

УДК 004.054

АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ДЛЯ ГРАФИЧЕСКОГО ВЕБ-ИНТЕРФЕЙСА

Степанюк А.В., магистрант гр. ПИМ-161, 2 курс

Павлова И.С., магистрант гр. ПИМ-161, 2 курс

Научный руководитель: Пимонов А.Г., д.т.н., профессор

Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

В ответственных проектах вопрос качества становится важнейшим. Сайт, портал, сервис, которому нельзя доверять – плохой продукт. Наличие ошибок в программном продукте избежать нельзя, ошибки есть и в больших, и в маленьких программах, и на сайтах, и в операционных системах. В настоящее время в IT-сфере как никогда стала актуальной профессия тестировщика программного обеспечения. Обычно их называют QA (Quality Assurance). Тестирование – это проверка соответствия программы требованиям, осуществляемая путем наблюдения за ее работой в специальных, искусственно созданных ситуациях, выбранных определенным образом. Разработчик тестов занимается тем, что он из огромного потенциально бесконечного набора тестов выбирает некоторый ограниченный набор [1].

В простых случаях (ручное тестирование) тестирование выполняет человек. Он проверяет работу различных сценариев использования в разных браузерах, для учётных записей с разными правами (оптовик, партнёр, гость, администратор и т.д.). Чем больше проект, тем больше проверок. Современное программное обеспечение является сложным многофункциональным объектом. Его ручная проверка требует значительных трудовых и временных затрат. На помощь приходят средства автоматизации тестирования. Если тестировщик автоматизирует тесты, то он не сам наблюдает за поведением программы – он делегирует эту задачу специальному инструменту или специальной программе, которую он сам написал. Именно она наблюдает, она сравнивает наблюдаемое поведение с ожидаемым, а тестировщику выдает только некоторый конечный результат – совпадает ли наблюдаемое поведение с ожидаемым, или не совпадает.

В каждой крупной компании, занимающейся разработкой программного обеспечения, имеется отдел тестирования. Не является исключением и компания OneVizion [2], которая разработала платформу, представляющую собой веб-приложение, предназначенное для решения проблемы управления сложносвязанной информацией. OneVizion непрерывно ведет разработку различных улучшений, которые упрощают работу с платформой. Процедура тестирования автоматизирована и проводится ежедневно. Также пишутся новые тесты для охвата новых функциональных возможностей.

Автоматизация тестирования имеет множество плюсов. Его ключевые преимущества это:

- 1) Повышение качества тестирования, поскольку при использовании средств автоматизации «человеческий фактор» не оказывает влияния на качество тестирования.
- 2) Возможность выполнения таких видов тестирования, которые либо не могут быть выполнены вручную, либо требуют значительных затрат (дополнительное оборудование, персонал).
- 3) Ускорение процесса тестирования без потери качества. Проведение того же объема работ ручным методом занимает больше времени. Использование средств автоматизации для тестирования позволяет запускать уже написанные скрипты без дальнейших доработок.
- 4) В ходе тестирования отчеты о результатах работы программных приложений рассылаются и сохраняются автоматически.

Автоматизированное тестирование GUI (графического интерфейса – то есть того, что пользователь видит на экране) позволяет эмулировать работу реальных пользователей с интерфейсом системы. При этом проверяется работа системы на соответствие техническому заданию при помощи современных инструментов автоматизированного тестирования.

Создание автоматических тестов для веб-приложения может быть сложным. Два из самых больших препятствий для начала работы – выбор инструмента автоматизации и разработка основы для написания тестов. Существуют разные подходы и инструменты для организации автоматизированного тестирования. Наиболее популярны тестирование отдельных частей проекта (модулей) и тестирование конкретных сценариев использования (например, добавление товара в корзину и оформление заказа). Тестирование по частям принято называть модульным или Unit-тестированием. Тестирование сценариев, которые позволяют пользователям решать определённые задачи (например, оформлять заказы), называется функциональным тестированием.

Для использования модульного тестирования весь проект должен разрабатываться как набор модулей и связей между ними. Если проект не запрограммирован аккуратно (с соблюдением принципов объектно-ориентированного программирования), его невозможно автоматически тестировать, и наоборот – если код должен быть покрыт автоматическими тестами, приходится думать об архитектуре проекта.

Функциональное тестирование не накладывает никаких ограничений на архитектуру проекта. Принцип тестирования прост: мы отправляем в браузер сценарий команд, который нужно выполнить, и проверяем результат. Прохождение таких тестов после завершения доработки сайта гарантирует работу основных сценариев работы проекта. Если после обновления сайта заказ перестал оформляться, вы об этом узнаете немедленно, а не через 1-2 дня простоя интернет-магазина.

Современные инструменты разработки позволяют запускать тесты нажатием одной кнопки. Эта задача решается с помощью специального программного обеспечения Jenkins. Данная система работает на отдельном сервере. Программист даёт сигнал о том, что он закончил решать задачу, и пора запускать тестирование. Jenkins получает изменения программиста, добавляет их на сайт и запускает тесты. По окончании тестирования Jenkins отправит всем заинтересованным письма с подробным отчётом о провалившихся тестах и результатах своей работы. Jenkins пришёл в веб из мира разработки десктопного программного обеспечения.

Процесс превращения программы из исходных кодов в нечто исполняемое и вразумительное называется сборкой. Обычно сборка состоит из нескольких шагов: проверка отсутствия синтаксических ошибок, тестирование модулей, компиляция программы и т. д. Каждая операция называется шагом сборки. Необходимость делить процедуру на шаги обусловлена экономией времени. Например, если мы нашли синтаксическую ошибку, не стоит продолжать сборку проекта.

Для своевременного выявления ошибок в работе платформы веб-приложение тестируется с использованием Selenium – инструмента для автоматизированного управления браузерами. Наиболее популярной областью применения Selenium является автоматизация тестирования веб-приложений [3]. Selenium даёт наиболее полноценное тестирование, так как можно запускать код в конкретной версии браузера (например, IE) под управлением конкретной ОС. Но его настройка сложнее настройки других инструментов, и он требует больше ресурсов. Когда выполняется тест, браузер должен быть запущен, и невозможно пользоваться компьютером, так как один лишний клик может сорвать выполнение теста. По этой причине для тестов обычно создают сеть виртуальных машин, в которых тесты и выполняются.

Веб-разработка по модели «continuous integration» с использованием автоматического тестирования требует заметных организационных и технических затрат. В небольших, коротких и простых проектах это не окупается. Если в разработке участвует больше 2 человек, постоянно меняющих код, или объем трудозатрат превышает 200 часов в месяц, внедрение этой технологии обязательно. На практике только при автоматизированном тестировании удастся добиться стабильности процесса и предсказуемости результата.

Список литературы:

1. Материалы сайта «Тестирование ПО» [Электронный ресурс]. – Режим доступа: <http://barancev.github.io/>, свободный (дата обращения: 19.03.2018).
2. Материалы сайта «OneVizion» [Электронный ресурс]. – Режим доступа: <https://onevizion.com>, свободный (дата обращения: 19.03.2018).
3. Материалы сайта «SELENIUM / WEBDRIVER» [Электронный ресурс]. – Режим доступа: <https://selenium2.ru>, свободный (дата обращения: 19.03.2018).