

Методы оценки внедрения автоматизированного тестирования программного обеспечения

Тимин Сергей Иванович

*Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых
магистрант*

Аннотация

В данной статье рассматривается вопрос автоматического тестирования программного обеспечения, который является одним из основных в методах управления качеством. Более распространен метод ручного тестирования, но такой метод иногда не может решить все задачи обеспечения качества, именно поэтому все чаще стал использоваться метод автоматизированного тестирования ПО.

Ключевые слова: автоматизированное тестирование, программное обеспечение, инструменты автоматизации, контроль качества, информационные технологии.

Methods for evaluating the implementation of automated software testing

Timin Sergey Ivanovich

*Vladimir State University named after Alexander Grigorievich and Nikolai Grigorievich Stoletovs
undergraduate*

Abstract

This article discusses the issue of automatic software testing, which is one of the main methods of quality management. Manual testing is more common, but this method sometimes cannot solve all the problems of quality assurance, which is why the method of automated software testing has become increasingly used.

Keywords: automated testing, software, automation tools, quality control, information technology.

Программирование на современном этапе развития, как и всякая деятельность человека, иногда происходит с ошибками. Ранние этапы развития процесса отладки компьютерных программ были достаточно примитивными, сейчас же ситуация изменилась в корне и порой человек сталкивается со сложными проблемами в тестировании программного обеспечения.

Процесс разработки программных продуктов постоянно подвергается изменениям и обновлениям. Такие доработки обусловлены тем фактом, что необходимо исправлять существующие ошибки, которые выявляются в

процессе их работы, либо появляется возможность внести дополнительные полезные изменения в программу. Все это влечет за собой появление новых функциональных требований, которые изначально не рассматривались вообще.

Инструменты автоматизации тестирования программного обеспечения могут быть очень полезны для организации, занимающейся тестированием программного обеспечения. Цель этих инструментов – увеличение охвата тестированием при одновременном снижении затрат на фактическую функцию тестирования, а также сокращение времени, необходимого для тестирования и требующего меньше ресурсов. Теоретически полная «автоматизация» означает, что никакого ручного вмешательства не требуется – такие действия включают в себя запуск и остановку скриптов, создание новых скриптов и их обслуживание по мере изменения потребностей проекта. Но хотя «автоматизация» звучит как отличная функция для инструментов тестирования программного обеспечения, сегодня на рынке нет такого инструмента, который был бы полностью автоматизирован на 100%.

Специалисты по обеспечению качества программного обеспечения могут оказаться на грани между финансовыми затратами, которые требуются для вложений в инструмент тестирования ПО, и возможностями инструмента тестирования ПО. Решение часто сводится либо к инструменту тестирования программного обеспечения с открытым исходным кодом, либо к приложению для тестирования программного обеспечения от поставщика. Уверенность в том, что организация покупает подходящий инструмент автоматизации тестирования программного обеспечения для работы, не может быть оценена.

Прежде чем что-либо делать, очень важно составить стратегию для поиска нужного инструмента. Важно изучить и проанализировать потребности компании и выяснить, какие проблемы может решить этот инструмент. Для этого нужно составить список задач, которые инструмент должен выполнять, а затем описать, как каждая задача улучшит соответствующую бизнес-функцию. Подумать нужно еще и о том, насколько большой или маленькой будет пользовательская база инструмента, и помнить об их требованиях при согласовании функции инструмента со стратегией компании.

После того, как стратегия ясна, нужно быть готовым к выбору инструмента. Можно выделить девять основных методов, которые следует учитывать при оценке внедрения автоматизированного тестирования ПО:

1. Долговечность обслуживания. В первую очередь, стоит задать вопрос: как долго поставщик сможет поддерживать приложение инструмента тестирования программного обеспечения? Нужно понять, достаточно ли стабильна компания для продолжения обслуживания, если инвестировать достаточно времени, денег и/или ресурсов. Приложение может предлагаться по сниженной цене (или даже бесплатно), но должно быть достаточно

персонала, чтобы довести до конца этот процесс и поддерживать продукт на должном уровне.

Если это приложение с открытым исходным кодом, и команда поддержки не справилась с работой из-за нехватки ресурсов или других проблем, то это может повлиять и на способность компании управлять продуктом. Важно понимать, что, если обслуживание продукта происходит самостоятельно, также необходимо протестировать его на месте, что может потребовать больше времени и рабочей силы, а также открыть окно для ошибок. Программное обеспечение с открытым исходным кодом отлично подходит для среды с длинными графиками проектов или там, где существуют высокоуровневые исследования и разработки. Они нереализуемы в средах, где все, что нужно выполнять, – это быстрые автоматизированные задачи.

2. Обновления. Здесь важным аспектом выступает вопрос о том, будет ли приложение с открытым исходным кодом или приложение от поставщика регулярно сможет обновляться новыми версиями программного обеспечения. Также важно понимать, что делать, если в старом программном обеспечении есть «ошибка». По мере изменения системной среды (операционных систем, плагинов, браузеров и т. д.) меняется и потребность в обновлениях. Обновления следует проводить еженедельно или, по крайней мере, ежеквартально. Это важный шаг к тому, чтобы группа обслуживания инструмента тестирования программного обеспечения получила отличную внутреннюю поддержку.

3. Удобство использования. Крайне важным моментом является факт понимания, что перед нами – разработки макета приложения с открытым исходным кодом или приложения на основе платформы поставщика, – какой из вариантов для работы с любой заданной функцией потребует меньше действий. Важно также понять, будет ли его легко использовать как новичку, так и опытному тестировщику. Макетом приложений с открытым исходным кодом может быть сложно управлять, и иногда требуется глубокое знание языка компьютерного программирования, такого как C+ или Java. И многие новые приложения с открытым исходным кодом выходят с макетами, сходными с их аналогами от поставщиков. Следовательно, необходимо определить, сколько времени потребуется для внедрения инструмента, и для какого специалиста он предназначен. В последнее время инструменты тестирования программного обеспечения стали более удобными для пользователя, но пользователь по-прежнему должен уметь полностью понимать тонкости стратегии тестирования и тестируемого приложения.

4. Совместимость и интеграция. Следующий метод позволяет ответить на такие вопросы: есть ли у приложения с открытым исходным кодом или у поставщика возможность интеграции с другими инструментами, такими как инструмент отслеживания дефектов и совместим ли этот инструмент с окружением компании.

Функциональные возможности инструмента увеличиваются, если у него есть встроенная возможность интеграции с другими приложениями и базами данных. Нужно убедиться, что инструмент автоматизации тестирования программного обеспечения совместим с приложением, используемым для автоматизации и тестирования, клиентской операционной системой, в которой он работает, веб-браузерами и другими элементами, такими как контроль изменений. Кроме того, совместимость инструмента является ключевым показателем того, насколько он будет доступен при выполнении скриптов. Крайне важно выяснить, можно ли запустить инструмент с любого компьютера или сначала нужно установить клиент автоматизации, и то, и другое будет определять стоимость лицензии.

5. Руководства. Важным моментом, который не стоит упускать из вида, является руководство по эксплуатации, которое поможет поддерживать его на протяжении всего использования. Всегда полезно иметь какой-то справочный материал для поддержки конечного пользователя программного обеспечения, особенно когда речь идет о выполнении уникальной функции, специфичной для этого инструмента. Если справочные документы недоступны – как в случае с некоторыми приложениями с открытым исходным кодом – следует подумать о создании своих собственных. Справочные руководства, которые компактны, но все же указывают на полную функциональность инструмента, являются большим преимуществом для конечного пользователя, особенно если исходное руководство по продукту является громоздким для обычного пользователя.

6. Служба поддержки. Предоставляется ли соответствующая поддержка клиентов продуктом или поставщиком, какой у них график работы, их скорость отклика, взимается ли за это дополнительная плата – эти вопросы стоит решить без отлагательств.

Часто, когда покупается инструмент для тестирования, организация действительно платит за маркетинг продукта, исследования, разработки и поддержку. Так что нужно иметь в виду, что если платить много денег за инструмент, он должен быть первоклассным или, в конце концов, «вы заплатите за то, что получите». Узнать о том, включает ли в себя продукт поддержку клиентов – очень важно. Некоторые приложения с открытым исходным кодом могут утверждать, что у них есть поддержка клиентов, но мелким шрифтом иногда написано самое главное, чтобы узнать, предоставляется ли поддержка на самом деле, важно узнать – поддержка только онлайн или действительно можно поговорить с живым человеком по телефону. Еще нужно выяснить, где физически находится служба поддержки. Если возникла срочная проблема в 7 часов утра, а организация находится в восточном регионе России, не стоит ожидать немедленного перезвона из центрального района обслуживания клиентов продавца, если он расположен в западном регионе. Некоторые поставщики могут даже иметь офшорные ресурсы, об этом не стоит забывать.

7. Интернет-сообщества пользователей – очень полезная вещь, которая предлагает онлайн-сообщество для пользователей, чтобы делиться информацией с другими пользователями.

Часто бывает, что возникает вопрос, на который кто-то уже знает ответ. Интернет-сообщество программного обеспечения – прекрасный ресурс, который можно использовать, когда нужно узнать, есть ли у других пользователей такие же проблемы / симпатии / антипатии к продукту. Интернет-сообщества – это дополнение к возможностям инструмента. Они не только помогают пользователям обмениваться информацией друг с другом, но и повышают лояльность к бренду и укрепляют доверие среди существующих и потенциальных пользователей. Исследуя данный вопрос, желательно узнать, есть ли у данного инструмента такое сообщество, остерегаться нужно множества неофициальных сообществ сотрудничества и веб-сайтов – для получения наиболее точной информации перейти нужно прямо на официальную веб-страницу онлайн-сообщества инструмента.

8. «Центр передового опыта в тестировании». Кто будет управлять инструментами автоматизации тестирования программного обеспечения внутри компании – вопрос, который не стоит забывать. Будет ли создана централизованная группа, которая будет управлять всеми инструментами автоматизации тестирования программного обеспечения отдела, или каждый отдел будет управлять инструментом самостоятельно – вот, что важно решить сразу и на месте.

Метод, с помощью которого управляют инструментом, будет способствовать его успеху или наоборот. Поддержание центра передового опыта в области тестирования имеет смысл – особенно с инструментами автоматизации тестирования программного обеспечения, потому что он действует как центральный узел департамента, где хранится вся информация об инструменте; от управления и установки до знаний и использования. Центр передового опыта помогает каждому отделу, который использует инструмент тестирования программного обеспечения внутри отделов. Иногда в небольших группах могут быть независимые тестировщики, которые перемещаются в другие отделы и обратно. В больших группах могут быть постоянные тестировщики, которые работают полный рабочий день только в одном отделе. Тем не менее, Центр передового опыта в области тестирования является опорой, а в некоторых случаях он может даже выступать в качестве заместителя тестировщиков отдела.

9. Продажи. Покупатели должны быть осторожны, ведь инструмент автоматизации тестирования программного обеспечения, который они собираются приобрести, включает в себя гораздо больше, чем может быть представлено торговым представителем. «Попробуйте, прежде чем покупать» – хороший лозунг, чтобы прочувствовать технические аспекты инструмента и понять, насколько он будет соответствовать потребностям или нет – это вещи, о которых торговый представитель, скорее всего, никогда не

расскажет. Нужно убедиться, что происходит установка инструмента в тестовой среде, использовать его часто, а затем составить список вопросов для поставщика и его технической группы. Помнить важно о множестве «услуг», которые продвигает продавец, и сделать шаг назад, проанализировав, насколько это рентабельно и соответствует ли он целям бизнеса. Команды продаж могут использовать инструмент автоматизации тестирования программного обеспечения как трамплин для предоставления дополнительных, часто ненужных услуг своим потенциальным клиентам.

В некоторых случаях может быть лучше сравнить несколько систем одновременно и провести собственный анализ. Каждый инструмент может действовать по-своему, в зависимости от того, как ваша компания проводит тестирование. Не стоит ждать, что поставщик или независимые агентства проведут сравнение инструментов, чтобы сэкономить время, и не следует полагаться на них как на экспертов. Как конечный пользователь, именно человек является экспертом в своей отрасли, и нужно найти время, чтобы понять, подходит ли приложение для конкретной среды. Если ускорить этот процесс без какого-либо направления или плана, можно застрять с инструментом, который не отвечает ни потребностям компании, ни пользователям.

Как упоминалось ранее, очень важно вовлечь группу конечных пользователей в процесс принятия решения о том, приобретать ли инструмент и какие-либо дополнительные услуги. Корпоративная политика иногда играет большую роль в процессе принятия решений, и конечные пользователи застревают с инструментом, который не работает для них и корпоративных целей. Так что нельзя дать себя обмануть отделу продаж. В конце концов, можно и проиграть.

Просто примечание – можно сделать снимки экрана приложения во время использования временной демонстрации, создав документ, в котором сравниваются функциональные возможности каждого снимка экрана. По истечении срока действия краткосрочной лицензии может понадобиться эта документация, чтобы помочь осмыслить положительные и отрицательные стороны инструмента в процессе принятия окончательного решения.

Авансовые и косвенные затраты. «Стоимость» означает не только цену инструмента, но и время установки, знания о наращивании мощности и другие косвенные затраты. Некоторые из этих факторов уже были указаны выше, например, обучение, скорость наращивания, ресурсы, документация, установка, совместимость и руководства. Иногда следует сделать шаг назад, чтобы учесть все затраты, особенно косвенные, потому что иногда общая стоимость инструмента на основе поставщика может быть дешевле, чем приложение с открытым исходным кодом, и наоборот, в зависимости от того, насколько хорошо открыт исходный продукт. В связи с этим косвенные затраты могут вызвать серьезную проблему и часто увеличиться, когда этого меньше всего ожидаешь. Например, может случиться такое, когда приложения были приобретены, но на их установку или среду не был

выделен бюджет, что приводило к тому, что инструмент работал неадекватно.

Данные девять методов следует иметь в качестве руководства при рассмотрении вариантов приобретения средств автоматизации тестирования программного обеспечения. Прежде чем приступить к процессу, нужно убедиться, что полностью понимается цель компании – это снизит риск выбора инструмента, который отвечает лишь части потребностей бизнеса. Помнить нужно о том, что «качество» – это основа жизненного цикла разработки программного обеспечения. Если приобретение инструмента автоматизации тестирования программного обеспечения будет успешным и безупречным, есть большая вероятность, что это вызовет эффект домино и проникнет в другие отделы. Итак, не стоит торопиться. Чем лучше понимание того, что покупается, тем увереннее будет окончательное решение.

Автоматизация тестирования не может рассматриваться как универсальное средство в борьбе за качество конечного продукта, потому что кроме явных преимуществ (таких, как повышение независимости экспертизы за счёт исключения человеческого фактора, например), у автоматизации тестирования есть и ряд недостатков. Автоматизация тестирования не всегда необходима, рационально обоснована или экономически оправданна. Только детальный анализ может помочь сделать осознанный выбор, а также выявить плюсы и минусы выбранного подхода. Методы, рассмотренные в данной статье, могут помочь избавиться от ложных ожиданий и составить более точное представление о результатах автоматизации.

Библиографический список

1. Галимова Е. Ю., Коваленко А. Н. Метод выбора между ручным и автоматизированным тестированием, основанный на свойствах программного продукта // *Advanced Engineering Research*. 2016. № 4 (87). С. 134-139.
2. Гребенюк В. М. Оценка целесообразности внедрения автоматизированного тестирования // *Вестник евразийской науки*. 2013. № 1 (14). URL: <https://cyberleninka.ru/article/n/otsenka-tselesoobraznosti-vnedreniya-avtomatizirovannogo-testirovaniya>
3. Кудрявцева Е. Ю. Автоматизированное тестирование веб-интерфейсов // *ГИАБ*. 2014. № 6. С. 354-356.
4. Пришутова Е. А., Васильева Д. В. Автоматизация тестирования web-приложений при разработке. Плюсы и минусы данного подхода // *Наука, техника и образование*, 2017. № 4 (34). URL: [https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaiya-web-prilozheniy-pri-razrabotke-plyusy-i-minusy-dannogo-podhoda](https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-web-prilozheniy-pri-razrabotke-plyusy-i-minusy-dannogo-podhoda)