

ЛАБОРАТОРНАЯ РАБОТА №6

Тема лабораторной работы: автоматизация тестирования

Общие сведения по работе

Автоматизированное тестирование является способом повысить скорость контроля качества при разработке программных систем. В данной лабораторной работе необходимо приобрести навыки создания автотестов для контроля функциональности. Работа основывается на разработанной ранее тестовой документации, а именно тест-кейсах, тест-сютах, планах тестирования. Автоматизация тестирования может работать с различными типами приложений: desktop, мобильные приложения, веб-приложения. Из-за особенностей приложений выбор инструмента будет сильно зависеть от тестируемого приложения. Наибольшее распространение в настоящее время получили инструменты, позволяющие тестировать веб-приложения и мобильные приложения. Это связано с возрастающей популярностью таких типов приложений и возможностями инструментов разработки, преследующих цель повышения кросс-платформенности.

Методические рекомендации и материалы

Автоматизированное тестирование программного обеспечения (Software Automation Testing) – это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования. Достоинствами автоматизации тестирования являются:

- Увеличение скорости тестирования без ущерба для результата.
- Возможность выполнять тесты 24/7.

- Уменьшение объема ручных, рутинных, постоянно повторяющихся операций.

- Снижение стоимости итерации тестирования.

- Автоматическое формирование отчетов о тестировании.

- Набор тестов ограничивается производительностью системы, а не доступным резервом времени людей.

В то же время есть ряд недостатков:

- На начальном этапе внедрения автотестов требуется больше времени, чем в сравнении с ручным функциональным тестированием

- Плохо спроектированные/реализованные автотесты ведут к нестабильным результатам и ложным срабатываниям.

- Требуются люди с высокой квалификацией в области тестирования и знанием языков программирования и технологий одновременно.

- Требуется дополнительно время на сопровождение автотестов.

- Дефекты в автотестах ведут к пропуску багов в тестируемом приложении.

Перед началом разработки автоматизированных тестов нужно определить, что нуждается в таком тестировании:

- Труднодоступные места в системе.

- Часто используемая функциональность.

- Рутинные операции.

- Проверка данных, требующих точных математических расчетов.

- Проверка форм с множеством полей.

Не нужно строить автотесты для:

- Постоянно изменяющейся функциональности.

- Случаев необходимого постоянного вмешательства оператора в ходе выполнения автотестов.

- Проверки содержимого изображений, видео ряда, т. д.

Проектирование автоматизированных тестов зависит от поставленной цели. Это может быть нагрузочное тестирование, в котором использование инструментов автоматизации позволяет организовать необходимый режим эксплуатации приложения для снятия показателей производительности и отказоустойчивости, т. е. проверка нефункциональных требований к системе. Другие цели использования автоматизации могут проверять функциональность приложения. Режимы выполнения тестов в таких случаях будут:

- Эмуляция действий пользователя при взаимодействии с элементами пользовательского интерфейса.

- Тестирование программных интерфейсов, если целевая программная система является библиотекой, сервисом или компонентом, способным самостоятельно функционировать и выполнять задачи.

Во всех рассмотренных вариантах автотестов необходимо использовать сценарный подход в качестве основы для скриптов, кода автотестов. Источником будут наборы тест-кейсов.

С технологической точки зрения особенностями процесса создания автотестов являются:

- Автоматизация развертывания тестовых стендов (приложение, СУБД, платформа и т. д.) для выполнения автотестов.

- Написание программного кода автотестов, являющегося или самостоятельным программным проектом или являющимся отдельным модулем тестирования внутри основного приложения.

Обозначенные требования позволяют сделать вывод о необходимости обладания тестировщиком дополнительными компетенциями для создания и проведения автоматизированного тестирования.

После уточнения целей создания автотестов необходимо выбрать инструмент для автоматизированного тестирования. Приведем некоторые примеры:

- Selenium WebDriver является одним из возможных инструментов для проведения функционального тестирования веб-приложений. Его особенностью является использование установленного в операционной системе браузера для выполнения манипуляций с элементами пользовательского интерфейса веб-приложений. Selenium WebDriver является набором абстракций над веб-страницами, позволяющими искать и выбирать необходимые элементы (по имени тега, классу, xpath, ...), взаимодействовать с ними (клик, ввод текста, ...), осуществлять навигацию по страницам веб-приложения;

- xUnit автоматизация тестирования приложений, созданных на .Net Framework;

- Xamarin UI Tests – тестирование кросс-платформенных мобильных приложений, созданных на платформе Xamarin;

- Xautomation – пакет для linux, позволяющий эмулировать действия пользователя при работе с приложениями, запущенными на ПК;

- Monkey testing – тестирование мобильных приложений путем создания случайных потоков событий взаимодействия с элементами пользовательского интерфейса;

- Apache JMeter – тестирование API, нагрузочное тестирование;

Примеры использовать Selenium WebDriver можно найти в проекте <https://git.athene.tech/romanov73/tis-2017>. Кроме использования абстракций самого инструмента необходимо создавать автотесты с учетом свойств сопровождаемости, надежности, структурированности. Другими словами, автотесты необходимо создавать так, чтобы требовалось меньше времени на их поддержку и доработку, а случае изменений требований к

программной системе автотесты можно было бы незатратно актуализировать.

Одним из примеров такого подхода является использование POM – PageObjectModel, подход, позволяющий абстрагировать представление тестируемого приложения от сценариев тестирования. Делается это с целью сокращения трудоемкости актуализации тестов в случае изменения верстки веб-страниц приложения, изменения адресации элементов.

Для отработки навыков автоматизации тестирования веб-приложений существует ряд сервисов, предоставляющих типовые веб-компоненты, различные режимы использования. Список таких сервисов:

<https://demoqa.com/>

<http://uitestingplayground.com/>

<http://the-internet.herokuapp.com/>

<https://www.globalsqa.com/angularJs-protractor/BankingProject>

<https://www.saucedemo.com/>

<https://formy-project.herokuapp.com/>

<http://automationpractice.com/index.php>

Задания к лабораторной работе

1. Основываясь на ранее разработанной тестовой документации, разработать автотесты для проверки функциональности программной системы по варианту в соответствии с требованиями:

- a. Автотесты необходимо проектировать, исходя из свойств сопровождаемости, надежности, структурированности.
- b. Автотесты можно реализовать в форме отдельного приложения в виде модульных тестов.
- c. Автотесты должны эмулировать поведение пользователя при взаимодействии с элементами пользовательского интерфейса.

Обязательно реализовать контроль результатов выполненного сценария.

2. В отчет по лабораторной работе включить:

- a. Цель работы.
- b. Описание реализованных автотестов: инструменты, подходы.
- c. Код автотестов.
- d. Отчет о тестировании (выполненные тест-кейсы, результат, выявленные дефекты).
- e. Выводы по работе.
- f. Список использованных источников.

3. Оформить и защитить отчет.

Контрольные вопросы

- 1. Каковы цели использования автоматизированного тестирования?
- 2. В каких случаях можно использовать автоматизированное тестирование?
- 3. Каковы недостатки автоматизированного тестирования?
- 4. Что такое локатор?
- 5. Каковы цели нагрузочного тестирования?

СПИСОК ЛИТЕРАТУРЫ

1. Богданов, М.Р. Разработка клиентских приложений Web-сайтов [Электронный ресурс] : учебное пособие / М.Р. Богданов. – Электрон. дан. – Москва, 2016. – 258 с. – Режим доступа: <https://e.lanbook.com/book/100451>. – Загл. с экрана. (дата обращения: 10.10.2022).

2. Антамошкин, О.А. Программная инженерия. Теория и практика [Электронный ресурс] : учебник / О.А. Антамошкин. – Электрон. дан. – Красноярск : СФУ, 2012. – 247 с. – Режим доступа: <https://e.lanbook.com/book/45709>. – Загл. с экрана. (дата обращения: 10.10.2022).

3. Кознов, Д.В. Введение в программную инженерию [Электронный ресурс]: учебное пособие / Д.В. Кознов. – Электрон. дан. – Москва, 2016. – 306 с. – Режим доступа: <https://e.lanbook.com/book/100704>. – Загл. с экрана. (дата обращения: 10.10.2022).

4. Петрухин, В.А. Методы и средства инженерии программного обеспечения [Электронный ресурс] : учебное пособие / В.А. Петрухин, Е.М. Лаврищева. – Электрон. дан. – Москва, 2016. – 467 с. – Режим доступа: <https://e.lanbook.com/book/100645>. – Загл. с экрана. (дата обращения: 10.10.2022).

5. Котляров, В.П. Основы тестирования программного обеспечения [Электронный ресурс] : учебное пособие / В.П. Котляров. – Электрон. дан. – Москва, 2016. – 248 с. – Режим доступа: <https://e.lanbook.com/book/100352>. – Загл. с экрана. (дата обращения: 10.10.2022).

6. Технологии тестирования [Электронный ресурс] – Доступен в Интернете: <http://www.protesting.ru/> (дата обращения: 10.10.2022).

Учебное издание

РОМАНОВ Антон Алексеевич

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Лабораторный практикум

Редактор Н.А. Евдокимова

Подписано в печать 21.11.2022. Формат 60×84/16.

Усл. печ. л. 2,79. Тираж 35 экз. Заказ 548. ЭИ № 1747.

УлГТУ, 432027, Ульяновск, Сев. Венец, 32.

ИПК «Венец», УлГТУ, 432027, Ульяновск, Сев. Венец, 32.