



Занятие №12. Автоматизация тестирования.

На этом занятии Вы узнаете:

- Что такое автоматизация
- Как и когда нужно применять автоматизированное тестирование
- Как посчитать ROI автоматизации

Что это такое?

Автоматизация тестирования – это процесс тестирования программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования.

Из чего состоит?

- Автотесты

Из чего состоит?

- Автотесты
- Система запуска

Из чего состоит?

- Автотесты
- Система запуска
- Анализ результатов

Зачем нужна автоматизация?

- Избавление от рутины

Зачем нужна автоматизация?

- Избавление от рутины
- Регрессионное тестирование

Зачем нужна автоматизация?

- Избавление от рутины
- Регрессионное тестирование

Объект

тестирования

- Web
- Desktop
- Web services (API)
- СУБД
- Мобильные приложения

Критерии выбора инструмента

- Простота написания скриптов и их поддержки (знание языка)
- Возможность записи скриптов (качество, удобство кода)
- Стоимость
- Тесты, управляемые данными
- Наличие техподдержки
- Наличие развитого сообщества
- Наличие понятных и информативных отчетов

Платные инструменты

- QuickTest Professional, WinRunner (Hewlett-Packard)
- Rational Robot, Rational Functional Tester (IBM Rational)
- TestComplete (AutomatedQA Corp)
- Microsoft VS (Microsoft)

Бесплатные инструменты

- Selenium (WebDriver)
- Sikuli
- Watir, Watij, Watin
- Autoit

Базовая классификация

Инструмент	Запись скриптов	Языки	Кроссбраузерность	Техподдержка	Сообщество	Отчеты
Selenium	+	+	+	-	+	+
Sikuli	+	+	+	-	+	+
Test Complete	+	+	+	+	-	+
QTP	+	-	+	+	-	+
Microsoft VS	+	+	-	+	-	+

Selenium

- Бесплатный.
- Развитое сообщество.
- Поддержка многих языков программирования.

Компоненты Selenium

- Selenium WebDriver
- Selenium IDE
- Selenium Server
- Selenium Grid

Официальный сайт:

<http://www.seleniumhq.org/>

Организация тестирования

1. Фреймворк тестирования
 - JUnit, TestNG (Java)
 - NUnit, Gallio (.Net)
 - RSpec, Cucumber (Ruby)
 - unittest, PyUnit, nose, py.test (Python)
2. Среда разработки (eclipse, VS ...)
3. Система автоматической сборки (Maven, Ant...)
4. Сервер непрерывной интеграции (Jenkins, Bamboo...)

Пример теста

@Before

```
public void setUp() throws Exception {  
    driver = new FirefoxDriver();  
    baseUrl = "https://e.mail.ru";  
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
}
```

@Test

```
public void auth_test() throws Exception {  
    driver.get(baseUrl + "/login");  
    driver.findElement(By.cssSelector("[name=Login]")).clear();  
    driver.findElement(By.cssSelector("[name=Login]")).sendKeys("test@mail.ru");  
    driver.findElement(By.cssSelector("[name=Password]")).sendKeys("12345");  
    driver.findElement(By.xpath("//*[@id='LoginExternal']/button[@type='submit']")).click();  
    Assert.assertTrue("Не удалось залогиниться", isElementPresent(By.cssSelector("#b-letters")));  
}
```

@After

```
public void tearDown() throws Exception {  
    driver.quit();  
}
```

Преимущества автоматизации

- Быстрый фидбек о состоянии системы в целом. (регулярные прогоны)
- Экономит время прогона теста.
- Исключает человеческий фактор в тестировании
- Экономит время на генерации тестовых данных
- Тесты становятся автономными

Недостатки автоматизации

- Требуется специфические знания (программист и тестировщик)
- Требуется машинные ресурсы
- Ограничение в проверке результатов теста
- Нецелесообразно при частых изменениях в продукте

Сложно автоматизировать

- Проверка содержимого видео
- Проверка воспроизведения аудио
- Юзабилити-тестирование

Как определить выгодно ли будет автоматизация

R Return

O On

I Investment

ROI

ROI (return on investment) – финансовый коэффициент, иллюстрирующий уровень доходности или убыточности бизнеса, учитывая сумму сделанных в этот бизнес инвестиций.

ROI

ROI = Прибыль \ Затраты =

Доходы – Затраты

Затраты

ROI = 1 *Нормально*

ROI > 1 *Хорошо*

ROI < 1 *Плохо*

Затраты

- Фиксированные
- Переменные

Фиксированные затраты

- Стоимость лицензии инструмента
- Обучение и поиск специалистов по автоматизации
- Начальная подготовка фреймворка
- Начальная подготовка инфраструктуры

Переменные затраты

- Разработка тестов
- Запуск тестов
- Поддержка тестов
- Поддержка фреймворка
- Поддержка инфраструктуры

Прибыль

$$\text{Прибыль} = (X - Y) * N$$

Где

X – время на ручное тестирование,

Y – время на выполнение автотестов,

N – кол-во запусков за определенный промежуток времени.

Как влиять на ROI

- Чаще запускайте тесты
- В первую очередь выгодные тесты
- Оптимизируйте фреймворк
- Оптимизируйте автотесты
- Делайте постоянные замеры ROI

Эффективные тестовые прогоны

Время прогона автотеста < Время прогона теста руками

Не стоит прогонять автотесты «в холостую»

Выгодные автотесты

- Минимальная сложность разработки
- Минимальная сложность поддержки
- Максимальная важность

Фреймворк и автотесты

- Оптимизация времени на поддержку фреймворка
- Замер и оптимизация времени прохождения автотеста.
- Оптимизация времени на поддержку автотеста.

Постоянные замеры ROI

Собирайте метрики и возможности
влияйте на ROI



ТЕХНОТРЕК

**Спасибо за
внимание!**