

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ВИДАВНИЧО-ПОЛІГРАФІЧНИЙ ІНСТИТУТ
Кафедра репрографії

«На правах рукопису»
УДК 004.42;004.05

До захисту допущено:
В. о. завідувача кафедри
_____ Євгеній ШТЕФАН
« ____ » _____ 2020 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра
за освітньо-професійною програмою
«Технології друкованих і електронних видань»
зі спеціальності 186 «Видавництво та поліграфія»
на тему: «Центр з розроблення веб-порталів зі створенням і дослідженням
архітектури системи безперервної інтеграції
для реалізації web-to-print»

Виконав:

студент II курсу, групи РП-91мп
Гуров Олександр Валерійович

Керівник:

доцент кафедри репрографії, к.т.н., доцент
Кушлик Богдан Ростиславович

Консультанти з:

проектної частини

доцент кафедри репрографії, к.т.н., доцент
Скиба Василь Миколайович

розроблення старт-ап проекту

доцент кафедри репрографії, к.т.н., доцент
Розум Тетяна Володимирівна

Рецензент:

доцент кафедри МАПВ, д.т.н., доцент
Шостачук Юрій Олександрович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.
Студент _____

Київ – 2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Видавничо-поліграфічний інститут
Кафедра репрографії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 186 «Видавництво та поліграфія»

Освітньо-професійна програма «Технології друкованих і електронних видань»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

_____ Євгеній ШТЕФАН

«__» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Гурову Олександру Валерійовичу

1. Тема дисертації «Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print», науковий керівник дисертації Кушлик Богдан Ростиславович, к.т.н., доцент, затверджені наказом по університету від «3» листопада 2020 р. №3201-с
2. Термін подання студентом дисертації «10» грудня 2020 р.
3. Об'єкт дослідження: архітектура системи безперервної інтеграції для реалізації web-to-print.
4. Вихідні дані. Вихідними даними до магістерської дисертації має бути огляд сучасного стану та перспектив розвитку технологій, програмного та апаратного забезпечення систем безперервної інтеграції; науково-технічна література та патенти за темою дисертації. Результатом дисертації повинно бути: за результатами досліджень створено архітектуру системи безперервної інтеграції для реалізації web-to-print та сучасний центр з розроблення веб-порталів, що оснащений відповідним апаратним та програмним забезпеченням. Підприємство повинно забезпечити високу якість, рівень автоматизації створених веб-порталів, що відповідають встановленим вимогам.
5. Перелік завдань, які потрібно розробити. Провести аналіз сучасної спеціалізованої літератури, нормативної документації, патентів, а також проаналізувати сучасний стан і перспективи розвитку технології, програмних продуктів і апаратного забезпечення архітектури системи безперервної інтеграції. Визначити фактори, що впливають на будову архітектури системи безперервної інтеграції та вибір базового програмного забезпечення. На підставі об'єкту та предмету дослідження обрати методи та засоби експериментальних досліджень, визначити тестові об'єкти для їх проведення. Провести дослідження та на їх основі створити архітектуру системи безперервної інтеграції для

реалізації web-to-print. Запроєктувати передовий центр з розроблення веб-порталів, який оснащений сучасним програмним та апаратним забезпеченням, а також відповідає нормам проєктування виробничих приміщень з відповідним інженерно-технічним забезпеченням та ефективною інфраструктурою.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: класифікації обладнання, технологій, програмних продуктів – 1–3 рисунки (обов’язково); графіки експериментальних досліджень – 1–5 рисунки (обов’язково); тестові об’єкти (сторінки) – 1–3 рисунки (обов’язково); моделювання технологічного процесу з використанням евристичних методів – 1 рисунок (обов’язково); технологічна схема виробничого процесу – 1–2 рисунки (обов’язково); структурна схема комп’ютеризованої видавничої системи – 1 рисунок (обов’язково); плани дільниць, цехів підприємства – 1–3 рисунки (обов’язково); 3D-модель приміщення 1 рисунок (обов’язково).

7. Орієнтовний перелік публікацій. Опублікувати тези доповідей за темою магістерської дисертації.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
3. Проєктна частина	Скиба В. М., доцент		
4. Розроблення старт-ап проєкту	Розум Т. В., доцент		

9. Дата видачі завдання 02 вересня 2020 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Вступ	до 15.09.2020 р.	
1	Теоретична частина	до 01.10.2020 р.	
2	Експериментальна частина	до 15.10.2020 р.	
3	Проєктна частина	до 01.11.2020 р.	
4	Розроблення старт-ап проєкту	до 15.11.2020 р.	
	Висновки та список використаних джерел	до 01.12.2020 р.	
	Оформлення магістерської дисертації і графічного матеріалу	до 10.12.2020 р.	
	Здавання дисертації на кафедру для рецензування	до 10.12.2020 р.	

Студент

Олександр ГУРОВ

Науковий керівник

Богдан КУШЛИК

АНОТАЦІЯ

Гуров О. В. Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print. – 2020. – 176 с.

Магістерська дисертація на здобуття ступеня магістра зі спеціальності 186 «Видавництво та поліграфія» та освітньої програми «Технології друкованих і електронних видань». – Київський політехнічний інститут імені Ігоря Сікорського, Київ, 2020.

У магістерській дисертації на тему «Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print» проведено дослідження із визначенням факторів впливу на будову архітектури системи безперервної інтеграції та вибір базового програмного забезпечення, на основі чого створено архітектуру системи безперервної інтеграції саме для web-to-print.

У ході роботи над дисертацією було створено систематизації видів систем безперервної інтеграції та метрик ефективності. На основі патентного пошуку визначено тенденції у сфері архітектури CI/CD. Побудовано технологічну блок-схему виробничого процесу створення веб-порталу, розроблено організаційну структуру виробництва. На основі промислового завдання було здійснено технологічні розрахунки, розроблено комп'ютеризовану видавничу систему із необхідною кількістю робочих станцій та побудовано плани центру з розроблення веб-порталів та його 3Д-модель.

Створено старт-ап проєкт центру з розроблення веб-порталів, де описано ідею проєкту та проаналізовано ринкові можливості запуску старт-ап проєкту.

Ключові слова: АРХІТЕКТУРА СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ, CI/CD, WEB-TO-PRINT, ЦЕНТР, ВЕБ-ПОРТАЛ, ПРОЄКТУВАННЯ, ТЕХНОЛОГІЧНИЙ ПРОЦЕС, ОРГАНІЗАЦІЯ, АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ДОСЛІДЖЕННЯ, ТЕХНОЛОГІЧНИЙ ПЛАН, ПРОЄКТ, СИСТЕМАТИЗАЦІЯ.

ANNOTATION

Hurov O. V. The center creating web-portals with setting up and research of the continuous integration architecture to realize the web-to-print concept. – 2020. – 176 p.

Master's dissertation for Master's Degree in Specialty 186 "Publishing and Printing" and educational program "Technologies of Printed and Electronic Editions". – Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2020.

The master's dissertation on "The center creating web-portals with setting up and research of the continuous integration architecture to realize the web-to-print concept" conducted research to determine the factors influencing the architecture of the system of continuous integration and selection of basic software architecture of the system of continuous integration for web-to-print.

To accomplish these tasks, the systematization of types of continuous integration systems and efficiency metrics was created. Based on the patent search, trends in the field of CI/CD architecture have been identified. The technological block diagram of the production process of creating a web portal is built, the organizational structure of production is developed. On the basis of the industrial task, technological calculations were made, a computerized publishing system with the required number of workstations was developed, and plans of the center for the development of web portals and its 3D model were built.

The start-up project of the web portal development center has been created, where the project idea is described and the market opportunities of starting a start-up project are analyzed.

Keyword list: CONTINUOUS INTEGRATION SYSTEM ARCHITECTURE, CI/CD, WEB-TO-PRINT, CENTER, WEB PORTAL, DESIGNING, TECHNOLOGICAL PROCESS, ORGANIZATION, HARDWARE, SOFTWARE, RESEARCH, TECHNOLOGY PLAN, PROJECT, ORGANIZING.

РЕФЕРАТ

Пояснювальна записка складається з: 176 сторінок, 67 рисунків, 66 таблиць, 14 формул, 78 літературних джерел та 11 додатків.

Тема магістерської дисертації – «Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print».

АКТУАЛЬНІСТЬ ТЕМИ. Дана тема є *актуальною*, оскільки аналіз потреб замовників веб-порталів показує, що очікується висока якість кінцевого продукту, яка виражається у відсутності помилок в роботі коду, швидкості роботи та можливості витримувати пікові навантаження, швидкості циклу зворотного зв'язку. Всі ці вимоги можуть бути виконані використанням системи безперервної інтеграції з правильно створеною архітектурою.

МЕТА ДОСЛІДЖЕННЯ. Створення продуктивної та ефективної архітектури системи безперервної інтеграції для веб-порталів типу web-to-print з визначенням факторів впливу на будову та вибір інструментів.

Об'єктом дослідження є: архітектура системи безперервної інтеграції для реалізації web-to-print.

Предметом дослідження є: фактори, що впливають на будову архітектури системи безперервної інтеграції та вибір базового програмного забезпечення.

МЕТОДИ ДОСЛІДЖЕННЯ. Для досягнення поставленої мети буде визначено та проаналізовано всі чинники, що впливають на будову архітектури системи безперервної інтеграції. Використано діаграму Парето, теоретичний метод на основі теорії графів, проведено патентний пошук, аналітичне дослідження; систематизовано інформацію; залучено евристичні та експериментальні методи аналізу. Розроблено програмний код для виконання експериментів та за допомогою спеціальних скриптів виконано вимірювання метрик.

НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ. Досліджено фактори впливу на архітектуру системи безперервної інтеграції та її

масштабованість під навантаженнями. Створено архітектуру системи безперервної інтеграції саме для веб-порталів типу web-to-print.

ПРАКТИЧНА ЦІННІСТЬ РЕЗУЛЬТАТІВ. Полягає в проєктуванні підприємства, що займається виготовленням веб-порталів web-to-print з використанням досліджень системи безперервної інтеграції, що дозволить пришвидшити процес створення веб-порталів, скоротить час знаходження та виправлення помилок, підвищить захищеність. Отримані результати дослідження, а саме створена архітектура системи безперервної інтеграції для web-to-print, може також використовуватися як основа для побудови архітектури для інших типів веб-порталів зі схожими характеристиками.

АПРОБАЦІЯ РЕЗУЛЬТАТІВ. Основні результати роботи представлено на XIX та XX Міжнародній науково-технічній конференції студентів та аспірантів «Друкарство молоде» НТУУ «КПІ ім. Ігоря Сікорського» (Київ, 2019 та 2020).

Ключові слова: АРХІТЕКТУРА СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ, CI/CD, WEB-TO-PRINT, ЦЕНТР, ВЕБ-ПОРТАЛ, ПРОЄКТУВАННЯ, ТЕХНОЛОГІЧНИЙ ПРОЦЕС, ОРГАНІЗАЦІЯ, АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ДОСЛІДЖЕННЯ, ТЕХНОЛОГІЧНИЙ ПЛАН, ПРОЄКТ, СИСТЕМАТИЗАЦІЯ.

ABSTRACT

The explanatory note consists of 176 pages, 67 figures, 66 tables, 14 formulas, 78 references, and 11 appendices.

The topic of the master's dissertation is "The center creating web-portals with setting up and research of the continuous integration architecture to realize the web-to-print concept".

RELEVANCE OF THE TOPIC. This topic is relevant because the analysis of the needs of web portals customers shows that the expected high quality of the final product, which is expressed in the absence of errors in the code, speed and ability to withstand peak loads, speed of the feedback cycle. All these requirements can be met using a system of continuous integration with a properly designed architecture.

THE AIM OF THE STUDY. Creating a productive and efficient architecture of the system of continuous integration for web portals such as web-to-print with the definition of factors influencing the structure and choice of tools.

The object of research is the architecture of the system of continuous integration for the implementation of web-to-print.

The subject of the study are factors influencing the architecture of the system of continuous integration and the choice of basic software.

RESEARCH METHODS. To achieve this goal, all factors influencing the architecture of the system of continuous integration will be identified and analyzed. Pareto diagram, theoretical method based on graph theory, patent search, analytical research; systematized information; heuristic and experimental methods of analysis are involved. The program code for performance of experiments is developed and by means of special scripts measurement of metrics is executed.

SCIENTIFIC NOVELTY OF THE RESULTS OBTAINED. The factors influencing the architecture of the system of continuous integration and its scalability under loads are investigated. The architecture of the system of continuous integration has been created for web-type web portals.

PRACTICAL VALUE OF RESULTS. It consists in designing an enterprise engaged in the production of web-to-print web portals using research of the system of continuous integration, which will speed up the process of creating web portals, reduce the time of finding and correcting errors, increase security. The obtained research results, namely the created architecture of the system of continuous integration for web-to-print, can also be used as a basis for building an architecture for other types of web portals with similar characteristics.

APPROVAL OF RESULTS. The main results of the work were presented at the XIX and XX International Scientific and Technical Conference of Students and Postgraduates "Young Printing" NTUU "KPI Igor Sikorsky"(Kyiv, 2019 and 2020).

Keyword list: CONTINUOUS INTEGRATION SYSTEM ARCHITECTURE, CI/CD, WEB-TO-PRINT, CENTER, WEB PORTAL, DESIGNING, TECHNOLOGICAL PROCESS, ORGANIZATION, HARDWARE, SOFTWARE, RESEARCH, TECHNOLOGY PLAN, PROJECT, ORGANIZING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	12
ВСТУП.....	14
РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА	17
1.1 Аналітичний огляд сучасного стану систем безперервної інтеграції ...	17
1.1.1 Аналіз технології безперервної інтеграції.....	17
1.1.2 Класифікація систем безперервної інтеграції, їх особливості	19
1.1.3 Типова архітектура, варіації та фактори впливу	21
1.1.4 Інструменти для забезпечення безперервної інтеграції.....	24
1.2 Чинники, що впливають на якість архітектури системи безперервної інтеграції	29
1.3 Предмет і регламент патентного пошуку	32
1.4 Завдання дослідження.....	44
РОЗДІЛ 2. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	45
2.1 Тенденції розвитку архітектури систем безперервної інтеграції за результатами патентного пошуку.....	45
2.2 Об'єкт та предмет дослідження.....	49
2.3 Розроблення тестових файлів	49
2.4 Методика проведення експерименту та оцінювання результатів дослідження	52
2.4.1 Оцінювання якості базового програмного забезпечення	52
2.4.2 Дослідження впливу навантаження на масштабування серверної частини за різної структури та інструментів.....	53
2.5 Результати досліджень	55
2.4.1 Оцінювання якості базового програмного забезпечення	55
2.4.2 Дослідження впливу навантаження на масштабування серверної частини за різної структури та інструментів.....	61
2.6 Моделювання технологічного процесу з урахуванням результатів дослідження	72
РОЗДІЛ 3. ПРОЄКТНА ЧАСТИНА	77
3.1 Проєктування інженерно-технічного забезпечення виробництва	77

	11
3.1.1 Промислове завдання на розробку проєкту	77
3.1.2 Вибір технології та структури виробничих процесів.....	83
3.1.3 Принципові рішення щодо автоматизації технологічного процесу	91
3.1.3.1 Вибір апаратно-програмного забезпечення	91
3.1.3.2 Організаційна структура виробництва	98
3.1.3.3 Основні характеристики проєкту та його цілі	99
3.1.4 Розрахунок розгорнутого промислового завдання.....	102
3.1.5 Розрахунок обсягу виробництва, трудомісткості робіт, необхідної кількості устаткування та робочих місць, кількості працюючих.....	104
3.1.6 Виробничо-технологічні плани виробничих приміщень.....	109
3.2 Завдання на інженерно-технічного забезпечення виробництва.....	111
3.2.1 Проєктування конструкцій перекриття та шумоізоляції виробничих приміщень	111
3.2.2 Розроблення ескізних креслень і 3Д-моделі генерального плану центру з розроблення веб-порталів.....	112
3.2.3 Складання завдання на інженерно-технічне забезпечення виробництва.....	114
3.2.4 Завдання на комп'ютерне забезпечення виробництва.....	116
3.3 Техніко-економічні показники проєкту.....	119
3.4 Принципові рішення щодо розроблення технологічної системи.....	125
РОЗДІЛ 4. РОЗРОБЛЕННЯ СТАРТ-АП ПРОЄКТУ	129
4.1 Опис ідеї проєкту	129
4.2 Технологічний аудит ідеї проєкту.....	131
4.3 Аналіз ринкових можливостей запуску старт-ап проєкту.....	132
4.4 Розроблення ринкової стратегії проєкту	139
4.5 Розроблення маркетингової програми старт-ап проєкту.....	141
ВИСНОВКИ.....	145
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	147
ДОДАТКИ.....	156

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

бібліотека (програмна) – збірка об'єктів чи підпрограм для вирішення близьких за тематикою задач

веб-портал – сайт, що надає користувачеві інтернету різні інтерактивні сервіси, які працюють у рамках єдиного сайту

КВС – комп'ютеризована видавнича система

логування – запис інформації про виконання програми

ЛОМ – локальна обчислювальна мережа

месенджер – телекомунікаційна служба для обміну текстовими повідомленнями між комп'ютерами або іншими пристроями користувачів через комп'ютерні мережі

ОЗП – оперативний запам'ятовувальний пристрій

ПЗП – постійний запам'ятовувальний пристрій

ПК – персональний комп'ютер

плагін – додаток, незалежно скомпільований програмний модуль, що динамічно підключається до основної програми, призначений для розширення або використання її можливостей

СУБД – система управління базами даних

ТЗ – технічне завдання

фреймворк – заготовки для програмної платформи, що визначають каркас програмної системи

API – опис способів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою

AWS – компанія та хмарне середовище Amazon Web Services

back-end – програмно-апаратна частина сервісу

Bootstrap – вільний набір інструментів для створення front-end частини сайтів

CD – безперервна доставка чи розгортання

CI – безперервна інтеграція

CMS – ядро, програмне забезпечення для створення, редагування, організації структури і управління веб-сайтом

docker – інструментарій для управління ізольованими Linux-контейнерами

front-end – клієнтська сторона призначеного для користувача інтерфейсу до програмно-апаратної частини сервісу

front-end сервер – сервер, який відповідає на статичні запити користувачів та виконує функції reverse проху, зазвичай використовують nginx, caddy, lighttpd

javascript – динамічна, об'єктно-орієнтована прототипна мова програмування

jQuery – бібліотека javascript з великою кількістю функцій

nodejs – програмна платформа для програмування за допомогою javascript на серверній частині

ORM – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних»

SEO-оптимізація – оптимізація веб-порталу для пошукових систем

UI – інтерфейс користувача

UX – дизайн взаємодії з користувачем

XSS – тип вразливості інтерактивних інформаційних систем у вебi

ВСТУП

У сучасній економіці використання цифрових інструментів в бізнес-рішеннях грає визначальну роль. Разом зі зростанням складності високотехнологічних платформ важливим фактором стає безперебійність роботи та постійні оновлення з метою дотримання безпекових заходів критично важливих ІТ-систем. Торкнулася ця тенденція і розробки програмного забезпечення для порталів типу web-to-print.

Безперервна інтеграція дає швидкий цикл зворотного зв'язку, що дозволяє майже миттєво визначити, наскільки якісними є зміни в код і функціональність продукту. При waterfall-підході можна швидко вносити зміни в код, але без постійних перевірок виявлення помилок потребує набагато більше часу і може статися вже після початку комерційної експлуатації, причому таких «відкладених» помилок в одному релізі може виявитися декілька.

З кожним днем кількість веб-порталів, які використовують даний підхід та технології зростає. Майже всі сучасні, особливо, великі за обсягом проєкти, використовують безперервну інтеграцію. При проєктуванні нового web-to-print веб-порталу буде дивно не використовувати її.

Отже, сучасні тенденції диктують використання засобів автоматизації тестування задля скорочення часу розробки, зменшення безпекових помилок та зручного постійного оновлення програмного коду продуктів.

Актуальність теми полягає в створенні сучасної та ефективної архітектури системи безперервної інтеграції. Адже якість кінцевого продукту, яка виражається у відсутності помилок в роботі коду, швидкість роботи та можливість витримувати пікові навантаження, швидкість циклу зворотного зв'язку суттєво впливають на успіх поліграфічного підприємства. Це ще раз доводить необхідність дослідження, вивчення та обґрунтування архітектури.

Завданням магістерської дисертації є проєктування прогресивного центру з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print. Використання

доцільної та сучасної архітектури дозволить значно підвищити якість кінцевого продукту – веб-порталу, а також скоротити цикл оновлення.

Мета роботи полягає у створенні продуктивної та ефективної архітектури системи безперервної інтеграції для веб-порталів типу web-to-print з визначенням факторів впливу на будову та вибір інструментів. Використовуючи сучасні комп'ютеризовані системи з їх інженерно-організаційним, апаратно-програмним, матеріально-технічним та інформаційним забезпеченням необхідно: схарактеризувати організаційну схему та діяльність центру, розрахувати промислове та розгорнуте промислове завдання, визначити виробничо-технічні характеристики веб-порталів, запроєктувати технологічний процес створення веб-порталів та блок-схему виробничо-технічних процесів, виконати розрахунок обсягу виробництва, устаткування, матеріалів, штату працівників та робочих місць, встановити техніко-економічні показники проєкту, створити виробничо-технологічні плани.

Об'єктом дослідження є: архітектура системи безперервної інтеграції для реалізації web-to-print.

Предметом дослідження є: фактори, що впливають на будову архітектури системи безперервної інтеграції та вибір базового програмного забезпечення.

Методи дослідження: визначення та аналіз всіх чинників, що впливають на будову архітектури системи безперервної інтеграції. Використано діаграму Парето, теоретичний метод на основі теорії графів, проведено патентний пошук, аналітичне дослідження; систематизовано інформацію; залучено евристичні та експериментальні методи аналізу. Розроблено програмний код для виконання експериментів та за допомогою спеціальних скриптів виконано вимірювання метрик.

Наукова новизна одержаних результатів: досліджено фактори впливу на архітектуру системи безперервної інтеграції та її масштабованість під навантаженнями. Створено архітектуру системи безперервної інтеграції саме для веб-порталів типу web-to-print.

Практична цінність результатів: полягає в проектуванні підприємства, що займається виготовленням веб-порталів web-to-print з використанням досліджень системи безперервної інтеграції, що дозволить пришвидшити процес створення веб-порталів, скоротить час знаходження та виправлення помилок, підвищить захищеність. Отримані результати дослідження, а саме створена архітектура системи безперервної інтеграції для web-to-print, може також використовуватися як основа для побудови архітектури для інших типів веб-порталів зі схожими характеристиками.

Апробація результатів: основні результати роботи представлено на ХІХ та ХХ Міжнародній науково-технічній конференції студентів та аспірантів «Друкарство молоде» НТУУ «КПІ ім. Ігоря Сікорського» (Київ, 2019 та 2020).

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналітичний огляд сучасного стану систем безперервної інтеграції

1.1.1 Аналіз технології безперервної інтеграції

Безперервна інтеграція – практика розробки програмного забезпечення, що полягає у частих складаннях проєкту з метою виявлення та вирішення інтеграційних і безпекових проблем. Також це дозволяє швидко запускати новий функціонал, зменшити витрати на тестувальників, легко брати нових членів у команду.

Варто зазначити, що термін «безперервна інтеграція» може використовуватись як в загальному значенні (практика розробки програмного забезпечення), так і в значенні одного з етапів автоматизації – автоматична збірка і тестування.

Є багато різних інструментів та архітектур систем безперервної інтеграції. Зазвичай їх розподіляють по рівнях цієї (автоматичної) інтеграції: лише для перевірки правильності нового коду та його інтегрованості або повного циклу (від тестування коду до розгортання на робочих серверах). Також за способом та умовах, в яких працюють такі системи можна поділити на хмарні та такі, що працюють на серверах компанії (власних чи арендованих).

Вибирають технології та архітектури в переважній більшості в залежності від таких факторів:

- кількості зліпків зі змінами коду, що надходять до системи контролю версій за певний проміжок часу;
- доступності (SLA);
- продуктивності та часу розгортання на серверах;
- можливості масштабування (scalability);
- фінансових затратах на апаратне забезпечення (власне чи хмарне) та підтримку працездатності системи.

Розглянемо детально ці технології.

Хмарні технології CI/CD дозволяють за допомогою одного чи декількох файлів-конфігів проводити операції ланцюгу послідовно або паралельно, виконуючи спочатку збірку проєкту, потім тестування та записуючи весь перебіг операції у спеціальні файли (логи), які потім спеціаліст може аналізувати в приємному веб-інтерфейсі. Окрім цього, вони показують всю історію збірок, помилки що виникли, статистику та багато чого іншого без будь-яких спеціальних налаштувань. Основні переваги – це швидко, просто, надійно, але серед недоліків – інколи (в залежності від сервісу) дорого та не має повного контролю над процесом.

Розгортання ж CI/CD на власних чи арендованих серверах значно складніше та потребує більше зусиль, часу, але дозволяє мати компанії повний контроль над даними та кодом, як завгодно налаштовувати.

Загалом ж процес безперервної інтеграції виглядає так, як зображено на рисунку 1.1.

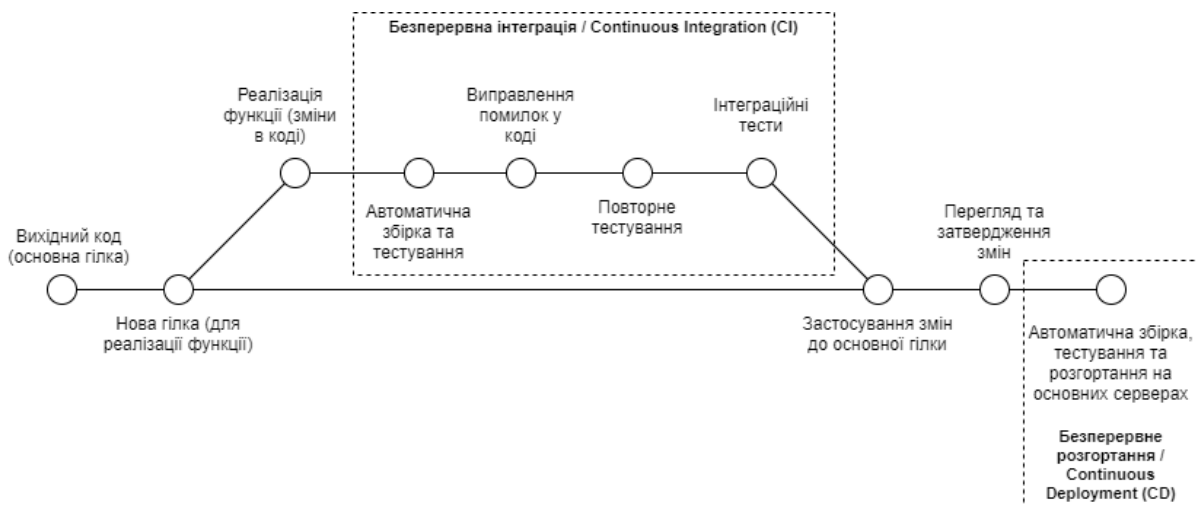


Рисунок 1.1 – Загальна схема процесу безперервної інтеграції

Необхідність самої безперервної інтеграції визначається бізнес-процесами компанії, однак всі сучасні тенденції свідчать про швидке тестування гіпотез, швидкий запуск нових функцій чи продуктів, а без автоматичного процесу збірки та тестування важко уявити це. Також ще, одним з головних факторів, являється якість кінцевого продукту, яка значно вища, а кількість критичних безпекових помилок значно менша [1].

1.1.2 Класифікація систем безперервної інтеграції, їх особливості

CI/CD системи можна поділити за різними ознаками на велику кількість підгруп, але основних чотири:

- спосіб конфігурації;
- середовище роботи;
- ступінь автоматизації;
- призначення.

Класифікацію наведено на рисунку 1.2 [2-4].

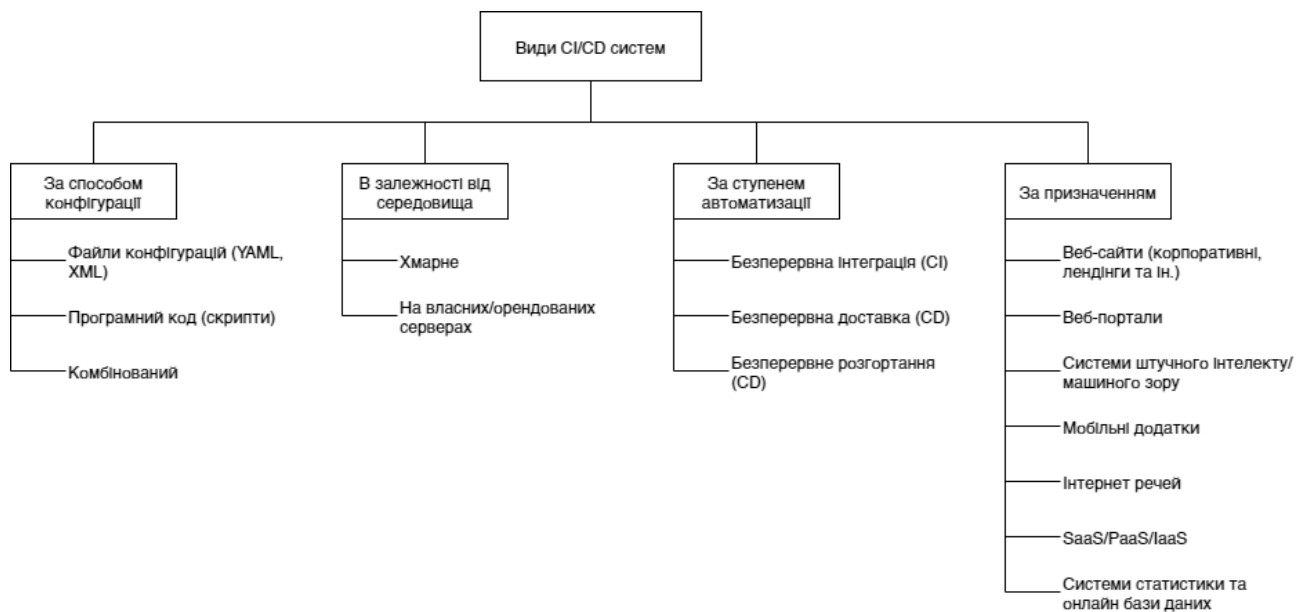


Рисунок 1.2 – Класифікація CI/CD систем

Конфігурація за допомогою файлів конфігурацій є найбільш поширеною, адже вона використовується для хмарних систем, а файли конфігурацій зберігаються в системі контролю версій в окремій директорії. Але, наприклад, Jenkins використовує конфігурацію за допомогою коду на мові програмування Groovy.

Якщо порівнювати формати YAML та XML, то перший набагато менш багатослівний. Відношення сигнал / шум вище без всіх дужок. Це суб'єктивно полегшує читання і редагування для багатьох людей. З іншого боку, це дещо складніше для аналізу. Однак найбільша відмінність полягає в тому, що XML

призначений для мови розмітки, а YAML – це скоріше формат даних. Подання простих, ієрархічних даних, як правило, виконується більш витончено в YAML, але фактичний розмічений текст представляти незручно.

Розгортають системи безперервної інтеграції на власних (в т.ч. арендованих) серверах доволі рідко. Набагато зручніше і простіше мати підписку на хмарний сервіс. Однак, якщо є спеціалісти, що можуть виконати налаштування – це допоможе зберегти кошти.

Є три ступеня автоматизації, причому кожен наступний включає в себе попередній.

При безперервній інтеграції розробники якнайчастіше зливають свої зміни з основною гілкою коду. Зміни розробника перевіряються шляхом створення збірки та запуску автоматизованих тестів для збірки. Це робиться для уникнення проблем інтеграції, які можуть трапитися під час оновлення робочих серверів.

Постійна інтеграція робить великий акцент на автоматизації тестування, щоб перевірити, чи працює програма, коли нові зміни інтегруються в основну гілку.

Безперервна доставка – це продовження безперервної інтеграції, оскільки вона автоматично розміщує всі зміни коду в тестовому та / або робочому середовищі після етапу збірки.

Це означає, що крім автоматизованого тестування, є автоматизований процес випуску, і є можливість розгорнути зміни в будь-який час, натиснувши кнопку.

Безперервне розгортання йде на крок далі, ніж безперервна доставка. Завдяки цій практиці кожна зміна, яка проходить усі етапи виробничого циклу, передається клієнтам. Тут немає втручання людини, і лише невдалий тест не дозволить застосувати нові зміни до робочих серверів.

Постійне розгортання – це чудовий спосіб пришвидшити цикл зворотного зв'язку зі своїми клієнтами та зняти тиск з боку команди, оскільки дня випуску вже немає. Розробники можуть зосередитись на створенні програмного

забезпечення, і вони бачать, як їх робота починає діяти через кілька хвилин після закінчення роботи над ним.

В залежності від призначення системи безперервної інтеграції, архітектура може мати свої особливості. В даній роботі розглядаються лише загальні архітектури, чи для веб-порталів типу web-to-print.

1.1.3 Типова архітектура, варіації та фактори впливу

Під архітектурою системи безперервної інтеграції розуміють не лише конфігурації та обрані інструменти, а також процес роботи з цією системою загалом, інтеграції із зовнішніми сервісами та ін.

Типова архітектура системи безперервної інтеграції з усіма основними компонентами, зв'язками між ними та етапами ланцюгу збірки й тестування зображено на рисунку 1.3.

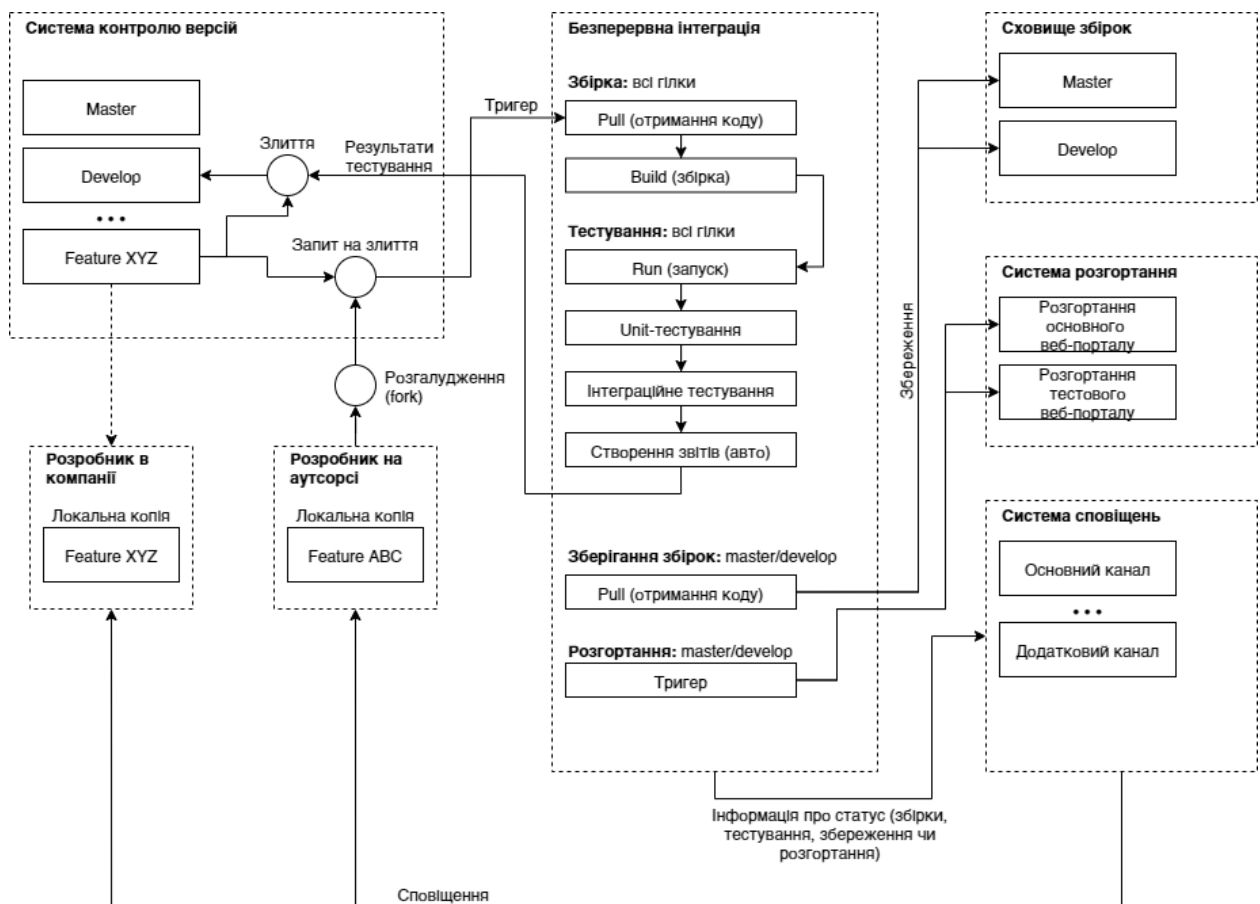


Рисунок 1.3 – Типова загальна архітектура системи безперервної інтеграції

Як видно зі схеми, дуже важливим є зв'язки системи безперервної інтеграції з системою контролю версій, сховищем збірок, системою розгортання та сповіщень. Дуже важливим є максимальна інтеграція з системою контролю версій, адже розробникам потрібно бачити які саме зміни в коді (зліпок зі змінами) не пройшли тести та виправляти згідно цієї інформації.

Незважаючи на те, що зображена на рисунку вище архітектура є найбільш часто вживаною – є і різновиди.

Доволі часто створюють архітектуру, за якої сховище збірок поєднане з системою розгортання, де першим етапом є збірка сирцевого коду. Такий зв'язок має деякі переваги, наприклад, для великих систем (при великій кількості збірок), або дозволяє застосовувати різні степені оптимізації під час виконання збірки (тестування потребує швидкої збірки, а під час розгортання вже більш важлива продуктивність роботи).

Система сповіщень може бути поєднана із системою контролю версій, яка знаходиться в хмарному середовищі та розробник бачить всі результати не через спеціальні сповіщення на електронну пошту чи месенджер, а у веб-інтерфейсі.

Часто, для великих систем, окрім інтеграційних тестів потрібно виконати тестування спеціалістом, тобто неможливо перейти відразу до етапу розгортання. Для цього роблять проміжний крок, де після успішного автоматичного тестування розгортання проводиться лише після натискання спеціальної кнопки в веб-інтерфейсі чи іншим способом.

Варто зазначити, що не обов'язково повинні бути розробники на аутсорсі, як це зображено на схемі. Процес роботи з кодом для молодших розробників також потрібно організовувати через запити на злиття, адже після такого запиту зміни в коді повинен перевірити розробник з більшим досвідом, який відповідає за певний функціонал.

Також було виокремлено ряд основних факторів, які мають безпосередній вплив на архітектуру системи безперервної інтеграції, а саме: доступність (Д), продуктивність (П), масштабування серверної частини (М), вартість підтримки системи (ВП), витрати на тестування (ВТ), час введення нового функціоналу (Ч),

кваліфікація технічних спеціалістів (КВ). Для оцінювання вагомості кожного фактора (показника) було використано метод експертних оцінок [5-8]. Сума балів, що були виставлені групою експертів є показником порівняння рівня пріоритету для безперервної інтеграції. Для опитування було залучено 9 осіб - спеціалісти з видавничо-поліграфічної справи, DevOps-и (спеціалісти безперервної інтеграції), результати занесені до таблиці 1.1. Для візуального представлення результатів експертних оцінок було побудовано діаграму Парето, яку представлено на рисунку 1.4.

Таблиця 1.1 – Результати експертної оцінки факторів основних впливу

Xi	Xj							$\sum a_j$	Вага параметру
	ВТ	М	ВП	КВ	Д	Ч	П		
ВТ	9,00	7,50	4,50	6,00	4,50	6,00	4,50	42,00	0,09
М	10,50	9,00	10,50	7,50	6,00	12,00	6,00	61,50	0,14
ВП	9,00	7,50	9,00	7,50	6,00	10,50	6,00	60,00	0,14
КВ	12,00	10,50	10,50	9,00	6,00	13,50	7,50	69,00	0,16
Д	13,50	12,00	12,00	12,00	9,00	13,50	10,50	82,50	0,19
Ч	12,00	6,00	7,50	4,50	4,50	9,00	4,50	48,00	0,11
П	13,59	12,00	12,00	10,50	7,50	13,50	9,00	78,00	0,18
$\sum a_j$								441,00	1,00

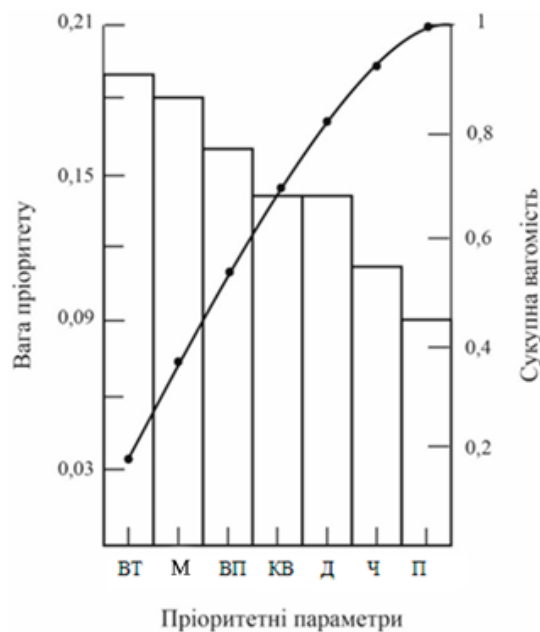


Рисунок 1.4 – Діаграма Парето для експертної оцінки основних факторів, що впливають на архітектуру системи безперервної інтеграції

Пояснення до рисунку 1.4:

Д – доступність, П – продуктивність, М – масштабування серверної частини, ВП – вартість підтримки системи, ВТ – витрати на тестування, Ч – час введення нового функціоналу, КВ – кваліфікація технічних спеціалістів.

1.1.4 Інструменти для забезпечення безперервної інтеграції

Для використання систем безперервної інтеграції необхідно декілька базових інструментів, без яких не можливо уявити їх роботу. Одним із таких інструментів є система контролю версій програмного коду.

Контроль версій – це процес управління програмним кодом та його змінами, який надає багато можливостей, таких як [9]:

- підтримка кількох версій коду;
- можливість повернутися до будь-якої попередньої версії;
- розробники можуть працювати паралельно;
- аудит з чітким уявленням про те, хто, як, коли, де і які зміни зробив;
- синхронізація коду;
- копіювання, об'єднання, скасування змін;
- з'ясування різниці між версіями;
- забезпечує повне резервне копіювання, не займаючи багато місця;
- перегляд історії змін;
- можливий як для малих, так і для великих проєктів;
- можливість ділитися та працювати над кодом по всьому світу.

Отже, система контролю версій дозволяє керувати та відстежувати весь сирцевий код, а також показує те, що всі зміни зберігаються у сховищі.

Є 3 основні системи: Git, Mercurial та SVN [9-10]. Порівняння кожної з них наведено нижче.

Git був створений Лінусом Торвальдсом у 2005 році. Він написаний на мовах C, Shell, Perl, TCL (мова команд інструментів) та Python [10].

Можливості:

- система контролю версій з відкритим кодом;

- забезпечує потужну підтримку для нелінійного розвитку;
- модель розподіленого сховища;
- сумісна із існуючими системами та протоколами, такими як HTTP, FTP, ssh;
- здатна ефективно обробляти малі та великі проєкти;
- криптографічна автентифікація історії;
- періодична явна упаковка об'єктів;
- сміття накопичується до збирання.

Переваги:

- надшвидка та ефективна робота;
- крос-платформне рішення;
- зміни коду можна дуже легко і чітко відстежувати;
- легко обслуговується та багатофункціональна;
- пропонує дивовижну утиліту командного рядка, відому як git bash;
- також пропонує графічний інтерфейс, де можливо дуже швидко повторно відсканувати, змінити стан, підписатись, зафіксувати та швидко закріпити зміни (коміт) код лише за кілька кліків.

Недоліки:

- складний і більший журнал історії стає важким для розуміння;
- не підтримує розширення ключових слів та збереження міток часу.

Mercurial був створений незадовго до Git, а також є розподіленою одноранговою системою. Mercurial - розподілений інструмент керування версіями, який написаний переважно на Python та деяких невеликих розділах мовою C [10].

Можливості:

- висока продуктивність та масштабованість;
- розширені можливості розгалуження та злиття;
- повністю розподілений спільний розвиток;
- децентралізований;
- надійно обробляє як звичайний текст, так і двійкові файли;

- має інтегрований веб-інтерфейс.

Переваги:

- швидкий і потужний;
- зрозумілий та з гарною документацією;
- легкий і портативний;
- концептуально простий.

Недоліки:

- часткові переходи заборонені;
- досить проблематично при використанні з додатковими розширеннями.

SVN – це безкоштовне програмне забезпечення з відкритим кодом під ліцензією Apache. Написано мовою C. SVN був створений як альтернатива CVS (система одночасних версій) з метою виправити деякі помилки та зробити його більш надійним у використанні [10].

Можливості:

- модель сховища клієнт-сервер. Однак SVN дозволяє мати розподілені гілки;
- каталоги мають версії;
- операції копіювання, видалення, переміщення та перейменування також мають версії;
- підтримує атомарні коміти;
- версійні символічні посилання;
- версійні метадані у вільній формі;
- ефективне двійкове диференційне зберігання;
- розгалуження не залежить від розміру файлу, і це дешева операція;
- відстеження злиття, повна підтримка MIME, авторизація на основі шляху, блокування файлів, автономна робота сервера.

Переваги:

- має перевагу завдяки хорошим інструментам графічного інтерфейсу, таким як TortoiseSVN;

- підтримує порожні каталоги;
- має кращу підтримку Windows у порівнянні з Git;
- простота налаштування та адміністрування.

Недоліки:

- досі містить помилки, пов'язані з перейменуванням файлів та каталогів;
- недостатня кількість команд управління сховищем;
- повільна порівняльна швидкість;
- не підтримує підписані коміти.

Другим, надзвичайно важливим інструментом, є базове програмне забезпечення, на основі якого будуються конфігурації системи безперервної інтеграції. Воно отримується від різних постачальників, таких як CircleCI, Bamboo, Jenkins, Travis CI, GitHub Actions та ін. Нижче наведено короткий опис основних.

Jenkins - це проєкт з відкритим кодом, написаний на Java, який працює на Windows, macOS та інших Unix-подібних операційних системах. Він безкоштовний, підтримується громадою та може бути інструментом першочергового вибору для постійної інтеграції. В основному розгортається локально, Jenkins також може працювати на хмарних серверах. Його інтеграція з Docker та Kubernetes використовує переваги контейнерів та виконує ще частіші випуски.

Jenkins може інтегруватися практично з будь-якою зовнішньою програмою, яка використовується для розробки програмного забезпечення. Це дозволяє використовувати контейнерні технології, такі як Docker та Kubernetes нестандартно. G2 Crowd стверджують: «Не існує кращого інструменту для інтеграції ваших сховищ та баз коду з інфраструктурою розгортання».

TeamCity від JetBrains - надійний та якісний сервер CI. Команди часто вибирають TeamCity для великої кількості функцій автентифікації, розгортання та тестування нестандартних функцій, а також підтримки Docker. Він є крос-платформним, підтримує всі останні версії Windows, Linux та macOS і працює з

Solaris, FreeBSD та ін. TeamCity працює відразу після встановлення, додаткові налаштування не потрібні. Він може похвалитися низкою унікальних функцій, таких як докладні звіти про історію, миттєвий відгук про невдалі тести та повторне використання налаштувань, тому ніколи не доведеться дублювати код заради цього.

Як одне з найстаріших рішень, Travis завоював довіру багатьох користувачів.

Travis CI може тестувати на Linux та macOS. Тим часом його документація попереджає, що ввімкнення тестування на декількох ОС може призвести до того, що деякі інструменти чи мови стануть недоступними, або призведе до помилок тестування через особливості поведінки кожної файлової системи.

Пропонуючи багато автоматизованих опцій CI, Travis усуває необхідність у виділеному сервері, оскільки він розміщений у хмарі. Однак він також має локальний продукт для компаній, які хочуть і надалі використовувати ті самі функції інструментів IT, що відповідають потребам безпеки на місці.

CircleCI - це гнучкий інструмент CI, який пропонує аж до 16-кратного розпаралелювання. Він розумно повідомляє користувачів, які надають лише відповідну інформацію по електронній пошті. CircleCI забезпечує просте налаштування та обслуговування.

CircleCI - це хмарна система, яка також пропонує комплексне рішення із захистом та конфігурацією для запуску інструменту у приватній хмарі чи центрі обробки даних.

CodeShip від CloudBees дає повний контроль над налаштуванням та оптимізацією робочого циклу CI та CD. Цей інструмент CI допомагає керувати командою та впорядковувати проекти. За допомогою паралельних конвеєрів, одночасної збірки та кешування CodeShip дозволяє встановлювати потужні конвеєри розгортання, які дозволяють легко розгортати кілька разів на день.

Останнім із популярних є GitHub Actions, який спрощує автоматизацію створення, тестування та розгортання проектів на будь-якій платформі, включаючи Linux, macOS та Windows. Безпроблемний запуск робочих процесів

в контейнері або на віртуальній машині. GitHub Actions також підтримують більше мов та фреймворків, ніж більшість інших, включаючи Node.js, Python, Java, PHP, Ruby, C / C ++, .NET, Android та iOS.

1.2 Чинники, що впливають на якість архітектури системи безперервної інтеграції

Для систем безперервної інтеграції не існує нормативних актів чи інших офіційних документів, які б регулювали цю область. Однак, різні хмарні сервіси чи компанії, які впроваджують систему обіцяють певні експлуатаційні характеристики. Для того, щоб зрозуміти є характеристика чи показник важливим і на що впливає – необхідно їх, як мінімум, класифікувати. Це було зроблено, результат представлено на рисунку 1.6.

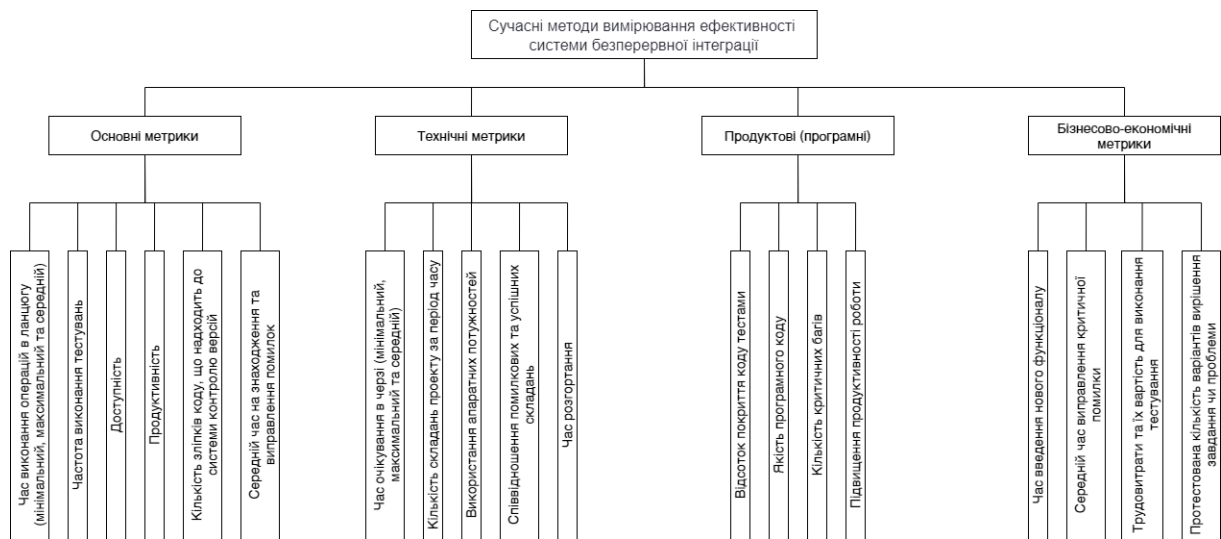


Рисунок 1.6 – Класифікація метрик систем CI/CD [11-15]

Як показано вище в класифікації, основними метриками є:

- час виконання операцій в ланцюгу (мінімальний, максимальний та середній);
- частота виконання тестувань;
- доступність;
- продуктивність;

- кількість зліпків коду, що надходить до системи контролю версій;
- середній час на знаходження та виправлення помилок.

Визначення допустимих відхилень від норми, а також самої «норми» лежить на компаніях, яким потрібна така система, або на хмарний сервісах, які пропонують послуги безперервної інтеграції майже «під ключ», адже потрібно лише налаштувати конфігураційні файли і все.

Велика частина метрик, а саме: доступність, продуктивність, час виконання операцій в ланцюгу, частота виконання тестувань в великій мірі залежить від апаратного забезпечення (серверів), тому невеликі компанії не висувають високих потреб. Для великих компаній, де працює багато технічних спеціалістів та програмістів, потрібно забезпечити працездатність системи при великій кількості користувачів, а також значно більшу доступність та продуктивність, яка б дозволила виконувати тестування програмного забезпечення одночасно для багатьох зліпків, надісланих до системи контролю версій коду.

Розглянемо вказані вище основні метрики та те, як їх вимірюють.

Доступність вимірює працездатність системи протягом часу, в результаті чого фіксується середній час простою (в хвилинах чи годинах) за певний проміжок часу (зазвичай прогнозують час недоступності на місяць чи рік). Такий простій називається недоступністю, яка може бути викликана не тільки технічними проблемами, а й бути запланованою – необхідне технічне обслуговування. При комерційних пропозиціях доступність зазвичай називають терміном SLA – угода про рівень обслуговування, який часто виражають у відсотках, найпоширенішими гарантіями працездатності системи є 99%, 99.5%, 99.9%, 99.95% та 99.99% [14].

Час виконання операцій в ланцюгу визначається лише для певної компанії експериментальним шляхом. Залежить як від апаратного забезпечення, так і необхідної кількості операцій в цьому ланцюзі. Хоча це і визначається потребами компанії індивідуально, але в більшості своїй всі орієнтуються на час менше 15-20 хв для стандартної збірки та тестування програмного коду проєкту, адже за

такої швидкості розробник програмного забезпечення бачить швидкий зворотній зв'язок та, в разі необхідно, швидко виправляє проблеми.

Частота виконання тестувань є важливою метрикою, яка в деякій мірі залежить і від швидкості виконання операцій в ланцюгу, однак розгортання можуть виконуватися паралельно. Потрібно забезпечити можливість хоча б декількох тестувань на день для кожного програміста, який працює з системою. Багато компаній роблять можливим і десятки розгортань на людину в день.

Продуктивність вимірюється в кількості виконання паралельних ланцюгів та загальній кількості виконаних (необхідних певній компанії) ланцюгів за певний проміжок часу, тобто в деякій мірі це можна назвати пропускнуою можливістю системи. Також сюди входить, в деякій мірі, час виконання ланцюгу та утилізація серверних ресурсів (ефективність використання системи). Необхідна продуктивність повинна розраховуватись і забезпечуватись апаратними можливостями та правильною архітектурою [13].

Кількість зліпків коду, що надходить до системи контролю версій показує наскільки великими кроками команда працює над тим чи іншим функціоналом, правками і т.д. Чим частіше, до певної межі розумності, це відбувається – тим краще. Адже рухаючись невеличкими кроками команда може швидше бачити проблемні місця в програмному коді (варто зауважити, що не завжди новий зліпок коду дорівнює виконанню ще одного тестування).

Середній час на знаходження та виправлення помилок – надзвичайно цікава метрика. Оскільки для автоматичного тестування необхідно написати ці самі тести, то більшість помилок не дійде до користувачів. Але, навіть, якщо це трапиться, то команда зможе швидко виправити за рахунок впевненості в тому, що автоматичні тести перевіряють працездатність абсолютної більшості функцій і те, що швидке виправлення помилки ніяк не вплинуло на інші частини web-to-print чи іншого сайту. Після впровадження безперервної інтеграції цей час зменшується в рази або на порядок.

1.3 Предмет і регламент патентного пошуку

Було проведено патентний пошук за обраною тематикою для того щоб дізнатися як розвиваються системи безперервної інтеграції протягом останніх років. Регламент та власне патентний пошук наведено в таблицях 1.1 та 1.2 відповідно.

Таблиця 1.1 – Регламент патентного пошуку

Предмет пошуку	Мета пошуку	Країни	Класифікаційні індекси	Ретро-спективність	Джерела інформації
1. Архітектура системи безперервної інтеграції 2. Структура ланцюгу CI/CD 3. Вплив вибору інструментів на ефективність процесу CI/CD 4. Автоматизація тестування та інтеграційне тестування 5. Автоматична збірка програмного коду 6. Масштабування за допомогою системи безперервної інтеграції	Визначити тенденції розвитку архітектур систем безперервної інтеграції, необхідних інструментів та способів автоматичного тестування і збірки, а також масштабування під час навантажень	Німеччина, США, Канада, Китай, Австралія, Японія, Корея, Україна, Росія	B41F13, G06F8, H04L29, G06F11, H04L63, H04L41, G06Q10, G06F15, H04L43, G06F40, G07C5, G06F13	2010-2020 (10 років)	Інтернет ресурси: er.espacenet.com ; patents.google.com ; library.uipv.org

Таблиця 1.2 – Патенти, відібрані в результаті пошуку

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
1.	США; US 20140053138 A1; G06F 11/36	SAP Portals Israel Ltd 20/02/2014	Перевірка якості сирцевого коду при відправці у систему контролю версій, а саме контроль за новим кодом, автоматичний контроль якості та надання звітів [16].
2.	США; US 20180285128 A1 G06F 9/451; G06F 11/36; G06F 8/38	International Business Machines Corporation 04/09/2018	Метод здійснення автоматичного тестування графічного користувацького інтерфейсу за допомогою програмного коду [17].
3.	США; US 20190370010 A1 G06F 8/41; G06F 11/36; G06F 9/44; G06F 9/448	Bank of America Corporation 09/08/2019	Архітектура системи автоматичної інтеграції сирцевого коду при його зміні в репозиторії [18].
4.	США; US 20190369980 A1 G06F 8/65; H04L 29/08	Palantir Technologies Inc. 05/12/2019	Ланцюг безперервної інтеграції з контрольними точками на певних етапах та можливістю відміни внесених у систему змін на будь-якому етапі [19].
5.	США; US 8863114 B2 G06F8/60	Red Hat Inc 14/10/2014	Клієнтська система контролю версій (VCS) звертається до віддаленої VCS для отримання різниці між поточними метаданими, що зберігаються локально, що представляють першу версію встановлених в даний час пакетів програмного забезпечення, і новими метаданими, що представляють другу версію пакетів програмного забезпечення, доступних з сервера для поширення, без необхідності завантажувати все нові метадані [20].
6.	США US 7519964 B1 G06F8/71	Oracle America Inc 16/12/2015	Система і спосіб розгортання додатків в кластері серверів з центрального домену. Спосіб може включати в себе збірку набору вихідного коду і пов'язаних файлів ресурсів на сервері адміністрування, пов'язаному зі сховищем додатків [21].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
7.	Японія JP 2011053966 A1 G06F8/60	Ricoh Co Ltd 17/09/2011	Система підготовки програмного забезпечення перед виконанням автоматизованої перевірки якості коду за допомогою програмних тестів. Під підготовкою розуміється спеціальний та новий процес збірки цього програмного забезпечення [22].
8.	Корея KR 20130068524 A1 G06F8/65	Лі Гю-хо 26/06/2013	Метод оновлення програмного забезпечення для автоматичних збірок, під час роботи технічного спеціаліста в терміналі операційної системи [23].
9.	США US 9710259 B2 G06F8/70	VMware Inc 18/07/2017	Система розгортання дозволяє розробнику налаштувати план розгортання, створений відповідно до логічних, багаторівневих планів додатків для розгортання декількох додатків в хмарній інфраструктурі. Використовуючи систему розгортання, розробник вставляє призначений для користувача сценарій або завдання в послідовність завдань, які необхідно виконати для розгортання компонента додатка на різних етапах (наприклад, установка, настройка, запуск) на віртуальній машині. Система розгортання прив'язує користувацький сценарій до різних етапів розгортання компонента додатка, так що налаштування в плані розгортання підтримуються шляхом внесення змін до базового проєкту програми [24].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
10.	США US 10120670 B1 G06F8/60	Capital One Services LLC 06/11/2018	Щонайменше один додаток може включати в себе інструкції, що містять інструкції програми та безліч окремих інструкцій визначення конвеєра. Інструкції додатка можуть знаходитися у віртуальному контейнері, що включає, щонайменше, одну програму, яка в загальному випадку виконується в безлічі різних середовищ безперервної інтеграції та доставки (CI/CD). Кожна з безлічі окремих інструкцій визначення конвеєра може бути налаштована для кожної з безлічі різних середовищ CI/CD, так що кожне визначення конвеєра може працювати тільки в середовищі CI/CD, для якого воно створена. Кожне визначення конвеєра може бути налаштоване так, щоб середовище CI/CD, для якого воно створений, виконувало щонайменше одну програму [25].
11.	Китай CN 101978366 B1 G06F8/65	О. Аміга, Дж. Макінтайр, Д. Шукла, Дж. Чжан 30/10/2013	Управління додатками в обчислювальному середовищі часто включає в себе значну кількість обчислювальних операцій, таких як отримання додатку, розгортання об'єктів даних в різних місцях і настройка програми щодо розгорнутого пристрою. Звичайні обчислювальні середовища можуть забезпечувати недостатню підтримку протягом типового життєвого циклу додатку, особливо для обчислювального середовища, розподіленої по багатьом пристроям. Замість цього обчислювальна середовище може бути представлена в розгорнутій ієрархії об'єктів, яке може бути розподілена по різних пристроях. Тому може бути надана послуга управління додатками для установки додатка в ієрархії об'єктів і для підтримки [26].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
12.	США US 9645858 B2 G06F8/60	VMware Inc 09/05/2017	Система розгортання дозволяє розробнику визначити логічний, багаторівневий проєкт програми, який можна використовувати для створення і управління (наприклад, повторного розгортання, оновлення, резервного копіювання, виправлення) декількох додатків в хмарної інфраструктурі. У проєкті програми розробник моделює загальну архітектуру програми або топологію, яка включає в себе окремі і кластерні вузли (наприклад, віртуальні машини), логічні шаблони, постачальників хмарних середовищ, середовища розгортання, програмні служби, код конкретного додатка, властивості і залежності між компоненти вищого і другого рівня [27].
13.	Китай CN 105516233 B1 G06F8/60	Дж. Джонстон, С.П. Генрі 04/06/2019	Спосіб розгортання в хмарне середовище включає в себе запит, виявлений для виконання програми в хмарній службі, для надання послуги або ресурсу в хмарній службі і системи для успішного виконання програми. У відповідь на запит запис дескриптора додатка витягується з файлу дескриптора. Запит ресурсів і послуг перетворюються в дію, що робиться в середовищі хмарної служби, для надання необхідного ресурсу додатка або послуги. Дії, які повинні бути виконані, передаються на основі дескриптора запису додатка, наданого в деталях і виконаного з попередньо визначеною послідовністю. Стан використовується для визначення того, чи були надані необхідний ресурс або служба для успішного виконання програми в хмарній службі. [28].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
14.	Канада CA 3019911 A1 H04L43/0817	Peter Nickolov, Ross Schibler, Bert Armijo 02/02/2017	Різні аспекти, описані в даному патенті, спрямовані на різні методи для: оцінки надійності серверної системи, уразливості і сумісності компонентів з використанням краудсорсингу даних сервера і уразливості; формування автоматизованих рекомендацій щодо поліпшення показників серверної системи; і автоматично і умовно оновлювати або оновлювати системні пакети / компоненти [29].
15.	Китай CN 104520818 A1 G06F11/3692	Adam SpektorInbar ShaniAmichai Nitsan 15/04/2015	Метод виконання тестів продуктивності додатка в CI-конвеєрі розгортання. Метод ідентифікує зміни коду - це дві різні збірки в середовищі тестування продуктивності. Метод отримує базовий результат тесту, виконуючи набір налаштованих тестових сценаріїв в базовій збірці з першою базою коду. Метод аналогічним чином тестує нову збірку, виконуючи набір параметрів тестових сценаріїв для нової збірки з другою базою коду, щоб отримати новий результат тесту. Значення продуктивності визначаються шляхом порівняння базового результату тесту і нового результату тесту [30].
16.	США US 8903943 B2 G06F9/5027	Salesforce com Inc 02/12/2014	Тут розкриті способи інтеграції хмарних додатків і віддалених завдань. У деяких реалізаціях запит на ініціювання процедури віддаленого виконання може бути прийнятий в першій обчислювальній системі. Перша обчислювальна система може управлятися першим об'єктом і може бути налаштована для надання обчислювальних послуг на вимогу безлічі об'єктів, включаючи другий об'єкт [31].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
17.	Австралія AU 2018203054 B2 G06F11/368	Solano Labs Inc 19/12/2019	Система для безперервної інтеграції редакцій вихідного коду, система містить: щонайменше, один процесор, підключений до пам'яті, процесор при виконанні налаштований для: реєстрації набору тестів для розподіленого виконання, причому набір тестів визначає вихідний код і, щонайменше, один тест, який повинен бути виконаний для додатка, в якому реєстрація включає в себе ідентифікацію сховища коду для вихідного коду, пов'язаного з тестованим додатком; автоматично визначати вимоги до конфігурації для набору тестів на основі аналізу сховища коду, причому визначення вимог до конфігурації включає в себе визначення залежностей виконання для набору тестів і додатки, яке буде тестуватися; генерувати інструкції для виконання безлічі екземплярів виконання, причому, щонайменше, деякі з безлічі екземплярів виконання сконфігуровані для: визначення середовища виконання на основі вимог до конфігурації [32].
18.	Китай CN 104185840 B1 G06F11/3688	Інбар ШАНІ, Нітсан Аміхай, Сигал МАОН 16/01/2018	Тип методу, який використовується для визначення пріоритетності декількох тестів в процесі оптимізації розгортання. Цей метод виконує класифікацію, щоб забезпечити оцінку тесту для кожного в декількох тестах на основі атрибута тестування і середовища тестування для декількох тестів. Цей метод призначений для продовження набір тестів в раціоналізації розгортання з використанням класу тесту. Цей метод виконує набір тестів в раціоналізації тривалого розгортання [33].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
19.	США US 9886375 B2 G06F11/3692	International Business Machines Corp 06/02/2018	У відповідь на запуск тестового прикладу модульного тестування, зазначеного в платформі модульного тестування в вихідній системі, для запуску хоча б на одній меті, вихідна система направляє запит на виконання інфраструктури автоматизації тестування програмного забезпечення в тестовому прикладі модульного тестування в віддалену систему для виконання Сценарій функціонального тесту, позначений в тестовому прикладі модульного тестування на цілі в віддаленій системі, сценарій функціонального тесту, що позначає функціональний тестовий приклад, який повинен бути запущений на цілі для перевірки роботи одного або декількох функцій мети [34].
20.	США US 20190138301 A1 G06F8/77	CSC Agility Platform Inc 09/05/2019	Системи, методи і постійні носії можуть отримувати інформацію про план розробки програми. План розробки програми може бути пов'язаний як мінімум з одним критерієм розгортання. Один або кілька доступних ресурсів інфраструктури можуть бути ідентифіковані на основі інформації про план розробки програми. Безліч варіантів розгортання може бути визначено на основі одного або декількох доступних ресурсів інфраструктури. Безліч варіантів розгортання може бути визначено як відповідне щонайменше одним критерієм розгортання. Безліч варіантів розгортання може бути ранжовано для створення упорядкованого набору параметрів розгортання [35].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
21.	Китай CN 102693183 B1 G06F11/368	Пісня Юаньюань, Гун Гуйдун, Джин Руйкі, Цзяо Ліанхен 01/04/2015	Винахід розкриває спосіб і систему для здійснення автоматичного тестування програмного забезпечення. Спосіб включає в себе наступні етапи: отримання об'єктного коду тестованого програмного забезпечення; отримання тестового прикладу для тестованого програмного забезпечення, при цьому тестовий приклад генерується шляхом прийняття технології орієнтованого об'єкта і включає в себе ключове слово тестового об'єкта і дані тест-драйву для тестованого програмного забезпечення; складання та збереження тестового скрипта відповідно до тестового прикладу; запуск сценарію тестування, щоб виконати автоматичне тестування програмного забезпечення об'єктного коду тестованого програмного забезпечення на основі ключового слова тестового об'єкта і даних тест-драйву [36].
22.	США US 10033833 B2 H04L67/34	Cisco Technology Inc 24/07/2017	Розкриті варіанти здійснення полегшують розподілену оркестровку і розгортання програми розподілених обчислень на основі хмари. У деяких варіантах здійснення може бути визначений перший контекст розгортання, пов'язаний з хмарним додатком. Перший контекст розгортання може містити безліч правил, де кожне правило може вказувати умови для застосування конфігурацій до ресурсів, пов'язаних з хмарним додатком. У деяких варіантах здійснення розгортання хмарного додатка в першому хмарі може потім бути організовано на основі, частково, першого контексту розгортання [37].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
23.	США US 9760366 B2 G06F9/44536	Amazon Technologies Inc 12/09/2017	Представлені методи управління конвеєром розгортання з використанням шаблону успадкованого і розширюється вихідного коду, зазвичай званого шаблоном живого конвеєра (LPT). Як описано, шаблони конвеєра в реальному часі можуть використовуватися для управління конвеєрами розгортання, які, в свою чергу, використовуються для запуску, обслуговування та оновлення сервісів і систем, використовуваних для розміщення та надання обчислювальних сервісів [38].
24.	США US 8813040 B2 G06F8/70	Versionone Inc 19/08/2014	Спосіб повідомлення про прогонах збірки включає в себе: отримання змін коду для програмного продукту; ідентифікують зміни коду які зачіпає робочий елемент розроблюваного програмного продукту, робочий елемент визначає функцію, яка повинна бути додана чи дефект, який повинен бути видалений з програмного продукту; генерування безлічі прогонів збирання програмного продукту, відповідні прогони збірки, відповідні одному або декільком змін коду; і дані, що зв'язують робочий елемент з одним або декількома прогонами збірки, кожен з яких відповідає щонайменше одному з одного або декількох змін коду, зазначених як включають робочий елемент; отримання призначеного для користувача введення, що вибирає робочий елемент; і у відповідь на призначений для користувача введення відображають відповідні ідентифікатори щонайменше одного прогону збірки одного або декількох прогонів збірки, пов'язаних з робочим елементом [39].

Продовження таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
25.	США US 9760343 B2 G06F8/75	SAP SE 12/09/2017	Структура для створення додатків описана в даному документі. Відповідно до одним аспектом метадані додатки можуть бути вказані користувачем. Метадані програми включають інформацію про конфігурацію програми. Метадані додатка і правила складання можуть бути введені в модуль попередньої збірки для генерації сценаріїв збірки для збірки додатку. Сценарії збірки можуть ґрунтуватися на метаданих додатка і правилах складання. Сценарії збірки можуть бути надані модулю збірки для збірки додатку для розгортання [40].
26.	США US 10437585 B2 G06F8/71	Oracle International Corp 01/09/2016	Запропоновано системи та методи для управління та відстеження конвеєрів процесу та інших потоків таким чином, що зменшує вимогу до центрального контролюючого процесу для управління кожним кроком потоку конвеєра та станом, пов'язаним з кожним запущеним екземпляром [41].
27.	Канада CA 2970824 C H04L29/10	Kodiak Networks Inc 14/01/2016	Спосіб включає в себе ініціювання службовим оркестратором створення одного або декількох екземплярів контейнера для першого кластера служб. Спосіб додатково включає створення менеджером контейнерів одного або декількох екземплярів контейнерів і відображення одного або декількох екземплярів контейнера першого кластера послуг на одну або більше перших віртуальних машин, що належать до першої групи серверів віртуальних машин, відповідно до платформи профіль першої групи серверів віртуальної машини та першої служби, що надається першим кластером послуг [42].

Кінець таблиці 1.2

№ п/п	Країна, що видала патент, вид і номер охоронного документу, класифікаційний номер МПК	Заявник з вказівкою країни, номеру заявки, дати пріоритету, конвенційний пріоритет, дата публікації	Суть поданого технічного рішення і мета його здійснення за змістом опису винаходу
28.	США US 10521747 B2 G06Q10/0633	Northrop Grumman Systems Corp 08/04/2015	Система забезпечення семантичного механізму масштабування, що керується політикою дій та даних, що змінюються динамічно. Контекст політики являє собою сукупність онтологій, що фіксують політики підприємства в семантичній, декларативній граматиці, в семантичній базі даних [43].
29.	США US 10411975 B2 H04L41/5054	CSC Agility Platform Inc 15/03/2013	У варіантах здійснення цього винаходу описані вдосконалені можливості для середовища віртуалізації, пристосованого для розробки та розгортання щонайменше одного робочого програмного забезпечення, при цьому середовище віртуалізації має структуру метамоделі, що дозволяє прив'язувати політику до робочого програмного забезпечення при розробці навантаження, яке застосовується при розгортанні програмного забезпечення [44].
30.	США US 10230571 B2 G06F8/30	Equinix Inc 29/10/2015	Запропонована система містить платформу мікросервісу для розробки та виконання безлічі мікросервісів, при цьому кожен мікросервіс містить службу, що розгортається самостійно, сконфігуровану для виконання однієї або декількох функцій контракту інтерфейсу. Також описано платформу оркестрації для розробки та виконання оркестратора організації мікросервісів для роботи платформи взаємозв'язку, використовуючи одну або кілька віртуальних схем [45].

1.4 Завдання дослідження

Завдання дослідження – створити архітектуру системи безперервної інтеграції для порталів web-to-print, визначивши які фактори на це впливають та підібравши доцільні інструменти.

Для виконання поставленого завдання у роботі сформульовано наступні задачі:

- 1) Визначити загальну схему процесу безперервної інтеграції та його етапи;
- 2) Провести аналіз наявних джерел інформації щодо типової архітектури CI/CD, використовуваних інструментів та факторів впливу на будову;
- 3) Провести експериментальні дослідження варіантів масштабування серверної частини, їх доцільності та ефективності;
- 4) Визначити необхідне програмно-апаратне забезпечення, режими та параметри його роботи для впровадження CI/CD;
- 5) Створити базову архітектуру системи безперервної інтеграції для порталів типу web-to-print;
- 6) Розробити старт-ап проєкт центру з розроблення веб-порталів;
- 7) Узагальнити результати та сформулювати висновки роботи.

Висновки до першого розділу

Було наведено загальну схему процесу безперервної інтеграції, класифікацію таких систем, наведено типову архітектуру, а також визначено фактори, які на неї впливають. Проведено класифікацію метрик для вимірювання ефективності системи CI/CD.

Виконано патентний пошук за тематикою дослідження. Знайдено більше 2000 патентів, з яких відібрано 30.

Отже, згідно результатів аналітичного огляду можна дійти висновку, що використання систем безперервної інтеграції та дослідження їх архітектури є надзвичайно актуальним.

РОЗДІЛ 2

ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

2.1 Тенденції розвитку архітектури систем безперервної інтеграції за результатами патентного пошуку

В результаті проведеного патентного пошуку було знайдено понад 2 тисячі патентів, розглянуто понад 200 патентів та наведено і класифіковано 30. Всі патенти відповідають предмету пошуку, що був регламентований. Ретроспективність проведеного патентного пошуку 10 років, оскільки більш ранні розробки вже можуть вважатися застарілими та не актуальними.

Джерелами пошуку були такі інтернет-ресурси: Укрпатент, Espacenet та Google Patents.

Регламентовані країни: Німеччина, США, Канада, Китай, Австралія, Японія, Корея, Україна та Росія.

Використання систем безперервної інтеграції при створенні складних веб-ресурсів (веб-порталу, інтернет-магазину, CRM-системи та ін.) вже не є чимось новим. Необхідність слідкувати за якістю змін, безпекою, наявністю змін, що «ламають» працездатність коду, необхідність масштабування під час збільшення кількості користувачів потребує застосування сучасних рішень – CI/CD. Метрики та характеристики системи безперервної інтеграції залежать від її архітектури та використовуваних інструментів, на яких вона побудована.

Про актуальність розвитку систем безперервної інтеграції свідчить велика кількість знайдених патентів у даній галузі впродовж розглянутого періоду (рисунок 2.1).

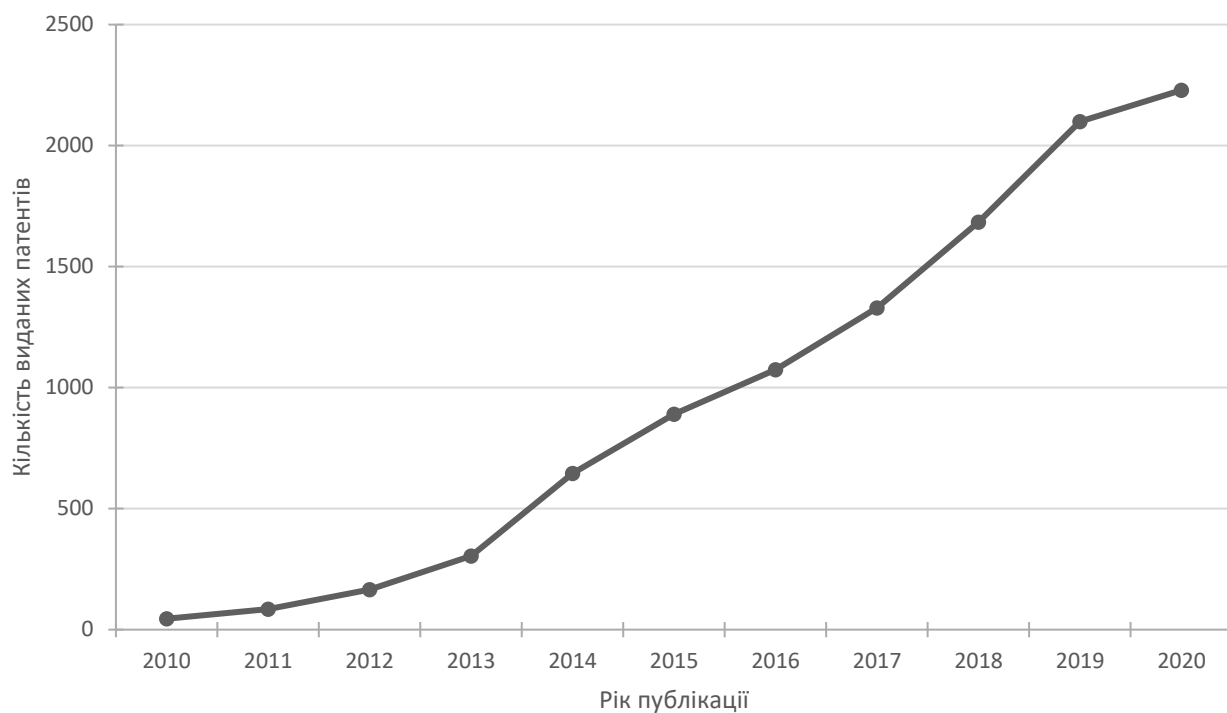


Рисунок 2.1 – Кумулятивна крива публікації патентів за ретроспективністю в 10 років

Динаміка публікації патентів за роками наведена на рисунку 2.2.

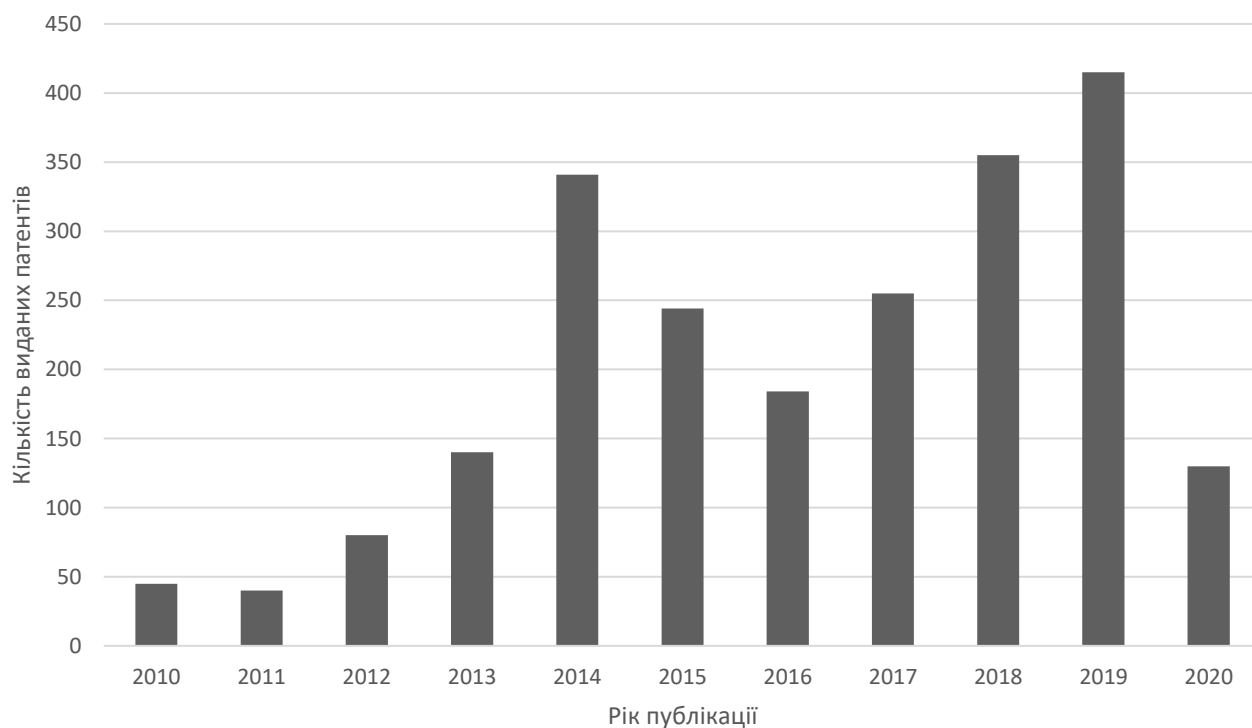


Рисунок 2.2 – Динаміка публікацій патентів за роками

Аналізуючи рисунок 2.2 можна помітити, що в 2014 році був сплеск, після чого падіння, а ріст знову почав продовжуватись тільки з 2017 року. Для того, щоб зрозуміти причину створено кумулятивні криві по кожному з досліджуваних напрямів (рисунок 2.3).

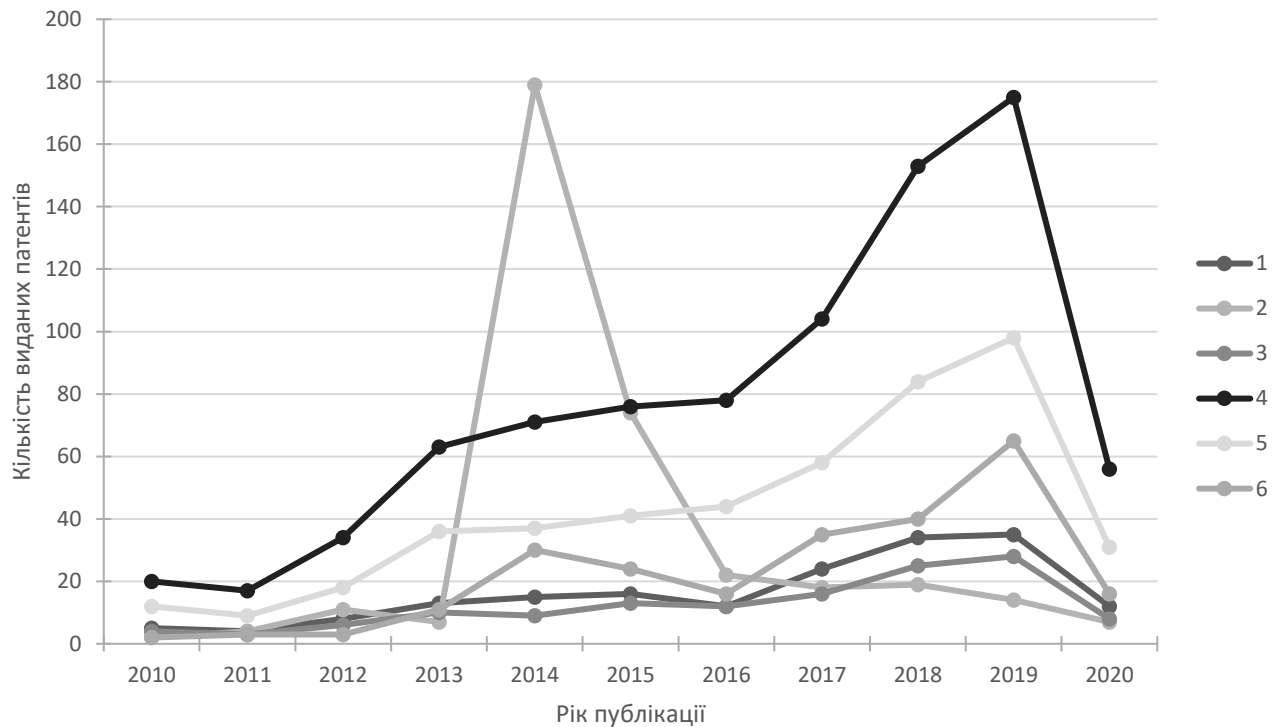


Рисунок 2.3 – Кумулятивна крива публікації патентів за окремими напрямками: 1 – архітектура системи безперервної інтеграції; 2 – структура ланцюгу CI/CD; 3 – вплив вибору інструментів на ефективність процесу CI/CD; 4 – автоматизація тестування та інтеграційне тестування; 5 – автоматична збірка програмного коду; 6 – масштабування за допомогою системи безперервної інтеграції.

З рисунку 2.3 видно, що сплеск патентування у 2014 році зумовлений інтересом до структури ланцюгу CI/CD. Це зумовлено виходом програмного забезпечення Docker у березні 2013, що дозволило спростити багато операцій, а також вплинуло на загальну структуру ланцюгів, що саме і почали патентувати.

На рисунку 2.4 представлено діаграму розподілу виданих патентів по країнах.

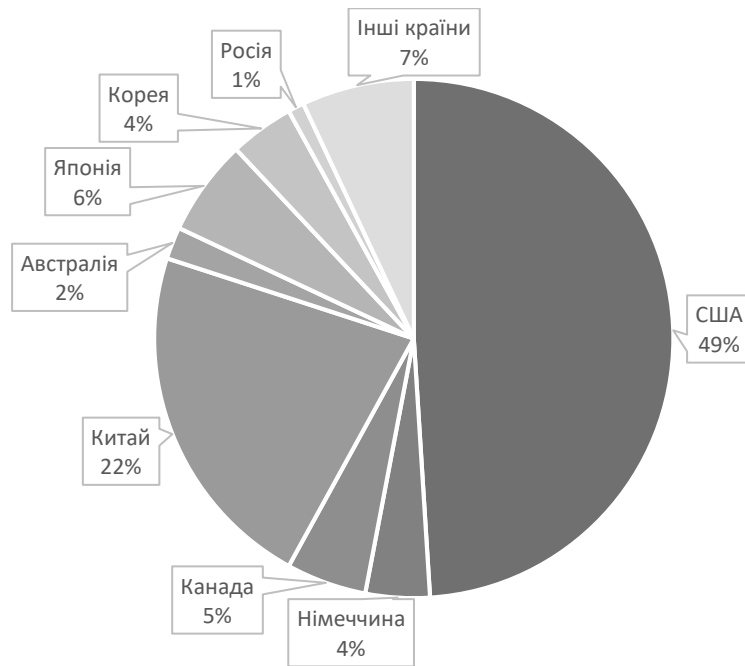


Рисунок 2.4 – Співвідношення патентування розробок за країнами-власниками патентів

Найбільша активність проведення досліджень в тематиці систем безперервної інтеграції приходить на США та Китай. Доля кожної з інших країн набагато менша, але Японія, Корея, Канада та Німеччина також проявляють високу активність.

Розподіл патентів за предметом пошуку наведено на рисунку 2.5.

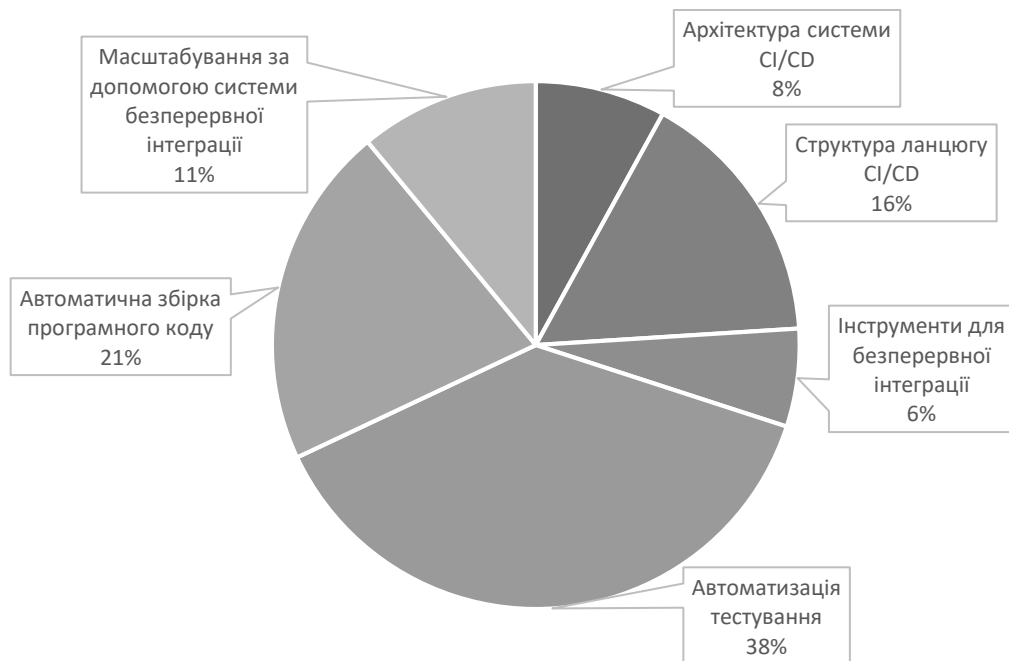


Рисунок 2.5 – Розподіл патентів за предметом пошуку

Аналіз рисунку 2.5 дає розуміння, що найбільше патентів подається в категорії автоматизації тестування. Це і не дивно, адже суть процесу безперервної інтеграції – автоматичне тестування на кожному маленькому етапі для запобігання помилок.

Отже, можна зробити висновок, що всі напрямки, за якими проводився патентний пошук зростають, однак деякі, типу автоматизації тестування, привертають більше уваги серед науковців.

Застосування систем безперервної інтеграції, зазвичай, обмежено проектами з великою кодовою базою та складністю. Веб-портали відносяться саме до таких мультимедійних продуктів. Однак, з кожним роком складність впровадження автоматизацій зменшується, саме тому це призведе до масового використання систем безперервної інтеграції при розробці програмних продуктів в найближчі 5-10 років. Виходячи з цього, можна сказати що тема є надзвичайно актуальною, має великий потенціал та постійно розвивається.

2.2 Об'єкт та предмет дослідження

У якості об'єкту дослідження обрано архітектуру системи безперервної інтеграції для реалізації web-to-print. А фактори, що впливають на будову архітектури системи безперервної інтеграції та вибір базового програмного забезпечення – предмет дослідження.

В результаті досліджень необхідно створити раціональну архітектуру системи безперервної інтеграції для web-to-print. Вибір програмних засобів буде виконано за допомогою оцінок експертів, а за допомогою тестування на типовому веб-порталі визначено структуру й інструменти масштабування серверної частини.

2.3 Розроблення тестових файлів

Експериментальне дослідження складається з двох етапів.

Перший етап передбачає експертне оцінювання базового програмного забезпечення, варіанти якого обрано завдяки аналізу статей і досліджень [46-49].

Для другого етапу, який передбачає вибір структури й інструментів масштабування серверної частини, необхідно підготувати тестовий веб-портал та програмний код тестування.

В якості програмного коду веб-порталу, на якому проводитимуться дослідження поведінки під навантаженням користувачів, обрано «Printlance». Для проведення тестувань на ресурс «Printlance» було додано можливість завантаження великих файлів (імітація роботи з файлами клієнтів для web-to-print). Приклад сторінок (скріншоти) веб-порталу наведено на рисунках 2.6-2.7.

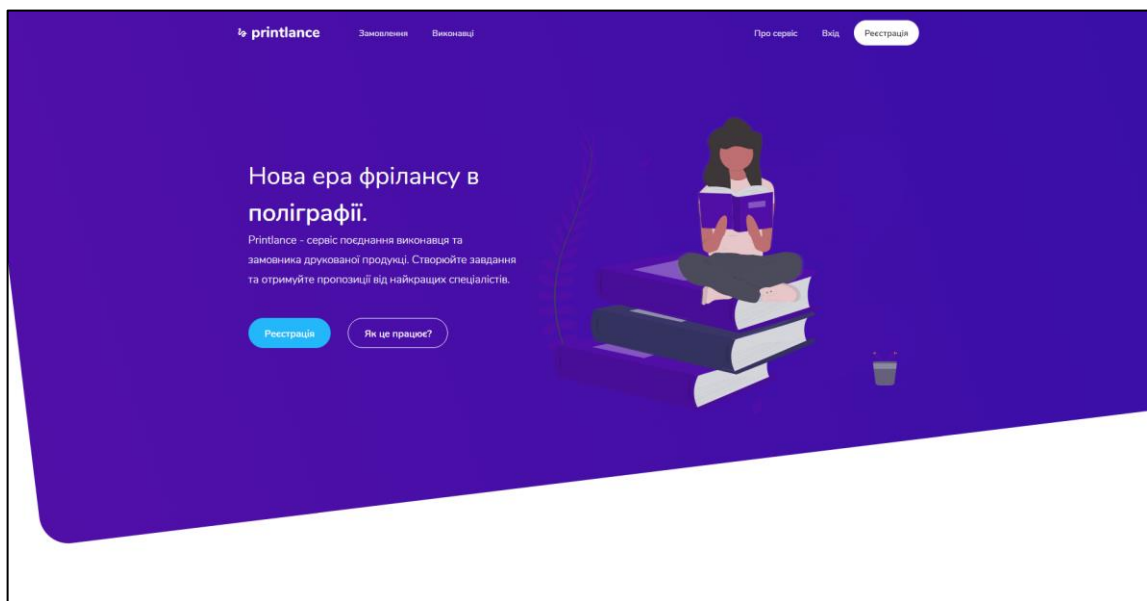
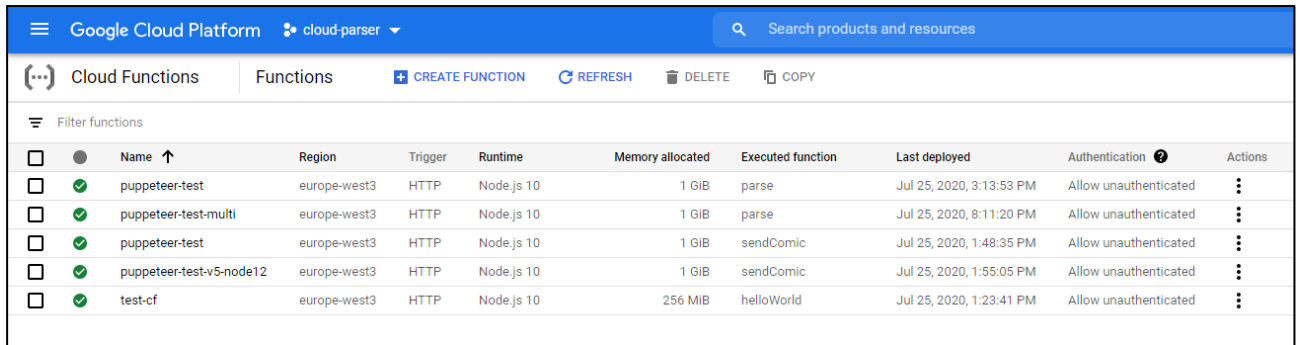


Рисунок 2.6 – Скріншот першого екрану головної сторінки

Скріншот сторінки створення замовлення на Printlance. Екран має темну синю шапку з заголовком «Нове замовлення» та підзаголовком «Створіть замовлення та отримайте пропозиції від найкращих фрілансерів.» Основна частина екрана містить форму з наступними полями: «Назва» (з підказкою «Назва до 80 символів»), «Опис» (з підказкою «Вкажіть деталі вашого замовлення.» та «Опис. Що потрібно зробити виконавцю?», а також примітку «Опис не повинен перевищувати 2000 символів.»), «Категорія» (з підказкою «Дизайн візиток»), «Бюджет» (з підказкою «Вкажіть бюджет у гривнях, не обов'язково») та розділ «Прикріпити файли (за бажанням)» з посиланням «Вибрати файли» та приміткою «Оберіть до 5-ти файлів (jpg, png, doc, docx, xls, pdf, gif, zip) розміром не більше 5mb.» Внизу екрана розташована кнопка «Створити замовлення».

Рисунок 2.7 – Скріншот сторінки створення замовлення

Програмний код для тестування запускається в середовищі Google Cloud Functions (рисунок 2.8).



	Name ↑	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
☐	☒ puppeteer-test	europa-west3	HTTP	Node.js 10	1 GiB	parse	Jul 25, 2020, 3:13:53 PM	Allow unauthenticated	⋮
☐	☒ puppeteer-test-multi	europa-west3	HTTP	Node.js 10	1 GiB	parse	Jul 25, 2020, 8:11:20 PM	Allow unauthenticated	⋮
☐	☒ puppeteer-test	europa-west3	HTTP	Node.js 10	1 GiB	sendComic	Jul 25, 2020, 1:48:35 PM	Allow unauthenticated	⋮
☐	☒ puppeteer-test-v5-node12	europa-west3	HTTP	Node.js 10	1 GiB	sendComic	Jul 25, 2020, 1:55:05 PM	Allow unauthenticated	⋮
☐	☒ test-cf	europa-west3	HTTP	Node.js 10	256 MiB	helloWorld	Jul 25, 2020, 1:23:41 PM	Allow unauthenticated	⋮

Рисунок 2.8 – Загальний вигляд консолі

Після початку тесту, за допомогою основного скрипта, запускаються хмарні функції (рисунок 2.9), що імітують користувачів згідно кривих навантаження. Принципи роботи, ліміти та вартість використання описано в документації до Google Cloud Functions [50].

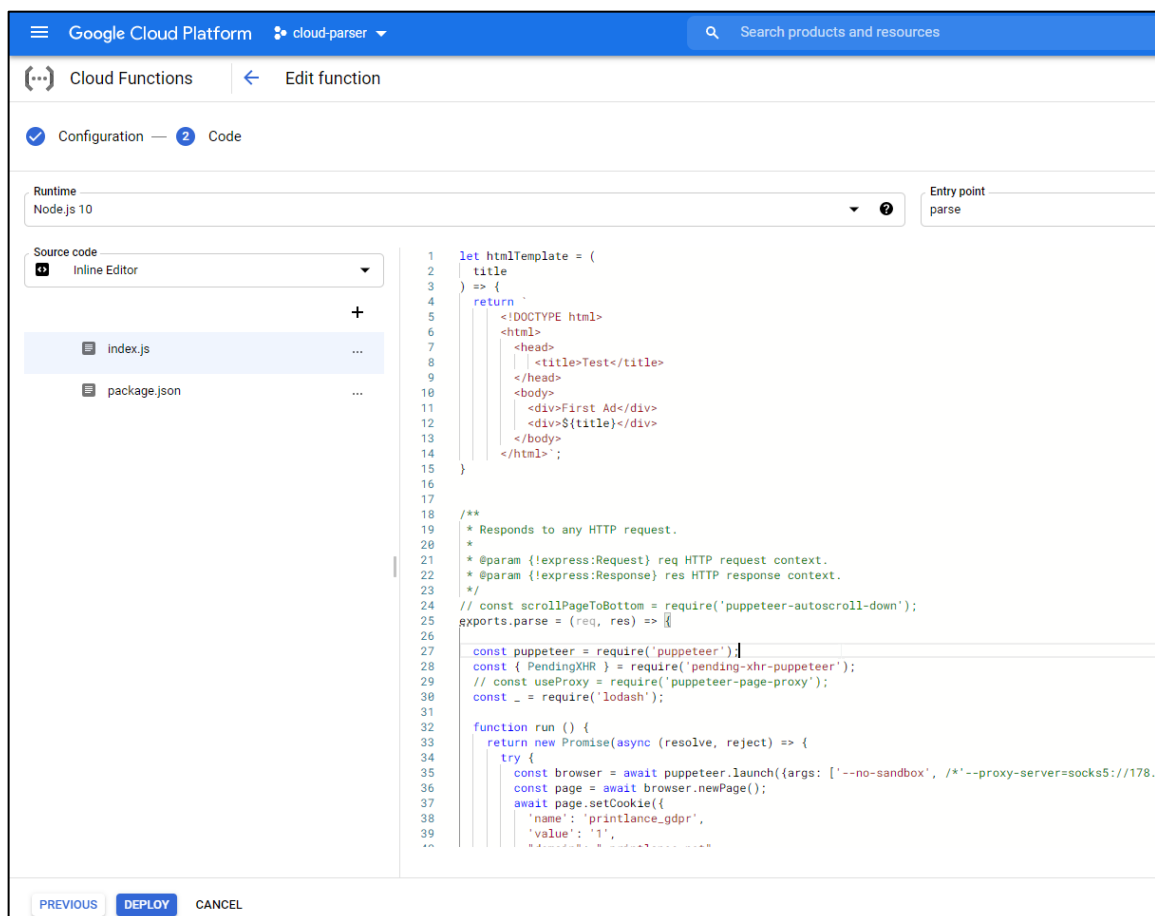


Рисунок 2.9 – Приклад скрипта-навантажувача

2.4 Методика проведення експерименту та оцінювання результатів дослідження

2.4.1 Оцінювання якості базового програмного забезпечення

Оскільки тестування різних варіантів базового програмного забезпечення для системи безперервної інтеграції коштує занадто дорого (десятки тисяч доларів), то вирішено звернутися до експертів.

Критерії впливу на вибір базового набору програмного забезпечення обрано виходячи з теоретичного аналізу у першій частині роботи:

- якість та повнота технічної документації;
- складність впровадження;
- спосіб, канали та якість технічної підтримки зі сторони постачальника;
- можливості та складність створення конфігураційних файлів ланцюгів;
- вартість (ліцензії чи підписки, впровадження);
- продуктивність та ефективність роботи, можливості масштабування системи;
- можливості адміністрування, розподілення ролей, якість та повнота статистичних даних.

Кожен із 10 експертів повинен оцінити за 7 критеріями варіант базового програмного забезпечення оцінкою від 1 до 10.

Варіанти базового програмного забезпечення:

- Jenkins;
- Circle CI;
- GoCD;
- GitHub Actions;
- Bamboo;
- GitLab;
- Travis CI.

Для виставлення оцінки експерти керуються, окрім свого досвіду, офіційними сайтами постачальників програмного забезпечення, статтями у вільному доступі, відгуками на професійних порталах.

2.4.2 Дослідження впливу навантаження на масштабування серверної частини за різної структури та інструментів

Буде розроблено 2 варіанти структури (з відповідно підібраними інструментами) враховуючи специфіку web-to-print.

Тестування проводитиметься на базі веб-порталу «Printlance» (Сервіс поєднання виконавця та замовника друкованої продукції) з деякими доопрацюваннями специфічного для web-to-print функціоналу.

Оцінка можливостей масштабування проводитиметься за 4 метриками для кожного із 3 можливих варіантів навантаження.

Метрики для оцінки:

- середній час відповіді сервера для динамічних сторінок (TTFB);
- кількість відповідей сервера з кодом помилки;
- швидкість завантаження статичних ресурсів (картинки, js, css);
- швидкість вивантаження клієнтських файлів.

Варіанти навантаження відрізняються кривою навантаження (кількість користувачів відносно часу) та характером (послідовність та варіативність HTTP-запитів).

Розроблено 3 варіанти навантаження:

- 1) поступове збільшення кількості користувачів із стандартною поведінкою («вдала рекламна кампанія»);
- 2) пік користувачів, які лише переглядають сторінки та швидко повернення до норми («опублікували посилання на головній сторінці urk.net»);
- 3) відносно стандартна кількість користувачів, але надзвичайно велика кількість завантажень файлів («потрібно виконати план до Нового Року»).

Для кожного з варіанту навантажень побудовано криву (кількість користувачів відносно часу). Криві наведено на рисунках 2.10-2.12.

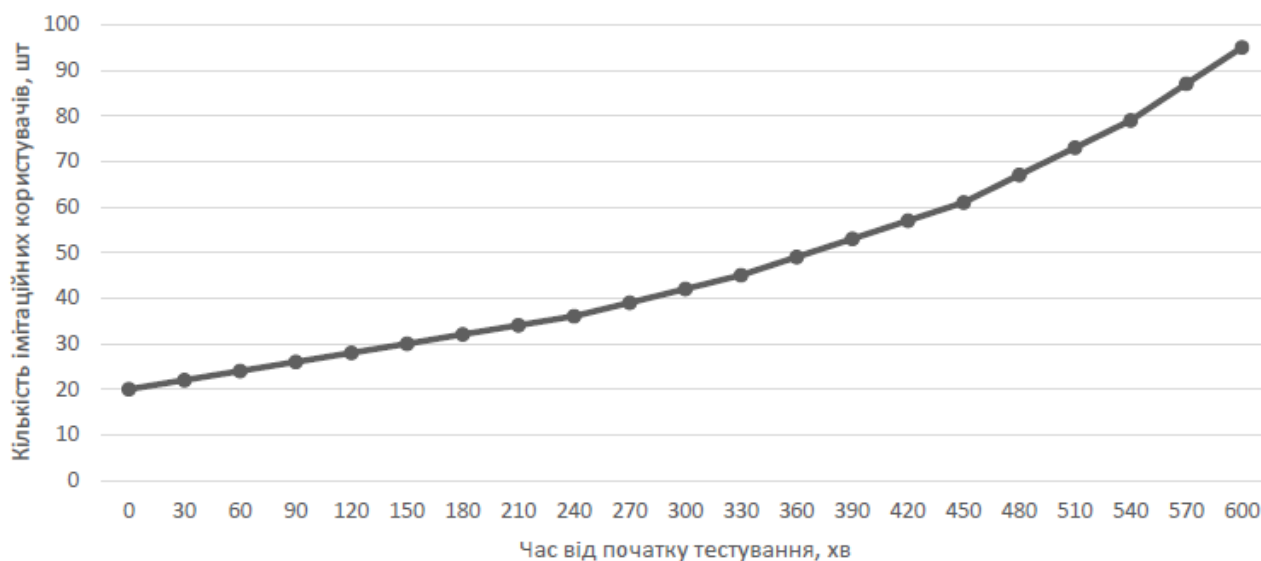


Рисунок 2.10 – Крива кількості імітаційних користувачів відносно часу для навантаження «вдала рекламна кампанія»

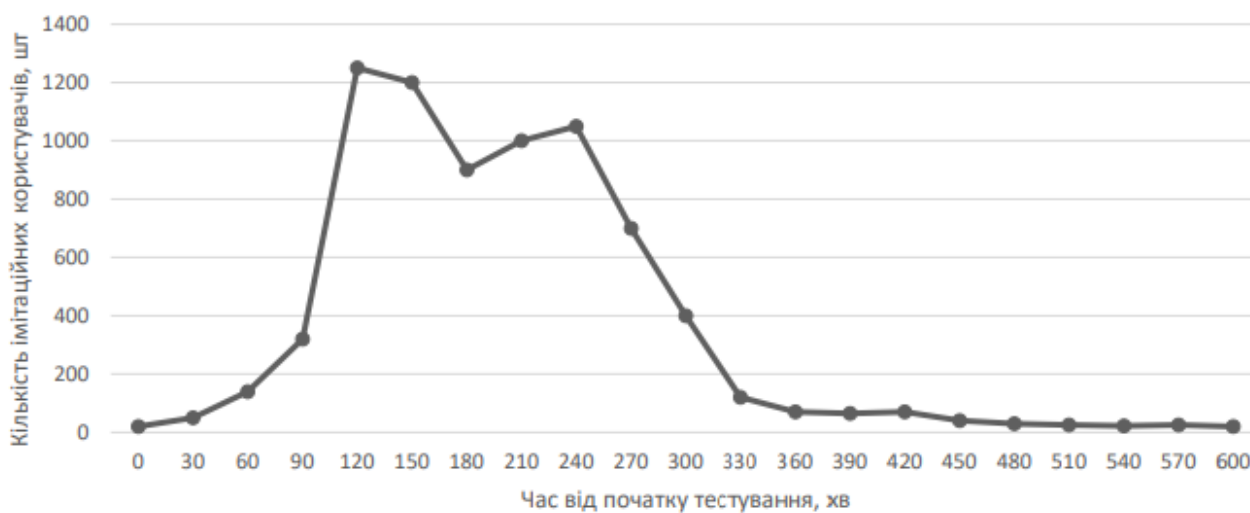


Рисунок 2.11 – Крива кількості імітаційних користувачів відносно часу для навантаження «опублікували посилання на головній сторінці urk.net»

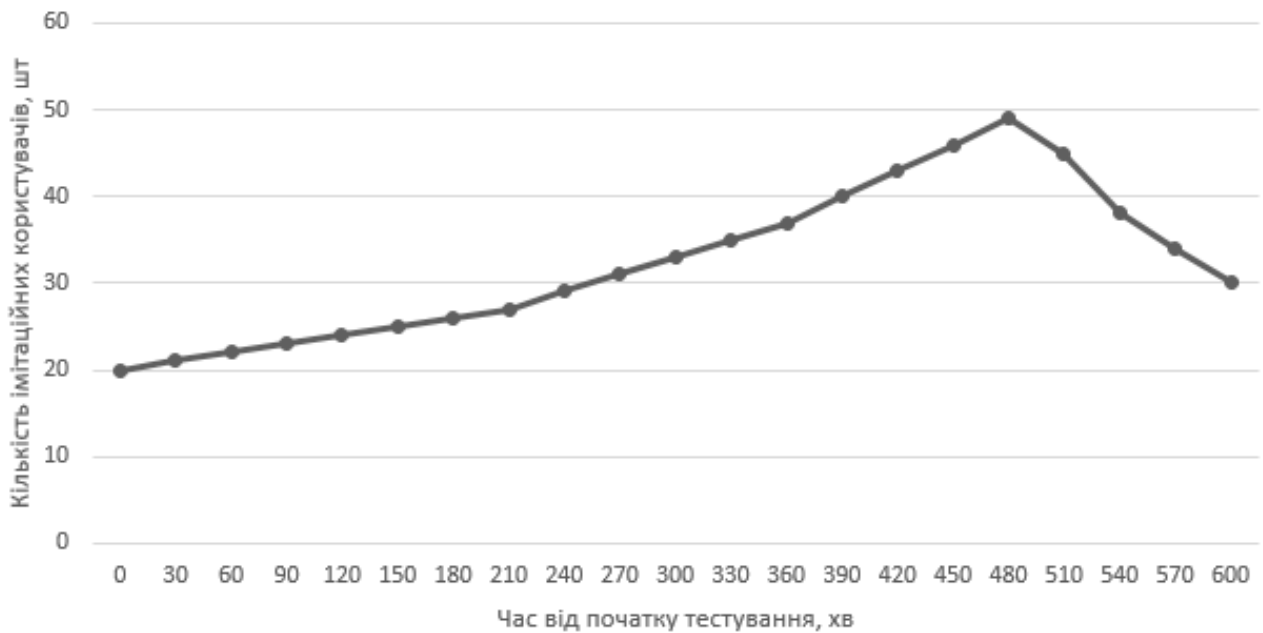


Рисунок 2.12 – Крива кількості імітаційних користувачів відносно часу для навантаження «потрібно виконати план до Нового Року»

Створення необхідного навантаження відбуватиметься за допомогою спеціально розроблених скриптів на Node.js [51] з використанням бібліотеки puppeteer [52], що запускаються у хмарному середовищі за допомогою Google Cloud Functions.

2.5 Результати досліджень

2.5.1 Оцінювання якості базового програмного забезпечення

Результати проведеного дослідження оцінки якості базового програмного забезпечення експертами наведено у вигляді таблиць та графіків.

Експерти провели оцінку варіантів базового ПЗ за кожним з критеріїв: якість та повнота технічної документації; складність впровадження; спосіб, канали та якість технічної підтримки зі сторони постачальника; можливості та складність створення конфігураційних файлів ланцюгів; вартість (ліцензії чи підписки, впровадження); продуктивність та ефективність роботи, можливості масштабування системи; можливості адміністрування, розподілення ролей, якість та повнота статистичних даних. Результати наведено у таблицях 2.1-2.7 відповідно.

Таблиця 2.1 – Результат оцінки якості та повноти технічної документації

№ експерта	Якість та повнота технічної документації						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	4	7	9	6	5	9	7
2	10	10	5	10	8	9	10
3	10	10	10	10	8	6	10
4	10	8	4	8	8	6	8
5	10	10	6	9	4	2	10
6	10	5	5	9	10	8	10
7	10	4	10	5	10	10	10
8	6	9	7	10	5	6	4
9	7	6	8	10	8	2	6
10	7	4	10	9	10	9	10

Таблиця 2.2 – Результат оцінки складності впровадження

№ експерта	Складність впровадження						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	5	7	9	6	5	10	8
2	5	9	5	10	4	10	10
3	3	9	9	9	9	5	6
4	10	6	10	10	5	3	10
5	10	6	8	5	10	2	4
6	9	8	8	10	10	10	7
7	9	10	6	4	5	10	9
8	4	6	7	3	4	2	10
9	5	4	10	10	6	10	10
10	9	5	9	9	8	8	10

Експерти оцінюють якість технічної підтримки за допомогою надсилання електронного листа з «проблемами» постачальнику, аналізу офіційних форумів, відповідей на часто задавані питання та ін.

Таблиця 2.3 – Результат оцінки способів, каналів та якості технічної підтримки зі сторони постачальника

№ експерта	Спосіб, канали та якість технічної підтримки зі сторони постачальника						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	9	9	3	10	7	9	3
2	6	5	9	10	8	10	6
3	2	8	3	10	10	10	7
4	10	10	9	4	3	9	5
5	10	4	8	10	4	10	7
6	10	9	10	2	10	4	3
7	10	10	5	10	5	6	10
8	10	4	8	3	4	10	10
9	4	6	8	10	5	8	10
10	10	5	8	6	4	3	6

Для визначення складності конфігураційних файлів, а також оцінки функціональності та можливостей, експертам запропоновано використовувати репозиторії відкритого сирцевого коду, де містяться конфігурації для системи безперервної інтеграції.

Таблиця 2.4 – Результат оцінки можливостей та складності створення конфігураційних файлів ланцюгів

№ експерта	Можливості та складність створення конфігураційних файлів ланцюгів						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	6	4	10	10	9	5	10
2	6	8	7	10	4	10	5
3	7	4	10	10	9	4	3
4	9	5	2	10	7	10	9
5	9	4	7	10	10	8	10
6	4	5	2	9	10	5	9
7	4	4	10	4	10	3	9
8	7	8	8	10	4	8	4
9	9	2	10	7	2	9	6
10	3	10	2	6	10	5	5

Таблиця 2.5 – Результат оцінки вартості (ліцензії чи підписки, впровадження)

№ експерта	Вартість (ліцензії чи підписки, впровадження)						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	10	7	10	10	5	10	10
2	8	9	10	9	3	7	4
3	8	6	2	10	10	10	10
4	10	6	9	3	3	9	7
5	10	4	10	10	3	10	5
6	10	4	10	7	4	10	6
7	10	10	9	6	7	10	10
8	10	4	4	10	10	2	7
9	7	7	2	8	10	9	9
10	10	7	6	3	9	6	5

Таблиця 2.6 – Результат оцінки продуктивності та ефективності роботи, можливості масштабування системи

№ експерта	Продуктивність та ефективність роботи, можливості масштабування системи						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	10	7	7	8	9	10	8
2	10	5	3	10	10	9	10
3	6	10	6	10	5	8	10
4	3	7	3	10	8	5	4
5	5	7	7	10	9	9	4
6	6	10	7	5	9	5	5
7	10	10	3	4	4	5	5
8	3	6	3	10	5	4	9
9	4	10	10	4	6	4	5
10	8	3	9	9	6	3	10

Для веб-порталів, звісно, важливим є можливість керувати рівнями доступу до важливого функціоналу системи безперервної інтеграції. Окрім цього наявність статистики використання за основними технічними показниками, визначення навантаженості системи та оцінка витрат на підтримання є надзвичайно важливим критерієм.

Таблиця 2.7 – Результат оцінки можливостей адміністрування, розподілення ролей, якості та повноти статистичних даних

№ експерта	Можливості адміністрування, розподілення ролей, якість та повнота статистичних даних						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
1	6	7	8	10	10	7	9
2	4	8	2	10	3	4	9
3	10	9	5	5	3	5	7
4	4	3	8	7	10	8	10
5	5	6	5	4	6	3	10
6	10	10	9	5	3	10	10
7	10	10	10	10	10	9	3
8	3	9	5	10	4	5	5
9	4	10	9	10	10	2	10
10	5	10	6	10	5	7	9

Для оцінок експертів за кожним критерієм визначено середнє значення для усіх варіантів базового програмного забезпечення (таблиця 2.8).

Таблиця 2.8 – Середня оцінка за кожним із критеріїв

Критерій	Варіант базового програмного забезпечення						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
Якість та повнота технічної документації	8.4	7.3	7.4	8.6	7.6	6.7	8.5
Складність впровадження	6.9	7	8.1	7.6	6.6	7	8.4
Спосіб, канали та якість технічної підтримки зі сторони постачальника	8.1	7	7.1	7.5	6	7.9	6.7
Можливості та складність створення конфігураційних файлів ланцюгів	6.4	5.4	6.8	8.6	7.5	6.7	7
Вартість (ліцензії чи підписки, впровадження)	9.3	6.4	7.2	7.6	6.4	8.3	7.3

Кінець таблиці 2.8

Критерій	Варіант базового програмного забезпечення						
	Jenkins	Circle CI	GoCD	GitHub Actions	Bamboo	GitLab	Travis CI
Продуктивність та ефективність роботи, можливості масштабування системи	6.5	7.5	5.8	8	7.1	6.2	7
Можливості адміністрування, розподілення ролей, якість та повнота статистичних даних	6.1	8.2	6.7	8.1	6.4	6	8.2

На основі таблиці 2.8 побудовано графік, що зображено на рисунку 2.13.

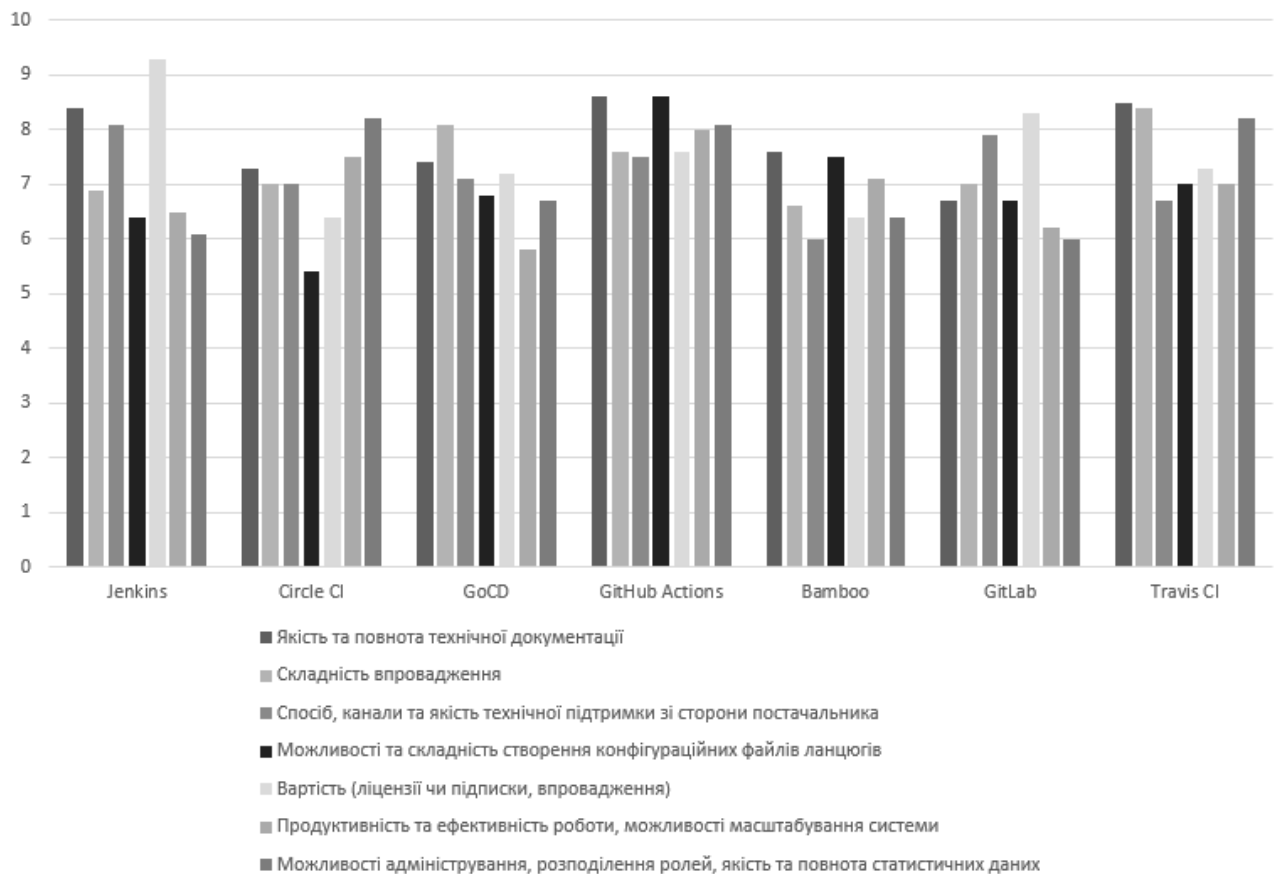


Рисунок 2.13 – Порівняння середніх значень по кожному критерію оцінки якості базового програмного забезпечення

На основі оцінок експертів можна зробити висновок, що найкращим варіантом базового програмного забезпечення для системи безперервної інтеграції веб-порталів типу web-to-print є GitHub Actions.

2.5.2 Дослідження впливу навантаження на масштабування серверної частини за різної структури та інструментів

Було розроблено 2 варіанти структури масштабування (з відповідно підібраними інструментами), що зображено на рисунках 2.14-2.15.

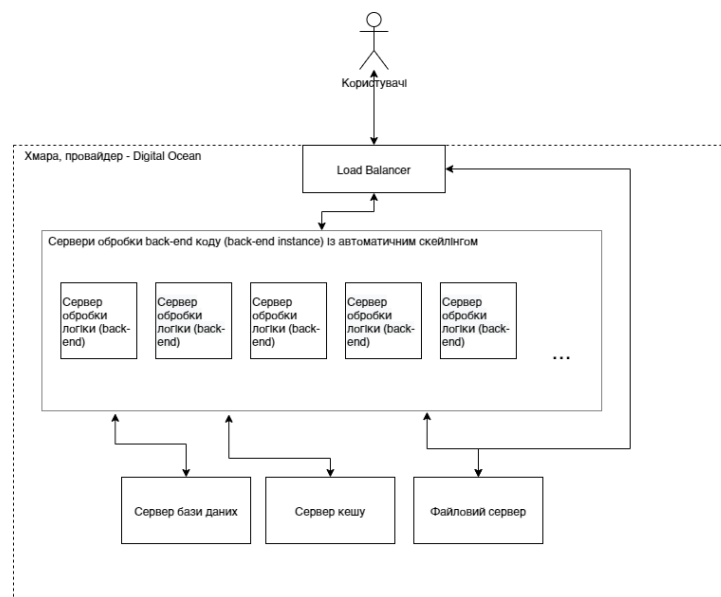


Рисунок 2.14 – Перший варіант структури масштабування

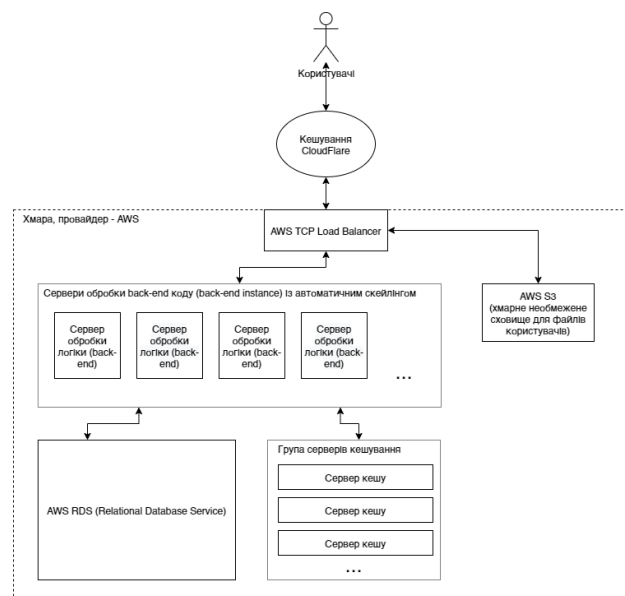


Рисунок 2.15 – Другий варіант структури масштабування

Варіанти масштабування розроблено згідно з сучасними тенденціями, врахуванням теоретичного аналізу у першому розділі та метриками, за якими виконується оцінювання. Щоб обрати найкращий варіант було проведено тестування навантаженням.

Для кожного варіанту структури проведено дослідження за 3 варіантами навантаження та оцінено поведінку за 4 метриками. Отже, загалом отримано 24 графіків для аналізу поведінки варіантів структури під різносторонніми навантаженнями. Результати наведено нижче.

Поведінку метрик першого варіанту структури при навантаженні «вдала рекламна кампанія» наведено на рисунках 2.16-2.19.

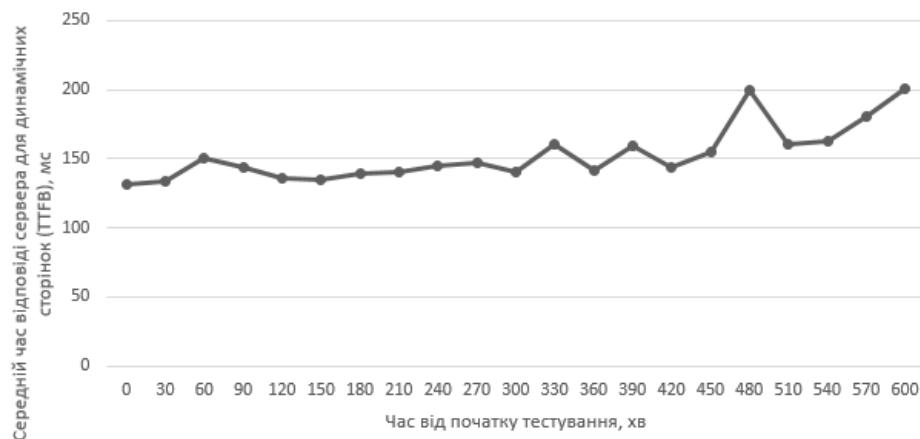


Рисунок 2.16 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

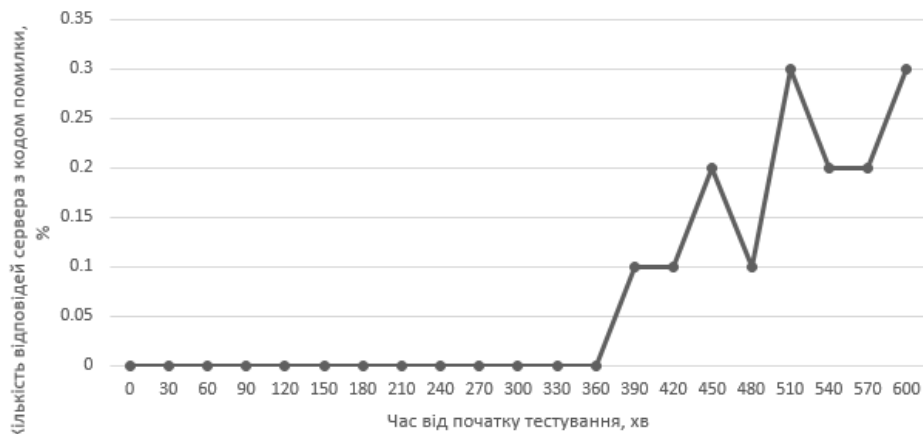


Рисунок 2.17 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

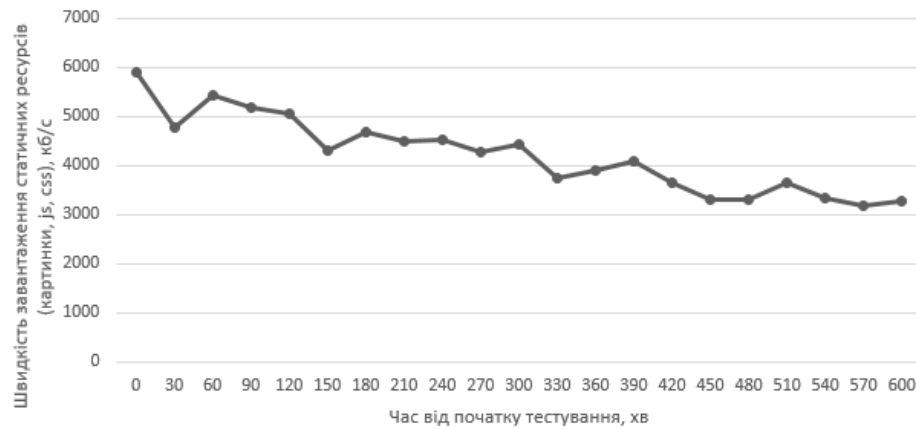


Рисунок 2.18 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

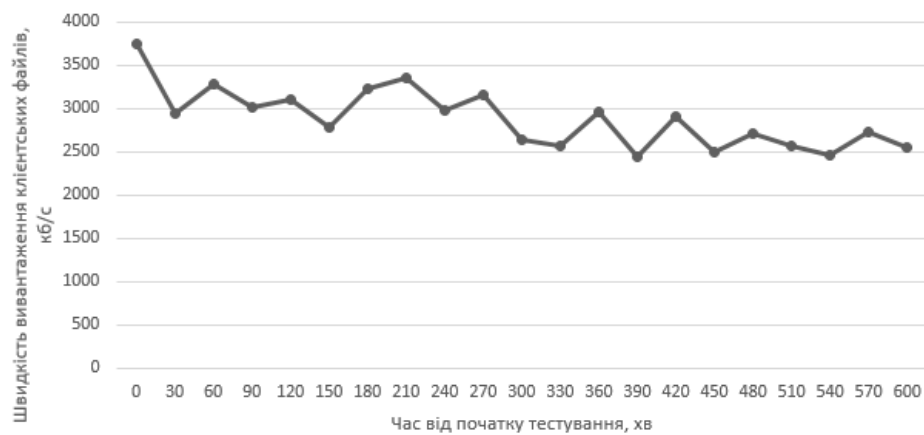


Рисунок 2.19 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Аналізуючи отримані результати можна сказати, що тестування пройдено успішно – поведінка майже не змінюється з поступовим збільшення навантаження.

Поведінку метрик першого варіанту структури при навантаженні «опублікували посилання на головній сторінці urk.net» наведено на рисунках 2.20-2.23.

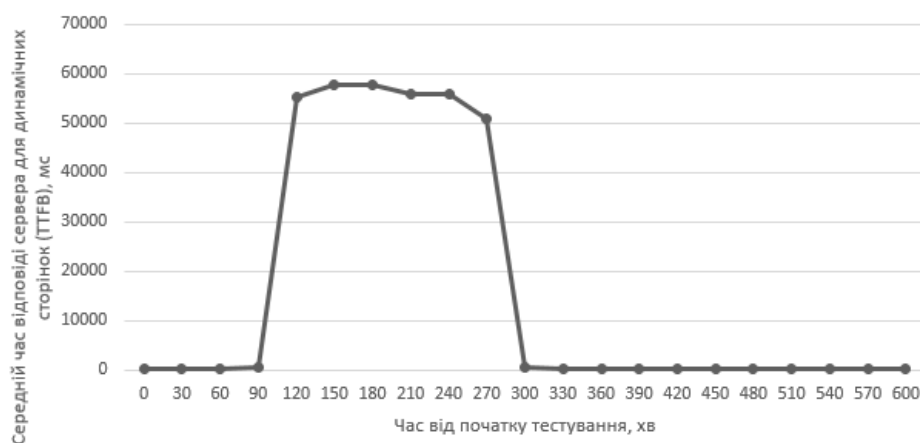


Рисунок 2.20 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

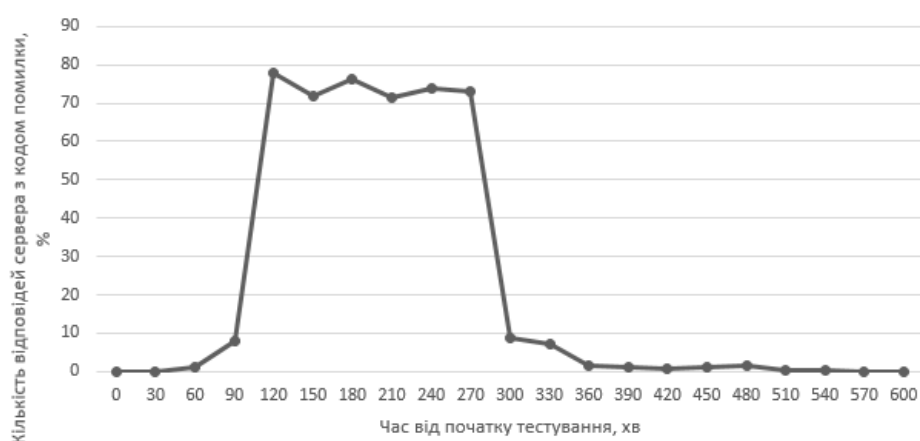


Рисунок 2.21 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

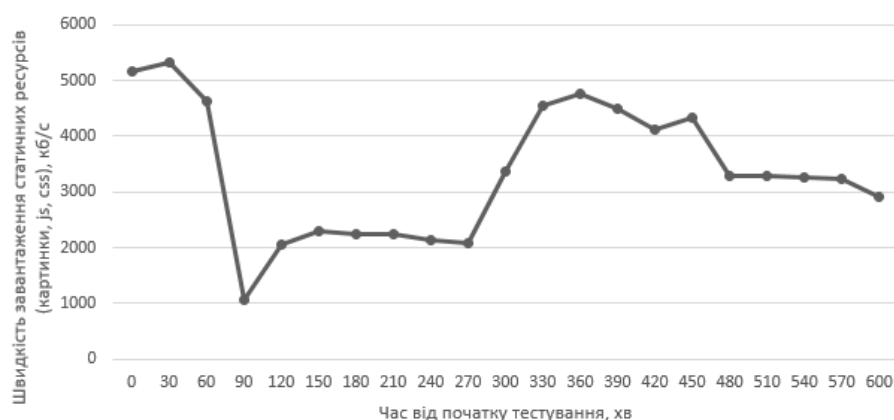


Рисунок 2.22 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

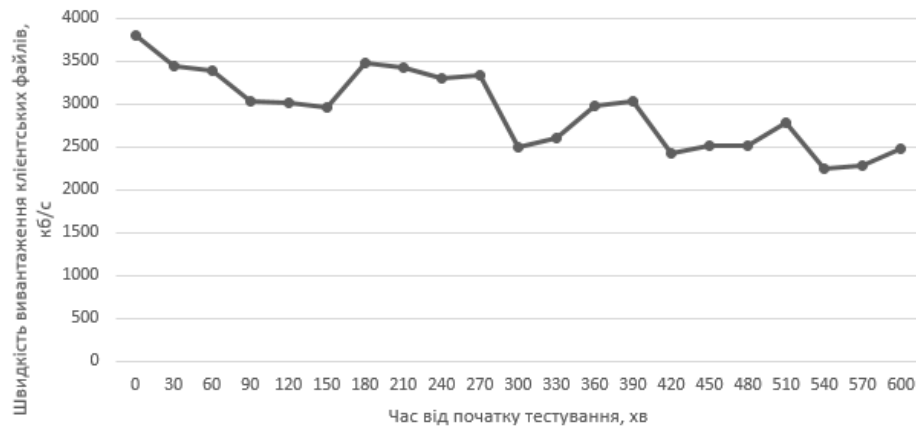


Рисунок 2.23 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Під час пікового навантаження перший варіант структури зовсім не справляється з імітаційною моделлю поведінки користувачів «опублікували посилання на головній сторінці urk.net», адже відповіді сервера займають аж до 1 хв часу та закінчуються помилкою в 80% випадків.

Поведінку метрик першого варіанту структури при навантаженні «потрібно виконати план до Нового Року» наведено на рисунках 2.24-2.27.

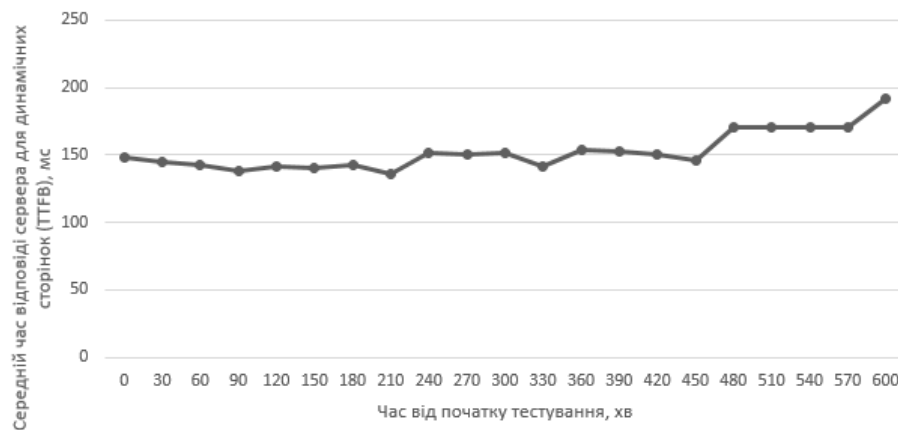


Рисунок 2.24 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

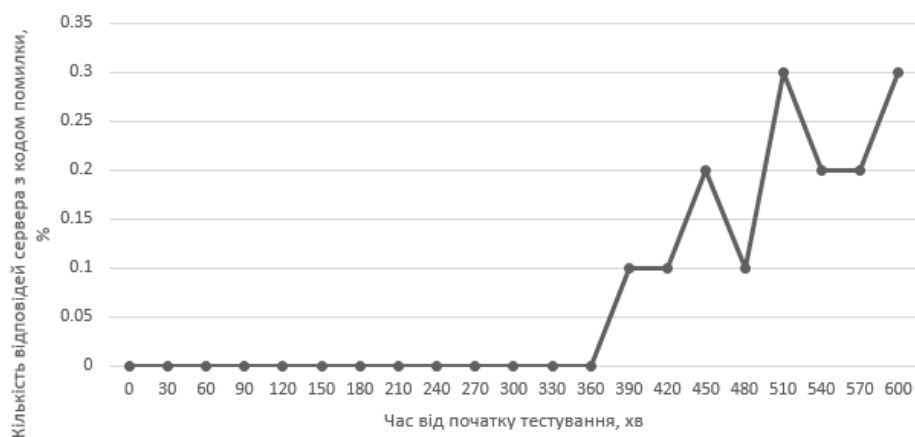


Рисунок 2.25 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

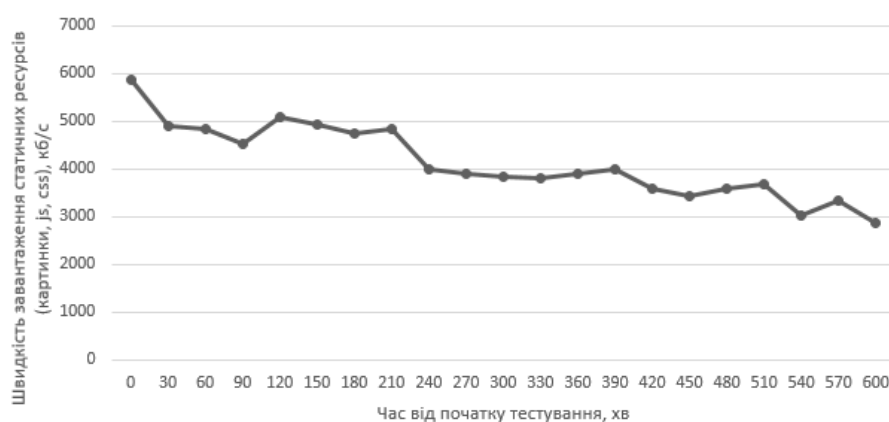


Рисунок 2.26 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

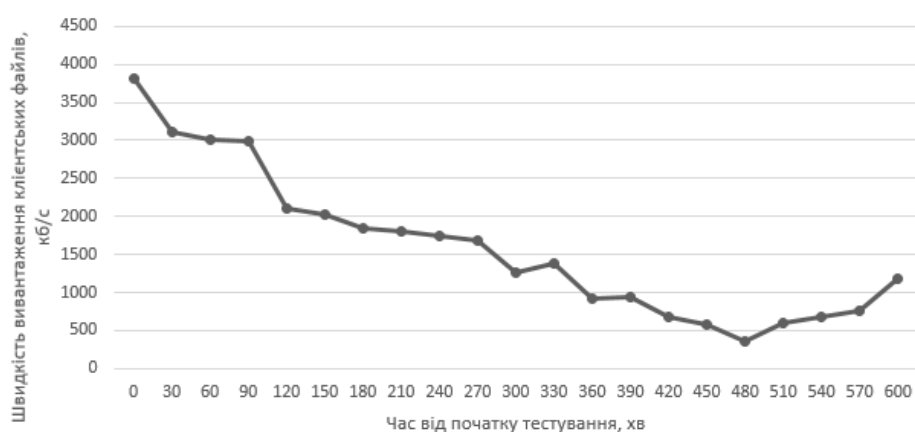


Рисунок 2.27 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Всі метрики в нормі, але швидкість вивантаження клієнтських файлів падає більше ніж у 10 разів. Такі результати призведуть до зниження зроблених замовлень користувачами, адже завантажувати великі за обсягом файли стає надзвичайно довго. Тобто перший варіант структури не справляється з таким видом навантаження.

Наступним кроком виміряно метрики для другого варіанту структури. Поведінку метрик другого варіанту структури при навантаженні «вдала рекламна кампанія» наведено на рисунках 2.28-2.31.

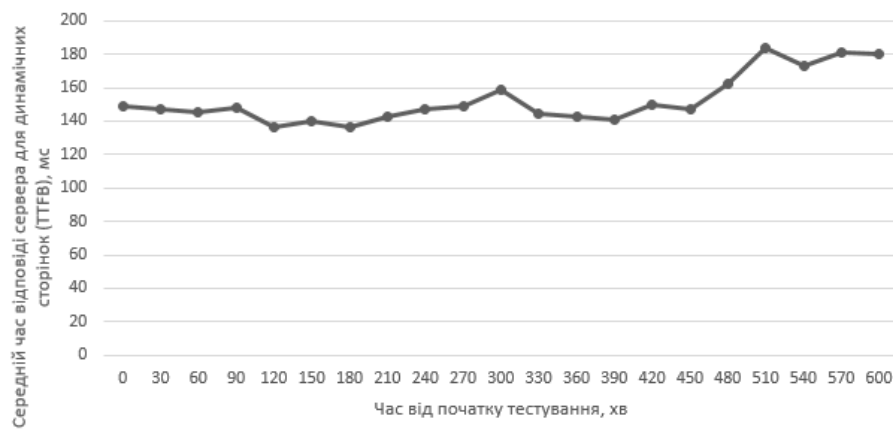


Рисунок 2.28 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

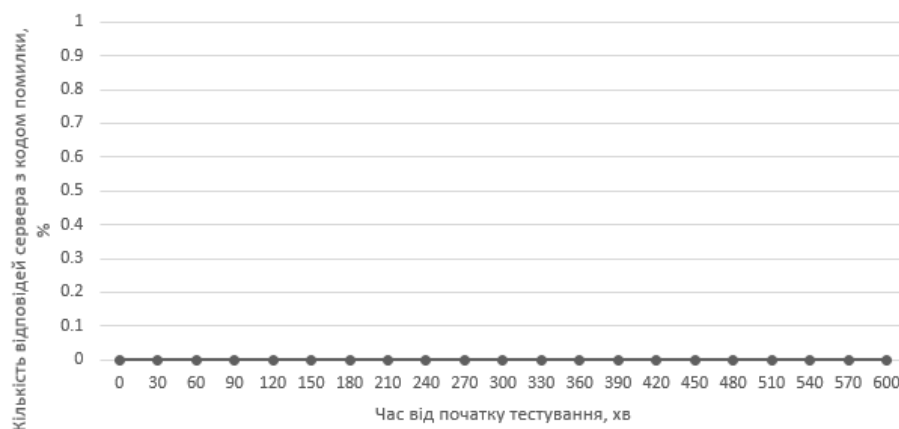


Рисунок 2.29 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

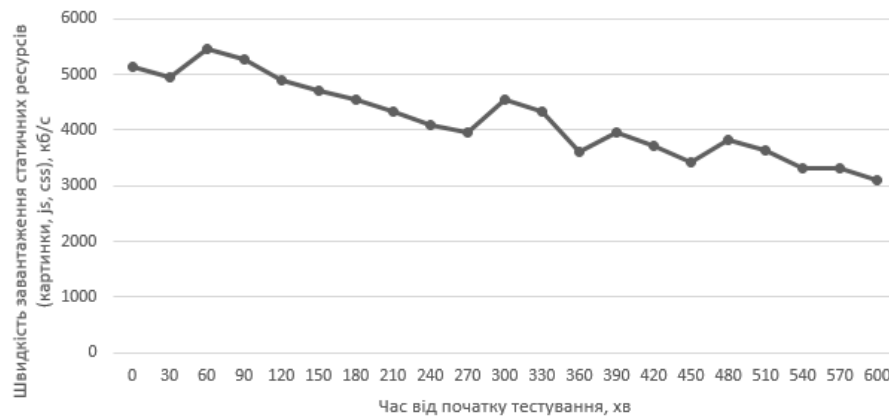


Рисунок 2.30 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

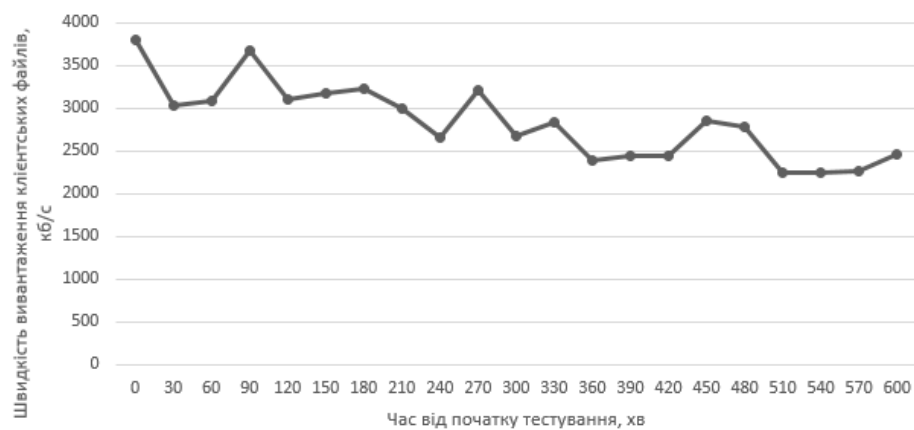


Рисунок 2.31 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Метрики під навантаженням майже не змінюються, тобто структура чудово справляється з таким видом навантаження.

Поведінку метрик другого варіанту структури при навантаженні «опублікували посилання на головній сторінці urk.net» наведено на рисунках 2.32-2.35.



Рисунок 2.32 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

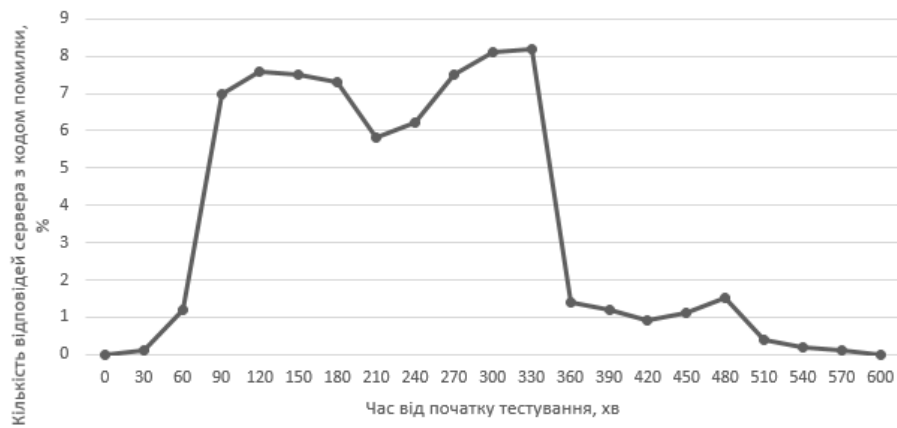


Рисунок 2.33 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

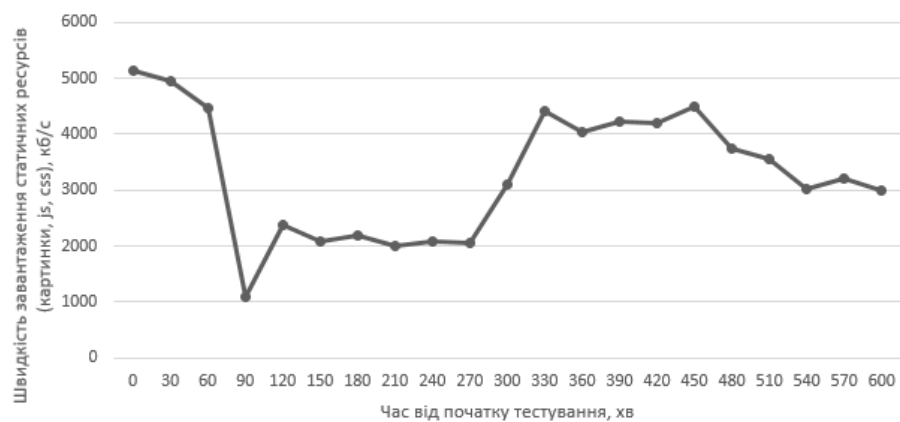


Рисунок 2.34 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

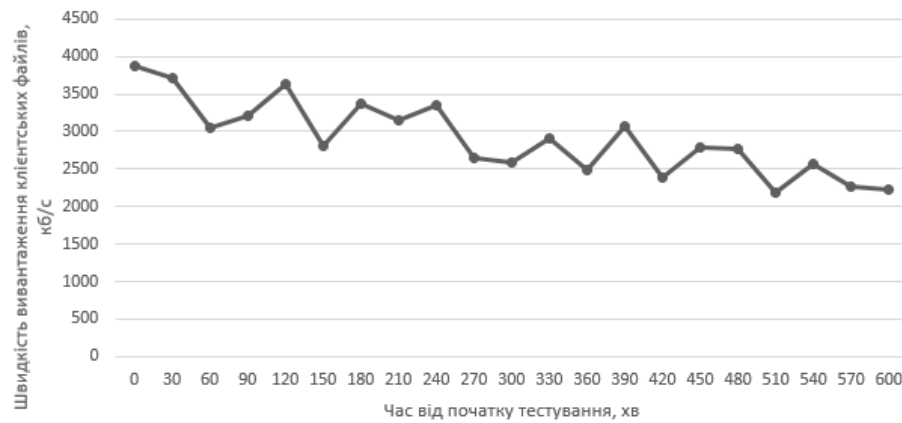


Рисунок 2.35 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Середній час відповіді трохи збільшується, але для кінцевих користувачів це не так помітно. Кількість помилок доволі суттєва, але перезавантаження сторінки майже завжди вирішить проблему. Отже, можна сказати що дана структура масштабування добре справляється з таким навантаженням.

Поведінку метрик другого варіанту структури при навантаженні «потрібно виконати план до Нового Року» наведено на рисунках 2.36-2.39.

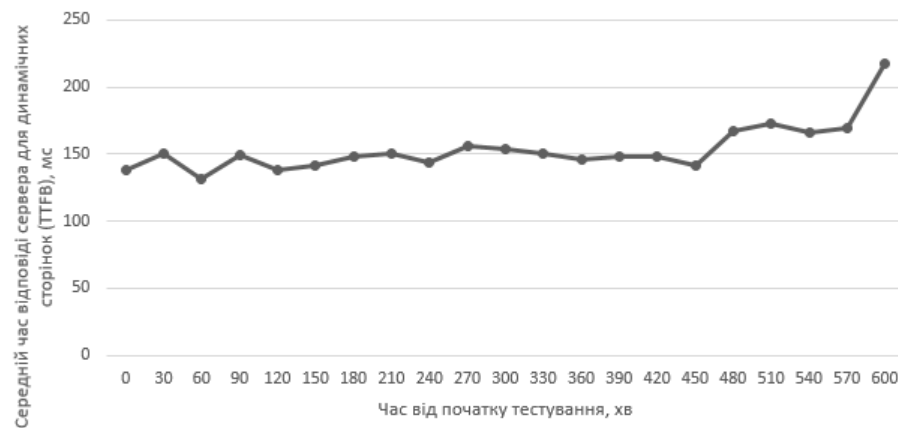


Рисунок 2.36 – Поведінка метрики «середній час відповіді сервера для динамічних сторінок» під час навантаження

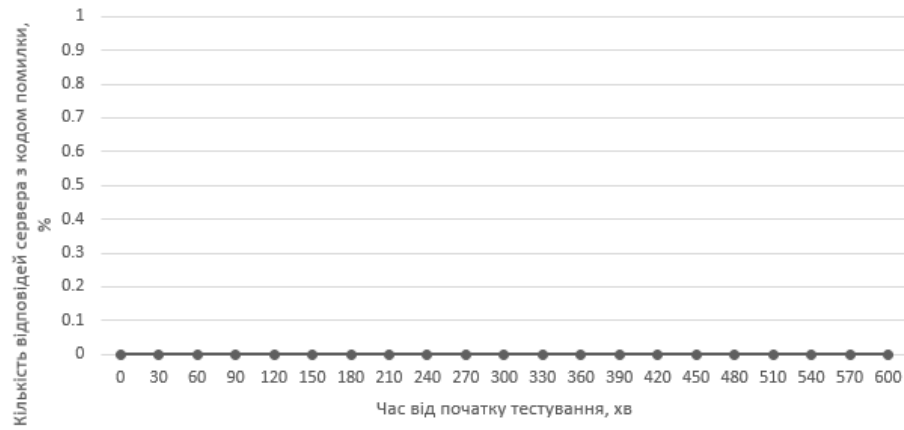


Рисунок 2.37 – Поведінка метрики «кількість відповідей сервера з кодом помилки» під час навантаження

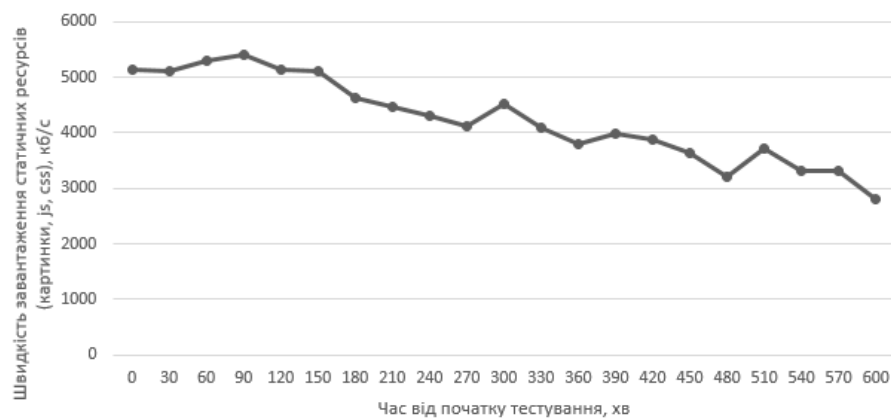


Рисунок 2.38 – Поведінка метрики «швидкість завантаження статичних ресурсів» під час навантаження

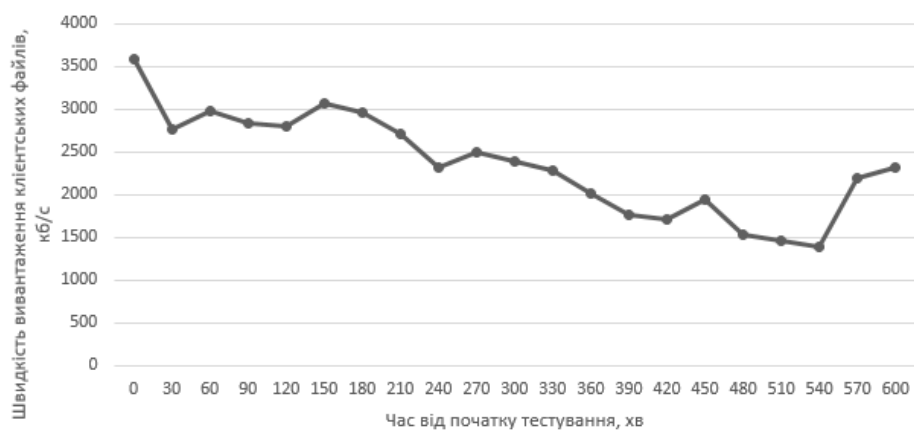


Рисунок 2.39 – Поведінка метрики «швидкість вивантаження клієнтських файлів» під час навантаження

Швидкість вивантаження клієнтських файлів в межах норми, клієнти не будуть відчувати перешкод. Всі інші показники майже не змінюються, а значить другий варіант структури відмінно справляється з таким варіантом навантаження.

Аналізуючи отримані результати, а саме виміряні метрики веб-порталу типу web-to-print під час різних навантажень, можна зробити висновок, що структура та інструменти, застосовані у 2-му варіанті масштабування найкраще справилися із задачею.

2.6 Моделювання технологічного процесу з урахуванням результатів дослідження

Важливо визначити найбільш впливові критерії, що впливають на вибір базового програмного забезпечення. Це було виконано за допомогою побудови домінантної ієрархічної впорядкованої моделі.

Сукупність критеріїв впливу на вибір базового набору програмного забезпечення для системи безперервної інтеграції становить множину елементів $C = \{c_1, c_2, \dots, c_n\}$, де:

- c_1 — якість та повнота технічної документації (Д);
- c_2 — складність впровадження (С);
- c_3 — спосіб, канали та якість технічної підтримки зі сторони постачальника (ТП);
- c_4 — можливості та складність створення конфігураційних файлів ланцюгів (КФ);
- c_5 — вартість впровадження, а також ліцензії чи підписки (В);
- c_6 — продуктивність та ефективність роботи, можливості масштабування системи на велику кількість серверів, якщо це не хмарна система (ПР);
- c_7 — можливості адміністрування, розподілення ролей, якість та повнота статистичних даних (А).

Проаналізувавши обрані критерії, було побудовано початковий орієнтовний граф, що відображає зв'язки між факторами впливу (рисунок 2.40).

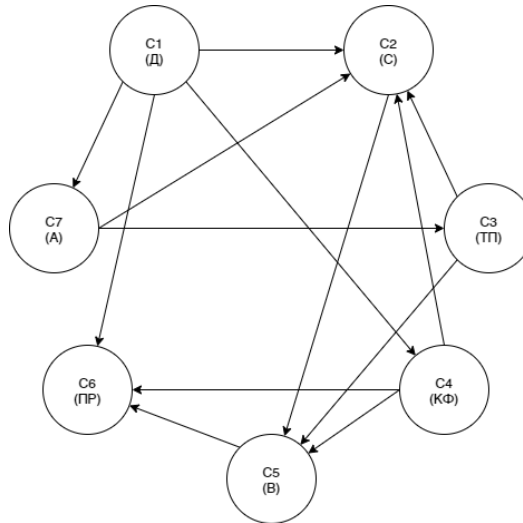


Рисунок 2.40 – Граф зв'язків між критеріями вибору параметрів, які впливають на вибір базового набору програмного забезпечення для системи безперервної інтеграції

Далі побудовано бінарну матрицю досяжності M' для множини вершин C , що представлена на рисунку 2.41.

$$M' = \begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 5 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 6 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Рисунок 2.41 – Бінарна матриця досяжності M' для множини вершин C

Наступним кроком було проведено аналіз матриці досяжності M' для множини вершин C за допомогою онлайн-сервісу Wolfram Alpha [53] та побудовано домінантну ієрархічну впорядковану модель критеріїв впливу (рисунок 2.42).

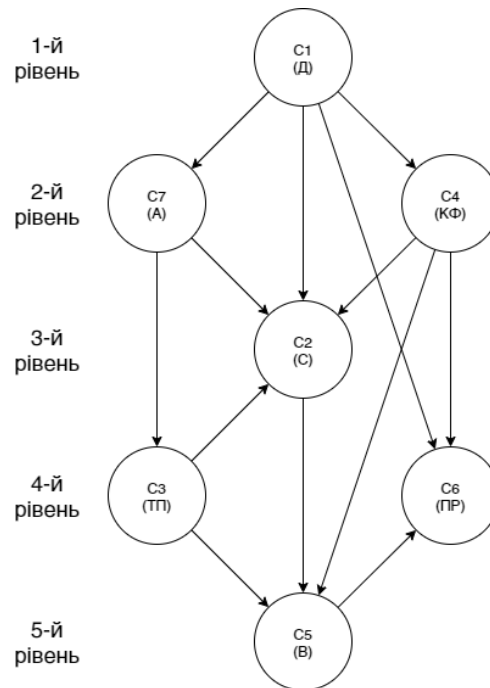


Рисунок 2.42 – Домінантна ієрархічна впорядкована модель критеріїв впливу

Для отримання вагових значень присвоюємо їм умовні числові значення. Нехай п'ятому рівню ієрархії відповідатиме число 10, а значення кожного вищого рівня збільшуватиметься в геометричній прогресії. Результатом таких припущень є вектор $\bar{I} = (160; 40; 20; 80; 10; 20; 80)$.

У результаті аналізу було виявлено, що найбільш впливовим чинником на вибір базового набору програмного забезпечення для системи безперервної інтеграції є якість та повнота технічної документації. Наступні параметри по ступеню впливу: можливості та складність створення конфігураційних файлів ланцюгів; можливості адміністрування, розподілення ролей, якість та повнота статистичних даних.

За результатами проведених досліджень, а саме: вибору раціонального базового програмного забезпечення (GitHub Actions) та структури масштабування, було створено архітектуру системи безперервної інтеграції для веб-порталів типу web-to-print. До уваги взято також всі проаналізовані джерела та теоретичну інформацію, тенденції розвитку. Результат представлено на рисунку 2.43.

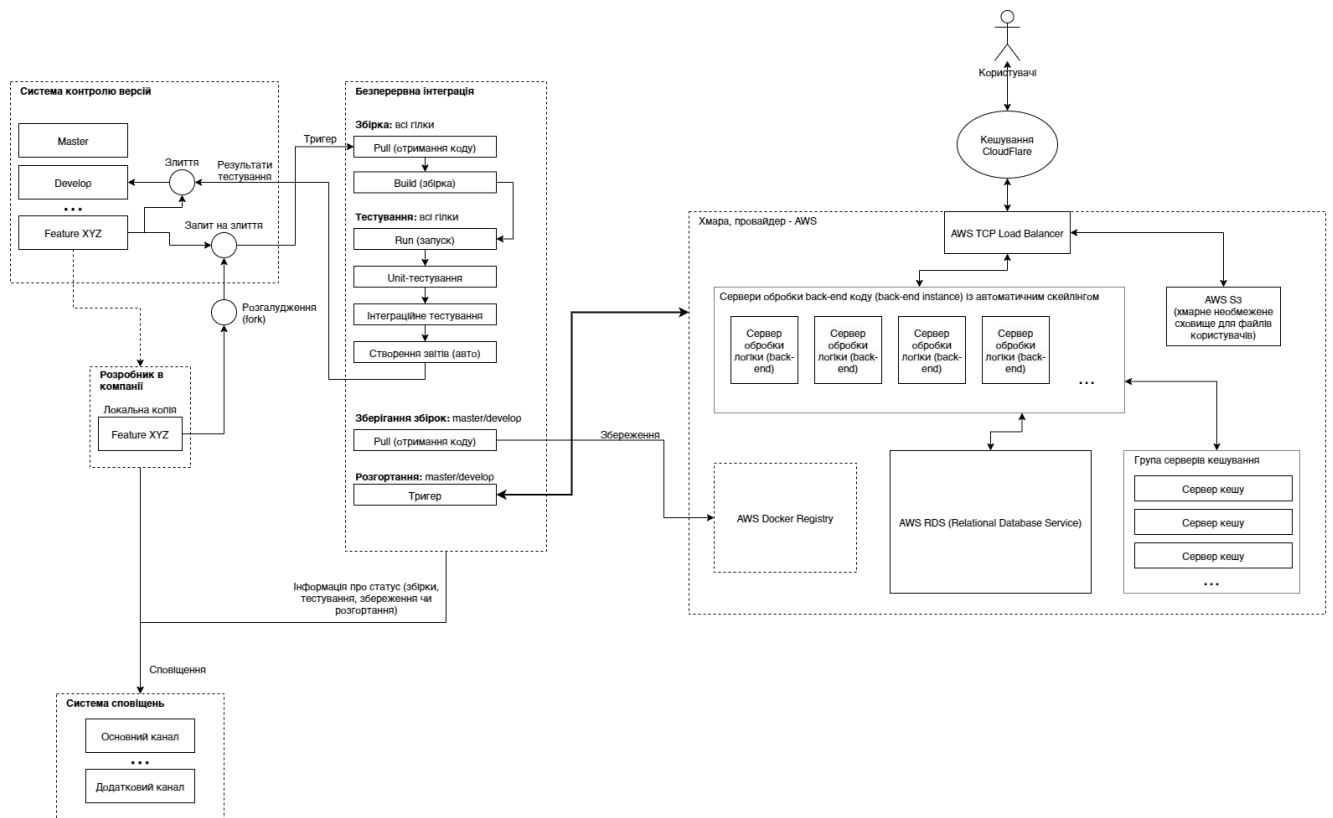


Рисунок 2.43 – Створена архітектура системи безперервної інтеграції для веб-порталів web-to-print

Висновки до другого розділу

На основі проведеного патентного пошуку проаналізовано актуальність теми дослідження за допомогою кумулятивної кривої публікації патентів та динаміки публікацій за роками. Було визначено, що тематика надзвичайно актуальна та стрімко розповсюджується на різні сфери. Кумулятивна крива публікації патентів за окремими напрямками дозволила виявити причину піку патентування в 2014 році – вихід системи контейнеризації Docker. А на теперішній момент основними драйверами галузі є автоматизація тестування та автоматизація збірки програмного коду.

Наступним кроком визначено об’єкт та предмет дослідження. Архітектура системи безперервної інтеграції для реалізації web-to-print та фактори, що впливають на будову відповідно.

Розроблено тестові файли (програмний код), що необхідні для проведення експерименту. Тестовий веб-портал запущено на основі сирцевого коду веб-порталу «Printlance», а скрипти тестування навантаження розроблено в середовищі Google Cloud Functions.

На основі семи критеріїв оцінено якість та обрано базове програмне забезпечення. Критеріями оцінки були: якість та повнота технічної документації; складність впровадження; спосіб, канали та якість технічної підтримки зі сторони постачальника; можливості та складність створення конфігураційних файлів ланцюгів; вартість (ліцензії чи підписки, впровадження); продуктивність та ефективність роботи, можливості масштабування системи; можливості адміністрування, розподілення ролей, якість та повнота статистичних даних. Обраним експертами програмним забезпеченням виявилось GitHub Actions.

Наступним кроком створено два варіанти структури масштабування для проведення дослідження навантаженням за допомогою 3 кривих імітаційних користувачів відносно часу. В результаті визначено структуру, яка краще справляється з навантаженням та рекомендовано до застосування у веб-порталах типу web-to-print.

Останнім етапом визначено найважливіші критерії впливу на вибір базового набору програмного забезпечення за допомогою домінантної ієрархічної впорядкованої моделі. А результатом досліджень стала створена архітектура системи безперервної інтеграції для веб-порталів типу web-to-print

РОЗДІЛ 3

ПРОЄКТНА ЧАСТИНА

3.1 Проєктування інженерно-технічного забезпечення виробництва

3.1.1 Промислове завдання на розробку проєкту

Центр зі створення веб-порталів створює та підтримує рішення web-to-print для замовників поліграфічної галузі. На основі аналізу діяльності центру було створено промислове завдання, яке наведено у таблиці 3.1.

Спільною рисою всіх типів веб-порталів є використання тих чи інших функцій web-to-print. Але кожен має свою специфіку та їх розробка відрізняється витратами часу на обробку текстової інформації, графічної інформації, розробку користувацьких інтерфейсів, налаштування системи безперервної інтеграції, розробку front-end та back-end частин.

Таблиця 3.1 – Промислове завдання

№	Тип веб-порталу	К-ть назв на рік	Сер. к-ть ілюстрацій	Обсяг макетів сторінок для верстки	Сер. відвідуваність, тис. осіб/міс.	Сер. к-ть переглядів сторінок за місяць	Середній обсяг однієї сторінки, Мб	Позиція держ-стандарту
1	2	3	4	5	6	7	8	9
1	Веб-портал типу web-to-print з підвищеним захистом даних користувачів	3	45	38	6,5	22	4,1	ДСТУ 7157:2010 Видання електронні. Основні види та вихідні відомості, ДСТУ EN ISO 14915-1:2009, ДСТУ EN ISO 14915-2:2013, ISO 9707:1991
2	Веб-портал для використання web-to-print державними структурами	11	25	60	29	7.5	2,9	
3	Корпоративний веб-портал поліграфії з деякими функціями web-to-print	26	350	45	0,4	31	1,9	

Кінець таблиці 3.1

№	Тип веб-порталу	К-ть назв на рік	Сер. к-ть ілюстрацій	Обсяг макетів сторінок для верстки	Сер. відвідуваність, тис. осіб/міс.	Сер. к-ть переглядів сторінок за місяць	Середній розмір однієї сторінки, Мб	Позиція держ-стандарту
4	Загальний інформаційний веб-портал з можливостями web-to-print	4	620	24	11,2	14	2,5	
5	Універсальний веб-портал web-to-print	2	400	120	140	19	2,2	

Наступним кроком йде визначення виробничо-технічних параметрів позицій промислового завдання (таблиця 3.1) для центру з розробки веб-порталів. Загальна структура та основні можливості у різних типів веб-порталів ідентичні, адже центр при розробці буде користуватися загальними напрацюваннями, а також намагатиметься широко застосовувати безкоштовне програмне забезпечення з відкритим сирцевим кодом.

Створення порталів починається зі складання ТЗ, в якому наводяться всі вимоги до роботи, функціональності, планованого максимального навантаження та інші вимоги за бажанням замовника. Структура модулів веб-порталу – одна із найважливіших частин технічного завдання.

Структуру основних модулів типового веб-порталу web-to-print з підвищеним захистом даних користувачів наведено на рисунку 3.1.

Структуру основних модулів типового веб-порталу для використання web-to-print державними структурами наведено на рисунку 3.2.

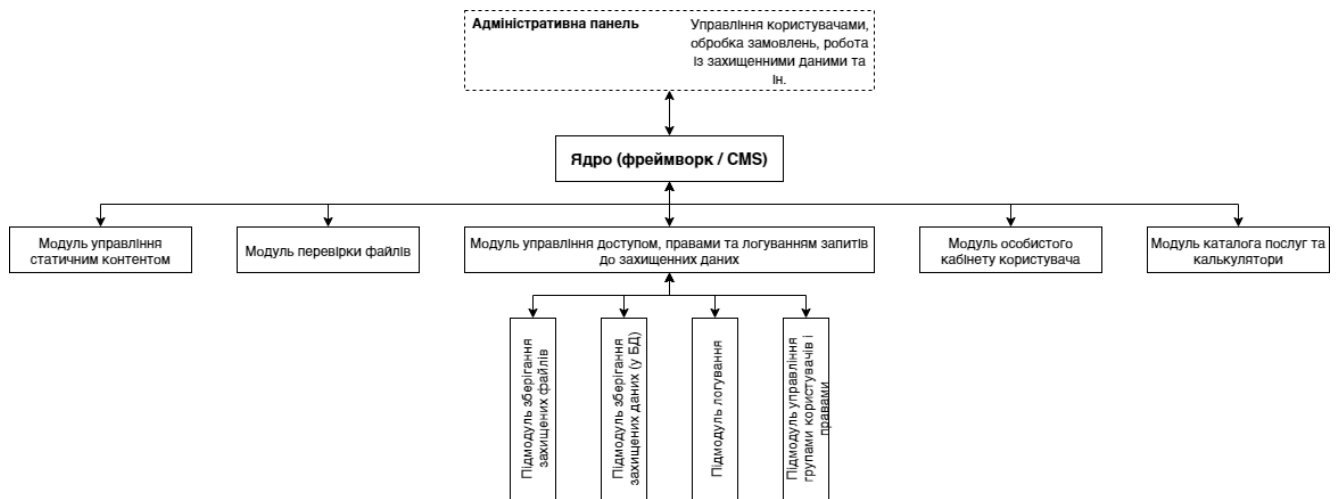


Рисунок 3.1 – Структура модулів типового веб-порталу web-to-print з підвищеним захистом даних користувачів

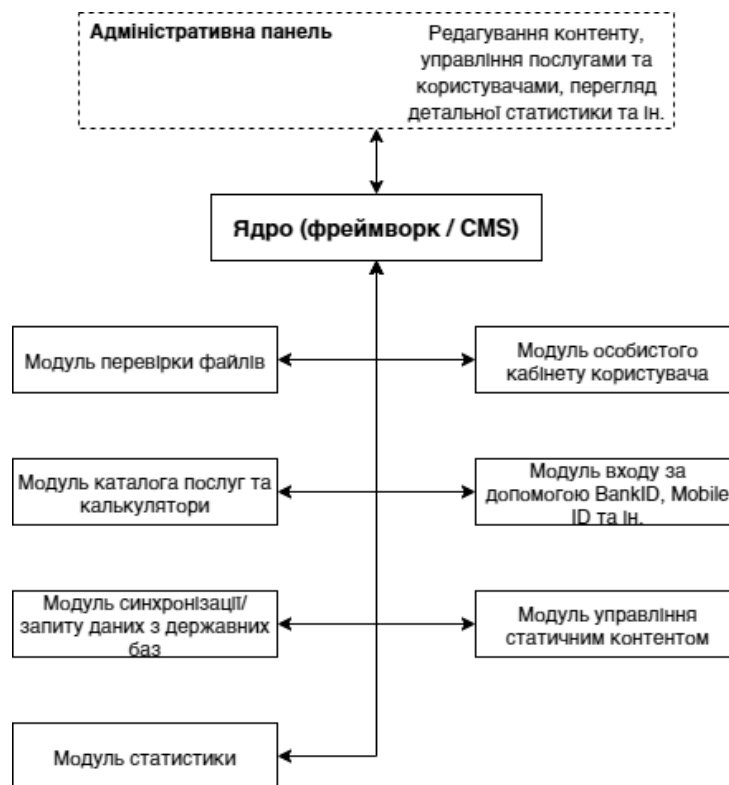


Рисунок 3.2 – Структура модулів типового веб-порталу для використання web-to-print державними структурами

Структуру основних модулів корпоративного веб-порталу поліграфії з деякими функціями web-to-print наведено на рисунку 3.3.



Рисунок 3.3 – Структура модулів типового корпоративного веб-порталу поліграфії з деякими функціями web-to-print

Структуру основних модулів типового загального інформаційного веб-порталу з можливостями web-to-print наведено на рисунку 3.4.

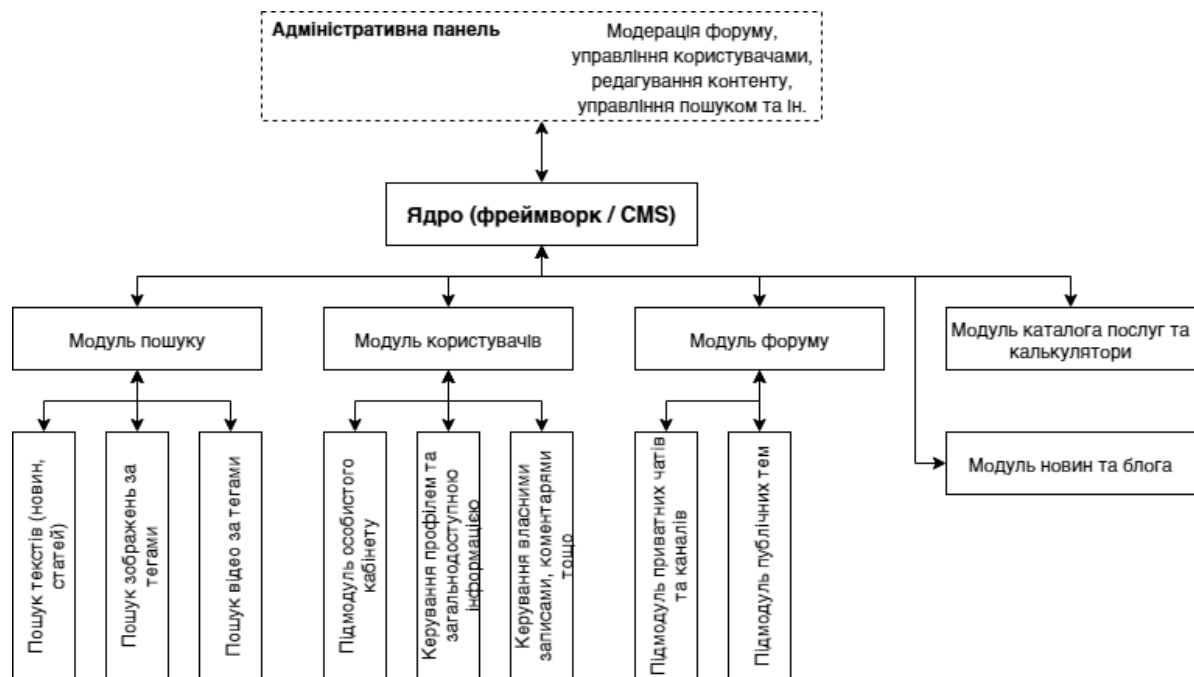


Рисунок 3.4 – Структура модулів типового загального інформаційного веб-порталу з можливостями web-to-print

Структуру основних модулів типового універсального веб-порталу web-to-print наведено на рисунку 3.5.

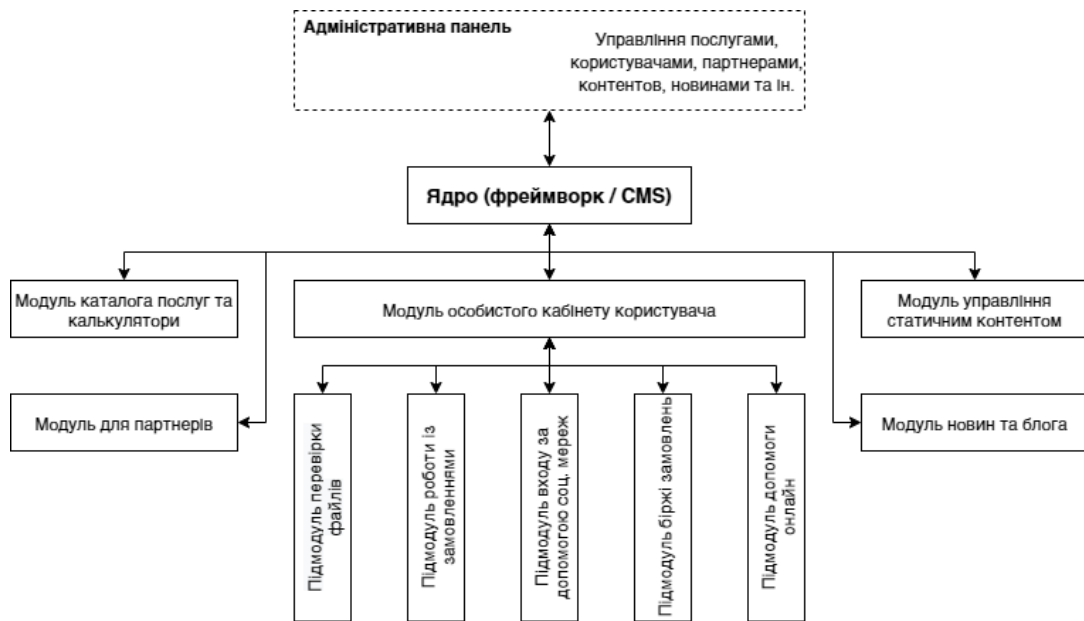


Рисунок 3.5 – Структура модулів типового універсального веб-порталу web-to-print

Виконавши аналіз структури основних модулів для кожного типу веб-порталу, було виявлено, що для всіх варіантів потрібна функціональна та потужна панель для адміністраторів. Окрім того, така панель повинна допомагати не лише в управлінні порталом за допомогою великої кількості функцій, а й забезпечувати високу швидкодію, адже за великої кількості відвідувачів створюється велика кількість даних, які необхідно обробляти.

Для виконання всіх потреб адміністративна панель повинна працювати за наступною схемою (рисунок 3.6):

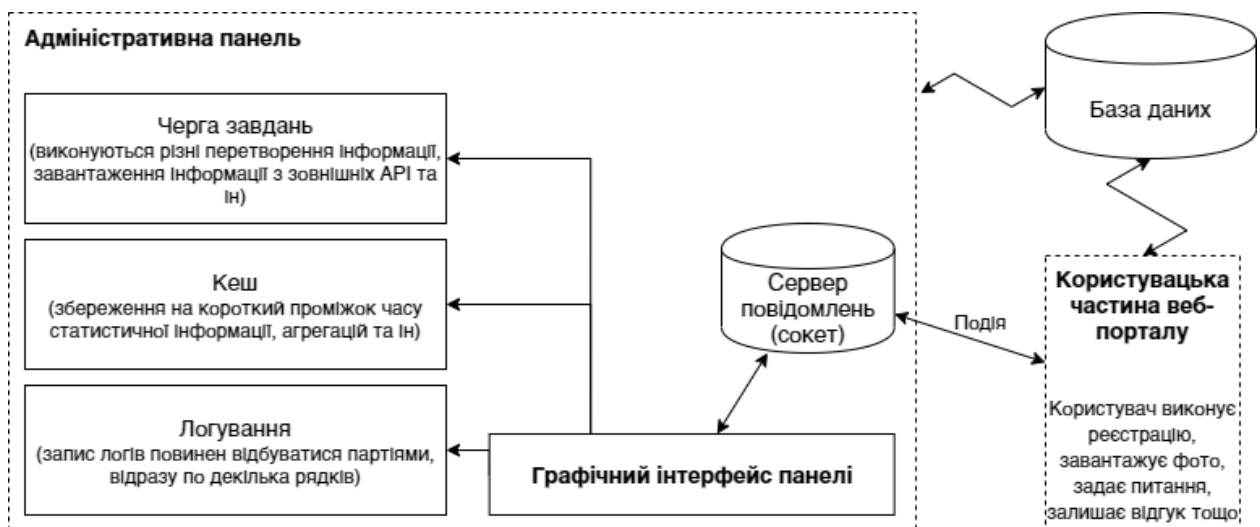


Рисунок 3.6 – Схема роботи панелі для адміністраторів

Деякі прототипи найбільш необхідних типового веб-порталу наведено на рисунках 3.7-3.8.

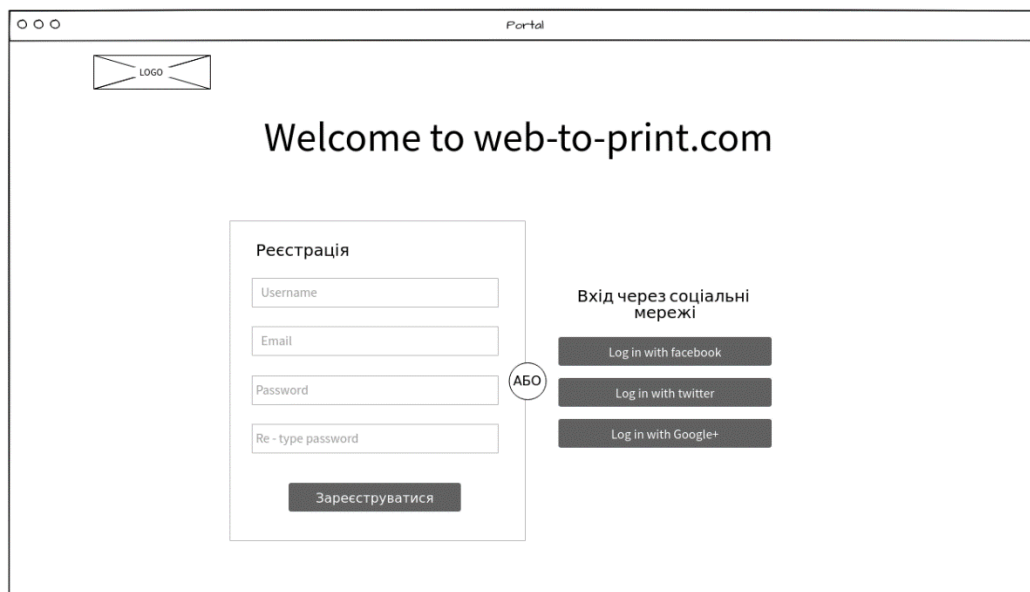


Рисунок 3.7 – Типова сторінка реєстрації

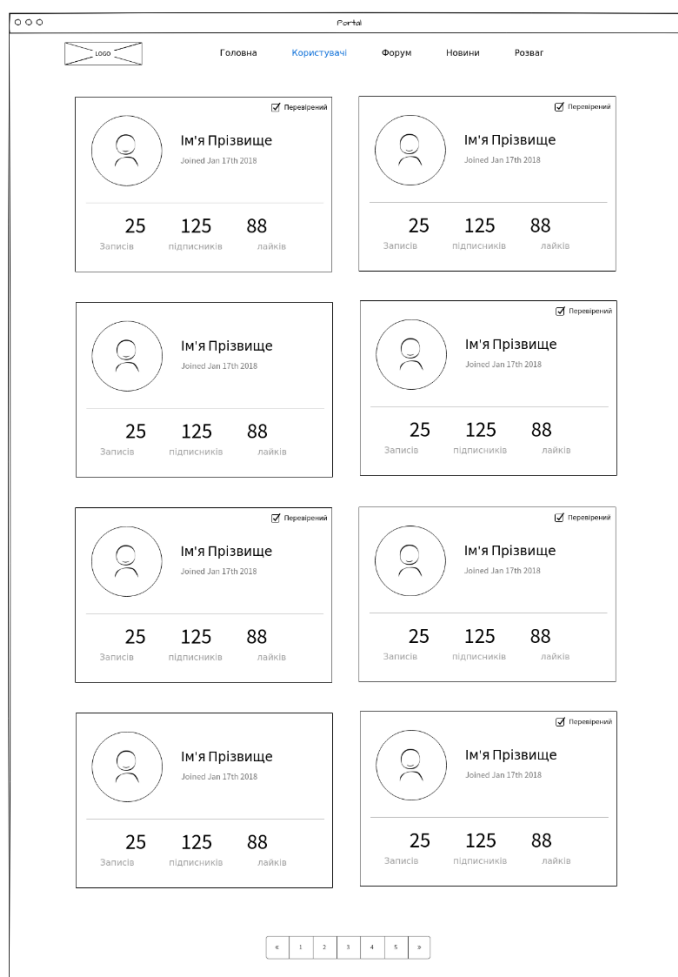


Рисунок 3.8 – Типова сторінка перегляду та пошуку серед користувачів (або компаній) веб-порталу

Надзвичайно важливою сторінкою на всіх веб-порталах, хоча, насправді, на кожному сайті, є контакти. Типову структуру сторінки контактів наведено на рисунку 3.9.

Рисунок 3.9 – Типова сторінка контактів на веб-порталі

3.1.2 Вибір технології та структури виробничих процесів

Для створення веб-порталів необхідно обрати технології back-end та front-end частин.

Вибір мови програмування залежить від її поширеності, існування та активності спільноти, швидкості коректувань багів та ще безлічі чинників. Для серверної частини варіантами є php, javascript (на базі nodejs), python, asp.net, тому що рештою варіантів практично не послуговуються для створення сайтів. Зрештою було обрано php, саме завдяки найбільшій спільноті, існуванню

бібліотек та фреймворків, швидкості роботи та пристосованості до веб-програмування.

Вибираючи технологію створення мультимедійного продукту було розглянуто чимало бібліотек, фреймворків, CMS. Враховуючи вимоги до якості сервісу, структури URL та структури сторінок, зрозуміло, що можливостей ніякої CMS не вистачить, тому що вони розраховані на стандартні рішення, по суті це конструктор з допоміжними плагінами. Отже, лишаються варіації створення з використанням фреймворків чи без будь-яких ключових функцій – за допомогою бібліотек. Отож, проаналізовано чотири технології створення:

- використовуючи тільки мову програмування php та бібліотеки;
- за допомогою фреймворку Laravel;
- за допомогою фреймворку Yii 2;
- використовуючи фреймворк Symfony.

Характеристики фреймворків наведено нижче, у таблиці 3.2.

Таблиця 3.2 – Порівняння характеристик фреймворків [54]

Характеристика	Фреймворк		
	Symfony	Laravel	Yii 2
Остання версія	4.1	5.7	2.0.15
Тип ліцензії	MIT	MIT	BSD
Версія PHP	>= PHP 5.5.9	PHP >= 7.0.0	>= PHP 5.4
Можливості локалізації	XML (XLIFF) CSV PHP	PHP arrays Gettext	PHP arrays Gettext database
Доступ до бази даних (ORM)	Doctrine 2, Propel (active record)	ELOQUENT ORM (active record)	Database Access Objects (DAO), Active Record (AR)
Dependency injection	+	+	-
Система шаблонів	PHP, Twig	Blade, PHP, Custom	PHP and Prado
Логування	+	+	-
XSS-захист	+	+	-
Модуль авторизації	+	+	-
Зручність, продуманість архітектури	7/10	10/10	8/10

Щоб вибрати найкращу технологію було застосовано аналітичний метод співставлення технологічних процесів «чорна скринька», який показано на рисунку 3.10.

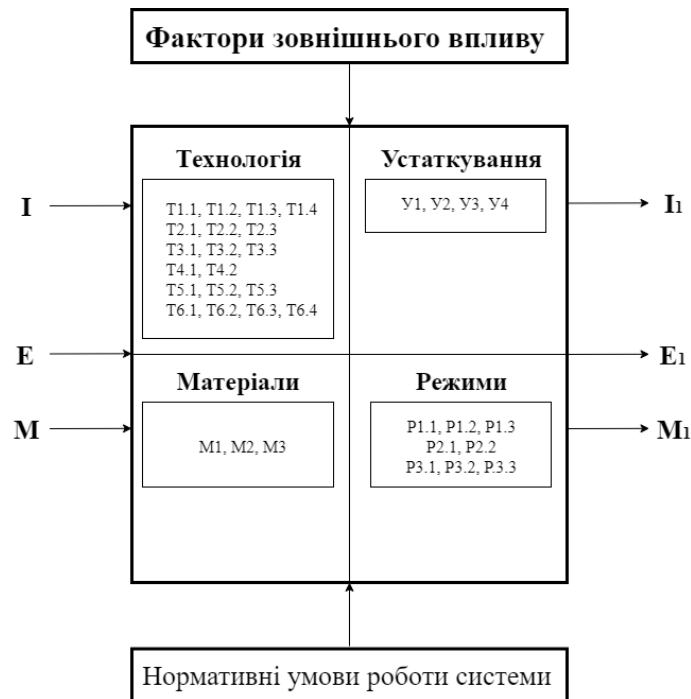


Рисунок 3.10 – Система «Чорна скринька»

Пояснення до рисунку 3.10:

I, I₁ — інформація, що вводиться (I) та виводиться (I₁) системою, метод та вигляд її подання; E, E₁ — енергія, яка необхідна для здійснення процесу (E) та витрачена (E₁); M, M₁ — матеріали до (M) та після (M₁) здійснення технологічного процесу; T1.1...Tn — варіанти технологічного процесу; У1...Уn — варіанти використання устаткування (обладнання); М1...Мn — матеріали; Р1.1...Рn — технологічні режими.

T1.1 — використання мови програмування php без каркасів (фреймворків); T1.2 — фреймворк Symfony (вся база компонентів); T1.3 — фреймворк Laravel та всі його composer-залежності (бібліотеки з онлайн бази); T1.4 — фреймворк Yii2 з усіма необхідними composer-залежностями;

T2.1 — зберігання інформації в СУБД mysql; T2.2 — зберігання інформації в СУБД postgresql; T2.3 — зберігання інформації у файлах;

T3.1 – «nginx» в якості front-end сервера та reverse proxy; T3.2 – «caddy» в якості front-end сервера та reverse proxy; T3.3 – «lighttpd» в якості front-end сервера та reverse proxy;

T4.1 – розгортання за допомогою файлу з інструкціями; T4.2 – розгортання за допомогою docker-контейнерів;

T5.1 – «redis» для зберігання завдань у черзі; T5.2 – «rabbitmq» для зберігання завдань у черзі; T5.3 – СУБД mysql для зберігання завдань у черзі;

T6.1 – використання ORM-бібліотеки; T6.2 – бібліотека для роботи з cookie – Cookie manager; T6.3 – бібліотека фільтрації SQL-ін'єкцій – SQL Sanitizer; T6.4 – бібліотека для роботи з датами – Carbon;

Y1 – сервер з конфігурацією: 512 мб ОЗУ, мінімум 30 гб жорсткого диску або краще SSD, процесор з як мінімум 1 ядром на 1.6 гГц;

Y2 – сервер з конфігурацією: 2 гб ОЗУ, мінімум 50 гб жорсткого диску або краще SSD, процесор з як мінімум 1 ядром на 2.4 гГц;

Y3 – сервер з конфігурацією: 2 гб ОЗУ, мінімум 60 гб жорсткого диску або краще SSD, процесор з як мінімум 1 ядром на 1.4 гГц;

Y4 – сервер з конфігурацією: 1 гб ОЗУ, мінімум 30 гб жорсткого диску або краще SSD, процесор з як мінімум 1 ядром на 1.6 гГц;

M1 – мережевий кабель, підключений до інтернету на швидкості 100 мбіт;

M2 – мережевий кабель, підключений до інтернету на швидкості 50 мбіт;

M3 – мережевий кабель, підключений до інтернету на швидкості 10 мбіт;

P1.1 – конфігурація для http-запитів у файлах .htaccess;

P1.2 – конфігурація для http-запитів у файлах «.conf», використовуючи директиви серверу nginx;

P1.3 – конфігурація для http-запитів у файлах «.conf» по правилам caddy;

P2.1 – зберігання сесійних даних користувачів сервісу у файлах спеціального формату;

P2.2 – зберігання сесійних даних користувачів сервісу в СУБД mysql;

P3.1 – запуск завдань в черзі за допомогою планувальника «cron»;

P3.2 – виконання необхідних завдань без черги – в момент запиту;

P3.3 – сторонній менеджер черги в хмарному середовищі AWS.

Щоб порівняти технології складемо ланцюги.

Для першої технології T1: T1.1 – T2.3 – T3.3 – T4.1 – T6.1 – T6.2 – T6.3 – T6.4; U1; M2; P1.1 – P2.1 – T3.3.

Для другої технології T2: T1.3 – T2.1 – T3.2 – T4.2 – T5.3; U2; M2; P1.3 – P2.2 – P3.1.

Для третьої технології T3: T1.4 – T2.2 – T3.1 – T4.1 – T5.1 – T6.4; U4; M1; P1.2 – P2.2 – P3.2.

Для четвертої технології T4: T1.2 – T2.1 – T3.2 – T4.2 – T5.2 – T6.2 – T6.4; U3; M2; P1.3 – P2.2 – P3.1.

При порівнянні вибраних технологій, устаткування, матеріалів та режимів було вибрано другий ланцюг із застосуванням фреймворку Laravel саме тому, що він коротший і технології, що використовуються, найбільш модерні.

Створення сервісу, звичайно, вимагає значно більше роботи з back-end частиною, проте не можна забувати front-end. Його можливо створити різними способами:

- за допомогою створення дизайну, верстки html/css та нативного JavaScript;
- за допомогою прототипу та відразу створення верстки з елементів Bootstrap, а більшість анімацій, валідацій зробити з плагінами JQuery.

Для визначення технології створення побудовано та проаналізовано циклограми технологічних процесів, що зображені на рисунку 3.11.

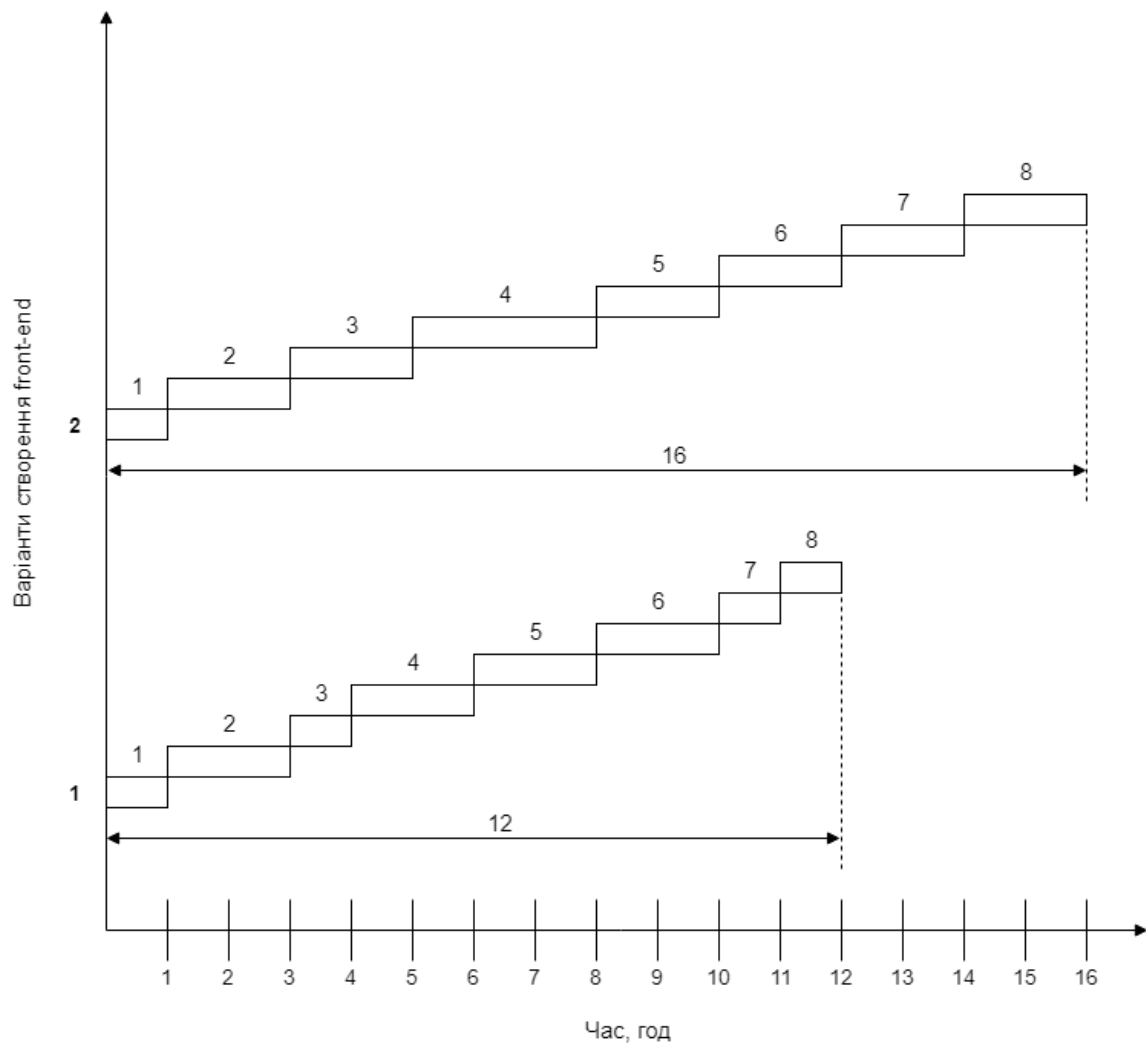


Рисунок 3.11 – Циклограми технологічних процесів створення front-end частини за допомогою: елементів Bootstrap та плагінів JQuery (1), верстки html/css та звичайного JavaScript (2)

Пояснення до рисунку 3.11:

1.1 – Налаштування середовища; 1.2 – Інсталяція залежностей; 1.3 – Підготовка необхідних компонентів та їх стилізація; 1.4 – Верстка основних сторінок; 1.5 – Верстка службових сторінок; 1.6 – Верстка внутрішніх сторінок; 1.7 – Налаштування плагінів та написання onload-коду; 1.8 – Тестування.

2.1 – Налаштування середовища; 2.2 – Інсталяція залежностей; 2.3 – Розробка базових компонентів; 2.4 – Верстка основних сторінок; 2.5 – Верстка службових сторінок; 2.6 – Верстка внутрішніх сторінок; 2.7 – Написання JavaScript-коду; 2.8 – Тестування.

Коефіцієнт технологічності системи $K_{\text{тех.}}$:

$$K_{\text{тех.}} = \frac{\sum P_{ij}}{m \times \sum N_{ij}}, \quad (3.1)$$

де $K_{\text{тех.}}$ – коефіцієнт технологічності системи;

$\sum P_{ij}$ – загальний час на виконання операцій технологічного циклу, год.;

m – устаткування;

$\sum N_{ij}$ – проєкція на вісь часу виконання операцій на циклограмі технологічного процесу, год.

Розрахуємо коефіцієнт технологічності системи для обох варіантів за формулою 3.1:

$$K_{\text{тех. (1)}} = \frac{1+2+1+2+2+2+1+1}{12} = 1$$

$$K_{\text{тех. (2)}} = \frac{1+2+2+3+2+2+2+2}{16} = 1$$

Коефіцієнт автоматизації:

$$K_{\text{авт.}} = \frac{P_{\text{авт.}}}{P_{\text{заг.}}}, \quad (3.2)$$

де $K_{\text{авт.}}$ – коефіцієнт автоматизації;

$P_{\text{авт.}}$ – число автоматизованих операцій;

$P_{\text{заг.}}$ – кількість всіх операцій.

Далі порахуємо ступінь автоматизації – кількість автоматизованих операцій до загальної кількості операцій, за формулою 3.2:

$$K_{\text{авт. (1)}} = \frac{3}{8} = 0,375$$

$$K_{\text{авт. (2)}} = \frac{2}{8} = 0,25$$

Рівень комп'ютеризації виробничих процесів – відношення комп'ютеризованих операцій до загального числа всіх операцій виробництва, визначається за наступною формулою:

$$K_{\text{комп.}} = \frac{P_{\text{комп.}}}{P_{\text{заг.}}}, \quad (3.3)$$

де $K_{\text{комп.}}$ – коефіцієнт комп'ютеризації;

$P_{\text{комп.}}$ – число комп'ютеризованих операцій;

$P_{\text{заг.}}$ – кількість всіх операцій.

Розрахунок за формулою 3.3 для обох варіантів створення:

$$K_{\text{комп. (1)}} = \frac{8}{8} = 1$$

$$K_{\text{комп. (2)}} = \frac{8}{8} = 1$$

Розглянувши одержані циклограми можливо зробити висновок, що застосування елементів Bootstrap набагато пришвидшує процедуру створення front-end частини сервісу. Отже, було вибрано технологію створення front-end частини за допомогою елементів Bootstrap та JQuery-плагінів.

Наступним кроком сформовано поопераційні карти найбільш трудомістких стадій виробничого процесу (таблиці 3.3-3.6).

Таблиця 3.3 – Поопераційна карта створення користувацьких інтерфейсів (UI/UX-дизайн)

№	Найменування технологічної операції	Технологічні розрахунки, хв
1	Пошук референсів	180
2	Вибір або створення шрифтового і колірного оформлення	90
3	Створення стилю та вигляду окремих елементів (UI-Kit)	920
4	Розробка макету головної сторінки та її погодження із замовником	600
5	Створення макетів всіх запланованих сторінок	960
6	Остаточні погодження із замовником, доопрацювання	420

Таблиця 3.4 – Поопераційна карта налаштування безперервної інтеграції

№	Найменування технологічної операції	Технологічні розрахунки, хв
1	Створення серверів та базові налаштування	150
2	Клонування репозиторію коду та створення файла із необхідними змінними середовища на серверах	30
3	Налаштування збірки та тестування контейнерів	240
4	Впровадження перенесення зібраних та протестованих контейнерів на робочі сервери	90
5	Тестування роботи та виправлення помилок	360

Таблиця 3.5 – Поопераційна карта розробки front-end частини

№	Найменування технологічної операції	Технологічні розрахунки, хв
1	Налаштування середовища	15
2	Інсталяція залежностей, підготовка необхідних компонентів та їх стилізація	90
3	Верстка основних сторінок	1420
4	Верстка службових та службових сторінок, знаходження та виправлення помилок	840
5	Налаштування плагінів та написання коду, що спрацьовує після завантаження сторінки	300
6	Тестування правильності роботи	180

Таблиця 3.6 – Поопераційна карта створення адміністративної панелі

№	Найменування технологічної операції	Технологічні розрахунки, хв
1	Визначення необхідного функціоналу, опис роботи	240
2	Створення правил доступу: списку ролей та прав користувачів	150
3	Створення швидких компонентів на основі Laravel Вакраск з використанням правил доступу	960
4	Тестування роботи та виправлення помилок	180
5	Навчання замовника та написання основної документації користувача адміністративної панелі	520

3.1.3 Принципові рішення щодо автоматизації технологічного процесу

3.1.3.1 Вибір апаратно-програмного забезпечення

Об Програмне забезпечення треба обирати починаючи з операційної системи. Беручи до уваги поставлене завдання, найкращою системою є Windows

10 Pro x64. Вона має досить невисокі системні вимоги, дає змогу користуватися звичними програмами та працює на різних типах процесорів, материнських плат, на відміну від MacOS. Системні вимоги показано у таблиці 3.7.

Таблиця 3.7 – Системні вимоги Windows 10 Pro x64 [55]

Назва вимоги	Мінімальна вимога	Рекомендовано
Мікропроцесор, тактова частота, ГГц	1	1,8
Ємність ОЗП, Гб	2	4
Ємність жорсткого диску або SSD, Гб	20	20
Дисплей, роздільна здатність в пікселях	800x600	1024x768

При створенні дизайну та у процесі верстки знадобиться програма для дизайну. Галузевим стандартом є програма Adobe Photoshop CC 2020, системні вимоги якої наведено у таблиці 3.8.

Таблиця 3.8 – Системні вимоги Adobe Photoshop CC 2020 [56]

Назва вимоги	Мінімальна вимога	Рекомендовано
Операційна система	Windows 7, 8.1, 10	Windows 10
Мікропроцесор	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 2 ГГц	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 3,4 ГГц
Ємність ОЗП, Гб	4	8
Ємність жорсткого диску або SSD, Гб	6,2	10,1
Дисплей, роздільна здатність в пікселях	1024x768	1280x800

Для створення логотипу, іконок потрібен векторний редактор графіки, наприклад, Illustrator CC 2020 компанії Adobe, котрий забезпечує сповна всі запити. Системні вимоги представлено у таблиці 3.9.

Таблиця 3.9 – Системні вимоги Adobe Illustrator CC 2020 [57]

Назва вимоги	Мінімальна вимога	Рекомендовано
Операційна система	Windows 7, 8.1, 10	Windows 10
Мікропроцесор	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 2 ГГц	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 3,4 ГГц
Ємність ОЗП, Гб	4	8
Ємність жорсткого диску або SSD, Гб	6,2	10,5
Дисплей, роздільна здатність в пікселях	1024x768	1280x800

Для написання коду потрібен швидкий редактор з підтримкою автодоповнення, з широкими можливостями підсвітки синтаксису, підтримкою імпорту опису функцій з бібліотек, можливостями роботи з системами контролю версій. Чи не найкращим для веб-програмістів є PhpStorm 2020.3, системні вимоги якого наведено у таблиці 3.10.

Таблиця 3.10 – Системні вимоги PhpStorm 2020.3 [58]

Назва вимоги	Мінімальна вимога	Рекомендовано
Операційна система	Windows 8.1, 10	Windows 10
Мікропроцесор	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 2 ГГц	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 3,4 ГГц
Ємність ОЗП, Гб	2	4
Ємність жорсткого диску або SSD, Гб	3,1	3,7
Дисплей, роздільна здатність в пікселях	1024x768	1280x800

Для програмування та дизайну (насамперед для веб-порталів) обов'язковим є процес пошуку інформації та матеріалів. Браузер повинен бути встановлений для того, щоб працювати з пошуковою системою «Гугл» тощо. Операційна система Windows 10 Pro x64 має вже вбудований Microsoft Edge 17, проте він не дуже добре працює та не має потрібних інструментів на панелі веб-

розробника. Тож, як браузер обрано Google Chrome 80 x64, системні вимоги якого наведено у таблиці 3.11.

Таблиця 3.11 – Системні вимоги Google Chrome 80 x64 [59]

Назва вимоги	Мінімальна вимога	Рекомендовано
Операційна система	Windows 7, 8.1, 10	Windows 10
Мікропроцесор	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 1 ГГц	Intel або AMD Athlon з підтримкою 64-розрядної версії та тактовою частотою 3,4 ГГц
Ємність ОЗП, Гб	0,5	2
Ємність жорсткого диску або SSD, Гб	0,256	4
Дисплей, роздільна здатність в пікселях	1024x768	1024x768

Крім того, потрібно вибрати додаткове програмне забезпечення. А це: управління контейнерами з програмами Docker, корпоративний месенджер Slack, версіонування коду GitHub, антивірус Bitdefender free та Mozilla Firefox 67 і Microsoft Edge 17 для тестування верстки.

Тож, було підібрано прикладне програмне забезпечення та наведено системні вимоги до кожної програми. Перелік програмних продуктів, що будуть встановлені на робочі станції наведено в таблиці 3.12.

Таблиця 3.12 – Мінімальні системні вимоги для програмного забезпечення [55-59]

Програмне забезпечення	Процесор	ОЗП, Мб	НЖМД, Мб	Дисплей	Додаткові пристрої
Операційна система					
Windows 10 Pro x64	1 ГГц чи вище	2048	20480	800x600	Клавіатура, маніпулятор
Програми зв'язку та роботи в Internet					
Google Chrome 80	Intel Core i3 чи вище	512	256	1280x1024	Клавіатура, маніпулятор
Mozilla Firefox 67	Intel Core i3 чи вище	512	512	1280x1024	Клавіатура, маніпулятор
Microsoft Edge 17	Intel Core i3 чи вище	256	256	1280x1024	Клавіатура, маніпулятор

Кінець таблиці 3.12

Програмне забезпечення	Процесор	ОЗП, Мб	НЖМД, Мб	Дисплей	Додаткові пристрої
Slack 3.4.0	Intel Core i3 чи вище	512	2048	1280x1024	Клавіатура, маніпулятор
Програми обробки ілюстраційної інформації					
Adobe Illustrator CC 2020	Intel Core i3 чи вище	4096	6114	1280x1024	Клавіатура, маніпулятор
Adobe Photoshop CC 2020	Intel Core i3 чи вище	4096	6114	1280x1024	Клавіатура, маніпулятор
Програми для програмування					
PhpStorm 2020.3	Intel Core i3 чи вище	2048	3072	1280x1024	Клавіатура, маніпулятор
Docker for Windows 2.0.0.3	Intel Core i3 чи вище	1024	2048	1280x1024	Клавіатура, маніпулятор
GitHub Desktop 1.6.6	Intel Core i3 чи вище	256	512	1280x1024	Клавіатура, маніпулятор
Програми захисту ПК					
Антивірус Bitdefender free 2020	Intel Core i3 чи вище	1024	2048	1280x1024	Клавіатура, маніпулятор

Для всіх працівників компанії (спираючись на характеристики основного та додаткового програмного забезпечення, встановленого системним адміністратором, а також, беручи до уваги підвищення системних вимог з оновленням програмного забезпечення на більш сучасне, використанням корпоративних месенджерів, браузерів, антивірусного пакету), було вирішено обрати однакові робочі станції. Це цілком раціонально та логічно, враховуючи відсутність потреби в обробці відео чи інших ресурсовитратних операціях. Придбання значної кількості ідентичного апаратного забезпечення зменшує вартість кожної одиниці, а також (за потреби) в майбутньому дозволить міняти на підприємстві структуру спеціалістів різних категорій. Ця тенденція є найбільш модерною. Таким чином компанії, які створюють мультимедійні продукти стараються у часи глобальних криз оптимізувати витрати.

Для ще більшої простоти та здешевлення вирішено обрати моноблоки, які одразу втілюють функції як системного блоку, так і монітору, а також мають комп'ютерну мишу та клавіатуру у комплекті. Для аналізу взято 3 варіанти від різних виробників та наведено їх основні характеристики в таблиці 3.13.

Таблиця 3.13 – Порівняльна характеристика моноблоків [60-62]

Характеристика	Модель моноблока		
	HP 200 G3 AiO	Asus Vivo AiO V241FAK-BA059D	Artline Business M61 v06
Діагональ (К1), дюймів	21.5	23.8	23.8
Роздільна здатність	1920x1080	1920x1080	1920x1080
Процесор (К2)	4 ядра, 2.2 ГГц	2 ядра, 3.9 ГГц	4 ядра, 1.8 ГГц
Обсяг оперативної пам'яті (К3), Гб	8	16	8
Обсяг HDD/SSD (К4), Гб	256	1000	240
Потужність блоку живлення (К5), Вт	65	90	48
Вага, кг	5.39	5.1	7
Розміри, мм	490 x 390 x 204	540 x 409 x 165	539 x 403 x 57
Ціна (К6), грн.	14699	21222	9777

Далі у таблиці 3.14 у зручному для розрахунку вигляді наведено порівнювані характеристики моноблоків. Опираючись на ці дані побудовано радіальний графік, що зображено на рисунку 3.12.

Характеристики моноблоків, що несуть позитивний ефект записуються згідно з формулою [7-8]:

$$a_i = \frac{10X_i^n}{X_j^{max}}, \quad (3.4)$$

де a_i – приведені позначення;

X_i^n – поточне значення;

X_j^{max} – максимальне значення для всіх варіантів.

Якщо характеристика є негативною, слід використати іншу формулу:

$$a_i = 10 \left(\frac{X_j^{max} - X_j^n}{X_j^{max}} \right) \quad (3.5)$$

Таблиця 3.14 – Показники якості устаткування

Показник якості	Модель моноблока			Коефіцієнт вагомості, a_i	Статус показника
	<i>HP 200 G3 AiO</i>	<i>Asus Vivo AiO V241FAK-BA059D</i>	<i>Artline Business M61 v06</i>		
K ₁	9	10	10	0,1	Позитивний
K ₂	10	8	8	0,25	Позитивний
K ₃	5	10	5	0,15	Позитивний
K ₄	4	10	4	0,1	Позитивний
K ₅	5	0	3	0,1	Негативний
K ₆	6	0	8	0,3	Негативний

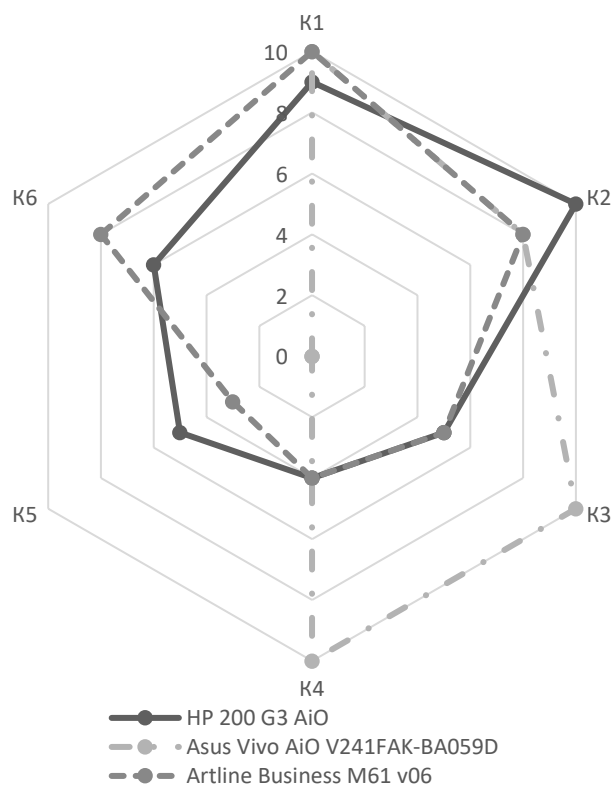


Рисунок 3.12 – Радіальний графік з вибору моноблока

Пояснення до рисунку 3.12:

K1 – діагональ;

K2 – процесор;

K3 – обсяг оперативної пам'яті;

K4 – обсяг HDD/SDD;

K5 – потужність блоку живлення;

K6 – ціна.

Для різних моделей моноблока розраховується інтегральний показник раціональності (за допомогою переліку основних характеристик (таблиця 3.13) та побудованого радіального графіка) за наведеною формулою:

$$S_1 = \frac{1}{2} \sin\left(\frac{360^\circ}{n-1}\right) \cdot \left(\sum_{i=1}^{n-1} a_i \cdot a_{i+1} + a_1 \cdot a_n\right), \quad (3.6)$$

де S_i – площа багатокутника для розрахунку інтегрального показника;

n – загальна кількість характеристик [8].

Інтегральний показник раціональності для першого моноблока (HP 200 G3 AiO) за формулою (3.6):

$$S_1 = \frac{1}{2} \sin\left(\frac{360^\circ}{6-1}\right) \cdot 7.145 = 3.4$$

Інтегральний показник раціональності для другого моноблока (Asus Vivo AiO V241FAK-BA059D) за формулою (3.6):

$$S_2 = \frac{1}{2} \sin\left(\frac{360^\circ}{6-1}\right) \cdot 6.5 = 3.09$$

Інтегральний показник раціональності для третього моноблока (Artline Business M61 v06) за формулою (3.6):

$$S_3 = \frac{1}{2} \sin\left(\frac{360^\circ}{6-1}\right) \cdot 7.04 = 3.35$$

На основі порівняння розрахованих інтегральних показників раціональності для різних моноблоків, бачимо, що найкращим є HP 200 G3 AiO.

3.1.3.2 Організаційна структура виробництва

Для вдалого виконання складних проєктів згідно промислового завдання необхідно мати професійну команду працівників. В цю команду входитимуть: копірайтери, які підготовлюють якісний контент для SEO-просування у пошукових системах; спеціалісти з підготовки графіки, що знаходять та згідно вимог готують графічний матеріал для використання в адміністративній панелі веб-порталів; UI- та UX-дизайнери, які займаються створенням якісної візуалізації задуманих ідей (створюють візуальні інтерфейси, які спочатку

тестуються на цільовій аудиторії); back-end розробники, що створюють логіку роботи (серверний код, php), програмують функціональність та зв'язують візуальні елементи з діями та запитами до бази даних; front-end розробники, які верстають та за допомогою javascript-коду створюють складні анімації та інтерактивні інтерфейси, реалізуючи задуми UI-дизайнерів; тестувальники обов'язково необхідні, адже неможливо створити новий продукт зовсім без помилок, а якість необхідна висока; спеціалісти з безперервної інтеграції відіграють важливу роль налаштовуючи роботу системи та автоматичне тестування й оновлення коду; системний адміністратор слідкує за роботою локальний серверів та мережі, допомагає з проблемами у колег.

Щоб організувати роботу підприємства необхідно мати посади креативного та технічного директорів, а також директора по роботі з контентом. Звісно, потрібні й фахівці по роботі з клієнтами, а також бухгалтер.

Організаційну структуру виробництва наведено на рисунку 3.13.



Рисунок 3.13 – Організаційна структура виробництва

3.1.3.3 Основні характеристики проекту та його цілі

На основі визначених тенденцій розвитку, структури підприємства і промислового завдання визначено головні особливості проекту:

- використання сучасного програмного забезпечення для систем безперервної інтеграції разом із спеціально розробленою архітектурою саме під портали типу web-to-print дає багато конкурентних переваг, зокрема: швидкість розробки, висока та перевірена якість, дотримання найвищих стандартів безпеки та ін.;

- спеціалізація на сегменті порталів типу web-to-print дозволяє створити велику базу напрацювань, які можна використовувати для нових проєктів, це не було б можливим за широкої спеціалізації;
- приділення уваги не лише користувачам, а й адміністраторам веб-порталу (працівники зі сторони замовника) дозволяє швидко реагувати на будь-які проблемні чи аварійні ситуації.

Ціль проєкту – підприємство, що спеціалізується на розробці сучасних веб-порталів типу web-to-print та є лідером в своїй ніші.

На рисунку 3.14 представлена узагальнена блок-схема технологічного процесу створення веб-порталу згідно зі структурою підприємства і виробничою програмою.

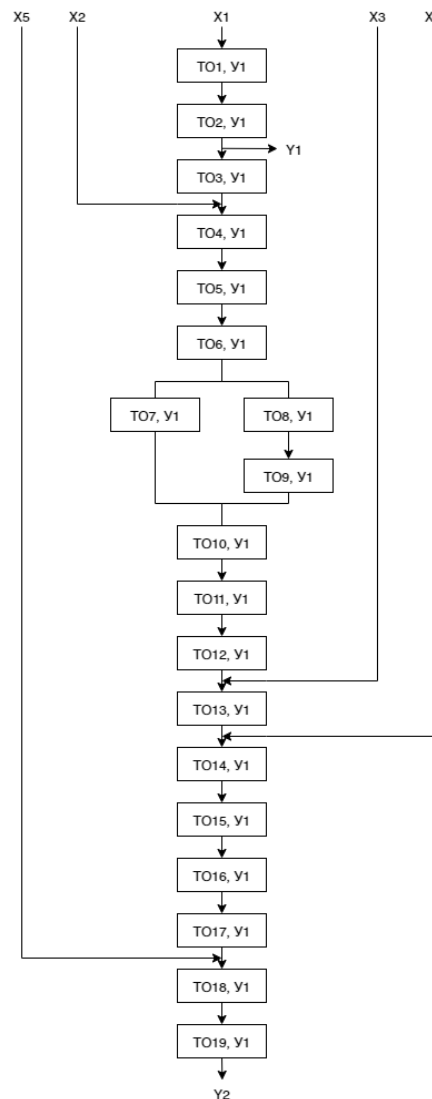


Рисунок 3.14 – Узагальнена блок-схема технологічного процесу створення веб-порталу

Пояснення до рисунку 3.14:

X1 – побажання замовника, референси; X2 – корпоративний аккаунт в системі контролю версій програмного забезпечення GitHub; X3 – аккаунти в сервісах, їх документація API, дані служб підтримки тощо; X4 – корпоративний аккаунт в Google; X5 – текстова та графічна інформація для наповнення веб-порталу (контент).

Y1 – готове технічне завдання; Y2 – розроблений та опублікований в мережу інтернет веб-портал.

TO1 – визначення цілей, розробка вимог, Y1 – моноблок HP 200 G3 AiO з операційною системою Windows 10 (64 бітна); TO2 – створення технічного завдання на розробку веб-порталу; TO3 – визначення дедлайну, підписання угоди; TO4 – ініціалізація проєкту (кодової бази) та створення конфігурацій для контейнерів Docker; TO5 – налаштування системи безперервної інтеграції та написання конфігураційних файлів; TO6 – проєктування структури сторінок та їх URL-адрес; TO7 – проєктування інтерфейсів та розробка дизайну; TO8 – створення опису API для подальшої роботи front-end розробників; TO9 – реалізація API згідно створеної документації; TO10 – програмування front-end частини; TO11 – попереднє тестування для виявлення основних помилок в коді; TO12 – виправлення помилок, знайдених при попередньому тестуванні; TO13 – налаштування інтеграцій з різними, необхідними для порталу, сервісами; TO14 – налаштування систем аналітики типу Google Analytics, теплових карт та ін.; TO15 – розгортання на основному сервері; TO16 – повне тестування роботи веб-порталу; TO17 – виправлення помилок, знайдених при тестуванні; TO18 – публікація в мережі інтернет (додавання в індекс пошукової системи Google) та наповнення початковим контентом; TO19 – здача проєкту замовнику та підписання акту прийому-передачі.

3.1.4 Розрахунок розгорнутого промислового завдання

Трафік, спожитий користувачами веб-порталу у гігабайтах на місяць, було розраховано за формулою 3.7, яку створено на підставі розрахунку обсягу спожитого трафіку [63-65]. Результати розрахунків представлено нижче.

$$K_{Гб} = \frac{K_{відв./міс.} \times K_{стор.} \times K_{с. розм.}}{1024}, \quad (3.7)$$

де $K_{Гб}$ – кількість використаного трафіку у гігабайтах;

$K_{відв./міс.}$ – кількість відвідувачів в місяць;

$K_{стор.}$ – середня кількість переглядів сторінок на відвідувача;

$K_{с. розм.}$ – середній розмір однієї сторінки веб-порталу.

Площа ілюстрацій вираховується за формулою:

$$S_{іл} (мп) = \frac{N_{іл} \times K_{н} \times N \times k}{1000 \times 1000}, \quad (3.8)$$

де $N_{іл}$ – кількість ілюстрацій для веб-порталу;

$K_{н}$ – кількість назв у виробничій програмі;

$N \times k$ – розмір у пікселях (розмір ілюстрації прийнятий за 1920x800 px).

Кількість текстової інформації:

$$O_{текст.} (Мбайт) = N_{стор} \times K_{н} \times N_{м}, \quad (3.9)$$

де $N_{стор}$ – кількість знаків на одній сторінці;

$K_{н}$ – кількість назв у виробничій програмі;

$N_{м}$ – кількість макетів сторінок.

Зважаючи на вищесказане, розроблено розгорнуте промислове завдання представлено у таблиці 3.15.

Таблиця 3.15 – Розгорнуте промислове завдання

№	Тип веб-порталу	К-ть назв на рік	Сер. к-ть ілюстрацій	Обсяг макетів сторінок для верстки	Сер. відвідуваність, тис. осіб/міс.	Сер. к-ть переглядів сторінок за місяць	Середній розмір однієї сторінки, Мб	Спожитий трафік, Гб/міс	Площа ілюст. інф., мп	К-ть текст. інф., тис. зн.	Додаткові можливості
1	2	3	4	5	6	7	8	9	10	11	12
1	Веб-портал типу web-to-print з підвищеним захистом даних користувачів	3	45	38	6,5	22	4,1	572,6	207,36	1694	Можливість хмарного захищеного зберігання важливих файлів чи документів (хмарний диск)
2	Веб-портал для використання web-to-print державними структурами	11	25	60	29	7.5	2,9	616	422,4	235	Вхід за допомогою BankID, Mobile ID та ін.
3	Корпоративний веб-портал поліграфії з деякими функціями web-to-print	26	350	45	0,4	31	2,2	26,6	13977,6	428	—
4	Загальний інформаційний веб-портал з можливостями web-to-print	4	620	24	11,2	14	2,5	382,8	3809,28	2241	Інтеграція з новинними ресурсами, форум
5	Універсальний веб-портал web-to-print	2	400	120	140	9	1,8	2214,8	1228,8	53	—

3.1.5 Розрахунок обсягу виробництва, трудомісткості робіт, необхідної кількості устаткування та робочих місць, кількості працюючих

Обчислення обсягу інформації для веб-порталів та інших інтернет-видань здійснюється у такий спосіб:

- 1) текстова інформація підраховується в кількості символів (знако-команд), яку далі можна перерахувати в об'єм у мегабайтах. Одна знако-команда в різних кодуваннях може займати різний об'єм (у байтах), але найменше – це 1 байт. Оскільки для веб-порталів та інших веб-ресурсів загальноприйнятим є кодування символів UTF-8, то один символ – це 1 байт для латиниці та 2 байти для кирилиці [66];
- 2) графічні матеріали (зображення) зберігають у файлі формату «jpg». Цей формат має властивість стиснення, яка умовно виражається числом від 0 до 100. При максимальному стисненні спотворюється зображення, саме тому зберігають файли зі стисненням 75-80. Кількість ілюстративного матеріалу виражають у мегапікселях (1000 на 1000 пікселів), а обсяг його при 75 стисненні в середньому становить 0,12 Мбайт [67-68].

Відповідно до вибраної технології створення веб-порталів, розгорнутих промислових завдань та норм на деякі процеси [69-70], можна провести виробничі розрахунки, а саме: розрахувати виробниче завантаження на обробку графічної інформації, обробку текстової інформації, розробку користувацьких інтерфейсів (UI/UX-дизайн), розробку front-end частини, розробку back-end частини, налаштування роботи системи безперервної інтеграції, наповнення контентом та тестування. Результати проведених розрахунків представлені в таблицях 3.16-3.23.

Таблиця 3.16 – Виробниче завантаження на обробку текстової інформації

№	Тип веб-порталу	Кількість назв на рік	Група складності	Заг.завд., тис. зн.	Одиниця обліку, зн	Норма часу на 1 обл. од., хв	Всього нормо-годин	Завдання в од. інформації, Мб
1	Веб-портал з підвищеним захистом	3	I	1694	1000	8,36	708.09	9.69
2	Державний веб-портал	11		235			360.18	4.93
3	Корпоративний веб-портал	26		428			1550.5	21.22
4	Інформаційний веб-портал	4		2241			1248.98	17.1
5	Універсальний веб-портал	2		53			14.77	0.2

Таблиця 3.17 – Виробниче завантаження на обробку графічної інформації

№	Тип веб-порталу	Кількість назв на рік	Середня кількість ілюстрацій	Одиниця обліку, мп	Норма часу на 1 обл. од., хв	Кількість облікових одиниць	Всього нормо-годин	Завдання в од. інформації, Мб
1	Веб-портал з підвищеним захистом	3	45	1	12	207.36	41.47	24.88
2	Державний веб-портал	11	25			422.4	84.48	50.69
3	Корпоративний веб-портал	26	350			13977.6	2795.52	1677.31
4	Інформаційний веб-портал	4	620			3809.28	761.86	457.11
5	Універсальний веб-портал	2	400			1228.8	245.76	147.46

Таблиця 3.18 – Виробниче завантаження на розробку користувацьких інтерфейсів (UI/UX-дизайн)

№	Тип веб-порталу	Кількість назв на рік	Одиниця обліку, стор.	Кількість облікових одиниць, стор.	Норма часу на 1 обл. од., хв	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1	38	85	161.5
2	Державний веб-портал	11		60	295	3245
3	Корпоративний веб-портал	26		45	160	3120
4	Інформаційний веб-портал	4		24	55	88
5	Універсальний веб-портал	2		120	115	460

Таблиця 3.19 – Виробниче навантаження на налаштування системи безперервної інтеграції

№	Тип веб-порталу	Кількість назв на рік	Одиниця обліку, налаштування	Кількість облікових одиниць, стор.	Норма часу на 1 обл. од., хв	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1	1	720	36
2	Державний веб-портал	11			1920	352
3	Корпоративний веб-портал	26			1200	520
4	Інформаційний веб-портал	4			960	64
5	Універсальний веб-портал	2			2880	96

Таблиця 3.20 – Виробниче навантаження на розробку front-end частини

№	Тип веб-порталу	Кількість назв на рік	Одиниця обліку, стор.	Кількість облікових одиниць, стор.	Норма часу на 1 обл. од., хв	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1	38	90	171
2	Державний веб-портал	11		60	115	1265
3	Корпоративний веб-портал	26		45	225	4387.5
4	Інформаційний веб-портал	4		24	80	128
5	Універсальний веб-портал	2		120	340	1360

Таблиця 3.21 – Виробниче навантаження на розробку back-end частини

№	Тип веб-порталу	Кількість назв на рік	Одиниця обліку, стор.	Кількість облікових одиниць, стор.	Норма часу на 1 обл. од., хв	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1	38	160	304
2	Державний веб-портал	11		60	460	5060
3	Корпоративний веб-портал	26		45	145	2827.5
4	Інформаційний веб-портал	4		24	65	104
5	Універсальний веб-портал	2		120	740	2960

Таблиця 3.22 – Виробниче навантаження на наповнення контентом

№	Тип веб-порталу	Кількість назв на рік	Заг.завд., тис. зн	Загальна кількість ілюстрацій	Одиниця обліку	Норма часу на 1 обл. од., хв	Кількість облікових одиниць	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1694	2150	1000 зн. або 5 ілюстр.	2.5	6372	265.5
2	Державний веб-портал	11	235.2	164			2948	122.83
3	Корпоративний веб-портал	26	428.4	142			11877	494.88
4	Інформаційний веб-портал	4	2241.7	1952			10529	438.71
5	Універсальний веб-портал	2	53.5	1350			647	26.96

Таблиця 3.23 – Виробниче навантаження на тестування

№	Тип веб-порталу	Кількість назв на рік	Одиниця обліку, стор.	Кількість облікових одиниць, стор.	Норма часу на 1 обл. од., хв	Всього нормо-годин
1	Веб-портал з підвищеним захистом	3	1	38	12	22.8
2	Державний веб-портал	11		60	35	385
3	Корпоративний веб-портал	26		45	6.5	126.75
4	Інформаційний веб-портал	4		24	6	9.6
5	Універсальний веб-портал	2		120	22	88

Опираючись на розрахунки виробничого навантаження, було обчислено потрібну кількість робочих місць основних виробничих операцій, устаткування, списковий та явочний штат робітників. Результати розрахунків [71] занесено до таблиць 3.24-3.25.

Фонд часу роботи на 2020 за 40-годинного робочого тижня становить 2002 години [72].

Таблиця 3.24 – Необхідна кількість устаткування та робочих місць

№	Повна назва робочого місця	Марка, фірма-виробник устаткування	Виробнича програма, нормо-годин	Необхідна кількість робочих місць, одиниць	
				Розрахункова	Прийнята проектом
1	2	3	4	5	6
1	УРС (універсальна робоча станція опрацювання текстової та графічної інформації)	Моноблок HP 200 G3 AiO з операційною системою Windows 10 (64 бітна)	7811.61	3.97	4
2	РСІ (робоча станція розробки користувацьких інтерфейсів)		7074.5	3.53	4
3	РСФ (робоча станція розробників front-end частини)		7311.5	3.65	4
4	РСБ (робоча станція розробників back-end частини)		11255.5	5.62	6
5	РСБІ (робоча станція для налаштування системи безперервної інтеграції)		1068	0.53	1
6	РСНТ (робоча станція наповнення контентом та тестування)		1981.03	0.99	1
				Всього	20

Таблиця 3.25 – Чисельність працюючих

№	Назва виробничої операції	Розрахункова кількість робочих місць, одиниць	Чисельність та розряд робітників	Явочна кількість робітників, осіб	Списочна кількість робітників, осіб	ІТР та службовців, осіб
1	2	3	4	5	6	7
1	Редагування, коригування та оформлення (обробка) текстової та графічної інформації	3.73	4 (5 розряд)	4	4	5

Кінець таблиці 3.25

№	Назва виробничої операції	Розрахунок ва кількість робочих місць, одиниць	Чисельність та розряд робітників	Явочна кількість робітників, осіб	Списочна кількість робітників, осіб	ІТР та службовців, осіб
1	2	3	4	5	6	7
2	Розробка користувацьких інтерфейсів (UI/UX-дизайн)	3.53	4 (6 розряд)	4	4	5
3	Розробка front-end частини веб-порталу (верстка, анімації та ін.)	3.65	4 (6 розряд)	4	4	5
4	Розробка back-end частини веб-порталу (робота з базою даних, обробка запитів, адміністративна панель тощо)	5.62	6 (6 розряд)	6	7	8
5	Налаштування системи безперервної інтеграції	0.53	1 (6 розряд)	1	1	2
6	Наповнення початковим (базовим) контентом (з адміністративної панелі) та тестування	0.99	1 (5 розряд)	1	1	2

3.1.6 Виробничо-технологічні плани виробничих приміщень

Було розроблено план центру з розроблення веб-порталів. Будівля буде одноповерховою. На одне робоче місце потрібно 4 м² у адміністративних приміщеннях та 6 м² у виробничих. Підсумувати площі усіх приміщень, можна дізнатися середню площу центру – 182 м². Однак рекомендовані відстані між колонами за стандартом 3, 6, 9, 12, 18. Тому для побудови сітки колон площу було округлено до 18, а саме:

$$S_{\text{сер}} \approx 216 \text{ (м}^2\text{)} = (6 + 6 + 6) \cdot (6 + 6)$$

За стандартом ДСТУ Б А.2.4.-7-95 (ГОСТ 21.501-93) висота виробничого приміщення – 3,3 м.; розміри колон – 0,4×0,4 м; ширина зовнішніх стін - 0,4 м; ширина внутрішніх стін – 0,3 м.

Експлікація центру представлено в таблиці 3.26.

Таблиця 3.26 – Експлікація центру

№	Найменування приміщення, призначення	Площа, м ²
I	Відділ підготовки текстової та графічної інформації	24
II	Відділ створення користувацьких інтерфейсів (UI/UX)	24
III	Відділ back-end розробників	36
IV	Відділ front-end розробників	25.5
V	Відділ безперервної інтеграції, наповнення та тестування	12
VI	Складське приміщення	4
VII	Приймальня і директор. Бухгалтерія	24
VIII	Вестибюль	8.5
IX	Серверна кімната	8
X	Туалет	12.75
XI	Буфет	12.75

План центру з розроблення веб-порталів наведений на рисунку 3.15, а в додатки винесено штамп і експлікацію обладнання.

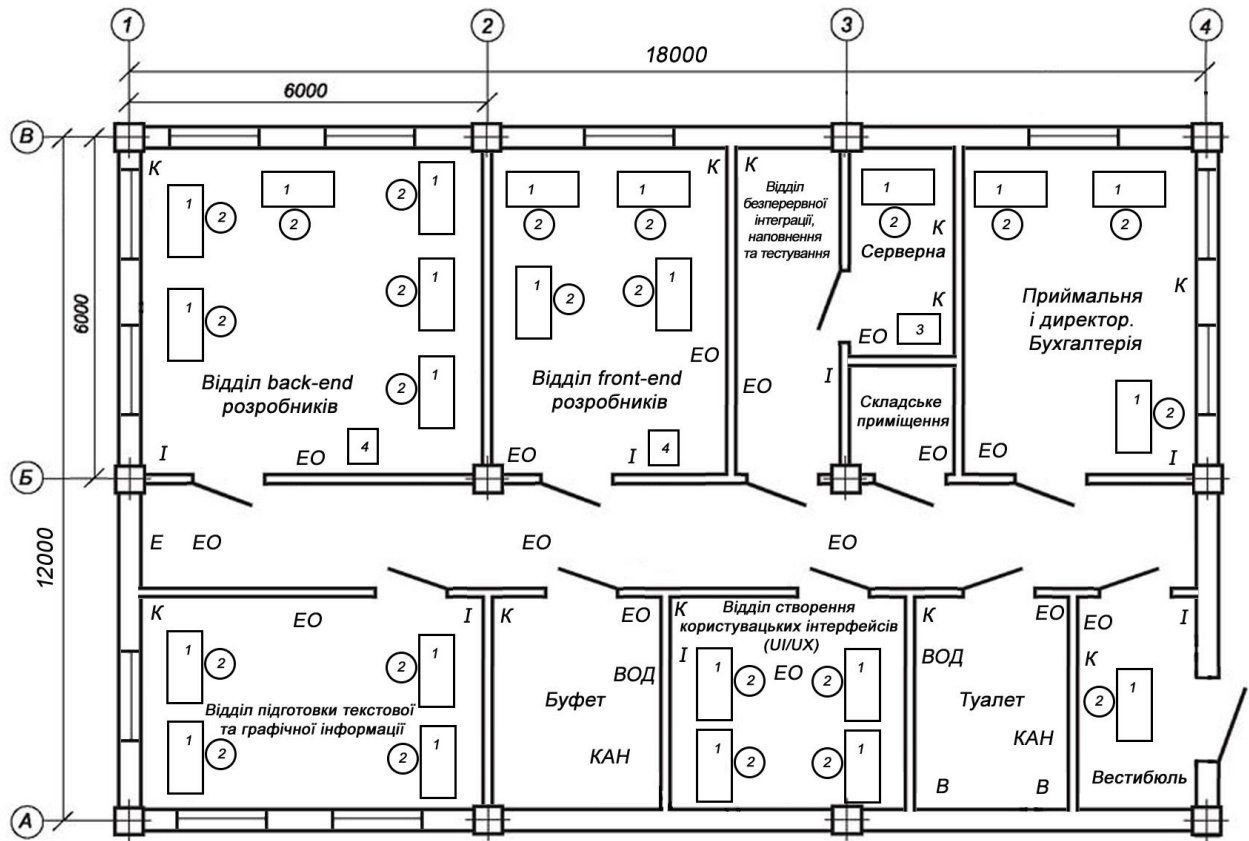


Рисунок 3.15 — План центру: В – витяжна вентиляційна система, ЕО – електричне освітлення, К – система кондиціонування, Е – електрощитова, ВОД – вода, КАН – загальна міська каналізаційна система, І – локальна мережа, 1 – робочий стіл, 2 – крісло, 3 – серверна шафа, 4 – шафа.

3.2 Завдання на інженерно-технічного забезпечення виробництва

3.2.1. Проектування конструкцій перекриття та шумоізоляції виробничих приміщень

Центр з розроблення веб-порталів знаходиться в одноповерховій будівлі (займає повністю). Всередині приміщення розташовані лише робочі станції та серверний шафа CMS 24U Type 2, яка має гарну звукоізоляцію.

Оскільки не розміщено ніякого великогабаритного обладнання, тому специфічних вимог щодо зведення конструкцій перекриття не ставиться. Отже, слід розрахувати вагу робочих станцій на 1 м^2 , тобто статичне навантаження. Шум та вібрація не перевищуватимуть норму.

Визначені розрахунки кількості устаткування всіх відділів зведено до загальної специфікації (таблиця 3.27), де наведено габарити оснащення робочих місць та їх параметри інженерного забезпечення, що є базою всіх ТЗ на розробку конструкцій перекриття.

Відповідно до норм загальної та локальної вібрації ДСН 3.3.6.039-99 припустимо максимальний рівень шуму 60 дБ, тому рівень вібрації буде становити 1,4 см/с.

Таблиця 3.27 – Специфікація (відомість) вихідних даних ТЗ на розробку конструкцій перекриття та шумоізоляції

№	Назва устаткування, оснащення робочого місця	Марка	Габарити, мм x мм	Необхідна площа для розміщення устаткування або оснащення робочого місця, м2	Маса устаткування, т
1	Робоча станція	HP 200 G3 AiO	1200 x 600	0,72	0,018
2	Серверна шафа	CMS 24U Type 2	800 x 500	0,4	0,035

Кінець таблиці 3.27

№	Статистичне навантаження, т/м2	Зусилля (тиснення, вирубки), т	Динамічне навантаження, Н	Вібрація, см/с	Максимальний рівень шуму, дБ
1	0,025	Відсутнє	Відсутнє	1,4	60
2	0,088				

3.2.2 Розроблення ескізних креслень і 3Д-моделі генерального плану центру з розроблення веб-порталів

Для розробки генерального плану за формулою $K_3 = S_{\text{заб}} / S_{\text{підп}}$ знайдено щільність забудови, підставивши дані: $K_3 = (216 + 5.5 * 6) / 432 = 0.57$.

З огляду на те, що на території підприємства не передбачено доріг, складів та інших споруд, крім основної, то коефіцієнт використання ділянки розраховується як $S_{\text{буд}}/S_{\text{підпр.}}$, тобто $216/432=0.5$.

Коефіцієнт об'єму будівель визначається так: $K=(V/S)$. Підставивши дані у формулу, було отримано: $K=(3.5*216)/432=1.75$.

Коефіцієнт щільності забудови включає кількість поверхів. Він буде дорівнювати коефіцієнту використання ділянки, тобто 50% або 0.5, оскільки будівля одноповерхова.

Тому, загальна площа підприємства становитиме 432м^2 , 216м^2 площі – одноповерхова будівля. Дороги, склади та інші споруди на ділянці відсутні. Оскільки персонал працює в одну зміну і його кількість дорівнює 25 – повинно бути облаштовано 2 паркомісця (відповідно до норм 7-10%), їх і було облаштовано.

Генеральний план центру наведений на рисунку 3.16.

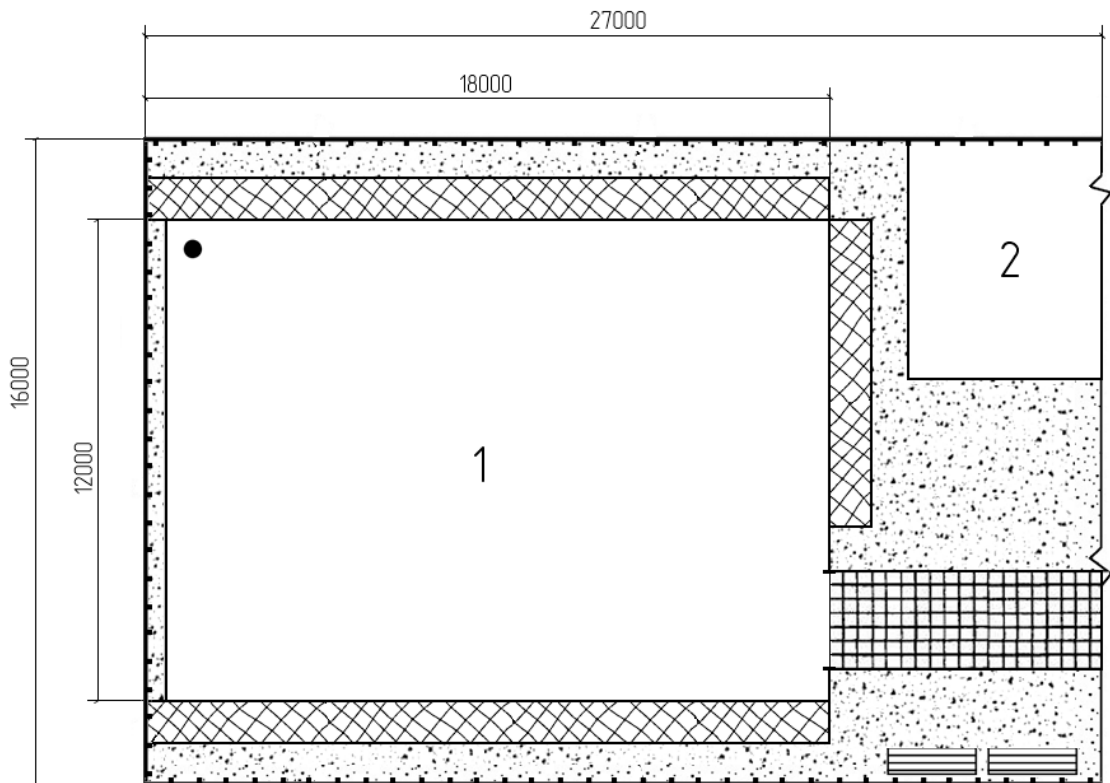


Рисунок 3.16 — Генеральний план центру: 1 – центр розробки веб-порталів, 2 – відкрита парковка

Крім того, було запроєктовано об'ємно-планувальну схему підприємства і виконано моделювання ландшафту, після чого зроблено 3D-рендеринг, результат якого зображено на рисунку 3.17.



Рисунок 3.17 – Візуалізація підприємства з елементами ландшафту

3.2.3. Складання завдання на інженерно-технічне забезпечення виробництва

Завдання на інженерно-технічне забезпечення виробничих процесів наведено в таблиці 3.28.

Таблиця 3.28 – Експлікація обладнання

№	Найменування	Кількість, шт	Марка та фірма виробник устаткування	Габарити, м
1	Робочий стіл	25	Стиль SL-82, AMF	1,2 x 0,6
2	Крісло	25	Neon HR, AMF	0,65 x 0,64
3	УРС (універсальна робоча станція опрацювання текстової та графічної інформації)	4	Моноблок HP 200 G3 AiO з операційною системою Windows 10 (64 бітна)	0,49 x 0,2
4	РСІ (робоча станція розробки користувацьких інтерфейсів)	4		

Продовження таблиці 3.28

№	Найменування	Кількість	Марка та фірма виробник устаткування	Габарити, м
5	РСФ (робоча станція розробників front-end частини)	4	Моноблок HP 200 G3 AiO з операційною системою Windows 10 (64 бітна)	0,49 x 0,2
6	РСБ (робоча станція розробників back-end частини)	6		
7	РСБІ (робоча станція для налаштування системи безперервної інтеграції)	1		
8	РСНТ (робоча станція наповнення контентом та тестування)	1		
9	РСК (робоча станція керівника або помічника)	2		
10	РСБ (робоча станція бухгалтера)	1		
11	РСР (робоча станція рецепції)	1		
12	РССА (робоча станція системного адміністратора)	1		
13	Серверна шафа	1	24U Type 2, CMS	0,8 x 0,5
14	Шафа	2	ОШ-2, Ferocon	0,5 x 0,6

Кінець таблиці 3.28

Потреба в технічному забезпеченні									
Електроенергія			Вода		Каналізація	Вентиляція		Зв'язок	Комп'ютеризація
Силова	Теплова	Освітлення	Холодна	Гаряча		Загальна	Місцева		
35782 кВт*год	14300 кВт*год	6616 кВт*год	672,7 л	34622 м ³	Загальноміська, 96315 м ³	Канальний кондиціонер, 17029440 м ³	Спліт-система кондиціонування	Ір-телефонія, мережа інтернет 1 Гбіт/с	ЛОМ, 10 Гбіт/с

3.2.4. Завдання на комп'ютерне забезпечення виробництва

Було розроблено завдання на комп'ютерне забезпечення технологічних та виробничих процесів, яке представлено у таблиці 3.29.

Таблиця 3.29 – Завдання на комп'ютерне забезпечення технологічних та виробничих процесів

№	Назва устаткування чи робочого місця	Рекомендоване комп'ютерне устаткування	Необхідне програмне забезпечення	Рекомендована потужність комп'ютера, МБайт	Операції та засоби контролю, що підлягають комп'ютеризації
1	УРС (універсальна робоча станція опрацювання текстової та графічної інформації)	Моноблок HP 200 G3 AiO	ОС: Windows 10 Pro; ПЗ: Google Chrome, пакет Microsoft Office 2020, Adobe Photoshop CC 2020, Sublime Text 3.2, Bitdefender Free	ОЗП: 4096 ПЗП: 46438	Калібрування, профілювання моніторів, антивірусна перевірка робочих станцій, створення резервних копій робочої інформації
2	РСІ (робоча станція розробки користувацьких інтерфейсів)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Adobe XD 27, Bitdefender Free	ОЗП: 8192 ПЗП: 31216	
3	РСФ (робоча станція розробників front-end частини)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Adobe XD 27, PhpStorm 2019.3, Docker for Windows 2.1, Bitdefender Free	ОЗП: 8192 ПЗП: 66386	
4	РСБ (робоча станція розробників back-end частини)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Mozilla Firefox, PhpStorm 2019.3, , Docker for Windows 2.1, Bitdefender Free		
5	РСБІ (робоча станція для налаштування системи безперервної інтеграції)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Docker for Windows 2.1, Sublime Text 3.2, Putty 0.7, Bitdefender Free	ОЗП: 4096 ПЗП: 32240	
6	РСНТ (робоча станція наповнення контентом та тестування)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Bitdefender Free	ОЗП: 4096 ПЗП: 16612	
7	РСК (робоча станція керівника або помічника)		ОС: Windows 10 Pro; ПЗ: Google Chrome, пакет Microsoft Office 2020, Bitdefender Free	ОЗП: 4096 ПЗП: 65230	

Кінець таблиці 3.29

№	Назва устаткування чи робочого місця	Рекомендоване комп'ютерне устаткування	Необхідне програмне забезпечення	Рекомендована потужність комп'ютера, МБайт	Операції та засоби контролю, що підлягають комп'ютеризації
8	РСБХ (робоча станція бухгалтера)	Моноблок HP 200 G3 AiO	ОС: Windows 10 Pro; ПЗ: Google Chrome, 1С: Бухгалтерія для України, пакет Microsoft Office 2020, Bitdefender Free	ОЗП: 4096 ПЗП: 26765	Калібрування, профілювання моніторів, антивірусна перевірка робочих станцій, створення резервних копій робочої інформації
9	РСП (робоча станція рецепції)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Bitdefender Free	ОЗП: 4096 ПЗП: 16612	
10	РССА (робоча станція системного адміністратора)		ОС: Windows 10 Pro; ПЗ: Google Chrome, Putty 0.7, FileZilla 3.4, WinSCP 5.17, Bitdefender Free	ОЗП: 8192 ПЗП: 63149	

Визначення побудови ЛОМ – це завершаюча стадія проектування системи. Через те, що вся робота, за винятком безперервної інтеграції, проходить за допомогою хмарних сервісів без локального серверу, тому вибрано однорангову ЛОМ. Виходячи із розробленого завдання на комп'ютерне забезпечення технологічних та виробничих процесів, було розроблено типову структуру КВС, схему якої наведено нижче на рисунку 3.18.

Для роботи ЛОМ обрано комутатор TP-LINK TL-SG105, характеристики якого наведено у таблиці 3.30.

Таблиця 3.30 – Характеристики TP-LINK TL-SG105 [73]

Характеристика	Значення
Порти, шт	5
Швидкість портів, мб	10/100/1000
Габарити, мм	99.8 x 98 x 25

В робочих станціях мережевий адаптер вмонтовано в материнську плату. Таким чином, як мережевий адаптер використовується контролер Realtek RTL8111H Gigabit Ethernet 10/100/1 000 Мбіт/с (RJ-45) [60].

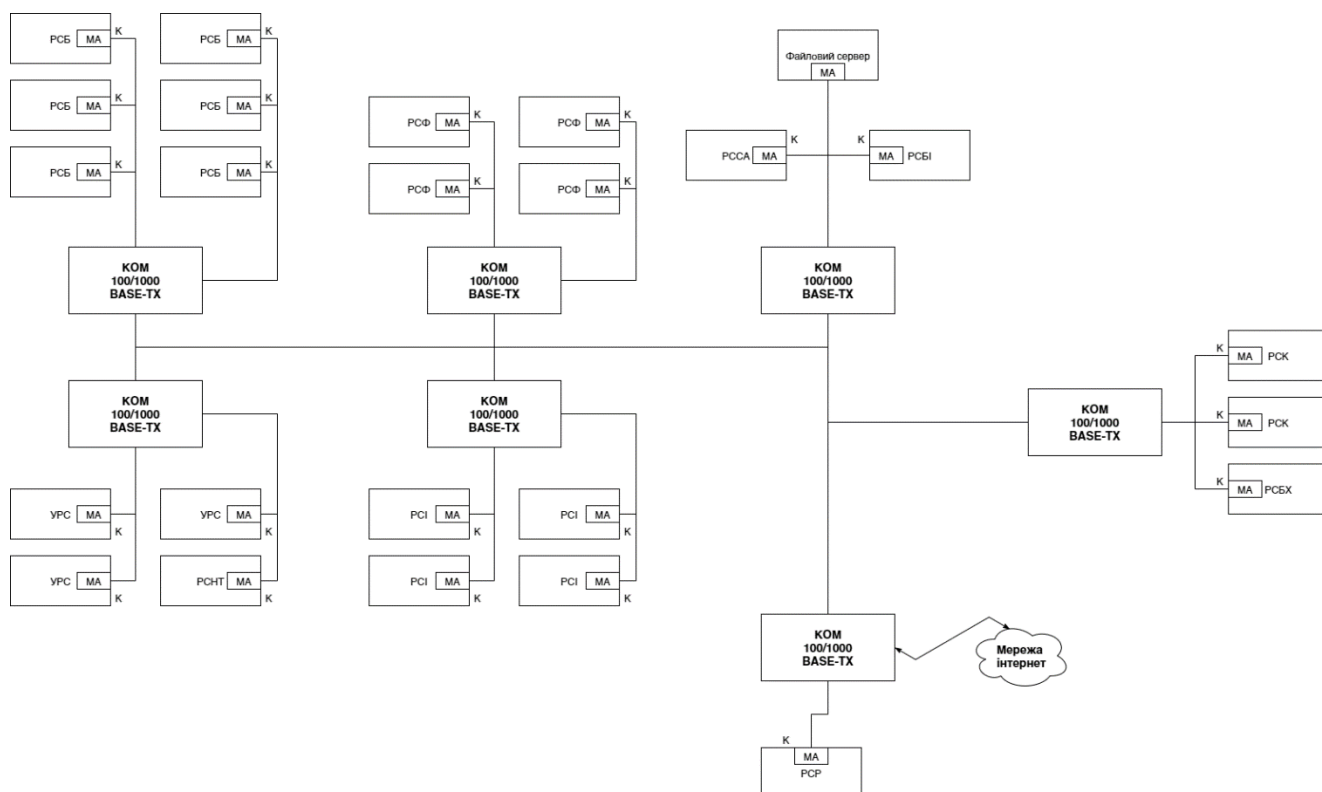


Рисунок 3.18 – Узагальнена структурна електрична схема комп'ютерної мережі:

УРС – робоча станція опрацювання текстової та графічної інформації;

РСІ – робоча станція розробки користувацьких інтерфейсів;

РСФ – робоча станція розробників front-end частини;

РСБ – робоча станція розробників back-end частини;

РСБІ – робоча станція для налаштування системи безперервної інтеграції;

РСНТ – робоча станція наповнення контентом та тестування;

РСК – робоча станція керівника або помічника;

РСБХ – робоча станція бухгалтера;

РСР – робоча станція рецепції;

РССА – робоча станція системного адміністратора;

КОМ – комутатор TP-LINK TL-SG105;

МА – мережевий адаптер Realtek RTL8111H Gigabit Ethernet;

К – кабель на скручених парах.

3.3 Техніко-економічні показники проєкту

Для перевірки доцільності впровадження даного проєкту – центру з розроблення веб-порталів, було розраховано техніко-економічні показники.

Абсолютні показники: річний випуск веб-порталів, загальна кількість працівників, кількість промислово-виробничого персоналу, загальна площа земельної ділянки, загальна площа центру, загальна кубатура центру, висота поверху центру, вартість впровадження проєкту центру.

Відносні показники: витрати електроенергії на освітлення, витрати електроенергії для технологічних потреб, трудомісткість, витрати на водопостачання, повна собівартість, ціна веб-порталу, прибуток, рентабельність, термін окупності.

Витрати електроенергії для створення веб-порталу розраховуються зважаючи на потужність обладнання та вартість електроенергії, яка в другому кварталі 2020 року коштує 1.57 грн/кВт·год за 1 клас напруги у компанії ПрАТ «ДТЕК Київські електромережі» [74]. Річні витрати на електроенергію для технологічних потреб було розраховано за формулою 3.10 [75]:

$$B_e = P \times T_o \times K_e \times C \times K_{\text{прац}}, \quad (3.10)$$

де P – потужність струмоприймачів, кВт;

T_o – час роботи устаткування, години;

C – ціна за 1 кіловат-годину електроенергії, грн;

$K_{\text{прац}}$ – кількість працівників центру;

K_e – коефіцієнт, що враховує втрати в електродвигуні та електромережі, прийнятий 1,1.

Результати розрахунку за формулою (3.10):

$$B_e = 0,065 \times 2002 \times 1,57 \times 1,1 \times 25 = 5617.5 \text{ грн}$$

Річні витрати електроенергії на освітлення приміщень центру було розраховано за наступною формулою 3.11 [75]:

$$W_{\text{річ.осв.}} = \frac{S_{\text{заг}} \times \omega \times K_{\text{осв}} \times T_{\text{осв}}}{1000} \times \text{Ц} \quad (3.11)$$

де $W_{\text{річ.осв.}}$ – річна кількість електроенергії на освітлення, кВт;

$S_{\text{заг.}}$ – загальна площа будівлі (виробничі, побутові, адміністративні приміщення, склади), м^2 ;

ω – питомі витрати електроенергії, Вт/м^2 , приймаються у межах 18-23 Вт/м^2 ;

$K_{\text{осв.}}$ – коефіцієнт, що враховує одночасність освітлення всіма світильниками, приймається у межах 0,8...0,9;

$T_{\text{осв.}}$ – час освітлення залежно від режиму роботи підприємства, год.;

1000 – коефіцієнт переведення одиниць Вт у кіловати (кВт);

Ц – ціна за 1 кіловат-годину електроенергії, грн.

Розрахуємо витрати електроенергії на освітлення:

$$W_{\text{річ.осв.}} = \frac{216 \times 18 \times 0,85 \times 2002}{1000} \times 1,57 = 10387,14 \text{ грн}$$

Для розрахунку витрат води було використано ціну за 1 м^3 води на 2020 рік – 22,99 грн від компанії «Київводоканал» [76]. Витрати розраховувалися за наступною формулою 3.12 [75]:

$$B_v = \frac{q_{\text{пит}} \times R \times T \times \text{Ц}}{1000} \quad (3.12)$$

де B_v – кількість води на господарсько-питні потреби працівників, грн;

$q_{\text{пит.}}$ – питомі витрати води на одного працівника, л/добу, приймається 12 л/добу [77];

R – розрахована спискова (прийнята) чисельність працюючих на підприємстві, чол;

T – річний фонд часу, год;

Π – ціна 1 м³ води, грн/м³.

$$B_v = \frac{12 \times 28 \times 2002 \times 22,99}{1000} = 15464,73 \text{ грн}$$

Прибуток було визначено за формулою 3.13 [75]:

$$\Pi = \Pi_n - C \quad (3.13)$$

де Π – прибуток;

Π_n – ціна мультимедійного видання;

C – повна собівартість.

Ціна за виготовлення продукції розраховується на базі повної собівартості продукції, до якої додається 30% [75]. Для визначення повної собівартості потрібно додати витрати на виготовлення мультимедійного видання, а також загальногосподарські, позавиробничі та загальновиробничі витрати.

Сума загальновиробничих витрат розраховується як 160% від базової заробітної плати виробничих робітників, а загальногосподарських – як 180%. Позавиробничі витрати являють собою 0,7% виробничої собівартості [75].

У таблиці 3.24 подано всіх робітників підприємства, яких можна віднести до виробничих робітників, загалом – 22 працівники. Середня зарплатня становить близько 34000 грн згідно сайту пошуку вакансій [78]. За рік центр створює 46 веб-порталів, а річний зарплатний фонд становить 8976 тис. грн, ЄСВ (22%) – 1974,72 тис. грн. Зарплатні витрати на створення веб-порталу дорівнюють 195,13 тис. грн, ЄСВ – 42,91 тис. грн.

Загальновиробничі витрати: $195,13 \times 1,6 = 312,2$ тис. грн.

Загальногосподарські витрати: $195,13 \times 1,8 = 351,23$ тис. грн.

Виробнича собівартість: $195,13 + 42,91 + 312,2 + 351,23 = 380,9$ тис. грн.

Позавиробничі витрати: $380,9 \times 0.007 = 2,67$ тис. грн.

Повна собівартість продукції: $C = 380,9 + 2,67 = 383,57$ тис. грн.

Ціна за виготовлення продукції: $C_n = 383,57 \times 1.3 = 498,64$ тис. грн.

Прибуток з одиниці продукції: $\Pi = 498,64 - 383,57 = 115,07$ тис. грн.

Рентабельність продукції було розраховано за формулою 3.14 [75]:

$$P_{\Pi} = \frac{\Pi \times 100}{C}, \quad (3.14)$$

де P_{Π} – рентабельність;

Π – прибуток;

C – повна собівартість.

$$P_{\Pi} = \frac{115,07 \times 100}{383,57} \approx 30\%$$

Результати розрахунків наведено в таблицях 3.31 – 3.32.

Таблиця 3.31 — Абсолютні техніко-економічні показники проєкту

№	Показник	Одиниця виміру	Значення
1	Річний випуск веб-порталів	шт.	46
2	Чисельність промислово-виробничого персоналу	ос.	22
3	Загальна чисельність працівників		25
4	Загальна площа центру	м ²	216
5	Висота поверху центру	м	3,3
6	Загальна кубатура центру	м ³	712,8
7	Загальна площа земельної ділянки	м ²	432
8	Коефіцієнт щільності забудови	—	0.5
9	Вартість впровадження проєкту	тис. грн	7950

Таблиця 3.32 – Відносні техніко-економічні показники

№	Техніко-економічні показники проєкту	Одиниця виміру	Розраховане значення
1.	Кількість продукції: на одиницю площі приміщення на одиницю об'єму приміщення на одиницю площі земельної ділянки	шт.	0,21 0,065 0,105
2.	Витрати електроенергії для технологічних потреб: на створення 1 веб-порталу на річний випуск веб-порталів на 1000 веб-порталів	грн	122,1 5617,5 122100
3.	Витрати електроенергії на освітлення	грн	10387,14
4.	Витрати на водопостачання: на створення 1 веб-порталу на річний випуск веб-порталів на 1000 веб-порталів	грн	336,2 15464,73 336200
5.	Трудомісткість: на створення 1 веб-порталу на річний випуск	нормо-год	786,2 36167,97
6.	Повна собівартість 1 веб-порталу	тис. грн	383,57
7.	Ціна веб-порталу	тис. грн	498,64
8.	Соціальна програма	—	На території даного підприємства розміщено автостоянку, буфет
9.	Прибуток: зі всіх розроблених веб-порталів на одиницю продукції на одиницю площі приміщення на одиницю об'єму приміщення на одиницю площі земельної ділянки	тис. грн	5293,22 115,07 24,5 7,43 12,25
10.	Рентабельність	%	30
11.	Термін окупності	рік	1,5

Крім того, було визначено та оцінено ризики, що можуть вплинути на розробку проєкту і діяльність центру надалі. Результати представлено в таблиці 3.33.

Таблиця 3.33 – Оцінювання ризиків

№	Найменування потенційного ризику	Думка експерта			Середнє арифметичне	Зважений результат
		№1	№2	№3		
1	Невчасне виконання проекту	10	8	8	8.67	11.26
2	Перевищення бюджету проекту	9	9	10	9.33	12.11
3	Проблеми з підрядниками	5	6	4	5	6.49
4	Проблеми з підключенням електроенергії чи її постачанням	7	7	6	6.67	8.66
5	Проблеми з отриманням дозволів чи інші проблеми з держ. органами	4	3	7	4.67	6.06
6	Стихійне лихо, пандемія чи інша форс-мажорна ситуація	10	9	10	9.67	12.56
7	Недостатня завантаженість замовленнями	10	10	10	10	12.98
8	Неякісний контроль етапів виконання проекту	7	8	8	7.67	9.96
9	Проблеми з пошуком спеціалістів або найм недостатньо досвідчених	9	10	10	9.67	12.56
10	Заморозки переказів SWIFT чи інші проблеми з банками	6	4	7	5.67	7.36

Було проведено ранжування ризиків від найбільшого до найменшого за зваженим результатом. Результат ранжування наведено в таблиці 3.34

Таблиця 3.34 – Ранжування ризиків

№	Найменування потенційного ризику	Зважений результат	Ступінь впливу
1	Недостатня завантаженість замовленнями	12.98	Високий
2	Стихійне лихо, пандемія чи інша форс-мажорна ситуація	12.56	Високий
3	Проблеми з пошуком спеціалістів або найм недостатньо досвідчених	12.56	Високий
4	Перевищення бюджету проекту	12.11	Високий
5	Невчасне виконання проекту	11.26	Середній
6	Неякісний контроль етапів виконання проекту	9.96	Середній
7	Проблеми з підключенням електроенергії чи її постачанням	8.66	Середній
8	Заморозки переказів SWIFT чи інші проблеми з банками	7.36	Низький
9	Проблеми з підрядниками	6.49	Низький
10	Проблеми з отриманням дозволів чи інші проблеми з держ. органами	6.06	Низький

Провівши аналіз таблиці 3.34 можна зробити висновок, що існує 4 головних ризики, а саме: недостатня завантаженість замовленнями; форс-мажорні обставини: стихійні лиха, пандемія чи інші; перевищення бюджету проєкту; труднощі з пошуком професійних працівників або набір без достатнього досвіду роботи. Аби мінімізувати шкідливу дію такого ризику як недостатня завантаженість замовленнями – потрібно компетентно скласти промислове завдання, коректно налагодити рекламу в соціальних мережах чи пошукових системах. Повністю уникнути форс-мажорних обставин неможливо, але реально попередньо підготуватися, наприклад, до віддаленої роботи з дому. Наступні фактори несуть найменші ризики: заморозки переказів банками, труднощі з підрядниками та труднощі з отриманням дозволів, а також інші труднощі у взаємовідносинах з державними органами.

3.4 Принципові рішення щодо розроблення технологічної системи

Було запроєктовано та розраховано промислове завдання для центру з розроблення веб-порталів, яке складається з таких позицій: веб-портал типу web-to-print з підвищеним захистом даних користувачів, веб-портал для використання web-to-print державними структурами, корпоративний веб-портал поліграфії з деякими функціями web-to-print, загальний інформаційний веб-портал з можливостями web-to-print, універсальний веб-портал web-to-print. Описано виробничо-технічні характеристики, розроблено структури модулів для кожного типу веб-порталу й загальну схему роботи панелі для адміністраторів.

За допомогою методу абстрактного моделювання функціональної системи «чорна скринька» виконано порівняння трьох технологій (фреймворків) для створення back-end частини, які мають такі основні характеристики: остання версія, тип ліцензії, можливості локалізації, тип доступу до бази даних, наявність логування, тип системи шаблонів, наявність XSS-захисту «з коробки», чи присутній модуль авторизації, а також зручність і продуманість архітектури.

За допомогою циклограми визначено технологічний процес створення front-end частини, а саме використовуючи елементи Bootstrap та плагіни JQuery.

Процес складається з таких етапів: налаштування середовища, інсталяція залежностей, підготовка необхідних компонентів та їх стилізація, верстка основних сторінок, верстка службових сторінок, верстка внутрішніх сторінок, налаштування плагінів та написання onload-коду, тестування.

На основі обраних технологій створення back-end та front-end частин складено поопераційні карти найбільш трудомістких етапів виробничого процесу: створення користувацьких інтерфейсів (UI/UX-дизайн), налаштування безперервної інтеграції, розробка front-end частини, створення адміністративної панелі.

Обрано сучасне програмне забезпечення для реалізації поставлених задач: Windows 10 Pro, Google Chrome 80, Mozilla Firefox 67, Microsoft Edge 17, Slack 3.4, Adobe Illustrator CC 2020, Adobe Photoshop CC 2020, PhpStorm 2020.3, Docker for Windows, GitHub Desktop 1.6.6, Антивірус Bitdefender free 2020. На основі обраного ПЗ та його вимог складено загальні системні вимоги до робочої станції. За допомогою аналізу з використанням пелюсткових діаграм обрано моноблок HP 200 G3 AiO.

Створено раціональну організаційну структуру виробництва, яка складається з чотирьох основних відділів. Кожним із відділів керує директор. Для підтримки роботи є також бухгалтер та системний адміністратор, а відділ роботи з клієнтами працює з новими клієнтами. Виробничим персоналом на підприємства є: копірайтери, спеціалісти з підготовки контенту, UI/UX-дизайнери, back-end розробники, front-end розробники, тестувальники, DevOps-спеціалісти.

Створено узагальнену блок-схему технологічного процесу створення веб-порталу згідно зі структурою підприємства і виробничою програмою. Основними етапами є: визначення цілей та розробка вимог; створення технічного завдання; визначення дедлайнів та підписання угод; ініціалізація проекту та налаштування системи безперервної інтеграції; проєктування інтерфейсів та розробка дизайну; програмування front-end та back-end;

тестування, виправлення помилок та налаштування інтеграцій; розгортання в робочому режимі; наповнення контентом та здача проєкту.

Проведено розрахунок виробничого навантаження на: обробку текстової інформації, обробку графічної інформації, розробку користувацьких інтерфейсів (UI/UX-дизайн), налаштування системи безперервної інтеграції, розробку front-end частини, розробку back-end частини, наповнення контентом та тестування. Також визначено кількість робочих місць та устаткування.

Запроектований центр розміщується в одноповерховій будівлі, загальним розміром 216 м² з сіткою колон 6000×6000 мм. Включає наступний поділ: відділ back-end розробки, відділ front-end розробки, відділ безперервної інтеграції та тестування, відділ підготовки текстової та графічної інформації, відділ створення користувацьких інтерфейсів, серверна, складське приміщення, буфет, туалет, вестибюль та кабінет директора. На основі цих даних розроблено технологічний та генеральний план центру з розроблення веб-порталів, а також 3Д-модель.

Наведено габарити облаштування робочих місць та параметри їх інженерного забезпечення. Специфічних вимог щодо зведення конструкцій перекриття не ставиться, тому що в центрі великогабаритне обладнання не розміщуватиметься.

Відповідно до проєкту розроблено завдання щодо інженерно-технічного та комп'ютерного забезпечення. Визначено потребу в технічному забезпеченні.

Створено узагальнену структурну електричну схему комп'ютерної мережі.

Розраховано відносні та абсолютні техніко-економічні показники, зокрема: рентабельність, термін окупності, прибуток на одиницю продукції, ціну веб-порталу, соціальну програму та ін.

Також виконано оцінювання ризиків, їх сортування від найбільшого до найменшого. Враховуючи ранжування ризиків – можна попередити найбільш небезпечні для реалізації проєкту: недостатня завантаженість замовленнями; форс-мажорні обставини: стихійні лиха, пандемія чи інші; перевищення бюджету проєкту; труднощі з пошуком професійних працівників або набір без достатнього досвіду роботи.

Висновки до третього розділу

В даному розділі створено проєкт сучасного підприємства, що займається розробкою веб-порталів з використанням систем безперервної інтеграції, архітектура яких створена за допомогою досліджень та аналізу великої кількості теоретичної інформації.

Обрані технології створення back-end та front-end частин, . що обрано за допомогою системи «чорна скринька» та циклограми відповідно, є раціональними та дозволяють покращити кінцеву якість продукту.

Програмне забезпечення дозволить виконувати всі необхідні операції, що пов'язані з діяльністю підприємства. На основі системних вимог кожної із програм складено загальні вимоги до апаратного забезпечення та зроблено вибір за допомогою порівняння із використанням пелюсткових діаграм.

Побудована організаційна структура та загальний технологічний процес дозволить створювати якісні веб-портали за розумний час, вартість, з високим рівнем автоматизації процесів, швидким виправленням програмних помилок, а за допомогою відділу по роботі з контентом й швидко просувати у пошукових системах (SEO-оптимізація).

Проведено розрахунки трудомісткості робіт, обсягу виробництва, необхідної кількості робочих місць (загалом 20) та устаткування, кількості працюючих.

Створено технологічний план центру з розроблення веб-порталів з наведенням експлікації обладнання. Наведено специфікацію вихідних даних ТЗ на розробку конструкцій перекриття та шумоізоляції. Також створено генеральний план і відповідний 3Д-рендер.

Розроблено узагальнену структурну електричну схему комп'ютерної мережі.

Розраховано відносні та абсолютні техніко-економічні показники, зокрема, прибуток, рентабельність та термін окупності.

Окрім всього, проведено оцінку можливих ризиків та ранжування задля попередження найбільш суттєвих.

РОЗДІЛ 4

РОЗРОБЛЕННЯ СТАРТ-АП ПРОЄКТУ

Проектується підприємство «Центр з розроблення веб-порталів із використанням систем безперервної інтеграції для реалізації web-to-print». Діяльність полягає у розробці та підтримці веб-порталів для підприємств поліграфічної галузі, що хочуть використовувати сучасні способи комунікації зі своїми клієнтами, широко рекламуватися в мережі інтернет, отримувати замовлення цілодобово та витратити мінімум часу своїх менеджерів.

Створенням веб-порталів займається чимало веб-студій, діджитал-агенств, але майже не має таких, що спеціалізуються на web-to-print та одночасно мають високий рівень експертизи в області систем безперервної інтеграції. Проведені дослідження архітектури систем безперервної інтеграції для реалізації web-to-print дозволяють отримати надзвичайно високий рівень якості кінцевого продукту (веб-порталу): високий рівень захищеності, легке масштабування при збільшенні навантаження (scalability), автоматичне тестування якості та працездатності при кожному оновленні перед розгортанням на робочих серверах та ін. Отримані напрацювання також зменшують строки розробки. Саме все це і вирізняє запроєктоване підприємство серед конкурентів.

4.1 Опис ідеї проєкту

У межах даного підпункту було описано:

- зміст ідеї;
- можливі напрямки застосування та вигоди для клієнтів;
- різницю за техніко-економічними характеристиками від пропозицій конкуруючих компаній.

Перші два пункти подані у вигляді таблиці 4.1 і дають повне уявлення про зміст ідеї та потенційні ринки, в межах яких необхідно знаходити потенційних клієнтів.

Таблиця 4.1 – Опис ідеї стартап проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка веб-порталів з використанням створеної для web-to-print архітектури системи безперервної інтеграції. Дана архітектура системи дозволяє: виконувати автоматичну збірку проєкту та його тестування, що дозволяє підтримувати високий рівень захищеності, якості та зменшити строки розробки; справлятися з підвищенням навантаження шляхом автоматичного керування хмарними серверними ресурсами; спростити процес оновлення та мінімізувати час недоступності	1. Діджиталізація поліграфічних підприємств для отримання замовлень та роботи з клієнтами за допомогою мережі інтернет	1. Новий канал клієнтів з широкими можливостями для реклами 2. Цілодобове отримання замовлень, можливість відслідковування клієнтами стану без зайвих проблем 3. Мінімізація часових витрат менеджерів компанії 4. Автоматична перевірка файлів клієнтів за основними вимогами
	2. Покращення існуючого веб-порталу за рахунок впровадження системи безперервної інтеграції	1. Значне зменшення кількості критичних помилок в роботі веб-ресурсу 2. Масштабування для компенсування пікових навантажень 3. Мінімізація часу простоїв та підвищення швидкості роботи сайту для замовників поліграфічної продукції

Порівняльний аналіз показників запропонованої ідеї та компаній-конкурентів також наведено за допомогою таблиці (таблиця 4.2), де кожному показнику проставлено W, S чи N, що означає гірше, нейтральне чи краще значення відповідно, у порівнянні з фірмами-конкурентами.

Сильними сторонами проєкту є надійність роботи, можливість масштабування для витримування пікових чи високих навантажень, рівень автоматизації операцій для економії часу менеджерів по роботі з клієнтами та наявність системи безперервної інтеграції з продуманою архітектурою, що дозволяє мати високу якість коду. Однак, система потребує більше часу на обслуговування ніж конкуренти.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№ п/п	Техніко- економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	W2P Shop	Key2Print			
1	Вартість типового веб-порталу	Середня	Висока	Середня		+	
2	Витрати часу на обслуговування	Вище середніх	Середні	Низькі	+		
3	Надійність роботи	Дуже висока	Середня	Низька			+
4	Захищеність даних замовників поліграфічної продукції	Висока	Висока	Середня		+	
5	Масштабування для витримування пікових чи високих навантажень	Авто	Відсутнє	Відсутнє			+
6	Рівень автоматизації операції (для економії часу менеджерів)	Дуже високий	Середній	Високий			+
7	Наявність системи безперервної інтеграції та продуманість її архітектури	Так, проводилися дослідження	Так, максимальна проста структура	Ні			+

4.2 Технологічний аудит ідеї проєкту

Реалізувати веб-портал можна за допомогою різних технологій, так само як і основу, на якій будується система безперервної інтеграції. Результати аудиту технологій занесено у таблицю 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка веб-порталів з використанням створеної для web-to-print архітектури системи безперервної інтеграції	Мова програмування PHP та власна структура	Наявна, але базову структуру потрібно розробити	Доступна
2		Мова програмування PHP та фреймворк Laravel	Наявна	Доступна
3		Мова програмування JavaScript (на базі Node.js)	Наявна	Доступна
4		Мова програмування C# та .NET Framework	Наявна, але втрачає популярність	Доступна
5		GitHub Actions	Наявна	Доступна
6		Jenkins	Наявна	Доступна
7		Circle CI	Наявна	Доступна
8		TeamCity	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: мова програмування PHP та фреймворк Laravel, із використанням GitHub Actions у якості основи для системи безперервної інтеграції				

4.3 Аналіз ринкових можливостей запуску стартап проєкту

Спочатку було проведено аналіз попиту, результати якого занесено до таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	5 500 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	22

Середню норму рентабельності було порівняно із банківським відсотком (6-12%) на вкладення та виявлено, що вкладати гроші в проєкт економічно вигідно.

Проаналізувавши таблицю 4.4 можна дійти висновку, що ринок є привабливим для входження.

Наступним кроком визначено потенційні групи клієнтів (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів старт-ап проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Цілодобове отримання замовлень через мережу інтернет, залучення клієнтів новими шляхами	Середні компанії поліграфічної галузі	Мінімізація часових витрат менеджерів, порівняно невисока ціна	Репутація компанії-розробника, простота використання
2	Веб-портал з високим рівнем автоматизації операцій та автоматичною перевіркою коду для зменшення безпекових помилок, компенсація пікових навантажень під час рекламних акцій чи свят	Великі компанії поліграфічної галузі	Автоматизація перевірки замовлень клієнтів, надійність роботи системи, мінімізація часу простоїв	Висока якість, бренд та успішні кейси

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища та визначено фактори загроз (таблиця 4.6) та можливостей (таблиця 4.7).

Хоча на даний момент і не має конкурентів, які б в повній мірі впровадили систему безперервної інтеграції на основі проведених ними досліджень, але в найближчі два роки очікується. Перешкодою були програмні платформи, які вже застаріли, адже створювалися багато років тому.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Впровадження технологій безперервної інтеграції конкурентами	В найближчі два роки очікується вихід на ринок декількох конкурентів з аналогічною архітектурою системи безперервної інтеграції	1. Якнайскоріший вихід на ринок для його завоювання, отримання успішних кейсів та підвищення впізнаваності бренду 2. Постійні інвестиції в удосконалення продукту 3. Подальше підвищення рівня автоматизації операцій
2	Зниження доходів потенційних клієнтів	Зниження доходів через нестабільну економічну ситуацію може призвести до призупинення замовлень веб-порталів	1. Пропонувати «коробочні» рішення за «підпискою» 2. Вихід на міжнародні ринки

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Збільшення попиту на веб-портали типу web-to-print	Постійне збільшення частки замовлень через мережу інтернет	Підтримка ціни на конкурентному рівні
2	Досягнення лідерських позицій на ринку України	Відсутність конкуренції з боку компаній на внутрішньому ринку	1. Участь в конференціях, виставках та інших локальних подіях 2. Підвищення впізнаваності бренду
3	Вихід на міжнародний ринок	Великий обсяг ринку та можливість залучення додаткових коштів	Підготовка англомовних інструкцій та спеціалістів
4	Розробка нових функцій автоматизації	Покращення продукту шляхом більшої автоматизації процесів	Підтримка інновацій шляхом постійних інвестицій

Наступним кроком було проведено аналіз пропозиції із визначенням загальних рис конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: олігополія	Існує декілька компаній-лідерів	Показати недоліки лідерів та свої переваги, постійно слідкувати за новими функціями конкурентів
2. Рівень конкурентної боротьби: міжнародний	Відсутність конкурентів на національному рівні	Адаптація продукту для вітчизняного ринку, так і підготовка спеціалістів для роботи із закордонними клієнтами
3. За галузевою ознакою: внутрішньогалузева	Продукт використовується в межах галузі	Зосередити зусилля на підтриманні конкурентних переваг на даному ринку
4. Конкуренція за видами товарів: товарно-видова	Конкуренція ведеться між рішеннями для web-to-print	Враховувати слабкі сторони конкурентів
5. Характер конкурентних переваг: неціновий	Веб-портали мають різну якість, функціонал та ефективність роботи з клієнтами	Підтримка якості, впровадження функціоналу шляхом «синхронізації» з новими потребами та регулярні оновлення
6. Інтенсивність: марочна	Впізнаваність бренду впливає на вибір клієнтів	Реклама в мережі інтернет, участь у галузевих подіях, успішні кейси з великими корпоративними клієнтами

Ступеневий аналіз конкуренції показав, що не дивлячись на конкуренцію зі сторони декількох західних компаній-лідерів, є високі шанси отримати лідерство на національному рівні завдяки адаптації для вітчизняного ринку, кращого функціоналу, більшої надійності та створеній архітектурі системи безперервної інтеграції.

Після ступеневого аналізу проведено аналіз умов конкуренції в галузі за М. Портером, який наведено у таблиці 4.9.

За результатами аналізу цієї таблиці зроблено висновок про можливість роботи на ринку. Для цього проєкт повинен мати готові рішення для web-to-print, прислухатися до потреб клієнтів, тобто бути максимально

клієнтоорієнтованим, адже саме вони диктують умови роботи на ринку. Замовникам потрібна максимальна автоматизація операцій для економії часу менеджерів, висока надійність роботи та захищеність, помірна ціна та можливість масштабування веб-порталу при зростанні навантажень чи під час пікових періодів.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу				
Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
W2P Shop, Key2Print, Racad Tech	Наявність готових рішень для web-to-print	Хмарні серверні ресурси на яких працює веб-портал: вартість, надійність та спектр	Якість та зручність	Поява єдиного онлайн-сервісу для замовлень web-to-print
Висновки:				
Помірна інтенсивність конкуренції	Є можливості входу на ринок. Строки їх виходу на ринок – 1.5-2 роки.	Постачальники не диктують умови роботи на ринку, адже їх надзвичайно велика кількість та вони майже аналогічні	Клієнти диктують всі умови роботи на ринку	Інтеграція в усі поліграфії, що бажають працювати з клієнтами через інтернет та необхідність клієнтам сплачувати комісії

Фактори конкурентоспроможності та їх обґрунтування наведено у таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1	Якість	Швидкість веб-порталу, робота без помилок та простоїв є надзвичайно важливою для кінцевих замовників поліграфічної продукції

Кінець таблиці 4.10

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
2	Рівень автоматизації	Автоматизація перевірки файлів від клієнтів, підрахунку ціни, проходження ланцюгу замовлення та оновлення статусів зберігає час менеджерів поліграфічного виробництва
3	Надійність та захищеність	Безперебійність роботи системи впливає на виконання виробничого плану, а захист даних клієнтів необхідний для того, щоб не було юридичних проблем та судів
4	Можливість масштабування при навантаженні	Під час проведення акцій, сезонних чи святкових скидок необхідно щоб веб-портал міг витримувати такі пікові навантаження. В інакшому випадку – втрата репутації та клієнтів
5	Вартість	Якщо є можливість отримати аналогічну за характеристиками систему web-to-print більш дешево – замовник це зробить
6	Витрати на обслуговування	Намагання зменшення витрат часу та коштів на обслуговування

За визначеними факторами конкурентоспроможності проаналізовано слабкі та сильні сторони проєкту (таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін старт-ап проєкту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Якість	18		+					
2	Рівень автоматизації	16	+						
3	Надійність та захищеність	15			+				
4	Можливість масштабування при навантаженні	20	+						
5	Вартість	14				+			
6	Витрати на обслуговування	12					+		

Проводячи аналіз таблиці 4.11 можна дійти висновку, що сильними сторонами проєкту є якість, рівень автоматизації та можливості

масштабування. А над зменшенням вартості потрібно обов'язково працювати, так само як і над здешевленням обслуговування.

SWOT-аналіз старт-апу було складено у таблиці 4.12.

Таблиця 4.12 – SWOT- аналіз старт-ап проєкту

Сильні сторони: 1. Високий рівень автоматизації 2. Якість 3. Можливості масштабування під час збільшення навантажень 4. Висока надійність та захищеність 5. Порівняно нейтральна вартість	Слабкі сторони: 1. Дещо більші витрати на обслуговування
Можливості: 1. Збільшення попиту на веб-портали типу web-to-print 2. Досягнення лідерських позицій на ринку України 3. Вихід на міжнародний ринок 4. Розробка нових функцій автоматизації	Загрози: 1. Конкуренція технологій безперервної інтеграції 2. Цінова конкуренція

Розроблено альтернативи ринкової поведінки (таблиця 4.13) на основі SWOT-аналізу, який наведено у таблиці 4.12. Для альтернатив визначено строки та ймовірність отримання ресурсів.

Таблиця 4.13 – Альтернативи ринкового впровадження старт-ап проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Вихід на ринок із базовою версією, активна реклама в інтернеті. Поступове підвищення якості та збільшення функціональності	70%	3 місяців
2	Залучення клієнтів через галузеві події, видання та мережу інтернет. Вихід на ринок з високим рівнем автоматизації та великою кількістю функцій	90%	7 місяців

Враховуючи те, що потенційні проєкти конкурентів з'являться щонайменше через рік, то обрано другу альтернативу – вона має більшу ймовірність отримання ресурсів.

4.4 Розроблення ринкової стратегії проєкту

Опис цільових груп потенційних споживачів наведено у таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі поліграфії	Низька	25%	Висока	Середня
2	Типографії середнього розміру	Висока	40%	Низька	Середня
3	Великі поліграфічні компанії	Висока	80%	Низька	Висока
Які цільові групи обрано: 2 та 3					

Наступним кроком сформовано базову стратегію розвитку (таблиця 4.15).

В якості базової стратегії розвитку обрано стратегію диференціації, яка передбачає надання продукту важливих з точки зору споживача відмінних властивостей, які роблять його відмінним від конкурентів, в нашому випадку впровадження системи безперервної інтеграції з архітектурою, яку було розроблено під час досліджень.

Далі було визначено базову стратегію конкурентної поведінки (таблиця 4.16) та стратегію позиціонування (таблиця 4.17).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Залучення клієнтів через галузеві події, видання та мережу інтернет. Вихід на ринок з високим рівнем автоматизації та великою кількістю функцій	Стратегія диференційованого маркетингу	Високий рівень якості, автоматизації процесів, надійності та захищеності, продумана архітектура системи безперервної інтеграції	Стратегія диференціації

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	В національному масштабі – так, у глобальному – ні	В переважній більшості – шукати нових	Ні	Стратегія заняття конкурентної ніші

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту (три ключових)
1	Висока якість, надійність, бренд, простота	Стратегія диференціації	Використання обраної шляхом досліджень архітектури системи безперервної інтеграції, високий рівень автоматизації та якості при помірній вартості	Якість, ергономічність, простота

В результаті виконання даного підрозділу отримано узгоджену систему рішень щодо ринкової поведінки стартап-компанії.

4.5 Розроблення маркетингової програми старт-ап проєкту

Виконано формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 4.18 підсумовано результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Отримання замовлень через інтернет	Клієнти самостійно створюють замовлення та заносять всі необхідні дані	Автоматизація перевірки файлів клієнтів
2	Надійна і безперебійна робота	Мінімізація часу простоїв	Автоматичне тестування будь-яких змін перед завантаженням на робочі сервери
3	Витримування навантажень під час пікових чи підвищених навантажень	Підвищення швидкості роботи за рахунок автоматичного керування серверними ресурсами та запобігання «падіння» під час пікових навантажень	Спеціально спроектована архітектура системи безперервної інтеграції

Наступним кроком розроблено тривірневу маркетингову модель товару, результати занесено до таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Зручне, надійне та захищене розташування замовлень кінцевими клієнтами за допомогою веб-порталу типу web-to-print в мережі інтернет		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Час генерації веб-сторінки сервером (TTFB)	мс	<200
	2. Вага веб-сторінки	мб	<3
	3. Висока якість сирцевого коду		
	4. Надійність роботи		
	5. Легкість масштабування при навантаженні (в т.ч. піках)		
6. Широкий функціонал автоматизації роботи менеджера			
7. Унікальна та продумана архітектура системи безперервної інтеграції			
	Якість: стандарт коду PSR-2, цикломатична складність до 10		
	Пакування: –		
	Марка: HALEX Studios, продукт: W2P Framework		
III. Товар із підкріпленням	До продажу: з’ясування конкретних потреб та проблем поліграфії-клієнта		
	Після продажу: постійне оновлення по «підписці»		
За рахунок чого потенційний товар буде захищено від копіювання: патент на корисну модель та комерційна таємниця			

Наступним кроком визначено межі ціни за допомогою аналізу експертним методом (таблиця 4.20).

Виконано визначення оптимальної системи збуту та результати занесено до таблиці 4.21.

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	5-15 тис \$	20-50 тис \$	200 тис \$	8-35 тис \$

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Поліграфії замовляють web-to-print рішення у центрів з розробки веб-порталів або створюють власні рішення, залучаючи необхідних спеціалістів	Встановлення та підтримка контактів із клієнтами	Нульового рівня (виробник продає товар клієнту)	Залучення клієнтів за допомогою мережі інтернет (власного сайту компанії)

Останнім кроком є визначення концепції маркетингових комунікацій. Концепцію наведено у таблиці 4.22.

Результатом виконання підрозділу стала ринкова програма, що включає в себе концепції товару, просування, збуту і попередній аналіз меж ціни, спирається на потреби та цінності потенційних клієнтів.

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти обирають зручний, простий, надійний та функціональний товар, який має велику кількість автоматизацій	Соціальні мережі, мобільний телефон та електронна пошта	Якість, зручність, надійність, простота, захищеність, автоматизація, доступна ціна	Демонстрація товару, його позиціонування з метою залучення клієнтів	«Web-to-print – цього хочуть ваші клієнти!»

Висновки до четвертого розділу

В даному розділі було розроблено старт-ап проєкт на основі підприємства, що проєктовано в попередньому розділі та з використанням

дослідів архітектури системи безперервної інтеграції, зроблених у другому розділі. Можна зазначити, що попит на виготовлення веб-порталів типу web-to-print збільшується щороку, адже це зручно, просто та швидко для кінцевих замовників поліграфічної продукції.

На ринку існує декілька компаній-лідерів, наявна олігополія. Для того, щоб вийти на ринок потрібно буде докласти зусиль, але стратегія зайняття конкурентної ніші дозволить уникнути прямої та непотрібної конкуренції з вже існуючими рішеннями. Висока конкурентоспроможність досягається завдяки високому рівню якості, надійності, автоматизації та впровадження системи безперервної інтеграції з архітектурою на основі досліджень. Також важливу роль грає відсутність конкурентів на національному рівні, всі рішення для web-to-print створені західними компаніями та не адаптовані під вітчизняний ринок.

Для ринкового впровадження стартап проєкту було обрано альтернативу, яка передбачає вихід повної версії продукту з великою кількістю функцій та автоматизацій. А потім реклама та створення бренду завдяки участі у галузевих конференціях, статтях у спеціалізованих виданнях та широкому застосуванню мережі інтернет (пошукові системи, соціальні мережі).

Отже, можна зробити висновок, що подальша імплементація є доцільною, адже стартап може знайти споживачів на динамічно зростаючому ринку web-to-print.

ЗАГАЛЬНІ ВИСНОВКИ

У першому розділі проведено аналітичний огляд за допомогою науково-технічної літератури та інтернет-джерел. Виявлено, що тема актуальна та стрімко розвивається. Представлено загальну схему процесу безперервної інтеграції, розроблено класифікацію систем безперервної інтеграції за основними ознаками. Побудовано типову архітектуру CI/CD-системи, а за допомогою експертних оцінок та діаграми Парето виокремлено ряд основних факторів, які мають безпосередній вплив на архітектуру системи безперервної інтеграції. Розглянуто програмне забезпечення, необхідне для функціонування системи. Визначено метрики, за допомогою яких вимірюється ефективність роботи системи та проведено патентний пошук сучасних розробок глибиною 10 років.

У другому розділі побудовано графік публікації патентів за роками та кумулятивну криву, які підтверджують актуальність теми та високу зацікавленість багатьох компаній. Увиражено об'єкт та предмет дослідження. Розроблено тестові файли, а точніше програмний код, за допомогою якого будуть проводитися дослідження поведінки веб-порталу під навантаженням. За допомогою оцінок експертів за семи критеріями визначено раціональне базове програмне забезпечення – GitHub Actions. Розроблено 2 варіанти структури з відповідно підібраними інструментами, враховуючи специфіку web-to-print. В якості середовища створення навантаження обрано Google Cloud Functions, а в якості тестового веб-порталу – «Printlance». За допомогою трьох варіантів навантаження, які представляють собою криву кількості користувачів відносно часу, виміряно зміну основних метрики та визначено структуру масштабування, що задовольняє потреби веб-порталів типу web-to-print. За допомогою домінантної ієрархічної впорядкованої моделі визначено найважливіші критерії впливу на вибір базового набору програмного забезпечення. Останнім кроком, на основі досліджень, створено архітектуру системи безперервної інтеграції для веб-порталів типу web-to-print.

У третьому розділі розроблено та розраховано відповідне до сучасних тенденцій промислове завдання. Для кожної з позицій наведено структуру основних модулів. Також наведено схему роботи панелі для адміністраторів і типові сторінки для всіх видів веб-порталів. За допомогою аналітичного методу «чорна скринька» обрано технологію створення back-end частини веб-порталів – фреймворк Laravel, а за допомогою циклограми обрано технології front-end – Bootstrap та JQuery. Для найбільш трудомістких етапів виробничого процесу створено поопераційні карти. Обрано сучасне програмне забезпечення та складено мінімальні системні вимоги. Враховуючи вимоги, за допомогою пелюсткової діаграми, обрано апаратне забезпечення – моноблок. Розроблено ефективну організаційну структуру виробництва та наведено узагальнену блок-схему технологічного створення веб-порталу. Вирахувано раціональний обсяг виробництва, трудомісткість робіт, необхідна кількість устаткування та робочих місць, кількість працюючих. Розроблено технологічний та генеральний плани підприємства з наведенням 3Д-моделі. Складено специфікацію на розробку конструкцій перекриття та шумоізоляції, завдання на інженерно-технічне забезпечення виробничих процесів. Створено узагальнену структурну електричну схему комп'ютерної мережі. Розраховано відносні та абсолютні техніко-економічні показники проєкту. Виконано оцінювання ризиків, їх сортування від найбільшого до найменшого.

У четвертому розділі описано ідею проєкту. Визначено сильні, нейтральні та слабкі характеристики проєкту. Проведено оцінку ринку та надано характеристику потенційних клієнтів старт-ап проєкту. Визначено фактори загроз та можливостей, проведено ступеневий аналіз конкуренції на ринку. Важливими факторами можливостей є: відсутність конкуренції з боку компаній на внутрішньому ринку, можливість виходу на міжнародний ринок та висока якість завдяки автоматизаціям. Створено ринкову програму. Визначено, що реалізація проєкту є доцільною, адже старт-ап володіє унікальними перевагами, зможе знайти клієнтів та отримувати прибуток.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуров О. В., Особливості та переваги використання систем безперервної інтеграції для web-to-print/20-а Міжнародна науково-технічна конференція студентів та аспірантів «Друкарство молоде». – К.: НТУУ «КПІ ім. Ігоря Сікорського», 07-09 квітня 2020 р., 63-65 с.

2. 20 Best Continuous Integration(CI) Tools in 2020 [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/top-20-continuous-integration-tools.html> – Назва з екрана.

3. What is CI/CD? Continuous integration and continuous delivery explained [Електронний ресурс]. – Режим доступу: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html> – Назва з екрана.

4. What is CI/CD? [Електронний ресурс]. – Режим доступу: <https://www.redhat.com/en/topics/devops/what-is-ci-cd> – Назва з екрана.

5. Величко, О. М. Проектування технологічних процесів видавничо-поліграфічного виробництва [Електронний ресурс] : навчальний посібник для студентів напряму підготовки 6.051501 «Видавничо-поліграфічна справа» / О. М. Величко, В. М. Скиба, А. В. Шангін; НТУУ «КПІ». – Електронні текстові дані (1 файл: 1,71 Мбайт). – Київ : НТУУ «КПІ», 2014. – 235 с. – Назва з екрана. – <http://ela.kpi.ua/handle/123456789/8538>.

6. Проектування та розрахунки виробничих процесів [Електронний ресурс] : електронні таблиці для виробничих розрахунків для студентів спеціальностей «Технологія друкованих видань», «Технологія розробки, виготовлення та оформлення пакувань»/НТУУ «КПІ»; уклад. В. І. Кульбич. – Електронні текстові дані (1 файл: 752 Кбайт; 1 файл: 1,04 Мбайт). – Київ : НТУУ «КПІ», 2010. – Назва з екрана. – <http://library.ntu-kpi.kiev.ua:8080/handle/123456789/460>.

7. Величко О. Видавничо-поліграфічна справа: Практикум з проектування і розрахунку технологічних і виробничих процесів / Олена

Величко [Текст] : навч. посіб. –К.: ВПЦ «Київський університет», 2009. – 520 с.

8. Методичні вказівки до виконання комп'ютерного практикуму з дисципліни «Проектування видавничо-поліграфічного виробництва. Модуль 1 – Проектування технологічних процесів» для студентів напрямку 6.051501 «Видавничо-поліграфічна справа» спеціальностей «Технології друкованих видань», «Технології електронних мультимедійних видань», «Комп'ютерні технології та системи видавничо-поліграфічних виробництв», «Матеріали видавничо-поліграфічних виробництв», «Технології розробки, виготовлення і оформлення пакувань» [Електронний ресурс] / НТУУ «КПІ»; уклад. О. М. Величко, В. М. Скиба. – Електронні текстові дані (1 файл: 499 Кбайт). – Київ : НТУУ «КПІ», 2014. – 25 с. – Назва з екрана. – <http://ela.kpi.ua/handle/123456789/7733>.

9. SVN, Git Out of Here: How to Choose Your Version Control System [Електронний ресурс]. – Режим доступу: <https://medium.com/@MentorMate/svn-git-out-of-here-how-to-choose-your-version-control-system-10b44750a75c> – Назва з екрана.

10. BEST Version Control Software (Source Code Management Tools) [Електронний ресурс]. – Режим доступу: <https://www.softwaretestinghelp.com/version-control-software/> – Назва з екрана.

11. 15 Metrics for DevOps Success [Електронний ресурс]. – Режим доступу: <https://stackify.com/15-metrics-for-devops-success/> – Назва з екрана.

12. DevOps KPI in Practice [Електронний ресурс]. – Режим доступу: <https://dzone.com/articles/devops-kpi-in-practicechapter-1deployment-speed-fr> – Назва з екрана.

13. Finding and defining good metrics in the DevOps environment [Електронний ресурс]. – Режим доступу: <https://www.aservo.com/en/know-how/slas-and-kpis-for-cicd> – Назва з екрана.

14. 7 Continuous Delivery Metrics to Track [Электронный ресурс]. – Режим доступа: <https://www.datical.com/blog/7-metrics-to-track-when-implementing-continuous-delivery/> – Назва з екрана.

15. Tech Talk: The Continuous Delivery Metrics That Will Make Your Pipeline Fast and Your Apps Better [Электронный ресурс]. – Режим доступа: <https://www.slideshare.net/CAinc/tech-talk-the-continuous-delivery-metrics-that-will-make-your-pipeline-fast-and-your-apps-better> – Назва з екрана.

16. US20140053138A1 - Quality on submit process - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US20140053138A1> – Назва з екрана.

17. US20180285128A1 - Automated user interface analysis - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US20180285128A1> – Назва з екрана.

18. US20190370010A1 - Integrated mainframe distributed orchestration tool - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US20190370010A1> – Назва з екрана.

19. US20190369980A1 - Constraint-based upgrade and deployment - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US20190369980A1> – Назва з екрана.

20. US8863114B2 - Managing software packages using a version control system - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US8863114B2> – Назва з екрана.

21. US7519964B1 - System and method for application deployment in a domain for a cluster - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US7519964B1> – Назва з екрана.

22. JP2011053966A - Information processing apparatus, software management system, and software management method - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/JP2011053966A> – Назва з екрана.

23. KR20130068524A - Method and apparatus for web runtime updating of terminal - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/KR20130068524A> – Назва з екрана.

24. US9710259B2 - System and method for customizing a deployment plan for a multi-tier application in a cloud infrastructure - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US9710259B2> – Назва з екрана.

25. US10120670B1 - Systems and methods for building generic CI/CD pipelines - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10120670B1> – Назва з екрана.

26. CN101978366B - Application management within deployable object hierarchy - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CN101978366B> – Назва з екрана.

27. US9645858B2 - Single, logical, multi-tier application blueprint used for deployment and management of multiple physical applications in a cloud infrastructure - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US9645858B2> – Назва з екрана.

28. CN105516233B - Method and system for application deployment portable on one or more cloud systems - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CN105516233B> – Назва з екрана.

29. CA3019911A1 - Techniques for evaluating server system reliability, vulnerability and component compatibility using crowdsourced server and vulnerability data - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CA3019911A1> – Назва з екрана.

30. CN104520818A - Performance tests in a continuous deployment pipeline - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CN104520818A> – Назва з екрана.

31. US8903943B2 - Integrating cloud applications and remote jobs - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US8903943B2> – Назва з екрана.

32. AU2018203054B2 - System of distributed software quality improvement - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/AU2018203054B2> – Назва з екрана.

33. CN104185840B - It is used for being prioritized the methods, systems and devices of multiple tests in lasting deployment streamline - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CN104185840B> – Назва з екрана.

34. US9886375B2 - Automated execution of functional test scripts on a remote system within a unit testing framework - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US9886375B2> – Назва з екрана.

35. US20190138301A1 - Systems and methods for providing ranked deployment options - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US20190138301A1> – Назва з екрана.

36. CN102693183B - Method and system for realizing automatic software testing - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CN102693183B/> – Назва з екрана.

37. US10033833B2 - Apparatus, systems and methods for automatic distributed application deployment in heterogeneous environments - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10033833B2> – Назва з екрана.

38. US9760366B2 - Maintaining deployment pipelines for a production computing service using live pipeline templates - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US9760366B2> – Назва з екрана.

39. US8813040B2 - Methods and systems for reporting on build runs in software development - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US8813040B2> – Назва з екрана.

40. US9760343B2 - Application builder based on metadata - Google Patents [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US9760343B2> – Назва з екрана.

41. US10437585B2 - Managing highly scalable continuous delivery pipelines [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10437585B2> – Назва з екрана.

42. CA2970824C - System and method for elastic scaling using a container-based platform [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/CA2970824C> – Назва з екрана.

43. US10521747B2 - System and method for providing a scalable semantic mechanism for policy-driven assessment and effective action taking on dynamically changing data [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10521747B2> – Назва з екрана.

44. US10411975B2 - System and method for a cloud computing abstraction with multi-tier deployment policy [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10411975B2> – Назва з екрана.

45. US10230571B2 - Microservice-based application development framework [Электронный ресурс]. – Режим доступа: <https://patents.google.com/patent/US10230571B2> – Назва з екрана.

46. 15 Best Continuous Integration Tools In 2020 (Compared) [Электронный ресурс]. – Режим доступа: <https://www.softwaretestinghelp.com/tools/24-best-continuous-integration-tool/> – Назва з екрана.

47. Best 14 CI/CD Tools You Must Know | Updated for 2020 [Электронный ресурс]. – Режим доступа: <https://www.katalon.com/resources-center/blog/ci-cd-tools/> – Назва з екрана.

48. 27 Of The Best CI/CD Tools Available Today [Электронный ресурс]. – Режим доступа: <https://www.lambdatest.com/blog/27-best-ci-cd-tools/> – Назва з екрана.

49. Top 10 CI/CD Tools – Which is Best for You? [Електронний ресурс]. – Режим доступу: <https://blog.testproject.io/2020/11/09/top-ci-cd-tools/> – Назва з екрана.

50. Cloud Functions documentation [Електронний ресурс]. – Режим доступу: <https://cloud.google.com/functions/docs?hl=ru> – Назва з екрана.

51. Node.js [Електронний ресурс]. – Режим доступу: <https://nodejs.org/uk/>

52. Puppeteer [Електронний ресурс]. – Режим доступу: <https://developers.google.com/web/tools/puppeteer> – Назва з екрана.

53. Wolfram|Alpha: Computational Intelligence [Електронний ресурс]. – Режим доступу: <https://www.wolframalpha.com/> – Назва з екрана.

54. Гуров О. В., Порівняльний аналіз фреймворків для створення веб-порталу/19-а Міжнародна науково-технічна конференція студентів та аспірантів «Друкарство молоде». – К.: НТУУ «КПІ ім. Ігоря Сікорського», 07-09 квітня 2019 р., 74-78 с.

55. Windows 10 System requirements [Електронний ресурс]. – Режим доступу: <https://www.microsoft.com/en-us/windows/windows-10-specifications> – Назва з екрана.

56. Photoshop CC system requirements [Електронний ресурс]. – Режим доступу: <https://helpx.adobe.com/photoshop/system-requirements/earlier-versions.html> – Назва з екрана.

57. Illustrator CC system requirements [Електронний ресурс]. – Режим доступу: <https://helpx.adobe.com/illustrator/system-requirements/earlier-versions.html> – Назва з екрана.

58. System Requirements [Електронний ресурс]. – Режим доступу: https://www.jetbrains.com/resharper/download/system_requirements.html – Назва з екрана.

59. Chrome System Requirements [Електронний ресурс]. – Режим доступу: <https://chrome.google.com/system-requirements> – Назва з екрана.

60. HP 200 G3 AiO [Электронный ресурс]. – Режим доступа: https://hard.rozetka.com.ua/hp_8pg80es/p185233336/characteristics/ – Назва з екрана.

61. Asus Vivo AiO V241FAK-BA059D – Режим доступа: https://hard.rozetka.com.ua/asus_90pt0292_m07070/p177430877/ – Назва з екрана.

62. Artline Business M61 v06 – Режим доступа: https://hard.rozetka.com.ua/artline_m61_v06/p177143310/ – Назва з екрана.

63. How to Estimate Your Hosting Bandwidth Usage [Электронный ресурс]. – Режим доступа: <https://www.hostpapa.com/blog/web-hosting/how-to-estimate-your-hosting-bandwidth-usage/> – Назва з екрана.

64. How to Estimate Bandwidth Requirements for a Business Site or Network [Электронный ресурс]. – Режим доступа: <https://helpdeskgeek.com/networking/how-to-estimate-bandwidth-requirements-for-a-business-site-or-network/> – Назва з екрана.

65. Bandwidth calculator [Электронный ресурс]. – Режим доступа: <https://oneringnetworks.com/bandwidth-calculator/> – Назва з екрана.

66. UTF-8 [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/UTF-8> – Назва з екрана.

67. JPEG [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/JPEG> – Назва з екрана.

68. Common Digital Photo File Formats [Электронный ресурс]. – Режим доступа: <http://www.rideau-info.com/photos/filetypes.html> – Назва з екрана.

69. Технология обработки текстовой информации: Часть I. основы технологии издательских и наборных процессов: Глава VII. основные правила и особенности набора текстов возрастающей сложности [Электронный ресурс] – Режим доступа: <http://www.hi-edu.ru/e-books/xbook337/01/part-008.htm> – Назва з екрана.

70. Технология обработки изобразительной информации: Глава I. Подготовка к печати, параметры изображений [Электронный ресурс] – Режим

доступу: <http://www.hi-edu.ru/e-books/xbook438/01/part-002.htm#i102> – Назва з екрана.

71. Розрахунок кількості робочих місць [Електронний ресурс] – Режим доступу: <https://studfile.net/preview/8168211/page:5/> – Назва з екрана.

72. Скільки працюємо у 2020 році [Електронний ресурс] – Режим доступу: <http://vobu.ua/ukr/directory/prodolzhitelnost-rabocheho-vremeny> – Назва з екрана.

73. Коммутатор TP-Link TL-SG105 [Електронний ресурс]. – Режим доступу: <https://secur.ua/setevoe-oborudovanie/kommutatoribezpoe/kommutator-tp-link-tl-sg105.html> – Назва з екрана.

74. Ціни на універсальні послуги [Електронний ресурс] – Режим доступу: <https://yasno.com.ua/business/b2b-tariffs> – Назва з екрана.

75. Методичні вказівки до виконання розрахункової роботи з дисципліни «Економіка та організація виробництва» для всіх форм навчання напряму підготовки (спеціальності) 6.050503 «Машинобудування»: [Електронний ресурс] / Укладачі: Я. В. Котляревський, М. В. Сірик. – Київ : НТУУ «Київський політехнічний інститут», 2015. – 18 с.

76. Тарифы на холодную воду в г. Киев [Електронний ресурс] – Режим доступу: https://www.prostobank.ua/spravochniki/indikatory_rynka/tarify_na_kholodnuyu_vodu/1 – Назва з екрана.

77. Про затвердження Методики розрах... | від 03.09.2012 № 449 [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z1606-12>. – Назва з екрана.

78. Вакансії DOU [Електронний ресурс] – Режим доступу: <https://jobs.dou.ua/> – Назва з екрана.

ДОДАТКИ

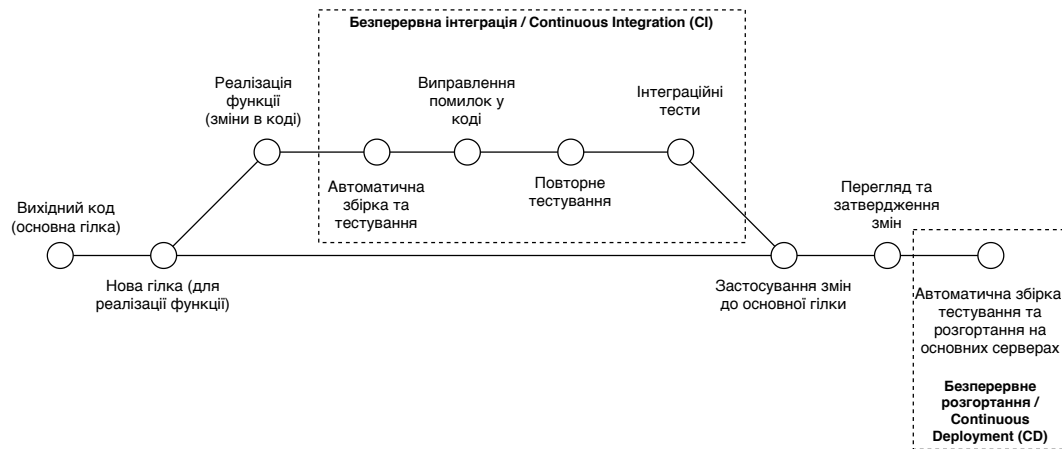


Рисунок 1.1 – Загальна схема процесу безперервної інтеграції



Рисунок 1.2 – Класифікація CI/CD систем

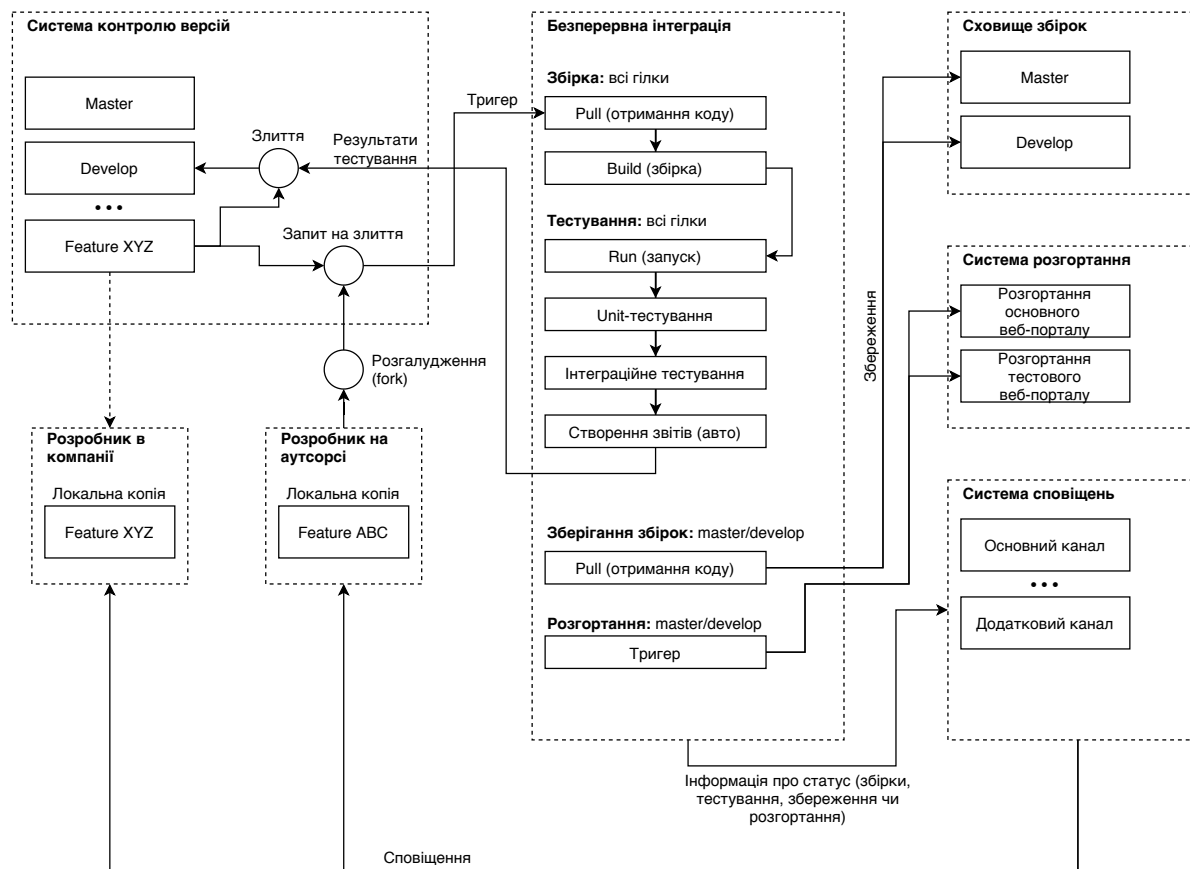


Рисунок 1.3 – Типова загальна архітектура системи безперервної інтеграції

				Центр з розроблення веб-порталів зі створенням і дослідженнями архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм.	Арх.	Підпис	Дата	Зм.	Маса	Місяць	Рік
Розробник	Гуров О. В.						
Перевірник	Кузнецов В. Р.						
Т. конгр.							
М. конгр.							
Виконав	Штефан С. В.			Арх. 1	Архушіт	11	
КТП ім. Ігоря Сікорського ВП				РП-91мп			

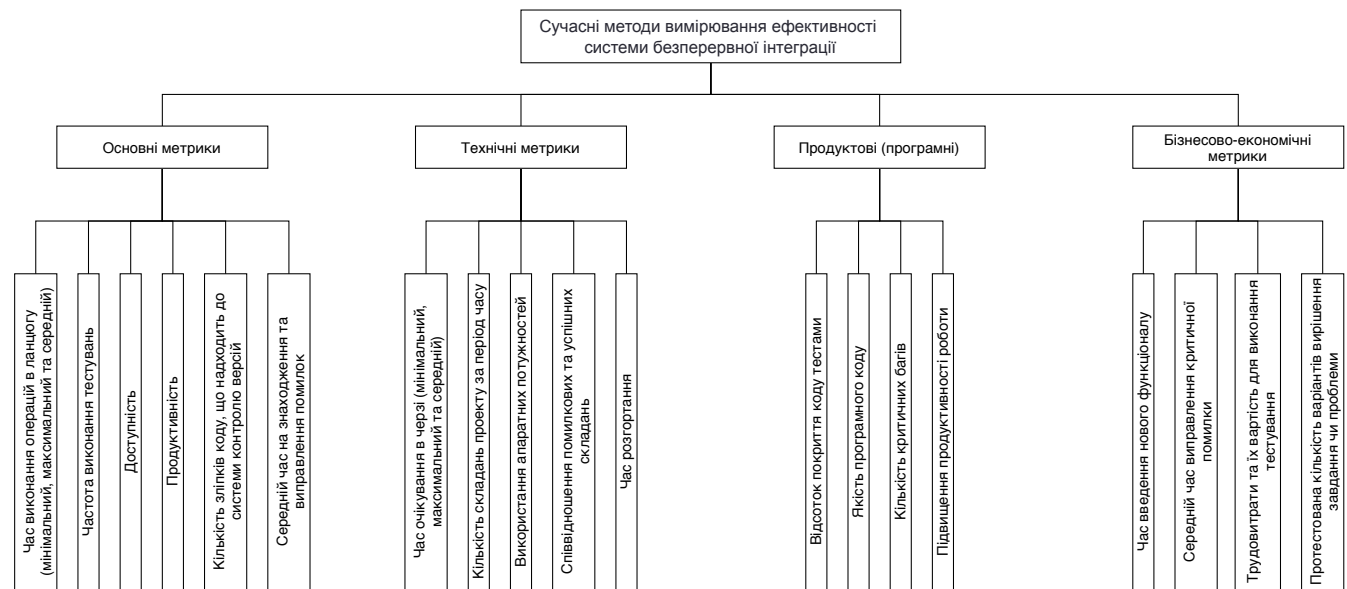
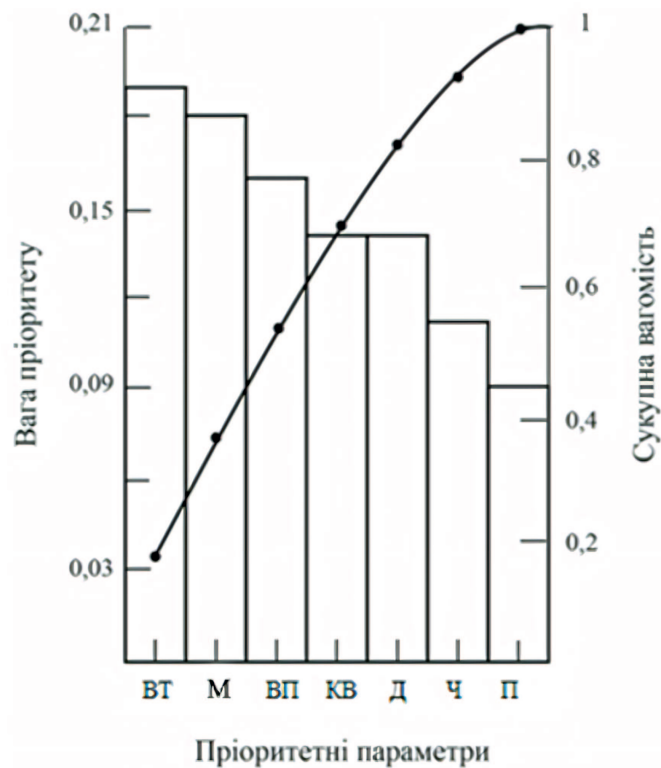


Рисунок 2.2 – Класифікація метрик систем CI/CD

Рисунок 2.1 – Діаграма Парето для експертної оцінки основних факторів, що впливають на архітектуру системи безперервної інтеграції:

Д – доступність,
 П – продуктивність,
 М – масштабування серверної частини,
 ВП – вартість підтримки системи,
 VT – витрати на тестування,
 Ч – час введення нового функціоналу,
 KB –кваліфікація технічних спеціалістів

				Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм.		Арх.	Підпис	Дата	Діп.	Маса	Масштаб
Розробка		Гуров О. В.					
Перевірка		Кудиних Б. Р.					
Т. контр.					Арх. 2	Архушія	11
Ш. контр.					КПІ ім. Ігоря Сікорського ВНП		
Затверд.		Штефан С. В.			РП-91мп		

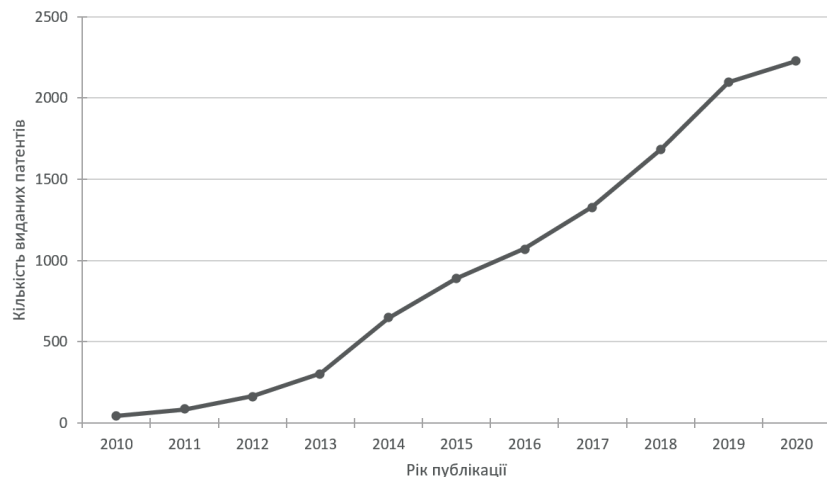


Рисунок 3.1 – Кумулятивна крива публікації патентів

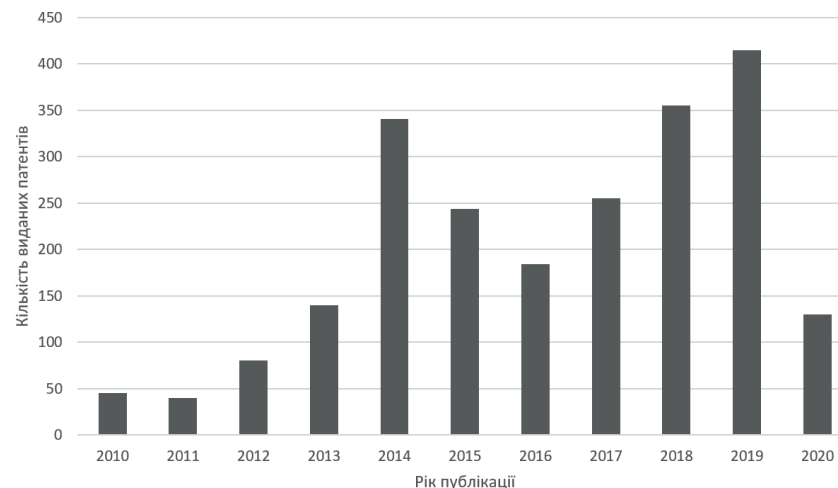


Рисунок 3.2 – Динаміка публікації патентів за роками

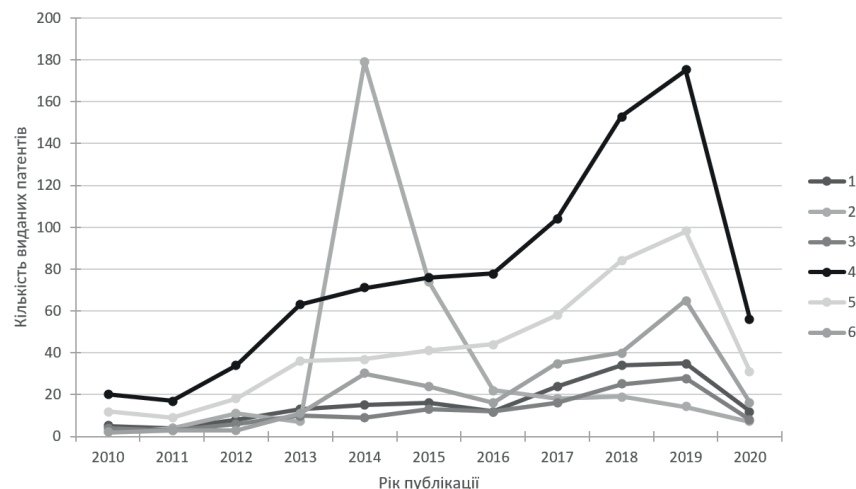


Рисунок 3.3 – Кумулятивна крива публікації патентів за окремими напрямками: 1 – архітектура системи безперервної інтеграції; 2 – структура ланцюгу CI/CD; 3 – вплив вибору інструментів на ефективність процесу CI/CD; 4 – автоматизація тестування та інтеграційне тестування; 5 – автоматична збірка програмного коду; 6 – масштабування за допомогою системи безперервної інтеграції.

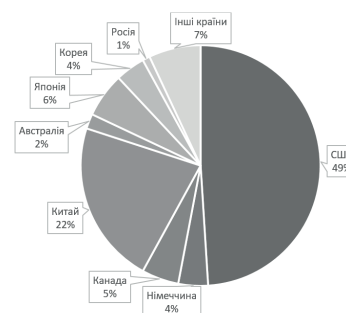


Рисунок 3.4 – Співвідношення патентування розробок за країнами-власниками патентів

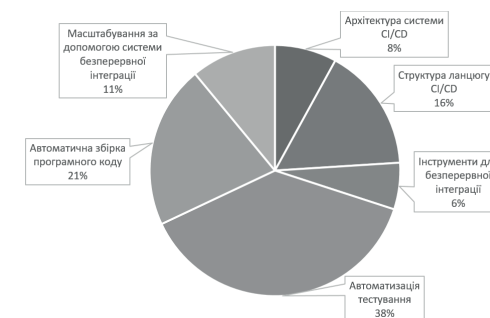


Рисунок 3.5 – Розподіл патентів за предметом пошуку

				Центр з розроблення веб-порталів зі створенням і дослідженнями архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм.	Арх.	Підпис	Дата				
Розробник	Гуров О. В.						
Перевірник	Кузнецов В. Р.						
Г. конгр.							
Модератор							
Інтерпретатор	Штефан С. В.						
				1. Кумулятивні криві та динаміка публікації патентів 2. Розподіл за країнами-власниками та предметом пошуку			
				КПІ ім. Ігоря Сікорського ВНП			
				РП-91мп			

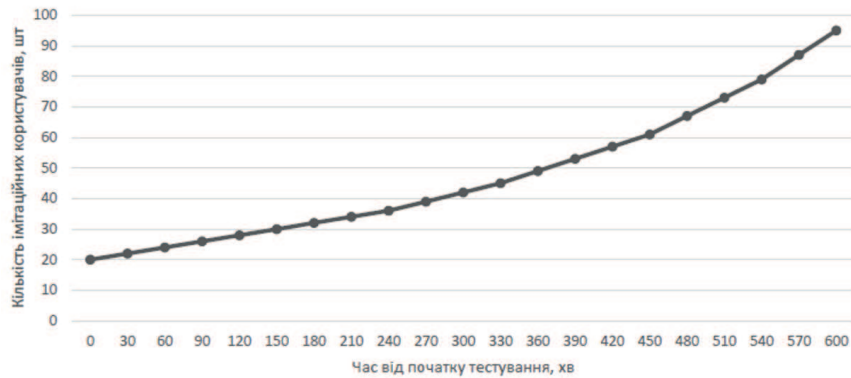


Рисунок 4.1 – Крива кількості імітаційних користувачів відносно часу для навантаження «вдала рекламна кампанія»

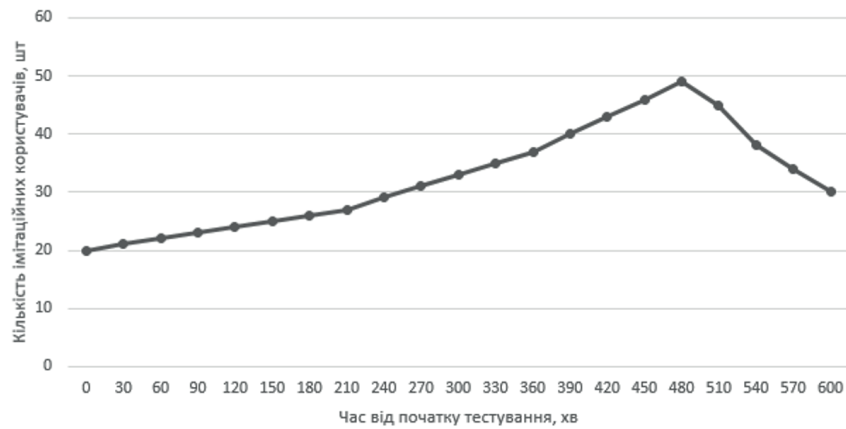


Рисунок 4.3 – Крива кількості імітаційних користувачів відносно часу для навантаження «потрібно виконати план до Нового Року»

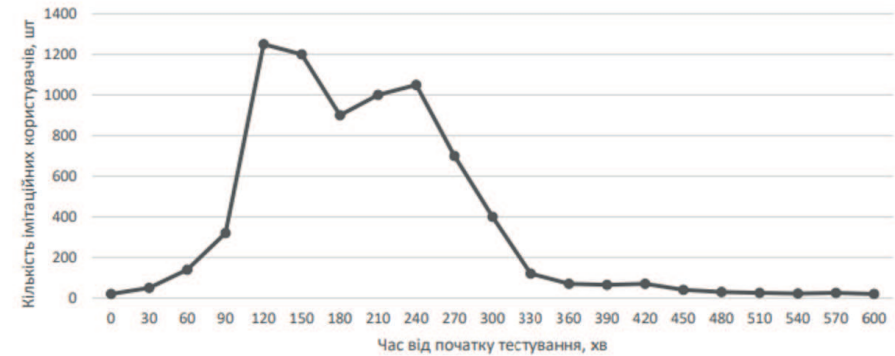


Рисунок 4.2 – Крива кількості імітаційних користувачів відносно часу для навантаження «опублікували посилання на головній сторінці urk.net»

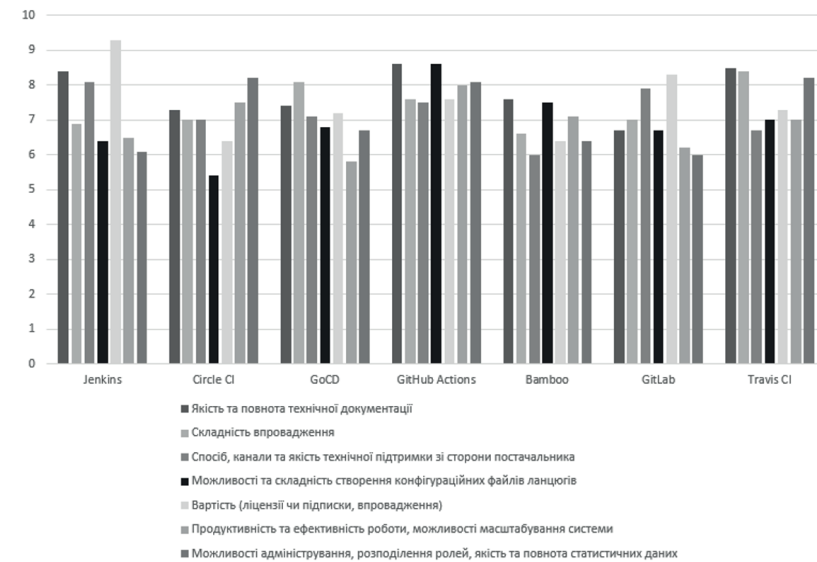


Рисунок 4.4 – Порівняння середніх значень по кожному критерію оцінки якості базового програмного забезпечення

				Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм.	Арх.	Підпис	Дата	1. Криві кількості імітаційних користувачів відносно часу	2. Порівняння середніх значень по кожному критерію оцінки якості базового програмного забезпечення	3. Порівняння середніх значень по кожному критерію оцінки якості базового програмного забезпечення	4. Порівняння середніх значень по кожному критерію оцінки якості базового програмного забезпечення
Розробник	Гуров О. В.						
Перевірник	Кудинюк В. Р.						
Г. контр.							
Штат	Штат						
Інтерпр.	Штат						
КПІ ім. Ігоря Сікорського ВПІ				РПІ-91мп			

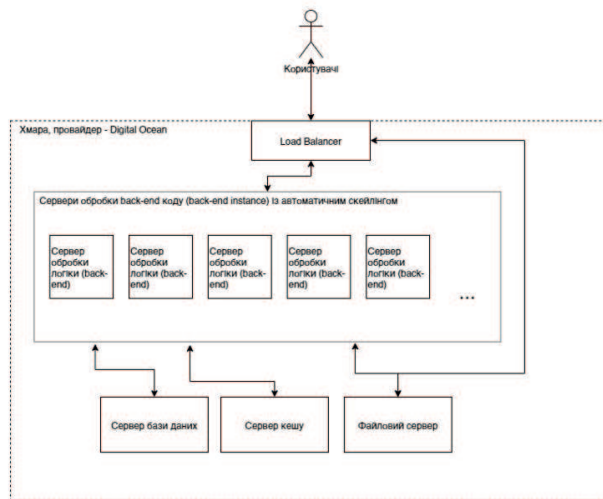


Рисунок 5.1 – Перший варіант структури масштабування

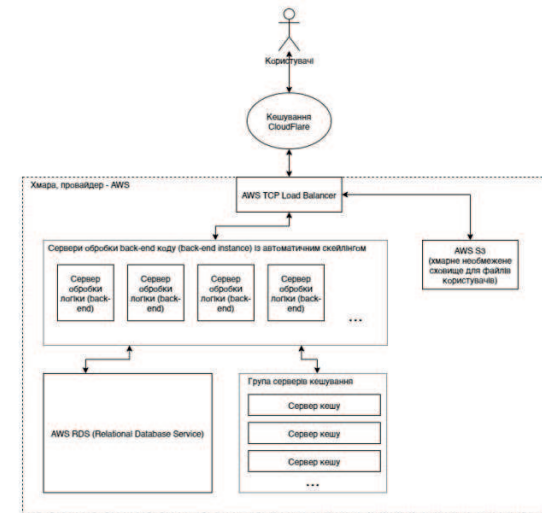


Рисунок 5.2 – Другий варіант структури масштабування

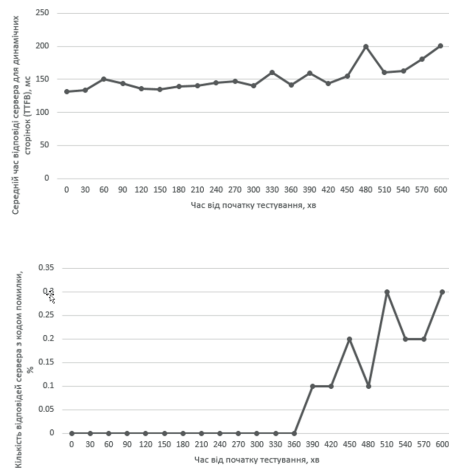


Рисунок 5.3 – Поведінка метрик першого варіанту структури при навантаженні «вдала рекламна кампанія»

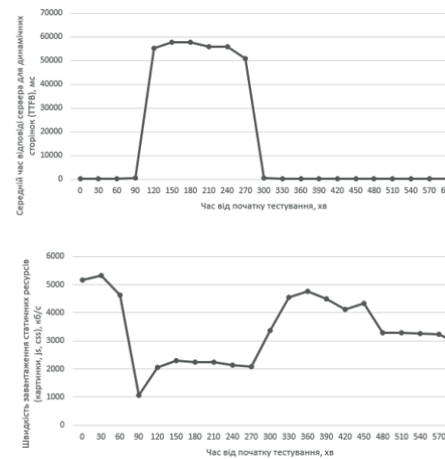
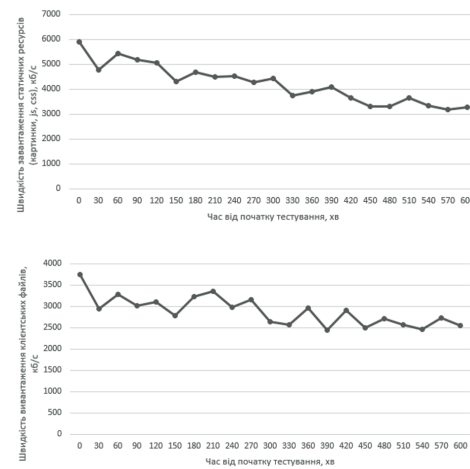


Рисунок 5.4 – Поведінка метрик першого варіанту структури при навантаженні «опублікували посилання на головній сторінці urk.net»

				Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм. Арх.	Розробл.	Гуров О. В.	Підпис	Дат.	Дат.	Маса	Масштаб
Перевір.	Кудиник В. Р.						
Г. контр.							
Підпис	Штефан С. В.						
Затверд.							
КПІ ім. Ігоря Сікорського ВНП				РП-91мп			

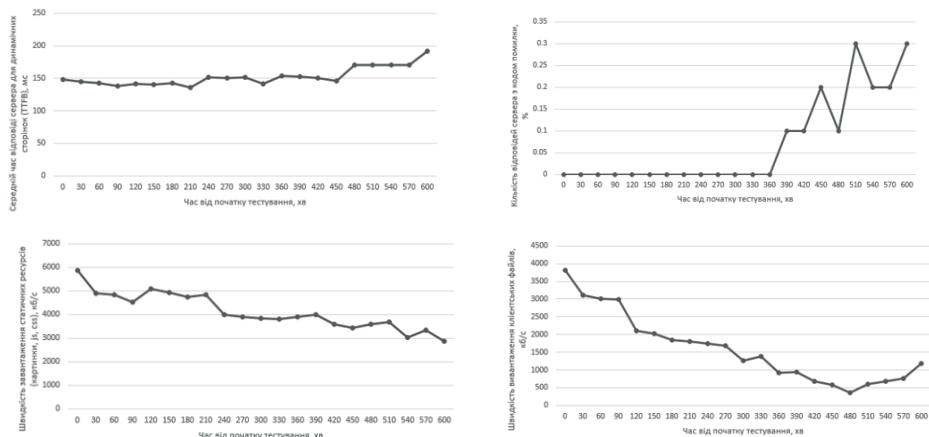


Рисунок 5.1 – Поведінка метрик першого варіанту структури при навантаженні «потрібно виконати план до Нового Року»

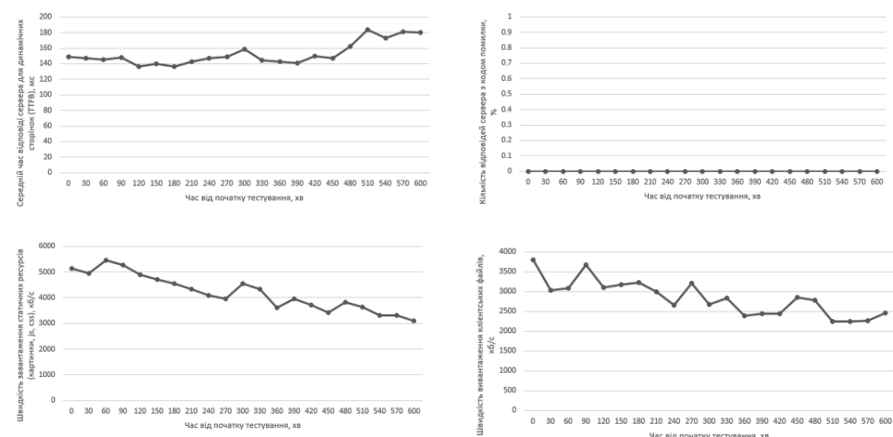


Рисунок 5.2 – Поведінка метрик другого варіанту структури при навантаженні «вдала рекламна кампанія»

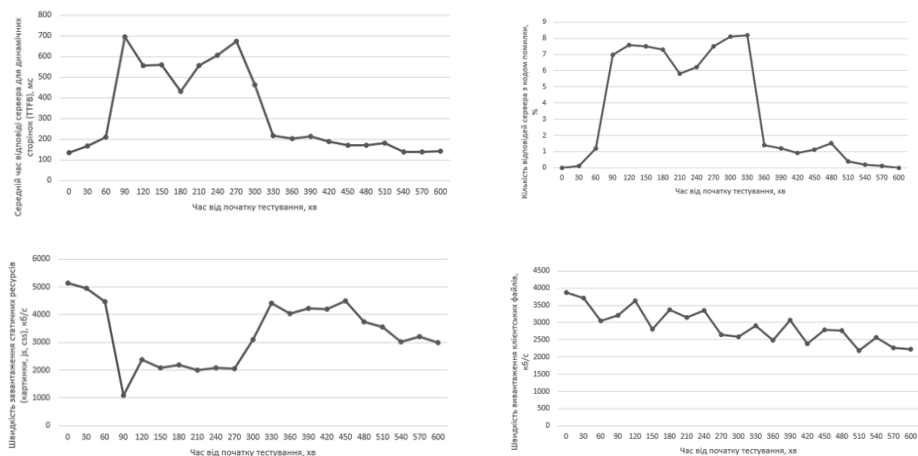


Рисунок 5.3 – Поведінка метрик другого варіанту структури при навантаженні «опублікували посилання на головній сторінці urk.net»

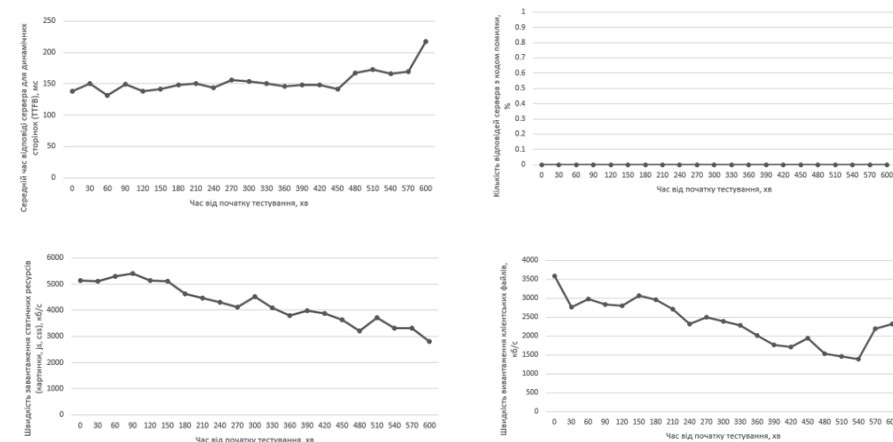


Рисунок 5.4 – Поведінка метрик другого варіанту структури при навантаженні «потрібно виконати план до Нового Року»

					Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print			
					1. Поведінка метрик першого та другого варіанту структури при навантаженні			
Зм.	Арх.	Розробил	Гуров О. В.	Підпис	Дат.	Діп.	Маса	Маса(г)
		Перевірив	Кудиник В. Р.					
		Г. контр.				Арх. 6	Архумін	11
Підготр.		Затвердив	Штефан С. В.			КПІ ім. Ігоря Сікорського ВПІ		
						РПІ-91мп		

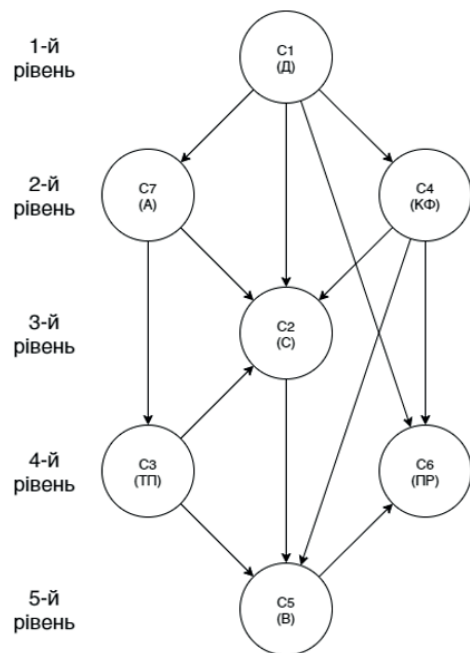


Рисунок 7.1 – Домінантна ієрархічна впорядкована модель критеріїв впливу на вибір базового набору програмного забезпечення для системи безперервної інтеграції:

- c1 – якість та повнота технічної документації (Д);
- c2 – складність впровадження (С);
- c3 – спосіб, канали та якість технічної підтримки зі сторони постачальника (ТП);
- c4 – можливості та складність створення конфігураційних файлів ланцюгів (КФ);
- c5 – вартість впровадження, а також ліцензії чи підписки (В);
- c6 – продуктивність та ефективність роботи, можливості масштабування системи на велику кількість серверів, якщо це не хмарна система (ПР);
- c7 – можливості адміністрування, розподілення ролей, якість та повнота статистичних даних (А).

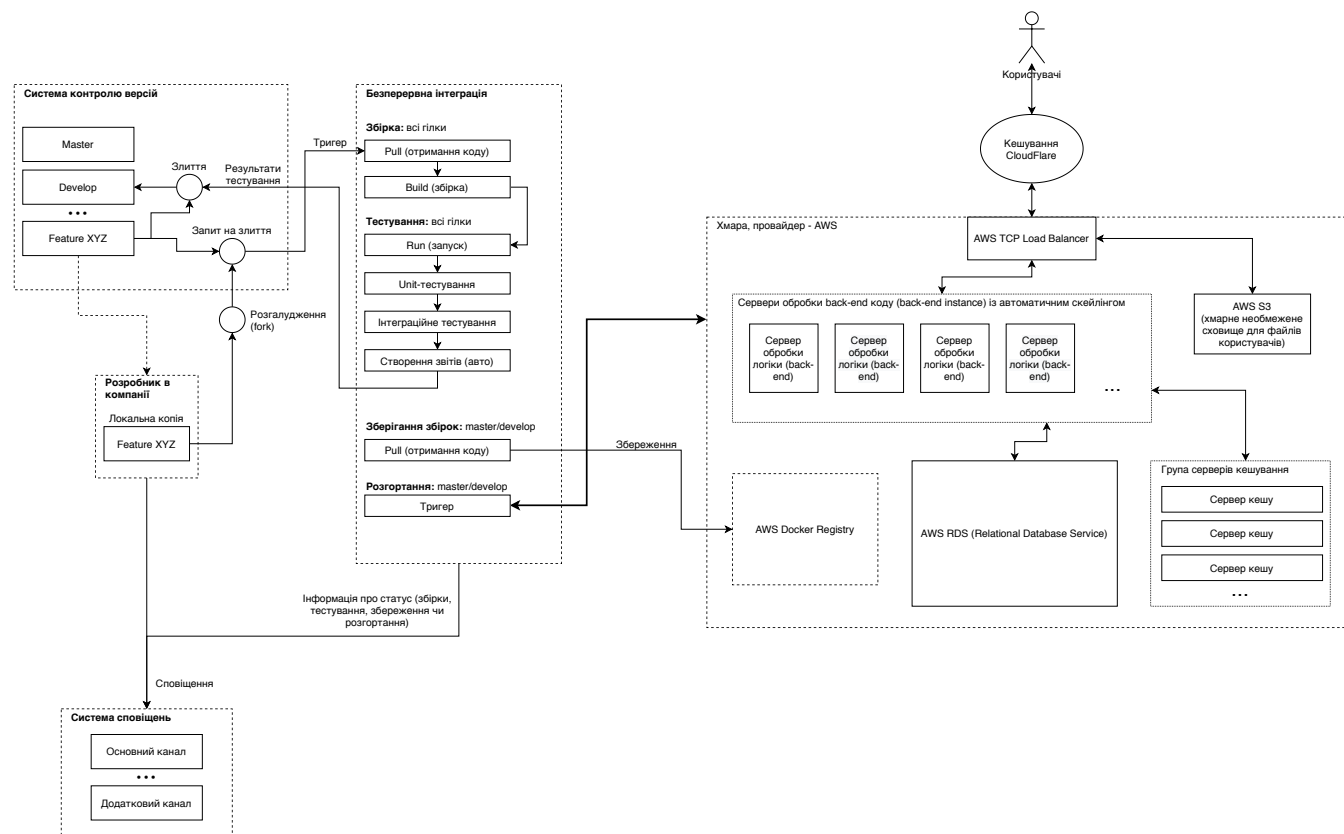


Рисунок 7.2 – Створена архітектура системи безперервної інтеграції для веб-порталів web-to-print

				Центр з розроблення веб-порталів зі створенням і дослідженням архітектури системи безперервної інтеграції для реалізації web-to-print			
Зм.	Арх.	Підпис	Дата	1. Домінантна ієрархія в порівняльній моделі	дн	Маса	Масштаб
Розробник	Гуров О. В.			2. Створена архітектура системи безперервної інтеграції для веб-порталів web-to-print			
Перевірив	Кузнецов В. Р.				Арх. 7	Архуші 11	
Т. контр.							
Ш. контр.							
Вітеран	Штефан С. В.			КТП ім. Ігоря Сікорського ВП		РП-91мп	

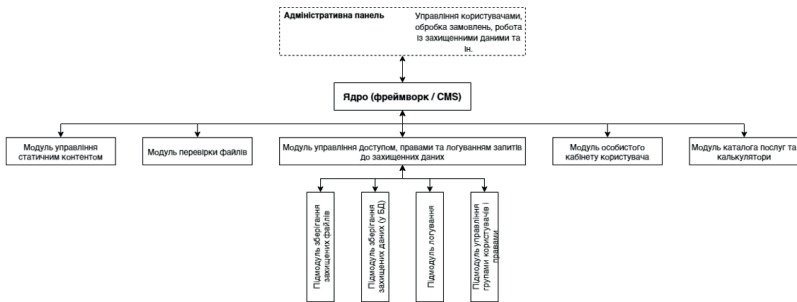


Рисунок 8.1 – Структура модулів типового веб-порталу web-to-print з підвищеним захистом даних користувачів

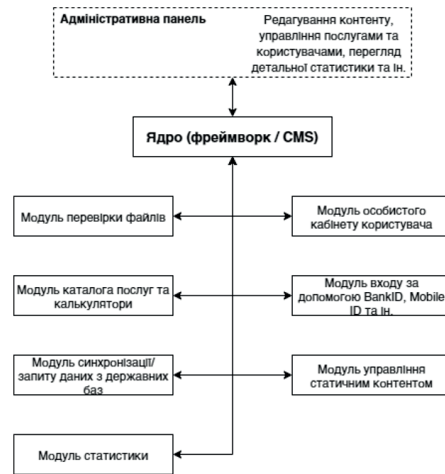


Рисунок 8.2 – Структура модулів типового веб-порталу для використання web-to-print державними структурами



Рисунок 8.3 – Структура модулів типового корпоративного веб-порталу поліграфії з деякими функціями web-to-print

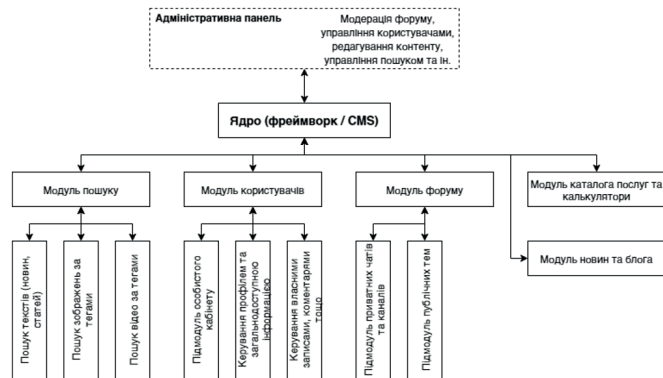


Рисунок 8.4 – Структура модулів типового загального інформаційного веб-порталу з можливостями web-to-print

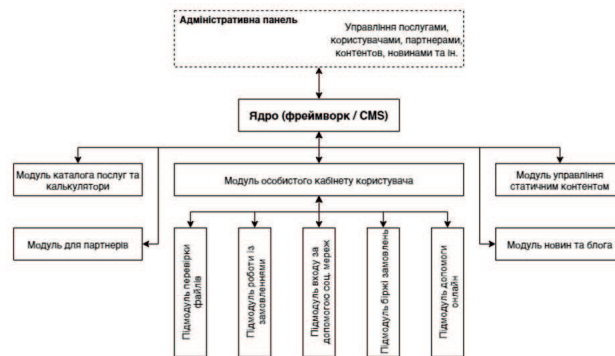


Рисунок 8.5 – Структура модулів типового універсального веб-порталу web-to-print

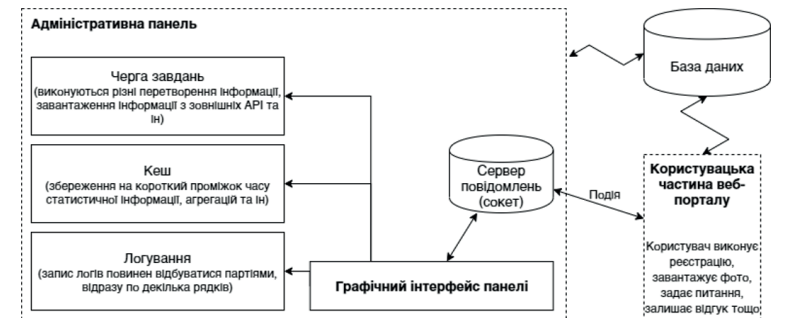


Рисунок 8.6 – Схема роботи панелі для адміністраторів

				Центр з розроблення веб-порталів зі створенням і дослідженнями архітектури системи безсерверної інтеграції для реалізації web-to-print			
Зм.	Арх.	Підпис	Дата	1. Структура модулів типового веб-порталу	Міс.	Маса	Масштаб
Розробник	Гуров О. В.			2. Схема роботи панелі для адміністраторів			
Перевірник	Кудиник В. Р.				Арх. 8	Архуші 11	
Г. контр.							
Ілюстратор				КПІ ім. Ігоря Сікорського ВНП			РПІ-91мп
Інтерпретатор	Штефан С. В.						

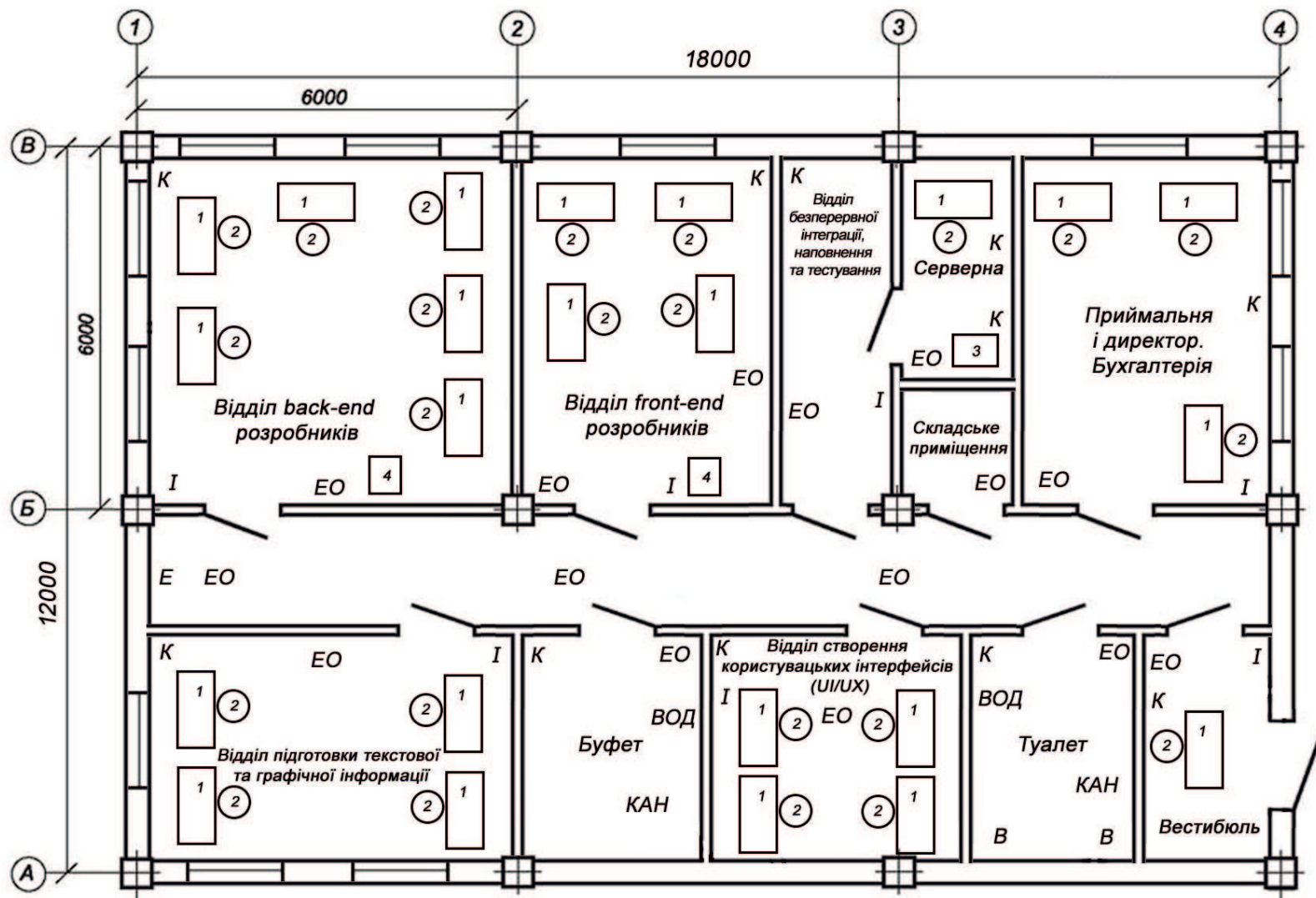


Рисунок 10.1 – Технологічний план центру з розроблення веб-порталів: В – витяжна вентиляційна система, ЕО – електричне освітлення, К – система кондиціонування, Е – електрощитова, ВОД – вода, КАН – загальна міська каналізаційна система, І – локальна мережа

№	Найменування устаткування	Марка устаткування	Кількість	Габарити, м
1	Робоча станція	SL-82	25	1,2x0,6
2	Крісло	Neon HR	25	0,65x0,64
3	Серверна шафа	24U	1	0,8x0,5
4	Шафа	ОЩ-2	2	0,5x0,6
Центр з розроблення веб-порталів зі створенням і дослідженнями архітектури системи безперервної інтеграції для реалізації web-to-print				
Зм.	Арх.	Підпис	Дата	Міс.
Розробник	Гуров О. В.			1:50
Перевірив	Кудинюк Б. Р.			
Г. контр.				Арх. 10 Арх. 11
Ілюстр.				
Затвердив	Штефан С. В.			
КПІ ім. Ігоря Сікорського ВНП				РП-91мп

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ВИДАВНИЧО-ПОЛІГРАФІЧНИЙ ІНСТИТУТ

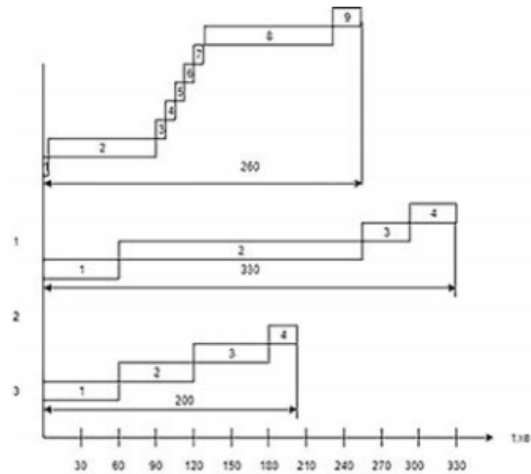
ГО «НАУКОВО-ТЕХНІЧНЕ ОБ'ЄДНАННЯ
ПОЛІГРАФІСТІВ»

ТЕЗИ ДОПОВІДЕЙ

19-Ї МІЖНАРОДНОЇ
НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І АСПІРАНТІВ
«ДРУКАРСТВО МОЛОДЕ»



КИЇВ
2019



Циклограми технологічних процесів створення анімації за допомогою технологій: 1 — HTML5, CSS3; 2 — HTML5, JS; 3 — React, SASS

анімації. Анімація дозволяє сфокусувати увагу користувача на потрібних авторів акцентах, продемонструвати логіку взаємодії, та дозволяє побачити користувачеві результат його дій.

Ефекти анімації для елементів навігації електронного мережевого видання можна реалізувати за допомогою трьох засобів веб-розробки, а саме комбінацій: 1 — HTML5, CSS3; 2 — HTML5, JS; 3 — React, SASS.

Для порівняння цих технологій було побудовано та проаналізовано циклограми технологічних процесів, які представлено на рисунку.

Пояснення до рисунку: 1.1 — Створення базової розмітки; 1.2 — Створення функціональної розмітки; 1.3 — Створення стилю анімації; 1.4 — Задання імені анімації; 1.5 — Задання типу анімації; 1.6 — Задання тривалості; 1.7 — Задання затримки перед початком анімації; 1.8 —



Задання ключових кадрів; 1.9 — Додавання CSS в HTML та збереження; 2.1 — Створення розмітки; 2.2 — Створення стилів та властивостей анімації із застосуванням API веб-анімації; 2.3 — Модифікація стилю сторінки; 2.4 — Оптимізація та збереження; 3.1 — Створення проекту анімації за допомогою React; 3.2 — Створення компонентів розмітки та анімації; 3.3 — Створення стилів та властивостей анімації; 1.4 — Інтегрування та збереження.

Визначено, що для створення ефектів анімації для елементів навігації електронного мережевого видання оптимально використовувати комбінацію React, SCSS, оскільки цей варіант має найменшу кількість операцій та час реалізації анімації.

УДК 004.42;004.05

© **Олександр Гуров**, студент 4-го курсу, ВПІ КПІ ім. Ігоря Сікорського, м. Київ, Україна, 2019 р.
Науковий керівник: Б. Р. Кушлик, к.т.н., доцент, ВПІ КПІ ім. Ігоря Сікорського



ПОРІВНЯЛЬНИЙ АНАЛІЗ ФРЕЙМВОРКІВ ДЛЯ СТВОРЕННЯ ВЕБ-ПОРТАЛУ

Unique web portals are developed using frameworks. This article provides a comparative analysis of the frameworks for their creation.

В наш час для створення веб-порталів у більшості випадків використовують системи управління контентом (CMS) або фреймворки, адже вони «з коробки» містять найбільш необхідні функції та допомагають уникнути виконання повторюваних дій. Якщо необхідно розробити типовий сайт — використовують CMS та інсталиують необхідні плагіни, які реалізують стандартні функції. Для створення унікальних рішень найбільш доцільно застосовувати фреймворки, які виступають у ролі каркаса та мають широкі можливості зміни функціоналу та вигляду.

Сучасні веб-портали використовують різноманітні фреймворки, зокрема: Laravel, CodeIgniter, Symfony, CakePHP, Yii, Zend Framework, Phalcon, FuelPHP, PHPixie, Slim тощо. Але найбільш універсальними, технологічними та популярними є Laravel, Symfony, Yii, їх і порівняємо. Кожен з них має свої переваги та недоліки. Laravel задовольняє потреби як маленьких, так і великих за обсягом кодової бази проектів, а вирішення типових задач здійснюється «в один клік». Незважаючи на простоту, має надзвичайно широкі можливості конфігурації, дозволяє використовувати найбільш сучасні підходи до розробки, підтримує роботу з усіма найбільш популярними базами даних, за умови дотримання структури побудови коду. Стандартні рішення в цьому фреймворці не завжди є оптимальними. Symfony насамперед орієнтований на корпоративний сектор, має надзвичайно широкі можливості, структура визначається розробником, але занадто складний для невеликих проектів, потребує багато знань для доцільного використання його функцій. Yii — це фреймворк із високою швидкістю, підтримує легку зміну тем, проте частина його функцій не документована, знайти вирішення якоїсь проблеми складніше, адже він не такий популярний як попередні.



Порівняння фреймворків проводилось за такими показниками (характеристиками): швидкість роботи, зручність шаблонізації, можливості кастомізації, наявність та різноманітність готових рішень, документація та спільнота. За цими п'ятьма показниками було виконано порівняння з побудовою пелюсткової діаграми, наведеної на рисунку.

Швидкість роботи (K1) у Yii найкраща за результатами багатьох тестів, Laravel трохи повільніший, а ось Symfony значно повільніший через свій широкий функціонал.

Зручність шаблонізації (K2) фреймворка Laravel дійсно надзвичайно висока, він дозволяє використовувати Blade, звичайний php-шаблон або, навіть, свою розмітку шаблонів, Symfony та Yii мають приблизно однакові можливості у своїх стандартних шаблонізаторах.

З можливостями кастомізації (K3) Symfony мало хто зможе зрівнятися, але Laravel — один з таких, Yii ж нав'язує свою структуру та підходи, через що багато розробників мігрувало на Laravel.

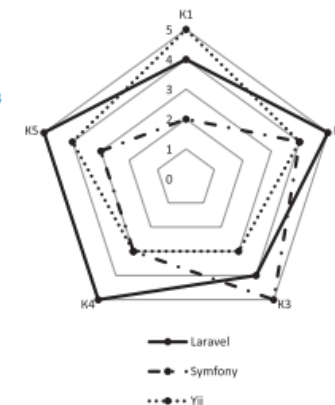
Наявність та різноманітність готових рішень (K4) — це про Laravel (якщо потрібно щось зробити, то скоріш за все це вже зроблено). Symfony та Yii мають приблизно однакову кількість готових рішень.

Документація та спільнота (K5) Laravel не має рівних, адже це найбільш популярний фреймворк за останні два роки. Офіційна документація дуже гарна, все описано з прикладами використання, є Laracasts — відеоуроки та багато іншого. Yii має достатньо недокументованих рішень, а ось Symfony, як фреймворк для великих проектів, має ще меншу спільноту, хоча його документація досить вичерпна та зрозуміла.

Важливо зазначити, що всі фреймворки мають продуману архітектуру, досить легку реалізацію багатомовності, зручну роботу з базами даних за допомогою Doctrine 2 в Symfony, Eloquent ORM в Laravel та Active Record в Yii. Також мають гарну захищеність та можливість збереження логів.



Порівняння фреймворків для створення веб-порталу: K1 — швидкість роботи; K2 — зручність шаблонізації; K3 — можливості кастомізації; K4 — наявність та різноманітність готових рішень; K5 — документація та спільнота



У результаті проведеного аналізу було визначено, що: Symfony найбільш доцільно обирати, якщо розроблюваний веб-портал — це великий корпоративний проект; Yii для проектів, у яких найважливішим показником є швидкість роботи; Laravel — найкращий вибір в більшості випадків, адже забезпечує високу швидкість, зручність розробки, має гарну документацію та спільноту.

УДК 004.422.833+004.915

© **Олександр Сабуров, Володимир Слющинський**,
студенти 4-го курсу, УАД, м. Львів, Україна, 2019 р.
Науковий керівник: Т. В. Нерода, к.т.н., доцент, УАД



ПОРІВНЯННЯ ПРИНЦИПІВ СКЛАДАННЯ НОТНИХ ТЕКСТІВ

Comparative description of construction principles of computing environments to the musical notation set for desktop publisher systems is given.

Розвиток інформаційних технологій зумовлює зростання вимог до якості та скорочення термінів виготовлення замовлень на підприємствах оперативної поліграфії. Особливі труднощі при цьому становлять публікації вищих груп складності, зокрема музично-орієнтовані видання, створення яких необхідно виконувати з використанням функціоналу редакторів (нотаторів), де забезпечено врахування технологічних норм оформлення друкованої продукції.

Вузька спеціалізація таких замовлень й високі норми виробітку часто відбиваються на ціннику, тому при організації автоматизованого робочого місця верстальника необхідно приділити увагу ретельному вибору середовища складання нотних текстів як компонента

комп'ютерно-видавничих систем (КВС), що зменшить вплив людського чинника та забезпечить ефективність й узгодженість процесу підготовки партитурного об'єкта та його інтегрування в оригінал-макет електронної публікації з дотриманням національних стандартів України.

Аналіз відомих нотаторів показав, що відповідно до застосовуваних принципів комп'ютерного складання партитури можна класифікувати як середовища з використанням *псевдографіки*, *мови розмітки* й *об'єктних шаблонів*. Подальші дослідження нотних редакторів (табл.) дозволили дійти висновку, що при розгортанні комп'ютеризованого видавничого комплексу закладу оперативної поліграфії оптимальним з огляду на гнучкість моделювання та редагування музичної символіки, підтримку базових парадигм організації застосунків і т. п. є середовище з алгоритмічно реалізованим складанням об'єктно-орієнтованих компонентів нотних текстів із врахуванням поліграфічних вимог.



Дослідження нотних редакторів

Критерії вибору середовища	Принципи складання нотних записів		
	Псевдо-графіка	Мова розмітки	Об'єктні шаблони
Інтуїтивність інтерфейсу	+/-	+/-	+
Наочність інструментарію	+/-	-	+
Гнучкість складання партитури	+/-	+/-	+
Зручність редагування	+/-	+/-	+
Підтримка WYSIWYG	-	-	+
Підтримка WYSIWYM	-	-	+
Інтегрування у КВС	-	-	+
Забезпечення поліграфічних норм	-	-	+



Іваськів Роман. Алгоритм оцифрування бібліотечних фондів для комп'ютеризованої інформаційної бібліотечної системи	49
Попель Ярослав. Розроблення інтерактивного каталогу друкарського устаткування	51
Каневський Богдан. Класифікація анімаційних ефектів для відео	53
Калиновський Кирило. Принципи гештальту у створенні відео	56
Марчук Іванна. Класифікація мультимедійних стандартів стиснення відеоінформації	58
Трофименко Вадим. Математична реалізація чат-ботів	60
Остапенко Єкатерина. Дослідження доступності інформації на різних пошукових сервісах для людей з обмеженими можливостями	61
Дякуненко Данііл. Вивчення можливостей оптимізації UV розгортки для подальшого текстурування у програмі «Maya»	63
Мазурчак Володимир. Сучасні засоби тестування веб-сайтів	64
Русаков Кирило. Визначення проблемних місць сайту та методи їх вирішення	67
Воробей Віталій. Систематизація метрик вимірювання контенту веб-сторінок	68
Діденко Марія. Оцінка якості користувацького інтерфейсу	71
Кислова Катерина. Систематизація програмних засобів для створення GIF-анімації	72
Ворожбит Вероніка. Вивчення можливостей трепінгу для реалізації маскування дефектів у додрукарській підготовці	74
Сабуров Олександр. Вибір графічного класу при опрацюванні музичної нотації для комп'ютерно-видавничих систем	75
Слющинський Володимир. Структурування музичної нотації при проектуванні цільового інструментарію спеціалізованого редактора	76
Сущенко Анастасія. Огляд технологій автоматизації розробки корпоративного стилю	78

Ратушна Анжела. Класифікація мережових ЗМІ	79
Підгурська Марія. Розробка темної теми для мобільних додатків UX	82
Дехтяр Оксана. Встановлення пріоритетних параметрів мобільного додатка з доповненою реальністю	83
Гуров Олександр. Особливості та переваги використання систем безперервної інтеграції для web-to-print	84
Ярова Тетяна. Аналіз стану електронних дошкільних дитячих видань	87
Рубан Ірина. Доповнена реальність (AR) у контексті книговидавання	89
Бардовський Богдан. Вибір способу розробки мультимедійного додатку	90
Мельниченко Світлана. Класифікація інфографіки для дітей	93
Минчик Юлія. Сравнительный анализ электронных мультимедийных обучающих изданий	96
Кушнір Катерина. Продуктивність процесів створення анімації для мультимедійних додатків	97
Горбачова Катерина. Чинники, що впливають на якість візуальних новел	100
Станиславчик Марія. Роль белорусских изданий с дополненной реальностью в формировании национального самосознания детей	103
Клименко Юлія. Деталізація технологічного процесу створення елементів доповненої реальності у вигляді масок	105
Яцковська Вікторія. Дослідження інструментарію цільового програмного забезпечення нейл-принтерів	107
Луніна Катерина. Використання доповненої реальності у дитячій літературі	108
Журавльова Катерина. Електронні додатки, як елемент творчого розвитку дитини	109
Пономаренко Анна. Методи реалізації анімації у сучасному веб-дизайні	111
Тараненко Олена. Особливості накладення текстур у 3D-моделюванні	112



**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

ВИДАВНИЧО-ПОЛІГРАФІЧНИЙ ІНСТИТУТ

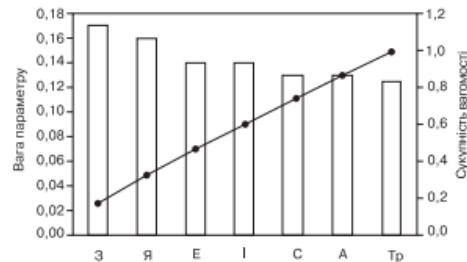
**ГО «НАУКОВО-ТЕХНІЧНЕ ОБ'ЄДНАННЯ
ПОЛІГРАФІСТІВ»**

ТЕЗИ ДОПОВІДЕЙ

**20-Ї МІЖНАРОДНОЇ
НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І АСПІРАНТІВ
«ДРУКАРСТВО МОЛОДЕ»**



**КИЇВ
2020**



Діаграма Парето для оцінки пріоритетних параметрів додатка з технологією доповненої реальності: зручність використання (З), якість 3D моделі (Я), економічність технологічного процесу (Е), інтерактивність (І), стабільність роботи (С), адаптивність (А), трудомісткість виконання (Тр)



моделі (Я), економічність технологічного процесу (Е), інтерактивність (І). Менш важливими слід вважати: стабільність роботи (С), адаптивність (А), трудомісткість виконання (Тр). Отож, для досягнення більш якісного рівня проектування та розробки додатка з технологією доповненої реальності, слід більше уваги звертати на встановлені пріоритетні параметри, що дозволить суттєво скоротити витрати часових та матеріальних ресурсів.

УДК 004.42;004.05

© **Олександр Гуров**, магістрант, ВПІ КПІ ім. Ігоря Сікорського, м. Київ, Україна, 2020 р.
Науковий керівник: Б. Р. Кушлик, канд. техн. наук, доц., ВПІ КПІ ім. Ігоря Сікорського

ОСОБЛИВОСТІ ТА ПЕРЕВАГИ ВИКОРИСТАННЯ СИСТЕМ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ДЛЯ WEB-TO-PRINT

Modern business requires the rapid implementation of functions in web-to-print portals. This article provides an overview of CI/CD solutions, the benefits of use and implementation.

У сучасній економіці використання цифрових інструментів у бізнес-рішеннях грає визначальну роль. Разом зі зростанням складності високотехнологічних платформ, важливим чинником стає безперебійність роботи та постійні оновлення з метою дотримання безпекових заходів критично важливих IT-систем. Торкнулася ця тенденція й розробки програмного забезпечення для порталів типу web-to-print.

При отриманні замовлень онлайн, клієнти завантажують великий обсяг даних у вигляді файлів оригінал-макетів. Система зберігання та контролю версій таких файлів потребує чимало серверних ресурсів, а також не допускається наявність помилок в ядрі програмного забезпечення. У даному випадку CI/CD дає швидкий цикл зворотного зв'язку, що дозволяє майже миттєво визначити, наскільки якісними є зміни у коді, а також перевірити функціональність продукту). При waterfall-підході (каскадна розробка програмного забезпечення, де розробники незалежно працюють над різними частинами системи, стадія інтеграції є заключною) можна швидко вносити зміни у код, але без постійних перевірок, бо виявлення помилок потребує набагато більше часу й може статися вже після початку комерційної експлуатації, причому таких «відкладених» помилок в одному релізі може виявитися декілька.

Одні помилки у програмному забезпеченні можуть містити інші, які включають у себе треті тощо. Чим більше помилок накопичується, тим складніше тестувати й знаходити їх, що у результаті може привести до неприємних наслідків. Тим часом у CI/CD-розробці автоматизовані тести у разі неуспішного їх проходження покажуть, що саме потрібно виправити. Для впровадження системи потрібен час, але це допоможе розробляти програмне забезпечення максимально швидко й зручно.

Автоматизовані процеси допомагають значно знизити трудовитрати різних відділів підприємства. Без автоматизації CI/CD можуть ставатися помилки, викликані людським фактором й необхідністю здійснення ручних операцій.



При аналізі досліджень та опитувань компаній, що впровадили CI/CD, було виявлено основні переваги та покращення процесів з точки зору цих компаній, які зображено на рисунку 1.

Отже, на підставі багатьох проаналізованих джерел, встановимо та узагальнимо основні переваги використання CI/CD для web-to-print:

1. Швидкість введення нової функціональності від ідеї до запуску в експлуатацію. Безперервна інтеграція дозволяє запускати оновлення за лічені дні або тижні порівняно з цілим календарним роком при класичному waterfall-підході. Нові сервіси — нові конкурентні бізнес-переваги. З'являється можливість не просто відтворювати функціональність рішень конкурентів, але й значно випереджати їх у розробці та впровадженні нових інструментів.

2. Можливість вибору оптимального варіанту за рахунок оперативного тестування й більшого числа ітерацій. Відмовившись від роботи над безперспективними рішеннями, можна досягти значного заощадження ресурсів компанії.

3. Якість кінцевого результату вище за рахунок можливості впровадження автоматичного тестування, яке охоплює всі аспекти продукту, що важко буде реалізувати при стандартному релізному підході. Всі помилки й тонкі місця виявляються та видаляються ще на ранніх етапах розробки.

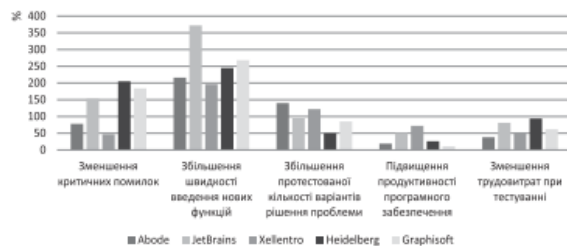


Рис. 1. Результати компаній після впровадження CI/CD

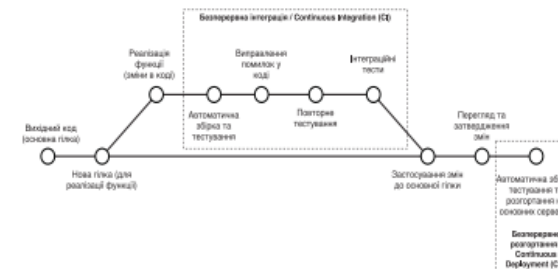


Рис. 2. Типовий процес CI/CD для web-to-print

На рис. 2 наведено типовий процес CI/CD для веб-порталів типу web-to-print.

Отже, у результаті аналізу особливостей та переваг запровадження й використання систем безперервної інтеграції для порталів типу web-to-print, можна зробити висновок, що: системи безперервної інтеграції CI/CD є потужним інструментом, який допомагає суттєво покращити робочий процес розробки програмного забезпечення із підвищенням його надійності та з можливістю внесення коректив та застосування ітераційного принципу розробки із скороченням циклу одиничних спринтів (одна ітерація життєвого циклу розробки програмного забезпечення).



УДК 004.422.8

© **Тетяна Ярова**, магістрантка, ВПІ КПІ ім. Ігоря Сікорського, м. Київ, Україна, 2020 р.

Науковий керівник: Т. А. Роїк, д-р техн. наук, проф., ВПІ КПІ ім. Ігоря Сікорського

АНАЛІЗ СТАНУ ЕЛЕКТРОННИХ ДОШКІЛЬНИХ ДИТЯЧИХ ВИДАНЬ

The state of children's electronic publications for pre-school age has been analyzed, and their classification has been developed.



Попель Ярослав. Використання бібліотеки Kordova.js при проектуванні онлайн-компонентів видавничих комплексів	66
Гнатенко Олександр. Систематизація методів створення фізичних моделей навчальних мультимедійних видань	68
Іванов Віталій. Дослідження методів та моделей лазерної діагностики режимів вібрації в багатосекційних машинах	69
Бабич Олександра, Остапчук Анатолій, Гурей Дмитро. Застосування технології інтернету речей для впровадження smart-систем в друкарському цеху	71
Оратовська Олена. Розробка і моделювання САР натягу стрічки PPM на основі нейронної мережі	73
Каневський Богдан. Систематизація програмних засобів створення візуального роману	75
Трофименко Вадим. Рекомендації щодо підвищення ефективності використання додатку Adobe After Effects	77
Воробей Віталій. Сучасні технології створення анімації для електронних видань у форматі EPUB	78
Драз Діана Марван. Використання Google Map при створенні сайтів	81
Мазурчак Володимир. Методика синтаксичного аналізу даних для наповнення інформаційних веб-порталів	82
Трусова Тая. Дослідження критеріїв оцінки UX	85
Марчук Іванна. Важливість розкадрування в медіавиробництві	86

Блиновська Катерина. Класифікація веб-сайтів	89
Кислова Катерина. Вибір технологічних процесів створення gif-анімації	91
Дехтяр Оксана. Розробка доповненої реальності для навчального мобільного додатку	92
Метліна Катерина. Алгоритм створення анімованого ролику	94
Поліщук Юлія. Розробка анімаційних ефектів елементів навігації електронного мережевого видання	97
Гуров Олександр. Порівняльний аналіз фреймворків для створення веб-порталу	99
Сабуров Олександр, Слющинський Володимир. Порівняння принципів складання нотних текстів	102
Орленко Кристина. Технологія передачі кольору в смартфонах	104
Горбачова Катерина. Аналітичний огляд програмних засобів для створення візуальних новел	105



III. Устаткування

Тришук Руслан. Комплексна технологія зміцнення деталей вузлів лінії для виготовлення інтегральних обкладинок	108
Тиндик Роман. Проект модернізації паркувального вузла у струминних широкоформатних плотерах	109
Ткаченко Владислав. Дослідження конструкції пресу термодрукарської машини	111
Бойко Владислав. Дослідження механізму подавання задрукованого матеріалу цифрової друкарської машини Mimaki JV3-160SP	112