

НЕИДЕАЛЬНОСТЬ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ

Лаппо К. Д.

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

Лагутин А.Е. – канд. техн. наук, доцент

Статья рассматривает понятие автоматизированного тестирования, его основных задач на сегодняшний день. Статья выделяет негативные стороны и подводные камни применения автоматизированного тестирования на проектах. Также приводятся советы, позволяющие снизить негативные эффекты использования автоматизации. Статья подходит для широкого круга читателей, вовлеченных в индустрию разработки и тестирования программного обеспечения.

Под автоматизированным тестированием подразумевают использование инструментов автоматизации для выполнения набора специально подготовленных тестовых сценариев. Такое тестирование резко отличается от традиционного ручного тестирования, выполняемого человеком, сидящим перед компьютером и тщательно выполняющим этапы тестирования. [1]

Основной причиной ввода автоматизированного тестирования на проектах в основной массе является желание сократить количество времени, затрачиваемое специалистами ручного тестирования на проведение регрессионного тестирования разрабатываемого продукта. Также зачастую приводятся следующие доводы в пользу автоматизированного тестирования: увеличение затрат по стоимости и времени полного тестирования приложения [1]; присутствие человеческого фактора при проведении тестирования (усталость, жизненные ситуации, выполнение однообразных действий); возможность проведения однотипных тестов при изменении начальных параметров, резко не меняющих работу приложения и, как следствие, не изменяющих сценарий теста; возможность проведения тестирования небольшого функционала по требованию и на постоянной(в идеале - ежедневной) основе.

При этом следует понимать, что и само по себе автоматизированное тестирование имеет собственные издержки. Автоматизированное тестирование подразумевает заблаговременное планирование, в котором происходит взаимодействие специалистов автоматизированного тестирования почти со всеми участниками разработки ПО. При раннем планировании можно заранее наметить как будет проводиться тестирование приложения, в каком объеме, по каким сценариям, которые скорее всего придется адаптировать под нужды автоматизации; заранее необходимо понять, что из себя будет представлять продукт в плане функциональности, используемой инфраструктуры; необходимо определиться с выбором инструментов тестирования, будь то написанный внутри компании фреймворк или сторонние утилиты и технологии; необходимо подготовить площадку для предоставления отчетности и проведения анализа результатов тестирования; также необходимо понимать где и как будет производиться само автоматизированное тестирование – то есть рассматривается инфраструктура и функциональность самих систем тестирования и написанных сценариев.

Сегодня существует множество методологий разработки ПО, одна из самых популярных сегодня – гибкая (agile) разработка. В любом случае, выбранную методологию необходимо будет учитывать при планировании применения автоматизации на проекте, так как создание, развитие и поддержка продукта автоматизации занимают определенное время и ресурсы, и может пройти несколько месяцев и кварталов, прежде чем автоматизация будет приносить очевидные выгоды и предоставлять полезную информацию о текущем состоянии проекта.

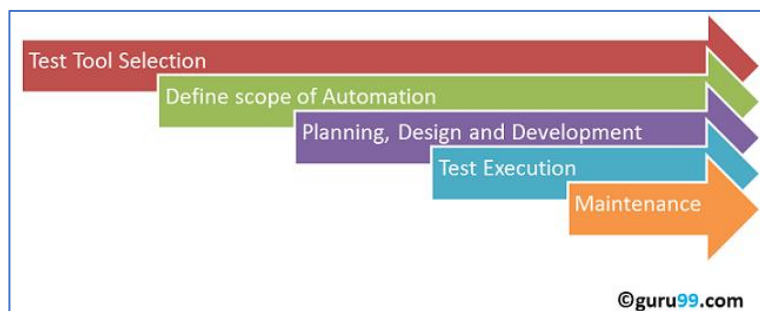


Рисунок 1 – Типичный шаги развития автоматизации тестирования на проекте. [1]

Под продуктом автоматизации стоит понимать набор выбранных заранее техник, технологий, прикладных программ и решений автоматизации и написанные на их основе и с их

помощью сценарии тестов. Как видно из рисунка 1, продукт автоматизации проходит несколько стадий начиная от подготовки (выбор инструмента, и тд) и «заканчивая» постоянной и продолжающейся поддержкой, расширением продукта.

Также время стоит отметить самый неприятный аспект автоматизированного тестирования – его малую гибкость и одновременно полную зависимость от разрабатываемого продукта, процессов разработки и контроля качества. Как было упомянуто выше сегодня популярным подходом при разработке приложений является гибкая разработка, при которой продукт за небольшой промежуток времени может существенно меняться. На это накладывається неидеальность самих процессов разработки и тестирования на проекте. Вследствие изменения функциональности проекта меняются и пользовательский опыт, и сценарии использования конечного продукта, что делает существующие сценарии тестирования устаревшими. Внесение изменений в сценарии тестирования должно происходить одновременно и специалистами ручного тестирования к спецификациям, и специалистами автоматизированного тестирования в продукт автоматизации, а если точнее – то конкретно в реализацию сценариев (код, скрипты, специальные файлы сценариев). Конечно, зачастую это все может совмещаться специалистом по автоматизированному тестированию, однако все же это занимает определенное время, а конечный продукт за это время может еще раз измениться.

Данная ситуация усугубляется большой частотой вносимых изменений, степенью их «переломности» (те являются ли они “breaking change” с точки зрения существующих сценариев использования), и временем их появления в продукте. Значительная часть времени в таком случае относится на подгонку и исправление продукта автоматизации под изменившиеся условия, в то время как уменьшается время на его развитие, расширение и увеличение покрытия тестирования. Еще стоит отметить, что порой само приложения становится «источником истины» вместо общей документации проекта, на которую опираются участники процесса разработки. В данном случае тестирование в целом, как и автоматизация должны подстраиваться под приложение.

К сожалению, ценность подобной автоматизации резко падает, в виду того, что результаты тестирования на самом деле не сообщают о состоянии качества конечного продукта, а только о соответствии реализованных автоматизированных тестовых сценариев работе конечного продукта. С течением времени продукт автоматизации становится действительно еще одним продуктом производимым в процессе разработки основного продукта, на который было затрачено немало усилий и времени и который, в тоже время, не приносит явную и ощутимую пользу, так как поддержка продукта автоматизации сводится к его приведению в рабочее состояние, как если бы вы на постоянной занимались починкой автомобильного средства и большую часть времени проводили на СТО.

Несмотря на такие серьезные негативные стороны, есть способы их минимизации. Существуют специальные метрики, которые позволяют узнать полезность и эффективность автоматизации на проекте (например, ROI). С периодичностью необходимо проводить анализ автоматизации тестирования на проекте и понимать, приносит ли или будет ли приносить эффект использование автоматизации. Также автоматизацию тестирования следует применять на серьезных, зрелых проектах с относительно неизменным базовым функционалом, который необходимо поддерживать и над которым необходимо проводить анализ качества. В небольших проектах, проектах-прототипах автоматизацию необходимо осторожно, либо вообще не использовать исходя из целей конкретного проекта – нет смысла тратить ресурсы на проект, который будет претерпевать множество итераций и изменений в базовых сценариях использования.

Еще один способ сделать автоматизацию полезной для проекта – это выбирать наиболее важные и критические сценарии для проверки определенного функционала конечного продукта. В таком случае сам проект автоматизации проще поддерживать и развивать. Но все же придется опытным путем искать «золотую середину» - соотношение количества автоматизированных сценариев и времени на их поддержку к покрытию (отношение тестируемой части функционала ко всему функционалу в целом).

Список использованных источников:

1. guru99 [Электронный ресурс] – AUTOMATION TESTING Tutorial: What is, Process, Benefits & Tools – Режим доступа: <https://www.guru99.com/automation-testing.html> Дата доступа: 03.02.2020