



A Guided Tour of Test Automation

George Dinwiddie

My slides are available for you at:

<http://idiacomputing.com/publications.html>

A “Test,” Check, or Scenario



Arrange

Act

Assert

Given

When

Then

“Arrange” – “Given”



- The preconditions for the interaction with the system
- What needs to be true before we can check this bit of functionality?
- In manual testing, this is often the end state of previous interactions, to save the tester work
- In test automation, this is preferred to be an independent setup of known conditions

“Act” – “When”



- The triggering interaction of the test
- How do we make the system do the behavior we're testing?
- In manual testing, this is almost always necessarily through the User Interface (UI)
- In test automation, this can be through the UI, through an Application Program Interface (API) that the UI calls, directly to code within the system, or any means that is accessible

“Assert” – “Then”



- The postconditions for the interaction with the system
- How can we verify that the action performed correctly?
- In manual testing, this may require many steps with the application UI or other tool
- In test automation, this can be through the UI, through an API, directly to code within the system, directly to a database where results are stored, or any means that is accessible

Examples



JUnit example

@Test

```
Public void ensureACoinAddsTimeToAnExpiredMeter() {  
    ParkingMeter parkingMeter = new ParkingMeter();  
    parkingMeter.reset();  
    parkingMeter.acceptQuarter();  
    assertEquals(15, parkingMeter.getMinutesRemaining());  
}
```

Arrange

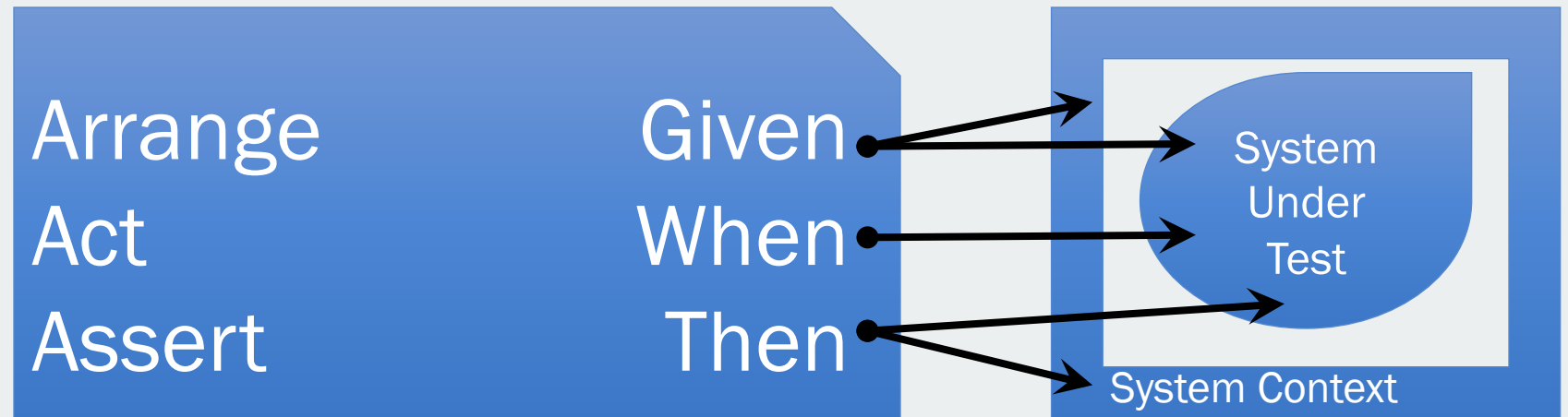
Act

Assert

Cucumber example

Scenario: A coin adds time to an expired meter
 Given the parking meter shows no time remaining
 When a quarter is inserted into the slot
 Then the parking meter shows 15 minutes remaining

Connections to System Under Test



Cucumber Step Definitions



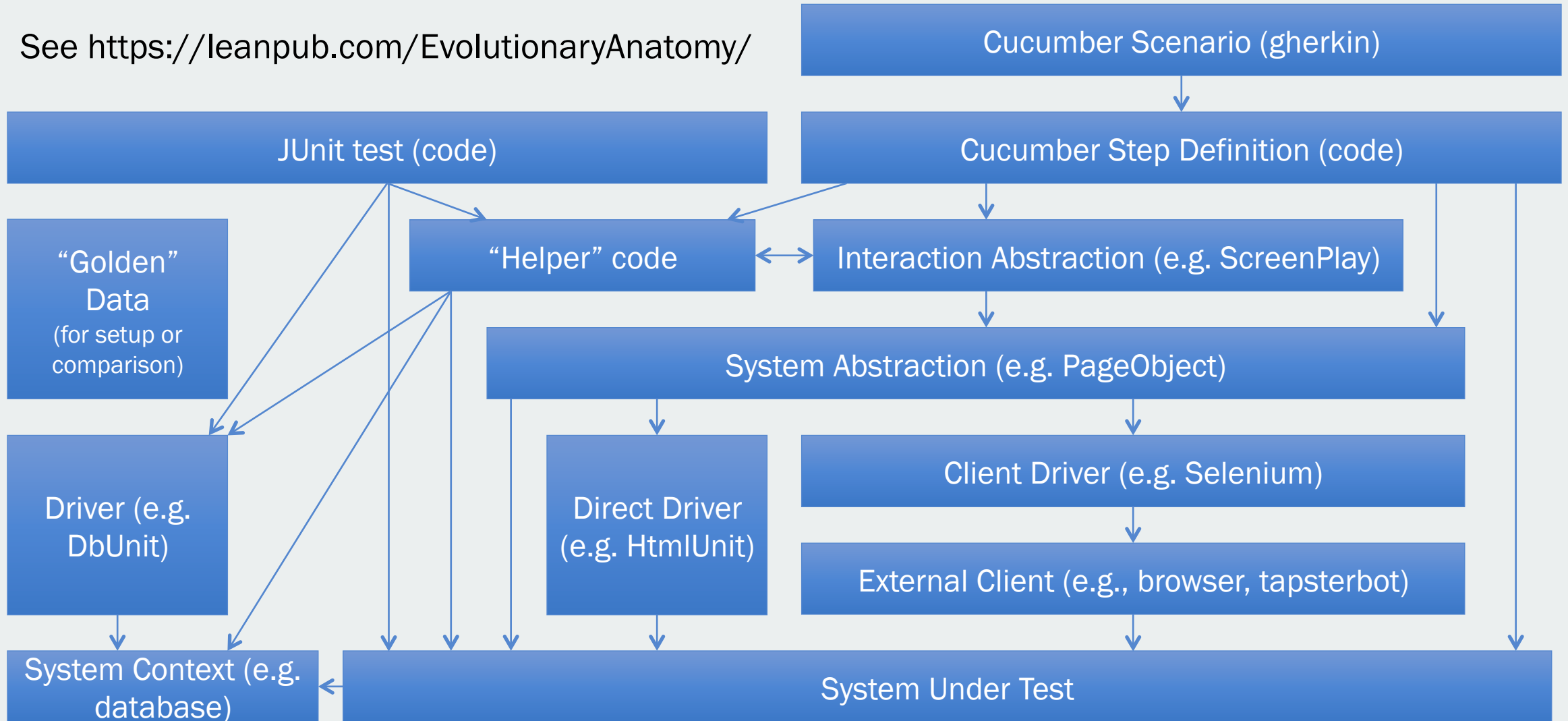
used by Cucumber-JVM example
to connect to system under test

```
public class ParkingMeterStepDefinitions {  
    private ParkingMeter systemUnderTest = new ParkingMeter();  
  
    @Given("the parking meter shows no time remaining")  
    public void the_parking_meter_shows_no_time_remaining() throws Throwable {  
        systemUnderTest.reset();  
    }  
  
    @When("a quarter is inserted into the slot")  
    public void a_quarter_is_inserted_into_the_slot() throws Throwable {  
        systemUnderTest.acceptQuarter();  
    }  
  
    @Then("the parking meter shows {int} minutes remaining")  
    public void the_parking_meter_shows_minutes_remaining(int minutes) throws Throwable {  
        assertEquals(minutes, systemUnderTest.getMinutesRemaining());  
    }  
}
```


Connecting to the System Under Test

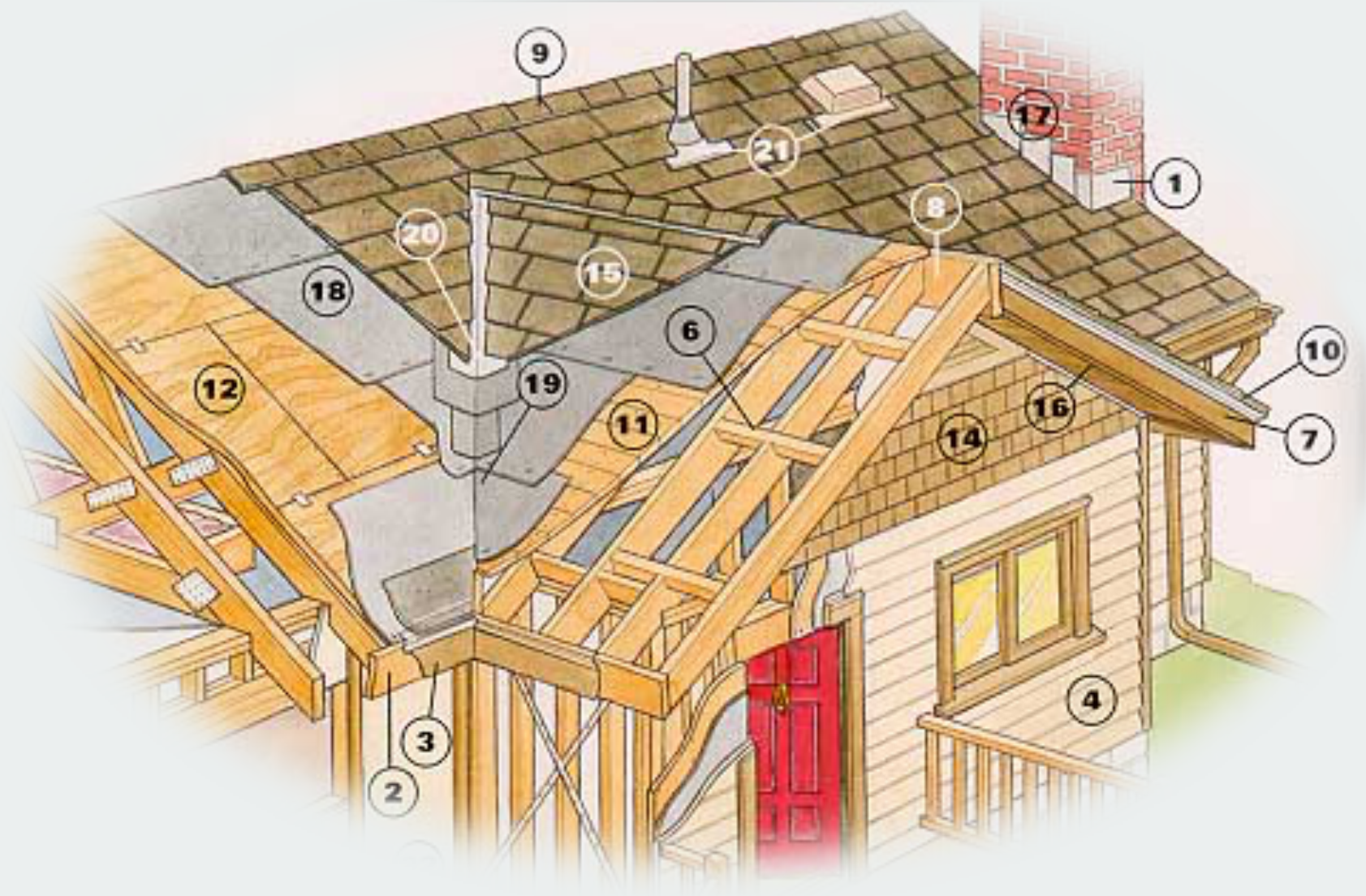


See <https://leanpub.com/EvolutionaryAnatomy/>

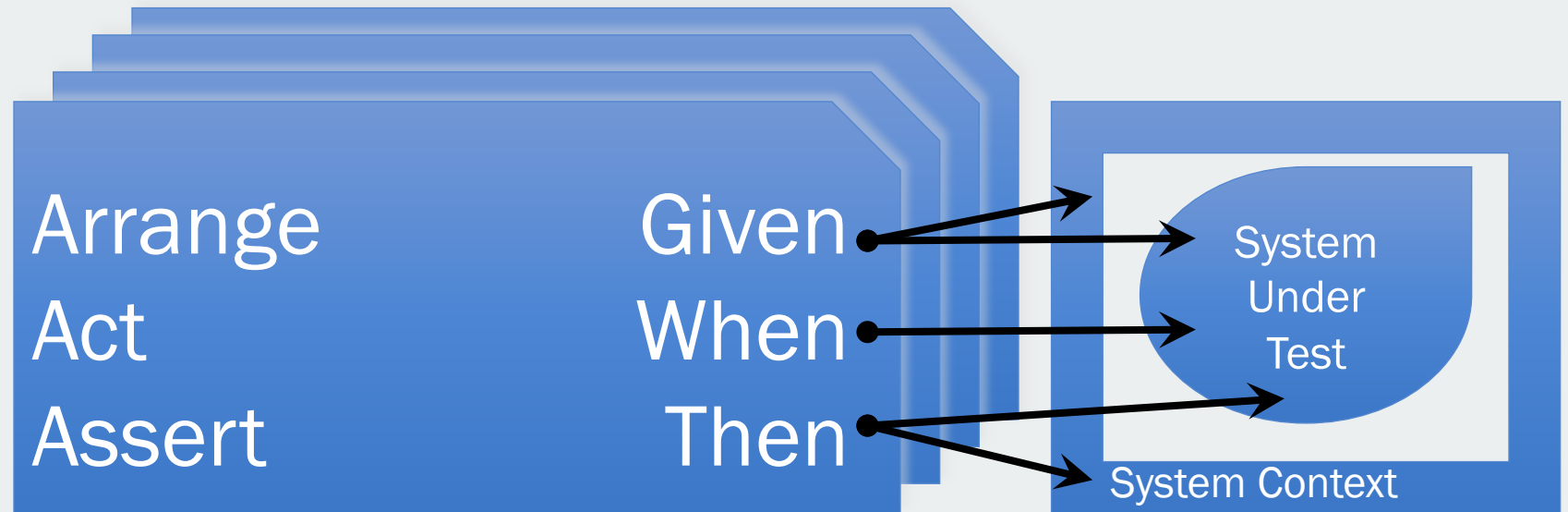


Protection in Depth

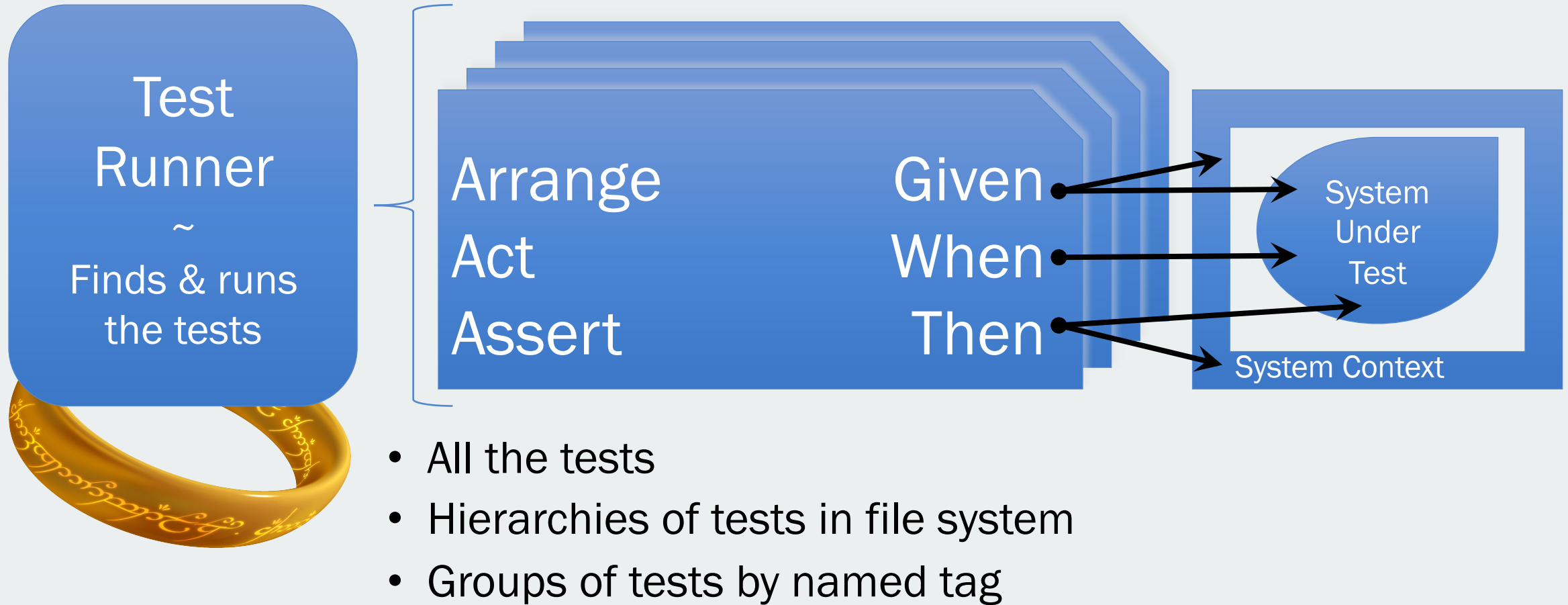
- Unit Tests
- Adapter Tests
- Subsystem Tests
- Integration Tests
- Below-The-GUI
- GUI Tests
- Exploratory Tests



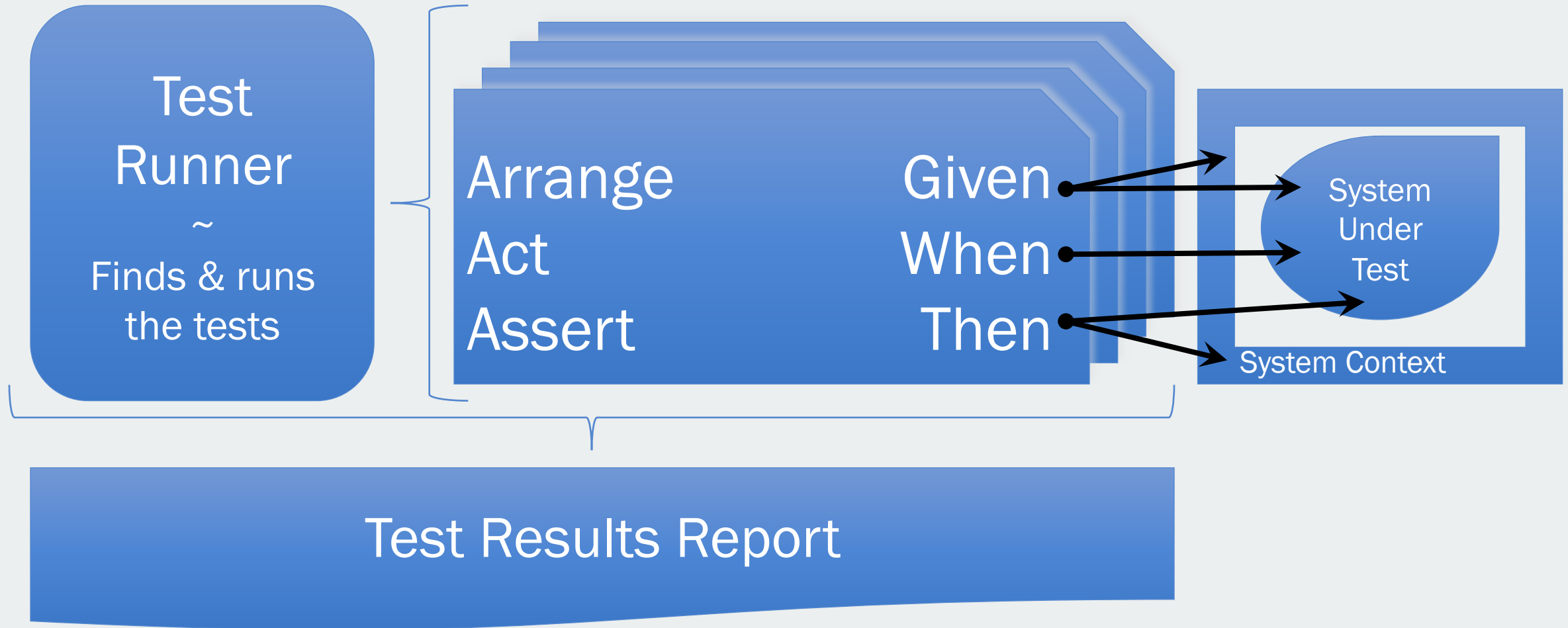
Test Suite or Feature



Test Runner



Reporting Results



Assertion Libraries



Java

- JUnit Assert
- Hamcrest
- AssertJ
- ApprovalTests.Java

Javascript

- Node.js Assert
- Chai
- should.js
- ApprovalTests.NodeJS

.NET

- NSpec Assertions
- Fluent Assertions
- Shouldly
- NFluent
- ApprovalTests.Net

PHP

- PHPUnit Assert
- Leo

and many others



Assertion Styles of Expression

- JUnit Assert

```
assertEquals(15, parkingMeter.getMinutesRemaining());
```

- TestNG Assert

```
assertEquals(parkingMeter.getMinutesRemaining(), 15);
```

- Hamcrest Java

```
assertThat(parkingMeter.getMinutesRemaining(), equalTo(15);
```

- AssertJ

```
assertThat(parkingMeter.getMinutesRemaining()).isEqualTo(15);
```

- Oleaster-Matcher

```
expect(parkingMeter.getMinutesRemaining()).toEqual(15);
```

Consider the way failures are reported, too.

THANKS FOR YOUR ATTENTION!

Feedback welcome! Evaluate my session and get the chance to win a free ticket for the Agile Testing Days 2019.



Go to agiletestingdays.com/session-ratings and give your rating until 30th of November 2018!

