CS608
Software Testing

Dr. Stephen Brown

Room Eolas 116

stephen.brown@mu.ie

CS608

SOFTWARE TEST AUTOMATION
TOOLS & FRAMEWORKS
IN USE TODAY

REAL-WORLD USAGE

- Hard to get reliable figures
- Some statistics from a survey by simform.com
- Tools survey by jetbrains.com
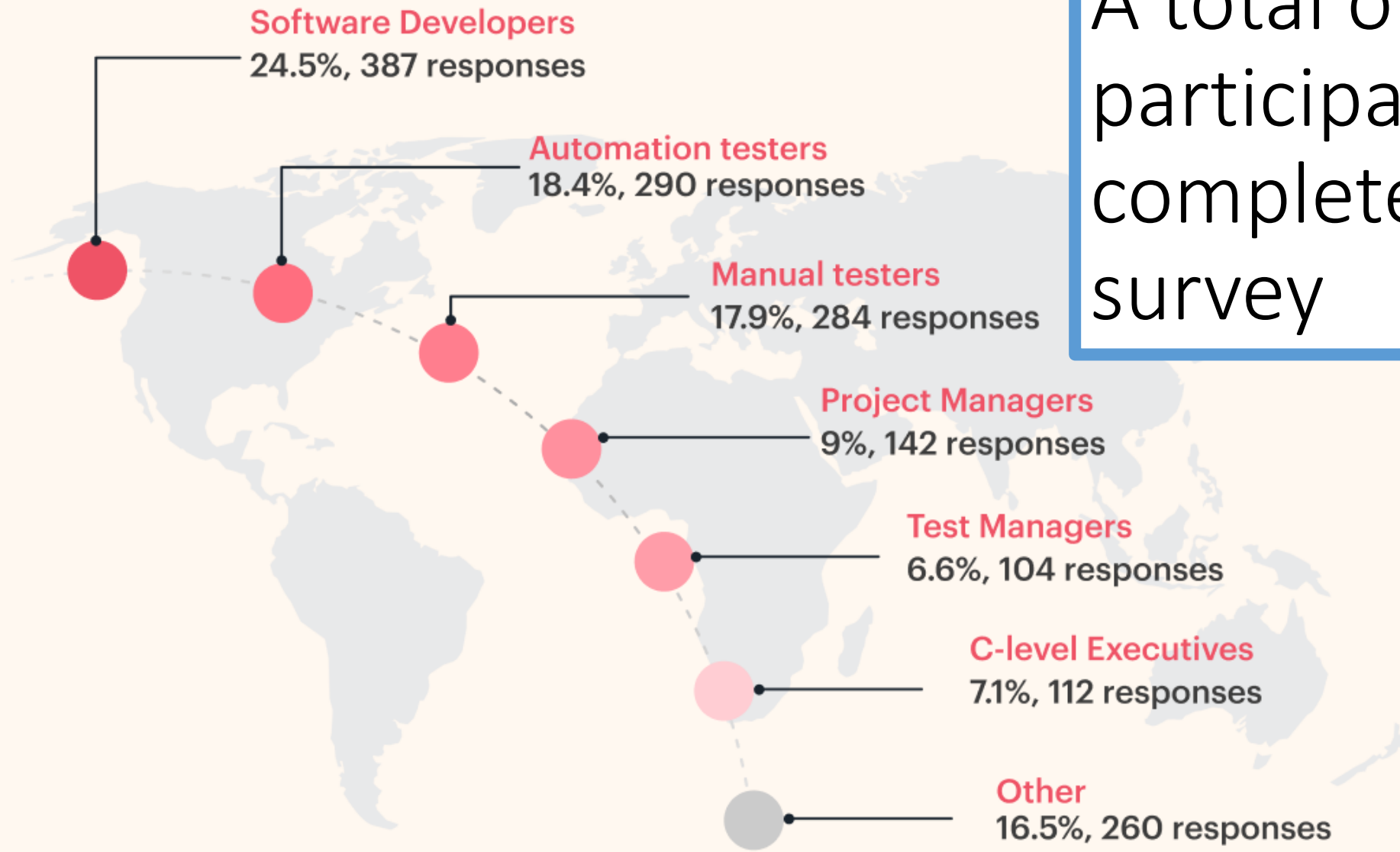- Some examples from saucelabs.com

SIMFORM SURVEY

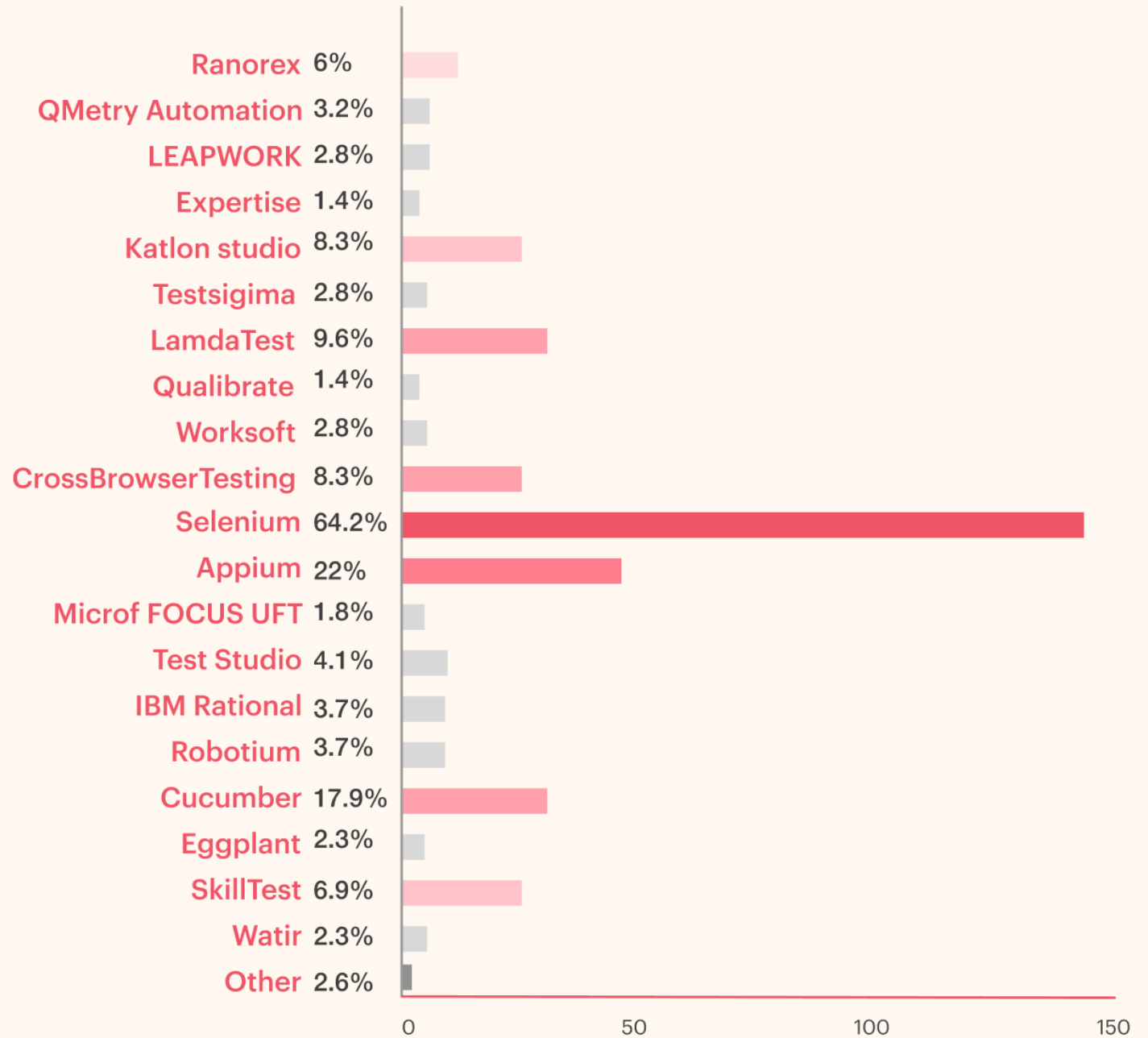Here we can see that the most preferred tests for automation are...

- ✓ *Functional testing(66.5%)*

- ✓ *API testing(54.2%)*

- ✓ *Regression testing(50.5%)*

- ✓ *Smoke testing(38.2)*

These tests involve a lot of repetitive work and data entry. Plus, they have clear pass/fail results, so they are the most preferred tests for automation.

Software Developers
24.5%, 387 responses

Automation testers
18.4%, 290 responses

Manual testers
17.9%, 284 responses

Project Managers
9%, 142 responses

Test Managers
6.6%, 104 responses

C-level Executives
7.1%, 112 responses

Other
16.5%, 260 responses

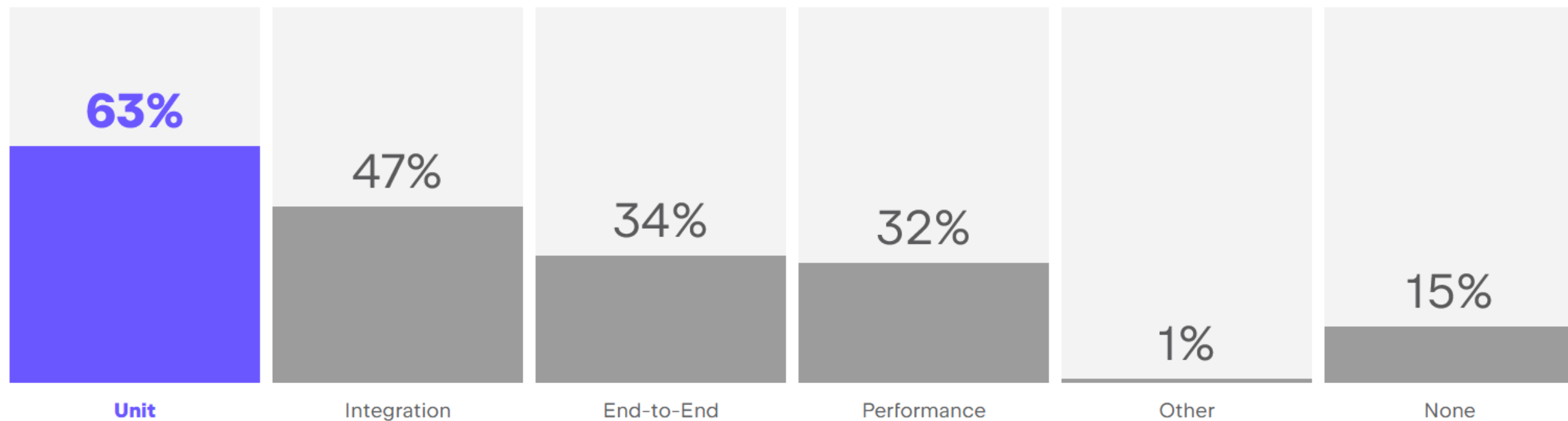A total of 1579 participants completed the survey

SIMFORM

When it comes to preferences for the [test automation tools](#), some of the highlighted ones are among the popular ones, Selenium (64.2%), Appium (22%), and Cucumber (18%)
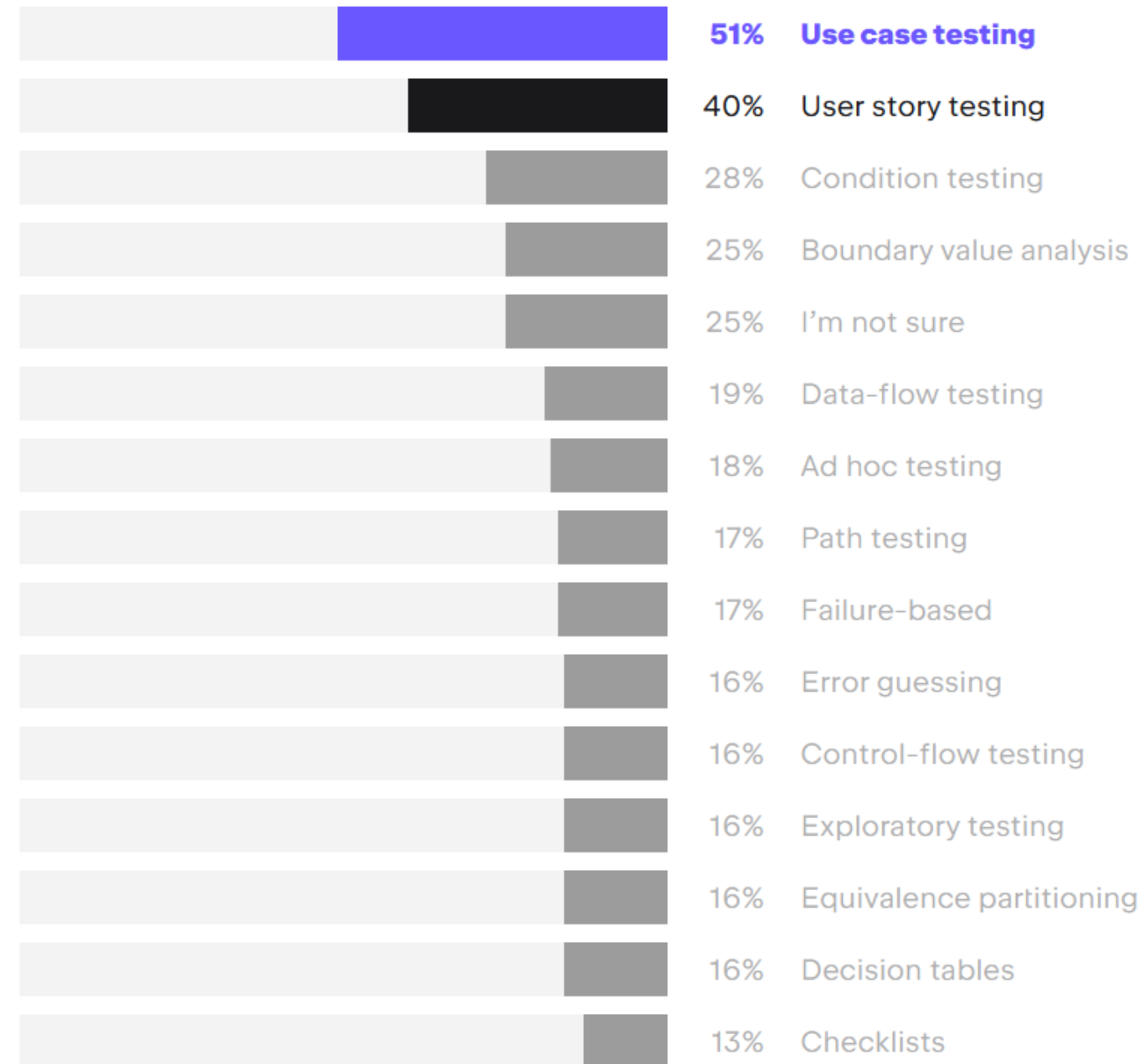
Tool	Percentage
Ranorex	6%
QMetry Automation	3.2%
LEAPWORK	2.8%
Expertise	1.4%
Katlon studio	8.3%
Testsigima	2.8%
LamdaTest	9.6%
Qualibrate	1.4%
Worksoft	2.8%
CrossBrowserTesting	8.3%
Selenium	64.2%
Appium	22%
Microf FOCUS UFT	1.8%
Test Studio	4.1%
IBM Rational	3.7%
Robotium	3.7%
Cucumber	17.9%
Eggplant	2.3%
SkillTest	6.9%
Watir	2.3%
Other	2.6%

SIMFORM

JetBrains Survey, 2023

What types of tests do you have in your projects?

Unit	Integration	End-to-End	Performance	Other	None
63%	47%	34%	32%	1%	15%

Which test design techniques do you use?

51%	**Use case testing**	
40%	User story testing	
28%	Condition testing	
25%	Boundary value analysis	
25%	I'm not sure	
19%	Data-flow testing	
18%	Ad hoc testing	
17%	Path testing	
17%	Failure-based	
16%	Error guessing	
16%	Control-flow testing	
16%	Exploratory testing	
16%	Equivalence partitioning	
16%	Decision tables	
13%	Checklists	

Which test design techniques do you use?

51%	**Use case testing**	←
40%	User story testing	
28%	Condition testing	
25%	Boundary value analysis	←
25%	I'm not sure	
19%	Data-flow testing	
18%	Ad hoc testing	
17%	Path testing	
17%	Failure-based	
16%	Error guessing	
16%	Control-flow testing	←
16%	Exploratory testing	
16%	Equivalence partitioning	←
16%	Decision tables	←
13%	Checklists	

9

Approximately how many of your project tests are automated?

All	Most	Half	Only some	I'm not sure	None
10%	24%	12%	**26%**	14%	14%
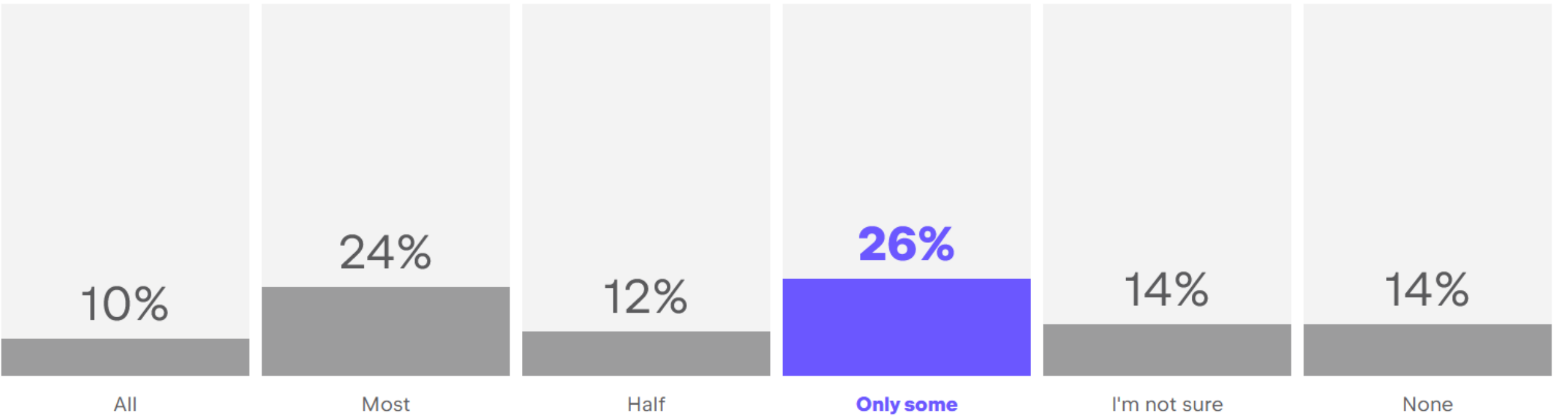
What is your primary testing target?
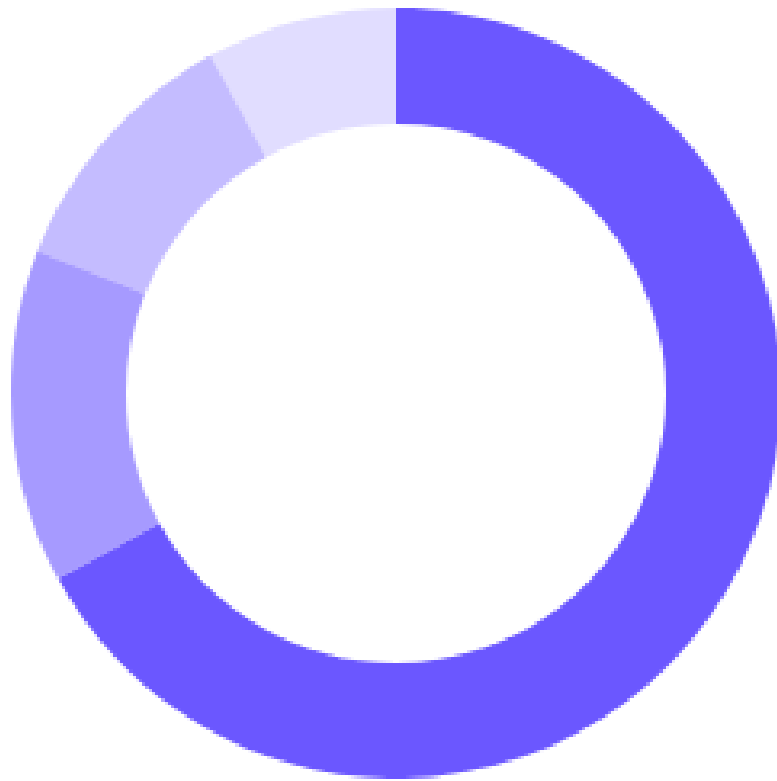
Web applications • 67%

Desktop applications • 14%

Mobile applications • 11%

Other • 8%

SAUCELABS/TEST AUTOMATION EXAMPLES

- Selenium

- Cypress

- Playwright

- Testcafe

- Robot Framework

- Appium

- Cucumber and Gherkin

"Top Test Automation Frameworks in 2024", Saucelabs

SAUCELABS/TEST AUTOMATION EXAMPLES

- Selenium ⬅
- Cypress
- Playwright
- Testcafe
- Robot Framework
- Appium
- Cucumber and Gherkin

"Top Test Automation Frameworks in 2024", Saucelabs

Selenium with Python

```python
1  # Create a new browser instance
2  driver = webdriver.Chrome()
3
4  # Navigate to the website
5  driver.get("https://www.example.com")
6
7  # Find the search bar element and enter a query
8  search_bar = driver.find_element_by_name("q")
9  search_bar.send_keys("test automation frameworks")
10
11  # Find the search button and click it
12  search_button = driver.find_element_by_name("btnI")
13  search_button.click()
14
15  # Verify that the search results page has loaded
16  assert "Example" in driver.title
17
18  # Close the browser
19  driver.close()
```

Saucelabs Bottom Line
Selenium is a robust and widely used test automation framework, but it can be difficult to learn and slow to run tests. It's best suited for testing web applications on multiple browsers.

```
Cypress with JavaScript

1  describe('My Test', () => {
2    it('Visits the homepage and checks the title', () => {
3      cy.visit('https://www.example.com');
4      cy.title().should('include', 'Example');
5    });
6  });
```

Saucelabs Bottom Line
Cypress's focus on web application testing and its requirement for JavaScript knowledge may only make it suitable for some projects. Still, Cypress has one of the most active and vibrant communities of the frameworks in this list; users can find support on its public Slack channel, GitHub issues, or the official forum on Cypress's website.

```
Playwright Example

 1  const { chromium } = require('playwright');
 2
 3  (async () => {
 4    const browser = await chromium.launch();
 5    const page = await browser.newPage();
 6    await page.goto('https://www.example.com');
 7
 8    const title = await page.title();
 9    console.log(title);
10
11    await browser.close();
12  })();
```

Saucelabs Bottom Line
Playwright is a promising new test automation framework that's simple and fast, with good cross-platform and cross-browser support. It may not have as many features and resources as some of the more established frameworks, but it's worth keeping an eye on as it continues to grow and mature.

TestCafe with JavaScript

```javascript
1  import { Selector } from "testcafe";
2
3  fixture`My Test`.page`https://www.example.com`;
4
5  test("Does something", async t => {
6    const searchBar = Selector("input[name=q]");
7    const searchButton = Selector("input[name=btn
8
9    await t
10     .typeText(searchBar, "test automation fram
11     .click(searchButton)
12     .expect(Selector("title").innerText)
13     .contains("Example");
14 });
```

Saucelabs Bottom Line
TestCafe is a commercial test automation framework designed to be fast and easy to use. Its support for multiple programming languages and good cross-browser support make it an attractive choice; however, its focus on web application testing and its cost may not make it suitable for all projects.

Appium with Python

```python
1   from appium import webdriver
2
3   desired_caps = {
4     "platformName": "iOS",
5     "deviceName": "iPhone 8",
6     "app": "/path/to/my.app"
7   }
8
9   driver = webdriver.Remote("http://example.com", desired_caps
10
11  # Find the search bar element and enter a query
12  search_bar = driver.find_element_by_name("q")
13  search_bar.send_keys("test automation frameworks")
14
15  # Find the search button and click it
16  search_button = driver.find_element_by_name("btnI")
17  search_button.click()
18
19  # Verify that the search results page has loaded
20  assert "Example" in driver.title
```

Saucelabs Bottom Line
Appium is specifically designed for testing mobile applications. Its support for multiple programming languages and good cross-platform support make it an attractive choice for many developers. However, its reliance on coding and its potential for reliability issues may not make it suitable for all projects

```
Robot Test Example

1  *** Test Cases ***
2  Check Title
3    Open Browser  https://www.example.com  chrome
4    Title Should Be  Example Domain
5    Close Browser
```

Saucelabs Bottom Line
Robot Framework is an open-source test automation framework with simple syntax and good cross-platform support. Still, its focus on web application testing and limited feature set may not make it suitable for all projects.

Examples

As previously mentioned, Cucumber tests are written in a simple syntax that's easy for non-technical stakeholders to understand. Here's an example of what a simple Cucumber test looks like:

Feature: My Test As a user I want to search for something on the website So that I can find what I'm looking for

Scenario: Search for something Given I am on the homepage When I enter "test automation frameworks" in the search bar and click the search button Then the search results page should be displayed

Saucelabs Bottom Line

Cucumber is an open-source test automation framework designed to be easy to use and maintain. Its simple syntax and good cross-platform support make it an attractive choice for developers. Still, its focus on web application testing and limited feature set may only make it suitable for some projects.