

開發、測試 軟體工程常用技術

by 姜林毅



yi.jianglin@stariver-dig.com

大綱

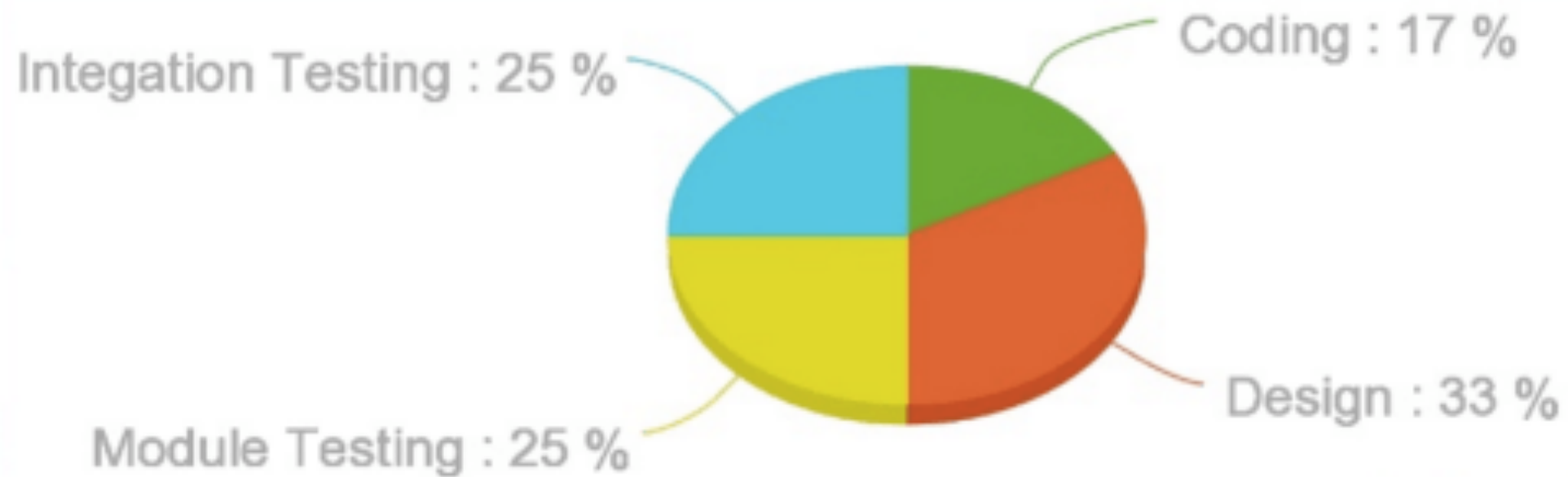
- 常見的專案開發模式
- 測試
- python單元測試
- Android自動化測試
- Sikuli
- ADB+OPENCV
- Docker+AndroidViewClient
- Git
- 面試題目、經驗分享

老闆：I have a DREAM

一個很厲害的“雲端計算機”



專案時間分配



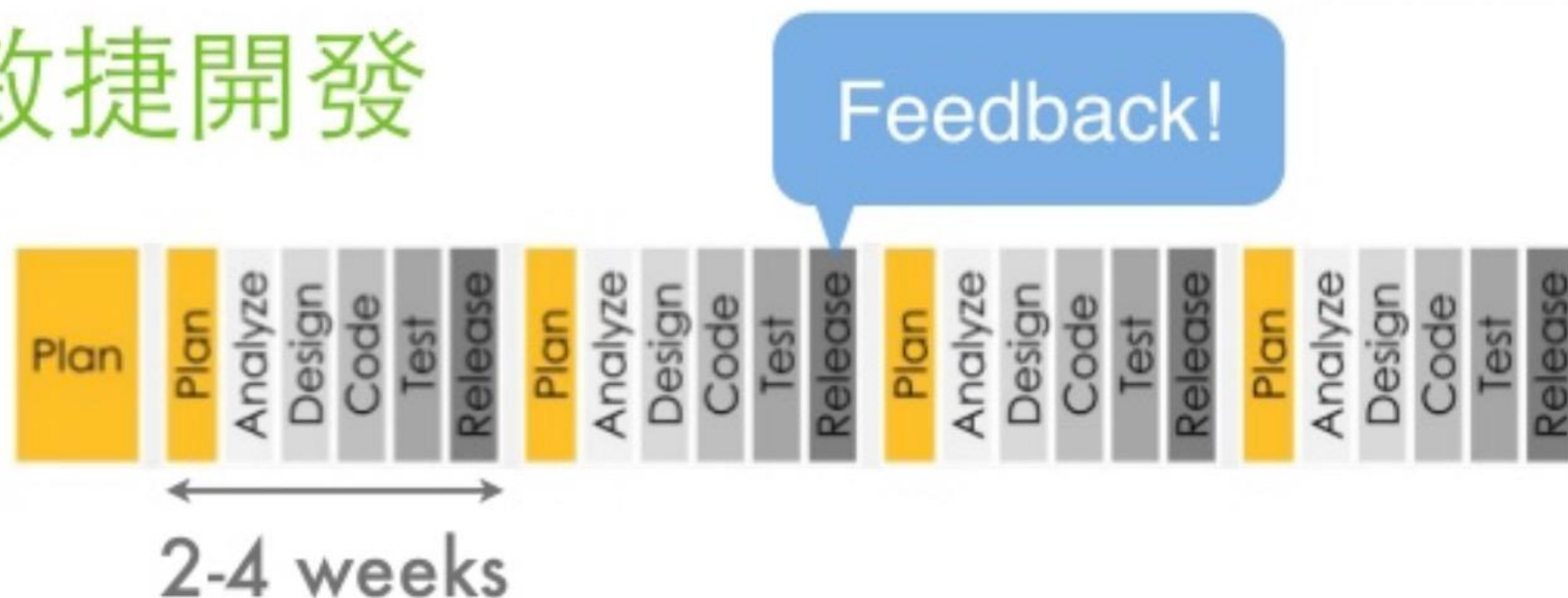
by : 人月神話

開發流程

瀑布開發



敏捷開發



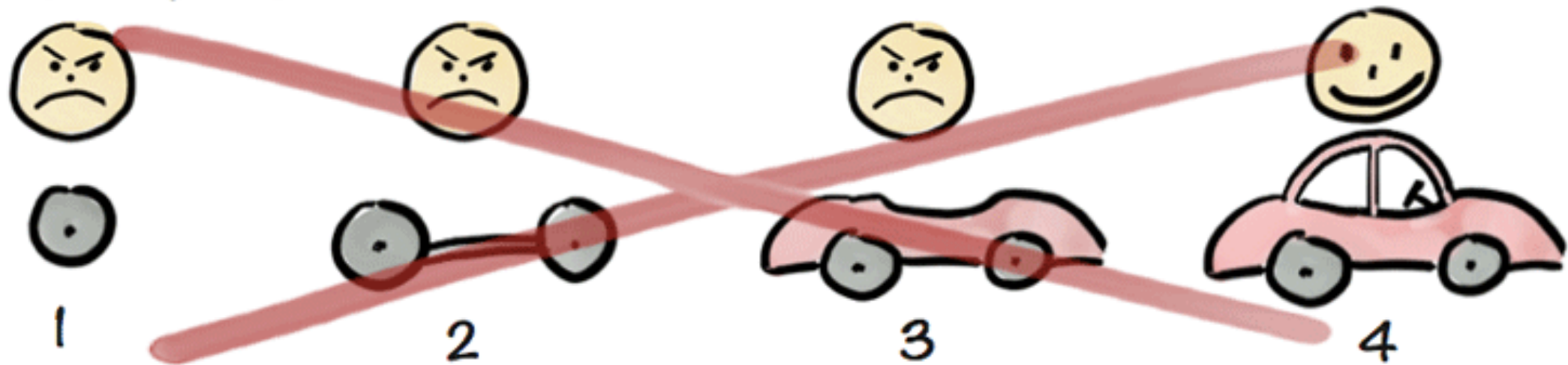
瀑布開發

- 目標、功能明確
- 不是第一次做這類型的案子
- 團隊成員非常有經驗

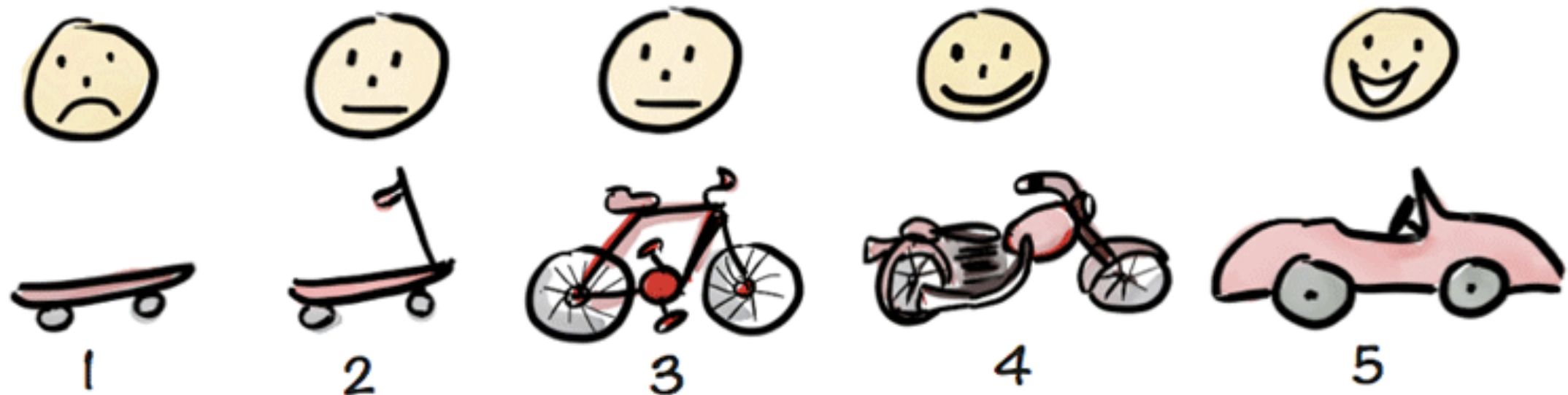
敏捷式開發

- 團隊成員都沒有經驗
- 創新型專案
- 產品的不確定性大

Not like this....



Like this!



By Henrik Kniberg

測試

- 測試是什麼
- 為什麼要測試
- 測試的方法、工具
- 測試工具還能做哪些有趣的玩意

測試

測試是什麼

- 測試的類型
 - 壓力測試
 - 效能測試
- 測試的階段
 - 單元測試
 - 整合測試
 - 系統測試
 - 回歸測試

為什麼需要測試

- 老闆
- pm
- 工程師
- 使用者

為什麼需要測試

推薦計算機類必讀書籍：

第一階段：

《C語言程序與設計》

《Java通識》

《C++進階寶典》

《Swift入門與實踐》

第二階段：

《教你怎麼不生氣》

《佛經》

《老子》

《沉默的憤怒》

第三階段：

《頸椎病康復指南》

《腰椎間盤突出日常護理》

《心髒病的預防與防治》

《高血壓降壓寶典》

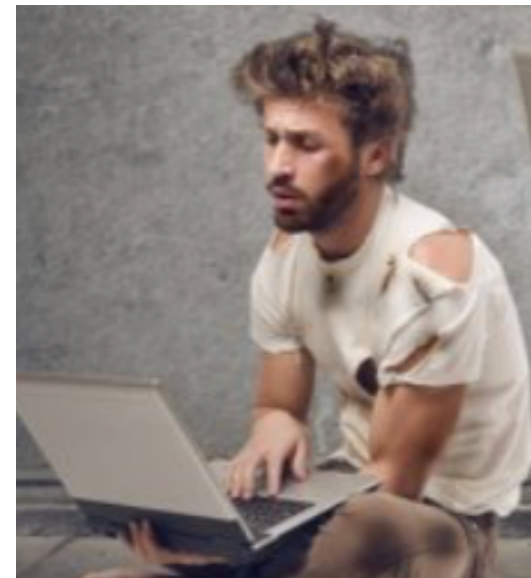
《強迫症的自我恢復》

《精神病症狀學》

第四階段：

《活著》

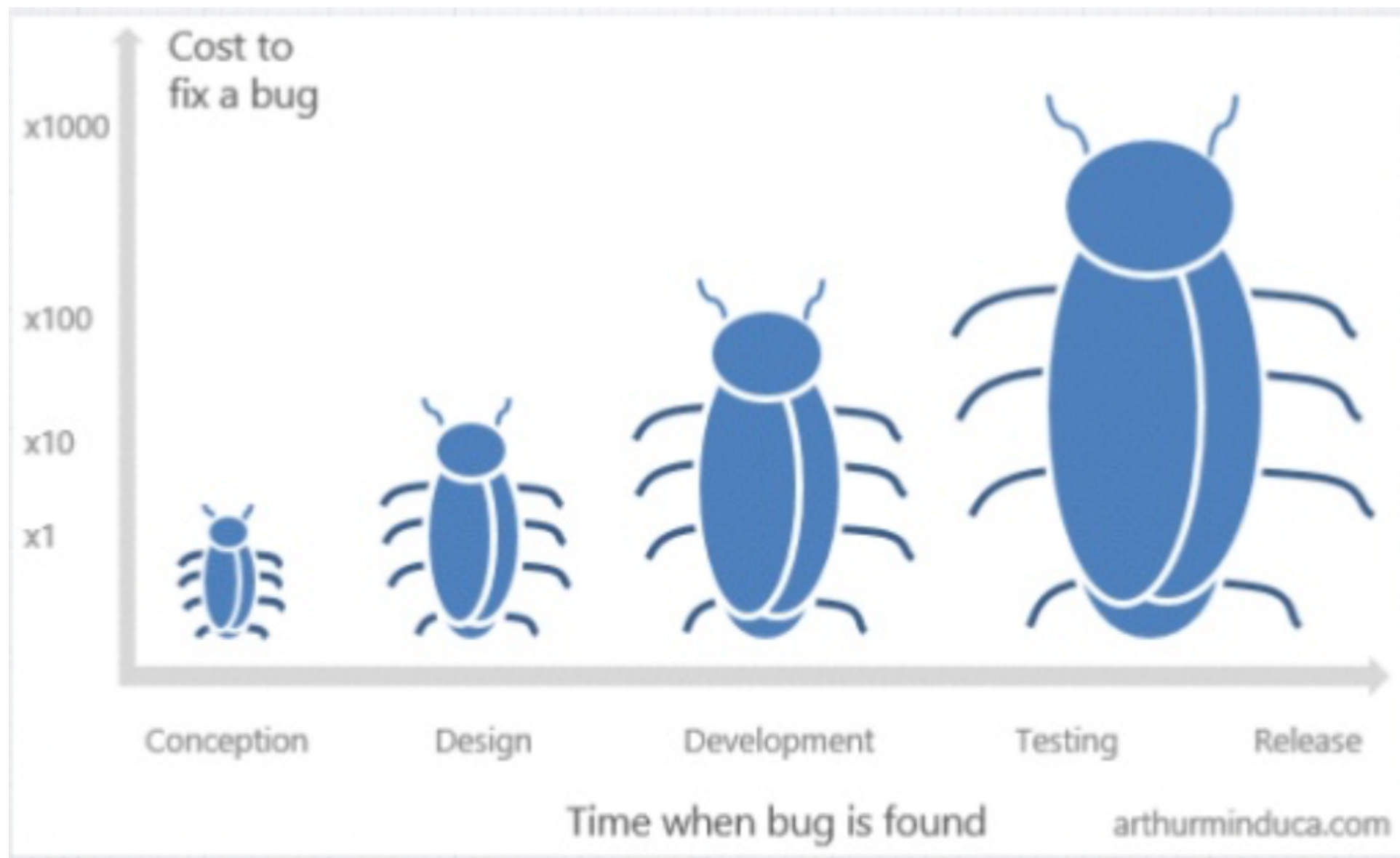
18則靠北工程師粉絲團中「讓人笑到崩潰的經典貼文」。#5會讓所有工程師哭到辭職...% 照片



當週末一堆人還在拼命的DEBUG，
我卻下班離開公司時



及早發現、及早治療



測試驅動開發 (TDD)

預防 > 治療

單元測試vs整合測試

- 手電筒
 - 鎢絲燈泡
 - 電池
 - 開關

自動化測試vs手動測試

..v/SomeScripts (zsh) 1

~ (zsh) 2

```
~/Dev/SomeScripts master x 1d
▶ python ui_automator.py 178440060@qq.com testpassword123
```

Android Emulator - Nexus_5X_-_N_-_A... 12:43

系统设置

系统设置

切换语言 跟随系统

清空缓存 59.62KB

更新

检查更新 当前版本(7.0.0-build270-dev-debug)

其他

重置通讯录

Power

Volume

Rotation

Camera

Search

Navigation

App Drawer

More

Navigation

Home

App Drawer

寫測試的最大敵人——

Deadline

工程師最不想點開的訊息



老婆

今天幾點回家

下午6:09

3



老闆

今天在做什麼

下午6:00

1



PM

這功能需要多久時間？

下午5:57

1



客戶

無痕模式是什麼？

下午4:52

8



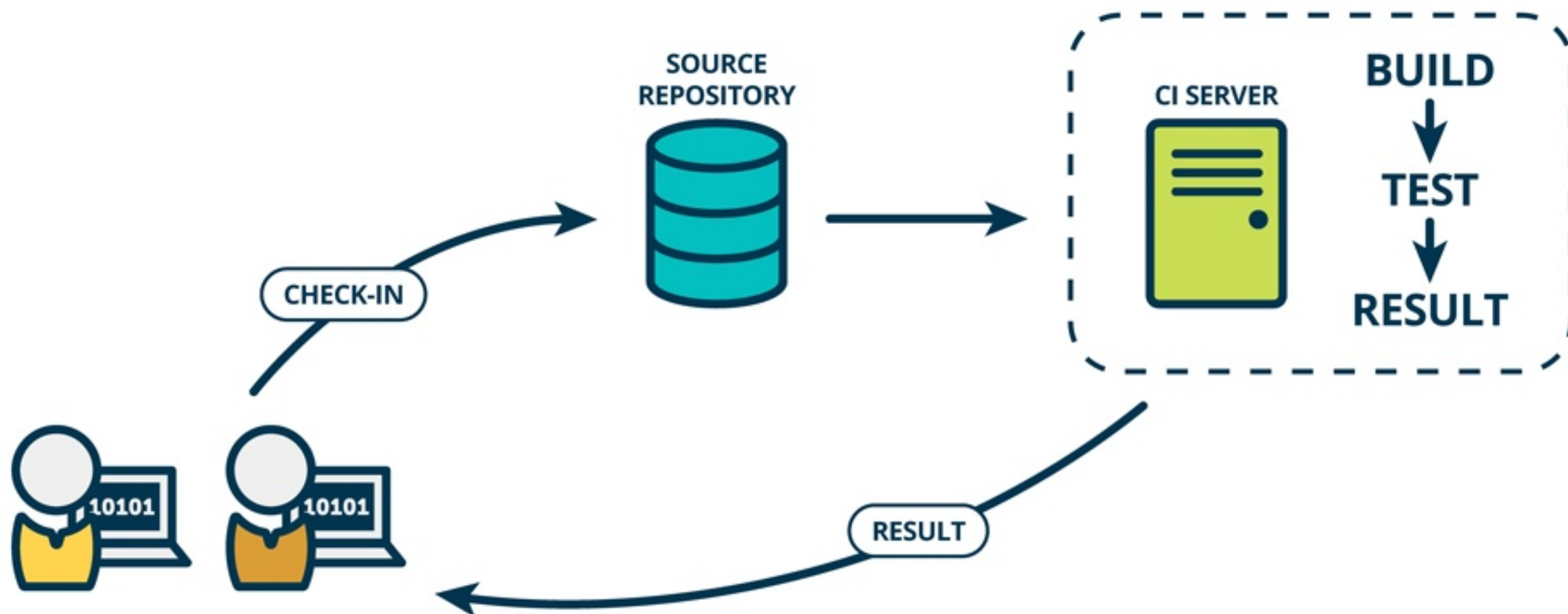
系統測試案（已結案）(8)

那個我這邊有一些問題 可以有空的時候打給我嗎

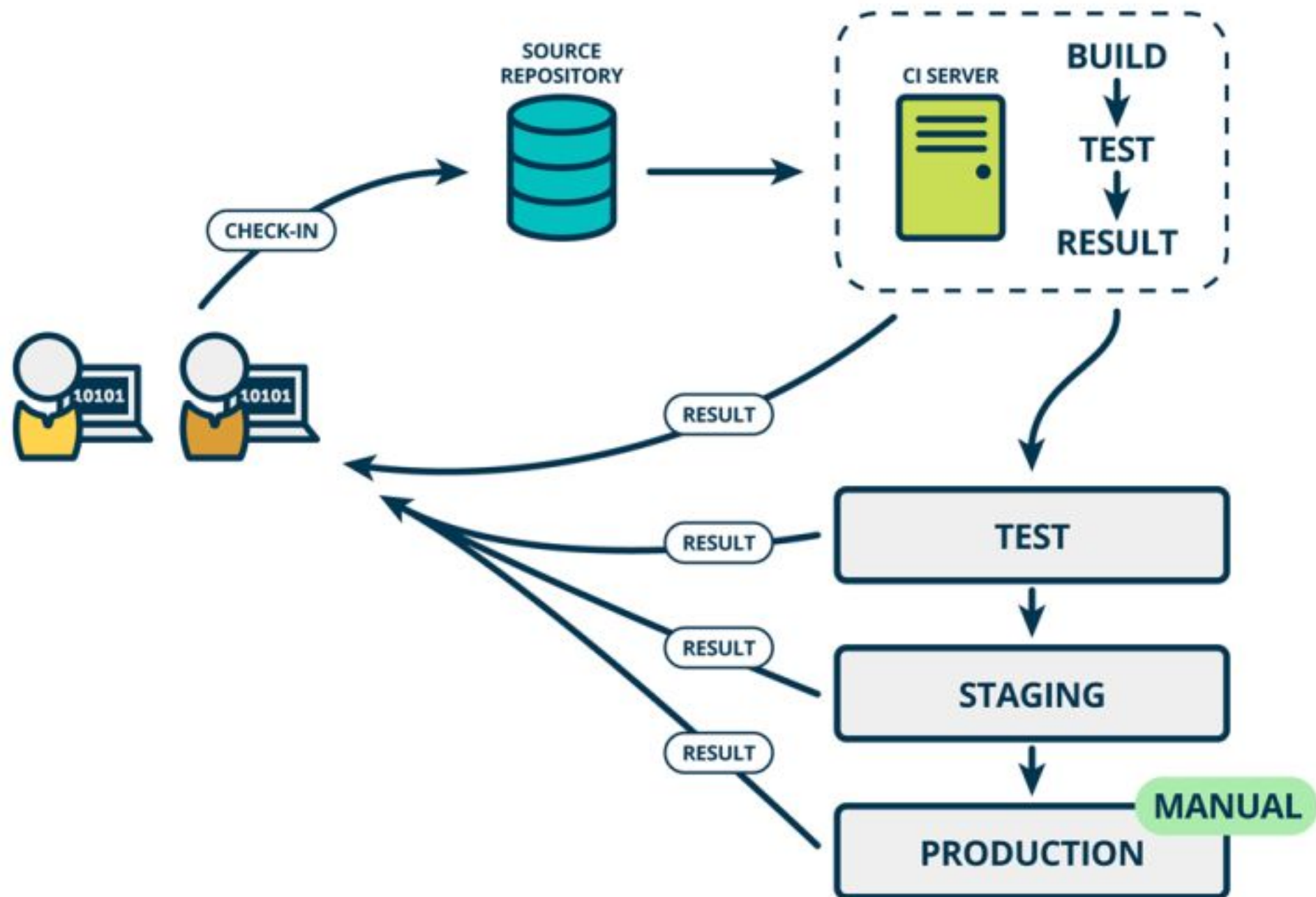
上午11:00

49

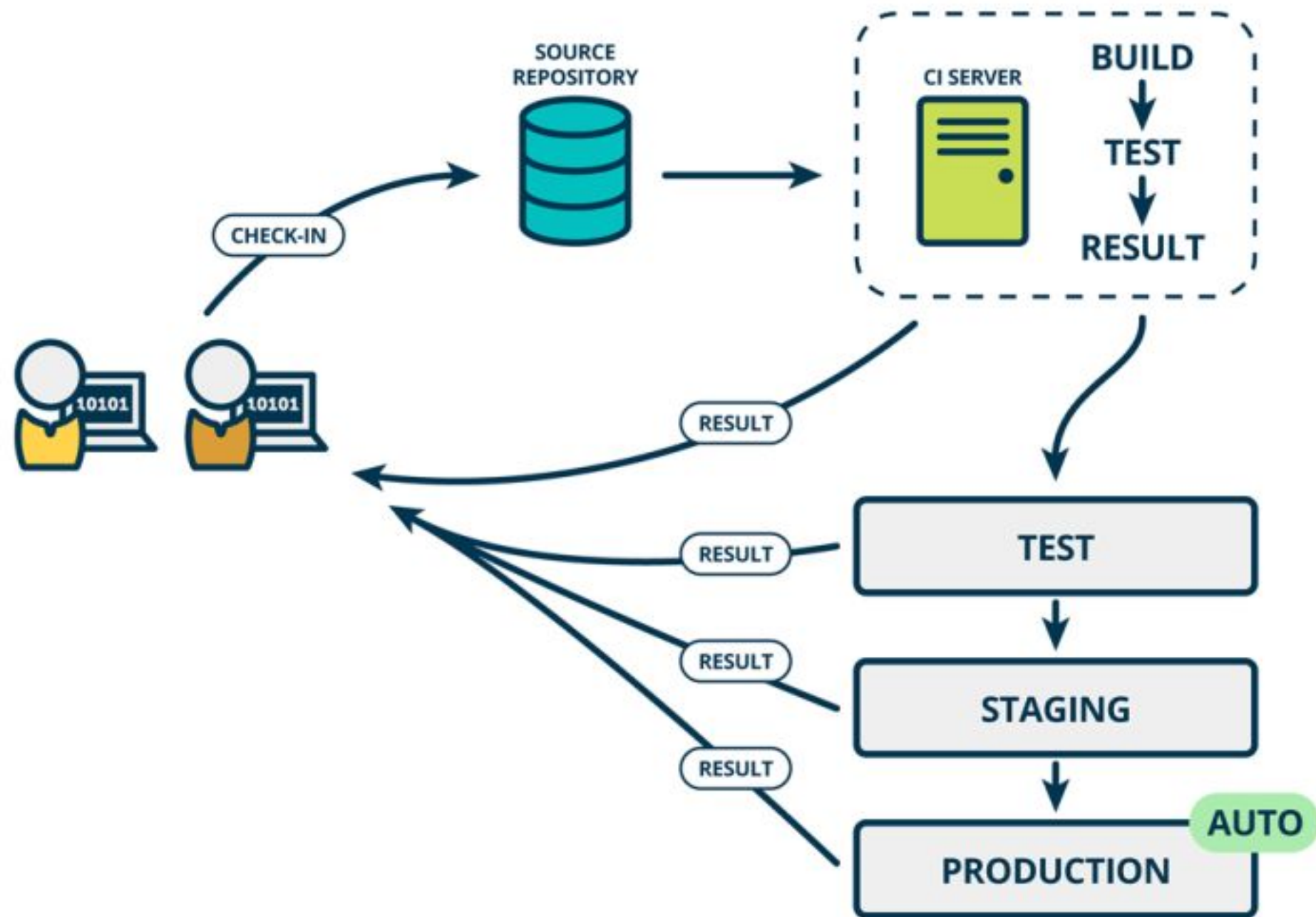
持續集成



持續交付



持續部署



什麼樣的專案需要導入軟體工程

- 一個以上的軟體設計師
- 程式規模龐大
- 問題與需求不是一目瞭然

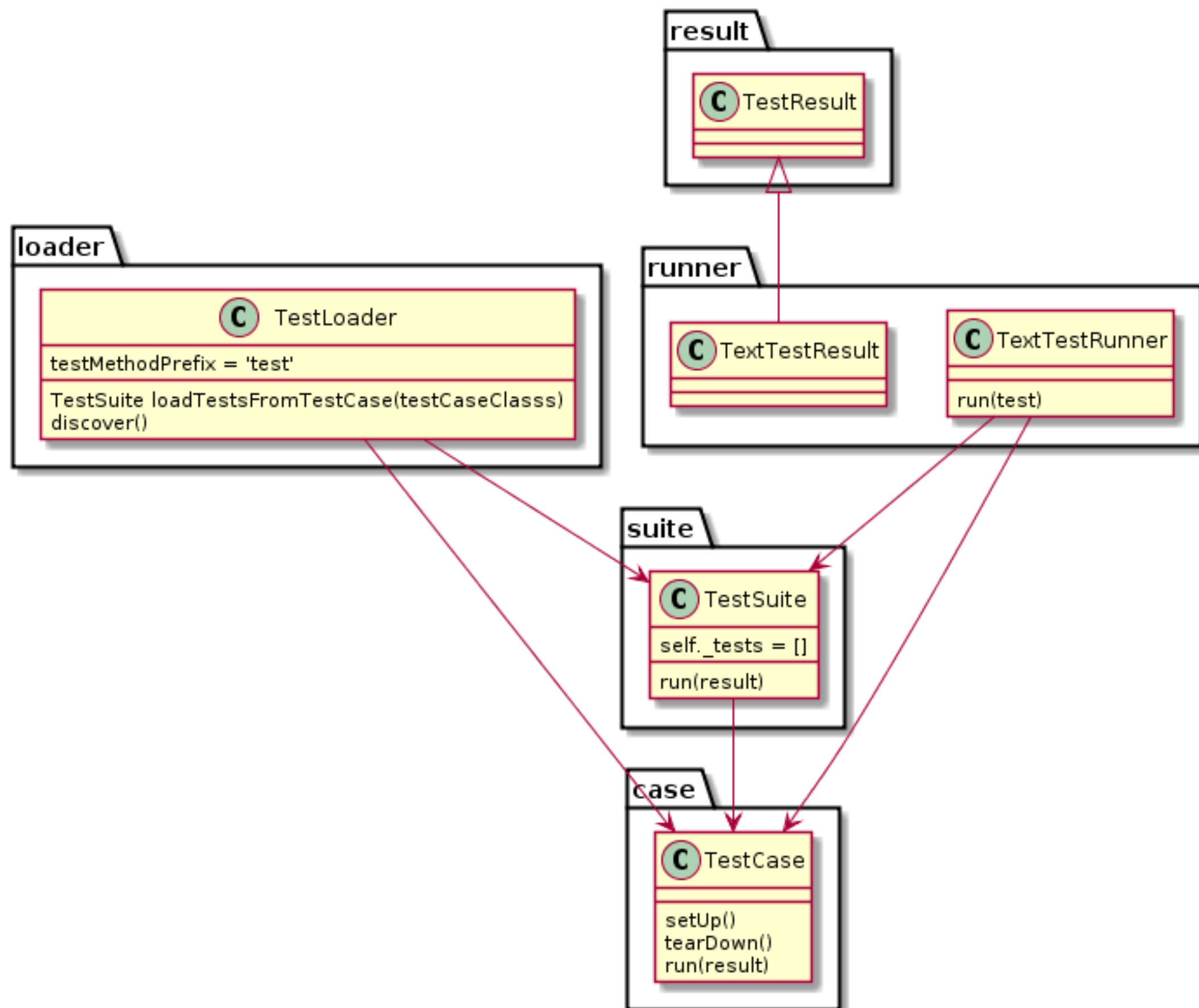
除非最後一個使用者死掉，
軟體是不會有完工的時候。

python 單元測試

Unittest

Unittest組件

- 測試案例(Test Case): 測試的最小單元。
- 測試套件(Test Suite): 一組測試案例、測試套件或者是兩者的組合。
- 測試加載器(Test Loader): 用以加載測試案例到測試套件。
- 測試執行器(Test Runner): 負責執行測試並提供測試結果的元件。
- 測試設備(Test Fixture): 執行一或多個測試前必要的預備資源，以及相關的清除資源動作。



安裝python

- `sudo apt-get install python2.7`
- `tar xvf sampleCode.tar`

```
1  def add(a, b):  
2      return a+b  
3  
4  def minus(a, b):  
5      return a-b  
6  
7  def multi(a, b):  
8      return a*b  
9  
10 def divide(a, b):  
11     return a/b  
12 |
```

Test Case

```
3 import unittest
4 from mathfunc import *
5
6
7 class TestMathFunc(unittest.TestCase):
8     """Test mathfuc.py"""
9     def setUp(self):
10         print "do something before test.Prepare environment."
11
12     def tearDown(self):
13         print "do something after test.Clean up."
14
15     def test_add(self):
16         """Test method add(a, b)"""
17         self.assertEqual(3, add(1, 2))
18         self.assertNotEqual(3, add(2, 2))
19
20     def test_minus(self):
21         """Test method minus(a, b)"""
22         self.assertEqual(1, minus(3, 2))
23
24     def test_multi(self):
25         """Test method multi(a, b)"""
26         self.assertEqual(6, multi(2, 3))
27
28     def test_divide(self):
29         """Test method divide(a, b)"""
30         self.assertEqual(2, divide(6, 3))
31         self.assertEqual(2.5, divide(5, 2))
32         self.assertEqual(2.5, divide(5.0, 2.0))
33
34 if __name__ == '__main__':
35     unittest.main()
36 # unittest.main(module=__file__, argv=None, exit=1)
```

```
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest# python test_mathfunc.py
.F..
=====
FAIL: test_divide (__main__.TestMathFunc)
Test method divide(a, b)
-----
Traceback (most recent call last):
  File "test_mathfunc.py", line 26, in test_divide
    self.assertEqual(2.5, divide(5, 2))
AssertionError: 2.5 != 2
-----
Ran 4 tests in 0.001s

FAILED (failures=1)
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest#
```

verbosity=2

```
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest# python test_mathfunc.py
test_add (__main__.TestMathFunc)
Test method add(a, b) ... ok
test_divide (__main__.TestMathFunc)
Test method divide(a, b) ... FAIL
test_minus (__main__.TestMathFunc)
Test method minus(a, b) ... ok
test_multi (__main__.TestMathFunc)
Test method multi(a, b) ... ok

=====
FAIL: test_divide (__main__.TestMathFunc)
Test method divide(a, b)
-----
Traceback (most recent call last):
  File "test_mathfunc.py", line 26, in test_divide
    self.assertEqual(2.5, divide(5, 2))
AssertionError: 2.5 != 2
-----

Ran 4 tests in 0.007s

FAILED (failures=1)
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest#
```

Test Suite & Test Runner

```
import unittest
from test_mathfunc import TestMathFunc

if __name__ == '__main__':
    suite = unittest.TestSuite()

    tests = [TestMathFunc("test_add"), TestMathFunc("test_minus"), TestMathFunc("test_divide")]
    suite.addTests(tests)

    runner = unittest.TextTestRunner(verbosity=2)
    runner.run(suite)
```

Test Fixture

- setUp()
- tearDown()

輸出結果到文件

```
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest# cat UnittestTextReport.txt
test_add (test_mathfunc.TestMathFunc)
Test method add(a, b) ... ok
test_minus (test_mathfunc.TestMathFunc)
Test method minus(a, b) ... ok
test_divide (test_mathfunc.TestMathFunc)
Test method divide(a, b) ... FAIL

=====
FAIL: test_divide (test_mathfunc.TestMathFunc)
Test method divide(a, b)
-----
Traceback (most recent call last):
  File "/home/jianglinyi/stust/stust_py/unittest/test_mathfunc.py", line 39, in test_divide
    self.assertEqual(2.5, divide(5, 2))
AssertionError: 2.5 != 2

-----
Ran 3 tests in 0.001s

FAILED (failures=1)
root@jianglinyi-VirtualBox:/home/jianglinyi/stust/stust_py/unittest#
```

輸出結果到網頁形成報表

MathFunc Test Report

Start Time: 2019-05-18 18:52:08

Duration: 0:00:00.000203

Status: Pass 2 Failure 1

generated by HTMLTestRunner.

Show [Summary](#) [Failed](#) [All](#)

Test Group/Test case	Count	Pass	Fail	Error	View
test_mathfunc.TestMathFunc: Test mathfuc.py	3	2	1	0	Detail
test_add: Test method add(a, b)	pass				
test_minus: Test method minus(a, b)	pass				
test_divide: Test method divide(a, b)	fail				
Total	3	2	1	0	

測試驅動開發 (TDD)

預防 > 治療

Android 自動化測試

方向

- Android 自動化測試框架
- 各大雲測試平台

UI介面測試工具

- 絕對座標
- 元件辨識
- 圖形辨識

Android自動化測試框架

- Espresso
- UI Automator
- Appium
- AndroidViewClient

自動化測試框架比較

	UiAutomator	Appium	Robotium	Espresso
支持语言	Java	Java、python、Ruby、PHP等	Java	Java
是否需要签名	不需要	不需要	不需要	不需要
是否支持跨应用测试	支持	不支持	不支持	不支持
是否跨平台	仅支持Android平台	Android、IOS都支持	仅支持Android平台	仅支持Android平台
是否支持WebView	不支持	支持Native App、Hybird App、Web App	不支持	不支持

项目工具	UiAutomator	Monkeyrunner	Robotium	realMonkey	Selendroid	Calabash	Frank	Appium+OpenCV
权限	普通	普通	普通	普通	普通	普通	普通	普通
是否支持WebView	否	否	4.0版本以上支持	否	是	是	?	是
是否支持跨应用测试	是	否	否	是	否	否	?	是
Android API要求	API 16	?	API 7	API 16	API 10	?	?	API 17
是否支持控件ID	是	是	是	是	是	是	?	是
是否开源	是	是	是	是	否	否	是	是
是否支持Hybrid APP测试	否	否	否	否	是	是	?	是
适用移动端系统	Android	Android	Android	Android	Android	Android/iOS	iOS	Android/iOS
是否支持脚本录制	否	是	是	否	否	否	是	Mac平台支持
是否依赖于内部资源	是	是	是	是	是	是	是	否

雲測試服務比較

	兼容性测试	性能测试	功能测试	遍历测试	稳定性测试	网络场景测试	真机租用
Testin云测	√ (Android: 自动遍历时长3分钟, 30秒monkey测试) (IOS[专业版]: 指定脚本10分钟)	√ (包含在兼容性测试中)	√ (Android) (录制工具/自行编写)	√ (Android) (3分钟, 包含在标准兼容测试中)	×	×	√ (Android、iOS均有)
百度MTC	√ (安装、启动、monkey、卸载、自动登录)	√ (1、包含在兼容测试中; 2、深度性能支持Android)	√ (Android) (录制工具/自行编写)	√ (Android)	×	×	×
阿里MQC	√ (安装、启动、monkey、卸载、覆盖登录)	√ (Android) (深度性能测试: 内存泄漏、溢出、抖动、卡顿、主线程、过度绘制)	√ (Android: 自行编写) (iOS: UI Automation框架)	×	√ (Android) (1小时monkey测试模拟器上)	×	√ (单机租用、双机租用、33款真机)
腾讯优测	√ (标准兼容Android: 安装、启动、登录、控件遍历5min, 卸载) (专家兼容Android&iOS: 人工测试, 定制化)	√ (云端+本地)	√ (需要写自动化脚本)	√	×	×	√ (Android) (单机租用964款真机)
AWS Device Farm	√ (支持Android和iOS)	√ (支持脚本)	√ (支持脚本)	×	×	√	√

Sikuli

Sikuli，一種圖型化程式語言，使用者只須有最基本的程式設計技能，不需要寫出一行行程式，而是用螢幕截圖的方式將截出來的圖形組合出程式。

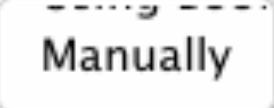
```
ip = input("Please enter the IP address:")  
gateway = ".".join(ip.split('.')[0:3] + ['254'])
```

```
switchApp("System Preferences.app")
```

```
click()
```

```
click()
```

```
click()
```

```
click()
```

```
wait()
```

```
type(ip + "\t")
```

```
type("255.255.255.0\t")
```

```
type(gateway+"\t")
```

```
click()
```

Sikuli

- 程式碼的可讀性(readability)、易用性(usability)
- 人機互動過程「數位化」的一種方法
- 電腦視覺的研究領域，從真實世界延伸到電腦的桌面
- open source

ADB+OPENCV
基於python

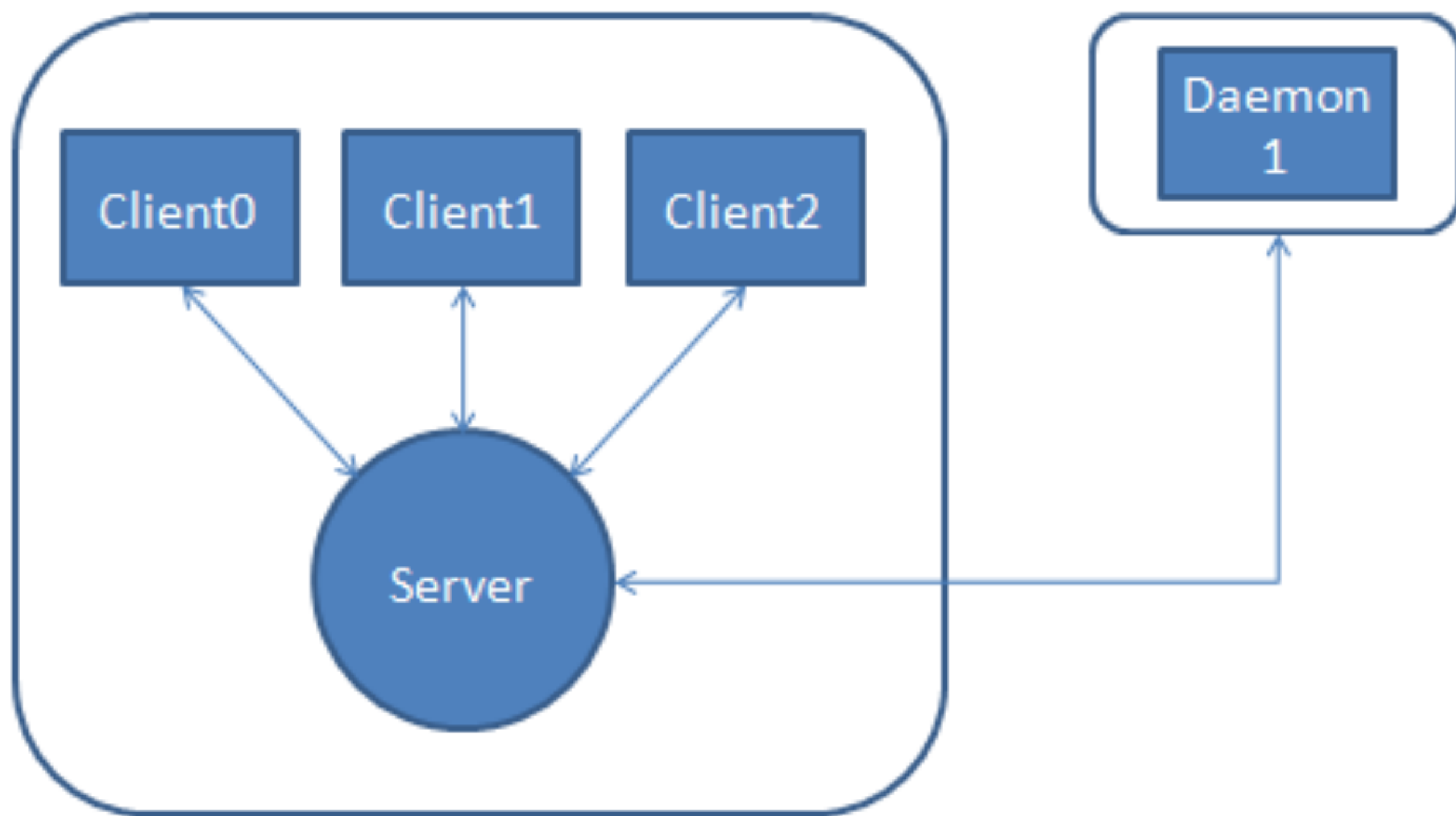
安裝模擬器

- `sudo add-apt-repository ppa:morphis/anbox-support`
- `sudo apt update`
- `sudo apt install -y anbox-modules-dkms`
- `sudo modprobe ashmem_linux`
- `sudo modprobe binder_linux`
- `sudo snap install --devmode --beta anbox`
- `sudo apt install android-tools-adb`

Android Debug Bridge

PC

手机



Android Debug Bridge

- adb client
- adb server
- adb daemon

Adb Command

adb截圖操作

- `adb shell screencap -p /sdcard/screencap.png`
- `adb pull /sdcard/screencap.png`

Demo & 實作

安裝截圖軟體

- `sudo add-apt-repository ppa:shutter/ppa`
- `sudo apt update`
- `sudo apt-get install shutter`
- `shutter &`

安裝python+adb+pip+opencv

- `sudo apt-get install python2.7`
- `sudo apt install python-pip`

安裝python+adb+pip+opencv

```
1 sudo gedit /usr/bin/pip
```

將原来的：

```
1 from pip import main
2 if __name__ == '__main__':
3     sys.exit(main())
```

改成：

```
1 from pip import __main__
2 if __name__ == '__main__':
3     sys.exit(__main__.__main__())
```

安裝python+adb+pip+opencv

- `pip install --upgrade setuptools`
- `pip install numpy Matplotlib`
- `pip install opencv-python`

安裝Sublime Text3

- `sudo add-apt-repository ppa:webupd8team/sublime-text-3`
- `sudo apt-get update`
- `sudo apt-get install sublime-text-installer`
- `subl`

圖形辨識自動化測試



Demo & 實作

ADB封装

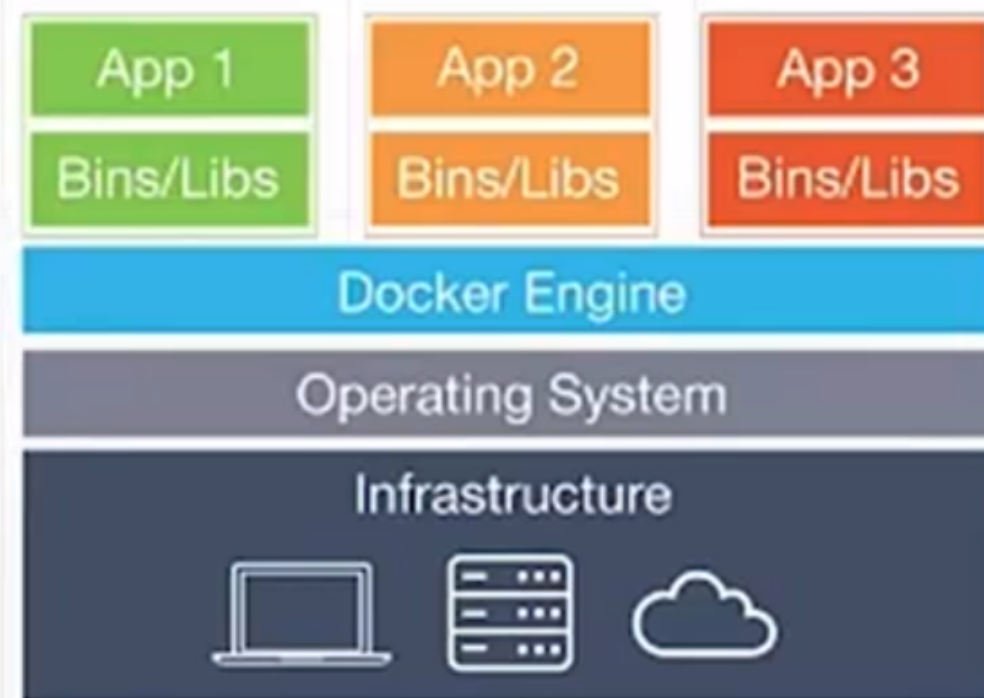
OPENCV影像識別

Docker+AndroidViewClient

Docker是什麼



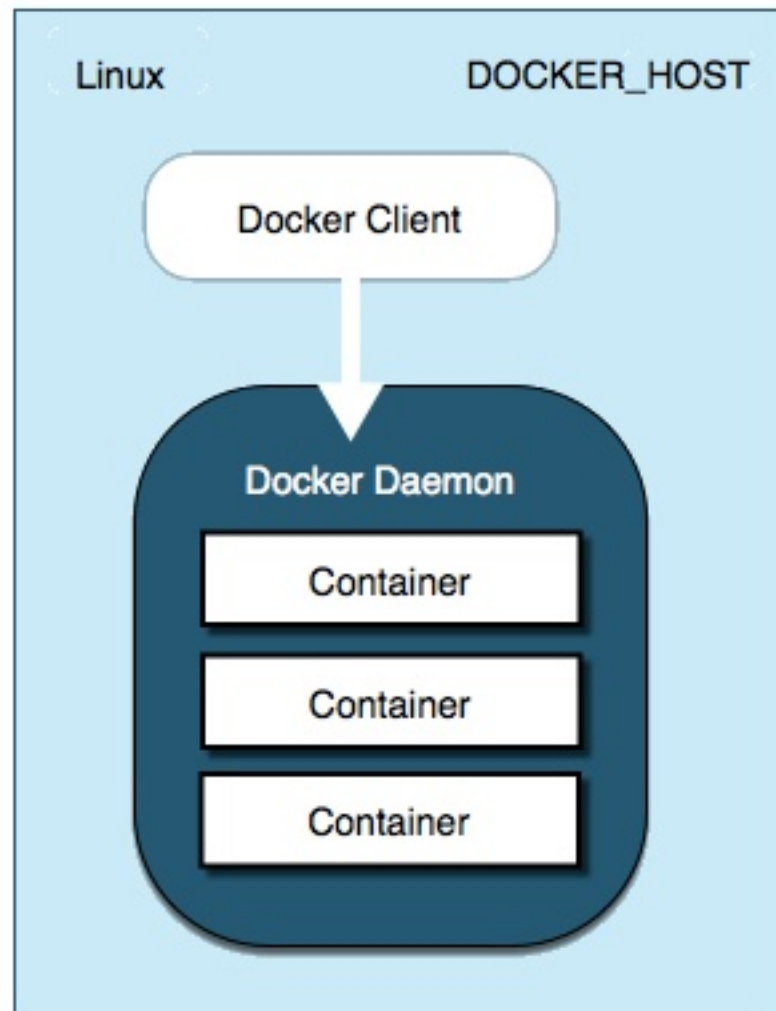
Virtual Machines



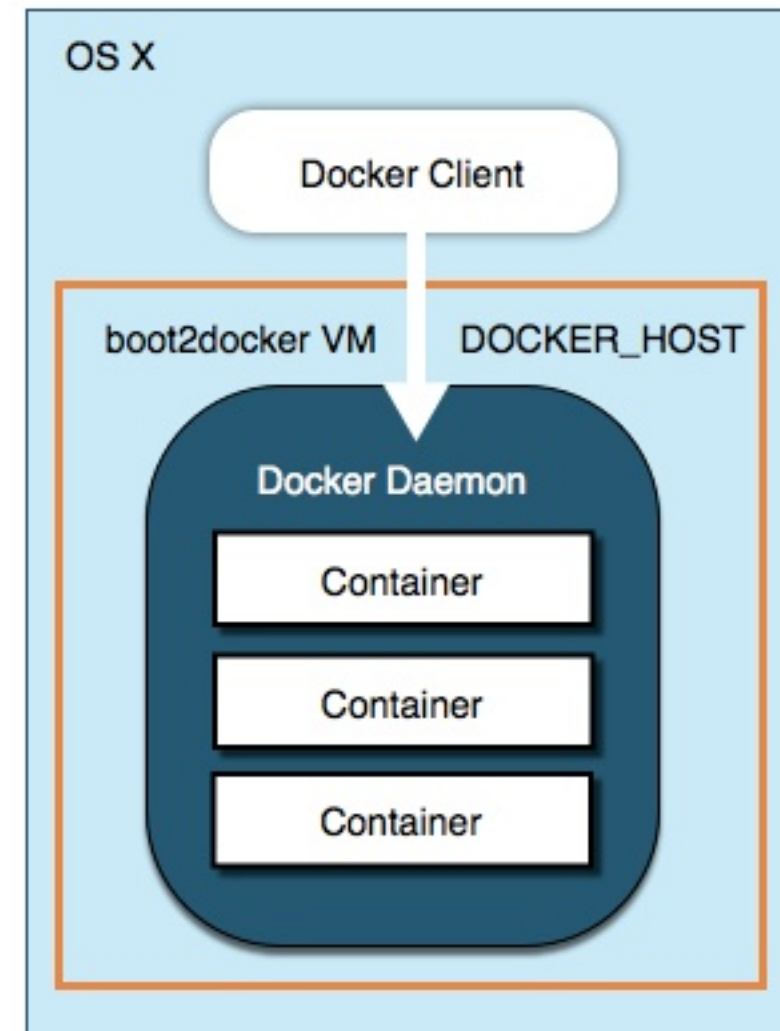
Containers

Difference between OS

Linux



OS X & Windows



為什麼要用Docker

- 更高效地利用系統資源
- 更快速的啟動時間
- 確保運行環境一致
- 更快速的交付和部署
- 更輕鬆的遷移
- 更簡單的管理、擴展

Docker vs 虛擬機

特性	容器	虛擬機
啟動	秒級	分鐘級
硬碟容量	一般為 MB	一般為 GB
效能	接近原生	比較慢
系統支援量	單機支援上千個容器	一般幾十個

Docker 基本概念

- 映像檔(Image)
- 容器(Container)
- 倉庫(Repository)

Docker運作原理



Docker install

- `sudo apt-get remove docker docker-engine docker.io`
- `sudo apt-get update`
- `sudo apt-get install \`
 `apt-transport-https \`
 `ca-certificates \`
 `curl \`
 `software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key`
 `add -`
- `sudo add-apt-repository \`
 `"deb [arch=amd64] https://download.docker.com/linux/ubuntu \`
 `$(lsb_release -cs) \`
 `stable"`

Docker install

- `sudo apt-get update`
- `sudo apt-get install docker-ce`
- `docker -v`

Docker sample

- `docker pull ubuntu:14.04`
- `docker run -t -i ubuntu:14.04 /bin/bash`
- `docker image pull hello-world`
- `docker run hello-world`

Dockerfile

By 《Docker —— 從入門到實踐》

- COPY 複製文件
- ADD 更高級的複製文件
- CMD 容器啟動命令
- ENTRYPOINT 入口點
- ENV 設置環境變量
- ARG 構建參數
- VOLUME 定義匿名卷
- EXPOSE 開放端口
- WORKDIR 指定工作目錄
- USER 指定當前用戶
- HEALTHCHECK 健康檢查
- ONBUILD 為他人作嫁衣裳

Dockerfile

```
FROM centos:7  
MAINTAINER jianglinyi
```

```
RUN yum install -y wget
```

```
RUN cd /
```

```
ADD jdk-8u152-linux-x64.tar /
```

```
RUN wget http://apache.stu.edu.tw/tomcat/tomcat-7/v7.0.94/bin/  
apache-tomcat-7.0.94.tar.gz
```

```
RUN tar zxvf apache-tomcat-7.0.94.tar.gz
```

```
ENV JAVA_HOME=/jdk1.8.0_152
```

```
ENV PATH=$PATH:/jdk1.8.0_152/bin
```

```
CMD ["/apache-tomcat-7.0.94/bin/catalina.sh", "run"]
```

Dockerfile

```
docker build -t mytomcat . --no-cache
```

```
docker images
```

```
docker run -p 8080:8080 mytomcat
```

匯出匯入映像檔

By 《Docker —— 從入門到實踐》

儲存映像檔

如果要建立映像檔到本地檔案，可以使用 `docker save` 命令。

```
$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
ubuntu              14.04              c4ff7513909d       5 weeks ago        225.4 MB
...
$ sudo docker save -o ubuntu_14.04.tar ubuntu:14.04
```

載入映像檔

可以使用 `docker load` 從建立的本地檔案中再匯入到本地映像檔庫，例如

```
$ sudo docker load --input ubuntu_14.04.tar
```

或

```
$ sudo docker load < ubuntu_14.04.tar
```

上傳你的映像檔

By 《Docker —— 從入門到實踐》

上傳映像檔

使用者可以通過 `docker push` 命令，把自己建立的映像檔上傳到倉庫中來共享。例如，使用者在 Docker Hub 上完成註冊後，可以推送自己的映像檔到倉庫中。

```
$ sudo docker push ouruser/sinatra
```

```
The push refers to a repository [ouruser/sinatra] (len: 1)
```

```
Sending image list
```

```
Pushing repository ouruser/sinatra (3 tags)
```

為什麼要用Docker

- 更高效地利用系統資源
- 更快速的啟動時間
- 確保運行環境一致
- 更快速的交付和部署
- 更輕鬆的遷移
- 更簡單的管理、擴展

Docker Hub

AndroidViewClient

AndroidViewClient install

- `docker pull panserbjorne/adb-avc`
- `docker run -t -i panserbjorne/adb-avc sh`
- `docker run --network=host -t -i -v /home/jianglinyi/stust_py/AndroidViewClient-master/:/home/panserbjorne/adb-avc sh`

AndroidViewClient

- `cd /home` 查看是否成功掛載
- `adb disconnect` 外部的ubuntu斷線
- `adb devices` 虛擬機內查看是否連接上
- 未連接上輸入 `adb connect 127.0.0.1`
- `adb shell pm list packages` 查看Android 所有app名稱
- `cd /home/examples`
- `python dump-simple.py`

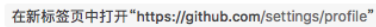
Demo & 實作

Git

申辦GitHub帳號

Add SSH KEY

- `ssh-keygen -t rsa -b 4096 -C "chianglin1993@gmail.ocm"`
- enter
- enter
- `cd ~/.ssh/`
- `cat id_rsa.pub` (返回結果複製起來)
- 登入GitHub上，右上角頭像->Setting->SSH and GPG keys
中，點擊new SSH key，貼上。
- `ssh -T git@github.com`



```
root@jianglinyi-VirtualBox:~/.ssh# ssh -T git@github.com
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGL7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
Hi love3499! You've successfully authenticated, but GitHub does not provide shell access.
root@jianglinyi-VirtualBox:~/.ssh#
```

新增一個repository

 [Explore](#) [Gist](#) [Blog](#) [Help](#) MarsW   



MarsW
MarsW

[JavaScript](#)
[NCTU](#)
[Taiwan](#)
[ymjh11436@gmail.com](#)
[http://www.marsw.tw](#)
Joined on Mar 07, 2012

1
follower

0
starred

0
following

[Contributions](#) [Repositories](#) [Public Activity](#) [Edit Your Profile](#)

[All](#) [Public](#) [Private](#) [Sources](#) [Forks](#) [Mirrors](#) [New](#)

 **marsw.github.com**
Last updated 2 months ago

JavaScript ★ 0 0





Search or type a command



[Explore](#) [Gist](#) [Blog](#) [Help](#)

MarsW



PUBLIC



Owner



MarsW

Repository name

test



Great repository names are short and memorable. Need inspiration? How about [ducking-octo-happiness](#).

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



☒ Initialize this repository with a README

This will allow you to `git clone` the repository immediately.

Add .gitignore: **None**

Add a license: **None**



Create repository



PUBLIC



MarsW / test

Unwatch 1

Star 0

Fork 0

Description

Short description of this repository

Website

Website for this repository (optional)

Save or cancel

1 commit

1 branch

0 releases

1 contributor



branch: master

test / +

Initial commit

MarsW authored a minute ago

latest commit e13cffa6ea

README.md

Initial commit

a minute ago

README.md

test

Code

Issues 0

Pull Requests 0

Wiki

Pulse

Graphs

Network

Settings

HTTPS clone URL

https://github.com/



You can clone with [HTTPS](#), [SSH](#), [Subversion](#), and [other methods](#).

Clone in Desktop

Download ZIP

本機端初始化

- `git config --global user.name "jianglinyi"`
- `git config --global user.email "chianglin1993@gmail.com"`
- `git clone 位置`

新增分支

- `git branch v1`
- `git checkout v1`

git 操作

- git status
- git add
- git commit -m “”

git 合并

- git checkout master
- git merge v1
- git push

總結

1.第一次初始

在Github開repo

`git clone [clone URL]` 先把code從Github抓下來

2.

進到該目錄中

`git branch [branch-name]` 生一個新branch

`git checkout [branch-name]` 切換到此branch上

3.

`git status` 看修改了哪些檔案

`git add [修改到的檔案]` 一個檔一個檔自己加，因為這樣可以Double check自己到底改了哪些檔

`git commit` 註解要寫清楚 完成一個小功能就要commit一次

重複步驟3.直到 **git status** 顯示沒有任何修改過的檔案

4.

`git checkout master`

`git merge [branch-name]` 和master merge起來

`git push` 然後接著輸入Github的帳號密碼

Git flow



Demo & 實作



軟工無用論，程式會動才是王道？

問題整理

- LeetCode刷題是必要的嗎？
- 該怎麼準備，拿到第一份offer
- 公職vs業界vs升學
- 我什麼都不會，我真的能把程式寫好嗎？
- 該學哪一個程式語言？

感謝聆聽