

CSE-322

Software Engineering Laboratory

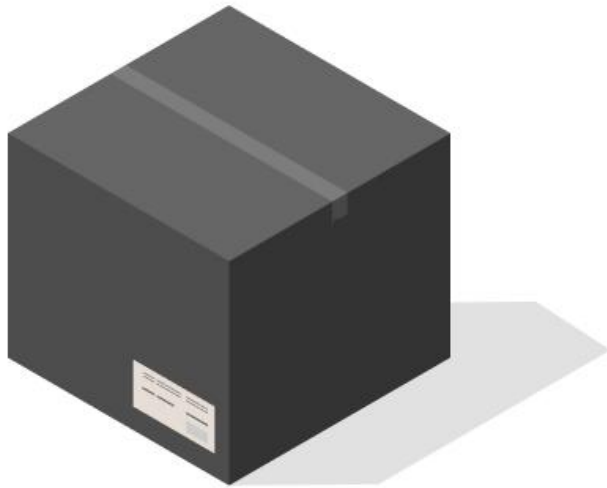
Black Box Testing

using Selenium

Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

Software Testing

QA testers



**Black box – we do not
know anything**

Developers

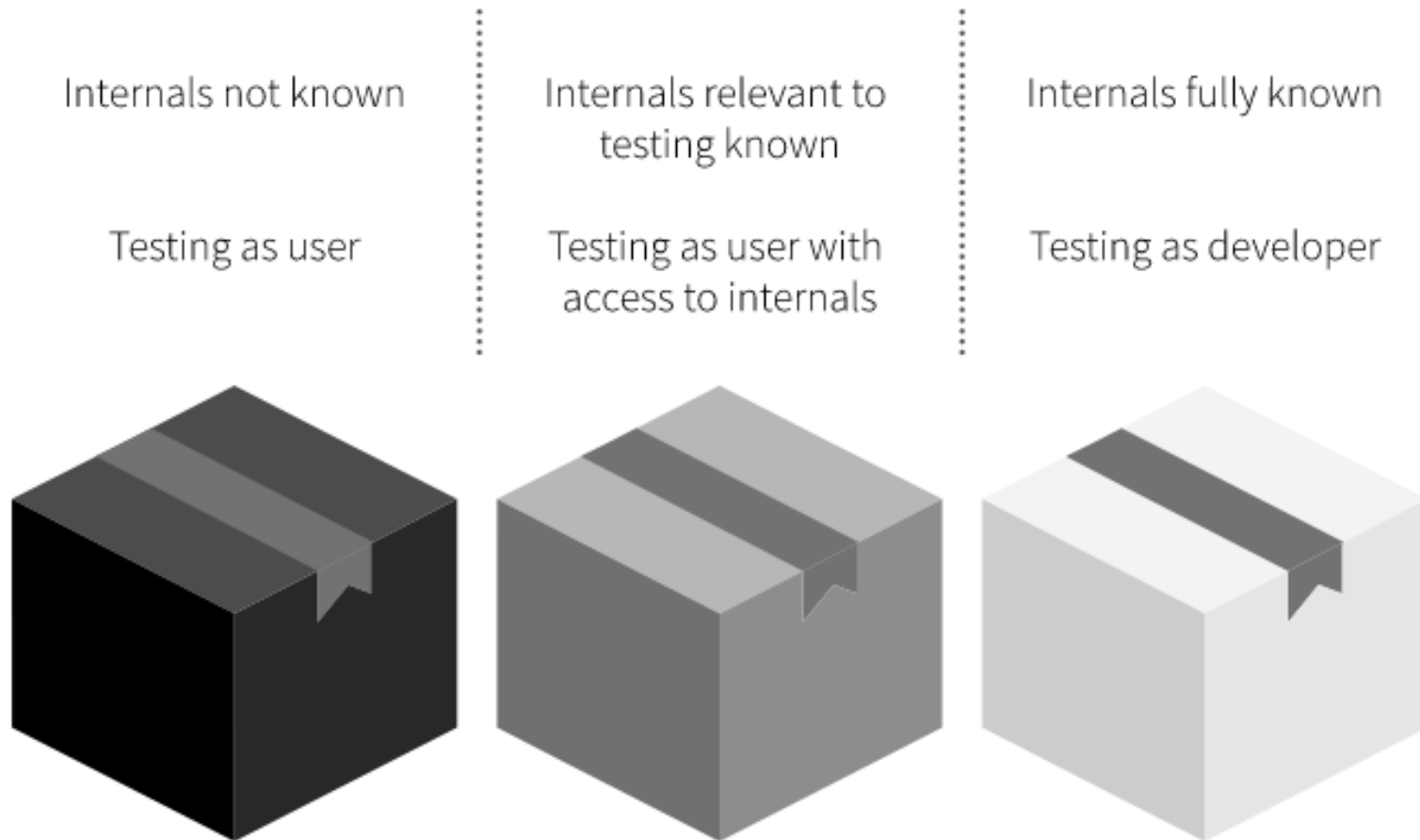


**White box – we know
everything**

Read more:

<https://www.careerist.com/insights/a-guide-to-white-box-black-box-and-gray-box-testing>

Software Testing



Read more:

<https://www.careerist.com/insights/a-guide-to-white-box-black-box-and-gray-box-testing>

Selenium

Selenium is an automation tool which is widely used to automate testing and web scrapping.

Run the following in command prompt to install it.

```
pip install selenium
```

```
pip install webdriver-manager
```

Documentation: <https://selenium-python.readthedocs.io/index.html>

Running a webdriver

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager

driver = webdriver.Chrome(ChromeDriverManager().install())
driver.get("https://www.google.com")
print(driver.title)
```

Functionalities

- To go to a url:
`driver.get("www.yahoo.com")`
- To view current url:
`driver.current_url`
- To select an element:
`elem = driver.find_element_by_id('...')`
`elem = driver.find_element_by_class('...')`
`elem = driver.find_element_by_xpath('...')`
- To click on a button
`elem.click()`
- To get the text content of an element:
`elem.text`
- To type something in an input-box:
`elem.clear()`
`elem.send_keys("Hello World")`

Functionalities

- To press a key*:

```
from selenium.webdriver.common.keys import Keys  
elem.send_keys('seleniumhq' + Keys.RETURN)
```

- To throw an error on a condition, use assert.

```
assert x == 5      #Raise an error if x is not equal to 5
```

- To pause:

```
import time  
time.sleep(5)      #Pause for 5 seconds before the next action
```

- To close the browser window:

```
driver.quit()
```

*Read more about keys:

<https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.keys.html>

Headless Driver

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.options import Options
options = Options()
options.headless = True

driver = webdriver.Chrome(ChromeDriverManager().install(), options=options)
```


Load Testing

```
import time
import threading

def do_work(n):
    time.sleep(n)
    print(f'work done in {n} seconds')

t1 = threading.Thread(target=do_work, args=[4])
t2 = threading.Thread(target=do_work, args=[2])

t1.start()
t2.start()
```

Load Testing

```
import time
import threading

def do_work(n):
    time.sleep(n)
    print(f'work done in {n} seconds')

for i in range(1, 1001):
    t = threading.Thread(target=do_work, args=[i])
    t.start()
```

Load Testing

```
import time
import threading

def do_simulation(...):
    #write code for simulation using WebDriver

for i in [...test_cases...]:
    t = threading.Thread(target=do_simulation, args=[...])
    t.start()
```

Black Box Testing

1. Requirement and specifications will be examined
2. Positive inputs, as well as negative inputs, will be given to the system to verify it
3. Outputs for the tests will be defined earlier
4. Test cases will be executed
5. Actual outputs and expected outputs will be compared
6. Fixed issues will be retested

Source: <https://www.testbytes.net/blog/black-box-testing/>

Reference

1. <https://pypi.org/project/selenium/>
2. <https://selenium-python.readthedocs.io/>
3. <https://realpython.com/modern-web-automation-with-python-and-selenium/#test-driving-a-headless-browser>