

## АНАЛИЗ ПОДХОДОВ К АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

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

Мищенко А. В., Мельников А. В.

Пискун Г.А. – канд. техн. наук, доцент

Ошибочный выбор подхода к автоматизации тестирования может оказать влияние на весь процесс тестирования, поэтому в зависимости от типа приложения и целей автоматизации необходимо правильно выбрать нужный подход. В данной статье рассматриваются основные подходы к автоматизации тестирования и приводятся условия при котором следует выбирать тот или иной подход

Автоматизация тестирования производится за счет создания специальных программ, позволяющих при выполнении тестирования использовать экономить затраты денег и времени на проведение тестирования за счет возможности многократного выполнения одних и тех же тестов, без необходимости их повторной разработки.

В зависимости от типа тестируемой системы набор утилит может отличаться исходя из специфики приложения и требований к решению по тестированию. Например, будет отличаться набор утилит и подходы к автоматизации тестирования интернет и мобильных приложений.

В целом можно выделить следующие основные подходы: тестирование на уровне кода и тестирование пользовательского интерфейса. К первому типу относится, в частности, модульное тестирование. Ко второму – имитация действий пользователя – функциональное тестирование (с помощью специальных тестовых фреймворков) [1].

Тестирование на уровне кода используется при модульном и регрессионном тестировании (тестирование направленное на поиск ошибок в ранее протестированном коде, например, при обновлении версии кода приложения). Модульное тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по отдельности (модули программ, объекты, классы, функции и т.д.).

Тестирование через пользовательский интерфейс является наиболее распространенной формой автоматизации. Популярность такого вида тестирования объясняется двумя факторами: во-первых, приложение тестируется тем же способом, которым его будет использовать человек, во-вторых, можно тестировать приложение, не имея при этом доступа к исходному коду. В зависимости от специфики приложения, будут отличаться также сценарии пользователя. В целом выделяют четыре основных подхода в рамках тестирования через пользовательский интерфейс:

1) Утилиты записи и воспроизведения записывают действия тестирующего во время ручного тестирования. Они позволяют выполнять тесты без прямого участия человека в течение продолжительного времени, значительно увеличивая продуктивность и устраняя повторение однообразных действий во время ручного тестирования. В то же время, любое малое изменение тестируемого ПО требует перезаписи ручных тестов. Поэтому это первое поколение инструментов не эффективно и не масштабируемо. В настоящий момент практически не используется;

2) Написание сценария — форма программирования на языках, специально разработанных для автоматизации тестирования ПО — смягчает многие проблемы инструментов записи и воспроизведения. Но разработкой занимаются программисты высокого уровня, которые работают отдельно от тестирующих, непосредственно запускающих тесты. К тому же скрипты более всего подходят для тестирования пользовательского интерфейса и не могут быть внедренными, пакетными или вообще каким-либо образом объединены в систему. Наконец, изменения в тестируемом ПО требуют сложных изменений в соответствующих скриптах, и поддержка все возрастающей библиотеки тестирующих скриптов становится в конце концов непреодолимой задачей;

3) Управляемое данными тестирование — методология, которая используется в автоматизации тестирования. Особенностью является то, что тестовые скрипты выполняются и верифицируются на основе данных, которые хранятся в центральном хранилище данных или базе данных. Роль базы данных могут выполнять базы данных, csv или xls файлы и т. д. Следует выбирать данный подход, если необходимо проводить тестирование на большом количестве данных, когда необходимо проверить одно и то же поведение или функциональность системы для различных условий;

4) Тестирование по ключевым словам – автоматизация подразумевает разделение процесса создания тест-кейсов на 2 этапа: этап планирования и этап реализации. В этом случае конечный тест представляет собой не программный код, а описание последовательности действий с их параметрами (например, «завести в базе данных пользователя с логином ХХХ и паролем УУУ»). При этом фреймворк отвечает за непосредственную реализацию ключевых слов (действий), а дизайнеру тестов достаточно иметь представление о всём наборе действий, реализованных во

фреймворке. Это даёт возможность создавать тесты людям, не имеющим навыков программирования.

В целом возможно также совмещать несколько подходов в одном с целью упрощения процесса создания автоматических тестов. Например, в [2] описан инструмент для записи действий тестировщика, при котором автоматически генерируются сценарий-подобные автоматические тест-кейсы.

Подход к автоматизации следует выбирать исходя из требований заказчика и специфики приложения.

**Список использованных источников:**

1. Автоматизированное тестирование [Электрон. ресурс] – Электрон. дан. Режим доступа: [https://en.wikipedia.org/wiki/Test\\_automation](https://en.wikipedia.org/wiki/Test_automation)
2. Фреймворк автоматизации тестирования пользовательского интерфейса : материалы 54 науч. конф. аспирантов, магистрантов и студентов, Минск, 23-27 апреля 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: А.Л. аднёнок [и др.]. – Минск : БГУИР, 2018. – 79 с.