

**26-28
ЖОВТНЯ
2021**

A hand in a grey suit jacket holds a black tablet. The background is a vibrant blue with a faint, glowing network of white lines and dots. Overlaid on this background is a cluster of white hexagons, each containing a different black icon. The icons represent various fields: a crossed-out power symbol, a cloud, a Wi-Fi signal, a crossed wrench and screwdriver, a lightbulb, a group of people, a pie chart, a world map, a factory, a circuit board, a gear with 'SERVICE' text, a head with gears, a computer monitor, a handshake, a truck, a hard hat, a flask, a camera with an arrow, a person on a motorcycle, and an '@' symbol. Some hexagons also contain binary code (0s and 1s).

Міністерство освіти і науки України
Інститут модернізації змісту освіти МОН України
Національний університет кораблебудування імені адмірала Макарова

II Всеукраїнська науково-практична інтернет конференція
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ:
МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2021)

26-28 жовтня
Миколаїв 2021

Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2021):
Матеріали II Всеукраїнської науково-практичної інтернет конференції
(26-28 жовтня 2021 р.). – Миколаїв: НУК імені адмірала Макарова, 2021. – 73 с.
Режим доступу: <http://itconf.nuos.edu.ua/2021/proceedings/>

ЗМІСТ

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ....5

Приходько С.Б., Пухалевич А.В., Халанчук І.П. НЕЛІНІЙНІ РЕГРЕСІЙНІ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНИХ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВАМИ JAVA ТА VISUAL BASIC ДЛЯ ПК	6
Трухов А.С., Приходько С.Б. АНАЛІЗ ІСНУЮЧИХ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ РОЗПІЗНАВАННЯ ОБРАЗІВ.....	8
Приходько С.Б., Приходько А.С., Шутко І.С. НЕЛІНІЙНІ РЕГРЕСІЙНІ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ PHP-ФРЕЙМВОРКІВ	11
Приходько С.Б., Сербулов Г.В. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ З КОДОМ НА PHP.....	13
Сорокунський О.Ю., Терлич С.В., Терлич А.С. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПОПЕРЕДЖЕННЯ ЗІТКНЕННЯ МОРСЬКИХ СУДЕН У ЗАЛЕЖНОСТІ ВІД ОСОБЛИВОСТЕЙ ОТОЧУЮЧОГО СЕРЕДОВИЩА	15
Приходько С.Б., Фаріонова Т.А. НЕЛІНІЙНІ РЕГРЕСІЙНІ РІВНЯННЯ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ РОЗРОБКИ JAVA-ЗАСТОСУНКІВ ДЛЯ ПЛАТФОРМИ MIDRANGE.....	20
Ponomarenko T., Gelenko Y., Kotula V. SOFTWARE APPLICATION FOR BUILDING A REGRESSION MODEL OF ESTIMATING THE PHP WEB APPLICATIONS EFFORT.....	23
Приходько С.Б., Смикодуб Т.Г. МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ПЕРЕВІРКИ ВЗАЄМОЗВ'ЯЗКІВ МІЖ ЧАСТИНАМИ JAVA-ЗАСТОСУНКІВ ЗА МЕТРИКАМИ RFC ТА СВО	25
Суслов С.В. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ПІДГОТОВЦІ ФАХІВЦІВ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ.....	27
Шакула О.В., Латанська Л.О. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ОЦІНЮВАННЯ РОЗМІРУ TELEGRAM БОТІВ НА PYTHON	30
Макарова Л.М., Нікітін О.В. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРОС-ПЛАТФОРМНОЇ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ПЛАТФОРМИ XAMARIN.....	32
Сторчеус О.В., Устенко І.В. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ CODEIGNITER	34
Чичерін В.О., Пухалевич А.В. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПЛАГІНІВ ДЛЯ WORDPRESS	36

УПРАВЛІННЯ ПРОЕКТАМИ І ПРОГРАМАМИ	38
Яковлева І.Р., Чернов С.К. МОДЕЛІ ТА ТЕХНОЛОГІЇ СУПРОВОДУ ПРОЦЕСІВ ДИСТАНЦІЙНОГО НАВЧАННЯ	39
Фараонов Ю.В., Чубчик Т.Т. ОСОБЛИВОСТІ ФОРМУВАННЯ ГІБРИДНОЇ КОМАНДИ ІТ-ПРОЕКТУ	42
Біла А.К. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОЛОГІЄЮ AGILE	45
Білявська О.В. ЖОРСТКІ НАВИЧКИ (HARD SKILLS) ЯК ДЕТЕРМІНАНТА УСПІХУ РІШЕННЯ ЗАВДАНЬ В УПРАВЛІННІ ПРОЕКТАМИ	48
Гайченко А.О. УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЕКТАМИ НА ОСНОВІ СУЧАСНИХ АНАЛІЗУЮЧИХ ПРОГРАМ	52
Корулик А.Д., Калабін С.Ю. ІНСТРУМЕНТИ УПРАВЛІННЯ ЯКІСТЮ В ПРОЕКТАХ ТУРИСТИЧНОЇ ГАЛУЗІ	54
Білоножко К.С., Білоножко М.С. ОРГАНІЗАЦІЯ РОБОЧОГО ПРОЦЕСУ ТА КОМУНІКАЦІЇ З КЛІЄНТОМ В УМОВАХ ДИСТАНЦІЙНОЇ РОБОТИ: ТЕХНІЧНА ПІДТРИМКА В ОНЛАЙН РЕЖИМІ	57
ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ	60
Шматко А.І., Барабаш О.В. АНАЛІЗ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ СПІВРОБІТНИКІВ ТА СТУДЕНТІВ КАФЕДРИ	61
Михалко М.І., Гайдаєнко О.В. АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ТА АДМІНІСТРУВАННЯ ВІДВІДУВАНOSTІ УНІВЕРСИТЕТУ	63
Черновалов О.В., Казимиренко Ю.О., Маршак О.І., Бобровський О.С. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ЗАДАЧ УПРАВЛІННЯ ТА ОРГАНІЗАЦІЇ НАВЧАЛЬНО-НАУКОВОЇ ЛАБОРАТОРІЇ ВИПРОБУВАНЬ НЕМЕТАЛІЧНИХ МАТЕРІАЛІВ	66
Lebed Y. MIGRATION OF HIGHLY CUSTOMIZED DESKTOP APPLICATIONS INTO CLOUD SERVICES.....	69
Заїченко Т.С., Книрік Н.Р. ДОСЛІДЖЕННЯ МЕТОДІВ І АЛГОРИТМІВ РОЗПІЗНАВАННЯ МОВИ ТА РОЗРОБКА СИСТЕМИ ГОЛОСОВОГО УПРАВЛІННЯ	70



МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ

УДК 004.9

НЕЛІНІЙНІ РЕГРЕСІЙНІ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНИХ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВАМИ JAVA ТА VISUAL BASIC ДЛЯ ПК

Приходько С.Б., д.т.н., проф.¹; Пухалевич А.В., к.т.н.²; Халанчук І.П.²
^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ sergiy.prykhodko@nuos.edu; ² andrii.pukhalevych@nuos.edu.ua

Анотація. Побудовано нелінійні регресійні моделі для оцінювання тривалості розробки в залежності від трудомісткості розробки програмних застосунків, що створюються мовами Java та Visual Basic для ПК.

Ключові слова: нелінійна регресія; оцінювання тривалості; розробка застосунків.

Вступна частина. Оцінювання тривалості розробки програмних застосунків є важливою задачею програмної інженерії, так як оцінки тривалості використовуються при заключенні договорів на розробку цих застосунків. Це вимагає математичних моделей, які дозволять виконати відповідне оцінювання.

До моделей, що використовуються для оцінювання тривалості розробки програмних застосунків, належать моделі ISBSG – нелінійні регресійні рівняння для оцінювання тривалості розробки програмного забезпечення в залежності від трудомісткості, які було побудовано для платформ PC (персональні комп'ютери – ПК), mid-range та mainframe [1]. Аналіз даних репозитарію ISBSG з релізу 2021 року [2] показав, що поділ моделей тільки по платформі вже є недостатнім, оскільки отримані моделі мають погані значення PRED та MMRE. Отже, виникає необхідність додавання інших ознак кластеризації даних. Однією з таких ознак, окрім платформи, природньо розглядати ще мову програмування.

Метою роботи є побудова нелінійних регресійних моделей для оцінювання тривалості розробки в залежності від трудомісткості розробки програмних застосунків, що створюються мовами Java та Visual Basic для ПК.

Основна частина. Для побудови нелінійних регресійних моделей для оцінювання тривалості розробки програмних застосунків, що створюються мовами Java та Visual Basic для ПК, використано метод побудови нелінійних регресійних моделей на основі нормалізуючих перетворень [3]. Цей метод полягає в тому, що емпіричні дані спочатку нормалізуються з застосуванням нормалізуючого перетворення, після чого будується лінійна регресійна модель за нормалізованими даними, а далі будується нелінійна регресійна модель з застосуванням перетворення, зворотного до нормалізуючого.

Щоб було можливо побудувати лінійну регресійну модель, нормалізовані дані мають мати нормальний розподіл, а відстань Махаланобіса для кожної пари значень має бути меншою чи дорівнювати критичному значенню. Розподіл залишків лінійної регресії повинен бути гаусівським, а нормалізовані значення емпіричних даних не повинні виходити за інтервал передбачення лінійної регресійної моделі. Тому з емпіричних даних потрібно видалити викиди, для чого виконуються наступні кроки:

1. Обчислюємо значення багатовимірної асиметрії нормалізованих емпіричних даних. Якщо це значення більше, ніж критичне значення, то з емпіричних даних видаляється пара значень з найбільшою відстанню Махаланобіса та знову переходимо до п. 1.

2. Обчислюємо значення багатовимірної експресії нормалізованих емпіричних даних. Якщо це значення більше, ніж критичне значення, то з емпіричних даних видаляється пара значень з найбільшою відстанню Махаланобіса та переходимо до п. 1.
3. За нормалізованими даними будуємо лінійну регресійну модель та перевіряємо закон розподілу залишків. Якщо розподіл не є гаусівським, то з емпіричних даних видаляємо пару значень з найбільшим за модулем залишку та переходимо до п. 1.
4. Якщо одне або більше нормалізованих значень даних виходить за інтервал передбачення лінійної регресії для нормалізованих даних, то ці значення видаляються та переходимо до п. 1.

Після видалення викидів за отриманою лінійною регресійною моделлю будується нелінійна регресійна модель із застосуванням перетворення, зворотного до нормалізуючого.

Емпіричними даними, що використовувалися при побудові нелінійних регресійних моделей для оцінювання тривалості розробки в залежності від трудомісткості розробки програмних застосунків, що створюються мовами Java та Visual Basic для ПК є дані репозитарію ISBSG з релізу 2021 року [2]. Нормалізація емпіричних даних виконувалася перетворенням у вигляді десятичного логарифму.

Для мови Java кількість емпіричних значень була 39, з них 7 значень було визначено як викиди. За значеннями, що залишилися було побудовано модель з наступними показниками якості: $R^2=0,9134$, $PRED(0,25)=0,9063$, $MMRE=0,1183$.

Для мови Visual Basic кількість емпіричних значень була 106, з них 27 значень було визначено як викиди. За значеннями, що залишилися було побудовано нелінійну регресійну модель з наступними показниками якості: $R^2=0,6487$, $PRED(0,25)=0,5063$, $MMRE=0,311$.

Висновки. В роботі було побудовано нелінійні регресійні моделі для оцінювання тривалості розробки програмних застосунків, що створюються мовами Java та Visual Basic для ПК. Показники якості побудованих моделей є кращими за відповідні показники нелінійної регресійної моделі, що створена для платформи ПК без врахування мови програмування. В подальшому планується розглянути можливість використання інших нормалізуючих перетворень при побудові нелінійних регресійних моделей для оцінювання тривалості розробки таких застосунків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Oligny, S., Bourque, P., Abran, A. & Fournier, B. (2000). Exploring the relation between effort and duration in software engineering projects. *Proc. of the 16th IFIP World Computer Congress* (pp. 175-178). Aug, 2000, Beijing, China
- [2] *International Software Benchmarking Standards Group*. (2021). Retrieved from <https://www.isbsg.org/>.
- [3] Приходько, С.Б. (2012). Метод побудови нелінійних рівнянь регресії на основі нормалізуючих перетворень. *Тези доповідей міждерж. наук.-методич. конф. "Проблеми математичного моделювання"* (с. 31-33). 13-15 червня 2012, Дніпродзержинськ, Дніпродзержинськ : ДДТУ.

Prykhodko S.B., Pukhalevych A.V., Khalanchuk I.P.

Nonlinear regression models for the development duration estimation of software applications created in Java and Visual Basic for PC

Abstract. Nonlinear regression models to estimate the development duration depending on the effort of developing software applications created in Java and Visual Basic for PC are built.

Keywords: nonlinear regression; duration estimation; software development.

УДК 004.93

АНАЛІЗ ІСНУЮЧИХ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ РОЗПІЗНАВАННЯ ОБРАЗІВ

Трухов А.С.¹, Приходько С.Б., д.т.н, проф.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ timka357@gmail.com; ² sergiy.prykhodko@nuos.edu.ua

Анотація. Розглянуто загальний алгоритм процесу розпізнавання образів. Проаналізовано існуючі математичні моделі для розпізнавання образів, а саме нейронні мережі та статистичні моделі, оцінено їх переваги та недоліки. Обґрунтовано необхідність розробки негаусівських ймовірнісних моделей для розпізнавання образів.

Ключові слова: розпізнавання образів; математичні моделі; негаусівські ймовірнісні моделі; нейронні мережі; статистичні методи.

Вступна частина. Методи розпізнавання образів – напрямок у технічних науках, пов'язаний з розробкою принципів і побудовою систем, що призначені для визначення належності досліджуваного об'єкту до конкретного класу із заздалегідь виділених. Досліджуваний об'єкт описується сукупністю визначальних характеристик. Обране вирішальне правило повинно бути засноване на використанні обмеженої кількості ознак, найбільш важливих для розрізнення одного образу від іншого.

Актуальність теми визначається стрімким розвитком технологічних досягнень, завдяки яким комп'ютерні системи та інформаційні технології розпізнавання образів конкурують з професіоналами з точки зору точності, але залишаються незмінними по швидкості і об'єму розглянутих випадків. Але сучасні системи та інформаційні технології розпізнавання образів, як правило, використовують вирішальні правила, які базуються на припущенні про нормальність розподілу, що в свою чергу призводить до зменшення вірогідності розпізнавання для негаусівських даних.

Метою роботи є підвищення вірогідності розпізнавання образів.

Основна частина. Завдання розпізнавання образів є завданням опису та класифікації об'єктів тієї чи іншої природи, це завдання створення поняття про клас об'єктів. Іншими словами, одне із завдань розпізнавання образів полягає в доцільному розбитті заданої сукупності об'єктів на однорідні класи. Об'єкти, що увійшли в один клас, повинні бути схожі один на одного або близькі один до одного за деяким критерієм схожості або близькості. Об'єкти ж з різних класів повинні бути досить різними, далекими один від одного.

Образ – це опис елемента як представника відповідного класу. Кожному образу x ставиться у відповідність певний вектор $x = (x_1, x_2, \dots)^T$ ознак x_i даного образу – елемент векторного простору X [1].

Загальний алгоритм розпізнавання образів виглядає так:

- попередня обробка вхідних даних;
- виділення контуру об'єктів;
- виділення характеристик;
- класифікація.

Найпопулярнішим методом є використання нейронних мереж. Нейрон має один вихід, який часто називають аксоном по аналогії з біологічним прототипом. З'єднаний виходом нейрона сигнал може надходити на довільне число входів інших нейронів. Якщо на виході нейрона є ненульовий сигнал (позитивний або негативний), то говорять, що нейрон активний або збуджений. У суматорі імпульси, що надходять на вхід нейрона, множаться на ваги входів і сумуються

$$x = \sum_{i=1}^n w_i x_i - c, \quad (1)$$

де n – кількість вхідних синапсів; w_i – ваги входів (позитивні або негативні); x_i – сигнали на входах; c – константа для формування порогу чутливості нейрона, яка називається зрушенням (bias). Відмінною особливістю нейронних мереж є здатність навчатися. Технічно, навчання – це процес знаходження коефіцієнтів зв'язків між нейронами.

Перевагами нейронних мереж є висока точність розпізнавання та здатність до навчання, як наслідок доволі просту побудову математичної моделі (1) при достатньо великому наборі навчальних даних.

Недоліком нейронних мереж є їхня недетермінованість. Мається на увазі те, що після навчання логіка прийняття рішень нейромережею зовсім схована від експерта. При невдалому навчанні будуть існувати об'єкти, що будуть помилково віднесені до іншого класу. А модифікувати таку систему в разі потреби вкрай важко, зазвичай використовуються спеціальні алгоритми донавчання моделі, які мають непередбачуваний вплив на всю систему в цілому. Також слід зазначити, що нейронні мережі потребують значно більше обчислювальних ресурсів для розпізнавання образу через велику кількість параметрів, кількість яких може становити більше ніж декілька мільярдів [2].

Підхід використання статистичних моделей базується на математичних правилах класифікації, які формуються і виводяться в термінах математичної статистики. Побудова статистичного класифікатора в загальному випадку передбачає застосування байєсового класифікаційного правила і його різновидів. Це правило забезпечує отримання оптимального класифікатора в тих випадках, коли відомі щільності розподілу для всіх сукупностей образів і ймовірності появи образів кожного класу. Вирішальні правила практично не залежать від специфіки об'єктів, оскільки вони мають справу не з самими об'єктами, а з деякими числовими характеристиками.

Якщо розглядати векторне представлення образів, то в загальному випадку, вектор ознак складається з компонент ознак, кожен з яких і їх сукупність в цілому характеризують образ з тим або іншим ступенем невизначеності. Ця невизначеність може, зокрема, носити і імовірнісний характер, хоча далеко і не вичерпується цим випадком. Іншими словами, кожен вектор ознак образу є багатовимірною випадковою величиною ξ . Поява того чи іншого образу є випадковою подією і ймовірність цієї події можна описати за допомогою закону розподілу ймовірностей цієї багатовимірної випадкової величини в тій чи іншій формі, наприклад, у формі щільності розподілу ймовірностей. Вид і параметри функції щільності визначаються конкретним середовищем, в якому працює система розпізнавання. Знаючи елементи навчальної вибірки – статистичну вибірку, можна відновити імовірнісні характеристики цього середовища [3].

Основними ймовірнісними характеристиками середовища є:

- функція щільності розподілу ймовірностей появи образу $f_{\xi}(x)$;
- умовні ймовірності приналежності деякого образу заданим класам $p(w_i|x^0)$;

- ймовірності появи класів $p_i = p(w_i)$;
- функції умовних щільностей розподілу ймовірностей образів всередині класів $f_i(x) = f_{\xi}(x|w_i)$.

В такому випадку справедлива формула ймовірності:

$$f_{\xi}(x) = \sum_{k=1}^m p_k f_k(x).$$

Формула Баєса:

$$p(w_i|x) = \frac{p_i f_i(x)}{f_{\xi}(x)}.$$

Перевагами використання статистичних моделей є те, що вони мають високу точність та прозорий процес прийняття рішення, у випадку необхідності їх можна легко модифікувати. Такий підхід використовує незначні ресурси, адже не виконує складних обчислень.

Основним недоліком є те, що такі моделі використовують вирішальні правила, які базуються на припущенні про нормальність розподілу, що в свою чергу призводить до зменшення вірогідності розпізнавання образів для негаусівських даних.

Висновки. В ході роботи було проаналізовано нейронні мережі, особливістю яких є здатність до навчання та статистичні моделі, які базуються на математичних правилах класифікації, що формуються і виводяться в термінах математичної статистики. Визначено, що вирішальні правила базуються на припущенні про нормальність розподілу, тому виникає необхідність розробки негаусівських ймовірнісних моделей для побудови вирішальних правил, які будуть використані для розпізнавання образів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Довбиш, А.С., Шелехов, І.В. (2015). *Основи теорії розпізнавання образів* (Ч. 1). Суми, Україна: СумДУ.
- [2] Редько, В.Г. (2010). *От моделей поведения к искусственному интеллекту*. Москва: КомКнига.
- [3] Лепский, А.Е., Броневиц, А.Г. (2009). *Математические методы распознавания образов*. Таганрог: ТТИ ЮФУ.

Trukhov A.S., Prykhodko S.B.

Analysis of existing mathematical models for pattern recognition

Abstract. The general algorithm of the image recognition process is considered during the research. Existing mathematical models for pattern recognition, namely neural networks and statistical models, are studied, their advantages and disadvantages are analyzed. The necessity of developing non-Gaussian probabilistic models for pattern recognition is substantiated.

Keywords: pattern recognition; mathematical models; non-Gaussian probabilistic models; neural networks; statistical methods.

УДК 004.412:519.237.5

НЕЛІНІЙНІ РЕГРЕСІЙНІ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ PHP-ФРЕЙМВОРКІВ

Приходько С.Б., д.т.н., професор¹, Приходько А.С.², Шутко І.С.³
^{1,2,3} Національний університет кораблебудування імені адмірала Макарова

^{1,2,3} Україна, Миколаїв

¹sergiy.prykhodko@nuos.edu.ua; ²whiterandrek@gmail.com; ³ishutko@gmail.com

Анотація. В роботі було побудовано нелінійні регресійні моделі для оцінювання кількості строк коду веб-застосунків, що створюються з використанням фреймворків CakePHP, Yii та Symfony.

Ключові слова: оцінка кількості строк коду програмного забезпечення; веб-застосунок; нелінійна регресійна модель.

Вступна частина. Оцінювання розміру веб-застосунків як і іншого програмного забезпечення (ПЗ) є важливою задачею на ранній стадії розробки програмного забезпечення, оскільки ця інформація використовується для прогнозування трудомісткості створення ПЗ за допомогою математичних моделей, наприклад, СОСОМО II. Зараз багато веб-застосунків створюються за допомогою різних фреймворків, які прискорюють розробку ПЗ. Найвідомішими з них є Symfony, Laravel, CakePHP, Yii.

На теперішній час відомі як лінійне регресійне рівняння [1], так і нелінійна регресійна модель [2] для оцінювання кількості строк коду інформаційних PHP-систем з відкритим кодом в залежності від трьох метрик концептуальної моделі даних у вигляді діаграми класів: загальна кількість класів, загальна кількість зв'язків та середня кількість атрибутів на клас. Але розмір веб-застосунків, що створюються з використанням різних фреймворків, може залежати від інших метрик [3]. Це потребує побудови відповідних моделей для оцінювання кількості строк коду веб-застосунків, що створюються з використанням PHP-фреймворків CakePHP, Yii та Symfony.

Метою дослідження є створення нелінійних регресійних моделей для оцінювання кількості строк коду веб-застосунків, що створюються з використанням фреймворків CakePHP, Yii та Symfony.

Основна частина. Розглянемо спочатку побудову нелінійної регресійної моделі для оцінювання кількості строк коду веб-застосунків, що створюються з використанням фреймворку CakePHP. Враховуючи на те, що CakePHP є PHP-фреймворком, метрики 38 веб-застосунків, які розміщені на сайті GitHub (<https://github.com>), були отримані за допомогою інструменту PhpMetrics (<https://phpmetrics.org/>). Ми брали такі метрики: розмір веб-застосунку Y у тисячах рядків коду; кількість класів X_1 ; середню кількість методів на клас X_2 ; глибину дерева спадкування X_3 . Ці метрики були обрані, тому що їх можна отримати з діаграми класів та між ними немає мультиколінеарності. Далі ми перевірили багатовимірні дані на нормальність за допомогою тесту Мардіа (Mardia's test), якій базується на багатовимірних асиметрії та ексцесі. Тест показав, що розподіл даних не є гаусівським. Тому для визначення наявності викидів у даних ми використовували тест на основі квадрату відстані Махаланобіса.

Використовуючи зібрані дані з із зазначених вище метрик веб-застосунків була побудована лінійна регресійна модель $Y = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon$, де $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ є оцінками параметрів. Причому змінна ε повинна бути випадковою величиною з розподілом Гаусу. Нульова гіпотеза про нормальність розподілу ε була перевірена за критерієм Пірсона. Ця гіпотеза була відкинута для рівня значимості 0,05. Те, що в розподіл ε не є гаусівським вказує на відсутність теоретичного обґрунтування застосування лінійної регресійної моделі та на необхідність побудови нелінійної регресійної моделі.

Нелінійна регресійна модель має вигляд $Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}$, де ε – гаусівська випадкова величина з нульовим математичним сподіванням та середньоквадратичним відхиленням, оцінка якого дорівнює 0,02976, $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ є оцінками параметрів, які мають відповідно такі значення: -1,8048, 0,9819, 1,1867 та 0,3198. Оцінки параметрів були отримані за допомогою методу найменших квадратів.

Створена нелінійна регресійна модель була перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, PRED(0,25). Допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Значення R^2 , MMRE і PRED(0,25) для створеної моделі дорівнюють відповідно 0,993, 0,107 і 0,929. Це свідчить про високу якість отриманої моделі.

Крім того були побудовані дві нелінійні регресійні моделі для оцінювання кількості строк коду веб-застосунків, що створюються з використанням фреймворків Yii та Symfony, які також мають добрі значення R^2 , MMRE і PRED(0,25).

Висновки. В роботі на основі десяткового логарифму було побудовано нелінійні регресійні моделі з трьома предикторами для оцінювання кількості строк коду веб-застосунків, що створюються за допомогою фреймворків CakePHP, Yii та Symfony. Створені моделі мають добру якість відповідного оцінювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Tan, H.B.K., Zhao, Y., Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models, *Software Engineering : the 28th International Conference (ICSE '06)* (pp. 321-330). Shanghai, China.
- [2] Prykhodko, N.V., Prykhodko, S.B. (2018). Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*, 6 (40), 101-110. DOI: 10.15407/emodel.40.06.101
- [3] Приходько, С.Б., Приходько, Н.В., Фаріонова, Т.А., Ворона, М.В. (2020). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом. *Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки»*, 31 (70), 124-131. DOI: 10.32838/2663-5941/2020.1-1/23

Prykhodko S.B., Prykhodko A.S., Shutko I.S.

Nonlinear regression models for estimating LOC of web applications created using PHP-frameworks

Abstract. In work three nonlinear regression models for estimating LOC of web applications created using the CakePHP, Yii, and Symfony frameworks have been created.

Keywords: LOC estimation; Web app; nonlinear regression model.

УДК 004.412:519.237.5

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ З КОДОМ НА РНР

Приходько С.Б., д.т.н., професор¹, Сербулов Г.В.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹sergiy.prykhodko@nuos.edu.ua; ²201250@nuos.edu.ua

Анотація. Удосконалено нелінійну регресійну модель для оцінювання розміру програмного забезпечення інформаційних систем з кодом на РНР. Ця модель має кращі значення показників R^2 , MMRE та PRED(0,25) у порівнянні з існуючими регресійними моделями, як лінійними, так і нелінійними.

Ключові слова: оцінювання розміру програмного забезпечення; інформаційна система; нелінійна регресійна модель; РНР.

Вступна частина. Задача оцінювання розміру програмного забезпечення (ПЗ) на ранній стадії його розробки є важливою, оскільки ця інформація використовується для прогнозування трудомісткості розробки ПЗ за допомогою такої відомої моделі як СОСОМО II (The Constructive Cost Model). Це призводить до необхідності розробки відповідних моделей та рівнянь для оцінювання розміру ПЗ, включаючи ПЗ інформаційних систем з кодом на РНР [1, 2].

У роботі [1] Hee Beng Kuan Tan, Yuan Zhao та Hongyu Zhang запропонували лінійні регресійні рівняння для оцінювання розміру (у тисячах строк коду) ПЗ інформаційних системи, що розробляються на базі певних мов програмування високого рівня, включаючи РНР. Запропоновані рівняння побудовано за допомогою множинного лінійного регресійного аналізу на основі метрик діаграми класів. Проте запропоноване в [1] лінійне регресійне рівняння для оцінювання розміру ПЗ інформаційних систем з відкритим кодом на РНР має незадовільні значення показників MMRE і PRED(0,25), які дорівнюють 0,4919 та 0,5313 відповідно. Тому в роботі [2] було побудовано нелінійне регресійне рівняння із застосуванням чотирирівневого перетворення Джонсона сім'ї S_B на основі тих же трьох метрик діаграми класів, що і в [1]: загальна кількість класів, загальна кількість зв'язків та середня кількість атрибутів на клас. Хоча запропоноване в [2] нелінійне регресійне рівняння для оцінювання розміру ПЗ інформаційних систем з відкритим кодом на РНР має добре значення MMRE – 0,2199, але значення PRED(0,25) складає 0,7188, що є меншим за 0,75. Тому виникає необхідність у подальшому удосконаленні нелінійного регресійного рівняння для оцінювання розміру ПЗ інформаційних систем з кодом на РНР.

Метою дослідження є удосконалення нелінійної регресійної моделі для оцінювання розміру ПЗ інформаційних систем з кодом на РНР.

Основна частина. Удосконалення нелінійної регресійної моделі для оцінювання розміру ПЗ інформаційних систем з кодом на РНР ми будемо за методом [3], що заснований на багатовимірних нормалізуючих перетвореннях та інтервалах передбачення. Суть цього методу [3] є такою. На першому етапі початкові дані перевіряються на наявність викидів і, якщо останні знайдено, то вони відкидаються. Для цього використовується критерій на основі квадрату відстані Махаланобіса для нормалізованих даних із 0,005 рівнем значущості. На другому етапі будується нелінійна регресійна модель із використанням відповідного

методу на основі нормалізуючих перетворень [4]. На третьому етапі визначаються границі інтервалу передбачення нелінійної регресії для рівня значущості, що дорівнює 0,05, за відповідним методом [4]. На останньому четвертому етапі перевіряють, чи є серед даних, за якими будувалася нелінійна регресійна модель такі, що виходять за границі інтервалу передбачення. Та, якщо відповідні дані знайдено, то вони відкидаються, і ми повторюємо знову всі етапи, починаючи з першого, для нових даних. Якщо таких викидів не було, то повторення етапів завершується, відповідна нелінійна регресійна модель побудована. Як і в [2-4], у якості багатовимірного нормалізуючого перетворення ми застосовуємо чотирьох-вимірне перетворення Джонсона для сімейства S_B .

Удосконалена нелінійна регресійна модель має таку ж саме структуру як і в [3, 4]. Ця модель була перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, PRED(0,25). Допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Значення R^2 , MMRE і PRED(0,25) для удосконаленої моделі дорівнюють відповідно 0,9885, 0,1725 і 0,7667, які є кращими за відповідні значення для лінійного та нелінійного регресійного рівнянь з [1] та [2]. Це свідчить про більш високу якість удосконаленої моделі у порівнянні з регресійними рівняннями з [1] та [2].

Висновки. На основі чотирьох-вимірного перетворення Джонсона сімейства S_B та видалення викидів у даних була удосконалена нелінійна регресійна модель з трьома предикторами для оцінювання розміру ПЗ інформаційних систем з кодом на PHP. Удосконалена модель має добру якість відповідного оцінювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Tan, H.B.K., Zhao, Y., Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models, *Software Engineering : the 28th International Conference (ICSE '06)* (pp. 321-330). Shanghai, China.
- [2] Prykhodko, S., Prykhodko, N., Makarova, L. (2019). Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis. *Proceedings of International Conference Advanced Computer Information Technologies (ACIT-2018). CEUR Workshop Proceedings, 2300*, 199-202.
- [3] Prykhodko, S., Prykhodko, N. (2021). Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations. In: Shkarlet S., Morozov A., Palagin A. (eds) *Mathematical Modeling and Simulation of Systems (MODS'2020)*. MODS 2020. Advances in Intelligent Systems and Computing, vol. 1265. (pp. 166-174). Springer, Cham. doi: https://doi.org/10.1007/978-3-030-58124-4_16
- [4] Prykhodko, N.V., Prykhodko, S.B. (2018). Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*, 6 (40), 101-110. doi: 10.15407/emodel.40.06.101

Prykhodko S.B., Serbulov G.V.

A nonlinear regression model for estimating a software size of PHP-based information systems

Abstract. A nonlinear regression model for estimating a software size of PHP-based information systems have been improved. This model, in comparison with other regression models both linear and nonlinear, has better values of metrics named R^2 , MMRE and PRED(0.25).

Keywords: software size estimation; information system; nonlinear regression model; PHP.

УДК 656.61

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПОПЕРЕДЖЕННЯ ЗІТКНЕННЯ МОРСЬКИХ СУДЕН У ЗАЛЕЖНОСТІ ВІД ОСОБЛИВОСТЕЙ ОТОЧУЮЧОГО СЕРЕДОВИЩА

Сорокунський О.Ю.¹; Терлич С.В., к.т.н.²; Терлич А.С.³

¹Херсонський морський коледж рибної промисловості;

²Національний університет кораблебудування імені адмірала Макарова;

³Гімназія Новокаховської міської ради

¹Херсон, Україна; ²Миколаїв, Україна; ³Нова Каховка, Україна

¹sorokunskiy3333@gmail.com; ²stterlych@ukr.net; ³slavuslimanskii@gmail.com

Анотація. Розроблено математичну модель запобігання зіткнення суден враховуючи фактори оточуючого середовища та можливості відхилення від Міжнародних правил попередження зіткнення суден (COLREG-72).

Ключові слова: зіткнення суден; зовнішнє середовище попередження; оптимізація; COLREG-72.

Вступ. Під час прийняття кращого рішення щодо попередження зіткнення суден судноводій, на якого лягає відповідальність за безпечного розходження, розглядає багато особливостей процесу попередження зіткнення суден під час транспортування вантажів морем:

- можливості свого та зустрічного судна для розходження;
- ризику та невизначеності стану судна та довкілля;
- можливості використання COLREG-72 [1], альтернативні дії, тощо.

Правилом 5 (Спостереження) [1] судноводій повинен «оцінити ситуацію та небезпечність зіткнення». Принципова складність цієї задачі оцінювання ситуації та вибору кращої дії у динамічній ситуації, яка відносно швидко змінюється при зустрічних курсах суден, полягає у неможливості апіорного визначення того, що назвати «кращим рішенням». Кожен судноводій вкладає свій зміст у це поняття. Більш того, невелике змінювання збудливої дії оточуючого середовища може призвести до змінення змісту «кращого рішення».

Метою дослідження є удосконалення математичного механізму прийняття рішень під час розходження морських суден. Для цього випадку прийняття рішення можна представити графічно, як показано на рис.1.

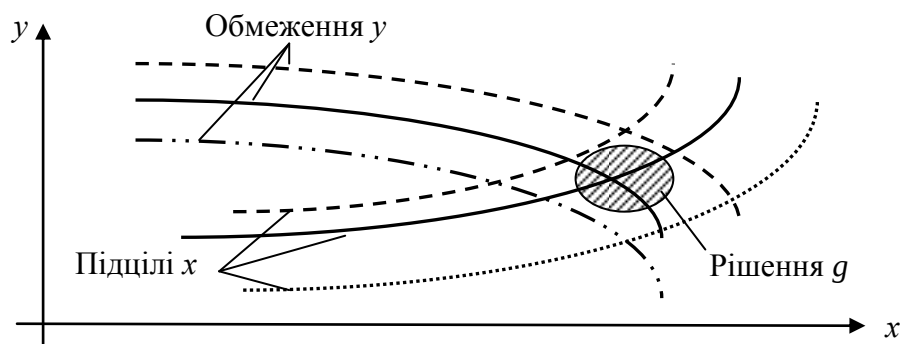


Рисунок 1 – Перетини та злиття підцілей та обмежень

Виходячи з рисунку 1 можна стверджувати, що

$$g \in (x \cap y), \quad (1)$$

де d – рішення; x – обмеження; y – підцілі процесу попередження зіткнення суден (ППЗТ).

На рис. 1 також видно, що чим більше факторів пред'явлено судноводію, тим область допустимих рішень стає більш розмитою. У визначені розпитою рішення (d) попередження зіткнення суден як перетин або злиття під цілей (x) та обмежень (y) як правило передбачає, що усі значення, які входять у (d) підцілі та обмеження мають у деякому сенсі однакову важливість. В той же час це не так. У деяких ситуаціях деякі підцілі та деякі обмеження становляться більш важливими, ніж інші і більш того за мірою зближень суден ці підцілі змінюються кількісно та якісно.

Основний текст. Поняття *кращого рішення* $\hat{g}$ залежить від досить великої кількості факторів, які практично неможливо врахувати у рамках фіксованої математичної моделі через особливість їх кількості, а також через неможливість математично описати різноманітних невизначеностей ППЗТ, які впливають на остаточне рішення [2].

Процес, який досліджується у статті схематично наведено на рис. 2. ППЗТ належить до рівня багаторівневих багатоцільових систем та складається із елементів прийняття рішення ($C_1, C_2, \dots, C_n$), які мають свої власні підцілі та (C_0) координатора. Отримання достатньо повного та точного оцінювання, навіть простої ситуації, вимагає врахування сукупності всіх підсистем $C = \{C_1; C_2; \dots, C_n\}$ у їх взаємозалежностях. Задачу прийняття кращого рішення серед під цілей ППЗТ треба шукати у ієрархічному підході [3]. Для цього спочатку розбивають сімейство які формують підцілі (рис. 2). Потім описують системи та їх призначення. Підсистема (C_1) – відповідає за генерацію правил для попередження зіткнення суден, які необхідні для забезпечення розходження суден на небезпечну відстань (D_i) компенсацією ситуаційного збурення на основі керованих та некерованих факторів складу судна та навколишнього середовища. Підсистема (C_2) – стан судна – ідентифікує експлуатаційні показники виконавчих пристроїв у відповідності із технічним регламентом та оперує факторами надійності, працездатності технічних засобів та механізмів судна. Підсистема (C_{n-1}) розв'язує задачі отримання максимального прибутку від транспортування вантажу, визначає його економічну ефективність та оперує комерційними факторами – коштовність, витрати, прибуток, ефективність тощо.

Для виконання безпечного розходження суден дії усіх підсистем повинні бути скоординованими. Ситуаційний оптимум ППЗТ формується на основі компромісу між усіма підцілями, так як рішення які вироблюють підсистеми, як правило не співпадають. У цьому випадку роль відводиться COLREG-72, тобто «командному елементу», який має пріоритет у випадку виникнення небезпеки зіткнення суден і може змінювати ієрархію підцілей. Таким чином, пророблення кращого рішення передбачає баланс взаємозв'язків:

складність оточуючого середовища $\rightarrow$ стан судна $\rightarrow D_i \rightarrow$ стратегія $\rightarrow$ краще рішення.

Перед керуванням Системою попередження зіткнення суден (СПЗС) слід розуміти спосіб переводу ситуації із небезпечного у безпечний стан за допомогою вектора стратегій [4]

$$Q_p^* = \{Q_1^*; Q_2^*; \dots; Q_n^*\}, \quad (2)$$

де p – ступінь небезпечної ситуації; Q_p^* – вектор стратегії управління СПЗС; $\{Q_1^*; Q_2^*; \dots; Q_n^*\}$ – вектори стратегії компенсації небезпечного ситуаційного збурення.

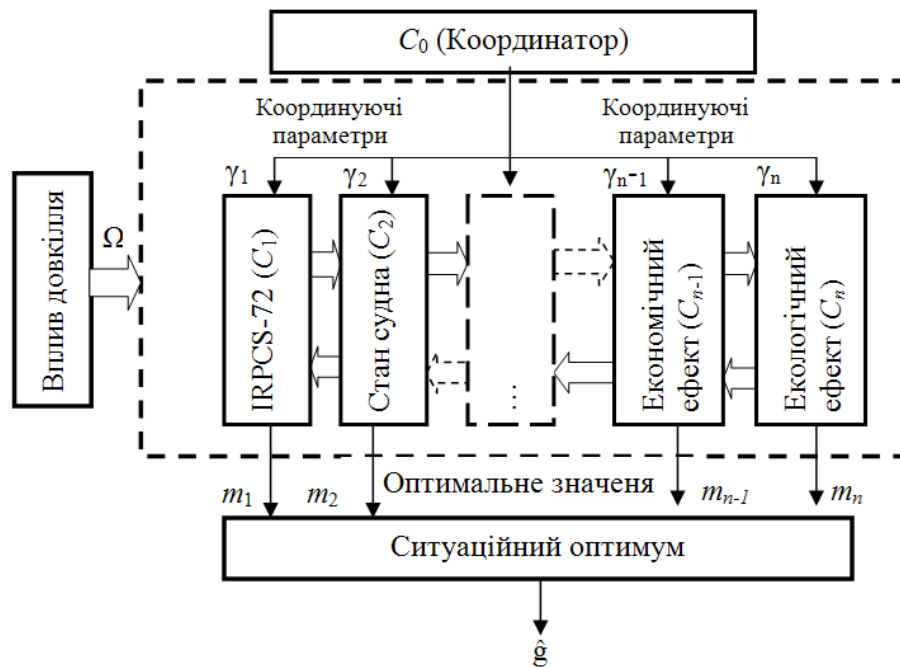


Рисунок 2 – Представлення моделі ППЗТ

Виходячи із цих параметрів, ступінь небезпеки оточуючої середовища (ρ) можна позиціонувати у одному із квадрантів матриці (табл. 1). Принципи координування СПЗС породжують сімейства стратегій маневрування у залежності від суперпозиції поставленої мети, яка в свою чергу у формі максимального ступеня рішення, яке треба досягти у вигляді кращого рішення – $\hat{g}$ у кожному квадранті табл. 1.

Квадрант 1 – характеризується низькою невизначеністю. В цьому випадку слід очікувати, що навколишнє середовище не зможе істотно впливати на рішення, які приймаються судноводієм.

Квадрант 2 – характеризується помірною невизначеністю і є вже більш напруженим для судноводіїв. Ступінь складності оточуючого середовища вводить елемент ризику у процес прийняття рішення.

Квадрант 3 – характеризується помірно високою невизначеністю, вимагає від судноводія достатньої «гнучкості» під час прийняття рішення (наприклад, Правило 17 COLREG-72).

Квадрант 4 – характеризується високою невизначеністю, уявляє собою найбільшу важкість під час управління та попередження зіткнення суден через те, що ситуація навкруги повна невідомих факторів, а для забезпечення безаварійного розходження суден необхідно відступити від вимог COLREG-72 згідно із *Правилом 2 (пункт b)*.

Таким чином, у процесах прийняття рішень можуть бути актуальні мінімум чотири ситуації у залежності від небезпеки оточуючого середовища та ступеня невизначеності. Виходячи із розглянутої схеми прийняття рішень в умовах невизначеності задача Координатора (рис. 2) складається з наступних пунктів:

- 1) формування множини рішень та множин стану оточуючого середовища;
- 2) визначення та формування складу задач основних показників якості ППЗТ, які входять до розрахунку функцій якості ППЗТ;
- 3) визначення критерію D_i у залежності від стану оточуючого середовища;
- 4) визначення стратегії поведінки судна $\{Q_1^* \vee Q_2^* \vee Q_3^* \vee Q_4^*\}$ у залежності від складу оточуючого середовища;
- 5) прийняття кращого рішення $\hat{g}(t)$ у залежності від обраної стратегії $\{Q_1^* \vee Q_2^* \vee Q_3^* \vee Q_4^*\}$.

Таблиця 1 – Знаходження кращого рішення у залежності від ступеня небезпеки оточуючого середовища p

Ступінь динамізму оточуючого	Низька	Ситуація низької невизначеності			Ситуація помірної невизначеності		
		Фактори	Стратегія D_1	Рішення $\widehat{g_1(t)}$	Фактори	Стратегія D_2	Рішення $\widehat{g_2(t)}$
		Малоподобні, незмінні			Істотно несхожі, не змінюються		
	Висока	Ситуація помірно-високої невизначеності			Ситуація високого ступеня невизначеності		
		Фактори	Стратегія D_3	Рішення $\widehat{g_3(t)}$	Фактори	Стратегія D_4	Рішення $\widehat{g_4(t)}$
		Малосхожі, постійно змінюються			Істотно несхожі, постійно змінюються		
	Низька				Висока		
	Ступінь складності оточуючого середовища						

У розгорнутій формі ситуація прийняття рішень характеризується матрицею, елементами f_{jk} якою є кількісні оцінки прийнятого рішення $m_k \in M$, за умови, що середа знаходиться у стані $\omega_j \in \Omega$:

$$\begin{array}{ccccccc}
 & m_1 & \dots & m_k & \dots & m_n & \\
 \omega_1 & f_{11} & \dots & f_{1k} & \dots & f_{1n} & \\
 \omega_2 & f_{21} & \dots & f_{2k} & \dots & f_{2n} & \\
 \omega_3 & f_{31} & \dots & f_{3k} & \dots & f_{3n} & \\
 \omega_4 & f_{41} & \dots & f_{4k} & \dots & f_{4n} &
 \end{array} \quad (3)$$

Прийняте рішення буде вважатися кращім, коли функція якості управління $F = \{f_{jk}\}$ намагається до максимізації досягнення D_i

$$F \rightarrow \max_{m_k \in M} \max_{D_i} f_{jk} . \quad (4)$$

Формально за ППЗТ у залежності від складності оточуючого середовища діють однакові найменування збурень, але різного ступеня впливу – $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Відповідно вплив цих збурень відображається на стратегії прийняття рішень також по різному. У відповідності із табл. 1 виконано декомпозицію «Ситуаційного оптимуму» (рис. 2) у залежності від ступеня небезпеки оточуючого середовища (рис. 3).

На рис. 3 множина можливих стратегій поведінки судна розбита на 4 підмножини $\{Q_1^* \vee Q_2^* \vee Q_3^* \vee Q_4^*\}$. Краще рішення $\hat{g}(t)$ може належати будь-якій із стратегій розходження суден у залежності від ступеня небезпеки оточуючого середовища.

Резюме. Формалізація процесу прийняття рішення щодо попередження зіткнення суден у залежності від ступеня небезпеки навколишнього середовища розбивається умовно на чотири підзадачі. Інтуїтивно зрозуміло, що така модель звужує параметри рішень, які приймаються із зменшенням «ступеня свободи» рішень ППЗТ, що означає підвищення ефективності самого рішення, яке прийнято та затраченого часу на його відпрацювання.

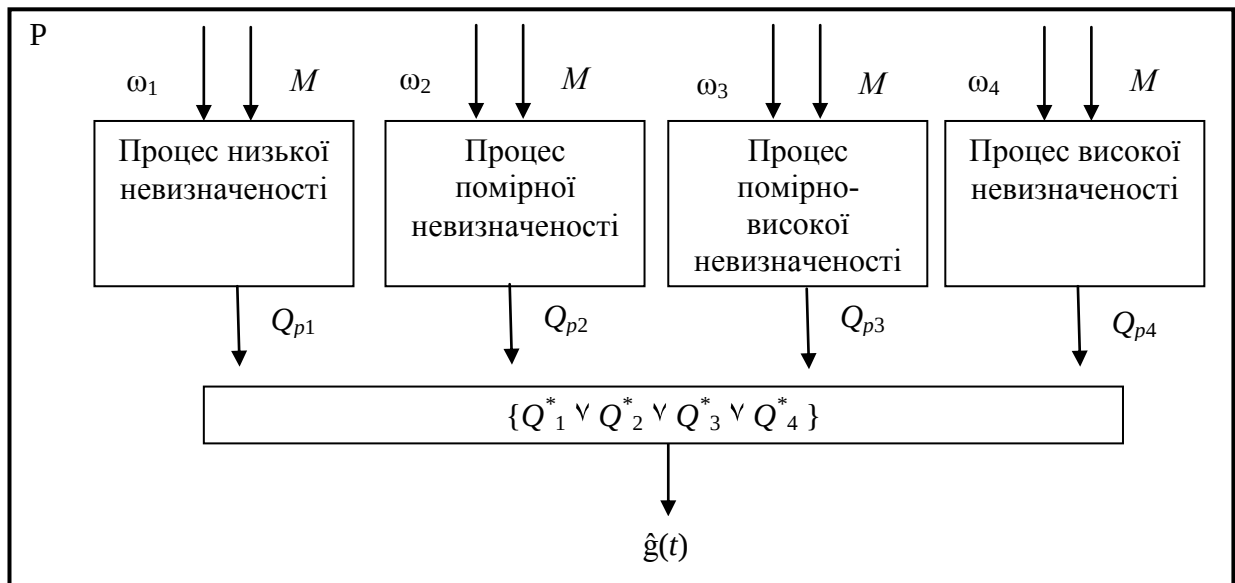


Рисунок 3 – Декомпозиція у залежності від ступеня оточуючого середовища

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Шарлай, Н.Г. (2017). *МППСС-72 с комментариями*. Москва: «МОРКНИГА», 137.
- [2] Лугінін, О.Є. (2007). *Статистика. Підручник. 2-е видання, перероблене та доповнене*. Київ: Центр учбової літератури, 608.
- [3] Заломихин, Д.В. (2020). Автоматизация и моделирование технических средств, элементов и методов судовождения. *Журнал «Транспортное дело»*, 5, 11-16.
- [4] Семушин, И.В. (2013). Ориентированная на фильтрацию Калмана математическая модель установившейся циркуляции для анализа траектории цели. *Автоматизация процессов управления*, 4 (34), 14-20.
- [5] Шапорев, С.Д. (2006). *Дискретная математика. Курс лекций и практических занятий*. СПб.: БХВ-Петербург, 400.

O. Sorokunsky, S. Terlych, F.Terlych

THE PROCESS FORMATION OF COLLISION VESSELS DEPENDING PREVENTION ON THE DEGREE OF THE ENVIRONMENT

Abstract. A mathematical model of ship collision prevention has been developed taking into account environmental factors and the possibility of deviations from the COLREG-72.

Keywords: collision of vessels; external warning environment; optimization; COLREG-72.

НЕЛІНІЙНІ РЕГРЕСІЙНІ РІВНЯННЯ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ РОЗРОБКИ JAVA-ЗАСТОСУНКІВ ДЛЯ ПЛАТФОРМИ MIDRANGE

Приходько С.Б., д.т.н., проф.¹; Фаріонова Т.А., к.т.н., доц.²

^{1, 2} Національний університет кораблебудування імені адмірала Макарова

^{1, 2} Україна, Миколаїв

¹ ORCID 0000-0002-2325-018X; ² ORCID 0000-0003-3384-4712

Анотація. Проведено оцінку існуючої моделі ISBSG для оцінювання часу розробки програмних проєктів в залежності від трудомісткості їх розробки для платформи midrange. Побудовано нелінійні регресійні рівняння тривалості розробки Java-застосунків для платформи midrange з урахуванням типів ПЗ. Також порівняно значення коефіцієнту детермінації R^2 , середньої величини відносної похибки MMRE та рівня прогнозування $PRED(0,25)$ для трьох побудованих нелінійних регресійних рівнянь.

Ключові слова: трудомісткість; тривалість розробки програмного забезпечення; нелінійна регресія; Java-застосунок; платформа midrange

Вступна частина. Midrange – клас комп'ютерних систем, який знаходиться між мейнфреймами і мікрокомп'ютерами. Це системи середнього рівня, в першу чергу високопродуктивні мережеві сервери та інші типи серверів, які можуть обробляти великомасштабну обробку багатьох бізнес-додатків. Вони не такі потужні, як мейнфрейми, однак їх покупка, експлуатація та обслуговування дешевше, ніж у мейнфреймів, і, таким чином, вони задовольняють обчислювальні потреби багатьох організацій. Midrange стали популярними як потужні мережеві сервери, які допомагають керувати великими веб-сайтами в Інтернеті, корпоративними інтрамережами, тощо. Сьогодні системи середнього рівня включають в себе сервери, що використовуються на промислових підприємствах і виробничих підприємствах, грають важливу роль в автоматизованому виробництві (CAM). Вони також можуть приймати форму потужних технічних робочих станцій для автоматизованого проектування (САПР) та інших обчислювальних і графічних додатків. Системи середнього рівня також використовуються в якості зовнішніх серверів для допомоги мейнфрейм-комп'ютерам в телекомунікаційній обробці та управлінні мережею [1].

Згідно з PMBOK та стандарту ISO/IEC 12207 – Software life cycle processes – процес оцінювання тривалості робіт з розробки програмного забезпечення (ПЗ) є невід'ємною складовою інженерії вимірювання ПЗ [2]. За статистикою біля 30% програмних проєктів мають труднощі з реалізацією (не вкладаються в бюджет, виходять з порушенням часових обмежень, або неналежної якості). Програмні проєкти для платформи midrange характеризуються своїм значним розміром, великою кількістю компонентів та спеціальними вимогами до надійності. Тому достовірне оцінювання тривалості розробки програмних проєктів для цієї платформи є актуальним завданням.

За статистичними даними компанії ISBSG (International Software Benchmarking Standards Group) (<https://www.isbsg.org/about-isbsg>), частка ПЗ для платформи midrange, основною мовою розробки якого є Java, складає приблизно 25%.

Метою роботи є підвищення достовірності оцінювання тривалості розробки Java-застосунків для платформи midrange за рахунок побудови нелінійних регресійних рівнянь.

Основна частина. Найбільш поширеними моделями, для оцінювання тривалості програмних проектів є COCOMO [3] і ISBSG [4]: нелінійні регресійні рівняння для оцінювання тривалості програмних проектів в залежності від трудомісткості. У зв'язку з тим, що емпіричні дані тривалості програмних проектів та трудомісткості програмних проектів мають закон розподілу, який відрізняється від гаусівського [3-5], для побудови таких моделей виконувалась нормалізація емпіричних даних з тривалості та трудомісткості розробки програмних проектів. Для побудови моделі ISBSG для платформи midrange, було використано десятковий логарифм для нормалізації емпіричних даних тривалості програмних проектів та емпіричних даних трудомісткості програмних проектів [4]:

$$D = 0,548 E^{0,360}. \quad (1)$$

Тут D –тривалість програмних проектів з розробки ПЗ (місяці), E – трудомісткість розробки програмних проектів для платформи midrange (люд.-год.). Однак, слід зазначити, що модель (1) була розроблена за даними проектів 1989-1996 рр.

Для оцінки моделі (1) було використано дані з 529 програмних проектів з репозитарію ISBSG та отримані такі результати її якості: $R^2=0,253$; $MMRE=0,708$; $PRED(0,25)=0,301$. Аналізуючи результати, можна зробити висновок, що модель (1) має невисоку якість.

Щоб вдосконалити модель ISBSG для оцінювання тривалості розробки програмних проектів для midrange (1) як додаткову ознаку впливу визначимо мову розробки ПЗ. А саме побудуємо нелінійне регресійне рівняння для оцінювання тривалості розробки Java-застосунків платформи midrange. В якості даних було використано 129 Java проектів з репозитарію ISBSG.

Емпіричні дані тривалості програмних проектів та трудомісткості мають закон розподілу, який відрізняється від гаусівського [4, 5]. Тому для побудови нелінійного регресійного рівняння авторами використані нормалізовані значення тривалості програмних проектів та нормалізовані значення трудомісткості. Нормалізацію випадкової величини виконуємо за перетворенням у вигляді десяткового логарифму. Це дозволить побудувати рівняння лінійної регресії для нормалізованих значень

$$\hat{z}_D(z_E) = b_0 + b_1 z_E \quad (2)$$

де z_D , z_E – нормалізовані випадкові величини D та E ; b_0 і b_1 – коефіцієнти рівняння лінійної регресії.

Після перевірки даних на викиди, коефіцієнти нелінійного регресійного рівняння для оцінювання тривалості розробки Java-застосунків платформи midrange мають наступні значення: $b_0 = -0,0778$, $b_1 = 0,273$. Побудоване рівняння нелінійної регресії тривалості програмних проектів для midrange має наступні характеристики: $R^2=0,3849$; $MMRE=0,2676$; $PRED(0,25)=0,57$. Таким чином, ми отримали кращі оцінки за параметрами R^2 , $MMRE$ та $PRED(0,25)$.

В наборі даних Java-застосунків платформи midrange, що використовувалися в розрахунках, були присутні як нові проекти (ND), так й такі, що вдосконалювалися (E). Тому авторами було запропоновано покращити оцінку за рахунок побудови окремо нелінійного рівняння регресії як для (ND), так й для (E). В табл. 1 наведені результати розрахунків.

З табл. 1 можна зробити висновок, що тип розробки (ND або E) має вплив на якість оцінки. Однак, необхідно проводити подальше дослідження факторів впливу з метою побудови покращеною нелінійної регресійної моделі для оцінювання тривалості розробки програмних проектів для midrange.

Таблиця 1 – Порівняльні результати розрахунків Java-застосунків платформи midrange

Показники моделі	New Development (ND), n=32	Enhancement (E), n=97
b_0	-0,74	-0,005
b_1	0,418	0,252
R^2	0,5916	0,3251
MMRE	0,3564	0,2979
PRED(0,25)	0,4	0,5974

Як видно з табл. 1, значення наведених параметрів кращі для нелінійного регресійного рівняння для проєктів, що вдосконалюються. Однак прийнятні значення MMRE та PRED(0,25) (не більше 0,25 та не менше 0,75 відповідно) для нелінійної регресії з використанням нормалізуючого перетворення у вигляді десяткового логарифму не досягнуті, що свідчить про необхідність застосування більш складних перетворень, наприклад, нормалізуючого перетворення Джонсона.

Висновки. Проведено оцінку існуючої моделі ISBSG для оцінювання часу розробки програмних проєктів в залежності від трудомісткості їх розробки для платформи midrange. Побудовано нелінійні регресійні рівняння тривалості розробки Java-застосунків для платформи midrange з урахуванням типів ПЗ. Порівняно значення оцінок R^2 , MMRE та PRED(0,25) для трьох побудованих нелінійних регресійних рівнянь. Перспектива подальших досліджень полягає у побудові нелінійних регресійних моделей із врахуванням впливу додаткових чинників, таких як сфера використання програмного продукту (банківська справа, виробництво, телекомунікації тощо), функціональний розмір тощо. Слід також дослідити використання нелінійних моделей, включення інших змінних у процес моделювання та використання інших видів нормалізуючих перетворень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Компьютер среднего уровня – Midrange computer. (2021). Retrieved from: https://star-wiki.ru/wiki/Midrange_computer.
- [2] Лавріщева, К.М. (2008). *Програмна інженерія*. Київ: Академперіодика
- [3] Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, NJ : Prentice Hall.
- [4] Oligny, S., Bourque, P., Abran, A., Fournie, B. (2000). Exploring the relation between effort and duration in software engineering projects. *In proc. of the World Computer Congress*, 175-178.
- [5] Приходько С.Б., Пухалевич А.В. (2012). Розробка нелінійної регресійної моделі тривалості програмних проєктів на основі нормалізуючого перетворення Джонсона. *Радіoeлектронні і комп'ютерні системи*, 4 (56), 90-93.

Prykhodko S.B., Farionova T.A.

Nonlinear regression equations for estimating the duration of Java-applications development for the midrange platform

Abstract. An evaluation of the existing ISBSG model for estimating the time of software project development depending on the effort of their development for the midrange platform. Nonlinear regression equations of duration of development of Java-applications for the midrange platform taking into account software types are constructed. The values of the coefficient of determination R^2 , the average value of the relative error MMRE and the prediction level PRED (0.25) for the three constructed nonlinear regression equations were also compared.

Keywords: effort; duration of software development; nonlinear regression; Java application; midrange platform.

УДК 005.8:004.94

SOFTWARE APPLICATION FOR BUILDING A REGRESSION MODEL OF ESTIMATING THE PHP WEB APPLICATIONS EFFORT

Ponomarenko T., candidate of technical science, PhD¹; Gelenko Y.²; Kotula V.³

^{1,2,3} National University of shipbuilding named by admiral Makarov

^{1,2,3} Ukraine, Mykolaiv

¹ tetyana.ponomarenko@nuos.edu.ua; ² urahelenko@gmail.com; ³ south-agrosnab@ukr.net

Abstract. *The article presents a description of software for building a multifactor regression model to estimate the effort of developing web applications in PHP.*

Keywords: *multifactor regression model; web applications in PHP; development complexity.*

Introduction. Regression analysis is an incredibly powerful machine learning tool used for analyzing data. It is a way of predicting future happenings between a dependent (target) and one or more independent variables (predictor). Regression differs from classification models because it estimates a numerical value, whereas classification models identify which category an observation belongs to. The main uses of regression analysis are forecasting, time series modeling and finding the cause and effect relationship between variables. It is well known that model will be more accurate if we will take into account the specificity of the data to construct the predictive models. As we see nowadays, field of php projects of github is huge - about 750 thousand projects, but each framework, for example Yii1 (420 projects) and Yii3 (223 projects), have its own circumstances, so models which will take into account them will bring more profit in future by the opinion of the authors. In this case datasets should be separated for each of the frameworks. This article deals with issues related with regression models and software creation for its implementation [1, 2, 3].

The aim of the work is to create software for building a regression model to estimate PHP-based projects effort, that it is possible to increase the reliability of the assessment of the dependent variable regression compared to using single-dimensional normalization transformations.

Background. Estimation of unknown parameters of the model is carried out solely based on the values metrics of the observations which we have got from GitHub PHP projects. Every tool from the list [4, 5] could be got for the observation extraction. All values are saved in a file that will be uploaded as in-data for the model construction. In Python for this case we should use Pandas, Matplotlib, Statsmodels, Numpy and Sklearn libraries. Lets import them:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.linear_model import LinearRegression
```

Multicollinearity occurs when there are two or more independent variables in a multiple regression model, which have a high correlation among themselves. When some features are highly correlated, we might have difficulty in distinguishing between their individual effects on the dependent variable. Multicollinearity can be detected using various techniques, one such technique being the Variance Inflation Factor(VIF). We will import VIF function from statsmodels.stats.outlier. Generally, a VIF above 5 indicates a high multicollinearity [2].

Then we should read the data from the file with observations that we have got from GitHub.

```
from google.colab import files
data = files.upload ()
plt.scatter(data.bugfixsize)
plt.figure(figsize=(10, 6))
plt.hist(data['effort'], bins=50, es='black', color='#2196f3')
plt.xlabel('bugfixsize')
plt.ylabel('pages_of_documentation')
plt.show()
regr = LinearRegression()
regr.fit(X_train, y_train)
print('r-squared:', regr.score (X_train,y_train))
print('Intersept:', regr.intersept_)
pd.DataFrame(data=regr.coef_,index=X_train.columns, columns = 'coef')
pd.DaraFrame({'coef_name':X_incl_const.columns,'vif':np.around(vif, 2)})
vif = [variance_inflation_factor(exog=X_incl_const.values,exog_inx=1) for i in range
(X_incl_const.shape[1])]
```

The non-linear regression model for estimating the effort of PHP projects which we have got from that software is $Y=2.341231+1.6159 x_1 +0.01438x_2$, where x_1 -size in bugfix in bugs, x_2 - size of documentation in pages. R-square 0.96.

Conclusions. Software of the model, built on the basis of multi-dimensional Johnson's family SB normalization transformation, compared to other regressive the models have a higher percentage of projected results, lower average values relative error and range of the interrelation of non-linear regression.

REFERENCES

- [1] Приходько, С.Б., Приходько, Н.В., Фаріонова, Т.А., Ворона, М.В. (2020). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*. 31 (70) № 1, 124-131. DOI: 10.32838/2663-5941/2020.1-1/23.
- [2] Приходько, С.Б., Приходько, Н.В., Ворона, М.В., Беловол, І.О. (2021). Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*, 2021.1.
- [3] Lépin , J.-F. (n.d.). PhpMetrics: Static analyzys for php. of Php. (2021). Retrieved from <https://www.phpmetrics.org/documentation/index.html>.
- [4] Визуализация качества кода с PhpMetrics. (2021). Retrieved from <https://habr.com/ru/post/254941/>.
- [5] PHP parser alternatives - PHP code analysis. LibHunt. (2021). Retrieved from <https://php.libhunt.com/php-parser-alternatives>.

Пономаренко Т., Геленко Ю., Котула В.

ПРОГРАМНИЙ ЗАСТОСУНОК ДЛЯ ПОБУДОВИ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВЕБ-ДОДАТКІВ PHP

Анотація. У статті представлено опис програмного забезпечення для побудови багатофакторної регресійної моделі для оцінювання трудомісткості розробки веб застосунків на PHP.

Ключові слова: багатофакторна регресійна модель; веб застосунки на PHP; трудомісткість розробки.

УДК 004.412:519.237

МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ПЕРЕВІРКИ ВЗАЄМОЗВ'ЯЗКІВ МІЖ ЧАСТИНАМИ JAVA-ЗАСТОСУНКІВ ЗА МЕТРИКАМИ RFC ТА СВО

Приходько С. Б., д.т.н., проф.¹; Смикодуб Т. Г.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ sergiy.prykhodko@nuos.edu.ua; ² tetiana.smykodub@nuos.edu.ua

Анотація. В роботі у якості математичної моделі для перевірки взаємозв'язків між частинами Java-застосунків пропонується використовувати рівняння трансформованого еліпсу передбачення в залежності від метрик RFC та СВО. Зазначене рівняння побудовано за значеннями RFC та СВО з 46 Java-застосунків з відкритим кодом на основі двовимірного перетворення Джонсона сім'ї S_B .

Ключові слова: Java-застосунок; метрики програмного забезпечення; еліпс передбачення; нормалізуюче перетворення; викид.

На сьогодні задача оцінювання якості програмних застосунків (ПЗ) є важливою для ІТ-компаній, оскільки тісно пов'язана з витратами, що виникають в процесі життєвого циклу програмного забезпечення. Якість об'єктно-орієнтовного проектування (ООП) визначають за допомогою певних метрик, серед яких застосовують метрики Чидамбера та Кемерера (СК): вага методів на клас (WMC, Weighted Methods Per Class), глибина дерева наслідування (DIT, Depth of Inheritance Tree), кількість нащадків (NOC, Number of Children), зв'язаність між об'єктами (СВО, Coupling Between Objects), відгук класу (RFC, the Response for Class) та відсутність згуртованості в методах (LCOM, Lack of Cohesion in Methods) [1]. Для перевірки взаємозв'язків між частинами ПЗ або відносин між класами ПЗ в [1] запропоновано використовувати разом дві метрики: СВО та RFC. Але на практиці, як правило, ці метрики аналізуються окремо без їх кореляції, а прийнятні їх значення визначаються лише для класу. Так для класу, значення СВО між 1 та 5 вважається добрим, а рекомендоване значення RFC складає від 0 до 50. Тому виникає потреба у математичній моделі, яка би дозволяла визначати припустимі значення СВО та RFC з урахуванням їх кореляції для застосунку в цілому.

У якості такої математичної моделі для перевірки взаємозв'язків між частинами Java-застосунків пропонується використовувати рівняння трансформованого еліпсу передбачення в залежності від метрик RFC та СВО. Як відомо [2], еліпс передбачення використовують для визначення викидів у двовимірних гаусівських даних. Але двовимірний розподіл даних метрик RFC та СВО для Java-застосунків не є гаусівським. Так, ми перевірили двовимірні дані метрик RFC та СВО з 46 Java-застосунків з відкритим кодом, які розміщені на сайті GitHub (<https://github.com>), на нормальність за допомогою критерію Мардіа (Mardia's test), якій базується на багатовимірних асиметрії та ексцесі [3]. За цим критерієм двовимірний розподіл зазначених даних не є гаусівським. Тому ми використовуємо трансформований еліпс передбачення, побудова якого наведена в [4]. Трансформований еліпс передбачення будується наступним чином. Спочатку за допомогою взаємо-зворотного нормалізуючого перетворення двовимірні негаусівські дані нормалізують таким чином щоб розподіл нормалізованих даних був наближений до гаусівського. Далі для нормалізованих даних будують еліпс передбачення. І нарешті, за еліпсом передбачення для нормалізованих даних із використанням зворотного до нормалізуючого перетворення отримують трансформований

еліпс передбачення. Вихід точки негаусівських даних за трансформований еліпс передбачення буде вказувати на наявність викиду з певною довірчою ймовірністю, для якої цей еліпс було побудовано. При цьому значення довірчої ймовірності ми пропонуємо брати 0,995. Це значення зазвичай обирають при визначенні викидів у багатовимірних даних [5, 6].

Для нормалізації двовимірних даних метрик RFC та CBO з 46 Java-застосунків з відкритим кодом ми використовували двовимірне перетворення Джонсона сім'ї S_B . На основі трансформованого еліпсу передбачення, рівняння для якого було побудовано за зазначеним перетворенням, виявлено два викиди за двома Java-застосунками, значення метрик RFC та CBO для яких є 785 та 132, 1729 та 397 відповідно. Відношення RFC до CBO для цих застосунків складає 5,947 та 4,355, а для інших 44 Java-застосунків це відношення знаходиться в межах від 0,908 до 2,212. Тому попередньо для Java-застосунків можна рекомендувати прийнятне відношення RFC до CBO в межах від 0,9 до 2,3. Але в подальшому цей діапазон потребує уточнення.

Висновки. У якості математичної моделі для перевірки взаємозв'язків між частинами Java-застосунків запропоновано використовувати рівняння трансформованого еліпсу передбачення, яке було побудоване на основі двовимірного перетворення Джонсона сім'ї S_B за даними метрик RFC та CBO з 46 Java-застосунків з відкритим кодом. Виконаний аналіз отриманих результатів показав, що для Java-застосунків можна рекомендувати прийнятне відношення RFC до CBO в межах від 0,9 до 2,3. В подальшому планується уточнити цей діапазон шляхом побудови інтервалу передбачення та використання додаткових даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Chidamber, S., Kemerer, C. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.
- [2] Prykhodko, S. B., Prykhodko, N. V., Makarova, L. M., Kudin, O. O., Smykodub, T. G., Prykhodko, A. S. (2017). *Detecting bivariate outliers on the basis of normalizing transformations for non-Gaussian data, Advanced Information Systems and Technologies: the 5th international scientific conference*, Sumy: Sumy State University, 95-97.
- [3] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 3(57), 519–530. doi:10.1093/biomet/57.3.519.
- [4] Prykhodko, S. B., Prykhodko, N. V., Kudin, O. O., Smykodub, T. G. (2017). Constructing the transformed prediction ellipses on the basis of normalizing transformations for bivariate non-Gaussian data. *Problems of information technology*, 1(021), 134-138.
- [5] Johnson, R. A., Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall.
- [6] Prykhodko, S., Prykhodko, N., Makarova, L., Pugachenko, K. (2017). *Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations, the First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, IEEE, 846–849. doi: 10.1109/UKRCON.2017.8100366.

Prykhodko S.B., Smykodub T.G.

The mathematical model for checking the relationships between parts of Java applications according to RFC and CBO metrics

Abstract. As a mathematical model to check the relationships between parts of Java applications, it is proposed to use the equation of the transformed prediction ellipse depending on the metrics of RFC and CBO. This equation is based on the two-variate Johnson transformation for the S_B family and the values of RFC and CBO metrics from 46 open-source Java applications.

Keywords: Java-application; software metric; prediction ellipse; normalizing transformation; outlier

УДК 004.415.2

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ПІДГОТОВЦІ ФАХІВЦІВ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Суслов С.В., к.т.н, доцент

Національний університет кораблебудування імені адмірала Макарова

Україна, Миколаїв

sergiy.suslov@nuos.edu.ua

Анотація. Запропоновано вдосконалення змісту навчального матеріалу при підготовці за спеціальністю «Програмна інженерія» з погляду на формування компетенції з формулювання вимог, проектування архітектури, моделювання процесів функціонування шляхом збалансованого поєднання розгляду певних методологій і загальних знань, що складають сутність багатьох методологій

Ключові слова: програмна інженерія; підготовка фахівців; методології розроблення; проектування.

Вступ. Інтенсивний розвиток ІТ галузі у світовому масштабі викликає потребу у все більшій кількості кваліфікованих фахівців. Нові робочі місця для розробників програмного забезпечення у різних країнах обчислюють сотнями тисяч при нестачі фахівців відповідної кваліфікації. Такий стан викликає підвищені вимоги до освіти з програмної інженерії з погляду на обсяги і зміст. У інженерії програмного забезпечення застосовують безліч методів і технологій розроблення зі сталою тенденцією до їх додаткової адаптації, що створює проблему для освіти. Змісту й обсягу знань, що мають отримувати випускники навчальних закладів за цією спеціальністю приділяють увагу провідні міжнародні об'єднання вчених й інженерів ІТ галузі ACM і IEEE, які при залученні відомих університетів з різних країн розробили таксономію знань з програмної інженерії SWEBOOK (Software engineering of body Knowledge) [1] і рекомендації з їх подання у навчальних закладах SE2014 [2]. Втім, обсяг матеріалу з проектування програмного забезпечення через обмеження часу, відведеного навчальною програмою, не відповідає реальному різноманіттю задач і методів їх розв'язання, з якими може зіткнутись студент після випуску з навчального закладу на майбутньому робочому місці.

Метою роботи є визначення оптимального змісту навчального матеріалу стосовно проектування програмного забезпечення з погляду на підготовку до різноманіття реальних задач розроблення.

Стандарти вищої освіти України (першого [3] і другого[4] рівнів) з інженерії програмного забезпечення частиною предметної області підготовки фахівців визначають методи аналізу і побудови моделей програмного забезпечення та методи проектування. Відповідні компетентності включають здатність проектувати структуру програмного забезпечення на її різних рівнях, включно з архітектурою, моделювати процеси функціонування окремих підсистем і модулів. Підготовка магістрів повинна надавати ще і здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.

Формування таких компетентностей передбачає знання сучасних і методів проектування, число яких на зараз є дуже великим, що спричинено різноманіттям видів самого програмного забезпечення і, відповідно, технологій його розроблення. Розвиток програмної інженерії шляхом зростання множини методів і технологій створює проблеми і у діяльності

фахівців галузі, серед яких, зокрема, брак консенсусу в порівнянні ефективності різних підходів. З погляду на це, група знаних практиків і науковців ініціювала рух з метою перетворення програмної інженерії у напрямку точної науки SEMAT (Software Engineering Method and Theory) [5].

Сучасний стан такого розвитку характеризує розробка теорії сутності програмної інженерії (Essence Theory of Software Engineering) [6]. Essence містить Ядро знань (Kernel) і Мову, які подані у специфікації [7]. Ядро структуровано по трьох областях уваги при розробленні програмного забезпечення: Клієнт, Рішення та Зусилля. У ньому визначені суттєві об'єкти, з якими виконують дії (так звані Alpha - аббревіатура Abstract-Level Progress Health Attribute), Області Діяльності – суттєві дії, які потрібно виконувати (так звані Activity Spaces) і Компетентності (Competencies) – представлення основних можливостей для виконання робіт програмної інженерії.

Таким чином у Ядрі визначено множина окремих, простих, ефективних сутностей, придатних до масштабування методів програмної інженерії, що можуть бути застосовані незалежно. Ядро надає загальний базис з практик розробки ПЗ, кожна з яких може бути визначена і застосована незалежно. Вони можуть бути об'єднані для створення конкретного методу програмної інженерії. Таке узагальнення зменшує обсяг детальної інформації потрібної для підготовки фахівця з програмної інженерії і тому в ряді університетів різних країн введені курси з Essence.

Досвід викладання такого курсу, що включав груповий проект, на бакалавраті у Норвезькому університеті науки і технології (NTNU) для більш ніж 540 студентів подано у статі [7]. У зроблених на цій основі висновках зазначено, що більшість студентів відчували труднощі у засвоєнні матеріалу з Essence. У той же час студенти відмічали полегшення за рахунок вивільнення від застосування призначених методів і можливості самостійного формування підходу до розв'язання завдань. Причинами зазначених труднощів автори вбачають недостатній досвід у студентів і брак навчального матеріалу.

Слід зазначити, що, дійсно, зміст інформації в Essence, її високий абстрактний рівень і форма подання розраховані на досвідчених фахівців, що володіють достатнім обсягом конкретних знань, які дозволяють розуміти значення абстракцій в Essence.

З погляду на наведене вище, виглядає доцільним стосовно викладання проектування програмного забезпечення застосувати комбінований підхід, що передбачає спочатку вивчення певних конкретних методів проектування і на їх базі розгляд узагальнень, пропонує у Essence. У переліку методів потрібні такі, що належать до різних методологій. Структурні методології можуть бути представлені методами Йордана-Де Марко і Гейна-Сарсона, як процедурно орієнтованими, та Джексона і Варнье-Орра, орієнтованими на дані. Об'єктно-орієнтовані методології доречно представити Рациональним уніфікованим процесом.

Висновки. У навчальний матеріал при викладанні проектування програмного забезпечення у сучасних умовах доцільно об'єднувати конкретні вживані зараз методи з узагальненим підходом до програмної інженерії, поданим Essence.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Bourque, P.and, R.E., Fairley, eds. (2014). Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society. Retrieved from www.swebok.org.

- [2] ACM, IEEE. (2015 February 23). Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Retrieved from <https://www.acm.org/binaries/content/assets/education/se2014.pdf>.
- [3] Міністерство освіти і науки України. (2018). Стандарт вищої освіти України першого рівня (ступінь бакалавра) галузь знань 12 – Інформаційні технології за спеціальністю 121 – Інженерія програмного забезпечення. <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzheneriya-programnogo-zabezpechennya-bakalavr.pdf>.
- [4] Міністерство освіти і науки України. (2020). Стандарт вищої освіти України другого рівня (ступінь магістра) галузь знань 12 – Інформаційні технології за спеціальністю 121 – Інженерія програмного забезпечення. https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2020/11/17/121_inzheneriya_prohramnoho_zabezpechennya_mahistr.doc.
- [5] Wikipedia contributors. (2021, August 14). SEMAT. In *Wikipedia, The Free Encyclopedia*. Retrieved 20:35, October 28, 2021, from <https://en.wikipedia.org/w/index.php?title=SEMAT&oldid=1038768931>.
- [6] Jacobson, I., Ng, P., McMahon, P. E., Spence, I., and Lidman, S.: (2012). The Essence of Software Engineering: The SEMAT Kernel. *ACMQueue*, 10, pp. 40-52.
- [7] Object Management Group. (2018). Essence – Kernel and Language for Software Engineering Methods. Version 1.2. Retrieved from <https://www.omg.org/spec/Essence/1.2/PDF>.
- [8] Kemell, KK., Nguyen-Duc, A., Wang, X., Risku, J., Abrahamsson, P. (2018). The Essence Theory of Software Engineering – Large-Scale Classroom Experiences from 450+ Software Engineering BSc Students. In Kuhrmann M. et al. (eds) *Product-Focused Software Process Improvement. PROFES 2018. Lecture Notes in Computer Science*, vol 11271. Springer, Cham. https://doi.org/10.1007/978-3-030-03673-7_9.

Serhii V. Suslov

DESIGNING OF SOFTWARE IN THE EDUCATION OF SOFTWARE ENGINEERS

Abstract. Propositions to enhance the content of education in the "Software Engineering" specialty are considered for the competences formation in software designing through a balanced combination of prominent methodologies and general knowledge, which is the essence of methodologies variety.

Keywords: Software Engineering; specialist education; development methodologies; designing.

УДК 004.9:620.92

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ОЦІНЮВАННЯ РОЗМІРУ TELEGRAM БОТІВ НА PYTHON

Шакула О.В.¹; Латанська Л.О., к.ф.-м.н., доцент²^{1,2} Національний університет кораблебудування імені адмірала Макарова^{1,2} Україна, Миколаїв¹ shakula13@gmail.com; ² llatanskaya@gmail.com

Анотація. На ресурсі GitHub зібрано емпіричні дані для побудови нелінійної регресійної моделі оцінювання розміру Telegram ботів на Python та виконано їх передобробку. Побудовано нелінійну регресійну модель для прогнозування розміру програмного забезпечення обраного типу та оцінено рівень довіри та прогнозу.

Ключові слова: передобробка даних, нелінійна регресійна модель; оцінювання розміру Telegram ботів на Python; рівень довіри та прогнозу.

Вступна частина. Розрахунок витрат – одне з найскладніших завдань в управлінні проектом. Процес оцінювання витрат на створення програмного забезпечення включає оцінку розміру програмного продукту, який розроблятиметься, оцінку необхідних зусиль і ресурсів, написання попередніх планів робіт і, нарешті, розрахунок повної вартості проекту.

Одним з основних показників, які впливають на вартість розробки, є розмір. Існує ряд методів та моделей для визначення розміру програмного забезпечення. Досить часто для прогнозування розміру використовуються регресійні моделі. Відомо, що розмір програми в значній мірі залежить від особливостей мови програмування, методології розробки, платформи розробки та інших факторів. Тому існуючі методи та моделі прогнозування розміру програмного продукту призначені для вузького класу задач. І тому для підвищення точності оцінювання розміру програмного забезпечення певного класу необхідно удосконалювати існуючі та розробляти нові методи та моделі, що робить задачу удосконалення регресійної моделі для оцінювання розміру Telegram ботів на Python актуальною [1].

Метою роботи є побудова нелінійної регресійної моделі для оцінювання розміру Telegram ботів на Python з використанням нормалізуючого перетворення десятковий логарифм.

Основна частина. Для досягнення поставленої мети було виконано наступну роботу:

– Зібрано емпіричні дані.

На ресурсі GitHub було зібрано інформацію по Telegram ботам на Python, а саме: кількість класів та кількість строк коду для 40 проектів.

– Виконано передобробку емпіричних даних.

За допомогою критерію згоди Пірсона χ^2 визначено, що емпіричні дані не являються нормально розподіленими. Тому виконано нормалізацію даних з використанням нормалізуючого перетворення десятковий логарифм.

З нормалізованих даних вилючено дублікати та викиди. При аналізі даних на викиди використано квадрат відстані Махаланобіса та допоміжну статистику, яку порівняно з критичним значенням Фішера $F_{k,N-k,\alpha}$ (N – кількість елементів у вибірці, k – кількість параметрів, α – рівень значущості). Ті елементи вибірки, для яких статистика перевищувала критичне значення Фішера, являлися викидами. Їх прибирали з вибірки і заново повторювали розрахунок. Було знайдено 2 викиди [2].

– Виконано специфікацію моделі, тобто обрано вид функціональної залежності.

В якості функціональної залежності обрано лінійну регресійну модель для очищених нормалізованих даних та нелінійну регресійну модель для вихідних даних.

– Виконано оцінювання параметрів моделі.

Побудовано лінійну регресійну модель для очищених нормалізованих даних та через зворотне перетворення отримано нелінійну регресійну модель для вихідних даних. Параметри лінійної регресії було визначено методом найменших квадратів [3].

– Виконано оцінювання рівня довіри до отриманої моделі

Побудовано довірчі інтервали для лінійної та, через зворотне перетворення, для нелінійної моделей.

– Виконано оцінювання прогнозних можливостей моделі.

Побудовано інтервали прогнозування для лінійної та нелінійної моделей. Обчислено коефіцієнт детермінації R^2 , середню величину відносної похибки MMRE та рівень прогнозування PRED(0,25) [4].

– Виконано порівняння лінійної моделі для вихідних даних у припущенні про нормальність їх розподілу та отриманої нелінійної моделі для вихідних даних по R^2 , MMRE та PRED(0,25).

По всім показникам побудована нелінійна модель є кращою, ніж лінійна, та значення R^2 , MMRE та PRED(0,25) знаходяться у допустимих межах (відповідно $>0,7$; $<0,3$ та $>0,7$). Таким чином, нелінійна модель більше підходить для прогнозування розміру Telegram ботів на Python.

Висновки. Побудовано нелінійну регресійну модель оцінювання розміру Telegram ботів на Python з використанням нормалізуючого перетворення десятковий логарифм. Виконано порівняння отриманої моделі з лінійною, побудованою у припущенні про нормальність розподілу емпіричних даних. В подальшому планується розробити програмне забезпечення для автоматизації процесу розрахунків за запропонованою нелінійною моделлю.

REFERENCES

- [1] Кантор, М. (2002). *Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения*. М.: Издательский дом «Вильямс».
- [2] Макарова, Л.М., Приходько, Н.В., Кудін, О.О. (2019). Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою Java. *Вісник Херсонського національного технічного університету*, 2 (69), 145-153.
- [3] Приходько, Н.В., Приходько, С.Б. (2018). Нелінійна регресійна модель для оцінювання розміру програмного забезпечення інформаційних систем на базі VB. *Інформаційні технології та комп'ютерна інженерія*, 3, 37-42.
- [4] Домбровский, В.В. (2004). *Эконометрика*. М.: Нов.учебник.

Oleksandr Shakula, Liudmyla Latanska

Nonlinear regression model for estimating the size of Telegram bots on Python

Abstract. Empirical data collected on the GitHub resource to build a nonlinear regression model for estimating the size of Telegram bots on Python and performed them preprocessing. A nonlinear regression model for predicting the size of the selected type of software is constructed and the level of trust and forecast is estimated.

Keywords: data preprocessing; nonlinear regression model; estimating the size of Telegram bots in Python; the level of trust and forecasting.

УДК 004.412:519:237.5

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРОС-ПЛАТФОРМНОЇ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ПЛАТФОРМИ XAMARIN

Макарова Л.М., к.т.н, доцент¹; Нікітін О.В.²^{1,2} Національний університет кораблебудування імені адмірала Макарова^{1,2} Україна, Миколаїв¹ lidia.makarova@nuos.edu.ua; ² 201247@nuos.edu.ua

Анотація. Побудована нелінійна регресійна модель для оцінювання розміру програмного забезпечення для крос-платформної розробки мобільних застосунків з використанням платформи Xamarin за рахунок використання одновимірного нормалізуючого перетворення Джонсона, що дозволило підвищити достовірність оцінювання розміру відповідного програмного забезпечення. Проведено порівняння отриманих результатів з лінійною моделлю.

Ключові слова: нелінійна регресійна модель; нормалізуюче перетворення Джонсона; розмір програмного забезпечення; Xamarin.

Вступна частина. Мобільні пристрої та технології – це атрибут життя сучасної людини та суспільства загалом. Банківські операції, покупки, спілкування, розваги – все базується на безперервному залученні людини у ці процеси за допомогою різних мобільних гаджетів. У свою чергу, подібні пристрої часто працюють під управлінням різних операційних систем і версій API, що погано позначається на сумісності програм і переносності (портативності) коду.

Платформа Xamarin є фреймворком для крос-платформної розробки мобільних додатків [1], що заснований на платформі з відкритим вихідним кодом Mono [2], яка, у свою чергу, сумісна з .Net фреймворком [3]. Xamarin дозволяє розробляти програми так, що написаний код буде здебільшого переносним (портативним) між такими платформами, як Android та iOS, не кажучи вже про сумісність з версіями різних систем і бібліотек, таких як OpenGL.

Xamarin продовжує розроблятися та доповнюватись, у тому числі різноманітними компонентами сторонніх розробників, не пов'язаними безпосередньо з Microsoft. Фреймворки та їх компоненти – це досить великі і складні проекти, і в цьому ключі, при розробці нового програмного продукту, який буде частиною Xamarin або компонентом на його основі, не погано було б мати метод оцінювання його розміру на ранній стадії проектування, що пов'язано з прогнозуванням собівартості продукту та трудовитрат на його розробку.

У цьому дослідженні з метою оцінювання розміру програмного забезпечення було обрано метод побудови нелінійної регресійної моделі, оскільки ця модель має високу точність прогнозування в порівнянні з лінійною регресійною моделлю.

Мета дослідження – побудова нелінійної регресійної моделі для оцінювання розміру програмного забезпечення для крос-платформної розробки мобільних застосунків з використанням платформи Xamarin.

Основна частина. Дані для побудови нелінійної регресійної моделі були знайдені на інтернет-ресурсі GitHub [4], де були вибрані проекти з відкритим вихідним кодом, що є частиною платформи Xamarin, яка використовується для крос-платформної розробки застосунків для мобільних операційних систем таких, як Android та iOS, а також проекти, які

є компонентами, розробленими на основі цього фреймворку. Було відібрано відносно великі проекти (від 1000 рядків коду і більше). Для кожного проекту було зібрано метрики програмного коду, такі як LOC (кількість рядків коду програми) – ця метрика була обрана як залежна змінна, та NC (загальна кількість класів) – ця метрика була обрана як незалежна змінна.

В результаті перевірки зібраних даних на нормальність розподілу з використанням критерію χ^2 Пірсона [5] з'ясувалося, що дані не підпорядковуються закону Гауса.

Для нормалізації даних було використано одновимірне нормалізуюче перетворення Джонсона сім'ї S_B [6], відповідно до значень асиметрії та ексцесу. Далі було проведено пошук та видалення викидів з використанням відстані Махаланобіса [7]. На наступному етапі, для нормалізованих даних була побудована лінійна регресійна модель з довірчим інтервалом і інтервалом передбачення.

На останньому етапі, використовуючи зворотне нормалізуюче перетворення, було побудовано нелінійну регресійну модель. Для перевірки якості отриманої моделі були розраховані характеристики, такі як R^2 , $MMRE$, $PRED(0.25)$. Розраховані значення знаходяться у припустимих межах ($MMRE \leq 0,25$; $PRED(0,25) \geq 0,75$; $R^2 \geq 0,7$).

Висновки. Використовуючи одновимірне нормалізуюче перетворення Джонсона сім'ї S_B , була побудована нелінійна регресійна модель для оцінювання розміру програмного забезпечення для крос-платформної розробки мобільних застосунків з використанням платформи Xamarin. Також передбачається розробка відповідної програми для автоматизації розрахунків та побудови нелінійної регресійної моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Xamarin: Open-source mobile app platform for .NET. (2021). Retrieved from <https://dotnet.microsoft.com/apps/xamarin>.
- [2] Mono. (2021). Retrieved from <https://www.mono-project.com//>
- [3] .NET: Free. Cross-platform. Open Source. (2021). Retrieved from <https://dotnet.microsoft.com/>.
- [4] GitHub. (2021). Retrieved from <https://github.com/>.
- [5] Kendall, M. G., & Stuart, A. (1967). *The Advanced theory of statistics* (Vol. 2). London: C. Griffin &.
- [6] Prykhodko, S. B., & Pukhalevych, A. V. (2014). Developing nonlinear regression models of software project duration based on Johnson transformation. *Collection of Scientific Publications NUS*, 0(3). doi:10.15589/jnn20140312
- [7] Prykhodko, S., Prykhodko, N., Makarova, L., & Pugachenko, K. (2017). Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations. *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. doi:10.1109/ukrcon.2017.8100366

Makarova L.M., Nikitin O.V.

A non-linear regression model for software size estimation for cross-platform development of mobile applications using Xamarin platform

Abstract. A non-linear regression model for software size estimation for cross-platform development of mobile applications using Xamarin platform was constructed using the univariate Johnson's normalizing transformation, which enabled to increase the reliability of the software size estimation. The obtained results were compared with the linear regression model.

Keywords: non-linear regression model; Johnson's normalizing transformation; software size; Xamarin.

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ CODEIGNITER

Сторчеус О. В.¹; Устенко І. В., к.т.н., доц., доц. каф. ПЗАС²

^{1,2}Національний університет кораблебудування імені адмірала Макарова

^{1,2}Україна, Миколаїв

¹olekiii639@gmail.com; ²ustenko.irina@gmail.com

Анотація: під час дослідження розроблено однофакторну нелінійну регресійну модель для оцінювання розміру web застосунків, що створюються з використанням фреймворку codeigniter, яка побудована на основі нормалізуючого перетворення за допомогою натурального логарифму для набору даних (розміру програмного забезпечення від кількості класів).

Ключові слова: регресійна модель; інтервал передбачення; оцінювання розміру; web-застосунок; нормалізуюче перетворення.

Нелінійна регресія - це вид регресійного аналізу, в якому експериментальні дані моделюються функцією, яка є нелінійною комбінацією параметрів моделі і залежить від однієї і більше незалежних змінних. Дані апроксимуються методом послідовних наближень.

Дані складаються з вільних від помилок незалежних змінних x і пов'язаних спостережуваних залежних змінних (відгуків). Кожна змінна у моделюється як випадкова величина із середнім значенням, що задається нелінійною функцією $f(x, \beta)$.

Регресія корисна для прогнозування відповіді на нові умови. Наразі все більше додатків та програмного забезпечення використовують багатофункціональні фреймворки. Таким чином, існує необхідність розробки однофакторної нелінійної регресійної моделі для оцінювання розміру web-застосунків.

Метою дослідження є оцінювання розміру web-застосунків та іншого програмного забезпечення на ранній стадії розробки, оскільки ця інформація використовується для прогнозування трудомісткості створення програмного забезпечення. У свою чергу зараз для створення web-застосунків широко використовують фреймворки, серед яких великою популярністю користується codeigniter – популярний MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-застосунків. Це потребує побудови відповідних моделей для оцінювання розміру web-застосунків, що створюються за допомогою фреймворку codeigniter. Як фактор візьмемо розмір програмного забезпечення від кількості класів.

При складанні математичних моделей може виявитися, що запропоноване рівняння лінійної регресії неадекватно описує досліджуваний процес або по виду поля кореляції видно, що лінія регресії криволінійної форми. У цьому випадку порядок рівняння, що описує процес, підвищується. Експоненціальне рівняння регресії має вигляд $y = a * \exp(bx)$. Оціночна модель регресії (побудоване за вибілковими даними) буде мати вигляд $y = a * \exp(bx) + \varepsilon$, де ε - випадкова помилка (відхилення).

Для розрахунку коефіцієнтів рівняння також як і в разі лінійної регресії можна застосувати метод найменших квадратів. При цьому математична модель підбирається так, щоб сума квадратів відхилень фактичних значень відгуку y від регресійних була мінімальна:

$S = \sum (y_i - \hat{y}_i)^2 \rightarrow \min$. Далі після складання системи нормальних рівнянь знайдемо коефіцієнт детермінації R^2 , що вказує, наскільки отримані спостереження підтверджують модель та визначається за формулою $R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$.

Потім знайдемо 2 інших коефіцієнти для повної оцінки моделі $MMRE = \frac{1}{n} \cdot \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$,
 $PRED(l) = \frac{k}{n}$.

Висновки: у ході даної роботи розробляється однофакторна нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку codeigniter. Параметри моделі потрібно оцінити за методом найменших квадратів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Tan, H.B.K., Zhao, Y., Zhang, H. (October, 2009). Conceptual data model-based software size estimation for information systems, *Transactions on Software Engineering and Methodology*, vol. 19, issue 2, article No. 4.
- [2] Приходько, С.Б. (2015). Моделювання гаусівських випадкових величин із використанням перетворення Джонсона із сім'ї SU. *Інформатика та математичні методи в моделюванні*, 1(5), 92-97.
- [3] Приходько, С.Б., Приходько, Н.В., Фаріонова, Т.А., Ворона, М.В. (2020). Трьохфакторна нелінійна регресійна модель для оцінювання розміру php-застосунків з відкритим кодом. *Науковий журнал Вчені записки Таврійського національного університету імені В. І. Вернадського*. Серія: Технічні науки, Том 31 (70), No 1, с. 124–131.
- [4] Chatterjee, S., Price, B. (2012). *Regression analysis by example*. New York USA: John Wiley & Son.

Storcheus O. V., Ustenko I. V.

IMPROVEMENT OF THE MATHEMATICAL MODEL FOR ESTIMATING THE SIZE OF WEB APPLICATIONS DEVELOPED USING THE CODEIGNITER AND WRITING A PROGRAM FOR IT'S REALISATION

Abstract: during the study, a one-factor nonlinear regression model was developed to estimate the size of web applications created using the codeigniter framework, which is based on a normalizing transformation using a natural logarithm for a data set (software size by number of classes).

Keywords: regression model; prediction interval; size estimation; web-application; normalization transformation.

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПЛАГІНІВ ДЛЯ WORDPRESS

Чичерін В.О.¹; Пухалевич А.В., к.т.н.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ vovaempire@gmail.com; ² andrii.pukhalevych@nuos.edu.ua

Анотація. Зібрано ООП метрики 128 плагінів для Wordpress на PHP. Побудовано трифакторну нелінійну регресійну модель для оцінювання розміру плагінів для системи управління вмістом Wordpress. Перевірено характеристики побудованої моделі.

Ключові слова: оцінювання розміру програмного забезпечення; Wordpress плагіни; нелінійна регресійна модель.

Вступна частина. При розробці програмного забезпечення, в тому числі плагінів для Wordpress – найпоширенішої системи управління вмістом веб-сайтів на PHP, дуже важливо мати оцінки розміру коду, так як вони потім використовуються для передбачення трудомісткості створення програмного забезпечення.

Для оцінювання розміру коду використовуються лінійні та нелінійні регресійні моделі [1] в залежності від трьох метрик: загальна кількість класів, загальна кількість зв'язків та середня кількість атрибутів на клас. Розмір плагінів, що створені для Wordpress, може також залежати від інших метрик та мати інші параметри [2]. Тому це потребує побудови математичних моделей для оцінювання розміру плагінів, що застосовуються для такої системи як Wordpress.

Метою дослідження є побудова нелінійної регресійної моделі для оцінювання розміру плагінів для системи управління вмістом Wordpress.

Основна частина. Так як плагіни для Wordpress написані мовою програмування PHP, ООП метрики 128 плагінів, розміщених на сайті Wordpress (<https://wordpress.org/plugins/>), були отримані за допомогою інструменту PhpMetrics (<https://phpmetrics.org/>). Було взято такі метрики: розмір плагіну Y у тисячах рядків коду; кількість класів X_1 ; середня кількість методів на клас X_2 ; глибина дерева спадкування X_3 . Вибір цих метрик пов'язаний з тим, що їх можна одержати з діаграми класів, а також тому, що між ними немає мультиколінеарності.

Отримані багатовимірні дані було перевірено на нормальність закону розподілу. Виявилось, що розподіл даних не являється гаусівським, і тому необхідно провести нормалізацію даних. Для цього було використано десятковий логарифм.

Після нормалізації дані було перевірено на наявність викидів за допомогою квадрату відстані Махаланобіса [3] та побудовано лінійну регресійну модель:

$$Y = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon,$$

де $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ являються оцінками параметрів, а ε - це випадкова помилка, яка повинна мати нормальний розподіл.

Було отримано наступні значення оцінок параметрів лінійної регресії: $\hat{b}_0 = 1,3663$; $\hat{b}_1 = 0,9989$; $\hat{b}_2 = 1,0229$; та $\hat{b}_3 = 0,0252$. Нормальність розподілу випадкової помилки ε була перевірена за критерієм Пірсона.

На наступному етапі було побудовано нелінійну регресійну модель. Побудована нелінійна регресійна модель була перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і PRED(0,25) – відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25. Допустимі значення R^2 , MMRE і PRED(0,25) складають не менше 0,7, не більше 0,25 і не менше 0,75 відповідно. Для побудованої моделі вони складають $R^2 = 0,92$; MMRE = 0,22; PRED(0,25) = 0,75. Це свідчить про середню якість отриманої моделі.

Висновки. В роботі, з використанням десятичного логарифму, було побудовано нелінійну регресійну модель з трьома факторами, для оцінювання розміру Wordpress плагінів. Побудована модель має достатні значення R^2 , MMRE і PRED, що свідчить про високу якість отриманої моделі. В подальшому планується створення програмного забезпечення, що реалізує побудовану модель для оцінювання розміру Wordpress плагінів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Tan, H.B.K., Zhao, Y., Zhang, H. (October, 2009). Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*, vol. 19, issue 2, article No. 4.
- [2] Tan, H.B.K., Zhao, Y., Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models. *Software Engineering : the 28th International Conference (ICSE '06)* (pp. 321-330). Shanghai, China.
- [3] Prykhodko, S., Prykhodko, N., Makarova, L., & Pugachenko, K. (2017). Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations. *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. doi:10.1109/ukrcon.2017.8100366

Chycherin V.O., Pukhalevych A.V.

THE NONLINEAR REGRESSION MODEL FOR ESTIMATING WORDPRESS PLUGINS SIZE

Abstract. OOP metrics for 128 Wordpress plugins are collected. A three-factor nonlinear regression model for estimating plugin size for the Wordpress content management system is built. The characteristics of the built model are checked.

Keywords: software size estimation; Wordpress plugins; nonlinear regression model.



УПРАВЛІННЯ ПРОЕКТАМИ І ПРОГРАМАМИ

УДК 37.018: 004.771

МОДЕЛІ ТА ТЕХНОЛОГІЇ СУПРОВОДУ ПРОЦЕСІВ ДИСТАНЦІЙНОГО НАВЧАННЯ

Яковлєва І.Р.¹; Чернов С.К. д.т.н., проф.²

Національний університет кораблебудування імені адмірала Макарова
Україна, Миколаїв

¹iryna.yakovlieva@nuos.edu.ua; ²19chk@gmail.com

Анотація. Підвищення рівня знань та умінь студентів відповідно до навчальних планів, спеціальності та курсу в різних напрямленнях та формах навчання.

Ключові слова: Дистанційне навчання; технології дистанційного навчання, структурно-функціональна модель дистанційного навчання.

Вступна частина. Наслідком бурхливого розвитку інформатизації суспільства, постійного зниження вартості послуг на використання глобальної мережі Internet та суттєве поглиблення процесів упровадження інформаційних технологій в освітню практику стала поява дистанційного навчання як найбільш перспективної, гуманістичної, інтегральної і орієнтованої на індивідуалізацію форми освітнього процесу [1].

Дистанційне навчання представляє собою нову організацію освіти, що ґрунтується на використанні як кращих традиційних методів отримання знань, так і нових інформаційних та телекомунікаційних технологій, а також на принципах самоосвіти. Дає змогу впроваджувати інтерактивні технології викладання матеріалу, здобувати повноцінну вищу освіту або підвищувати кваліфікацію і має такі переваги, як гнучкість, актуальність, зручність, модульність, економічна ефективність, інтерактивність, відсутність географічних кордонів для здобуття освіти.

Гнучкість дистанційного навчання полягає у можливості викладання матеріалу курсу з урахуванням підготовки та здібностей студентів. Це досягається створенням альтернативних сайтів для одержання більш детальної або додаткової інформації із складних тем, або низки питань-підказок.

При дистанційному навчанні змінюється роль і вимоги до викладачів. Лекції складають лише невелику частку учбового процесу. Основне завдання – зорієнтувати студентів на творчий пошук інформації, уміння самостійно набувати необхідні знання і застосовувати їх у вирішенні практичних завдань з використанням сучасних технологій. Викладачі дистанційних курсів повинні мати універсальну підготовку – володіти сучасними педагогічними та інформаційними технологіями, бути психологічно готовими до роботи зі студентами у новому навчально-пізнавальному середовищі. Завдяки таким засобам дистанційного навчання, як дискусійні форуми, електронні обговорення засвоєного матеріалу створюється нове навчальне середовище, в якому студенти почувають себе невід’ємною частиною колективу, що посилює мотивацію до навчання. У свою чергу, викладачі повинні володіти методами створення і підтримки такого навчального середовища, розробляти стратегії активної взаємодії між учасниками навчального процесу, підвищувати творчу активність студентів і власну кваліфікацію [2].

Широкий доступ студентів до освітніх ресурсів, не обмежені часом та відстанню можливості працювати з електронними варіантами програмно-методичного забезпечення по тим чи іншим дисциплінам формують відповідний рівень предметного діалогу «викладач-студент». У зв’язку із цим, новою функцією викладача є пошук діалогових комунікацій, що підвищують мотивацію навчання, сприяють розвитку пізнавальних інтересів студентів.

Для того, щоб навчання було максимально ефективним, його потрібно правильно організувати за допомогою системи обов'язкових організаційних заходів. По-перше, необхідно встановити рівень базових знань з дисципліни у кожного студента (вхідний контроль), а також виявити їх індивідуальні дані – інтереси, потреби, ціннісні орієнтації. По-друге, з урахуванням мети і завдань навчальної дисципліни, а також результатів аналізу інформації про ціннісні орієнтації студентів, що була отримана на попередньому організаційному етапі, сформувати для студентів пакет навчально-методичного забезпечення в електронному вигляді. По-третє, ознайомити студентів з наявними навчально-методичними матеріалами, надати рекомендації щодо організації та методики роботи з різними складовими цього пакету, а також конкретну інформацію про доступ до інших сайтів, зміст яких буде корисним при самостійній навчально-пізнавальній роботі. Крім того, необхідно довести до відома студентів заплановані контрольні заходи, пояснити, у якій формі вони будуть проводитись, надати критерії оцінки знань [3].

Актуальність. Дистанційне навчання проявляється у можливості упровадження новітніх педагогічних, психологічних і методологічних розробок з розбиттям матеріалу на окремі функціонально завершені модулі (теми), які вивчаються у міру засвоєння і відповідають здібностям окремого студента або групи загалом.

Основна частина. Педагогічні технології дистанційного навчання - це технології опосередкованого активного спілкування викладачів зі студентами з використанням телекомунікаційного зв'язку та методології індивідуальної роботи студентів з структурованим навчальним матеріалом, представленим у електронному вигляді.

Інформаційні технології дистанційного навчання - це технології створення, передачі та збереження навчальних матеріалів, організації і супроводу навчального процесу дистанційного навчання за допомогою телекомунікаційного зв'язку.

Мережна технологія – це використання засобів віддаленого доступу для одержання потрібної навчальної інформації, проведення контролю знань та індивідуальних консультацій з викладачами. Ця технологія значно ефективніша за традиційну тому, що вона орієнтується на індивідуалізацію навчання.

Технології дистанційного навчання можуть використовуватись не тільки в дистанційній освіті, а й в інших формах навчання: очній, заочній, екстернаті; крім того - в окремих дисциплінах або блоках дисциплін, що призначені для підвищення освітнього рівня чи кваліфікації окремих осіб та (або) груп слухачів.

Ефективна структурно-функціональна модель дистанційного навчання має наступні складові:

1. Цільовий який визначає мету навчального процесу.
2. Змістовий залежить від об'єкта навчального процесу – галузі суспільної діяльності, яку опановує реципієнт, та тих компетентностей, що формує викладач (також визначають навчальні програми конкретної дисципліни).
3. Діяльнісний що передбачає засвоєння знань, формування вмінь у процесі діяльності. Головним орієнтиром у дистанційній формі навчання має стати не максимальна кількість запропонованих для виконання завдань, а їхня якість, різноманітність та можливість вибору студентом.
4. Діагностично-оцінювальний. Це невід'ємний складник освітнього процесу. На цьому етапі відбувається контроль знань студентів, оцінювання, аналіз перебігу педагогічного процесу.
5. Соціально-емоційний. Основним недоліком дистанційного навчання є обмеження безпосереднього спілкування, взаємодії з іншими учасниками процесу [4].

Висновки. У ході виконання науково-дослідницької роботи були досліджені моделі дистанційного навчання, розглянуто та проаналізовано технології дистанційного навчання, застосування автоматизованих систем контролю знань. Дистанційне навчання – це перспективна форма організації навчання, у якій студент виступає активним суб'єктом навчання, що сприяє його самостійності, підвищенню навчальної мотивації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Андрєєв, А.А. (2013). Дидактичні основи дистанційного навчання. Взято з <https://uadoc.zavantag.com/text/3276/index-1.html?page=2>.
- [2] Побудова системи дистанційного навчання. (2013). Взято з https://otherreferats.allbest.ru/programming/00258547_0.html.
- [3] Системи дистанційного навчання. Переваги та недоліки. (2019). Взято з <https://studfile.net/preview/9437881/page:20/>
- [4] Стадній, А. (2020). Моделі дистанційного навчання. Взято з <https://op.ua/ru/pedclass/nauchnaya-statya/modeli-distanciynogo-navchannya>.

Yakovleva I.R., Chernov S.K.

Models and technologies for supporting distance learning processes

Abstract. *Improving the level of knowledge and skills of students in accordance with the curriculum, specialty and course in various areas and forms of education.*

Keywords: *Distance learning; distance learning technologies; structural and functional model of distance learning.*

УДК 005.8

ОСОБЛИВОСТІ ФОРМУВАННЯ ГІБРИДНОЇ КОМАНДИ ІТ-ПРОЕКТУ

Фараонов Ю.В., магістрант¹; Чубчик Т.Т., ст. викладач²

^{1,2}Національний університет кораблебудування імені адмірала Макарова

^{1,2}Україна, Миколаїв

¹19faraonov79@gmail.com; ² tetiana.chubchuk@nuos.edu.ua

Анотація. В роботі розглянуто основні елементи формування гібридної команди ІТ-проектів. Проведено аналіз недоліків та переваг кожної з 7 складових, на які необхідно звернути увагу для створення ефективної команди. Правильне поєднання саме цих семи елементів додасть силу і синергію команді, та забезпечить якісну розробку програмного забезпечення для великих складних проектів.

Ключові слова: ІТ-проект; гібридна команда; складні проекти.

Вступна частина. Інтернет технології здобувають все більшого значення в сучасному розвитку як економіки, так і суспільства в цілому. Про це свідчить динаміка кількості користувачів Інтернету відносно загальної кількості світового населення. Аналіз динаміки розвитку Інтернету створює умови для ведення бізнесу на основі побудови успішних платформ в Інтернеті шляхом розробки та впровадження Інтернет-проектів, основним завданням яких є бути корисними для користувачів та спрямованими на залучення максимальної аудиторії з можливістю давати комерційний прибуток [1].

Пандемія і криза змінили багато. У тому числі запити і звички покупців. Значно змінилася поведінка споживачів, виник новий клієнтський досвід, змінюються ланцюги доданої вартості. Ринки змінилися і під змінившися попит необхідно змінити або переконфігурувати продукти і рішення. Однак, продукти більшості компаній змінити не встигли. І одне з найважливіших питань - яким чином організувати роботу команди в нових умовах, щоб налаштуватися на нові потреби клієнтів. Тут допоможе підхід, названий «гібридні команди».

Основна частина. Термін гібридна команда до нинішньої пандемії не використовувалася широко. Якщо говорити просто, то спочатку гібридна команда - це команда, частина членів якої працює з дому, а частина - з одного або декількох офісів. І зрозуміло, що до 2020 року такі ситуації були рідкістю. Але більш важливо, що зараз виникає прагнення спробувати в гібридній команді об'єднати окремі елементи всіх інших типів команд, компенсувавши тим самим їх недоліки [2].

Гібридну команду можна назвати частково віртуальною командою, але вона не може бути віртуальною в чистому вигляді. Її можливо побудувати як частково міжфункціональну команду, що складається з фахівців різних відділів і вирішує складну задачу або проблему. Однак, при вирішенні складного завдання комунікації і обмін знаннями між експертами досить критичні і така команда вже не може управлятися так само, як і традиційна міжфункціональна команда. Більш того, не всі складні завдання можна буде доручити такій гібридній команді.

При формуванні ефективної гібридної команди ІТ – проектів експерти радять звертати увагу на 7 ключових моментів [3]:

Здібності і компетенції. Тут виділяється три ключових здатності членів команди: експертні знання та вміння їх приміряти (hard skills), вміння вирішувати проблеми, навички

ефективних комунікацій (soft skills). Знайти таких людей не завжди виходить навіть у випадку традиційних локальних і міжфункціональних команд. А в разі гібридних команд вимоги до комунікаційних навичок стають ще більш суворими, так як в разі віддаленої роботи саме комунікації провисають.

Гнучкість. Цей момент описує рухливість практик, норм і форматів роботи команди. Такі норми і практики частково визначаються керівництвом, частково складаються поступово шляхом самоорганізації команди. Практично у всіх, навіть самих традиційних командах можна спостерігати елементи самоорганізації. І члени команди повинні вміти підлаштовуватися як під норми що вводяться зверху, так і під норми, які складаються в процесі роботи. Це найважливіший момент і при формуванні команди, на старті, і необхідно враховувати. При формуванні гібридної команди цей момент ще більш критичний, так як всі ці норми і формати повинні контролюватися в тому числі і віддалено, а ймовірність їх порушення зростає. Тому члени гібридної команди повинні бути вдвічі більше гнучкими.

Неоднорідність. Цей момент дуже важливий для вирішення складних завдань. Складна система - це не система, що складається з великої кількості однакових елементів, в ній важлива різноманітність елементів. Аналогічно, для вирішення складного завдання, повинні залучатися команди, що складаються з різних співробітників, які дотримуються різних підходів, цінностей, шкіль мислення, різних за своїм експертним бекграундом. При цьому, виникає проблема обміну знаннями між експертами з різних бекграундів, яку необхідно вирішувати за рахунок зростання довіри та обміну знаннями. Не секрет, що експерти не завжди терпимі і деякі з них абсолютно не готові працювати з колегами протилежних поглядів. З цієї точки зору гібридна команда не відрізняється від традиційних. Однак, в гібридній команді, через віддаленість її членів виникають труднощі з комунікаціями і передачею знань. В результаті, підтримувати високу неоднорідність в гібридній команді істотно складніше, ніж в традиційній. Треба уважніше ставитися до терпимості членів команди, наскільки вони толерантні при роботі з іншими членами команди.

Особистісні якості. Не всі створені бути командними гравцями. Існує модель з п'яти якостей, які необхідні членам будь-якої команди: емоційна стійкість, добропорядність, чесність, товарищескість і екстраверсія. Відсутність будь-якого з цих якостей сильно підриває атмосферу довіри, яка повинна скластися в команді. У цьому моменті підбір і підтримка атмосфери довіри в гібридній команді практично нічим не відрізняється від таких же завдань у традиційній. Однак, є співробітники, яким дуже складно працювати в дистанційному режимі. Їх соціалізація дуже висока, а самотність і відсутність особистих комунікацій, практично вибивають їх з колії. І про це необхідно пам'ятати, організовуючи гібридну команду.

Ролі, необхідні у команді. Це традиційний і зрозумілий момент для будь-яких команд і гібридні команди мають тут трохи специфіки. Єдино, про що варто сказати - це інструменти контролю керівників або лідерів команди, які ускладнюються через віддаленість комунікацій.

Культура. Це момент, який враховує культурні особливості членів команди. Наприклад, європейська та американська культури більшою мірою побудовані на індивідуалізмі і схильні до оцінки індивідуальних результатів роботи. І далеко не всі люди готові відмовитися від культурних партнерів. Є азіатські культури, налаштовані на колективну і командну роботу. Українська культура в цьому сенсі знаходиться посередині і для більшості дослідників ще не до кінця зрозуміло.

Розмір команди. Дослідження показують, що для ефективного дистанційного управління необхідно зменшити оптимальний розмір команди. Якщо для традиційних команд оптимальна чисельність оцінювалася в 8 - 10 осіб, то для віртуальних і гібридних команд вона повинна бути зменшена до 5 - 6 чоловік.

Правильне поєднання саме цих семи складових і забезпечує силу і синергію команд.

Висновки. В роботі розглянуті основні особливості формування гібридних команд. Головна цінність таких команд - можливість колосально знизити або оптимізувати витрати, при цьому зберігаючи професіоналізм і примножуючи творчий потенціал людей, які працюють над продуктом. Якщо мета - надійний і багатоскладовий продукт, воно того варто. Такий підхід забезпечує якісну розробку програмного забезпечення для великих складних проєктів, це підтверджує досвід і наукові дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Ладонина, Л. Ю. (2008). *Книга руководителя интернет-проекта. Готовые маркетинговые решения*. СПб.: Питер.
- [2] Роббинз, С. П. (2005). *Основы организационного поведения*. Москва: Вильямс.
- [3] Шантаренкова, М. Зимин, К. (2021). Будущее за гибридными командами. Retrieved from <https://upr.ru/article/chto-takoe-gibridnaia-komanda/>.

Faraonov Yu.V., Chubchuk T.T.

Features of formation of hybrid team of IT-project

Abstract. *The paper considers the main elements of forming a hybrid team of IT projects. An analysis of the disadvantages and advantages of each of the 7 components, which must be addressed to create an effective team. The right combination of these seven elements will add strength and synergy to the team, and provide quality software development for large complex projects.*

Keywords: *IT – project; hybrid team; complex projects.*

УДК 005.8

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОЛОГІЄЮ AGILE

Біла А. К., магістрант

Національний університет кораблебудування імені адмірала Макарова

Україна, Миколаїв

nastya.belaya.917@gmail.com

Анотація. У статті ставиться завдання проаналізувати методологію Agile, її переваги та недоліки у проектах розробки програмного забезпечення. З метою розробки якісного та економічно ефективного програмного забезпечення компанії намагаються прийняти нові методології для цього. Використання Agile в індустрії розробки програмного забезпечення має свої переваги та недоліки.

Ключові слова: методології управління проектами; Agile-підхід; інновації; програмне забезпечення; проект.

Вступна частина. Актуальність теми дослідження полягає в тому, що не існує ідеальної системи управління проектами, що підходить для кожного з видів проектів. Однак за час існування проектного управління було створено чимало ефективних підходів, методів і стандартів, які можна ефективно використовувати. Agile-методика управління проектами вважається однією з найефективніших, її успішно застосовують менеджери проектів з різних країн світу. Безліч підприємств, які займаються розробкою програмного забезпечення намагаються здійснювати великі види діяльності з такими можливостями, як обмеження бюджету, скорочення робочої сили та зменшення операцій [1]. Такі результати обумовлюють необхідність оптимізації кожної процедури, ресурсу та діяльності з точки зору управління проектами. Останнім часом збільшилася потреба в переході від традиційного управління проектами до більш гнучкої методології, яка може дозволити досягнутих найкращих результатів за менший час та гарантує, що компанії отримуватимуть більший прибуток, використовуючи мінімальні ресурси на відміну від традиційних методів.

Основна частина. Agile це сукупність гнучких ітеративно-інкрементальних методів до управління проектами та продуктами. Відповідно до даного підходу, проект розбивається на маленькі підпроекти, які потім «збираються» в готовий продукт. Центальною рисою цього методу управління проектами є досягнення цінності для клієнтів, оскільки досягнення швидкої реалізації проектів відбувається завдяки співпраці невеликих спільно розташованих груп, що включає кінцевого користувача або споживача, бізнес-аналітиків, спеціалістів та керівника проекту [1].

За словами Тернера, Agile - це поступова, інтерактивна техніка управління, яка найкраще підходить до проектування інформаційних технологій, інженерії та інших сфер бізнесу, яка спрямована на надання нової послуги чи продукту у гнучкій формі [2]. Теорія Agile-методу передбачає зміни, додаткові можливості та вдосконалення, включені до циклу розробки продукту.

Впровадження Agile-методики у глобальній розробці програмного забезпечення має більше переваг, ніж недоліків. Такий підхід зосереджений головним чином на тісній співпраці між замовниками та розробниками програмного забезпечення.

При Agile-методології використовується парне програмування, тобто призначається двох різних експертів для однієї роботи, що допомагає підвищити якість результату та

роботи. Досвід багатьох компаній показує, що розробники повинні часто спілкуватися із клієнтом, щоб отримати зворотний зв'язок та зібрати всю інформацію для проекту. При цьому різний часовий пояс є основним недоліком, з яким стикаються розробники та клієнти. Щоб подолати таку проблему, розробники повинні мати належний зв'язок між собою та між замовником, адже часті зустрічі між ними розвивають дружні стосунки, а це позитивно впливає на хід проекту [3].

Сильні сторони Agile

Невеликі випуски	Допоможе зменшити недоліки для майбутнього програмного забезпечення, що розробляється.
Парне програмування	Підвищує якість продукту і прискорює результат.
Збір вимог	Постійний збір вимог на основі рішень замовника.
Відгуки клієнтів	Відгуки клієнтів під час кожного випуску зменшать кількість виправлень в кінці.

Крім того, Agile відмінно підходить для проектів з «відкритим кінцем» - наприклад, запуску сервісу або блогу. Як вже згадувалося раніше такий метод має деякі недоліки.

Найголовніші з них:

- Складність у ствердженні проектів, для яких не складені точні графіки, а також не визначено точний обсяг;
- важко складати довгострокові плани, коли не має інформації, чи будуть ресурси вільні для наступного проекту або задіяні в поточному Agile-проекті;
- складно оцінювати хід виконання через використання численних спринтів, які можуть налаштовуватися і виконуватися як окремі міні-проекти;
- виконання Agile-проектів часто забирає більше часу і сил, оскільки багато методологій вимагають проведення щоденних нарад і постійної взаємодії з клієнтом;
- тривалість і обсяг проекту можуть вийти з-під контролю через відсутність чітких меж, які не задаються спочатку.

Підсумовуючи все вище сказане, можна сказати, що Agile-методологія в сучасному світі та у IT-сфері була прийнята з великим успіхом. У сфері розробки програмного забезпечення немає непорушних основ і універсальних рішень. Кожному керівнику треба знайти для себе найкращий метод, який буде задовольняти всі його потреби. Не дивлячись на деякі недоліки, Agile має чудову ефективність та результативність. Багато фахівців кажуть про те, що Agile є більш успішною методологією порівняно з іншими методологіями у всіх аспектах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Highsmith, J., Cockburn, A. (2001). *Agile software development: The business of innovation*. *Computer*, 34(9), 120-127.
- [2] Nerur, S., Mahapatra, R., Mangalaraj, G. (2005). *Challenges of migrating to agile methodologies*. *Communications of the ACM*, 48(5), 72-78.
- [3] Turner, J.R. (1995). *The Commercial Project Manager*, McGraw-Hill, London, 408.

Bila A.

Software development according to Agile methodology

Abstract. *The article sets the task to analyze the Agile methodology, its advantages and disadvantages in software development projects. In order to develop high quality and cost-effective software, companies are trying to adopt new methodologies for this. The use of Agile in the software development industry has its advantages and disadvantages.*

Keywords: *project management methodologies; Agile methodology; innovation; software; project*

УДК 005.8

ЖОРСТКІ НАВИЧКИ (HARD SKILLS) ЯК ДЕТЕРМІНАНТА УСПІХУ РІШЕННЯ ЗАВДАНЬ В УПРАВЛІННІ ПРОЕКТАМИ

Білявська О. В., магістрант

Національний університет кораблебудування імені адмірала Макарова

Україна, Миколаїв

alexandra.belyavskaya@gmail.com

Анотація. У даній статті розглядається значення жорстких навичок як професійних інструментів досягнення цілей у сфері управління проектами. Описуються п'ять найнеобхідніших жорстких навичок, які сприяють ефективному вирішенню завдань проекту: управління контрактами та закупівлями, управління ризиками, моніторинг та оцінка ефективності проекту, складання бюджету, планування.

Ключові слова: жорсткі навички; планування; управління проектами; матеріальні ресурси; аналіз.

Вступна частина. Актуальність теми дослідження є беззаперечною, адже неможливо працювати над виконанням проекту не маючи хоча б елементарних технічних знань зі спеціальності, і з глобальним технічним прогресом, ця актуальність зростає в геометричній прогресії. Без розвитку жорстких навичок ніколи не буде особистісного росту, не говорячи вже про можливість професійного підвищення, яке передбачає вирішення складних завдань. З цього випливає **мета** дослідження- виявити важливість жорстких навичок як фактора досягнення максимального успіху у вирішенні професійних завдань.

Основна частина. Управління проектами - одна зі сфер управління бізнесом, де керівник проекту несе виключну відповідальність за ефективне ведення процесом проекту. Отже, керівнику проекту потрібні певні навички для його управління. Ці навички загалом згруповані в м'які і жорсткі. М'які навички відносяться до навичок міжособистісного спілкування і включають такі якості, як лідерство, спілкування, ведення переговорів, емпатія, креативність і таке подібне. Вони в значній мірі нематеріальні, їх важко виміряти і не пов'язані з конкретними показниками. М'які навички зазвичай використовуються без наявності інструментів або шаблонів [1]. Жорсткі ж навички є технічними аспектами ролі менеджера проекту і включають розробку матеріальних результатів, таких як графіки проекту, бюджети і структурні декомпозиції робіт. Жорсткі ж навички в управлінні проектами включають в себе управління бюджетом проекту, створення структур робіт, оцінку прибутку або складання діаграм критичних шляхів. Оскільки всі ці навички є специфічними, чітко визначеними та вимірюваними, їх вважають жорсткими.

Жорсткі навички часто передбачають використання шаблонів та інструментів, таких як програмне забезпечення, планування, електронні таблиці та засоби моделювання. Розвиток жорстких навичок - це навчання в школі та університеті або коледжі, онлайн-курси, майстер-класи, воркшопи, стажування, інтенсиви і професійні конференції- тобто усі бажаючі можуть легко здобувати необхідний для них рівень компетентності [2].

У чому полягає роль жорстких навичок? Професійні технічні навички безцінні як для співробітників, так і для роботодавців; це ті барометри, за допомогою яких лідер команди може оцінити загальний прогрес проекту. Без жорстких навичок та точного аналізу проект не

був би спроможний навіть розпочати своє існування. Здобуваючи необхідні технічні навички, співробітники можуть виконувати свою роботу більш ефективно і результативно. А наймаючи співробітників з добре розвиненими жорсткими навичками, роботодавці отримують вигоду у вигляді збільшення продуктивності і підвищення задоволеності споживачів, а отже- і більше проектних замовлень. Саме тому, під час працевлаштування у свою компанію, роботодавці звертають пильну увагу на жорсткі навички потенційних кандидатів [3].

Нижче наведені часто необхідні жорсткі навички управління проектами:

1. Управління контрактами та закупівлями

Менеджер проекту повинен бути обізнаний про дії, необхідні для забезпечення виконання контракту і відповідних законів. Він також повинен бути знайомий з процесом перетворення потреб бізнесу або проекту в товари і послуги, передані на аутсорсинг, і з тим, як вибирати кращих постачальників; повинен знати, як управляти зацікавленими сторонами в ланцюжку поставок.

2. Управління ризиками

Інститут управління проектами (PMI) визначає ризик проекту як «невизначену подія або умову, яка, якщо вона відбудеться, матиме позитивний або негативний вплив на цілі проекту». Управління ризиками включає ідентифікацію ризиків, аналіз ризиків, а також пріоритизацію і контроль ризиків. Він спрямований на моніторинг і управління ризиками для мінімізації або уникнення їх негативного впливу на досягнення цілей проекту. Менеджери проектів часто використовують Систему управління ризиками Національного інституту стандартів і технологій (NIST) для структурованого управління ризиками.

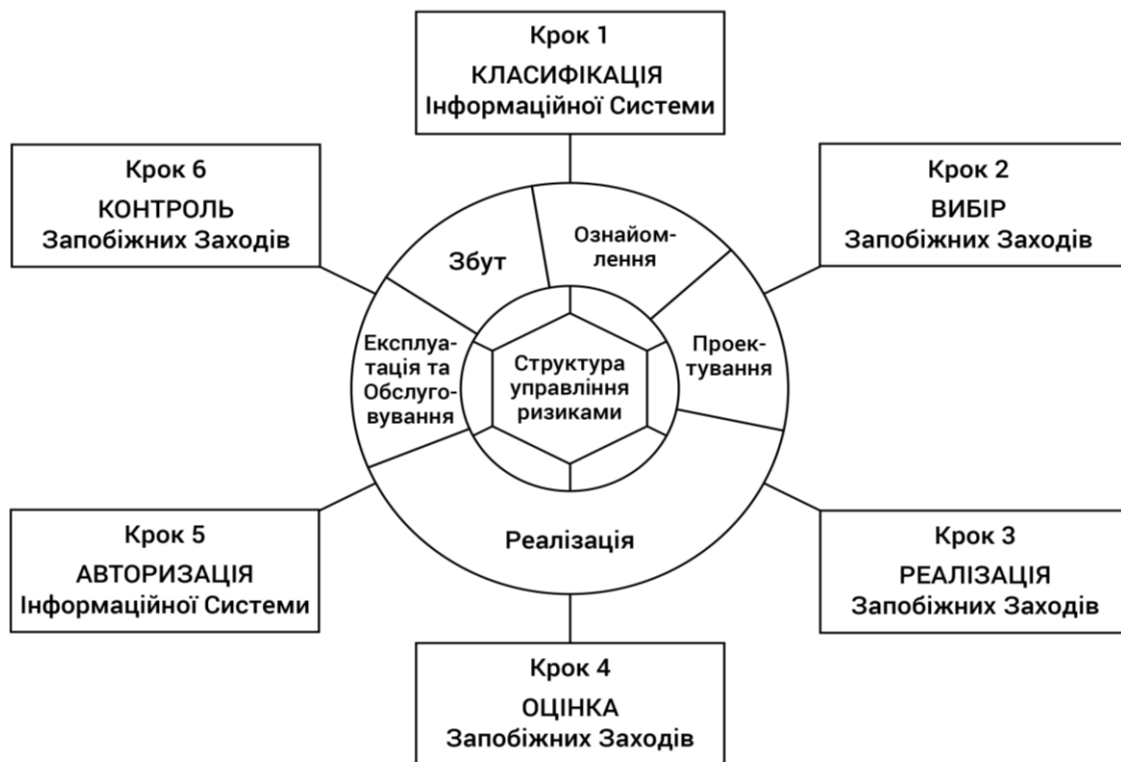


Рисунок 1 - Структура управління ризиками

3. Моніторинг та оцінка ефективності проекту.

Моніторинг та оцінка - це постійний аналіз реалізації проекту відповідно до погоджених графіків з метою поліпшення поточного і майбутнього управління результатами, наслідками і впливом. Він надає менеджерам та іншим зацікавленим сторонам постійний зворотний зв'язок щодо реалізації проекту і виявляє фактичні або потенційні успіхи і проблеми якомога раніше, щоб полегшити своєчасне коректування роботи проекту. Оцінка часто буває ретроспективною, і її найкраще проводити незалежною особою або органом. Метрика - це стандарт виміру, за допомогою якого можна оцінити ефективність, прогрес, продуктивність, або якість проекту чи продукту. Метрики управління проектами допомагають оцінити статус поточного проекту з точки зору графіка, вартості та рентабельності. Ці показники також допомагають передбачити будь-які потенційні ризики і виявити проблеми задовго до того, як вони стануть серйозними. Вони також допомагають відслідковувати прибутковість проекту, продуктивність його команди і якість результатів роботи. Деякі показники ефективності проекту включають: графік і відхилення трудовитрат, планове значення (PV), освоєну вартість (EV), відхилення графіка (SV) і використання ресурсів.

4. Складання бюджету

Бюджет вказує, як кошти будуть використовуватися і витрачатися протягом терміну дії проекту. В рамках бюджету всі зусилля, що витрачаються на досягнення результатів кожного завдання в проекті, визначаються з точки зору витрат, а графік показує послідовність дій.

Після затвердження менеджер проекту повинен контролювати бюджет і основні етапи графіка і забезпечувати їх дотримання. Бюджет і графік входять в число інструментів управління проектом.

5. Планування

План проекту - це офіційний затверджений документ, який використовується як для виконання, так і для управління проектом. Згідно Зводу знань з управління проектами (Project Management Body of Knowledge), план проекту документує припущення і рішення при плануванні, сприяє обміну інформацією між зацікавленими сторонами і засвідчує затверджений обсяг, вартість і базові плани графіка.

План проекту повинен відповідати на чотири основні питання про проект: яка робота виконується? Чому це робиться? Хто це робить? Коли це буде зроблено?

Інструменти планування включають WBS (ієрархічну структуру робіт), метод аналізу оцінки програми (PERT), діаграми Ганта та діаграми методу критичного шляху (CPM).

Висновки. Таким чином, можна підбити підсумки, що жорсткі навички це здібності, яких індивід набуває шляхом практики та навчання, їх важливість вважається беззаперечною, адже саме практичні технічні навички відіграють вирішальну роль у досягненні якісного та швидкого виконання завдань проекту. Їх наявність або відсутність можна легко перевірити за допомогою тестування або просто упевнитися, що співробітник має відповідний сертифікат або диплом, який засвідчуватиме рівень його технічної кваліфікації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Dr. Anupama, J., Swasti, Satpathy, Suvasweta. (2017). Importance of Soft Skills in Project Management. *International Journal of Scientific Research and Management (IJSRM)*. Retrieved from <https://www.ijssrm.in/index.php/ijssrm> DOI : 10.18535/ijssrm.
- [2] Kagan, J. (2021). Hard Skills. Retrieved from <https://www.investopedia.com/terms/h/hard-skills.asp>.
- [3] Malsam, W. (2019). Hard Skills Soft Skills: Understanding the Benefits of Both. Retrieved from <https://www.projectmanager.com/blog/hard-skills-vs-soft-skills>.

Biliavska O.

Hard skills as a determinant of success in solving problems in project management

Abstract. *This article explores the importance of hard skills as professional tools for achieving project management goals. Describes the five most essential tough skills that help to effectively solve project problems: contract and procurement management, risk management, monitoring and evaluating project performance, budgeting, planning.*

Keywords: *hard skills; planning; project management; material resources; analysis.*

УДК 005.8

УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЕКТАМИ НА ОСНОВІ СУЧАСНИХ АНАЛІЗУЮЧИХ ПРОГРАМ

Гайченко А. О., магістрант
Національний університет кораблебудування імені адмірала Макарова
Україна, Миколаїв
gaychenkon@gmail.com

***Анотація.** Ця стаття присвячена актуальній проблемі транспортної логістики, а саме – її автоматизації. Багато логістичних процесів потребують досить детального підходу, багаточасових розрахунків та великої концентрації уваги. Якщо брати до уваги людський фактор, то в наш час він навпаки ускладнює роботу та зростає відсоток помилок, тому що людський фактор не сприяє точності. Чим більша компанія, тим більше ризик людського фактору*

Основна мета цієї статті донесення важливості автоматизованих аналітичних технологій у транспортній логістиці.

***Ключові слова:** транспортна логістика; автоматизація; аналітичні технології; логістична оптимізація; проект.*

Вступна частина. Актуальність теми дослідження полягає у тому, що в сучасному світі автоматизація різних форм бізнесу досить серйозно змінює логістичну систему в сфері транспорту, починаючи з виробництва товару і завершуючи його споживанням. У 2019 році пандемія внесла корективи в сучасний світ і вказала на слабкі місця в сфері логістики і таким чином призвела до поліпшення автоматизації логістичних технологій, своєчасну доставку вантажу, швидке його завантаження та безконтактну доставку. Протягом часу пандемії COVID-19 велика кількість компаній почали досить багато коштів вкладати в автоматизацію та налаштування транспортних програм, для того щоб залишатися конкурентоспроможними.

Основна частина. Таким чином дослідниця Костишева Я. В. [1] виділяє перелік сучасних логістичних технологій, які допомагають безперешкодно пересуватися вантажу:

- автоматизована система планування поставок товарів, а також автоматизоване управління операціями по дистрибуції на основі SCM-технологій і засобів електронної комерції (E-commerce). Обробка отриманих замовлень та диспетчеризація здійснюється за допомогою центру виклику (call-центри) і навігаційної системи на основі стільникових мереж;

- застосування аутсорсингу в різних його формах. Найчастіше застосовується інформаційний аутсорсинг на основі ASP і DataCenter - технологій, а також транспортний аутсорсинг, застосовуваний по відношенню до регулювання операцій в ланцюгах поставок і моніторингу даних за допомогою віртуальних центрів даних (DataCenter).

Крім того, набуває поширення практика надання інтерактивних послуг з організації та планування маршрутів транспортування. Серед систем, які надають дані послуги можна виділити наступні: www.ati.su, www.gruzonline.ru, www.trans.ord.ua, www.belcargo.com і ін .;

- постійний моніторинг та навігація операцій, здійснюваних в ланцюгах поставок за допомогою Інтернету і RFID-технологій.

Дослідники І. Н. Лавриков та Н. В. Пеншин [2] виділяють, що найбільш часто в інформаційних логістичних системах використовуються наступні програмні продукти: Комплексна інформаційна система «Галактика» застосовується в системах автоматизації всіх напрямків спектра фінансово-господарської діяльності середніх і великих підприємств, в тому числі: «Управління постачанням», «Управління договорами», «Управління збутом» «Постачальники, одержувачі ». Також, важливим аспектом в сфері логістики є здатність

компанії адаптуватися до умов, що змінюються, як внутрішньої, так і зовнішнього середовища [3]. Дане завдання компанія в повній мірі може вирішити, забезпечуючи себе сучасними аналітичними інформаційними системами. Важливо створювати ті умови, які вимагає сучасне суспільство, щоб в першу чергу не допустити витоку клієнтів до конкурентів, а в другу чергу - забезпечити максимальну ефективність логістичної системи компанії.

Якщо відповідати на питання, які ж саме вигоди приносить автоматизація транспортної логістики? То можна відповісти: технології створенні для того, щоб спростити людині життя, підвищити ефективність, зекономити кошти та час, знизити ризики та підвищити прибуток.

Система автоматизованої транспортної логістики створюється спеціально під певну компанію та структуру та має в собі лише ті функції, які будуть використовуватися, для того щоб система вирішувала завдання, які ставляться перед нею.

Для кожного з користувачів система має свій рівень доступу, який дозволяє вирішувати всі необхідні завдання безпосередньо в самій системі, наприклад: для водіїв, для менеджерів і для постачальників.

Висновки. Отже, основним призначенням транспортно-логістичного програмного продукту є складання найбільш ефективного маршруту, шляхом постійного моніторингу та навігації операцій, здійснюваних в ланцюгах поставок за допомогою Інтернету та програмного забезпечення для того щоб реалізувати проект доставки, враховуючи при цьому такі параметри, як: часовий інтервал доставки, точки адреси поставки. Програми призначені для малих, середніх і великих підприємств. Важливою перевагою програм є наявність модуля GPS, що дозволяє здійснювати контроль руху транспорту в режимі реального часу, а також переміщати запис шляху проходження в архів. Таким чином автоматизація - це рішення не тільки для транспортно-логістичних компаній, а для будь-якої компанії, яка має потребу в регулярному перевезенні або відправленні товарів: великих інтернет-магазинів, виробництв сільськогосподарських підприємств та інших.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Костишева, Я.В. (2013). Эффективность применения программных обеспечений в области транспортной логистики. *Экономикс*, (1), 47.
- [2] Лавриков, И.Н., Пеньшин, Н.В. (2003). *Транспортная логистика*. М.: Экзамен.
- [3] Лавриков, И.Н., Пеньшин, Н.В. (2016). *Транспортная логистика*. Тамбов : Изд-во ФГБОУ ВО «ТГТУ».

Haichenko A.

Management of logistics projects based on the use of modern applications

Abstract. *This article is devoted to the current problem of transport logistics, namely - its automation. Many logistics processes require a very detailed approach, long-term calculations and great concentration. If we take into account the human factor, in our time, on the contrary, it complicates the work and increases the percentage of errors, because the human factor does not contribute to accuracy. The larger the company, the greater the risk of the human factor*

The main purpose of this article is to convey the importance of automated analytical technologies in transport logistics.

Keywords: *transport logistics; automation; analytical technologies; logistics optimization; project.*

ІНСТРУМЕНТИ УПРАВЛІННЯ ЯКІСТЮ В ПРОЕКТАХ ТУРИСТИЧНОЇ ГАЛУЗІ

Корулик А.Д., магістрант¹; Калабін С.Ю., магістрант²

^{1,2}Національний університет кораблебудування імені адмірала Макарова

^{1,2}Україна, Миколаїв

¹barina16@gmail.com; ²s1kalabin999@gmail.com

Анотація. В статті досліджено інструменти управління якістю проектів туристичної галузі. Якість розглядається як один з основних керованих параметрів проекту. Сім інструментів управління якістю забезпечують розуміння складних ситуацій і дозволяють полегшити задачу управління якістю шляхом поліпшення процесу проектування продукції або послуги.

Ключові слова: управління якістю; проект; туристична галузь.

Вступ. Сучасний етап розвитку туристичної діяльності містить позитивні зміни, головні з яких повинні враховувати прагнення покращити їх економічний показник для рахунку, що не входить до моделей, методів та систем якості на базах міжнародних стандартів. Задача забезпечення якості проекту актуальна на всіх фазах його життєвого циклу, через що виникає проблема неефективності систем управління в проектах туристичної області.

Туристична сфера в усьому світі зазнає збитків внаслідок заходів щодо протидії пандемії COVID-19. Водночас, ситуацію, що склалася, слід розглядати не лише як проблему, але і як можливості для виведення туристичної сфери в Україні на якісно новий рівень.

Основна частина. Питання ефективного управління проектами у сфері туризму є актуальними, оскільки успішна реалізація призведе до розвитку галузі. Існує багато аспектів, які можуть призвести до ефективності проекту. Розглянемо інструменти управління якістю проектів у сфері туризму.

Мета політики якості полягає в тому, щоб врахувати інтереси споживачів при розробці та наданні послуг, постійного поліпшення якості без додаткової витрати матеріальних, людських і фінансових ресурсів підприємства. Який би сучасний засіб вирішення ділової задачі ми не взяли до розгляду (проект або бізнес-процес), їм завжди властиві три конкуруючих один з одним параметра: тривалість, вартість і якість. Не можна сказати, що якість є найскладнішим з вказаною трійці, швидше за все, витрати на його забезпечення є найбільш небажаними, оскільки вони здатні істотно погіршити бюджет. Однак управління якістю проекту є одним з найважливіших аспектів проектно-управлінської доктрини.

У будь-якому випадку продукт діяльності повинен бути розглянутий нами як предмет, що володіє складом певних властивостей, які важливі як для його виробника, так і для споживача. З позиції абсолютно різних підходів до визначення якості пропонується розуміти ступінь відповідності вимогам ряду властивих предмету характеристик [1].

Якість і задоволення клієнтів є однією з основних цілей, яких має досягти проект. Відповідно до PMI (Інститут Управління Проектами) якість – це ступінь, до якої набір властивих характеристик відповідає вимогам [1].

Управління якістю проекту сумісне зі стандартами та інструкціями щодо якості ISO 9000 та ISO 10000. Управління якістю проекту включає процеси та діяльність виконавчої

організації, які визначають політику якості, цілі та відповідальність, щоб проект задовольнив потреби, для яких він був здійснений [1].

Найчастіше використовувані інструменти якості в управлінні проектами у сфері туристичної галузі - це: діаграми Парето, контролювати діаграми, мозковий штурм, діаграму розсіювання тощо (рис.1).

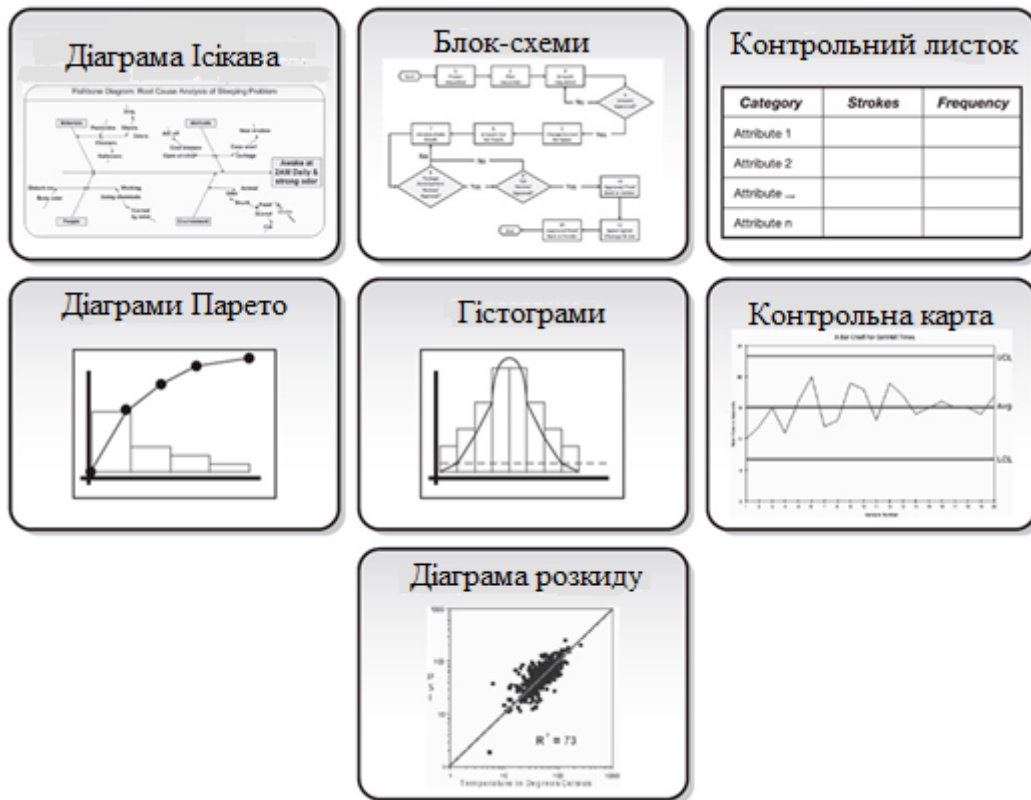


Рис.1.Інструменти планування, що застосовуються при управлінні якістю проектів туристичної сфери.

Діаграма Ісікава (причинно-наслідковий діаграма) - інструмент, який дозволяє виявити найбільш істотні фактори (причини), що впливають на кінцевий результат (наслідок).

Блок-схеми - це один з основних інструментів якості, який можна використовувати для аналізу послідовності подій. Інструмент відображає послідовність подій, які відбуваються послідовно або паралельно. Блок-схема може бути використана для розуміння складного процесу, щоб знайти взаємозв'язку і залежності між подіями. Блок-схеми можуть використовуватися для будь-якого поля, щоб простим способом ілюструвати складні процеси. Для малювання блок-схем розроблені спеціальні програмні інструменти, такі як MS Visio.

Контрольний листок - інструмент для збору даних і їх автоматичного упорядкування для полегшення подальшого використання зібраної інформації.

Діаграма Парето - інструмент, що дозволяє об'єктивно представити і виявити основні фактори, що впливають на досліджувану проблему, і розподілити зусилля для її ефективного вирішення.

Гістограма - інструмент, що дозволяє візуально оцінити розподіл статистичних даних, згрупованих за частотою попадання даних в певний (заздалегідь заданий) інтервал.

Контрольна карта - інструмент, що дозволяє відстежувати хід протікання процесу і впливати на нього (за допомогою відповідної зворотного зв'язку), попереджаючи його відхилення від пред'явлених до процесу вимог [2].

Діаграма розкиду (розсіювання) - інструмент, що дозволяє визначити вид і тісноту зв'язку між парами відповідних змінних.

Сім інструментів управління якістю забезпечують розуміння складних ситуацій і дозволяють полегшити задачу управління якістю шляхом поліпшення процесу проектування продукції або послуги. Інструменти управління якістю підсилюють процес планування завдяки їх здатності: з'ясовувати завдання; усувати недоліки; сприяти поширенню та обміну інформацією між зацікавленими сторонами; використовувати побутову лексику. В результаті інструменти управління якістю дозволяють приймати оптимальні рішення в найкоротші терміни.

Якість розглядається як один з основних керованих параметрів проекту, тому завдання забезпечення якості проекту повинна вирішуватися на кожному етапі життєвого циклу проекту системою, основу якої складають найбільш ефективні інструменти управління якістю.

Висновок. В управлінні проектами одне з ключових місць займає процес управління якістю (ступінь відповідності характеристик проекту вимогам). Процес управління якістю - проектів можна назвати основним аспектом управління проектом нарівні з управлінням змістом проекту і його ризиками. Процес управління якістю в даний час грає стратегічно важливу роль для утримання компаніями свого місця на ринку [3].

Поліпшення якості туристичних послуг, задоволення споживачів, отримання певного соціального результату стане важливим внеском туристичних підприємств у регіональний розвиток.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Подходы к управлению качеством проекта.(2020).Retrieved from <http://projectimo.ru/realizaciya>.

[2] Мазур, И.И., Шапиро, В. Д. (2003). *Управление качеством: учебное пособие*. М.: Высшая школа.

[3] Копець, Г. Р. , Кулиняк, І. Я. (2020). Сучасні напрями розвитку підприємств у сфері туризму в Україні. *Вісник Національного університету "Львівська політехніка". Серія "Проблеми економіки та управління"*, 2(8), 37-48.

Korulyk A.D., Kalabin S.Yu.

Quality management tools in tourism projects

Abstract. *The article investigates the tools of quality management of tourism projects. Quality is considered as one of the main managed parameters of the project. Seven quality management tools provide an understanding of complex situations and facilitate the task of quality management by improving the process of designing products or services.*

Keywords: *quality management; project; tourism industry.*

УДК 658.5:004.771

ОРГАНІЗАЦІЯ РОБОЧОГО ПРОЦЕСУ ТА КОМУНІКАЦІЇ З КЛІЄНТОМ В УМОВАХ ДИСТАНЦІЙНОЇ РОБОТИ: ТЕХНІЧНА ПІДТРИМКА В ОНЛАЙН РЕЖИМІ

Білоножко К. С., магістрант¹, Білоножко М. С., магістрант²
Національний університет кораблебудування імені адмірала Макарова
Україна, Миколаїв

¹ belonozhkokatya@ukr.net; ² bilonozhkomasha@ukr.net

Анотація. В ході дослідження було проаналізовано сучасні підходи організації робочого процесу та технічної підтримки в умовах дистанційної роботи. Використання інформаційно-комп'ютерних технологій для взаємозв'язку та обслуговування клієнтів через програми віддаленого доступу.

Ключові слова: організація; віддалений доступ; TeamViewer; технічне обслуговування; ІТ-адміністрування.

Вступна частина. Сучасний світ розвивається дуже стрімко. Ми живемо в час розвитку інформатизації та автоматизації. Глобалізація і прискорений інформаційний потік спричиняють необхідність постійних змін в організації і вимагають, щоб управлінські рішення були частішими і швидшими [1]. На сьогоднішній день постає актуальність та важливість віддаленої роботи. Така робота регулюється законопроектами та законодавчими актами, але має ще багато недоліків. У цьому напрямку діяльності ще багато питань, тому до законодавчої бази державні органи вносять зміни та поправки в ході подій та застосування її на практиці. Саме віддалена робота наразі розглядається як перспективний варіант у подальшому розвитку в умовах пандемії. Метою даного дослідження є забезпечення ефективного та успішного процесу управління за допомогою сучасних автоматизованих методів, що відповідають вимогам часу і розвитку суспільства.

Основна частина. Дистанційна робота виконується за допомогою інформаційно-комунікаційних технологій. У неї є свої недоліки та переваги. Отже, перед керівниками ІТ-компаній, які обслуговують програмне забезпечення, постала задача організації та забезпечення належних комфортних умов праці. Значна увага має приділятися управлінню внутрішньому процесу та поєднання комунікації з клієнтами. Тут враховується різні фактори: психологічний, який впливає на пристосування до нових умов та змін звичної атмосфери. Технічне забезпечення, інформаційна грамотність, оперативність та гнучкість реагування на будь-які ситуації, що виникають у ході дистанційної роботи.

Звичним варіантом організації віддаленого доступу стає використання відповідного програмного забезпечення. У цьому випадку два комп'ютери будуть пов'язані між собою з використанням інтернету або через локальну мережу [2]. Цей модернізований спосіб дозволяє:

- побачити на екрані власного пристрою дані з екрану іншого комп'ютера;
- безперешкодно обмінюватися файлами;
- забезпечувати скачування, установку і настройку ПО;
- створювати скріншоти екрану;
- здійснювати шифрування даних;

- проводити конференції та аудіо-відеоформаті;
- підключати мережеві пристрої і здійснювати їх настройку.

Такий формат пропонує нові альтернативні підходи та методи організації процесу роботи. Розглянемо деякі з них:

Підготовка IT служби компанії, яка має надати віддалене адміністрування через спеціальні програми або додатки віддаленого доступу. Саме через неї співробітник зможе в режимі он-лайн надавати послуги для клієнта. В свою чергу клієнт має користуватися такою самою програмою для підключення. Необхідно враховувати якісний доступ до мережі Інтернет. Це дуже важливо у роботі, і має бути в пріоритеті.

1. TeamViewer є швидким і безпечним комплексним рішенням для одержання віддаленого доступу до комп'ютерів і мереж. Завдяки безлічі ефективних функцій для віддаленого доступу, які спрощують віддалене управління, проведення онлайн-нарад і роботу хмарної служби підтримки, існує цілий ряд ресурсів для демонстрації всіх можливостей TeamViewer. На висоті і функціонал – в наявності маршрутизація, автоматичні настройки пропускну здатності мережі, можливість інтелектуального підключення. При простоті використання TeamViewer цікава досвідченому користувачеві і зрозуміла початківцям. Вивчивши доступні керівництва користувача, інструкції з початку роботи і загальні практичні вказівки, ви дізнаєтеся про всі можливості TeamViewer. IT-підрозділ або постачальник керованих послуг, що використовує дане комплексне рішення, отримує в своє розпорядження всі ці інтуїтивно зрозумілі функції. Сьогодні вона активно використовується в багатьох країнах – для використання пропонується 30 мов.

2. AnyDesk. Це потужний інструмент, що дозволяє організувати роботу віддалено. Він охоплює максимум завдань і є повністю безкоштовним – правда, якщо мова йде про некомерційне використання. Серверна частина пропонує портативну версію, яка працює, в тому числі, і без установки. Можливе підключення до платформ iOS, macOS, Linux, Windows, Android. У списку платформ, з яких може здійснюватися робота – ті ж варіанти.

3. DameWare. Цікаве рішення, що дозволяє групі підтримки віддалено контролювати комп'ютер користувача. Тобто фахівець може без проблем отримати доступ до пристрою, досліджувати і усунути проблеми. Система захищена шифруванням, а тому використання її через Інтернет безпечно.

4. Наступний крок – це використання електронної пошти для верифікації документів, щодо проведеної роботи чи наданих технічних послуг шляхом дистанційного обслуговування клієнта. Це робиться і в звичайних умовах для контролю виконання завдання згідно внутрішніх інструкцій та звітності. Також це доказ того, що послуга була надана і підтверджена клієнтом задля уникнення будь яких конфліктних ситуацій в подальшому.

5. Використання он-лайн чатів для спілкування в режимі реального часу. Це досить зручно і поширено серед співробітників. Він надає можливість для обговорення проблеми, обміну інформацією.

6. Важливо зазначити та приділити увагу хмарним технологіям – вони забезпечують комплексне рішення та безпечне зберігання даних, інформації компанії.

Висновки. Віддалене адміністрування сьогодні стає поширеним і затребуваним. Якщо раніше дані можливості використовувалися лише для моніторингу робочих столів, то сьогодні можливий повний контроль ПК, а також можливість його використання в особистих цілях. На даний момент завдяки використанню віддаленого адміністрування у людей з'явився цілий ряд можливостей, серед яких аутсортинг, використання хмарних сервісів та ін. Навіть

найменший офіс має потребу у віддаленому адмініструванні – це допоможе уникнути витрат на утримання фахівця і придбання дорогого устаткування [2].

Якісне управління проектами та взаємозв'язок співробітників з клієнтами, яких вимагає будь-яка діяльність або сфера, особливо ІТ має бути продуктивною і націленою на гарний результат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Ноздріна, Л. В. (Ред.). (2010). *Управління проектами*. К.: Центр учбової літератури.

[2] ТОП-4 программ удаленного доступа к компьютеру в 2021 году. (2021). Retrieved from <https://softlist.com.ua/articles/top-5-programm-udalennogo-dost/>.

Bilonozhko K. S., Bilonozhko M. S.

Organization of work process and communication with the client in the conditions of remote work: technical support online

Abstract. During the study, modern approaches to the organization of the work process and technical support in the conditions of remote work were analyzed. Use of information and computer technologies for communication and customer service through remote access programs.

Keywords: organization; remote access; TeamViewer; maintenance; IT-administration.



ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УДК 004.42

АНАЛІЗ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ СПІВРОБІТНИКІВ ТА СТУДЕНТІВ КАФЕДРИ

Шматко А.І., магістрант¹; Барабаш О.В., д.т.н., професор²^{1,2} Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського»^{1,2} Україна, м. Київ¹ andriyshmatko1999@gmail.com; ² bar64@ukr.net

Анотація. Застосування інформаційних технологій для ефективного і якісного управління кафедрою. Запропонована система передбачає здійснення ефективного обліку співробітників та студентів кафедри.

Ключові слова: система обліку; автоматизація обліку; облік співробітників кафедри; інформаційні системи.

Вступна частина. Ефективне використання ресурсів, у тому числі кадрових, є важливою ланкою на шляху до конкурентоспроможності закладів вищої освіти. Вирішення освітніх питань потребує змін та створення інформаційної системи для кращого управління. Специфіка об'єкта управління, яким є освітня діяльність, і слабкий розвиток інформаційних систем для освітньої галузі роблять актуальним завдання розробки автоматизованих систем управління процесами життєвого циклу освітньої діяльності вишу та його основних виробничих підрозділів - кафедр.

Основна частина. Ефективне і якісне управління кафедрою, засноване на застосуванні інформаційних технологій є однією з основних умов для її успішного розвитку, а також випуску затребуваних ринком праці фахівців [1; 2]. Це в свою чергу збільшує конкурентоспроможність кафедри на ринку освітніх послуг.

Впровадження інформаційної системи на кафедрах є важливим кроком у комплексній автоматизації всього навчального закладу.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- вивчити теорію управління освітніми установами та ВНЗ;
- вивчити проблеми організації управління ВНЗ;
- проаналізувати діяльність кафедри та побудувати процесну модель;
- спроектувати інформаційну систему;
- реалізувати програмне забезпечення системи;
- впровадити реалізовані частини системи на кафедрі;
- отримати і проаналізувати результати виконаної роботи.

Одним з найважливіших показників вишу є його кадрове забезпечення. Система автоматизації обліку співробітників допоможе виконати наступні задачі:

- відстежувати динаміку змін кадрового складу кафедр;
- вести статистику за науковими ступенями та науковими званнями;
- контролювати підвищення кваліфікації працівників, перепідготовку, сертифікацію;
- вести облік видання наукових праць та винаходів;
- здійснювати формування звітних документів про кадрове забезпечення кафедри.

В даний час для оцінки ефективності праці співробітника вишу став використовуватися показник - публікаційна активність [3]. Він враховує науково-дослідницьку роботу і оцінюється сумарною кількістю публікацій автора в національних і зарубіжних журналах, у збірниках праць конференцій, а також у вигляді виданих монографій. Таким чином, актуальним стає завдання обліку публікацій, з подальшим аналізом і підрахунком балів публікаційної активності співробітника вишу, отримання різних форм звітності. Запропонована автоматизована інформаційна система дозволяє вести облік наукової активності співробітника кафедри. До вхідних документів належать дані виконаної та виданої наукової роботи викладача кафедри. У системі до вхідної оперативної інформації можна віднести дані видання, в якому друкується наукова робота співробітника кафедри. Таким чином, на основі введення вхідної інформації, формуються результуючі дані про наукову діяльність кафедри. Вся результуюча інформація підрозділяється на інформацію, що виводиться в екранні форми і друкується на принтері. На формі зазвичай формуються дані для перегляду і редагування, а також дані попереднього розрахунку, готові до друку.

Показником роботи студентів і викладачів, які навчають їх, є успішність з дисциплін, що знаходиться в прямопропорційній залежності від відсотка відвідуваності занять. Керівництво навчального закладу зацікавлене в тому, щоб здійснювався безперервний облік і контроль за діяльністю студентів, зокрема за їх відвідуваністю. Автоматизована система зробить даний облік менш трудомістким та більш ефективним. Підвищення оперативності обліку і контролю відвідуваності та успішності студентів сприятиме збільшенню продуктивності і зниженню трудомісткості вирішуваних завдань.

Висновки.

Зроблений аналіз методів сприяє розробці і впровадженню інформаційної системи, яка дозволяє автоматизувати процес обліку студентів та співробітників кафедри.

Система допоможе з більшою ефективністю накопичувати, зберігати, обробляти та аналізувати інформацію про кадровий склад кафедри; ефективніше використовувати кадрові ресурси.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Баженов Р.І., Кузнєцова Я.І. (2014). Про розробку інформаційної системи з обліку благодійної допомоги в малій організації. *Сучасна техніка і технології*, 6(34).
- [2] Баженов Р.І., Нікітін А.В. (2014). Про розробку інформаційної системи з контролю над пролонгацією страхових договорів. *Сучасні наукові дослідження та інновації*, 6(38).
- [3] Борівський М.А., Шевченко І. К., Масич М. А. (2013). Інструментальні засоби кваліметрування результативності діяльності науково-педагогічних співробітників в системі управління вузом. *Бізнес інформ*, 5, 106-112.

Shmatko A.I., Barabash O.V.

Analysis of methods for developing an information system for accounting of employees and students of the department

Abstract. *Application of information technologies for effective and high-quality management of the department. The proposed system provides for the effective accounting of employees and students of the department.*

Keywords: *accounting system; accounting automation; accounting of employees of the department; information systems.*

УДК 004.414.2

АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ТА АДМІНІСТРУВАННЯ ВІДВІДУВАНOSTІ УНІВЕРСИТЕТУ

Михалко М. І.¹; Гайдаєнко О.В., к.т.н., доцент кафедри ІУСТ²

^{1,2}Національний університет кораблебудування імені адмірала Макарова

^{1,2}Україна, Миколаїв

¹mikhalko98@gmail.com; ²okotsur80@gmail.com

Анотація. *Темою роботи є розробка автоматизованої інформаційної системи контролю та адміністрування відвідуваності університету. Система спрямована на підвищення ефективності діяльності навчального відділу та студента. В результаті роботи було спроектовано систему, в основу якої покладено технологію NFC для ідентифікації студентів. Програмне забезпечення виконане мовами програмування Java, PHP, JS.*

Ключові слова: *Інформаційна система; NFC; контроль відвідуваності; мобільний додаток; android.*

Вступна частина. Широкомасштабне використання у всіх сферах діяльності інформаційних технологій, що являють собою складний комплекс інформаційно-телекомунікаційних систем, потребує впорядкування та оптимізації процесів, їх розбудови та підтримки. Відвідування навчального закладу, тобто систематична присутність студента на заняттях завжди була однією з проблем університету. Збільшення показників присутності студентів дозволяє покращити якість засвоєваних знань. Підвищення рівня відвідуваності студентів університету сприяє вирішенню низки організаційних питань і створює якісний імідж навчального закладу.

Метою роботи є підвищення ефективності контролю за відвідуванням занять студентами навчального закладу та спрощення механізму роботи з розкладом занять шляхом впровадження автоматизованої інформаційної системи контролю та адміністрування відвідуваності вищого навчального закладу.

Основна частина. При дослідженні питання адміністрування відвідуваності університету було виявлено, що на сьогоднішній день не існує комплексної інформаційної системи, яка б містила в собі процес ідентифікації осіб при вході до приміщення, розклад занять, сповіщення про зміни в розкладі, електронний журнал для обліку роботи академічної групи та автоматизоване адміністрування даних з журналу обліку роботи академічної групи. Актуальність роботи полягає в тому, що автоматизована система значно зменшить витрати часу співробітників навчального відділу університету та студентів, а також покращить якість показників відвідуваності університету [1].

Застосування сучасних технологій може вирішити проблеми, які було детально описано при дослідженні. Для цього було спроектовано автоматизовану інформаційну систему контролю та адміністрування відвідуваності університету. Система являє собою комплекс програмних рішень, які взаємодіючи вирішують низку організаційних питань навчального закладу.

За основу було взято технологію NFC, яка активно розвивається та є дуже перспективною. NFC (Near-Field Communication) – це технологія бездротового високочастотного зв'язку малого радіусу дії “в один дотик”. За допомогою цієї системи смартфони, що перебувають на відстані близько 10 см можуть обмінюватися даними.

Вагомою перевагою даної технології є висока швидкість взаємодії між пристроями ($\sim 0,1\text{с}$). NFC працює на частоті 13,56 МГц та складається із зчитувача та антени, або із мітки та антени. Зчитувач – це пристрій NFC, який працює в режимі активної комунікації. Мітка – це пристрій NFC, який працює в режимі пасивної комунікації [2].

Існують три режими роботи пристрою NFC:

1. Режим емуляції карти.
2. P2P (Peer-to-peer) режим.
3. Режим читання/запису.

Щоб вирішити представлені в роботі задачі було обрано режим роботи пристрою NFC – режим читання/запису. Даний режим дозволяє пристроям з підтримкою NFC зчитувати інформацію, яка зберігається в NFC-тегах (чи мітках), або взаємодіяти з іншими NFC-пристроями в режимі читання/запису. Ініціюючий пристрій може зчитувати дані з другого пристрою або записувати дані на нього. Схематично режим читання/запису представлений на рисунку 1 [3].

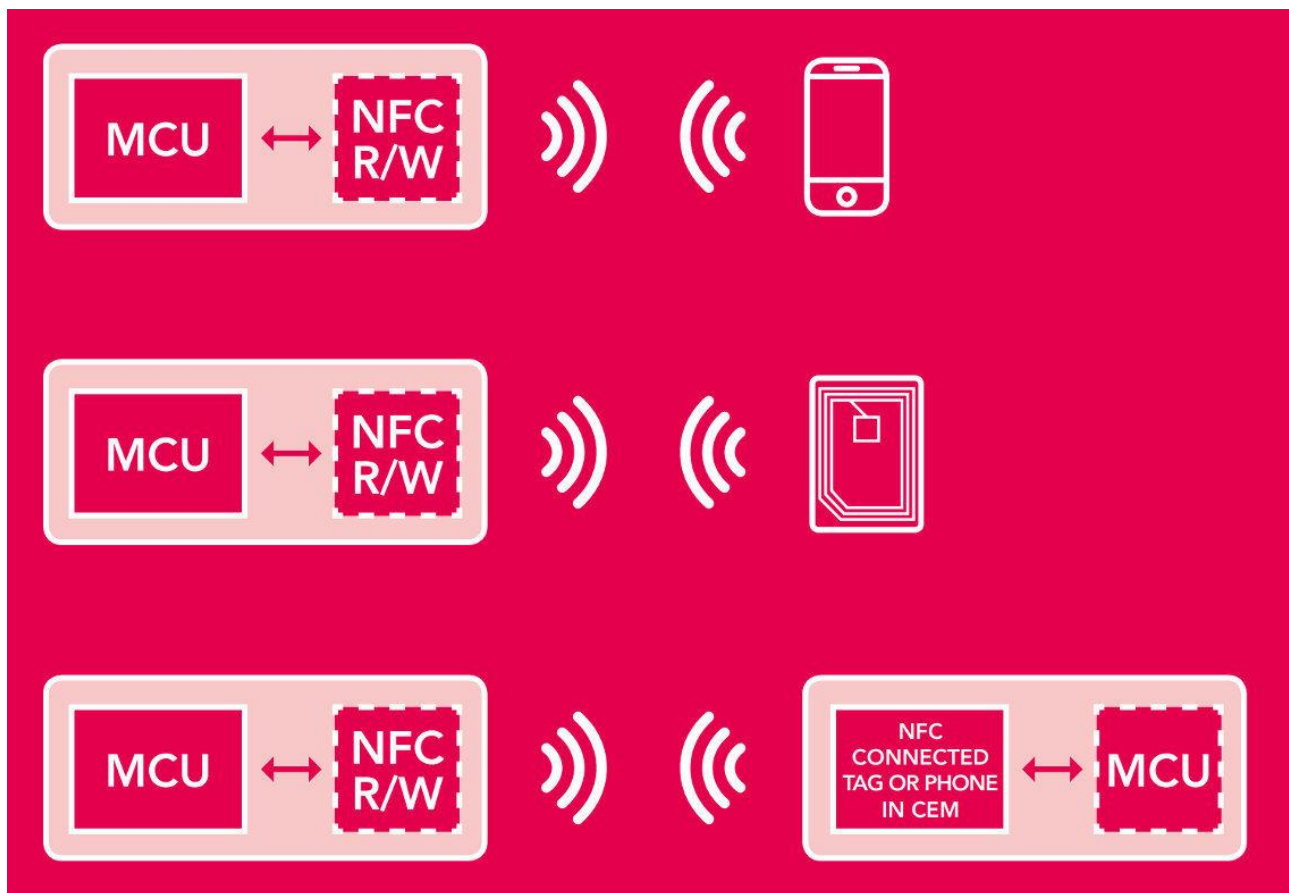


Рисунок 1 – Схема режиму читання/запису NFC-пристрою

Створена система містить в собі наступні функції:

1. Ідентифікація особи за допомогою технології NFC.
2. Електронний розклад.
3. Електронний журнал обліку роботи академічної групи.
4. Сповіщення в особистому кабінеті про зміни в розкладі.
5. Адміністрування даних збережених в журналі обліку роботи академічної групи.
6. Формування звітів.

Висновки. В результаті дослідження було спроектовано автоматизовану інформаційну систему, яка спрямована на підвищення ефективності діяльності навчального відділу та студента, а також покращення якості показників відвідуваності університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Ворона, В. А., Тихонов, В. А. (2016). *Системы контролю та управління доступом*. Москва: “Горячая линия - Телеком”.
- [2] Tom, I., Don, C., Brian, J. (2014). *Beginning NFC: Near Field Communication with Arduino, Android, and PhoneGap*. United States of America: O'Reilly Media, Inc.
- [3] Anne-Marie, L., Serge. M. (2017). *The Art and Science of NFC Programming*. London SW12 4EU UK: JohnWiley & Sons, Inc.

Mykhalko M.I., Haydaenko O.V.

Automated information system for control and administration of university attendance

Abstract. *The theme of the work is the development of an automated information system for control and administration of university attendance. The system is aimed at improving the efficiency of the educational department and the student. As a result, a system based on NFC technology for student identification was designed. The software is made in programming languages Java, PHP, JS.*

Keywords: *information system; NFC, traffic control; mobile application; android.*

УДК 004.12

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ЗАДАЧ УПРАВЛІННЯ ТА ОРГАНІЗАЦІЇ НАВЧАЛЬНО-НАУКОВОЇ ЛАБОРАТОРІЇ ВИПРОБУВАНЬ НЕМЕТАЛІЧНИХ МАТЕРІАЛІВ

Черновалов О. В., магістрант¹; Казимиренко Ю. О., д.т.н. наук, професор²;

Маршак О. І., старший викладач³; Бобровський О. С., провідний фахівець⁴

^{1,2,3,4}Національний університет кораблебудування імені адмірала Макарова

^{1,2,3,4} Україна, Миколаїв

¹alex.chernovalov@gmail.com; ²u.a.kazimirenko@gmail.com;

³olena.marshak@nuos.edu.ua; ⁴oleksii.bobrovskyi@nuos.edu.ua

Анотація. Проаналізовано шляхи підвищення ефективності управління та організації науково-дослідною роботою студентів вишу через застосування професійної інформаційної підтримки та впровадження web-технологій. Процес розглянуто на прикладі вузькопрофільної лабораторії випробувань неметалічних матеріалів, для якої визначено принципи побудови інформаційної платформи.

Ключові слова: Інформаційна підтримка. Web-технології. Науково-дослідна робота. Онлайн-обговорення. Віртуальний режим. Освітня складова.

Вступна частина. Науково-технічне забезпечення випробувань неметалічних матеріалів є важливою науково-технічною проблемою, вирішення якої не можливе без застосування інформаційних технологій. Вся інформація зосереджується у окремих метрологічних підрозділах та вузькоспеціалізованих лабораторіях, які не матимуть нічого спільного з навчальним процесом майбутніх фахівців. Тому виникатимуть питання щодо створення навчально-наукових лабораторій, які є підрозділами у складі кафедр і центрів. Вирішення організаційних питань полягає у встановленні зв'язку між замовниками, персоналом, програмами випробувань та завантаженістю обладнання.

Метою роботи є розробка інформаційної платформи для управління та організації навчально-дослідницької діяльності лабораторії випробувань неметалічних матеріалів як самостійного структурного підрозділу у виші.

Основна частина. Методологічну базу досліджень становлять основи методології та організації наукових досліджень [1], сучасний досвід застосування web-технологій для організації ефективної науково-дослідної роботи у інженерній галузі [2] та інноваційні розробки з постановки лабораторних досліджень в області теорії твердого деформівного тіла [3]. Нормативними підставами вважається законодавча база України, а саме Закони України: «Про освіту», «Про вищу освіту» та «Про науково-технічну діяльність».

Основним завданням сучасних навчально-наукових лабораторій експериментальних досліджень, зокрема випробувань неметалічних матеріалів, є підготовка здобувачів вищої освіти, наукова робота студентів, науково-педагогічних працівників і допоміжного персоналу, а також наукова робота з учнями старших класів ЗОШ, ліцеїв, студентів коледжів. Метою є популяризація досягнень вітчизняної науки, а також надання додаткових послуг за визначеним профілем. Суттєвим недоліком у організації досліджень є низький рівень інформування про наукові досягнення у суміжних галузях або у потребах вітчизняних та зарубіжних підприємств. Це потребує аналізу та вивчення web-технологій з урахуванням специфіки

наукових досліджень. Розвиток застосування мережевих технологій дасть змогу розширити наукові інтереси і зробити цікавими дослідження фізико-механічних властивостей матеріалів не тільки для студентів технічних спеціальностей, а й майбутніх фахівців економічного, екологічного, юридичного та дизайнерського профілю. Крім того, в умовах дистанційного навчання стає можливою організація роботи у віртуальному режимі, що сприятиме розширенню сфери наукових інтересів та залучення до досліджень та обговорення студентів інших вишів, міст та країн.

Головними вимогами до розробки інформаційної платформи є інтуїтивно-зрозумілий інтерфейс користувача, зручна і логічна навігація, кросбраузерність, оптимізований пошук по сайту, висока швидкість завантаження сторінок, наявність зворотнього зв'язку, відповідність контенту тематиці сайту.

Поряд із загальноприйнятими розділами, такими як «Про нас», «Новини», «Контакти» сайт повинен містити розділи, які б у повній мірі висвітлювали науковий доробок та освітню складову. Це розділи:

- *«Напрямки наукових досліджень»* – розташовано матеріали про наукову роботу, наукові проекти, замовлення, контракти зі стислим описом змісту, наукові публікації, патенти, методики науково-педагогічних працівників і студентів, матеріали конференцій та практичних семінарів за їх участю тощо;

- *«Обладнання»* – наведено опис та фотографії установок, лабораторного устаткування з методичними рекомендаціями щодо роботи на ньому;

- *«Освітня діяльність»* – містить підрозділи, які висвітлюють зв'язок лабораторії з освітнім процесом, а саме освітні програми, робочі програми навчальних дисциплін, методичні вказівки до лабораторних робіт та практикумів, пов'язані з направленням діяльності лабораторії;

- *«Майстер-класи»* – наводиться інформація щодо проведення майстер-класів, семінарів та інших видів занять за вимогами Замовників.

Особливість структури сайту, що розробляється, полягатиме у наявності розділу «Форум», на який виносяться питання для тематичних обговорень, що реалізує потреби у онлайн-спілкуванні науковців з фахівцями підприємств та студентами, для чого вони реєструються та проходять відповідну процедуру модерації. Крім того, у цьому розділі можуть бути опублікованими консультації провідних фахівців з вибору матеріалів, пристосування існуючих технологій у певних виробничих умовах, методів їх зміцнення, захисту конструкцій та ін. Макет титульної сторінки форуму наведено на рисунку 1.

Попередній аналіз сайтів навчально-наукових лабораторій вишів свідчить, що зазвичай формою зворотнього зв'язку є розташовані у розділі «Контакти» адреса електронної пошти керівника лабораторії та телефон. Це, безумовно, надає можливість зацікавленим сторонам поставити запитання, надіслати відгук або пропозицію, але не створює умови для сумісного обговорення поставленої проблеми науковою спільнотою.

Для підвищення ефективності комунікацій на матеріалах сторінок сайту реалізовано функцію швидкого створення гілки обговорення контенту через автоматичний модуль керування форумом. Кнопка «Обговорити на форумі» надає можливість зареєстрованому користувачу надати запит на створення гілки теми форуму безпосередньо зі сторінки розділу, який містить питання тематичного обговорення.

Крім того, на форумі реалізована система ранжування повідомлень за рейтингом та автоматичного відображення найвищого за рейтингом повідомлення у примітках матеріалу, що є предметом обговорення.

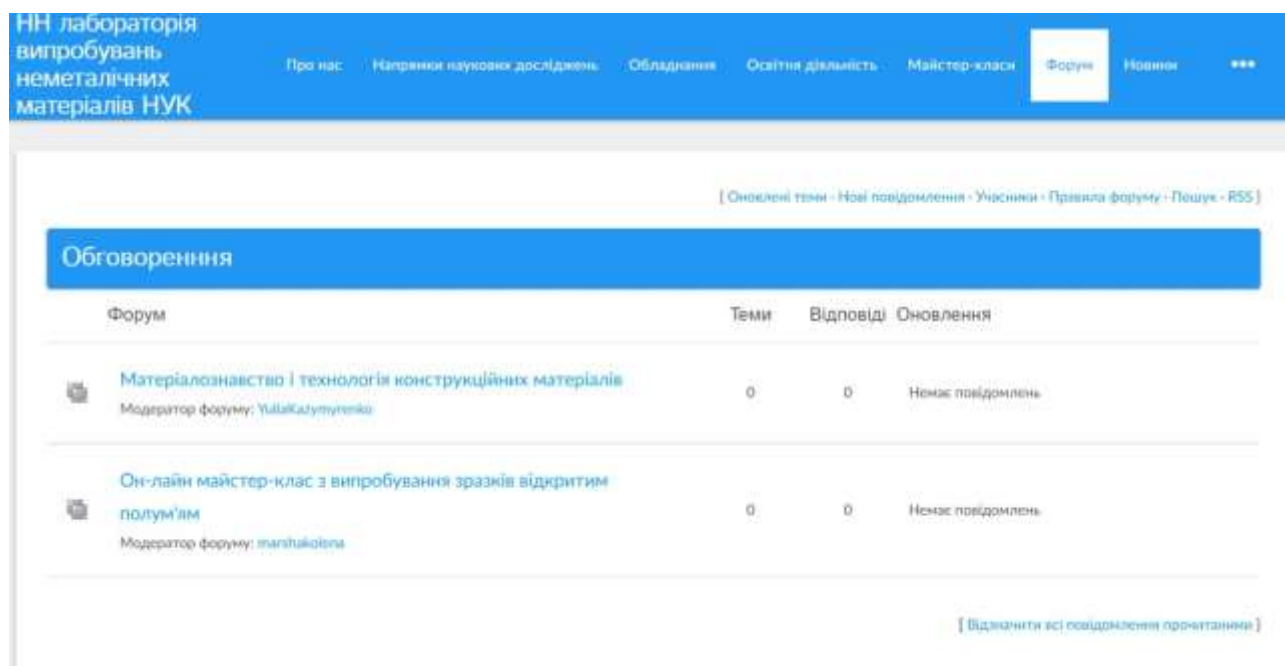


Рисунок 1 – Макет сторінки форуму

Перспективи подальших досліджень пов'язано з апробацією інформаційної платформи та запуском пілотного проекту.

Висновки.

1. На прикладі проекту лабораторії випробувань неметалічних матеріалів запропоновано принципи створення інформаційної платформи для управління та організації навчально-дослідницької діяльності, що ґрунтується на просуванні web-технологій з урахуванням специфіки наукових досліджень.

2. Застосування web-ресурсів у навчально-науковій діяльності дозволить надати інформацію користувачам, запровадити віртуальні форми досліджень, навчання та консультацій, розширити аудиторію зацікавлених сторін.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Луцик, І. Б. (2014). Застосування web-технологій для організації ефективної науководослідної роботи у галузі енергозбереження. *Науковий вісник Ужгородського національного університету*, 34, 115-116.
- [2] Важинський, С. Е., Щербак Т. І. (2016). *Методика та організація наукових досліджень: навчальний посібник*, СумДПУ імені А. С. Макаренка, 260 с.
- [3] Карпов, Г. Н. *Инновации в лабораторных работах по теории твердого деформируемого тела* : монография. Москва-Берлин, 2016, 70 с.

Chernoalov O.; Kazymyrenko Y.; Marshak O.; Bobrovskyi O.

Information support of management tasks and organization of educational and scientific laboratory for testing non-metallic materials

Abstract. Ways to increase the efficiency of management and organization of research work of university students through the use of professional information support and implementation of web-technologies were analyzed. The process is considered on the example of a narrow-profile laboratory for testing non-metallic materials, for which the principles of building an information platform are defined.

Keywords: Information support. Web-technologies. Scientific research. Online discussion. Virtual node. education component.

УДК 004.2

MIGRATION OF HIGHLY CUSTOMIZED DESKTOP APPLICATIONS INTO CLOUD SERVICES

Lebed Y.N., engineer

CCSI Limited

Republic of Ireland, Dublin; Spain, Barcelona

yuriy.lebed@ccsi.ie

Abstract. During the development were implemented a few different solution architectures with analysis by the following criteria: final reliability, performance, expandability and customisation. The best results were achieved by architecture with microservices with generation of visual content.

Keywords: solution architecture; software architecture; customisation; microservice.

As the PC era started multiple aspects of businesses were attempted to be automated. The spreadsheets application really made a big push for areas where calculations were straightforward and not too complex. Next step, creation of desktop applications supposed to increase usability and durability. Over time some of them were picked up by big players and we can see accounting, warehousing, CRM, etc. standard applications broadly offered and used by millions of customers which contribute to their profitability.

On the other end, there were a number of applications which evolved into quite a knowledge or science-intensive software product with relatively narrow demand. Where institutional investments have no interest because of the relatively small client base and low cash flow.

But for those whose business niche was continuing to expand a desktop application becomes more and more difficult to maintain and enhance. The obvious solution is to go online [1]. Which solution architecture to choose?

What do we have to deal with? The research is based on software for the selection of air handling units and their components, multilanguage, multicurrency, 24/7 application, with users spread around the globe. Picture 1.



Picture 1. Users' locations.

Apart from this, the application uses circa forty third-party libraries some of them developed more when twenty years ago and have never been updated by the manufacturer. Some of them thread-unsafe.

The common solutions were:

- to convert the desktop app into a thick client with sufficiently reduces overheads for license management;
- thread-unsafe modules were wrapped by messaging queue clients which have eliminated interference between instances processed in parallel;
- the desktop's functionalities were one by one gradually moved to web-based services.
- the web core fully maintains user management and project management where the individual product functionalities were implemented using the following solution architectures;
- the software architecture chosen is MVC;
- where controllers implemented design patterns such as abstract factory, strategy, decorator, interpreter, command [2];
- coding practice follows the SOLID principles [3];
- types of databases used: SQL, no-SQL, blob storage.

The solution architectures were considered.

Standalone web application. An application where all functionality presents in one deployment.

Pros: easy deployment, requiring lesser developers' skills.

Cons: because the app is massive, practically not possible to scale dynamically, high probability for errors affecting all users, each change request requires full redeployment, adding new modules makes it even more massive, difficult to accommodate all third-party libraries inside one environment.

Standalone web application with scalable microservices. An application where performance-critical functionality is moved to easily scalable separate services.

Pros: split deployment, dynamic scalability, the lesser probability of fatal errors, individual containers for each third-party requiring a specific environment.

Cons: difficult to maintain seamless UI customization, still a lot of work to add new functionality.

Web portal with UI microservices. An application where all specific functionality and related UI content are generated on separate dedicated services.

Pros: all from above, plus comfortable and secure customization of UI and functionality.

Cons: more efforts to manage a larger number of software components, require a unified solution for localization.

Conclusions.

Development of the portal with UI microservices responsible for specific functionality allows to deploy independent parts of the application with minimum risk of fatal errors, makes easy deep customization including UI, and utilises scalability to keep performance at the required level.

REFERENCES

- [1] Small, M.J. (2018). *Stress Free Cloud Migration*. RMS Technology Consulting, LLC.
- [2] Gamma, E., Helm, R., Jonson, R., Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.
- [3] Martin, R. (2002). *Agile Software Development, Principles, Patterns, and Practices*. Pearson.

Yuriy N. Lebed

Migration of highly customized desktop application into cloud services

Анотація. У ході роботи було реалізовано декілька архітектур програмного комплексу та зроблено порівняльний аналіз за критеріями: загальна остаточно надійність, продуктивність, придатність до розширення функціональності та налаштування для окремого користувача. Найбільш перспективною виявилися архітектура з використанням мікросервісів з генерацією візуального контенту.

Ключові слова: архітектура рішення; архітектура програмно забезпечення; кастомізація; мікросервіс.

УДК 004.8

ДОСЛІДЖЕННЯ МЕТОДІВ І АЛГОРИТМІВ РОЗПІЗНАВАННЯ МОВИ ТА РОЗРОБКА СИСТЕМИ ГОЛОСОВОГО УПРАВЛІННЯ

Заїченко Т. С.¹, Книрик Н. Р.²

Національний університет кораблебудування імені адмірала Макарова

Україна, Миколаїв

zaichenko.link@gmail.com¹

Анотація. Створення додатку голосового асистента дає змогу полегшити голосове управління комп'ютерами. На даний момент існує низка проблем, з якими зустрічаються користувачі, а саме обмеженість використання голосового асистента в деяких країнах, можливість роботи такого додатку на комп'ютері, погане розуміння мовлення та можливість працювати з додатком без підключення до мережі Internet.

Ключові слова: голосовий асистент; розпізнавання мови; Alexa; Google; Siri; Cortana; Python.

Вступна частина. В ході дослідження було виявлено, що на сьогоднішній час широко використовуються голосові асистенти, адже вони спроможні економити час користувача на пошук інформації в ситуаціях коли потрібна мультизадачність користувача в роботі. На даний момент тема створення та розвитку системи голосового управління є актуальною, оскільки все більше технологій можуть працювати з голосовим помічником, починаючи від мобільних пристроїв та розумної техніки та закінчуючи машинами на автопілоті.

Основна частина. Голосовий помічник є високотехнологічною системою управління, яка може реагувати на голосові команди, тому все більше компаній намагаються створити свого асистента, який приверне увагу їх клієнтів та зможе полегшити їм життя.

Основною особливістю помічника є його розвинутий штучний інтелект та самонавчання, що дозволяє йому розуміти голос користувача та виконувати указані дії. Тобто головною задачею голосового помічника є максимально швидке виконання запитів та їх обробка.

Принцип роботи голосового асистента такий же простий, як і його використання [1, 2]:

1. пасивне зчитування звуку, за допомогою активації функції кодовим словом;
2. фільтрація сигналу – етап фільтрації звуку від шуму та звукових перешкод, котрі з'являються під час запису голосового запиту;
3. оцифровка звуку – виконується перетворення звукового сигналу в цифровий вид, який буде зрозумілий комп'ютеру;
4. аналіз сигналу – виділяються частини з голосом та виконується оцінка параметрів, таких як частина мовлення, форма слова, зв'язок в один запит;
5. пошук шаблонів в даних – штучний інтелект збирає різні раніше почуті слова, порівнює з шаблонами та видає результат.

Таку технологію можна використовувати навіть в психології, оскільки систему можна навчити за допомогою алгоритму виявляти настрій користувача та надавати всі зібранні данні, котрі будуть включати в себе всі тонкощі

Перед створенням голосового асистента необхідно врахувати досить багато нюансів [3]. Для початку варто ознайомитись з популярними системами, а також проаналізувати переваги та недоліки кожної з них. До ключових критеріїв вибору можна віднести:

– сумісність – для кожної операційної системи є свої віртуальні асистенти. Наприклад, для Windows з великим попитом використовується Cortana. Для гаджетів Apple розроблена система Siri, яка має широкий спектр функцій та високу швидкість пошуку інформації. Однією з самих популярних для Android вважається Аліса.

– функціональні можливості – в цьому випадку багато чого залежить від особистих бажань користувача. Одна система може краще виконувати пошук інформації, інша – більш проста в керуванні ПК або смартфоном. Є навіть помічники, які виступають в якості розвинутої навігаційної системи для полегшення подорожей.

– рівень «людяності» – багато систем мають досить високий рівень штучного інтелекту, що на деякому рівні можуть імітувати розмову з живою людиною.

Тож у кожній системі голосового управління є свої можливості, з якими можна ознайомитись в таблиці 1.

Таблиця 1 Функціонал популярних голосових асистентів

Голосовий асистент	Siri	Google Assistant	Cortana	Alexa	Аліса
Компанія представник	Apple	Google	Windows	Amazon	Яндекс
Управління розумним будинком/пристроями	Так	Так	Так	Так	Так
Розуміння людського мовлення	Ні	Так	Так	Так	Так
Доступність в Україні	Так	Так	Ні	Ні	Ні
Можливість взаємодіяти з іншими додатками	Так	Так	Так	Так	Так
Можливість працювати без підключення до мережі Internet	Так	Так	Так	Ні	Ні
Платформи	iOS	Android, iOS	Android, iOS, PC	Android, iOS	Android, iOS, PC

Розібравши самих відомих голосових асистентів, було виявлено, що Аліса та Alexa не можуть використовуватись на території України, наприклад Alexa не має в своєму арсеналі української або російської мови, та має дуже малий процент використання її, як голосового асистента в Україні; Cortana може використовуватись на території України лише тоді, якщо провести деякі маніпуляції з нею; Siri відома тим, що вона працює лише на пристроях виготовлених компанією Apple, що робить її ексклюзивним продуктом, але не поширеним для користувачів котрі використовують пристрої інших компаній; Google Assistant може використовуватись лише на смартфоні, оскільки вона була створена початково для мобільної версії пошукової системи Google.

Тож на даний момент ми не маємо системи голосового управління саме для користувачів комп'ютеру, котрий буде виконувати прості запити користувача для прискорення роботи. Гарним рішенням буде саме створення цієї системи на базі Python, оскільки ця мова програмування дає можливість написати простий та зрозумілий код, що дозволяє розробляти програми для машинного навчання, не замислюючись про особливості синтаксису. Також Python доволі зрозумілий та простий у вивченні та має велику кількість бібліотек, які необхідні

будуть при створенні системи голосового управління. Якщо голосовий помічник буде створений, як окремий додаток на пристрій та зможе працювати без підключення до мережі Internet (окрім сервісів (запитів) які будуть потребувати підключення до мережі) то такий додаток буде працювати не залежачи від серверу виробника, як в більшості компаній, тому робота голосового асистента буде безперебійною.

З розвитком цієї системи стає зрозуміло, що використання цих систем буде мати більш ширший спектр використання, якщо людина буде вести діалог з комп'ютером під час роботи, а саме, стане можливим керувати машиною за допомогою голосу в реальному часі, а також введення та виведення інформації у вигляді звичайної людської мови. Така система матиме змогу зробити користувача мультизадачним, знявши з нього частину обов'язків у вигляді виконання простих завдань, чим прискорить виконання роботи користувача взагалі. Так наприклад більшість браузерів, телефонів, ПК можуть керуватись за допомогою голосу, однак окрім невеличких технологій до яких звикли всі, голосового помічника використовують також і в автомобілях. Тобто під час поїздки водію не потрібно відволікатись від дороги використовуючи запити до голосового асистента, та окрім цього машинне навчання допомогло створити види транспорту, які вже використовують систему голосового управління, тобто під час подорожі, навчений асистент керує машиною не гірше людини та під час своєї роботи може виконувати запити користувача. Голосові асистенти дозволяють нам виконувати прості, повсякденні задачі швидко, не відволікаючись від основного завдання. Така система вже стає невід'ємною частиною нашого життя.

Висновки. У ході роботи були розглянуті принцип роботи типової системи голосового управління та виявлені основні недоліки існуючих голосових асистентів провідних компаній розробників.

Система голосового управління повинна полегшити та прискорити роботу користувачів, а маючи технологію машинного навчання, автоматизувати роботу деяких систем. Користувачі можуть навчати систему самостійно, під час виконання запитів та спілкуванню з голосовим асистентом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Поначугін О. В., Пічужкіна Д. Ю., Смекалова К. С. (2020). Голосовий помічник як технологія обробки даних. *Наука без меж*, 6, 96-98.
- [2] Крогер Д., Люц О., Рашки Ф. (2020). Наслідки аналізу голосу та мовлення для конфіденційності – розкриття інформації шляхом спостереження. *Конфіденційність та ідентифікація*, 242-244.
- [3] Смірнов І. В., Панов О. І., Скриннік О. О., Чистова К. В. (2019). Персональний когнітивний асистент: концепція та принципи роботи. *Інформатика та її застосування*, 13(3), 105-107.

Zaichenko T. S., Knyrik N. R.

Research of language recognition methods and algorithms, and development of voice control system

Abstract. *Creating a voice assistant application makes it easier to voice control computers. At the moment, there are a number of problems that users face, namely the limited use of voice assistant in some countries, the ability to run such an application on a computer, poor speech comprehension and the ability to work with the application without an Internet connection.*

Keywords: *voice assistant; speech recognition; Alexa; Google; Siri; Cortana; Python.*