

ОСНОВНЫЕ ЭТАПЫ ОРГАНИЗАЦИИ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПРИЛОЖЕНИЙ

Ишманов Айдар Флюрович

магистрант, кафедра автоматизации технологических процессов Уфимского государственного технического авиационного университета, РФ, г. Уфа

Сафин Рустам Ильдарович

магистрант, кафедра автоматизации технологических процессов Уфимского государственного технического авиационного университета, РФ, г. Уфа

Хасанов Наиль Салаватович

магистрант, кафедра автоматизации технологических процессов Уфимского государственного технического авиационного университета, РФ, г. Уфа

Процесс автоматизации тестирования – это интеллектуальное творчество ИТ-специалистов высокой квалификации, но для достижения поставленных целей его тоже необходимо вести планомерно. На каждом этапе специалисты выбирают правильную стратегию испытаний при проверке качества исследуемого объекта. Основные этапы автоматизации тестирования:

Подготовка – выбор бизнес-операций, подлежащих автоматизации тестирования, определение требований к Системе Автоматизированного Функционального Тестирования (САФТ), согласование проектных сроков, выбор инструмента автоматизации, оценка возможных рисков.

Проведение – производится запуск автоматизированных тестов и проведение регрессионного автоматизированного тестирования, если необходимо.

Отчет – составляется итоговый документ с результатами тестирования, который содержит обнаруженные дефекты, отклонения от нормативов и предложения по улучшению системы. Создаются руководство пользователя и инструкции по настройке и сопровождению системы автоматизированного функционального тестирования.

Инструменты тестирования позволяют делать выбор между подходами: работать по Behaviour Driven Development или просто писать сценарные тесты для их автоматического запуска и проверки поведения системы. Разница примерно такая же как между Test Driven Development и покрытием юнит тестами. TDD говорит о том, что модульный тест (юнит тест), которому должен удовлетворять модуль (объект, метод) пишется до того, как написана реализация программного модуля. Максимум, что может быть описано одновременно с тестом – это интерфейс – сигнатура метода, интерфейс объекта. Первый запуск теста обязан "упасть" – подтвердить, что текущая реализация не работает.

Суть в том, что мы таким образом описываем, что хотим получить от модуля, функции или процедуры. Описываем входящие параметры и ожидаемые результаты, в том числе при граничных условиях. Мы сосредотачиваемся на роли модуля и правильных результатах, а не наоборот, подгоняем входящие параметры под уже написанный код, боясь его изменений. Сам же модуль на время написания тестов остается черным ящиком – автоматом, который затем эволюционирует, чтобы удовлетворить всем тестам.

Покрытие тестами без применения TDD – это наоборот, написание модульных тестов на уже реализованный функционал. Покрытие тестами делается для безопасного рефакторинга модуля и регрессионного тестирования. Того же самого результата позволяет добиться TDD, но TDD это другая концепция. И в том, и в другом случае количество кода, создаваемого программистами, сильно увеличивается, стоимости разработки растут в начале, но компенсируется сокращением сроков на развитие, отладку, тестирование, рефакторинг и поддержку. Аналогично BDD отличается от применения сценарных тестов для проверки уже реализованного функционала. BDD говорит о том, что фича-файл и содержащиеся в нем сценарии являются спецификацией на часть программного продукта. И эта спецификация пишется до реализации программного продукта. Но в отличие от TDD, где речь идет лишь о коде и взаимодействии программиста и кода, в BDD речь идет о поведении пользователя в системе и о постановке задачи разработчику бизнес-аналитиком. Или о совместной проработке спецификации бизнес аналитиком и программистом. При этом сценарии, описывающие поведение пользователя и смотрящие на систему глазами пользователей, опираются не на интерфейс метода, не на сигнатуру функции, а на пользовательский графический интерфейс приложения.

Не смотря на все перечисленные преимущества автоматизированного тестирования, создание подобной системы является нетривиальной и очень специфической задачей. В зависимости от функциональности разрабатываемого ПО, языков и средств программирования, задача организации системы кардинально меняется, приобретая свои персональные проблемы.

Список литературы:

1. Автоматизированное тестирование, автоматизация тестирования приложений. URL: <https://daglab.ru/avtomatizirovannoe-testirovanie-avtomatizacija-testirovanija-prilozhenij/> (дата обращения: 20.12.2021).
2. Автоматизация тестирования. URL: <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/ruchnoe-testirovanie/avtomatizatsiia-testirovaniia> (дата обращения: 21.12.2021).
3. Разработка системы автоматизированного тестирования. URL: https://dspace.spbu.ru/bitstream/11701/4023/1/diploma_-_diploma.pdf (дата обращения: 22.12.2021).
4. Лекция 8: Автоматизация тестирования. URL: <https://intuit.ru/studies/courses/48/48/lecture/1438?page=1> (дата обращения: 21.12.2021).