



Universitatea Tehnică a Moldovei

**Automatization of functional web application
testing**
**Automatizarea testării funcționale a aplicațiilor
web**
**Автоматизация функционального
тестирования веб приложений**

**Masterand:
Vladimir Zubenco**

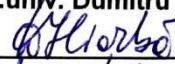
**Conducător:
lect. univ. Mariana Catruc**

Chișinău – 2019

MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII
al REPUBLICII MOLDOVA
FACULTATEA Calculatoare, Informatică și Microelectronică
DEPARTAMENTUL Ingineria Software și Automatică

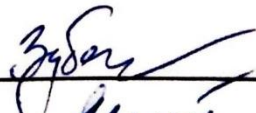
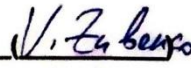
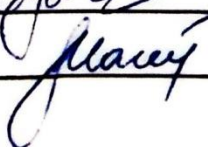

Admis la susținere

Șef de departament: dr.conf.univ. Dumitru Ciorbă


„17” decembrie 2019

Автоматизация функционального тестирования веб приложений

Teză de master în Tehnologii Informaționale

Masterand:  
Conducător:  

Chișinău – 2019

ADNOTARE

Teza cu tema „Automatizarea testării funcționale a aplicațiilor web” este constituită din introducere, trei capitole, concluzii și recomandări, bibliografie cu 34 titluri, 3 anexe, 110 pagini de text de bază, 6 figuri, 16 tabele.

În teză au fost utilizate cuvinte-cheie, cum ar fi: testarea, cerințele, funcționalitatea, software-ul, acoperirea, agile, testabilitatea, unit testing.

Testarea software-ului vă ajută să economisiți bani și efort pe dezvoltarea de software, ajutând la detectarea problemelor cu mult înainte ca procesul de dezvoltare să meargă prea departe. Se știe că cu cât defectul rămâne mai puțin detectat, cu atât este mai scump procesul de corectare. Costul corectării acestor erori face ca testarea să fie o parte importantă a procesului de dezvoltare.

Testarea poate fi costisitoare și consumatoare de timp, mai ales în cazurile în care este efectuată manual. S-a stabilit că eforturile de testare a software-ului consumă cel puțin jumătate din totalul forței de muncă a proiectului. Automatizarea ajută la testarea funcționalității de orice dimensiune mai exactă și mai ieftină, eliberează echipa de testele de regresie nesfârșite efectuate manual, permite teste mai specifice și mai detaliate. Lucrarea mea se concentrează pe modul de automatizare a testării software-ului și a modului în care automatizarea poate îmbunătăți eficiența echipei de dezvoltare software, precum și calitatea produsului final.

АННОТАЦИЯ

Магистерская работа на тему: «Автоматизация функционального тестирования веб приложений» состоит из введения, трех глав, вывода и рекомендаций, библиографии из 34 пунктов, 3 приложений, 110 страниц основного текста, 6 рисунков, 16 таблиц.

В работе были использованы ключевые слова, такие, как: тестирование, требования, функционал, программное обеспечение, покрытие, agile, тестируемость, unit testing.

Тестирование программного обеспечения помогает сэкономить деньги и усилия, потраченные на создание программного, помогая обнаружить проблемы задолго до того, как процесс разработки пойдет слишком далеко. Известно, что чем дольше дефект остается необнаруженным, тем дороже обходится процесс его исправления. Стоимость исправления подобных ошибок делает тестирование важной частью процесса разработки.

Тестирование может быть дорогостоящим и трудоемким, особенно в тех случаях, когда оно выполняется вручную. Установлено, что усилия, направленные на тестирование программного обеспечения, потребляют по меньшей мере половину общей рабочей силы проекта. Автоматизация помогает сделать тестирование функционала любых размеров более точным и дешевым, освобождает команду от бесконечных регрессионных тестов, выполняемых вручную, дает возможность проведения более конкретных и углубленных испытаний. В моей работе основное внимание уделяется методам автоматизации тестирования программного обеспечения и тому, как автоматизация может повысить эффективность работы команды разработчиков программного обеспечения, а также качество конечного продукта.

ANNOTATION

The master's thesis by the topic: "Automatization of functional web application testing" consists of an introduction, three chapters, a conclusion with recommendations, a bibliography with 34 items, 3 appendixes, 110 pages of general text, 6 figures, 16 tables.

Following key-words have been used in the thesis body: testing, requirements, functionality, software, coverage, agile, testability, unit testing.

Software testing helps save money and effort on software development, helping to detect problems well before the development process goes too far. It is known that the longer the defect remains undetected, the more expensive the process of fixing it. The cost of such errors makes testing an important part of the development process.

Testing can be costly and time consuming, especially in cases where it is performed manually. It has been established that efforts to test software consume at least half of the total project workforce. Automation helps to make the testing of functionality of any size more accurate and cheap, frees the team from endless regression tests performed manually, allows for more specific and in-depth tests. My work focuses on how to automate software testing and how automation can improve the efficiency of the software development team, as well as the quality of the final product.

Оглавление

ВВЕДЕНИЕ	10
1. ПРИЕМУЩЕСТВА, НЕДОСТАТКИ И СТРАТЕГИИ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ	12
1.1. Увеличение производительности	12
1.2. Повышение надежности	13
1.3 Уменьшение времени, затраченного на выполнение	13
1.4 Типы стратегий тестирования	13
1.5 Проверка на тестируемость (testability)	14
1.6 Порядок тестирования	15
1.7 Пример недоверия к автоматическим тестам	15
1.8 Снижение сложности автоматизированных тестов	17
1.9 Затраты на запуск	19
1.10 Возможность поддержки	20
1.11 Коммерческие тестовые пакеты	21
1.12 Тестовые пакеты с открытым исходным кодом	21
2. АВТОМАТИЗАЦИЯ В РАЗЛИЧНЫХ МЕТОДОЛОГИЯХ РАЗРАБОТКИ, КОММЕРЧЕСКИЕ И ОТКРЫТЫЕ ИНСТРУМЕНТЫ	23
2.1 Методология разработки Agile	23
2.2 Методология разработки Extreme Programming (XP)	23
2.3.1 Библиотека для тестирования JUnit	24
2.3.2 Библиотека для тестирования TestNG	24
2.4 Тестирование как сервис (Testing as a Service - TaaS)	25
2.5 Тестируемое приложение Recipe Box	26
2.6 Виртуальные машины	27
2.7 Коммерческий инструмент тестирования T-Plan Robot	28
2.8 Коммерческий инструмент тестирования QF-Test	30
3. ТЕСТЫ, НАПИСАННЫЕ БЕЗ ПОМОЩИ КОММЕРЧЕСКИХ ИНСТРУМЕНТОВ	33
3.1 Требования к тестируемому приложению Recipe Box	35

3.1.1 Введение	35
3.1.2 Общее описание	37
3.1.3 Требования	38
3.1.4 Глоссарий	40
3.1.5 Ссылка на источник формата и содержания документа	40
ВЫВОД	41
БИБЛИОГРАФИЯ	42
ПРИЛОЖЕНИЕ А. ПЛАН ТЕСТИРОВАНИЯ RESCPE BOX	45
ПРИЛОЖЕНИЕ Б. КОД, ТЕСТИРУЮЩИЙ RESCPEBOX	56
ПРИЛОЖЕНИЕ В. ИСХОДНЫЙ КОД ПРИЛОЖЕНИЯ RESCPEBOX	84

