

Functional Test Automation Options

Dev/Test

Development tests are more difficult to create, but are also more flexible. In addition to UI/page-level testing, they could help with system-level testing, for example we could send/receive email, probably perform batch testing, etc.

Capybara

- Platforms: Linux, OS X, Windows
- Browsers: Rack, Firefox, WebKit
- Multiple drivers:
 - Rack (No JS)
 - Selenium (JS, drives Firefox)
 - Capybara-webkit (JS, headless, works on OS X or Linux)

PROS:

- Integrates with RSpec
- Wide adoption
- Active development
- Nice DSL e.g. `click_link('id-of-link')`

CONS:

- Requires developers to create a script
- Limited support for browsers
- Default Selenium driver (Firefox) is slower

Selenium

- Platforms: Linux, OS X, Windows
- Browsers: All

PROS:

- Wide adoption
- Supports all browsers

CONS:

- Requires developers to create a script
- Clunky syntax e.g. `ancestor_element.find_element(:tag_name, 'a').click`
- No direct integration to RSpec

Watir

- Platforms: Linux, OS X, Windows
- Browsers: IE, Firefox, Chrome
- Inactive?

PROS:

- Integrates with RSpec
- Nice DSL

CONS:

- Requires developers to create a script
- Inactive?

GUI

GUI tests would be easier to create, but less flexible. They would be most applicable to UI/web page interactions rather than system functionality.

Selenium IDE

- Platforms: Linux, OS X, Windows

- Browsers: Firefox

PROS:

- Anybody can create a script

CONS:

- Only supports Firefox
- Slower (drives browser)
- No integration with developer workflow

Sahi Open Source

- Platforms: Linux, OS X, Windows
- Browsers: All

PROS:

- Anybody can create a script
- Supports all platforms and browsers

CONS:

- Limited open source version
- Slower (drives browser)
- No integration with developer workflow

Twist

- Platforms: Linux, OS X, Windows
- Browsers: All

PROS:

- Anybody can create a script
- Supports all platforms and browsers

CONS:

- \$\$\$
- Slower (drives browser)
- No integration with developer workflow

Fake

- Platforms: OS X
- Browsers: Safari

PROS:

- Anybody can create a script

CONS:

- Only supports OS X & Safari
- Slower (drives browser)
- No integration with developer workflow