof. kaf. information systems, Semyon Kuznets Kharkiv National University of Economics, [vasyl.kotsyuba@gmail.com](mailto:vasyl.kotsyuba@gmail.com), tel. +38 067-573-27-89.

УДК 004.514

DOI: 10.30977/BUL.2219-5548.2022.96.0.22

## РОЗРОБЛЕННЯ ГРАФІЧНОГО КОМПОНЕНТА ЯК СКЛАДНИКА ІНФОРМАЦІЙНО-НАВІГАЦІЙНОЇ СИСТЕМИ СУЧАСНОГО УНІВЕРСИТЕТУ

Гризун Л. Е., Біда Б. О.

Харківський національний економічний університет ім. С. Кузнеця

**Анотація.** Запропоновано розроблення графічного компонента як складника інформаційно-навігаційної системи університету. Висвітлено специфікацію вимог до графічного компонента зазначеної системи; уточнено постановку завдання проєктування; розкрито основні етапи проєктування та розроблення графічного компонента інформаційно-навігаційної системи.

**Ключові слова:** інформаційно-навігаційна система, графічний компонент, технологія Angular, реактивне програмування, бібліотека RxJs, бібліотека SVG.js.

### Вступ

Останнім часом все більше аспектів життя спрощуються завдяки використанню вебтехнологій, усе частіше необхідну інформацію можливо отримати максимально зручним способом – онлайн та (або) відразу в мобільному пристрої, за допомогою програмного застосунку. Одним із таких аспектів стає завдання орієнтування в приміщеннях (зокрема в університетах) з великою кількістю поверхів, кімнат і локацій різного призначення. Часто потреба в зручній навігації в приміщеннях сучасного університету виникає в студентів-першокурсників, нових викладачів та відвідувачів університету, незнайомих із розташуванням навчальних корпусів та їхньою топологією, що актуалізує розроблення інформаційно-навігаційної системи. У цьому контексті залучення вебтехнологій для розв'язання завдань навігації має переваги, адже вебзастосунки не потребують заздалегідь встановленого програмного забезпечення, окрім браузерів, що зазвичай є доступними в кожному сучасному пристрої, не потребують установлення на пристрій та мають змогу використовувати кеш браузера та технологію PWA для роботи офлайн.

Як зазначалося вище, основною потребою навігації в університеті є орієнтування та супроводження користувача в приміщеннях та численних локаціях навчального закладу. З цією метою доцільно користуватися відповідною інформаційно-навігаційною системою (далі ІНС). Відповідно до джерел [4], під ІНС розуміється обчислювальна система, що допомагає в навігації. ІНС вирішують питання визначення місцезнаходження користувача та інколи прокладання маршруту з точки А в точку Б, також доповнюючи вказаний вище функціонал певною корисною ін-

формацію, важливою користувачеві для полегшення орієнтування.

Загалом архітектура ІНС [4] є сукупністю компонентів для забезпечення користувача всіма зазначеними можливостями, використовуючи в цьому випадку різні методи: від застосування магнітних компасів, радіосистем, даних космічних супутників, до звичайних карт місцевості та позначок із направленням сторін світу. Особливе місце в ІНС посідає графічний компонент, що є посередником між користувачем та всією бізнеслогікою застосунку – від оброблення даних до виконання складних алгоритмів. Його роль полягає в наданні користувачеві зручного та зрозумілого інтерфейсу для взаємодії з функціоналом застосунку, а також візуалізації мапи та навігаційної інформації (назв конкретних локацій, додаткових позначок тощо). Тому графічний компонент разом з іншими підсистемами має спростити завдання навігації в університеті та вивести зручність орієнтування користувача в будівлі на новий рівень.

### Аналіз публікацій

Зосередимось на аналізі ІНС, що забезпечують саме навігацію в приміщенні. Такі системи не можуть повністю використовувати можливості GPS навігації (як це роблять ІНС місцевості, наприклад Google Maps, 2Gis, OpenStreetMaps тощо) через те, що навігація відбувається в невеликих масштабах і зазвичай у критичних приміщеннях. Отже, залучення технологій GPS-навігації призведе до значних похибок. У зв'язку з цим більшість ІНС для орієнтування у приміщеннях використовують indoor-навігацію [8; 12].

Під indoor-навігацією розуміють виявлення об'єктів (пристроїв або людей) та орієнтування всередині будівлі за допомогою радіохвиль, магнітного поля, акустичних сигналів або інших технологій, розташованих усередині будівлі [8]. Проте проблема такої навігації в тому, що кожна будівля, де треба використовувати таку навігацію, повинна мати спеціальні пристрої, налаштовані на роботу з ІНС, що робить розроблення такої системи дорогим і довготривалим, а також ускладнює її тестування, супроводження та розширення її можливостей.

Для аналізу можливостей було обрано декілька популярних застосунків, що спеціалізуються на навігації в приміщенні, використовуючи indoor-навігацію, і реалізують такі загальні функції: перегляд схеми будівлі / поверху, можливість пересування по поверху або перегляду всіх частин приміщення, надають візуальну інформацію для спрощення сприйняття схеми будівлі та інтуїтивної навігації. Під час аналізу ми зосередились на можливостях саме графічних компонентів аналогів ІНС та систем загалом за такими критеріями: (1) функціонал прокладення шляху; (2) адаптивність інтерфейсу; (3) зручність інтерфейсу користувача; (4) якість візуалізації графічного контенту; (5) естетичність дизайну; (6) ціна; (7) загальне враження. Зокрема за такими критеріями було оцінено графічні компоненти низки застосунків: Mapsindoors (розширення платформи MapsPeople, побудованої за технологією Google Maps, що забезпечує перехід від зовнішньої до внутрішньої навігації та її швидке впровадження); Situm Mapping Tool (сервіс indoor-навігації, що забезпечує супровід відвідувачів конкретної будівлі для знаходження шляху в реальному часі); AnyPlace (безкоштовний відкритий сервіс indoor-навігації, що забезпечує визначення місцезнаходження та пошук усередині будівель за допомогою смартфонів); BSB Navigator (застосунок для смартфонів, що є навігатором по бібліотеці на Людвігштрассе в Мюнхені та використовує технологію маяків із застосуванням Bluetooth смартфона) [6; 7; 10; 11].

Незважаючи на значні функціональні можливості цих ІНС та їхніх графічних компонентів, варто наголосити на їхніх основних недоліках і обмеженнях. Більшість ІНС прив'язані до конкретної будівлі, та їх не можна використовувати в інших приміщеннях; не мають функціоналу розширення та доопрацювання; не забезпечують користува-

ча мовною локалізацією; є або вебсервісами, або мобільними застосунками.

Зазначені обмеження доцільно взяти до уваги під час визначення функціональних і нефункціональних вимог до графічного компонента ІНС сучасного університету, а також для формулювання логічної постановки завдання його проєктування. Викладене вище обумовлює актуальність пошуку нових технологічних підходів до розроблення графічного компонента ІНС університету.

### Мета та постановка завдання

Метою цієї роботи є проєктування графічного компонента інформаційно-навігаційної системи ХНЕУ ім. С. Кузнеця та його розроблення на основі застосування вебтехнологій.

Для досягнення поставленої мети необхідно розробити та висвітлити специфікацію вимог до графічного компонента зазначеної системи; уточнити постановку завдання проєктування; здійснити та висвітлити основні етапи проєктування та розроблення графічного компонента інформаційно-навігаційної системи.

### Виклад основного матеріалу

У процесі специфікації вимог до графічного компонента ІНС університету було побудовано діаграму використання компонента користувачем, який є будь-якою людиною, що має на смартфоні встановлену ІНС як вебзастосунок або вебпосилання на нього.

Діаграма використання, яку опускаємо для статті, містить усі передбачені варіанти застосування графічного компонента користувачем, а саме: перегляд інструкцій з використання; перегляд налаштувань (зокрема зміну мовної локалізації); вибір стартової точки (точки, яка відповідає координатам розміщення фізичного QR-коду щодо схеми будівлі); вибір кінцевої точки (будь-якої локації в приміщенні університету, нанесеної на інтерактивну мапу, для якої користувач хоче знайти маршрут); отримання візуалізованого прокладеного маршруту між стартовою та кінцевою точками; використання сканера QR-кодів; отримання прокладеного маршруту; вільний перегляд схеми поточного поверху; зміна масштабу мапи; зміна поточного поверху.

Для всіх варіантів використання проведено їхню специфікацію з визначенням контексту, дійової особи, передумови, тригера, сценарію та постумови кожного варіанта вико-

ристання. Приклад специфікації варіанта використання «Одержання візуалізованого прокладеного маршруту між стартовою та кінцевою точками» наведено в табл. 1.

Таблиця 1 – Специфікація варіанта використання «Одержання візуалізованого прокладеного маршруту»

Характеристика	Значення
Контекст використання	Дозволяється будь-якому користувачеві побачити прокладений шлях від стартової до кінцевої точки
Дійові особи	Будь-який користувач сайту ІНС
Передумова	Користувач обрав стартову та кінцеву точку
Тригер	Необхідність прокласти маршрут
Сценарій	<ol style="list-style-type: none"> <li>1. Введення назви стартової точки, де є QR-код, або обрання її із списку, або сканування QR-коду за допомогою сканера</li> <li>2. Натискання кнопки з іконкою збільшуючої лупи у відповідному полі введення</li> <li>3. Введення назви локації кінцевої точки або обрання її у списку</li> <li>4. Натискання кнопки з іконкою збільшуючої лупи у відповідному полі введення</li> </ol>
Постумова	Після успішного виконання умови користувач може побачити анімацію, яка візуалізує маршрут до кінцевої точки (шуканої локації)

На основі аналізу специфікацій варіантів використання сформульовано низку функціональних вимог до графічного компонента ІНС університету. До них належать такі вимоги, як: зміна локалізації; вибір наявних на мапі точок; візуалізація прокладеного маршруту; огляд схеми поверху; вільна зміна поверхів; зміна масштабу мапи; надання підказок щодо типів локацій; використання спливаючих вікон для розміщення певного функціоналу; адаптивність інтерфейсу під різні розміри пристроїв та їхні графічні налаштування.

Уточнюючи та конкретизуючи постановку завдання проєктування, візьмемо до уваги таке. Графічний компонент ІНС університету, що є вебзастосунком, необхідний для спрощення процесу використання алгоритмів прокладання шляху та орієнтування на схемах поверхів університету для всіх користу-

вачів системи. Основне його завдання – це забезпечити максимально ефективне застосування всіх функцій ІНС та додати зручний і привабливий інтерфейс для використання.

Аналізуючи місце графічного компонента в архітектурі ІНС, варто зазначити, що він забезпечує не тільки зовнішній вигляд застосунку з його інтерфейсом, а й логіку взаємодії даних, які належать до візуальної частини системи (коефіцієнт масштабу мапи, позиція щодо екрана, обрані користувачем налаштування тощо). Загалом графічний компонент є сукупністю елементів інтерфейсу та програмного коду, що дозволяє користувачеві взаємодіяти із системою, впливати на її стан, а також отримувати певну інформацію. Крім цього, графічний компонент реалізує візуалізацію карти приміщення та маршруту користувача для його навігації. Важливою частиною графічного компонента є інтерактивна мапа поточного поверху. Графічний компонент забезпечує можливість змінення положення екрана щодо мапи, а також її масштабу. Для цього використовується тачпад пристрою.

Зазначимо також, що до графічного компонента належить така сукупність елементів інтерфейсу: меню пошуку; кнопки для управління масштабом мапи; перемикач поверхів; вікно з довідковою інформацією про використання застосунку; доступ до сканера QR-кодів, що, використовуючи вебкамеру пристрою, дозволяє сканувати коди щодо інформації про стартову та/або кінцеву точку; вікно налаштувань для зміни мови інтерфейсу користувача тощо. Для того щоб отримати інформацію про те, як дістатися певної локації університету, необхідно визначити кінцеву точку. Зробити це можна декількома способами: ввести назву кінцевої точки (наприклад номер аудиторії) у меню пошуку та натиснути іконку пошуку або відсканувати QR-код, який зберігає інформацію про кінцеву точку. Якщо потрібно не тільки побачити локацію на мапі, а й отримати шлях до неї, треба визначити точку старту також одним із зазначених вище способів. Зазвичай точка старту – це місце, де фізично розташований QR-код у будівлі, що зберігає в собі ідентифікатор із точкою. Точка старту відповідає місцезнаходженню користувача, проте не гарантує цього, тому що прив'язана до конкретного фізичного місця на мапі й не змінює свого положення. Це дає змогу користувачеві виконувати навігацію не тільки в реальному часі й для себе, а і в будь-який час

для будь-якої людини, що перебуває в будівлі.

Отже, ключовою проблемою проектування графічного компонента ІНС стає управління даними, що відповідають за інтерфейс та стан мапи (коефіцієнт масштабу, позиція мапи щодо екрана, поточний поверх тощо). Часто зміна якогось параметра спричиняє зміну більшості інших та потребує виконання певних дій, що стає значною проблемою під час розширення такої системи та її функціоналу.

Аналіз сутності графічного компонента ІНС, що розробляється на основі вебтехнологій, особливостей його функціоналу та необхідності подолання зазначених вище обмежень, притаманних графічним компонентам наявних ІНС (відсутність функціоналу розширення, переорієнтації на навігацію в іншій будівлі, а також мовної локалізації; жорстка залежність від платформи функціонування), вимагає залучення сукупності технологій. Зокрема вважаємо за доцільне обрати технологію Angular як платформу для розроблення ефективних та складних односторінкових застосунків, що використовує діалект TypeScript мови JavaScript для опису компонентів інтерфейсу із застосуванням принципів об'єктно орієнтованого програмування [5]; логіку реактивного програмування за допомогою бібліотеки RxJs [9] та інші прийоми для спрощення розроблення. Крім цього, доцільним є залучення бібліотеки SVG.js [1], яка застосовує можливості HTML та JavaScript для динамічного оброблення, генерації та анімації векторних документів, що допоможе у створенні інтерактивної карти.

Отже, постановку завдання проектування графічного компонента ІНС університету варто визначити як забезпечення таких складників:

1) дизайн і реалізація компонентів елементів управління у вигляді шаблону, стилів та контролера, html-, css- та ts-файлів для кожного з компонентів;

2) реалізація адаптивного та гнучкого інтерфейсу користувача як групування всіх елементів управління з додатковою логікою відображення;

3) організація бізнес-логіки інтерфейсу у вигляді сервісів, що зберігають у собі дані та засоби їхнього оброблення, використовуючи реактивне програмування за допомогою RxJs, також пов'язуючи між собою певні компоненти;

4) організація утилітарних класів та методів, що дозволять винести повторювані алгоритми, які безпосередньо не належать до бізнес-логіки роботи всього графічного компонента;

5) опис всіх моделей, що забезпечують чітку типізацію та зручний і безпечний контроль за сутностями вихідного коду.

Відповідно до поставленого завдання та визначених вище функціональних вимог до графічного компонента, було реалізовано його проектування як складника архітектури ІНС університету.

Розроблено мокапи та прототипи інтерфейсу вебзастосунку, а також дизайн схем будівлі для кожного окремого поверху. Мокапи було спроєктовано для максимальної ергономічності користування. Для цього екран було поділено на три інтерактивні зони, які залежно від розмірів екрана пристрою користувача масштабуються, змінюють відступи внутрішніх елементів та адаптуються під зручне користування та досяжність. Зокрема було виокремлено такі зони екрана.

Верхня зона – зона пошуку, де містяться всі елементи інтерфейсу, що відповідають за встановлення стартової та кінцевої точки. Однією з найважливіших функцій ІНС є визначення місцезнаходження та пункту призначення для прокладення маршруту або орієнтування в університеті, оскільки застосунк не використовує відслідковування місцезнаходження користувача в реальному часі, яке прив'язано не до його реального місцезнаходження, а до стартових точок, що є фізичними QR-кодами. Отже, виникає потреба в самостійному та вільному виборі стартової та кінцевої точок для пошуку та візуалізації маршруту (або просто кінцевої точки), для чого інтерфейсом ІНС передбачено верхню зону екрана, що складається з двох форм пошуку (рис. 1).



Рис. 1. Зона пошуку із елементами інтерфейсу для встановлення стартової та кінцевої точок

Для того, щоб користувач міг повноцінно застосовувати ІНС, графічний компонент містить елементи управління для зміни позиції мапи, її масштабу тощо. Такі елементи було розміщено в лівій зоні, щоб не заважати користувачеві розглядати мапу, а також запобігаючи випадкового натискання будь-якої

кнопки фалангою пальця під час тримання мобільного девайса, що може зіпсувати користувацький досвід. Отже, ліва зона екрана містить елементи інтерфейсу, що дозволяють впливати на масштаб мапи, її центрування та зміну поточного поверху (рис. 2).

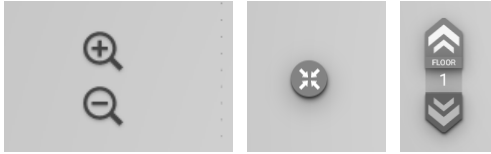


Рис. 2. Елементи інтерфейсу лівої зони екрана для керування інтерактивною мапою

Права зона екрана відповідає за загальний функціонал застосунку та концентрує елементи інтерфейсу, що надають доступ до унікальних функцій застосунку (надання довідки про роботу застосунку, налаштування та сканер QR-кодів) (рис. 3).

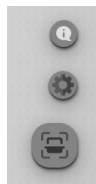


Рис. 3. Елементи інтерфейсу правої зони екрана

Визначено кольорову гаму інтерфейсу. Виходячи із загальних фізичних умов використання ІНС (у світлу пору доби та в добре освітлених приміщеннях університету), зображення має бути досить яскравим та контрастним. Як основний колір інтерфейсу користувача було обрано світло-блакитний (#358DFF), білий та відтінки сірого із застосуванням плавних та легких градієнтів. Також використовуються заокруглені кути й більш м'які форми елементів та різноманітні іконки для полегшення сприйняття користувачем.

Однією з переваг графічного компонента, що представляється, є адаптивність та гнучкість, яку він забезпечує всій ІНС. Унаслідок застосунок розраховано на використання не тільки мобільними пристроями, а також і на портативних комп'ютерах, ноутбуках, планшетах тощо. Водночас інтерфейс користувача коректно відображається на екрані пристрою, зберігаючи звичне розташування елементів управління, ергономічність та не

втрачає своєї унікальності. Адаптивний дизайн було отримано завдяки врахуванню не тільки ширини девайса, а й висоти та, згідно з [2], щільності віртуальних пікселів щодо фізичних. Також було враховано можливі налаштування масштабу відображення браузера та девайса користувача. На мобільних пристроях із невеликими розмірами екрана, низькою щільністю пікселів або в альбомному режимі змінюються відступи від країв екрана та між елементами управління, а також напрямок розташування кнопок управління масштабом мапи, щоб не займати багато місця та дати достатньо простору для верхньої зони, кнопок центрування мапи та перемикача поверхів (рис. 4).

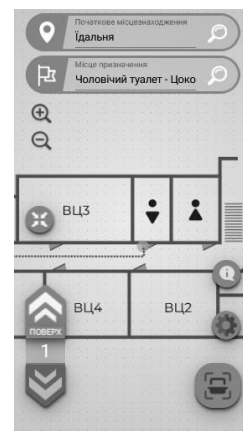


Рис. 4. Вигляд інтерфейсу користувача на смартфоні iPhone 5/SE у портретному режимі

Інтерфейс добре адаптовано під старі моделі мобільних пристроїв, продукти компанії Apple та має підтримку щодо старих версій браузерів.

Розроблено також дизайн інтерактивної мапи шляхом оцифрування схем поверхів будівель університету за допомогою інструмента редагування векторних зображень Adobe Illustrator. Результат оцифрування схеми одного з поверхів університету та його інтерактивна мапа, згенерована графічним компонентом ІНС, наведені на рис. 5–6.

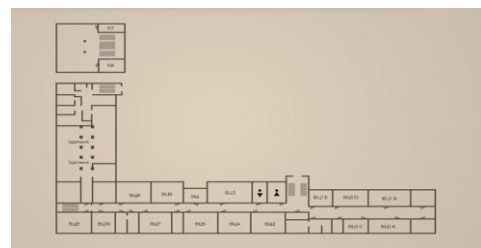


Рис. 5. Результат оцифрування схеми одного з поверхів університету

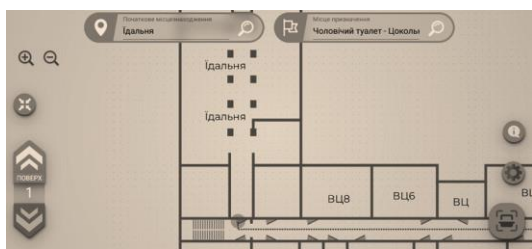


Рис. 6. Інтерактивна мапа, візуалізована графічним компонентом ІНС

Завдяки застосуванню саме векторної графіки для кожного поверху було одержано повноцінний документ зі своїм синтаксисом та правилами, який описує за допомогою атрибутів та формул, як має виглядати той чи інший графічний елемент зображення. Це дало змогу динамічно взаємодіяти з таким документом за допомогою програмного коду, що дозволить надалі легко адаптувати вхідні дані графічного компонента для розширення можливостей ІНС. Крім цього, застосування документів із схемами, підготовлених засобами векторної графіки, дає перевагу у швидкості завантаження та якості зображення перед растровими аналогами.

Кінцевим результатом розроблення графічного компонента як складника ІНС є вебзастосунок, тож він не потребує інсталювання на пристроях користувачів, проте має таку можливість завдяки технології PWA. Для користування розробленою ІНС не тільки локально, а й з будь-якого пристрою, підключеного до інтернету, всі файли зібраного проекту треба розмістити на видаленому сервері. Оскільки кінцевий артефакт є нативні html-, css-, js-файли, а також ресурси (зображення, json-файли та інші медіафайли), то додаткового оточення та програмного забезпечення сервер не потребує, тому розгортання продукту стає досить простим порівняно з іншими застосунками, що використовують серверну частину для повноцінної роботи. Для тестового деплою проекту було обрано Firebase Hosting. Розміщений на сервері програмний продукт може бути запущений на будь-якій системі, портативних комп'ютерах, мобільних пристроях.

#### Результати апробації розробленого графічного компонента

Графічний компонент як складник цілісної архітектури інформаційно-навігаційної системи в її тестовій версії було впроваджено в практику освітньої діяльності ХНЕУ ім. С. Кузнеця. Зокрема у відповідних лока-

ціях університету було розміщено QR-коди, що дозволяють будь-якому відвідувачу університету за допомогою смартфона отримати доступ до ІНС загалом та до її графічного компонента і скористатися описаним вище функціоналом. У цих самих місцях розташовано також спеціальні коди для зворотного зв'язку з користувачами й одержання відгуків від них (рис. 7). У процесі апробації було зібрано та проаналізовано відгуки користувачів системи саме щодо її інтерфейсних і графічних можливостей, а також щодо доцільності запровадження системи в освітню практику.



Рис. 7. QR-коди, розміщені в певних локаціях ХНЕУ ім. С. Кузнеця для отримання доступу до ІНС, її функціоналу та зворотного зв'язку

Користувачам було запропоновано заповнити форму, за допомогою якої їм необхідно було оцінити якість реалізації як функціональних, так і нефункціональних вимог до графічного компонента за п'ятибальною шкалою. Серед користувачів, які надіслали відгуки, було виявлено 34 % вітчизняних студентів різних курсів, 23 % іноземних студентів, 15 % школярів з учителями, що відвідували університетські заходи, 10 % працівників університету, 18 % відвідувачів університету.

Загалом було отримано схвальні відгуки щодо якості інтерфейсу, швидкості завантаження інтерактивної мапи та зручності роботи з нею, зрозумілості візуалізації стартового місцезнаходження користувача та кінцевої

точки, а також коректності запропонованого системою маршруту. Позитивне враження справила апробована користувачами можливість ділитися посиланням на застосунок з уже побудованим маршрутом від найближчої точки з QR-кодом до точки потрібної локації. На думку користувачів, це виявилось корисним під час організації масових заходів, адже забезпечило швидке й надійне орієнтування одночасно значної кількості відвідувачів університету. Англійськомовні студенти та відвідувачі перевагами навігаційної системи вважають також реалізовану англійськомовну локалізацію інтерфейсу ІНС.

Серед побажань щодо вдосконалення роботи ІНС було запропоновано підвищити якість анімованої візуалізації прокладеного маршруту між локаціями на різних поверхах, що, на думку деяких користувачів, є недостатньо чітко промальованою.

Отримані під час апробації відгуки від користувачів засвідчують доцільність розроблення та застосування графічного компонента як складової системи для навігації в університеті. Побажання будуть ураховані для розширення та вдосконалення ІНС університету, що є перспективами нашої роботи.

### Висновки

Отже, розроблений графічний компонент ІНС університету надає користувачеві зручну та надійну можливість: взаємодіяти з усіма підсистемами ІНС; візуалізувати інтерактивну мапу приміщення з маршрутом користувача, отриманим системою на основі алгоритмів пошуку, що забезпечує орієнтування в приміщеннях університету; повноцінно застосовувати весь функціонал ІНС.

Зауважимо також, що за рахунок запропонованих технологічних рішень розроблений графічний компонент може бути впроваджений у системи навігації в інших будівлях (торгових центрах, офісах, культурних об'єктах, паркувальних майданчиках тощо).

### Література

1. Документація бібліотеки `svg.js`. URL: <https://svgdotjs.github.io/docs/3.0/>
2. Пиксели реальные и виртуальные. URL: <https://www.sea.com.ua/ua/led-ekrany/news/pikseli-realnye-i-virtualnye-anatomia-svetodiodnogo-ekrana-cast-11/>
3. Савчук В. В., Пасічник В. В. Інтелектуальна система «Мобільний інформаційний асистент туриста»: функціональні та технологічні особливості // Вісник Нац. ун-ту «Львівська

політехніка». 2015. № 832: Інформаційні системи та мережі. С. 225–241.

4. Stephen F. Appleyard (23 January 2006). *Marine Electronic Navigation*. Routledge. ISBN 978-1-134-96309-6.
5. Файн Я., Моисеев А. *Angular и TypeScript. Сайтостроение для профессионалов*. Санкт-Петербург: Питер 2018. 464 с.
6. AnyPlace indoor service. URL: <https://play.google.com/store/apps/details?id=com.dmsl.anyplace&hl=ru&gl=US>.
7. BSB Navigator. URL: <https://www.bsb-muenchen.de/en/search-and-service/apps/bsb-navigator/>
8. Integrated Indoor Navigation System for Ground Vehicles with Automatic 3D Alignment and Positionin Initialisation / Mohamed M. Atia, Shifei Liu, Heba Nematallah, Tashfeen B. Karamat, Aboelmagd Noureldin // *IEEE Transactions on Vehicular Technology*. 2015. Vol. 64, Issue 4, April. P. 1279–1292.
9. Learn RxJS. URL: <https://www.learnrxjs.io/>
10. MapsIndoors documentation. URL: <https://docs.mapsindoors.com>.
11. Indoor navigation with Situm. URL: <https://situm.com/en/indoor-navigation/>
12. Transforming indoor wayfinding. URL: <https://www.liveli.com.au/OurTechnology/>

### References

1. Documentation of the library `svg.js`. URL: <https://svgdotjs.github.io/docs/3.0/> [in Ukrainian].
2. Pixels real and vertual. URL: <https://www.sea.com.ua/ua/led-ekrany/news/pikseli-realnye-i-virtualnye-anatomia-svetodiodnogo-ekrana-cast-11/> [in Russian].
3. Savchuk V. V., Pasichnyk V. O. Intellectual system “Mobile inforamation assistant of a tourist: functional and technological features // *Visnyk Nats. Un-tu “Lvivska politechnika”*. 2015. № 832: Information systems and networks. P. 225–241 [in Ukrainian].
4. Stephen F. Appleyard (23 January 2006). *Marine Electronic Navigation*. Routledge. ISBN 978-1-134-96309-6.
5. Fine Y., Moiseev A. *Angular and TypeScript. Sitebuilding for pffessionals*. Piter, 2018. 464 p. [in Russian].
6. AnyPlace indoor service. URL: <https://play.google.com/store/apps/details?id=com.dmsl.anyplace&hl=ru&gl=US>
7. BSB Navigator. URL: <https://www.bsb-muenchen.de/en/search-and-service/apps/bsb-navigator/>
8. Integrated Indoor Navigation System for Ground Vehicles with Automatic 3D Alignment and Positionin Initialisation / Mohamed M. Atia, Shifei Liu, Heba Nematallah, Tashfeen B. Karamat, Aboelmagd Noureldin // *IEEE Transactions on Vehicular Technology*. 2015. Vol. 64, Issue 4, April . P. 1279–1292.



9. Learn RxJS. URL: <https://www.learnrxjs.io/>
10. Mapsindoors documentation. URL: <https://docs.mapsindoors.com>.
11. Indoor navigation with Situm. URL: <https://situm.com/en/indoor-navigation/>
12. Transforming indoor wayfinding. URL: <https://www.liveli.com.au/OurTechnology/>

**Гризун Людмила Едуардівна**, д.п.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, Lgr2007@ukr.net, тел. +38 0954139386,

**Біда Богдан Олексійович**, магістрант каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, bida1bohdan@hneu.net, тел. +38 095-706-15-07

### **Development of graphic component as an integral part of information and navigation system of modern university**

**Abstract. Problem.** The need for convenient navigation in the premises of a modern university arises for first-year students, new teachers and university visitors who are unfamiliar with the location of its buildings and their topology, which actualizes the development of information and navigation system (INS). One of the integral parts of the architecture of the INS is its graphic component, which performs several functions at once. The analysis of functionality of the existing INSs which apply indoor navigation testifies that despite their great functionality they have some disadvantages and limitations: most INS are building-specific and cannot be used in other areas; do not have the function of expansion and elaboration; do not provide the user with language localization; are either web services or mobile applications. These limitations encourage to find out new technological approaches to the development of the graphic com-

ponent of the university INS. **Goal.** The goal is the development of the graphic component of the university INS based on the specification of its functional requirements and technological solutions. **Methodology.** The analytical methods of research are used. **Results.** The developed graphic component of the INS of the university, built basing on web-technology, provides the user with a convenient and reliable opportunity to: interact with all subsystems of the INS; visualize an interactive map of the premises with the user's route, obtained by the system on the basis of search algorithms, which provides orientation in the university premises; make full use of all INS functionality. **Originality.** It is suggested to apply special technological solutions: angular framework, logic of reactive programming using RxJs library, and SVG.js library which applies HTML and JavaScript means for dynamic generation, proceeding and animation of digital images. **Practical value.** The use of the proposed technological solutions provides the possibility for the developed component to be implemented in navigation systems in other buildings (shopping malls, offices, cultural facilities, parking lots, etc.).

**Key words:** information and navigation system, graphic component, Angular framework, reactive programming, RxJs library, SVG.js library.

**Gryzun Liudmyla**, professor, DSci, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Lgr2007@ukr.net, tel. +38 0954139386,

**Bida Bogdan**, master-student, Information Systems Department, bida1bohdan@hneu.net, Simon Kuznets Kharkiv National University of Economics, tel. +38 095-706-15-07.

УДК 004.514

DOI: 10.30977/BUL.2219-5548.2022.96.0.30

## ВИБІР ІНСТРУМЕНТУ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Гризун Л. Е., Скорін Ю. І., Деточенко І. М.

Харківський національний економічний університет ім. С. Кузнеця

**Анотація.** У роботі проаналізовано можливості основних інструментів безперервної інтеграції для автоматизованого тестування та визначено критерії їхнього оцінювання. На основі визначених критеріїв розроблено модель оцінювання зазначених інструментів за допомогою СППР та використано її для вибору відповідного інструменту. Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту. Отримано та проаналізовано результати використання моделі оцінювання.

**Ключові слова:** інструменти безперервної інтеграції, автоматизоване тестування ПЗ, модель оцінювання, система підтримки прийняття рішень.

### Вступ

Автоматизоване тестування сьогодні є невід'ємним складником процесу розроблення програмного забезпечення. Під автоматизованим тестуванням розуміють залучення до тестування програмних продуктів спеціальних інструментів та технологій з метою підвищення якості ПЗ.

У практиці сучасних компаній існує проблема впровадження автоматизованого тестування, що пов'язано із недостатністю засобів для оцінювання та вибору відповідних інструментів для реалізації цього процесу.

Автоматизоване тестування безпосередньо пов'язано із безперервною інтеграцією (continuous integration / CI), яка полягає у виконанні частих автоматизованих об'єднань складників проекту для швидшого виявлення та вирішення інтеграційних проблем. Кожний коміт у репозиторії з кодом має перевірятися автоматично, а результати повідомлятися розробникам. CI/CD дозволяє розв'язати проблеми, які інтеграція нового коду може спричинити для команд розробників.

Відповідно до джерел [3; 4], безперервна інтеграція / безперервна доставка (CI/CD) є методом доставки програм клієнтам шляхом впровадження автоматизації на етапах розроблення застосунків. Постійне розгортання за допомогою автоматизації вирішує проблему перевантаження операційних команд ручними процесами, які уповільнюють доставку застосунків. Вибір коректного інструменту безперервної інтеграції для автоматизованого тестування є запорукою ефективної безперервної інтеграції.

### Аналіз публікацій

Аналіз спеціалізованих джерел засвідчує, що проблеми автоматизованого тестування програмного забезпечення вивчаються у безпосередньому зв'язку із завданнями безперервної інтеграції, яка передбачає, що всі, хто бере участь у проекті розробки ПЗ, регулярно публікують свої зміни в кодовій базі та перевіряють, чи працює код належним чином після кожної зміни. Безперервна інтеграція вважається дослідниками ключовим елементом DevOps-підходу до розроблення та випуску програмного забезпечення, який сприяє спільній роботі, автоматизації та швидкому отриманню зворотного зв'язку [3; 4; 6].

Основними елементами безперервної інтеграції є [6; 7]: джерело або система контролю версій з єдиною кодовою базою, зокрема файли вихідного коду, бібліотеки, конфігураційні файли та скрипти; автоматизовані білд-скрипти; автоматизовані тести; інфраструктура для складання та тестів.

Щоб усі учасники проекту працювали з однаковим кодом, вони мають використовувати один репозиторій та регулярно публікувати свої зміни. Наступним кроком після відправлення змін є проведення автоматизованих тестів для перевірки поведінки коду. Автоматизація цього процесу є невід'ємною частиною безперервної інтеграції. Якщо тестування здійснюється вручну, це забирає багато часу і може призвести до помилок. Унаслідок щоденної інтеграції змін втрачає сенс. Конкретні інструменти складання та фреймворки тестування залежать від робочої мови програмування.

Після налаштування скриптів та тестів необхідно контролювати та за необхідності

оновлювати процес. Це передбачає додавання автоматизованих тестів із появою будь-яких нових функцій, усунення збоїв та відстеження продуктивності.

Концепція безперервної інтеграції стала вирішенням проблем, з якими зіткнулися команди, коли намагалися звести до купи роботу безлічі розробників, щоб створити цілісний продукт після тривалого періоду написання коду. Процес інтеграції незалежно розроблених модулів міг призвести до істотних затримок у роботі над проектом, оскільки припущення та залежності виявлялися через тривалий час після написання коду та часто вимагали його суттєвого перероблення [10; 11].

У цьому контексті в літературі й на практиці розглядається також безперервне тестування як процес виконання автоматизованих тестів як частини конвеєра постачання програмного забезпечення з метою якомога швидшого отримання зворотного зв'язку щодо бізнес-ризиків, пов'язаних із випуском програмного забезпечення [1; 2]. Серед ключових характеристик безперервного тестування дослідники виокремлюють такі: (1) тестування є вбудованим у процес розроблення, а не буде відкладено до фінальних етапів; (2) забезпечує ефективний зворотний зв'язок, відповідний для кожного етапу; (3) передбачає постійний перегляд і оптимізацію набору тестів для усунення надмірності та максимального покриття бізнес-ризиків; (4) end-to-end-тести, що оцінюють роботу кінцевого користувача з усіма пов'язаними технологіями (фронт-енд і бек-енд).

Здійснення безперервного автоматизованого тестування стає можливим завдяки спеціальним інструментам безперервної інтеграції. Аналіз теорії та практики засвідчує, що вибір адекватного інструменту безперервної інтеграції для підтримки тестування, який буде відповідати усім вимогам бізнесу та клієнта, залишається проблемою. Відповідно до джерел, складним завданням також є пошук та поєднання інструментів, сумісних між собою, з метою створення функціонального конвеєра. Для того, щоб повністю впровадити конвеєр безперервного розгортання, потрібно багато інструментів (автоматизована система складання, тестування, розгортання тощо). Проблема, що виникає внаслідок комбінації інструментів, полягає в їхній конфігурації. Більшість проектів потребують своїх власних конфігурацій для задоволення бізнес-потреб. Самі інструменти мають власні способи конфігурації, які ще не стандар-

тизовані. Це призводить до гальмування процесу налаштування кожного сегмента конвеєра. Для деяких проектів ця кількість зусиль для створення конвеєру є однією з основних причин, через які вони не використовують постійне розгортання [7; 10; 11].

Викладене вище актуалізує вирішення завдання вибору відповідного інструменту CI для автоматизованого тестування ПЗ шляхом побудови моделі оцінювання таких інструментів.

### Мета та постановка завдання

Метою цієї роботи є створення моделі оцінювання інструментів CI для процесу тестування програмних продуктів за допомогою системи підтримки прийняття рішень (СППР).

Для досягнення поставленої мети необхідно проаналізувати можливості основних інструментів CI та визначити критерії їхнього оцінювання; обрати СППР та розробити модель оцінювання інструментів CI за допомогою обраної системи; запровадити рекомендований системою інструмент CI для тестування програмного продукту, отримати та проаналізувати результати використання моделі оцінювання.

### Виклад основного матеріалу

Під час роботи було проведено порівняльний аналіз переваг та недоліків найбільш уживаних сьогодні засобів CI, таких як Jenkins, GitlabCI, CircleCI, TeamCity, Bamboo [3; 6; 8; 9; 10].

Виходячи із проведеного аналізу характеристик зазначених інструментів, було визначено основні критерії, що полегшують вибір CI-засобу для автоматизованого тестування. До них належать такі характеристики:

- вартість інструменту;
- інтеграція з інструментами сторонніх розробників (Maven, Allure Report);
- синхронізація з системами управління репозиторіями (Github/Gitlab);
- можливість налагодження для перевірки коду, який був оновлений;
- середовище розміщення інструменту (чи є інструмент готовим рішенням із вебінтерфейсом або потребує інсталяції).

Для побудови моделі оцінювання інструментів CI за визначеними критеріями було використано СППР «Вибір», що є аналітичною системою, основою на застосуванні методу аналізу ієрархій [5]. Метод аналізу ієрархій (MAI), що належить до класу критеріальних методів, є математичним інструме-

нтом системного підходу до складних проблем прийняття рішень.

На підготовчому етапі побудови моделі оцінювання в СППР було встановлено вагомість визначених критеріїв шляхом опитування експертів команди DevOps. Результати опитування (рис. 1) показали, що найбільш вагомими критеріями є вартість інструменту та синхронізація із системами управління репозиторіями. Однаково важливими є інтеграція з інструментами сторонніх розробників та можливість налагодження, а найменш вагомим критерієм є середовище розміщення продукту.



Рис. 1. Результати опитування експертів щодо вагомості критеріїв оцінювання CI інструментів

Крім цього, експертам було запропоновано оцінити відповідність кожного інструменту критеріям за 5-бальною шкалою. Результати опитування експертів було використано для побудови моделі оцінювання інструментів.

Унаслідок було отримано ієрархію для оцінювання інструментів, яка пов'язує мету оцінювання, критерії та альтернативи, що є в нашому випадку інструментами CI, з яких необхідно здійснити вибір. Отримана ієрархія для оцінювання інструментів зображена на рис. 2.

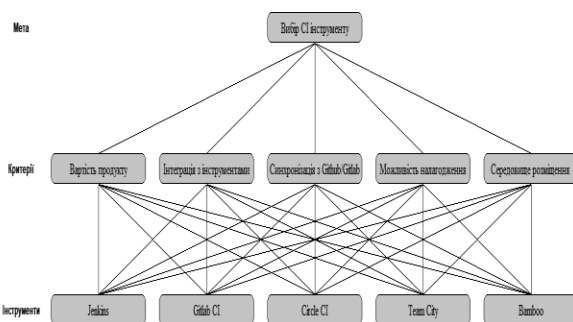


Рис. 2. Побудована ієрархія для оцінювання інструментів CI

Далі було побудовано матрицю попарних порівнянь критеріїв (рис. 3). Вона була запо-

внена результатами опитування експертів щодо вагомості критеріїв.

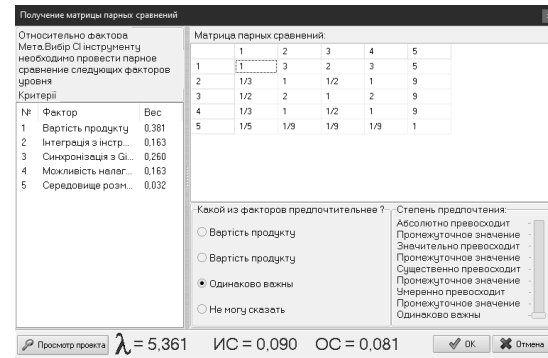


Рис. 3. Побудована матриця попарних порівнянь критеріїв

На цьому етапі також було розраховано індекс узгодженості (IU) суджень експерта, який визначається за такою формулою:

$$IU = \frac{\lambda_{\max} - n}{(n-1)}$$

де  $\lambda_{\max}$  – максимальне власне значення (ВЗ);  $n$  – розмірність матриці.

Розрахунок максимального власного значення  $\lambda_{\max}$  здійснюється за матрицею парних порівнянь таким чином: підсумовується кожен стовпець суджень, сума першого множиться на величину першої компоненти нормалізованого вектора пріоритетів, сума другого стовпця – на величину другої компоненти тощо, потім отримані числа підсумовуються.

Відношення узгодженості розраховується як  $VU = (IU / C3) * 100\%$ . Якщо VU більше ніж 10 %, необхідний перегляд суджень.

На наступному етапі були складені матриці парних порівнянь альтернатив за всіма критеріями, і в такий спосіб проведено порівняння інструментів CI за всіма критеріями (рис. 4).

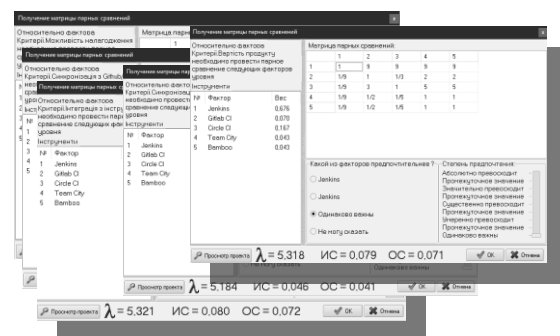


Рис. 4. Побудовані матриці попарних порівнянь інструментів CI

Після використання моделі оцінювання було отримано результат у вигляді діаграми (рис. 5), де рекомендованим інструментом інтеграції визначено інструмент CircleCI.

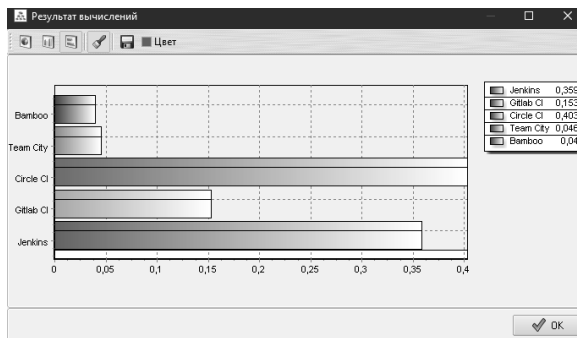


Рис. 5. Результат використання моделі оцінювання інструментів CI

Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту (вебзастосунку для роботи з постачальниками деяких товарів).

У процесі підготовки до використання рекомендованого інструменту було розроблено 20 тест-кейсів та відповідний конфіг-файл, після чого до CircleCI було приєднано репозиторій із тестами. Завдяки налаштуванням у конфіг-файлі тести запускалися автоматично кожного дня. Унаслідок на головній сторінці інструменту відображалися всі запуски, що виконувалися після кожного коміту до репозиторію. Перегляд запусків дає змогу зрозуміти, чи є проблеми з продуктом після певної зміни в коді. Разом із статусом виконання вказується назва гілки в Git та код коміту, що дозволяє дізнатися, які саме зміни вплинули на роботу вебзастосунку.

Підсумовуючи результати використання інструменту CircleCI, рекомендованого СППР для автоматизованого тестування реального програмного продукту, можна зробити висновки щодо доцільності його застосування у практиці безперервної інтеграції. Зокрема встановлено, що використання CircleCI полегшує роботу тестувальників за рахунок налаштованої автоматизованості, а звіт Allure Report, що є артефактом кожного запуску тестів, значно підвищує швидкість формування звітів під час регресійного тестування.

Установлено також, що інструмент Circle CI відповідає загалом усім критеріям оцінювання. Вартість інструменту виправдана,

оскільки для забезпечення автоматизованого тестування невеликого проекту достатньо безкоштовної версії. Проблем з інтеграцією з інструментами сторонніх розробників не виявлено, проте встановлено, що артефакти доступні для завантаження лише протягом 30 днів. Під час використання інструменту репозиторій з кодом розміщується на GitHub, проблем із синхронізацією не виникало, вона реалізована зручно та швидко. Робота з інструментом CircleCI, що є самостійним вебзастосунком з інтуїтивно зрозумілим інтерфейсом і не потребує спеціальних завантажень та налаштувань, не викликала ускладнень і додаткового навчання персоналу.

### Висновки

Отже, у роботі проаналізовано можливість основних інструментів CI для автоматизованого тестування та визначено критерії їхнього оцінювання. На основі визначених критеріїв розроблено модель оцінювання інструментів CI за допомогою СППР та використано її для вибору відповідного інструменту. Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту. Отримано й проаналізовано результати використання моделі оцінювання.

### Література

1. Автоматизоване тестування: підготовка стратегії. URL: <https://dou.ua/lenta/articles/preparation-for-test-automation/>.
2. Автоматизація тестування: як уникнути поширених помилок. URL: <https://www.globallogic.com/ua/insights/blogs/qa-automation-2/>.
3. Безперервна інтеграція: CircleCI vs Travis CI vs Jenkins. URL: <https://habr.com/ru/company/southbridge/blog/>.
4. Принципи CI/CD. URL: <https://medium.com/southbridge/>.
5. СППР Вибір. URL: <https://allsoft.ru/software/vendors/tsiritas/-spgr-vybor/>.
6. Що таке безперервна інтеграція Дженкінса. URL: <https://uk.myservername.com/jenkins-ci>.
7. A Basic Introduction to DevOps Tools. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.735.292>.
8. Atlassian Bamboo. URL: <https://www.atlassian.com/ru/software/bamboo>.
9. Comparison of Most Popular Continuous Integration Tools: Jenkins, TeamCity, Bamboo, Travis CI and more. URL: <https://www.altexsoft.com/blog/engineering/comparison-of-most-popular-continuous-integration-tools>.

10. Continuous integration and delivery – Circle CI. URL: <https://circleci.com/>.
11. DevOps methodology and process. URL: <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>.

### References

1. Automated testing: strategy preparation. URL: <https://dou.ua/lenta/articles/preparation-for-test-automation/> [in Ukrainian].
2. Automated testing: how to avoid common mistakes. URL: <https://www.globallogic.com/ua/insights/blogs/qa-automation-2/> [in Ukrainian].
3. Continuous integration: CircleCI vs Travis CI vs Jenkins. URL: <https://habr.com/ru/company/southbridge/blog/332836/> [in Ukrainian].
4. Principles of CI/CD. URL: <https://medium.com/southbridge/> [in Ukrainian].
5. SSTD Vybor. URL: <https://allsoft.ru/software/vendors/tsiritas/-sppr-vybor/> [in Russian].
6. What is continuous integration of Jenkins. URL: <https://uk.myservername.com/-jenkins-ci-tutorial-what-is-jenkins/> [in Ukrainian].
7. A Basic Introduction to DevOps Toolsю. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.735.292>.
8. Atlassian Bamboo. URL: <https://www.atlassian.com/ru/software/bamboo>.
9. Comparison of Most Popular Continuous Integration Tools: Jenkins, TeamCity, Bamboo, Travis CI and more. URL: <https://www.altexsoft.com/blog/engineering/comparison-of-most-popular-continuous-integration-tools>.
10. Continuous integration and delivery – Circle CI. URL: <https://circleci.com/>
11. DevOps methodology and process. URL: <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>.

**Гризун Людмила Едуардівна**, д.п.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, Lgr2007@ukr.net, тел. +38 0954139386,

**Скорін Юрій Іванович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, skorin.yuriy@gmail.com, тел. +38 066-748-47-51,

**Деточенко Ірина Миколаївна**, магістрант каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця,

detochenko@hneu.net, tel. +38 066-423-85-02.

### Choosing a continuous integration tool for software automated testing

**Abstract. Problem.** Automated testing is directly related to continuous integration (CI), which is the performance of frequent automated grouping of project components to identify and resolve integration issues. Choosing the adequate CI tool for automated testing is the key to effective CI process. Analysis of theory and practice shows that the choice of such an instrument to support testing remains a challenge. According to the sources, finding and combining compatible tools to create a functional pipeline is also a difficult task. The problem that arises due to the combination of tools is their configuration. The said above actualizes the search of the solution to the problem of choosing the appropriate CI tool for software testing by building a model for evaluating these tools. **Goal.** The aim of the work is to create a model for evaluating CI tools for the software testing process using the decision support system (DSS). **Methodology.** The analytical methods of research, the method of the hierarchy analysis, and the method of modeling are used. **Results.** In the course of the work, the possibilities of the main CI tools for automated testing were analyzed and the criteria for their evaluation were determined. **Originality.** Based on the defined criteria, a model for evaluating CI tools using DSS was developed and used to select the appropriate instrument. **Practical value.** The tool for continuous integration recommended by DSS, CircleCI, has been implemented at the enterprise to test a real software product. The results of using the evaluation model are obtained and analyzed.

**Key words:** tools for continuous integration, automated software testing, evaluation model, decision support system.

**Gryzun Liudmyla**, Doct. of Science, Professor, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Lgr2007@ukr.net, tel. +38 0954139386,

**Skorin Yuriy**, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, skorin.yuriy@gmail.com, tel. +38 066-748-47-51,

**Detochenko Iryna**, master-student, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, detochenko@hneu.net, tel. +38 066-423-85-02.

## ІМПЛЕМЕНТАЦІЯ ТА РЕЙТИНГ РІШЕНЬ WEB-АНАЛІТИКИ НА ОСНОВІ ХМАРНИХ СЕРВІСІВ

Знахур Л. В., Знахур С. В.

Харківський національний економічний університет ім. С. Кузнеця

*Анотація.* Вартість володіння інфраструктури для WEB-аналітики e-commerce сайтів є важливим аспектом ефективності інтернет-бізнесу. У роботі проведено аналіз сучасних рішень, архітектур щодо розв'язання наскрізного завдання створення та імплементації WEB-аналітики e-commerce сайтів. Проблемою, яку вирішує дослідження, є визначення оптимальної архітектури хмарних сервісів для реалізації WEB-аналітики e-commerce сайтів на основі використання інтегрального показника, який містить адитивну функцію показників функціональності рішення та вартості володіння обраних хмарних сервісів для побудови рішення.

*Ключові слова:* e-commerce, ML, WEB-аналітика, інтегральна оцінка, імплементація.

### Вступ

Криза 2020–2021 рр. показала, що функціональність та вартість володіння інфраструктури для WEB-аналітики e-commerce сайтів є дуже важливим аспектом ефективності інтернет-бізнесу. Ціна володіння інфраструктури провідних провайдерів хмарних сервісів має свої особливості та залежить від обраної архітектури, кількості та специфіки сервісів, часу їхнього використання. Також необхідно враховувати можливість масштабування та інтеграції рішення до власних сервісів або даних за рахунок використання нативних конекторів, API та різних додаткових служб, що дозволяють підключати інші зовнішні сервіси, необхідні для побудови наскрізної WEB-аналітики e-commerce сайтів.

### Аналіз публікацій

Дослідженню питань побудови наскрізних рішень щодо аналізу трафіку сайтів e-commerce на основі інструментів веб-аналітики присвячено праці таких учених, як Х. Карджалото, П. Сінгх, К. Родден, Р. Томас, Х. Хатчінсон, Б. Янсен та ін. Проте є необхідність практичного дослідження сучасних рішень та розроблення методичних підходів щодо комплексного обґрунтованого вибору архітектур та інтеграції рішень WEB-аналітики.

Згідно з дослідженнями [3, 4], сучасний інтернет-бізнес потребує вирішення таких завдань:

1. Створення та актуалізація системи наскрізної аналітики (dwh, etl, дашборди).
2. Створення оперативного маркетингового аналізу діяльності бізнесу та товарної аналітики на основі даних із різних джерел.

3. Автоматизованого впровадження рекомендаційної системи на основі ML у сфері аналітики та сегментації аудиторії.

Інтерес до ML зростає щороку. Лідери e-commerce залучені до R&D та застосування машинного навчання та поглибленого аналізу даних для завдань бізнесу та WEB-аналітики. Упроваджуються результати успішних проектів на рівні промислової експлуатації. Виникає потреба й у методології вибудовування бізнес-процесів навколо машинного навчання та в інструментах автоматизації цих процесів для повного циклу WEB-аналітики. Цей напрям отримав назву ModelOps WEB-аналітики. На ринку вже є понад 70 рішень та інструментів, так чи інакше пов'язаних із завданнями ModelOps, серед яких безкоштовні, відкриті рішення (наприклад MLFlow, KubeFlow) та рішення від лідерів у галузі Data Science, таких як AWS, GCP, Azure та IBM. Архітектурні рішення мають аналогічні підходи щодо методології управління життєвим циклом моделей машинного навчання, але імплементації моделей залежать від обраної концепції загальної архітектури. Досить популярним рішенням є мікросервісна архітектура або використання API для моделей.

Проблемою, яку вирішує дослідження, є визначення оптимальної архітектури хмарних сервісів для реалізації WEB-аналітики e-commerce сайтів на основі використання інтегрального показника, що містить адитивну функцію показників функціональності рішення та вартості володіння обраних хмарних сервісів для побудови рішення. Для вирішення проблеми вибору та побудови оптимальної архітектури було визначено такі завдання:

1. Дослідити необхідні інструменти, стандарти та показники WEB-аналітики (у роботі розглянуто тільки Google Analytics).

2. Проаналізувати сучасні архітектури, технології та сервіси щодо впровадження сайтів та інтеграції Google Analytics, рішень ML.

3. Реалізувати різні хмарні архітектури та підключити різні аналітичні індикатори Google Analytics.

4. Побудувати рейтинг архітектурних рішень на основі інтегрального показника.

Використання Google Analytics 4 (GA4) суттєво змінило сучасний облік WEB-аналітики. Однією з основних переваг GA4 є машинне навчання та NLP-функції, які застосовуються для того, щоб:

передбачати можливість конверсії та створювати на основі цих прогнозів аудиторії для Google Ads;

попереджати про важливі тенденції в даних. Наприклад, про товари, на які зростає попит через те, що змінюються потреби користувачів;

знаходити аномалії;

передбачати можливість відтоку клієнтів.

Детальний аналіз наявних рішень показав, що проблема інтеграції Google Analytics до хмарних рішень провайдерів має декілька рівнів. Перше: 2020 р. було змінено стандарт та концепцію Google Analytics, тобто зроблено перехід на стандарт GA4. Це суттєво змінило кількість сервісів конекторів до даних GA4, оскільки більшість комерційних рішень були реалізовані для Google Analytics (universal). Друге: рішення на основі сервісів AWS або Azure потребують додаткові налаштування, дозволи та ролі для підключення до GA. Також рішення AWS та Azure потребують додаткові процедури ETL (вони можуть бути реалізовані на основі лямбда-функцій) для процесингу даних GA. Ці особливості використання даних Google Analytics суттєво збільшують трудомісткість та вартість володіння архітектур, які не базуються винятково на сервісах GCP [6].

### Мета та постановка завдання

Метою роботи є аналіз сучасних рішень, архітектур щодо вирішення наскрізного завдання створення та імплементації WEB-аналітики e-commerce сайтів.

Завдання роботи – визначити необхідні інструменти, сервіси та показники для WEB-аналітики; проаналізувати сучасні архітектури, технології та сервіси щодо впро-

вадження сайтів та інтеграції Google Analytics, упровадження рішень ML на основі даних Google Analytics; здійснити реалізацію різних архітектурних рішень щодо оперативного аналізу даних Google Analytics, формування аналітичної звітності, побудови рейтингу архітектурних рішень.

Об'єктом дослідження є впровадження інформаційних технологій, сервісів для побудови рішень щодо реалізації ML-моделей на основі даних WEB-аналітики e-commerce сайтів.

Предметом дослідження є побудова та обґрунтування оптимальної архітектури WEB-аналітики e-commerce сайтів для оперативного та ML-аналізу даних Google Analytics.

Наукова новизна роботи полягає в розробленні методики щодо рейтингу рішень відомих провайдерів, які будуть зіставлені згідно з параметрами функціональності та вартості володіння.

### Виклад основного матеріалу

Для виконання завдань роботи щодо побудови рейтингу рішень WEB-аналітики e-commerce сайтів було здійснено дослідження особливостей побудови та імплементації WEB-аналітики на основі хмарних сервісів AWS, GCP, Azure. Загальна концепція вибору оптимальної архітектури рішення передбачає розрахунок компонент оцінки функціональності WEB-аналітики та компонент TCO (total cost of ownership) – вартості володіння рішень. На рис. 1 наведено контекстну діаграму «Проведення експерименту щодо імплементації та рейтингу рішень WEB-аналітики для e-commerce сайтів», на рис. 2 наведено її декомпозицію.

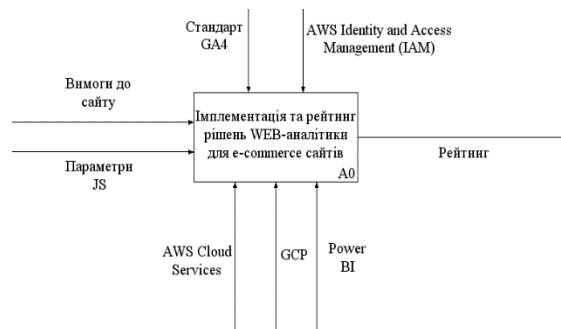


Рис. 1. Контекстна діаграма «Проведення експерименту щодо імплементації та рейтингу рішень WEB-аналітики для e-commerce сайтів»



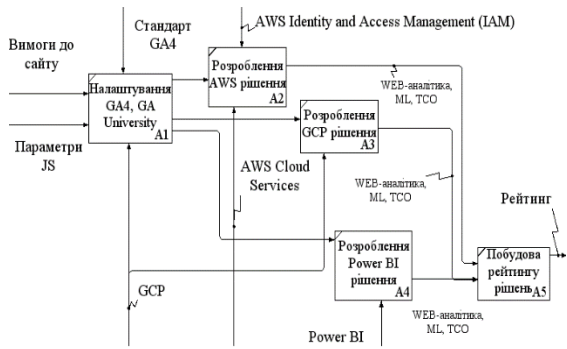


Рис. 2. Декомпозиція процесу «Проведення експерименту щодо імплементації та рейтингу рішень WEB-аналітики для e-commerce сайтів»

Запропонований у роботі підхід щодо побудови рейтингу рішень WEB-аналітики для e-commerce сайтів містить такі етапи:

1. Розроблення (або отримання дозволу щодо адміністрування) e-commerce сайту.
2. Створення GA-акаунтів для e-commerce сайту. Пропонується використати стандарт GA (universal) та GA4. Це дозволить отримати різні дані аналітики та використати різні конектори в архітектурі та сервісах.
3. Реалізація архітектури AWS для аналізу даних GA, GA4. Варіант реалізації необхідно здійснити в межах одного часового інтервалу трафіку на e-commerce сайт. Розраховується час виконання запитів та побудови дашбордів. Для аналізу часу отримання результатів моделей ML пропонується реалізувати класичну модель ML (Tree Decision) на основі даних GA.
4. Розрахунок вартості володіння архітектурою AWS на базі AWS Pricing Calculator (<https://calculator.aws/#/estimate>).
5. Реалізація обраної архітектури GCP для аналізу даних GA, GA4. Варіант реалізації необхідно здійснити в межах одного часового інтервалу трафіку на e-commerce сайт. Розраховується час виконання запитів та побудови дашбордів. Для аналізу часу отримання результатів моделей ML пропонується реалізувати класичну модель ML (Tree Decision) на основі даних GA.
6. Розрахунок вартості володіння архітектурою GCP на базі <https://cloud.google.com/products/calculator>.
7. Побудова моделі для дашборду GA на основі використання конектора GA (universal) PowerBI. Конектор GA4 не реалізовано в PowerBI. Для аналізу часу отримання результатів моделей ML пропонується ре-

алізувати класичну модель ML (Tree Decision) на основі даних GA.

8. Формування показників за методикою побудови інтегрального показника.

9. Розрахунок рейтингу для кожної архітектури рішення на основі значень інтегрального показника.

Для побудови інтегрального показника архітектурних рішень було запропоновано використати такі показники функціональності WEB-аналітики, складності архітектури та компонент TCO (табл. 2). Вибір показників є результатом анкетування WEB-аналітиків, DevOps та MLOps інженерів.

Таблиця 2 – Показники функціональності WEB-аналітики, складності архітектури та компонент TCO

Показник	Опис
Кількість сервісів, використаних в архітектурі	Складність архітектури визначається кількістю хмарних сервісів, що необхідно використати для побудови масштабованого рішення. Збільшення кількості сервісів збільшує складність архітектури за рахунок інтеграції, налаштування IAM (ролі, повноваження). Еталонне значення – максимальне
Час оновлення даних	Технологічні особливості оновлення даних GA. Еталонне значення – мінімальне
Вартість рішення	TCO – розраховується на основі відповідних калькуляторів хмарних провайдерів. Еталонне значення – мінімальне
Кількість показників аналітики	Максимальна кількість показників, які підтримує обрана технологія інтеграції. Еталонне значення – максимальне
Джерело даних	Старий GA (0) або новий стандарт GA4 (1). Еталонне значення – максимальне
Інтеграція (або наявні) ML	Нативна підтримка або інтеграція ML (так чи ні) (0 або 1). Еталонне значення – максимальне
Масштабування рішення	Можливість масштабування (так чи ні) (0 або 1). Еталонне значення – максимальне
Імплементація аналітики у WEB	Можливість імплементації у Web на основі iframe (так чи ні) (0 або 1). Еталонне значення – максимальне

Розрахунок інтегрального показника базується на формулі:

$$I_j = \sum_{i=1}^m P_{ij}^n, \quad (1)$$

де  $P_{ij}^n$  –  $i$ -ий нормований показник моделі для  $j$ -архітектури;  $m$  – кількість архітектур.

$$P_{ij}^n = P_{ij} \div P_{ij\max}, \quad (2)$$

де  $P_{ij}$  –  $i$ -ий показник моделі для  $j$ -архітектури;  $P_{ij\max}$  – максимальне значення  $i$ -го показника  $j$ -ої архітектури.

Максимальне значення показника розраховується відповідно до напрямку дії показника. Так, еталонний показник може бути максимальним (наприклад, кількість показників WEB-аналітики в рішенні), або мінімальним (наприклад, час розрахунку або трансферу даних). Якщо використовується як еталон мінімальне значення, тоді  $P_{ij}$  розраховується, як  $1 \div P_{ij}$ .

Для оцінювання та рейтингу рішень використано підхід розрахунку інтегральних показників із нормуванням. Результати розрахунку показників для отримання інтегрального показника наведено в табл. 1.

Таблиця 1 – Результати розрахунку показників архітектур

Показники	Оцінка / AWS	Оцінка / GCP	Оцінка / Azure
Сервіси, використані в архітектурі	0.16/S3, AppFlow, Amazon Athena, Amazon QuickSight	0.3/Data Studio, BigQuery, AppEngine	1/Microsoft Power BI
Час оновлення даних (до 1 Гб)	0,2/10 хв	1/2 хв	0,01/24 год
Вартість рішення	0,37	0,38	1
Кількість показників	1/50 спеціальних параметрів і 50 спеціальних показників	1/50 спеціальних параметрів і 50 спеціальних показників	0,4/20 показників
Джерело даних	0/GA4	0/GA4	0/GA (universal)
Інтеграція ML	0/QuickSight, SageMaker	0/python (function), BigQuery	0/Python Script, API
Масштабування рішення	1/існує	1/існує	0/не існує
Імплементация	1/iframe	1/iframe	1/iframe
Інтегральна оцінка	3,73	4,68	3,41

Для розрахунку показників реалізовано скрипт оцінки часу виконання запитів та розрахунок вартості володіння архітектурами.

особливості імплементации рішення або ML-моделей можуть змінити рейтинг.

### Висновки

Формальний підхід щодо оцінки архітектурних рішень показав, що фаворитом є архітектурне рішення на основі GCP із максимальним результатом 4,68 балів. Архітектурне рішення AWS має 3,73 бали, а рішення Azure – 3,41. Але варто врахувати, що в аналізі не було використано ваг для параметрів оцінок. Так, наприклад, час оновлення даних зазвичай має максимальну вагу та може суттєво впливати на результат рейтингу. Необхідно також зазначити, що вибір сервісів для архітектур GCP та AWS містить тільки базові сервіси з мінімальними характеристиками щодо інстанс, зберігання даних, безпеки та масштабування. Збільшення вимог щодо продуктивності промислових архітектур та

### Література

1. AWS: быстрая разработка, сборка и развертывание приложений на AWS. URL: <https://aws.amazon.com/ru/codestar/>
2. BigQuery. URL: <https://cloud.google.com/bigquery>.
3. Burby, J., & Brown, A. Web Analytics Definitions – Version 4.0. URL: <http://www.digitalanalyticsassociation.org/>
4. Data Age 2025: The digitization of the world – From edge to core. White paper. Farmingham, MA. URL: <https://www.seagate.com/files/www-content/our-story/trends/files/idcseagate-dataage-whitepaper.pdf>.
5. Google Analytics (n.d.). Features. URL: <http://www.google.com/analytics/features/>
6. GCP. URL: <https://cloud.google.com/solutions/retail>.

### References

1. AWS: fast development, assembly and deployment of applications on AWS. URL: <https://aws.amazon.com/ru/codestar/>
2. BigQuery. URL: <https://cloud.google.com/bigquery>
3. Burby, J., & Brown, A. Web Analytics Definitions – Version 4.0. URL: <http://www.digitalanalyticsassociation.org/>
4. Data Age 2025: The digitization of the world - From edge to core. White paper. Farmingham, MA. URL: <https://www.seagate.com/files/www-content/our-story/trends/files/idcseagate-dataage-whitepaper.pdf>
5. Google Analytics (n.d.). Features. URL: <http://www.google.com/analytics/features/>
6. GCP. URL: <https://cloud.google.com/solutions/retail>

**Знахур Людмила Володимирівна**, ст. викл. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 050-65-11-282, [razinalv@gmail.com](mailto:razinalv@gmail.com),  
**Знахур Сергій Вікторович**, к.е.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 050-30-04-390, [serhii.znakhur@gmail.com](mailto:serhii.znakhur@gmail.com).

### Implementation and rating of Web-analysis solutions based on cloud services

**Abstract.** *The total cost of ownership WEB-analytics of e-commerce sites is a very important aspect of the Internet business effectiveness. The thesis includes the modern solutions, architectures for solving the cross-cutting problem of creating and implementing*

*WEB-analytics of e-commerce sites. Goal. The goal is to determine the optimal architecture of cloud services for the implementation of WEB-analytics of e-commerce sites. The object of research is the introduction of information technology, services for building and implementation of ML models based on data WEB-analytics of e-commerce sites. The subject of the research is the architecture development for WEB-analytics and ML analysis of Google Analytics data. The scientific novelty of the work is to develop a methodology for rating cloud providers solutions which are compared according to the parameters of functionality and cost of the ownership.*

**Methodology.** *The system analysis and analytical and empirical methods of research were used for development and testing of cloud solutions. Results.* *Solving includes investigation of the necessary tools, standards and indicators of WEB-analytics, architectures, technologies, services for the implementation of sites and integration of Google Analytics, ML solutions. Cloud solutions of different architectures were developed and tested. Practical value.* *Rating methodology was used for the best Cloud solutions estimation.*

**Key words:** *e-commerce, ML, WEB-analytics, TCO, architecture, cloud service.*

**Znakhur Liudmyla**, Lecturer of Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 050-65-11-282, [razinalv@gmail.com](mailto:razinalv@gmail.com),  
**Znakhur Serhii**, Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 050-300-43-90, [serhii.znakhur@gmail.com](mailto:serhii.znakhur@gmail.com).

УДК 629.341

DOI: 10.30977/BUL.2219-5548.2022.96.0.40

## РЕАЛІЗАЦІЯ ПОШУКУ АНАЛОГІЧНИХ ТОВАРІВ НА ОСНОВІ FAISS

Знахур С. В., Знахур Л. В.

Харківський національний економічний університет ім. С. Кузнеця

***Анотація.** Проблема отримання повної множини аналогічних товарів різних постачальників з урахуванням інформації щодо зображення та опису товару є актуальною для e-commerce та практично цікавою. Робота містить алгоритм побудови загальної архітектури рішення засобами GCP та створення AI компонент на базі FAISS та Deep Learning для реалізації Product Quantization пошуку товарів на основі їхніх векторів текстових ознак та зображень. Запропонований підхід пошуку аналогічних товарів використовує мережу DNN для виділення features-зображень, текстових тегів та формування на їхній основі векторів, що застосовуються для FAISS-кластеризації.*

***Ключові слова:** AI, ML, Deep Learning, FAISS, GCP, e-commerce, кластеризація.*

### Вступ

Сучасні e-commerce, що працюють за схемами дропшипінгу чи маркетплейсу, на відміну від звичайних магазинів, не закуповують товар у постачальників, вони використовують зовнішню інформацію (відповідно до своєї системи) про їхні товари для організації бізнесу B2B або B2C. Організація інформаційних потоків e-commerce базується на достовірній та максимально повній інформації щодо товарів різних постачальників. Тому один із необхідних елементів автоматизації процесів e-commerce – це автоматичне створення опису товарів та їхнє зіставлення. Автоматичне зіставлення товарів допомагає виявити схожі товари й вказати ступінь схожості. Це особливо важливо для товарів, що складно піддаються порівнянню, наприклад одяг, аксесуари. Отже, у роботі визначимо базовими завданнями функціоналу маркетплейсу такі: пошук релевантного товару серед аналогічних товарів на маркетплейсі; отримання повної множини аналогічних товарів різних виробників і постачальників з урахуванням нечіткої або неповної інформації опису товару.

### Аналіз публікацій

Проведений аналіз показав напрями використання AI, які є комерційними перевагами сучасних e-commerce магазинів та маркетплейсів [1, 2]:

1. Тегування товарів на основі їхнього зображення та тегування товарів на основі NLP-аналізу їхнього текстового опису.

2. Пошук товарів на основі їхнього зображення або текстового опису.

3. Удосконалення рекомендаційної системи на основі масштабування, використання

гібридних підходів DSSM, Attentive Collaborative Filtering.

4. Підбір набору товарів та комбінації товарів для покупця.

5. Персоналізація пошуку та вибору товарів на основі поведінкової моделі користувача.

6. Використання онлайн-моделювання в процесі вибору або підбору товарів (зокрема 3D-моделювання).

Аналіз показав, що більшість AI рішень та сервісів є типовими для конкуруючих компаній у сфері e-commerce. Зазвичай це такі групи сервісів: тегування товарів на основі їхнього зображення та тегування товарів на основі NLP-аналізу їхнього текстового опису; пошук товарів на основі зображення; рекомендаційні системи та інструменти мапінгу (зіставлення зображень товарів або опису). Точність результатів рішень у конкурентів кардинально не відрізняється, оскільки вони використовують спеціалізовані DNN (Deep Neural Network), які мають точність від 90 до 97 % [1, 5, 6]. Особливістю є застосування власних наборів даних та розмітки для тегування товарів, постійна актуалізація та тюнінг DNN-мереж. Найбільш складною та унікальною розробкою є рекомендаційна система для користувачів щодо підбору товарів. Якість її роботи визначає точність підбору та ступінь персоналізації результатів для споживача. Рекомендаційну систему можливо побудувати на основі моделей DNN, що показали найбільшу точність для пошуку товарів на основі визначених атрибутів та ознак features (наприклад ResNet, EfficientNet) та на основі алгоритмів і мереж для рекомендації товарів на базі поведінкової моделі Deep Learning Recommender

System [2, 3]. Але необхідно зазначити, що базовим функціоналом кожної рекомендаційної системи є пошук схожих товарів.

### Мета та постановка завдання

Мета роботи – побудова алгоритму та архітектурного рішення пошуку аналогічних товарів маркетплейсу на основі використання алгоритму FAISS та Deep Learning. Робота передбачає огляд напрямів використання AI та передумов дослідження інтелектуального пошуку товарів для e-commerce, загальний алгоритм пошуку схожих товарів, реалізацію AI-компонент на базі FAISS і DeepLearning для Product Quantization пошуку товарів на основі використання їхніх текстових ознак та векторів зображень.

### Виклад основного матеріалу

Ключові передумови дослідження інтелектуального пошуку товарів для e-commerce такі:

1. Наявність тренду smart-покупки. Користувачі прагнуть купувати брендові або якісні речі за максимально низькими цінами, тому здійснюють пошук аналогічних товарів у різних постачальників.

2. Користувачі все більше здійснюють покупки, застосовуючи мобільні пристрої в режимі онлайн, що потребує швидкого та інтуїтивно зручного пошуку.

3. Наявність високої конкуренції серед маркетплейсів та електронних магазинів. Ефективний та інтелектуальний пошук товарів є значною перевагою в e-commerce.

4. Використання AI та Big Data у маркетплейсі, як обов'язкового компоненту.

5. Наявність значної кількості провайдерів AI та Big Data сервісів для маркетплейсів, агрегаторів, магазинів.

6. Доступність сучасних технологій, архітектур та технічних ресурсів для побудови або оренди AI-рішень. Існують відкриті рішення, навчені мережі для розпізнавання зображень товарів, аксесуарів на фото, тегування, кластеризації об'єктів (наприклад YOLO v3).

Розглянемо три напрями розвитку AI та Big Data рішень, які є актуальними для e-commerce:

1. Рішення, що оптимізують процеси (flow) покупки та надають додаткові цінності для покупця. Наприклад, це рішення, які побудовані на основі використання AI, Big Data, WEB-аналітики, карт поведінки, кліків, статистики поведінки та вибору користува-

чів, прогнозування на основі моделі поведінки, профілю користувачів тощо. Ці рішення дозволяють у режимі реального часу пропонувати (рекомендувати) товари певної групи (категорії), зменшити кількість пропозицій. Такі рішення становлять 80 % трудомісткості робіт із розроблення AI, Big Data та операційних витрат BI сучасних маркетплейсів.

2. AI та Big Data рішення, що оптимізують процеси (flow) продавця щодо зручності завантаження, контролю та моніторингу даних. Наприклад, завантаження даних і валідація щодо повноти опису товару, якості фото, наявності описових тегів. Для процесів (flow) продажів продавця – це аналіз даних продажів, пошук закономірностей продажів за товарами та їхніми групами, прогнозування та рекомендації щодо товарів.

3. AI та Big Data рішення, які забезпечують та оптимізують роботу адміністраторів маркетплейсу. Наприклад, оптимізація зберігання даних, оптимізація ETL, оптимізація запитів, пошук та виявлення інцидентів, передбачення інцидентів, оптимізація виконання запитів для агрегацій та звітів.

Найбільш цікавим є рішення AI, які оптимізують процеси (flow) покупки. Більшість AI та Big Data рішень e-commerce недоступні для детального аналізу на рівні архітектури, але можливо дослідити моделі, які покладені в їхню основу. Найбільш поширені моделі AI – це моделі класифікації та кластеризації товарів або запитів. Результати аналізу публікацій щодо методів та моделей класифікації або кластеризації товарів, що використовуються для пошуку та мапінгу, такі:

1. Точність алгоритмів та методів ґрунтується на використанні спеціалізованих, розмічених наборів даних (наприклад, MNIST [5])

2. Тегування зображень товарів (ідентифікація та класифікація) вже мають максимальну точність 92–96 % [4, 5], що дозволяє суттєво знизити помилки пошуку.

3. Пошук аналогічних об'єктів (товарів) ефективно реалізовувати в системах інтеграції на основі кластеризації (найбільша оперативність в умовах онлайн-пошуку).

4. DNN більш ефективні, ніж класичні методи ML для визначення ознак товарів та вирішення завдання класифікації [5].

Отже, для виконання завдань пошуку аналогічних товарів використаємо сучасний підхід на основі кластеризації, що передбачає вилучення n-вимірної вектора ознак товару (embeddings vector); зіставлення товарів за

векторами ознак (на основі кластеризації). Для реалізації цих кроків пропонується використати бібліотеку TensorFlow. Пошук товарів за векторами можливо реалізувати на основі бібліотеки FAISS [6, 8]. FAISS – це бібліотека для ефективного пошуку схожості і кластеризації векторів. Вона містить алгоритми пошуку в наборах векторів будь-якого розміру. Бібліотека має підтримку GPU, що значно пришвидшує процеси пошуку.

Загальний алгоритм пошуку схожих товарів наведено нижче та містить такі блоки:

1. Угрупування товарів за категоріями. Для практичного використання рішення рекомендується угрупувати товари тільки верхнього рівня (наприклад, групи верхнього одягу, взуття, аксесуари), оскільки категорії можуть надалі додаватися й глибина декомпозиції може збільшуватися. Ця процедура попереднього угрупування необхідна для зменшення часу побудови кластерів у пам'яті. Що менше товарів використовується в групі товарів, то швидше можна побудувати кластери та розрахувати відстані між ними.

2. На етапі ETL-фідів товарів формуються текстові теги на основі текстового опису товару. Для зручності кластеризації пропонується використовувати попередньо отримані теги (текстові атрибути) товарів, а не повний текст опису товарів. На основі набору тегів товару формуємо вектор певної довжини (наприклад 512-Flat). Для розрахунку вектора пропонується використати навчену модель нейронної мережі з архітектурою ResNet-18. Робота з нейронною мережею може бути здійснена на базі фреймворку TensorFlow.

3. На основі Product Quantization вектори розмірності (наприклад 512) розбиваються на  $n$ -частин.

4. Необхідно завантажити для всіх товарів вектори індексів (IVF-індекси). Для оптимізації необхідно завантажити вектор лише певної категорії товарів.

5. Використовуємо алгоритм кластеризації faiss.Clustering. Простір векторів розбивається методом  $k$ -means на  $k$  кластерів. Налаштовуємо кількість кластерів (наприклад, квадратний корінь із кількості товарів у категорії).

6. Створюємо Inverted Lists IVF простору векторів на диску. Навчаємо IVF-індекс із параметрами максимально можливого обсягу даних, який може бути завантажений у пам'ять сервера.

7. У навчений індекс частинами додаємо вектори, що були в навчанні, записуємо на диск IVF-індекс для кожної з частин (зберігаємо їх у БД).

8. Етап експлуатації для векторів на основі використання текстового опису такої:

У запиті (API) надсилаємо (id) товару та ідентифікуємо його вектор (у пам'яті простору IVF). Для пошуку схожих товарів використовуємо отриманий IVF-індекс і визначаємо найближчих сусідів (на відстані) для обраного товару. Ранжуємо знайдених найближчих сусідів (не більше ніж 10) за їхньою віддаленістю від нашого товару.

9. Аналогічно виконуємо процедуру кластеризації для векторів, які отримуємо на основі перетворення зображення на вектор. Етапи 3–7 аналогічні. Результатом є  $m$  найближчих сусідів для обраного вектора зображення товару.

10. Зіставляємо результати кластеризації (товари – найближчі сусіди). Вибираємо результати, які мають перетин товарів як для векторів за текстовими ознаками, так і для векторів за зображеннями (це необхідно для зменшення ймовірності помилки. За наявності великого набору товарів та постачальників можливо для кожного способу пошуку найближчих сусідів встановлювати вагу, наприклад, більшу вагу встановлюємо результатам за векторами зображення, меншу вагу – за текстовим описом).

11. Надаємо отриманий набір схожих товарів (id) як відповідь на запит пошуку (за допомогою API).

Фрагмент архітектури рішення щодо розв'язання завдання кластеризації наведено на рис. 1.

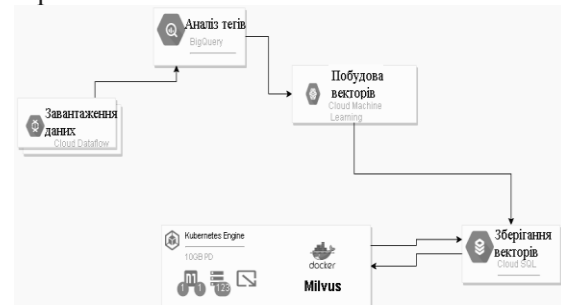


Рис. 1. Фрагмент архітектури рішення

Загальна архітектура базується на використанні сервісів GCP. Платформа Google Cloud (GCP) – це сукупність обчислювальних ресурсів Google. Для рішення пропонується такі сервіси:

1. Google Dataflow надає аналітичне рішення для отримання, оброблення й аналізу потоків подій в інфраструктурі GCP та є альтернативою пакетному ETL (Execute-Transform-Load). Потокова аналітика в GCP спрощує конвеєри ETL за допомогою API Java або Python. В архітектурі використовуємо Google Cloud Dataflow Template Pipelines для трансформації JSON даних товарів (фід) в дані BigQuery.

2. BigQuery – це безсерверне сховище даних, що використовується для зберігання та аналізу наборів даних фідів товарів (після препроцесингу – тегів). BigQuery дозволяє: будувати власні схеми, які організують дані в набори даних та таблиці; завантажувати дані з різних джерел, зокрема потокові дані; використовувати запити SQL; використовувати WEB-інтерфейс або API; здійснювати завантаження, експорт та копіювання даних за допомогою завдань.

3. Платформа GCP AI використовується для машинного навчання (ML) та побудови векторів зображень та тегів. Існує можливість використовувати API, які надають змогу застосовувати заздалегідь підготовлені моделі DNN. Також можна створити й навчити власні масштабні, складні моделі, використовуючи TensorFlow. Реалізацію завдань embedding та побудови векторів пропонується реалізувати в GCP AI на основі ResNet-18 у фреймворку TensorFlow.

4. Cloud SQL – служба реляційних баз даних для MySQL, PostgreSQL та SQL Server. Для інтеграції з рішенням MILVUS пропонується використовувати PostgreSQL для зберігання даних щодо векторів.

5. Kubernetes Engine (GKE) – це середовище для розгортання контейнерних застосунків. GKE забезпечує швидке розроблення та ітерацію застосунків, полегшуючи розгортання, оновлення та управління програмами та послугами. Підтримка апаратних прискорювачів дозволяє запустити машинне навчання, GPU загального призначення, високо-ефективні обчислення та інші робочі навантаження [9]. Реалізацію завдання кластеризації (FAISS) можливо здійснити на основі вже розробленого докеру проєкту MILVUS. Масштабування рішення здійснюється за рахунок налаштування GKE.

5. Хмарні функції GCP (FaaS) забезпечують середовище виконання без сервера для побудови та підключення інших хмарних сервісів. Хмарні функції в роботі використовуються для: попереднього оброблення да-

них фідів товарів та завантаження зображень у Cloud Storage; відповіді на тригери HTTP; роботи з API; реалізації WEB-бекенда.

### Висновки

Сучасні рішення AI можуть бути максимально практичні для e-commerce, вони допомагають утримувати або повертати клієнтів, рекомендувати товари або визначати цільову аудиторію тощо. Унаслідок дослідження було запропоновано алгоритм щодо реалізації рішення на базі GCP, розроблено, протестовано ML-компоненти задачі пошуку аналогічних товарів на основі використання алгоритмів FAISS та Deep Learning в GCP. Але залишилось багато практичних питань щодо оптимізації рішення, наприклад, вибір моделей векторизації даних щодо товару або препроцесингу фідів, які суттєво впливають на швидкість, точність та оперативність рішення.

### Література

1. 2021: A Year Full of Amazing AI papers. URL: [https://github.com/louisfb01/best\\_AI\\_papers\\_2021](https://github.com/louisfb01/best_AI_papers_2021)
2. Marketplace for AI Models. URL: <https://arxiv.org/pdf/2003.01593.pdf>.
3. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. URL: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/frp1159-songA.pdf>.
4. GAN-based-Recommender-System. URL: <https://github.com/jihoo-kim/GAN-based-Recommender-System>.
5. Image classification on fashion-MNIST. URL: <https://github.com/cmasch/zalando-fashion-mnist>
6. Benchmark Fashion MNIST. URL: <http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/#>
7. FAISS. URL: <https://github.com/facebookresearch/faiss>
8. MILVUS. URL: <https://github.com/milvus-io/pymilvus/tree/0.2.14>.
9. Product quantization for nearest neighbor search. URL: [http://lear.inrialpes.fr/pubs/2011/JDS11/jegou\\_searching\\_with\\_quantization.pdf](http://lear.inrialpes.fr/pubs/2011/JDS11/jegou_searching_with_quantization.pdf).

### References

1. 2021: A Year Full of Amazing AI papers. URL: [https://github.com/louisfb01/best\\_AI\\_papers\\_2021](https://github.com/louisfb01/best_AI_papers_2021)
2. Marketplace for AI Models. URL: <https://arxiv.org/pdf/2003.01593.pdf>.
3. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. URL: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/frp1159>

- songA.pdf.
4. GAN-based-Recommender-System. URL: <https://github.com/jihoo-kim/GAN-based-Recommender-System>.
  5. Image classification on fashion-MNIST. URL: <https://github.com/cmasch/zalando-fashion-mnist>.
  6. Benchmark Fashion MNIST. URL: <http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/#>
  7. FAISS. URL: <https://github.com/facebookresearch/faiss>.
  8. MILVUS. URL: <https://github.com/milvus-io/pymilvus/tree/0.2.14>.
  9. Product quantization for nearest neighbor search. URL: [http://lear.inrialpes.fr/pubs/2011/JDS11/jegou\\_searching\\_with\\_quantization.pdf](http://lear.inrialpes.fr/pubs/2011/JDS11/jegou_searching_with_quantization.pdf).

**Знахур Сергій Вікторович**, к.е.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 050-30-04-390, [serhii.znakhur@gmail.com](mailto:serhii.znakhur@gmail.com),  
**Знахур Людмила Володимирівна**, ст. викл. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 050-65-11-282, [razinalv@gmail.com](mailto:razinalv@gmail.com).

#### Similar goods search based on FAISS

**Abstract.** *The problem of obtaining a complete set of similar goods from different manufacturers based on the image and description of the product is relevant and interesting. The article includes description of algo and architecture solution for online goods clustering. Goal.* *The goal of the work is to create*

*and explore models of e-commerce ML (online goods clustering). The work includes developing AI components based on FAISS and Deep Learning to implement Product Quantization of goods searches based on their embedding vectors and vectors of images. The proposed approach to the search for similar goods uses the DNN for features detection and embedding vectors, which are used for FAISS clustering. Methodology.* *The analytical and empirical methods of research based on the development and DNN are used, ML methods to determine features of goods and solve classification problems. Results.* *The architecture of the solution is based on the use of GCP services. Practical value.* *AI solutions have practical value for e-commerce, they help retain or return customers, and recommend products. As a result of the work, a basic solution to the problem of matching similar products based on the use of FAISS and Deep Learning algorithms in GCP was developed and tested.*

**Key words:** *GCP, ML, Deep Learning, FAISS, images, tags, clustering.*

**Znakhur Serhii**, Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 050-300-43-90, [serhii.znakhur@gmail.com](mailto:serhii.znakhur@gmail.com),

**Znakhur Liudmyla**, Lecturer of Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 050-65-11-282, [razinalv@gmail.com](mailto:razinalv@gmail.com).



УДК 004.031.4

DOI: 10.30977/BUL.2219-5548.2022.96.0.50

## ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ХМАРНОЇ ІТ-ІНФРАСТРУКТУРИ У БАНКІВСЬКІЙ СФЕРІ

Золотарьова І. О., Беседовський О. М.

Харківський національний економічний університет ім. С. Кузнеця

*Анотація.* Розглянуті можливості використання інфраструктурних сервісів IaaS, PaaS, SaaS, VaaS хмарних технологій для вдосконалення роботи банківських систем. Проаналізовано функції захисту платежів, модуля-антиспаму, середовища для обміну даними.

*Ключові слова:* хмарні технології, інфраструктурні сервіси, банківські системи.

### Вступ

Використання хмарних технологій у банківській діяльності істотно розширюють їхні можливості, оскільки це зручно й економічно, конкурентноспроможно й актуально, інноваційно й безпечно. За даними досліджень, у банків за умови використання хмар на 10–20 % скорочуються операційні витрати, на 30–50 % покращується показник часу виведення на ринок нових рішень, на 40–50 % прискорюється виділення віртуальних ресурсів для вирішення поточних завдань. Утім, лише 35 % банків повністю задоволені наявними можливостями хмарної ІТ-інфраструктури й бажають її адаптації [1].

### Аналіз публікацій

Особливості використання банками хмарних технологій у загальних рисах відомі [2]. На сьогодні в міжнародній практиці існує успішний досвід перенесення банками в хмарну інфраструктуру критичних бізнес-функцій, таких як обліково-операційна діяльність, управління ризиками та інформаційна безпека. Водночас зазвичай використовується публічна хмарна інфраструктура від одного з технологічних гігантів, таких як Amazon, Microsoft або Google. Так, найбільший цифровий банк *Capital One* (США) 2018 р. відмовився від 5 з 8 власних дата-центрів на користь хмарної інфраструктури в форматі PaaS [3]. Цифровий банк *Bunq* (Нідерланди) повністю мігрував свої ІТ-системи на IaaS від Amazon Web Services [4].

Інколи хмарні банківські ІТ-системи створюються під конкретний банк. Так, DBS Bank відкрив у Індії банк *DigiBank* – мобільний банк у публічній хмарі, що використовує обліково-операційну систему материнського банку, розміщену на приватній інфраструктурі [5]. *Atom Bank* (Великобританія) – мобільний банк без відділень і вебсайту, розвивається на хмарній інтеграційній платформі від постачальника *MuleSoft* (США), з викорис-

танням хмарної інтеграційної платформи *iPaaS* [6]. *Oak North Bank* (Великобританія) надає кредитні та депозитні продукти малому й середньому бізнесу в режимі онлайн, використовуючи хмарну автоматизовану банківську систему компанії *Mambu* (Німеччина) [7].

### Мета та постановка завдання

Дослідити можливості застосування різноманітних сервісів хмарної ІТ-структури для вдосконалення банківської інформаційної системи. Виявити переваги й ризики їхнього використання.

### Виклад основного матеріалу

Хмарні технології – це модель забезпечення зручного мережного доступу до ресурсів, які конфігуруються й можуть бути оперативно надані, масштабовані та звільнені з мінімальними експлуатаційними витратами і зверненнями до постачальника. Хмарні сервіси поділяються на кілька моделей надання послуг – від базових інфраструктурних сервісів до комплексу готових бізнес-функцій, наприклад сервісів обліково-операційної діяльності банків.

Інфраструктура як послуга (англ. Infrastructure as a service, IaaS) – це модель обслуговування, у межах якої споживачу надається можливість керувати засобами оброблення та збереження, комунікаційними мережами, та іншими фундаментальними обчислювальними ресурсами, на базі яких споживач може розгортати й виконувати довільне програмне забезпечення, до складу якого можуть входити операційні системи та прикладні програми [8].

Платформа як послуга (англ. Platform as a service, PaaS) – модель надання хмарних обчислень, за умови якої споживач отримує доступ до використання інформаційно-технологічних платформ: операційних систем, систем управління базами даних,

зв'язного програмного забезпечення, засобів розроблення й тестування розміщених у хмарних провайдерах [8].

Програмне забезпечення як послуга (англ. Software as a Service, SaaS) – у цій моделі клієнт отримує вже готову функціональність у застосунку. У цьому разі розвиток і супровід програми залишається в зоні відповідальності постачальника послуги SaaS, наприклад, клієнт може купити підписку на хмарний CRM чи систему автоматизації банківського обслуговування [8].

Таблиця 1 – Порівняння можливостей IaaS, PaaS, SaaS

Можливість	IaaS	PaaS	SaaS
Закупівля та підтримка обладнання	+	+	+
Віртуалізація	+	+	+
Адміністрування на фізичному та мережному рівні	+	+	+
Налаштування на рівні операційної системи		+	+
Бази даних		+	+
Програмне забезпечення		+	+
Наповнення сайту			+

Бізнес як сервіс BaaS (англ. Bank / Business as a Service, BaaS) є принципово новим рівнем застосування хмарних технологій. Клієнту надаються не технологічні можливості, а готовий автоматизований бізнес-процес за моделлю підписки, яка дозволяє гнучко управляти обсягом робіт, переданих на аутсорсинг. Наприклад, якщо в моделі SaaS споживач замовляє хмарну автоматизовану банківську систему, в якій буде працювати й вибудовувати бізнес-процеси самостійно, то в BaaS він замовляє готові обліково-операційні сервіси, що не потребують витрат на їхню організацію [9].

Розглянемо можливі сценарії надання послуг платіжних систем із використанням хмарних технологій для кредитно-фінансового ринку.

Інформаційна безпека як послуга. Традиційно кожний комерційний банк упроваджував політику інформаційної безпеки на власній інфраструктурі. Перехід від приватних рішень до хмарних дозволяє впроваджувати високоякісні й сучасні системи інформаційної безпеки в разі сукупного зниження вартості володіння для організації за рахунок відмови від установки й супроводу програмного й апаратного забезпечення, а також скорочення витрат на персонал, що обслуговує вказане обладнання. Найбільш поширені

хмарні безпекові рішення: антивірусні сервіси; сервіси захисту від спаму; сервіси захисту зберігання інформації; сервіси захисту від шахрайства і DDoS-атак, послуги для зберігання та використання закритих ключів електронного підпису [10].

Функція захисту платежів присутня в розширених версіях платних антивірусів і спрямована на запобігання крадіжки грошей під час онлайн-оплати. Хмарні провайдери не здійснюють платіжні послуги, а лише можуть надати ресурси в оренду організації, що здійснює оброблення платіжних карт. Отже, провайдери не обробляють і не зберігають дані про власників карт (CHD) або конфіденційні дані аутентифікації (SAD) під час транзакцій. Утім, багато клієнтів великого хмарного провайдера (банки або платіжні системи) надають платіжні послуги населенню або бізнесу. У цьому випадку обов'язки щодо захисту платежів поділяються між провайдером і клієнтом, але і провайдер, і клієнт у цьому випадку мають відповідати вимогам PCI DSS у тому обсязі, в якому це визначено договором між цими сторонами. Отже, спільними зусиллями і провайдера, і клієнта досягається відповідність усім вимогам PCI DSS, і цей тягар не лягає лише на одного клієнта [11, 12].

Велике безпекове значення належить *антиспам*-модулю, що фільтрує поштовий трафік, відправляючи підозрілі листи в окрему папку, в якій під час відкриття листів блокуються мультимедійні дані для запобігання проникнення черв'яків [13].

Важливим елементом хмарної інфраструктури для фінансового сектора є середовище для обміну даними на основі відкритих API, що забезпечує інтеграцію між хмарними компонентами постачальника інфраструктури та приватними компонентами одного з клієнтів-учасників. Наприклад, хмарна банківська система й хмарний шлюз до платіжної системи можуть бути інтегровані з приватної реалізацією мобільного банку-клієнта. В іншому випадку кредитна організація може інтегрувати свій процес розгляду кредитних заявок із сервісами, що надає хмарний постачальник KYC-аналітики або альтернативного кредитного скорингу. Водночас у споживача послуги має бути змога паралельно використовувати сервіси декількох постачальників з метою визначення найбільш відповідного для вирішення конкретних завдань. Отже, «хмара» є форматом надання послуги обміну даними, а відкриті API – способом публікації та використання цієї послуги [14].

Постачальник хмарної платформи може надавати шлюзи до платіжних систем за моделлю підписки аналогічно оренді обліково-операційних послуг, створюючи безпечну й масштабовану інтеграцію з платіжними системами. Зі свого боку організація-клієнт, розмістивши свій застосунок усередині контуру хмари, зможе отримати спрощену схему інтеграції з платіжними системами, істотно скоротивши витрати на інфраструктуру, необхідну для підключення до них. Необхідно використовувати єдиний платіжний хаб у хмарі, який з'єднує всі дочірні компанії та забезпечує цілісну центральну платформу для підключення до локальних ERP-систем, забезпечує єдиний інтерфейс на всі платіжні транзакції, конвертує в необхідний формат платежів, підписує з допомогою надійної криптографії та підключається до всіх банків. Від власника хмари клієнт отримує такі послуги, як упровадження платформ в їхню систему, а також постійна підтримка протягом всього періоду співпраці [15].

### Висновки

Проведені дослідження показали, що, незважаючи на певні побоювання з приводу забезпечення інформаційної безпеки, альтернативи використання хмарних технологій у банків немає. Тут можливі різні варіанти: використання приватної хмари, повноцінна міграція в публічну хмару; розміщення в хмарі всієї інфраструктури банку або лише частини його сервісів.

### Література

- Accenture: банки, использующие облака, растут в два раза быстрее конкурентов. URL: <https://servernews.ru/1032570> (дата звернення: 20.12.2021).
- Як банк модернізував застарілі IT-системи та мігрував у «хмару». URL: <https://dou.ua/lenta/columns/modernization-of-old-it-systems/> (дата звернення: 20.12.2021).
- Capital One Credit Cards, Bank, and Loans. URL: <https://www.capitalone.com/> (дата звернення: 30.12.2021).
- Bunq Bank of The Free. URL: <https://www.bunq.com/home> (дата звернення: 30.12.2021).
- DBS Bank Singapore. URL: <https://www.dbs.com.sg/index/default.page> (дата звернення: 30.12.2021).
- Atom – The app-based bank that makes money simple. URL: <https://www.atombank.co.uk/> (дата звернення: 24.12.2021).
- Oak North Bank. URL: <https://www.oaknorth.co.uk/> (дата звернення: 24.12.2021).
- Deploying the company's IT infrastructure in the cloud: SaaS, PaaS, IaaS. URL: <https://www.it.ua/knowledge-base/architecture-security/cloud-infrastructure-saas-paas-iaas> (дата звернення: 24.12.2021).
- How the banking-as-a-service industry works and BaaS market outlook For 2021. URL: <https://www.insiderintelligence.com/insights/banking-as-a-service-industry/> (дата звернення: 24.12.2021).
- Нікішин Д., Федюшин О. Ризики інформаційної безпеки в хмарних сервісах. URL: <https://openarchive.nure.ua/bitstream/document/10550/1/NIKISHYN.pdf> (дата звернення: 24.12.2021).
- PCI DSS Compliant Tokenization Vault. URL: [https://pcivault.io/?gclid=CjwKCAiAz--OBhBIEiwAG1rIOosw3h8rCMf19CRvBBosGtDoNwaL0BpUAySPyWCRvnVP-5ol2cOBnxoCANKQAvD\\_BwE](https://pcivault.io/?gclid=CjwKCAiAz--OBhBIEiwAG1rIOosw3h8rCMf19CRvBBosGtDoNwaL0BpUAySPyWCRvnVP-5ol2cOBnxoCANKQAvD_BwE) (дата звернення: 24.12.2021).
- Що таке PCI DSS і чому це потрібно кожній комерційній компанії. URL: <https://www.myspecialist.com/uk/what-is-pci-dss-ua/> (дата звернення: 24.12.2021).
- Облачные антивирусы – обзор на LiveBusiness URL: [https://www.securno.ru/tags/oblachnye\\_antivirus/](https://www.securno.ru/tags/oblachnye_antivirus/) (дата звернення: 24.12.2021).
- Соммер Р. Сравнение облачных хранилищ – самые популярные облака. URL: <https://www.boxcryptor.com/ru/blog/post/list-best-clouds-private-use/> (дата звернення: 24.12.2021).
- Alevate Payments – Bobsguide. URL: <https://www.bobsguide.com/products/alevate-payments/> (дата звернення: 24.12.2021).

### References

- Accenture: банки, использующие облака, растут в два раза быстрее конкурентов. *Accenture: banki, ispol'zuyushchiye oblaka, rastut v dva raza bystreye konkurentov.* (Accenture: Cloud-based banks are growing twice as fast as their competitors). URL: <https://servernews.ru/1032570> (accessed: 24 December 2021).
- Як банк модернізував застарілі IT-системи та мігрував у «хмару». *Yak bank modernizuvav zastarili IT-systemy ta mihruvav u «khamaru»* (How the bank upgraded outdated IT systems and migrated to the cloud). URL: <https://dou.ua/lenta/columns/modernization-of-old-it-systems/> (accessed: 24 December 2021).
- Capital One Credit Cards, Bank, and Loans. URL: <https://www.capitalone.com/> (accessed: 24 December 2021).
- Bunq Bank of The Free. URL: <https://www.bunq.com/home> (accessed: 24 December 2021).
- DBS Bank Singapore. URL: <https://www.dbs.com.sg/index/default.page> (accessed: 24 December 2021).

6. Atom – The app-based bank that makes money simple. URL: <https://www.atombank.co.uk/> (accessed: 24 December 2021).
7. Oak North Bank. URL: <https://www.oaknorth.co.uk/> (accessed: 24 December 2021).
8. Deploying the company's IT infrastructure in the cloud: SaaS, PaaS, IaaS. URL: <https://www.it.ua/knowledge-base/architecture-security/cloud-infrastructure-saas-paas-iaas> (accessed: 24 December 2021).
9. How the banking-as-a-service industry works and BaaS market outlook For 2021. URL: <https://www.insiderintelligence.com/insights/banking-as-a-service-industry/> (accessed: 24 December 2021).
10. Нікішин Д., Федюшин О. Ризики інформаційної безпеки в хмарних сервісах. *Nikishyn D., Fedyushyn O. Ryzkyk informatyynoyi bezpeky v khmarnykh servisakh.* (Nikishyn D., Fedyushyn O. Risks of information security in cloud services). URL: <https://openarchive.nure.ua/bitstream/document/10550/1/NIKISHYN.pdf> (accessed: 24 December 2021).
11. PCI DSS Compliant Tokenization Vault. URL: [https://pcivault.io/?gclid=CjwKCAiAz--OB-hBIEiwAG1rIOosw3h8rCMf19CRvBBosGtDoNwaL0BpUAYsPyWCRvnVP-5ol2cOBnxoCAHkQAvD\\_BwE](https://pcivault.io/?gclid=CjwKCAiAz--OB-hBIEiwAG1rIOosw3h8rCMf19CRvBBosGtDoNwaL0BpUAYsPyWCRvnVP-5ol2cOBnxoCAHkQAvD_BwE) (accessed: 24 December 2021).
12. Що таке PCI DSS і чому це потрібно кожній комерційній компанії. *Shcho take PCI DSS i chomu tse potribno kozhniy komertsyyniy kompaniyi* (What is PCI DSS and why every commercial company needs it). URL: <https://www.my-itspecialist.com/uk/what-is-pci-dss-ua/> (accessed: 25 December 2021).
13. Облачные антивирусы – обзор на LiveBusiness. *Oblachnyye antivirusy – obzor na LiveBusiness* (Cloud Antivirus – LiveBusiness Review). URL: [https://www.securno.ru/tags/oblachnyye\\_antivirus\\_y/](https://www.securno.ru/tags/oblachnyye_antivirus_y/) (accessed: 24 December 2021).
14. Sommer P. Сравнение облачных хранилищ – самые популярные облака. *Sommer R. Sravneniye oblachnykh khranilishch – samyye populyarnyye oblaka.* (Sommer R. Comparison of cloud storage - the most popular clouds). URL: <https://www.boxcryptor.com/ru/blog/post/list-best-clouds-private-use/> (accessed: 25 December 2021).
15. Alevate Payments – Bobsguide. URL: <https://www.bobsguide.com/products/alevate-payments/> (accessed: 24 December 2021).

**Золотарьова Ірина Олександрівна**, к.е.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, [iryna.zolotaryova@hneu.net](mailto:iryna.zolotaryova@hneu.net), тел. +380677596158,

**Беседовський Олексій Миколайович**, к.е.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, [balex78@gmail.com](mailto:balex78@gmail.com), тел. +380501403585.

**Application of cloud IT infrastructure in banking**  
**Abstract. Problem.** *Competition forces banking organizations to test technological innovations in order to keep up with more progressive start-ups and be able to compete for customers in the market. At the same time, the cost of cloud technology is lower than in-house infrastructure, especially when you consider constant expansions, costs for upgrades, software and innovations, and maintenance of in-house servers. Security is another shift from in-house servers to cloud services. The architecture allows load balancing of servers, thus controlling resource consumption. This structure also facilitates financial-technical changes or upgrades for its own development, without affecting other system elements. It can be very advantageous for large companies with many projects. Additional insurance can be the use of several cloud services to distribute the load on servers. The use of cloud technology in banking significantly extends their capabilities, because it is convenient and cost-effective, competitive and relevant, innovative and secure. However, only 35% of banks are fully satisfied with the existing capabilities of cloud IT infrastructure and wish to adapt it. Goal.* *The goal is to explore the possibilities of applying various cloud IT structure services to improve the banking information system, identify advantages and risks of their use. Methodology.* *The introduction of cloud services is in demand among banks focused on innovative technology and continuous development. For competitive banks, the cloud has become part of the plan for technological development. Results.* *The studies have shown that banks have no alternative to using cloud technology. Each bank can use different variants of working with cloud technologies: using a private cloud, full migration to a public cloud, placing the entire infrastructure of the bank or only part of its services in the cloud. Particular attention should be paid to information security issues. Originality.* *The problematic areas of banks in the cloud IT infrastructure are identified and the recommendations for improvement are provided. Practical value.* *The developed recommendations can be applied by banks when using the cloud IT infrastructure.*

**Key words:** *cloud technology, infrastructure services, banking systems.*

**Zolotaryova Iryna**, PhD, professor, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, phone +380677596158, [iryna.zolotaryova@hneu.net](mailto:iryna.zolotaryova@hneu.net),  
**Besedovskyi Oleksii**, PhD, ass. professor, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, phone +380501403585, [balex78@gmail.com](mailto:balex78@gmail.com).

УДК [004.5+004.89]:37.02

DOI: 10.30977/BUL.2219-5548.2022.96.0.54

## ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ НАВЧАЛЬНОЮ ДІЯЛЬНІСТЮ ЯК ЗАСІБ ОПАНУВАННЯ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ СЕРВІСІВ ІНТЕРНЕТ

Колгатін О. Г., Парфьонов Ю. Е.

Харківський національний економічний університет ім. С. Кузнеця

***Анотація.** У дослідженні побудовано модель реалізації технології навчання програмування сервісів Інтернет. Модель передбачає вибір способу опрацювання навчального матеріалу згідно з наявними загальними компетентностями, знаннями, уміннями та очікуваним розвитком студента в процесі опанування освітньої дисципліни. Варіанти реалізації технології навчання передбачають застосування методу проєктів та використання інтерактивних елементів системи Moodle, таких як Lesson (Урок), Workshop (Семинар), Quiz (Тест).*

***Ключові слова:** інтернет, програмування, навчання, педагогічна діагностика, інформаційна система, Moodle.*

### Вступ

Сучасний навчальний процес усе більше орієнтується на самостійну роботу студентів, що підтримується потужними інформаційними системами управління навчальною діяльністю. Однією з таких відомих систем є LMS Moodle, що активно використовується в Харківському національному економічному університеті імені С. Кузнеця. Вагомою перевагою потужної інформаційно-комунікаційної підтримки освітнього процесу є можливість індивідуального проєктування навчальної діяльності кожного студента, застосування доцільних варіантів технології навчання. Основною вимогою адаптивної організації навчального процесу та, відповідно, управління самостійною роботою студентів є наявність автоматизованої системи педагогічної діагностики та педагогічного прогнозування, яка будується на розвиненій базі даних і елементах штучного інтелекту щодо прийняття рішень [1]. Так, О. Г. Колгатін [1] пропонує покласти в основу бази даних модель цілей навчання, психолого-педагогічну модель студента й модель реалізації технології навчання. Модель цілей навчання будується як система діагностичних завдань, що повністю покривають матеріал навчальної дисципліни. Психолого-педагогічна модель будується на підґрунті моделі цілей навчання, відтворює ступінь наближення студента до поставленої мети в динаміці згідно з критеріями, що можуть бути об'єктивно виміряні в автоматизованій системі педагогічної діагностики: мотиваційно-цільовий, навчально-змістовий, організаційно-діяльнісний, рефлексивно-прогностичний [1]. Модель реалізації техно-

логії навчання забезпечує обґрунтований вибір методів і прийомів навчання певного матеріалу для конкретного студента [1]. Саме ця модель відбиває методику навчання, тому її побудова є важливим елементом методичного забезпечення дисципліни.

Отже, застосування інформаційної системи управління освітньою діяльністю для навчання технологій програмування сервісів Інтернет передбачає побудову адаптивної системи методів навчання, що спираються на використання компонентів LMS Moodle. Ця робота спрямована на навчання іноземних студентів за освітньою програмою «Інженерія програмного забезпечення» освітнього рівня «бакалавр». Як базову мову програмування під час вивчення навчальної дисципліни обрано Python. Курс побудовано згідно з робочою програмою навчальної дисципліни, розробленою Ю. Е. Парфьоновим [2].

### Аналіз публікацій

Існує значна кількість якісних інструктивних матеріалів із використання мови програмування Python та фреймворку Django для програмування сервісної частини інтернет-орієнтованих інформаційних систем. Насамперед це офіційні сайти підтримки Python [3] та Django [4], де наведено докладний опис мови та функцій. Звернемо увагу на підходи до застосування інформаційних систем управління освітньою діяльністю в межах навчання програмування Інтернет з точки зору побудови адаптивних систем навчального призначення. Проблеми впровадження мережних технологій відкритих систем у навчальний процес присвячено чимало досліджень. Так, щодо дисципліни «Організаційна

інформатика» в роботі Т. Я. Вдовичин [5] обґрунтовано основні компоненти методики використання мережних технологій відкритих систем у навчанні майбутніх бакалаврів інформатики. Значну увагу приділяють дослідники проблемі адаптивного навчання в інформаційно-комунікаційному педагогічному середовищі. Як параметр управління традиційно використовують рівень знань [6] та застосовують алгоритми, що ґрунтуються на використанні дерева прийняття рішень [6]. Інший підхід передбачає врахування стилю навчання як критерію вибору варіанта реалізації технології навчання. Такий підхід використовувався для підтримки вивчення об'єктно орієнтованого програмування мовою Java для студентів інженерних спеціальностей [7]. Студенти в цьому дослідженні були диференційовані за допомогою профільного тесту, щоб запропонувати їм одну з трьох альтернативних навчальних пропозицій. У праці [8] запропоновано огляд та систематизацію досліджень із зазначеного напрямку. Окремо варто звернути увагу на дослідження, в яких пропонується розвинений математичний апарат для прийняття рішень щодо ефективності тих чи інших варіантів технології навчання. Із цього приводу цікавою є праця [9], в якій для оцінювання ефективності електронного курсу програмування застосовано експертну систему на основі нечіткої логіки.

#### Мета та постановка завдання

За результатами проведеного аналізу можна стверджувати, що, незважаючи на значний обсяг і глибину відомих досліджень, проблема побудови моделей реалізації технології навчання для конкретних освітніх дисциплін залишається актуальною.

Метою цієї роботи є розроблення варіантів навчання програмування Інтернет-сервісів, що можуть бути реалізовані в інформаційно-комунікаційному педагогічному середовищі на базі інформаційної системи управління навчальною діяльністю Moodle.

#### Виклад основного матеріалу

Розглянемо варіанти реалізації технології навчання на прикладі дисципліни «Програмування Інтернет» за освітньою програмою бакалавра «Інженерія програмного забезпечення» [2]. Метою навчальної дисципліни є формування компетентності щодо використання сучасних засобів програмування Інтернет та технологій розроблення серверної частини

вебзастосунків мовою Python. У процесі навчання студенти повторюють основні прийоми використання мови Python та опановують спеціальні засоби Django. Передбачається вивчення таких питань: розподілені програмні системи; підтримка інтернет-протоколів та суміжних технологій у Python; основні функції модуля socket; WSGI та ASGI використання протоколів SMTP, POP3, IMAP тощо; основи Web Scraping; об'єктно-реляційне відображення даних, моделі, міграції бази даних; розроблення видів, зіставлення URL та видів, основи мови шаблонів; процес опрацювання форм у Django; сесії, система авторизації та автентифікації у Django; інтернаціоналізація та локалізація; розгортання вебзастосунку на хостингу [2]. Навчальний процес має практико-орієнтований характер.

Аналіз способів опрацювання освітнього матеріалу проведемо відповідно до структури моделі реалізації технології навчання. Ця модель передбачає поєднання формалізованих вимог до студента щодо його наявних загальних компетентностей та результатів навчання (табл. 1) з вимогами до очікуваного розвитку студента (табл. 2) та моделі цілей навчання.

Основою навчальної діяльності студентів є проектування певних компонентів інформаційної системи із застосуванням технологій та прийомів програмування, що вивчаються. Такий спосіб опрацювання навчального матеріалу забезпечує високий рівень освітніх досягнень та розвиток студента, проте висуває певні вимоги до фахових і загальних компетентностей студента, а саме: високий рівень рефлексії як до результату, так і до процесу діяльності; зацікавленість процесом навчальної діяльності; пізнавальний інтерес; стабільність темпу навчальної праці та усвідомленість освітньої дисципліни, зокрема терміну виконання етапів роботи. Знижувати успіх може висока значущість результату навчальної діяльності (склав / не склав) на фоні низької теоретичної та практичної підготовленості студента. Саме тому цей основний спосіб опрацювання навчального матеріалу має бути доповнений іншими варіантами, що забезпечать необхідний початковий рівень освітніх досягнень та розвиток студента щодо вміння оцінювати якість результату діяльності, аналізувати процес діяльності й планувати власні дії, сприятимуть формуванню пізнавального інтересу та стабільності темпу навчальної праці.

Таблиця 1 – Вимоги до наявних компетентностей студента та результатів навчання

Параметр	Спосіб опрацювання навчального матеріалу		
	метод проєктів	програмоване навчання	взаємне рецензування
Рефлексія щодо результату діяльності	високі вимоги	-	помірні вимоги
Рефлексія щодо процесу діяльності	високі вимоги	-	-
Значущість результату навчальної діяльності	помірні вимоги	високі вимоги	високі вимоги
Зацікавленість процесом навчальної діяльності	високі вимоги	-	помірні вимоги
Пізнавальний інтерес	високі вимоги	-	помірні вимоги
Усвідомленість навчальної дисципліни	високі вимоги	помірні вимоги	високі вимоги
Стабільність темпу навчальної праці	помірні вимоги	-	високі вимоги
Здатність до мобілізації енергії, наполегливості й волі	помірні вимоги	-	високі вимоги
Рівень навчальних досягнень	достатній або високий	середній	достатній або високий
Навички предметної діяльності	високі вимоги	-	помірні вимоги
Повнота знань	помірні вимоги	-	високі вимоги
Глибина знань	помірні вимоги	-	високі вимоги
Гнучкість знань	помірні вимоги	-	високі вимоги
Оперативність знань	високі вимоги	-	високі вимоги
Систематичність знань	помірні вимоги	-	високі вимоги

Програмоване навчання є відомим і частково забутим способом опрацювання навчального матеріалу на репродуктивному рівні. Цей спосіб забезпечує середній або достатній (невисокий) рівень освітніх досягнень, проте відсутні спеціальні вимоги до студента, крім значущості результату навчальної діяльності. Тобто студент має подолати не дуже складне й не дуже цікаве навчальне завдання, тому що це потрібно для накопичення певних балів. Для підвищення мотивації у процесі програмованого навчання доцільно використовувати елементи гейміфікації, змагання, рейтингу

тощо. Важливим аспектом також є якість оформлення навчальних матеріалів, застосування елементів мультимедіа, цікавої інформації, що виходить за межі обов'язкового для вивчення матеріалу. В інформаційній системі управління навчальною діяльністю Moodle програмоване навчання забезпечується за допомогою елемента Lesson (Урок) і передбачає завдання в тестовій формі після кожної невеликої порції навчального матеріалу. Розвинена система побудови завдань у тестовій формі дає змогу запропонувати студенту вставити пропущені фрагменти коду або проаналізувати можливий результат виконання певного алгоритму. Програмоване навчання з використанням елементів Lesson (Урок) сприятиме формуванню повноти знань, деяких навичок репродуктивної діяльності, оперативності знань, стабільності темпу навчальної праці.

Таблиця 2 – Очікувані результати застосування варіантів реалізації технології навчання

Параметр	Спосіб опрацювання навчального матеріалу		
	метод проєктів	програмоване навчання	взаємне рецензування
Рівень навчальних досягнень	достатній або високий	середній або достатній	високий
Навички предметної діяльності	так	слабко	частково
Повнота знань	частково	так	частково
Глибина знань	так	слабко	так
Гнучкість знань	так	слабко	так
Оперативність знань	так	так	слабко
Систематичність знань	слабко	так	слабко
Мотивація до досягнення навчальних цілей	так	слабко	так
Уміння оцінювати якість результату діяльності	так	слабко	так
Уміння аналізувати процес діяльності й планувати власні дії	так	слабко	частково
Стабільність темпу навчальної праці	так	так	так
Пізнавальний інтерес	так	слабко	частково

Потужним методом опрацювання навчального матеріалу є взаємне рецензування, для організації якого можна застосувати елемент Workshop (Семинар). Студенти виставляють певні продукти навчальної діяльності на оці-

нювання. У випадку програмування це мають бути незначні за обсягом модулі, оскільки аналіз повноцінного проєкту є дуже складним завданням для студентів і потребує багато часу. Після завершення терміну подання робіт починається етап оцінювання. Така навчальна праця сприяє набуттю освітніх досягнень високого рівня, формуванню здатності до оцінювання, глибини й гнучкості знань. Необхідність синхронізувати власний графік навчальної праці з певними часовими межами подання та оцінювання робіт у групі дисциплінує студента, сприяє формуванню стабільності темпу навчальної праці. Проте застосування такого способу опрацювання навчального матеріалу потребує високої значущості результату освітньої діяльності для студента, здатності до мобілізації енергії, наполегливості й волі та певних навичок планування власної навчальної діяльності. Наші емпіричні дослідження показують, що далеко не всі студенти, особливо іноземні, здатні до такого планування. Більш того, метод працює тільки в разі виконання студентами досить простих завдань, які здатні виконати більшість студентів академічної групи.

Розглянуті підходи до організації самостійної роботи студентів у інформаційній системі управління навчальною діяльністю стають елементом вибору з боку студента за рекомендацією системи педагогічного прогнозування на підставі педагогічної діагностики.

### Висновки

У межах викладання дисципліни «Програмування Інтернет» побудовано модель реалізації технології навчання щодо управління самостійною роботою студентів у системі Moodle. Зазначена модель є основою для прийняття рішень в інформаційній системі педагогічного прогнозування та проєктування на підставі педагогічної діагностики.

Перспективу подальшого дослідження проблеми вбачаємо в експериментальній перевірці параметрів зазначених способів опрацювання навчального матеріалу, поширенні запропонованого підходу на інші навчальні дисципліни.

### Література

1. Колгатін О. Г. Базові моделі в комп'ютерно орієнтованій системі педагогічної діагностики // *Інформаційні технології в освіті*: збірник наукових праць. Херсон, 2012. Вип. 12. С. 14–20.

2. Парфьонов Ю. Е. Програмування Інтернет: робоча програма навчальної дисципліни для студентів спеціальності 121 «Інженерія програмного забезпечення». Харків, 2021. URL: [http://repository.hneu.edu.ua/bitstream/123456789/26544/1/Програмування Інтернет.pdf](http://repository.hneu.edu.ua/bitstream/123456789/26544/1/Програмування%20Інтернет.pdf) (дата звернення: 10.01.2020).
3. Python 3.10.1 documentation. URL: <https://docs.python.org/3/> (дата звернення: 10.01.2020).
4. Django documentation. URL: <https://docs.djangoproject.com/en/3.1/> (дата звернення: 10.01.2020).
5. Вдовичин Т. Я. Використання мережних технологій відкритих систем у навчанні бакалаврів інформатики: загальні висновки. *Інформаційні технології в освіті*: збірник наукових праць. Херсон, 2016. Вип. 2. С. 167–185.
6. AL-Chalabi H. K. M., Hussein A. M. A., Apoki U. C. An Adaptive Learning System Based on Learner's Knowledge Level. *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2021. P. 1–4. URL: <https://ieeexplore.ieee.org/document/9515158> (accessed: 10.01.2022).
7. Díaz F. S., Rubilar T. P., Figueroa C. C., Silva R. M. An Adaptive E-Learning Platform with VARK Learning Styles to Support the Learning of Object Orientation. *2018 IEEE World Engineering Education Conference (EDUNINE)*, 2018. P. 1–6. URL: <https://ieeexplore.ieee.org/abstract/document/8450990> (accessed: 10.01.2022).
8. Khamparia A., Pandey B. Association of learning styles with different e-learning problems: a systematic review and classification // *Educ Inf Technol.* 25, P. 1303–1331 (2020). URL: <https://doi.org/10.1007/s10639-019-10028-y> (accessed: 10.01.2022).
9. Khawar K., Munawar S., Naveed N. Fuzzy Logic-based Expert System for Assessing Programming Course Performance of E-Learning Students. *Journal of Information Communication Technologies and Robotic Applications*. Vol. 11, No. 1. 2020. P. 54–64. URL: <https://www.jictra.com.pk/index.php/jictra/article/view/206/120> (accessed: 10.01.2022).

### References

1. Kolgatin O. H. Bazovi modeli v kompiuterno orientovaniy systemi pedahohichnoi diahnostyky [Basic models in computer-based pedagogical diagnostic system]. *Informatsiini tekhnologii v osviti: zbirnyk naukovykh prats*. Kherson: 2012. Issue 12. P. 14–20 [in Ukrainian].
2. Parfonov Yu. E. Prohramuvannia Internet: robocha prohrama navchalnoi dystsypliny dlia studentiv spetsialnosti 121 "Inzheneriia programnoho zabezpechennia" [Internet programming: working program of the discipline for students majoring in 121 "Software Engineering"]. Kharkiv: 2021. URL:



- [http://repository.hneu.edu.ua/bitstream/123456789/26544/1/Програмування Інтернет.pdf](http://repository.hneu.edu.ua/bitstream/123456789/26544/1/Програмування%20Інтернет.pdf) (accessed: 10.01.2020) [in Ukrainian].
3. Python 3.10.1 documentation. URL: <https://docs.python.org/3/> (accessed: 10.01.2020).
  4. Django documentation. URL: <https://docs.djangoproject.com/en/3.1/> (accessed: 10.01.2020).
  5. Vdovychyn T. Ya. Vykorystannia merezhnykh tekhnolohii vidkrytykh system u navchanni bakalavriv informatyky: zahalni vysnovky [The use of open source network technologies in the teaching of bachelors of computer science: general conclusions]. *Informatsiini tekhnolohii v osviti : zbirnyk naukovykh prats*. Kherson, 2016. Issue. 2. P. 167–185 [in Ukrainian].
  6. AL-Chalabi H. K. M., Hussein A. M. A., Apoki U. C. An Adaptive Learning System Based on Learner's Knowledge Level. *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2021. P. 1–4. URL: <https://ieeexplore.ieee.org/document/9515158> (accessed: 10.01.2022).
  7. Díaz F. S., Rubilar T. P., Figueroa C. C., Silva R. M. An Adaptive E-Learning Platform with VARK Learning Styles to Support the Learning of Object Orientation. *2018 IEEE World Engineering Education Conference (EDUNINE)*, 2018. P. 1–6. URL: <https://ieeexplore.ieee.org/abstract/document/8450990> (accessed: 10.01.2022).
  8. Khamparia A., Pandey B. Association of learning styles with different e-learning problems: a systematic review and classification. *Educ Inf Technol*. 25, P. 1303–1331 (2020). URL: <https://doi.org/10.1007/s10639-019-10028-y> (accessed: 10.01.2022).
  9. Khawar K., Munawar S., Naveed N. Fuzzy Logic-based Expert System for Assessing Programming Course Performance of E-Learning Students. *Journal of Information Communication Technologies and Robotic Applications*. Vol. 11, No. 1. 2020. P. 54–64. URL: <https://www.jictra.com.pk/index.php/jictra/article/view/206/120> (accessed: 10.01.2022).

**Колгатін Олександр Геннадійович**, д.п.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 (057) 702-18-31, [kolgatin@ukr.net](mailto:kolgatin@ukr.net),

**Парфонов Юрій Едуардович**, к.т.н., с.н.с., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 057-702-18-31, [Yurii.Parfonov@m.hneu.edu.ua](mailto:Yurii.Parfonov@m.hneu.edu.ua).

### Learning management system as a means of mastering Internet programming technologies

**Abstract. Problem.** A step towards building an automated system of pedagogical diagnostics and forecasting based on the information system for managing educational activities Moodle has been taken.

**Goal.** The goal of this work is to develop variants for learning the Internet programming, which can be implemented in the information and communication pedagogical environment based on the information system for managing learning activities Moodle. The aim of the discipline is to develop competence in the use of modern Internet programming tools and technologies for developing the server part of web applications in Python. **Methodology.** The method of information-logical modelling was used to design automated pedagogical diagnostics system. The model of training technology realization for Internet programming study according to the educational program of bachelors' instruction in a specialty "Software engineering" was constructed as a part of this system database and cooperates with the model of educational purpose and the psychologic and pedagogic model of student. **Results.** Automated pedagogical diagnostics system helps a student to choose a variant of learning educational material in accordance with the existing competencies of the student, knowledge and skills and the expected development of the student in the process of mastering the discipline. Variants for implementing learning technology include the use of the project method as the main way to develop learning material and use interactive elements of Moodle, such as Lesson for programmed learning, Workshop for inter-reviewing process, Quiz for training and diagnosis at the preparatory stages of independent student work. **Originality.** A model of implementation of learning technology according to study of the discipline "Internet Programming" was built for managing independent work of students in the Moodle system. **Practical value.** The proposed model should become the basis for automated pedagogical forecasting and design of an individual version of the learning technology implementation for each student.

**Key words:** Internet, programming, learning, pedagogical diagnostics, information system, Moodle.

**Kolgatin Oleksandr**, D. Sc., Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38 (057) 702-18-31, [kolgatin@ukr.net](mailto:kolgatin@ukr.net),  
**Parfonov Yurii**, Ph.D., Assoc. Prof., Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38 057-702-18-31, [Yurii.Parfonov@m.hneu.edu.ua](mailto:Yurii.Parfonov@m.hneu.edu.ua).

## ВИБІР СЛУЖБИ ВЕБХОСТИНГУ ДЛЯ ЗАСТОСУНКІВ НА БАЗІ ФРЕЙМВОРКУ DJANGO

Парфьонов Ю. Е., Колгатін О. Г.

Харківський національний економічний університет ім. С. Кузнеця

**Анотація.** Статтю присвячено питанням вибору провайдера хостингу для вебзастосунків на базі фреймворку Django. Розглянуто використання мови Python для розроблення вебзастосунків, особливості створення серверної частини таких застосунків, використання для цього серверного вебфреймворку Django, показники якості функціонування застосунків, що розгорнуті на будь-якому вебхостингу, етапи вибору вебхостингу на прикладі вебзастосунку KnAS4Web. Виконано порівняльний аналіз служб вебхостингу Heroku та PythonAnywhere. Зроблено висновок про доцільність використання служби PythonAnywhere для хостингу вебзастосунку KnAS4Web. **Ключові слова:** вебхостинг, вебфреймворк, Python, Django, Heroku, PythonAnywhere.

### Вступ

Протягом декількох років застосунки мовою Python широко використовуються для автоматизації різноманітних предметних галузей. Насамперед це стосується вебсайтів, зокрема вебзастосунків.

Відомо, що більшість застосунків на базі вебтехнологій є розподіленими програмними системами клієнт-сервер. Через це часто виникає потреба в розгортанні та забезпеченні функціонування серверної частини таких застосунків на певному хостингу в мережі Інтернет.

### Аналіз публікацій

На сьогодні Python є однією з найбільш популярних мов програмування у світі [1–3], зокрема завдяки різноманітності його можливого застосування. Зараз Python широко використовується для машинного навчання, розпізнавання образів, побудови нейронних мереж, а також для розроблення вебзастосунків, комп'ютерних ігор, графічних інтерфейсів користувача, корпоративних застосунків та багато іншого [4].

Використання мови Python для веброзробки обумовлено її простотою, гнучкістю та масштабованістю, можливістю швидкого створення прототипів вебзастосунків, наявністю стандартних бібліотек для вирішення широкого кола проблем та значної спільноти розробників [4–6].

Практично всі серверні вебзастосунки розробляються з використанням того чи іншого вебфреймворку – програмного забезпечення, що надає інструменти та бібліотеки для полегшення розроблення, підтримки та масштабування вебзастосунків. Вони надають інструменти та бібліотеки, що вико-

нують загальні завдання веброзробки, такі як маршрутизація HTTP-запитів, взаємодія з базами даних, підтримка сесій, авторизації та аутентифікації користувачів, форматування вихідних даних та підвищення безпеки від вебатак [7].

Серед серверних вебфреймворків на базі мови програмування Python одним із найкращих часто вважається Django [8, 9]. Цей фреймворк використовується на таких відомих вебсайтах, як Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic, Open Knowledge Foundation, Pinterest, Open Stack та багатьох інших.

Зараз існує досить багато хостингів із підтримкою мови програмування Python та серверного вебфреймворку Django [10]. Отже, вибір хостингу залежить від багатьох чинників і є актуальним.

### Мета та постановка завдання

Метою статті є вибір вебхостингу для забезпечення функціонування та підтримки вебзастосунку.

Для досягнення поставленої мети необхідно проаналізувати переваги й недоліки наявних служб вебхостингу, апаратних та програмних вимог до функціонування вебзастосунку, способів його розгортання та особливостей підтримки протягом життєвого циклу.

### Виклад основного матеріалу

Не існує універсального вебхостингу, тому що кожний вебзастосунок має свої потреби щодо зберігання даних, продуктивності тощо. Отже, спочатку необхідно визна-

чити вимоги до хостингу конкретного застосунку.

Важливі питання щодо цього, які необхідно вирішити насамперед, – це стек серверних технологій, що застосовується службою вебхостингу, та припустимі грошові витрати на використання вебхостингу. Це дозволяє суттєво обмежити коло потенційних служб вебхостингу.

Далі треба розглянути більш конкретні показники [11, 12], щоб у підсумку визначити раціональний варіант хостингу для вебзастосунку. Найбільш важливими з них є такі:

1. Обсяг доступного дискового простору.

Він визначає обсяг вмісту вебзастосунку, який можна зберігати, зокрема йдеться про текст, зображення, програмний код, бази даних тощо.

2. Час безперервної роботи служби вебхостингу.

Це дуже важливий показник щодо забезпечення безперебійної роботи користувачів. Якщо вебзастосунок, такий як вебсайт, недоступний для користувача, він скоріше за все покине його та відвідає інший. Прості служби вебхостингу можуть завдати бізнесу величезних втрат у вигляді незалучених клієнтів, втрачених доходів, зриву важливих заходів тощо.

Але забезпечити безперервну роботу будь-якої комп'ютерної системи неможливо насамперед через технічні причини, коли служба вебхостингу може бути тимчасово недоступною. Зазвичай час безперервної роботи вимірюється у відсотках від загального часу роботи хостингу.

Для критично важливих застосунків час безперервної роботи менший за 99 % вважається неприпустимим.

3. Час завантаження вебсторінок.

Це час необхідний для повного відображення вмісту вебсторінки. Зокрема дослідження [13] показало, що 53 % відвідувачів залишають сайт, якщо завантаження сторінки триває понад три секунди. Також зазначено, що користувачі не повертатимуться на сайти, завантаження яких займає понад чотири секунди, і формують «негативне сприйняття» компанії з погано зробленим сайтом або розповідають про свій досвід родичам та друзям. Отже, можна рекомендувати, щоб вебхостинг забезпечував час завантаження вебсторінок не більше ніж три-чотири секунди.

Також варто зазначити, що час завантаження вебсторінки може залежати від багатьох чинників, таких як технічні можливості служби вебхостингу, особливості реалізації вебзастосунку, швидкості інтернет-з'єднання, використовуваного користувачем веббраузера та багатьох інших. У будь-якому випадку для збільшення швидкості завантаження вебсторінок рекомендується виконувати оптимізацію вебзастосунку, зокрема рефакторинг його вихідного коду.

4. Обсяг трафіку за певний період.

Обсяг трафіку – це кількість даних, яку служба вебхостингу може надати протягом певного періоду часу. Отже, що вищий цей показник, то краще.

5. Місце розташування центру оброблення даних.

Цей показник впливає на загальну продуктивність вебсайту, зокрема на швидкість завантаження вебсторінок. Низка провайдерів вебхостингу навіть дозволяють обрати бажане розташування центру оброблення даних.

Тому перевагу доцільно надати службі вебхостингу, яка має центр оброблення даних у країні перебування більшості користувачів вебзастосунку або поблизу.

6. Можливість використання HTTPS-з'єднання.

Рекомендується [14], щоб на будь-якому вебсайті використовувався протокол HTTPS для забезпечення певного рівня впевненості його відвідувачам та захисту інформації. Це особливо правильно, якщо вебзастосунок обробляє конфіденційну інформацію, таку як паролі, дані кредитної картки тощо. HTTPS допомагає захистити транзакції вебзастосунку, шифруючи будь-яку інформацію, що проходить через нього.

Використання протоколу HTTPS пов'язано з додаванням сертифіката безпеки на ваш сайт та зазвичай є додатковими витратами. Однак деякі хостинг-провайдери пропонують безкоштовний сертифікат безпеки як частину своїх тарифних планів.

7. Можливість отримання оперативної допомоги.

Останнім важливим показником, що необхідно враховувати, є наявність підтримки клієнтів служби вебхостингу. Від цього залежить можливість швидкого та ефективного вирішення потенційних технічних проблем.

Також додатковими показниками у виборі служби вебхостингу можуть бути необ-

хідний рівень навичок з адміністрування вебзастосунку, частота поновлення підписки на послуги вебхостингу тощо.

Розглянемо питання вибору вебхостингу на прикладі вебзастосунку KпAS4Web [15].

Він призначений для комп'ютерного оцінювання рівня навчальних досягнень студентів.

Для забезпечення функціонування застосунку KпAS4Web цільова служба вебхостингу має підтримувати мову програмування Python 3.8, вебфреймворк Django 3.2 та одну з реляційних СКБД MySQL, PostgreSQL або Oracle.

З огляду на те, що застосунок KпAS4Web використовується за призначенням не дуже часто та його функціональність постійно вдосконалюється, у виборі служби вебхостингу доцільно розглянути тільки ті, що мають безплатні тарифні плани.

На сьогодні цим обмеженням задовольняють вебхостинги Heroku (<https://www.heroku.com/>) та PythonAnywhere (<https://www.pythonanywhere.com/>).

Зазвичай KпAS4Web використовується до десяти разів за навчальний рік. Середній розмір вебсторінки становить 76 Кб. Максимальна тривалість сеансу його застосування становила 50 хв. Протягом цього часу максимальна кількість користувачів не перевищувала 100 та кожен із них виконував не більше ніж 80 HTTP-запитів. Отже, до вебзастосунку надходило до трьох запитів за секунду.

Максимальний обсяг даних, що надходить до вебхостингу (BwR) за сеанс роботи, можемо обчислити як:

$$BwR = APS \cdot MRN \cdot MUN \cdot Rdn, \quad (1)$$

де APS – середній розмір вебсторінки, байт; MRN – максимальна кількість запитів за сеанс від одного користувача; MUN – максимальна кількість користувачів у сеансі; Rdn – коефіцієнт надмірності на випадок несподіваних стрибків трафіку (рекомендоване значення Rdn становить 1,5).

Тоді за формулою (1) маємо, що BwR дорівнює 879 Мб.

Далі проаналізуємо можливість застосування служб вебхостингу Heroku та PythonAnywhere з погляду задоволення апаратних вимог щодо функціонування вебзастосунку KпAS4Web, а також основних показників, визначених раніше (табл. 1).

З наведених даних випливає, що обидві служби вебхостингу задовольняють вимоги щодо доступного дискового простору, обсягу трафіку за місяць та можливості використання HTTPS-з'єднання.

Таблиця 1 – Характеристики безплатних тарифних планів служб вебхостингу Heroku та PythonAnywhere

Показник	Heroku	PythonAnywhere
Доступний дисковий простір	512 Мб	512 Мб
Середній час цілодобової безперервної роботи за рік	76,4 %	99,98 %
Обсяг трафіку за місяць	Немає жорсткого обмеження	Немає жорсткого обмеження
Місце розташування центрів оброблення даних	США, штат Вірджинія; Ірландія, Дублін	США, штат Вірджинія; Німеччина, Франкфурт
Можливість використання HTTPS-з'єднання	Так	Так
Можливість отримання оперативної допомоги	Ні	Так

Безперервну роботу, майже в режимі 24x7 протягом року, за замовчуванням гарантує тільки PythonAnywhere. Для того, щоб служба вебхостингу Heroku працювала подібним чином, існує вимога зазначення реквізитів банківської картки під час реєстрації користувача на відповідному вебсайті. Але з різних причин це може бути недоцільно або взагалі неможливо.

Як Heroku, так і PythonAnywhere дозволяють обрати місце розташування вебзастосунку в центрах оброблення даних, які розташовані на східному узбережжі США. Але користувачі вебзастосунку KпAS4Web на сьогодні перебувають в Україні, здебільшого в Харківській області. Тому для забезпечення мінімальної латентності доцільно розгорнути його на серверах, що розміщені в Європейському регіоні. Для цього можна скористатися центром оброблення даних Heroku в Ірландії (м. Дублін) або відповідним центром PythonAnywhere в Німеччині (м. Франкфурт). Оскільки меншу латентність скоріше за все буде забезпечувати європейський центр PythonAnywhere, то обираємо саме його.

Можливість отримання оперативної допомоги, зокрема електронною поштою, забезпечує тільки служба вебхостингу PythonAnywhere. Heroku надає таку можливість тільки в платних тарифних планах.

Отже, для хостингу вебзастосунку KnAS4Web обираємо службу PythonAnywhere.

### Висновки

Розглянуто проблему вибору провайдера вебхостингу для застосунків на базі фреймворку Django. Зазначено, що використання мови Python для розроблення вебзастосунків обумовлено її особливостями, такими як простота, гнучкість та масштабованість, наявність стандартних бібліотек для вирішення широкого кола проблем тощо. Визначено, що майже всі серверні вебзастосунки розробляються з використанням того чи іншого вебфреймворку. Серед серверних вебфреймворків на базі мови програмування Python одним із найкращих вважається Django. Розглянуто показники якості функціонування застосунків, що розгорнуті на будь-якому вебхостингу, та етапи вибору вебхостингу на прикладі вебзастосунку KnAS4Web. Виконано порівняльний аналіз безоплатних тарифних планів служб вебхостингу Heroku та PythonAnywhere. З наведених даних випливає, що обидві служби вебхостингу задовольняють вимоги щодо доступного дискового простору, обсягу трафіку за місяць та можливості використання HTTPS-з'єднання. Безперервну роботу, майже в режимі 24x7 протягом року, за замовчуванням гарантує тільки PythonAnywhere. Як Heroku, так і PythonAnywhere дозволяють обрати місце розташування вебзастосунку в центрах оброблення даних. Але для забезпечення мінімальної латентності доцільно розгорнути його на серверах, що розташовані в Європейському регіоні. Для цього доцільно скористатися центром оброблення даних PythonAnywhere в Німеччині (м. Франкфурт).

Для хостингу вебзастосунку KnAS4Web рекомендується використовувати службу PythonAnywhere.

### Література

1. TIobe Index for December 2021. 2021. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 10.12.2021).
2. Top Programming Languages 2021. 2021. URL: <https://spectrum.ieee.org/top-programming-languages-2021> (дата звернення: 10.12.2021).
3. The PYPL PopularitY of Programming Language Index. 2021. URL: <https://pypl.github.io/PYPL.html> (дата звернення: 10.12.2021).
4. Pros and Cons of using Python for Web Development. 2021. URL: <https://www.datasciencecentral.com/profiles/blogs/pros-and-cons-of-using-python-for-web-development> (дата звернення: 10.12.2021).
5. The 16 Most Important Pros and Cons of using Python for Web Development. 2021. URL: <https://djangostars.com/blog/python-web-development/> (дата звернення: 10.12.2021).
6. A complete guide to web development in Python. 2020. URL: <https://www.educative.io/blog/>
7. Server-side web frameworks. 2021. URL: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\\_steps/Web\\_frameworks](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks) (дата звернення: 11.12.2021).
8. The official site of the Django Project. 2021. URL: [www.djangoproject.com/](http://www.djangoproject.com/) (дата звернення: 11.12.2021).
9. The 2020 Stack Overflow Developer Survey. 2020. URL: <https://insights.stackoverflow.com/survey/2020/> (дата звернення: 11.12.2021).
10. Best Django Hosting of 2021. 2021. URL: <https://www.hostingadvice.com/how-to/best-django-hosting/> (дата звернення: 10.12.2021).
11. How To Choose A Web Host: 23 Factors To Consider. 2021. URL: <https://bloggingwizard.com/choose-a-web-host/> (дата звернення: 29.12.2021).
12. How to Choose a Web Host: A 15-Point Checklist/ 2021. URL: <https://www.dreamhost.com/blog/how-to-choose-web-host-checklist/> (дата звернення: 29.12.2021).
13. How fast should a website load in 2022? 2021. URL: <https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/> (дата звернення: 28.12.2021).
14. Secure your site with HTTPS. 2021. URL: <https://developers.google.com/search/docs/advanced/security/https> (дата звернення: 29.12.2021).
15. Комп'ютерна програма «Knowledge Assessment System for World Wide Web». Свідоцтво про реєстрацію авторського права на твір від 24.02.2020 № 96294 // *Авторське право і суміжні права: офіційний бюлетень*. Київ: Державне підприємство «Український інститут інтелектуальної власності», 2020. № 57. С. 494.

### References

1. TIobe Index for December 2021. Available at: <https://www.tiobe.com/tiobe-index/> (accessed: 10.12.2021).

2. Top Programming Languages 2021. Available at: <https://spectrum.ieee.org/top-programming-languages-2021> (accessed: 10.12.2021).
3. The PYPL PopularitY of Programming Language Index. Available at: <https://pypl.github.io/PYPL.html> (accessed: 10.12.2021).
4. Pros and Cons of using Python for Web Development. 2021. Available at: <https://www.datasciencecentral.com/profiles/blogs/pros-and-cons-of-using-python-for-web-development> (accessed: 10.12.2021).
5. The 16 Most Important Pros and Cons of using Python for Web Development. Available at: <https://djangostars.com/blog/python-web-development/> (accessed: 10.12.2021).
6. A complete guide to web development in Python. Available at: <https://www.educative.io/blog/> (accessed: 10.12.2021).
7. Server-side web frameworks. 2021. Available at: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\\_steps/Web\\_frameworks](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks) (accessed: 11.12.2021).
8. The official site of the Django Project. 2021. Available at: [www.djangoproject.com/](http://www.djangoproject.com/) (accessed: 11.12.2021).
9. The 2020 Stack Overflow Developer Survey. 2020. Available at: <https://insights.stackoverflow.com/survey/2020/> (accessed: 11.12.2021).
10. Best Django Hosting of 2021. 2021. Available at: <https://www.hostingadvice.com/how-to/best-django-hosting/> (accessed: 10.12.2021).
11. How To Choose A Web Host: 23 Factors To Consider. 2021. Available at: <https://bloggingwizard.com/choose-a-web-host/> (accessed: 29.12.2021).
12. How to Choose a Web Host: A 15-Point Checklist/ 2021. Available at: <https://www.dreamhost.com/blog/how-to-choose-web-host-checklist/> (accessed: 29.12.2021).
13. How fast should a website load in 2022? 2021. Available at: <https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/> (accessed: 28.12.2021).
14. Secure your site with HTTPS. 2021. Available at: <https://developers.google.com/search/docs/advanced/security/https> (accessed: 29.12.2021).
15. Komputerna programa [Computer program] «Knowledge Assessment System for World Wide Web». Svidotstvo pro reyestratsiyu avtorskoho prava na tvir vid 24.02.2020 № 96294 // *Avtorske pravo i sumizhni prava: ofitsiyny byuleten*. Kyiv: Derzhavne pidpryyemstvo «Ukrayinskyy instytut intelektualnoyi vlasnosti», 2020. № 57. С. 494.

**Парфонов Юрій Едуардович**, к.т.н., с.н.с., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 057-702-18-31, [Yurii.Parfonov@m.hneu.edu.ua](mailto:Yurii.Parfonov@m.hneu.edu.ua),

**Колгатін Олександр Геннадійович**, д.п.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 057-702-18-31, [kolgatin@ukr.net](mailto:kolgatin@ukr.net).

### Choosing a web hosting service for applications based on Django framework

**Abstract. Problem.** The article deals with the issues of choosing a hosting provider for Django-based web applications. It is noted that the use of Python for web development is due to its simplicity, flexibility, and scalability, the possibility of rapid prototyping of web applications, the availability of standard libraries to solve a wide range of problems, and a large community of developers. It is indicated that almost all server-side applications are developed using one or another web framework. Among server-side web frameworks based on Python programming language, Django is often considered one of the best. This framework is used on many well-known websites. **Goal.** The goal is to choose a web hosting service to deploy a Django-based web application. **Methodology.** The article describes the performance metrics of applications deployed on any web hosting service. We consider the steps to choose a web hosting service using KnAS4Web application as an example. **Results.** A comparative analysis of free plans for Heroku and PythonAnywhere web hosting services is performed. From the given data it is concluded that the reasonable choice is to use PythonAnywhere service for hosting web application KnAS4Web. To ensure minimal latency, it makes sense to deploy it on the servers located in the European region, namely, at the data center in Frankfurt, Germany. **Originality.** For production-ready web applications such as KnAS4Web, it is suggested to consider only web hosting services with free tariff plans available. To compare the services, we analyze their disk space available, annual uptime, the maximum amount of traffic per month, data centers location, and ability to provide HTTPS connection as well as operative help. **Practical value.** The approach allows to significantly limit the range of potential web hosting services and helps to choose the service for deploying a web application. **Key words:** web hosting, web framework, Python, Django, Heroku, PythonAnywhere.

**Parfonov Yurii**, Ph.D., Assoc. Prof., Information Systems Department, Simon Kuznets Kharkov National University of Economics, tel. +38 057-702-18-31, [Yurii.Parfonov@m.hneu.edu.ua](mailto:Yurii.Parfonov@m.hneu.edu.ua),

**Kolgatin Oleksandr**, D. Sc., Prof., Information Systems Department, Simon Kuznets Kharkov National University of Economics, tel. +38 057-702-18-31, [kolgatin@ukr.net](mailto:kolgatin@ukr.net).

## APPROACHES TO WEB APPLICATION PERFORMANCE TESTING AND REAL-TIME VISUALIZATION OF RESULTS

Ushakova I., Plokha O., Skorin Yu.

Simon Kuznets Kharkiv National University of Economics

**Abstract. Problem.** Today, performance testing is an integral part of the web applications quality assurance whose performance failures and performance issues affect the business of their owners. **Goal.** The goal of the work is to generalize approaches and methods to improve the quality of web applications and develop recommendations for improving performance testing using open source tools. The object of research is the processes of testing web applications. The subject of research is the approaches, methods and tools of performance testing. **Methodology.** The study identified the impact of software performance testing on its quality and its main types, namely load testing, stress testing, volume testing, stability testing. The main stages of performance testing and their content were identified. To implement modern automated testing technologies, the advantages and disadvantages of the most popular tools for testing performance in the modern IT market and continuous visualization of their results were analyzed and identified. The following factors should be considered when selecting a performance testing tool: compatibility, scalability, clarity, and monitoring. Time series databases and visualization tools are used for continuous monitoring of test results together with testing tools. **Results.** During the practical implementation of the research results, the goals of different types of performance testing, indicators of normal operation of the system without exceeding the permissible limits, test scenarios and test results were identified. Visualization of test results in JMeter is shown and a board for continuous real-time visualization is created. **Originality.** The originality of the study lies in unlocking the potential of open source tools for testing the performance of web applications and visualizing its results. On the basis of comparative analysis the spheres of application of tools for performance testing are substantiated. **Practical value.** The practical value lies in the development of methodological bases for testing the performance of web applications in real time on the example of the connection of tools Jmeter – InfluxDB – Grafana.

**Keywords:** testing, web application, performance, load, software quality, quality metrics.

### Introduction

Nowadays, performance testing is an integral part of web application quality assurance. Performance testing is a set of types of testing, the purpose of which is to determine the efficiency, stability of resource consumption and other attributes of application quality under different loads and usage scenarios [1]. Performance testing tries to find possible vulnerabilities and defects in the system during its development in order to prevent their negative impact on the operation of the program in use.

Website malfunctions and performance issues affect the business of their owners. Thus, in 2021, the largest failure in the history of the Internet was recorded, when Facebook, Instagram and WhatsApp stopped working for several hours. About \$ 6,6 billion was lost then, and the company's management had to substantiate itself to users [2]. There are plenty of such examples. The Apple Store lost about \$ 25 million in profits due to a 12-hour delay. Delta Airlines canceled about 2,000 flights and suffered \$ 150 million in

losses due to the failure of the computer system's operations center for 5 hours.

According to Gartner, the average cost of downtime for IT giants is about \$ 300,000 per hour of forced inactivity, and in extreme cases can reach \$ 540,000 per hour [3]. Companies are losing money, but worst of all, they are losing their business reputation. That is why it is important for businesses to correctly calculate the potential load on their websites both in normal operation and at peak times. Companies turn to performance testing to find out the causes of failures, but it must be done in a timely manner.

About 80 % of users admit that they are unlikely to buy goods or services from a company whose site "hangs". This can be illustrated by the fact that, for example, Pinterest increased site traffic by 15 % without any marketing costs, only speeding it up by 40 %, and the BBC found that they lose 10 % of users for every second of site load speed [4]. These and many similar examples emphasize the need to spend money on performance testing. Performance testing itself can

ensure the stability of a web application and improve its quality, so improving the performance testing process when creating such programs is important.

### Analysis of Publications

Analysis of web application testing technologies allows to divide them into groups, namely: functional, aimed at verifying the compliance of functional requirements of the software to its actual characteristics, and non-functional, aimed at verifying properties that do not belong to the functionality of the system. Non-functional properties characterize reliability, performance, ease of use, scalability and security.

Performance testing is one of the key components of non-functional testing [5–11], because it helps to test the behavior of the program in different situations. The system can work effectively with a certain number of concurrent users, but may become inoperable with many additional thousands of users during peak traffic. Performance testing help determine the speed, scalability, and stability of software. There are different types of performance testing that simulate possible user scenarios and record program indicators of behavior

Performance testing does not necessarily reflect defects in the application. It must ensure that the program works properly regardless of fluctuations in network settings, availability and bandwidth or traffic load. This is practically a subset of performance engineering, as a set of measures for software development and improving all life cycle processes of its development, which are aimed at meeting the requirements of productivity [12]. Therefore, the development and implementation of these tests are crucial to ensure the stability of the web application.

Automating performance testing adds significant benefits to improving the quality of web applications. There are various tools used for these purposes, both licensed and open source [12, 13]. Free tools are of particular interest to small and medium-sized enterprises. The advantages of the open source tool Jmeter are emphasized in [6], and Grinder, NeoLoad, LoadRunner in [12].

### Purpose and Tasks

The goal of the research is to generalize approaches and methods to improve the quality of web applications and to develop recommendations for improving performance testing using tools for real-time results analysis.

To achieve this goal, it is necessary to dis-

close the impact of software testing on its quality, determine the metrics and stages of performance testing, the choice of tools and develop recommendations for the performance testing of web applications.

### The impact of software testing on quality

Software testing is an integral part of any modern software development methodology. The essence of the software testing process and its importance as a component of software development are revealed in the literature [14]. The role of testing is one of the key in the life cycle model, because it depends on how high quality the product will reach the customer.

Software testing saves development time and defect correction costs because the cost of troubleshooting is proportional to the time it takes to detect it. Fig. 1 shows the impact of software testing costs on the cost of its quality. Decision making to increase or decrease the number of tests can lead to both the detection and omission of many defects. Therefore, determining the optimal number of tests allows you to minimize the time and cost of testing. The figure clearly shows that the optimal cost of testing is when it is equal to the cost of defect correction.

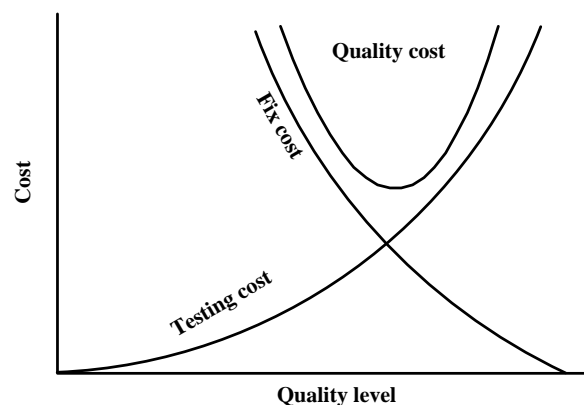


Fig. 1. Dependence of software quality on testing

Recently, the web development market is gaining momentum. And this trend is only growing with each passing year, as entrepreneurs' interest in websites and their mobile versions grows with the transition to the digital economy. As mentioned earlier, an important type of web application testing is performance testing, which includes [1]:

- load testing;
- stress testing;
- volume testing;
- stability testing.

Load testing is performed in order to investigate the possibility of the application to keep the



specified quality indicators under load within the specified limits, as well as a certain excess of these limits to determine the margin of safety. Sometimes this type of testing is used as a synonym for “performance testing”, but this is not always legitimate, because performance testing is a broader concept.

Stress testing is performed to study the behavior of the program with “abnormal” changes in load in abnormal conditions. It allows you to set limits on the bandwidth of the application, the reliability of the system at extreme or disproportionate loads and answers the question of the required performance of the system if the current load significantly exceeds the expected maximum.

Volume testing is used to study the performance of the program when processing different amounts of data without increasing the load and operating time.

Stability testing is performed to make sure that the program will withstand the expected load for a long time. During this type of testing, memory consumption is monitored to assess potential losses. Also, this testing allows you to detect performance degradation by reducing the speed of information processing and increasing the response time of the program after prolonged use of the application.

**Defining metrics and performance testing stages**

For the web application to work successfully, you need to check:

bandwidth, i.e. how fast the server can process requests when a different number of users are connected to the system;

how many simultaneous connections the system can process;

what is the system response time, etc.

The bandwidth of the system includes two components: the number of requests received by the system per second  $QP_S$  and the number of responses (transactions) provided by the system per second  $TP_S$ :

$$QP_S = QP / T_q,$$

where  $QP$  – number of user requests,  
 $T_q$  – total query execution time;

$$TP_S = TP / T_{tr},$$

where  $TP$  – number of system responses,  
 $T_{tr}$  – total transaction execution time.

The number of concurrent connections is determined by the number of concurrent users  $K$ .

The response time  $T_r$  consists of the data transfer time in the network  $T_{rn}$  and the processing time  $T_{ro}$  (fig. 2):

$$T_r = T_{rn} + T_{ro},$$

where  $T_{rn} = N_1 + N_2 + N_3 + N_4$ ,  
 $T_{ro} = A_1 + A_2 + A_3$ ,  
 $A_1, A_3$  – server processing time,  
 $A_2$  – database processing time.

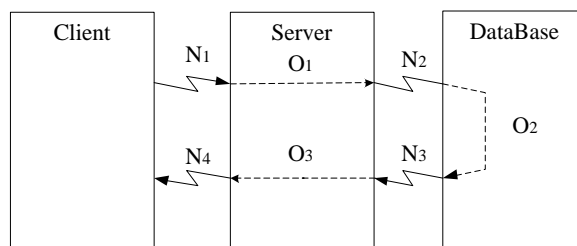


Fig. 2. Components of system response time

Performance testing as a process involves certain stages that need to be clearly defined before testing can begin. Usually distinguish the following stages [14]:

- collecting information for testing;
- definition of test environment;
- testing planning;
- creation of tests;
- environment configuration;
- testing and visualization;
- analysis of results and reporting.

The first step in gathering information about web application performance involves determining:

- critical functionality to be tested;
- expected system response time;
- the expected number of users working simultaneously with the system;
- expected use of system resources;
- future growth of the load on the system, etc.

The definition of a test environment includes the hardware and software and other tools needed to perform the tests. It is necessary that the test environment was as close as possible to the real environment.

Performance testing planning involves actions aimed at defining the main goals of testing and the tasks required to achieve these goals, namely the team of testers, tools, approaches, metrics and priorities. The schedule of performance testing is drawn up at this stage.

The creation of tests includes their design, development of load models, and generation of test data and development of test scenarios in accordance with the project.

The configuration of the environment includes its configuration both on a separate personal computer of the tester for cases of small loading, and a separate usually distributed environment for considerable loading.

Performing stress testing involves running and monitoring test results and usually takes several hours (average two to three hours). If the test is negative, you need to be able to stop the process in real time. In addition, during the experiment, you need to collect the values of many metrics as the server: response time, bandwidth, and the system itself: resource loading, system errors, and so on. To do this, it is convenient to use monitoring systems. They allow you to display all the results in one place and thus quickly notice the relationships between different indicators and decide whether to continue or stop testing.

Collected and analyzed results, identified dependencies are provided to all stakeholders to make further decisions on the quality of the application.

#### Rationale for choosing testing tools, storing and visualizing results

The following factors should be considered when choosing a performance testing tool:

- interoperability,
- scalability,
- clarity,
- monitoring.

Interoperability. Keep in mind that the tools will be used in general in the company, not just for an individual project. Therefore it is necessary to consider in addition the following factors:

protocols used by the system and which of them will be checked;

interfaces to external components such as software components, or possibly to full integration, for example in the CI process;

interoperability with platforms and their versions used to host tools and platforms with which tools interact to monitor and create loads.

Scalability. Performance testing should allow you to track how software is behaving under pressure and provide information on how it can handle scalability.

Clarity should be taken into account given the technical knowledge required by professionals to use the tool. This is often ignored and can lead to unqualified testers setting up tests incorrectly, which in turn can lead to inaccurate results. Some open source testing tools require coding skills. The team must make sure that the tester has the necessary skills, experience and training for testing, which requires complex scenarios and a high

level of programming and configuration.

Monitoring is taken into account for its sufficiency. In addition, find out the availability of other monitoring tools available in the environment and which can be used to supplement with this tool. Determine whether monitoring can be correlated with certain operations.

There are many tools on the market for load testing: Kinsta APM, WebLOAD, Apache Jmeter, LoadNinja, Loadero, SmartMeter.io, StormForge, LoadView, NeoLoad, LoadUI Pro, Silk Performer, AppLoader, Gatling, BlazeMeter, Rational Performance Tester, k6, Eggplant, Loadster, Akamai CloudTest, Parasoft Load Test, Locust, Grinder, Loader.io, LoadStorm, SolarWinds, Test Studio, Taurus [13]. Therefore, there is a need to justify their choice. Among the most popular tools that meet the need for performance testing are: Apache Jmeter [15] and Loadrunner [16]. They are the market leaders and are popular among testers and developers of leading IT companies.

Table 1 compares Apache Jmeter and Loadrunner performance testing tools.

The main difference between the Jmeter and LoadRunner tools is the openness of the software and its price. Jmeter is open source software that can be easily downloaded from the official website. LoadRunner software is available as a paid version, and the user must pay for its use.

Table 1 – Comparison of performance testing tools

Specifications	Apache Jmeter	Loadrunner
Cost	Free	- Community Edition free for 50 users, - \$0.56 per virtual user per day
Code	Open	Micro-Focus (HP)
Platforms and protocols	Java objects Servlets FTP server queries HTTP SOAP Pearl scripts and other	Web services, .net, J2EE, SAP, Siebel, PeopleSoft, Wireless media and other
User interface	Comfortable simple	Comfortable with complex structure
Functionality	Limited	Powerful
Function settings	Yes	No
Users	Developers, small and medium companies	Medium and large companies

Another difference between benchmarking performance testing tools is supported platforms.

Jmeter can support a variety of platforms, such as Java objects, servlets, FTP servers, database queries, HTTP, SOAP, Pearl scripts, and more. It can be easily run for testing on all mentioned platforms. LoadRunner can support platforms such as web services, .net, J2EE, SAP, Siebel, PeopleSoft, wireless media, and more. And all of these platforms can be used to test performance.

Another difference between Jmeter software and LoadRunner is the user interface. The user interface in Jmeter is user-friendly, but less experienced and has fewer features than LoadRunner. The LoadRunner toolkit is technically more advanced and has more advanced functionality but the user interface structure is more complex compared to Jmeter.

Another difference between Jmeter software and LoadRunner is the configuration of software features. Jmeter is an open source tool, so it provides functionality to customize existing features and modify them as required. LoadRunner is not open source, so you have to use existing functions.

Thus, both tools have their advantages and disadvantages, but they remain market leaders in performance testing. Loadrunner will be the best choice for large businesses and Apache Jmeter for small and medium. Therefore, the Apache Jmeter tool will be used for further research.

Jmeter can save the results of each test to files, but after a long time the number of files will be too large. Also, Jmeter generates an extended report only after the tests are completed, so detailed error logs can only be analyzed after the test is completed. But the development team and any stakeholders need to have this information at all times during testing. This will allow you to track test results such as slow transactions, information about API query errors in real time. This is especially true during long tests, because defects can be detected in the test system at the beginning of testing, which leads to the inexpediency of further testing.

Therefore, the next step is to choose the tools for easier storage of test results and their continuous visualization, which will allow monitoring of test results in real time. Typically, non-relational databases are used to store results, which will write real-time test logs to a spreadsheet, and then these data are visualized by a graphing tool.

In the practice of performance testing usually use one of the following approaches [17, 18]:

- InfluxDB та Grafana;
- Elasticsearch, Logstash та Kibana (ELK).

To choose the tools for storing and visualizing

the results, their functional features were analyzed (table 2).

Table 2 – Comparison of functionality

Functionality	ELK	Grafana
Input data formats	++	+
Built-in integrations	+++	+
GUI data streams	++	+
Parsing named	+++	a
Input data processing and enrichment	++	a
Processor templates	++	a
Data visualization	+++	+++
Alert	p	+
Own agents	+	+
Expansion opportunities	+++	+
Documentation	++	++
Installation process	+	+

In the table, the signs "+" indicate the presence of a functional with increasing degree of capabilities, the sign "a" means that this functionality is provided by agents, the sign "p" – paid functionality. The analysis of the considered software solutions revealed a trend on delegation of functionality of primary processing of logs to the local agents that simplifies functionality as it is already visible on Grafana's example. For ELK, there is a trend of gradually reducing free functionality and the emergence of more and more paid. ELK-stack is the most full-featured solution, but the alert is only available in paid versions. Grafana is the simplest and most frivolous solution and is suitable for solving narrow problems related to metric data analysis. Therefore, the approach using free tools was chosen: database - InfluxDB and Grafana visualization tool.

### Performance testing with Apache JMeter, InfluxDB and Grafana

The study of load testing processes focused on the need to reduce economic risks and take into account all the necessary business and technical components. It was determined to conduct all key performance tests, the components of which are listed in table 3.

Performance testing metrics and metrics were determined according to web application quality requirements. The requirements are listed in Table 4, and the most important metrics for identifying problems are in table 5.

The test collections were created based on the test scripts created in Apache Jmeter. Also, groups of test users for stress testing under nor-

mal conditions, as well as for stress loads were prepared.

Table 3 – Functional objectives

Type of testing	Objectives
Load	Assessing software response speed Assessing work speed of hardware Measurements on the number of users Defining productivity limits
Stress	Assessing software response speed Assessing hardware speed Assessing the ability to restore the system Assessing the stress loads impact
Volumetric	Assessing the dependence of the system on the size of the processed data Assessing the number of simultaneously working with the system users Assessing the capabilities of data warehouses
Stability	Memory loss assessment Detection of errors related to data collection Assessing the stability of work for a long time

Table 4 – Load testing requirements

Testing option	Value
Active users limit	200
Number of flows	50
Response time	15 sec.
Bandwidth	1 Mbps
Server memory usage	500 megabytes
Number of records read/written to the data warehouse at the same time	1000 records

Table 5 – Load testing metrics

Name	Definition
CPU usage	Determines how much time from the specified interval was spent by the processor on the calculation
Memory usage	Availability of physical memory for processing in the system
Bandwidth	The highest possible data transfer rate in the network
Response time	The time between user queries and application responses
Speed hits	The speed of loading application pages per second
Active sessions	The maximum number of sessions that can be activated at one time

The parameters used to test the normal load on the system are shown in table 6. Expected test result: responses to the queries were received correctly, the response delay is not more than 5 seconds.

Table 6 – Indicators of normal operation of the system without exceeding the permissible limits

№	Value
1	200 users log in to the site at the same time
2	All users change the page from 1 to 10 seconds
3	Users leave a request for a consultation
4	Users periodically create orders

Graphs of queries (fig. 3) and responses to system queries (fig. 4) were obtained after running the test in JMeter. Fig. 3 shows the maximum number of requests processed per second in green and the minimum number in red. Fig. 4 shows the maximum number of system responses per second in yellow and the average in green. Their analysis shows that our system processes an average of 30 requests per second, and the median response to the request is 63 milliseconds, which meets our requirements.

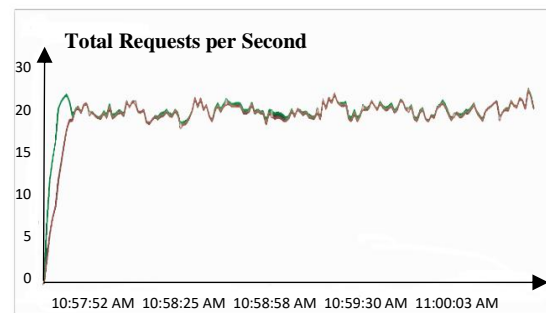


Fig. 3. Number of requests processed per second

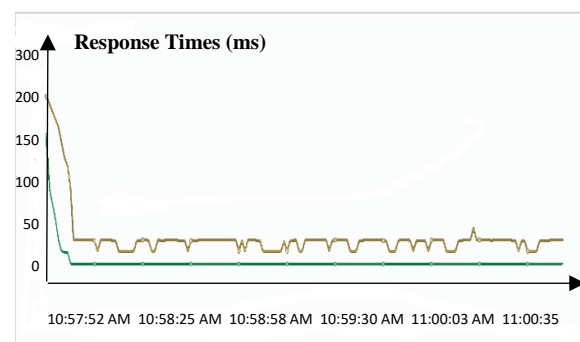


Fig. 4. The number of system responses per second

A Dashboard was created to visualize the results in Grafana. It allows you to track their dynamics throughout the time of performance testing (fig. 5). There are four graphs on the board

that show the results of system performance testing when processing requests from two sites in Chrome: histogram of the number of requests per second (by status); graphs of the number of requests per second (by instances); latency percentiles of request (green for 99 percentiles, yellow for 50 percentiles, and blue for average); distribution of request latency.

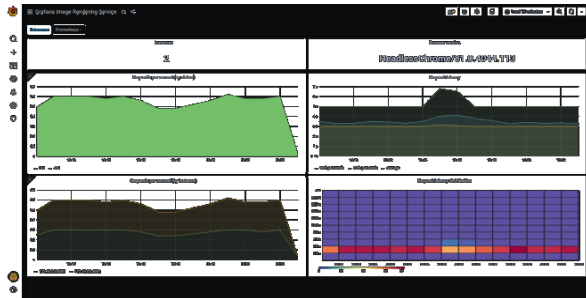


Fig. 5. Visualization of results in Grafana

Modification of indicators occurs with each new pass of test scenarios.

The interactive dashboards helps track performance degradation during testing and identify system vulnerabilities on each of the measured metrics. Analysis of the results of the prepared test scenarios for all types of performance testing makes it possible to determine the readiness of the system for active use, as well as its quality in accordance with the requirements. The results of performance testing according to the developed test scenarios are shown in table 7.

The conducted allows us to draw the following conclusions. The system passed most of the testing scenarios successfully. But checking the time to refresh the application page after prolonged inactive use and checking the execution time of database transactions in stressful situations were not passed.

Table 7 – Testing results

№	Test scenario	Expected result	Yes/No
1	2	3	4
<b>Load testing</b>			
1	Check the operation of the system while the normal number of users are	Work without failures	Yes
2	Check the system operation while a normal number of users are with the authorization page	Work without failures	Yes
3	Check the speed of processing the request to the server with a normal number of requests	Processing speed in an acceptable range	Yes
4	Check the application response time for a normal network connection	The response time is in the acceptable range	Yes
5	Determine the maximum number of users that the program can work with before it shuts down	Maximum number of users when working without failures	Yes
6	Check database execution time when 500 records are read/written at one time	Execution time in an acceptable range	Yes
7	Check database execution time when 1000 records are read/written at one time	Execution time in an acceptable range	Yes
8	Test CPU and memory usage by the application and database server under normal load	Use of resources in the normal range	Yes
9	Check the application response under normal load	Response time in the normal range	Yes
10	Check the response time of the application under low load conditions	Response time in the normal range	Yes
11	Check the response time of the application under moderate load	Response time in the normal range	Yes
<b>Stress testing</b>			
12	Check the operation of the system while interacting with an excessive number of users	System shutdown	Yes
13	Check the operation of the system while interacting with an excessive number of users with the authorization page	System shutdown	Yes
14	Check the speed of processing the request to the server in case of excessive number of requests	Requests are processed in turn	Yes
15	Check the response time of the downloaded program with a slow network connection	Response time in the normal range	Yes
16	Check the runtime of the database when 1500 entries are read or written simultaneously	Execution time in an acceptable range	No

1	2	3	4
17	Check CPU and memory usage by program and database server under overload	The system stops working	Yes
18	Check the response time of the program under overload conditions	Response time increases	Yes
Volume testing			
19	Check the system operation when simultaneously downloading files to the warehouse with a normal number of downloads and with a file of normal size	File uploaded successfully	Yes
20	Check the operation of the system when uploading files to the warehouse with an excessive number of downloads and with a file with an excessive volume	File upload declined	Yes
Stability testing			
21	Check for memory loss during peak system interactions	The system is working properly	Yes
22	Check the load time of the application page during peak interaction	File upload declined	Yes
23	Check the refresh time of the application page after an hour of inactive use	The page reloaded successfully without delay	No
24	Check page refresh time during peak user interaction	The page reloaded successfully without delay	Yes

The results of load testing indicate a high level of quality of the software product in terms of performance in accordance with the requirements. Most of the test scenarios were passed with success. These results indicate the readiness of the system for active use at high and stressful loads.

During the testing process, defects that affected system performance and could affect financial losses due to temporary unavailability of the system for customers were identified. Performance and load testing has prevented these risks and identified system vulnerabilities that need to be addressed to further scale the software product.

### Conclusion

Digitalization of the business contributes to the further spread of web applications, the quality of which is directly related to performance testing. Performance testing involves researching a system with different loads and monitoring the test results and usually lasts several hours. If testing gives negative results, it should be possible to stop the process in real time to reduce the cost of unnecessary testing.

Many metrics of the system need to be collected during the experiment. The use of a monitoring system during testing allows you to quickly determine the relationships between different indicators and decide whether to continue or stop testing.

An approach to web application performance testing is proposed, which will provide continuous monitoring of performance test results through the use of technologies for storing test

results and their visualization. The choice of testing tools was justified, namely Apache Jmeter – for performance testing, InfluxDB – for storing test results, Grafana – for creating dashboards with results. The proposed approach is described by a real example.

### References

1. Тестування продуктивності. Qalight. URL: <https://qalight.ua/baza-znaniy/testuvannya-produktivnosti> (дата звернення: 02.01.2022).
2. Locked out and totally down: Facebook's scramble to fix a massive outage. URL: <https://www.theverge.com/2021/10/4/22709575/facebook-outage-instagram-whatsapp> (дата звернення: 02.01.2022).
3. The Cost of Downtime. Gartner. URL: <https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime> (дата звернення: 02.01.2022).
4. Обновление PageSpeed Insights: что изменилось, на какие метрики обращать внимание? URL: <https://siteclinic.ru/blog/technical-aspects/obnovlenie-pagespeed-insights> (дата звернення: 02.01.2022).
5. Draheim D., Grundy J., Hosking J. Lutteroth C., Weber G. Realistic Load Testing of Web Applications. *Conference on Software Maintenance and Reengineering (CSMR'06). IEEE Xplore*. 2006. 11 p.
6. Hamza Z. A., Hammad M. Testing Approaches for Web and Mobile Applications: An Overview. *International Journal of Computing and Digital Systems*. 2020. Vol. 9. No. 4. P. 657–664.
7. Israr Gh., Wan M.N., Ahmad M. Web Service Testing Techniques: A Systematic Literature Review. *International Journal of Advanced Comput-*

- er Science and Applications*. 2019. Vol. 10. No. 8. P. 443–458.
8. Kao Ch., Lin Ch., Lu H. Toward Automatic Performance Testing for REST-based Web Applications. *ICSEA 2016: The Eleventh International Conference on Software Engineering Advances*. 2016. P. 68–71.
  9. Legramante G., Bernardino M., Rodrigues E., Basso F. Systematic Literature Review on Web Performance Testing. *Conference: Escola Regional de Engenharia de Software*. 2020. No. 4. P. 285–295.
  10. Legramante G., Bernardino M., Rodrigues E., Basso F. Systematic Literature Review on Web Performance Testing. *2020: Proceedings of the 4th Regional School of Software Engineering*. 2020. 11 p.
  11. 10 Best Practices for Application Performance Testing: Leveraging Agile Performance Testing for Web and Mobile Applications. Orasi Software, Inc. 2018. 9 p.
  12. Bui S., Shrivastava M., lee E., Dhaliwal J. A case study of testing a web-based application using an open-source testing tool. *Journal of Information Technology Management*. 2015. Vol. XXVI. No. 1. P. 19–30.
  13. Top 27 Performance Testing Tools to Use in 2022. URL: <https://kinsta.com/blog/performance-testing-tools> (дата звернення: 02.01.2022).
  14. Crispin L., Gregory J.: Agile testing. Addison-Wesley, 2014. 464 с.
  15. Apache JMeter™. URL: <https://jmeter.apache.org> (дата звернення: 02.01.2022).
  16. LoadRunner Professional. URL: <https://www.microfocus.com/en-us/products/loadrunner-professional/overview> (дата звернення: 02.01.2022).
  17. Grafana. Dashboard anything. Observe everything. URL: <https://grafana.com/grafana/>
  18. What is the ELK Stack? URL: <https://www.elastic.co/what-is/elk-stack> (дата звернення: 02.01.2022).
  6. Hamza Z. A., Hammad M. Testing Approaches for Web and Mobile Applications: An Overview. *International Journal of Computing and Digital Systems*, 2020, vol. 9, no. 4, p. 657–664.
  7. Israr Gh., Wan M. N., Ahmad M. Web Service Testing Techniques: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*, 2019, vol. 10, no. 8, p. 443–458.
  8. Kao Ch., Lin Ch., Lu H. Toward Automatic Performance Testing for REST-based Web Applications. *ICSEA 2016: The Eleventh International Conference on Software Engineering Advances*, 2016, p. 68–71.
  9. Legramante G., Bernardino M., Rodrigues E., Basso F. Systematic Literature Review on Web Performance Testing. *Conference: Escola Regional de Engenharia de Softwar*, 2020, no. 4, p. 285–295.
  10. Legramante G., Bernardino M., Rodrigues E., Basso F. Systematic Literature Review on Web Performance Testing. *2020: Proceedings of the 4th Regional School of Software Engineering*, 2020, 11 p.
  11. 10 Best Practices for Application Performance Testing: Leveraging Agile Performance Testing for Web and Mobile Applications. Orasi Software, Inc. 2018. 9 p.
  12. Bui S., Shrivastava M., lee E., Dhaliwal J. A case study of testing a web-based application using an open-source testing tool. *Journal of Information Technology Management*, 2015, vol. XXVI, no. 1, p. 19–30.
  13. Top 27 Performance Testing Tools to Use in 2022. Available at: <https://kinsta.com/blog/performance-testing-tools> (accessed: 02 Jan. 2022).
  14. Crispin L., Gregory J. Agile testing. Addison-Wesley, 2014. 464 с.
  15. Apache JMeter™. Available at: <https://jmeter.apache.org> (accessed: 02 Jan. 2022).
  16. LoadRunner Professional. Available at: <https://www.microfocus.com/en-us/products/loadrunner-professional/overview> (accessed: 02 January 2022).
  17. Grafana. Dashboard anything. Observe everything. Available at: <https://grafana.com/grafana/>
  18. What is the ELK Stack? Available at: <https://www.elastic.co/what-is/elk-stack> (accessed: 02 January 2022).

### References

1. Performance testing. Qalight. Available at: <https://qalight.ua/baza-znaniy/testuvannya-produktivnosti> (accessed: 02 January 2022).
  2. Locked out and totally down: Facebook’s scramble to fix a massive outage. 2021. Available at: <https://www.theverge.com/2021/10/4/22709575/facebook-outage-instagram-whatsapp> (accessed: 02 Jan. 2022).
  3. The Cost of Downtime. Gartner. 2021. Available at: <https://blogs.gartner.com/andrew-lerner/2014/07/6/the-cost-of-downtime> (accessed: 02 Jan. 2022).
  4. Обновление PageSpeed Insights: что изменилось, на какие метрики обращать внимание? 2019. Accessed 02 Jan. 2022 <https://siteclinic.ru/blog/technical-aspects/obnovlenie-pagespeed-insights> (accessed: 02 Jan. 2022).
  5. Draheim D., Grundy J., Hosking J. Lutteroth C., Weber G. Realistic Load Testing of Web Applications. *Conference on Software Maintenance and Reengineering (CSMR'06). IEEE Xplore*, 2006, 11 p.
- Ushakova Iryna**, Ph.D., Assoc. Prof. Informaton System Department, Simon Kuznets Kharkiv National Economic University, tel. +38 066-785-09-92, [iryana.ushakova@hneu.net](mailto:iryana.ushakova@hneu.net),  
**Plokha Olena**, Ph.D., Assoc. Prof. Informaton System Department, Simon Kuznets Kharkiv National Economic University, tel. +38. 095-570-47-11, [badhel@i.ua](mailto:badhel@i.ua),  
**Skorin Yuri**, Ph.D., Assoc. Prof. Informaton System Department, Simon Kuznets Kharkiv National

Economic University, tel. +38 066-748-47-51, skorin.yuriy@gmail.com.

### **Підходи до тестування продуктивності вебзастосунків і візуалізації результатів у реальному часі**

***Анотація.** Досліджено вплив тестування продуктивності програмного забезпечення на його якість. Для впровадження сучасних технологій автоматизованого тестування були проаналізовані й визначені переваги та недоліки найбільш популярних на сучасному ІТ-ринку інструментальних засобів тестування продуктивності й візуалізації їхніх результатів, що використовуються для безперервного моніторингу в режимі реального часу. Визначені цілі різних видів тестування продуктивності, показники нормальної роботи системи без перевищення допустимих меж, наведені тестові сценарії та результати тестування. Показана візуалізація результатів тестування в JMeter та*

*створена дошка для безперервної візуалізації в реальному часі.*

***Ключові слова:** тестування, вебзастосунок, продуктивність, навантаження, якість програмного забезпечення, метрики якості.*

**Ушакова Ірина Олексіївна**, к.е.н., доцент кафедри інформаційних систем,  
Харківський національний економічний університет ім. С. Кузнеця,

тел. +38 066-785-09-92, iryna.ushakova@hneu.net,

**Плоха Олена Борисівна**, к.е.н., доцент кафедри інформаційних систем,

Харківський національний економічний університет ім. С. Кузнеця,

тел. +38 095-570-47-11, badhel@i.ua,

**Скорін Юрій Іванович**, к.т.н., доцент кафедри інформаційних систем,

Харківський національний економічний університет ім. С. Кузнеця,

тел. +38 066-748-47-51, skorin.yuriy@gmail.com.

---



## АНАЛІЗ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ МОБІЛЬНИХ ЗАСТОСУНКІВ ДЛЯ ОС ANDROID

Федорченко В. М.<sup>1</sup>, Поляков А. О.<sup>1</sup>, Сєверінов О. В.<sup>2</sup>

<sup>1</sup> Харківський національний економічний університет ім. С. Кузнеця

<sup>2</sup> Харківський національний університет радіоелектроніки

**Анотація.** Запропонована стаття резюмує результати проведеного авторами дослідження щодо вибору платформ розроблення мобільних додатків для ОС Android з урахуванням їхньої продуктивності. Пропонується метод аналізу продуктивності на основі профілювання додатку за пам'яттю, за завантаженням процесора, за кількістю коду та мережної активності. За результатами дослідження платформа Xamarin має обмеження з боку дозволів, потребує більше ресурсів як пам'яті, так енергії живлення, що може бути відчутним у великих застосунках.

**Ключові слова:** ОС Android, мобільна платформа, native-платформа, мультиплатформна розробка, кросплатформність, Android profiler, платформа Xamarin.

### Вступ

Android – це мобільна операційна система (ОС) з відкритим вихідним кодом із найбільшою кількістю користувачів у світі [8]. Згідно зі статистикою використання різних мобільних операційних систем у світовому масштабі на грудень 2021 р. Android є системою, що використовують понад 70 % користувачів [9], але цей показник може відрізнятися залежно від регіону світу, що подано в табл. 1.

Таблиця 1 – Розподіл мобільних ОС залежно від регіону світу, Україна та Worldwide

Регіон	Android	IOS	Samsung
Europe	63,65 %	35,66 %	0,63 %
North America	44,12 %	55,54 %	0,29 %
Asia	82,09 %	17,09 %	0,41 %
Oceania	53,87 %	44,55 %	1,52 %
Africa	82,69 %	15,45 %	0,2 %
South America	87,52 %	12,2 %	0,22 %
Ukraine	77,9 %	21,77 %	0,22 %
Worldwide	70,05 %	29,21 %	0,43 %

Але не всі програми мають однакову продуктивність на кожному пристрої Android, оскільки кожен пристрій має різні обчислювальні можливості. Він має обмеження щодо потужності, пам'яті та ресурсів, які впливають на аспекти продуктивності [11]. У процесі розроблення застосунків для Android важливо враховувати продуктивність застосунку, оскільки кожна програма має різні характеристики та підходи. Як правило, продуктивність залежить від платформи, на якій розробляється та виконується застосунок. Необхідно ще наголосити на низці фреймво-

рків, за допомогою яких можна розробити застосунки під ОС Android, але використання різних технологій, імовірно, обмежить можливості кінцевого продукту. Тому розвиток мобільних платформ вимагає чіткого розуміння, які платформи доцільно використовувати для вирішення завдань, що виникають у процесі розроблення.

Native-застосунки інтегруються з ОС пристрою так, що мають змогу працювати швидше та гнучкіше, ніж альтернативні типи застосунків, розроблені за іншими платформами. Наприклад, застосунок Facebook колись був написаний із використанням WebView віджета в HTML5, щоб застосовувати той самий код для iOS, Android та мобільної мережі. Проте програма була повільною для користувачів iOS, що привело розробників Facebook до створення окремого коду для iOS.

Отже, можна зробити висновок, що не кожний підхід буде однаково гарно вирішувати будь-яке завдання, що розробляється для мобільної ОС. Щодо native-застосунків, то вони мають підтримувати всі native-технології та апаратні можливості конкретної платформи, у цьому дослідженні ОС Android [3].

Залежно від використаних технологій на вигляд однакові програми можуть споживати різну кількість ресурсів смартфона, тобто в них буде різна продуктивність. Застосунки можуть споживати неоднакову кількість пам'яті, процесорного часу, по-різному впливати на витрату батареї, завантажувати більше або менше даних по мережі. Зрештою все це впливає на “Look and Feel” і продук-

тивність вашої андроїд програми. Тому на-самперед у проектуванні самого застосунку треба думати про продуктивність за багатьма критеріями. Отже, потрібне наукове дослідження для порівняння кількох аспектів використання платформ мобільної розробки, щоб з'ясувати переваги та недоліки кожної з них.

### Аналіз публікацій

Питання продуктивності застосунків постійно досліджуються в науковій спільноті. У роботі Doug Sillars [11] аналізує питання, пов'язані з продуктивністю мобільного застосунку в розрізі продажів та втримання користувачів застосунку; особливостями створення пулу пристроїв для максимізації покриття тестування інтерфейсу користувача, функціональності та продуктивності; моніторингу з витоків пам'яті та проблем із процесором, що впливають на продуктивність; особливостями роботи застосунку в мобільних пристроях.

Існують дослідження архітектури MVC та VIPER native Android застосунків у роботі В. Гуменюка [7]. Аналіз продуктивності архітектур MVC, MVP та MVVM у мобільних пристроях було виконано в роботі Т. Лоу [12]. У ній аналіз архітектур виконувався за методикою «Architecture Tradeoff Analysis Method» (АТАМ), в якій три фактори якості – тестованість, можливість модифікації та продуктивність – розглядаються як стандарт для оцінювання. Також робиться висновок, що за продуктивністю архітектура MVC споживає більше пам'яті, ніж MVP і MVVM, але останні дві архітектури потребують більш розгорнутого аналізу. Такий аналіз проведено в роботі В. Wisnuadhi [6], де встановлено, що продуктивність архітектури MVVM краща у використанні ЦП і часу виконання, тоді як MVP краща щодо використання пам'яті.

Сучасні застосунки стають усе більш кросплатформні за рахунок використання проміжного шару адаптації коду, або на етапі компіляції, або під час виконання – і все це додатково знижує продуктивність щодо native Android застосунку. Отже, методи та підходи, що використовувалися для аналізу продуктивності архітектур, можуть бути застосовані під час аналізу продуктивності різних платформ Android для розроблення застосунків.

### Мета та постановка завдання

Метою цього дослідження є огляд методів і засобів для створення native-застосунків, а завдання – розробити застосунок з однако-вим функціоналом, але виконаний на різних платформах. Але за умови такого постійного зростання кількості конкурентів перед розробниками мобільних застосунків постає питання: який підхід і технології варто використовувати в розробленні, щоб нові програми не тільки мали змогу доступу до native-функцій, визначалися швидкодією та високою продуктивністю, а й були конкурентоспроможними. Наприклад, така можливість native-застосунків, як позначення геолокації дозволяє компаніям підлаштовувати свої програми таким чином, щоб споживачі могли отримувати повідомлення, коли вони перебувають біля фізичних магазинів.

### Аналіз технологій платформ і мов розроблення мобільних застосунків для ОС Android

Беручи до уваги рейтинг мов програмування для мобільної розробки на 2021 р. з офіційного сайту спільноти програмістів DOU [2] (рис. 1), очевидно, що треба провести додаткову класифікацію для виявлення платформ, наближених до native Android.

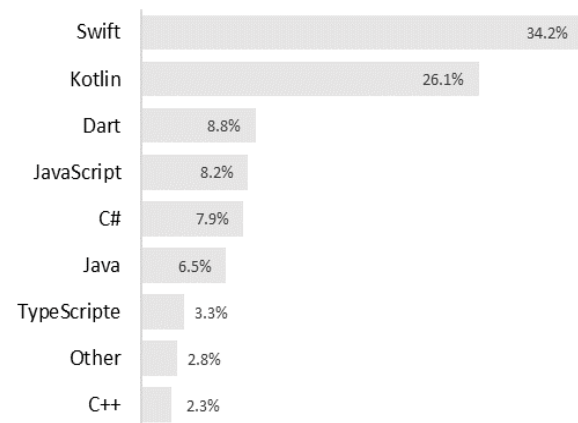


Рис. 1. Рейтинг мов програмування станом на 2021 р. з dou.ua

Окремо можна розглядати платформу Xamarin, яка реалізує застосунки, що мають ознаки native-коду. Загалом програми під Android можна розробляти практично будь-якою об'єктно орієнтованою мовою програмування. Найкраще для цього завдання підійдуть Java або Kotlin. У травні 2019 р. компанія Google оголосила Kotlin кращою мовою для Android-розробки [1]. З одного боку, Java все ще можна вважати офіційною

мовою андроїду, але вже існує низка факторів, здатних схилити чашу терезів на користь Kotlin, такі як: менша кількість коду; досить передбачувана перспектива розвитку, орієнтована саме на Android-розробку; позитивні особливості, які відсутні в Java. Дійсно, Java вимагає написання набагато більшої кількості коду, ніж Kotlin, тому існує більш високий ризик помилок. Узагальнювальна порівняльна інформація про мови Java та Kotlin наведена в таблиці 2.

Незважаючи на всі переваги Kotlin, його документація передбачає знання Java [8]. Так, з технічних аспектів різниця суттєва, але щоб займатися Android-розробкою, треба

вчити обидві мови. Більшість популярних бібліотек підтримують зворотню сумісність із Kotlin, тобто код Java можна використовувати в Kotlin і навпаки, а ось щоб вирішити проблему або банально зрозуміти всі нюанси документації, варто знати Java.

Щодо Xamarin, то його можна використовувати за допомогою C# або F#. Xamarin.Android – бібліотека класів для C#, що надає розробнику доступ до Android SDK.

Тому оцінюючи мови, можна зробити висновок, що для порівняння найкращим варіантом буде обрати мови програмування Java і C# і використати технології Android SDK і Xamarin відповідно.

Таблиця 2 – Порівняння мов програмування Java і Kotlin

Критерії / мова програмування	Kotlin	Java
Платформи / підтримка	open source конвертер Java в Kotlin підтримка ООП і ФП	open source (лише реалізація OpenJDK) ООП
Час компіляції	повільніше, ніж у Java (генерує додатковий байт-код)	швидко (native)
Об'єм коду початковий код	коротка (компактна)	надмірно багатослівна
Проміжний код (байт-код)	генерує додатковий байт-код для підтримки конструкцій мови	генерується компактний код (байт-код орієнтований для Java)
Швидкість розробки	висока	досить висока
Підтримка ком'юніті	зростаюча спільнота, слабкий канал ком'юніті	гігантська спільнота на GitHub, Reddit та StackOverflow
Безпека	більш безпечно (завдяки нульовій безпеці)	безпечно

Можна визначити два види платформ:  
*одноплатформні* (single platform) – native-застосунки;

*мультиплатформні* (multiple platform) – гібридні або native-застосунки.

*Одноплатформні*: кожна операційна система має власний набір пакетів SDK для розроблення застосунків із швидким часом відгуку, графічною продуктивністю та підтримкою вбудованих сервісів (location, cellular, camera, sensors, Bluetooth тощо, рис. 2). Такі застосунки чутливі (responsive), мають високу швидкість, безперервну підтримку та широкий спектр функцій у SDK, але необхідно зазначити: якщо застосунок розроблений для понад однієї платформи, то підтримка двох різних кодових баз, відповідних команд розробників та синхронізація функцій між застосунками стає здебільшого слабким місцем.

Мови Java / Kotlin та Objective-C / Swift треба розглядати як native для відповідних ОС Android та iOS.

Платформа Xamarin

Xamarin – це платформа з відкритим кодом для створення сучасних та ефективних застосунків для iOS, Android та Windows за допомогою .NET. Ця платформа розширює платформу .NET спеціальними бібліотеками для iOS, macOS, Android тощо. Xamarin – це абстрактний рівень, який управляє зв'язком спільного коду з базовим кодом платформи. Xamarin працює в керованому середовищі, що забезпечує такі зручності, як виділення пам'яті та збір сміття [4]. Ця платформа дозволяє розробникам писати всю свою бізнес-логіку однією мовою і водночас досягати native-характеристик та вигляду на кожній платформі.

Фреймворк містить декілька основних компонентів:

*Xamarin.iOS* – бібліотека класів для мови C#, що надають розробнику доступ до платформи iOS SDK;

*Xamarin.Android* – бібліотека класів для мови C#, що надають розробнику доступ до Android SDK;

компілятори для iOS і Android;  
IDE Xamarin Studio;  
плагіни для Visual Studio.

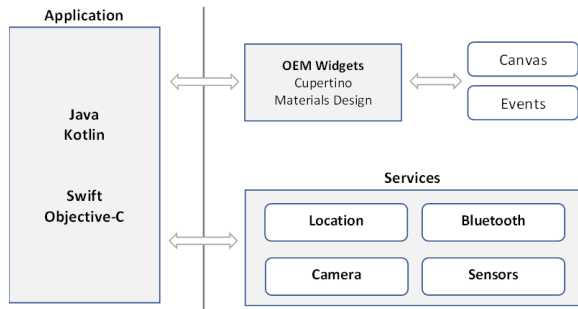


Рис. 2. OEM SDK – пряма інтеграція з native-функціями ОС

Щодо виконання застосунків, то між iOS і Android є одна ключова різниця – спосіб їхньої попередньої компіляції. Як відомо, для виконання застосунків на Android використовується реєстр орієнтований JVM Dalvik або Android runtime (ART). Застосунки, що пишуться мовою Java, компілюються в проміжний байт-код, який інтерпретується Dalvik у команди процесора в момент виконання програми (тобто аналогічно тому, як працює CLR в .NET). Це так звана компіляція Just-in-time (компіляція на льоту). В iOS використовується інша модель компіляції – Ahead-of-Time (компіляція перед виконанням). Xamarin враховує цю різницю, надаючи окремі компілятори для кожної з цих платформ, що дозволяють на виході отримувати справжні native-застосунки, які можуть використовувати всі апаратні й програмні ресурси платформи.

Для Android у процесі компіляції програми відбувається переклад коду з C# у проміжний байт-код, зрозумілий віртуальній машині Mono, і сама ця віртуальна машина також додається в упакований застосунок. Компоненти Mono та Dalvik/ART написані на C і працюють поверх ядра Linux (варто зазначити, що Android основана на Linux). Під час запуску програми на Android обидві віртуальні машини починають працювати паралельно й обмінюються даними через спеціальний механізм wrapper-ів [4] (рис. 3).

Xamarin.Native (у цьому дослідженні Xamarin.Android) дозволяє досягти повторного застосування коду та максимально використати функціональність .NET під час створення мобільного застосунку. Використовуючи Xamarin, уся бізнес-логіка та серверний код є повністю спільними та мають

повний доступ до базової платформи. Коли розробник починає писати свій застосунок на Xamarin, перед ним постає питання: Xamarin.Native або Xamarin.Forms. Зі знаннями .Net можна писати як з Xamarin.Native, так і Forms, але з XF інтерфейс створюється з простим у використанні XAML. З іншого боку, якщо в наявності команда з певним досвідом роботи з Android та iOS, тоді Xamarin.Native може окупитися в плані часу та якості.

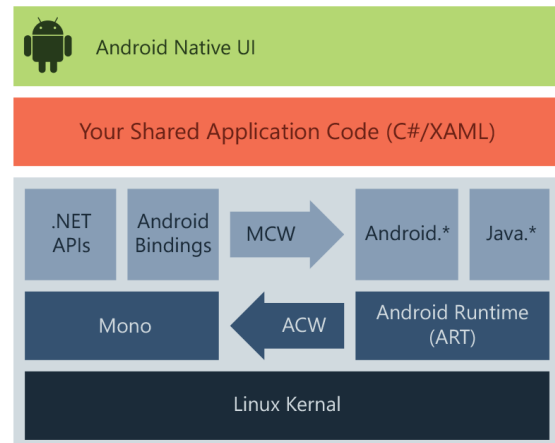


Рис. 3. Схема архітектури Xamarin.Android

Більшість власників бізнесу обирають платформу Xamarin, оскільки це скорочує час виходу на ринок (Time-To-Market) та інженерні витрати за рахунок спільного використання коду й застосування єдиного стека технологій.

#### Платформа Android SDK

Android SDK (Software Development Kit) – набір усіх компонентів від Google, пакет засобів для розроблення програм на Android – містить усе необхідне для створення, компіляції та запакування Android-програм. Ці програми здебільшого розробляються із застосуванням мови програмування Java. Усе, що ми робимо на Android за допомогою Java, залежить від Android SDK – якщо ми створюємо застосунок під певну версію, наприклад, для Android Nougat, то в нас мають бути встановлені відповідні інструменти SDK. Це треба враховувати в процесі розроблення [10].

Android SDK містить Android Debug Bridge, який є інструментом, що дозволяє підключитися до віртуального або реального пристрою для керування ним чи налагодження програми [5].

На сьогодні компанія Google пропонує тільки одне інтегроване середовище для роз-

роблення нових програм, і це Android Studio (інтегроване середовище розроблення для ОС Android від Google, створене на базі JetBrains IntelliJ IDEA та адаптоване під Android-розробку). Інше середовище розробки Andmore: Development Tools for Android (ADTA) базується на Eclipse IDE. ADTA – це набір компонентів (плагінів), які розширюють можливості Eclipse IDE для розроблення під операційну систему Android.

Обидва середовища містять увесь необхідний функціонал для створення, компіляції, налагодження та розгортання Android-програм. Вони також дозволяють розробнику створювати й запускати віртуальні пристрої Android для тестування. У SDK є плагіни для IDE, інструменти для збірки коду, інструменти для відлагодження, бібліотеки, емулятори.

Розглядаючи архітектуру Android-застосунків, можна побачити, що код поділений на два рівні: рівень даних містить DataManager і набір класів-помічників, рівень уявлення складається з класів Android SDK, таких як Activity, Fragment, ViewGroup тощо (рис. 4). *Helper-Class* (класи-помічники, третя колонка в діаграмі) мають дуже обмежені сфери відповідальності, і їх реалізують в послідовній манері. Наприклад, більшість проектів мають класи для доступу до REST API, читання даних із DB або взаємодії з SDK від сторонніх виробників. У різних застосунків формується різний набір helper-class, але найбільш часто використовуваними будуть такі:

*PreferencesHelper* – працює з даними в SharedPreferences;

*DatabaseHelper* – працює з SQLite;

*servicui Retrofit*, що виконують звернення до REST API.

*DataManager* є центральною частиною

архітектури та широко використовує оператори RxJava для того, щоб комбінувати, фільтрувати й трансформувати дані, отримані від помічників. Завдання *DataManager* полягає в тому, щоб звільнити Activity і Fragments від роботи щодо оброблення даних – він буде виконувати всі потрібні трансформації всередині себе й віддавати назовні дані, готові до відображення.

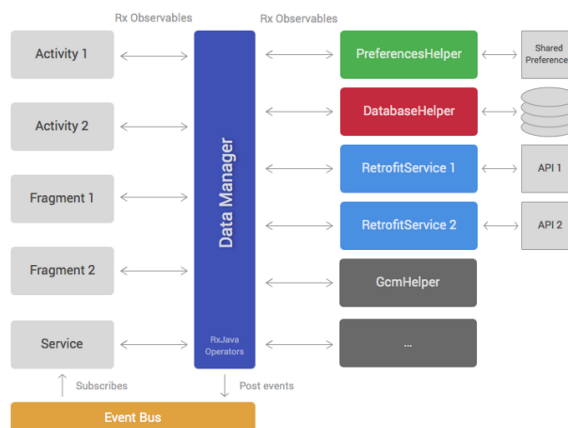


Рис. 4. Архітектура Android-застосунку

*Мультиплатформні*: застосунки підтримують декілька платформ з уніфікованим стилем і кодовою базою.

#### Мета та постановка завдання

Android SDK та Xamarin насамперед класифікуються як «Набір для розробки ПЗ» та «Платформа для кросплатформної розробки» відповідно. Порівняємо обрані технології за такими показниками: стек технологій, використання коду платформою, UI/UX (User Interface / User Experience), продуктивність, можливості апаратних засобів та IDE. Результат порівняння наведено в таблиці 3.

Таблиця 3 – Порівняння Xamarin і Android SDK

Критерії / мобільна платформа розробки	Xamarin	Android SDK
Стек технологій	.Net framework + native бібліотеки	native-бібліотеки
Використання коду	спільне використання з іншою платформою до 96 % з Xamarin.Forms	єдиний код
UI/UX (User Interface / User Experience)	можливе повне налаштування UI для платформи	платформозалежний UI
Продуктивність	близька до рідної	відмінна (native)
Можливості апаратних засобів	високі (Xamarin використовує платформозалежні API і підтримує зв'язок з native-бібліотеками)	високі (native-інструменти мають повну підтримку для можливостей системи)
IDE	Visual Studio	Android Studio

У разі створення застосунків зі складним інтерфейсом із використанням Хамагін кількість загального коду кардинально зменшується. Отже, кросплатформна розробка втрачає свою основну перевагу. Але це не робить застосунки на Хамагін менш якісними, просто трохи збільшує витрати. Проте жоден із інших кросплатформних інструментів не може мати такого самого рівня продуктивності, які пропонує платформа Хамагін.

Метою дослідження є розроблення двох native-застосунків, що мають такі функціонали:

- можливість відкрити телефонну книгу користувача;
- змога обрати контакт;
- відображення інформації про обраний контакт.

Застосунки мають бути розроблені з використанням таких технологій:

a) мови програмування: у прикладі показано:

- Java/Kotlin
- C#

b) інструменти / платформи:

- Android SDK
- Xamarin

c) середовища розробки:

- Android Studio
- Visual Studio

d) застосунки мають підтримувати версію Android 6.0.1 і пізнішу (що покриває майже 98 % використовуваних пристроїв).

Метою дослідження є порівняння таких показників, як вага застосунку, швидкодія та продуктивність.

### Метод аналізу продуктивності профілюванням

Дуже зручно, що деякі середовища для розроблення навіть дозволяють відстежити всі основні метрики продуктивності вашого застосунку. Наприклад, якщо існує проєкт в Eclipse, можна відкрити перспективу DDMS. Там є змога профілювати програми та отримати уявлення про те, скільки часу займає кожен метод / що займає найбільше часу.

Також дуже зручний вбудований в Android Studio профілювальник дозволяє відстежити всі основні метрики продуктивності, такі як: обсяг пам'яті, завантаження процесора, використання мережі та споживання енергії (див. рис. 5).

Для того щоб обрати платформу, треба чітко знати завдання, що має виконувати застосунок, і яким саме функціоналом він має

володіти. Оцінюючи технології розроблення мобільних застосунків, необхідно зазначити, що цей напрям вимагає вибору критеріїв оцінювання.

Показники оцінювання native-застосунків можна поділити на технічні та якісні. На швидкість та якість розроблення впливають такі якісні показники:

- популярність мови;
- наявність фахівців;
- наявність повної документації та технічна підтримка;
- зручність розроблення і налагодження (зручний синтаксис (синтаксичний цукор), наявність колекції (контейнерів), лямбда-вирази тощо).

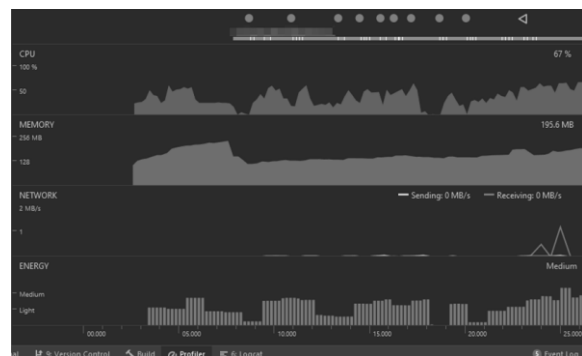


Рис. 5. Профільювальник застосунку в Android Studio

Щодо технічних вимог, то можна виокремити такі показники:

- вага застосунку;
- можливість доступу до native-функцій (безпроблемна взаємодія програми з мобільною ОС);
- швидкість роботи й відгуку інтерфейсу;
- швидкодія;
- продуктивність (споживання пам'яті, процесорного часу, вплив на витрату батареї, завантаження даних по мережі);
- надійність;
- безпеку;
- легкість підтримки / оновлення.

На основі порівняльної таблиці 3, а також якісних параметрів у межах роботи буде доцільно розглянути саме Xamarin і Android SDK. Для порівняння таких технічних критеріїв, як вага застосунку, швидкодія та продуктивність необхідно розробити два застосунки, які б задіяли native-функції, наприклад доступ до телефонної книги.

### Результати аналізу ефективності мобільних застосунків

Аналіз застосунку, розробленого на Android SDK.

Застосунок на Java / Kotlin розроблений в Android Studio, яка містить зручний вбудований профілювальник, що дозволяє відстежи-

ти всі основні метрики продуктивності: пам'ять, процесор, використання мережі й споживання енергії (див. рис. 6).

Застосунок займає 22,01 МБ пам'яті пристрою. Варто зауважити, що цьому застосунку не потрібен дозвіл на використання даних.

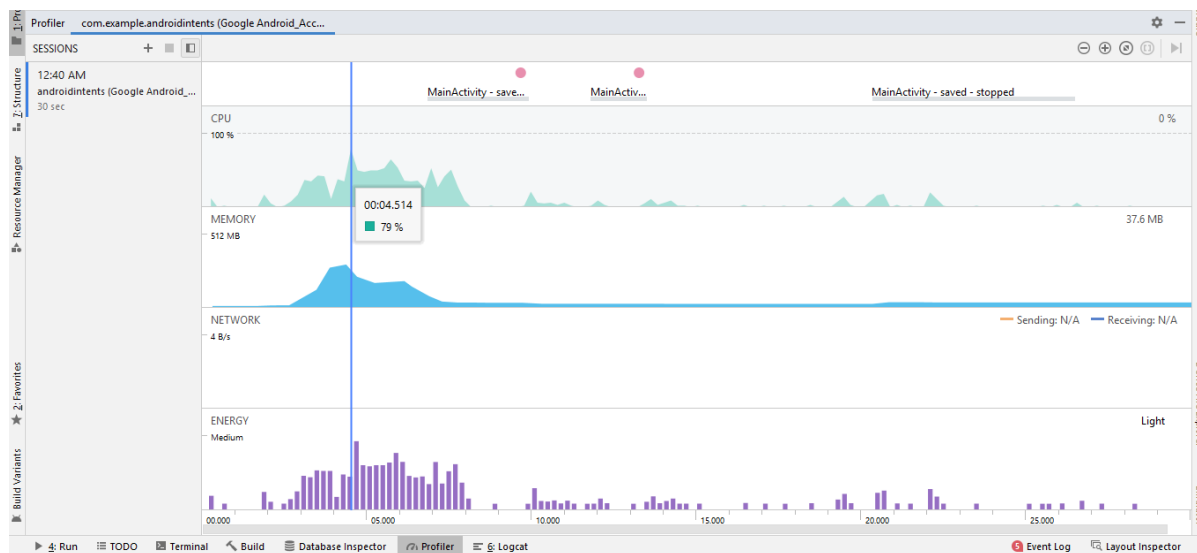


Рис. 6. Профілювання застосунку на Java

### Аналіз застосунку на C#

Для застосунків на Xamarin існує Xamarin Profiler – графічний інтерфейс для профілювання застосунків Android та iOS на Windows. Але щоб скористуватися ним, потрібно бути передплатником Visual Studio Enterprise.

Також Android Studio містить засіб Android Profiler. Його можна використовувати для вимірювання продуктивності застосунку

Xamarin Android, створеного за допомогою Visual Studio. Однак, на відміну від Xamarin Profiler, засіб Android Profiler не інтегровано в Visual Studio й може використовуватися лише для реєстрації APK, що було створено та імпортовано в Android Profiler.

Після завантаження APK-файлу Android Studio відображає інформацію про розмір застосунку та всіх його частин (див. рис. 7).

APK size: 15.2 MB, Download Size: 7.4 MB			
File	Raw File Size	Download Size	% of Total Download ...
assemblies	10.5 MB	3.5 MB	46.8%
lib	1.9 MB	1.9 MB	25%
classes.dex	1.7 MB	1.7 MB	22.5%
res	344.2 KB	304.3 KB	4%
resources.arsc	334.1 KB	69.7 KB	0.9%
META-INF	36.5 KB	35.7 KB	0.5%
typemap.mj	227.3 KB	14.3 KB	0.2%
typemap.jm	198.8 KB	14.2 KB	0.2%
AndroidManifest.xml	1.3 KB	1.3 KB	0%
environment	187 B	187 B	0%
NOTICE	121 B	121 B	0%

Рис. 7. Відображення інформації про розмір застосунку

Для того щоб отримати всі необхідні для аналізу показники, потрібно натиснути Profile 'назва\_арк' у верхньому лівому кутку

студії. Android Studio проаналізує застосунок, щоб відстежити всі основні метрики продуктивності (рис. 8).

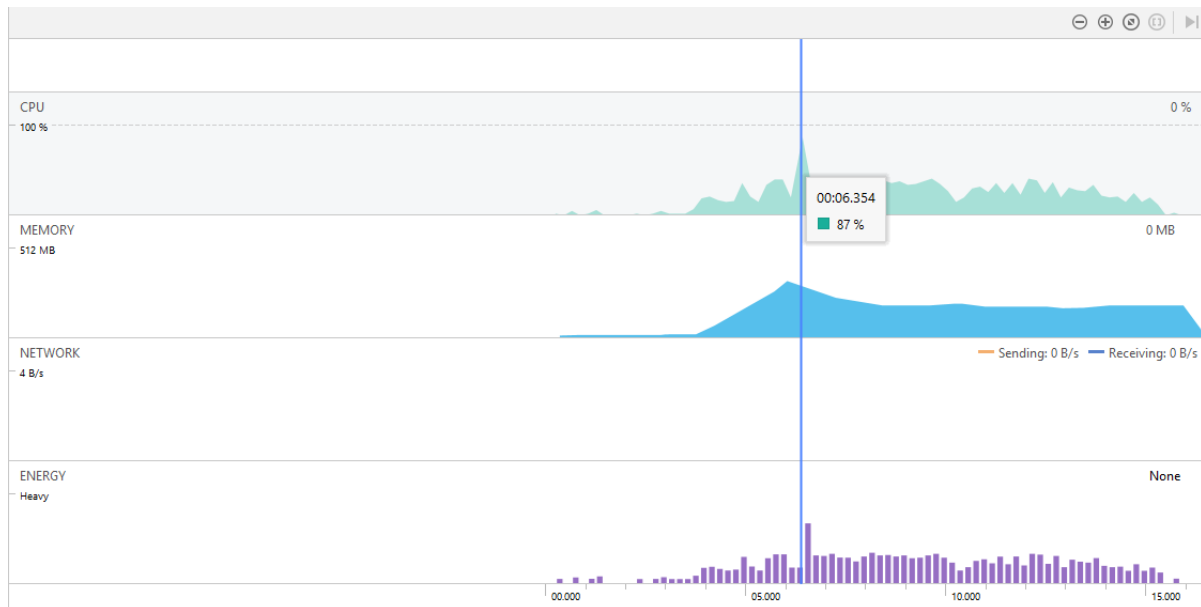


Рис. 8. Профілювання застосунку, розробленого на платформі Xamarin мовою C#

Варто зазначити, що застосунку на Xamarin необхідний дозвіл, щоб використовувати «Контакти пристрою», тоді як застосунок на Java просто використовує вже реалізовані можливості. Застосунок займає 28,78 МБ з пам'яті пристрою.

#### Аналіз застосунків

Застосунки встановлювалися й тестувалися на смартфоні з найменшою версією Android 6.0.1, версія прошивки ядра – Linux 3.10.49. Занесемо в таблицю 4 всі перелічені показники, а саме: розмір (size) застосунку, а також обсяг використовуваної пам'яті (memory Mb), процесорне завантаження та споживання енергії (рис. 6 і 8). Використання мережі залишилося стале в обох випадках. Energy Profiler відображає приблизне споживання енергії застосунком, що тестується.

Таблиця 4 – Порівняння результатів профілювання застосунків

	Застосунок на Xamarin (C#)	Застосунок на Java / Kotlin
Size (Mb)	28,78	22,01
CPU (%)	87	79
Memory (Mb)	304,7	273,2
Energy	00:04,513	00:01,434

Аналізуючи всі показники, можна зробити висновок, що використання native-технологій, а саме мови програмування Java / Kotlin і Android SDK, є найефективнішими в цьому дослідженні.

#### Висновки

У дослідженні розглянуті види та проведений аналіз типів мобільних застосунків і самої операційної системи Android. Крім того, розглянуто й проаналізовано технології для розроблення під Android. Також у роботі розглянуті основні компоненти архітектури мобільних застосунків для Android, що є блоками для побудови програми. За допомогою середовищ розробки Visual Studio і Android Studio було створено два застосунки, які можна встановити на мобільний пристрій та переглядати з нього інформацію про контакти користувача. Застосунки були розроблені мовами програмування Java і C#, використані Android SDK і Xamarin відповідно.

Java і C# є популярними мовами програмування, тому важко сказати, яка з них краща за такими якісними показниками, як наявність повної документації, технічна підтримка й наявність фахівців. Обидві ці мови є C-подібними, тому мають схожий зрозумілий синтаксис, зручний у розробленні й налагодженні. Розглядаючи такі технічні характеристики, як безпроблемна взаємодія програми з мобільною ОС, варто зазначити, що платформи, що використовувалися в розробці, мають повний доступ до основних native-функцій. Як особливість варто зазначити, що під час використання застосунку на C#, програма запитувала доступ на використання контактів, тоді як застосунок мовою Java дозволу не потребує.

Для того, щоб дати однозначну відповідь на питання, яка з мов має вищу швидкість роботи та відгуку інтерфейсу, виконуються



багато тестів у «реальних умовах», де перевіряється швидкість значних обчислень або активних запитів до бази даних. У цьому дослідженні два застосунки експериментально порівнювалися за такими показниками: вага додатків та їхня продуктивність. У процесі аналізу native-застосунків, розроблених мовою Java, показав кращі результати. Проте не можна зробити однозначного вибору на користь однієї технології, адже під час створення того чи іншого мобільного застосунку необхідно чітко визначати мету, що й буде керувати кінцевим вибором технологій.

Як висновок можна зазначити, що для найбільш ефективних результатів, за умов наявності необхідних навичок, бюджету й особливо часу, доцільно використовувати повністю native-платформу. Якщо ж необхідний застосунок для декількох ОС, а його функціонал не вимагає на кожній операційній системі дуже високої продуктивності, то логічніше було б використовувати кросплатформний фреймворк. Він би забезпечив максимальне покриття за короткий час, ніж native-технології для кожної операційної системи окремо.

### Література

1. Kotlin (programming language) // Wikipedia.: [https://en.wikipedia.org/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/Kotlin_(programming_language)) (дата звернення: 12.01.22).
2. Рейтинг мов програмування 2021: частка Python зменшується, а TypeScript обійшов C++ // ДОУ. URL: <https://dou.ua/lenta/articles/language-rating-jan-2021/> (дата звернення: 08.01.22).
3. Android. Разработка приложений / Р. Роджерс, Д. Ломбардо, З. Медникс, Б. Мейк. Москва: ЭКОМ Паблишерз, 2010. 400 с.
4. Что такое Xamarin? // Xamarin Microsoft. URL: <https://docs.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin>. (дата звернення: 10.12.21).
5. Янчук К. В., Федорченко В. М. Огляд платформ для створення мобільних додатків. Черкаси – Харків – Баку – Бельсько-Баяла, 2020. 64 с.
6. Wisnuadhi B., Munawar G., Wahyu U. Performance Comparison of Native Android Application on MVP and MVVM // *Advances in Engineering Research*. 2020. Vol. 198. P. 276–282.
7. Humeniuk V. Android Architecture Comparison: MVP vs. VIPER. 2019.
8. Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017 / Meulen R. van der // Gartner. URL: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017> (дата звернення: 16.12.21).
9. Mobile Operating System Market Share Worldwide // StatCounter Global Stats. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. (дата звернення: 19.11.21).
10. SDK Platform release notes // Android Developers. URL: <https://developer.android.com/studio/releases/platforms>. (дата звернення: 20.11.21).
11. Sillars D. High performance Android apps: improve ratings with speed, optimizations, and testing. Beijing; Boston: O'Reilly, 2015. 247 p.

### Reference

1. Kotlin (programming language) // Wikipedia. URL: [https://en.wikipedia.org/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/Kotlin_(programming_language)) (accessed: 12.01.22).
2. Programming language rating 2021: Python share decreases and TypeScript bypasses C++ // DOU. URL: <https://dou.ua/lenta/articles/language-rating-jan-2021/> (accessed: 08.01.22).
3. Android. Application Development / R. Rogers, D. Lombardo, Z. Mednieks, B. Meik. Moscow: ECOM Publishers, 2010. 400 p.
4. What is Xamarin? // Xamarin Microsoft. URL: <https://docs.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin> (accessed: 10.12.21).
5. Yanchuk K. V., Fedorchenko V. M. Review of platforms for creating mobile applications. Cherkasy – Kharkiv – Baku – Belsko-Bayala: 2020. 64 p.
6. Wisnuadhi B., Munawar G., Wahyu U. Performance Comparison of Native Android Application on MVP and MVVM // *Advances in Engineering Research*. 2020. Vol. 198. P. 276–282.
7. Humeniuk V. Android Architecture Comparison: MVP vs. VIPER. 2019.
8. Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017 / Meulen R. van der // Gartner. URL: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017> (accessed: 16.12.21).
9. Mobile Operating System Market Share Worldwide // StatCounter Global Stats. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed: 19.11.21).
10. SDK Platform release notes [Electronic Resource] // Android Developers. URL: <https://developer.android.com/studio/releases/platforms> (accessed: 20.11.21).
11. Sillars D. High performance Android apps: improve ratings with speed, optimizations, and testing. Beijing; Boston: O'Reilly, 2015. 247 p.

12. Tian Lou A Comparison of Android Native App Architecture – MVC, MVP and MVVM. Eindhoven University of Technology, 2016.

**Федорченко Володимир Миколайович**, к.т.н., доц. каф. електронних обчислювальних машин, каф. безпеки інформаційних технологій, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 099-640-52-78, Volodymyr.Fedorchenko@hneu.net,

**Поляков Андрій Олександрович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 099-4151200, Andrii.Poliakov@hneu.net,

**Сєверінов Олександр Васильович**, к.т.н., доц. каф. безпеки інформаційних технологій, Харківський національний університет радіоелектроніки, тел. +38 (066) 175-86-68, oleksandr.sievierinov@nure.ua.

#### Analyzing the efficiency of technologies for developing mobile applications for Android OS

**Abstract. Problem.** Mobile application development is quite a complex process that requires developers with high expertise in a particular development platform for Android. There are also several single-platform, cross-platform, Xamarin development platforms but there is no comparison of their performance as with native ones. **Methodology.** As a method, it is proposed to use an experimental approach in which a mobile application is developed with the necessary components, and then the performance of the profiler on the metrics of CPU usage, memory, power consumption and network is investigated. **Results.** According to the study, the Xamarin platform has additional restrictions on permissions,

requires more resources, both memory and power, which can be felt in large applications, but allows you to attract a large community of .Net developers.

**Originality.** This approach to the analysis of productivity will reveal at a preliminary stage the problematic aspects of the mobile development platform, its integration with native Android services, as well as possible user preferences. **Practical value.** The proposed technique will introduce a preliminary analysis of the various components integrated into the Android OS. This approach will provide a more reasonable management decision for the full process of developing mobile applications for Android OS, considering such criteria as speed of development, professional level of developers to be involved in the development process, application integration with Android components, integration of new functionality to the application and others.

**Key words:** Android OS, mobile platform, native platform, multi-platform development, cross-platform, Android profiler, Xamarin platform.

**Fedorchenko Volodymyr**, Ph.D., Assoc. Prof. Electronic Computers Department, and Information Systems Department, <sup>1</sup>Simon Kuznets Kharkiv National University of Economics, tel. +38 099-640-52-78, volodymyr.fedorchenko@nure.ua,

**Poliakov Andrii**, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 099-41-51-200, Andrii.Poliakov@m.hneu.edu.ua,

**Sievierinov Oleksandr**, Ph.D., Assoc. Prof. Department of Information Technology Security, Kharkiv National University of Radio Electronics, тел. +38 (066) 175-86-68, oleksandr.sievierinov@nure.ua.

УДК 004.4:658.3(6)

DOI: 10.30977/BUL.2219-5548.2022.96.0.91

## ПІДВИЩЕННЯ ЯКОСТІ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ШЛЯХОМ РОЗРОБЛЕННЯ Й ТЕСТУВАННЯ МЕТОДУ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ CRM-ПЛАТФОРМ НА БАЗІ ЗАСТОСУВАННЯ СИСТЕМИ SALESFORCE

Федорченко В. М.<sup>1</sup>, Федорченко Р. В.<sup>2</sup>, Поляков А. О.<sup>1</sup>

<sup>1</sup> Харківський національний економічний університет ім. С. Кузнеця

<sup>2</sup> ІТ-компанія NIX Solutions LTD

**Анотація.** Стаття підсумовує результати проведеного авторами дослідження щодо пошуку шляхів удосконалення процесу автоматизації роботи підприємства з допомогою використання передових розробок у галузі програмного забезпечення. Доцільною формою підвищення ефективності автоматизації бізнес-процесів було визначено впровадження cloud CRM платформ, побудованих на базі віртуальних ресурсів компаній вендорів, і віртуалізації бізнес-процесів. Розроблено та обґрунтовано заходи щодо вдосконалення інформаційно-аналітичної системи для управління організацією.

**Ключові слова:** CRM, управління взаємодією з клієнтами, інформаційні системи, KPI, порівняльний аналіз, Salesforce.

### Вступ

Для сучасного підприємства успішність діяльності залежить від ефективності впровадження інформаційних технологій і систем. Автоматизація бізнес-процесів дозволяє оптимізувати бізнес-функції діяльності підприємства й створити сучасну систему управління підприємством із застосуванням інформаційних технологій.

Для оцінювання ефективності автоматизації бізнес-процесів застосовують низку методик і моделей. Ефективність від впровадження інформаційних систем визначають за допомогою динамічних показників: сукупна вартість володіння (ТСО), коефіцієнт оцінки повернення інвестицій (ROI), чиста приведена вартість (NPV), середньозважена вартість капіталу (WACC).

Варто зауважити, що малі та середні підприємства (МСП) відіграють вирішальну роль в економічному розвитку України. Зростання кількості малих і середніх підприємств спричиняє жорсткішу конкуренцію на ринках [1]. Це змушує підприємства стикатися з постійним зовнішнім тиском від конкурентів. Малому та середньому бізнесу приділяється менше уваги в плані оцінювання його ефективності, на відміну від вимірювання ефективності у великих компаніях. Отже, вони потребують певної підтримки для утримання та збільшення їхніх бізнес-досягнень. Особливо, коли йдеться про впровадження ІТ-продукту, що вимагає певних експертних навичок від фахівця [2].

Більшість малих і середніх підприємств не можуть дозволити собі наймати штат кваліфікованих фахівців із забезпечення, розроблення та налаштування системи управління ІТ-продуктами. Хоча насправді така технологія може бути дуже корисною для них, у межах зменшення витрат на оплату праці та підвищення ефективності бізнес-процесів. Дослідження впровадження SaaS-технологій на великих підприємствах проводяться вже кілька десятиліть, на відміну від його застосування в межах малих і середніх підприємств.

Поява сучасних хмарних обчислень і пов'язаного з ними програмного забезпечення як концепції сервісу може змінити всю ситуацію. Однак відсутність методологій, інструкцій та інструментів, які могли б допомогти організаціям з їхніми ІТ-стратегіями, аутсорсингом та управлінням ІТ-послугами, разом із випадковими або необґрунтованими рішеннями, можуть спричинити серйозні ускладнення щодо якості, зручності використання та інтеграції, що в підсумку впливають на загальну вартість послуг або продуктів компанії.

Саме з метою пошуку новітніх шляхів на базі комплексного дослідження теоретичних і практичних аспектів підвищення якості автоматизації бізнес-процесів за допомогою CRM проаналізовано ефективності використання традиційних засобів ведення бізнесу порівняно з перспективами комп'ютерної віртуалізації бізнес-процесів.

Результати теоретичних і емпіричних досліджень показали необхідність отримання та дослідження позитивних результатів (зростання продуктивності, розширення організації, ефективності, результативності, конкурентоспроможності тощо) впровадження технологій у різних організаціях.

### Аналіз публікацій

Будь-який інвестиційний проєкт потребує чіткого та прозорого обґрунтування своєї необхідності та життєздатності. Кожна ідентифікована бізнес-проблема, визначене бізнес-завдання, як правило, мають декілька способів успішного вирішення. Перед керівництвом фірми та її персоналом постає проблема вибору найбільш ефективного способу із запропонованих шляхом його обґрунтування.

Проблеми у сфері взаємовідносин із клієнтами та виявлення пріоритетів їхнього усунення можуть бути вирішені шляхом використання CRM-системи. Проблематика впровадження CRM-систем на підприємствах уже тривалий час вивчається як вітчизняними, так і зарубіжними фахівцями. Під час аналізу використання CRM-систем були використані дослідження Г.В. Мозгова, Т.А. Петросян, а також аналітиків дослідницького ресурсу ARTW та компанії Salesforce. Salesforce – це платформа, що повністю хоститься на серверах компанії Salesforce в клауді.

Salesforce зі свого боку дозволяє створювати та розгортати індивідуальні рішення, автоматизувати бізнес-процеси, інтегруватися із зовнішніми застосунками. Більшість світових ентерпрайз-компаній є клієнтами компанії Salesforce і використовують цю платформу як рішення для своїх бізнес-потреб.

Salesforce, як компанія, є партнером різноманітних представників ІТ-світу і, відповідно, вони пропонують готові інтеграційні рішення, удосконалені сервіси для задоволення вимог кінцевих клієнтів і ще багато різних можливостей. Серед них – Apple, Microsoft, Google, Amazon. Станом на 2019 р. Salesforce пропонує цілий набір продуктів: Sales Cloud, Service Cloud, Marketing Cloud, e-Commerce Cloud, Heroku, Integration, Community Cloud, Einstein Analytics тощо [3].

Проте як компанії вимірюють зиск від впровадження цих систем? [4] Показники ефективності КРІ – інструмент вимірювання визначених цілей. Якщо показник не пов'язаний з метою, тобто не утворюється,

виходячи з її змісту, тоді не можна використовувати цей показник як КРІ. Технології постановки, перегляду й контролю цілей і завдань лягли в основу концепції, яка стала підґрунтям сучасного управління і називається «управління по цілях».

Дослідженню теоретико-методологічних підходів щодо запровадження КРІ присвячено наукові праці таких іноземних учених: Б. Марр, Р. Каплан, Д. Нортон, Д. Парментер, Р. Фрейзер, Д. Хоуп та ін. Проте існує необхідність у дослідженні особливостей запровадження зазначених підходів підприємствами України в сучасних умовах. Термін КРІ (Key Performance Indicator) з англійської мови найчастіше перекладається як «ключовий показник ефективності», що не є цілком коректним, оскільки ефективність характеризує співвідношення між досягнутим результатом і витраченими ресурсами, а за допомогою КРІ можна вимірювати й інші параметри / показники процесів певної діяльності або досягнення конкретної мети. Тому вважаємо, що правильнішим є переклад «ключовий показник діяльності». Також існує хибна думка про те, що КРІ має пряме відношення до BSC (Balanced Scorecard, тобто «збалансована система показників»).

Проте Д. Нортон і Р. Каплан – автори публікацій щодо збалансованої системи показників – не використовували термін КРІ, а застосовували поняття «measure» – «міра», «вимірник». Тому між КРІ і BSC можна побачити лише непрямий зв'язок, у якому BSC описує перспективу «бізнес-процесів», на якій знаходяться цілі, пов'язані з ними. А от вимірниками досягнення цілей можна використовувати показники цих бізнес-процесів – КРІ. Отже, КРІ – показники, за якими оцінюється ефективність процесів / дій, функцій управління щодо досягнення поставлених цілей.

Для діагностики результатів діяльності організації необхідно обирати обмежений перелік найважливіших показників, який стане основним переліком КРІ, за яким керівництво буде ухвалювати поточні й стратегічні управлінські рішення. Показники мають відображати найважливішу інформацію для ефективного управління організацією, процесами її управління, основними процесами та процесами, що забезпечують основні, а також управління результатами діяльності підрозділів і працівників. Сучасні організації використовують безліч КРІ, що належать до різних сфер управління бізнес-процесами.

Вони можуть характеризувати як загальні моменти, єдині для всіх працівників організації, так і специфічні норми праці й трудової поведінки, властиві для конкретної посади чи конкретного робочого місця.

#### Мета та постановка завдання

Проведений аналіз довів перевагу комп'ютерних Customer relationship management (CRM) систем перед традиційними способами ведення бізнесу та управління діловою взаємодією з клієнтами. Це уможливило створення віртуальних комп'ютерних застосунків на базі CRM-платформ та інтерактивних систем дистанційного управління.

Актуальність і прикладна значущість такого напряму полягає в тому, що впровадження програмних засобів, призначених для автоматизації вирішення завдань управління взаємовідносинами з клієнтами, за даними різних джерел, дозволяє [5]:

- 1) знизити адміністративні витрати на 10–20 %;
- 2) збільшити обсяги продажів на 10–30 % за рік на одного менеджера;
- 3) збільшити кількість укладених угод на 5–15 %.

Загалом, ефективність CRM можна оцінити як рівень задоволеності, досягнутий впровадженням та використанням CRM. Хоча CRM стала основною бізнес-стратегією для електронної комерції, проте в Україні було проведено досить мало досліджень щодо оцінювання ефективності CRM. Оскільки важко продемонструвати відчутну віддачу від ресурсів, вкладених у планування, розроблення, впровадження та експлуатацію CRM, мета цього дослідження полягає у виявленні та вимірі атрибутів переваг впровадження CRM, таких як, наприклад, підвищення вартості, ефективність, інновації та покращення якості послуг, кількість постійних клієнтів. У межах цього дослідження запропоновано та описано орієнтовану на клієнта модель для оцінювання ефективності CRM, а потім ілюстровано її за допомогою тематичного дослідження.

Загалом, інформатизація всіх сфер діяльності суспільства принципово зміцнює роль інформації та інформаційних технологій. Особливо актуальні ці питання для підприємств, управління якими – постійний процес переробки значних обсягів різноманітної інформації.

На сьогодні інформаційна сфера набуває

системоутворювального значення для функціонування підприємств. Це зумовлено тим, що інформація стає стратегічним ресурсом, який забезпечує підприємству конкурентну перевагу. Згідно із сучасними поглядами, інформаційні системи матимуть значно більше значення не лише в досягненні тактичних, але й стратегічних цілей компанії [6].

Ефективність ведення бізнесу визначається тим, яку інформацію використовує управлінець і як він нею розпоряджається. Витрати на збір інформації, її передачу, оброблення, зберігання та передачу користувачеві мають бути мінімальними. Технічний прогрес у галузі інформаційних технологій та систем за останні роки зробив розв'язання цієї проблеми цілком можливим і доступним.

Цьому сприяло впровадження в управління комп'ютерної техніки та новітніх інформаційних технологій, через те дуже важливо вичерпно оцінити ефективність впровадження тих чи інших інформаційних систем, обрати критерії їхньої ефективності. Вибір критерію ефективності – це центральний, найвідповідальніший момент дослідження системи. Критерій ефективності формується на основі показників, що кількісно описують (мета або цілі) прийняття рішення.

Об'єктом дослідження статті є процес вибору й обґрунтування показників і критеріїв ефективності для оцінки CRM-застосунків.

Предметом є методи, технології та засоби проведення аналізу показників ефективності автоматизації бізнес-процесів на підприємстві за допомогою CRM-застосунків, дослідження та тестування комбінації методів визначення впливу впровадження CRM-системи на підприємстві.

Завдання – аналіз підходу до вирішення питання визначення ефективності впливу впровадження CRM-системи Salesforce як SaaS-технології на підприємстві.

Метою цієї роботи є аналіз, дослідження та тестування теоретико-методичних основ та надання характеристики критеріїв підвищення ефективності проведення автоматизації бізнес-процесів за умов використання CRM-технологій на підприємстві, а також побудова моделі оцінювання ефективності впровадження інформаційних систем даного типу на підприємствах.

Для ефективного впровадження та аналізу доцільності CRM необхідно враховувати діяльність компанії, виявити її особливості, визначити основні критерії показників ефективності та обрати найкращу стратегію, яка б

задовольнила потреби компанії та мала найкращий результат.

Ураховуючи викладене, актуальним є з'ясування сутності інформаційних CRM-систем підприємства з погляду їхнього впливу на загальну ефективність діяльності організації та визначення найвагоміших напрямів удосконалення цих систем, реалізація яких дозволить створювати дієві інноваційні інформаційні системи на базі впровадження передових методів організації інформаційного забезпечення та прогресивних інформаційних технологій.

### Виклад основного матеріалу

Найкращий спосіб дізнатися, наскільки вагомими є внесок і важливість CRM для компанії, – це вимірювання рівня використання системи для задоволення бізнес-потреб (кількість бізнес-процесів, що можуть бути автоматизовані за допомогою CRM) та ефективності цієї автоматизації. Існує низка методів, що використовуються для вимірювання продуктивності CRM та для задоволення потреб клієнтів та компанії.

Прикладом цього можуть бути такі застосунки та методи: Body Check, CRM Scorecard, CRM Maturity Model, Method of qualification of development alability, CRACK Model, paired T-test тощо. Більшість із них побудовані на основі складних структур анкет та їхньої подальшої об'єктивної оцінки. Метою цього дослідження є створення методології для вимірювання рівня ефективності впровадження саме Salesforce CRM [7]. Для цього було вирішено спиратися на концепцію CRM, орієнтовану на стратегію роботи компанії, оскільки майже неможливо створити загальну методологію, що підходила б для кожного конкретного випадку. З іншого боку, можна розробити базову конструкцію системи вимірювання та методу процедур, які можна було б ефективно адаптувати до вимог будь-якої компанії.

Вважається, що система вимірювання ефективності CRM має пов'язувати довгострокові бачення, стратегію та цілі у використанні CRM із конкретними короткостроковими тактичними, дійовими та оцінними планами, створюючи рушійну силу для впровадження систем цього типу. Тому було вирішено провести порівняння простим способом, тобто порівнюватимуться мінімальні, реальні та оптимальні значення, отримані в процесі визначення відповідних значень CRM, і щодо компанії та ключових клієнтів.

Під час дослідження, аналізуючи окремі методи, важливо визначити, які змінні у сфері управління відносинами з клієнтами компанія хоче досліджувати, і відповідно до цього їй необхідно вирішити, який метод вона використовуватиме під час вимірювання ефективності CRM. На основі проведеного аналізу й порівняння окремих методів було вирішено використовувати для вимірювання рівня та продуктивності CRM метод Customer Relation Analysis Complex Kit, який було застосовано під проведення дослідження (до обраної компанії та її ключових клієнтів) [8].

Оцінювання продуктивності використання CRM було поділено на такі основні частини: визначення контрольованих змінних, критеріїв, розрахунок важливості окремих, оцінювання з боку компанії, розрахунок комплексного показника рівня CRM, оцінка з погляду клієнтів, конфронтація, або порівняння отриманих висновків з оцінки, зробленої компанією та клієнтами [9].

З огляду на вищезазначене необхідно було ретельно продумати вибір конкретних критеріїв, які будуть предметом моніторингу. Обираючи, варто було враховувати такі фактори:

- перспективи компанії на майбутнє в основних сферах, а не тільки в минулому й теперішньому стані;

- контроль внутрішніх процесів та діяльності компанії, що тісно впливають на подальшу поведінку та ставлення клієнтів, а не зосередження лише на поведінці та ставленні останніх до самої компанії.

Вибір критеріїв завжди здійснювався на основі обговорення з представниками відповідних відділів, а саме відділів маркетингу, бізнесу, логістики, виробництва, а також з директором компанії [10]. Були обговорені всі критерії, викладені в моделі, але для дослідження були обрані лише ті, які можна кількісно проаналізувати та оцінити з погляду конкретної компанії. Наприклад, були обрані та оцінені такі критерії: сфера «Маркетинг – бренд», сфера «Клієнти», сфера «Послуги», сфера «Управління процесом надання послуг / товарів», сфера «Комплексні показники».

Наступний крок (після вибору відповідних показників для вимірювання та оцінювання рівня ефективності CRM) полягав у необхідності встановлення ваги цих критеріїв. Це потрібно через те, що не кожен критерій має однакову важливість порівняно з

іншими. Перед установленням ваги кожного критерію були визначені переваги між окремими сферами. Метод кількісного порівняння критеріїв використаний для встановлення переваг і подальшого розрахунку ваг.

На цьому кроці були встановлені вагові коефіцієнти окремих критеріїв з усіх обраних сфер. Після встановлення ваг для окремих областей та критеріїв необхідно було «об'єктивувати» ці вагові показники. Річ у тому, що потрібно перерахувати значення ваг таким чином, щоб їхня кількість дорівнювала одиниці (адже вага всіх наявних критеріїв – це 100 %, тобто 1).

Тільки так можна використовувати вагові коефіцієнти в подальшому оцінюванні продуктивності CRM. Перерахунок ваг проводиться таким чином, що кожна з критеріальних ваг помножувалась на значення ваги певної сфери [11].

Наступним важливим кроком для оцінювання ефективності CRM є отримання значень для окремих критеріїв. Зазвичай значення вагомості отримують від співробітників, відповідальних за підтримку та впровадження CRM, але в цьому випадку не було змоги визначити конкретну особу, яка брала б на себе цю відповідальність. Тому було вирішено отримати оцінки від більшої кількості представників компанії. Кожному оцінювачу була надана анкета, яка містила деяку кількість запитань, пов'язаних з окремими критеріями оцінювання таким чином, щоб можна було отримати значення для кожного з них. Власне, у запропонованому анкетуванні йшлося про визначення важливості (0–10) або відсоткового значення (0–100 %) наявних критеріїв.

Кожен із представників компанії оцінював усі критерії незалежно від того, входить певна сфера безпосередньо до його компетенції чи ні. Однак для досягнення об'єктивності кінцевого значення показників було вирішено встановити ваги окремих підрозділів під час відповіді на конкретні запитання. Цим кроком було виявлено, що, наприклад, критерії у сфері маркетингу є основною компетенцією представника цього відділу, але також певну вагому частку мають бізнес-відділ та директор компанії.

На основі отриманих значень згодом було розроблено середнє зважене та отримано рівень ефективності впровадження CRM у відсотках (тоді як 100 % було прийнято як теоретичний ідеальний стан системи після автоматизації всіх сфер роботи компанії).

Для розрахунку середньозваженого було використано таке співвідношення:

$$S_v = \frac{p_1 \text{per} \cdot w_1 + p_2 \text{per} \cdot w_2 + \dots + p_i \text{per} \cdot w_i}{w_1 + w_2 + \dots + w_i} = \frac{\sum_{i=1}^k p_i \text{per} \cdot w_i}{\sum_{i=1}^k w_i}, i = 1, 2, \dots, k,$$

де  $S_v$  – середньозважене;  $p_i \text{per}$  – значення критерію у відсотках;  $w_i$  – вага окремих критеріїв;  $k$  – кількість критеріїв.

Наприклад, якщо отримане значення середньозваженого становить 75,35 % – це означає, що рівень дієвості впровадження CRM-системи в спостережуваній компанії становить 75,35 %. Варто зауважити, що для досягнення теоретичного ідеалу йому не вистачає лише 24,65 відсоткових пунктів. На основі цього процесу та за отриманим значенням рівня дієвості CRM можна констатувати, що компанія розташована на достатньо гарному рівні за значенням дієвості впровадження CRM, хоча відстань від «ідеалу» є передумовою для подальшого покращення здебільшого шляхом ретельного аналізу досліджуваних сфер і критеріїв. Це впливає з уже згаданої та частково використованої моделі CRACK, а також із консультацій із представниками компанії, що досягти ідеального 100 % значення більш-менш нереально.

Тому було вирішено встановити оптимальні значення, які б відображали рівень окремих сфер і критеріїв, що є свого роду реальним максимумом для певної компанії. За умови досягнення цих значень компанія буде задоволена, оскільки буде досягнутий необхідний запланований рівень. З іншого боку, важливо встановити також так звані «мінімальні» значення. Це створило б якусь критичну межу як для сфер, так і для окремих критеріїв. У разі досягнення встановлених мінімальних значень або якщо значення будуть ще нижчими, компанії потрібно буде вжити швидких заходів, щоб змінити погану ситуацію, яка може серйозно поставити її продуктивність під загрозу, особливо відносно з ключовими клієнтами.

Згадані оптимальні та мінімальні значення встановлювалися окремими підрозділами, проте варто пам'ятати, що кожен відділ має різні компетенції. Для зменшення впливу цього фактора отримані значення було помножено на ваги окремих підрозділів для кожного критерію. Ці значення дають змогу для

швидкого та простого порівняння, де розміщується та аналізується реальний рівень розрахованої ефективності впровадження та використання CRM порівняно з оптимальним та мінімальним рівнями. На основі цих результатів можливо визначити, де розміщений поточний рівень CRM у проміжку між оптимальним і мінімальним рівнем [12]. З визначення цієї відстані впливає, чи поточний рівень CRM ближчий до оптимального та чи існує відсоток, що показує простір для покращень.

Хоча компанія вже встановила свій поточний рівень дієвості від упровадження CRM, він оцінюється лише зсередини. Оскільки основною метою впровадження CRM є підвищення ефективності управління відносинами з клієнтами, важливо з'ясувати, як CRM компанії сприймається її клієнтами. Насамперед CRM орієнтована на найважливіших клієнтів, і тому, виявляючи сприйняття CRM клієнтами, було вирішено звертатися тільки до тих, хто визначений керівництвом як ключовий і, отже, найважливіший і найкомпетентніший клієнт.

Для розрахунку та визначення ефективності CRM-системи з погляду клієнтів було використано ту саму процедуру, що й у випадку самої компанії. Для забезпечення оптимальної порівнянності оцінок зберігалися однакові критерії та індивідуальні значення, отримані від клієнтів за допомогою анкет.

Однак для забезпечення більш точного «погляду клієнта» всім діловим партнерам було запропоновано висловити свої переваги порівняно з окремими критеріями, а також окремими сферами, що містять ці критерії.

Як і у випадку з оцінюванням з боку компанії, матриця оцінок клієнтів також була встановлена на основі уподобань. Через них було розраховано ваги кожної сфери, а також ваги окремих критеріїв. Остаточна вага критеріїв була встановлена так само, як і у випадку компанії, на основі множення їхньої ваги на вагу сфери, до якої вони належать. Остаточні ваги були використані в подальших розрахунках, що виконувалися для визначення рівня сприйняття CRM клієнтами.

Для розрахунку рівня доцільності використання CRM з погляду клієнтів було застосовано середньозважене значення так само, як і для розрахунку з погляду компанії. За цими розрахунками виведено значення рівня ефективності CRM, сприйнятого клієнтами, що становить, наприклад, 77,22 %, що порівняно з рівнем значущості CRM з погляду підприємства (75,35 %) навіть трохи вище (на 1,87 %), з чого можна припустити, що в загальному масштабі сприйняття клієнтів знаходиться на тому самому рівні або навіть трохи краще, ніж погляд компанії на свої внутрішні автоматизовані бізнес-процеси [13].

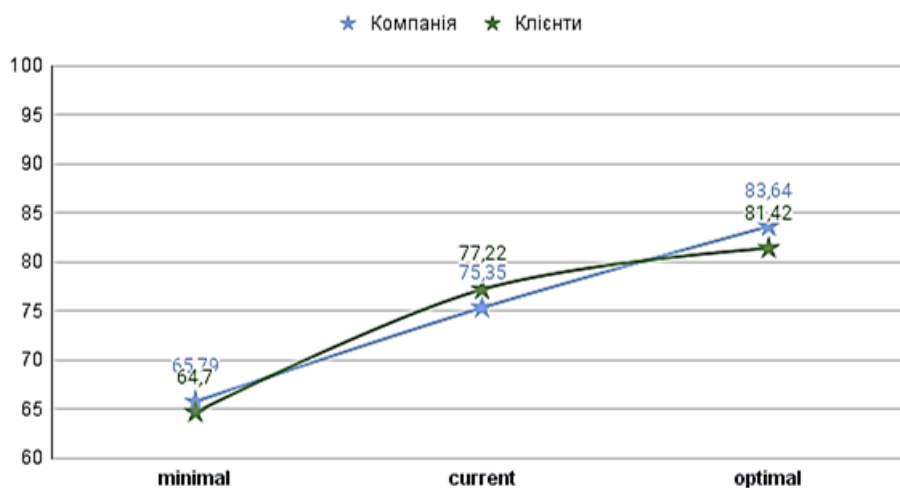


Рис. 1. Графічне відображення порівняння розрахованого рівня ефективності від упровадження CRM-системи на підприємстві з погляду підрозділів компанії та її клієнтів

Виходячи з того, що було отримано подібні результати від оброблення обох підходів, тобто від компанії та від клієнтів, можна констатувати, що відсоток ефективності й задоволеності від використання CRM у компанії розташований на хорошому рівні, тому

що для досягнення теоретичного максимуму йому не вистачає приблизно 25 %, що не так багато. Але навіть ця різниця створює простір для вдосконалення і здебільшого не у внутрішньому процесі роботи підприємства, а з погляду взаємодії з клієнтами.



### Висновки

З викладених у статті даних для порівняння видно, що загалом вимоги клієнтів до CRM нижчі, ніж у компаній. Здебільшого можна констатувати, що ця ситуація є вигідною для компанії. Насправді, мінімум оцінки CRM-системи клієнтами нижчий від встановленого компанією мінімуму, і тому останній достатньо досягти рівня свого мінімуму, і вона все одно буде в зоні, коли вимоги ключових клієнтів будуть задоволені.

З іншого боку, нижчий показник цінності оптимізму клієнта порівняно з компанією може свідчити про те, що компанія легше матиме змогу досягти вартості, яку очікують клієнти, а в разі досягнення власного оптимізму навіть перевершить їхнє очікування від взаємодії із CRM-системою. Можна зробити висновок, що в якийсь момент часу задоволеність клієнтів перевищить рівень, який вони вважають реальним максимумом. Позитивним є те, що поточне значення рівня ефективності роботи CRM, на думку клієнтів, вище, ніж те, як його сприймає компанія.

Отже, у статті:

- виконувалося дослідження, аналіз і тестування теоретико-методичних основ та надання характеристики критеріїв підвищення ефективності проведення автоматизації бізнес-процесів за умов використання CRM-технологій на підприємстві;

- визначено актуальність поставленого завдання, обґрунтовано доцільність використання CRM-технологій на підприємстві, що дозволяє забезпечити надійне підґрунтя для використання подальших досліджень у цій галузі.

Також було проведено аналіз та розглянуто основні показники KPI [14] у впровадженні CRM-системи Salesforce, побудовано модель оцінювання ефективності впровадження CRM-Salesforce: модель базується на розподілі функціоналу системи на кілька основних груп (схоже до випадку із порівнянням різних CRM-систем) та розрахунку важливості кожної з груп, а також наявних у ній підгруп (за допомогою вагових коефіцієнтів, наданих як працівниками компанії, так і її клієнтами). Після проведення тестувань було визначено що в першому випадку Salesforce CRM перевищує очікування своїх користувачів на 3 %, що дає компанії змогу значно покращити ефективність взаємодії із своїми клієнтами [15].

Однак важливо також те, що поточний рівень дієвості CRM необхідно підвищити на

1,33 %, аби мати змогу досягти очікуваного від системи максимуму. Знання важливості окремих критеріїв для клієнтів, а також їхніх очікувань у виконанні кожного з них є чудовою передумовою для досягнення цього оптимального рівня ефективності CRM, до того ж не тільки з погляду кастомерів, але й з погляду самої компанії.

Дослідження, проведене в межах написання цієї публікації, дозволяє стверджувати, що впровадження Salesforce CRM на малих і середніх підприємствах може забезпечити їм реінжиніринг управління базою даних, що містить облік потенційних клієнтів, обліковий запис, контакти, угоди, планувальник завдань і зустрічей, звіт і аналіз.

### Література

1. Jyoti D., Hutcherson J. A. Salesforce Architect's Handbook: A Comprehensive End-to-End Solutions Guide. Apress, 2021. 412 p.
2. KPI Examples and Templates // Klipfolio.com. URL: <https://www.klipfolio.com/resources/kpi-examples> (accessed: 17.11.21).
3. Reinartz W., Krafft M., Hoyer W. The Customer Relationship Management Process: Its Measurement and Impact on Performance // Journal of Marketing Research – J MARKET RES-CHICAGO, 2004. Vol. 41. P. 293–305.
4. Bahrami M., Naderi A., Arabzad S. M. Introducing Proposed Framework for Effective Use of Information Technology on Customer Relationship Management // The 9th International Conference on e-Business (iNCEB2010), November 18th – 19th, 2010–2011. 75–81 p.
5. Юрчук Н. П. CRM-системи: Особливості функціонування та аналіз українського ринку // Науковий вісник Ужгородського національного університету. Серія: Міжнародні економічні відносини та світове господарство. 2019. No. 23(2). P. 141–147.
6. Key Performance Indicator Creation Process Overview: SCA Metrics and Durations Taskforce URL: <https://www.slideshare.net/Handsomea/enterprise-kpidevelopment-process>.
7. Mohamed Y., Mahmood A. K., Almomani M. A., Alkaws G. Customer relationship management (CRM) in healthcare organization: A review of ten years of research // 2016 3rd International Conference on Computer and Information Sciences. 2016. 97–102 p.
8. Khabala T., Mang'anyi E., Govender K., Rajkaran S. Exploring the e-CRM – e-customer – e-loyalty nexus: a Kenyan commercial bank case study // Management & Marketing. 2017. Vol. 12. P. 674–696. DOI:10.1515/mmcks-2017-0039
9. Kim J., Suh E., Hwang H. A model for evaluating the effectiveness of CRM using the balanced

- scorecard // Journal of Interactive Marketing. 2003. Vol. 17, No. 2. P. 5–19. URI: <https://www.sciencedirect.com/science/article/abs/pii/S1094996803701315>.
10. Малукіна А. О. Обґрунтування впровадження CRM-системи для удосконалення процесу прийняття управлінських рішень на підприємстві // Науковий вісник Полісся. 2015. Вип. 1 (1). С. 85–90.
  11. Measuring Adoption: Is your CRM solution healthy and growing? // Salesforce.org. URL: <https://www.salesforce.org/blog/measuring-adoption-is-your-crm-solution-healthy-and-growing/> (accessed: 17.11.21).
  12. Легкий О. А., Сохацька О. М. Ефективність цифрових маркетингових комунікацій: від постановки мети до оцінювання результату // Маркетинг і цифрові технології. 2017. Т. 1. № 2. С. 4–31.
  13. Centobelli P., Cerchione R., Esposito E., Raffa M. Digital Marketing in Small and Medium Enterprises: The Impact of Web-Based Technologies // Advanced Science Letters. 2016. Vol. 22, No. 5–6. P. 1473–1476.
  14. Salesforce Help. URL: [https://help.salesforce.com/s/?language=en\\_US](https://help.salesforce.com/s/?language=en_US).
  15. The CRM Metrics: How to measure the performance of CRM in 2021 – Reviews, Features, Pricing, Comparison // PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices. URL: <https://www.predictiveanalyticstoday.com/the-crm-metrics-how-to-measure-the-performance-of-crm/> (accessed: 23.10.21).

### References

1. Jyoti D., Hutcherson J. A. Salesforce Architect's Handbook: A Comprehensive End-to-End Solutions Guide. Apress, 2021. 412 p.
2. KPI Examples and Templates // Klipfolio.com. – Access Mode: URI: <https://www.klipfolio.com/resources/kpi-examples> (accessed: 17.11.21).
3. Reinartz W., Krafft M., Hoyer W. The Customer Relationship Management Process: Its Measurement and Impact on Performance // Journal of Marketing Research – J MARKET RES-CHICAGO. 2004. Vol. 41. P. 293–305.
4. Bahrami M., Naderi A., Arabzad S. M. Introducing Proposed Framework for Effective Use of Information Technology on Customer Relationship Management // The 9th International Conference on e-Business (iNCEB2010), November 18th – 19th, 2010–2011. 75–81 p.
5. Yurchuk N. P. CRM-Systems: Peculiarities of Functioning and Analysis of the Ukrainian Market // Scientific Bulletin of Uzhgorod National University. Series: International Economic Relations and the World Economy. 2019. № 23 (2). P. 141–147.
6. Key Performance Indicator Creation Process Overview: SCA Metrics and Durations Taskforce URL: <https://www.slideshare.net/Handsomea/enterprise-kpiddevelopment-process>.
7. Mohamed Y., Mahmood A. K., Almomani M. A., Alkaws G. Customer relationship management (CRM) in healthcare organization: A review of ten years of research // 2016 3rd International Conference on Computer and Information Sciences. 2016. P. 97–102.
8. Khabala T., Mang'unyi E., Govender K., Rajkaran S. Exploring the e-CRM – e-customer – e-loyalty nexus: a Kenyan commercial bank case study // Management & Marketing. 2017. Vol. 12. P. 674–696. DOI:10.1515/mmcks-2017-0039
9. Kim J., Suh E., Hwang H. A model for evaluating the effectiveness of CRM using the balanced scorecard // Journal of Interactive Marketing. 2003. Vol. 17, No. 2. P. 5–19. URI: <https://www.sciencedirect.com/science/article/abs/pii/S1094996803701315>.
10. Malyukina A. O. Justification of the Implementation of CRM-System for Improving the Management Decision-Making Process at the Enterprise // Scientific Bulletin of Polissya. 2015. No 1(1). P. 85–90.
11. Measuring Adoption: Is your CRM solution healthy and growing? // Salesforce.org. URL: <https://www.salesforce.org/blog/measuring-adoption-is-your-crm-solution-healthy-and-growing/> (accessed: 17.11.21).
12. Lehky O. A., Sohatskaya O. M. Digital marketing communications effectiveness: from target setting to result evaluation // Marketing and digital technologies. 2017. Vol. 2. P. 4–31.
13. Centobelli P., Cerchione R., Esposito E., Raffa M. Digital Marketing in Small and Medium Enterprises: The Impact of Web-Based Technologies // Advanced Science Letters. 2016. Vol. 22, No. 5–6. P. 1473–1476.
14. Salesforce Help. URL: [https://help.salesforce.com/s/?language=en\\_US](https://help.salesforce.com/s/?language=en_US).
15. The CRM Metrics: How to measure the performance of CRM in 2021 – Reviews, Features, Pricing, Comparison // PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices. URL: <https://www.predictiveanalyticstoday.com/the-crm-metrics-how-to-measure-the-performance-of-crm/> (accessed: 23.10.21).

**Федорченко Володимир Миколайович**, к.т.н., доц. каф. електронних обчислювальних машин, каф. безпеки інформаційних технологій, Харківський національний університет радіоелектроніки, тел. +38 099-640-52-78, [Volodymyr.Fedorchenko@hneu.net](mailto:Volodymyr.Fedorchenko@hneu.net),

**Федорченко Руслана Володимирівна**, магістр комп'ютерних наук, NIX Solutions LTD, тел. +38 063-325-46-97,

[Ryslana.Fedorchenko@hneu.net](mailto:Ryslana.Fedorchenko@hneu.net),

**Поляков Андрій Олександрович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця,

тел. +38 099-41-51-200,  
Andrii.Poliakov@m.hneu.edu.ua.

**Improving the quality of business process automation by developing and testing a method for evaluating the effectiveness of the use of CRM-platforms based on the use of Salesforce system**

**Abstract. Problem.** In order to develop your business, you need to keep track of products that interest people and the services they buy from your company, their place of work and other connections, giving an idea of the customers. **Goal.** The purpose of this article is to analyze, study and test the theoretical and methodological principles and characterize the criteria for improving the efficiency of business process automation using CRM-technologies in the enterprise, as well as building a model to assess the effectiveness of information systems such as Salesforce-CRM. **Methodology.** Methods, technologies, and tools are used for analyzing the effectiveness of automation of business processes of the enterprise using CRM-applications, research, and testing combinations of methods to determine the impact of the Salesforce-CRM-system's implementation in the enterprise. **Results.** Development and justification of measures to improve the information and analytical management system of the organization. **Methods** of effective implementation and analysis of CRM effectiveness include determining its features and the main criteria of performance indicators and choosing the best strategy that would correspond to the company's business processes. **Originality.** Since the aim of the article is to analyze the theoretical and methodologi-

cal foundations and determine the criteria for improving the efficiency of business process automation using CRM-technologies in the enterprise, the following tasks were performed: the purpose and role of CRM-system in enterprise automation were distinguished; a model for evaluating the effectiveness of CRM-system implementation for business process automation was built; quantitative estimates were obtained and recommendations for the Salesforce-CRM implementation were provided. **Practical value.** The proposed research can be used to identify and measure the attributes of the benefits of CRM implementation, such as, for example, cost, efficiency, innovation and improvement of the quality of services, the number of regular customers. As part of the writing of this paper, a customer-oriented model for evaluating the effectiveness of CRM was built.

**Key words:** CRM, Customer Relationship Management, Information Systems, KPIs, Benchmarking, Salesforce.

**Fedorchenko Volodymyr**, Ph.D., Assoc. Prof. Electronic Computers Department, and Information Systems Departmenta, tel. +38 099-640-52-78, volodymyr.fedorchenko@nure.ua,

**Fedorchenko Ryslana**, Master of Computer Science, NIX Solutions LTD, tel. +38 063-325-46-97, Ryslana.Fedorchenko@hneu.net,

**Poliakov Andrii**, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, tel. +38 099-41-51-200, Andrii.Poliakov@m.hneu.edu.ua.

---

УДК 339.944:004

DOI: 10.30977/BUL.2219-5548.2022.96.0.100

## ВИДИ ТА КЛАСИФІКАЦІЯ ДОГОВОРІВ У СФЕРІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Чирва Ю. Є., Євстрат Д. І.

Харківський національний економічний університет ім. С. Кузнеця

**Анотація.** *Визначено, що, незважаючи на особливості різних моделей бізнесу у сфері інформаційних технологій та переважання у вітчизняному економічному просторі аутсорсингу, умовою забезпечення ефективного бізнесу та уникнення різноманітних штрафів і фінансових втрат є побудова діяльності на основі використання системи прийнятних договорів. Доведено, що систематизована класифікація договорів дає змогу не тільки визначити основні ознаки чи істотні риси того чи іншого договору, але й сприяє активізації розвитку бізнесу у сфері інформаційних технологій, адже поліпшує взаєморозуміння розробників та замовників. Розглянуто наявні види та класифікації договорів за певними ознаками, визначено обмеження їхнього практичного застосування. На основі здійсненого аналізу фахової літератури та узагальнення практичного досвіду визначено обмеження наявних класифікацій, систематизовано та вдосконалено види договорів у сфері інформаційних технологій. Висвітлено критерії систематизації та розкрито сутність видів договорів, їхні особливості застосування з урахуванням різноманітних критеріїв.*

**Ключові слова:** *інформаційні технології, компанія, договір, класифікація, систематизація.*

### Вступ

ІТ (інформаційні технології) – це досить молода сфера діяльності, пов'язана зі збором, обробленням, зберіганням, передачею та управлінням інформацією за допомогою комп'ютерних та телекомунікаційних пристроїв.

У загальному випадку можна виокремити три моделі ведення вітчизняного бізнесу у сфері інформаційних технологій.

Перша – це продуктова модель, що передбачає створення та подальший продаж власних програмних продуктів широкому колу покупців. Друга – сервісна (аутсорсингова) модель бізнесу – полягає у розробленні програмного забезпечення на замовлення. І остання, третя, – гібридна модель – має місце в тому випадку, коли компанія комбінує дві попередні моделі, тобто паралельно зі створенням власного програмного продукту надає широкий спектр послуг, пов'язаних із налаштуванням, упровадженням, обслуговуванням, навчанням, інтеграцією з іншими системами, технічною підтримкою, забезпеченням безперебійної роботи корпоративних інформаційних систем [1].

Незважаючи на сутнісні особливості кожної моделі бізнесу та переважання у вітчизняному економічному просторі аутсорсингу, захист результатів інтелектуальної діяльності у сфері ІТ та ефективне забезпечення їхньої комерціалізації вимагають уточнення сутно-

сті видів та систематизації класифікаційних ознак видів договорів.

Здійснення класифікації договорів дає змогу визначити основні ознаки чи істотні риси того чи іншого договору та сприяє активізації розвитку ІТ-бізнесу. Інакше кажучи, коректно побудована класифікація сприяє систематизації накопичених знань, забезпечує належне використання термінів, поліпшує взаєморозуміння розробників та замовників.

### Аналіз публікацій

Розгортаючи різноманітну палітру договорів, насамперед варто звернути увагу на проблеми, пов'язані з дослідженням видів договорів.

Так, аналіз публікацій свідчить, що, незважаючи на окремі розробки із зазначеної проблеми [2–8], питання щодо чітких уявлень про види договорів залишається актуальним для різних сфер господарювання. Не є винятком сфера інформаційних технологій, яка останнім часом розвивається досить швидкими темпами, є однією з пріоритетних у вітчизняній економіці, та у якій на сьогодні відсутня загальноприйнята класифікація видів договорів. Це ускладнює взаємовідносини між замовниками та споживачами й не сприяє розвитку галузі.

У сучасній практиці та публікаціях пропонуються до розгляду лише окремі види

договорів без їхнього узагальнення та структуризації.

Водночас необхідно зазначити, що деякі науковці, як О. Руденко і Є. Кудінова, пропонують класифікувати договори, зокрема аутсорсингових компаній, відповідно до функцій менеджменту, виходячи з визначення аутсорсингу як технології менеджменту. Функцію планування пропонують реалізувати за рахунок аутсорсингу маркетингу й логістики, організації – виробничого аутсорсингу й аутстафінгу, мотивації – аутсорсингу корпоративного харчування та медичного обслуговування, контролю – ІТ-аутсорсингу [10, с. 42]. Важко не погодитись із тим, що аутсорсинг є «технологія менеджменту», проте така класифікація вимагає конкретизації, деталізації та подальших наукових досліджень.

Більш прийнятною до застосування є класифікація договорів у ІТ-сфері, виокремлена за типом робіт [11]: на розроблення сайту, застосунку, програмного рішення для бізнесу, технічне обслуговування, SEO (пошукову оптимізацію сайту), упровадження програмних систем, верстання, тестування, копірайт, графічний дизайн, відчуження авторських прав тощо.

Така класифікація є прикладною, проте комплексний підхід вимагає дослідження різних ознак для систематизації.

#### **Мета та постановка завдання**

Метою дослідження є узагальнення видів договорів, що використовуються в ІТ-сфері, з виокремленням класифікаційних ознак.

Іншим аспектом є вирішення прикладного завдання щодо визначення особливостей застосування тих чи інших договорів для різних моделей ІТ-бізнесу.

#### **Виклад основного матеріалу**

В умовах сучасної економічної системи договір є основним правовим засобом, що визначає та регулює відносини між суб'єктами ІТ-ринку.

Відповідно до Цивільного кодексу України, договір – це домовленість двох або більше сторін, спрямована на встановлення, зміну або припинення цивільних прав та обов'язків [10].

Адекватно обраний вид договору для кожного окремого проєкту надає ІТ-компанії низку переваг [11].

1. Забезпечення реальності домовленостей, тобто фіксування угоди.

2. Створення словника спільної мови із замовником, що є робочим інструментом взаємодії сторін, який визначає алгоритм дій сторін та формує загальний бізнес-процес.

3. Усвідомлення ризиків, які бере на себе ІТ-компанія. Ризики в ІТ-бізнесі досить різноманітні й можуть мати характер як матеріальних втрат (унаслідок неплатоспроможності замовника), так і нематеріальних (втрата власної репутації). Також важливим у цьому контексті є усвідомлення межі відповідальності ІТ-компанії у випадку недосягнення очікуваного результату та суми можливих штрафних санкцій.

Отже, договір – це основний документ, що легалізує домовленість між ІТ-компанією та замовником.

В основі класифікації ІТ-договорів можуть бути різні ознаки – технічні, економічні, юридичні або їхня комбінація.

У сучасних публікаціях [5–6] досить багато уваги приділяється Waterfall (каскадна система) та Agile (гнучка методологія) договорам із висвітленням їхньої сутності й особливостей та без зазначення класифікаційної ознаки. У зв'язку з чим вважаємо за потрібне виокремити зазначені види договорів за такою ознакою, як методологія розроблення.

Отже, з технічної погляду досить важливою є класифікація договорів за методологією розроблення програмного забезпечення: Waterfall – каскадна система, Agile – гнучка система.

Так, класична методологія Waterfall – це підхід до розроблення програмного забезпечення, побудований відповідно до технічного завдання, що заздалегідь написано та узгоджено. Водночас перехід від однієї фази розроблення до іншої відбувається тільки після повного та успішного завершення попередньої фази, а отже, переходів назад чи вперед або перекриття фаз – не відбувається. Завдяки зрозумілій та відносно простій структурі процесу розроблення, зручній звітності (що дозволяє відстежувати ресурси, ризики, витрачений час і фінанси), можливості чіткого визначення бюджету та строків розроблення програмного забезпечення [6] навіть в умовах сьогодення замовники інколи виявляють інтерес до такого підходу.

Водночас Waterfall як методологія розроблення програмного забезпечення позбавлена гнучкості. І якщо реалії життя вимагають більше зусиль і часу на розроблення, то в умовах незмінного бюджету аутсорсинговим ІТ-компаніям доводиться інколи жертвувати

етапом тестування, що в перспективі призводить до значно вищих витрат замовника на пошук багів на етапі впровадженого програмного забезпечення.

Зазначений недолік досліджуваного типу договорів доповнюється жорстким каркасом етапів розроблення програмного забезпечення і підвищеним ризиком унаслідок здійснення тестування не поетапно, а вже готового продукту.

У зв'язку з цим за сучасних умов господарювання та темпів економічного життя сфера застосування економічних догорів Waterfall обмежується головним чином державними замовленнями та науковими розробленнями.

Абсолютно іншою є гнучка модель розробки Agile, основана на ітеративних принципах [5, 12].

Відповідно до договорів, що ґрунтуються на «гнучкій» методології розроблення, після кожної ітерації замовник отримує певні результати й розуміння того, чи задовольняють вони його чи ні. До інших переваг договорів, основаних на гнучкій методології, належать високий рівень взаємодії між членами команди проекту, швидкий результат (робочий код) унаслідок «спринтів», стимулювання змін і поліпшень продукту під час його розроблення, безпосереднє залучення замовника до робочого процесу. Agile є групою методологій, у кожній з яких простежується прагнення до безупинного виконання циклів зворотного зв'язку та безперервного вдосконалення. Отже, нині саме гнучка методологія є найбільш ефективною для ІТ-компаній, стартапів, проектів в інноваційних сферах [5].

За юрисдикцією як наступною ознакою класифікації можна виокремити цілу низку різноманітних догорів [13], проте доцільно сконцентрувати увагу на тих із них, які найбільш властиві для ІТ-сфери, а саме аутстафінгові, фрилансерні, субпідрядні та офертні.

Якщо за договорами, які ґрунтуються на методологіях Waterfall та Agile, співробітники ІТ-компаній здійснюють безпосередньо виконання замовленого проекту, відповідно, каскадним або гнучким шляхом, то аутстафінговий договір (Out staff) є певною мірою іншим варіантом регулювання відносин замовника та ІТ-компанії. За аутстафінговим договором замовник сплачує не за замовлений проект, а за суттю отримує в розпорядження певну кількість годин необхідних фахівців із відповідними навичками. Досить часто роботу таких

фахівців координує менеджер, якого отримує замовник також за аутстафінговим договором. Сфера застосування таких договорів є об'єктивно вужчою порівняно, наприклад, з Agile, адже такі договори підходять досвідченим замовникам у сфері розроблення софту.

Пандемія і карантин значно активізували фриланс як договірні цивільно-правові відносини між виконавцем і замовником, зокрема, у сфері ІТ. І найкращий спосіб легалізувати ці стосунки – це оформити ФОП (фізична особа – підприємець), адже зі зростанням середнього чеку фінансові ризики також збільшуються. У такому випадку відносини між ІТ-компанією та фрилансером будуються на основі відповідних договорів.

З іншого боку, досить поширеною є ситуація, коли для оптимізації оподаткування та робочого процесу, аутсорсингові й гібридні ІТ-компанії співпрацюють із девелоперами в статусі ФОПів на основі відповідних договорів, що, власне, є договорами субпідряду.

Останні дві віддзеркалюють найбільш поширені схеми ІТ-бізнесу в Україні. Проте запровадження спеціального правового режиму для ІТ-індустрії – «Дія Сіті» неодмінно призведе до зміння схем функціонування ІТ-бізнесу й структури договорів.

Публічна оферта (Terms of use) – це пропозиція укласти договір на заздалегідь визначених умовах [14]. Процес переговорів та досягнення домовленостей щодо умов не відбувається. Зазвичай такі договори використовуються для визначення основних прав і обов'язків власника сайту та його користувачів.

Під час запуску нового сайту або початку нового стартапу, розрахованого на невизначену кількість користувачів, дуже важливо правильно прописати взаємні права й обов'язки, щоб у майбутньому мінімізувати ризик їхнього оскарження. Незважаючи на різноманітність за змістом, метою, способами надання згоди, договори оферти покликані визначити обов'язкові умови укладення угод та максимально знімати відповідальність із власника сайту в межах законодавства.

За системою оплати доцільно розрізняти два види ІТ-договорів – Time&Material (мається на увазі продаж людино-годин, коли оплата здійснюється по факту наданих послуг, виконаних робіт) та Fixed Price, або Fixed Bid, або Fixed Fee (передбачає фіксовану оплату за обсяг наданих послуг) [13, 15].

Якщо договір з оплатою по факту використовується сервісними компаніями у випадку, коли проект є не рутинним, інноваційним і досить складно визначити його вартість наперед, то договір із фіксованою ціною застосовується під час реалізації стандартних, нескладних проектів, строк виконання яких та бюджет можна визначити наперед.

Відповідно до договору із системою оплати Time&Material фактична вартість реалізації проекту для замовника може значно перевищити попередні розрахунки [13]. З іншого боку, увесь час, що фактично витрачено на реалізацію завдання співробітниками аутсорсингової або гібридної компанії буде належним чином оплачено. У контексті технічної сторони проект за договором Time&Material реалізується як певна послідовність короткострокових технічних завдань, що можуть підлягати оперативному втручання та коригуванню з метою поліпшення кінцевого результату. Що ж до Fixed Price, то відповідно до такого договору всі вимоги прописуються в технічному завданні до початку реалізації проекту й фактичні можливості їхнього коригування практично відсутні.

Наступна ознака класифікації договорів у IT-сфері перебуває більше в межах юридичного поля. Так, без виокремлення критерію, прихильником поділу договорів на односторонні та двосторонні є В. Сінайський. На його думку, якщо лише одна сторона бере на себе зобов'язання щодо іншої сторони, то такий договір називається одностороннім. Двостороннім же є договір, за яким дві сторони беруть на себе зобов'язання [16, с. 315]. На наш погляд, важко не погодитися з С. Вавженчуком, який вважає, що необхідно також розглядати багатосторонні договори і як критерій класифікації використовувати кількість сторін договору [17, с. 5–10].

Необхідно зазначити, що на сьогодні договори в IT-сфері є публічною офертою, в якій однією стороною є компанія-розробник, а з іншого – необмежена кількість фізичних та юридичних осіб. Крім того, прикладом двостороннього документа є договір на розроблення програмного забезпечення, або мобільного застосунку, чи вебсайту, що укладається між компанією-замовником та IT-компанією. І нарешті, багатосторонніми є договори на розроблення софтверних рішень, що укладаються більш ніж двома сторонами [13].

Виходячи з того, що IT-бізнес – це спрямована на отримання доходу діяльність у сфері створення й комерційного поширення продуктів і послуг, створених на основі інформаційних технологій, значення інформації та технологій як запоруки успішності IT-бізнесу переоцінити важко. Розуміння важливості інформаційної безпеки висуває на одне з перших місць проблему її забезпечення, що втілюється у відповідних договорах. Non-Disclosure Agreement (NDA) – договір, який забороняє певній особі розголошувати інформацію, що вважається конфіденційною [18]. Такий договір є одним із найпопулярніших у контексті досліджуваного питання. Така угода є дієвим механізмом для замовників виконання робіт у сфері IT від небажаного поширення інформації про діяльність компанії та її комерційні таємниці. Штрафи за розголошення інформації можуть бути більші за всю винагороду розробника за декілька років.

Не менш важливим за попередній документ є Non Compete Agreement (NCA) – договір про уникнення конкуренції, що укладається між роботодавцем і працівником та обмежує останнього у праві працювати в будь-якому подібному бізнесі на певний період часу після припинення трудових відносин [19]. Зазвичай NCA укладають разом із NDA.

Для попередження випадків, коли колишні співробітники використовують напрацьовану базу клієнтів з метою створення власного бізнесу або в іншій компанії, а конкуренти надсилають джоб офери співробітникам своїх колег, на ринку використовують Non-Solicitation Agreement (NSA) – договір про «непереманювання» фахівців та/або клієнтів. Також цілком можливим є переманювання клієнтів замовником. Укладений NSA допоможе, по-перше, попередити такі випадки, а по-друге, буде підставою вимагати відшкодування збитків та/або виплати неустойки.

Незважаючи на те, що такі види договорів стають майже традиційними у вітчизняній IT-практиці та висвітлюються в публікаціях, класифікаційна ознака не знайшла належного відтворення. На нашу думку, договори захисту від нерозголошення, уникнення конкуренції та «непереманювання» доцільно виокремлювати за такою ознакою, як методи захисту від недобросовісної поведінки співробітників та контрагентів.

Також, на наш погляд, спираючись на норми міжнародного та національного права,

доцільно виокремити за географічною ознакою міжнародні та національні договори. Якщо міжнародні договори укладаються між контрагентами різних країн, то національні – між резидентами однієї країни. У цьому контексті доцільно зазначити, що ІТ в Україні – це переважно експортно орієнтований аутсорсинг. Отже, вітчизняні ІТ-компанії здебільшого укладають міжнародні договори з країнами Європейського союзу та США. Обсяг національних договорів нині обмежується темпами розвитку внутрішнього аутсорсингового ринку інформаційних технологій.

І останнє, спираючись на національне законодавство, залежно від моменту, з якого договір вважається укладеним, можна розрізняти консенсуальні та реальні договори. Перші вважаються укладеними з моменту, як тільки сторони досягли згоди за всіма істотними умовами договору, що має відтворюватися у формі, яка передбачена для цього виду договору. Реальні договори є дійсними лише після вчинення на основі досягнутої згоди певної дії з передачі предмета договору (наприклад, технічної документації для виконання проекту).

### Висновки

Отже, на основі проведеного дослідження здійснено систематизацію та вдосконалення класифікації договорів у сфері інформаційних технологій. Така класифікація договорів дає змогу продуктивним, сервісним і гібридним компаніям з'ясувати суттєві властивості того чи іншого договору та в щоденній практиці застосовувати саме ті договори, які найбільше відповідають особливостям їхніх моделей бізнесу з урахуванням різноманітних критеріїв.

Питання ж ліцензійних договорів як засобів комерціалізації результатів творчої діяльності у сфері інформаційних технологій як для аутсорсингових, так і продуктивних компаній у сучасних умовах потребують також наукових пошуків і переосмислення, проте, на жаль, не є метою цієї публікації.

### Література

- Сьомкіна Т. В., Литвинова О. В., Лобань О. О. Особливості моделей функціонування ІТ-компаній в Україні // Науковий вісник Ужгородського національного університету. 2018. Вип. 18. Ч. 3. С. 84–87.
- Тверезенко О. О. Класифікація договорів у сфері інтелектуальної власності. URL: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?pdfurl=http%3A%2F%2Fp>

- pp-ournal.kiev.ua%2Farchive%2F2011%2F10%2F32.pdf&clen=299515&chunk=true.
- Договірне право України. Загальна частина: навч. посіб. / Т. В. Бондар та ін.; за ред. О. В. Дзери. Київ: Юрінком Інтер, 2008. С. 321–325.
- Рузакова О. А. Право интеллектуальной собственности / Московская финансово-промышленная академия; Москва, 2004. 308 с.
- Agile чи Waterfall – який варіант відповідає вашому бізнесу. URL: <https://worksection.com/ua/blog/waterfall-vs-agile.html>.
- Waterfall методология разработки. URL: <https://qaevolution.ru/metodologiya-menedzhment/waterfall>.
- Основні види договорів ІТ. URL: <https://www.mak-mel.com/podatkove-pravo/17740>.
- Руденко О. Н., Кудинова Е. А. Актуальные вопросы определения аутсорсинга // Международный научно-исследовательский журнал. 2013. № 4–2(11). С. 40–43.
- Святошнюк А. Л. Класифікація договорів аутсорсингу. URL: [http://webcache.googleusercontent.com/search?q=cache:l\\_XRa2JMmBoJ:dspace.onu.edu.ua:8080/bitstream/123456789/11789/1/222-224.pdf+&cd=1&hl=uk&ct=clnk&gl=ua](http://webcache.googleusercontent.com/search?q=cache:l_XRa2JMmBoJ:dspace.onu.edu.ua:8080/bitstream/123456789/11789/1/222-224.pdf+&cd=1&hl=uk&ct=clnk&gl=ua).
- Цивільний кодекс України. URL: <https://zakon.rada.gov.ua/laws/show/435-5#Text>
- Тертичний Т. У чому цінність договору для бізнесу? URL: <https://blog.liga.net/user/ttertychnyi/article/41929>.
- Евдокимов А. Обзор Agile. Что это: методология, метод или философия? URL: <https://scrumtrek.ru/blog/agile-scrum/4029/metodologiya-agile>.
- Сталиваров В. ІТ-контракти. Види та класифікація договорів. URL: <https://stalirov.lawyer/uk/posts/it-kontrakti-vidi-ta-klasifikaciya-dogovoriv>.
- Что такое Terms of Use или Как онлайн-бизнесу избежать проблем с пользователями. URL: [https://habr.com/ru/company/icon\\_partners/blog/531730](https://habr.com/ru/company/icon_partners/blog/531730).
- Korotia Ye. Time-and-Materials vs Fixed Price: Which to Choose for Your Project? URL: <https://medium.com/@Eugeniya/time-and-materials-vs-fixed-price-which-to-choose-for-your-project-11dc6adc758b>
- Синайский В. И. Русское гражданское право. Москва: Статут, 2002. 638 с.
- Вавженчук С. До питання класифікації договорів на односторонні та взаємні, відплатні й безвідплатні // Цивільне право і процес. 2021. № 6. С. 5–10.
- Non-Disclosure Agreement (NDA). URL: <https://www.investopedia.com/terms/n/nda.asp>.
- Oliveros F. What is a Non-Compete Agreement? Do You Need One for Your Business? URL:



<https://gusto.com/blog/people-management/non-compete-agreement/>

### References

1. S'omkina T. V., Lytvynova O. V., Loban O. O. Features of models of functioning of IT companies in Ukraine // Scientific Bulletin of Uzhgorod National University, 2018. Issue. 18, Ch. 3. P. 84–87.
2. Tverezenko O. O. Classification of contracts in the field of intellectual property. Access mode: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?Pdfurl=http%3A%2F%2Fppp-journal.kiev.ua%2Farchive%2F2011%2F10%2F32.pdf&clen=299515&chunk=true.
3. Contract law of Ukraine. General part: textbook. / T. V. Bondar, O. V. Dzero, N. S. Kuznetsova, etc.; for ed. O. V. Dzeri. Kyiv: Jurinkom Inter, 2008. P. 321–325.
4. Ruzakova O. A. Intellectual property law / Moscow Academy of Finance and Industry; Moskva, 2004. 308 p.
5. Agile or Waterfall – which option is right for your business. URL: <https://worksection.com/ua/blog/waterfall-vs-agile.html>.
6. Waterfall development methodology. URL: <https://qaevolution.ru/metodologiya-menedzhment/waterfall>.
7. The main types of IT contracts. URL: <https://www.mak-mel.com/podatkovopravo/17740>.
8. Rudenko O. N., Kudimova E. A. Actual issues of outsourcing // International Research Journal. 2013. № 4–2 (11). P. 40–43.
9. Svyatoshnyuk A. L. Classification of outsourcing agreements. URL: [http://webcache.googleusercontent.com/search?q=cache:l\\_XRa2JMmBoJ:dspace.onu.edu.ua:8080/bitstream/123456789/11789/1/222-224.pdf+&cd=1&hl=uk&ct=clnk&gl=ua](http://webcache.googleusercontent.com/search?q=cache:l_XRa2JMmBoJ:dspace.onu.edu.ua:8080/bitstream/123456789/11789/1/222-224.pdf+&cd=1&hl=uk&ct=clnk&gl=ua).
10. Civil Code of Ukraine. URL: <https://zakon.rada.gov.ua/laws/show/435-5#Text>
11. Tertychny T. What is the value of the contract for business? URL: <https://blog.liga.net/user/tertychnyi/article/41929>.
12. Evdokimov A. Review of Agile. What is it: methodology, method or philosophy? URL: <https://scrumtrek.ru/blog/agile-scrum/4029/metodologiya-agile>.
13. Stalivarov V. IT-contracts. Types and classification of contracts. URL: <https://stalirov.lawyer/uk/posts/it-kontrakti-vidi-ta-klasifikaciya-dogovoriv>.
14. What are the Terms of Use or How to avoid problems with online business users. URL: [https://habr.com/ru/company/icon\\_partners/blog/531730](https://habr.com/ru/company/icon_partners/blog/531730).
15. Korotia Ye. Time-and-Materials vs Fixed Price: Which to Choose for Your Project? URL: [\[materials-vs-fixed-price-which-to-choose-for-your-project-11dc6adc758b\]\(https://medium.com/@Eugeniya/time-and-materials-vs-fixed-price-which-to-choose-for-your-project-11dc6adc758b\)](https://medium.com/@Eugeniya/time-and-</a></li>
</ol>
</div>
<div data-bbox=)

16. Sinaisky V. I. Russian civil law. Moscow: Statute, 2002. 638 p.
17. Vavzhynchuk S. On the question of classification of agreements on unilateral and mutual, repayable and gratuitous // Civil law and process. 2021. № 6. P. 5–10.
18. Non-Disclosure Agreement (NDA). URL: <https://www.investopedia.com/terms/n/nda.asp>.
19. Oliveros F. What is a Non-Compete Agreement? Do You Need One for Your Business? URL: <https://gusto.com/blog/people-management/non-compete-agreement>.

**Чирва Юлія Євгенівна**, к.е.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 066-329-75-50, [chirva\\_juliya2015@ukr.net](mailto:chirva_juliya2015@ukr.net),

**Євстрат Дмитро Іванович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 066-765-48-76, [dmitry.yevstrat@hneu.net](mailto:dmitry.yevstrat@hneu.net).

### Types and classification of contracts in the field of information technologies

**Abstract. Problem.** It is determined that despite the peculiarities of different models of IT business and the predominance of outsourcing in the domestic economic space, the condition for ensuring efficient business and avoiding various fines and financial losses is to build activities based on a system of acceptable contracts. It is proved that the systematized classification of contracts allows not only to determine the main features or essential features of a contract, but also promotes the development of IT - business, because it improves mutual understanding between developers and customers. The existing types and classifications of contracts on certain grounds are considered, the limitations of their practical application are determined. **Goal.** The goal of the research is to generalize the types of contracts used in the IT field, with the separation of classification features. **Methodology.** Methods of analysis and synthesis, induction and deduction are used; analogies and comparisons; methods of classification and systematization. **Results.** On the basis of the analysis of special literature and generalization of practical experience the limitations of the existing classifications are determined, the systematization and improvement of the types of contracts in the field of information technologies is carried out. The criteria of systematization are highlighted and the essence of the types of contracts, their peculiarities of application taking into account various criteria are revealed. **Originality.** It is proposed to classify contracts in the field of information technology according to the methodology of software development into Waterfall

– cascade system and Agile – flexible system; by jurisdiction – to distinguish out staffing, freelance, subcontracting and offering; by payment system – Time & Material and Fixed Price (Fixed Bid, Fixed Fee); by the number of parties to the agreement – unilateral, bilateral and multilateral; by methods of protection against unscrupulous behavior of employees and contractors – Non-Disclosure Agreement, Non-Compete Agreement, Non-Solicitation Agreement; by geographical feature – international and national; depending on the moment from which the contract is considered concluded – consensual and real contracts. **Practical value.** The application of the proposed classification of contracts allows product, service and hybrid companies to find out the essential features of a

contract and in everyday practice to apply those contracts that best meet the characteristics of their business models based on various criteria.

**Key words:** information technologies, IT-company, contract, classification, systematization.

**Chyrva Yuliia**, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National Economic University, tel. +38 066-329-75-50, [chirva\\_juliya2015@ukr.net](mailto:chirva_juliya2015@ukr.net),

**Yevstrat Dmitry**, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National Economic University, tel. +38 066-765-48-76, [dmitry.yevstrat@hneu.net](mailto:dmitry.yevstrat@hneu.net).

---

## ПРОФЕСІЙНА ОСВІТА

УДК 621.317

DOI: 10.30977/BUL.2219-5548.2022.96.0.141

**ПІДВИЩЕННЯ ЯКОСТІ НАВЧАЛЬНОГО ПРОЦЕСУ ШЛЯХОМ РОЗРОБЛЕННЯ І ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ НА БАЗІ ВІРТУАЛЬНИХ КОМП'ЮТЕРНИХ ТРЕНАЖЕРІВ****Скорін Ю. І., Щербаков О. В., Ушакова І. О.****Харківський національний економічний університет ім. С. Кузнеця**

***Анотація.** Запропонована стаття підсумовує результати проведеного авторами дослідження щодо пошуку шляхів удосконалення навчального процесу із застосуванням новітніх розробок у галузі програмного забезпечення. Доцільною формою підвищення ефективності професійного навчання визначено впровадження імітаційних віртуальних тренажерів, побудованих на базі віртуальних вимірювальних приладів, і віртуалізації вимірювальних процесів. Аналіз та оцінювання переваг і недоліків традиційних методів і засобів вимірювання довели перспективність віртуалізації. За результатами досліджень розроблено пропозиції щодо побудови віртуальних комп'ютерних тренажерів на основі віртуальних вимірювальних приладів. Такий підхід не тільки забезпечить високу якість навчання, унаочнивши його, а й уможливить створення нових віртуальних комп'ютерних систем дистанційної освіти як необхідної форми набуття знань у сучасному світі.*

***Ключові слова:** віртуалізація, прилад, ефективність, тренажер, метод, засіб, парк, система.*

**Вступ**

Інтерактивні комп'ютерні тренажери є складними комплексами реалізації симуляції та моделювання віртуальних і фізичних моделей. Їхнє програмне забезпечення надає змогу розробляти методики та прийоми на допомогу в формуванні й ухваленні рішень у багатьох галузях діяльності.

Аналіз сучасного стану в розвитку вимірювальних технологій окреслює майбутні тенденції, спрямовані, поряд із удосконаленням традиційних, на розроблення віртуального покоління вимірювальних приладів.

Цим процесам сприяють такі чинники:

- значний прогрес у розвитку комп'ютерних технологій зробив персональний комп'ютер звичним інструментом у роботі інженерів, учених, викладачів тощо;
- необхідність у підвищенні темпів поповнення й оновлення парків засобів вимірювальної техніки.

Водночас порушення інтеграційних зв'язків значно ускладнює процеси розроблення та виробництва сучасних засобів вимірювальної техніки [3].

Отже, сучасні реалії вимагають пошуку альтернативних способів удосконалення парку вимірювальних приладів, серед яких найбільш застосовним можна вважати створення віртуальних їхніх видів.

Орієнтація на комп'ютеризацію всіх галузей господарства потребує використання потужного технологічного потенціалу, і саме віртуалізація допоможе вдосконалити процеси, якими оперують вимірювальні системи.

Саме з метою пошуку новітніх шляхів на базі комплексного дослідження теоретичних і практичних аспектів підвищення якості навчального процесу було проведено аналіз ефективності використання традиційних вимірювальних засобів порівняно з перспективами комп'ютерної віртуалізації процесів вимірювання.

За результатами розроблено застосовні до практики пропозиції щодо комплексного використання традиційних і віртуальних приладів.

Крім методу комбінованого використання комп'ютерних віртуальних тренажерів у навчальному процесі, розроблено оригінальні ActiveX – елементи, завдяки яким інтерфейс віртуальних тренажерів повністю повторює інтерфейс традиційних приладів.

У виробленні підходу до розроблення й тестування програмного забезпечення для віртуальних комп'ютерних тренажерів був обраний дискурс, висвітлений у науковій літературі.

Основою стали критерії новизни, сформульовані авторами робіт [6; 7]. Так, А. Бер-

мус [6] вважає, що новизна дослідження визначається тим «наскільки є сучасними й оригінальними використовувані в ньому уявлення та методи».

Гідна уваги позиція М. Солнишкова [7], який цілком доречно вводить ці критерії в параметри оцінювання наукової новизни поряд із фіксацією фактів приросту знань тощо.

### Аналіз публікацій

Створення та поширення новітніх технологій щільно пов'язане зі зміцненням економіки як окремих підприємств і галузей, так і держави загалом.

Тому так важлива ефективна професійна підготовка фахівців, які б вільно володіли найсучаснішими науково-технічними розробками.

Підвищенню якості навчального процесу, безперечно, сприяє проектування навчальних середовищ із використанням комп'ютерних тренажерів.

З огляду на важливість проблеми, науковці завжди приділяли значну увагу питанням удосконалення навчального процесу.

Так, різні аспекти та шляхи підвищення ефективності використання перспективних форм інтерфейсів і комп'ютерних тренажерів розглядалися в роботах цілої низки авторів [5–8].

Питанням ергономічного проектування перспективних форм інтерфейсів комп'ютерних тренажерів приділяли увагу автори робіт [4; 8].

Практикою побудови програмного забезпечення комп'ютерних віртуальних тренажерів для широкого кола об'єктів у різний час займалися вчені [6–8].

Досліджували феномени, пов'язані з віртуальною реальністю та інтерактивністю, науковці [7; 8].

### Мета та постановка завдання

Проведений аналіз довів перевагу комп'ютерних віртуальних приладів перед традиційними.

Це уможливує створення віртуальних комп'ютерних тренажерів та інтерактивних систем дистанційного управління, покликаних забезпечити унаочнення й тим самим підвищення ефективності навчального процесу.

Актуальність і прикладна значущість такого напряму полягає в тому, що [3]:

- склад штатних засобів вимірювальної техніки, який є в наявності й потрібен для забезпечення навчального процесу, є обмеженим, часто вимагає ремонту, відновлення або заміни, тому значення віртуальних комп'ютерних тренажерів у таких випадках важко переоцінити;

- за допомогою віртуальних комп'ютерних тренажерів можна забезпечити набуття практичних навичок роботи з найсучаснішими приладами, які, на жаль, через обмеження технічних або економічних можливостей не використовуються в навчальному процесі;

- віртуальні комп'ютерні тренажери можуть використовуватися студентами під час самостійної підготовки до занять: вони досить прості в експлуатації, не вимагають спеціальних знань із програмування, не є критичними до апаратного складу та програмного забезпечення, містять підказки та коментарі, які управляють діями студента, відпрацьовують його помилки;

- віртуальні комп'ютерні тренажери доцільно використовувати на попередньому етапі підготовки до робіт на штатній техніці, під час самостійної підготовки до занять, заочній формі навчання тощо, тобто в тих випадках, коли доступ до штатних засобів вимірювальної техніки є обмеженим або недоцільним;

- віртуальному комп'ютерному тренажеру можна надати додаткові функції, які не притаманні реальному приладу, наприклад: відображати у вигляді графіків фізичні процеси, що відбуваються в приладі під час проведення вимірювального експерименту, надавати довідкову інформацію, здійснювати оброблення та зберігання результатів вимірювань і діагностики, проводити тестування і контроль рівня знань студентів тощо;

- віртуальні комп'ютерні тренажери, що розглядаються в статті, мають інтерфейс, зовнішній вигляд якого повністю відповідає інтерфейсу реальних приладів, що є важливим із погляду ефективності та наочності навчального процесу.

Отже, можна сформулювати цілі проведених досліджень, а саме: обґрунтування альтернативних способів вдосконалення парку засобів вимірювальної техніки шляхом розроблення віртуальних вимірювальних приладів і підвищення ефективності навчального процесу шляхом розроблення та впровадження віртуальних комп'ютерних тренажерів на базі створених віртуальних приладів.

### Виклад основного матеріалу

Віртуальні прилади є концепцією, яка створює передумови для організації програмно-керованих систем збирання даних й управління широкою номенклатурою різних технічних об'єктів і технологічних процесів [1–3].

Система реалізується за допомогою створення програмної моделі певного гіпотетичного або реального вимірювального приладу або іншого об'єкта, до того ж і засоби управління, і сама логіка роботи приладу реалізуються програмним шляхом [5].

Значною перевагою віртуальних приладів є їхня універсальність і практично необмежений потенціал щодо розширення функціональних можливостей, до того ж без зміни апаратного складу приладів, а тільки за рахунок удосконалення програмного забезпечення [1; 3].

Найбільш перспективним, на наш погляд, є підхід, в основу якого покладено принцип об'єднання комп'ютера з блоком управління, базованого на платі збирання та перетворення даних.

Отже, віртуальний прилад складається з двох основних компонентів, а саме: пристрою управління та оброблення інформації, тобто персонального комп'ютера, та плати збирання і перетворення даних.

Перший компонент – персональний комп'ютер – є необхідним атрибутом сучасності й обов'язковим інструментом на робочому місці інженера-метролога. Тому будемо розглядати його як уже реальний компонент віртуального вимірювального приладу.

Другий компонент – блок управління – містить плату збирання та перетворення даних (наприклад SDI-ADC16-16). Виробництво плати збирання й перетворення даних у рази дешевше за сам прилад, що підтверджено проведеним аналізом орієнтовних цін.

Плата збирання та перетворення даних відрізняється простотою у використанні й обслуговуванні.

Завдяки наявності в програмному забезпеченні віртуального приладу певної системи підказок робота з віртуальним приладом не вимагає від студента спеціальних знань у галузі програмування.

Отже, можна констатувати, що сучасний віртуальний вимірювальний прилад є технічно об'єднаною сукупністю персонального комп'ютера з вбудованою спеціальною платою збирання та перетворення даних або з

додатковим блоком, підключеним до персонального комп'ютера за допомогою з'єднувального кабелю, якщо проводяться більш складні та багатофункціональні вимірювання [1–3].

Під час використання подібної плати збирання й перетворення даних, а також відповідного програмного забезпечення розробник ніби проєктує конкретний засіб вимірювань, оптимізуючи його для проведення певного вимірювального експерименту або конкретного метрологічного завдання [3].

Так, на базі плати збирання й перетворення даних ADC 16-32 було розроблено чинний макет комп'ютерного віртуального вимірювального приладу, а саме віртуального вольтметра постійного струму, і пакет програмного забезпечення для його програмної реалізації.

Дослідження приладу та тестування програмного забезпечення показали, що метрологічні характеристики комп'ютерного віртуального пристрою практично повністю відповідають аналогічним характеристикам поширених штатних цифрових вольтметрів.

Також на базі розроблених віртуальних приладів створено комп'ютерний віртуальний вимірювальний комплекс у вигляді пакета програмного забезпечення під загальною назвою «Віртуальна вимірювальна лабораторія», до складу якого ввійшли кілька комп'ютерних тренажерів цифрових і аналогових приладів.

Такі комп'ютерні тренажери можуть використовуватися в навчальному процесі як відокремлено, так і в складі загального циклу-практикуму.

Методика проведення вимірювального експерименту за допомогою того чи іншого комп'ютерного віртуального приладу-тренажера практично не відрізняється від усталених методик, притаманних відповідним традиційним вимірювальним приладам, тому не розглядається у цій статті.

Розроблений програмний продукт за принципом побудови є модульною структурою. Він містить блок управління або програмну оболонку, загальну для всіх віртуальних тренажерів, що міститься в складі комп'ютерної віртуальної вимірювальної лабораторії.

### Висновки

Важливою особливістю розробленого програмного продукту є те, що його робота може бути реалізована в режимі підказки. Тобто програмне забезпечення фактично управляє діями студента, надаючи коментарі

та підказки, водночас воно блокує дії оператора, якщо вони можуть викликати критичну помилку.

Практично необмеженими є функціональні можливості комп'ютерного тренажера, насамперед не притаманні традиційному приладу [1–3].

Тому залежно від призначення кожного конкретного віртуального тренажера деякі модулі програмного продукту містять інтерактивні електронні таблиці, часові діаграми, графіки, що відображують фізичні процеси, які відбуваються в приладі.

Розроблений пакет програмного забезпечення є закінченим і самодостатнім програмним продуктом, що містить інсталяційний модуль, адаптований під більшість платформ програмного забезпечення.

Запропанований програмний продукт повністю адаптований до використання в мережі Інтернет або локальних комп'ютерних мережах. Ще одна важлива особливість програмного продукту полягає в тому, що він є базовим для побудови віртуальних вимірювальних приладів і комп'ютерних тренажерів інших видів і типів.

Але варто зазначити, що впровадження комп'ютерних тренажерів у процес навчання жодним чином не передбачає підміну штатних традиційних приладів їхніми комп'ютерними моделями, а навпаки – тільки доповнює та розширює можливості як викладачів, так і студентів.

Питання, пов'язане з виробленням концепції, методики спільного використання в навчальному процесі як штатних традиційних приладів, так і їхніх комп'ютерних моделей-тренажерів виходить за межі цієї статті.

У плані подальшого розвитку пакета програмного забезпечення необхідно зазначити, що можливості поповнення парку віртуальних приладів є практично необмеженими, тому цікавою є ідея побудови, наприклад, віртуальних аналогових приладів, аналізаторів спектра тощо.

Також перспективною є сфера використання розроблених віртуальних приладів.

На їхній основі можна будувати вимірювальні системи для досліджень не тільки автономних засобів вимірювань, а й вимірювально-інформаційних систем, параметри та зовнішній вигляд яких можна корегувати як на стадії розроблення, так і в процесі роботи [1–4].

Отже у статті:

- проаналізовано традиційних підходів до вимірювального процесу;

- обґрунтовано вибір віртуалізації вимірювального процесу як найбільш ефективного засобу вдосконалення приладового парку;

- на базі плати збирання й перетворення даних розглянуто чинний макет віртуального вимірювального приладу та пакет програмного забезпечення для його реалізації;

- визначено віртуальні прилади як базові для побудови на їхній основі віртуальних тренажерів, що формують передумови для створення й удосконалення систем дистанційного навчання;

- вирішено важливе прикладне завдання, а саме: віртуальні комп'ютерні тренажери, що розглядаються в статті, мають інтерфейс, зовнішній вигляд якого повністю відповідає інтерфейсу реальних приладів;

- для досягнення цього ефекту створено нестандартні ActiveX елементи, що є важливим з погляду ефективності та наочності навчального процесу.

### Література

1. Скорін Ю. І., Стаднік В. В., Клименко А. М. Віртуальні прилади у вимірювальній лабораторії. Вісник Національного технічного університету «ХПІ»: збірник наукових праць. Серія: «Інформатика та моделювання». Харків: НТУ «ХПІ». 2012. № 38. С. 84–92.
2. Скорин Ю. И., Стадник В. В. Создание виртуальных измерительных приборов средствами технологии Windows Presentation Foundation: материалы 10-й Международной научно-технической конференции «Приборостроение-2017», 1–3 ноября 2017 г. Минск: БИТУ, 2017. С. 185–187.
3. Скорін Ю. І. Щербачов О. В., Магдалиць Т. І. Віртуальні вимірювальні та діагностичні прилади. Системи обробки інформації: збірник наукових праць. Вип. 4(102). Т. 1. Інформаційні технології та захист інформації. Харків: ХУПС, 2012. С. 65–68.
4. Виртуальные измерительные приборы. URL: <https://strpo.ru/electricity/viii-virtual-measuring-instruments/> (дата звернення: 26.11.2021).
5. Принципы построения виртуальных тренажеров. URL: <https://www.sunspire.ru/articles/part-35/> (дата звернення: 06.10.2020).
6. Бермус А. Г. Общие основы педагогики: учеб. пособие. Ростов-на-Дону: Изд-во Ростов, гос. пед. ун-та, 1999. 114 с.
7. Солнышков М. Е. Критерии новизны научно-педагогических исследований. URL: <https://cyberleninka.ru/article/v/kriteriy-novizny-nauchno-pedagogicheskikh-issledovaniy> (дата звернення: 26.11.2021).

8. Рахманов Ф. Г. Применение имитационных виртуальных тренажеров в процессе профессионального обучения. Молодой ученый. 2015. № 9. С. 1173–1175. URL: <https://moluch.ru/archive/89/17867/> (дата звернення: 26.11.2021).

### References

1. Skorin Yu. I., Stadnik V. V., Klymenko A. M. Virtual'ni pryklady u vymiryuval'niy laboratorii / Visnyk Natsional'noho tekhnichnoho universytetu "KHPI". Zbirnyk naukovykh prats'. Seriya: Informatyka ta mode-lyuvannya. Kharkiv: NTU "KHPI". № 38. 2012. P. 84–92.
2. Skorin Yu. I., Stadnik V. V. Sozdaniye virtual'nykh izmeritel'nykh priborov sredstvami tekhnologii Windows Presentation Foundation. Materialy 10-y Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii "Priborostroyeniye-2017", 1–3 noyabrya 2017. Minsk: BITU, 2017. P. 185–187.
3. Skorin Yu. I., Shcherbakov O. V., Mahdalyts' T. I. Virtual'ni vymiryuval'ni ta diahnostychni pryklady. Systemy obrobky informatsiyi. Zbirnyk naukovykh prats'. Issue 4(102), volume 1. Informatsiyi tekhnolohiyi ta zakhyst informatsiyi. Kharkiv: KHUPS, 2012. P. 65–68.
4. Virtual'nyye izmeritel'nyye pribory. URL: <https://strpo.ru/electricity/viii-virtual-measuring-instruments/> (accessed: 26.11.2021).
5. Printsipy postroyeniya virtual'nykh trenazherov. URL: <https://www.sunspire.ru/articles/-part35/> (accessed: 06.10.2020).
6. Bermus A. G. Obshchiye osnovy pedagogiki: ucheb pocobiye. Roctov-na-Donu: Izd-vo Roctov, gos. ped. un-ta, 1999. 114 p.
7. Solnyshkov M. E. Kriterii novizny nauchno-pedagogicheckikh icclodovaniy. URL: <https://cyberleninka.ru/article/v/kriteriy-novizny-nauchno-pedagogich-eskih-issledovaniy> (accessed: 26.11.2021).
8. Rakhmanov F. G. Primeneniye imitatsionnykh virtual'nykh trenazherov v protsesse professional'nogo obucheniya. Molodoy uchenyy. 2015. № 9. P. 1173–1175. URL: <https://moluch.ru/archive/89/17867/> (accessed: 26.11.2021).

**Скорін Юрій Іванович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 066-748-47-51, [skorin.yuriy@gmail.com](mailto:skorin.yuriy@gmail.com),

**Щербаків Олександр Всеволодович**, к.т.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 050-636-20-09, [oleksandr.shcherbakov.kafis@gmail.com](mailto:oleksandr.shcherbakov.kafis@gmail.com),

**Ушакова Ірина Олексіївна**, к.е.н., зав. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38 066-785-09-92, [varavina.ira@gmail.com](mailto:varavina.ira@gmail.com).

### Enhancing educational process by development and testing of information and measurement system software based on virtual computer simulators

**Abstract. Problem.** The proposed article summarizes the results of a study conducted by the authors in order to find the ways to improve the educational process using the latest developments in the field of software. The article outlines the introduction of simulation-virtual simulators, built on the basis of virtual measuring devices and virtualization methods as a way to improve the educational process, increase the efficiency of using promising forms of vocational training. **Goal.** The objectives of the research are to substantiate alternative ways to improve the fleet of measuring equipment through the development of virtual measuring instruments and increase the efficiency of the educational process through the development and implementation of virtual computer systems. **Methodology.** The research is based on the analysis of traditional methods and means of measurement and the proposal of an alternative solution to the problem of virtualization of the measurement process. Analysis and evaluation of the advantages and disadvantages of traditional measurement methods and tools have proven the viability of virtualization. **Results.** Based on the results of the research, the proposals were developed for the construction of virtual computer simulators based on virtual measuring instruments. This approach will not only provide high-quality education through visualization of educational processes, but will also allow the creation of new virtual computer systems for distance learning as a necessary form of obtaining knowledge in the modern world. **Originality.** The developed software package is a complete and self-sufficient software product, which includes an installation module adapted to most software platforms. **Practical value.** The introduction of virtual simulators built on the basis of virtual measuring instruments and virtualization of measuring processes was identified as an appropriate form of increasing the effectiveness of vocational training.

**Key words:** virtualization, device, efficiency, simulator, method, means, park, system.

**Skorin Yuriy**, Ph.D., Assoc. Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets,

tel. +38 066-748-47-51, [skorin.yuriy@gmail.com](mailto:skorin.yuriy@gmail.com),  
**Shcherbakov Alexander**, Ph.D., Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets,  
tel. +38 050-636-20-09,  
[oleksandr.shcherbakov.kafis@gmail.com](mailto:oleksandr.shcherbakov.kafis@gmail.com),

**Ushakova Iryna**, Ph.D., Assoc. Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets,  
tel. +38 066-785-09-92, [vara-vina.ira@gmail.com](mailto:vara-vina.ira@gmail.com).

УДК 378.14

DOI: 10.30977/BUL.2219-5548.2022.96.0.146

## ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИВЧЕННЯ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЗАЦІЇ З ДОПОМОГОЮ ВИКОРИСТАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ТЕСТУВАННЯ

Щербаков О. В., Скорін Ю. І.

Харківський національний економічний університет ім. С. Кузнеця

***Анотація.** У статті розглядаються шляхи підвищення ефективності вивчення основ програмування та алгоритмізації. Перспективною формою підвищення ефективності навчального процесу є використання автоматизованих систем тестування. Аналіз наявних підходів до вирішення цього питання дозволив виокремити декілька різних варіантів організації навчального процесу з використанням автоматизованих систем тестування та виявити переваги й недоліки кожного з них. За результатами досліджень було запропоновано застосування автоматизованої тестувальної системи, що дозволяє швидко та ефективно організувати процес автоматичної перевірки програм студентів, надає студентам змогу в будь-який час перевірити свої рішення, а також залишає викладачеві можливість максимально контролювати навчальний процес і керувати ним.*

***Ключові слова:** програмування, алгоритмізація, автоматизована система тестування, ефективність навчального процесу.*

### Вступ

Ефективне та якісне вивчення студентами ІТ-спеціальностей основ програмування та алгоритмізації є запорукою підготовки висококваліфікованих фахівців для ІТ-галузі, вимоги до рівня підготовки яких щороку зростають. Поява нових мов програмування, розвиток теоретичних основ комп'ютерних технологій вимагають постійного вдосконалення змісту навчальних дисциплін і навчальних планів. Аналіз міжнародних стандартів у галузі комп'ютерних наук [1] свідчить про необхідність формування у випускників ІТ-спеціальностей університетів широкого спектру професійних та фахових компетентностей у сфері інформаційних технологій, зокрема здатність алгоритмічного мислення, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їхньої ефективності та складності, здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування та здатність формулювати й забезпечувати вимоги щодо якості програмного забезпечення. Одними з основних фундаментальних дисциплін у підготовці майбутніх ІТ-фахівців є дисципліни «Програмування» та «Алгоритми та структури даних». Їхньою метою є вивчення основних алгоритмічних і синтаксичних конструкцій мов програмування, ефективних структур і алгоритмів оброблення даних в інформацій-

них системах, формування у студентів компетентностей у проектуванні, розробленні й аналізі алгоритмів, оцінюванні їхньої складності для адекватного моделювання предметних галузей, а також реалізації алгоритмів у вигляді програмних та інформаційних систем. У більшості університетів у межах дисципліни «Алгоритми та структури даних» вивчаються такі теми, як алгоритми сортування та пошуку, алгоритми на графах, алгоритми обчислювальної геометрії, алгоритми пошуку в рядках, комбінаторні й евристичні алгоритми. Основними структурами даних, що розглядаються в межах цієї дисципліни, є лінійні списки, черги, стеки, дерева та графи [2–4]. Реалізація всіх перелічених вище алгоритмів і структур даних у вигляді програм, з одного боку, дозволяє студентам глибше зрозуміти теоретичний матеріал і набути практичних навичок його використання, а з іншого – дає викладачам змогу переконатися, що в студентів сформувалися необхідні компетентності. Але водночас актуальним стає питання тестування програм, що розробляють студенти. Очевидно, що так зване ручне тестування є надто трудомістким процесом і малоефективним, тому все більше викладачів використовують автоматизовані тестувальні системи.

### Аналіз публікацій

Проведений аналіз публікацій [5–8] показав, що використання в навчальному процесі



автоматизованих систем тестування є досить поширеним підходом. У багатьох навчальних закладах для вивчення дисциплін, пов'язаних з алгоритмізацією та програмуванням, використовуються різні автоматизовані засоби навчання та перевірки. Можна виокремити два основні підходи до організації автоматизованого тестування студентських програм.

Перший підхід полягає у використанні спеціальних інформаційних систем, що виконують автоматичну перевірку програм. Одне з визначень поняття «автоматизована система тестування» наведено в [7] і сформульовано так: це комплекс програмних (або програмно-апаратних) засобів, які дозволяють в автоматичному режимі проводити тестування рішень користувачів у межах об'єкта тестування. Існує досить значна кількість таких систем, наприклад, такі як PCMS2, Contester, DOMjudge тощо, але однією з найпоширеніших є Ejudge [9]. Ejudge – це система для проведення різноманітних заходів, в яких потрібна автоматична перевірка програм. Незважаючи на те, що основне призначення системи Ejudge – це проведення олімпіад та інших змагань із програмування, система знайшла також широке застосування в навчанні програмування в різних університетах [8]. Але такий підхід має один істотний недолік. Система Ejudge не має бази завдань із тестами, викладач повинен їх розробити самостійно. Ця обставина суттєво ускладнює процес і потребує чимало часу.

Другим можливим варіантом застосування автоматизованих систем тестування для вивчення алгоритмів і структур даних, позбавлених зазначеного недоліку, є використання інтернет-ресурсів, розрахованих на застосування в процесі навчання або тренування вже готових завдань. Типовими представниками таких систем є сайт «Школа програміста» [10] або сайт «E-olymp» [11]. База даних кожного з цих двох сайтів містить понад тисячу завдань різного рівня складності. Методика використання цих ресурсів для підготовки учнів до олімпіад із програмування викладена в роботах [12–14]. Наприклад, на сайті «E-olymp» викладач може створювати групи учнів (студентів), запрошувати до цих груп учасників, які попередньо вже зареєструвалися на цьому сайті, створювати на базі наявних завдань різноманітні змагання, переглядати загальний рейтинг учасників (за результатами всіх змагань) та рейтинг за результатами окремих змагань. Хоча основним призначенням сайту є підготовка до

олімпіад, про що говорить сама назва сайту, його можна використовувати й на заняттях із програмування [14].

Безумовно, наявність готової бази завдань суттєво скорочує час на підготовку турніру. Але разом із тим відсутність можливості побачити код кожного рішення студента, відсутність можливості анулювати рішення, якщо його автор порушує встановлені викладачем вимоги, інші обмеження знижують ефективність такого підходу саме в процесі вивчення алгоритмізації.

### Мета та постановка завдання

Проведений аналіз довів перевагу використання автоматизованих систем тестування для перевірки програм студентів перед традиційними підходами, так званим «ручним тестуванням».

Разом із тим, не всі системи автоматизованого тестування можна легко адаптувати для проведення навчального процесу. Більшість систем, які аналізувалися, мають певні недоліки, що знижують ефективність їхнього застосування в цьому питанні.

Метою цієї статті є аналіз наявних підходів і пошук ефективного вирішення проблеми автоматизованого тестування програм студентів під час вивчення програмування та алгоритмізації.

### Виклад основного матеріалу

Автори статті вже кілька років під час проведення лабораторних і практичних занять із навчальних дисциплін «Програмування», «Основи алгоритмізації», «Алгоритми та структури даних» використовують систему DOTS. DOTS (Distributed Olympiad Test System) – це онлайн-платформа для навчання програмування й розвитку алгоритмічного мислення [15]. Система дозволяє студентам більш ефективно вивчати алгоритмізацію та програмування, дає змогу в будь-який момент відправити своє рішення у вигляді програми на перевірку та практично миттєво отримати вердикт перевіркою системи на своє рішення.

Широкі можливості система надає викладачам. Він може для кожної дисципліни для кожної групи створити лабораторну або практичну роботу (у системі вони називаються «турніри»), наповнити її завданнями з бази готових завдань, установити час початку та закінчення й отримує максимальний контроль над процесом виконання студентами завдань. Система підтримує 28 різних компіляторів усіх поширених мов програмування, таких як

C, C++, Java, C#, Python, Kotlin, Haskell, Ruby, JavaScript тощо.

Система має значну базу готових завдань, поділених на три великі групи – завдання для новачків на елементарні алгоритмічні конструкції, такі як розгалуження або цикли, завдання для вивчення алгоритмів і структур даних, а також завдання олімпіадного характеру. У межах кожної групи всі завдання також поділені на теми. Наприклад, для вивчення основ програмування передбачені окремо задачі на лінійні алгоритми, задачі, для реалізації яких потрібно використовувати умовні оператори й задачі на циклічні алгоритми. А для вивчення алгоритмів та структур даних виокремлено такі теми: алгоритми цілочисельної арифметики, алгоритми сортування, алгоритми на графах, алгоритми на деревах, завдання на динамічне програмування, алгоритми роботи з рядками, алгоритми обчислювальної геометрії, комбінаторні алгоритми тощо. Кожне завдання має свій рівень складності, і викладач легко та швидко може сформувати комплект завдань для кожної навчальної групи й навіть для окремих студентів з урахуванням рівня попередньої алгоритмічної підготовки. За необхідності викладач може будь-якої миті змінити набір завдань, додавши до нього або додаткові завдання, або, навпаки, замінити складне завдання на більш легке.

Усі завдання з алгоритмів та структур даних у системі мають однаковий формат. Існує чітке формулювання завдання, указаний формат вхідних даних, в якому прописані всі обмеження на вхідні дані, а також вказано порядок введення вихідних даних. Обов'язково вказується формат вихідних даних, щоб студент розумів, у якій послідовності виводити результати. Це дуже важливий момент, оскільки за умови автоматичної перевірки система порівнює результат роботи програми студента з результатом еталонного рішення й неправильний формат виведення результатів може призвести до вердикту WA – Wrong Answer (неправильна відповідь). Крім того, обов'язково вказується обмеження на час виконання програми й обмеження на кількість пам'яті, що виділяється. Отже, автоматично перевіряється, наскільки ефективним у плані обчислювальної складності є рішення студента. Якщо рішення виконується довше за встановлений час, система автоматично припиняє подальше тестування та видає вердикт TL – Time Limit Exceeded (перевищено час виконання). Якщо ж рішення вимагає більше

пам'яті, ніж передбачено обмеженнями завдання, система видає повідомлення ML – Memory Limit Exceeded (перевищено обмеження пам'яті).

Для кожного турніру система автоматично формує рейтингову таблицю, де вказується кількість балів, набраних кожним студентом групи за кожне завдання, загальна кількість балів, набрана студентом за всі завдання, а також повна статистика групи загалом. Викладач бачить, який середній бал у групі з кожного завдання, скільки студентів намагалося вирішувати кожне завдання та скільки з них виконали це завдання на повний бал.

Кількість максимальних балів можна встановлювати для кожного завдання окремо, і ця кількість може бути довільною. Отже, можна налаштувати систему оцінювання під будь-яку шкалу оцінок.

Значною перевагою цієї системи для викладача є змога переглянути вихідний код усіх рішень, які студенти відправили на перевірку. До того ж навіть якщо студент відправив повторно своє рішення за одним і тим самим завданням, у системі зберігаються всі його попередні рішення. У системі передбачено режим, в якому оцінка виставляється не автоматично, а лише після того, як викладач переглянув вихідний код програми та прийняв це рішення. Отже, можна контролювати не тільки відповідність результату еталонному, але й перевіряти, наскільки студент виконав додаткові вимоги до цього завдання, наприклад, реалізував саме той алгоритм або спосіб вирішення завдання, що визначив викладач. Крім того, якщо викладач дійшов висновку, що студент виконував завдання не самостійно або серйозно порушив вимоги до вирішення, то таке рішення викладач може скасувати.

### Висновки

У статті обґрунтовано використання автоматизованих систем тестування для вивчення основ програмування та алгоритмізації. Проаналізовано наявні підходи та виявлено їхні переваги та недоліки. Автоматизованою системою тестування для організації перевірки рішень студентів у процесі вивчення дисциплін «Програмування» та «Алгоритми та структури даних» запропоновано використовувати систему DOTS, яка, на думку авторів, є найбільш ефективним інструментом для досягнення поставленої мети. Перевагами цієї системи є наявність великої бази завдань різного рівня складності, починаючи від найпростіших і до завдань олімпіадного характеру, широту охоплення тем за алгорит-

мами та структурами даних, легкість та зручність створення нових турнірів, наповнення їх завданнями та додавання до цих турнірів студентів, можливість автоматичної перевірки всіх завдань та формування підсумкових таблиць із результатами цієї перевірки; можливість для викладача контролювати всі етапи виконання студентами завдання, переглядати вихідний код програм, відправлених на перевірку та керувати всім процесом. Досвід застосування системи DOTS дозволяє з усією впевненістю стверджувати, що використання автоматизованої системи тестування у вивченні основ програмування й алгоритмізації у підсумку дозволяє ефективно формувати компетентності випускників у проєктуванні, розробленні та аналізі алгоритмів, а також реалізації алгоритмів у вигляді програмних та інформаційних систем.

### Література

1. Computing Curricula 2020: Paradigms for Global Computing Education. URL: <https://dl.acm.org/doi/book/10.1145/3467967>.
2. Мелешко Є. В., Якименко М. С., Поліщук Л. І. Алгоритми та структури даних: навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. Кропивницький: Видавець – Лисенко В. Ф., 2019. 156 с.
3. Ткачук В. М. Алгоритми і структура даних: Навчальний посібник. Івано-Франківськ: Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
4. Коротеєва Т. О. Алгоритми та структури даних: навч. посібник. Львів: Видавництво Львівської політехніки, 2014. 280 с.
5. Зубков О. В., Семичева Н. Л. Роль автоматизованих тестируючих систем в процесі формування навчальних алгоритмічного мислення // Педагогический ИМИДЖ. 2019. № 4 (45). С. 550–565.
6. Айдаров К. А., Балакаева Г.Т. Применение тестирующей онлайн системы для развития практических навыков обучающихся в программировании // Материалы 48-ой научно-методической конференции «Роль высших учебных заведений в модернизации общественного сознания: переход к модели “Университет 4.0”». 18–19 января 2018 г. Алматы: Каз. ун-т, 2018. С. 17–20.
7. Гаспарян А. В., Тимошина Н. В. Особенности автоматизации проверки задач по программированию // ИТ-портал. 2018. № 2 (18). URL: <http://itportal.ru/-science/tech/osobennosti-avtomatizatsii-proverki/> (дата обращения: 02.07.2019).
8. Горчаков Л. В., Стась А. Н., Карташов Д. В. Обучение программированию с использованием системы Ejudge // Вестник Томского государственного педагогического университета. 2017. Вып. 9 (186). С. 109–112.
9. eJudge – система автоматической проверки программ. URL: <https://ejudge.ru/>
10. Школа программиста. URL: <https://acmp.ru/>
11. Интернет-портал организационно-методического обеспечения дистанционных олимпиад с программирования для одарованной молодежи. URL: <https://www.e-olymp.com>.
12. Алексеев А. В., Беляев С. Н. Подготовка школьников к олимпиадам по информатике с использованием вебсайта: учебно-методическое пособие для учащихся 7–11 классов. Ханты-Мансийск: РИО ИРО, 2008. 284 с.
13. Жуковский С. С. «E-olimp» – система автоматической проверки задач та проведення олімпіад з інформатики в інтернеті // Комп'ютер у школі та сім'ї. № 1 (65). 2008. С. 48–50.
14. Жуковский С. С. Використання Інтернет-порталу організаційно-методичного забезпечення «E-OLIMP» для підготовки одарованих школярів до олімпіади з інформатики // Комп'ютер у школі та сім'ї. 2010. № 8. С. 47–48. URL: [http://nbuv.gov.ua/UJRN/komp\\_2010\\_8\\_14](http://nbuv.gov.ua/UJRN/komp_2010_8_14).
15. DOTS – онлайн-платформа для обучения программированию и развития алгоритмического мышления. URL: <https://dots.org.ua/>

### References

1. Computing Curricula 2020: Paradigms for Global Computing Education. URL: <https://dl.acm.org/doi/book/10.1145/3467967>.
2. Meleshko Ye. V., Yakymenko M. S., Polishchuk L. I. Alhorytmy ta struktury danykh: Navchalnyi posibnyk dlia studentiv tekhnichnykh spetsialnostei dennoi ta zaochnoi formy navchannia. Kropyvnytskyi: Vydavets – Lysenko V. F., 2019. 156 s.
3. Tkachuk V. M. Alhorytmy i struktura danykh: Navchalnyi posibnyk. Ivano-Frankivsk: Vydavnytstvo Prykarpatskoho natsionalnoho universytetu imeni Vasyilia Stefanyka, 2016. 286 s.
4. Korotieieva T. O. Alhorytmy ta struktury danykh: navch. posibnyk. Lviv: Vydavnytstvo Lvivskoi politekhniki, 2014. 280 s.
5. Zubkov O. V., Semicheva N. L. Rol' avtomatizirovannyh testiruyushchih sistem v processe formirovaniya navykov algoritmicheskogo myshleniya // Pedagogicheskij IMIDZH. 2019. № 4 (45). S. 550–565.
6. Ajdarov K. A. Balakaeva G. T. Primenenie testiruyushchej onlajn sistemy dlya razvitiya prakticheskikh navykov obuchayushchihsya v programmirovani // Materialy 48-oj nauchno-metodicheskoy konferencii «Rol' vysshih uchebnyh zavedenij v modernizacii obshchestvennogo soznaniya: perekhod k modeli «Universitet 4.0»». 18–19 yanvarya 2018 g. Almaty: Kaz. un-t, 2018. S. 17–20.

7. Gasparyan A. V., Timoshina N. V. Osobennosti avtomatizatsii proverki zadach po programmirovaniyu // ITportal. 2018. № 2 (18). URL: <http://itportal.ru/science/tech/osobennosti-avtomatizatsii-proverki/>
8. Gorchakov L. V., Stas' A. N., Kartashov D. V. Obuchenie programmirovaniyu s ispol'zovaniem sistemy Ejudge // Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta. 2017. Vyp. 9 (186). S. 109–112.
9. eJudge – sistema avtomaticheskoy proverki programm. URL: <https://ejudge.ru/>
10. SHkola programmista. URL: <https://acmp.ru/>
11. Internet-portal orhanizatsiino-metodychnoho zabezpechennia dystantsiinykh olimpiad z prohramuvannia dlia obdarovanoi molodi. URL: <https://www.e-olymp.com>.
12. Alekseev A. V., Belyaev S. N. Podgotovka shkol'nikov k olimpiadam po informatike s ispol'zovaniem veb-sajta: uchebno-metodicheskoe posobie dlya uchashchihsya 7–11 klassov. Hanty-Mansijsk: RIO IRO, 2008. 284 s.
13. Zhukovskiy S. S. “E-olimp” – sistema avtomaticheskoy perevirky zadach ta provedennia olimpiad z informatiky v interneti // Kompiuter u shkoli ta simi. №1 (65). 2008. S. 48–50.
14. Zhukovskiy S. S. Vykorystannia Internet-portalu orhanizatsiino-metodychnoho zabezpechennia “E-OLIMP” dlia pidhotovky obdarovanykh shkolariv do olimpiady z informatiky // Kompiuter u shkoli ta simi. 2010. № 8. S. 47–48.
15. DOTS – onlajn-platforma dlya obucheniya programmirovaniyu. URL: <https://dots.org.ua/>

**Щербаків Олександр Всеволодович**, к.т.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38050-636-20-09, [oleksandr.shcherbakov@hneu.net](mailto:oleksandr.shcherbakov@hneu.net),  
**Скорін Юрій Іванович**, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, тел. +38066-748-47-51, [skorin.yuriy@gmail.com](mailto:skorin.yuriy@gmail.com).

### Improving the efficiency of learning programming and algorithmization through the use of an automated testing system

**Abstract. Problem.** One of the main fundamental disciplines in training of future IT-specialists are the disciplines “Programming” and “Algorithms and data structures”. The issue of testing the programs

developed by students becomes relevant. Obviously, the so-called manual testing is too time-consuming and inefficient, so the increasing number of teachers use automated testing systems. **Goal.** The goal is to analyze the existing approaches and find an effective solution to the problem of automated testing of student programs when studying the basics of programming and algorithmization. **Methodology.** The analysis carried out showed that there are two main approaches to the organization of automated testing of student programs. The first is the use of special information systems that automatically check the programs. The second is the use of Internet resources designed to use ready-made tasks in the process of learning or training. **Results.** The use of automated testing systems in the study of programming, algorithms and data structures was substantiated. The existing approaches were analyzed and their advantages and disadvantages were identified. It was proposed to use the DOTS system, which, according to the authors, is the most effective tool to achieve this goal. **Originality.** The advantages of the proposed approach are the presence of a large database of tasks of various levels of complexity, the ease and convenience of creating new tournaments, filling them with tasks and adding students to these tournaments, the ability to automatically check all tasks and generate final tables with the results. This gives the possibility for the teacher to control all stages of the assignment by students, view the source code of the programs sent for verification and manage the entire process. **Practical value.** The use of an automated testing system in the study of the basics of programming and algorithmization ultimately makes it possible to effectively form the competence of graduates in the design, development and analysis of algorithms, as well as in the implementation of algorithms in the form of software and information systems.

**Key words:** programming, algorithmization, automated testing system, efficiency of educational process.

**Shcherbakov Oleksandr**, Ph.D., Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38050-636-20-09, [oleksandr.shcherbakov@hneu.net](mailto:oleksandr.shcherbakov@hneu.net),  
**Skorin Yuriy**, Ph.D., Assoc. Prof. Information Systems Department, Kharkiv National Economic University named after S. Kuznets, tel. +38066-748-47-51, [skorin.yuriy@gmail.com](mailto:skorin.yuriy@gmail.com).