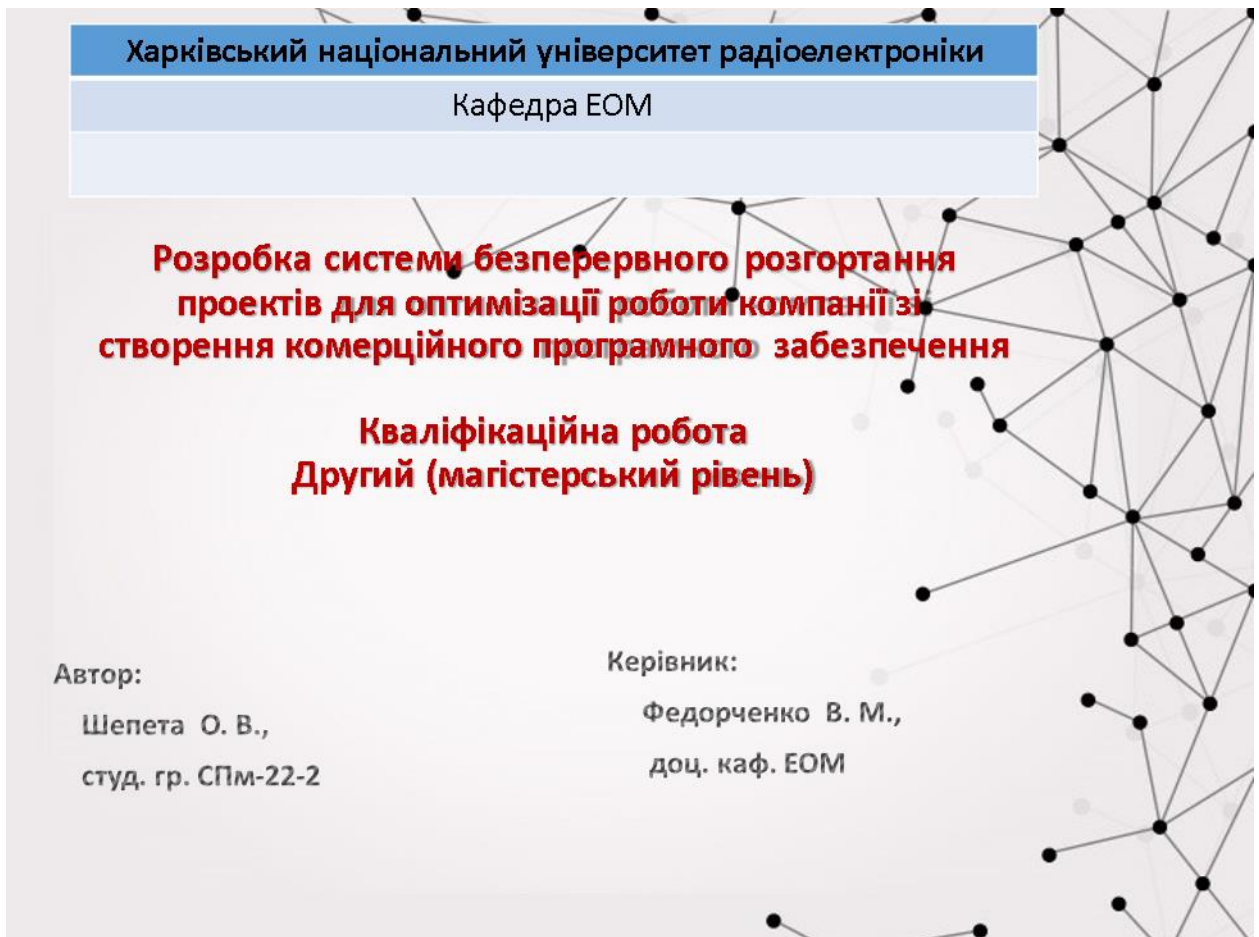


ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



Харківський національний університет радіоелектроніки

Кафедра ЕОМ

Розробка системи безперервного розгортання проектів для оптимізації роботи компанії зі створення комерційного програмного забезпечення

**Кваліфікаційна робота
Другий (магістерський рівень)**

Автор:
Шепета О. В.,
студ. гр. СПм-22-2

Керівник:
Федорченко В. М.,
доц. каф. ЕОМ

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ:

2

- **Метою роботи** є підвищення конкурентоспроможності ІТ-компанії зі створення комерційного програмного забезпечення на основі системи безперервного розгортання проектів.
- **Об'єктом дослідження** є процес підвищення конкурентоспроможності ІТ-компанії на основі збільшення продуктивності команди програмістів за рахунок впровадження системи безперервного розгортання проектів.
- **Предмет дослідження** – економічний механізм формування конкурентоспроможності та вплив впровадження системи безперервного розгортання проектів на показники ефективності

ПОНЯТТЯ БЕЗПЕРЕРВНОГО РОЗГОРТАННЯ

3

- **Безперервна інтеграція** – це практика розробки комерційного програмного забезпечення, що полягає у автоматизованому інтегруванні програмних модулів в єдину систему з використанням програмних тестів для уникнення помилок.



АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ СИСТЕМ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ПРОЄКТІВ

4

- **Скорочення ручних операцій** – етапи створення, збирання і тестування програмного забезпечення проводяться в автоматичному режимі;
- **Наявність робочої ІТ-системи** протягом всього процесу розробки – у проєктної команди завжди є свіжа версія рішення для демонстрації замовнику, отримання зворотного зв'язку і швидкого доопрацювання;
- **Якість програмного забезпечення** – в рамках Continuous Integration використовуються різні програмні засоби для контролю якості коду, що дозволяє скоротити кількість помилок;
- **Мінімізація ризиків** – дефекти виявляються на ранніх стадіях розробки програмного забезпечення, що допомагає уникнути збільшення термінів і вартості проєкту;
- **Потенційна необхідність** у спеціальному навчальному сервері для потреб безперервної інтеграції;
- **Проблеми інтегрування** разі використання системи контролю версіями вихідного коду з підтримкою розгалуження.

ПОНЯТТЯ КОНКУРЕНТОСПРОМОЖНОСТІ ПІДПРИЄМСТВА

5

- **Конкурентоспроможність підприємства** – це здатність певних об'єктів чи суб'єктів підприємства перевершити конкурента у заданих умовах.



МЕТОД ОЦІНКИ КОНКУРЕНТОСПРОМОЖНОСТІ

6



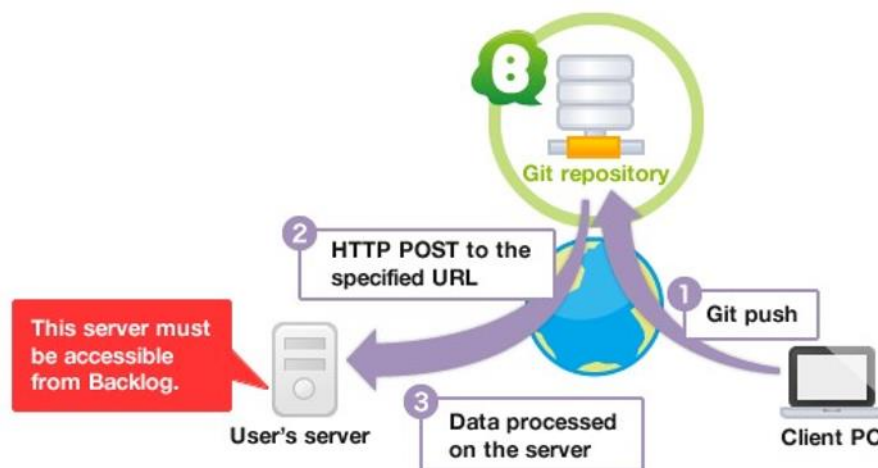
ОЦІНКА ЦІННІСНИХ ДІЙ ДО ВПРОВАДЖЕННЯ СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ

Загальні висновки по ціннісним блокам

Вхідна логістика	3,5
Виробництво	3,4
Вихідна логістика	4,1
Маркетинг та продажі	3,3
Обслуговування	3,6
Загальний середній бал у основній діяльності підприємства	3,6
Розвиток технології	2,7
Управління персоналом	3,2
Інфраструктура	3,1
Загальний середній бал у допоміжній діяльності підприємства	3
Загальна оцінка конкурентоспроможності підприємства:	3,3

ВПРОВАДЖЕННЯ СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ПРОЄКТІВ

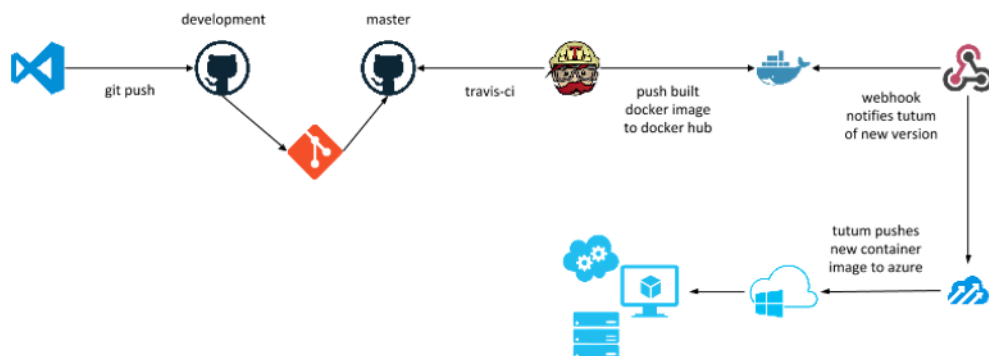
● **Webhooks** дозволяють будувати або налаштувати інтеграцію, яка виконується на певні події в системі контролю версій. Коли одна з цих подій спрацює, система контролю версій запускає скрипт інтерпретації або компіляції проєкту.



ВПРОВАДЖЕННЯ СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ПРОЄКТІВ

9

- Використання **Webhooks** для системи безперервного розгортання. На виділеному інтеграційному сервері організовується служба, до завдань якої належать: отримання вихідного коду зі сховища; збірка проєкту; виконання тестів; розгортання готового проєкту; формування та відправка звітів



ОГЛЯД СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ПРОЄКТІВ GITLAB

10

GitLab – це веб-додаток для хостингу вихідного коду проєктів, заснований на системі контролю версій Git. Своїм функціоналом GitLab дуже нагадує GitHub, проте заточений під командну роботу, в той час як GitHub віддає перевагу індивідуальній роботі.

GitLab існує як у вигляді **SAAS** веб-сайту з відкритою реєстрацією, так і в якості індивідуального рішення GitLab Community Edition, яке можна встановити на свій сервер і налаштувати під власні потреби.



GitLab

РЕЗУЛЬТАТИ ТА АНАЛІЗ ВПРОВАДЖЕННЯ СИСТЕМИ БЕЗПЕРЕРВНОГО РОЗГОРТАННЯ НА ПОКАЗНИКИ КОНКУРЕНТОСПРОМОЖНОСТІ ¹¹

Вхідна логістика	3,5	3,5
Виробництво	3,4	4,2
Вихідна логістика	4,1	4,1
Маркетинг та продажі	3,3	3,3
Обслуговування	3,6	3,6
Загальний середній бал у основній діяльності підприємства	3,6	3,6
Розвиток технології	2,7	3,7
Управління персоналом	3,2	3,2
Інфраструктура	3,1	3,1
Загальний середній бал у допоміжній діяльності підприємства	3	3
Загальна оцінка конкурентоспроможності підприємства:	3,3	3,53

ВИСНОВКИ

12

- Проведена оцінка конкурентоспроможності підприємства до впровадження системи безперервної інтеграції проектів;
- Впроваджено систему безперервної інтеграції проектів шляхом використання систем GitLab та GitLab CI ;
- Проведений аналіз впливу впровадження системи безперервної інтеграції на показники конкурентоспроможності підприємства.



ПЕРЕЛІК ПУБЛІКАЦІЙ ЗА ТЕМОЮ РОБОТИ

13

Вимоги до якості програмних систем [Текст] :
матеріали одинадцятої міжнародної
науково-технічної конференції.,
16 – 17 листопада 2023 року.
– Баку – Харків – Бельсько-Бяла. – 52 с.

ДОДАТОК Б

Налаштування системи контролю версій

```

<?php
/**
 * GitLab Web Hook
 *
 * This script should be placed within the web root of your desired deploy
 * location. The GitLab repository should then be configured to call it for the
 * "Push events" trigger via the Web Hooks settings page.
 *
 * Each time this script is called, it executes a hook shell script and logs all
 * output to the log file.
 *
 * This hook uses php's exec() function, so make sure it can be executed.
 * See http://php.net/manual/function.exec.php for more info
 */
// CONFIGURATION
// =====
/* Hook script location. The hook script is a simple shell script that executes
 * the actual git push. Make sure the script is either outside the web root or
 * inaccessible from the web
 *
 * This setting is REQUIRED
 */
$hookfile = '.hooks/gitlab-webhook-push.sh';
/* Log file location. Log file has both this script's and shell script's output.
 * Make sure PHP can write to the location of the log file, otherwise no log
 * will be created!
 *
 * This setting is REQUIRED
 */
$logfile = '.hooks/gitlab-webhook-push.log';
/* Hook password. If set, this password should be passed as a GET parameter to
 * this script on every call, otherwise the hook won't be executed.
 *
 * This setting is RECOMMENDED
 */
// $password = 'a_strong_password';
/* Ref name. This limits the hook to only execute the shell script if a push
 * event was generated for a certain ref (most commonly - a master branch).
 *
 * Can also be an array of refs:
 *
 * $ref = array('refs/heads/master', 'refs/heads/develop');
 *
 * This setting does not support the actual refspec, so the refs should match
 * exactly.
 *
 * See http://git-scm.com/book/en/Git-Internals-The-Refspec for more info on
 * the subject of Refspec
 */

```

```

* This setting is OPTIONAL
*/
$ref = 'refs/heads/master';
// THE ACTUAL SCRIPT
function log_append($message, $time = null)
{
    global $logfile;
    $time = $time === null ? time() : $time;
    $date = date('Y-m-d H:i:s');
    $pre = $date . ' (' . $_SERVER['REMOTE_ADDR'] . '): ';
    file_put_contents($logfile, $pre . $message . "\n", FILE_APPEND);
}
function exec_command($command)
{
    $output = array();
    exec($command, $output);
    foreach ($output as $line) {
        log_append('SHELL: ' . $line);
    }
}
if (isset($password))
{
    if (empty($_REQUEST['p'])) {
        log_append('Missing hook password');
        die();
    }
    if ($_REQUEST['p'] !== $password) {
        log_append('Invalid hook password');
        die();
    }
}
// GitLab sends the json as raw post data
$input = file_get_contents("php://input");
$json = json_decode($input);
if (!is_object($json) || empty($json->ref)) {
    log_append('Invalid push event data');
    die();
}
if (isset($ref))
{
    $_refs = (array) $ref;
    if ($ref !== '*' && !in_array($json->ref, $_refs)) {
        log_append('Ignoring ref ' . $json->ref);
        die();
    }
}
log_append('Launching shell hook script...');
exec_command('sh ' . $hookfile);
log_append('Shell hook script finished');

```