Lessons Learnt From TA Practices

Xiao Shiliang-Shelwin (肖世良)

2017.05.25

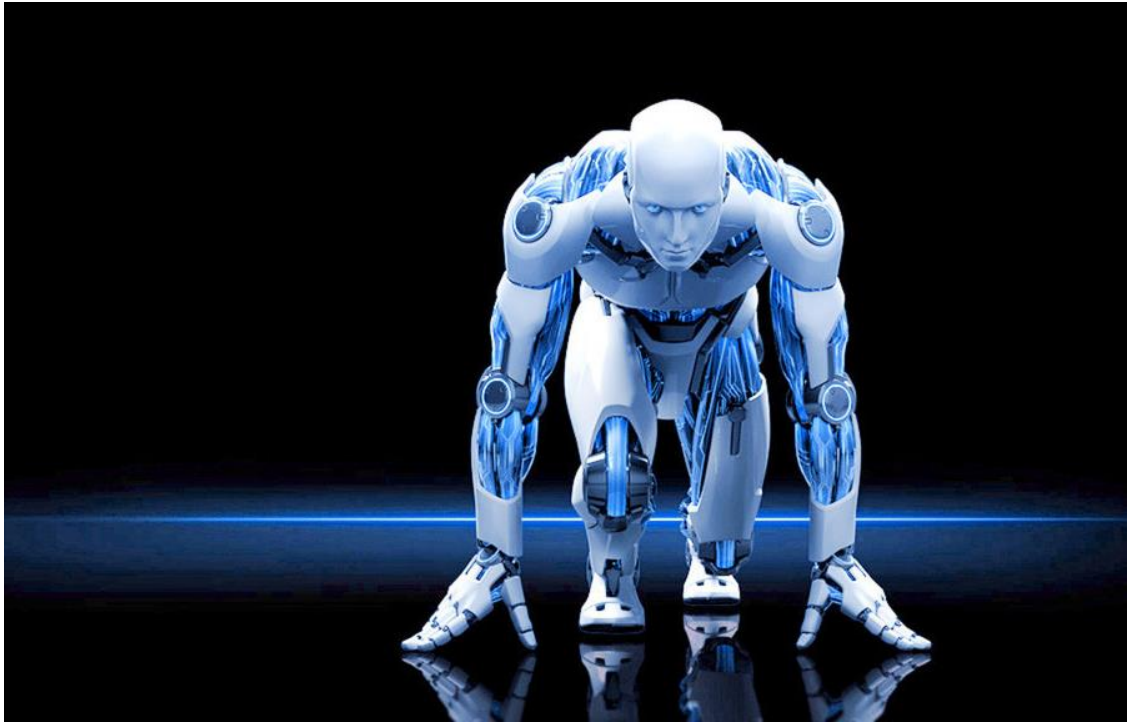
TA: Test Automation

“Use of special software to control the execution of software tests and the comparison of actual outcomes with predicted outcomes”

—— Wiki

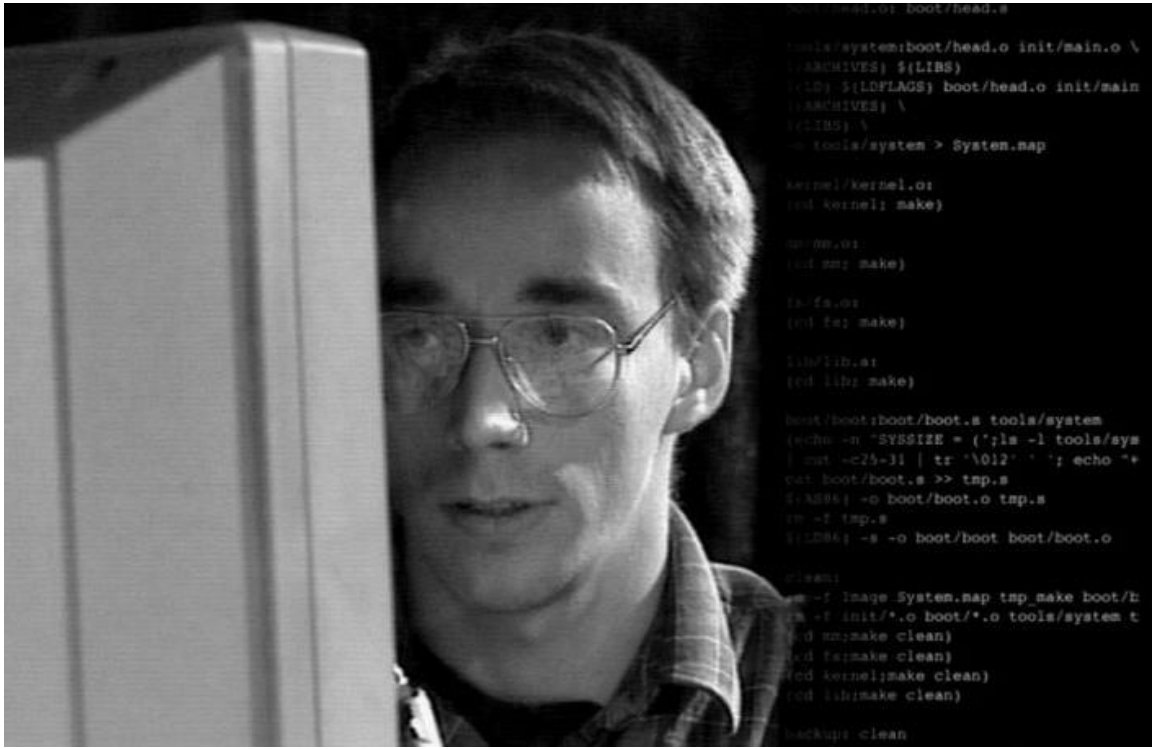
“Let computer do software testing for human”

—— Somebody



Essence of Test Automation

Test Automation is
Software Development

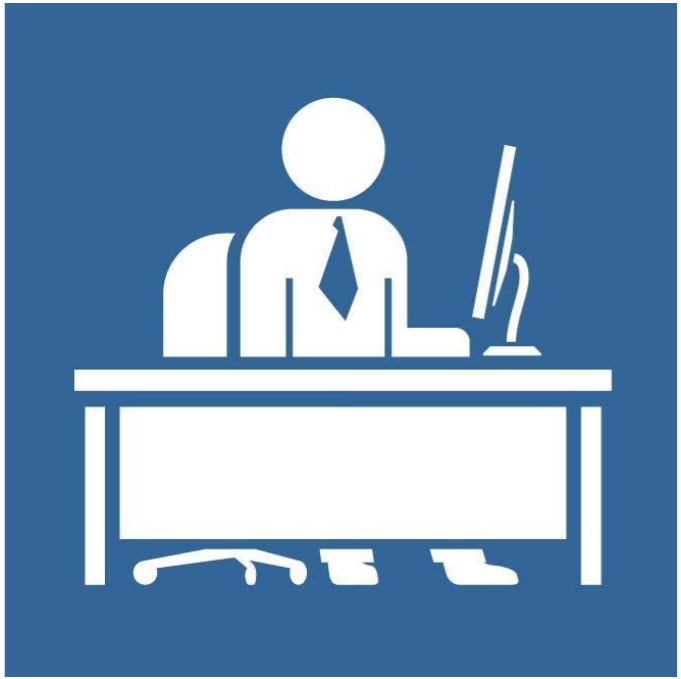


Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

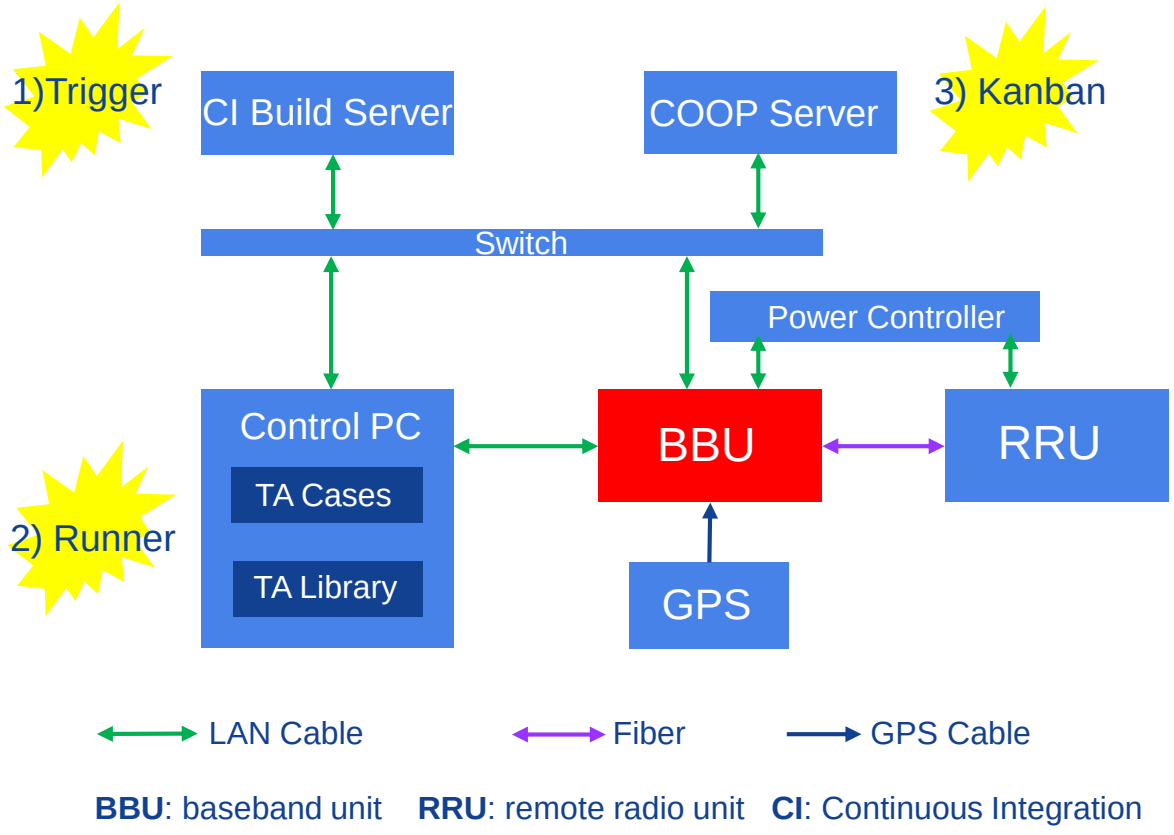
- Lesson: What a good TA is
- Lesson: How to achieve a good TA

TA Practice



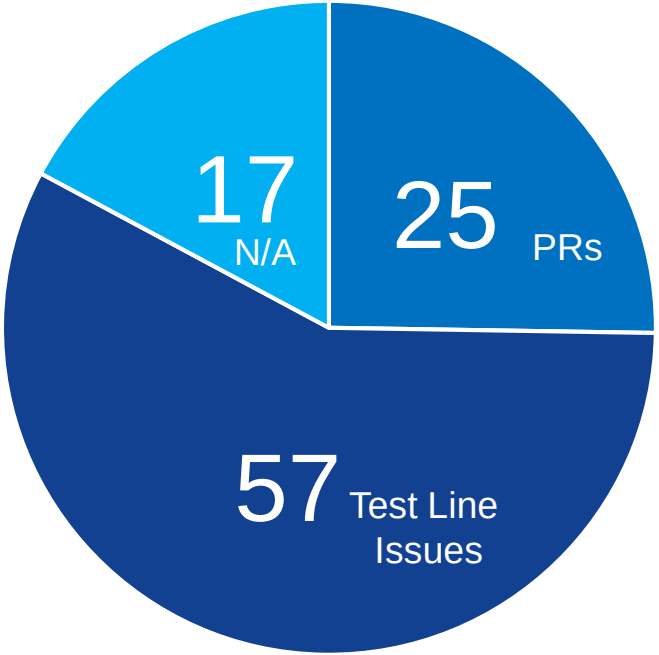
- **Project**: TDLTE BTS CRT (Continuous Regression Testing)
- **Time**: 2016.03 —— 2016.06
- **Participant**: myself

BTS CRT


1)Trigger
CI Build Server
COOP Server
3) Kanban
Switch
Power Controller
Control PC
TA Cases
TA Library
BBU
RRU
2) Runner
GPS
LAN Cable
Fiber
GPS Cable
BBU: baseband unit
RRU: remote radio unit
CI: Continuous Integration

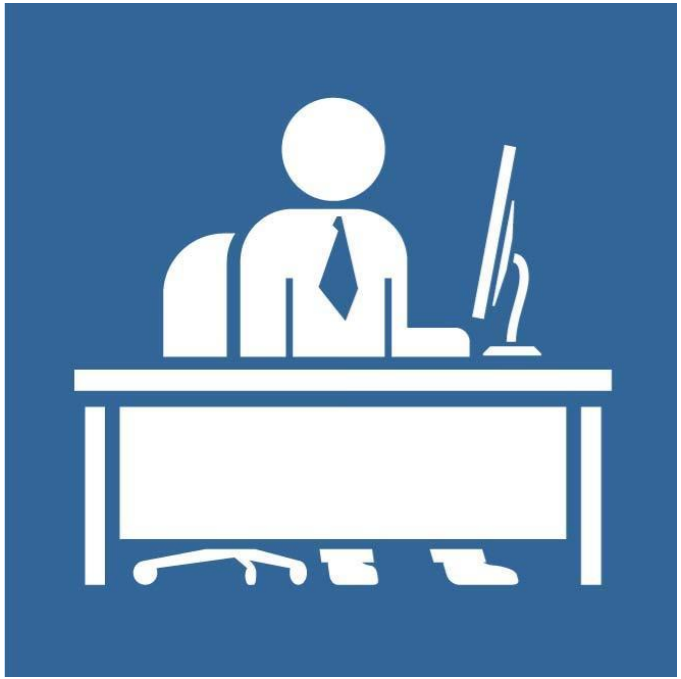
Test Summary

Issue Summary



Lost in thinking ……

TA Practice



- **Project**: SRAN BTSMED ET (Entity Testing)
- **Time**: 2016.08 —— now
- **Participants**: Dev HZ3 FV1 team, led by Ye Jason.

BTSMED ET

Operator

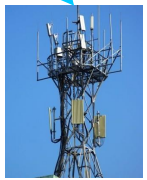
NMS(NetAct)

BTSMED

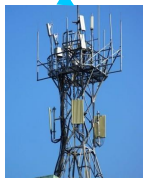
BTS

BTS

……



BTS



BTS

CI Build Server

COOP Server

Switch

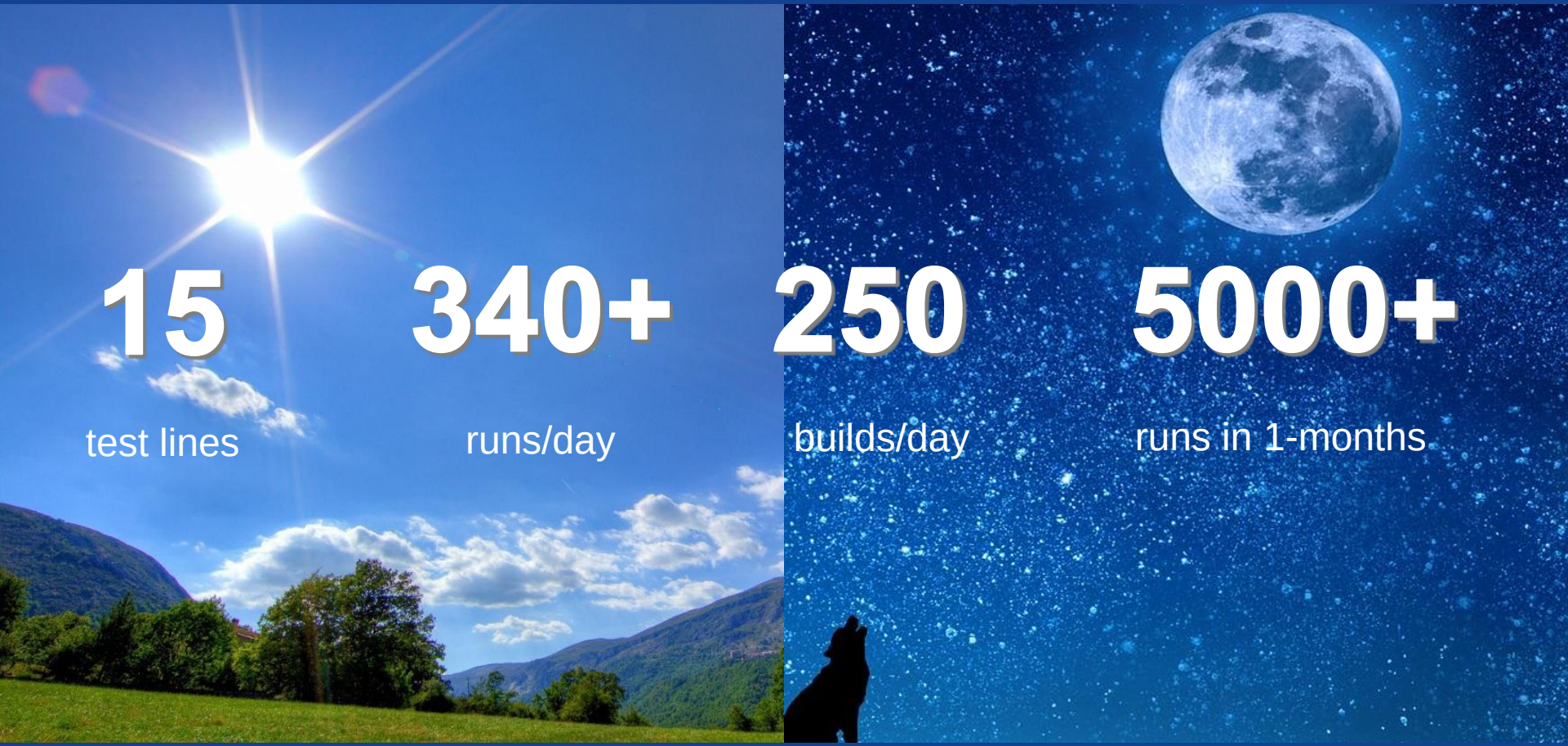
ET Server

TA Case

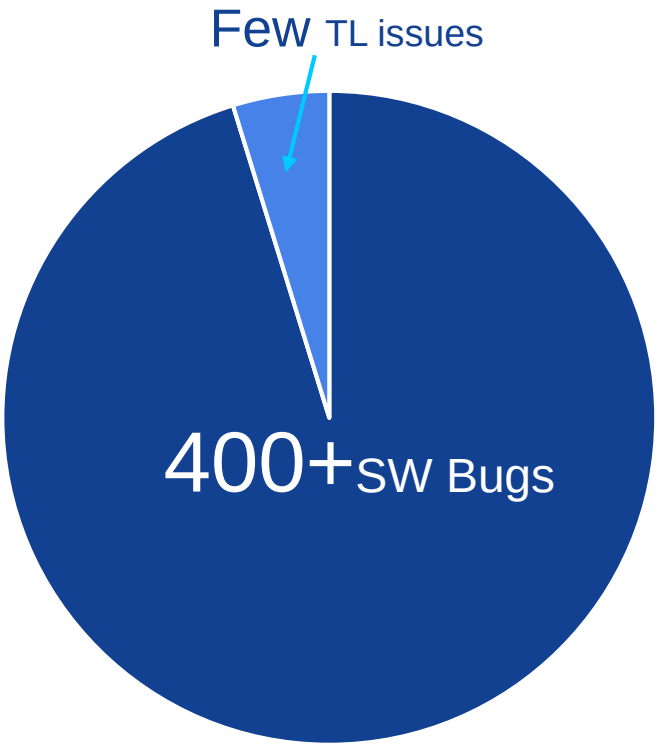
Simulator

BTSMED

Test Summary



Issue Summary



Lost in thinking, again ……

What a good TA

is ?

A not-so-good TA

Cannot find SW bugs efficiently

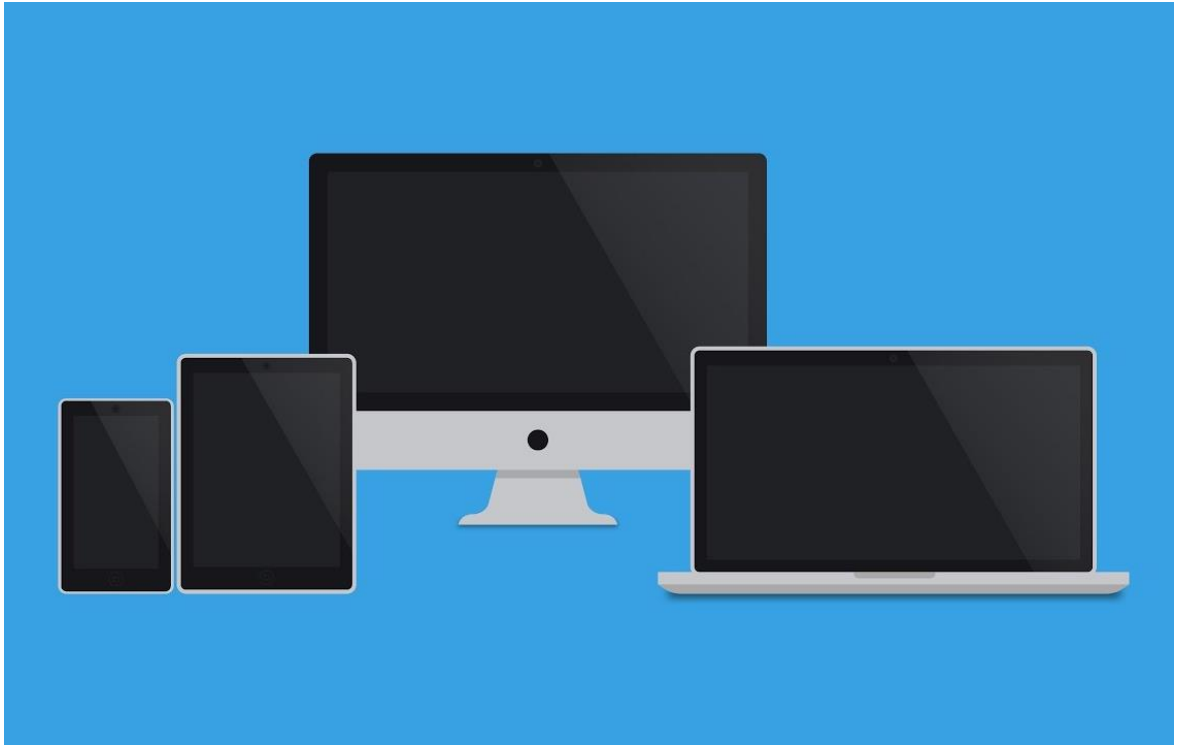
Find many TA issues

Test ENVs are unstable

Test lib/cases are hard to maintain

A Good TA

Simple & Reliable
OR
Efficient & Productive



TA Simplicity (Efficiency)

How much efforts are needed to develop and maintain TA libraries and TA cases?

TA Reliability (Productivity)

How much confidence do we have about that case failure is caused by SW bugs, not by TA itself?

How to achieve a good TA?

A few personal [P]roposals….

Achieving A Good TA

[P1] Add *TA Grooming* as part of software testing process.

TA Grooming

TA or not TA for each case?

New lib/keyword requirement?

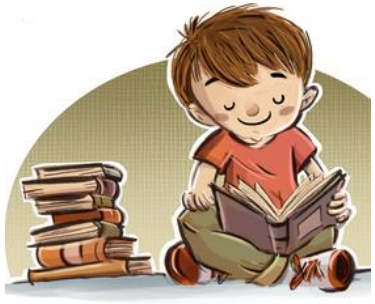
Action plan

Start TA Grooming As Early As Possible

Achieving A Good TA

[P2] Add *TA Case Review* as part of software testing process.

TA Case Review



Cases be as readable as requirement docs

All cases follow common paradigms

Large-screen meeting review

Everybody involved

Achieving A Good TA

[P3] *Test* of Test Automation is needed, especially for TA library/tools.

A Good Example: NBS

NBS: Newbie Simulator

“Simulating NetAct & SOAM BTS for BTSMED Entity Testing & Performance Testing”

http://gitlab.china.nsn-net.net/ta/nbs

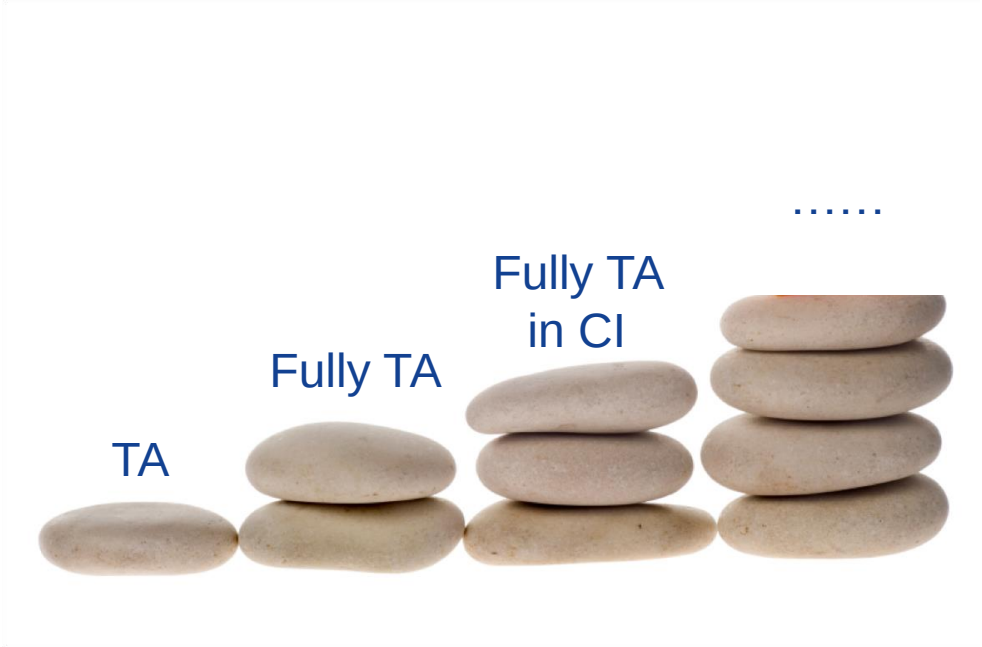
A Good Example: NBS

1	212	~20	2146	257
mocked BTSMED	unit test cases	seconds	commits	versions

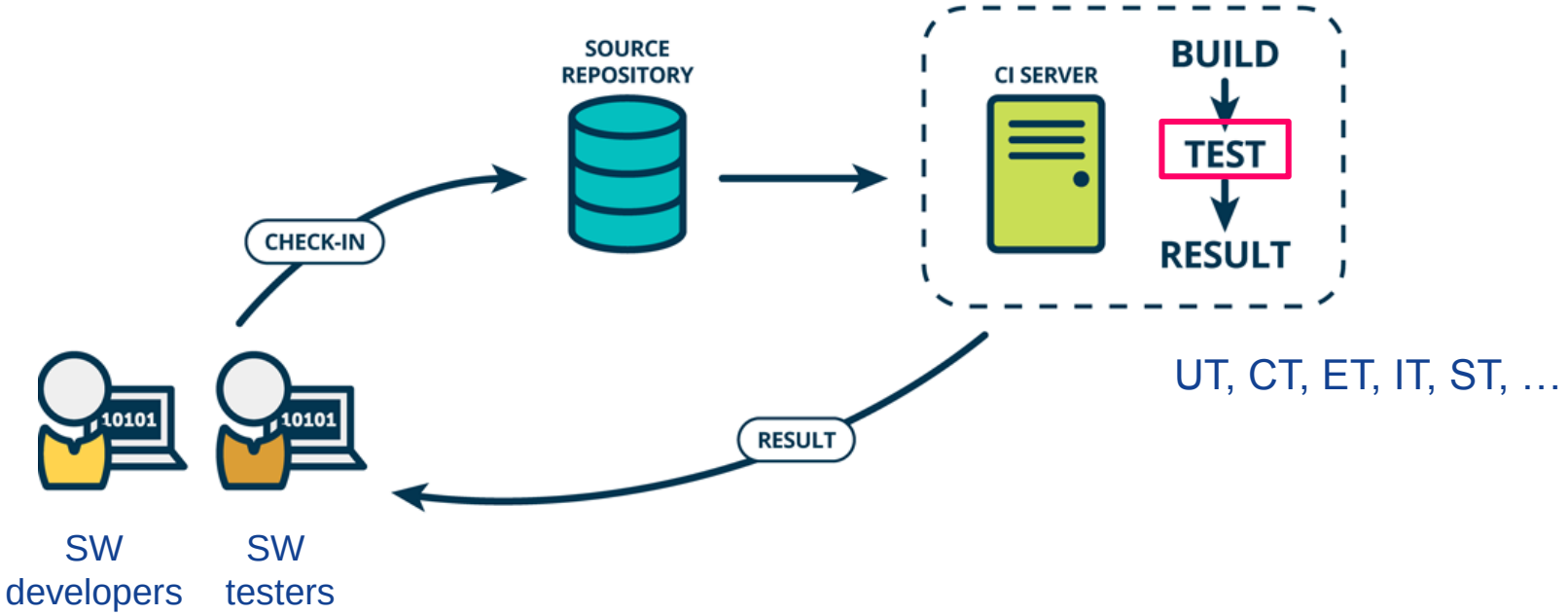
Our response time to issue pre-check/bug fixing is in hours, NOT in days

Achieving A Good TA

[P4] Add fully automated testing into continuous integration (CI) system.



Add Fully TA into CI



A more aggressive example: BTSMED ET

NOBODY can bring change to master repository with a failed ET !

Job name	Case Num	TL Num	Time	Rounds	Comments
Branch ET	163	15	~7 min	~250	Run on every code change quickly
Trunk ET	190	1	~40 min	~15	Run on every released build
Trunk Stress ET	190	1	~40 min	~75	Run on newest builds stressfully

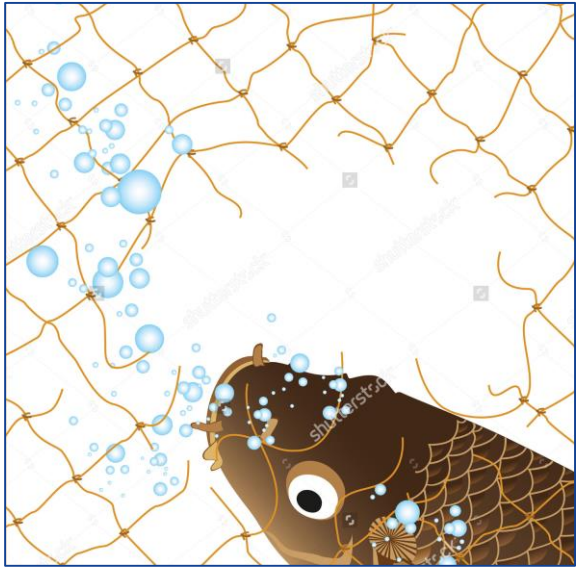
But it is NEVER an easy work ……

It is worth doing since "QUALITY MATTERS"

[P5] Continuously improve TA by RCA/EDA.

RCA/EDA of Test Automation

1. For each issue proven to be TA bug, do RCA (root cause analysis)



2. For each SW bug found by post test stages, do EDA (escaped defect analysis)

First of all, reproducing bug by changing test code

[P6] TA Left-shift: invest more on automation of *early* test stages.

Necessity(必要性) of TA Left-shift

The Google Testing Law (谷歌测试定律)：

“As SW test proceeds(UT->CT->IT->ST or small->medium->large test), the cost of fixing a discovered SW bug increases at an exponential scale”.

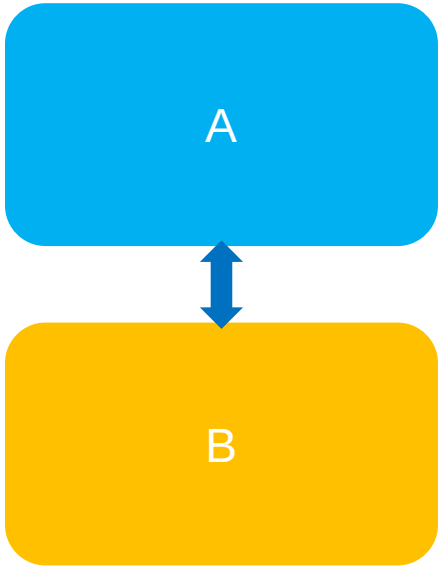
Feasibility(可行性) of TA Left-shift

The Testing Coverage Law (测试覆盖定律)：

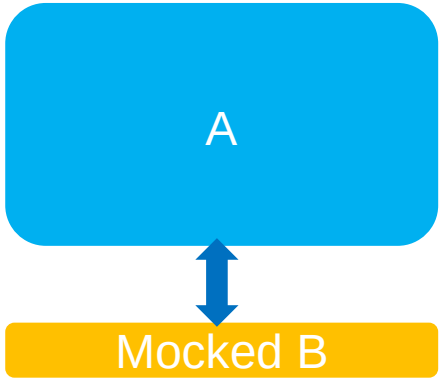
"For multi-stages SW testing, any SW bug discovered at the current test stage, could have been discovered at the former stages by increasing or modifying one test case"

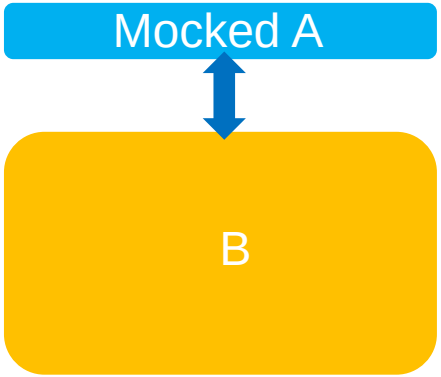
[P7] Use Mock technique as much as possible.

What is Mock ?



Test Stage *N+1*





Test Stage *N*

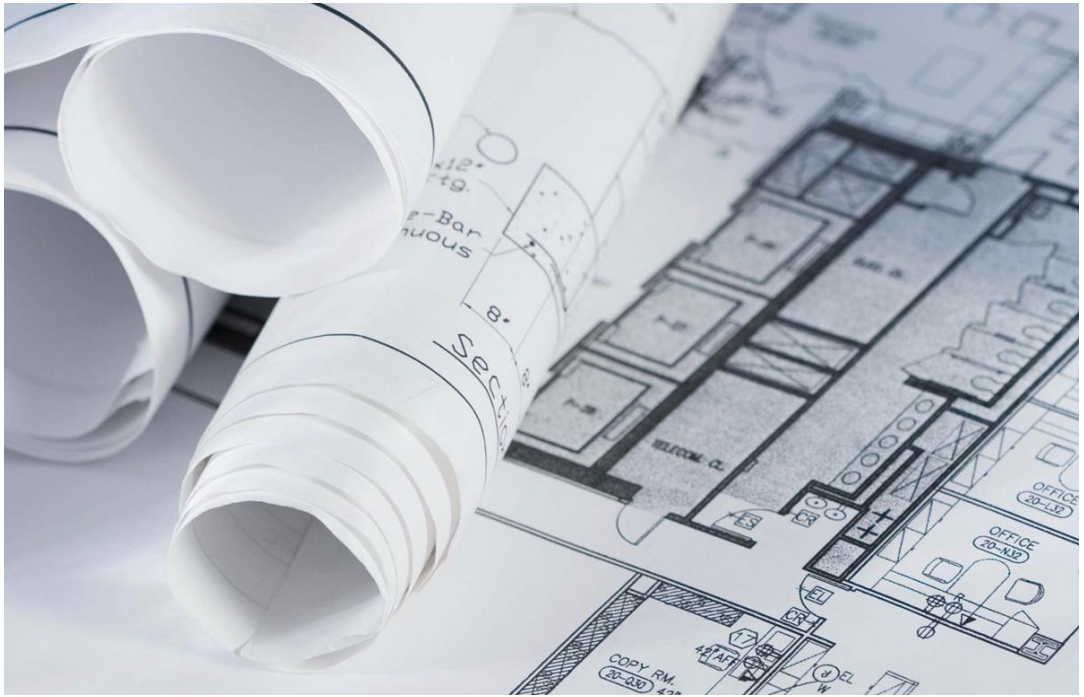
Benefits of Mock

Focus on the tested object

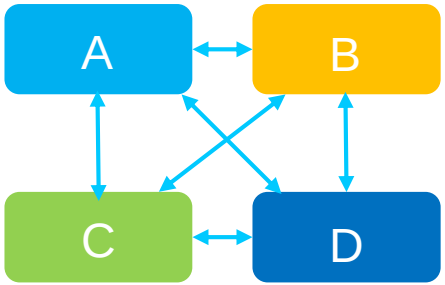
Starts test early

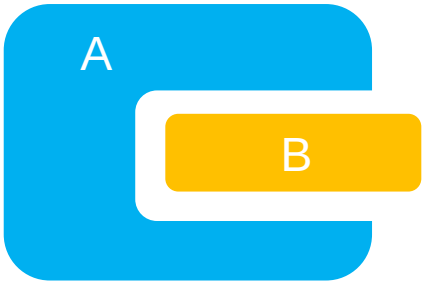
Cheap since only interface needs mocking

[P8] Take *testability* into account when designing software.

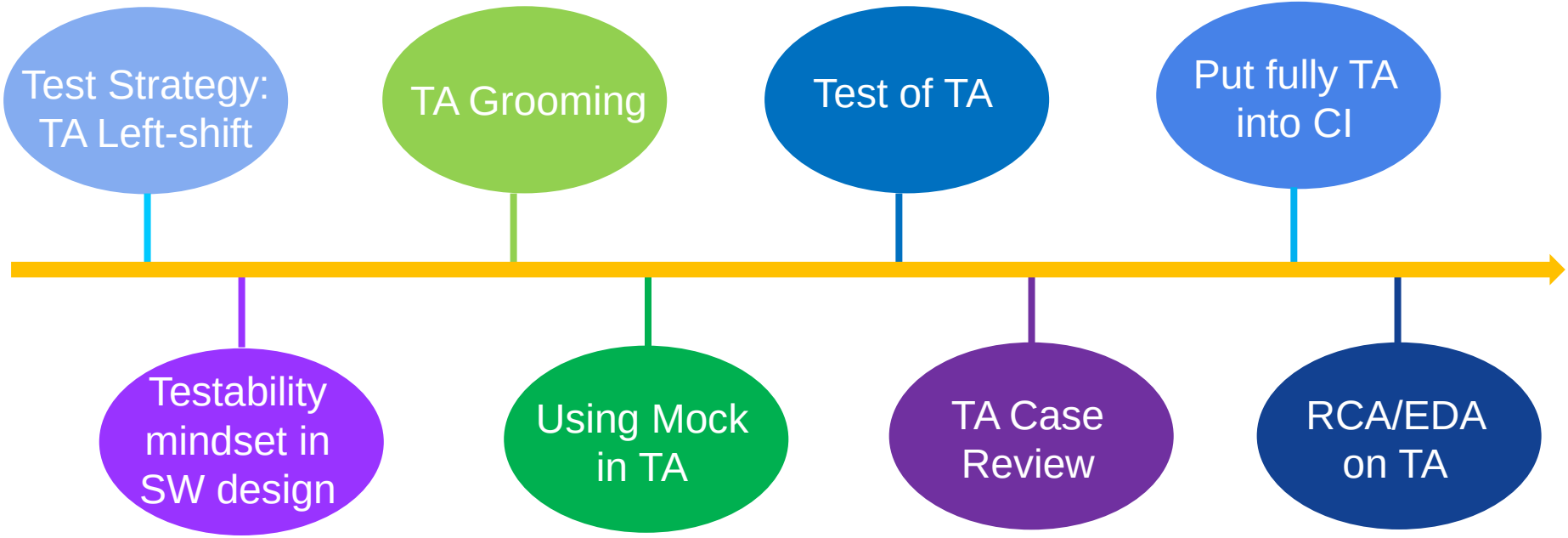


Some Bad Examples





Summary of my proposals towards a good TA



To repeat……

Test Automation is
Software Development

Learn More

My Blog

GTAC

Q & A