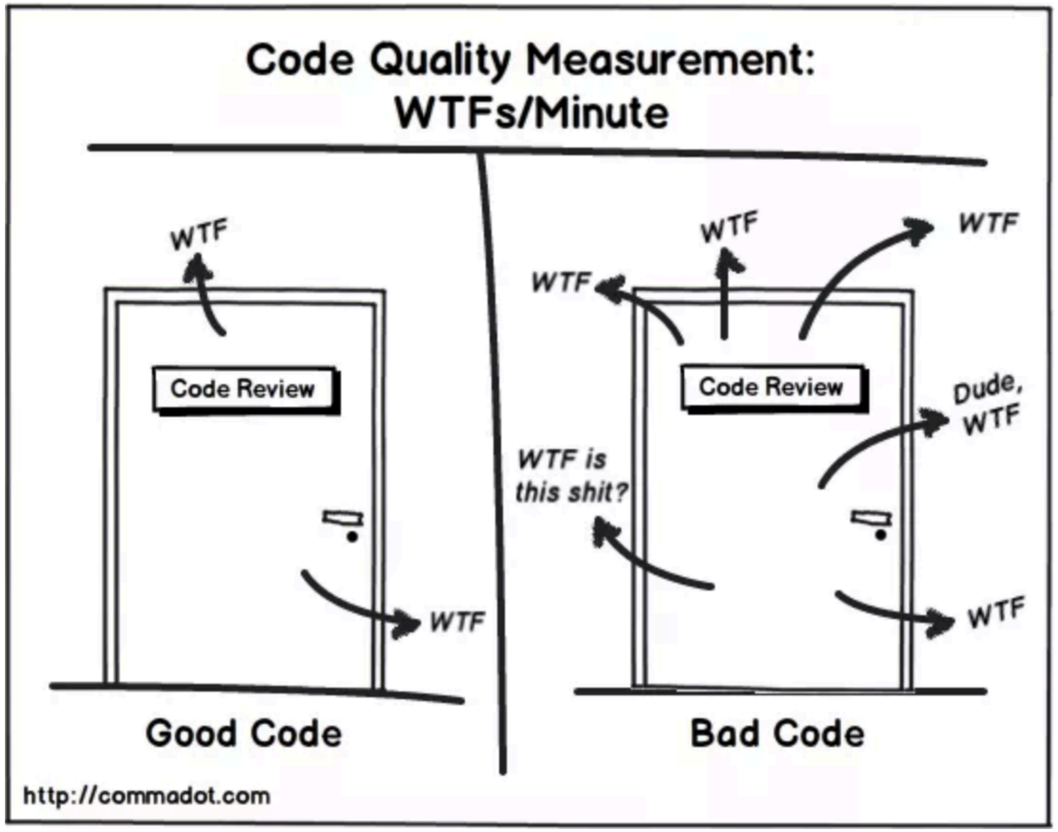
Web 前端测试指南

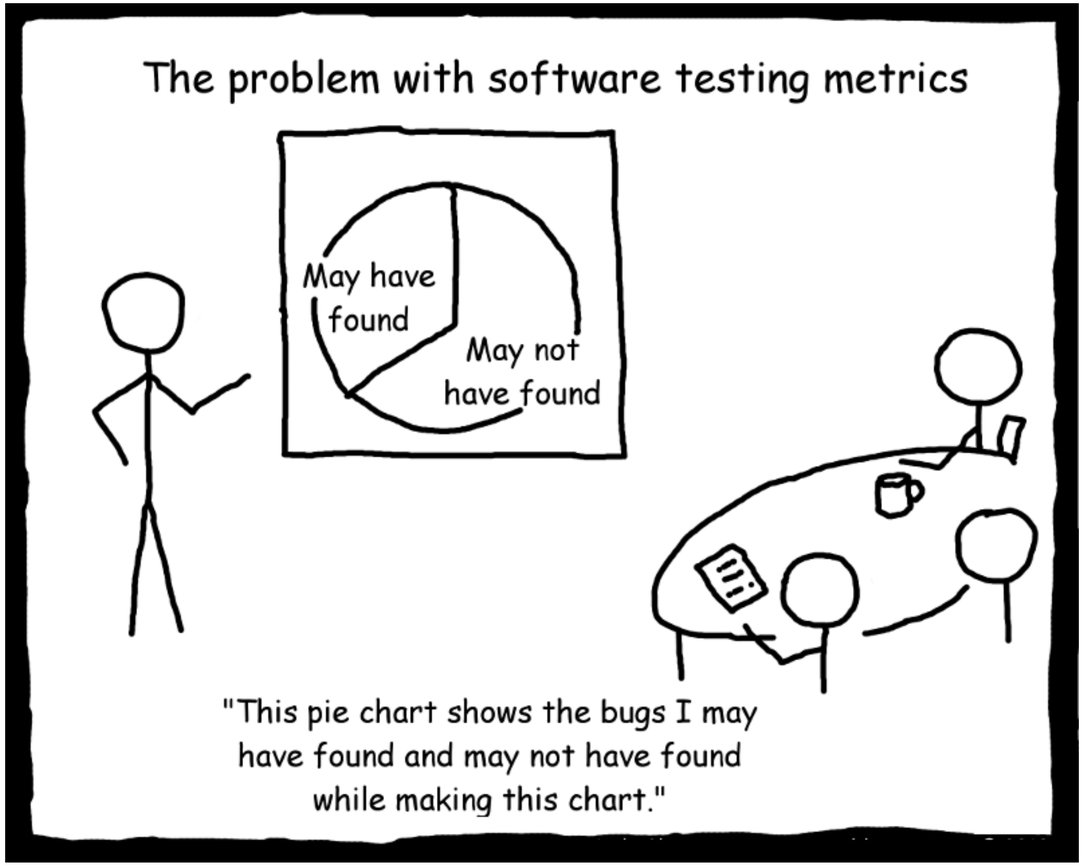
The Guide of Web Test

Jack Pu

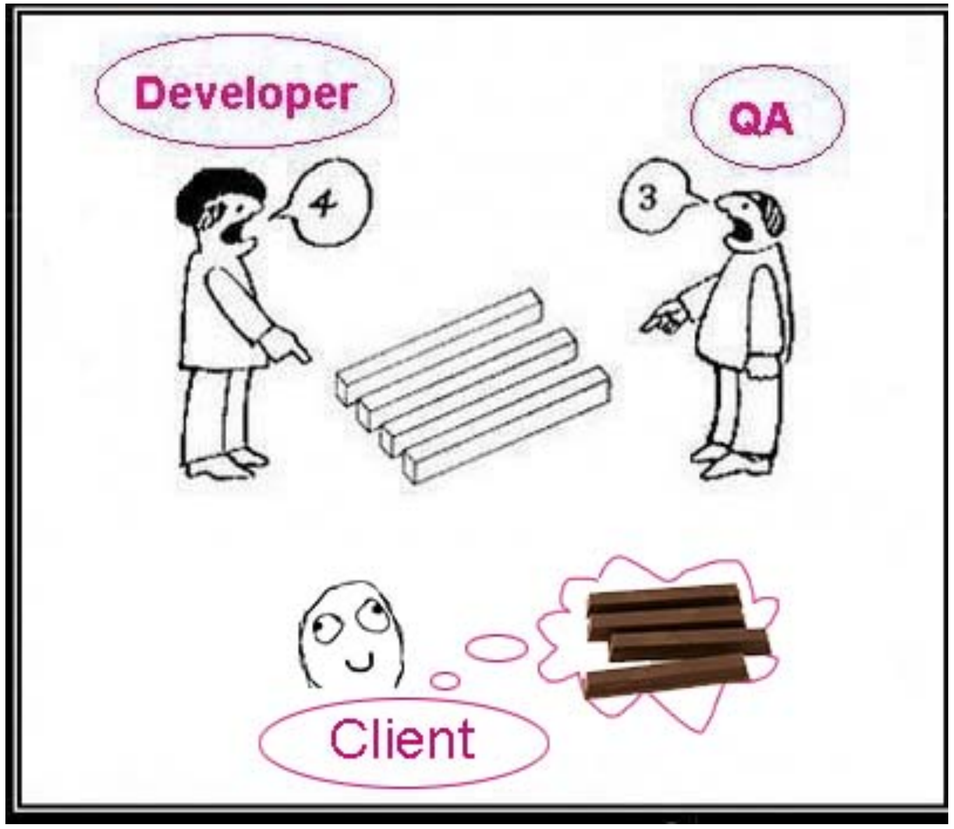
Code Quality



Not Found Bug



Dev VS QA VS Client



Web 前端测试

npm run lint

npm run test:unit

npm run test:e2e

Web 前端测试

```
├── test
|   ├── e2e
|   |   ├── specs
|   |   |    └── index.test.js
|   |   └── nightwatch.night.js
|   ├── fixtures // 存放 mock 的数据或者测试资源
|   |    └── data/html/images
|   ├── unit
|   |   └── specs
|   |       ├── unit.test.js
|   |       └── index.test.js
|   └── karma.conf.js
└ ...
```

代码规范

代码规范

行业流行规则 + 团队内部约束

Eslint / Standrad / Prettier / JSHint

Code Review

了解功能实现，从自己的角度去理解代码

Arcanist

单元测试

来自开发

开发人员编写的、用于检测在特定条件下目标代码正确性的代码

优化设计

从调用者的角度观察、思考，会让使用者把程序设计成易于调用和可测试，并且解除软件中的耦合，剔除无效代码

易于回归

代码在做迭代更新时候，确保功能快速回归，提升开发测试效率

前端单元测试框架

Jest

配置少，API简单
基于 Jasmine
来自 Facebook
比较新颖

React Vue

Mocha

灵活
社区成熟
需要较多配置

JS App

Jasmine

开箱即用
非常成熟
支持断言和仿真

JS React

AVA

异步，性能好
简约，清晰快照测试断言需要三方支持

Node

写第一个单元测试

```
var calculator = {
    sum: function(x, y) {
        return 2; // <-- note this is hardcoded
    }
}

describe('calculator', function () {

    it('1 + 1 should equal 2', function () {
        expect(calculator.sum(1, 1)).toBe(2);
    });

});
```

测试的内容

测试 Case

写第一个单元测试

```
describe('cookie-util.js test', () => {
  beforeEach(() => {
    // running before each test
  });
  it('# get()', () => {
    // case1
    // case 2
  });
  it('# set()', () => {
    // case 1
    // case 2
    // case 3
  });
  it('# remove()', () => {
    // case 1
    // case 2
  });
});
```

测试 case 可以有很多，
尽可能的详细的描述出需要被测试的特点

测试 异步 或者 超时

```
describe('timeout test', () => {
  // set max timeout time
  jasmine.DEFAULT_TIMEOUT_INTERVAL = 6000;

  it('# emit()', (done) => {
    setTimeout(() => {
      // your code
      expect(...).toHaveBeenCalled();
      done();
    }, 4000);
  });
});
```

Done 标识这次测试结束

```
beforeEach(function() {
  return new Promise(function(resolve, reject) {

    // your async code requet, file handle etc.

    resolve();
  });
});
```

测试框架支持对 async 函数
以及 Promise 的调用

测试函数执行被调用

```
describe('function call test', () => {

  let foo;
  let bar;

  beforeEach(() => {
    foo = {
      setBar(value) {
        bar = value;
      },
    };
    spyOn(foo, 'setBar');
  });

  it('# func exec()', () => {
    ee.on('add', foo.setBar);
    ee.emit('add', true);
    expect(foo.setBar).toHaveBeenCalled();
  });
});
```

spyOn 支持对于对象方法执行进行监听，
从而达到某个函数是否执行的判断

https://jasmine.github.io/2.0/introduction.html#section-Spies

使用 beforeEach 进行前置设置

```
describe('cookie test', () => {
  beforeEach(() => {
    // clear cookie
    document.cookie.split(';').forEach((c) => {
      document.cookie = c.trim().split('=')[0] + '='
        + 'expires=Thu, 01 Jan 1970 00:00:00 UTC;';
    });
  });
  it('# get()', () => {
    // test code
  });
  it('# set()', () => {
    // test code
  });
  it('# remove()', () => {
   // test code
  });
});
```

BeforeEach 会在每个 测试 case 运行前都执行一次

https://jasmine.github.io/2.0/introduction.html#section-Setup_and_Teardown

使用 enzyme 测试 React 应用/组件

安装 enzyme

```
$ npm install babel-preset-airbnb react-addons-test-utils enzyme  --save-dev
```

测试组件

```
import React from 'react';
import {
  shallow,
} from 'enzyme';
// test componet
import Footer from '../src/components/footer';

describe('Component Tests', function () {
  it('contains copyright words',function() {
    expect(shallow(<Footer />).contains(<p className="other-info">All Rights Reserved By Jack Pu</p>)).toEqual(true);
  })
  it('contains four link',function() {
    expect(shallow(<Footer />).find('a').length).toBe(4);
  })
});
```

https://jasmine.github.io/2.0/introduction.html#section-Setup_and_Teardown

Enzyme

Enzyme 是Airbnb推出一款用于测试React编写的组件的测试工具。
通过它你可以轻松的完成断言，DOM操作以及遍历 React Components 输出。

Enzyme 支持多种测试类库，比如Chai.js ,Mocha,或者Jasmine.
你也可以用它来测试你的React-Native / React-360 程序。

Enzyme 支持构建工具集成，对 webpack / babel 友好。

熟悉 enzyme 基本 API

Shallow

执行 constructor / render ， 但是不渲染子组件

Mount

执行 full dom render ,可以测试到 componentDidMount 生命周期

Render

执行 render 函数，并且渲染所有子组件

Find

匹配 选择器里面 render tree 节点

setProps

手动触发 prop 改变，触发组件生命周期 componentWillReceiveProps

https://airbnb.io/enzyme/docs/api/

使用 enzyme 模拟 用户交互

simulate 支持对现有基本的基本交互事件 click, mouseover, focus 等进行模拟支持

```
import React from 'react';
import Form from '../../../src/components/form';
import { mount } from '../../enzyme';


describe('Dorm component test', () => {

  it('# onChange ()', () => {
    const wrapper = mount(
      <Form />
    );
    wrapper.find('input').simulate('focus');
    wrapper.find('input').simulate('change', { target: { value: '13' } });
    wrapper.find('input').simulate('change', { target: { value: '1378' } });
    expect(wrapper.find('.js-error').length).toBe(1);
  });

})
```

https://airbnb.io/enzyme/docs/api/ReactWrapper/simulate.html

使用 redux-mock-store 模拟 store

```
const mockState = {
  guide: {
    data: [
      {
        name: 'KungFu Hustle',
        rank: '8.3',
      },
      {
        name: 'CJ-7',
        rank: '6.4',
      },
    ],
    tags: ['movie', 'comedy'],
  },
};

export default mockState;
```

```
import React from 'react';
import configureStore from 'redux-mock-store'
import { Provider } from 'react-redux';
import Guide from '../../../src/components/guide';
import { mount } from '../../enzyme';
// 用于模拟状态得数据
import mockState from '../fixtures/state';


const mockStore = configureStore();
let wrapper;
let store;
beforeEach(() => {
  // 创建关联 store
  store = mockStore(mockState);
  // 渲染测试的组件将 store 传入
  wrapper = mount(<Provider store={store}><Guide /></Provider>)
});
describe('Links component test', () => {
  it('render movies and tags', () => {
    expect(wrapper.find('.movie-item').length).toBe(2);
    expect(wrapper.find('.tag-item').length).toBe(2);
  });
});
```

https://github.com/dmitry-zaets/redux-mock-store

集承测试

保证 UI

确保浏览器渲染效果正常，实现快照对比

克服兼容

在集成测试端，快速进行多种浏览器的测试，发现兼容问题

易于回归

易于流程控制，用户交互模拟，方便测试用例编写

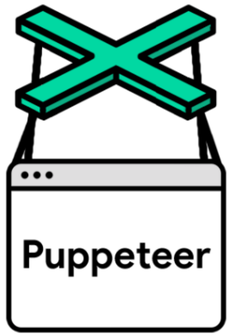
E2E测试框架

Cypress

不使用 Selenium
运行速度要快
比较新颖, 问题较多

Nightwatch

上手较快
社区成熟
API丰富



Puppeteer

基于 headless Chrome
支持测试 Chrome 插件
可控性大
只支持 Chrome

使用 nightwatch 进行 E2E 测试

安装 nightwatch

```
$ npm i selenium-server nightwatch chromedriver --save-dev
```

编写 nightwatch.conf.js

```
src_folders: [
  'specs',
],
output_folder: 'reports', // Where to output the test reports
```

```
  chrome: {
    desiredCapabilities: {
      browserName: 'chrome',
      javascriptEnabled: true,
      acceptSslCerts: true,
      nativeEvents: true,
    }
  },
  // firefox: {

  // },
  // safari: {

  // }
```

https://github.com/dwyl/learn-nightwatch

编写 E2E 测试用例

```
// http://nightwatchjs.org/guide/#writing-tests

module.exports = {
  beforeEach: (browser) => {
    browser
      .url('http://localhost:8080/test/fixtures/test.html')
      .waitForElementVisible('body')
      .waitForElementVisible('#app');
  },
  'basic title': (browser) => {
    browser
      .assert.visible('#app > a', 'Check if element created')
      .assert.title('alita');
  },
  after: browser => browser.end(),
};
```

https://jasmine.github.io/2.0/introduction.html#section-Setup_and_Teardown

使用 nightwatch 进行 E2E 测试

```
module.exports = {
  beforeEach: (browser) => {
    browser
      .url('https://veervr.tv/')
      .waitForElementVisible('body')
      .waitForElementVisible('#app');
  },
  'basic title': (browser) => {
    browser
      .assert.visible('#app > a', 'Check if element created')
      .assert.title('VR/360视频，全景照片以及互动体验全球VR内容分享社区 | VeeR VR');
  },
  'oauth test': (browser) => {
    browser
      .assert.visible('.header-signup', 'signup button')
      .assert.visible('.header-login', 'login button')
      .click('.header-login')
      .pause(1000)
      .assert.visible('user-login-modal', 'show login modal')
      .browser.setValue('#loginIdentifier', 'jackpu')
      .browser.setValue('#loginPassword', '111111')
      .click('.user-login-modal .submit-btn')
      .pause(2000);
  },
  after: browser => browser.end(),
};
```

确保页面 ready

模拟用户行为

https://github.com/dwyl/learn-nightwatch

性能测试

页面性能

关注页面加载运行的指标，比如 loadtime / interactive time / memory

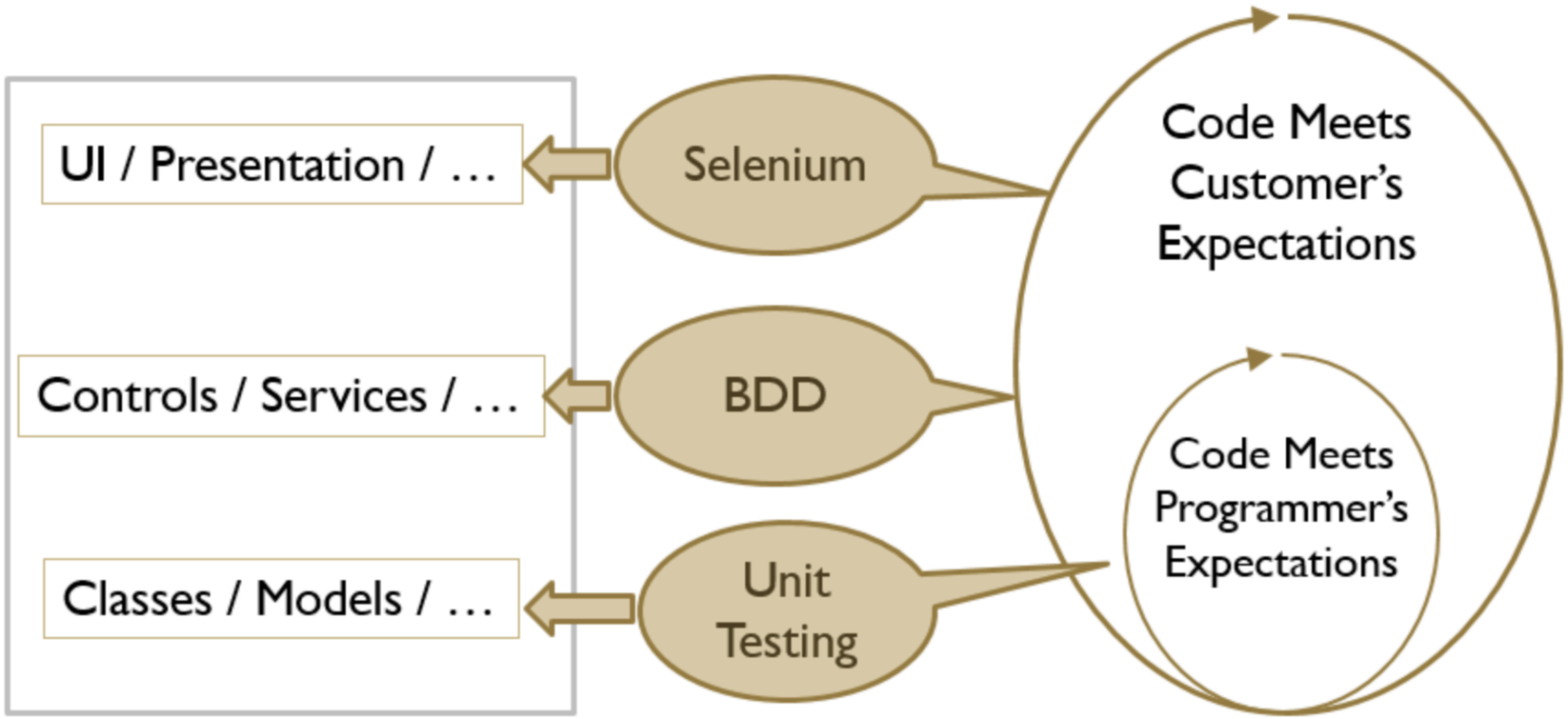
Pagespeed(Audits) / WebPageTest

压力测试

确保应用在高并发时候的承载能力 （**2/8 法则**），关注 实例响应延迟，CPU 运算，内存

阿里云 PTS / Testable / JMeter / http_load

小结


UI / Presentation / …
Controls / Services / …
Classes / Models / …
Selenium
BDD
Unit Testing
Code Meets Customer's Expectations
Code Meets Programmer's Expectations

THANK YOU

The Guide of Web Test

Jack Pu