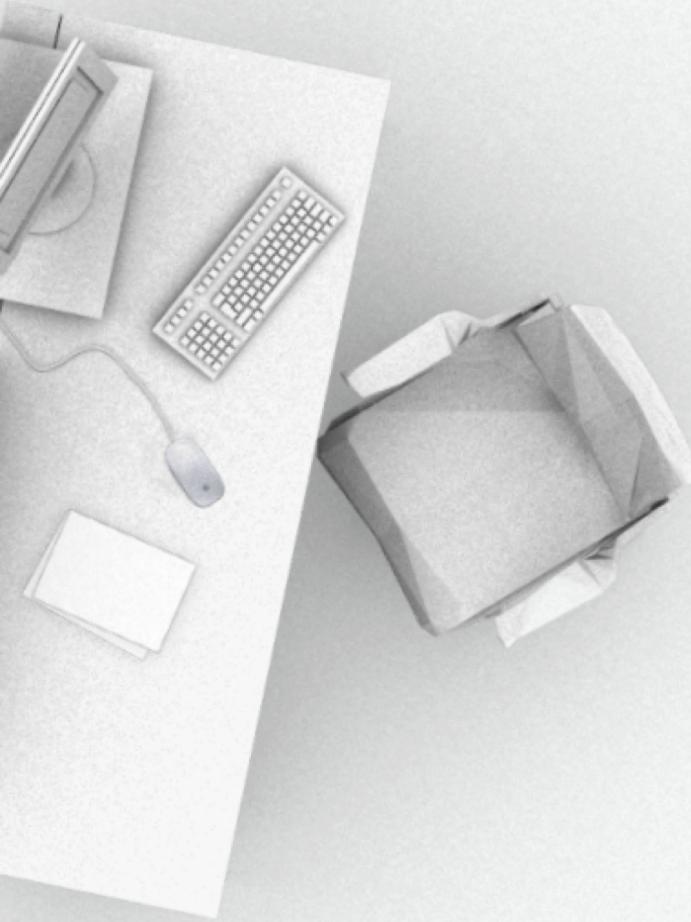


Автоматизированное тестирование

Илья Кириллов



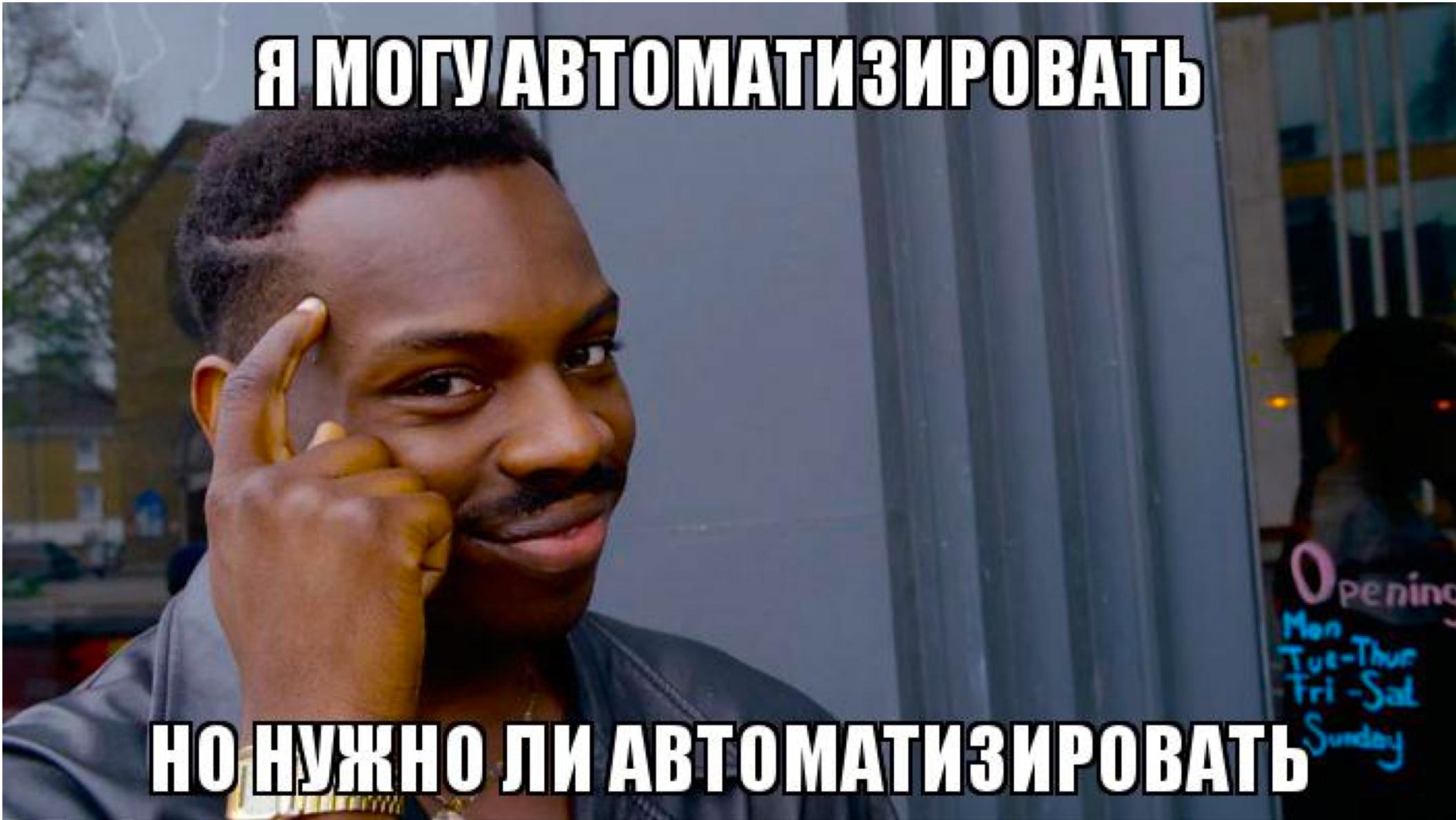


Автоматизированное тестирование

Что такое автоматизированное тестирование?

Это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически

Я МОГУ АВТОМАТИЗИРОВАТЬ



НО НУЖНО ЛИ АВТОМАТИЗИРОВАТЬ

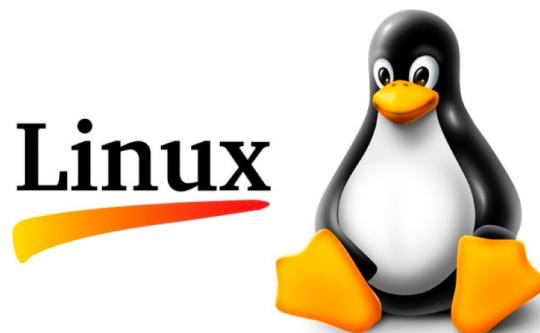
Преимущества автоматизированного тестирования

- Быстрое выполнение
- Повторяемость
- Меньше затрат на поддержку
- Отчеты

Цели автоматизации тестирования

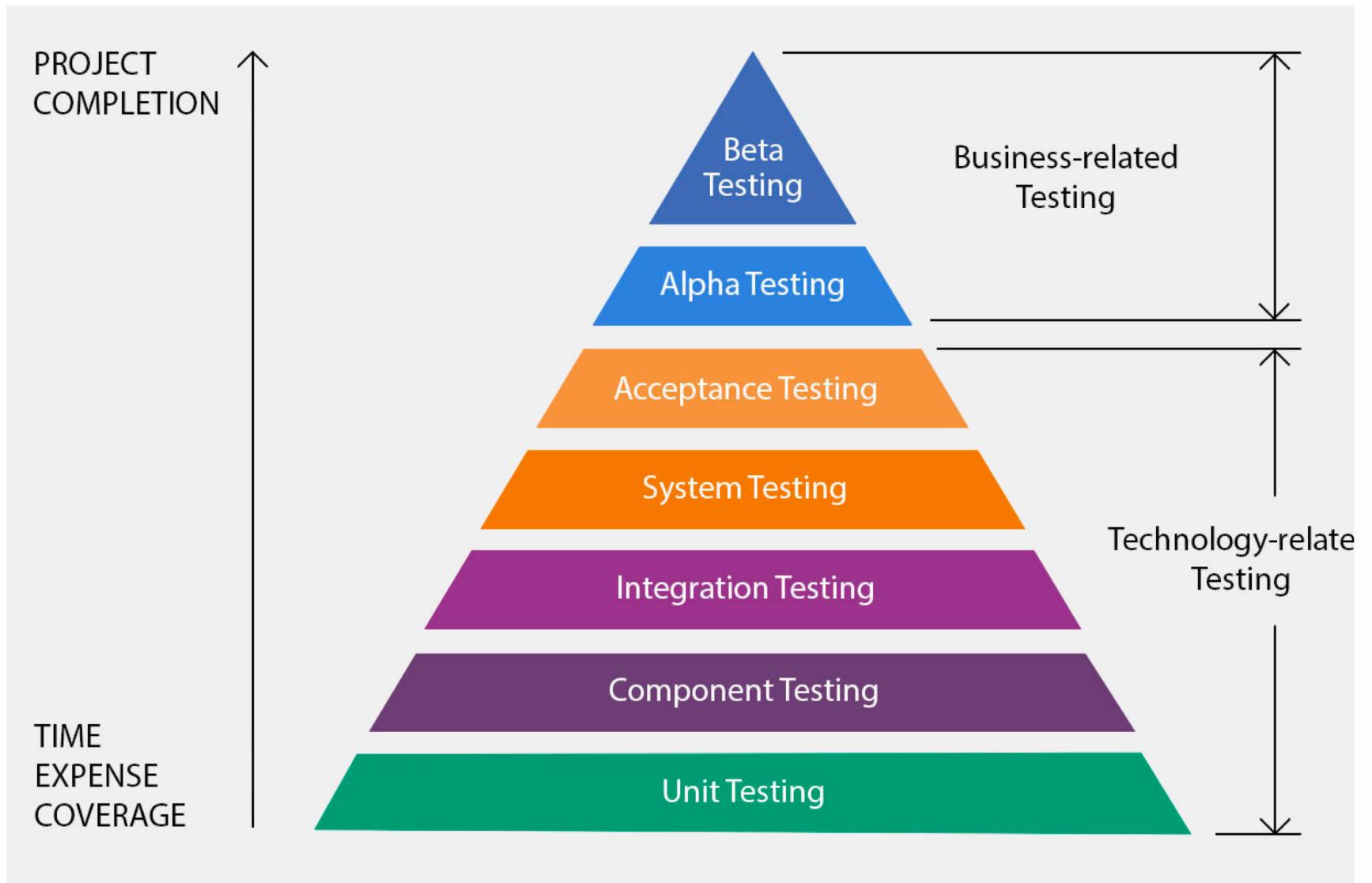
- Повышение производительности труда
- Повышение надежности
- Повышение скорости разработки

Использование различных ОС при автоматизации тестирования



MacTM OS

Уровни тестирования



НЕХОЧУ НИЧЕГО НАСТРАИВАТЬ



#09



Инструменты



cypress.io

🎭 Playwright

The SoapUI logo, featuring a large green stylized letter "S" with an arrow pointing right, followed by the word "SoapUI" in a bold, green, sans-serif font.

SoapUI

The Robot Framework logo, which is a black square containing a white icon of a face with a neutral or slightly grumpy expression.

Robot Framework

The POSTMAN logo, which is a red circle containing a white icon of a person holding a pen and writing.

POSTMAN

The appium logo, which consists of three overlapping semi-circles in blue, purple, and light blue.

appium

The XCTest logo, which is a grey rounded square containing a white checkmark icon and the letters "Se".

XCTest

The Espresso Test Recorder logo, which is a green rounded square containing a white checkmark icon.

Espresso Test Recorder

The Espresso Test Recorder logo, which features a brown Android robot head on top of a white coffee cup. A green Android icon is on the side of the cup.

UI Testing for Android
Espresso Test Recorder

#UITU

Selenium + Appium



#0...

Selenium + Appium



#012

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("http://www.mail.ru")
assert "Mail.ru" in driver.title
assert "Yandex" not in driver.page_source

driver.close()
```

CSS ID

```
<html>
<body>
  <form id="loginForm">
    <input name="username" type="text" />
    <input name="password" type="password" />
    <input name="continue" type="submit" value="Login" />
  </form>
</body>
<html>
```

```
login_form = driver.find_element_by_id("loginForm")
login_forms = driver.find_elements_by_id("loginForm")
```

CSS CLASS NAME

```
<html>
  <body>
    <p class="content">Site p content goes here.</p>
    <a class="content">Site a content goes here.</a>
  </body>
<html>
```

```
content = driver.find_element_by_class_name('content')
content = driver.find_elements_by_class_name('content')
```

CSS SELECTOR

```
<html>
  <body>
    <p class="content">Site p content goes here.</p>
    <a class="content">Site a content goes here.</a>
  </body>
<html>
```

```
content = driver.find_element_by_css_selector('p.content')
content = driver.find_elements_by_css_selector('p.content')
```

CSS NAME ATTRIBUTE

```
<html>
  <body>
    <form id="loginForm">
      <input name="username" type="text" />
      <input name="password" type="password" />
      <input name="continue" type="submit" value="Login" />
      <input name="continue" type="button" value="Clear" />
    </form>
  </body>
<html>
```

```
username = driver.find_element_by_name('username')
password = driver.find_element_by_name('password')
```

XPATH

```
<html>
  <body>
    <form id="loginForm">
      <input name="username" type="text" />
      <input name="password" type="password" />
      <input name="continue" type="submit" value="Login" />
      <input name="continue" type="button" value="Clear" />
    </form>
  </body>
<html>
```

```
login_form = driver.find_element_by_xpath("/html/body/form[1]")
login_form = driver.find_element_by_xpath("//form[1]")
login_form = driver.find_element_by_xpath("//form[@id='loginForm']")
```

```
<input type="text" name="passwd" id="passwd-id">Input password here</input>
```

```
element = driver.find_element_by_id("passwd-id")
element = driver.find_element_by_name("passwd")
element = driver.find_element_by_xpath("//input[@id='passwd-id']")
element = driver.find_element_by_xpath("//input[contains(text(), 'Input password')]")
```

```
>>> element
<selenium.webdriver.remote.webelement.WebElement>
    id: '4e4387a1-6987-4050-a296-a1f395ca6d81'
    location: {'x': 1042, 'y': 90}
    rect: {'height': 37, 'width': 224, 'x': 1041.859375, 'y': 89.90625}
    size: {'height': 37, 'width': 224}
    tag_name: 'input'
    text: 'text'
```

```
>>> dir(element)
is_displayed
send_keys
clear
click
```

#000

```
<input type="text" name="passwd" id="passwd-id" />
```

```
from selenium.webdriver.common.keys import Keys
```

```
element = driver.find_element_by_id("passwd-id")
element.clear()
element.send_keys("MyPassword")
element.send_keys(Keys.RETURN)
```

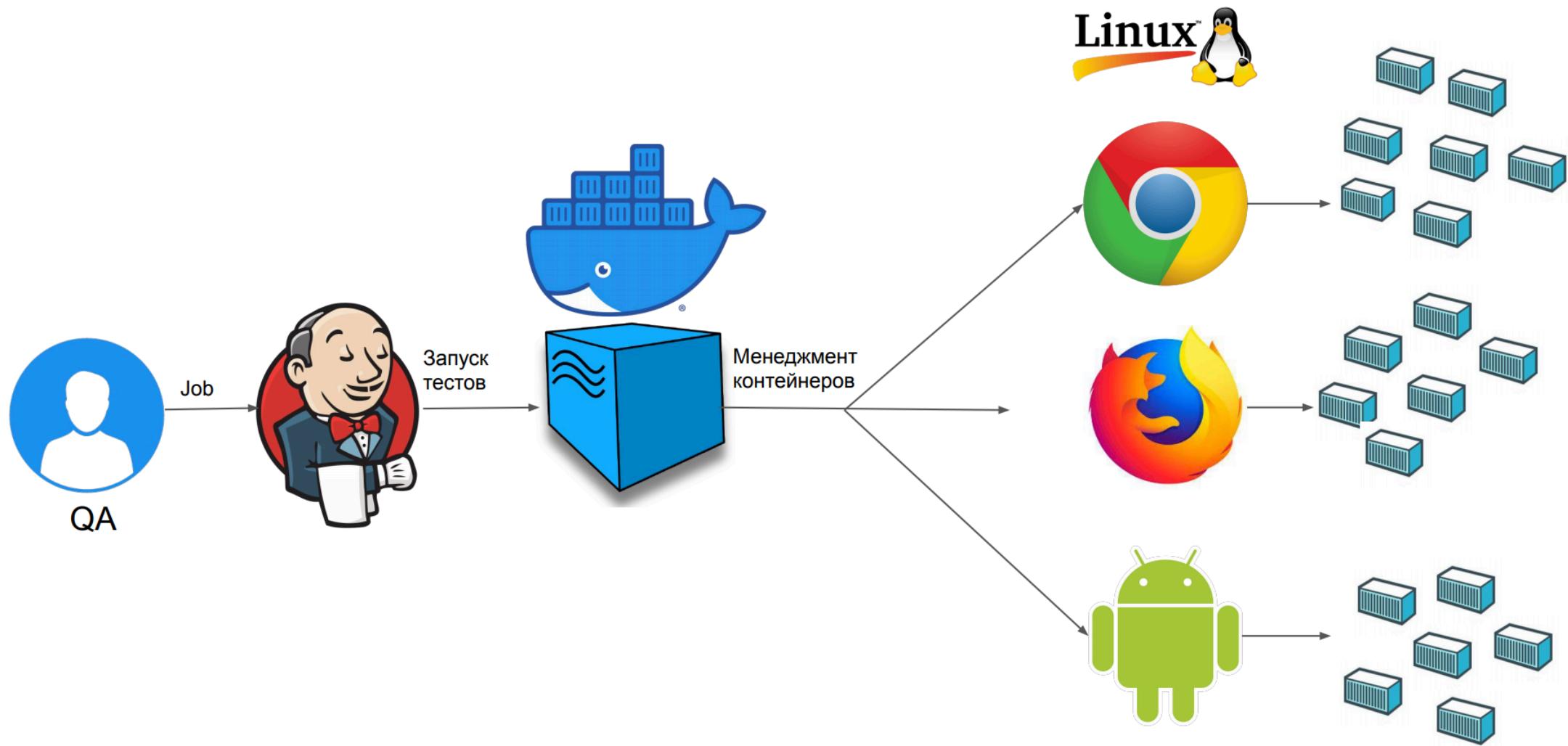
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

def test():
    driver = webdriver.Chrome()
    driver.get("http://www.python.org")
    assert "Python" in driver.title

    elem = driver.find_element_by_name("q")
    elem.send_keys("pycon")
    elem.send_keys(Keys.RETURN)
    assert "No results found." not in driver.page_source

    driver.quit()
```

Selenoid



BDD

```
Feature: Write comments
As a blog reader
I want to be able to write a comment
So that I tell the author my opinion / feedback

| Background: Login with author user
|   Given that the author adds a new post with title "Post to comment"

| Scenario: Leave a comment with all info filled in
|   Given that I select the post
|   When I add a new comment with name, email and body
|   Then I will see the comment on the blog

| Scenario: Leave a comment with name field not filled in
|   Given that I select the post
|   When I add a new comment with email and body
|   Then I will see the message "ERROR: please fill the required fields (name, email)."

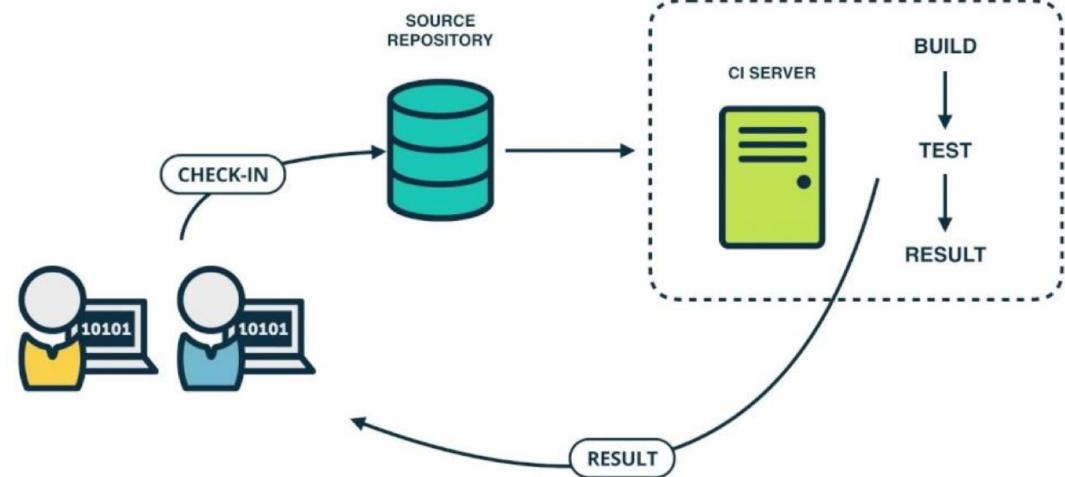
| Scenario: Leave a comment with email field not filled in
|   Given that I select the post
|   When I add a new comment with name and body
|   Then I will see the message "ERROR: please fill the required fields (name, email)."

| Scenario: Leave a comment without body
|   Given that I select the post
|   When I add a new comment with name and email
|   Then I will see an empty comment on the blog
```

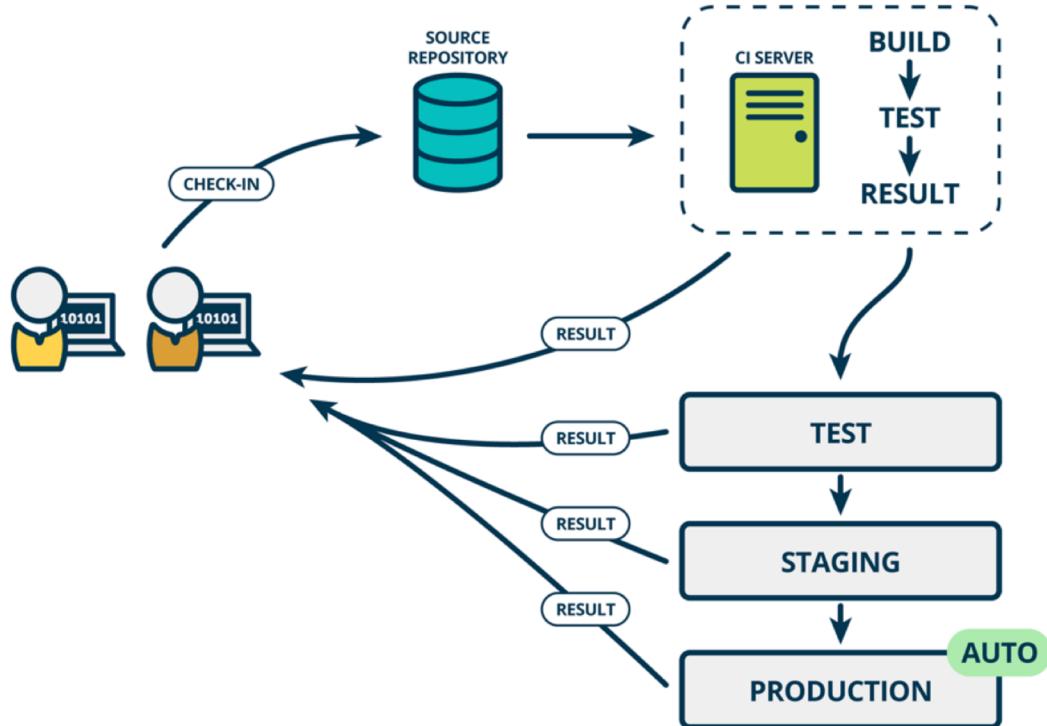
CI

Continuous Integration –
*непрерывная интеграция,
автоматическая сборка
программного обеспечения и
его тестирование на
корректность*

Continuous Integration (CI)



CD

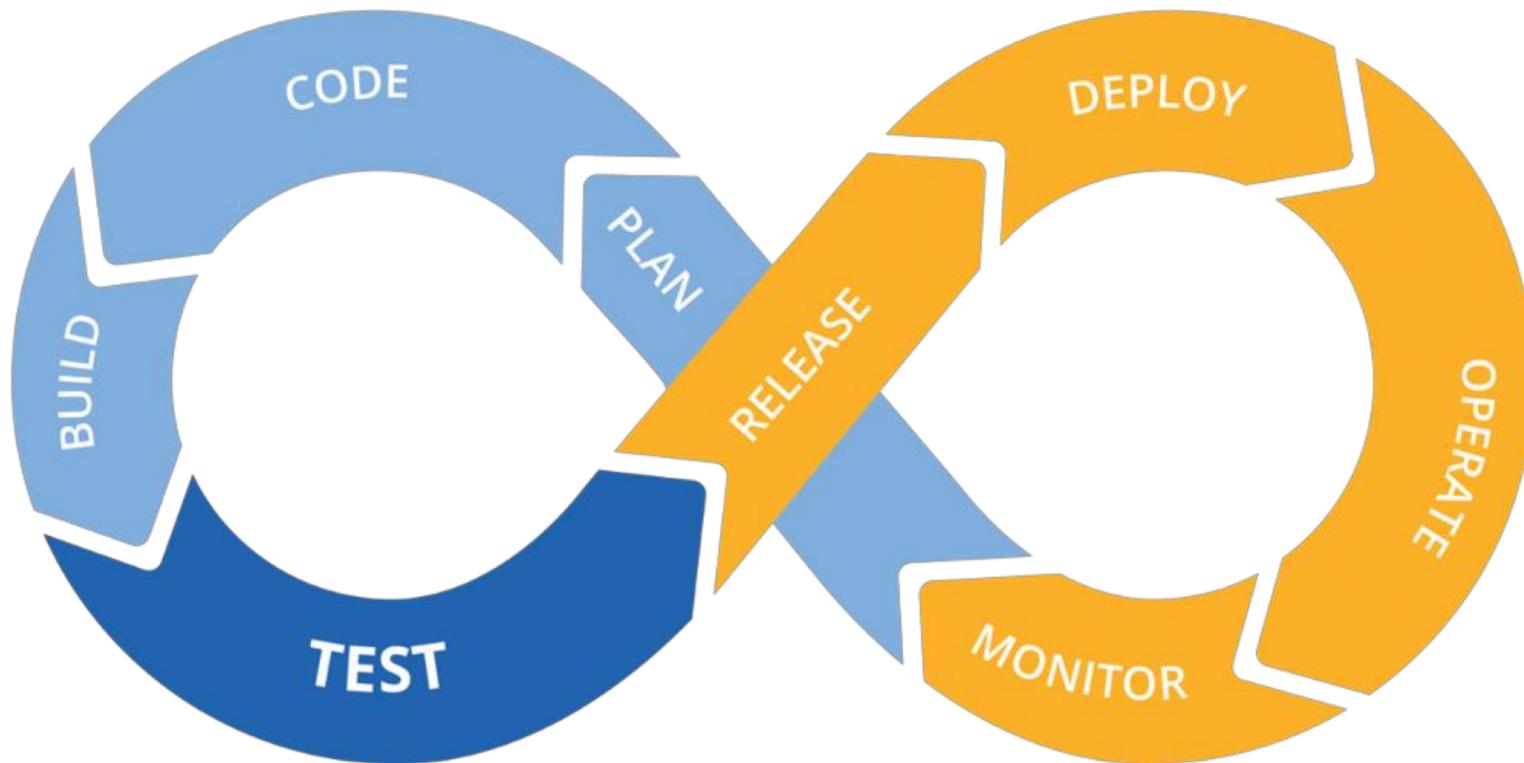


stage – среда интеграции
(предпродакшн), в которой происходит проверка перед боем

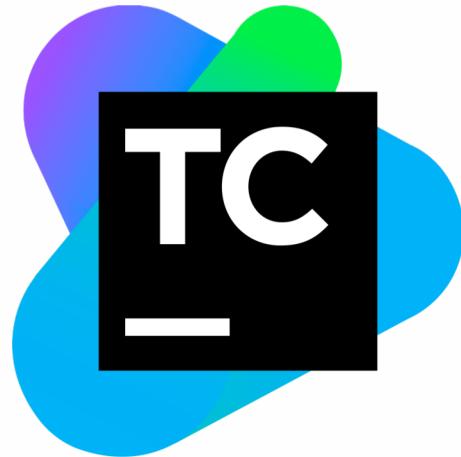
Continuous Delivery – непрерывная доставка, автоматическая установка изменения кода на **stage** серверах

Continuous Deployment –
непрерывное развертывание

Цикл CI/CD



#026

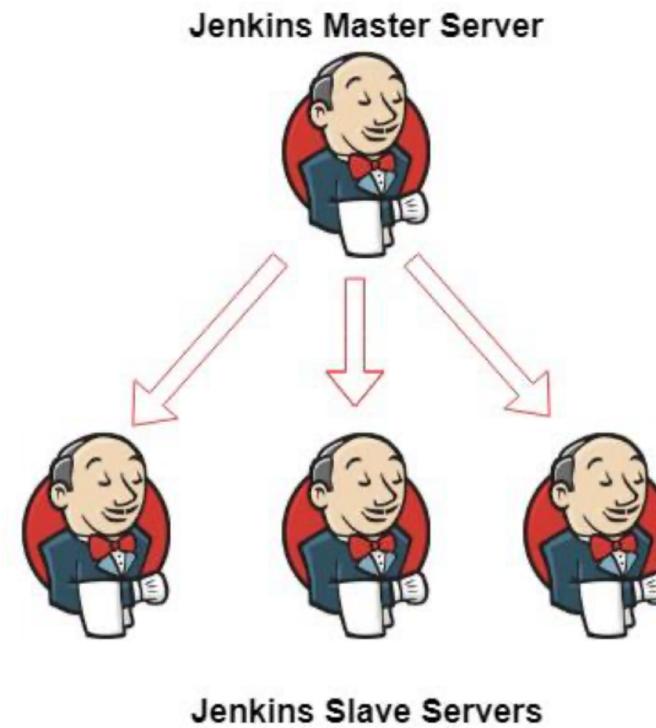


#027

Jenkins

Jenkins — программная система с открытым исходным кодом на Java, предназначенная для обеспечения процесса непрерывной интеграции программного обеспечения.

- Бесплатный
- Надежный
- Камьюнити, мануалы, образы
- Легко развернуть и настроить





py.test



**СПАСИБО
ЗА ВНИМАНИЕ**

