

УДК 004.415:004.415.533

ПОДХОД К АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА IOS И ANDROID ПУТЕМ ГЕНЕРАЦИИ ИСХОДНОГО КОДА

Барсуков И.А. (Университет ИТМО), Бострикова Д.К. (Университет ИТМО),

Наумова Н.А. (Университет ИТМО)

Научный руководитель – преподаватель Гаврилов А.В.

(Университет ИТМО)

Введение. В работе излагается подход к созданию инструмента автоматизации для проведения унифицированного тестирования пользовательского интерфейса мобильных приложений, реализованных нативно под основные мобильные операционные системы iOS и Android с использованием подхода генерации исходного кода, что позволяет существенно сократить время, затрачиваемое на тестирование мобильных приложений.

Цель работы – сокращение времени прохождения этапа тестирования мобильных приложений путем автоматизации создания тестовых сценариев, объектов взаимодействия со страницами приложения, а также автоматизации процесса непрерывной интеграции с облачным сервисом.

Разработка крупных программных систем является сложной задачей для компаний-разработчиков программного обеспечения. Современные приложения проходят сверхбыстрые циклы разработки, поскольку бизнес-потребности требуют не только выполнения поставленных задач в кратчайшие сроки, но также и обеспечения качества [1]. Компаниям, занимающимся разработкой программного обеспечения, необходимо выпускать новые версии программного обеспечения настолько быстро, насколько это возможно в существующих ограничениях по времени на разработку. Для того, чтобы убедиться, что разработанное программное обеспечение соответствует требованиям, используется тестирование, задача которого — проверить корректность работы программы. Кроме того, тестирование позволяет повысить степень уверенности в качестве и надежности разрабатываемого продукта, а также повысить уровень пользовательского доверия в том, что программа функционирует корректно во всех необходимых обстоятельствах.

Существует множество различных видов тестирования: модульное, интеграционное, системное, нагрузочное. Но одним из самых трудоемких является UI-тестирование, требующее, зачастую, ручную взаимодействие со множеством элементов пользовательского интерфейса приложения [2]. Процесс проведения ручного тестирования мобильных приложений длительный и трудоемкий, учитывая, что при его проведении необходимо производить проверку приложения на обеих поддерживаемых мобильных платформах – Android и iOS.

Основная часть. Производится тестирование одного приложения с общей бизнес-логикой, но разной реализацией некоторых действий на различных платформах. Кроме того, UI-тесты, как правило, взаимодействуют с моделью объекта тестирования – логической частью страницы приложения (что возможно организовать с использованием паттерна PageObject [3]). Таким образом, процесс тестирования мобильных приложений можно организовать путем создания общего тестового сценария под обе платформы, но с использованием различных объектов PageObject, специфичных для платформы приложения. Процесс взаимодействия с элементами пользовательского интерфейса (ввод текста в текстовое поле, клик по элементу) отличается в зависимости от платформы, но организуется общим заранее известным набором команд и имеет внутри себя большое количество однотипного кода. Поэтому создание объектов модели взаимодействия с интерфейсом можно реализовать с использованием инструмента генерации кода из заранее определенного достаточно простого языка.

В мире тестирования мобильных приложений существуют лишь отдельные технологии, позволяющие организовывать UI-тестирование, тогда как готовых решений для автоматизации этого процесса в целом не существует. Подобными свойствами генерации

тестовых сценариев и объектов страниц обладают технологии, используемые в тестировании веб-приложений, которые не подходят для тестирования мобильных приложений из-за отсутствия необходимой функциональности [4, 5].

Для автоматизации процесса создания объектов страниц и тестовых сценариев необходимо определить их структуру. Так, был разработан шаблон объекта страницы – сущность, имеющая внутри себя уникальный идентификатор, позволяющий однозначно определить данную страницу, поля с идентификаторами объектов страниц и методы, которые, путем взаимодействия с основным тестовым фреймворком, взаимодействуют с элементами страницы приложения, а также шаблон тестового сценария, основной задачей которого является взаимодействия с объектами страниц приложений таким образом, чтобы такой сценарий мог быть использован под обе платформы без необходимости внесения в него изменений.

После того, как были определены составляющие данных сущностей, которые содержат схожие элементы, был реализован модуль генерации исходного кода тестовых сценариев и объектов страниц приложения на основе YAML-описания.

Предлагаемое решение – приложение, состоящее из нескольких компонентов, использующее в качестве базового тестового фреймворка Appium и позволяющее взаимодействовать с приложениями на iOS и Android. В качестве входных данных приложению передаются описания тестовых сценариев и объектов страниц, Далее система преобразовывает входные данные в код на языке программирования Kotlin и помещает результат работы в архив. Отдельный модуль приложения отвечает за выгрузку сборок приложений из удаленного центра приложений, а, благодаря интеграции с облачной системой тестирования приложений на реальных мобильных устройствах, производит загрузку артефактов и непосредственно приложений в облако, запуск процесса тестирования, и возвращает результат выполнения сценариев.

Такой подход обеспечивает полноценную автоматизацию тестирования мобильных приложений, а благодаря модульности разработанного решения обеспечивается возможность не только запускать весь процесс тестирования, но и каждый этап отдельно, в том числе локально.

Выводы. Из всего вышесказанного можно сделать вывод о том, что реализованная инфраструктура может быть использована для любых мобильных приложений. Модуль генерации объектов страниц позволяет создавать объекты как только для одной из платформ, так и для обеих сразу, поэтому использование модуля генерации объектов страниц оправдано даже в том случае, когда необходимо протестировать приложение только для одной платформы. Что касается расширения функциональности модуля генерации кода, то это возможно путем модернизации структуры модели с минимальными изменениями в части непосредственно генерации кода объектов.

По результатам внедрения данного в решения в цикл создания мобильных приложений и автоматизации части тестовых сценариев, выявлено ускорение процесса проведения тестирования.

Список использованных источников:

1. Sommerville I. Software engineering 10th Edition //ISBN-10. –2015. – Т. 137035152. – С. 18.
2. APOGEN: automatic page object generator for web testing / A. Stocco [и др.] // Software Quality Journal. — 2017. — Т. 25, No 3. — С. 1007—1039.
3. Morgado I. C., Paiva A. C. R., Faria J. P. Automated pattern-based testing of mobile applications //2014 9th International Conference on the Quality of Information and Communications Technology. – IEEE, 2014. – С. 294-299.
4. Ramya P., Sindhura V., Sagar P. V. Testing using selenium web driver //2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). — IEEE. 2017. — С. 1—7.

5. Stocco A. et al. Why creating web page objects manually if it can be done automatically?
//2015 IEEE/ACM 10th International Workshop on Automation of Software Test. – IEEE, 2015. –
C. 70-74.