

**УДК 004.415.53**

**Садыхян В.К., Лаврушина Е.Г.**

**ИССЛЕДОВАНИЕ ПОДХОДОВ АВТОМАТИЗАЦИИ ПРОЦЕССОВ  
ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ НА ПРИМЕРЕ  
ПОРТАЛА “MYAUDIENCE”**

*Владивостокский государственный университет экономики и сервиса,  
Владивосток, Гоголя 41, 690014*

**UDC 004.415.53**

**Sadikhyn V.K., Lavrushina E.G.**

**RESEARCH APPROACH AUTOMATING THE PROCESS OF TESTING OF  
WEB APPLICATIONS AN EXAMPLE  
PORTAL “MYAUDIENCE”**

*Vladivostok State University of Economics and Service,  
Vladivostok, Gogolya str. 41, 690014*

*Традиционные методы разработки тестов вручную уже не могут обеспечить быстрое и качественное тестирование современных программных систем. В данной статье рассматривается подход к автоматизации процесса тестирования. В настоящее время необходимость систематизированного тестирования в промышленной разработке программного обеспечения (ПО) общепризнанна и неоспорима.*

*Ключевые слова: веб-проект, тестирование, инструменты тестирования, скрипт, тестовые сценарии, фреймворк.*

*Traditional methods of developing tests manually can no longer provide fast and quality testing of modern software systems. This paper discusses an approach to automating the testing process. At present, the need for systematic testing in the industrial development of software (software) is well recognized and undisputed.*

*Keywords: web projects, testing, testing tools, scripts, test scripts, framework.*

Тестирование — самая популярная методика повышения качества, подкрепленная многими исследованиями и богатым опытом разработки коммерческих приложений. Существует множество видов тестирования: одни обычно выполняют сами разработчики, а другие — специализированные группы. Одним из видов тестирования, который выполняет специализированная группа, является регрессионное тестирование. Регрессионное тестирование включает: проверку исправления вновь найденного дефекта, проверку, что исправленный ранее и верифицированный дефект не воспроизводится в системе снова, а также проверку того, что не нарушилась работоспособность работающей ранее функциональности.

Единственный практичный способ управления регрессионным тестированием — его автоматизация. Многократное выполнение одних и тех же тестов, приводящих к тем же результатам, вводит людей в транс. Они утрачивают концентрацию и начинают упускать ошибки, что сводит на нет эффективность регрессионного тестирования. Известно, что уровень ошибок, допускаемых при тестировании вручную, сравним с уровнем ошибок в самом тестируемом коде. По оценкам, при тестировании, проводимом вручную, должным образом выполняется лишь около половины всех тестов [1].

Автоматизированное тестирование программного обеспечения — часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения.

Существующие на сегодняшний день методы тестирования ПО не позволяют однозначно и полностью выявить все дефекты и установить корректность функционирования анализируемой программы, поэтому все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого ПО.

Такой процесс формальной проверки, или верификации, может доказать, что дефекты отсутствуют с точки зрения используемого метода. Существует множество подходов к решению задачи тестирования и верификации ПО, но эффективное тестирование сложных программных продуктов — это процесс

творческий, не сводящийся к следованию строгим и чётким процедурам или созданию таковых.

Существует два основных подхода к автоматизации тестирования: тестирование на уровне кода и тестирование пользовательского интерфейса. К первому типу относится, в частности, модульное тестирование. Ко второму — имитация действий пользователя с помощью специальных тестовых фреймворков.

Наиболее распространенной формой автоматизации является тестирование приложений через графический пользовательский интерфейс. Популярность такого вида тестирования объясняется двумя факторами: во-первых, приложение тестируется тем же способом, которым его будет использовать человек, во-вторых, можно тестировать приложение, не имея при этом доступа к исходному коду.

Данная работа посвящена исследованию и анализу существующего современного подхода, основой которого является автоматизация процесса тестирования пользовательского интерфейса веб-приложения.

Веб-приложением называют клиент-серверное приложение, в котором клиентом выступает веб-браузер, а сервером – веб-сервера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети.

Для эффективной автоматизации процесса тестирования необходимо на начальном уровне тщательно изучить все функции веб-приложения, а затем составить требования к механизму тестирования.

В результате анализа веб-портала “MyAudience” были выделены его основные функции:

- Управление лицензионной политикой программных продуктов myAudience;
- Регистрация программных продуктов myAudience;
- Построение и отображение статистических данных на графике;

- Управление профилем пользователя;
- Ведение журнала событий и действий пользователя.

Требования чаще всего составляются в зависимости от ожидаемого результата и вида тестирования. На основе требований разрабатываются тестовые случаи на естественном языке.

Тестовый случай - это процесс, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Под тестовым случаем понимается структура вида:

**Действие => Ожидаемый результат => Результат теста**

Тестовые случаи должны максимально охватывать все возможные пользовательские сценарии и покрывать все изученные ранее функции веб-ресурса.

Ниже представлен пример тестового сценария:

### **1. Действие**

Открыть страницу "Login"

### **2. Ожидаемый результат**

Окно "Login" открыто

Название окна - "Login"

Логотип компании отображается в левом верхнем углу

На форме "Login" 2 поля – "Email" и "Password"

Кнопка "Login" доступна

Ссылка "Forgot your password? [Click here](#)" – доступна

### **3. Результат теста**

Тест пройден/не пройден

На основе разработанных тестовых сценариев необходимо создать автотесты, используя выбранное программное средство для автоматизированного тестирования.

Выполнение комплексного тестирования с использованием средств автоматизированного тестирования для последовательно возрастающего

количества версий программного обеспечения представляет большую ценность. С каждой новой версией появляются новые тесты, но можно использовать и ранее разработанные тестовые скрипты. В условиях постоянных изменений и добавлений требований и программного обеспечения автоматизированное тестирование служит важным механизмом проверки, обеспечивающим точность и стабильность программного продукта при выходе каждой новой версии.

Автотест — форма программирования на языках, специально разработанных для автоматизации тестирования ПО. Автотесты более всего подходят для тестирования пользовательского интерфейса приложения и могут быть объединены в систему [2].

Для запуска и быстрой разработке автотестов необходимо определиться с программным средством для автоматизированного тестирования, отвечающего критериям среды системной разработки проекта, на ранних стадиях жизненного цикла. Критерии оценки и выбора инструмента тестирования ПО:

- Оценка возможностей усовершенствования;
- Подход к выбору надлежащего средства;
- Диапазон затрат на инструментарий;
- Дополнительное время на внедрение средства;
- Экспертиза инструмента;
- Стоимость обучения использованию средства;
- Процесс внедрения средства.

При наличии нескольких инструментов соответствующих всем критериям оценки нужно выявить среди кандидатов лучший. Кандидатуры средств тестирования оцениваются на основе функциональных критериев, отражающих требования к инструменту.

При выборе средства важно учитывать поддержку конкретной среды разработки, возможность построения отчетов о тестировании, автоматическая регистрация обнаруженных дефектов, наличие сценариев восстановления. Важную роль играет наличие документации и линии технической поддержки —

современные инструменты тестирования не менее сложны, чем средства разработки. Следует обратить внимание и на наличие специализированных форумов, посвященных средствам тестирования, – существование форума по конкретному инструменту и, главное, многочисленной группы активных пользователей говорит о широкой распространенности данного инструмента, что в дальнейшем поможет быстрее получить ответ на интересующий вопрос. Кроме того, нужно обращать внимание на возможность интеграции инструментов тестирования с программным обеспечением, которое используется в компании.

Результаты анализа средств автотестирования для веб-портала “MyAudience” представлены в таблице 1.

**Таблица 1**

**Сравнительный анализ программных средств**

<b>Название</b>	<b>Распространение</b>	<b>Язык тестов</b>	<b>Поддерживаемые технологии</b>
Canoo WebTest	Бесплатный	Groovy	веб технологии
Codeception	Бесплатный	PHP	веб технологии
Coded UI Test	Платный	C#	NET 2.0, 3.0, 3.5,
Selenium	Бесплатный	Java, C#, Perl, PHP, Python, и Ruby	DHTML, JavaScript, Ajax

Анализ показал, что подходящим программным средством является Codeception. Codeception – фреймворк для написания функциональных тестов на PHP с возможностью выполнять их в эмуляторе браузера. Отличительной чертой программного средства является простота написания тестов. Тест пишется в особой манере - каждая команда эмитирует действие тестера.

Преимущества codeception

- Простая установка;
- Интеграция с наиболее известными фреймворками;
- Хорошая документация;
- Стил тестов приближен к естественному языку (BDD);

После выбора программного средства следует перейти к непосредственному созданию автотестов. Созданные наборы автотестов являются модулями системы автотестирования. Модули разделены по функциональным возможностям в соответствии с функциями web-портала “MyAudience”:

- Модуль тестирования страницы «Вход»;
- Модуль тестирования страницы «Информация»;
- Модуль тестирования страница «Группы и записи»;
- Модуль тестирования страницы «Метрики»;
- Модуль тестирования страницы «Профиль»;
- Модуль тестирования страницы «Лицензии»;
- Модуль тестирования всех страниц.

Итоги запуска и выполнения всех автотестов должны отобразиться в отчете по тестированию.

Отчёт о результатах тестирования – часть тестовой документации, включающая в себя описание процесса тестирования, суммарную информацию о протестированных за подотчётный период разделов, информацию о деятельности тестировщика, а также некоторые статистические данные.

Цель написания отчета – предоставление лицам, заинтересованным в проекте, полной и объективной информации о текущем состоянии качества проекта. Эта информация выражается в конкретных фактах и цифрах.

Вывод:

В результате исследования данного метода тестирования можно сказать о его положительном эффекте. Анализ показал, что данный подход удовлетворяет требованиям автотестирования веб-приложений.

## Сравнительный анализ методов тестирования

	Ручное тестирование	Автотестирование
Количество тестов (шт)	120	120
Время выполнения тестирования (чел/ч)	16	0,1
Работники отдела тестирования (чел)	2	1

Данные, приведенные в таблице 2, доказывают, что основная задача автотестирования, а именно сокращение временных и материальных затрат, достигнута.

Стоит отметить, что исследованный метод можно использовать не только при автоматизации регрессионного тестирования, но и при автоматизации функционального и системного тестирования.

Преимущества подхода:

- Позволяет создавать тесты, не дожидаясь окончания разработки приложения, руководствуясь только требованиями и дизайном интерфейса.
- Стиль создания прост и удобен, тесты может разрабатывать человек, не владеющий навыками программирования.

Недостатки подхода:

- Отсутствие автоматизации процесса разработки теста. Все тестовые наборы разрабатываются вручную, это приводит к проблемам на стадии доработки тестов или их редактирования.

Возможные направления развития данного подхода:

- Автоматизация поддержки тестового набора в актуальном состоянии при изменении интерфейса веб-приложения. Единственным решением является локализация компонентов теста, зависящих от детализации интерфейса веб-

приложения, позволяет сократить усилия по проведению тестов в соответствии с изменяющимся интерфейсом.

•Расширение видов тестирования, поддерживаемых инструментов. Начиная от генерации некорректных Http-запросов для тестирования устойчивости приложения и заканчивая использованием разработанных наборов тестов для нагрузочного тестирования.

#### Литература:

1. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Пер. с англ. М: Лори. 2003. 592 с. ISBN – 5-85582-186-2
2. Майерс Г. Искусство тестирования программ. 3-е изд. Пер. с англ. М: Диалектика, Вильямс. 2012. 272 с.

#### References:

1. Dustin E., Rashka J., Paul J. Automated software testing: Introduction, Management, and Performance. Pages: 573 Publisher: Addison Wesley Published: 1999
2. Glenford J. Myers ; Revised and updated by TomBadgett and Todd Thomas, with Corey Sandler.—2nd ed.p. cm.ISBN 0-471-46912-/ Copyright © 2004 by Word Association, Inc. All rights reserved.Published by John Wiley & Sons, Inc., Hoboken, New Jersey.Published simultaneously in Canada.Myers, Glenford J.The art of software testing