

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ПЕТРА МОГИЛИ

**Померанцева Марія Андріївна**

УДК 004.925.5

**СИСТЕМА QA ДЛЯ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ  
ПРОГРАМНИХ ЗАСТОСУНКІВ**

Спеціальність 123 – Комп'ютерна інженерія

Автореферат  
магістерської роботи  
на здобуття кваліфікації магістра з комп'ютерної інженерії

Миколаїв – 2020

Робота виконана у Чорноморському національному університеті ім. Петра Могили.

- Керівник:** кандидат технічних наук, доцент  
**Ірина Олександрівна Калініна,**  
ЧНУ ім. Петра Могили,  
доцент кафедри інтелектуальних інформаційних систем
- Рецензент:** д-р педагогічних наук, професор  
**Мещанінов Олександр Павлович,**  
ЧНУ ім. Петра Могили,  
професор кафедри інтелектуальних  
інформаційних систем
- Консультант:** д-р біол. наук, професор  
**Григор'єва Людмила Іванівна,**  
ЧНУ ім. Петра Могили,  
завідувач кафедри екології Медичного інституту

Захист відбудеться « 26 » лютого 2020 р. о 9<sup>00</sup> на засіданні  
Державної екзаменаційної комісії в ЧНУ ім. Петра Могили, ауд. 2-406

З магістерською роботою можна ознайомитись на сайті ЧНУ ім. Петра Могили  
за посиланням <http://chmnu.edu.ua>

Автореферат оприлюднений « 24 » лютого 2020 р.

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

**Актуальність теми.** У сучасному світі інформаційні системи використовуються всюди, багато коштів витрачається на розробку програмного і апаратного забезпечення, очікується, що продукт або система будуть якісними та виконувати поставлені задачі. Але навіть достатній бюджет не може гарантувати якості програмного забезпечення. Адже навіть найкращі розробники допускають помилки у системі. Ручна перевірка якості системи, її компонентів, взаємодії додатків займає багато часу та не є ефективною при багаторазовому повторенню однакових дій чи процесів, які необхідно тестувати періодично. Автоматизоване тестування програмних, апаратних додатків та їх сумісності зменшує витрати на розробку, оновлення системи за допомогою швидкості виконання, можливості повторного використання, охоплення різноманітних кейсів, високої точності без людської похибки.

**Мета:** підвищити якість та стійкість програмних застосунків у системі, шляхом розробки автоматизованої системи QA..

Для досягнення мети в бакалаврській роботі поставлені та вирішені наступні **задачі:**

- аналіз існуючих типів та методів тестування, особливостей автоматизованого тестування;
- дослідження існуючих методів тестування програмних застосунків;
- аналіз апаратних складових для проведення тестування;
- проектування та розроблення автоматизованої системи QA, з урахуванням загальних методів тестування, операційних систем та архітектур комп'ютерів.

**Об'єкт:** процес перевірки програмних застосунків.

**Предмет:** автоматизована перевірка програмних застосунків.

**Використані методи:**

методи емпіричного дослідження:

1. Опис існуючих методів автоматизованого тестування, методик ручного тестування.

2. Вимірювання ефективності комп'ютерної програми при стандартній та модернізованій системі автоматизації перевірки програмних застосунків.

3. Порівняння ефективності комп'ютерної програми при стандартній та модернізованій системі автоматизації перевірки програмних застосунків.

Методи проміжного рівня дослідження:

1. Аналіз існуючих методів автоматизованого тестування, методик ручного тестування.

2. Аналіз ефективності комп'ютерної програми при стандартній та модернізованій системі автоматизації перевірки програмних застосунків.

3. Моделювання модернізованої системи автоматизованої перевірки програмних застосунків.

Методи теоретичного рівня дослідження:

1. Методи автоматизованого тестування, ручного тестування.

2. Методи вимірювання якості програмних застосунків.

3. Удосконалення стандартних методів автоматизації перевірки якості програмних застосунків.

**Практичне значення одержаних результатів:** результати роботи використані у поточній діяльності підприємства «Hosting Max» для автоматизованої перевірки якості проекту «XpressDocx».

**Апробація результатів** бакалаврської роботи відбулася під час:

- XXII Всеукраїнської науково-практичної конференції «Могилянські читання-2019» (Миколаїв, 2019).

**Структура та обсяг роботи.** Магістерська робота складається з анотації на 2 сторінках, вступу, трьох розділів, висновків, переліку джерел посилання з 30 найменувань, 2 додатків на 20 сторінках. Основна частина роботи становить 70 сторінки, серед яких \_\_ рис. та \_ табл.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** подано обґрунтування актуальності теми бакалаврської роботи, зазначено її зв'язок із науковою програмою, планами і темами, сформульовано мету та завдання дослідження, вказано практичне значення одержаних результатів, наведено відомості про апробацію результатів роботи та публікації автора. Задача побудови автоматизованих систем тестування програмних застосунків набуває своєї актуальності разом зі збільшенням числа програмних та апаратних додатків.

У **першому розділі** бакалаврської роботи «**Аналітичний огляд видів тестування, особливості автоматизованого тестування, архітектур**» проведено огляд теоретичні основи тестування, продемонстровані основні класифікації тестування, були наведені найбільш поширені методології тестування програмного забезпечення, розглянуто архітектури комп'ютерів для розуміння взаємодії впливу програмного застосунку на операційну систему та ядро процесора, розглянуто емулятори, як засоби заміни апаратної частини при тестуванні.

Аналіз проводився у декілька етапів. На початковому етапі був проведений аналітичний огляд видів тестування. Отримана інформація відображає, що для проведення тестування програмно-апаратних компонентів необхідно використовувати методи таких типів тестування, як: конфігураційне, тестування встановлення, переносимості. Методи тестування для програмних застосунків відрізняються відповідно до сфери застосування (web-застосунки, мобільні застосунки, ін.), На наступному етапі проведено аналіз особливостей автоматизованого тестування. Виявлено, що автоматизоване тестування є доцільним для тестування, яке кероване даними, повторюваних тестів, тестів, які не потребують експертної оцінки. На заключному етапі визначено які архітектурні рішення використовуються для сучасних комп'ютерів.

Дослідження проводилось за допомогою порівняльного аналізу, зокрема функціональності, актуальності інформації. Отримана інформація відображає недоліки та переваги кожного етапу аналізу.

У другому розділі магістерської роботи «Апаратні та інструментні рішення» було проведено дослідження апаратної складової та інструментів проекту. Проаналізовані основні технічні характеристики процесорів комп'ютерів, визначено кращі процесори для роботи з багатопотоковістю в системі, виявлено, що найкращими рішеннями процесорів для паралельного програмування є процесори AMD виробництва, а саме: AMD Ryzen 9 3900X і AMD Ryzen 5 3600 (Рисунок 1).



Рисунок 1 – Обрані процесори: а – AMD Ryzen 9 3900X; б – AMD Ryzen 5 3600.

Досліджено використання операційних систем, результати наведені у Таблиці 1, визначено і описано інструменти, які краще застосовувати для системи тестування.

Таблиця 1 – Провідні ОС за 2020 рік

|         |        |
|---------|--------|
| Windows | 35.23% |
| Android | 39.67% |

|       |       |
|-------|-------|
| IOS   | 14.5% |
| OS X  | 7.71% |
| Linux | 0.88% |
| Інші  | 2.01% |

Проведено опис інструментів для проведення автоматизованого тестування програмних додатків.

У **третьому розділі** бакалаврської роботи «**Розробка програмної частини**» описано процес реалізації автоматизованої системи тестування для web – застосунку. Для системи використані наступні інструменти: EclipseIDE, Java, фреймворк Cucumber, Git, бібліотека Selenium, Junit, Jenkins та деякі інші бібліотеки. Для паралельного запуску тестів реалізовано багатопотоковість.

Життєвий цикл виконання одного окремого тесту – наступний:

1. Встановлення вказівника на необхідний тест.
2. Запуск WebDriver для можливості доступу до браузеру.
3. Зчитування кроку Gherkin сценарію.
4. Зв'язування Gherkin сценаріїв зі програмною реалізацією.
5. Виконання алгоритму проходження тесту (включаючи пошук необхідних елементів, виконання дії, порівняння очікуваного та фактичного результатів, логування).
6. Завершення тесту.

До зовнішніх програмних компонентів даної системи потрібно віднести: використані програмні бібліотеки, базу даних, інструмент для створення процесу неперервної інтеграції Jenkins, Git репозиторій, web-драйвер.

Для реалізації підключення бази даних у проєкті розроблено клас ReadFromDatabase, який надає можливість тестувальникам запускати db запити на рівні gherkin. Підключення до бази даних відбувається за стандартною процедурою з використанням класу бібліотеки. Для считування даних із бази

необхідно вказати середовище (тестовий сайт), посилання на базу даних, вказати ім'я користувача авторизованого до бази даних та його пароль.

Схема взаємодії WebDriver з тестовою середою зображено на Рисунку 2.

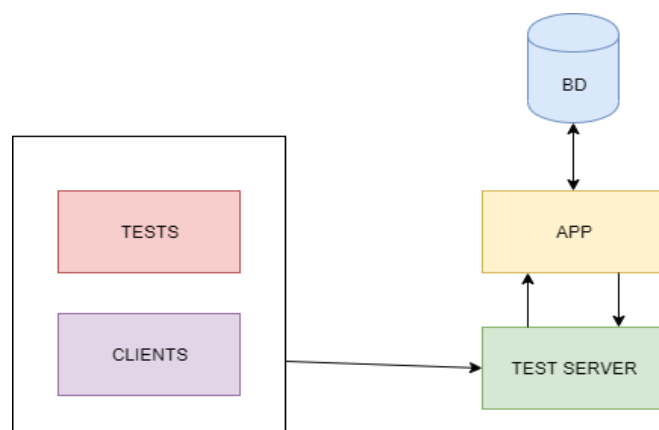


Рисунок 3.7 – Взаємодія драйверу з тестовою середою

Для забезпечення неперервного процесу деплою нового коду в проєкті використовується Jenkins. Схема взаємодії Jenkins з проєктом зображено на Рисунку 3.

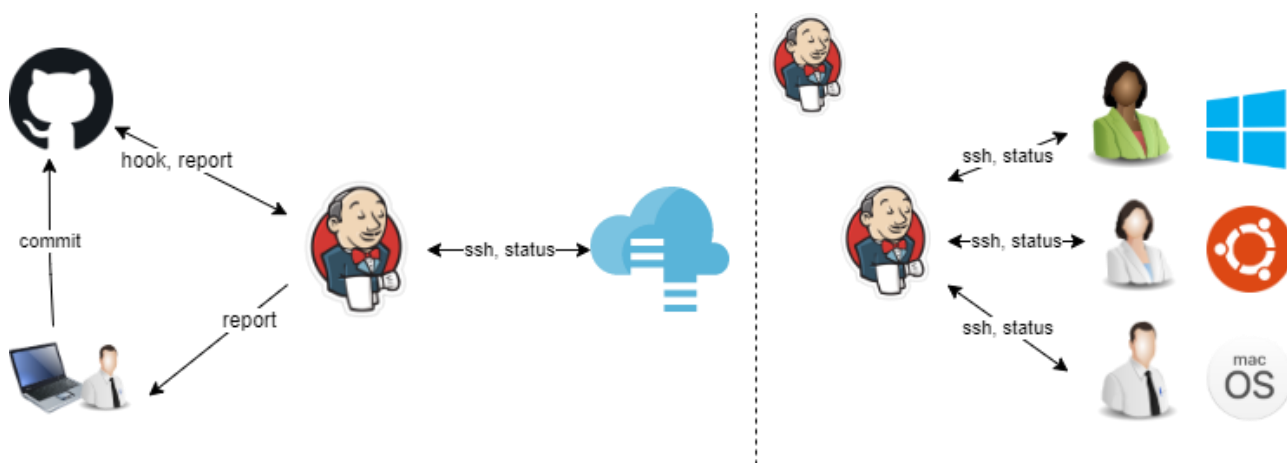


Рисунок 3 – Взаємодія Jenkins з проєктом

Додатки містять лістинг коду застосунку.

У спеціальній частині «Охорона праці та безпека у надзвичайних ситуаціях» наведено аналіз факторів виробничого середовища у приміщенні



на підприємстві ТЗОВ «Хостінг Макс», а також визначений вплив цих факторів на здоров'я та працездатність працівників. Слід зазначити, що було встановлено відповідність всіх розглянутих показників чинним санітарним нормам та виявлено, що умови праці в ТЗОВ « Хостінг Макс » є оптимальними.

## ВИСНОВКИ

В результаті виконання дипломної роботи:

1. На основі проведеного аналітичного огляду типів та методів тестування, особливостей автоматизованого тестування виявлено, що методи тестування для програмних застосунків відрізняються відповідно до сфери застосування, тестування сумісності та інтерфейсу краще проводити вручну або з використанням інструментів для перевірки навантаження, роботи системи при різних умовах. Автоматизоване тестування найкраще підходить для регресійних тестів;
2. Здійснено аналіз існуючих методів тестування програмних застосунків. Для тестування програмних застосунків необхідно виділити сферу тестування і згідно з обраним використовувати необхідні інструменти. Для тестування web-застосунку можна використати наступні інструменти: EclipseIDE, Java, фреймворк Cucumber, Git, бібліотека Selenium, Junit, Jenkins, ін. Для тестування мобільних застосунків: Java/Kotlin, Android Studio, XCode, Git, Appium Server/Client, Selenium, ін. Для тестування API: Java, Rest Assured. Тестування навантаження проводиться з використанням готових програмних рішень.
3. Виявлено, що характеристики процесорів комп'ютерів впливають на продуктивність системи, в якій реалізовано можливість роботи з потоками. Найкращими рішеннями є процесори розроблені AMD категорії Ryzen.

Гарними рішеннями є процесори: AMD Ryzen 9 3900X (високої цінової категорії), AMD Ryzen 5 3600 (середньої цінової категорії).

4. Було спроектовано і розроблено рішення для покращення автоматизованої системи QA для сайту “XpressDocx” шляхом впровадження паралельного запуску тестів та застосуванням шаблону тестування PageObject з урахуванням загальних методів тестування, операційних систем та архітектур комп’ютерів.
5. У спеціальному розділі з охорони праці наведено аналіз факторів виробничого середовища у приміщенні на підприємстві ТЗОВ «Хостінг Макс», а також визначений вплив цих факторів на здоров’я та працездатність працівників.

## **СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ РОБОТИ**

1. “Могілянські читання - 2019”

### **АНОТАЦІЯ**

**Померанцева М. А.** Система QA для автоматизованої перевірки програмних застосунків. – На правах рукопису.

Магістерська робота на здобуття освітньої кваліфікації «Магістр комп’ютерної інженерії». – Чорноморський національний університет імені Петра Могили, Миколаїв, 2020.

У сучасному світі інформаційні системи використовуються всюди, багато коштів витрачається на розробку програмного і апаратного забезпечення, очікується, що продукт або система будуть якісними та виконувати поставлені задачі. Але навіть достатній бюджет не може гарантувати якості програмного забезпечення. Адже навіть найкращі розробники допускають помилки у системі. Ручна перевірка якості системи, її компонентів, взаємодії додатків займає багато часу та не є ефективною при

багаторазовому повторенню однакових дій чи процесів, які необхідно тестувати періодично. Автоматизоване тестування програмних, апаратних додатків та їх сумісності зменшує витрати на розробку, оновлення системи за допомогою швидкості виконання, можливості повторного використання, охоплення різноманітних кейсів, високої точності без людської похибки.

Пояснювальна записка бакалаврської роботи складається зі вступу, трьох розділів, висновків, переліку джерел посилання, 2 додатків та спеціальної частини з охорони праці.

У вступі визначається актуальність теми, наведені задачі, які заплановано вирішити для досягнення поставленої мети. У першому розділі проводиться аналіз сучасного стану мануального та автоматизованого тестування. У другому розділі проведено дослідження апаратної складової та інструментів проекту. У третьому розділі описано реалізацію автоматизованої системи тестування для web – застосунку «XpressDocx», впроваджено методи для покращення системи. Розробка системи тестування здійснена в Eclipse IDE, програмний продукт створювався для платформи Java з версією JDK 11.0. У висновках наводяться підсумки проведеної роботи та основні переваги та недоліки розробленої системи.

Магістерська робота містить рисунки, 30 літературних джерела та 2 додатків. Загальний обсяг дипломної роботи складає 75 сторінок.

**Ключові слова:** : *QA, автоматизація, тестування, програмні застосунки, java, Selenium, Cucumber, Jenkins, процесор, операційна система, паралельні процеси.*

## ABSTRACT

**Pomerantseva M.** QA system for automation testing of software applications.  
– On the rights of the manuscript.

Master's work for obtaining an educational qualification "Master of Computer Engineering". – Petro Mohyla Black Sea National University, Mykolaiv, 2020.

In the modern world informational system are used everywhere. When money is expended on developing of the software and hardware it`s expected that the product or system will be of good quality and fulfill tasks. But even a sufficient budget cannot guarantee that software will have high quality because the best developers make mistakes in the system. Manually checking the quality, components and application interaction of a system, is time-consuming and ineffective in repeatedly repeating the same actions or processes that need to be tested periodically. Automated testing of software, hardware applications and their compatibility reduces development costs, system upgrades with runtime, reusability, coverage of various cases, high accuracy without human error.

The thesis consists of an introduction, three chapters, conclusions, references, two appendices and a special chapter which covers problems of health and safety in emergency. The third section describes the implementation of an automated testing system for the XpressDocx web application, and introduces methods for improving the system. Testing system was developed in Eclipse IDE, the software was created for Java platform with JDK 11.0 version. The conclusions summarize the work done and the main advantages and disadvantages of the developed system.

The work contains \_\_ drawings, 30 references and 2 appendices. The total volume of the thesis is 70 pages.

**Keywords:** *QA, automation, testing, software, java, Selenium, Cucumber, Jenkins, processor, operating system, parallel processes.*