

測試是什麼!?

PIXNET - Yvonne Yu

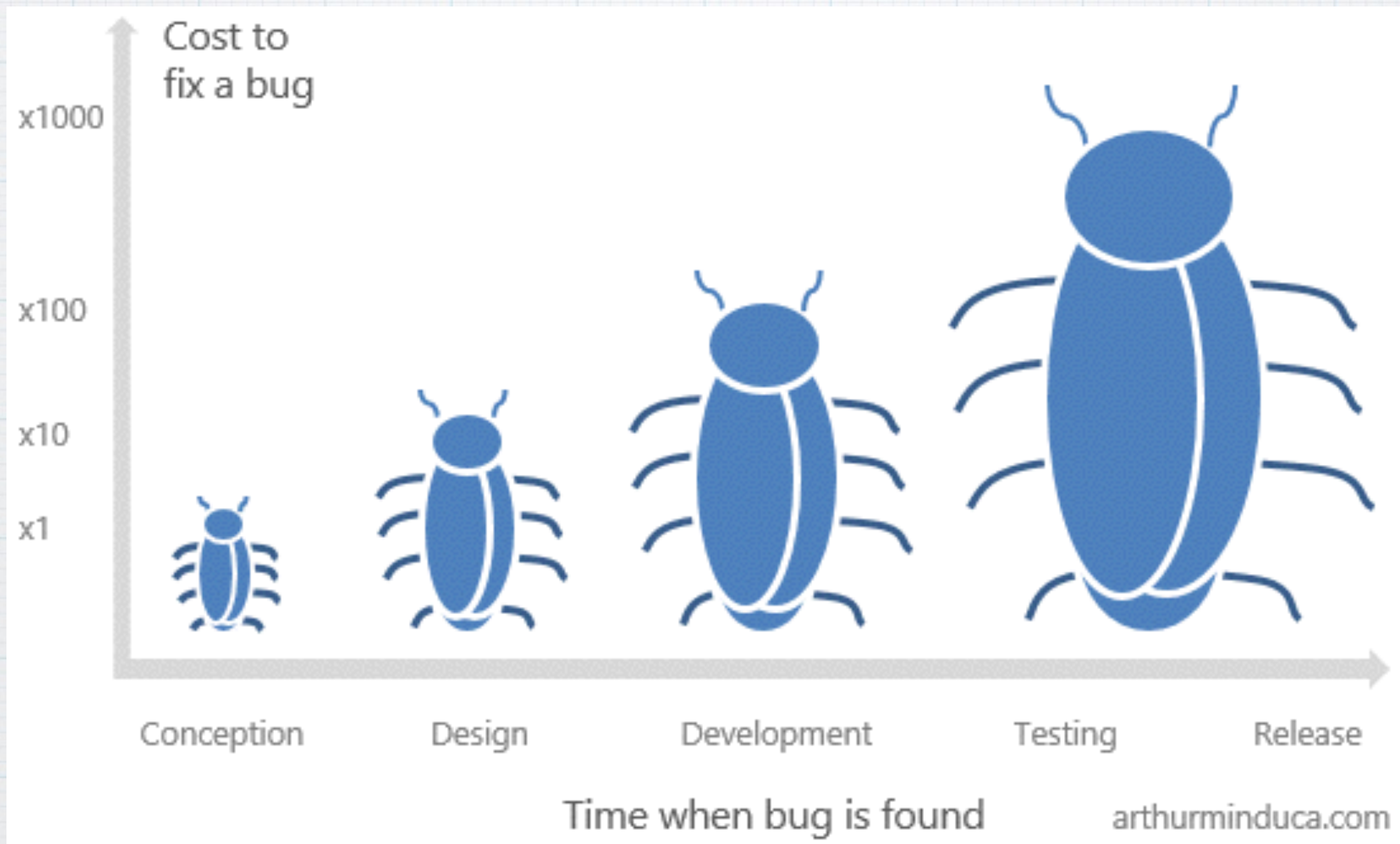
為什麼要寫測試？

是人，都會犯錯



QA

越晚發現 bug 解決時間就越久



測試使你思考



比較安心...?



有沒有改壞程式，
跑測試就知道了啊



WEB

測試是？

原理
(philosophy)

TDD, Continuous Integration

方法
(stages)

Unit Testing, End-to-End, integration Testing

結構
(constructs)

Test double, Test fixture, Test Harness,
Web Driver, Headless Browser

架構
(frameworks)

Mocha, Jasmine, Chai, Sinon

工具
(tools)

PhantomJS, Karma, Protractor, Selenium,
Jenkins, internJS, istanbul

Unit Testing - 單元測試

- * 針對程式的最小測試單位
- * 測試對象為單一函式
- * 獨立，不應跟其他測試程式碼相依
(不應測試後端程式碼)



Jasmine

```
function addTwo (num) {  
  return num + 2;  
};  
  
describe("測試 addTwo method", function() {  
  it("1 加 2 要等於 3", function() {  
    expect(addTwo(1)).toBe(3);  
  });  
  
  it("1 加 2 要不等於 4", function() {  
    expect(addTwo(1)).not.toBe(4);  
  });  
});
```


開始思考 Unit Testing!

Story：使用者成功登入後，會導到後台

➡ Features：使用者登入後台的表格

➡ Units：測試登入表格

```
describe('使用者登入表格', function() {  
  // 一定要測  
  it('正確的 email 格式跟密碼可以登入', function() {});  
  it('不正確的 email 格式不可以登入', function() {});  
  it('不正確的密碼不可以登入', function() {});  
  it('登入成功網址會變到後台網址', function() {});  
  
  // nice to have  
  it('密碼錯誤後，跳出提示視窗', function() {});  
  it('按下 enter 可送出登入表格', function() {});  
});
```


End-to-End Testing

- * 模擬使用者跟軟體的互動行為
- * 測試一整個 User Story

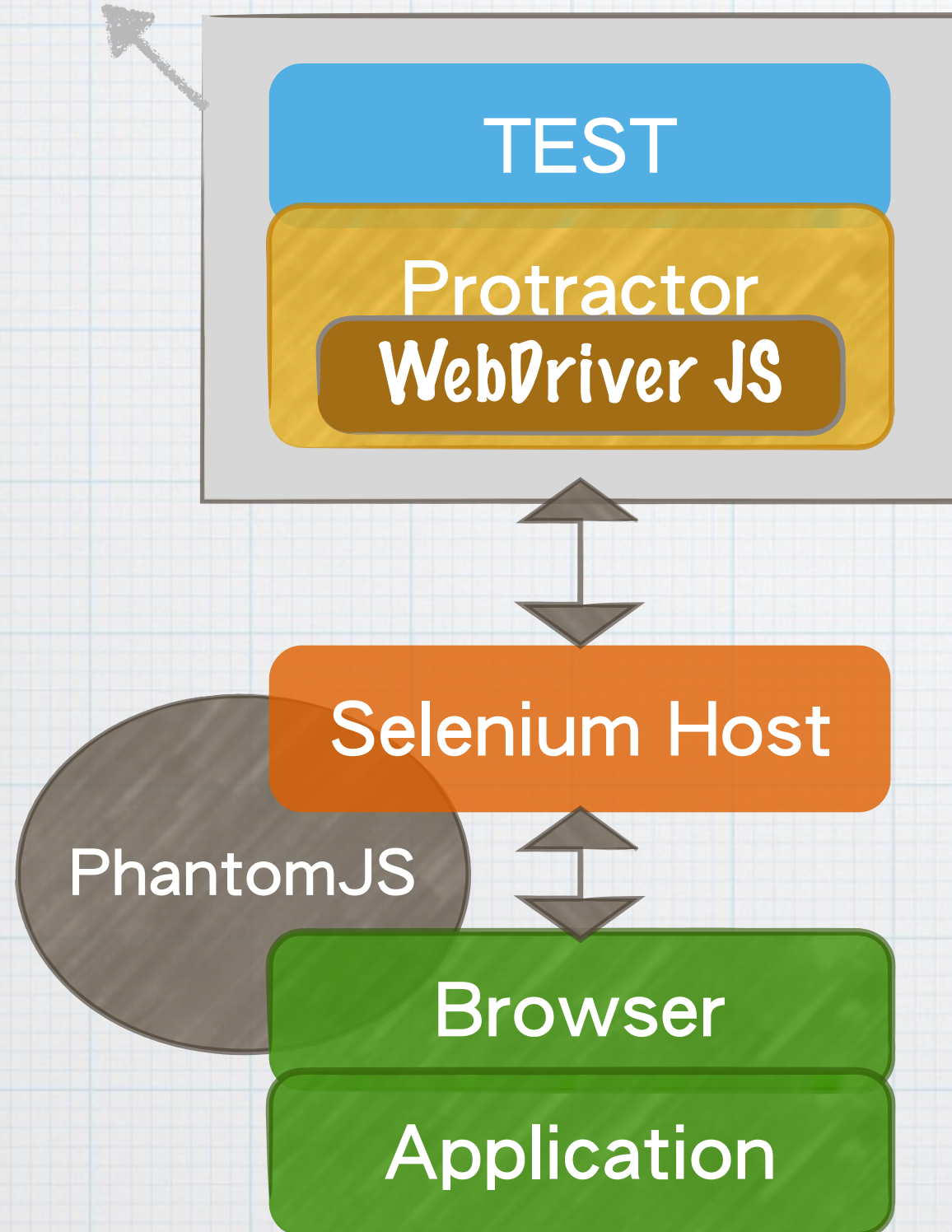


Protractor

```
describe("使用者登入後台", function() {  
  it('確定使用者可以在登入頁登入', function() {  
    browser().navigateTo('#/login');  
    expect(browser().location().path()).toBe("/login");  
    // 輸入登入表格  
    input('email').enter('test@test.com');  
    input('password').enter('password');  
    element('submit').click();  
    // 是否轉頁到後台  
    expect(browser().location().path()).toBe("/dashboard");  
    // 確定 email 欄位是否有使用者剛剛登入的 email  
    expect(element('#email').html()).toContain('test@test.com');  
  });  
});
```


E2E 流程

node JS



Write User story
(Jasmine)

Protractor 包裝
WebDriver

Selenium 驅動
Browser 跑模擬測試

with application

Unit Testing v.s E2E

	Unit Testing	E2E
Pro's	快速 可靠 測試單一功能	檢測程式整合 測試使用者真實情境
Con's	需花時間完整開發 需長期維護	很難測試每種情況 難以精準問題點
測試環境	console	browser

Unit Testing v.s E2E

- * Do Both!!
- * Unit Testing 可明確的找到問題點，
將問題範圍縮小
- * E2E 是測試當程式碼整合在一起後，
會遇到的問題

什麼什麼...?

Unit
Testing

* 工程師寫測試會有盲點，所以一定要 QA 來寫？

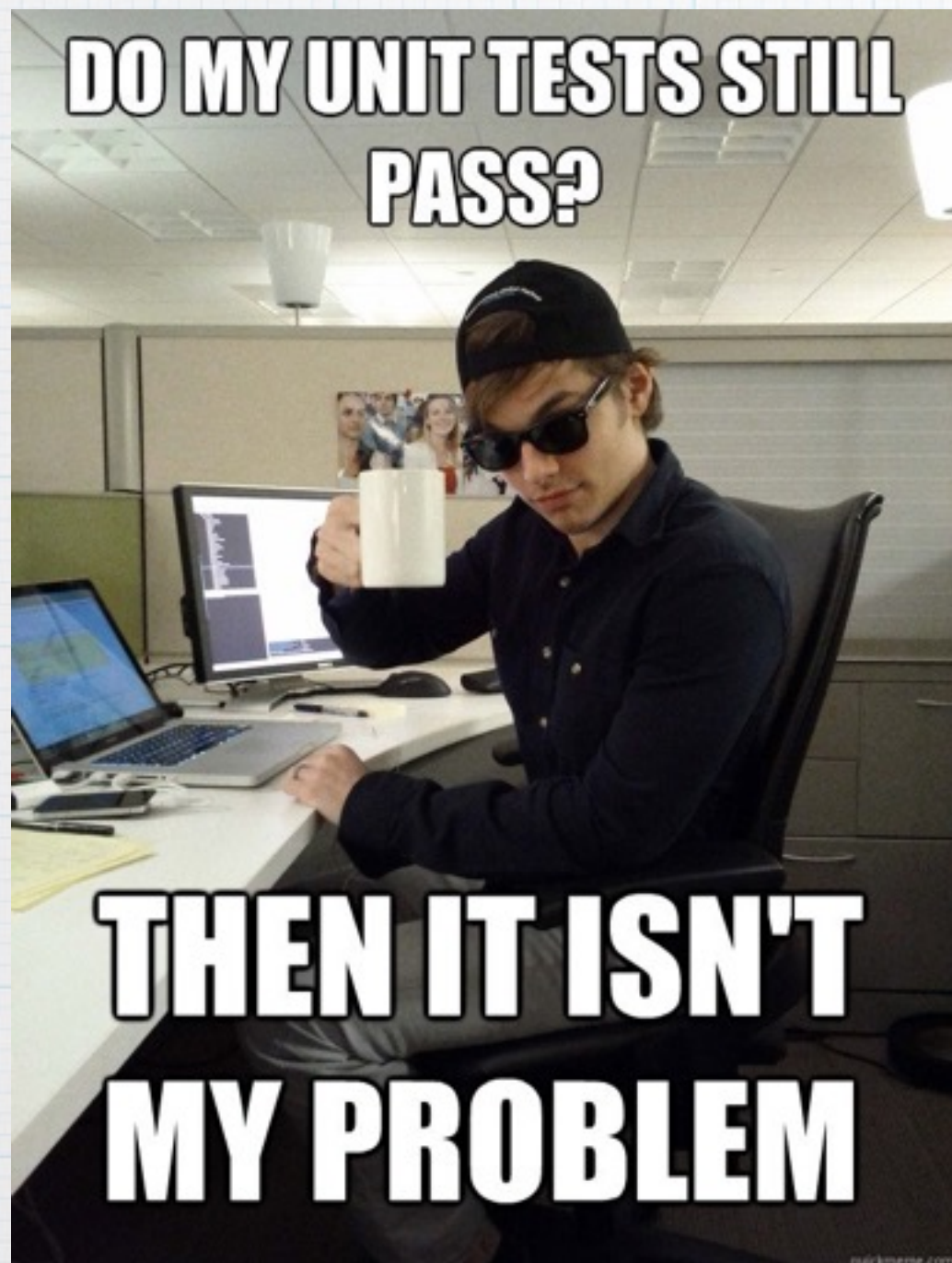
- 程式碼是工程師設計的，測試也要工程師自己寫最準
- QA 可策劃/協助 User story 的部分

E2E

* 我程式都寫完了，跑起來也都對。這時寫測試程式一點意義都沒有...

- 寫測試的三個時機：
 1. 當程式碼需要被其他程式呼叫的時候
 2. 修復 bug 前
 3. 需求變更的時候

完成 Unit Testing 後



完成 E2E 後



References

- * Automated Web Testing using JavaScript
- * Unit-testing and E2E testing in JS
- * Front-End Testing: Demystified
- * Unit Testing Best Practices in AngularJS
- * 為什麼要寫 unit test ? 為什麼要先寫測試 ?
- * 30天快速上手 TDD - 陳仕傑
- * 自动化e2e测试 - WebDriverJS , Jasmine和Protractor
- * Unit and integration Testing