

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження оптимального набору показників продуктивності веб-базованої
інформаційної системи
(тема)

Виконав:
студент 2 курсу, групи ІУСТМ-20-1
Хоменко К. О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі
системи та технології
(повна назва освітньої програми)

Керівник проф. Євланов М. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Петров К.Е.
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Інформаційних управляючих систем
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
« ____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентці Хоменко Ксенії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження оптимального набору показників продуктивності веб-базованої інформаційної системи

затверджена наказом університету від 05 листопада 2021 р. № 1645 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 08 грудня 2021 р.

3. Вихідні дані до роботи: науково-технічні публікації та інтернет джерела з тематики атестаційної роботи

4. Перелік питань, що потрібно опрацювати в роботі: аналіз існуючих методів оцінювання продуктивності веб-базованих інформаційних систем; опис теоретичного вирішення задачі; дослідження методів регресійного аналізу результатів моніторингу продуктивності веб-базованої інформаційної системи; практична апробація результатів магістерської кваліфікаційної роботи.

5. Перелік графічного матеріалу із зазначенням схем, плакатів, комп'ютерних ілюстрацій (слайдів): схема загальної архітектури веб-застосунку, схеми однорівневої, дворівневої та тривірневої архітектури веб-застосунку, схема моделі зчитування показників продуктивності сервісів, відображення частини значень показників продуктивності, зібраних у ході проведення моніторингу експлуатованої ВБІС, відображення підсумку гребеневої регресії.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз існуючих методів оцінювання продуктивності веб-базованих інформаційних систем	08.11.2021 – 11.11.2021	
2	Опис теоретичного вирішення задачі	12.11.2021 – 16.11.2021	
3	Дослідження методів регресійного аналізу результатів моніторингу продуктивності веб-базованої інформаційної системи	17.11.2021 – 21.11.2021	
4	Практична апробація результатів магістерської кваліфікаційної роботи	22.11.2021 – 28.11.2021	
5	Оформлення пояснювальної записки	29.11.2021 – 01.12.2021	
6	Оформлення графічного матеріалу	02.12.2021 – 04.12.2021	
7	Розробка презентаційного матеріалу	05.12.2021	
8	Перевірка на плагіат	06.12.2021	
9	Захист кваліфікаційної роботи	09.12.2021	

Дата видачі завдання 08 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Євланов М. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 85 сторінок, 7 рисунків, 3 таблиці, 1 додаток, 21 джерело.

ВЕБ-ЗАСТОСУНОК, ІНФОРМАЦІЙНА СИСТЕМА, МЕТОДИ РЕГРЕСІЙНОГО АНАЛІЗУ, МОНІТОРИНГ, ПАСИВНИЙ ЕКСПЕРИМЕНТ, ПОКАЗНИКИ ПРОДУКТИВНОСТІ

Об'єктом дослідження є веб-базована інформаційна система у ході промислової експлуатації на підприємстві. Розглядаються лише існуючі готові системи, передані розробником замовнику і введені в експлуатацію.

Метою кваліфікаційної роботи є побудова моделі, яка дасть можливість кількісно оцінити ефективність експлуатації веб-базованої інформаційної системи, ґрунтуючись на відомому наборі показників, значення яких вимірюються в ході моніторингу експлуатації системи.

Методами дослідження є методи регресійного аналізу, що дозволяють визначити оптимальний набір показників продуктивності експлуатованої веб-базованої інформаційної системи.

Теоретичними результатами проведення наукового дослідження є опис моделі, яка дозволить визначити оптимальний набір показників продуктивності, для яких здійснюватиметься моніторинг. Практичними результатами є використання даної моделі з метою виявлення на практиці оптимального набору показників продуктивності.

Новизна результатів проведення наукового дослідження полягає у розробці вдосконаленої моделі, за якою проводитиметься подальший моніторинг продуктивності веб-базованих інформаційних систем. Така модель дозволить скоротити час обробки результатів моніторингу і, як наслідок, сприятиме зниженню витрат на проведення моніторингу.

ABSTRACT

Explanatory note for the qualification work: pages, 7 figures, 3 tables, 1 attachment, sources.

INFORMATION SYSTEM, MONITORING, PASSIVE EXPERIMENT, PERFORMANCE INDICATORS, REGRESSIONAL ANALYSIS METHODS, WEB APPLICATION

The object of the study is a web-based information system observed during industrial operation at the enterprise. Only existing web-based systems handed over by the developer to the customer and put into operation are considered.

The purpose of the qualification work is to build a model that will quantify the effectiveness of the web-based information system based on a known set of indicators the values of which are measured during the monitoring of system operation.

The research methods are the methods of regression analysis which allow to determine the optimal set of performance indicators of the operated web-based information system.

The theoretical result of the research is a description of the model that will determine the optimal set of performance indicators for which monitoring will be carried out. The practical result of the research is the use of created model in order to identify in practice the optimal set of performance indicators.

The novelty of the results of the research is the development of an improved model based on which the further monitoring of the performance of web-based information systems will be carried out. This model will reduce the processing time of monitoring which will help to reduce the cost of the monitoring.

ЗМІСТ

	С.
Скорочення та умовні позначки.....	8
Вступ	9
1 Аналіз існуючих методів оцінювання продуктивності веб-базованих інформаційних систем	10
1.1 Дослідження існуючих видів моделей веб-базованих інформаційних систем	10
1.2 Дослідження проблеми оцінювання продуктивності веб-базованих інформаційних систем	19
1.3 Висновки та постановка задачі дослідження	26
2 Опис теоретичного вирішення задачі	28
2.1 Опис основного способу моніторингу експлуатації веб-базованої інформаційної системи	28
2.2 Дослідження багатовимірних моделей моніторингу продуктивності веб-базованої інформаційної системи	34
2.3 Дослідження показників, що впливають на продуктивність веб-базованої інформаційної системи	37
2.4 Висновки до другого розділу	39
3 Дослідження методів регресійного аналізу результатів моніторингу продуктивності веб-базованої інформаційної системи	41
3.1 Дослідження методу покрокової регресії	41
3.2 Дослідження методу гребеневої регресії	47
3.3 Порівняльний аналіз та обґрунтування вибору методу регресійного аналізу	49
3.4 Висновки до третього розділу	55
4 Практична апробація результатів магістерської кваліфікаційної роботи	56
4.1 Опис експлуатованої веб-базованої інформаційної системи	56

4.2 Проведення аналізу результатів моніторингу із використанням методу гребеневої регресії	61
4.3 Висновки до четвертого розділу	63
Висновки	65
Перелік джерел посилання	66
Додаток А. Графічний матеріал кваліфікаційної роботи	68

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВБІС – веб-базована інформаційна система;

ІС – інформаційна система;

ПЗ – програмне забезпечення;

CRM – customer relationship management;

ISO – international organization for standardization;

ITIL – information technology infrastructure library;

KPI – key performance indicator;

OLA – operational level agreement;

RSS – really simple syndication;

SLA – service level agreement;

UC – underpinning contracts.

ВСТУП

Веб-базовані застосунки за рахунок зручності використання нині введені в експлуатацію на великій кількості підприємств. Клієнт-серверна архітектура дозволяє здійснювати швидкий обмін даними та надавати користувачеві необхідну інформацію за запитом. Але зі зростом популярності таких систем, звісно, зростає і кількість користувачів. Внаслідок цього можуть виникати проблеми із продуктивністю веб-базованих інформаційних систем.

При використанні веб-базованої інформаційної системи найважливішим чинником задоволення користувачів при користуванні системою є саме продуктивність, адже вона найповніше характеризується часом відгуку інформаційної системи [1].

Метою магістерської кваліфікаційної роботи є побудова моделі, яка дасть можливість оцінити ефективність експлуатації веб-застосунку, ґрунтуючись на відомому наборі показників, значення яких вимірюються в ході моніторингу експлуатації системи.

Для досягнення визначеної мети у кваліфікаційній роботі мають бути вирішені такі задачі: аналіз існуючих методів оцінювання продуктивності веб-базованих інформаційних систем; опис теоретичного вирішення задачі; дослідження методів регресійного аналізу результатів моніторингу продуктивності веб-базованої інформаційної системи; практична апробація результатів магістерської кваліфікаційної роботи.

Актуальність кваліфікаційної роботи полягає у тому, що результатом проведення дослідження є побудова вдосконаленої моделі, що дозволить визначити оптимальний набір показників продуктивності веб-базованої інформаційної системи.

Кваліфікаційну роботу оформлено згідно з державним стандартом із оформлення звітів у сфері науки і техніки [2].

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОЦІНЮВАННЯ ПРОДУКТИВНОСТІ ВЕБ-БАЗОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Дослідження існуючих видів моделей веб-базованих інформаційних систем

Досягнення в галузі технологій, безпеки та швидкості Інтернету значно збільшили потенційні можливості веб-базованих інформаційних систем. Їхня актуальність зростає з кожним днем, оскільки веб-застосунки мають значні переваги порівняно з програмним забезпеченням.

Система – це набір елементів або компонентів, які взаємодіють один з одним для досягнення певної мети [3]. Насправді, більшість систем не мають єдиної мети – вони складаються з підсистем з підцілями, які індивідуально вносять вклад в загальну мету системи. Завдання системи – приймати вхідні дані, маніпулювати або перетворювати такі вхідні дані у вихідні. Очікується, що кожна система буде мати форму введення, механізм обробки, форму виведення, зворотного зв'язку і механізмів управління. Тобто система обробляє введення (дані) для створення виведення (інформації); в кожній системі необхідний наявний механізм зворотного зв'язку для підтримки та моніторингу продуктивності; також повинен бути наявним механізм контролю для виправлення проблем і забезпечення виконання системою її призначення. Елементи системи та відношення, що існують між ними, визначають, як система працює [4]. Для інформаційної системи відсутнє єдине визначення, що є однією з основних перешкод в області інформаційних систем.

Одним з наслідків глобалізації в діловому світі стала можливість отримання більшої кількості інформації за набагато менший час, що означає, що компанії повинні витратити більшу частину свого часу та енергії на обробку збільшеного об'єму інформації.

Інформаційна система для підприємства являє собою групу взаємопов'язаних компонентів, які працюють разом для виконання дій введення, обробки, виведення, зберігання та контролю інформації з метою перетворення даних в матеріальну інформацію, яка може використовуватися при прогнозуванні, координації прийняття рішень і діяльності всередині організації [5]. На сьогодні переважна більшість підприємств використовує саме веб-базовані інформаційні системи.

Веб-базована інформаційна система – це інформаційна система, яка використовує веб-технології Інтернету для надання інформації та послуг користувачам чи іншим інформаційним системам або програмам [6].

На сьогодні існує два принципових підходи до створення веб-застосунків: традиційні веб-застосунки, більша частина логіки яких виконується на сервері, та односторінкові застосунки, логіка інтерфейсу користувача яких виконується переважно у веб-браузері, а взаємодія з веб-сервером здійснюється головним чином через веб-API [7]. Односторінковий застосунок використовує єдиний HTML-документ як оболонку для всіх веб-сторінок і організує взаємодію з користувачем через HTML, CSS, JavaScript, що динамічно підвантажуються, зазвичай за допомогою AJAX.

Також можливий гібридний підхід, при якому в найпростішому випадку в рамках великого традиційного веб-застосунка розміщуються один або кілька повнофункціональних підлеглих застосунків, побудованих на основі односторінкової моделі.

Повний та точний опис веб-застосунків є критичним фактором при проектуванні системи, оскільки він забезпечує точну оцінку якості обслуговування, необхідного для мережевої інфраструктури, ефективний розподіл ресурсів та ефективну оцінку продуктивності системи в цілому. На даний час існують текстові описи моделей веб-базованих ІС та ефективності їх експлуатації, але публікацій математичних моделей веб-базованих ІС дуже мало; вони практично відсутні у зарубіжному друці, як і конкретний склад формальної моделі, яка описувала би веб-базовані ІС у ході експлуатації.

Один з найбільш загальних описів архітектури наведено на рисунку 1.1 [8]. На наведеному рисунку елементи, що відділені пунктирною лінією, розміщені не на веб-серверах системи.

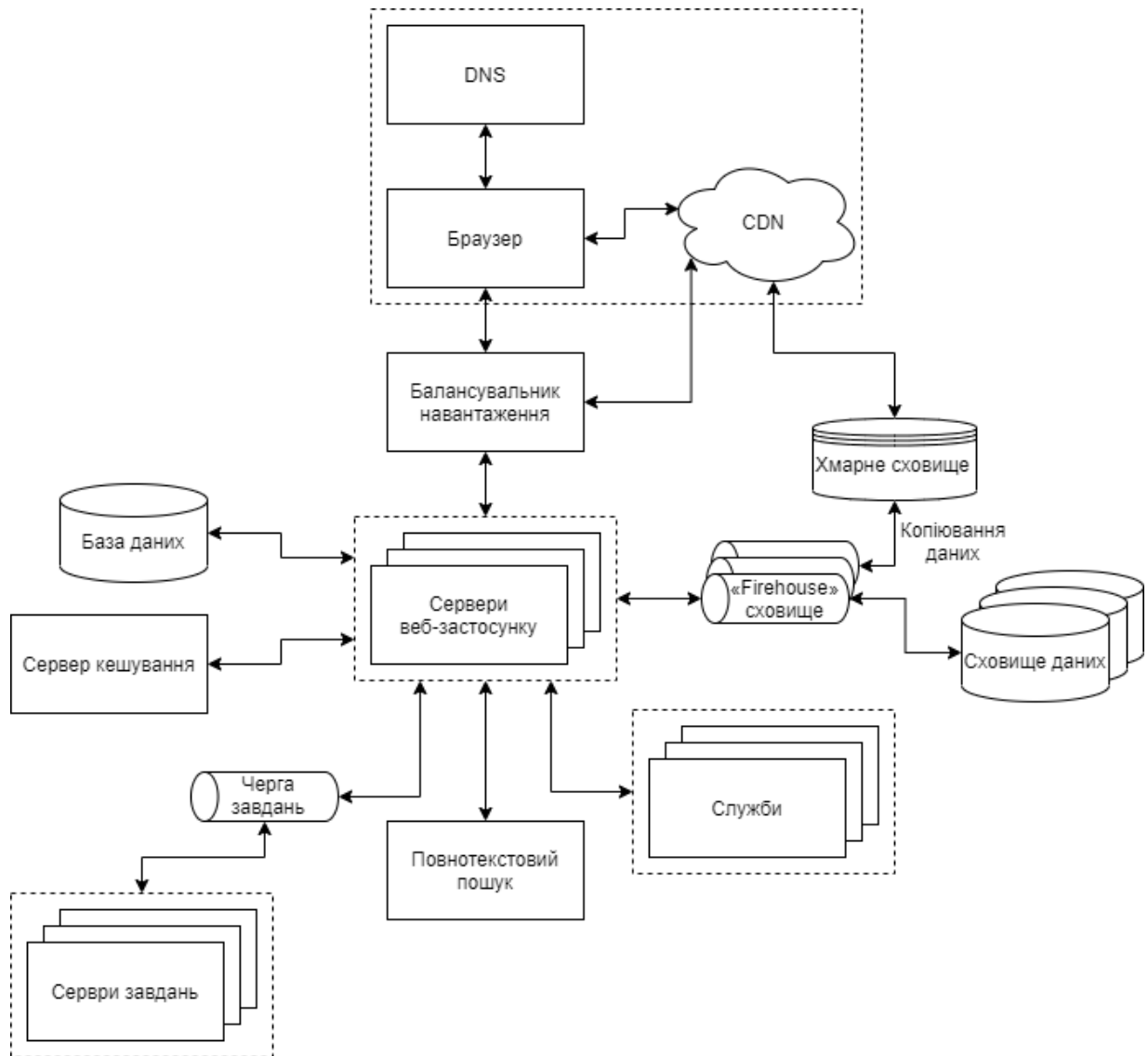


Рисунок 1.1 – Схема загальної архітектури веб-застосунку

DNS, або система доменних імен, – це базова технологія, яка уможливорює роботу Інтернету. На базовому рівні DNS забезпечує пошук пари з доменного імені та IP-адреси, що дозволяє комп'ютеру надіслати запит на відповідний сервер.

Балансувальники навантаження забезпечують горизонтальне масштабування. Вони надсилають вхідні запити на один із безлічі серверів програми, які зазвичай є дзеркальними копіями один одного, і надсилають відповідь назад користувачеві.

Сервери веб-застосунків виконують основну бізнес-логіку, яка обробляє запит користувача і відправляє HTML назад браузеру. У більшості випадків сервери веб-застосунків безпосередньо обмінюються інформацією з серверами завдань. Крім того, кожна серверна служба може мати відповідну базу даних, ізольовану від решти застосунку.

Служба кешування надає просте сховище даних у форматі ключ-значення, яке дозволяє зберігати та шукати інформацію за час, близький до лінійного. Зазвичай програми використовують функції кешування, щоб зберігати результати дорогих обчислень і скористатися ними пізніше з кешу, а не перераховувати їх ще раз.

Виконувати асинхронну роботу дозволяють різні архітектури, але найпоширенішою є архітектура «черга завдань». Вона складається з двох компонентів: черги «завдань», які необхідно виконати, та одного або кількох робочих серверів (часто званих «працівниками»), які обробляють завдання із черги. У чергах завдань зберігаються списки завдань, які необхідно виконати асинхронно.

Більшість веб-застосунків підтримують функцію пошуку за текстовим введенням (часто званим «запитом»), в якому застосунок повертає результати, що найбільш відповідають запиту. Технологія, яка використовує цю функцію, зазвичай називається «повнотекстовим пошуком» і використовує інвертований індекс для швидкого пошуку документів, які містять ключові слова запиту.

CDN, або система доставки контенту, дозволяє набагато швидше, ніж з вихідного сервера, відправляти статичні файли та зображення HTML, CSS та JavaScript. Вона розповсюджує контент із багатьох кінцевих серверів по

всьому світу, щоб користувачі завантажували різні ресурси з них замість вихідного сервера.

Веб-застосунки зазвичай працюють на робочих станціях. У користувачів є власна робоча станція; серверні процеси виконуються на виділених серверах. На продуктивність застосунків також впливає продуктивність окремого серверного комп'ютера.

Існують наступні архітектури веб-застосунків: однорівнева (монолітна), багаторівнева (два та три рівня) та шаблон модель-представлення-контролер (Model-View-Controller, MVC).

Для однорівневої архітектури характерні наступні властивості:

- усі три шари знаходяться на одній машині (весь код та обробка зберігаються на одній машині);
- шари представлення, логіки та даних тісно пов'язані між собою.

Внаслідок залежності наведених шарів один від одного виникають наступні проблеми:

- масштабованість: наявність одного процесору означає, що важко збільшити обсяг обробки даних;
- переносимість: перехід на нову машину може означати переписування всього коду програми;
- технічне обслуговування: зміна одного шару вимагає зміни інших шарів.

Таким чином, монолітний застосунок повністю замкнений у контексті поведінки. Під час роботи він може взаємодіяти з іншими службами або сховищами даних, однак основа його поведінки реалізується у власному процесі, а весь застосунок зазвичай розгортається як один елемент. Для горизонтального масштабування така програма зазвичай повністю дублюється на кількох серверах або віртуальних машинах. Схему однорівневої архітектури наведено на рисунку 1.2 [9].

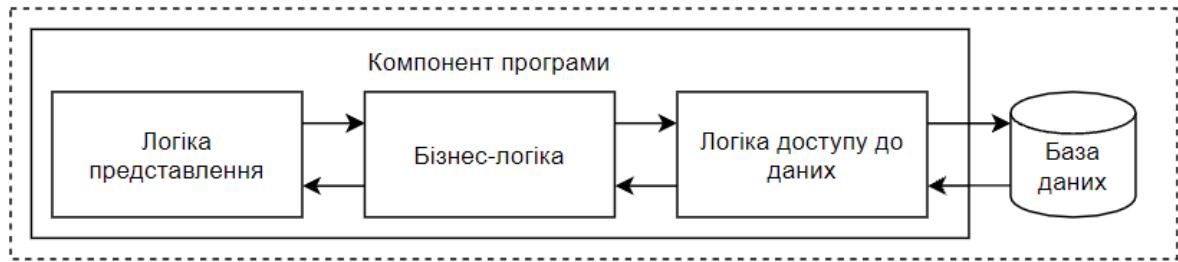


Рисунок 1.2 – Схема однорівневої архітектури веб-застосунку

Багаторівневі архітектури мають однакові компоненти: представлення, бізнес-логіка та дані. Для багаторівневих архітектур існує поділ компонентів на різні рівні або шари – рівень фізичного поділу та шар логічного поділу.

Рівень представлення – це статичний або динамічно створюваний вміст, що відображається браузером; він також:

- надає інтерфейс користувача;
- відповідає за взаємодію з користувачем;
- іноді називається графічним інтерфейсом користувача, представленням клієнта або «front end»;
- не повинен містити бізнес-логіку або код доступу до даних.

Рівень бізнес-логіки містить сервер додатків на рівні динамічної обробки та генерації вмісту, наприклад, Java EE, ASP.NET, PHP. Даний рівень:

- це набір правил обробки інформації;
- може передбачати одночасну роботу багатьох користувачів;
- іноді називається проміжним програмним забезпеченням або «back end»;
- не повинен містити код представлення або доступу до даних.

Рівень даних складається з бази даних, що містить як набори даних, так і СУБД або програмне забезпечення у вигляді реляційної СУБД, яке керує та забезпечує доступ до даних. Рівень даних:

- це фізичний рівень зберігання даних;
- керує доступом до БД або файлової системи;

- іноді називається «back end»;
- не повинен містити код представлення або бізнес-логіки.

У застосунках з багатошаровою архітектурою можуть встановлюватися обмеження на взаємодію між шарами. Така архітектура допомагає реалізувати інкапсуляцію. При зміні або заміні шару будуть змінені лише ті шари, які працюють безпосередньо з ним. Обмежуючи залежності шарів один від одного, можна зменшити наслідки внесення змін, внаслідок чого одинична зміна не впливатиме на всю програму.

Дворівнева архітектура моделі веб-базованого застосунку передбачає, що база даних працює на сервері, тобто вона є відділеною від клієнта. Таким чином, користувачу легко переключитися на іншу базу даних.

Однак шари представлення та логіки все ще тісно пов'язані, через що виникають наступні проблеми:

- велике навантаження на сервер;
- можливі перевантаження в мережі;
- представлення все ще прив'язане до бізнес-логіки.

Схему дворівневої архітектури наведено на рисунку 1.3 [9].

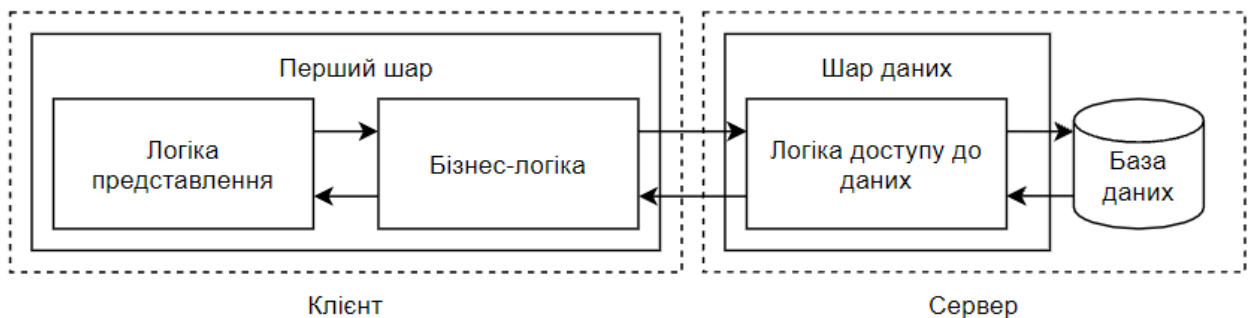


Рисунок 1.3 – Схема дворівневої архітектури веб-застосунку

Трирівнева архітектура передбачає незалежність шарів представлення, логіки та даних один від одного. Усі шари потенційно можуть працювати на різних машинах.

Для типової трирівневої архітектури характерні наступні принципи:

- кожен рівень (представлення, логіка, дані) повинен бути незалежним і не повинен викривати залежності, пов'язані з впровадженням;
- непов'язані рівні не повинні взаємодіяти;
- зміна платформи має впливає лише на шар, що працює на цій конкретній платформі.

Таким чином, для трирівневої архітектури можна виділити такі переваги, як незалежність шарів, через що система є простішою в обслуговуванні, компоненти можна використовувати багаторазово, а також додатку гарантується швидший розвиток (розподіл роботи). Схему трирівневої архітектури наведено на рисунку 1.4 [9].

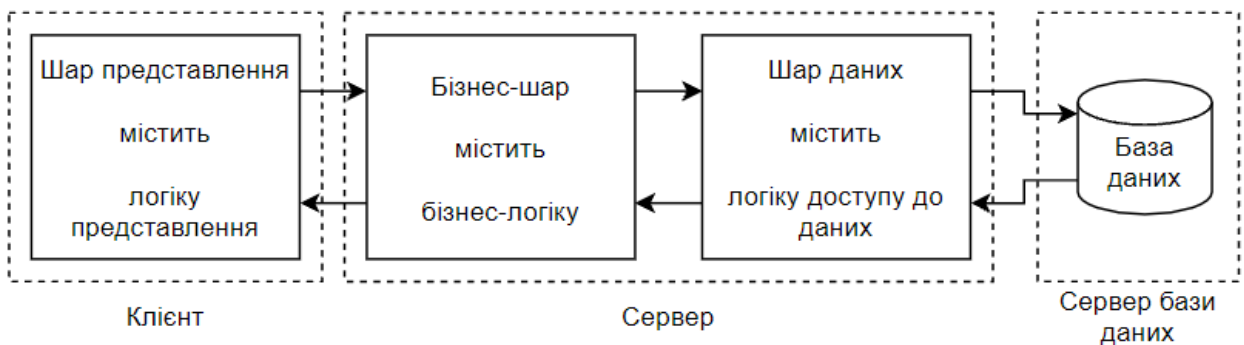


Рисунок 1.4 – Схема трирівневої архітектури веб-застосунку

Шаблон модель-представлення-контролер – це шаблон проектування графічних систем, що сприяє розмежуванню моделі та представлення.

За допомогою цього шаблону логіка, необхідна для обслуговування даних (БД, текстовий файл), відокремлюється від способу перегляду даних та способу взаємодії з даними (графічний інтерфейс, командний рядок).

Модель:

- керує поведінкою та даними домену програми;
- відповідає на запити про інформацію про її стан (зазвичай надходять від представлення);

– слідує інструкціям щодо зміни стану (зазвичай надходять від контролера).

Представлення надає моделі можливість взаємодії, зазвичай з використанням інтерфейсу користувача (для однієї моделі може існувати декілька представлень для різних цілей).

Контролер:

- отримує вхідні дані користувача та ініціює відповідь, здійснюючи виклики об'єктів моделі;
- приймає вхідні дані від користувача та на основі цього вводу вказує моделі та області перегляду необхідні для виконання дії.

На практиці модель містить спеціальні доменні значення, записує стан програми (наприклад, які товари є у кошику); модель часто пов'язана з БД, не залежить від представлення. Одна модель може посилатися на різні представлення. Представлення надає дані користувачеві, дозволяє взаємодію з користувачем, а також не виконує обробку. Контролер визначає, як інтерфейс користувача реагує на введення інформації користувачем (події), отримує повідомлення від представлення (звідки надходять події) та надсилає повідомлення моделі (повідомляє, які дані відображати) [10].

Нижче наведено склад даного шаблону для веб-додатків.

Модель:

- таблиці БД (постійні дані);
- інформація про сеанс (поточні дані про стан системи);
- правила, що регулюють транзакції.

Представлення:

- (X)HTML;
- таблиці стилів CSS;
- шаблони на стороні сервера.

Контролер:

- скрипти на стороні клієнта;
- обробка запитів http;

- бізнес-логіка та попередня обробка.

Таким чином, для шаблону модель-представлення-контролер можна виділити такі переваги, як чіткість дизайну (методи моделі надають інтерфейс прикладного програмування для даних та стану), ефективна модульність (будь-які компоненти можна легко замінити), наявність декількох представлень, легша розробка та обслуговування.

Можна виділити наступні відмінності між трирівневим видом архітектури та шаблоном модель-представлення-контролер:

- у трирівневій архітектурі відсутній безпосередній зв'язок між рівнями представлення та логіки (лінійна топологія), тоді як архітектурі модель-представлення-контролер наявний зв'язок між усіма рівнями (трикутна топологія);

- трирівнева архітектура переважно використовується у веб-програмах, де усі рівні працюють на фізично окремих платформах, тоді як шаблон модель-представлення-контролер здебільшого використовується в додатках, які працюють на одній робочій станції.

1.2 Дослідження проблеми оцінювання продуктивності веб-базованих інформаційних систем

Суть моделей сучасних ВБІС полягає у наступному: ці моделі описують ВБІС та її елементи як сукупність чорних ящиків, які взаємодіють один з одним через обмежену кількість заздалегідь відомих інтерфейсів [11]. Якщо кожен елемент моделі є чорним ящиком, вплив усього, що відбувається в цьому елементі, на інші елементи не справляється – вплив справляється тільки на результат. Існуючий вплив визначається параметрами діючого інтерфейсу. У інтерфейсі, який використовується для передачі результатів роботи цього чорного ящика, режим роботи інтерфейсу буде визначальним.

Побудова архітектури веб-додатків у вигляді сукупності чорних ящиків є суттєвим недоліком при оцінюванні продуктивності веб-базованої ІС, оскільки нині наявні моделі оцінювання ефективності експлуатації пропонують оцінювання ефективності лише окремих компонентів, що призводить до неможливості оцінювання ефективності експлуатації системи в цілому.

Для будь-якої ІС необхідним є проведення оцінки ефективності експлуатації системи з метою вивчення її проблем та недоліків. Даний процес допоможе в подальшому, базуючись на результатах, підвищити ефективність експлуатації, а також уникнути потенційного зниження задоволеності з боку користувача.

Проблема полягає у тому, що у діючих бібліотеках кількісних і якісних показників, що дозволяють оцінити ефективність експлуатації системи, усі підсумкові показники зводяться до задоволення вимог користувача. Ефективність експлуатації зводиться до ступеня задоволеності користувача; такий показник не може гарантовано відображати дійсний стан речей, тобто бути гарантовано об'єктивним.

Також на даний момент існують різні моделі оцінювання ефективності експлуатації, проте вони пропонують оцінювання ефективності лише окремих компонентів, що призводить до неможливості оцінювання ефективності експлуатації системи в цілому.

ISO 20000 – це міжнародний процесно-орієнтований стандарт управління ІТ-послугами. Мета його розробки полягала у створенні універсальних критеріїв, за допомогою яких будь-яка фірма чи служба, що надає ІТ-послуги, зможе оцінювати їх ефективність та виконання вимог замовників з урахуванням їхнього бізнесу [12]. У першій частині ISO 20000 визначено загальний підхід до планування і реалізації управління сервісами, в основу якого покладено цикл PDCA. Даний цикл складається з наступних етапів:

- Plan – ідентифікація та аналіз існуючої проблеми; оцінка можливостей і планування необхідних змін;
- Do – пошук вирішення проблеми і здійснення запланованих заходів;
- Check – оцінка результатів і висновки відповідно до поставлених задач;
- Act – прийняття рішення на основі отриманих висновків; якщо зміна не вирішує поставлену задачу, слід повторити цикл, внісши корективи в план.

Для кожного з етапів визначаються цілі та дії, які необхідно виконати. Основними достоїнствами ISO 20000 є можливість його практичного використання при проведенні сертифікації та аудиту організацій, а також його придатність при формуванні організаційного управління послугами. Основними недоліками даного стандарту можна вважати відсутність механізмів, які дозволяють привести існуючі сервіси у відповідність до вимог стандарту, і його описовий характер, а також відсутність формалізованих представлень, що ускладнює роботу зі стандартом. Таким чином, досягнення відповідності організації стандарту ISO 20000 забезпечує можливість надання послуг, виходячи з потреб бізнесу, а не тільки вирішення технічних задач.

Після впровадження системи управління IT-послугами стає можливою сертифікація підприємства за стандартом ISO 20000. Можливі такі варіанти сертифікації [12]:

- розширення сфери реєстрації;
- сертифікат ISO 20000.

У ході проведення узагальнення передового досвіду в галузі інформаційних технологій виникла потреба зумовити націленість узагальнюваних практик не лише на підвищення безперебійності роботи IT-підрозділів, а й на виконання вимог користувача та замовника IT-послуг. Таким чином було розроблено першу версію ITIL, яку згодом було удосконалено. ITIL – бібліотека інфраструктури інформаційних технологій, або ж набір детальних методів роботи з IT, таких як управління IT-послугами

та ІТ-активами, які зосереджені на узгодженні ІТ-послуг із потребами бізнесу. ІТІЛ описує процеси, процедури, завдання та контрольні списки, які не належать ні до організації, ні до технології, але можуть застосовуватися організацією для досягнення стратегії, надання цінності та підтримки мінімального рівня компетентності. Це дозволяє організації встановити базовий рівень, на основі якого вона може планувати, впроваджувати та проводити вимірювання. ІТІЛ використовується для демонстрації відповідності та вимірювання покращень. В організації немає формальної незалежної оцінки відповідності вимогам ІТІЛ.

Друга версія ІТІЛ складається із наступних семи основних книг [13]:

- підтримка послуг (Service Support);
- надання послуг (Service Delivery);
- планування запровадження управління послугами (Planning to Implement Service Management);
- управління застосунками (Application Management);
- управління інфраструктурою інформаційно-комунікаційних технологій (ICT Infrastructure Management);
- управління безпекою (Security Management);
- бізнес-перспектива (The Business Perspective).

Також до другої версії ІТІЛ включають додаткову книгу «Управління конфігураціями ПЗ» (Software Asset Management).

Третя версія ІТІЛ складається з наступних книг [13]:

- стратегія обслуговування (ITIL Service Strategy);
- проектування послуг (ITIL Service Design);
- перетворення послуг (ITIL Service Transition);
- надання послуг (ITIL Service Operation);
- безперервне вдосконалення послуг (ITIL Continual Service Improvement).

Нові книги акцентують увагу на тому, що одиночні процеси не забезпечують достатнього ефекту для бізнес-процесів підприємства. Найбільший ефект ці процеси забезпечують у разі, якщо вони повністю охоплюють бізнес-стратегію та безпосередньо взаємопов'язані з результатами діяльності підприємства.

Для ITIL SLA – це угоди про рівень сервісу, що визначають, яку якість обслуговування готовий надати IT-підрозділ (IT-підприємство) клієнтам або користувачам. Насправді, SLA є набором планових показників, значення яких прямо чи опосередковано визначають цільові значення параметрів IT-сервісу.

OLA розглядаються як угоди про рівні операційної підтримки IT-сервісу. Дані угоди деталізують глобальне SLA, декомпозуючи окремі глобальні показники до сукупності локальних техніко-економічних, технічних і технологічних показників та визначаючи їх цільові значення. У випадку, якщо IT-підрозділ (IT-підприємство) взаємодіє із зовнішніми постачальниками, подібні угоди позначаються як UC. CSF розглядаються як критерії, яким має задовольняти надання IT-сервісу, щоб виконувалося SLA.

KPI розглядаються як ключові показники ефективності, які кількісно чи якісно характеризують ступінь забезпеченості CSF конкретним IT-сервісом. Вимірювання KPI дозволяє надати власнику вимоги докладний і конкретний звіт про те, наскільки ефективно IT-підрозділ (IT-підприємство) забезпечує обумовлений рівень сервісу.

У ITIL також описуються метрики якості сервісів, значення яких у даному випадку є значеннями окремих показників продуктивності ВБІС. Однак оцінка якості за критеріями в даному випадку носить суб'єктивний характер, що негативно впливає на якість прийнятих управлінських рішень; дані показники зводяться до задоволення вимог користувача.

Прикладами таких метрик у третій версії ITIL є:

- зменшення у відсотках недоступності сервісів та компонентів;
- доступність черги (швидкість реакції);

- кількість і співвідношення подій у порівнянні з кількістю інцидентів;
- кількість інцидентів на одного співробітника ІТ-служби;
- кількість і відсоток подій, які потребували людського втручання, і чи було воно виконано;
- кількість інцидентів, вирішених на одній лінії;
- загальна кількість інцидентів (як контрольний захід);
- середня тривалість рішення;
- середній час ескалації інциденту;
- середній час закриття вирішеного інциденту;
- середній час, витрачений на обробку кожного типу запиту на обслуговування;
- загальна кількість дзвінків (за часом доби і днями тижня);
- збільшення кількості постачальників, що виконують цілі в рамках контракту;
- відсоток звернень, вирішених в рамках SLA;
- кількість нових сервісів;
- результати атестації співробітників тощо.

Оцінювання ефективності експлуатації елементів ВБІС відбувається шляхом проведення постійного моніторингу. Під моніторингом ІС розуміється збір ключових показників продуктивності системи через періодичні інтервали часу [14]. На даний момент існують різні інформаційні технології для моніторингу; кожна з цих технологій охоплює свій окремий аспект інформаційної системи і пропонує свій набір показників, що робить неможливою об'єктивну оцінку ефективності експлуатації веб-базованої інформаційної системи в цілому.

Результатом моніторингу буде набір значень деяких метрик веб-застосунку – описаних вище чорних ящиків. Проблема полягає у тому, що не всі елементи ВБІС можуть безпосередньо впливати на її продуктивність, або ж впливають лише у сукупності один з одним. У такому випадку моніторинг

цих елементів призводить до надмірних витрат при проведенні моніторингу ВБІС в цілому.

Отже, для уникнення подібної ситуації пропонується введення моделей ВБІС, належність досліджуваних веб-базованих ІС до яких можна уявити як гіпотезу про чорні ящики, вплив яких один на одного опосередковано через інтерфейси, що з'єднують ці ящики.

Таким чином, підтвердження або доказове спростування гіпотези передбачає виявлення набору показників, що суттєво впливають на продуктивність ВБІС. Одним із способів, що дозволяють висувати гіпотези про структуру та зміст моделі ВБІС та підтверджувати або спростовувати їх, є моделі та методи регресійного аналізу. Основна схема вирішення даної проблеми полягає у виключенні з подальшого розгляду тих показників продуктивності елементів веб-базованої ІС, значення яких лінійно залежні або корелюють зі значеннями показників продуктивності інших елементів ІС.

При вивченні залежностей між показниками використовуються наступні види регресійного аналізу:

- покрокова регресія (дозволяє встановити, яка підмножина показників дійсно впливає на результуючу функцію) – зчитуються усі показники; є ті, які суттєво впливають, та ті, які майже не впливають і заміряти ці показники немає сенсу;

- гребенева регресія (дозволяє виявити вид залежності і перевіряє результати покрокової регресії) – вивід підмножини показників, що реально впливають на результат оцінювання ефективності експлуатації ІС, з урахуванням не тільки лінійного, але й нелінійного зв'язку між ними;

- метод групового урахування аргументів (дозволяє групувати окремі показники за окремими елементами веб-базованої ІС у групи та враховувати їх сукупний вплив на систему).

Отже, задля оптимізації витрат на проведення моніторингу ВБІС пропонується провести перевірку гіпотези про вид моделі веб-базованої ІС на

основі декількох наборів вибірок даних за результатами моніторингу продуктивності ВБІС.

1.3 Висновки та постановка задачі дослідження

Існує припущення, або ж гіпотеза про структурний та змістовний вигляд сучасної моделі веб-базованої ІС, її інформаційне та програмне забезпечення. Дана гіпотеза про вигляд моделі ВБІС подається як гіпотеза про чорні ящики, вплив яких один на одного опосередковано через інтерфейси, що з'єднують ці ящики. Кількість інтерфейсів є кінцевою та можна заздалегідь визначити їх вплив один на одного, при цьому гіпотеза потребує перевірки.

Якщо виявиться, що за результатами аналізу набори показників, що характеризують роботу окремих елементів програмного забезпечення, програмних модулів, мережі та роботу бази даних не залежать один від одного ані лінійно, ані нелінійно, то можна стверджувати, що ця гіпотеза, в цілому, підтверджується.

Якщо буде виявлено вплив, тоді необхідно визначити вид впливу. Якщо залежність є нелінійною, її можна списати на роботу інтерфейсів; якщо вона є лінійною, то або спостережувана веб-базована ІС не відповідає прийнятій моделі, або гіпотеза про вид моделі є неправильною.

Головною метою науково-дослідної роботи є побудова моделі, яка дасть можливість кількісно оцінити ефективність експлуатації веб-базованої інформаційної системи, ґрунтуючись на відомому наборі показників, значення яких вимірюються в ході моніторингу експлуатації системи. Існуючі кількісні показники необхідно привести до спільного знаменника.

Для того, щоб модель не була занадто складною, із всієї множини показників необхідно обрати лише ті, які є найбільш значущими для

формування підсумкової оцінки. Відбір здійснюється з використанням регресійного аналізу.

Оскільки веб-базована система є досить складною і розглядається як множина закритих чорних або сірих ящиків, виникає необхідність згрупувати наявні показники для опису окремих елементів системи, тобто використовувати метод групового врахування аргументів. У даному випадку покрокова регресія є найпростішим інструментом, що дозволить зробити попередній висновок про значущість тих чи інших показників у цілому.

Об'єктом дослідження є веб-базована ІС у ході промислової експлуатації на підприємстві замовника. Розглядаються лише існуючі готові системи, передані розробником замовнику і введені в експлуатацію.

Предметом дослідження є моделі та методи, що дозволяють підтвердити або спростувати гіпотезу про вигляд моделі експлуатованої веб-базованої ІС.

Дослідницька робота зводиться до перевірки гіпотези про вид моделі веб-базованої ІС на основі декількох наборів вибірок даних за результатами моніторингу інформаційного та програмного забезпечення веб-базованої ІС.

2 ОПИС ТЕОРЕТИЧНОГО ВИРШЕННЯ ЗАДАЧІ

2.1 Опис основного способу моніторингу експлуатації веб-базованої інформаційної системи

Зростання та широке поширення великих розподілених клієнт-серверних додатків (ERP-систем, систем автоматизації бізнесу тощо) породжує численні спеціальні завдання щодо їх розробки. Одне з таких завдань – реалізація вимог щодо продуктивності. Продуктивність є однією з ключових бізнес характеристик великих клієнт-серверних систем, а їх невідповідність таким вимогам може спричинити великі фінансові та фізичні втрати при експлуатації і чи не повну повторну реалізацію.

Складність розробки великих клієнт-серверних додатків, що задовольняють заданим вимогам з продуктивності, полягає в необхідності створення складних архітектурних рішень, заснованих на паралельній обробці запитів, асинхронній взаємодії компонентів, алгоритмах кешування даних, балансування навантаження тощо. При цьому необхідно точно відобразити вимоги щодо продуктивності на архітектуру, щоб вона могла забезпечити потрібну продуктивність і не була надмірною, оскільки подібна надмірність збільшує витрати на розробку системи. При цьому повинні надаватися можливості збільшення продуктивності системи за рахунок додавання нових апаратних модулів.

Продуктивність, яку відчуває користувач – це єдина продуктивність, яка має значення. Користувачі взаємодіють із системою за допомогою введення вхідних даних, при цьому у відповідь вони отримують реакцію системи. Продуктивність – це якість того, як система реагує на дії користувача.

Продуктивність веб-базованої інформаційної системи є найважливішим чинником задоволення користувачів при її використанні. Відомо, що продуктивність інформаційної системи найповніше характеризується часом її відгуку.

Головна мета веб-додатків – обробка сотень і навіть тисяч запитів за секунду. Щоб забезпечити успіх таким програмам, дуже важливо визначити потенційні вузькі місця та зробити все можливе, щоб уникнути проблем у ході експлуатації.

Веб-ефективність – це об’єктивне вимірювання та сприйняття досвіду користувача під час використання веб-сайту чи веб-програми. Сюди входять такі основні напрямки:

- зменшення загального часу завантаження: скільки часу потрібно, щоб файли, необхідні для відтворення веб-сайту, завантажувалися на комп’ютер користувача? На це, як правило, впливає затримка, розмір файлів, кількість файлів та інші фактори. Загальна стратегія полягає в тому, щоб зробити файли якомога меншими, зменшити кількість запитів HTTP, наскільки це можливо, і використовувати розумні методи завантаження (наприклад, попереднє завантаження), щоб зробити файли доступними швидше;

- якнайшвидше зробити сайт придатним для використання: це в основному означає завантаження ресурсів веб-сайту в розумному порядку, щоб користувач міг дійсно швидко почати його використовувати. Будь-які інші ресурси можуть продовжувати завантажуватися у фоновому режимі, поки користувач виконує основні завдання, та іноді ресурси завантажуються лише тоді, коли вони дійсно потрібні (це називається відкладеним завантаженням). Вимірювання того, скільки часу потрібно сайту, щоб почати роботу після його завантаження, називається часом до інтерактивності;

- плавність та інтерактивність: чи є програма надійною та приємною у використанні? Чи плавна прокрутка? Чи кнопки реагують належним чином на натискання? Чи швидко відкриваються спливаючі вікна, і чи плавно вони анімуються при цьому? Існує багато найкращих методів, які слід розглянути, щоб зробити додатки плавними, наприклад, використовувати для анімації CSS, а не JavaScript, і мінімізувати кількість перефарбовування інтерфейсу, що необхідні користувачу через зміни в DOM;

– сприйнята продуктивність: те, наскільки швидким веб-застосунок здається користувачеві, має більший вплив на користувацький досвід, ніж те, наскільки швидким є веб-застосунок насправді. Те, як користувач сприймає ефективність, настільки ж важливе або, можливо, важливіше за будь-яку об'єктивну статистику, але це суб'єктивно і не так легко виміряти. Сприйнята продуктивність – це показник лише з точки зору користувача. Навіть якщо операція триватиме багато часу (через затримку чи що завгодно), можна задіяти користувача, поки він очікує, показуючи індикатор завантаження або серію корисних підказок;

– вимірювання продуктивності. Веб-продуктивність включає вимірювання фактичної та передбачуваної швидкості програми, оптимізацію, де це можливо, а потім моніторинг продуктивності. Це включає в себе ряд показників (вимірюваних показників, які можуть вказувати на успіх або невдачу) та інструментів для вимірювання цих показників.

Підсумовуючи, багато функцій впливають на продуктивність, включаючи затримку, розмір програми, кількість вузлів DOM, кількість зроблених запитів на ресурси, продуктивність JavaScript, навантаження на процесор тощо.

Продуктивність веб-базованих ІС може зменшуватись у наступних випадках:

- виникнення великого об'єму даних, що обробляються;
- програма стає більшою (сотні користувальницьких бізнес-сценаріїв, десятки екранів, форм і так далі);
- велика кількість клієнтів з різних регіонів;
- висока конкуренція на ринку (користувач віддасть перевагу додатку конкурента, якщо він працює швидше);
- бажання збільшити кількість користувачів або прибуток;
- очікувані та заплановані періоди пікового навантаження;

- підвищення продуктивності на основі бізнес-плану або викликане реальною чи очікуваною (прогнозованою) проблемою продуктивності з метою забезпечення доступності і ефективності при підвищеному навантаженні;
- необхідність мобільна версія (досі браузері на мобільних пристроях страждають від проблем із продуктивністю).

Сучасні веб-базовані ІС будуються на основі мікросервісної, або багаторівневої архітектури – це стиль, який структурує додаток як набір сервісів. Кожен такий сервіс є максимально автономним, є необхідний для виконання конкретної задачі та підтримується конкретною командою. Така архітектура дозволяє застосовувати модульний принцип побудови застоснків з урахуванням потреб клієнтів. На даний момент основним способом моніторингу експлуатації ВБІС є використання пасивного експерименту.

Пасивний експеримент – це метод, який передбачає проведення великої серії дослідів із почерговим варіюванням кожної із змінних – значень показників продуктивності ВБІС у ході експлуатації та реалізації різних бізнес-сценаріїв [15].

Завдання під час планування пасивного експерименту:

- вибір кількості та частоти вимірювань;
- вибір методу обробки результатів вимірювань.

Найчастіше метою пасивного експерименту є побудова математичної моделі об'єкта, яку можна розглядати або як добре, або як погано організований об'єкт.

Пасивному експерименту властиві наступні особливості:

- пасивний експеримент означає збір даних у режимі нормальної роботи ВБІС;
- пасивний експеримент полягає у спостереженні та реєстрації значень показників продуктивності без активного втручання дослідника в хід технологічного процесу, а також без внесення до нього навмисного впливу;

– для обробки зібраних експериментальних даних після закінчення експерименту застосовуються математико-статистичні методи;

– пасивні експерименти характеризуються відсутністю впливу дослідника на зміну значень показників продуктивності ВБІС, що зчитуються в ході експлуатації; у цьому випадку дослідник виступає в ролі спостерігача, не втручаючись у хід експерименту.

Таким чином, пасивний експеримент полягає у отриманні статистичної інформації про ВБІС за даними її нормальної експлуатації; це відбувається шляхом фіксації значень показників продуктивності та подальшим аналізом отриманих результатів. Обробка дослідних даних задля побудови математичної моделі проводиться методами класичного регресійного і кореляційного аналізу.

Моніторинг продуктивності ВБІС проводиться на найнижчому рівні абстракції, тобто проводиться спостереження показників лише окремих сервісів, або компонентів. Модель такого спостереження наведено на рисунку 2.1 [16].

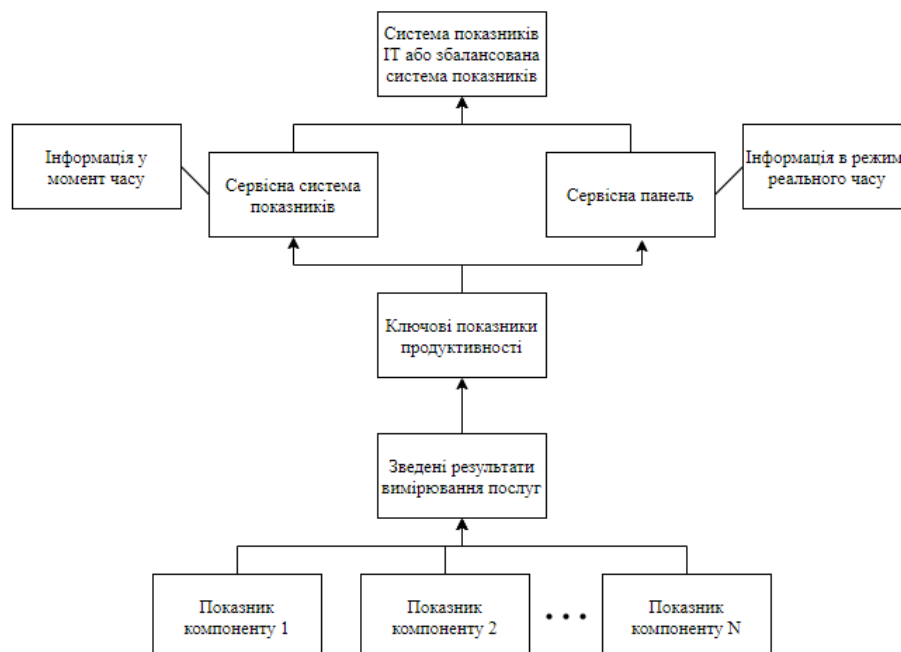


Рисунок 2.1 – Схема моделі зчитування показників продуктивності сервісів

Із зниженням рівня абстракції кількість показників суттєво збільшується, тому при моніторингу усіх можливих компонентів мережа може бути перенавантажена, тому раціональним буде враховувати лише ті показники, що суттєво впливають на продуктивність веб-базованої ІС.

Отже, головним недоліком пасивного експерименту при проведенні моніторингу продуктивності ВБІС є те, що даний метод дозволяє проводити замір показників продуктивності лише окремих елементів – сервісів – веб-базованої ІС без урахування можливого впливу цих елементів один на одного.

Веб-базована інформаційна система складається із множини елементів, які у ході експлуатації можуть по-різному впливати один на одного. Даний вплив обумовлений реалізацією у системі різних бізнес-сценаріїв.

Бізнес-сценарій – це опис поведінки системи, коли вона взаємодіє з кимось (або чимось) із зовнішнього середовища, тобто послідовність роботи елементів системи в залежності від поставленої задачі [17]. Бізнес-сценарій передбачає задіяння системою певного набору елементів у певній послідовності, що з точки зору визначення продуктивності може означати, що загальна продуктивність виконання бізнес-сценарію може відрізнятись від суми продуктивностей окремих елементів веб-базованої ІС, що беруть участь у виконанні даного сценарію.

Тому при виконанні різних бізнес-сценаріїв користувачів одні й ті самі елементи веб-застосунку можуть по-різному впливати один на одного і, як наслідок, може змінюватись загальна продуктивність ВБІС.

Таким чином, загальну продуктивність ВБІС можна розглядати як багатовимірну модель: продуктивність бізнес-сценарію, продуктивність сценаріїв виконання окремих функцій та продуктивність окремих елементів веб-базованої ІС, приклад яких зазначено на рисунку 1.1. Саме продуктивність останніх можливо відстежувати за допомогою моніторингу, але, як було зазначено вище, користувачеві важливе виконання бізнес-сценарію цілком, а не окремих його елементів.

2.2 Дослідження багатовимірних моделей моніторингу продуктивності веб-базованої інформаційної системи

Враховуючи наявні способи опису експлуатованої ВБІС, можна подати її у вигляді вектору, елементами якого є лінійно співзалежні елементи:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}, \quad (2.1)$$

де i – ідентифікатор елемента ВБІС, $i = 1, \dots, n$;

x_i – множина показників, які вимірюються у ході управлінням i -го елемента.

Даний опис веб-базованої ІС можна вважати коректним лише тоді, коли її елементи лінійно залежать один від одного. Але, як було зазначено вище, при реалізації різних бізнес-сценаріїв вид впливу елементів один на одного може змінюватися, тому критичним фактором при проведенні моніторингу продуктивності ВБІС є визначення виду залежності та впливу показників.

Вид залежності можна визначити за допомогою методу регресійного аналізу; для цього спочатку необхідно зафіксувати значення показників у ході експлуатації веб-застосунку та, що є найбільш важливим, у ході виконання різних бізнес-сценаріїв. Таким чином визначається вид залежності та вид моделі експлуатованої ВБІС.

Моніторинг продуктивності веб-базованої ІС відбувається у трьох вимірах: рівень бізнес-сценарію, рівень сценаріїв виконання окремих функцій та рівень окремих елементів ВБІС. При цьому продуктивність також залежить від архітектури веб-застосунку.

Для обміну даними на рівні бізнес-сценарію використовується поняття бізнес-транзакції. Бізнес-транзакція – це узгоджена зміна стану відносин двох

і більше сторін, де кожна сторона готова до цієї зміни та знає, що її узгоджено приймуть усі сторони [17]. На даному рівні проводиться моніторинг продуктивності усього бізнес-сценарію в цілому, враховуючи вид залежності окремих функцій та елементів один на одного.

Так, до опису транзакції входять такі дані:

- сторони та виконавці транзакції;
- об'єкти, над якими проводиться транзакція, у тому числі форми та документи, які обробляє транзакція;
- призначення транзакції;
- обмеження, які регламентують будь-які аспекти транзакції, у тому числі часові обмеження (максимальний час, протягом якого транзакція може бути активна);
- обмеження, зовнішні стосовно транзакції та її учасників, які виражають вимоги закону, правила торгівлі та терміни контракту, публічні, прийняті політики, закони та регулювання, які застосовні для учасників транзакції;
- особливості зв'язку та взаємодії з іншими транзакціями.

Бізнес-транзакції можуть реалізовуватись як класичними засобами інтеграції інформаційних систем, так і технологіями, що забезпечують комунікацію роз'єднаних та слабопов'язаних систем (наприклад, за допомогою технологій веб-служб).

Транзакції бізнес-процесів – це дія або набір дій, відповідальних за виконання певної частини бізнес-логіки застосунку [17]. Такі транзакції можуть включати та координувати транзакції двох інших типів, розподілених за різними бізнес-середовищами. Кожна транзакція бізнес-процесу поділяється на бізнес-завдання, а завдання виконується в рамках окремого середовища. Кожне середовище може рекурсивно поділятися на «підсередовища», поділяючи таким чином і бізнес-завдання.

На рівні сценарію виконання окремих функцій для обміну даними використовуються функціональні транзакції, або просто транзакції. На цьому рівні виконується моніторинг продуктивності сукупності елементів, які беруть участь у виконанні конкретної функції бізнес-сценарію. Загальні характеристики системи можна представити у вигляді суми продуктивностей усіх окремих функцій. Для рівня окремих елементів веб-базованої ІС використовуються технічні транзакції.

Отже, бізнес-транзакцію можна описати задоволеністю користувача, а також конкретними показниками, у тому числі й продуктивністю. Якщо елементами транзакції виділяються функції сервісів, всі вони пов'язані відносинами передачі даних, які у свою чергу обмежені інтерфейсними контрактами.

Запишемо загальну продуктивність у вигляді суми продуктивностей функцій:

$$x_{perf} = (a_1 * x_{tr,a_1} + b_1) + \dots + (a_j * x_{tr,a_j} + b_j) = \sum_{i=1}^j a_i * x_{tr,a_i} + b_i, \quad (2.2)$$

де x_{tr} – продуктивність функції (функціональної транзакції), що бере участь у виконанні бізнес-сценарію;

b – продуктивність інтерфейсу, через який опосередковано вплив функцій одна на одну;

a – номер функції;

z – кількість функцій.

У ході проведення регресійного аналізу та виявлення виду залежностей між елементами та функціями ВБІС може виявитись, що залежність між ними не є лінійною.

У свою чергу, продуктивність функції визначається продуктивностями елементів, що беруть участь у транзакціях:

$$\begin{aligned}
 x_{tr,a} = & (a_1 * x_{t,a_1b_1} + b_1) + \dots + (a_k * x_{t,a_kb_k} + b_k) + \dots + (a_m * \\
 & * x_{t,a_mb_m} + b_m) + \dots + (a_k * x_{t,a_kb_k} + b_k) + \dots + (a_1 * x_{t,a_1b_1} + \\
 & + b_1) = (a_m * x_{t,a_mb_m} + b_m) + 2 * \sum_{k=1}^m a * x_{t,ab} + b, \quad (2.3)
 \end{aligned}$$

де $x_{t,ab}$ – продуктивність b -го елемента ВБІС, що бере участь у a -ій транзакції;
 y – кількість елементів;
 a – номер транзакції;
 b – продуктивність інтерфейсу, через який опосередковано вплив елементів один на одного.

2.3 Дослідження показників, що впливають на продуктивність веб-базованої інформаційної системи

Згідно із формулою (2.3), у визначенні продуктивності функціональної транзакції беруть участь окремі елементи, тобто сервіси, що виконуються у конкретній транзакції. Продуктивність транзакції обумовлюється продуктивністю елементів та інтерфейсів, через які вони взаємодіють, тому загальна формула для функціональних транзакцій лишатиметься однаковою.

Для різних архітектур, описаних у розділі 1, коефіцієнти рівняння будуть відрізнятися один від одного, адже у залежності від моделі веб-застосунку у транзакціях можуть брати участь різні компоненти. Продуктивність бізнес-сценаріїв для усіх видів архітектур буде сумою продуктивностей функцій, але із різними коефіцієнтами. Таким чином, у ході аналізу результатів моніторингу необхідно визначити, чи впливає модель веб-застосунку на продуктивність ВБІС, та якщо впливає, то як сильно.

Оскільки моніторинг передбачає проведення пасивного експерименту, до його проведення неможливо визначити, які показники продуктивності

можуть бути лінійно залежними один від одного, а які можуть впливати на продуктивність системи в цілому. При проведенні аналізу результатів моніторингу може виникнути ситуація, при якій виявиться, що продуктивність окремої функції є занадто малою у порівнянні із іншими функціями; у такому випадку слід звернути на цю функцію основну увагу.

Усі елементи рівняння 2.3 визначаються за результатами проведеного пасивного експерименту. Основне завдання пасивного експерименту – за результатами спостережень зробити деякі висновки щодо параметрів математичної моделі експерименту, при цьому передбачається, що вигляд моделі відомий, але параметри – невідомі [18]. При проведенні пасивного експерименту спочатку необхідно визначити, які саме показники вимірюватимуться; потім необхідно визначити матрицю пасивних експериментів, яка являє собою набір значень показників, що замірюються у ході експерименту, у конкретні моменти часу. Дана матриця описує результат проведення пасивного експерименту у вигляді спостережуваних значень показників продуктивності.

У експерименті, у якому проводиться n вимірювань значень показників x_i , де $i = 1, 2, \dots, k$ – кількість показників, значення показнику x_i у j -му вимірюванні ($j = 1, 2, \dots, n$) позначається як x_{ij} . Тоді набір точок x^j називатиметься планом експерименту, а планом експерименту буде матриця наступного вигляду:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{k1} \\ x_{21} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, \quad (2.4)$$

У ході проведення експерименту як на систему, так і на ІТ-інфраструктуру здійснюється різних зовнішній та внутрішній вплив; за результатами пасивного експерименту необхідно вивести рівняння регресії,

тобто отримати конкретні значення коефіцієнтів як для рівня сценарію виконання окремих функцій, так і для рівня окремих елементів веб-базованої ІС. За результатом проведення експерименту необхідно визначити, чи є наявними лінійно залежні показники та визначити, які з елементів рівнянь (2.2) та (2.3) є значущими для кожної конкретної бізнес-транзакції та функціональної транзакції.

Отже, результатом аналізу проведеного моніторингу мають бути відповіді на наступні питання:

- чи є у системі лінійно залежні показники продуктивності (з метою подальшого зменшення оброблюваної інформації і, як наслідок, навантаження на мережу та зменшення витрат на моніторинг);

- наскільки значущими для продуктивності системи в цілому є продуктивності окремих технічних та функціональних транзакцій.

2.4 Висновки до другого розділу

У даному розділі було описано основний спосіб моніторингу продуктивності ВБІС на підприємстві у вигляді проведення пасивного експерименту. У ході проведення експерименту зчитуються значення показників лише найнижчого рівня абстракції, тобто значення окремих сервісів (компонентів). Такий підхід призводить до перенавантаження мережи; також існує можливість, що не усі зчитувані показники є лінійно незалежними або значущими для продуктивності веб-базованої ІС. З метою зменшення витрат на проведення моніторингу доцільним буде визначити набір показників, що суттєво впливають на продуктивність ВБІС, та не є лінійно залежними один з одним. Зробити це можна за допомогою використання методів регресійного аналізу.

Було запропоновано формули для визначення продуктивності як на рівні бізнес-сценарію (загальну продуктивність), так і на функціональному рівні (продуктивність окремої функції). Елементи зазначених рівнянь визначаються за результатами проведеного пасивного експерименту, тому до проведення експерименту та аналізів його результатів неможливо зробити висновок, чи є змінні, що беруть участь у моніторингу, лінійно залежними або значущими. У ході проведення регресійного аналізу за результатами моніторингу необхідно дати відповідь на наступні питання: чи наявні в системі лінійно залежні показники продуктивності (їхнє виявлення та усунення дозволить зменшити витрати на обробку інформації) та чи є виявлені лінійно незалежні показники продуктивності як технічних, так і функціональних транзакцій значущими для продуктивності веб-базованої ІС.

3 ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЙНОГО АНАЛІЗУ РЕЗУЛЬТАТІВ МОНІТОРИНГУ ПРОДУКТИВНОСТІ ВЕБ- БАЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Дослідження методу покрокової регресії

Мета покрокової регресії полягає у відборі з великої кількості предикатів невеликої підгрупи змінних, які роблять найбільший внесок у варіацію залежної змінної.

Покрокова регресія – це метод підбору регресійних моделей, у якому вибір прогнозних змінних здійснюється автоматичною процедурою. На кожному кроці розглядається змінна для додавання або віднімання з набору пояснювальних змінних на основі деякого заздалегідь заданого критерію [19]. Зазвичай це має форму прямої, зворотної або комбінованої послідовності F-тестів. Покрокова регресія використовується у вирішенні завдань із попиту, прибутковості акцій, щодо функції витрат виробництва, в макроекономічних розрахунках та інших питаннях економіки.

Через витрати, пов'язані з отриманням інформації при великій кількості показників продуктивності та їх подальшої перевіркою, необхідно прагнути, щоб рівняння включало якнайменше незалежних змінних.

Вхідні дані для проведення процедури покрокової регресії мають зберігатися у числовому форматі. На вхід алгоритму подаються матриця спостережень X розмірності $m \times n$ та вектор-стовпець відгуків у розмірності m , де n – об'єм вибірки (кількість регресорів), а m – кількість спостережень (показників продуктивності). Кожен елемент повинен мати чіткий набір значень; рядки з пропущеними значеннями активних змінних ігноруються. Оцінювання проводиться за спостереженнями за вхідними даними (рядки матриці спостережень X) та вихідними даними (елементи вектора відгуків Y).

Існують такі припущення множинної регресії:

- між вихідною та незалежними змінними має існувати лінійна залежність;
- множинна регресія передбачає, що залишки розподілені нормально;
- відсутність мультиколінеарності – незалежні змінні не сильно корельовані одна з одною;
- гомоскедастичність – дисперсія помилок є однаковою для значень незалежних змінних.

F-тест, або критерій Фішера – статистичний критерій, тестова статистика якого при виконанні нульової гіпотези має розподіл Фішера (F-розподіл). Критерій дозволяє оцінити доцільність введення додаткової незалежної змінної в лінійну модель множинної регресії, рівняння якої має наступний вигляд:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon, \quad (3.1)$$

де y – залежна змінна;

$x_k, k = 1, 2, \dots, n$ – незалежні змінні;

$\beta_k, k = 1, 2, \dots, n$ – параметри моделі;

n – об'єм вибірки;

ε – випадкова складова.

Ідея цього критерію заснована на такому понятті, як сума квадратів регресії SSR :

$$SSR = \sum_{k=1}^n (\widehat{y}_k - \bar{y})^2, \quad (3.2)$$

де n – об'єм вибірки;

\widehat{y}_k – оцінка, отримана на основі регресійної моделі, $k = 1, 2, \dots, n$;

\bar{y} – середнє за всіма спостереженнями y .

Даний показник характеризує частку загальної варіації (мінливості) результативної ознаки y , яку вдалося пояснити з допомогою регресії. Кожну відстань $\widehat{y}_k - \bar{y}$ можна розкласти на два складники: $(\widehat{y}_k - \bar{y})$ – було пояснено регресією та $(y_k - \widehat{y}_k)$ – не вдалося пояснити. Тоді модель вважається тим кращою, чим ближче до нуля величини $y_k - \widehat{y}_k$.

Припустимо, що на основі змінних x_k , $k = 1, 2, \dots, n$ було побудовано регресійну модель, для якої частка мінливості, пояснена лінійною залежністю, становить величину $SSR_{initial}$. Припустимо, що ми хочемо ввести у модель нову ознаку x_{extra} ; сума квадратів регресії, побудованої на незалежних змінних x_k , $k = 1, 2, \dots, n$ та x_{extra} складає SSR_{full} . Тоді $SSR_{full} \geq SSR_{initial}$.

Розрахуємо, наскільки збільшилася пояснювальна здатність моделі внаслідок введення нової змінної x_{extra} :

$$SSR_{extra} = SSR_{full} - SSR_{initial}, \quad (3.3)$$

де SSR_{extra} – внесок змінної x_{extra} у пояснення загальної мінливості y .

Отже, чим більше дане значення, тим вагоміший внесок. Поріг для SSR_{extra} задля визначення цієї величини досить великою і, відповідно, задля ухвалення рішення про значущість ознаки x_{extra} , визначається за допомогою F-тесту. Даний критерій дозволяє перевірити гіпотезу H_0 : внесок SSR , що вноситься x_{extra} , є недостатньо великим, через що цю змінну не слід включати в модель, та її альтернативу H_1 : внесок SSR , що вноситься x_{extra} , є значущим, через що цю змінну слід включати в модель.

Для того, щоб перевірити ці гіпотези, слід перейти від показника SSR_{extra} до статистики наступного виду:

$$\gamma = \frac{SSR_{extra}}{MSE_{full}}, \quad (3.4)$$

де MSE_{full} – сума квадратів помилок SSE , що припадає на одну ступінь свободи df_{SSE} , тобто розрахункове значення F-критерію.

Значення MSE_{full} знаходиться за наступною формулою:

$$MSE_{full} = F_{real} = \frac{SSE}{df_{SSE}} = \frac{\sum_{k=1}^n (y_k - \hat{y}_k)^2}{n - m - 2}, \quad (3.5)$$

де y_k – істинне значення результуючої змінної;

\hat{y}_k – оцінка, отримана на основі регресійної моделі;

n – об'єм вибірки;

m – кількість змінних вихідної моделі (без x_{extra}).

Статистика, сформована відповідно до (3.4), за справедливості гіпотези H_0 розподілена за законом Фішера (F-розподіл). Тоді перевірка гіпотези H_0 буде зводитись до наступної послідовності дій:

– задається рівень значущості α , наприклад, 0,01 або 0,05. Ця величина характеризує ризик прийняття неправильного рішення.

– за спеціальними таблицями знаходиться 100α -процентну точку K_α розподілу Фішера із ступенями свободи $d1 = 1$ та $d2 = n - m - 2$. Це значення буде граничним для статистики (3.4).

– порівнюється знайдена 100α -процентну точку K_α зі значенням статистики γ . Якщо виявиться, що $\gamma > K_\alpha$, то робиться висновок про значущість ознаки x_{extra} і, відповідно, її слід включити в модель (надається перевага гіпотезі H_1 з ймовірністю α помилитися).

Якщо $\gamma < K_\alpha$, то приймається рішення про неефективність включення змінної x_{extra} у модель (тобто гіпотеза H_0 приймається з ймовірністю $1 - \alpha$ як така, що не суперечить експериментальним даним).

Основні підходи до покрокової регресії:

– прямий відбір, який включає початок без змінних в моделі, тестування додавання кожної змінної з використанням обраного критерію відповідності моделі, додавання змінної (якщо є), включення якої дає найбільш статистично

значуще поліпшення відповідності, і повторення цього процесу до тих пір, поки жоден з них не покращує модель у статистично значущій мірі;

- зворотне усунення, яке включає початок з усіма змінними-кандидатами, тестування видалення кожної змінної з використанням обраного критерію відповідності моделі, видалення змінної (якщо така є), втрата якої призводить до найбільш статистично незначного погіршення відповідності моделі, і повторення цього процесу до тих пір, поки не зникнуть змінні, що можуть бути видалені без статистично значущої втрати відповідності;

- двоспрямоване усунення, комбінація перерахованого вище, перевірка на кожному етапі включення або виключення змінних.

Метод прямого відбору передбачає виконання наступних етапів.

Етап 1. Зі списку всіх можливих вхідних змінних вибирається та, яка має найбільшу кореляцію з y , після чого модель, що містить лише одну вибрану незалежну змінну, перевіряється на значущість за допомогою F-критерію. Якщо значущість моделі не підтверджується, то алгоритм на цьому закінчується через відсутність істотних вхідних змінних. В іншому випадку ця змінна вводиться в модель і здійснюється перехід до наступного пункту алгоритму.

Слід зазначити, що в даному випадку перевірка на значущість усієї моделі в цілому буде рівносильна перевірці на значущість обраної незалежної змінної, тому що на даному етапі модель не містить інших вхідних змінних.

Етап 2. За всіма змінними на підставі формули (3.4) розраховується значення статистики γ , яка являє собою відношення приросту суми квадратів регресії, що досягається за рахунок введення в модель відповідної додаткової змінної (порівняно з величиною $SSR_{initial}$, розрахованої тільки на основі раніше вже введених змінних), до величини MSE_{full} .

Етап 3. З усіх змінних-претендентів на включення до моделі вибирається та, яка має найбільше значення критерію, розрахованого на етапі 2.

Етап 4. Проводиться перевірка на важливість обраної на етапі 3 незалежної змінної. Якщо її значущість підтверджується, вона вводиться у модель, і здійснюється перехід до етапу 2 (але з новою незалежною змінною у складі моделі). В іншому випадку алгоритм зупиняється.

Метод зворотного усунення схожий на попередній метод, але з тією відмінністю, що спочатку в модель включені усі змінні, і поступово здійснюється «відсіювання» тих, які не проходять перевірку на значущість. Даний метод передбачає виконання наступних етапів.

Етап 1. У модель включаються всі незалежні змінні.

Етап 2. За змінними, включеними в модель, розраховується величина, що є різницею між сумою квадратів регресії, побудованої за всіма поточними змінними моделі, і аналогічним показником, розрахованим без урахування однієї змінної, для якої обчислюється цей показник. За кожною знайденою величиною $SSR_{initial}$ розраховується статистика γ .

Етап 3. Вибирається змінна з мінімальним значенням γ .

Етап 4. Вирішується питання про доцільність присутності у моделі обраної на етапі 3 змінної. Якщо вона не проходить перевірку на значущість, то проводиться її виключення із моделі, після чого здійснюється перехід до етапу 2, але вже з розрахунку, що зазначена змінна у моделі не присутня. В іншому випадку, коли змінна виявляється значущою, алгоритм зупиняється.

Метод двоспрямованого усунення є модифікацію методу прямого відбору, що відрізняється від нього тим, що на кожному кроці після включення нової змінної в модель здійснюється перевірка на значущість інших змінних, які вже були введені в неї раніше. Якщо такі змінні будуть виявлені, їх слід вивести зі складу моделі. Після коригування списку включених у модель змінних здійснюється чергова ітерація процедури прямого відбору пошуку нової змінної, що задовольняє умовам включення її до складу моделі.

Результатом застосування методу покрокової регресії є нові знання у вигляді рівняння регресії із визначеними значущими параметрами.

3.2 Дослідження методу гребеневої регресії

Гребенева регресія – це метод оцінки коефіцієнтів моделей множинної регресії у сценаріях, у яких незалежні змінні сильно корельовані. Метою даного виду регресії є усунення мультиколінеарності у моделях даних. Гребенева регресія була розроблена як можливе вирішення проблеми неточності оцінок найменших квадратів, коли моделі лінійної регресії мають кілька мультиколінеарних (сильно корельованих) незалежних змінних – шляхом створення оціночної функції [19]. Це забезпечує більш точну оцінку параметрів, так як оцінка дисперсії та середнього квадрата часто менша, ніж оцінки, отримані раніше методом найменших квадратів. Даний метод використовується у багатьох областях, включаючи економетрику, хімію та інженерію.

Гребенева регресія використовується для вирішення наступних двох типів проблем:

- точки даних менше, ніж кількість змінних;
- між змінними існує колінеарність.

В цілому, гребеневу регресію можна вважати удосконаленням лінійної регресії з підвищеною стійкістю до помилок, що накладає обмеження на коефіцієнти регресії для отримання більш наближеного до реальності результату. До того ж, цей результат набагато простіше інтерпретувати. Застосовується метод боротьби з надмірністю даних, коли незалежні змінні мультиколінеарні. Отже, гребенева регресія має такі властивості:

- гребенева регресія може вирішити проблему з більшою кількістю функцій, ніж розмір вибірки;
- гребенева регресія як алгоритм скорочення може визначати, які функції є важливими або неважливими, що певною мірою схоже на ефект зменшення розмірності;

– алгоритм скорочення можна розглядати як збільшення зміщення моделі при одночасному зменшенні дисперсії.

Вхідні дані для проведення процедури гребеневої регресії мають зберігатися у числовому форматі. На вхід алгоритму також подаються матриця спостережень X розмірності $m \times n$ та вектор-стовпець відгуків у розмірності m , де n – об'єм вибірки, а m – кількість спостережень. Кожен елемент повинен мати чіткий набір значень; рядки з пропущеними значеннями активних змінних ігноруються.

Припущення гребеневої регресії такі ж, як і у звичайній множинній регресії: лінійність, постійна дисперсія (без викидів) та незалежність. Оскільки гребенева регресія не забезпечує довірчих інтервалів, нормальність розподілу залишків не потрібна.

Загальне рівняння лінійної моделі множинної регресії у результаті застосування методу гребеневої регресії приймає вигляд (3.1).

У стандартній лінійній регресії вектор-стовпець y розміром $n \times 1$ має бути спроектований на простір стовпців матриці даних X $n \times m$, стовпці якої сильно корельовані. Звичайна оцінка методом найменших квадратів коефіцієнтів β має наступний вигляд:

$$\beta = (X^T X)^{-1} X^T y, \quad (3.6)$$

де X^T – транспонована матриця X .

Метод найменших квадратів (МНК) – це тип лінійного методу найменших квадратів для оцінки невідомих параметрів моделі лінійної регресії. МНК вибирає параметри лінійної функції набору незалежних змінних за принципом найменших квадратів: мінімізуючи суму квадратів відмінностей між залежною змінною (значення спостережуваної змінної) в даному наборі даних і передбаченими лінійною функцією незалежної змінної.

В умовах мультиколінеарності матриця $X^T X$ стає погано обумовленою. Для вирішення цієї проблеми на величину коефіцієнтів накладається обмеження, тоді рівняння приймає вигляд:

$$\beta = (X^T X + \lambda I_n)^{-1} X^T y, \quad (3.7)$$

де λ – невід’ємний параметр;

I_n – одинична матриця розмірністю $n \times n$.

Ця зміна збільшує власні значення матриці $X^T X$, але не змінює її власні вектори. Діагональна матриця λI_n називається гребнем. Додавання параметра λ вирішує проблему поганої обумовленості матриці $X^T X$; ці оцінки зміщені, на відміну МНК-оцінок, при цьому збільшується визначник матриці. Проте доведено, що є така λ , при якій ці оцінки ефективніші, ніж оцінки МНК (оцінки МНК ефективні серед лінійних незміщених оцінок). Чітких правил вибору цього параметра немає.

Якщо $\lambda = 0$, ми отримуємо ті самі коефіцієнти, що і за простої лінійної регресії; якщо $\lambda = \infty$, коефіцієнти дорівнюють нулю через нескінченну вагу в квадраті коефіцієнтів.

Результатом використання методу гребеневої регресії є нові знання у вигляді рівняння регресії із визначеними значущими параметрами та лінійно залежними змінними.

3.3 Порівняльний аналіз та обґрунтування вибору методу регресійного аналізу

Із розглянутих методів регресійного аналізу необхідно виділити такий метод, що дозволить повністю вирішити завдання, наведені у розділі 2. Для

вибору доцільно розробити систему критеріїв, що дозволять об'єктивно обґрунтувати вибір методу.

Розглянемо детальніше роботи з оцінювання та вибору одного з розглянутих методів регресійного аналізу. Суть цих робіт полягає у наступному:

- створення на основі висунутих вимог опису «ідеального» методу регресійного аналізу;
- вибір з усієї множини критеріїв оцінювання методу регресійного аналізу підмножини критеріїв, які є найбільш важливими для вирішення поставлених задач;
- проведення оцінювання конкретних методів регресійного аналізу за обраною підмножиною найбільш важливих критеріїв;
- порівняння оцінених методів регресійного аналізу з описом «ідеального» методу;
- вибір методу регресійного аналізу, результати оцінювання якого найбільш співпадають з описом «ідеального» методу.

Отже, в загальному випадку задачу вибору методу регресійного аналізу пропонується розглядати як задачу пошуку такого методу, опис якого найменш відмінний від опису «ідеального» методу регресійного аналізу.

Опис методу регресійного аналізу, що дозволить повною мірою вирішити задачу, наведені у розділі 2, позначимо M_{RA}^* , при цьому описи розглянутих методів регресійного аналізу позначимо M_{R1} та M_{R2} для методу покрокової та гребеневої регресії відповідно ($M_{Ri}, i = 1,2$). Тоді задача вибору методу регресійного аналізу буде полягати у виділенні такого опису методу регресійного аналізу, відмінності якого від опису M_{RA}^* будуть мінімальними. Дана задача може бути описана наступною моделлю:

$$F = M_{RA}^* - M_{Ri} \rightarrow \min. \quad (3.8)$$

Оскільки M_{RA}^* та M_{Ri} повинні існувати як текстові описи, необхідне виконання наступних умов:

$$\begin{cases} M_{RA}^* \neq \emptyset \\ M_{Ri} \neq \emptyset, i = 1, 2. \end{cases} \quad (3.9)$$

Кількість критеріїв оцінювання в кожному конкретному процесі вибору методу регресійного аналізу є скінченною, тому виникає можливість перейти до кількісного представлення цієї моделі.

Для кількісного опису відмінностей опису M_{RA}^* від M_{Ri} введемо інтегральний критерій збігу $C_{conc}(Ri)$. Даний критерій буде приймати значення у відсотках, що характеризують ступінь збігу опису M_{RA}^* та опису конкретного досліджуваного методу регресійного аналізу M_{Ri} . Якщо виконується умова $M_{RA}^* \equiv M_{Ri}$, то $C_{conc}(Ri) = 100\%$.

Узагальнений опис M_R буде складатися із множини локальних критеріїв, яку можна описати наступним кортежем:

$$M_R = \langle C_1, C_2, C_3, C_4, C_5, C_6 \rangle, \quad (3.10)$$

де C_1 – критерій «відповідність опису вихідних даних»;

C_2 – критерій «придатність працювати із лінійно залежними значеннями»;

C_3 – критерій «ступінь відповідності даних припущенням методу»;

C_4 – критерій «складність (при виконанні методу витрачається небагато часу)»;

C_5 – критерій «невибагливість до знань та вмінь аналітика, що використовує метод»;

C_6 – критерій «ступінь відповідності результату застосування методу результату вирішення поставлених задач».

Для підвищення точності порівняння змісту описів методів подамо інтегральний критерій $C_{conc}(Ri)$ як нормовану суму критеріїв збігу окремих

розділів описів методів:

$$C_{conc}(Ri) = \frac{\sum_{j=1}^6 C_{conc}(C_j)}{6}, \quad (3.11)$$

де j – кількість критеріїв описів, $j = 1, 2, \dots, 6$;

$C_{conc}(C_j)$ – критерій збігу j -го локального критерію описів M_{RA}^* та M_{Ri} .

Критерії $C_{conc}(C_j)$ також приймають відсоткові значення, що характеризують ступінь збігу критеріїв описів методів регресійного аналізу.

Для кожного конкретного підприємства важливість критеріїв вибору методу регресійного аналізу буде різною, тому використовувати вагові коефіцієнти, які дозволяють встановити ступінь відносної важливості коефіцієнтів описів методів регресійного аналізу, недоцільно.

Для використання моделі (3.8) – (3.9) у процесі вибору методу регресійного аналізу пропонується визначати значення локальних критеріїв збігу $C_{conc}(C_j)$ за допомогою методів експертного оцінювання. Дані методи базуються на виконанні наступних дій [20]:

- висування вимог до експертів, які будуть приймати участь в оцінюванні;
- формування експертних груп;
- формування списку варіантів, які повинні бути пред'явлені експертам;
- проведення опитування експертів відповідно до однієї із прийнятих форм (інтерв'ю або анкетування різного виду);
- оцінювання експертами запропонованих варіантів відповідно до обраного формулювання задачі оцінювання та шкали оцінювання;
- обробка результатів оцінювання з урахуванням тренованості та компетентності експертів.

Отже, базуючись на запропонованій моделі (3.8) – (3.9) можна виділити спосіб вибору методу регресійного аналізу, що складатиметься з таких етапів.

Етап 1. Визначення множини локальних критеріїв оцінювання,

зазначених у (3.10).

Етап 2. За обраними на Етапі 1 критеріями визначити множину потреб аналітиків, що проводять аналіз моніторингу продуктивності ВБІС, у методі регресійного аналізу.

Етап 3. В результаті обробки визначених на Етапі 2 потреб з врахуванням відповідності обраним на Етапі 1 критеріям оцінювання елементів опису M_{RA}^* необхідно сформулювати опис такого методу регресійного аналізу M_{RA}^* , який надасть можливість в повній мірі вирішити поставлені задачі.

Етап 4. Сформулювати описи досліджуваних методів регресійного аналізу M_{Ri} шляхом пошуку інформації, що знаходиться у відкритому доступі, з врахуванням відповідності елементів кожного опису M_{Ri} обраним на Етапі 1 критеріям оцінювання.

Етап 5. Обрати групу з p експертів із числа аналітиків, що проводять аналіз моніторингу продуктивності ВБІС на підприємстві.

Етап 6. Запропонувати експертам групи, обраної на Етапі 5, для порівняння паспорти M_{RA}^* та M_{Ri} .

Етап 7. Визначити значення локальних критеріїв $C_{conc}(C_j)$ шляхом анкетування експертів групи, обраної на Етапі 5.

Етап 8. Розрахувати усереднене значення локального критерію за середнім арифметичним.

Етап 9. Розрахувати значення інтегрального критерію збігу $C_{conc}(C_j)$ за формулою (3.11), базуючись на отриманих на Етапі 7 результатах.

Етап 10. Знайти рішення задачі вибору методу регресійного аналізу за допомогою моделі (3.8) – (3.9) та завершити виконання методу.

Під час виконання Етапів 6, 7 та 8 було проведено анкетування обраних експертів, визначені значення локальних критеріїв $C_{conc}(C_j)$. Результати наведено у таблицях 3.1-3.2.

Таблиця 3.1 – Результати визначення та обробки значень локальних критеріїв $C_{conc}(C_j)$ для покрокового методу регресійного аналізу

Локальний критерій	Значення першого експерту, %	Значення другого експерту, %	Значення третього експерту, %	Значення четвертого експерту, %	Усереднене значення локального критерію, %
$C_{conc}(C_1)$	95	100	96	99	97,5
$C_{conc}(C_2)$	5	3	10	5	5,75
$C_{conc}(C_3)$	100	95	94	96	93,75
$C_{conc}(C_4)$	94	95	100	95	96
$C_{conc}(C_5)$	100	96	95	95	96,5
$C_{conc}(C_6)$	75	80	73	70	74,5

Таблиця 3.2 – Результати визначення та обробки значень локальних критеріїв $C_{conc}(C_j)$ для гребеневого методу регресійного аналізу

Локальний критерій	Значення першого експерту, %	Значення другого експерту, %	Значення третього експерту, %	Значення четвертого експерту, %	Усереднене значення локального критерію, %
$C_{conc}(C_1)$	100	95	96	99	97,5
$C_{conc}(C_2)$	95	90	100	93	94,5
$C_{conc}(C_3)$	100	96	93	96	96,25
$C_{conc}(C_4)$	93	92	95	96	94
$C_{conc}(C_5)$	90	94	96	95	93,75
$C_{conc}(C_6)$	100	95	95	96	96,5

Під час виконання Етапу 8 були розраховані значення інтегрального критерію $C_{conc}(C_j)$ для кожного з досліджених методів регресійного аналізу за формулою (3.11). Результат наведено у таблиці 3.3.

Таблиця 3.3 – Результати розрахунків значень інтегрального критерію

№ методу	Назва методу	Значення $C_{conc}(C_j)$, %
1	Покрокова регресія	77,33
2	Гребенева регресія	95,42

3.4 Висновки до третього розділу

У третьому розділі було досліджено методи покрокової та гребеневої регресії, визначено їхні цілі та наведено припущення даних методів. Було визначено, що метод покрокової регресії дозволяє визначити лише значущі коефіцієнти рівняння регресії, але при наявності мультиколінеарності необхідно проводити додатковий аналіз даних.

Метод гребеневої регресії дозволяє як знайти лінійно залежні змінні, так і визначити значущі коефіцієнти рівняння регресії, що є значною перевагою при проведенні вибору методу регресійного аналізу для проведення аналізу моніторингу продуктивності ВБІС.

Також було введено інтегральний критерій, за допомогою якого було проведено вибір методу регресійного аналізу. Інтегральний критерій складається із наведених у підрозділі 3.3 локальних критеріїв, значення яких було отримано у ході анкетування експертів групи, обраної із числа аналітиків, що проводять аналіз моніторингу продуктивності ВБІС на підприємстві. У результаті проведення розрахунків за (3.11) було визначено, що метод гребеневої регресії має більший інтегральний коефіцієнт, отже, більше підходить для вирішення поставлених задач.

4 ПРАКТИЧНА АПРОБАЦІЯ РЕЗУЛЬТАТІВ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

4.1 Опис експлуатованої веб-базованої інформаційної системи

У якості експлуатованої веб-базованої ІС розглядається комплексна система автоматизації страхової компанії (КСАСК), розроблена компанією «ПрофІТсофт». Дана компанія займається розробкою, постачанням та документуванням готового програмного забезпечення індивідуального та загального користування, інформаційно-обчислювальним обслуговуванням, виробництвом програмних засобів, а також виготовленням програм на замовлення користувача. Ключовим ринком компанії є Європа; на ринку України компанія відома як розробник комплексної системи автоматизації роботи страхової компанії.

Компанія була заснована у 2002 році випускниками Харківського національного університету радіоелектроніки за участі німецьких партнерів, які й стали першими замовниками. Мета діяльності «ПрофІТсофт» полягає у отриманні прибутку та реалізації соціальних та економічних інтересів учасників, а також задоволення суспільних потреб у виконанні товариством робіт та послуг шляхом здійснення науково-дослідної, виробничої та комерційної діяльності. Місією «ПрофІТсофт» для внутрішнього ринку України є комплексна розробка веб-базованих інформаційних систем для будь-якого суб'єкта суспільства задля справедливого та максимально надійного страхування.

ТОВ «ПрофІТсофт» спеціалізується на наступних предметних областях:

- страхування;
- фінанси;
- облікові системи;
- освітні портали.

При цьому спеціалізацією компанії є Web 2.0 застосунки, Legacy системи і проекти, веб-сервіси та оптимальні рішення складних завдань.

Серед практик, використовуваних компанією «ПрофІТсофт», наявні гнучкі методології розробки (Agile), використання процесу розробки програмного забезпечення через приймальні тести, постійний контроль якості коду, якісний опис завдань і вимог, забезпечення необхідного рівня контролю якості, а також ефективне використання ресурсів.

Веб-базована інформаційна система КСАСК передбачає повну автоматизацію страхового бізнесу [21]. Дана система складається із підсистем «back-office» та «front-office», а також наступних модулів:

- страхові калькулятори;
- облік договорів;
- імпорт списків даних;
- андеррайтинг;
- облік бланків;
- комісійні;
- фінанси;
- врегулювання;
- облік договорів добровільного медичного страхування;
- звітність у Моторне (транспортне) страхове бюро України;
- адміністрування;
- CRM;
- бізнес-процеси;
- перестраховання;
- інтернет-магазин;
- електронний поліс та інтеграція з зовнішніми системами;
- релізи;
- допомога.

КСАСК є об'єктом моніторингу, який проводиться під час експлуатації системи. Під об'єктом моніторингу варто розуміти систему загалом та,

зокрема, її окремі компоненти. Об'єкт моніторингу визначає набір джерел показників, які можуть бути доступні в системі моніторингу, на базі яких здійснюватиметься аналіз продуктивності. Об'єкти моніторингу зазвичай ієрархічні і збирання інформації з них варто розглядати в контексті дерева ієрархії.

Для аналізування стану об'єкта у моніторингу існує поняття метрики. У загальному випадку метрика може бути представлена наступними параметрами:

- тип метрики;
- ідентифікатор об'єкта моніторингу в ієрархії;
- дата зняття метрик;
- значення метрики.

Наступним базовим елементом будь-якої концепції моніторингу є агент. Ідея агенту полягає в тому, щоб виявити об'єкт моніторингу та встановити на нього датчик, який зніматиме метрику. У найпростішому вигляді цим агентом може бути людина, яка встановить необхідне програмне забезпечення для об'єкту моніторингу та налаштує його роботу. Датчик повинен проводити формування інформації та надсилати її для подальшої обробки.

Сформулюємо основні вимоги для таких агентів:

- агенти повинні ефективно встановлюватись на основні елементи інфраструктури (сервера, віртуальні машини, активні мережеві елементи);
- агенти повинні автоматично встановлювати датчики на об'єкт моніторингу, запущені на елементах інфраструктури;
- агенти повинні інструментувати об'єкти моніторингу під час експлуатації, не вимагаючи внесення змін до вихідного дистрибутиву;
- агенти повинні взаємодіяти з відомими протоколами та підтримувати додавання додаткових параметрів для забезпечення можливості відстеження бізнес-транзакції через кілька компонентів;
- агенти повинні встановлюватися як на серверні, і на клієнтські компоненти системи;

– агенти повинні підтримувати модель, що розширюється, щоб розробники конкретних систем могли підключати до моніторингу свої пропрієтарні рішення.

Завдяки такому підходу системи моніторингу можна розвивати як окремі незалежні елементи. Цільова система, у свою чергу, повинна враховувати ті обмеження, які накладаються системами моніторингу, щоб процес інтеграції був менш витратним.

На даний момент використовуються наступні засоби для моніторингу КСАСК:

- бібліотека `sam-monitor` – для моніторингу продуктивності КСАСК на промислових серверах;
- моніторинг життєдіяльності сервісів у `back-office`;
- `Zabbix` – для моніторингу доступності КСАСК для клієнтів, моніторинг локальних ресурсів компанії «ПрофІТсофт», ресурси серверів.

В `back-office` по кожному зверненню користувача до системи, або при запуску системних процесів, по їх завершенню відбувається запис відомостей про операцію та час її виконання, використання ресурсів сервера у спеціальний лог-файл. Це робить бібліотека `sam-monitor`. Такі записи робляться на кожному екземплярі системи окремо, в окремі лог-файли, які потім завантажуються в індекси `ElasticSearch` компанії.

`ElasticSearch (ES)` – масштабована утиліта повнотекстового пошуку та аналітики, яка дозволяє в режимі реального часу швидко зберігати, шукати та аналізувати великі обсяги даних. `ES` є ядром `ELK-стеку (Elastic Stack)`, до складу якого, крім `ElasticSearch`, входять такі продукти:

- `Logstash` – інструмент збору, перетворення та збереження в загальному сховищі подій з різних джерел (файли, бази даних, логи тощо) в режимі реального часу;
- `Kibana` – веб-інтерфейс для `ElasticSearch` для взаємодії з даними, що зберігаються в його індексах через динамічні панелі моніторингу, таблиці, графіки та діаграми, що відображають зміни в `ES`-запитах у реальному часі;

– FileBeat – агент на серверах для відправлення різних типів оперативних даних в ES.

Запис відомостей про операцію та час її виконання відбувається у два етапи. На першому файл архівується і розміщується в спеціальну папку на сервері клієнта. На другому етапі файли з папки автоматично зчитуються з цільових папок за допомогою спеціальної окремої програми sam-importer; з них зчитується вся інформація і відповідні дані записуються в індекси Elasticsearch.

В режимі реального часу проводиться моніторинг наступних сервісів КСАСК:

- електронний поліс (ЕП);
- електронно-цифровий підпис (ЕЦП);
- 1С бухгалтерія – зв’язок з back-office;
- курс валют НБУ;
- генерація документів;
- доступність сервісу Fishka;
- сповіщення користувачів;
- обробка запитів користувачів.

За допомогою моніторингу Zabbix налаштовано функціонал для перевірок доступності стартових сторінок (сторінка входу до КСАСК) промислових та тестових КСАСК клієнтів компанії.

Також проводиться моніторинг:

- життєвого циклу SSL сертифікатів;
- життєвого циклу ЕЦП сертифікатів;
- сторінки логів у back-office;
- сторінки логів у back-office;
- обробки запитів у базі даних.

Наразі сповіщення користувачів здійснюється засобами Pushbullet та повідомляє про недоступність або про проблеми в роботі КСАСК клієнтів та проблеми локальних сервісів. Pushbullet – це сервіс, який дозволяє

обмінюватися контентом між пристроями; він вміє відображати на комп'ютері сповіщення зі смартфона та слугує RSS клієнтом. Для сповіщення про проблеми функціоналу КСАСК клієнтів використовується сповіщення засобами ElastAlert (плагін ElasticSearch).

4.2 Проведення аналізу результатів моніторингу із використанням методу гребеневої регресії

Моніторинг веб-базованої ІС проводився протягом доби. Значення показників збиралися щогодини у кількості 21 значень для одного елемента системи за годину. Файл моніторингу містить номер елемента системи, який описує окрему функцію бізнес-транзакції; даний елемент містить 21 значення окремих показників, тобто окремих сервісів, що беруть участь у даній функціональній транзакції.

Аналіз результатів проведеного моніторингу проведено з використанням програмного забезпечення Statistica – програмного пакету для статистичного аналізу, розробленого компанією StatSoft, що реалізує функції аналізу даних, управління даними, видобутку даних, візуалізації даних із залученням статистичних методів. Дане ПЗ має необхідний функціонал для проведення аналізу із використанням методу гребеневої регресії.

Задля проведення аналізу необхідно завантажити до програмного пакету файл даних із результатами моніторингу. Вихідний файл є пласким файлом із розширенням txt; для проведення аналізу його було конвертовано у xlsx файл із розмеженням на відповідні рядки та стовпці. Замірювані значення елементів системи у ході проведення аналізу виступають як незалежні змінні та позначаються $X_i, i = 1, 2, \dots, 12$; параметр функції є залежною змінною та позначається Y . Стовпець «Година» не використовується при проведенні аналізу. Частина файлу даних наведено на рисунку 4.1.

Дані з моніторингу													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
Година	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	Y
1	1 8455,064	140,97	1588,36	19,90817	44,35711	8,3875	23,98166	98,67104	1827,471	389	518,67	0,013151	0,262886
2	1 5347,259	1503,373	0,03	7,419245	67,94697	47,53	90,75185	103,0829	724,4776	9062,87	642,03	0,077143	0,567261
3	1 4438,211	1202,124	392	8,526448	93,56469	38,79	46,51523	68,39912	1967,338	2388	1589,92	0,106705	0,968989
4	1 8343,132	1964,316	1406,58	5,763166	58,48795	8138,91	5,605058	143,7761	1121,153	9057,65	1408,39	0,771589	1,142715
5	1 5296,89	1237,973	38,88	16,70424	125,9984	8134,71	13,89273	6,685375	4069,171	0,03	14,62	0,939116	1,149048
6	1 9950,422	712,7673	1401,42	11,99009	93,70457	100	72,58879	4,28862	5546,184	8141,49	21,61	0,105263	0,217975
7	1 1651,79	475,0435	553,31	13,4794	128,3542	8,4337	42,45189	122,6046	1787,41	47,16	553,4	0,398045	2,576149
8	1 3001,993	265,1628	2388,15	9,484875	86,90782	47,43	51,76661	109,5069	4876,794	21,61	2388,07	0,505407	1,532368
9	1 7541,883	2217,944	0,03	2,152735	149,8179	38,7	29,97249	61,83777	4252,149	2388,03	9053,65	0,255994	2,566316
10	1 3383,283	2087,192	9058,86	7,852471	73,29234	522,22	38,14575	109,0047	5693,092	391	1,3	0,797087	2,880084
11	1 4590,764	1337,452	1,3	6,752851	95,41225	21,61	33,91613	66,00476	4348,992	8,414	47,71	0,491923	0,155267
12	1 6508,605	627,8576	2388,13	11,78917	26,3503	2388,08	1,288576	132,6222	5397,609	521,97	522,2	0,264127	1,829231
13	1 2540,168	527,371	2388	4,578114	15,49371	9049	68,21862	59,76985	1022,928	8,4277	2388,13	0,119506	1,679905
14	1 7663,022	2130,061	642,03	19,73044	116,5855	8,4139	36,67318	139,8708	1157,615	100	8131,17	0,040005	0,279467
15	1 3478,767	676,4846	553,01	2,623421	103,0982	2388,06	30,80447	35,08266	1278,54	1396,57	8,433	0,713315	1,269543
16	1 5824,282	1788,045	8132,69	0,243167	100,7208	9055,37	92,55016	147,3795	3362,27	100	0,03	0,616758	1,199607
17	1 6069,699	609,7054	1593	17,98855	27,14215	2388,13	21,36911	55,97132	2422,806	23,3718	392	0,971083	1,878216
18	1 9731,492	26,96478	9050,6	1,498853	65,8038	642,22	52,09938	64,65279	2769,947	47,14	2388	0,729927	0,54926
19	1 2033,461	273,5361	2388,12	10,24281	92,17393	393	37,0917	101,9363	3014,409	8,3569	100	0,907518	0,33831
20	1 8850,385	162,3259	8,4439	1,968011	112,1759	100	48,7403	148,2516	1212,143	1391,67	38,99	0,907943	0,636701
21	1 6708,568	2486,893	8133,81	0,646609	10,20472	38,8	21,01357	7,770841	3690,026	39,08	23,296	0,721618	1,841974
22	2 6127,262	1370,028	521,43	12,8598	26,58297	2388,08	22,3578	145,7292	294,5038	47,27	518,67	0,918431	0,696241

Рисунок 4.1 – Відображення частини значень показників продуктивності, зібраних у ході проведення моніторингу експлуатованої ВБІС

Після проведення аналізу можна побудувати рівняння регресії із використанням виведеної таблиці «Підсумок гребеневої регресії». Таким чином, можна визначити, які змінні є лінійно залежними одна від одної, а також які із змінних є значущими для продуктивності усієї системи.

Якщо отриманий коефіцієнт при певній змінній за значенням суттєво перевищує інші коефіцієнти, можна зробити висновок, що дана змінна найбільш сильно впливає на продуктивність порівняно з іншими; таким чином у разі модифікації системи продуктивність збільшуватиметься за рахунок більш значущих показників.

Рівняння регресії будується з використанням значення параметру b , який відображається у таблиці «Підсумок гребеневої регресії». Значущі коефіцієнти виділені червоним кольором, при цьому значення параметру p -value повинне бути меншим за 0,05.

Результат проведення аналізу із використанням методу гребеневої регресії наведено на рисунку 4.2.

Ridge Regression Summary for Dependent Variable: Y (Дані з моніторингу in ANALYSIS)						
N=504	b*	Std.Err. of b*	b	Std.Err. of b	t(491)	p-value
Intercept			1,414595	0,186925	7,567713	0,000000
X1	-0,042918	0,043585	-0,000013	0,000013	-0,984714	0,325250
X2	0,107068	0,043640	0,000129	0,000053	2,453455	0,014496
X3	0,055750	0,044079	0,000019	0,000015	1,264760	0,206557
X4	-0,021288	0,043729	-0,003250	0,006676	-0,486820	0,626603
X5	-0,026617	0,044267	-0,000532	0,000884	-0,601285	0,547928
X6	0,007077	0,043856	0,000008	0,000048	0,161379	0,871861
X7	0,026875	0,044056	0,000855	0,001402	0,610024	0,542128
X8	-0,022562	0,043782	-0,000447	0,000867	-0,515327	0,606557
X9	-0,025168	0,044118	-0,000014	0,000024	-0,570459	0,568628
X10	-0,039673	0,043734	-0,000014	0,000015	-0,907144	0,364776
X11	0,065651	0,044116	0,000023	0,000015	1,488140	0,137356
X12	0,030104	0,044018	0,091559	0,133877	0,683904	0,494359

Рисунок 4.2 – Відображення підсумку гребеневої регресії

Як можна побачити на рисунку 4.2, для досліджуваного набору даних значущим виявився лише коефіцієнт при змінній X_2 , а також вільний член рівняння регресії; p-value для даних коефіцієнтів є меншим за 0,05.

Таким чином, рівняння регресії матиме наступний вигляд: $Y = 1,1416 + 0,000129 * X_2$.

4.3 Висновки до четвертого розділу

У четвертому розділі було визначено та описано веб-базовану інформаційну систему, для якої на підприємстві «ПрофІТсофт» проводиться моніторинг продуктивності. Система являє собою комплексну систему автоматизації роботи страхової компанії; вона складається з двох підсистем:

«back-office» та «front-office», а також має у наявності вісімнадцять модулів, що дозволяють користувачу виконувати різні бізнес-сценарії. Дана система є об'єктом моніторингу; для аналізування його стану використовуються певні метрики.

Аналіз результатів моніторингу зберігається у пласкому файлі, у даному випадку проводився аналіз даних, що було записано до текстового файлу. Даний файл було конвертовано у формат `xlsx` у зв'язку з вимогами ПЗ, за допомогою якого проводився аналіз. Результатом проведення аналізу є побудоване рівняння регресії, з якого можна зробити висновок, що значущий коефіцієнт належить лише змінній X_2 із досліджуваного набору даних; інші змінних наявна виражена лінійна залежність, тому при розрахунку продуктивності як функціональних, так і технічних транзакцій.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було проведено дослідження існуючих видів моделей веб-базованих інформаційних систем та визначено відмінності між ними. Водночас усі сучасні веб-застосунки мають спільну особливість: моніторинг продуктивності для них проводиться як для сукупності чорних ящиків, які взаємодіють один з одним через певні інтерфейси. Було визначено, що такий підхід унеможливорює оцінювання продуктивності ВБІС в цілому та може стати причиною збільшення витрат на проведення процесу моніторингу.

Було визначено основний спосіб здійснення моніторингу продуктивності у вигляді проведення пасивного експерименту. У ході проведення такого експерименту фіксуються значення показників найнижчого рівня абстракції. Запропоновано моделі визначення продуктивності як окремих транзакцій, так і бізнес-сценаріїв.

У якості методів проведення аналізу моніторингу продуктивності ВБІС було запропоновано покроковий та гребеневий методи регресійного аналізу та складено їхній опис. Вивчення покрокової регресії виявило, що даний метод потребує додаткової обробки результатів моніторингу через неможливість обробляти лінійно залежні дані, які можуть з'явитися у ході проведення моніторингу. При проведенні порівняльного аналізу із використанням інтегрального критерію збігу та експертних оцінок було зроблено висновок, що для проведення аналізу результатів моніторингу доцільно використовувати метод гребеневої регресії.

Із використанням даних моніторингу продуктивності експлуатованої веб-системи КСАСК було проведено аналіз даних із використанням гребеневої регресії, який дозволив побудувати регресійне рівняння для вказаного набору даних та виявити значущі показники продуктивності ВБІС.

ПРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chris Love. *Progressive Web Application Development by Example*. Birmingham : «Packt Publishing», 2018. 354 p.
2. Державний стандарт України ДСТУ 3008-2015: Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. Київ : ДП «УкрНДНЦ», 2016. 31 с.
3. Amit Sheth, Miltiadis D. Lytras. *Semantic Web-Based Information Systems: State-of-the-Art Applications*. Hershey : «IGI Global», 2006. 317 p.
4. Сусіденко В. Т. Інформаційні системи і технології в обліку : навч. посіб. К. : «Центр учбової літератури», 2016. 224 с.
5. Сазонець О. М. Інформаційні системи і технології в управлінні зовнішньоекономічною діяльністю. К. : «Центр учбової літератури», 2015. 256 с.
6. Guangzhi Zheng. *A Historical Perspective of Web Engineering*. *Encyclopedia of Networked and Virtual Organizations*. Hershey : «IGI Global», 2008. P. 660-667.
7. Common web application architectures. URL: <https://bit.ly/2ZG4jJw> (дата звернення 19.11.2021).
8. Modern Web Application Architecture Explained: Components, Best Practices and More. URL: <https://litslink.com/blog/web-application-architecture> (дата звернення 20.11.2021).
9. Luke Welling, Laura Thomson. *PHP and MySQL Web Development*. Indianapolis : SAMS, 2003. 672 p.
10. Jess Chadwick, Todd Snyder, Hrusikesh Panda. *Programming ASP.NET MVC 4: Developing Real-World Web Applications with ASP.NET MVC*. Sebastopol : «O'Reilly Media», 2012. 492 p.
11. Klaus-Dieter Schewe, Bernhard Thalheim. *Design and Development of Web Information Systems*. Berlin : «Springer», 2019. 590 p.

12. ISO/IEC 20000-1:2018. URL: <https://www.iso.org/standard/70636.html> (дата звернення 26.11.2021)
13. What is ITIL? URL: <https://bit.ly/31HqCzc> (дата звернення 29.11.2021).
14. Web Performance Monitoring: A How to Guide for Developers. URL: <https://stackify.com/web-performance-monitoring-guide-developers> (дата звернення 23.10.2021).
15. A. N. Chubinskij, D. S. Rusakov, I. M. Batyreva. Metody planirovaniya i obrabotki rezul'tatov eksperimentov. SPb : SPbGLTU, 2018. 109 s.
16. Rob England. Owing Itil: A Skeptical Guide For Decision-Makers. Hastings : «Two Hills», 2009. 178 p.
17. Artamonov I. V. Modelirovanie biznes-tranzakcij. Irkutsk : «Izd-vo BGU», 2016. 190 s.
18. The Problem of Passive Observation. URL: <https://bit.ly/3EzYQn6> (дата звернення 30.11.2021).
19. Ludwig Fahrmeir, Thomas Kneib, Stefan Lang. Regression: Models, Methods and Applications. New York City : «Springer», 2013. 698 p.
20. Gucykova S. V. Metod ekspertnyh ocenok. Teoriya i praktika. Moskva : «Institut psihologii RAN», 2011. 170 s.
21. Комплексна система автоматизації страхової компанії «ProfITsoft» <http://www.profitsoft.ua/dsk.php> (дата звернення 30.11.2021)