

# **Advanced Acceptance tests**

Aurelijus Banelis

# About me

**Software developer**

**aurelijus@banelis.lt**  
**aurelijus.banelis.lt**



# Why we need tests?



Usually: less stress during deployment, lower risk to loose money because of error / failure

# Unit vs Acceptance tests



**Many scenarios,  
but tested independently**



**Few long and complex  
end-to-end tests**

# When you use Acceptance tests

- 1. Debug**
- 2. Speed**
- 3. Changes**
- 4. Iteraction**

**Consider those  
before choosing  
right tools or  
practices**

# Debug



**Makes debugging harder:**

So choose tools accordingly

**Sync/async**

**Continue from the middle**

**Local, production**



# Debug



Problem: tests are failing

**Some validation error.**

**Exception silently ignored.**

**Requests via SSL (wireshark is useless)**

Cause of the problem (found after debugging):

**JavaScript change field name just before POST**

```
$this->log->crit(__METHOD__ . ':' . __LINE__, "Updated");
```

```
tail -f logfile1.log logfile2.log log/path* | grep CRIT
```

# Speed



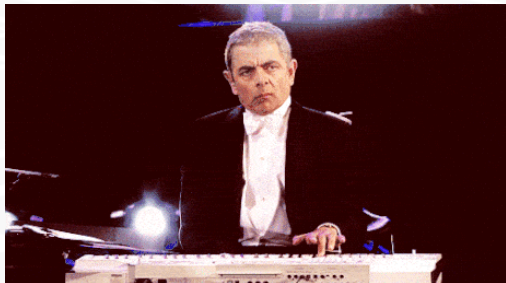
**Make tests  
faster**

**Chrome vs  
Phantom vs  
Firefox**

**Run on CI in  
parallel**



# Speed



Need to rerun test every time, because:

**Unique data needed**

Best practice:

**Fix in one specific environment/suite/scenario**  
**If successful run whole**

```
vendor/bin/codecept run  
--env=testing1-de_DE  
cartSuite  
PayPal.php::seePayPalDetailsPage
```

# Changes



**Solution to  
sensitive tests:**

**class="tst-buy  
js-buy buy"**

**data-tst-id=  
"product-123"**

**DRY,  
constants,  
reuse tests**

# Changes



Requirements (so tests could run faster on Continuous Integration systems like Jenkins):

**JavaScript modal/popup with new field.**  
**Test with primitive/PHP browser.**

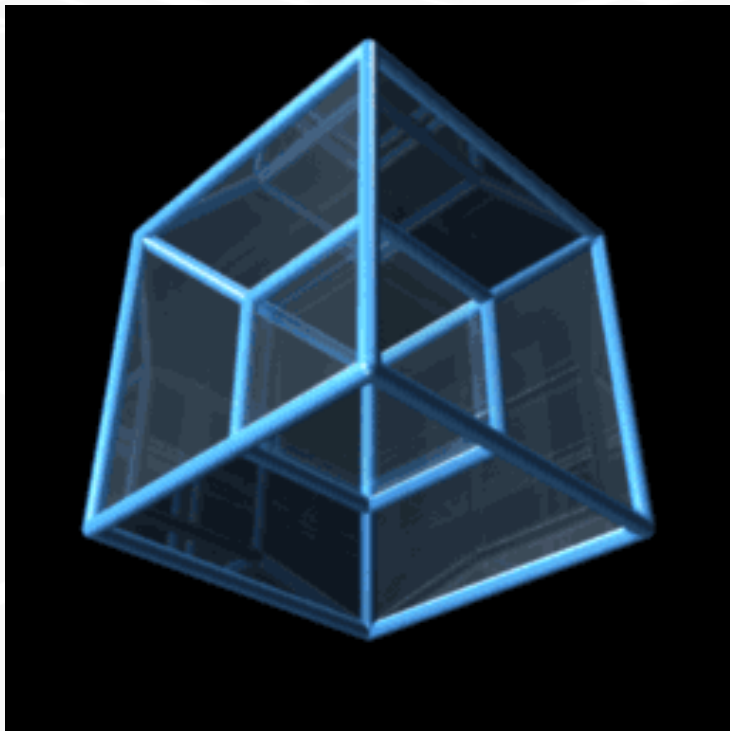
Identifying elements in Acceptance test scenarios. Common mistake: JQuery not equal querySelector

**Search elements by Xpath, CSS, JQuery**

**document.querySelector**

**class="tsy-modal-new" class="tst-modal-edit"**

# Interaction



**Test frameworks simulate only simple interaction:**

**E.g. click, move, mouse over**

Selenium drivers have wait for HTML element to appear functionality:

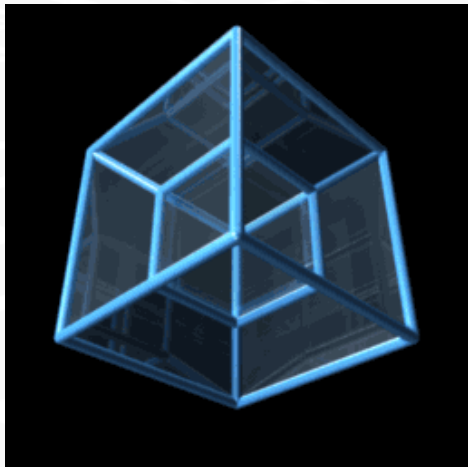
**Wait for Javascript framework**

```
//div[@data-reactid=".0"]
```

But to simulate more complex user inputs/interaction, you need to:

**Call your event functions directly**

# Interaction



Requirements (complex user input, run on production)

## Dragging, touch Minified React.js

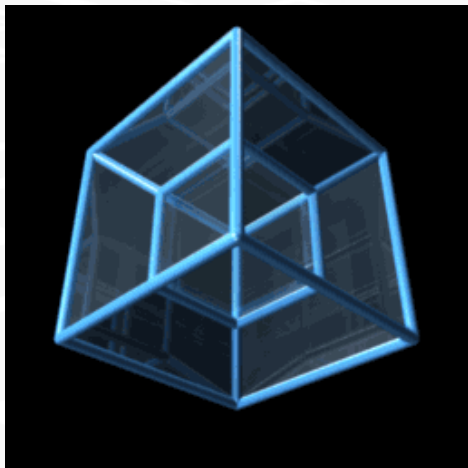
Problems:

**PhantomJs:** not all events exits, not all documented, checked even in source

**Chrome:** touch event invocation not by standards

```
React.addons.TestUtils.Simulate.$eventName(element, event)
```

# Interaction



**Dragging, touch  
Minified React**

PhantomJs not documented, check source  
Chrome: touch event invocation not standards

**React.min.js does  
not have TestUtils  
addon**

**React.addons.TestUtils.Simulate.*\$eventName*(element, event)**

**Rebuild react locally with one line changed**



# Summary

**Change your software  
to test it easier**

# Questions

- **Short intro into Acceptance tests**
- **Tips and tricks from practice**
  - **Debug**
  - **Speed**
  - **Changes**
  - **Iteration**

# References

- [https://en.wikipedia.org/wiki/Acceptance\\_testing](https://en.wikipedia.org/wiki/Acceptance_testing)
- <http://codeception.com/>
- <http://behat.org/>
- <http://casperjs.org/>
- <http://www.seleniumhq.org/>
- <https://devblog.supportbee.com/2014/10/27/setting-up-cucumber-to-run-with-Chrome-on-Linux/>
- <https://github.com/ariya/phantomjs/tree/master/src/qt/qtwebkit/Source/WebCore/dom>
- <https://chromium.googlesource.com/chromium/src.git>
- <https://facebook.github.io/react/docs/test-utils.html>