

An introduction to Software Testing

The state of art
software engineering practice
nowadays

The Nature of Software Testing

我們搞軟體的品質問題與其他工程領域比較起來如何？



A Joke

有天晚上我參加一個老同學的喜宴，
同桌的人有律師、會計師，
只有我一個是電腦業界的人。

為了引起話題，我說我羨慕其他行業的人，
像是律師、會計師等，在學校學的那一套
終生都受用，每年就算有新法條也不會改
太多。

不像我們電腦界的人，每天都在K新的技術手
冊，新產品新技術隨時大翻新，改朝換代
的速度讓人措手不及，所以學電腦的人看
起來都比較蒼老，因為太辛苦了。

我話剛說完，一位會計師馬上回我一句：
「你錯了，其實我們最羨慕你們電腦界的
人，因為沒有任何行業的消費者像你們電
腦界的消費者一樣好欺負。」

頓時我成了眾矢之的，
這些電腦使用者的抱怨全部集中到我身上
來。
其中一位仁兄還舉了個有趣的例子，

他說：

「如果傢俱業和電腦業一樣，世界會變成
什麼樣子？」以下是他講的故事：

如果傢俱業跟電腦業一樣，比如說我到傢
俱店買了一張桌子，搬回家往地板上一放，
啪啦一聲桌子就塌了。這時候我不會生氣、
不會罵人，我會先自己檢查一下出了什麼
錯。

我會先檢討自己，
是不是我做錯了什麼，
或是我對桌子的使用不夠熟悉。

於是我會去買書來看
(書名可能是《快快樂樂學修桌子》、
《21天學會修桌子》、《修桌子技巧與實例》、
或是《修桌子的聖經》)。

要是書看得不太懂，
我會再花錢去報名上修桌子的課程。

學完之後還是修不好，
我會請其他比較懂得修桌子的朋友來幫忙。

最後沒有辦法，
終於我打電話給原先的傢俱行，
(可能還要購買《技術支援方案》)，
結果他們跟我說：
「唉呀！你買到的是搶鮮版啦！
本來就應該有問題的」。

於是我恍然大悟原來是自己的錯，
我就再去買一張「正式版」的桌子。

回家一擺還是啪啦又塌了！

修了半天還是有問題，
再請傢俱行的技術人員來做仔細檢查，
最後終於發現問題的所在——
「我家的地板和桌子不相容」，
又是我自己的錯，
於是我得趕快幫家裡的地板升級.....。

等一切都忙完了，
桌子可以使用了，
我趴在桌上寫字，
心裡充滿了成就感，
我很得意地跟網友分享我修理桌子的經驗，
並暗自慶幸自己在科技的潮流上沒有落伍.....。

There was an ACM Survey sent to thousands of software companies

What a software engineer should learn in school?

The importance of testing

- Seldom recognized by academic
- Professors don't care
- students don't care
- but ACM survey more than 1000 software companies. Three topics are suggested to be important to CS students
 - Teamwork
 - Software Engineering
 - **Software Testing**

Why testing?

- Let's look at some well-known software defects
 - 台灣高鐵售票系統：系統上線之後當機連連，系統無法應付突然湧現的購票人潮（no stress testing）
 - 台灣彩卷系統：類似的問題
 - 2000-2005- 巴拿馬國家癌症中心—5個病人接受過量的迦瑪射線照射死亡。15人引發了嚴重的併發症
 - 2003—軟體造成美國東北部及加拿大停電。5000萬人受影響，3人喪生
 - 2000 美國海軍飛機墜落，4人喪生（控制軟體問題）
 - 1997 韓國空難，225人喪生（雷達控制軟體問題）
 - 1995 美國航空在哥倫比亞撞山159人喪生（導航軟體問題）

Statistics

- 2002- 美國國家標準局報告—軟體品質每年造成595億美元（0.6 % GDP）
- 國內又如何？
 - 台北縣政府校務行政系統案例
 - 台師大林口校區學務系統案例

Quality Control and Quality Assurance

- Software is a product.
- Good quality product requires 品質控管 quality control (QC) and quality assurance (QA) 品質確保

我們來看看其他產業是怎麼進行品質控管與改良的

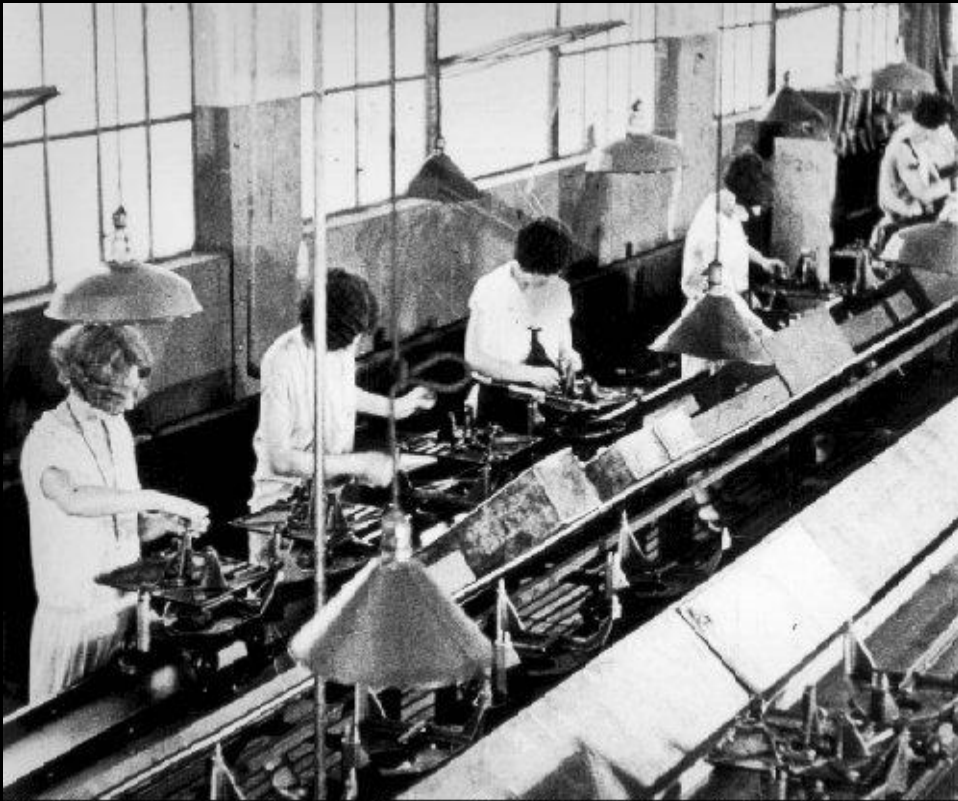
Manufacturing in other discipline

- 在製造業的世界裡面
 - 產品品質(quality)是很重要的一個因素
 - 當你投入非常多成本與原物料時你永遠希望你生產出來100個產品有100個都是可以賣的
- QC (Quality Control) 品質控管
 - 製造過程從設計到生產可能有數十道到幾百道程序
 - 每一道程序都可能影響到後面的品質
 - 如何透過製造流程的改善(process improvement)來提高良率，一直都是製造工業的重要課題

How to improve quality?

- 在製造過程中如果能夠能夠使用機器，最好的改善品質的方式就是自動化。機器不容易犯錯而且可以不斷地重複單調無聊的工作。機器會出錯的時候通常是由於製造機器的磨損，必須重新校正。
 - Ex. 日本的步進馬達精確度超高的秘密。
- Question
 - 當某部分的工作不能由機器來做的時候，製造過程如何做到品質控管？

Assembly line in manufacturing



U.S. DEPARTMENT OF THE INTERIOR, NATIONAL PARK SERVICE, EDISON NATIONAL HISTORIC SITE



生產線

- 每個生產線員工只做一件夠簡單到不容易出錯的工作
- 生產線員工不需要夠聰明
- 生產線將複雜的產品製造切割成許多小步驟可以在短時間以及最少的技術內可以完成
- 發現產品的問題並不難.

你能如法炮製嗎？

The answer is: NO, you cannot, software developing is a design job. If it is so, you will not be so expensive.

Programmers v.s. Assembly line worker

- Programming is a job which requires perfection !! Humans are not designed to complete perfect jobs
- Programmers must deal with large complexity and details
- Humans are not designed to complete perfect jobs (we are a product based on "Natural Selection" & "Survival of the fittest")



Th



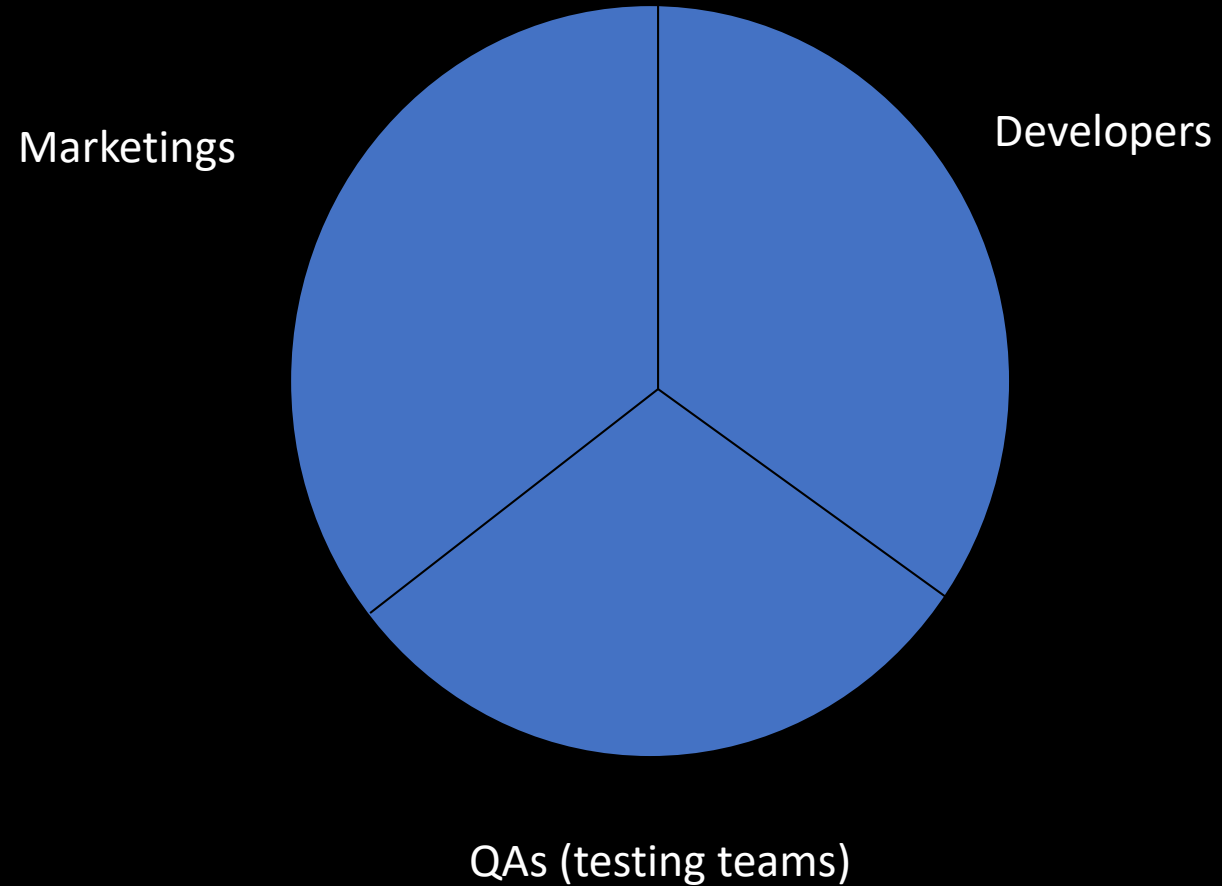
TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

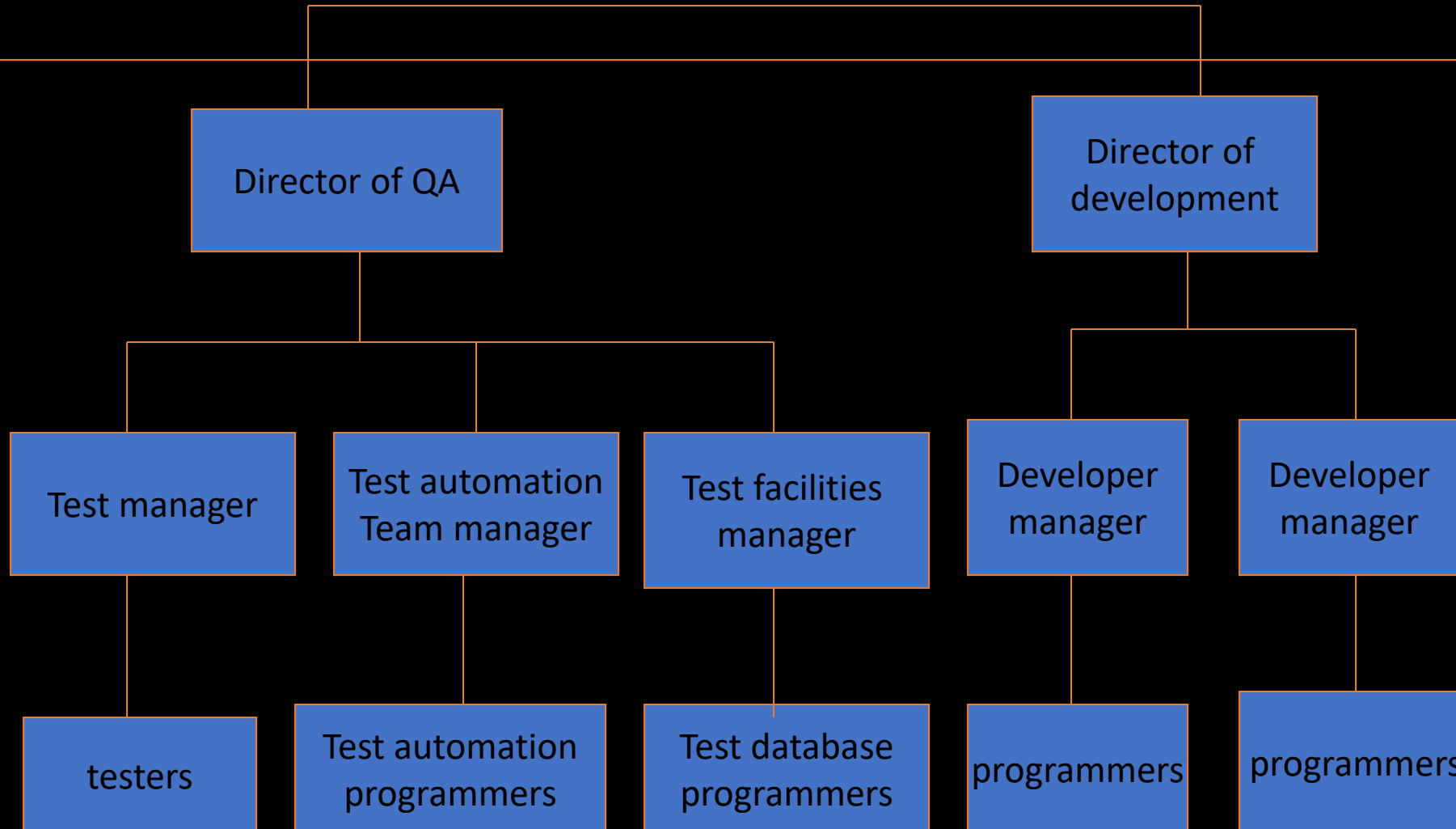
Traditional Software Testing?



Company Organization



Management Hierarchy



Important reasons to make QA team and developer team equal and separated

- 只有極少數處女座的 programmers 才會有品質（完美主義）的概念
- 叫 programmer 寫完程式，自己找bugs 就如同叫法官判完案之後，承認自己的判案是錯誤的
 - 如果他真的找出很多bug，他決不敢聲張與邀功，因為這代表自己程式寫太爛
 - 找的bug越多，表示待完成的工作越多，沒有人會自己找碴，讓自己無法喘息
- 上述「人」的因素使得
 - QA 必須在軟體品質上與 programmer 站在對立面
 - QA 的升遷管道必須是抓到越多 bug，功勞越高，QA 才會努力找碴
 - QA 不能歸 developer team 管轄，這就像是司法不能在行政部門底下是一樣的



Marketing

恐怖平衡



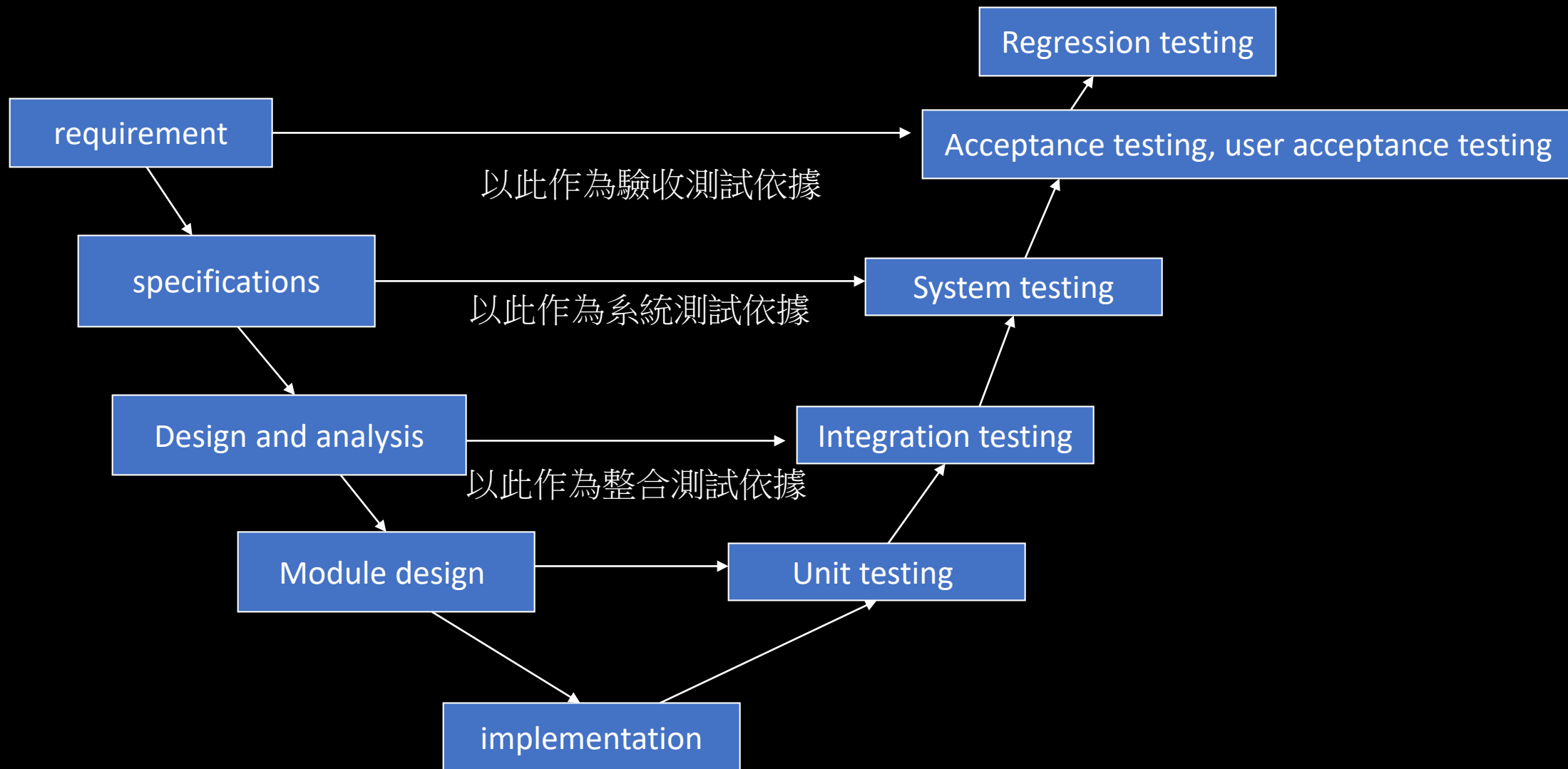
QA (testers)



Developer



Software Testing V model



SDLC (Software Development Life Cycle)

- Requirement analysis (marketing research)
- Software function/performance specification (QAs should join the project here to write a test plan)
- Analysis and Design 分析
- Coding and unit testing (by programmers)
- Integration testing (by programmers/QAs)
- **Alpha testing (by QAs – a set of test cases is prepared)**
 - Most software functions and features are basically completed
 - All functions are tested, no functions will be added beyond this point
 - Serious flaws (high severity) are solved and addressed (show stopper)
- **Beta testing (by Beta users)**
 - sub-serious bugs are all fixed
 - Test plan has been completely executed
 - Bug discovering rate is lower than bug fixing rate
- **Release**
 - Bug discovering rate is lower than bug fixing rate for a long period of time.
 - The version after fixing bugs has been regressively tested (regression test)
 - Quality is formally proved by QA team
 - All documents are ready

Testing tasks to be done

1. **Write Test plan (often seen in large software company with formal QA team)**
2. write test cases, prepare test data
3. test case management
4. file bugs
5. regression testing
6. test automation (with tools)
7. stress testing/load testing
8. security testing
9. test as assets.

Test Plan?

- There are many test plan template
- The major contents in a test plan
 - what to test?
 - what not to test?
 - estimate the cost and efforts of testing
 - prioritize testing tasks
 - how test data should be prepared.
 - testability
 - testing schedule
- via review, developers and testers can communicate and check
 - if specs are understood mutually (communication is not as intuitive as you think)
 - raise any testing issues while studying the system specs

NOTES

- QA occurs at the end of development
- Programmers are only responsible for releasing a workable build to QA
- QA test for a period of time, called Alpha testing

No test cases, no quality !!!

The problem continues.....

Are the test cases in your brain or in some places systematically maintained.

No test cases, no quality !!!

The problem continues.....

Are the test cases in your brain or in some places systematically maintained.

No test cases, no quality !!!

(講三遍，因為很重要)

The problem continues.....

Are the test cases in your brain or in some places systematically maintained.

Test Case 怎麼做?
In your brain?



TestID

T-VM-PLUG-0031A-Test plug-in : Minerva plug-in

Revision History

2013/01/21 created

Summary

利用視覺化一個cube_ubvm來測試Minerva plug-in是否能與xDIVA正常溝通

Tested Story**Creator**

Zoe

Test Steps

1. open the ubvm.java with the Minerva
2. Set break points at line 15 // ivar = 100
3. Run xDIVA
4. click the Debug menu -> Visualize ivar
5. When the mapping dialog pops up, do mapping



1. Place a cube_ubvm (select tab "ubvm")
2. Set the attribute of cube: select the tab "Input" on the right, check the sizex to make sizex visible
3. Link ivar to ports "_vm_name", _vm_sizex and _vm_clickmsg
4. Press "Apply"

6. -> cube sizex is 50



A REAL TEST CASE SAMPLE

Test Case ID example

- T-SSSS-CCCCC-XXX-DDDDD TITLE
- SSSS: Test Suite name
- CCCCC: Component or Requirement abbreviation name –
 - by this name, a tester can quickly grasp what this test case is designed for
 - it can be a component or requirement abbreviation names.
- XXX: X 的值可以是
 - 'A' the test case be automatically executed
 - 'M' the test case must be manually executed
 - 'P' this is a positive case
 - 'N' this is negative case
 - 'B' this is a boundary case
 - '0' test case priority (smoke test case can be 0)
 - '1'..'9'
- DDDD: test case ID in a test suite. ID is not necessary be an integer. If it is an integer like “0008”, an alternative 0008A means a similar test cases.
- TITLE: 50 words that can be easily displayed in a test case management tools.
-

Testing tasks to be done

1. Write Test plan (often seen in large software company with formal QA team)
2. write test cases, prepare test data
3. **test case management**
4. file bugs
5. regression testing
6. test automation (with tools)
7. stress testing/load testing
8. security testing
9. test as assets.

Test Case Management example using redmine

✓ #	Tracker	Status	Priority	Subject	Assignee	Updated	test result
<input type="checkbox"/> 985	test case	New	Normal	T-VM-LAYOT-0033-Varname : used in layout	琨宜 何	01/21/2013 12:38 pm	passed
<input type="checkbox"/> 984	test case	New	Normal	T-VM-PLUG-0032B-Test to visualize two variables with difference plug-in : Eclipse plug-in		01/21/2013 12:35 pm	none
<input type="checkbox"/> 983	test case	New	Normal	T-VM-PLUG-0032A-Test to visualize two variables with difference plug-in : Minerva plug-in		01/21/2013 12:32 pm	none
<input type="checkbox"/> 982	test case	New	Normal	T-VM-PLUG-0032-Test to visualize two variables with difference plug-in : VS plug-in		01/21/2013 12:27 pm	none
<input type="checkbox"/> 981	test case	New	Normal	T-VM-PLUG-0031B-Test plug-in : Eclipse plug-in	Ricky Chien	01/21/2013 12:19 pm	passed
<input type="checkbox"/> 980	test case	New	Normal	T-VM-PLUG-0031A-Test plug-in : Minerva plug-in	琨宜 何	01/21/2013 12:18 pm	passed
<input type="checkbox"/> 979	test case	New	Normal	T-VM-PLUG-0031-Test plug-in : VS plug-in	冠華 丁	01/21/2013 12:16 pm	passed
<input type="checkbox"/> 978	test case	New	Normal	T-VM-VP-0030-animation : by visual point		01/21/2013 12:11 pm	none
<input type="checkbox"/> 977	test case	New	Normal	T-VM-LAYOT-0029-Layout : attribute in the container		01/21/2013 12:04 pm	none
<input type="checkbox"/> 976	test case	New	Normal	T-VM-VP-0028A- visual point : when the variable doesn't exist		01/21/2013 11:57 am	none
<input type="checkbox"/> 975	test case	New	Normal	T-VM-VP-0028- visual point : a ball change with its radius		01/21/2013 11:52 am	none
<input type="checkbox"/> 974	test case	New	Normal	T-VM-TM-0027B- Array : size 0 array		01/21/2013 11:42 am	none
<input type="checkbox"/> 973	test case	New	Normal	T-VM-TM-0027A- Array : array unfold		01/21/2013 11:36 am	none
<input type="checkbox"/> 972	test case	New	Normal	T-VM-TM-0027- Array : array type mapping		01/21/2013 11:28 am	none
<input type="checkbox"/> 971	test case	New	Normal	T-VM-TM-0026D-Test unfolding : Can't unfold when pointer is null	冠華 丁	01/21/2013 11:22 am	passed
<input type="checkbox"/> 970	test case	New	Normal	T-VM-TM-0026C-Test unfolding : Can't unfold when the type is loop reference	新志 陳	01/21/2013 11:17 am	none
<input type="checkbox"/> 968	test case	New	Normal	T-VM-TM-0026B-Test unfolding : Can't unfold when the type is the same as class type	Ricky Chien	01/21/2013 10:44 am	passed
<input type="checkbox"/> 967	test case	New	Normal	T-VM-TM-0026A-Test unfolding : Can't unfold when the type is doing mapping	琨宜 何	01/21/2013 10:40 am	passed
<input type="checkbox"/> 966	test case	New	Normal	T-VM-TM-0026-Test unfolding : binary tree	冠華 丁	01/21/2013 10:36 am	passed
<input type="checkbox"/> 965	test case	New	Normal	T-VM-TM-0025E-Test type mapping dialog : ref type variable only connect to reference type VM	新志 陳	01/21/2013 10:24 am	passed
<input type="checkbox"/> 964	test case	New	Normal	T-VM-TM-0025D-Test type mapping dialog : Type mapping should has single root	Ricky Chien	01/21/2013 10:13 am	failed
<input type="checkbox"/> 963	test case	New	Normal	T-VM-TM-0025C-Test type mapping dialog : TMD doesn't have the "apply all"		01/21/2013 10:03 am	none
<input type="checkbox"/> 962	test case	New	Normal	T-VM-TM-0025B-Test type mapping dialog : TMD input port has default gates	冠華 丁	01/21/2013 10:00 am	passed
<input type="checkbox"/> 961	test case	New	Normal	T-VM-TM-0025A-Test type mapping dialog : save a binary tree type node and import to toolbar	新志 陳	01/21/2013 09:56 am	failed
<input type="checkbox"/> 960	test case	New	Normal	T-VM-TM-0025-Test type mapping : binary tree	Ricky Chien	01/21/2013 09:48 am	passed

Programmers 通常不喜歡寫 test cases 認為那是 QA 的事情
(我寫程式都快煩死了，還寫 test case?)

擁抱新世界與最新發展

- 寫測試是你(programmer)的事情
- 你可能還得把測試變成自動化 (unit tests)
- 全世界的主要公司 QA 的職缺都在萎縮
- 那品質誰把關? YES, you as a programmer
- So, yes, you need to understand the principle of testing
- 我還是認為 QA 的工作不會消失，但是他的工作內容會更複雜，尤其是完成 end to end (e2e) test automation

測了就會有 bug

Bug report 是一個關鍵的軟體工程實踐
(可以觀察你的產品是否是玩真的)



Testing tasks to be done

1. Write Test plan (often seen in large software company with formal QA team)
2. write test cases, prepare test data
3. test case management
4. **file bugs in bug report system**
5. regression testing
6. test automation (with tools)
7. stress testing/load testing
8. security testing
9. test as assets.

軟體開發現場

- Hello, 我這邊有個 bug (.....) 你幫我修一下
- 修完了，你幫我測試一下
- 可以動了 !!
- 好的

軟體開發HORROR現場

- A: Hello, 我這邊有個 bug (.....) 你幫我修一下
 - 你的 bug 有可能是其他人的 bug 或關聯，最保險的策略是 let everything is traceable。
- B: 修完了，你幫我測試一下
 - 你怎麼知道他的測試會是足夠？會不會連動地影響到其他功能或是效能？
 - 軟體工程 is all about communication。
- A: 可以動了 !!
- B: 好的.....
 - 這個修正可能永遠存在於你們兩個人之間，從你的 source code update 看起來可能完全不知道在 A 與 B 之間發生了什麼事情。
 - 你可以把 A, B 抓來問，他們一定都會跟你說，還好啊？就修了啊
 - 你可以這麼做，每一個 source commit about bug fix 一定要關聯一個 bug issue 否則 code review 都不會過
 - Quality problem 可能造成歸咎問責問題，我如果是你，一定乖乖把每個 bug 記錄下來







File a bug in a modern wiki bug report system

Create New Ticket

Properties

Summary:

Reporter:

Description: **B I A**      

Hardware: x86
OS: Window 7
1. run calc.exe
2. press 5
3. press +
4. press 5
5. press =
6. result should be 10, but show wrong answer
[[Image(calc.jpg)]]

Type:

Milestone:

Version:

Cc:

Assign to:

Priority:

Component:

Keywords:

☒ I have files to attach to this ticket

Ticket #1 (new defect)

The result is in correct

Opened **6 minutes** ago

Reported by:	ypc	Owned by:	student01
Priority:	critical	Milestone:	M1 - Obama
Component:	component1	Version:	2.0
Keywords:		Cc:	

Description

Hardware: x86
 OS: Window 7
 1. run calc.exe
 2. press 5
 3. press +
 4. press 5
 5. press =
 6. result should be 10, but show wrong answer

[Reply](#)

A sample bug report



Attach any Files that Concern the Bug

- Files to attach include
 - input file/data
 - output file/data
 - Log files
 - Core dump (memory dump)
 - Error log files
 - Tracing information (generated by Tracing tools)
 -
- A test report should provide any information as much as possible.

寫下 bug report 的操作順序很煩，Can we do it in better way?

GitLab Projects Groups More

Search or jump to...

xdiva2.0-portal-center

Project overview

Repository

Issues 209

List

Boards

Labels

Service Desk

Milestones

Merge requests 0

CI/CD

Security & Compliance

Operations

Packages & Registries

Analytics

Wiki

Members

Settings

xDiva > xdiva2.0-portal-center > Issues > #265

Open Created 2 months ago by Yung-Pin Cheng Developer Close issue

#264之後，uninstall後重啟電腦，重新install xDiva跟vscode套件，照正常操作卻仍沒有物件顯示

版本、log、操作影片

version 8012_3

typemappingdialog.log

2021-08-14_14-24-35.mkv

自從#264之後，照正常程序走Viewer都不會有任何東西顯示，vscode裡也沒有成功的提示

GDB-plugin status: **connected** connect

Viewer status: **connected** connect

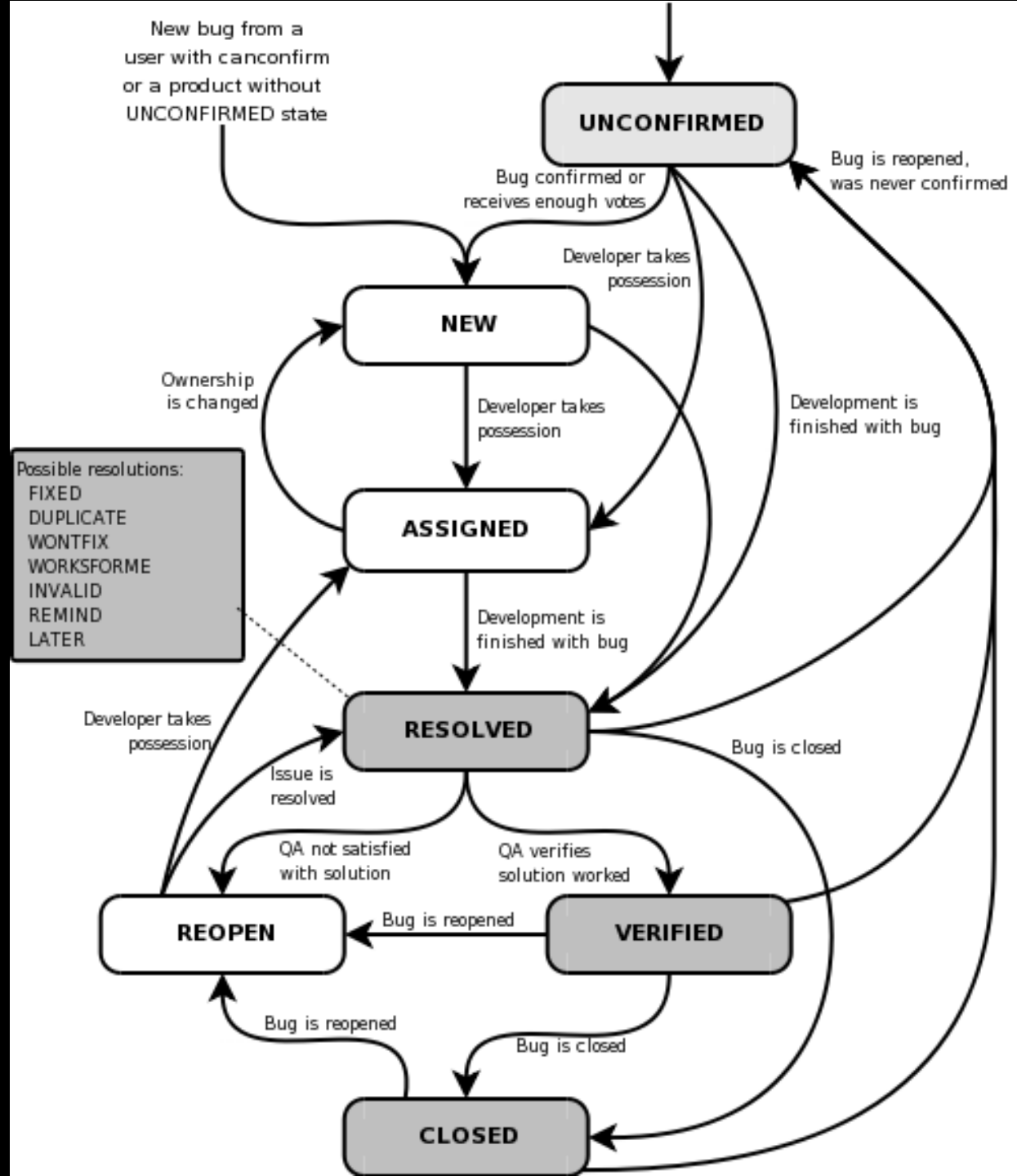
有確認都有connect到

a

uninstall xDiva跟vscode 套件，重新開機再次安裝也沒有出現任何東西

利用 Screen casting
tool (OBS) 對螢幕錄
音錄影

- bug life cycle



Severity of Software Defects

- show stopper (severity 1)
- unusable software (severity 1)
- Microsoft ranks the bug severity from 1-30
- killing people, waste a lot of money.

嚇人的問題來啦
QA的工作跑到我身上了
尤其是 regression testing
正在發生的大趨勢 DEVOP



Testing tasks to be done

1. write Test plan (often seen in large software company with formal QA team)
2. write test cases, prepare test data
3. test case management
4. file bugs in bug report system
- 5. regression testing & Test Automation (with tools)**
6. stress testing/load testing
7. security testing
8. test as assets.

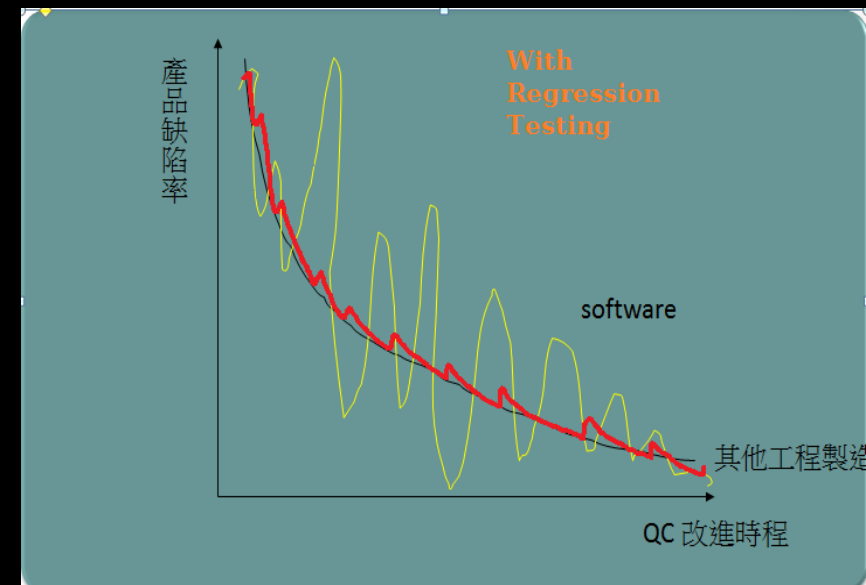
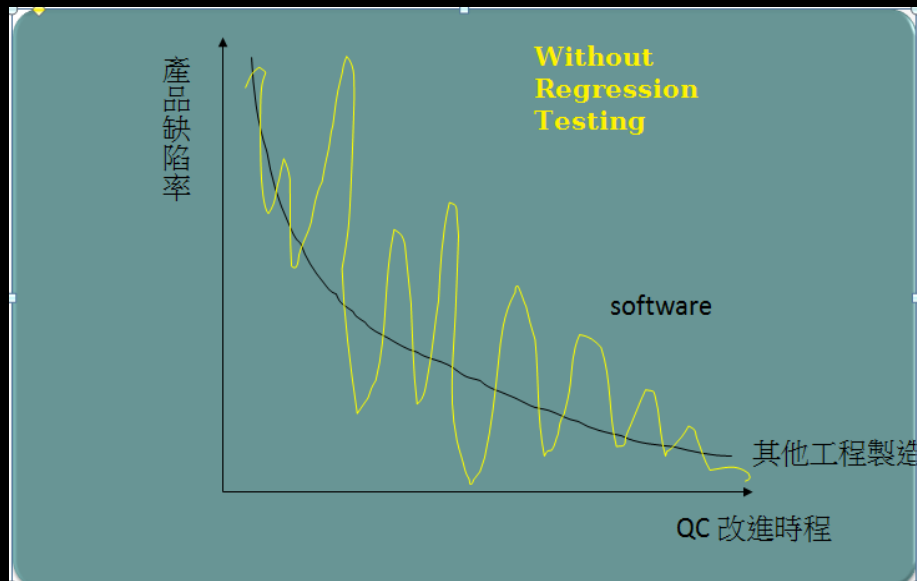
Regression Testing

Regression:
"when you fix one bug, you
introduce several newer bugs."

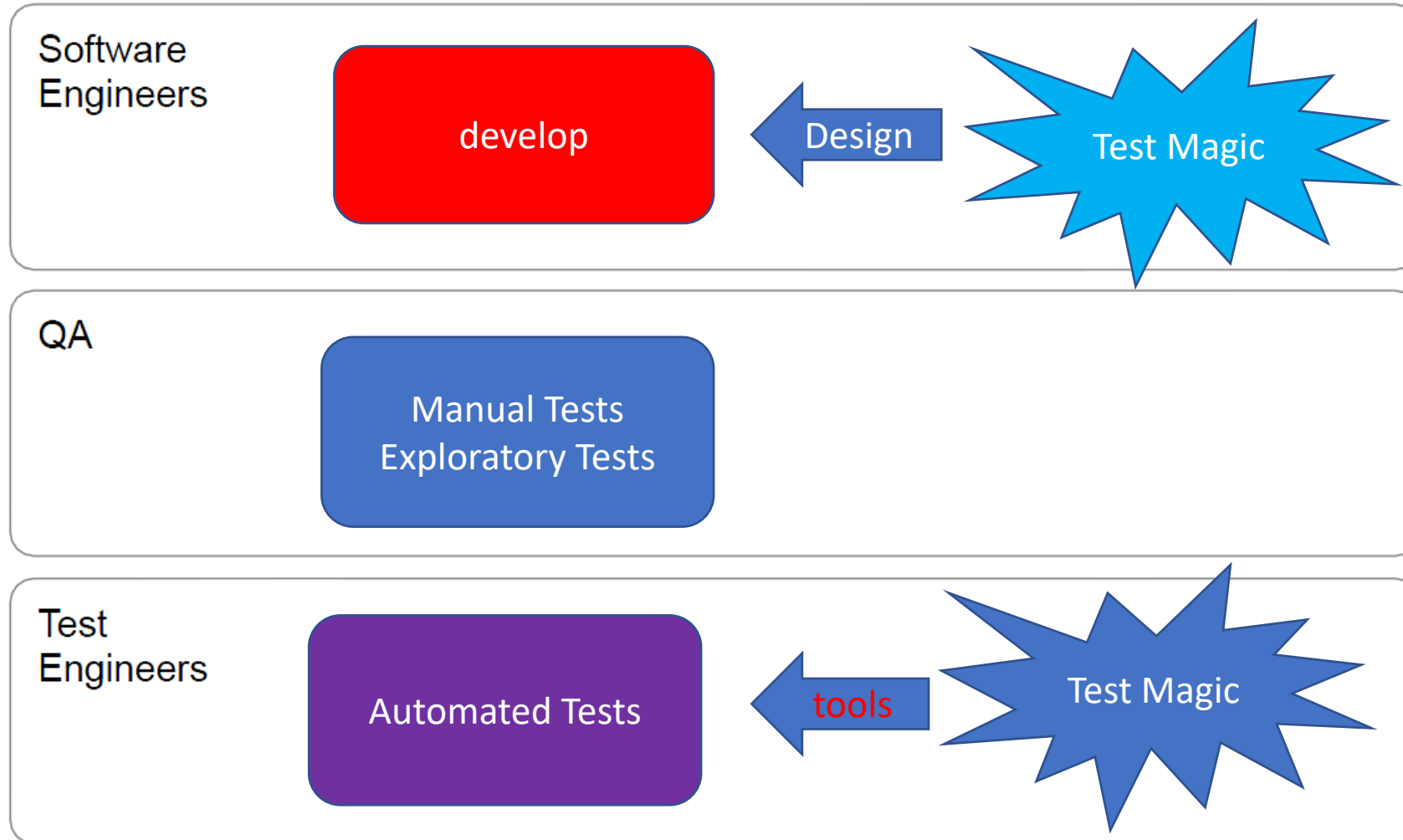


Regression Testing

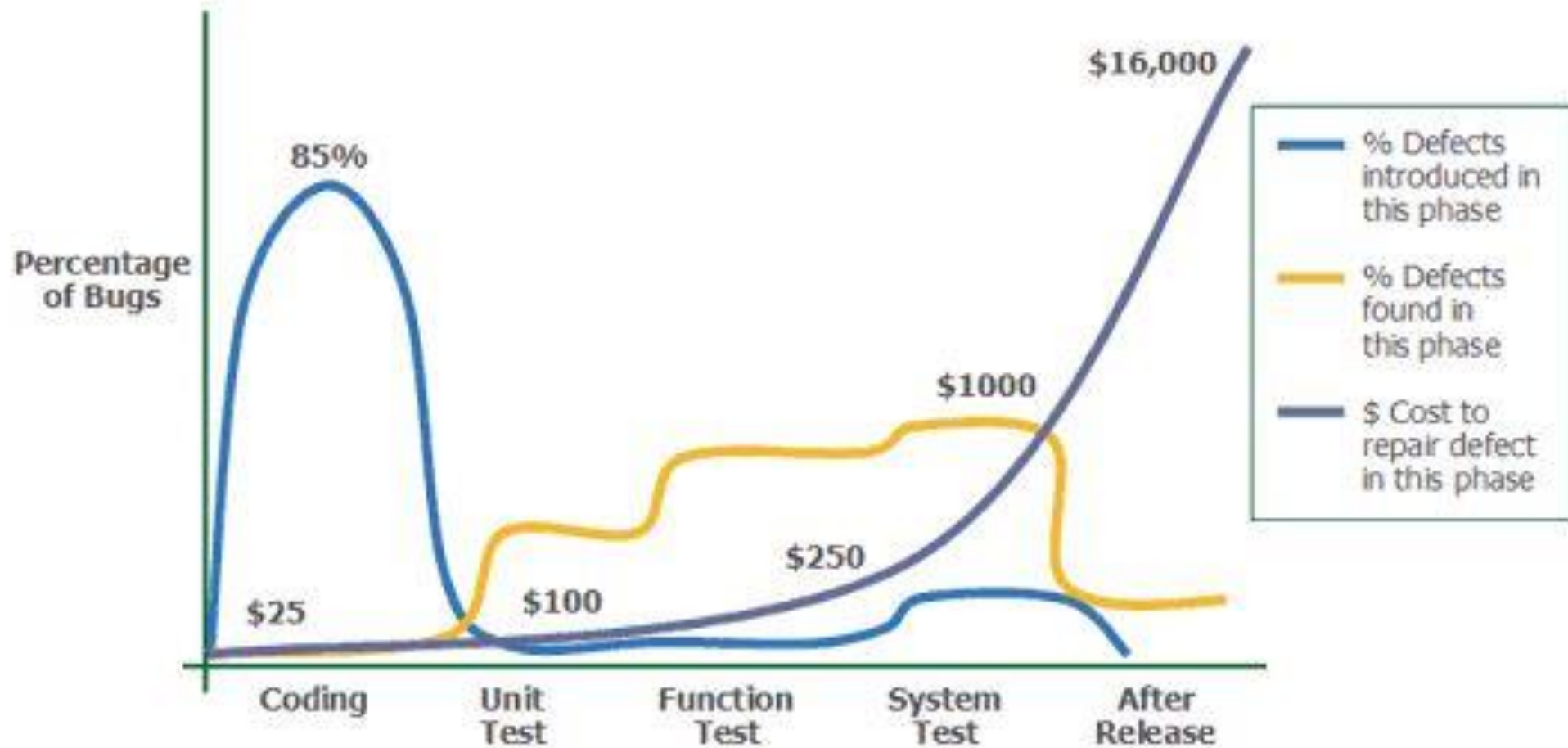
- Rerun all the test cases when the program is modified, extended, or revised.
- Automation could be hard depending on applications



Development Model



The cost of finding a bug – an evaluation



Source: *Applied Software Measurement*, Capers Jones, 1996

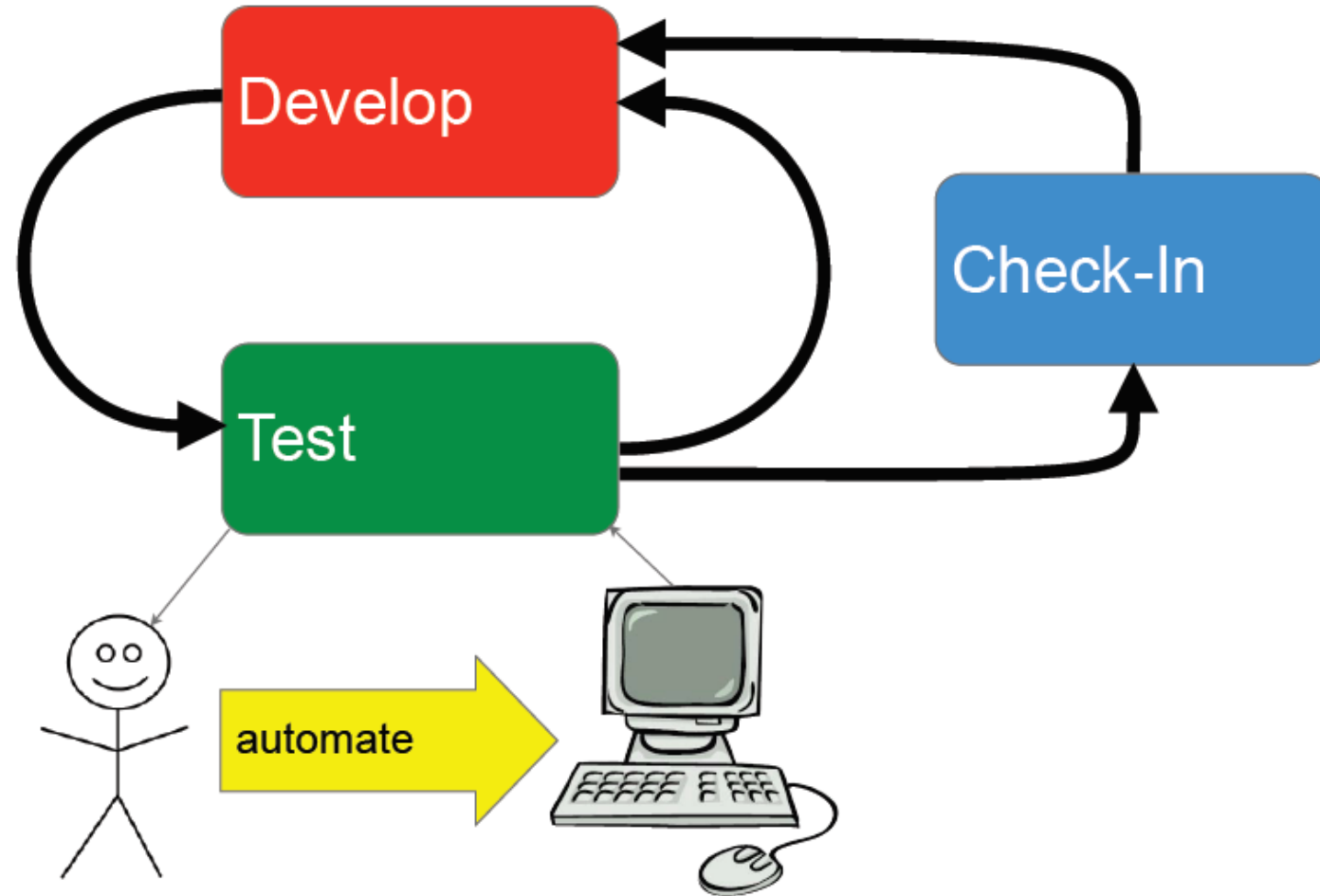
其實絕大部分的工程領域早就
知道這個道理，就是我們一直
後知後覺

一個簡單到爆(但是我們搞軟體的人都不懂)的原理

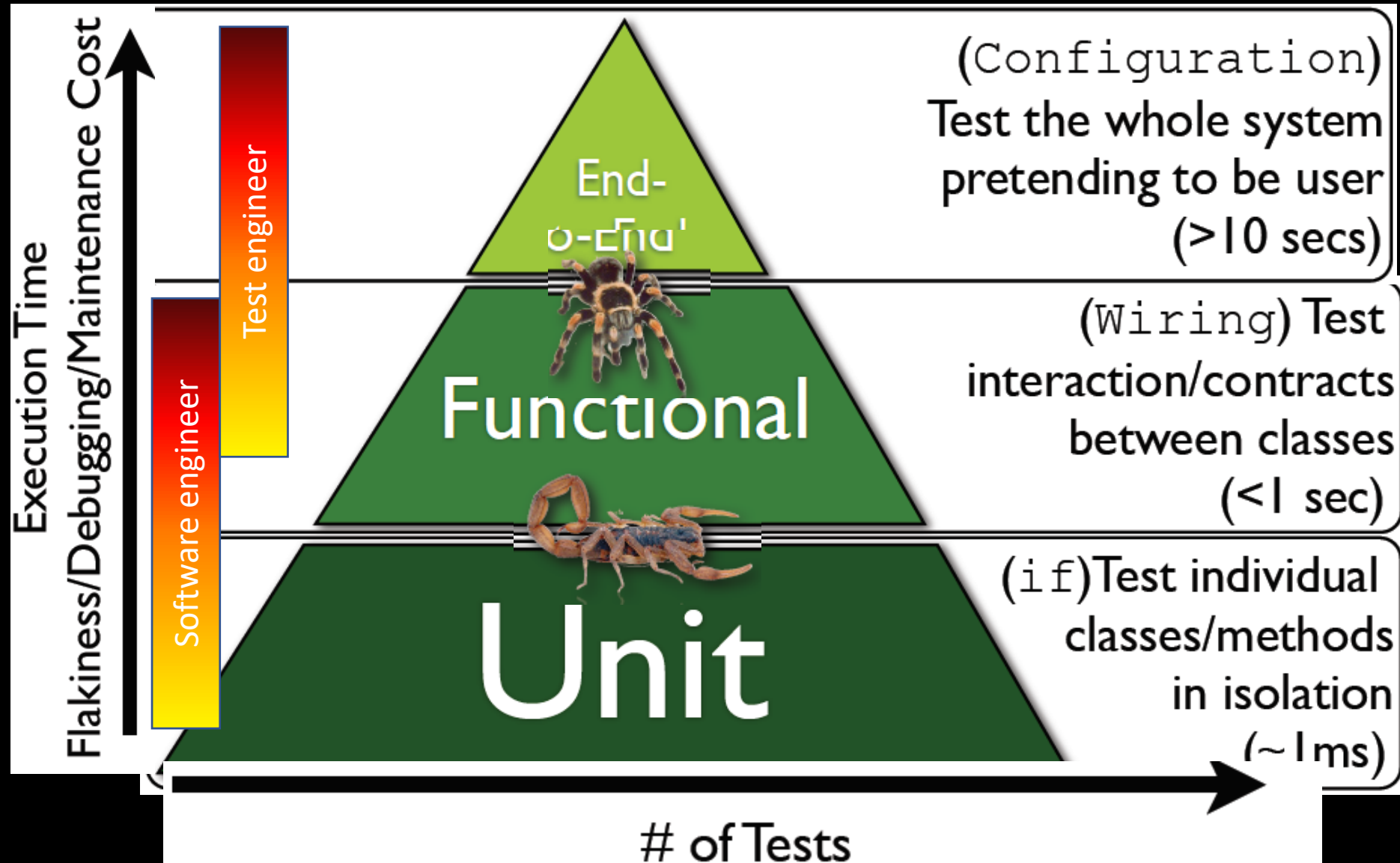
- 假設你今天要生產一個產品，由100個零件組成
- 如果設計沒有問題，你認為最終品質的改善來自於
 - 組裝程序?
 - 零件品質?
- 如果是零件品質? 你怎麼做?



Micro Development Model



Different Kind of Testing



Dev Model Revisited

Education

Testing on the Toilet
Tech Talks / Blogs
1 on 1 Training
Mercenaries
Immersion

Enforcement

RoboCop
Testability
Coverage

Develop

CI-build

Check-In

Test

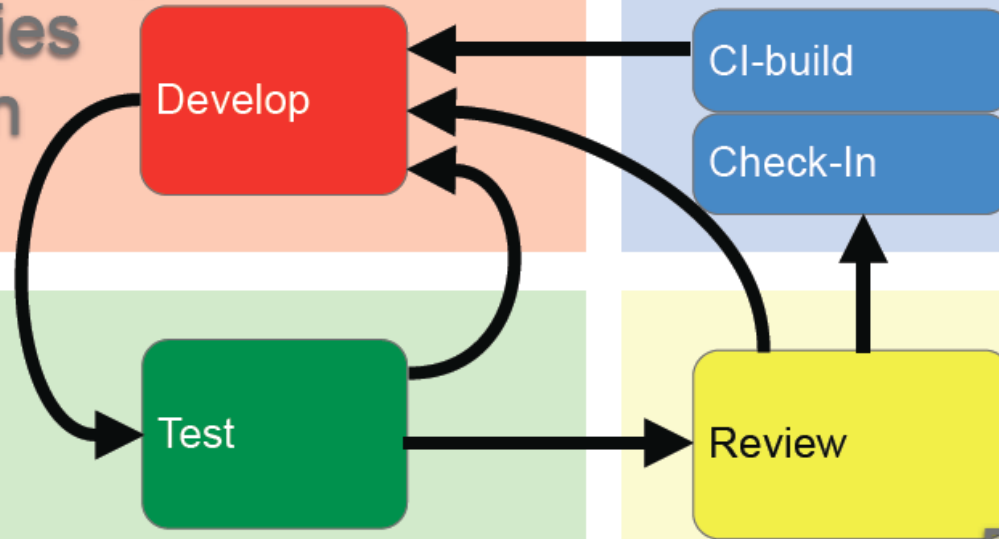
Review

Tools
Analysis
Reports

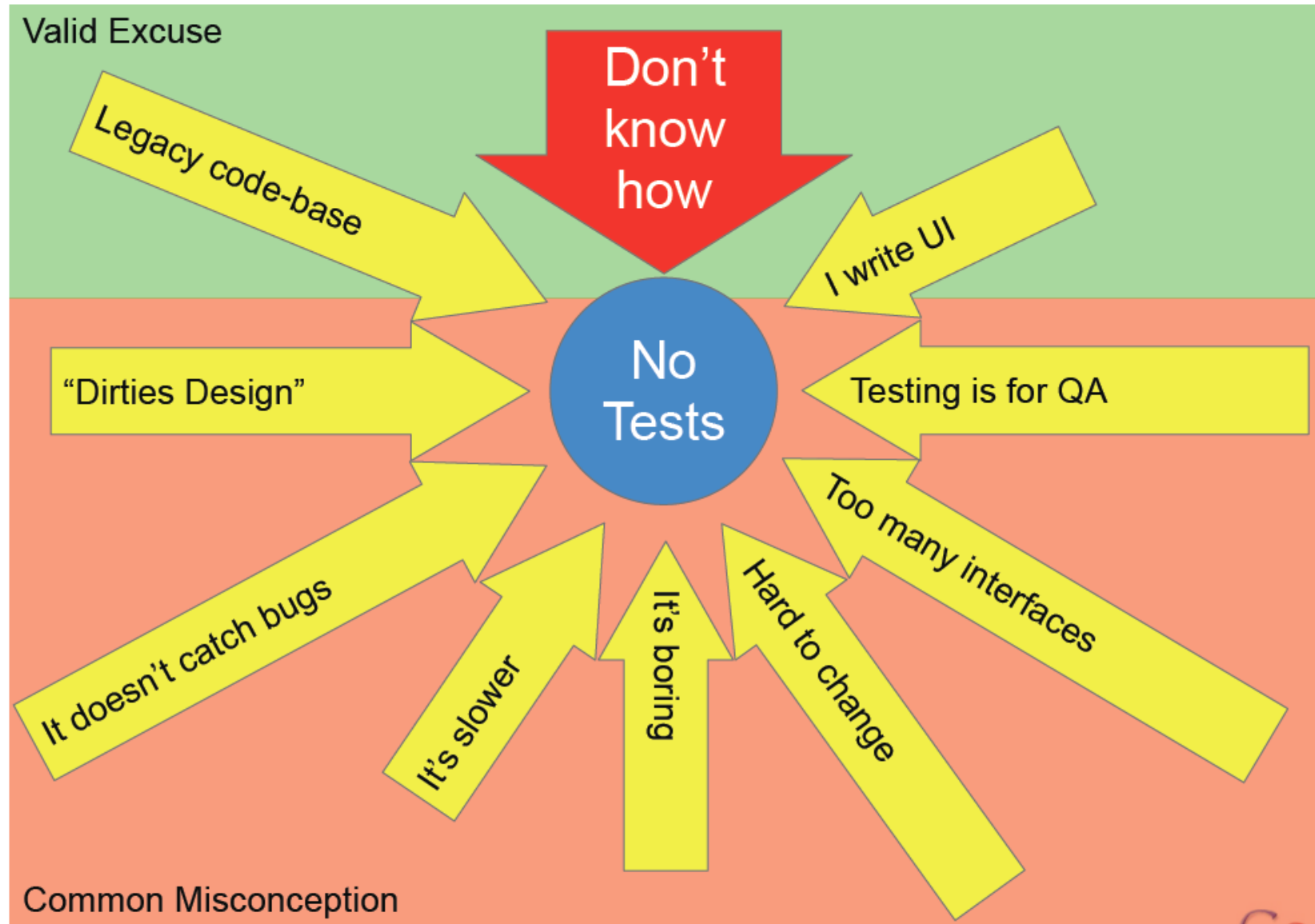
Games
Dashboards
Scoreboards
Trend Graphs

Tools

Visibility



Excuses



The COST and Difficulty of building test automation in practice

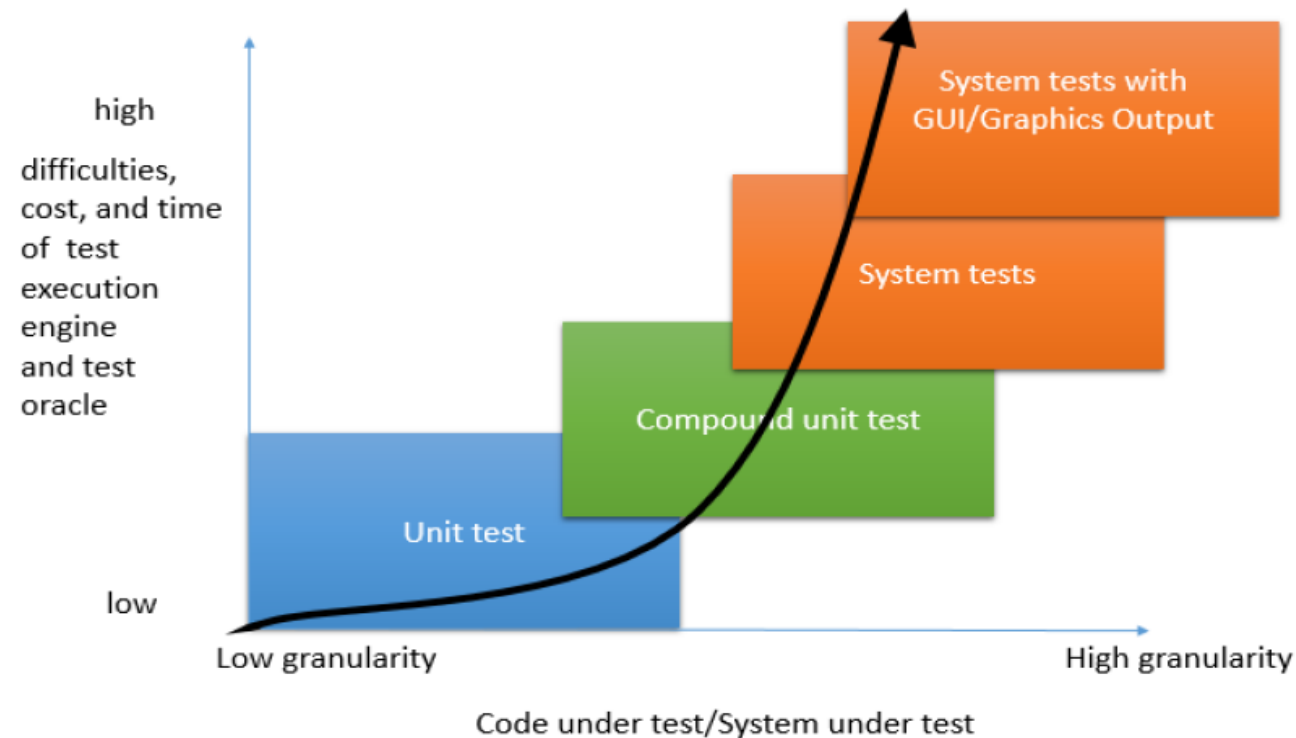


Fig. 1. The technical difficulties of test execution engine and test oracles as the granularity of code under test grows.