IT Home

IT Home

How to speed up 1k UI tests

Yury Barsukou

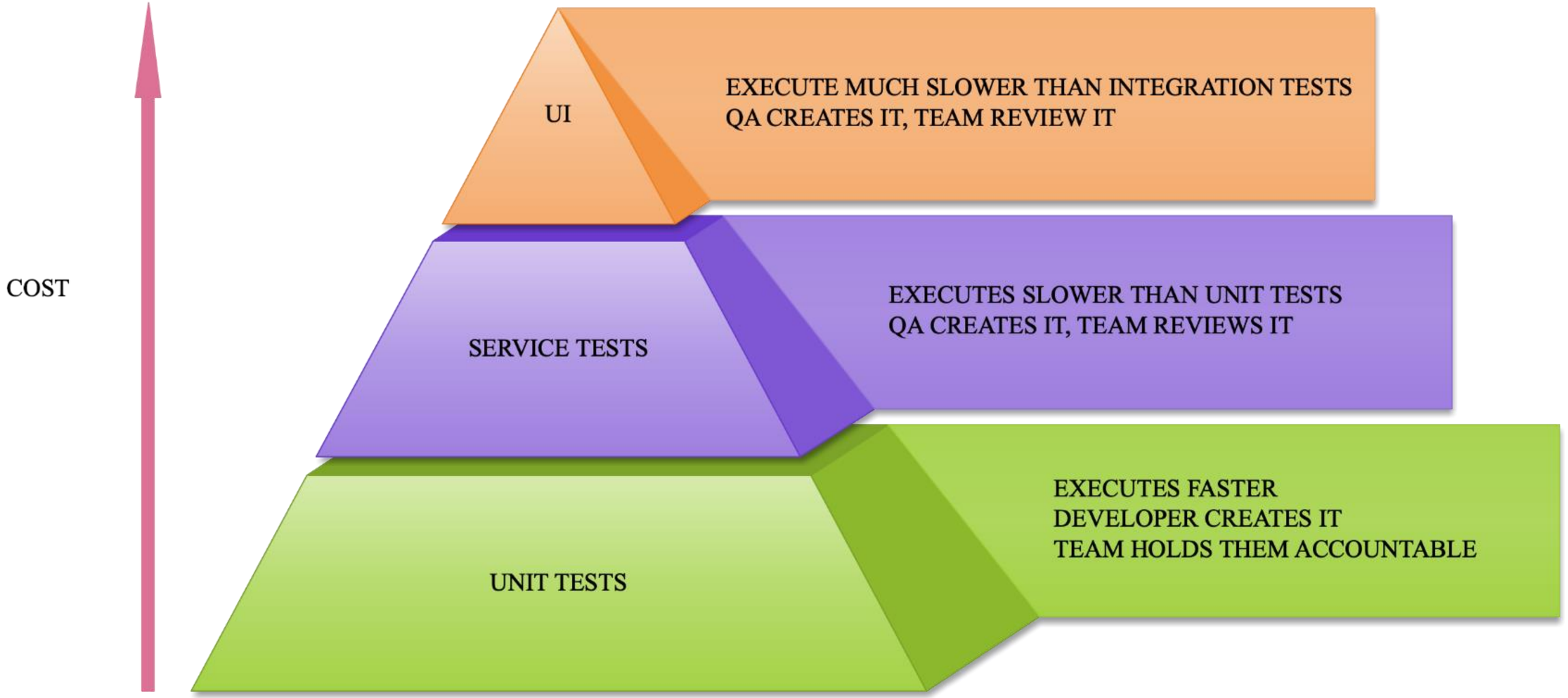
Lead Software Test Automation Engineer

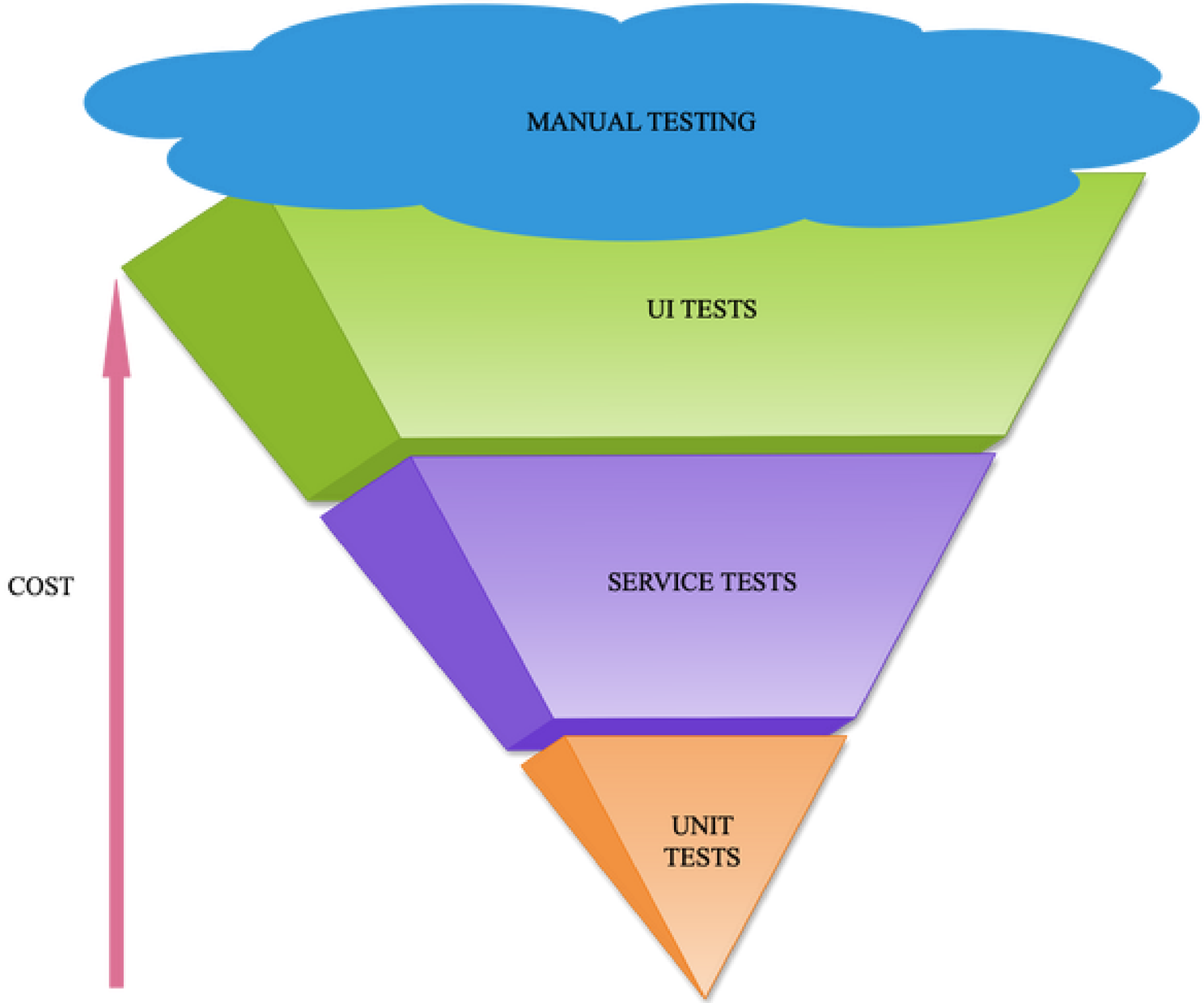
- 11 years **in IT**
- 5 years **in EPAM**
- 3+ years **in Mentoring**
- 1 year **in a Team Lead role**
- 2+ years **in Resource Management**
- I am fond of **playing** guitar and **singing** :)

IT Home

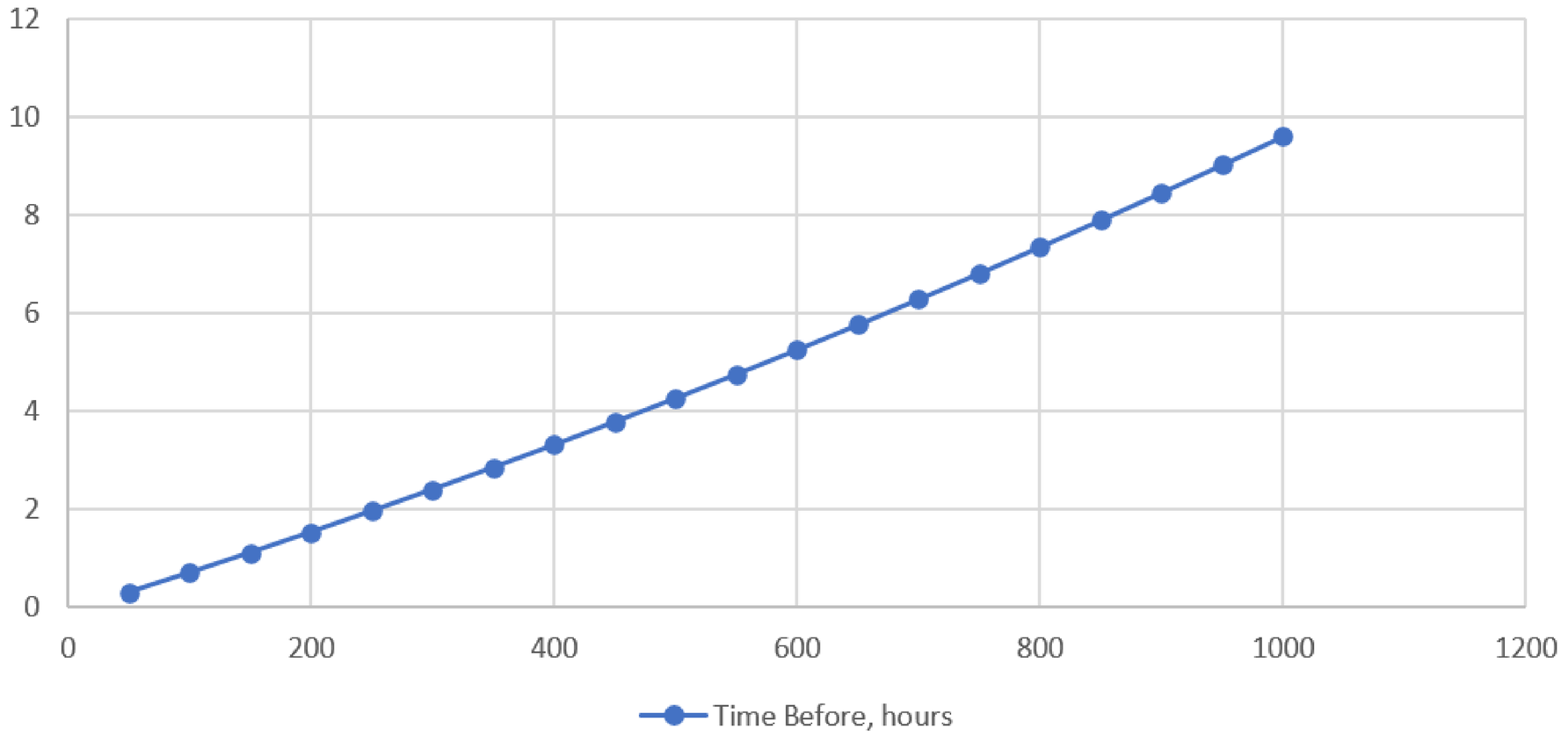
Agenda

TEST PYRAMID


COST
UI
EXECUTE MUCH SLOWER THAN INTEGRATION TESTS
QA CREATES IT, TEAM REVIEW IT
SERVICE TESTS
EXECUTES SLOWER THAN UNIT TESTS
QA CREATES IT, TEAM REVIEWS IT
UNIT TESTS
EXECUTES FASTER
DEVELOPER CREATES IT
TEAM HOLDS THEM ACCOUNTABLE

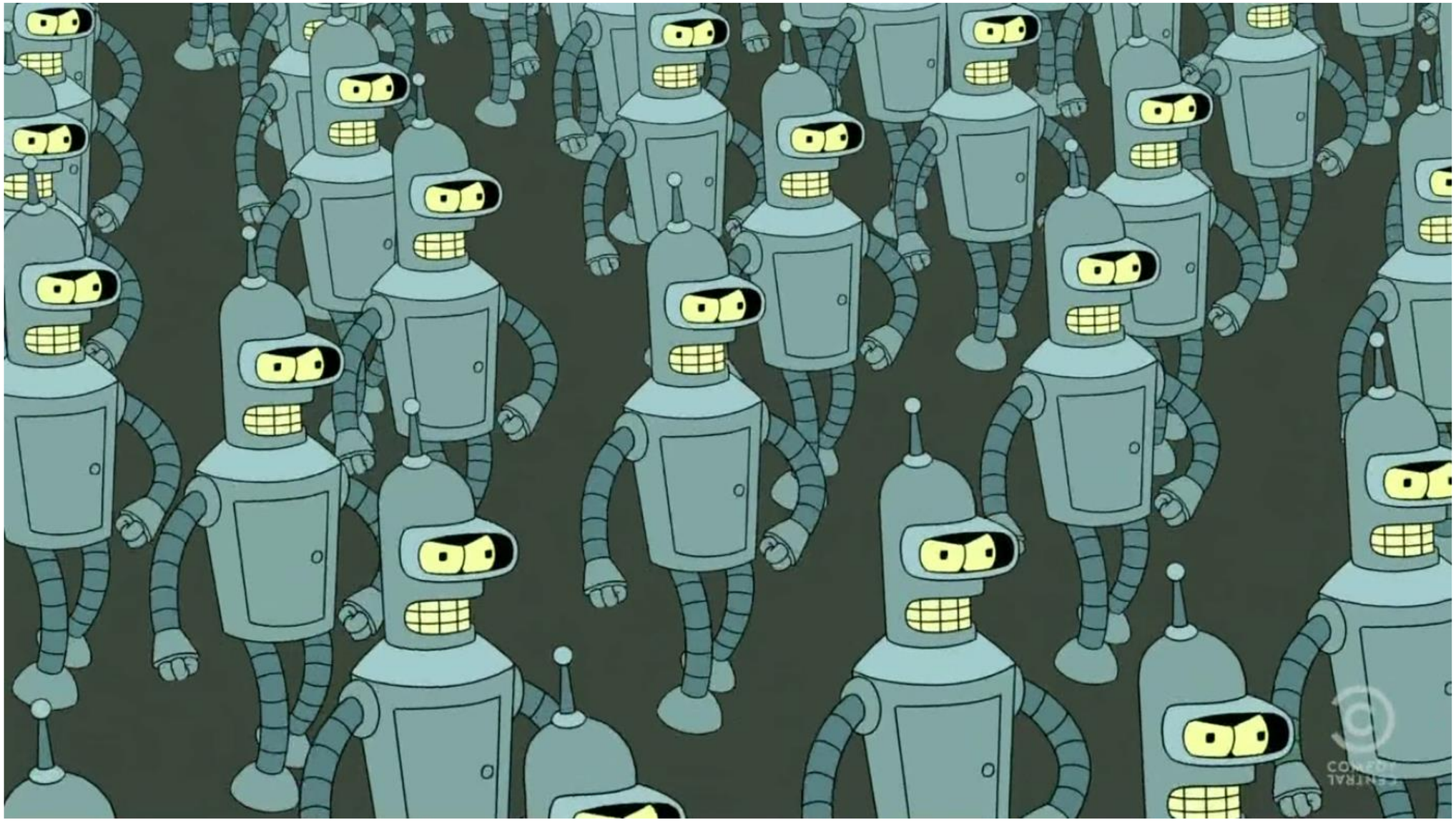

MANUAL TESTING
UI TESTS
SERVICE TESTS
UNIT
TESTS
COST

Regression tests execution. Before applying improvements

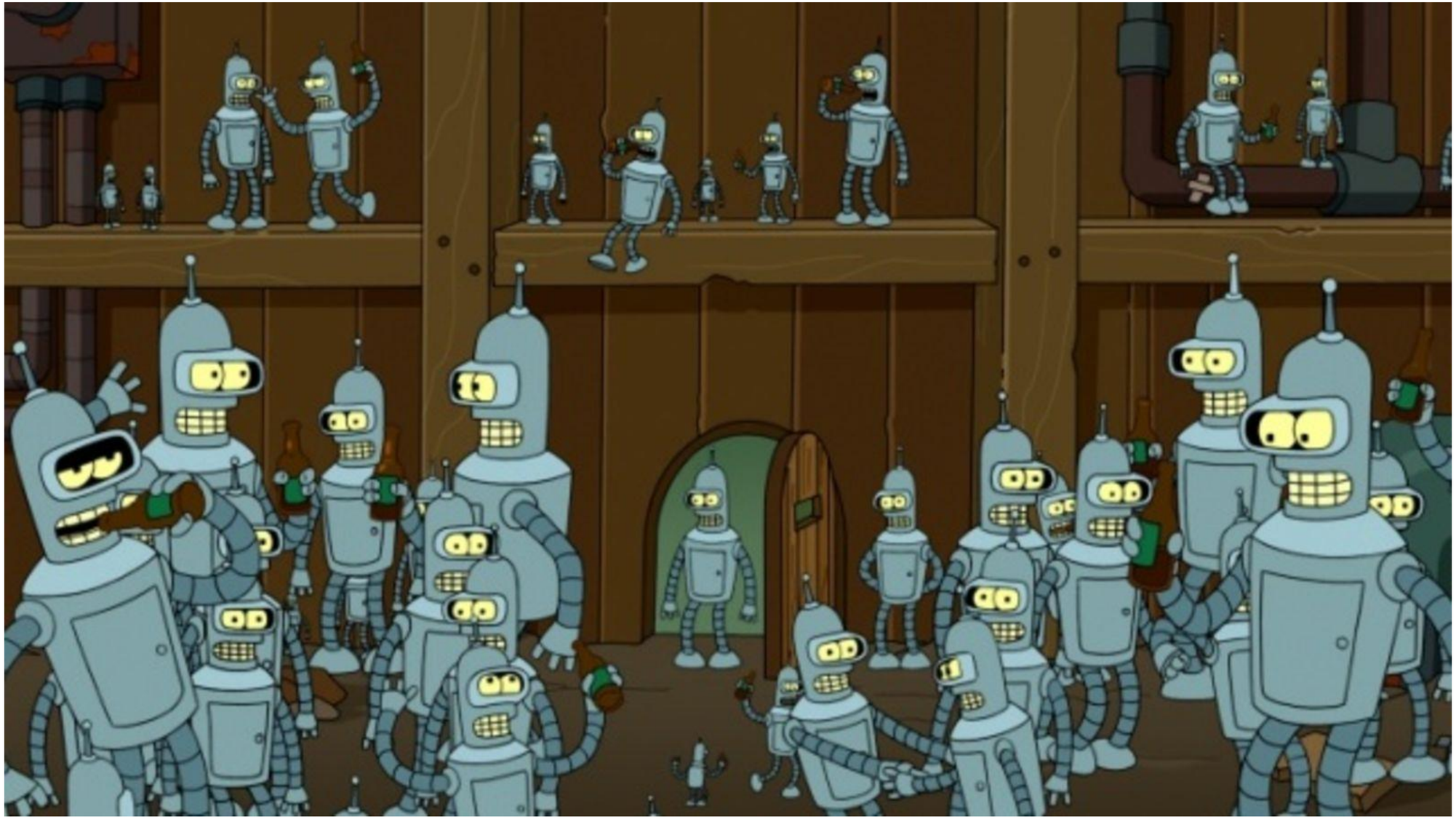


MULTI-THREAD MODE

Expectation



Reality



PROS

- x-times faster execution
- Flexible configuration
- Easy to manage

TESTNG.XML SUITE

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Regression Test Suite" parallel="" thread-count="10">

  <test name="Test 1" group-by-instances="tru
    <packages>
      <package name="com.epam.test.sign_in*"/
      <package name="com.epam.test.sign_up*"/
    </packages>
  </test>

  <test name="Test 2" group-by-instances="tru
    <packages>
      <package name="com.epam.test.cart*"/>
      <package name="com.epam.test.purchase*"/>
    </packages>
  </test>

</suite>
```

```
classes
false
instances
methods
none
tests
true
Press Ctrl+. to choose the selected (or first) suggestion and insert a dot afterwards >> π
```

CONS

- 10 threads != 10 times faster
- Concurrent access to the test data
- Instances limit on a remote GRID
- Limited SUT performance
- Hard to make right choose between variety of parallel modes

COMMON PRECONDITIONS

PROS

- Execute only once for several scenarios
- Reduce code duplication
- Easy to maintain

CONS

- Failed precondition will fail all scenarios
- Mess in test classes

EXPLICIT/FLUENT WAITS

PROS

- Pure WebDriver style
- Perform on a code side
- Full control of what and how long to wait
- Help tests to fail faster
- Variety of conditions from the box

CONS

- Not obvious results when mixing with implicit waits
- Need to know which condition to wait
- Wait too long for an unlikely happened condition

IMPLICIT WAIT INSIDE EXPECTED CONDITION

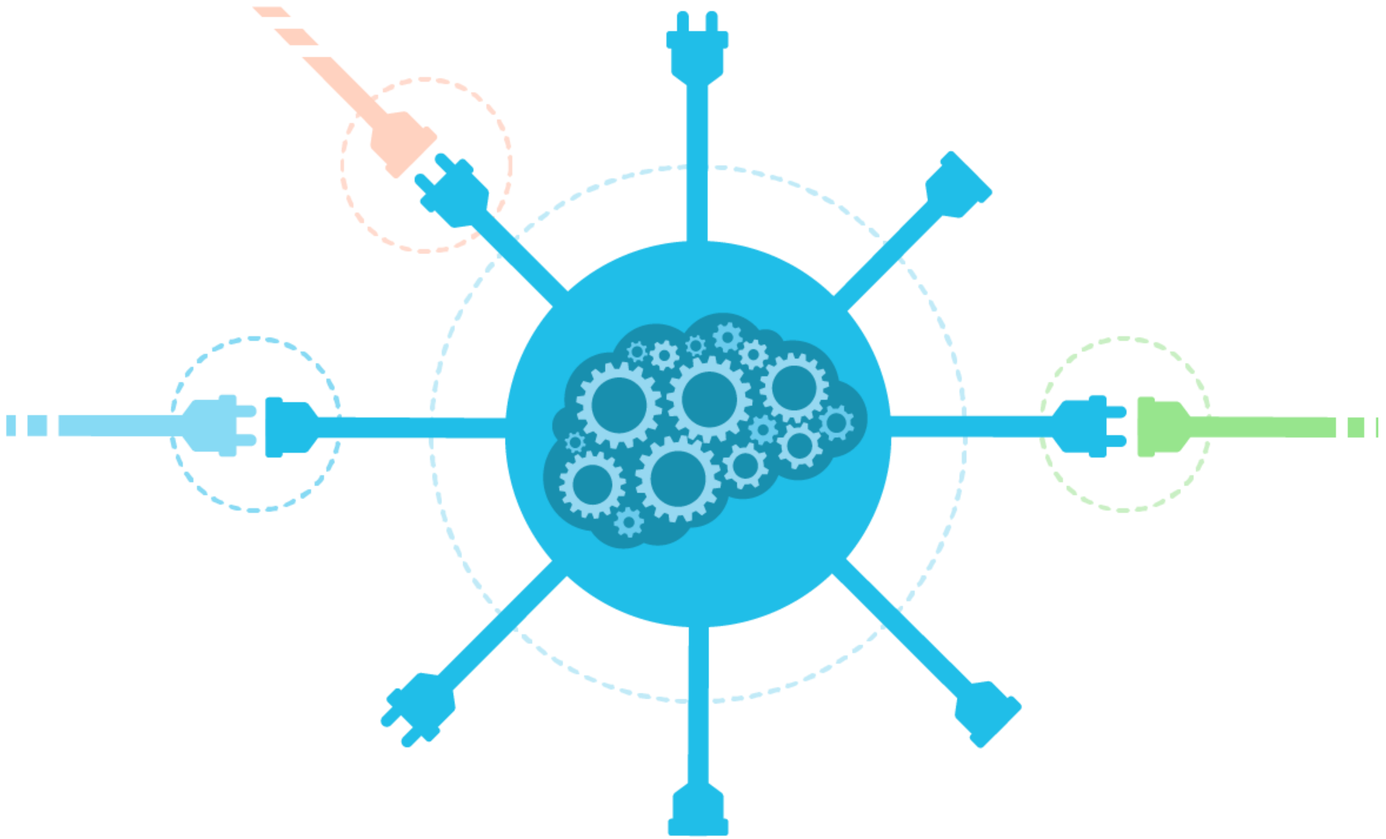
```
public static ExpectedCondition<WebElement> visibilityOfElementLocated(final By locator) {
  return new ExpectedCondition<WebElement>() {
    @Override
    public WebElement apply(WebDriver driver) {
      try {
        return elementIfVisible(driver.findElement(locator));
      } catch (StaleElementReferenceException e) {
        return null;
      }
    }

    @Override
    public String toString() {
      return "visibility of element located by " + locator;
    }
  };
}
```

```
/**

 * This method is affected by the 'implicit wait'
times in force at the time of execution.

 */
WebElement findElement(By by);
```

API SERVICES

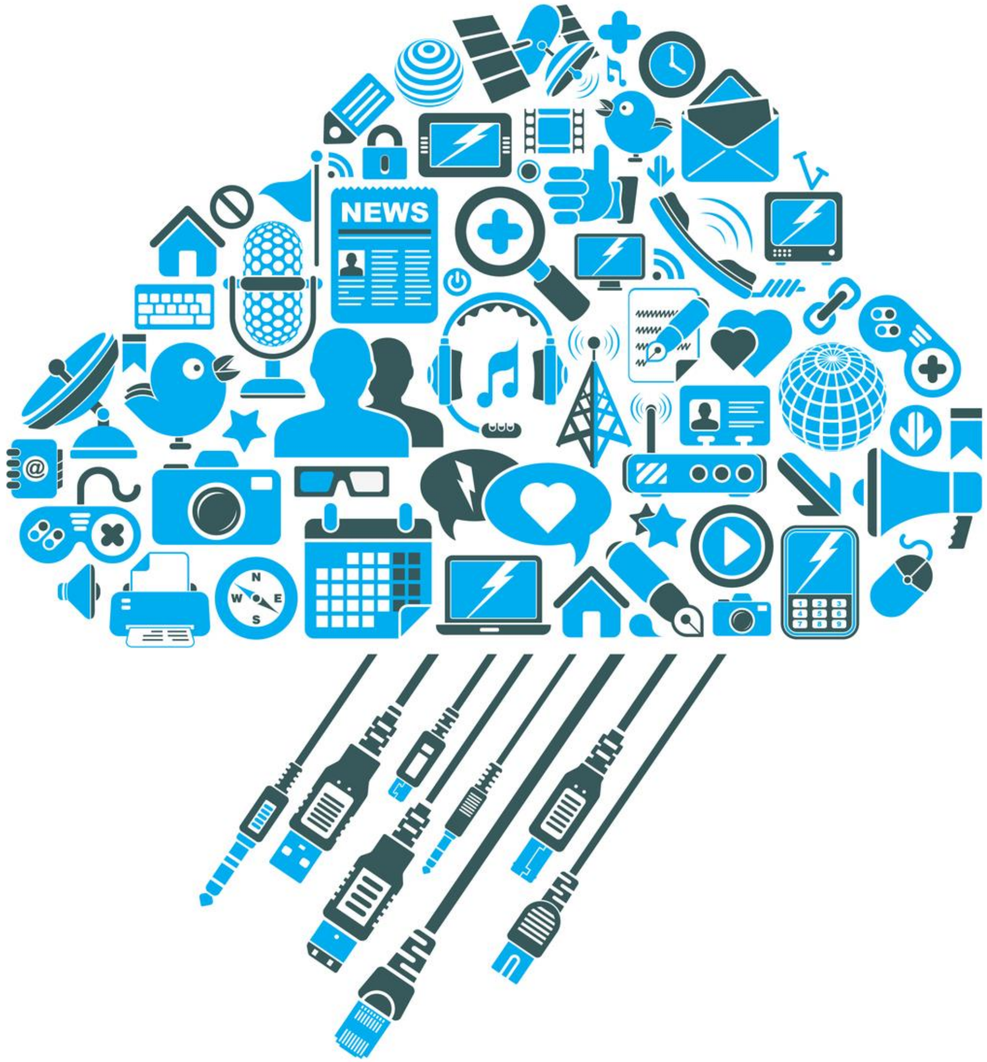
PROS

- Much faster than UI
- Much stable than UI
- Often are implemented by DEVs



CONS

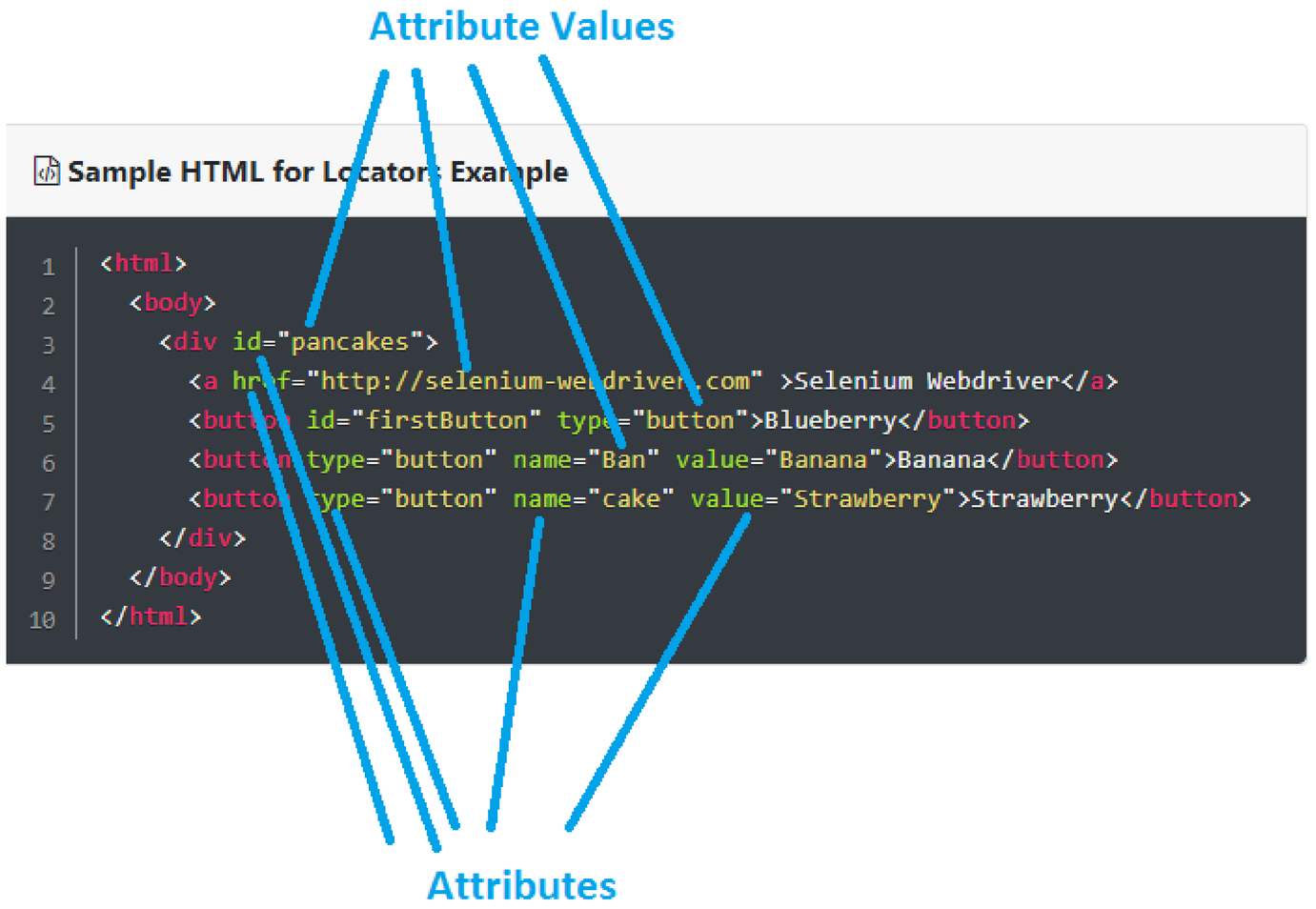
- Need to be implemented by DEVs
- Additional skills are required
- Don't detect UI bugs



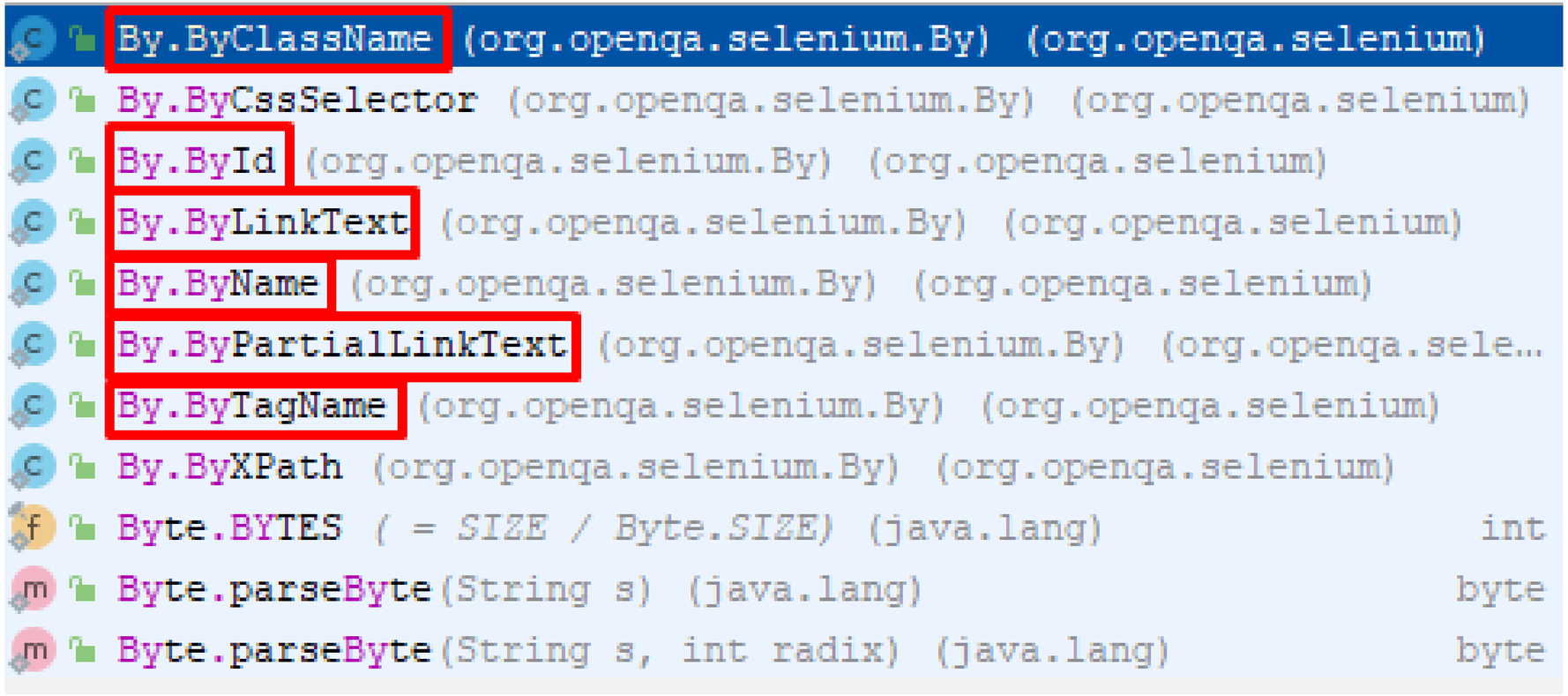
ELEMENT ATTRIBUTES

PROS

- Don't require mastery of writing xpath, css locators
- Easy to use from the box
- Faster than xpath, css

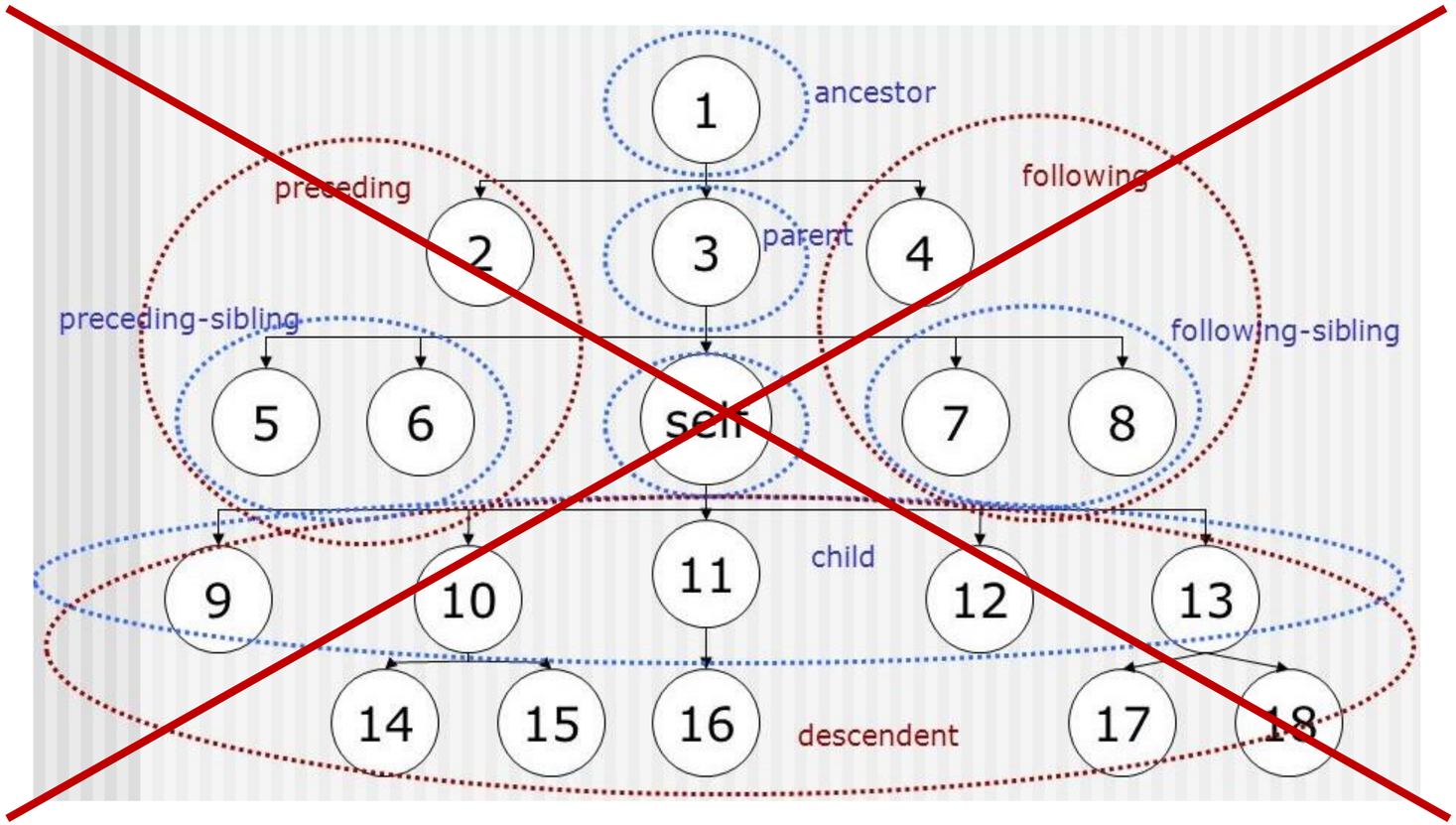


SELENIUM API



CONS

- Often belong to different elements
- Unable to use axis
- Unable to build dynamically



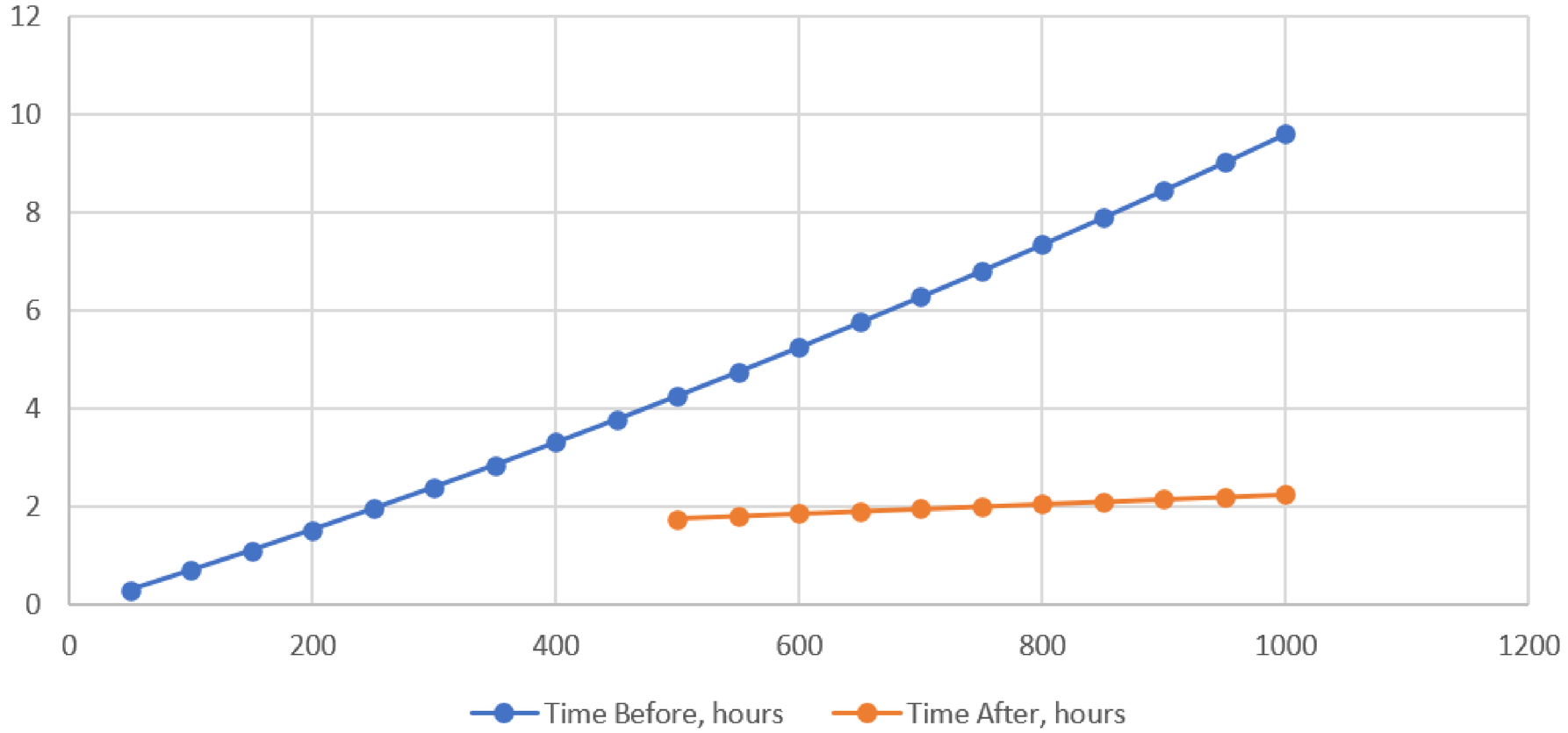
XPATH vs getElementById

Testing in Chrome 75.0.3770 / Windows 10 0.0.0		
Test		**Ops/sec**
xpath evaluate	`var bar = document.evaluate("//*[@id='bar']",document,null,9,null).singleNodeValue;`	3,806 ±1.44% 100% slower
getElementById	`var bar = document.getElementById("bar");`	26,183,888 ±0.42% fastest

EXTRAS

- Don't take screenshots for every action
- Don't modify UI without real need
- Reduce quantity of 'heavy' calls
- Mock Back-End calls
- //div[@class='app'] works faster than //*[@class='app']
- Organize your Test Suite
- Clean up similar test date from one Equivalence Class
- Get rid of *Thread.sleep()* (except debugging or demo purposes)
- Use Retry Analyzers wisely
- Adjust logging
- Consider using headless mode for browsers

Regression tests execution. After applying improvements



USEFUL LINKS

- 6 Ways to Speed Up Your Tests

https://www.thoughtworks.com/insights/blog/6-ways-speed-your-tests

- How to speed up JavaScript testing

https://medium.com/javascript-in-plain-english/how-to-achieve-faster-javascript-automated-ui-tests-399b86e46122

- Примеры ускорения авто тестов

https://www.youtube.com/watch?v=GZXq6oovGcw

Thank you!