**Внедрение автоматизированного тестирования сайта**

**доставки готовых блюд «О!Пельмень»**

В условиях пандемии все кафе и рестораны перешли на заказы в режиме онлайн. У сети кафе «О!Пельмень» значительно возросла нагрузка на сайт, который стал одним из основным источником дохода от заказов в компании. Через него проходит весь процесс обработки заказов: от принятия до онлайн-контроля доставки заказов.

По бизнес-сценарию в ближайшее время компания планирует масштабирование бизнеса в другие регионы, расширение меню и введение новых актуальных функционалов.

В связи с этим появляется необходимость внедрения автоматизированного тестирования.

Предполагается, что в штате компании набраны оптимальное количество опытных разработчиков и специалистов по автоматизированному тестированию, а также имеется вся необходимая техническая база для его налаживания и функционирования.

Заказчик на основании расчета трудозатрат и окупаемости проекта удостоверился, что введение автоматизированного тестирования принесет несомненные экономические выгоды, связанные с удержанием действующих и привлечением большего количества новых клиентов за счет более продуманного, качественного и бесперебойного сервиса по сравнению с конкурентами.

Автоматизация тестирования должна внедряться на ранних стадиях проекта и ее объём должен определятся контекстом проекта. Не нужно автоматизировать все тесты подряд, некоторые тестовые случаи просто невозможно автоматизировать, да и не нужно.

В первую очередь необходимо автоматизировать часто встречающиеся тесты, простые для автоматизации, хорошо понятные кейсы и сценарии, тесты проверки работоспособности базового функционала, проверка правильности поиска данных.

В нашем проекте в первую очередь следует автоматизировать тесты, отвечающие за регистрацию клиентов и ее подтверждение, обновление корзины покупателей и расчета заказов, а также сохранение результатов теста в базе данных для последующего анализа и составления статистики по различным параметрам.

Таким образом, в первую очередь необходимо автоматизировать следующие кейсы:

1. Участки кода, исполнение которых трудно визуализировать и получить осязаемую информацию о протекающих процессах (back-end процессы, занесение в базу данных, занесение логов в файл).

2. Функциональность продукта, которая будет использоваться наиболее часто и возникновение ошибок которой связано с достаточно высоким риском.

3. Типовые часто выполняемые операции, которые обычно связаны с обработкой данных. Например, заполнение формы заказа под каждого покупателя. Цель – автоматизация занесения требуемых данных в нужные поля.

4. Сообщения об ошибках.

5. Комплексная проверка поведения всей системы, как целостного объекта (end-to-end testing).

6. Проверка расчета стоимости заказа с учетом выбранных рекламных акций.

7. Тестирование корректности отображаемых результатов поиска в ответ на запрос по нужным данным.

Ручные проверки рекомендуются при следующих видах тестирования:

1. Тестирование сайта в реальном времени (исследовательское тестирование), так как незначительные изменения могут быть исследованы сразу, без написания кода и его исполнения, что значительно экономит время.

2. Исследовательское тестирование – его целью является проверка разнообразных возможностей приложения. Ключевым моментом здесь является то, что тестировщиком используются не заранее составленные тест-кейсы, а придуманные сходу сценарии. Тестировщик полагается на свои знания в сфере, опыт, логические умозаключения и интуицию. Данный вид тестирования помогает за короткое время обнаружить наиболее критичные дефекты.

3. Тестирование, направленное на борьбу с ботами.

4. Тестирование юзабилити - при проведении важно проанализировать, насколько удобным будет сайт для конечного пользователя. Главное для голодного человека – это интуитивно понятный и удобный интерфейс, привлекательнее и аппетитнее, чем у конкурентов.

5. Тестирование доработки сайта по обратной связи от потенциального пользователя, что даст возможность своевременного улучшения функционала для привлечения большего числа постоянных клиентов.

Для введения автоматизации заказчиком был рассмотрен ряд инструментов:

1. Selenium WebDriver - **это драйвер браузера, то есть не имеющая пользовательского интерфейса программная библиотека, которая позволяет различным другим программам взаимодействовать с браузером, управлять его поведением, получать от браузера какие-то данные и заставлять браузер выполнять какие-то команды.** Таким образом,  **WebDriver не имеет прямого отношения к тестированию.** Он предоставляет автотестам доступ к браузеру. Selenium WebDriver является бесплатным, то выгодно отличает его от других инструментов.

Однако с его помощью нельзя тестировать мобильные приложения, что не удовлетворяет требованиям заказчика, так он планирует запуск мобильного приложения для оформления заказов. Точно также, данный инструмент не позволяет тестировать десктопные приложения. Для формирования отчетов необходимо использовать дополнительные внешние инструменты (например, TestNG). Для написания скриптов в Selenium необходимо знание языков программирования, но в нашем случае данное требование не критично.

1. Katalon Studio –эффективный инструмент для автоматизации процесса тестирования веб-приложений, мобильных приложений и веб-сервисов. Данный инструмент может быть интегрирован в CI/CD, хорошо работает в связке с популярными инструментами qTest, JIRA, Jenkins и Git в процессе тестирования. Katalon Studio является потомком таких фреймворков, как Selenium и Appium.

Встроена функция **Katalon Analytics**, позволяющая пользователю получать полное представление о непосредственном процессе тестирования. Для этого на экран выводятся специальные отчёты, оформленные в виде графиков, метрик, диаграмм.

3. Appium — кроссплатформенный инструмент с открытым исходным кодом, который помогает автоматизировать тестирование мобильных приложений для Android и iOS. Appium поддерживает мобильные, нативные и гибридные приложения. Это бесплатное приложение. В качестве плюсов можно выделить то, что данный инструмент поддерживает любые языки программирования, дает возможность тестирования iOS и Android, поддерживает несколько фреймворков, можно использовать общепринятый программный API, является кросс-платформенным с открытым исходным кодом и совместим с Selenium webdriver. При этом, к его минусам можно отнести невозможность сравнения изображений, долгое время для настройки Appium для Android и iOS, а также слишком большое количество неожиданных ошибок.

Учитывая все плюсы и минусы рассмотренных инструментов автоматизации, рациональным решением видится использование инструмента Selenium WebDriver с опорой на Appium при тестировании мобильного приложения. Из-за особенности задачи выбрать один инструмент не представляется возможным. Учитывая опытную команду специалистов, основным направлением является разработка уникального инструментария для автоматизированного тестирования путем совмещения имеющихся инструментов.