

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ФУНКЦИОНАЛЬНОСТИ И ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПРОГРАММНОГО ПРИЛОЖЕНИЯ

Д.В. Ялакова

Научный руководитель: Ю.А Мартынова, ассистент каф. АИКС ИК ТПУ

(г. Томск, Томский политехнический университет)

e-mail: yalakova_dasha@yahoo.com

TEST AUTOMATION OF FUNCTIONALITY AND BEHAVIOUR OF USER INTERFACE SOFTWARE APPLICATIONS

D.V. Yalakova

(Tomsk, Tomsk Polytechnic University)

Abstract. This work is devoted to the analysis of test automation process. Practical application benefits and problems that may be encountered in the implementation of test automation at the enterprise. The article also discusses the most popular software tools to automate testing of the functionality and behavior of the user interface of software applications

Keywords: test automation, functional testing, user interface, test builder, tester.

Введение. Одной из неизбежных проблем, с которой сталкиваются предприятия, занимающиеся разработкой и внедрением программных продуктов, на сегодняшний день, является проблема возникновения различных ошибок в работе приложения. Они могут возникнуть как на этапе разработки, так и при использовании программного продукта пользователем. Безусловно, каждый руководитель стремится к тому, чтобы минимизировать возможность возникновения ошибок в работе созданного программного продукта и тем самым улучшить его качество.

На сегодняшний день, все чаще принимается решение о том, что необходимо принимать в штат специалистов, которые бы занимались тестированием продуктов, разрабатываемых программистами. Однако данные специалисты, чаще всего занимаются так называемым «ручным» тестированием.

Основная цель автоматизации тестирования – сокращение расходов на испытания программы после ее модернизации и повышение качества испытаний, что гарантирует надежность продукта.[1] Периодически повторяемые однотипные проверки отнимают много времени в цикле разработки. Автоматизация сокращает этап тестирования и высвобождает главный ресурс компании – рабочее время специалистов.

Закодированные тесты интерфейса пользователя. Автоматизированные тесты интерфейса пользователя(ИП) приложений называются закодированными тестами ИП (*coded user interface tests - CUIT*). Эти тесты обеспечивают функциональное тестирование элементов управления ИП. Они позволяют убедиться, что все приложение, в том числе интерфейс, работает правильно.

Чтобы создать закодированный тест ИП, необходимо запустить построитель закодированных тестов (*test builder*), и вручную выполнить тест. Можно указать значения, которые должны быть в определенных полях. Построитель тестов ИП записывает действия пользователя и создает на их основе код. После создания теста можно изменить последовательность действий в специальном редакторе.

Построитель закодированных тестов ИП и редактор упрощают создание и изменение закодированных тестов ИП, даже если специалист разбирается больше в тестировании, чем в программировании. Но если разработчик захочет расширить возможности теста, он легко сможет скопировать и адаптировать код, потому что он структурирован.

При запуске теста, движок закодированного теста ИП должен найти каждый элемент управления, который используют записанные действия. Это достигается с помощью прохода по дереву представления, используя имена элементов user interface. Если пользовательский интерфейс был изменен, тесты могут не работать из-за того, что не смогут найти элементы.

Положительные и отрицательные стороны автоматизации тестирования.

С автоматизацией тестирования связано много неверных представлений. Чтобы избежать неэффективного применения автоматизации, необходимо избегать ее недостатков и максимально использовать преимущества.

Преимущества автоматизации тестирования:

- Повторяемость – написанные тесты всегда будут выполняться однообразно, то есть исключен «человеческий фактор».
- Быстрое выполнение – автоматизированному скрипту не нужно сверяться с инструкциями и документациями, что экономит время выполнения.
- Меньшие затраты на поддержку – когда автоматические скрипты уже написаны, на их поддержку и анализ результатов требуется, как правило, меньше времени чем на проведение того же объема тестирования вручную.
- Отчеты – автоматически рассылаемые и сохраняемые отчеты о результатах тестирования.
- Выполнение без вмешательства – во время выполнения тестов инженер-тестировщик может заниматься другими полезными делами, или тесты могут выполняться в нерабочее время.

Недостатки автоматизации тестирования:

- Повторяемость – это одновременно является и недостатком, так как тестировщик, выполняя тест вручную, может обратить внимание на некоторые детали и, проведя несколько дополнительных операций, найти дефект. Скрипт этого сделать не может.
- Затраты на поддержку – в случае автоматизированных тестов они меньше, чем затраты на ручное тестирование того же функционала, но они все же есть. Чем чаще изменяется приложение, тем они выше.
- Большие затраты на разработку – разработка автоматизированных тестов это сложный процесс, так как фактически идет разработка приложения, которое тестирует другое приложение. В сложных автоматизированных тестах также есть фреймворки, утилиты, библиотеки и прочее. Все это нужно тестировать и отлаживать, а это требует времени.
- Стоимость инструмента для автоматизации – в случае если используется лицензионное программное обеспечение, его стоимость может быть достаточно высока. Свободно распространяемые инструменты как правило отличаются более скромным функционалом и меньшим удобством работы.
- Пропуск мелких ошибок – автоматический скрипт может пропускать мелкие ошибки на проверку которых он не запрограммирован. Это могут быть неточности в позиционировании окон, ошибки в надписях, которые не проверяются, ошибки контролов и форм с которыми не осуществляется взаимодействие во время выполнения скрипта.

Обзор программных средств.

Рассмотрим несколько наиболее популярных программных инструментов, сравнив их по нескольким основным критериям.

Таблица 1. Критерии сравнения программных средств

	<i>Ranorex</i> (компания <i>Ranorex</i>)	<i>Coded UI</i> (компания <i>Microsoft</i>)	<i>Unified Functional Testing</i> (компания <i>Hewlett Packard Enterprise</i>)
Стоимость лицензии	<i>Runtime</i> : 49 095 RUB; <i>Premium (Single User License)</i> : 141 592 RUB;	<i>Visual Studio Enterprise</i> 2015:	В зависимости от комплектации >10000 USD, уточняется у продавца

	<i>Premium (Shared License):</i> 248321 RUB;[2]	427 857 RUB;[3]	компания
Поддерживаемые языки тестовых скриптов	<i>C# и VB.NET</i>	<i>C#</i>	<i>Java и VBasic.NET</i>
Поддержка динамически генерируемых контролов	+	+	+
Простота изучения и использования тестировщиками	+	+	+
Запись тестов с помощью встроенного рекордера	+	+	+

Можно заметить, что все эти инструменты распространяются на коммерческой основе, в целом, у них много общих функций. Можно сказать, что, как программы специально предназначенные для автоматизации тестирования *Ranorex* и *Unified Functional Testing*, имеют довольно удобный интерфейс и обладают хорошим набором функционала, так же они имеют пробную версию, чтобы тестировщик мог попробовать поработать с продуктом и принять решение.

Заключение.

Автоматизация тестирования несомненно может помочь при разработке программных приложений. С ее помощью можно освободить некоторое время, затрачиваемое на ручное тестирование функциональности пользовательского интерфейса, сократить затраты на поддержку тестов, однако все это будет доступно лишь тогда, когда четко определены задачи тестирования, когда тестировщики понимают зачем на предприятии вводится автоматизированное тестирование и какой это даст результат.

Перед тем, как внедрять автоматизированное тестирование необходимо провести совокупный анализ разрабатываемого программного продукта и определить стоит ли это делать. Потому что, в некоторых случаях, необдуманные попытки внедрения автоматизации тестирования могут привести к существенным как временным, так и денежным затратам предприятия, но не принести никакого результата.

ЛИТЕРАТУРА

1. Автоматизация тестирования [Электронный ресурс]. – режим доступа: <http://software-testing.ru/library/around-testing/processes/437-learn-to-save>
2. Ranorex [Электронный ресурс]. – режим доступа: <http://www.ranorex.com/purchase/buy-now.html>– 30.03.16
3. Visual Studio [Электронный ресурс]. – режим доступа: <https://www.visualstudio.com/en-us/products/how-to-buy-vs.aspx>– 30.03.16