

ФРЕЙМВОРК ДЛЯ АВТОМАТИЗАЦИИ ПРИЕМОЧНОГО ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Матюшенко А.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Научный руководитель: Давыдович К. И – магистр техн. наук, ассистент кафедры ИПиЭ

Аннотация. Важный этап разработки любого ИТ-продукта – это тестирование. Правильно организованный процесс тестирования создаваемого программного продукта в конечном счёте позволит сэкономить деньги и время на устранении ошибок, и более того увеличить прибыль от хорошей репутации компании.

Ключевые слова: Автоматизация тестирования, фреймворк

Введение. Целью проекта является разработка фреймворка для повышения уровня тестирования программного обеспечения. Основная задача фреймворка это осуществление комплексного тестирования, необходимого для определения уровня готовности системы к последующей эксплуатации. Объектом разработки является фреймворк для автоматизации приемочного тестирования.

Основная часть. При разработке фреймворка использовался следующий стек технологий: *HTML, CSS, Java, Spring*. Предметная область проекта – разработка и тестирование программного обеспечения. Фреймворк позволяет клиентам, тестировщикам и программистам сравнить фактический результат с ожидаемым результатом, выявив дефекты программного обеспечения. Тестирование проводится на основании набора автоматизированных тестовых сценариев, покрывающих основные бизнес-операции системы.

Актуальность данного проекта обосновывается риском пропуска дефектов связанных с бэкенд структурой приложений мануальными тестировщиками. Данный фреймворк для автоматизации приемочного тестирования является набором уникальных по сравнению с другими программными платформами руководств по автоматизации, которые:

- 1) Поддерживают последовательность тестирования.
- 2) Улучшают структурирование теста.
- 3) Позволяют использовать минимальное количество кода.
- 4) Уменьшают затраты на обслуживание кода.
- 5) Повышают удобство повторного использования нетехническим тестировщикам участвовать в кодировании тестов.
- 6) Помогают сократить срок обучения использованию инструмента [1].

Нельзя забывать о том, что стоимость тестирования, особенно выполняемого вручную, слишком высока. Для приёмочного тестирования среднего проекта необходимо выполнить цикл, состоящий из 4500 тестовых задач, на выполнение одного цикла потребуется больше 40 человеко-дней. При обнаружении ошибки и после ее исправления соответственно необходимо повторно выполнить цикл тестовых задач. Время разработки фреймворка составило 95 человеко-дней. Учитывая количество циклов задач, необходимых для выполнения в ходе проектов и статистических данных, представленных выше очевидно, что представляемая платформа, существенно сокращает время выполнения цикла проверок приемочной части тестирования.

Фреймворк существенно исключает возникновение ошибок тестировщика, основанных на человеческом факторе, а также, снижает затраты организации на работу функциональных

тестировщиков. Все вышеперечисленные факторы определяют явные экономические и профессиональные выгоды от разработки и внедрения фреймворка для автоматизации приёмочного тестирования [2].

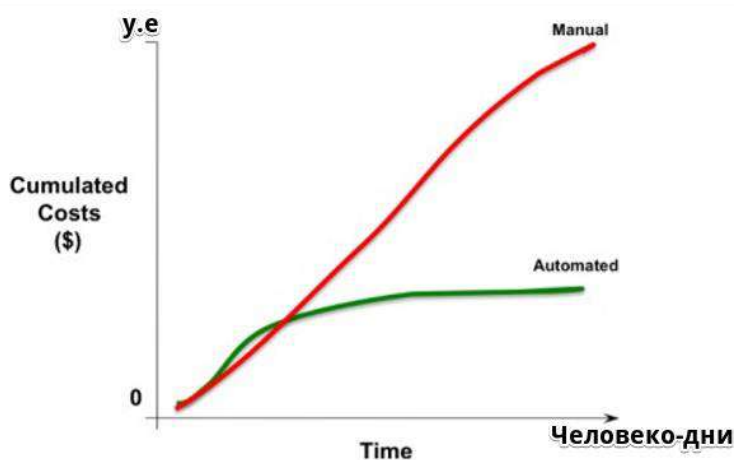


Рисунок 1 – Сравнительная диаграмма сравнения факторов времени и цены при осуществлении ручного и автоматизированного тестирования [2]

При написании автоматизированных тестовых сценариев используется система контроля версий *Git*. Для совместного проведения инспекции исходного кода мы используем свободное программное обеспечение с веб-интерфейсом *Gerrit*. Целью данной технологии является проведение просмотра тестовых сценариев специалистами по проверке кода на пригодность. Был использован сервер непрерывной интеграции *Jenkins*, для осуществления деплоя всех изменений кода тестовых сценариев на различные среды взаимодействия с пространственными пользователями программы.

Данная конфигурация позволяет оперативно осуществлять развертывание новой версии программного продукта сразу после внесения изменений в исходный код, хранящийся в репозитории. В случае ошибок конфигурация сервера всегда может быть возвращена к предыдущему состоянию [3].

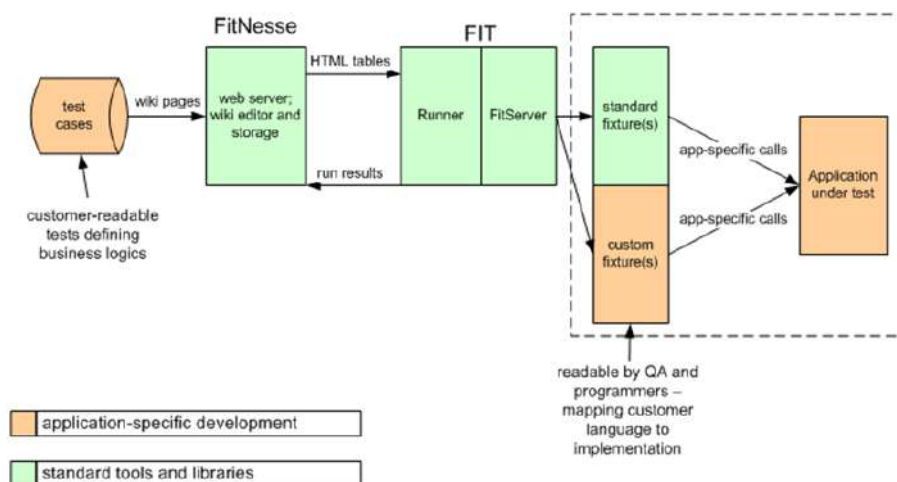


Рисунок 2 – Структура фреймворка для автоматизации приёмочного тестирования ПО [4]

Фреймворк сфокусирован на удобстве использования, как инструмента тестирования. Таким образом, он включает все необходимые компоненты: после запуска инструмент запускает встроенный веб сервер, который позволяет получить доступ к страницам с

одинаковой лёгкостью как с локальной машины, так и через интернет. Встроенный сервер довольно лёгкий и может быть запущен как на ноутбуке, так и на специализированном сервере.

Сама структура работы с фреймворком начинается с первоначального планирования перенесения тестовых сценариев, написанных специалистами функционального тестирования для автоматизации. Для этого оставляются таблицы, содержащие все необходимые переменные для написания теста, после ядро интегрированного фреймворка FIT которое в действительности обрабатывает каждую таблицу созданную в тестах, после проведения тестов фреймворком предоставляется результат, от которого зависит дальнейшее решение по сценарию, если он успешно пройден, то он помечается в тест-трекинговой системе, как успешный, если же нет, то тестирующий предоставляет разработчику результаты теста, для последующей корректировки кода.

Приложение возможно использовать не только для сепаративных проектов, но и для отделов работающих, с проектами, предусматривающие собой одинаковую логику для бэкенд структуры приложения, но разную структуру пользовательского интерфейса и функций, которые работают на клиентской стороне веб-сайта или приложения [4].

Заключение. Итогом проектирования является разработанный фреймворк для автоматизации процессов приемочного тестирования. В процессе разработки приложения были проведен анализ задачи, где был подробно рассмотрены все детали для реализации проекта и проанализированы основные запросы потенциальных пользователей, и подобран список необходимой литературы. Были выполнены поставленные задачи, такие как: создание информативного и удобного интерфейса, грамотное внедрение базовых веб-сервисов для работы, а также осуществление необходимой работы с системами контроля версий и разработка программной части, позволяющей применять все возможные достоинства при тестировании программного обеспечения.

Список литературы

1. Искусство тестирования программ / Гленфорд Майерс, Том Баджетт, Кори Сандлер. – Москва: Компания ООО Т-Айти, 2010. – 117 с.
2. Гибкое тестирование / Лиза Кристин, Джанет Грегори – Москва: Компания ООО Т-Айти, 2014. – 96 с.
3. Автоматизация приемочного тестирования – Москва, 2019. - Режим доступа: <https://omni.korusconsulting.ru/blog/bystryy-start-kak-avtomatizirovat-testirovanie/>. – Дата доступа: 22.08.2019
4. Приемочное тестирование для повышения качества программного продукта – Москва, 2020. - Режим доступа: <https://habr.com/ru/post/85831/> – Дата доступа: 12.04.2020

UDC 004.4'23

FRAMEWORK FOR AUTOMATING SOFTWARE ACCEPTANCE TESTING

Matyushenok A.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Davydovich K.I. – master of technical sciences, sssistant of the department of EPE

Annotation. An important stage in the development of any IT product is testing. A properly organized process of testing the software product being created will ultimately save money and time on fixing errors, and moreover increase profits from the good reputation of the company.

Keywords: testing automation, framework