

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники
Кафедра инженерной психологии и эргономики

УДК 331.101.1:004.415.53

Корнеев
Максим Николаевич

**ПОВЫШЕНИЕ ЭРГОНОМИЧНОСТИ АВТОМАТИЗИРОВАННОЙ
СИСТЕМЫ ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА
ВЕБ-ПРИЛОЖЕНИЙ**

АВТОРЕФЕРАТ

на соискание степени магистра технических наук

1-23 80 08 – Психология труда, инженерная психология, эргономика

Магистрант М.Н. Корнеев

Научный руководитель
О.Н. Малышева, кандидат
физико-матем. наук, доцент

Заведующий кафедрой ИПиЭ
К.Д. Яшин, кандидат
технических наук, доцент

Нормоконтролер
В.С. Гладкая,
ассистент кафедры ИПиЭ

Минск 2019

ВВЕДЕНИЕ

В век информационных технологий многие процессы автоматизируются и систематизируются благодаря различным программным средствам. Очень важно, чтобы, несмотря на общую доступность и удобство обмена информацией с помощью веб-сайтов, пользователи были уверены в сохранности личной информации и в том, что, качество веб-приложения которое они используют было на высоком уровне.

Целью данной диссертации является эргономическое проектирование и разработка автоматизированной системы тестирования пользовательского интерфейса веб-приложений. Система должна автоматически проверять качество пользовательского интерфейса веб-сайта.

Магистерская диссертация посвящена эргономическому проектированию и разработке одного из видов современных систем – автоматизированной системе тестирования пользовательского интерфейса веб-приложений. Прделанная работа направлена на создание успешной системы: эффективной с точки зрения затрат на её создание, функционирование, обучение пользователя и др.; обеспечивающей условия рабочей среды, не наносящей вред пользователям; способствующей личностному и профессиональному развитию пользователей. Данная система должна соответствовать уровню развития науки на момент её создания, быть эстетически привлекательной и удобной для работы в ней человека.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы обусловлена несоответствием пользовательского интерфейса систем тестирования необходимым свойствам эргономичности.

Объектом исследования является система «человек-веб-сайт».

Предметом исследования выступает повышение эргономичности пользовательского интерфейса системы тестирования веб-приложений.

Цель работы повышение эргономичности пользовательского интерфейса с помощью разработки системы тестирования. Разработанная система должны быть объединена с непрерывной системой интеграции.

Задачи исследования:

- Изучить научно-техническую литературу по существующим системам тестирования пользовательского интерфейса веб-приложений;
- Провести анализ систем тестирования и их свойств эргономичности;
- Разработать и испытать систему тестирования пользовательского интерфейса веб-приложений.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Автоматизированное тестирование использует программные средства для выполнения тестов и проверки корректности результатов выполнения, что упрощает тестирование и сокращает его длительность. Главное преимущество автоматизированного тестирования состоит в возможности повторного прогона тестов без участия человека.

Автоматизированные тесты, как правило, являются регрессионными, (от латинского Regression-движение назад), то есть направленными на обнаружение ошибок в уже протестированных участках исходного кода при внесении изменений.

Одной из главных проблем автоматизированного тестирования является его трудоемкость: несмотря на то, что оно позволяет устранить часть рутинных операций и ускорить выполнение тестов, большие ресурсы могут тратиться на обновление самих тестов. Тем не менее подобные инвестиции в большинстве случаев оправданны, так как ручное тестирование требует гораздо больше ресурсов.

Фреймворк автоматизированного тестирования (англ. Test automation framework) - это набор предположений, концепций и инструментов, которые обеспечивают поддержку для автоматизированного тестирования программного обеспечения. Основным преимуществом такой структуры является низкая стоимость обслуживания.

Если есть изменения в любом тест-кейсе, то необходимо обновить только тестовый файл тест-кейса, а сценарий драйверов и сценарий запуска останется прежним. В идеале, нет необходимости обновлять скрипты в случае внесения изменений в приложение.

Выбор правильной техники фреймворка/сценариев помогает для снижения затрат. Расходы, связанные с тестовыми сценариями обусловлены разработкой и поддержкой усилий. Подход к сценариям, используемый при автоматизации тестирования отображается на затратах.

Функции фреймворка автоматизированного тестирования:

- определение формата, который отображает ожидания;
- создание механизма для подключения или выполнения в тестируемого приложения;
- выполнение тестов;
- отчетность о результатах.

Различают техники фреймворков/сценариев:

- Линейный (англ. Linear)
- Структурированный (англ. Structured)

- Движимый данными (англ. Data-driven)
- Движимый ключевым словом (англ. Keyword-driven)
- Гибридный (англ. Hybrid) (два или более из этих шаблонов используются)

Автоматизированное тестирование (АТ) наиболее эффективно, когда реализовано с помощью фреймворка. Несмотря на то, что в фреймворк зачастую используется для описания совокупности объектов, которая формирует инструмент модульного тестирования, эта статья будет в основном сфокусирована на фреймворках другого рода. Мы обсудим типы фреймворков, которые могут быть определены как совокупность абстрактных понятий, процессов, процедур и сред, с помощью которой автоматические тесты проектируются, создаются и реализуется. Кроме того, это определение фреймворка включает в себя физические объекты, используемые для создания тестов и их реализации, а также для организации логического взаимодействия между компонентами .

Автоматизированное тестирование (и, следовательно, фреймворки) развивалось годами, формируясь и усложняясь с каждой новой фазой эволюции. Эти фазы могут быть описаны в терминах трех поколений, каждое из которых обладает набором недостатков и преимуществ, благодаря которым каждое из них остается актуальным, несмотря на новые разработки. Представленные ниже понятия обычно используются для автоматизации функционального тестирования, но в некоторых случаях их можно применить и для решения задач модульного тестирования.

Тестирование веб-приложений является обязательным условием для правильной и корректной работы. На тестирование отводится много времени и средств.

Фреймворк, для корректной работы и выполнения необходимых проверок, должен состоять из следующих слоёв:

- Тесты. Это непосредственно сами модульные тесты.
- Страницы. Классы этого слоя – это PageObjects. Каждый класс этого слоя предоставляет средства взаимодействия с элементами страницы. Методы доступны вышележащим слоям: сервисному и тестам. Но тесты без необходимости напрямую со страницами не работают.
- Элементы страницы. Классы этого слоя предоставляют методы работы с отдельными элементами страниц.
- Кроме того, фреймворк должен использовать современные технологии. Основной технологией и наиболее популярной технологией является Selenium WebDriver.

Selenium – это инструмент для тестирования веб-приложений.

Page Object - это популярный в автоматизированном тестировании шаблон проектирования, который упрощает поддержку написанных тестов и уменьшает количество дублируемого кода. Это объектно-ориентированный класс, который выступает интерфейсом веб-страницы тестируемого приложения. Соответственно, все изменения, которые необходимо осуществить для реализации поддержки нового интерфейса будут сосредоточены в одном месте.

Шаблон проектирования “Page Object” обладает следующими преимуществами:

1. Четкое разделение между непосредственно кодом тестов и зависящего от страницы кода, вроде локаторов (или их применения, если используется карта пользовательского интерфейса, разметка);
2. Наличие единого репозитория для всех служб и операций, представленных на странице, а не множества разбросанных по всему тестовому набору [14].

В обоих случаях при изменениях в пользовательском интерфейсе нужно модифицировать код только в одном месте. Довольно много полезной информации об этой технике можно найти в различных блогах, так как данный шаблон проектирования становится все более и более популярным в среде тестировщиков.

Спроектированная система должна отвечать следующим критериям:

- система должна обладать механизмом подключения и отключения от тестируемого веб-приложения;
- система должна содержать слои – логики, тестов и утилит;
- слои системы должны быть разделены и обладать своей зоной ответственности;
- слой логики должен содержать page object шаблон;
- система должна выполнять автоматическое тестирование пользовательского интерфейса веб-приложения;
- система должна обладать отчетом о результатах проверок;
- система должна обладать логированием;

Для управления процессом контроля качества веб-приложения, было выбрано техническое решение в виде средство – Jenkins. Данная система для непрерывной интеграции была выбрана не только из-за удовлетворяющих технический возможностей нуждам решаемой проблемы, но и по причине эргономичного интерфейса для пользователя.

ЗАКЛЮЧЕНИЕ

В результате выполнения диссертации был произведен анализ существующих систем и методов для тестирования пользовательского интерфейса веб-приложений, произведено эргономическое проектирование и разработана автоматизированная система тестирования пользовательского интерфейса веб-приложений повышенной эргономичности.

Разработанное веб-приложение обладает следующими достоинствами:

- система базируется на непрерывной системе интегрирования, что позволяет планировать запуск выполнения процесса поиска дефектов;
- по окончании выполнения тестирования система готовит отчет результатов состояния тестируемой системы и имеет возможность рассылки их по почте;
- функциональные возможности, предоставляемые системой, позволяют расширять покрытие тестируемых веб-приложений и увеличивать количество тестируемых продуктов;
- система тестирования веб-приложений обладает простым и понятным интерфейсом;
- система отлажена, протестирована и готова для демонстрации и практического использования;

Недостатком системы является:

- необходимость установки и настройки непрерывной системы интеграции Jenkins;

Результаты исследований доложены на 54 СНТК студентов, магистрантов, аспирантов БГУИР в 2018 г.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Список использованных источников

[1-А] Корнеев, М.Н. Повышение эргономичности автоматизированной системы тестирования пользовательского интерфейса веб-приложений / М.Н. Корнеев // Повышение эргономичности автоматизированной системы тестирования пользовательского интерфейса веб-приложений: Тезисы докл. к конф. – Минск, 2018.