

*Иванников Дмитрий Владимирович*

магистрант

*Толчинский Алексей Юрьевич*

магистрант

*Солодко Алексей Андреевич*

магистрант

ФГБОУ ВО «Московский государственный  
технологический университет «СТАНКИН»

г. Москва

## **КРИТЕРИИ ВНЕДРЕНИЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**

*Аннотация:* в настоящей научной работе представлена одна из задач тестирования пользовательских интерфейсов с применением автоматизированных инструментов: определение критериев внедрения автоматизированного тестирования UI. Данная задача является ключевой при организации тестирования веб-сервисов, поскольку на ранних этапах необходимо определить область представления базовых элементов и архитектуры проекта тестирования.

*Ключевые слова:* тестирование, автоматизация тестирования, тестирование пользовательского интерфейса, разработка программного продукта, архитектура программного обеспечения, пользовательские сценарии, тестирование пользовательского опыта.

### *Введение*

Внедрение приёмочного автоматизированного тестирования является одной из ключевых задач команд тестирования. По мнению экспертов именно на начальных этапах разработки допускаются архитектурные ошибки в проектах тестирования, что усложняет их поддержку, дальнейшее масштабирование и развитие. Данная проблема часто оказывается темой обсуждения на международных конференциях по разработке программного продукта таких как Agile Acceptance Testing Days и SQA Days.

Таким образом тестирование на уровне пользовательского интерфейса часто ставится под вопрос, поскольку многие из опрошенных команд для такого тестирования выбирают уровень бизнес-функциональной логики и спустя некоторое время, примерно год активной разработки, сталкиваются с проблемой высоких денежных затрат на поддержание UI-тестов, по сравнению с низкой эффективностью выявления ошибок на исследуемом проекте. Встречались случаи остановки поддержки проектов автоматизированного тестирования и потери вложенных средств. Ниже представлены критерии внедрения UI-тестов, которые помогут избежать подобных проблем.

### *Уровни абстракции тестирования пользовательских интерфейсов*

Хорошей практикой при проектировании тестирования пользовательского интерфейса является разделение абстракции на три основных уровня.

1. Уровень бизнес логики: каждый проект можно разделить на набор действий доступный пользователю, например если речь идёт о интернет-магазине книг, то можно выявить такие действия как:

- авторизация;
- поиск книги;
- добавление книги в корзину;
- оформление заказа;
- оплата.

2. Уровень рабочего процесса или порядок действий выполняемых команд: данный уровень определяет последовательность действий, указанных в пункте 1, так, как сама последовательность может задавать бизнес логику больших итераций, например, если после выполненного заказа совершить следующий, то пользователю должна быть предложена скидка.

3. Технический уровень проекта: на данном уровне должна быть представлена работа с драйверами, например если в проекте используется selenium фреймворк, то он требует описания взаимодействия с конкретной операционной системой, скачивания определённой библиотеки для вашей версии браузера.

Также к данному уровню можно отнести функциональные особенности взаимодействия с элементами, например:

- дождаться появления элемента на странице;
- взаимодействия с фото;
- клик мыши по элементу;
- набор текста клавиатурой.

Данный подход позволяет повысить эффективность проекта тестирования за счёт того, что бизнес логика самого проекта меняется не так часто, как технические компоненты и методы реализации, что позволяет абстрагироваться от этого уровня и сосредоточиться на технической реализации, что повышает простоту и скорость поддержки продукта.

### *Список литературы*

1. Алиев Р. С. о. Информационная система электротехнических расчётов WebЭнерго (тезисы доклада). Семинар «Компьютерные средства подготовки персонала», ВВЦ, павильон «Электрификация», 16–20 октября 2000.
2. Алиев Р. С. о. Математическая модель оценки сложности разработки программного обеспечения // Вестник МГТУ «Станкин». – М., 2015. – №2 (33). – С. 93–97.
3. Документация программы SoapUI [Электронный ресурс]. – Режим доступа: <https://www.soapui.org/soapui-projects/soapui-projects.html> (дата обращения: 20.04.18).
4. Документация библиотеки Junit [Электронный ресурс]. – Режим доступа: <https://junit.org/junit5/docs/current/user-guide/> (дата обращения: 20.04.18).
5. Кузнецов В.Е. Представление в ЭВМ неформальных процедур. – М.: Наука, 1989. – 160 с.
6. Официальный сайт языка программирования Kotlin [Электронный ресурс]. – Режим доступа: <https://jetbrains.ru/products/kotlin/> (дата обращения: 01.05.18).
7. Тестирование. Фундаментальная теория [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/279535/> (дата обращения: 20.03.18).

8. Charitha Kankanamge. Web Services Testing with soapUI. Birmingham: Packt Publishing Ltd, 2012.
9. McConnell S. Best Practices. IEEE Software, Vol. 13, No. 6, December 1996.
10. SOAP UI и тестирование сервисов [Электронный ресурс]. – Режим доступа: <https://www.dataart.ru/news/soap-ui-i-testirovanie-servisov/> (дата обращения: 20.04.18).