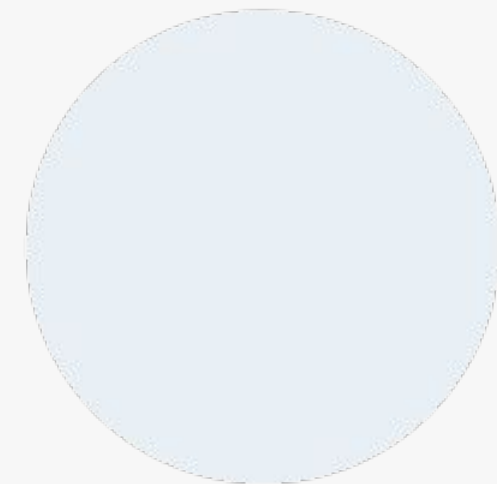


ASTON.



Введение в автоматизацию тестирования



astondevs.ru

Понятие автоматизации



Важный этап разработки любого ИТ-продукта – это тестирование. Оно помогает определить, соответствует ли программное обеспечение требованиям, все ли элементы приложения работают как надо и, если есть проблемы в первых двух случаях, что нужно сделать, чтобы их исправить. По сути, тестирование – это репетиция перед согласованием с клиентом и выходом в продакшн.

Проверить, качественный ли продукт и соответствует ли он ожиданиям, можно вручную или с помощью автоматизации тестирования. В первом случае тестировщик воспроизводит действия пользователя и фиксирует ошибки, если таковые имеются.

Во втором случае запуск, анализ, выдача результата происходят автоматически с использованием ПО. ИТ-специалист лишь обрабатывает собранную информацию. Поэтому автоматизация тестирования – это своего рода инструмент оптимизации процессов.

Автоматизация тестирования (test automation)



Автоматизация тестирования (test automation) – это набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения НЕКОТОРЫХ задач в процессе тестирования.

Программное обеспечение для автоматизации тестирования также может вводить тестовые данные в тестовую среду, сравнивать ожидаемые и фактические результаты и создавать подробные отчеты о тестах. Как правило, автоматизация тестирования требует значительных вложений денег и ресурсов.

Последовательные циклы разработки, особенно в крупных компаниях (Google, Facebook, Альфа-Банк, Тинькофф и т.д.) потребуют многократного выполнения одного и того же набора тестов. Используя инструмент автоматизации тестирования, можно записать этот набор тестов и при необходимости воспроизвести его. После автоматизации набора тестов вмешательство человека не требуется. Это улучшает ROI автоматизации тестирования.

Цель автоматизации –
уменьшить количество
тестовых примеров,
которые нужно запускать
вручную, а не полностью
исключить ручное
тестирование.

Автотесты



Автотесты можно условно разделить на три типа:

модульные (Unit)

интеграционные (API)

end-to-end (UI/API)

Цель модульных тестов – убедиться в исправности отдельно взятых функций в рамках одного модуля, из которых построено ПО.

Интеграционные тесты нужны для проверки функциональности, которую невозможно сделать с помощью модульного теста, а использовать end-to-end – излишне. End-to-end тесты нужны для проверки приложения в целом. Их задача – убедиться, что все части приложения вместе работают верно. Длительность таких тестов от нескольких секунд до нескольких минут.

В процессе автоматизации выполняются следующие шаги:



ШАГ 1. Выбор тестового инструмента.

Выбор средства тестирования во многом зависит от технологии, на которой построено тестируемое приложение.

ШАГ 2. Определение объема автоматизации.

Объем автоматизации – это область тестируемого приложения, которая будет автоматизирована. Его помогают определить следующие пункты:

- Функции, важные для бизнеса
- Сценарии с большим объемом данных
- Общие функции приложений
- Техническая осуществимость
- Частота повторного использования бизнес-компонентов
- Сложность тестовых случаев
- Возможность использовать одни и те же тестовые сценарии для кросс-браузерного тестирования

В процессе автоматизации выполняются следующие шаги:



ШАГ 3. Планирование, дизайн и разработка.

На этом этапе создается стратегия и план автоматизации, которые содержат следующие детали:

- Выбранные инструменты автоматизации
- Конструкция каркаса и его особенности
- Входящие и выходящие за рамки элементы автоматизации
- Подготовка стендов автоматизации
- График и временная шкала сценариев и выполнения
- Результаты тестирования автоматизации

ШАГ 4. Выполнение теста.

На этом этапе выполняются сценарии автоматизации.

Сценариям необходимо ввести тестовые данные, прежде чем они будут запущены. После выполнения они предоставляют подробные отчеты об испытаниях. Выполнение может быть выполнено с использованием инструмента автоматизации напрямую или с помощью инструмента управления тестированием, который вызовет инструмент автоматизации.

В процессе автоматизации выполняются следующие шаги:



ШАГ 5. Техническое обслуживание.

Этот этап автоматизированного тестирования проводится для проверки того, как работают новые функции, добавленные в программное обеспечение: нормально или нет.

Сопровождение в автотестировании выполняется, когда добавляются новые сценарии автоматизации, и их необходимо проверять и поддерживать, чтобы повышать эффективность сценариев автоматизации с каждым последующим циклом выпуска.

Инструментальное средство автоматизированного тестирования

Инструментальное средство автоматизированного тестирования (test automation tool) – это программа (или набор программ), позволяющая создавать, редактировать, отслеживать и выполнять автоматизированные тесты, а также собирать статистику их выполнения.

Для получения максимальной рентабельности инвестиций в автоматизацию следует соблюдать следующие правила:

ПРАВИЛО 1. Объем автоматизации необходимо детально определить до начала проекта. Это позволит убедиться, что ожидания от автоматизации будут оправданы.

ПРАВИЛО 2. Определите правильный инструмент автоматизации: инструмент не должен выбираться на основании его популярности, он должен соответствовать требованиям автоматизации на конкретном проекте.



Инструментальное средство автоматизированного тестирования (test automation tool) – это программа (или набор программ), позволяющая создавать, редактировать, отслеживать и выполнять автоматизированные тесты, а также собирать статистику их выполнения.

Инструментальное средство автоматизированного тестирования



ПРАВИЛО 3. Стандарты создания сценариев. При написании сценариев для автоматизации необходимо соблюдать стандарты. Вот некоторые из них:

- создайте единые скрипты, комментарии и отступы кода;
- разработайте правила наименования тестовых сценариев;
- прикладывайте необходимые документы, если, например, сложно понять прохождение тестового сценария без скриншота и/или спецификации.

ПРАВИЛО 4. Определите метрики и следите за ними.

Успех автоматизации нельзя определить лишь путем сравнения затраченных усилий, на тот или иной вид тестирования. Вот лишь некоторые:

- процент обнаруженных дефектов;
- время, необходимое для тестирования автоматизации выпуска каждого нового цикла;
- минимальное время требуемое для выпуска;
- индекс удовлетворенности клиентов;
- улучшение производительности

Выбирая между ручным тестированием и автоматизацией, можно придерживаться следующих положений.



Ручное тестирование будет лучшим решением, если вы:

1. ещё в процессе осознания функциональности вашего продукта;
2. нуждаетесь в проверке юзабилити и вёрстки;
3. хотите локализовать и описывать возникающие проблемы;
4. быстро проводите ручные тесты, а их количество совсем небольшое;
5. не обладаете нужными бюджетом и временем на автоматизацию тестирования.

Автоматизированное тестирование рекомендуется к внедрению в случаях, когда у вас:

1. много ручных тестов и нет времени на постоянную проверку всего регресса;
2. проходит много времени между обнаружением ошибки и её исправлением;
3. подготовка к тестированию – формирование данных и настройка конфигурации – требует много времени;
4. требуется поддержка старых версий софта – нужно проверять новые сервис-паки и патчи;
5. нет уверенности, что новый код не сломает результат работы других разработчиков.

Задачи и цели автоматизации



Для бизнеса важно как можно скорее доставлять изменения пользователям, опережать конкурентов. Это возможно благодаря гибким методологиям разработки, а также таким практикам, как continuous integration и continuous delivery. Последние подразумевают использование автоматизированного тестирования.

При регулярном запуске автотестов есть уверенность, что приложение работает так, как бизнес рассчитывает. А если возникают проблемы — о них узнают в кратчайшие сроки.

Цели автоматизации



К числу целей автоматизации тестирования программного обеспечения можно отнести следующие:

- Сократить стоимость и время тестирования;
- Уменьшить избыточность;
- Ускорить процесс тестирования;
- Помочь улучшить качество продукта;
- Улучшить тестовое покрытие;
- Сократить ручное вмешательство;
- Оперативно предоставлять заказчику информацию о состоянии продукта.

Задачи автоматизации



Одними из задач тестирования являются:

- **Проверка ссылок.**

Проверка ссылок в любом веб-ориентированном приложении является неотъемлемым этапом тестирования. Эта задача в тестировании может быть полностью автоматизирована (в том числе с применением бесплатных инструментов).

- **Проверка работоспособности стандартного, типичного для множества проектов функционала.**

Есть функции, повторяющиеся во множестве проектов. Они полностью идентичны или обладают незначительными отличиями. Автоматизация тестирования таких функций экономит много времени. Примеры таких функций:

- Аутентификация пользователя;
- Регистрация пользователя;
- Добавление товара в корзину.

- **Проверка стандартных элементов управления.**

Изолированная проверка стандартных элементов управления хорошо автоматизируется как в настольных приложениях, так и в веб-ориентированных.

- **Базовая проверка безопасности.**

Полноценную проверку безопасности автоматизировать едва ли представляется возможным, однако, проверить приложение или всю систему на наличие «глупых и известных» уязвимостей можно и нужно с помощью имеющихся в широком ассортименте средств автоматизированной проверки безопасности. К базовой проверке можно отнести: исследование известных уязвимостей тех или иных версий ПО, неверно выставленные права доступа (открыты какие-то файлы, которые не должны быть доступны извне), использование паролей по умолчанию, разрешенный доступ к СУБД не с localhost и т.п.

Задачи автоматизации



- **Тестирование производительности и нагрузочное тестирование.** Этот вид тестирования относится к числу тех, в которых без автоматизации обойтись невозможно или почти невозможно. Так, например, вручную крайне тяжело проэмулировать работу нескольких десятков тысяч пользователей, одновременно собирая полную статистику по работе каждого из них и реакции приложения.
- **Смоук-тест для крупных систем.** В крупных системах смоук-тест может быть достаточно длинным и утомительным. При этом он, как правило, содержит достаточно простые для быстрой автоматизации операции. В данном случае автоматизация тестирования приносит ощутимый выигрыш.
- **Регрессионное тестирование.** В регрессионном тестировании достаточно большой процент тестов повторяется снова и снова на каждом цикле тестирования. Их автоматизация может сэкономить в конечном итоге много сил и времени.
- **Конфигурационное тестирование (проверка работоспособности приложения при разных настройках).** Представим ситуацию: у приложения 100 параметров в файле конфигураций. Каждый из них может принимать 100 значений. Итого, может быть 10000 вариантов конфигурационных файлов. Проверять такое вручную можно десятилетиями.

Задачи автоматизации



- **Часто повторяющиеся тесты, простые для автоматизации.** В случае наличия таких тестов автоматизация позволяет как сократить время, затраченное на цикл тестирования, так и минимизировать риски – подобные тесты легко поддерживать. Более того – их почти никогда не приходится подолгу отлаживать и дорабатывать, т.к. они просты.
- **Длительные утомительные для человека тесты.** Заполнение форм, регистрация тысяч пользователей и т.п. задачи крайне утомительны для человека и являются неприятной рутиной, которой хотелось бы избежать. Автоматизация такой работы проста и приносит ощутимый эффект. Дополнительным преимуществом является исключение человеческого фактора, заключающегося в том, что на подобных утомительных операциях люди часто совершают множество ошибок.

Преимущества и недостатки автоматизации



К числу преимуществ автоматизации можно отнести:

- Скорость (компьютер работает быстрее человека; мы экономим время и, как следствие, деньги);
- Надежность (компьютер не допускает «человеческих ошибок»);
- Мощность (можно выполнить действия, недоступные человеку);
- Средства автоматизации тестирования собирают числовую информацию и представляют её в удобной для понимания человеком форме;
- Средства автоматизации тестирования в сложных ошибочных ситуациях способны выполнять «низкоуровневые действия», сложные для человека.

К числу недостатков можно отнести:

- Необходим квалифицированный персонал;
- Необходима грамотная стратегия разработки и управления тестами, тестовыми данными и т.п.;
- Необходимы специальные инструментальные средства (зачастую – ОЧЕНЬ дорогие);
- В случае серьёзных изменений в приложении многие автоматизированные тесты становятся бесполезными и/или требуют серьёзной переработки;
- На автоматизацию часто возлагают неоправданные надежды, что ведет к срывам сроков и прочим проблемам;
- Все области применения/продукта не могут быть автоматизированы. Там будут определенные ограничения в определенных областях.

Необходимость автоматизации



Говоря о необходимости автоматизации можно выделить тесты, которые нужно автоматизировать, и тесты, необходимость в автоматизации которых отсутствует.

К числу первых относятся:

- Бизнес-критические пути: фичи или пользовательские сценарии, при падении которых будет нанесен существенный урон бизнесу.
- Тесты, которые должны прогоняться на каждом билде/релизе приложения – например, smoke, sanity, регресс.
- Тесты, которые нужно прогонять на разных конфигурациях – разных операционных системах и браузерах.
- Тесты, которые используют один и тот же сценарий, но разные данные для каждого прогона (тесты, управляемые данными).
- Тесты, которые завязаны на большие объемы данных (например, заполнение очень больших форм).
- Тесты, которые могут быть использованы для тестирования производительности (стресс, нагрузочное тестирование).

Необходимость автоматизации



- Тесты, которые занимают много времени на выполнение и могут быть запущены во время перерывов или ночью.
- Тесты, во время которых необходим захват изображения, чтобы доказать, что приложение ведет себя верно, или проверка того, что ряд веб-страниц одинаково выглядит в разных браузерах.

К числу тестов, которые не нужно автоматизировать, можно отнести:



- Тесты, которые прогоняются только один раз или незначительное количество раз.
- Тесты пользовательского опыта и удобства использования, требующие реакции пользователя на простоту использования приложения.
- Тесты, которые нужно запустить очень срочно. Обычно новая, только что разработанная фича требует быстрой обратной связи, и ее тестируют вручную.
- Тесты, которые требуют доменного знания и опыта – исследовательское тестирование.

- Прерывающиеся тесты. Тесты с непредсказуемыми результатами не особенно ценны. Чтобы получить наилучший выхлоп от автоматизации, тесты должны давать предсказуемые и надежные результаты, чтобы проходить pass/fail условия.
- Тесты, требующие визуального подтверждения. Однако мы можем делать скриншоты страниц во время автотестов, а затем вручную проверять их.
- Тесты, которые не могут быть на 100% автоматизированы, не должны быть автоматизированы вообще, если только их автоматизация не сэкономит значительное время.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ



Сегодня заданием будет поискать в интернете и изучить дополнительную информацию по данной теме.

В последующем все задания будут сдаваться проверяющим путём предоставления ссылки на репозиторий.

Сразу оговоримся, в каком виде это будет происходить.

Во-первых, необходимо создать себе аккаунт на GitHub (если у вас его нет).

Во-вторых, создать себе репозиторий, в который вы будете складывать выполненные домашние задания.

В-третьих, сразу договоримся, что для каждого урока вам необходимо будет создавать отдельную ветку с названием вида "Lesson_1" (если иное не указано в условии).

В-четвёртых, условимся, что будем использовать Java версии 11.