T9
Mobile Testing
2019-05-02 11:15

Visual Regression Testing: A Critical Part of a Mobile Testing Strategy

Presented by:

Dmitry Vinnik

Salesforce

Brought to you by:

888-268-8770 · 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

Dmitry Vinnik

Dmitry Vinnik is a Lead Software Engineer at Salesforce and has been passionate about Software Quality since the very beginning of his career. He started out as a Quality Engineer, and was able to bring test expertise into his current Software Engineering role to ensure delivery of a high quality product. Dmitry is also a Scrum Master focused on making his team more efficient and productive. His background involves studying medicine and bioinformatics in addition to software and quality engineering.

Uphill Battle Of Mobile Visual Regression

Dmitry Vinnik

@DmitryVinnik

dvinnik.com

What are our goals?

Goals

Visual Testing (VT):
What, Why, How and When

New Perspective on Test Pyramid

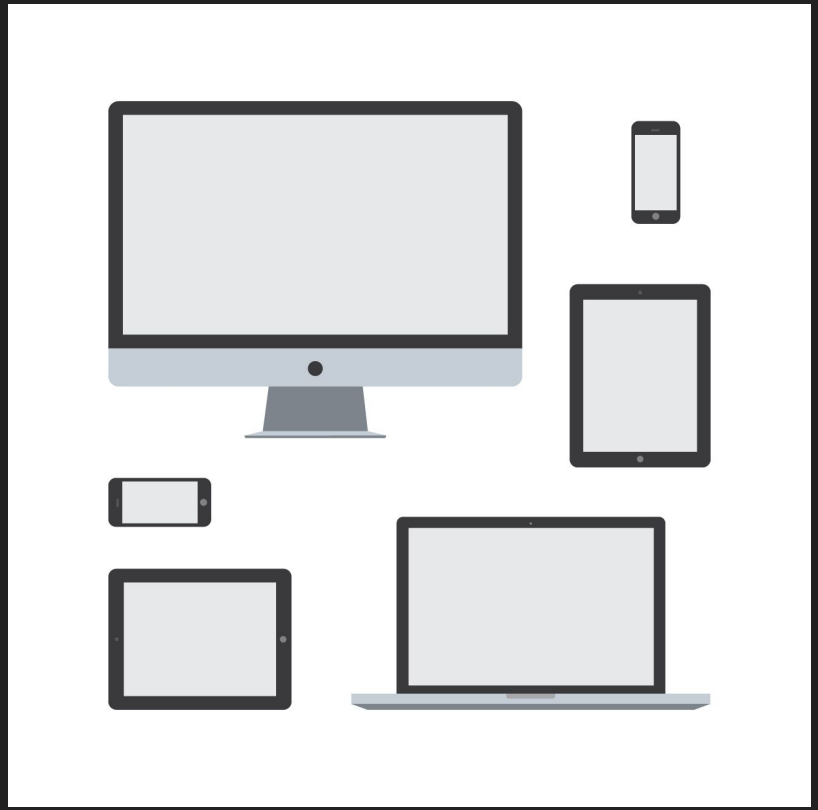
Finding VT Tool For Your Team

Target Platforms?

Target Platforms

iOS

Android

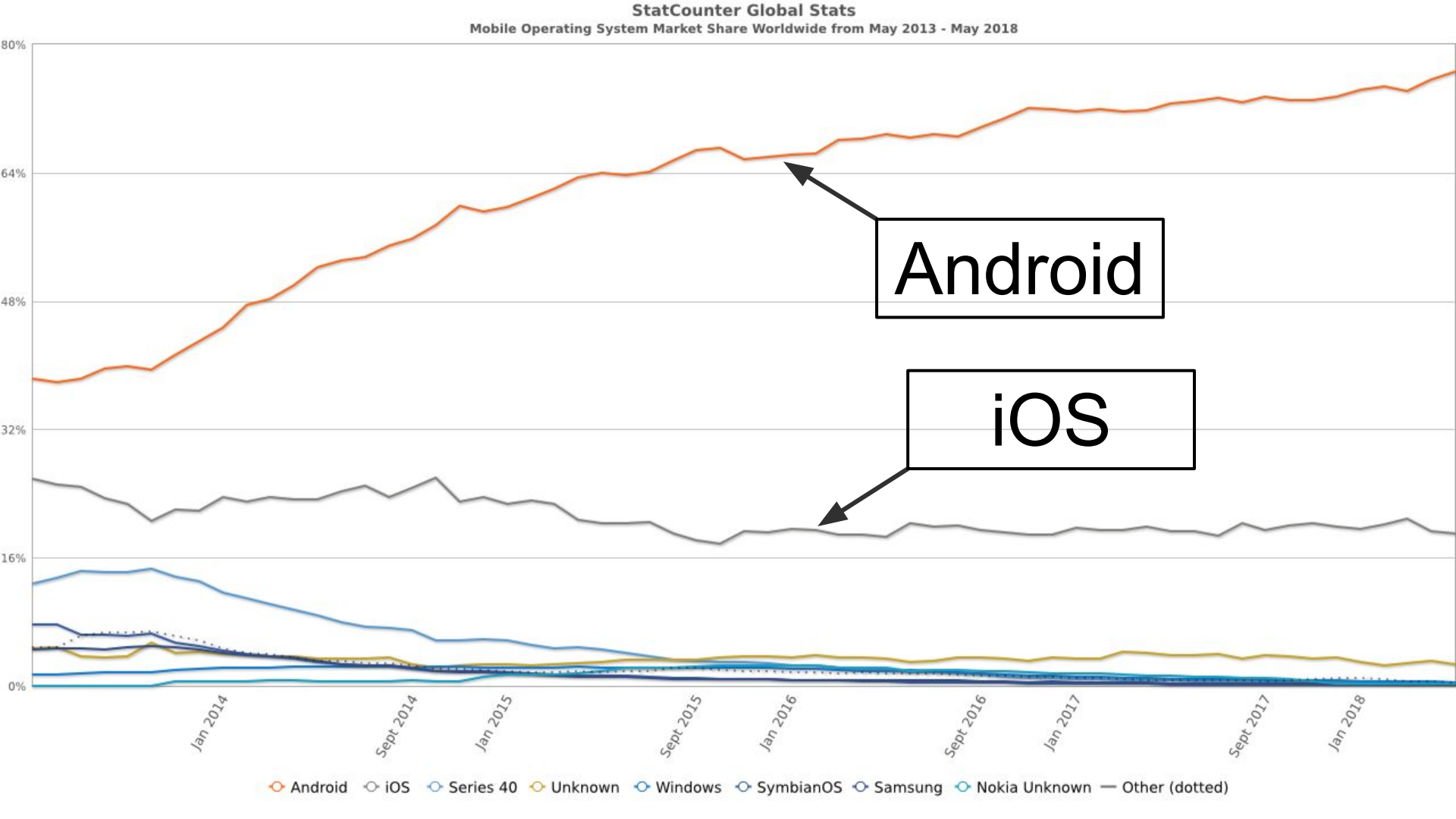


Responsive Apps

Hybrid Apps

@DmitryVinnik

dvinnik.com

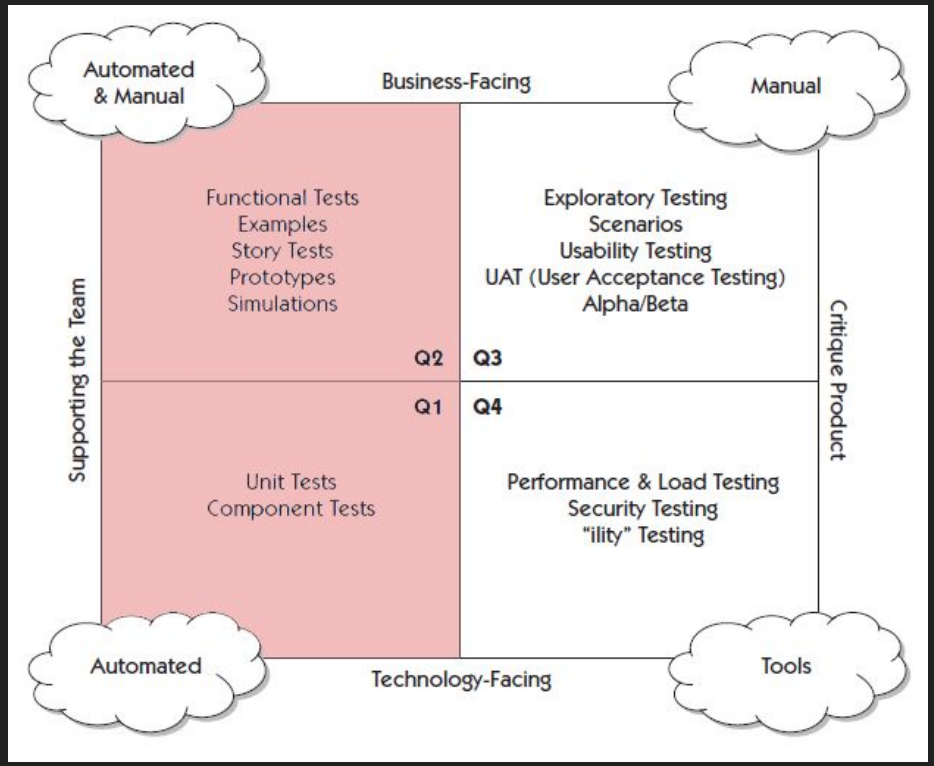

StatCounter Global Stats
Mobile Operating System Market Share Worldwide from May 2013 - May 2018
80%
64%
48%
32%
16%
0%
Android
iOS
Jan 2014
Sept 2014
Jan 2015
Sept 2015
Jan 2016
Sept 2016
Jan 2017
Sept 2017
Jan 2018
Android
iOS
Series 40
Unknown
Windows
SymbianOS
Samsung
Nokia Unknown
Other (dotted)

How do we organize QA Effort?

From the Top Level

Testing Quadrants

Agile Testing Quadrants



by Lisa Crispin and Janet Gregory

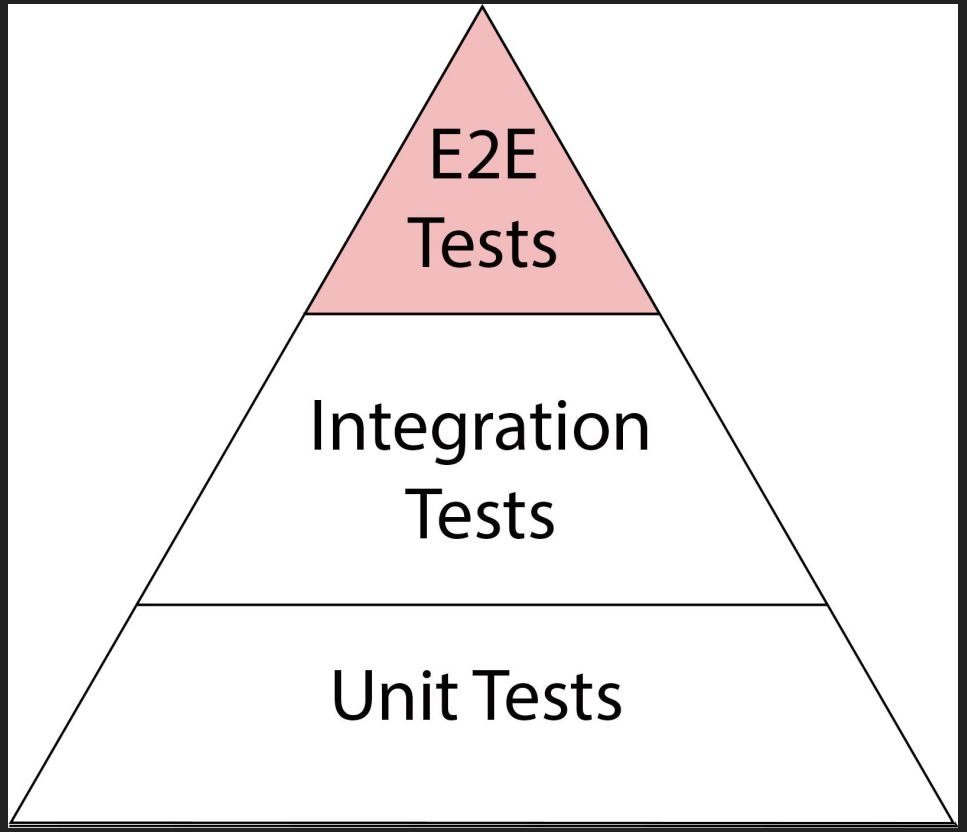
@DmitryVinnik

dvinnik.com

Let’s go in more details

How do we usually test?

Test Pyramid



↑ Fidelity

↓ Speed

↑ Cost

↓ Fidelity

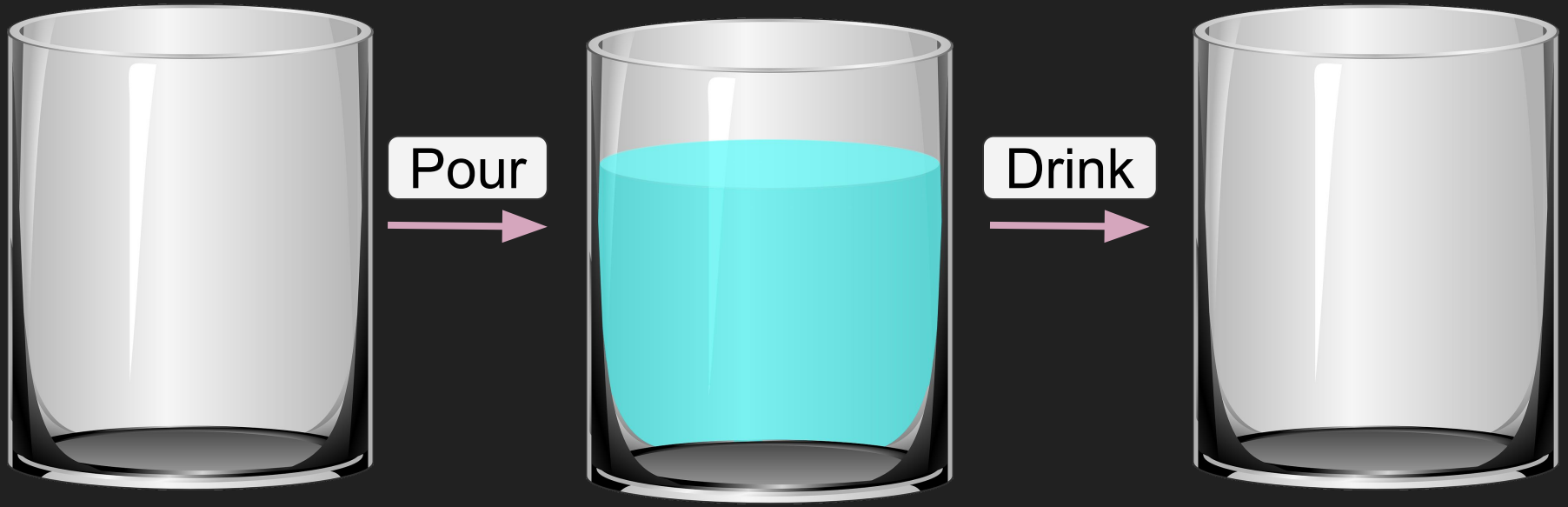
↑ Speed

↓ Cost

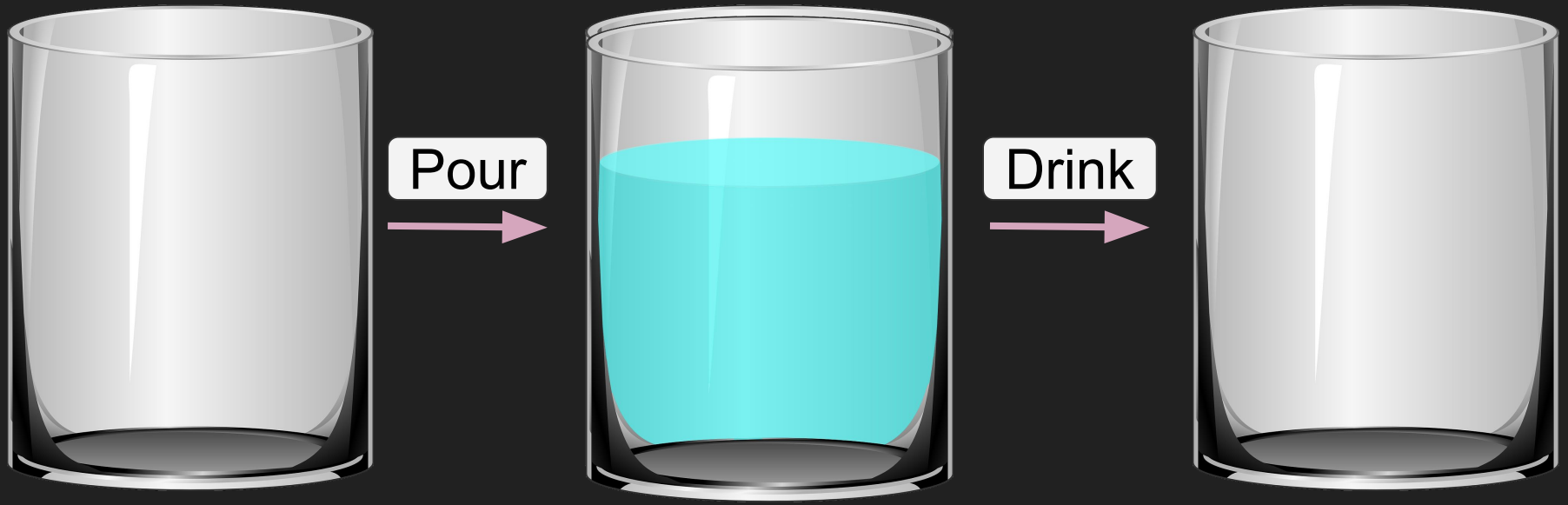
Let’s try get more practical

Is anyone thirsty?

Drinking Glass



Test Pyramid & Drinking Glass



Unit

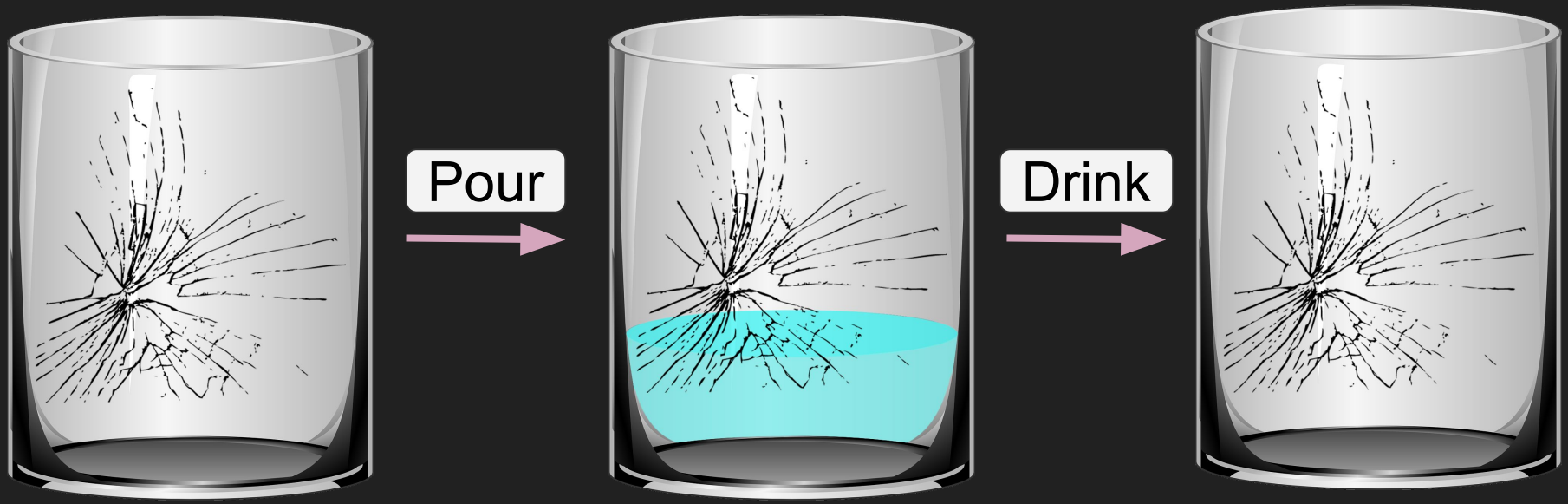
Integration

E2E

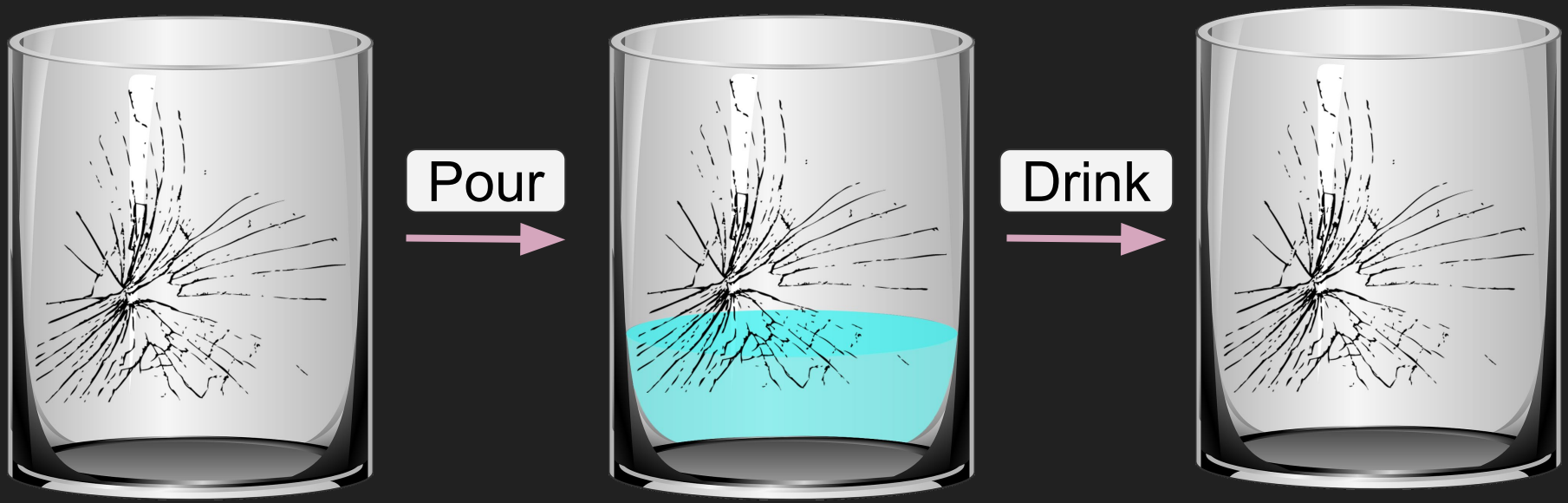
Are we done?

Not really...

Drinking Glass



Test Pyramid & Drinking Glass



Unit

Integration

E2E

@DmitryVinnik

dvinnik.com

What are we missing?

What are we missing?

Same
Functionally

Different
Visually

Visual Content

Visual Animation

Visual Testing

Page Layout

Responsive Design

Functional
Manual Testing
is **Difficult**

Visual
Manual Testing
is **Nearly**
Impossible

A Little Bit of Science

A Little Bit of Science

Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments

HECTOR YEE, SUMANTA PATTANAIK and DONALD P. GREENBERG

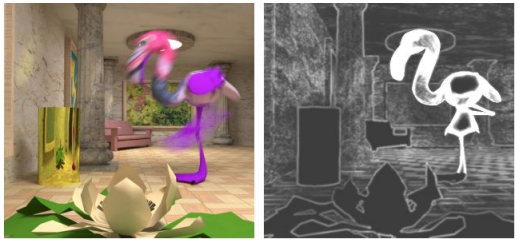
Program of Computer Graphics, Cornell University

We present a method to accelerate global illumination computation in pre-rendered animations by taking advantage of limitations of the human visual system. A spatiotemporal error tolerance map, constructed from psychophysical data based on velocity dependent contrast sensitivity, is used to accelerate rendering. The error map is augmented by a model of visual attention in order to account for the tracking behavior of the eye. Perceptual acceleration combined with good sampling protocols provide a global illumination solution feasible for use in animation. Results indicate an order of magnitude improvement in computational speed.

Keywords: Animation, Computer Vision, Human Visual Perception, Illumination, Monte Carlo Techniques

1 INTRODUCTION

Global illumination is the physically accurate calculation of lighting in an environment. It is computationally expensive for static environments and even more so for dynamic environments. Not only are many images required for an animation, but the calculation involved increases with the presence of moving objects. In static environments, global illumination algorithms can precompute a lighting solution and reuse it whenever the viewpoint changes, but in dynamic environments, any moving object or light potentially affects the illumination of every other object in a scene. To guarantee accuracy, the algorithm has to recompute the entire lighting solution for each frame. This paper describes a perceptually-based technique that can dramatically reduce this computational load. The technique may also be used in image based rendering, geometry level of detail selection, realistic image synthesis, video telephony and video compression.



Reference Image (a)

Spatiotemporal error tolerance map (Aleph Map) (b)

Figure 1: Global Illumination of a Dynamic Environment (*see color plate*). Global illumination correctly simulates effects such as color bleeding (the green of the leaves on to the petals), motion blur (the pink flamingo), caustics (the reflection of the light by the golden ash tray on the wall), soft shadows, anti-aliasing, and area light sources (a). This expensive operation benefits greatly from our perceptual technique, which can be applied to animation as well as motion-blurred still images such as shown above. The spatiotemporal error tolerance map (which we call the Aleph Map) is shown on the right (b). Bright areas on the map indicate areas where less effort should be spent in computing the lighting solution. The map takes a few seconds to compute but will save many hours of calculation.

Perceptually-based rendering operates by applying models of the human visual system to images in order to determine the stopping condition for rendering. In doing so, perceptually assisted renderers attempt to expend the least amount of work to obtain an image that is perceptually indistinguishable from a fully converged solution. The technique described in this paper assists rendering algorithms by producing a spatiotemporal error tolerance map (Aleph Map) that can be used as a guide to optimize rendering. Figure 1 shows a scene containing moving objects (a) and its Aleph Map (b). The brighter areas in the map show regions where sensitivity to errors is low, permitting shortcuts in computation in those areas.

1

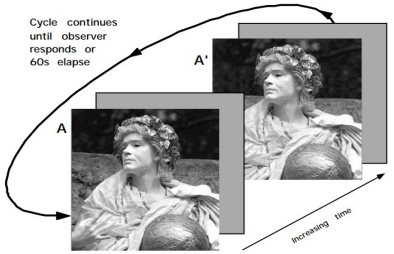


FIGURE 9.1. Example of flicker paradigm. Sequence alternates between original and modified image until observer responds. Display times (on-times) are typically 200 - 600 ms, while blank times (off-times) are 80-800 ms. In the stimulus here, original image A (statue with wall in background) and modified image A' (statue with wall lowered) appear in the order A, A', A, A',...with gray fields placed between successive images. For this example, over 40 alternations are required on average before observer detects the change.

Indeed, change blindness can be induced by a large number of techniques, such as making the change during:

- an eye movement (Bridgeman, Hendry, & Stark, 1975; Grimes, 1996),
- an eye blink (O'Regan, Deubel, Clark, & Rensink, 2000)
- a movie cut (Levin & Simons, 1997)
- occlusion of the changing item (Simons & Levin, 1998)
- small transient "splats" elsewhere in the image (Rensink et al., 2000).

(See Rensink, 2000c or Simons & Levin, 1997 for a more complete review of various change-blindness studies.) Given that change blindness can be induced in many ways and that it has a strong phenomenological effect, it follows that change blindness is not an aberrant phenomenon occurring only under a special set of conditions. Rather, it appears to touch on something important, something central to the way that the world is perceived.

9.1.2 *Coherence theory*

Why is it that change blindness can be so easily induced? And if it is so easy to induce, why are we nevertheless so good at seeing changes in everyday life?

All the techniques used to induce change blindness share a comment element: the transients associated with the change are swamped (or otherwise neutralized) so that there is little information about its location in the display. This leads to the suggestion that *focused attention is necessary to see change* (Rensink et al., 1997). The world is by and large a quiet

Conclusion?

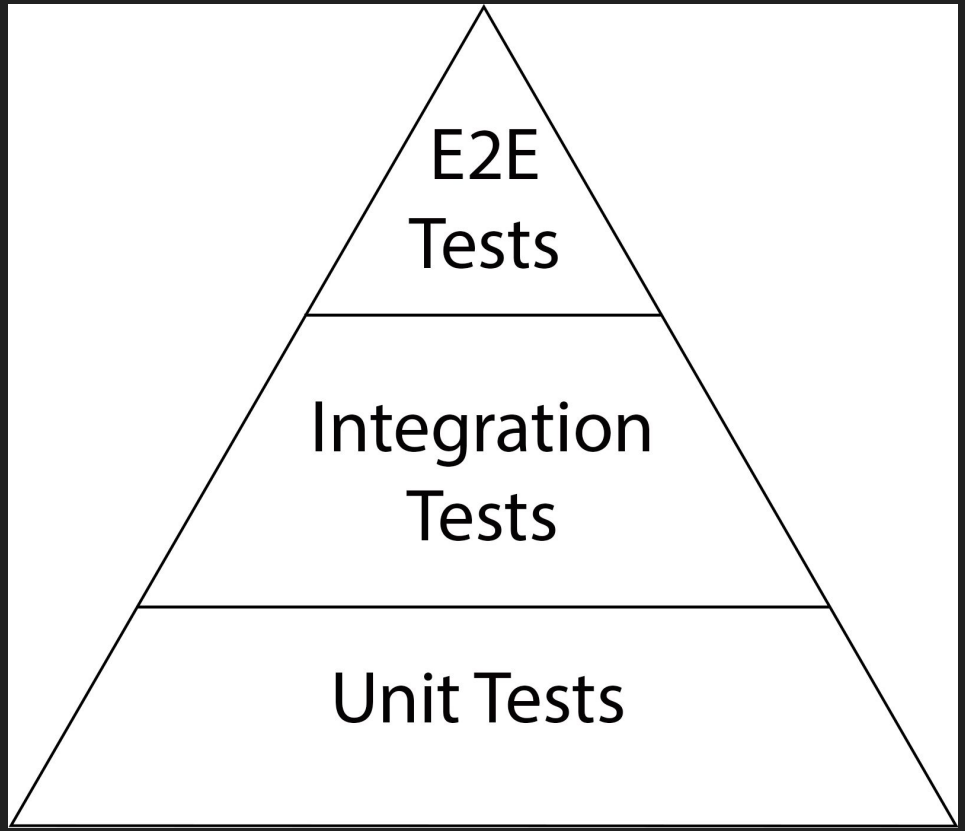
Visual Testing requires **Automation**

But!

Where does it fit on Test Pyramid?

On Every Level!

Test Pyramid



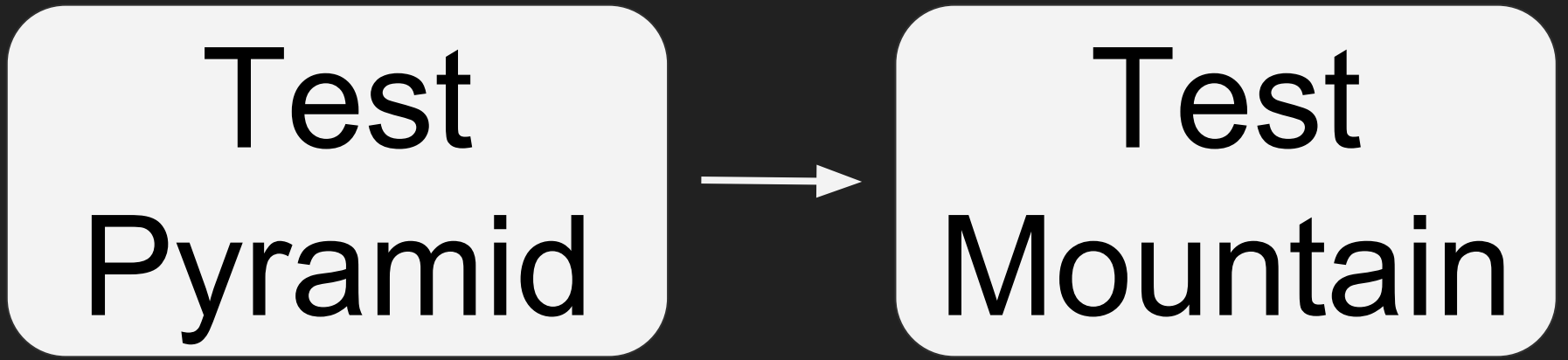
↑ Fidelity
↓ Speed
↑ Cost

↓ Fidelity
↑ Speed
↓ Cost

Remember the Title?

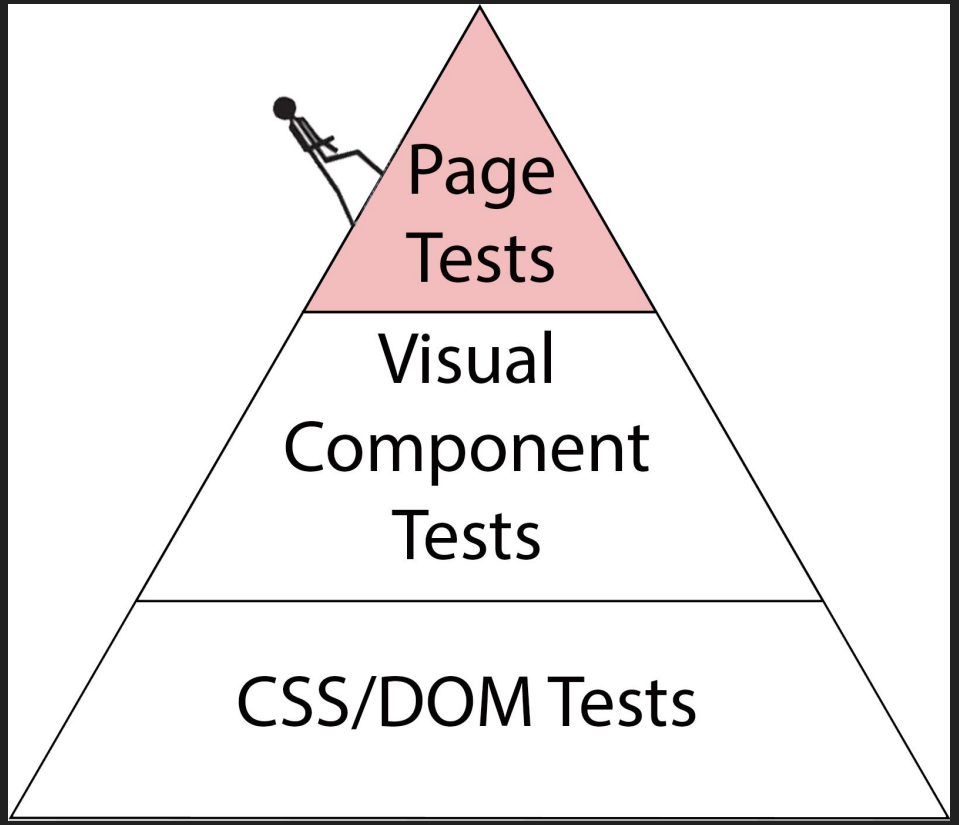
Uphill Battle Of Mobile Visual Regression

Uphill Battle


Test
Pyramid
Test
Mountain

Visual Test Mountain

Visual Test Mountain

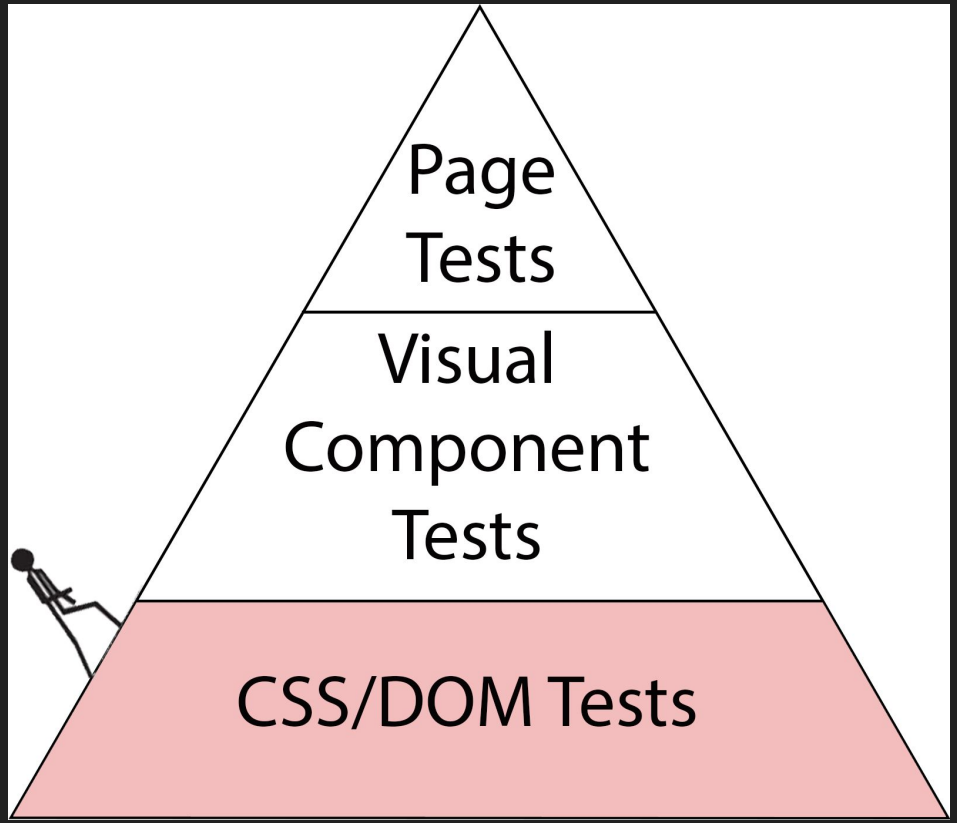


↑ Fidelity
↓ Speed
↑ Cost

↓ Fidelity
↑ Speed
↓ Cost

Let’s Start Our Climb!

CSS/DOM Tests



Responsive App

Hybrid App

Testing Demos

Android

iOS

Demo Time:
Snapshot Testing

Is it enough?

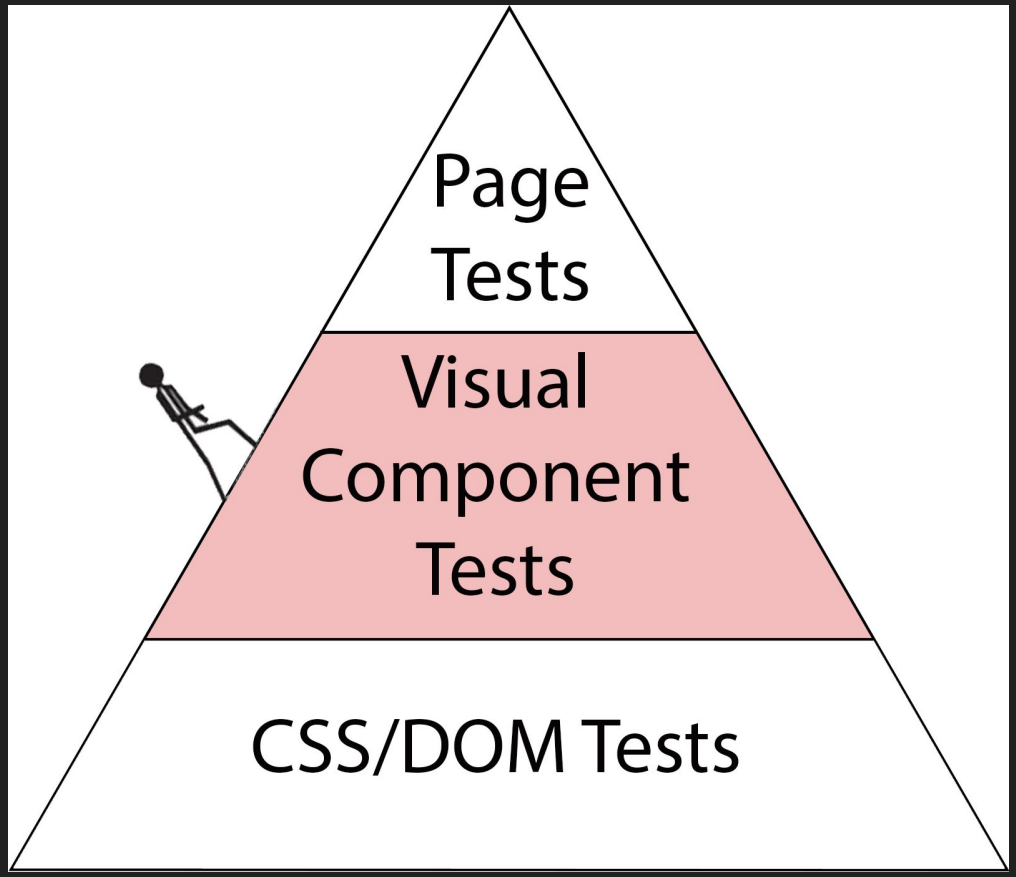
No!

Let me show you why

Quick Glance: **Responsive Design in Action**

Solution?

Visual Component Testing



Let’s Step Back First

Reusable

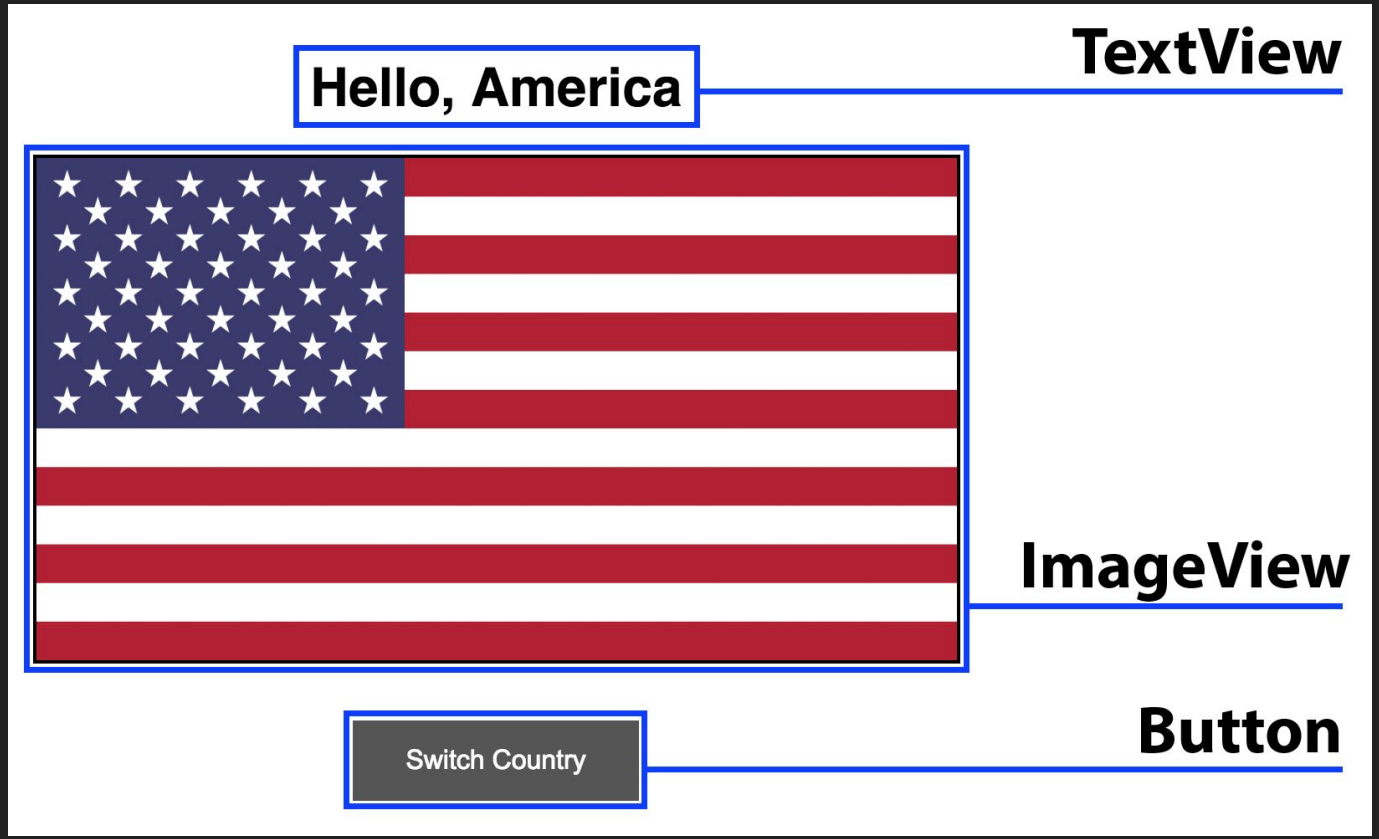
Encapsulated

Component Driven Development

Composable

Semantic

CDD in Practice



Library

Testing Ground

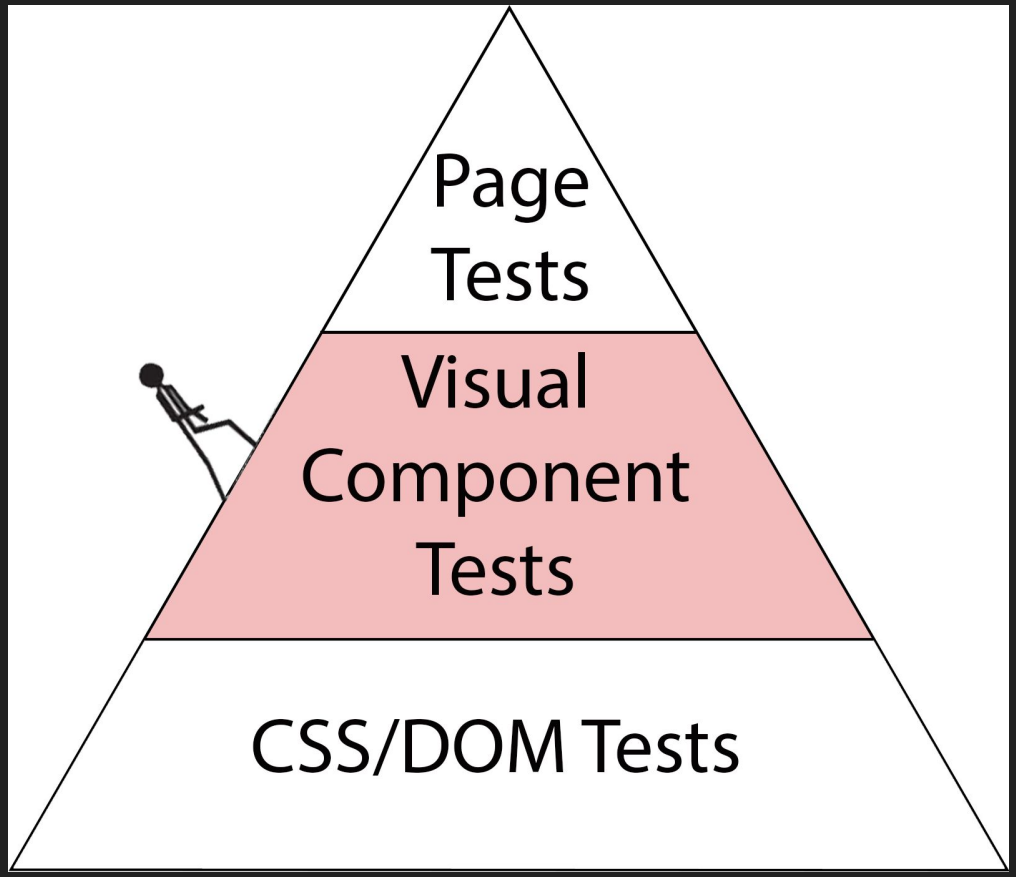
Storybook

Styleguide

IDE

Quick Glance: **Storybook**

Visual Component Testing



Responsive App

Hybrid App

Testing Demos

Android

iOS

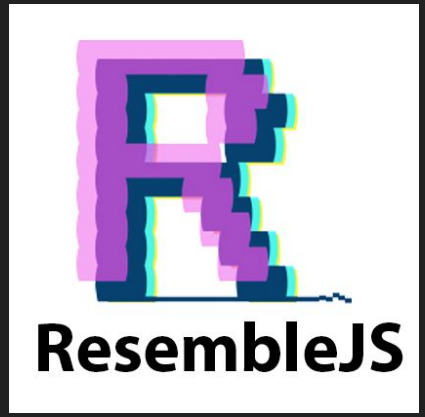
View Loader

Action Driver

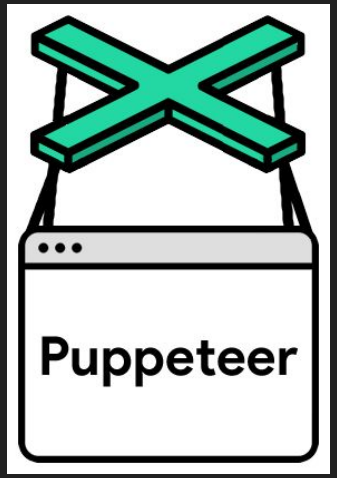
Screenshot Testing

Comparison Driver

PhantomCSS

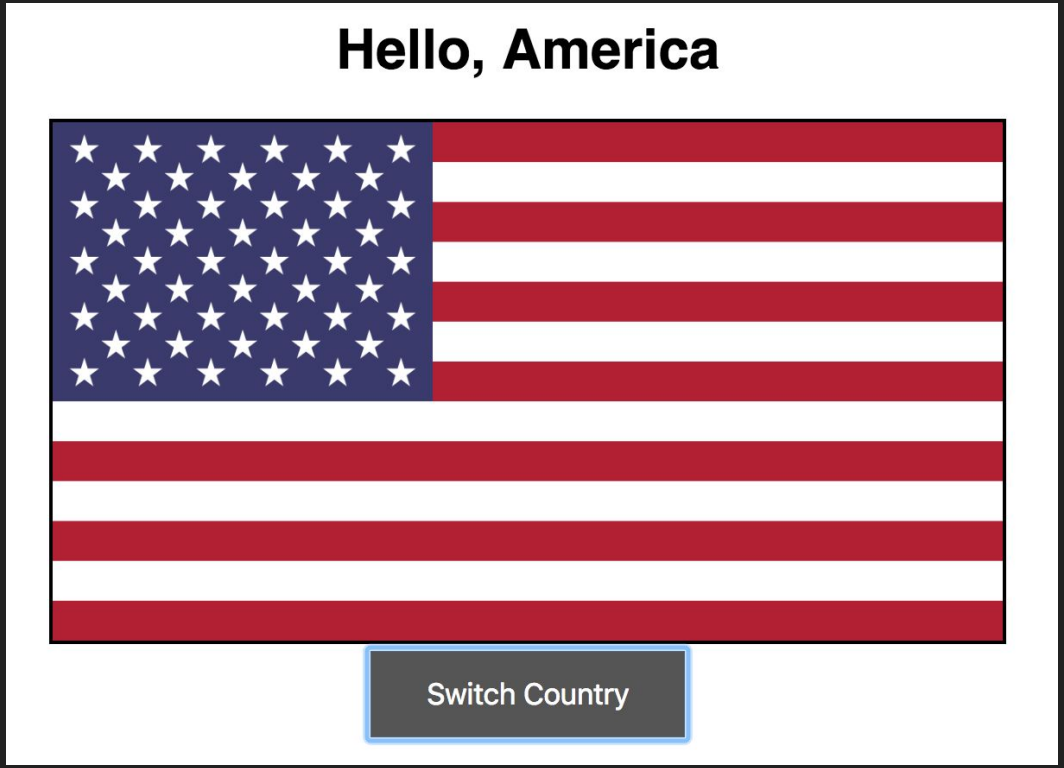


Demo Time:
PhantomCSS



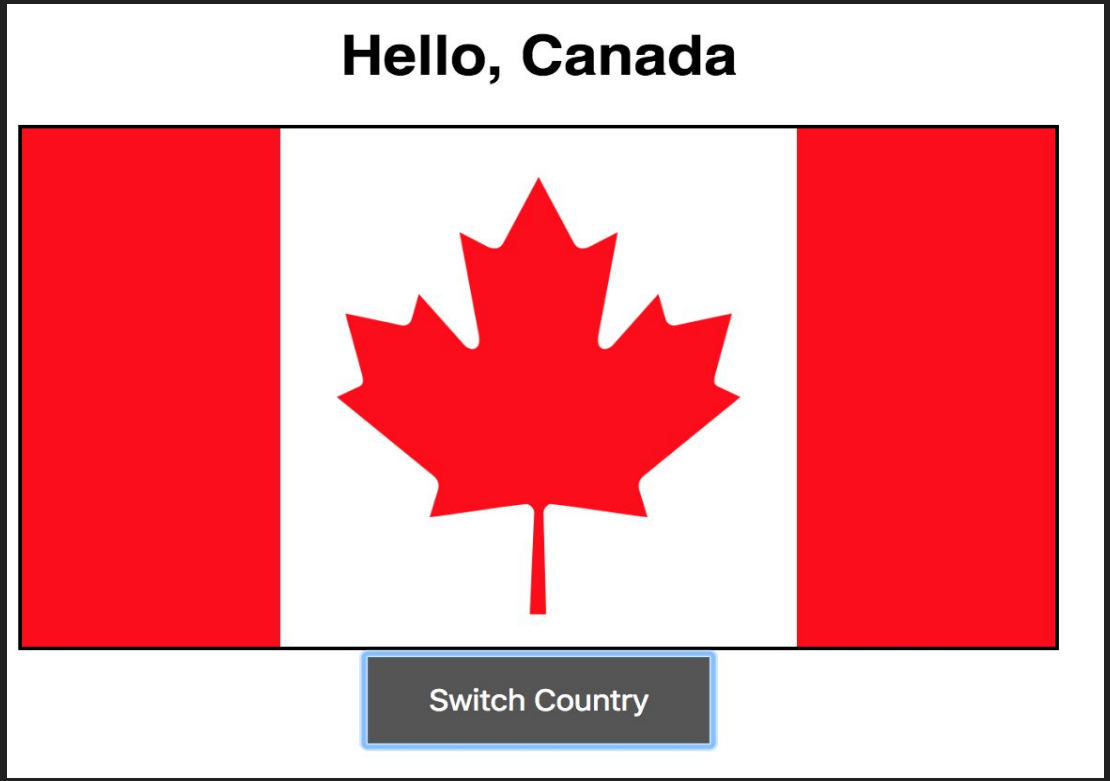
BackstopJS

BackstopJS



BackstopJS

BackstopJS



BackstopJS

```
{
  "viewports": [
    {
      "label": "tablet",
      "width": 1024,
      "height": 768
    },
    {
      "label": "phone",
      "width": 374,
      "height": 667
    }
  ],
}
```

BackstopJS

```
{
"scenarios": [
    {
        "label": "American Flag",
        "url": "http://localhost:3000/",
        "delay": 0,
        "selectors": [".countryFlag"],
        "misMatchThreshold" : 0.1
    }
]
}
```

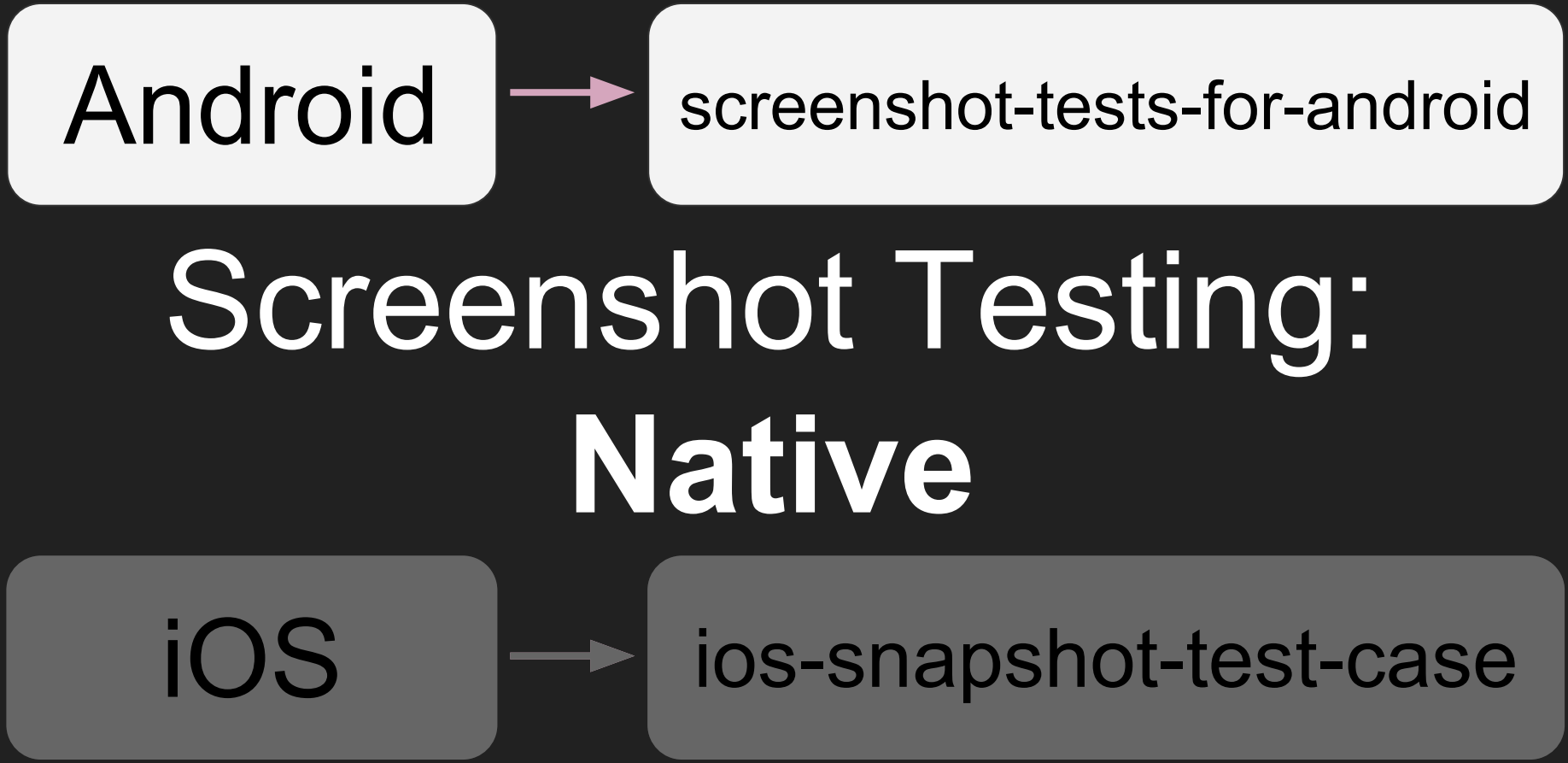
Demo Time:
BackstopJs

View Loader

Action Driver

Screenshot Testing: **Native**

Comparison Driver


Android
screenshot-tests-for-android
Screenshot Testing:
Native
iOS
ios-snapshot-test-case

Demo Time:
Screenshot Testing Android

Percy.io

Flawlessapp.io

Enterprise

Chromatic

Screener.io

Screenshot Names

Component Testing First

Best Practices

Git Media

Standard Test Pyramid

DOM/CSS
Not Enough

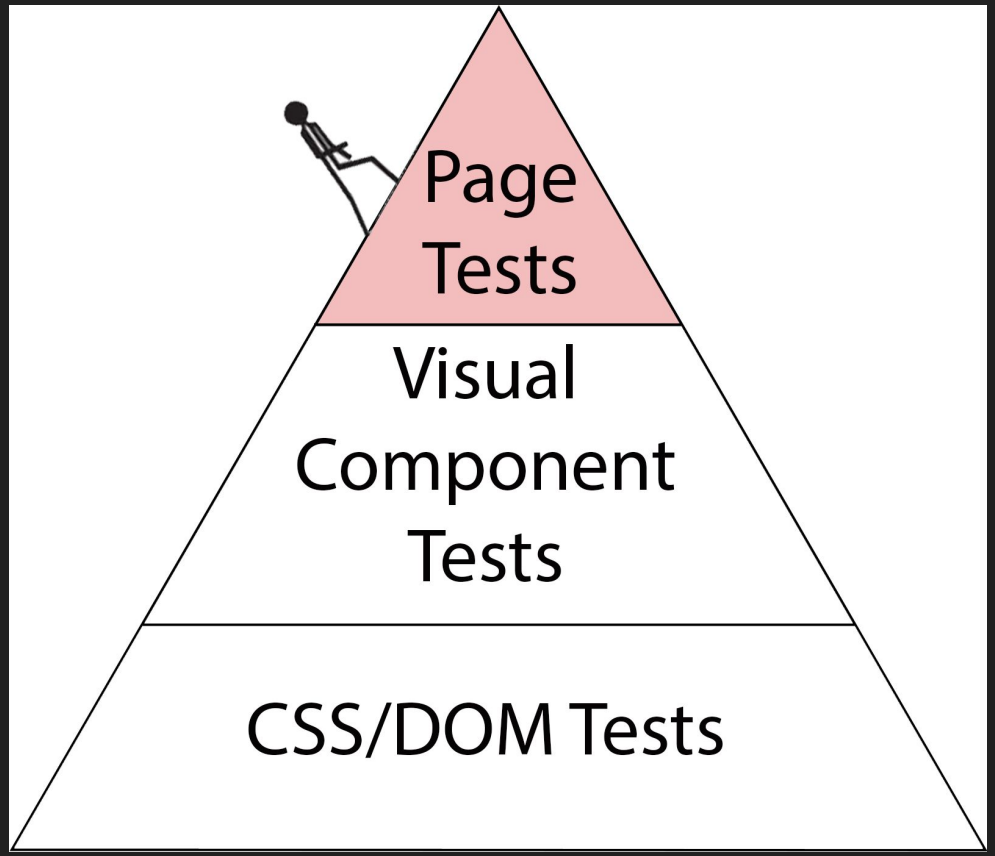
Dev = Test

Summary

Choose
Solution

Build
Solution

Page Visual Tests



Window Size

DOM/CSS

Responsive Testing

Transition States

Screenshots

Demo Time:
Responsive Testing
with **Selenium**

Demo Time:
Native Visual Testing
with
Appium & Applitools

Applitools

BrowserStack

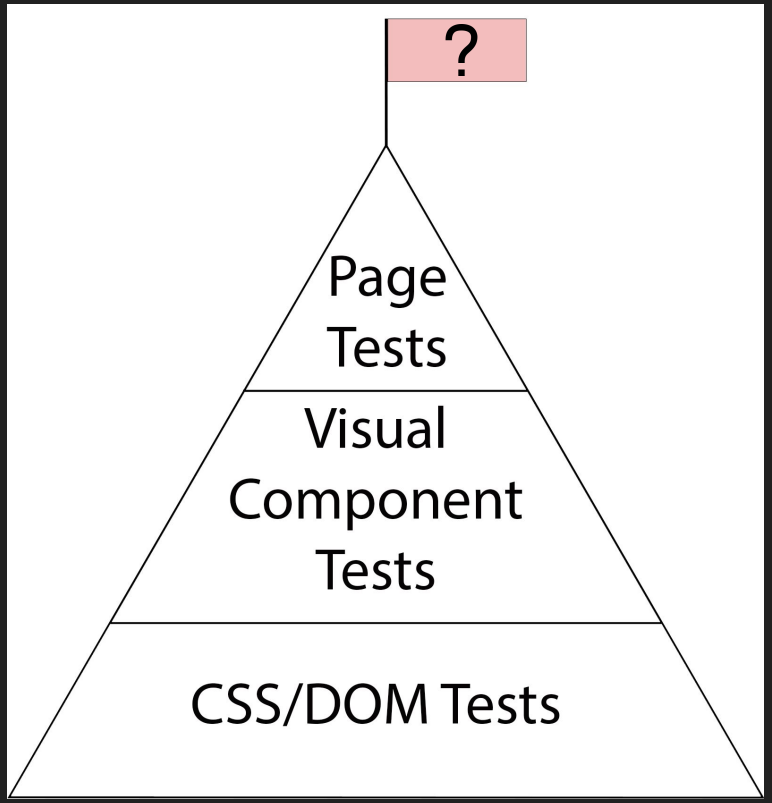
Enterprise

Sauce

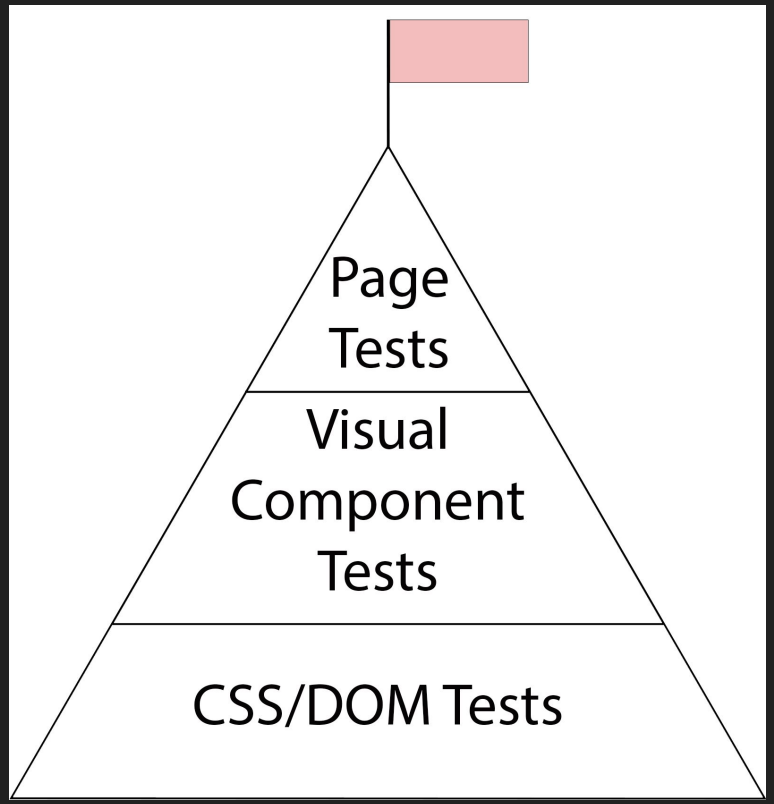
SauceLabs

Object

Summit



Summit



+ You

Start the Climb

Visual Unit Tests

Call For Action

Visual Component Tests

Visual Page Tests

Q/A

About Speaker

Twitter: @DmitryVinnik
Blog: dvinnik.com
LinkedIn: in/dmitry-vinnik/
Email: dmitry@dvinnik.com