

**Камінський Олег Євгенович**

*кандидат економічних наук, доцент,*

*доцент кафедри інформаційного менеджменту*

*Київський національний економічний університет імені Вадима Гетьмана*

**Каминский Олег Евгеньевич**

*кандидат экономических наук, доцент,*

*доцент кафедры информационного менеджмента*

*Киевский национальный экономический университет имени Вадима Гетьмана*

**Kaminsky Oleg**

*PhD in Enterprise Economics, Associate Professor,*

*Associate Professor of Information Management Department*

*Kyiv National Economic University named after Vadym Hetman*

DOI: 10.25313/2520-2294-2018-7-4038

## МЕТОДОЛОГІЯ РОЗРОБКИ ХМАРНИХ СЕРВІСІВ В УМОВАХ ІЄРАРХІЇ ХМАРИ

## МЕТОДОЛОГИЯ РАЗРАБОТКИ ОБЛАЧНЫХ СЕРВИСОВ В УСЛОВИЯХ ИЕРАРХИИ ОБЛАКА

## METHODOLOGY FOR DEVELOPING CLOUD SERVICES IN A CLOUD HIERARCHY

**Анотація.** Парадигма хмарних обчислень та пов'язані з нею технології набувають популярності серед підприємств IT-галузі через те, що вони дозволяють використовувати дорогі технології за низькою ціною. В даний час хмарні сервіси найчастіше розробляються на базі моделі PaaS, яка впроваджена багатьма провайдерами. У літературі ми знаходимо деякі наукові дослідження, що описують структури та архітектури для розробки хмарного програмного забезпечення, однак бракує загальної методології, яка охоплювала б весь життєвий цикл розробки хмарних сервісів. Багато розробників програмного забезпечення намагаються використовувати переваги хмарних обчислень і хочуть отримати доступ до хмарних середовищ за низькою вартістю та легким доступом. Багато проблем виникає при розробці та впровадженні хмарних платформ відносно моделей програмування в хмарі, архітектури хмарних сервісів, API та послуг, які вони надають. Процеси створення і використання хмарних сервісів мають свою економічну складову, вони характеризуються витратами праці і коштів з одного боку, і отриманням прямого або непрямого прибутку – з іншого. Складність сучасних хмар визначає необхідність звернення до методів економіко-математичного моделювання. Метою статті є розробка теоретико-методологічних підходів та створення бази для нової моделі розробки програмного забезпечення хмарних сервісів, яка може задовольнити вимоги користувачів щодо якості та вартості послуг, та враховує багаторівневу структуру хмарного середовища. Сутність цього підходу полягає у систематизації даних щодо функціональної придатності різних структурних елементів хмарних сервісів, у визначенні наборів можливих платформ та елементів і в порівняльному їх аналізі з погляду тривірневої ієрархії хмари для оцінювання проектів розробки хмарних сервісів.

**Ключові слова:** інформаційні технології, хмарні обчислення, хмарні сервіси, моделі, хмарні платформи, системний аналіз.

**Аннотация.** Парадигма облачных вычислений и связанные с ней технологии приобретают популярность среди предприятий IT-отрасли из-за того, что они позволяют использовать дорогостоящие технологии по низкой цене. В настоящее время облачные сервисы чаще всего разрабатываются на основе модели PaaS, которая внедрена многими провайдерами. В литературе мы находим некоторые научные исследования, описывающие структуры и архитектуры

для разработки облачного программного обеспечения, но не хватает общей методологии, которая охватывала бы весь жизненный цикл разработки облачных сервисов. Многие разработчики программного обеспечения пытаются использовать преимущества облачных вычислений и хотят получить доступ к облачным средам по низкой стоимости и легким доступом. Много проблем возникает при разработке и внедрении облачных платформ относительно моделей программирования в облаке, архитектуры облачных сервисов, API и услуг, которые они предоставляют. Процессы создания и использования облачных сервисов имеют свою экономическую составляющую, они характеризуются затратами труда и средств с одной стороны, и получением прямого или косвенного дохода – с другой. Сложность современных облаков определяет необходимость обращения к методам экономико-математического моделирования. Целью статьи является разработка теоретико-методологических подходов и создания базы для новой модели разработки программного обеспечения облачных сервисов, которая может удовлетворить требования пользователей к качеству и стоимости услуг, и учитывает многоуровневую структуру облачной среды. Суть этого подхода заключается в систематизации данных о функциональной пригодности различных структурных элементов облачных сервисов, в определении наборов возможных платформ и элементов и в сравнительном их анализе с точки зрения трехуровневой иерархии облака для оценки проектов разработки облачных сервисов.

**Ключевые слова:** информационные технологии, облачные вычисления, облачные сервисы, модели, облачные платформы, системный анализ.

**Summary.** The paradigm of cloud computing and its associated technology is gaining momentum among IT companies because they allow you to use expensive technology at a low cost. Nowadays, cloud applications are developed in Platforms as a Service following the PaaS architecture imposed by several providers. We find in the literature some works describing frameworks and architectures for cloud software development, but there is a lack of a generic methodology which covers the whole application development lifecycle. Many software developers trying to get the benefits of cloud computing and want to access the cloud environments at low cost and easy access. Many issues are encountering in development and deployment of these platforms regarding programming models, application architecture, APIs and services it provided. The processes of development and using cloud services have their own economic component, they are characterized by labor costs and funds on the one hand, and the receipt of direct or indirect income – on the other. The complexity of modern clouds determines the need to apply methods of economic and mathematical modeling. The purpose of the article is to develop theoretical and methodological approaches and create a basis for a new model of software development for cloud services that can meet the requirements of users regarding the quality and cost of services, and takes into account the multi-level structure of cloud environments. The essence of this approach is to systematize the data on the functional suitability of the various structural elements of cloud services, to determine the sets of possible platforms and elements, and in their comparative analysis in terms of the three-tier hierarchy of the cloud for the evaluation of cloud service development projects.

**Key words:** information technologies, cloud computing, cloud services, models, cloud platforms.

**Постановка проблеми.** Хмарні технології забезпечують доступ до інформаційних ресурсів (обчислювальних потужностей, IT-інфраструктури, сховищ даних, операційних систем, мереж, програмного забезпечення та середовищ розробки) в якості послуги, за допомогою інтернет-технологій, і на основі аналізу попиту серед споживачів [1–2]. Використовуючи хмарні сервіси, користувачі можуть зберігати свої дані на серверах і отримувати доступ до своїх даних з будь-якого місця через Інтернет. Користувач не повинен турбуватися про технічний стан системи або помилки жорсткого диска. Хмарні обчислення надають хмарні сервіси всім, скрізь і в будь-який час за низькою ціною та масштабованістю [3]. Парадигма хмарних обчислень та пов'язані з нею технології набувають популярності серед підприємств IT-галузі саме через те, що вони дозволяють використовувати дорогі технології за низькою ціною.

Хмарні обчислення надають безліч переваг для розробки програмного забезпечення [4], зокрема завдяки масштабованості та гнучкому розподілу ресурсів, без необхідності турбуватися про базову інфраструктуру чи налаштування середовища розробки. Іншою перевагою цієї нової парадигми програмного забезпечення є те, що компанії та організації можуть створювати масові розподілені програмні системи шляхом динамічного поєднання різних хмарних сервісів. Середовище хмарних обчислень складається з трьох основних шарів, тобто програмного забезпечення як сервіс (SaaS), платформа як служба (PaaS) та інфраструктура як сервіс (IaaS) [5].

Але у контексті хмарних обчислень хмарні провайдери за моделлю PaaS встановлюють жорстку архітектуру для розробки програмного забезпечення, а також вимагають від розробників підготовки IT-професіоналів, оскільки їм потрібні технічні

знання (наприклад, для Google App Engine, користувачеві потрібно освоїти мову програмування Java або Python).

Існуючі моделі процесів розробки програмного забезпечення не є достатньою для виконання вимог хмарного середовища. Необхідно розробити нові практики та моделі розробки програмного забезпечення хмарних сервісів для врахування нових тенденцій розвитку хмарних технологій.

**Аналіз останніх досліджень та публікацій.** В даний час існують різні моделі розробки програмного забезпечення, які є доступними та використовуються різними постачальниками для виконання вимог хмарного середовища [6]. В роботі [7] представлено новий методологічний підхід, для полегшення процесу розробки програмного забезпечення, використовуючи хмарну платформу але не враховуючи архітектуру хмарного середовища. Дослідники Guha R., та Al-Dabass D., в роботі [8] проаналізували та розширили існуючу Agile-методологію, доповнили її функціями хмарного провайдера та адаптували запропоновану модель для процесу розробки хмарних сервісів. Але дослідники зосередили свою увагу лише на конкретному хмарному провайдеру, а також на його конкретній моделі програмного забезпечення та архітектури.

Вчені K. Tang та J. M. Zhang у дослідженні [7] запропонували новий підхід до розробки хмарних додатків. Вони запропонували архітектуру життєвого циклу розробки хмарних додатків, в якій розробник сервісу зосереджує свою увагу лише на розробці самого сервісу, а не на проблемах хмарного середовища. Запропонований підхід має узагальнену структуру для різних хмарних платформ, які надають підтримку та платформи для розробки додатків. Оскільки кожна платформа має свої прикладні архітектури та моделі програмування, відповідно їх API та бізнес-моделі мають свої переваги та обмеження. Запропонована архітектура дозволила вирішити проблему відмінності між різнорідними хмарними платформами. Без врахування обмежень API та додаткових витрат розробник сервісів може створювати та розгортати нові хмарні сервіси або переносити існуючі програмні додатки (додатки CRM, корпоративні системи) в хмару. Також в роботі було досліджено переваги та недоліки моделей IaaS та PaaS та акцентовано увагу на питаннях архітектурного та програмного рівнів, таких як масштабованість та залежність від провайдера. Переваги та недоліки розробки хмарних сервісів на рівні підприємства були класифіковані. Однак, запропонований підхід орієнтований на програмні додатки, і автома-

тизовані інструменти розробки мають обмежене використання. Запропонована модель була розроблена лише на теоретичній основі, математичні та аналітичні результати відсутні.

Підбиваючи підсумок, можна заявити, що на теперішній час немає єдиного наукового дослідження, що описує всі практики та моделі розробки програмного забезпечення хмарних сервісів з врахуванням ієрархії хмарного середовища та економічних чинників.

**Формулювання цілей статті.** Метою статті є розробка теоретико-методологічних підходів та створення бази для нової моделі розробки програмного забезпечення хмарних сервісів, яка може задовольнити вимоги користувачів щодо якості та вартості послуг, та враховує багаторівневу структуру хмарного середовища.

**Основний матеріал дослідження.** Часто при побудові сервісу потрібно зробити вибір не одного структурного елементу сервісу та технології, а деякої сукупності різних взаємозв'язаних елементів та технологій. Як ми знаємо, хмарні обчислення визначаються за їхніми рівнями та моделями доставки сервісу (SaaS, PaaS та IaaS). Наприклад, для створення хмарного сервісу потрібно вибрати такі структурні елементи, як засіб розробки (мови c#, java, go, elixir, javascript тощо), інфраструктуру (сервери, засоби комутації, засоби віртуалізації тощо), систему управління базами та сховищами даних (MySQL, MS SQL, Redis, MongoDB, Oracle, PostgreSQL тощо). Тобто необхідно прийняти рішення відразу на кількох рівнях хмарної структури:

$$\{ \langle E^u, F^u, R^u \rangle \}, u = \overline{1, nl}, \quad (1)$$

де  $u$  — номер рівня хмарної моделі;

$nl$  — кількість рівнів (IaaS, PaaS, SaaS).

$E = \{ E_i \}, i = \overline{1, n}$  — множина структурних елементів хмарного сервісу, який розробляється ( $n$  — кількість таких елементів);

$F = \{ F_j \}, j = \overline{1, m}$  — множина функцій сервісу ( $m$  — кількість усіх функцій);

$R = \{ r_{ij} \}, i = \overline{1, n}, j = \overline{1, m}$  — матриця, яка визначає реалізацію елементом  $E_i$  функції  $F_j$ .

Структурні елементи хмарного сервісу на різних рівнях хмари можуть бути несумісні між собою, наприклад, деякі системи управління базами даних можуть працювати тільки в рамках якоїсь однієї операційної системи, яка встановлена на віртуальних машинах. Для визначення можливості спільного використання елементів сервісу різних рівнів пропонується використовувувати матриці сумісності:

$$\{c^{ug}\}, u = \overline{1, nl}, g = \overline{1, nl}, u \neq g; \quad (2)$$

$$c^{ug} = \{c_{ij}^{ug}\}, i = \overline{1, n_u}, g = \overline{1, n_g},$$

$$c_{ij}^{ug} = \begin{cases} 1, \text{якщо платформи } s_i^u \text{ та } s_j^g \text{ є сумісними} \\ 0, \text{в інших випадках} \end{cases}$$

Отже, матриці сумісності характеризують сумісність кожного елементу певного рівня хмари з усіма платформами, елементами та технологіями інших рівнів. Рис. 1 ілюструє завдання вибору сукупності елементів та технологій. Пунктирні лінії показують відносини сумісності між системами.

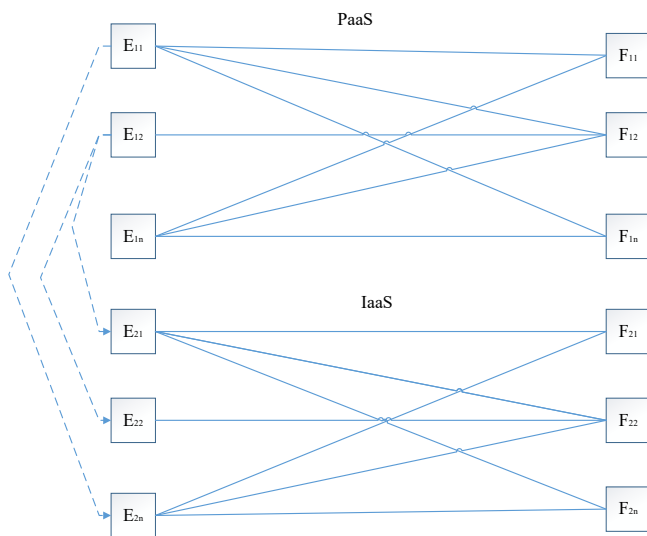


Рис. 1. Граф вихідних даних для систем з двох рівнів (PaaS та IaaS).

Відповідно, табл. 1–3 містять матриці сумісності різних рівнів.

У деяких випадках за розвинутої відкритої архітектури хмари для розглянутого хмарного сервісу може мати значення тільки сумісність груп елементів сервісу сусідніх рівнів, тоді кількість матриць сумісності може бути зменшена. Для прийняття рішення в умовах ієрархії хмари можна скористатися одним зі способів: або зібрати в єдину матрицю всі можливі поєднання структурних елементів усіх рівнів, або провадити аналіз в кілька етапів. Але перший варіант може призвести до надмірного збільшення матриці  $R$ , однак можливі й ситуації, коли подібного не станеться, наприклад, якщо потрібно вибрати СУБД (Oracle чи Mongo DB) або операційну систему. На відміну від великої кількості платформ для розробки та мов програмування, кількість СУБД і операційних систем обмежена.

Отже, для побудови єдиної матриці необхідно:

- 1) об'єднати стовпці матриць усіх рівнів, що являють множину функцій  $F' = F^1 \cap F^2 \cap \dots \cap F^{nl}$ ;
- 2) додати рядки для всіх можливих комбінацій елементів структури всіх рівнів:  $E' = \{(e_{i1}^1, e_{i2}^2, \dots, e_{inl}^{nl}) | C_{iu,ig}^{ug} = 1, \forall u, g = \overline{1, nl}\}$ , тобто відбираються такі комбінації структурних елементів хмарних сервісів, для яких усі компоненти сумісні між собою.

В інших випадках необхідно провадити аналіз в кілька етапів, виключаючи структурні елементи,

Таблиця 1

Матриця сумісності елементів сервісу першого та другого рівнів хмари (IaaS та PaaS)

	Елемент 21	Елемент 22	Елемент 23	Елемент 24
Елемент 11	1	1	0	0
Елемент 12	1	1	1	1
Елемент 13	0	1	1	1

Таблиця 2

Матриця сумісності елементів сервісу першого та третього рівнів хмари (IaaS та SaaS)

	Елемент 31	Елемент 32	Елемент 23
Елемент 11	1	0	0
Елемент 12	1	1	1
Елемент 13	1	1	1

Таблиця 3

Матриця сумісності елементів сервісу другого та третього рівнів хмари (SaaS та PaaS)

	Елемент 21	Елемент 22	Елемент 23	Елемент 24
Елемент 31	1	1	1	1
Елемент 32	1	1	0	1
Елемент 33	1	1	1	1



які не відповідають вимогам користувача на кожному з рівнів.

Отже, на кожному рівні провадиться вибір елементів або платформ для розробки, які найбільш задовольняють вимоги користувача (відповідно до стандартної методики системного аналізу). Далі з відібраних систем кожного рівня формується єдина матриця  $R'$ , як це описано раніше. З декартового добутку відібраних структурних елементів сервісу відкидаються ті, які не відповідають умовам сумісності  $C^{12}$ ,  $C^{13}$ ,  $C^{23}$ .

Для запропонованого прикладу будемо відбирати структурні елементи на кожному рівні за розрахованими матрицями поглинання  $U^1$ ,  $U^2$ ,  $U^3$ . Виберемо порогові значення  $\varepsilon_u^1 = 0,77$ ,  $\varepsilon_u^2 = 0,77$ ,  $\varepsilon_u^3 = 0,77$  (у загальному випадку порогові значення для різних рівнів можуть бути різні).

Для першого рівня:  $U_{01}^1 = 0,77$ ,  $U_{02}^1 = 1$ ,  $U_{03}^1 = 0,77$ ; оскільки  $U_{01}^1 \geq \varepsilon_u^1$ ,  $U_{02}^1 \geq \varepsilon_u^1$ ,  $U_{03}^1 \geq \varepsilon_u^1$ , то вибираємо елементи  $E'_1 = \{\text{елемент 11, елемент 12, елемент 13}\}$ .

Для другого рівня:  $U_{01}^2 = 1$ ,  $U_{02}^2 = 0,5$ ,  $U_{03}^2 = 1$ ,  $U_{04}^2 = 0,5$ ; відповідно до граничного значення  $\varepsilon_u^2 = 0,77$  відбираємо елементи  $E'_2 = \{\text{елемент 21, елемент 23}\}$ .

Для третього рівня:  $U_{01}^3 = 1$ ,  $U_{02}^3 = 0,5$ ,  $U_{03}^3 = 0,5$ . Відповідно до граничного значення  $\varepsilon_u^3 = 0,67$  вибираємо елемент  $E'_3 = \{\text{елемент 31}\}$ .

З декартового добутку  $E^1 \times E^2 \times E^3$  відкидаються елементи, які не відповідають умовам сумісності:

$$E'' = \left\{ (e_{i1}^1, e_{i2}^2, \dots, e_{inl}^n) \mid C_{iu,ig}^{ug} = 1, \forall u, g = \overline{1, n} \right\}$$

У результаті дістаємо таку підмножину декартового добутку, яка задовольняє умови сумісності:

(елемент 11, елемент 21, елемент 31)

(елемент 12, елемент 21, елемент 31)

(елемент 12, елемент 23, елемент 31)

(елемент 13, елемент 23, елемент 31).

Множина функцій трьох рівнів об'єднується:

$$F' = F^1 \cap F^2 \cap \dots \cap F^n.$$

Запропоновану схему можна модифікувати. Для того щоб під час проведення аналізу на окремих рівнях уникнути оптимізації за кожним з них, незалежно від інших можна зв'язати результати аналізу на

кожному рівні. Після відбору структурних елементів хмарного сервісу на першому рівні SaaS (відповідно до стандартної методики аналізу складних систем) множина елементів другого рівня модифікується, відкидаються елементи, які не сумісні хоча б з одним з відібраних на першому рівні:  $E^{2'} = \{e_i^2 \mid \exists e_j^1, c_{ji}^{12} = 1\}$ . Далі за стандартною методикою провадиться аналіз структурних елементів, що належать до другого рівня. А потім результати аналізу першого і другого рівнів разом з відповідними матрицями сумісності використовуються для скорочення множини елементів третього рівня:  $E^{3'} = \{e_i^3 \mid \exists e_j^1, c_{ji}^{13} = 1 \mid \exists e_k^2, c_{ki}^{23} = 1\}$ .

Доповнений і вдосконалений метод аналізу складних систем може використовуватися для вибору елементів і побудови проектів хмарних сервісів на всіх рівнях хмар.

**Висновки з даного дослідження і перспективи подальших досліджень у даному напрямі.** Особливістю хмарних сервісів є їх слабка структурованість в порівнянні з традиційними інформаційними системами, а також свобода користувачів і розробників в плані кількісного і якісного розвитку системи. Процеси створення і використання хмарних сервісів мають свою економічну складову, вони характеризуються витратами праці і коштів з одного боку, і отриманням прямого або непрямого прибутку — з іншого. Складність сучасних хмар визначає необхідність звернення до методів економіко-математичного моделювання. В статті була розроблена нова модель формування структури хмарного сервісу на основі методів аналізу складних систем. Сутність цього підходу полягає у систематизації даних щодо функціональної придатності різних структурних елементів хмарних сервісів, у визначенні наборів можливих платформ та елементів і в порівняльному їх аналізі з погляду трирівневої ієрархії хмари. Подальші дослідження мають стосуватися вдосконалення моделі шляхом встановлення залежності між функціями хмарного сервісу та групування даних функцій. Розглянуті методи та моделі можуть використовуватися в різних областях економіки та управління для оцінювання проектів розробки хмарних сервісів.

#### Література

1. M. Uddin, J. Memon, R. Alsaqour, A. Shah and M. Z. A. Rozan, «Mobile Agent Based Multi-Layer Security Framework for Cloud Data Centers» / *Indian Journal of Science and Technology*. — том 8. — № 12. — 2015. — с. 1–10.
2. Waqas A., Mahessar A. W., Mahmood N., Bhatti Z., Karbasi M. i Shah A., «Transaction Management Techniques and Practices In Current Cloud Computing Environments: A Survey» / *International Journal of Database Management Systems*. — том 7. — № 1. — 2015. — с. 41–59.
3. Shah M., Shah A. S. та Ijaz I., «Implementation of User Authentication as a Service for Cloud Network» / *International Journal of Grid and Distributed Computing (IJGDC)*. — том 9. — № 10. — 2016. — с. 197–210.
4. Uddin M., Rahman A. A., Shah A. та Memon J. «Virtualization Implementation Approach for Data Centers to Maximize Performance» / *Asian Network for Scientific Information (ANSINET)*. — том 5. — № 2. — 2012. — с. 45–57.
5. Chhabra B., Verma D. i Taneja B. «Software Engineering Issues from the Cloud Application Perspective» / *International Journal of Information Technology and Knowledge Management*. — 2010. — том 2, 2. — с. 669–673.
6. Ali M. F., Bashar A. та Shah A., «SmartCrowd: Novel Approach to Big Crowd Management using Mobile Cloud Computing» / *2015 International Conference on Cloud Computing (ICCC)*. — 2015. — с. 1–4.
7. Tang K., Zhang J. M. and Feng C. H., «Application Centric Lifecycle Framework in Cloud» / *International Conference on e-Business Engineering*. — том 0. — 2011. — с. 329–334.
8. Guha R., та Al-Dabass D., «Impact of Web 2.0 and Cloud Computing Platform on Software Engineering» / *International Symposium on Electronic System Design*. — 2010. — с. 213–318.

#### References

1. M. Uddin, J. Memon, R. Alsaqour, A. Shah and M. Z. A. Rozan, «Mobile Agent Based Multi-Layer Security Framework for Cloud Data Centers» / *Indian Journal of Science and Technology*. — tom. 8. — # 12. — 2015. — s. 1–10.
2. Waqas A., Mahessar A. W., Mahmood N., Bhatti Z., Karbasi M. i Shah A., «Transaction Management Techniques and Practices In Current Cloud Computing Environments: A Survey» / *International Journal of Database Management Systems*. — tom. 7. — # 1. — 2015. — s. 41–59.
3. Shah M., Shah A. S. and Ijaz I., «Implementation of User Authentication as a Service for Cloud Network» / *International Journal of Grid and Distributed Computing (IJGDC)*. — tom. 9. — #10. — 2016. — s. 197–210.
4. Uddin M., Rahman A. A., Shah A. and Memon J., «Virtualization Implementation Approach for Data Centers to Maximize Performance» / *Asian Network for Scientific Information (ANSINET)*. — tom. 5. — # 2. — 2012. — s. 45–57.
5. Chhabra B., Verma D. i Taneja B. «Software Engineering Issues from the Cloud Application Perspective» / *International Journal of Information Technology and Knowledge Management*. — 2010. — tom. 2, 2. — s. 669–673.
6. Ali M. F., Bashar A. and Shah A., «SmartCrowd: Novel Approach to Big Crowd Management using Mobile Cloud Computing» / *2015 International Conference on Cloud Computing (ICCC)*. — 2015. — s. 1–4.
7. Tang K., Zhang J. M. and Feng C. H., «Application Centric Lifecycle Framework in Cloud» / *International Conference on e-Business Engineering*. — tom. 0. — 2011. — s. 329–334.
8. Guha R., та Al-Dabass D., «Impact of Web 2.0 and Cloud Computing Platform on Software Engineering» / *International Symposium on Electronic System Design*. — 2010. — s. 213–318.