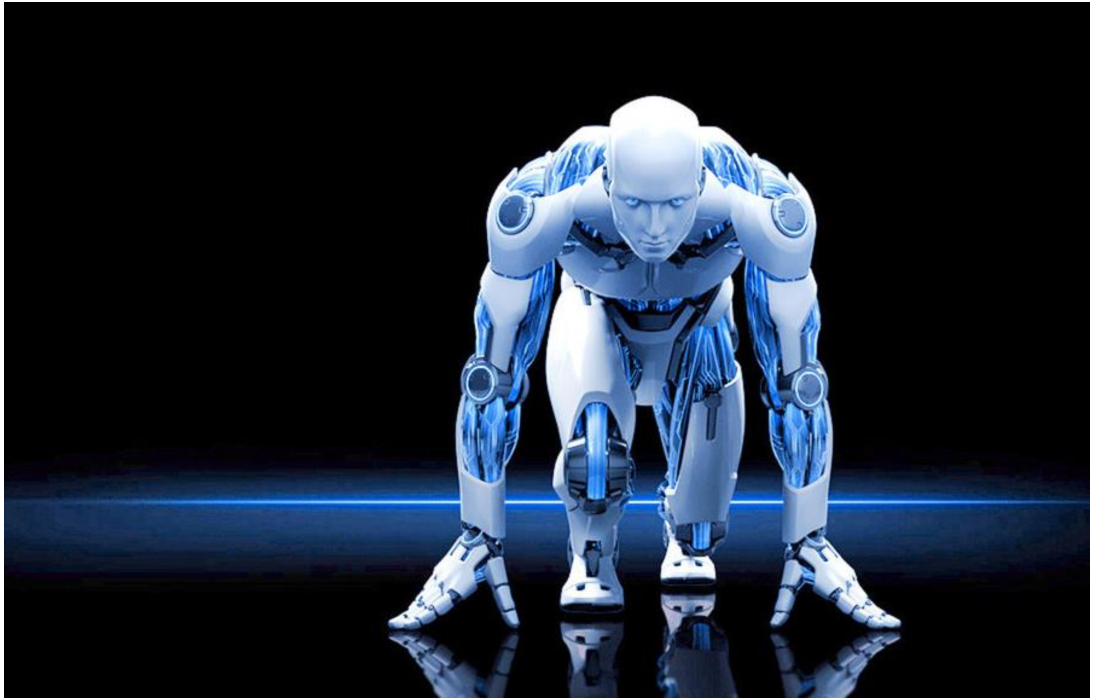
# Lessons Learned From TA Practices

Xiao Shiliang-Shelwin (肖世良)

2017.05.25

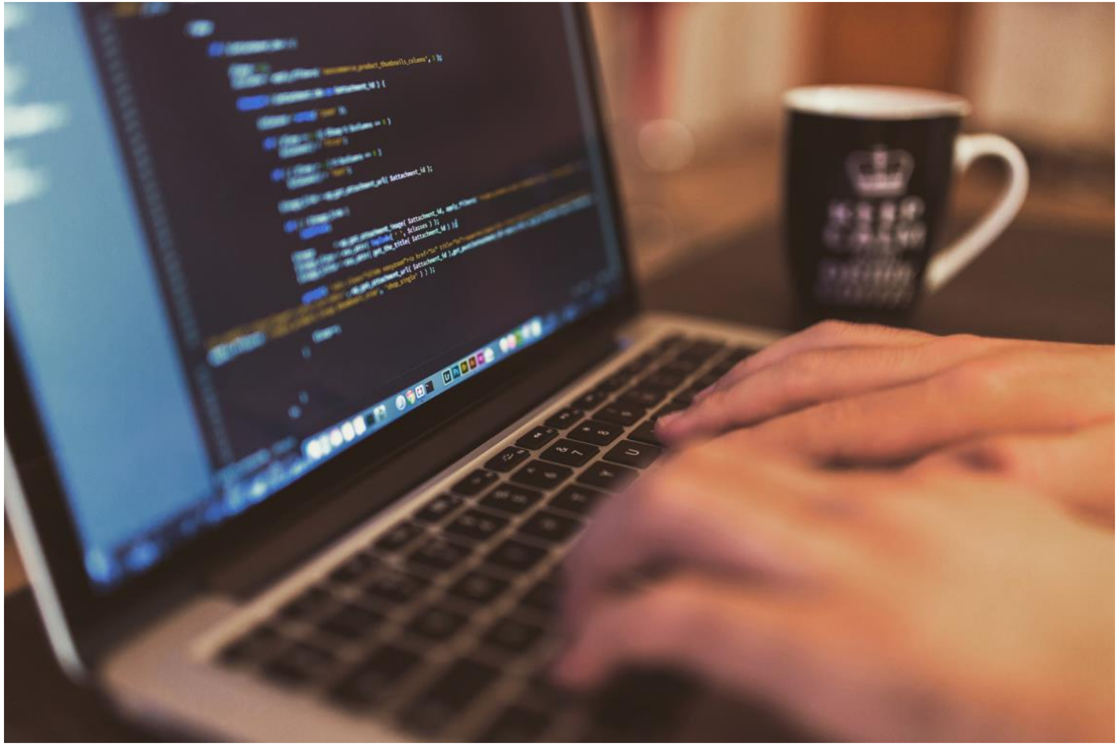
# TA: Test Automation

*Let computer do software testing for human*



# Fact of Test Automation

*Test Automation is*
*Software Development*



# Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

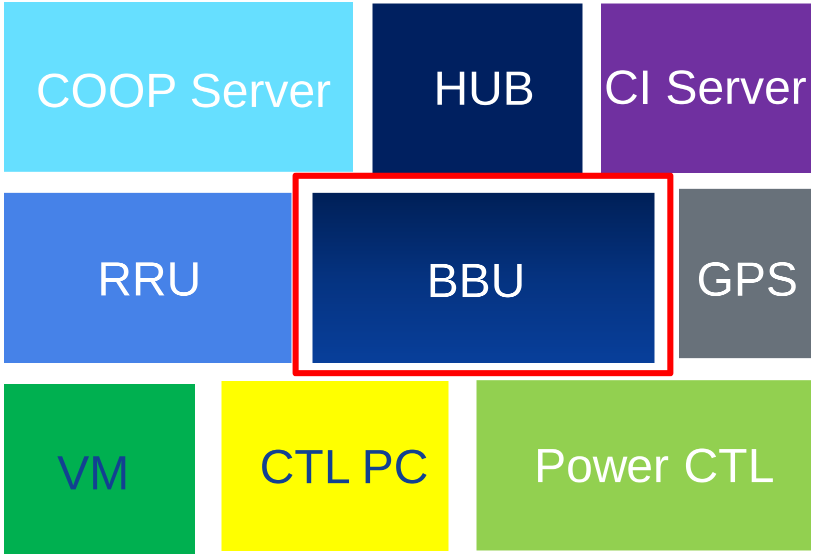
- Lesson: What a good TA is
- Lesson: How to achieve a good TA

# Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

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

# TDLTE BTS CRT



COOP Server

HUB

CI Server

RRU

BBU

GPS

VM

CTL PC

Power CTL

# Test Line Info

| TL conf. | UTE IP/port/loc. | UTE Host IP | UTE S1 port | BBU ID | BBU IP | BBU S1 port | BBU loc. | BBUPB IP/port | BBU PB S1 port | BBU GW | BBU mask | RRU loc. | RRU PB IP/port |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **FSIH FZHM 8Pipe** | 10.69.69.117<br>Eth2<br>AA1-5-1 | 10.69.69.121 | | 1697 | 10.69.68.116 | | AA3-6 | 10.69.68.97<br>Port6 | | 10.69.68.126 | 255.255.255.224 | AA3-6 | |
| **FSIH FZHS 2Pipe** | 10.69.69.118<br>Eth3<br>AA1-5-1 | 10.69.69.121 | | 1422 | 10.69.3.82 | Switch 2-8 | Y5-1 | 10.69.3.74[illegible]<br>Port1 | [illegible]ch1 [illegible]16 | 10.69.68.126 | 255.255.255.224 | Z6-4 | |
| **FSIH FZND 2Pipe** | 10.69.69.100<br>P1p1<br>AA1-1-2 | 10.69.69.99 | | 513 | 10.69.68.73 | | AA[illegible] | [illegible]68.97<br>[illegible]rt4 | | 10.69.68.126 | 255.255.255.224 | AA5-1 | |
| **FSIH FZHQ 8Pipe** | 10.69.69.119<br>Eth4<br>AA1-5-1 | 10.69.69.121 | | 672 | 10.69.68.[illegible] | | AA2-4 | 10.69.68.114<br>Port4 | | 10.69.68.94 | 255.255.255.224 | AA6-6 | |
| **FSIH FZFF 8Pipe** | 10.69.3.111<br>P10p1<br>Y1-1-1 | 10.69.69.112 | Switch 1-4 | 14[illegible] | [illegible].3.134 | Switch 2-16 | Y4-1 | 10.68.184.193<br>Port1 | Hub2-1 Switch1-35 | 10.69.3.126 | 255.255.255.224 | Z4-1 | |
| **AirScale FZHM 8Pipe** | 10.69.69.105<br>Eth1<br>AA1-3-1 | 10.69.69.104 | | | 10.69.68.93 | | AA9-6-L | 10.69.68.103<br>Port2 | | 10.69.68.94 | 255.255.255.0 | AA8-7 | |

8 x 6 = **48** cables, 15 x 6 = **90** ports

# Test Summary

**50+**

runs/day

**2~4**

builds/day

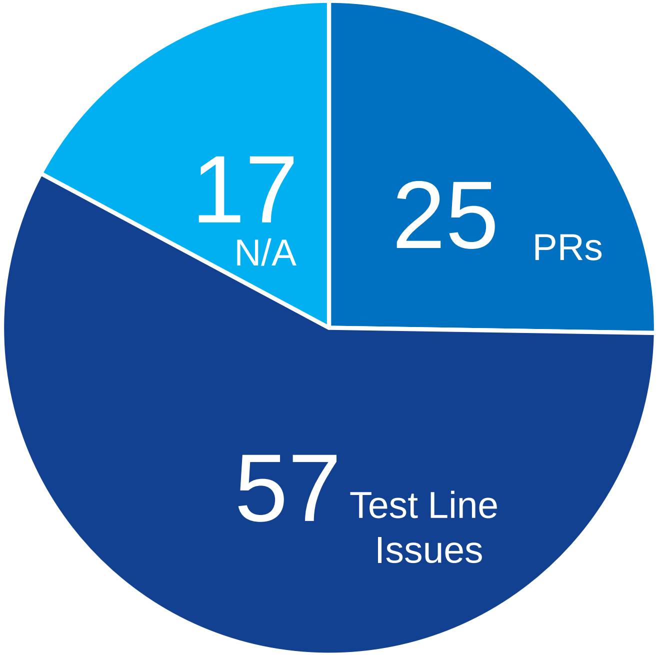
**5000+**

runs in 3-months

**87**

reports in 3-months

# Issue Summary

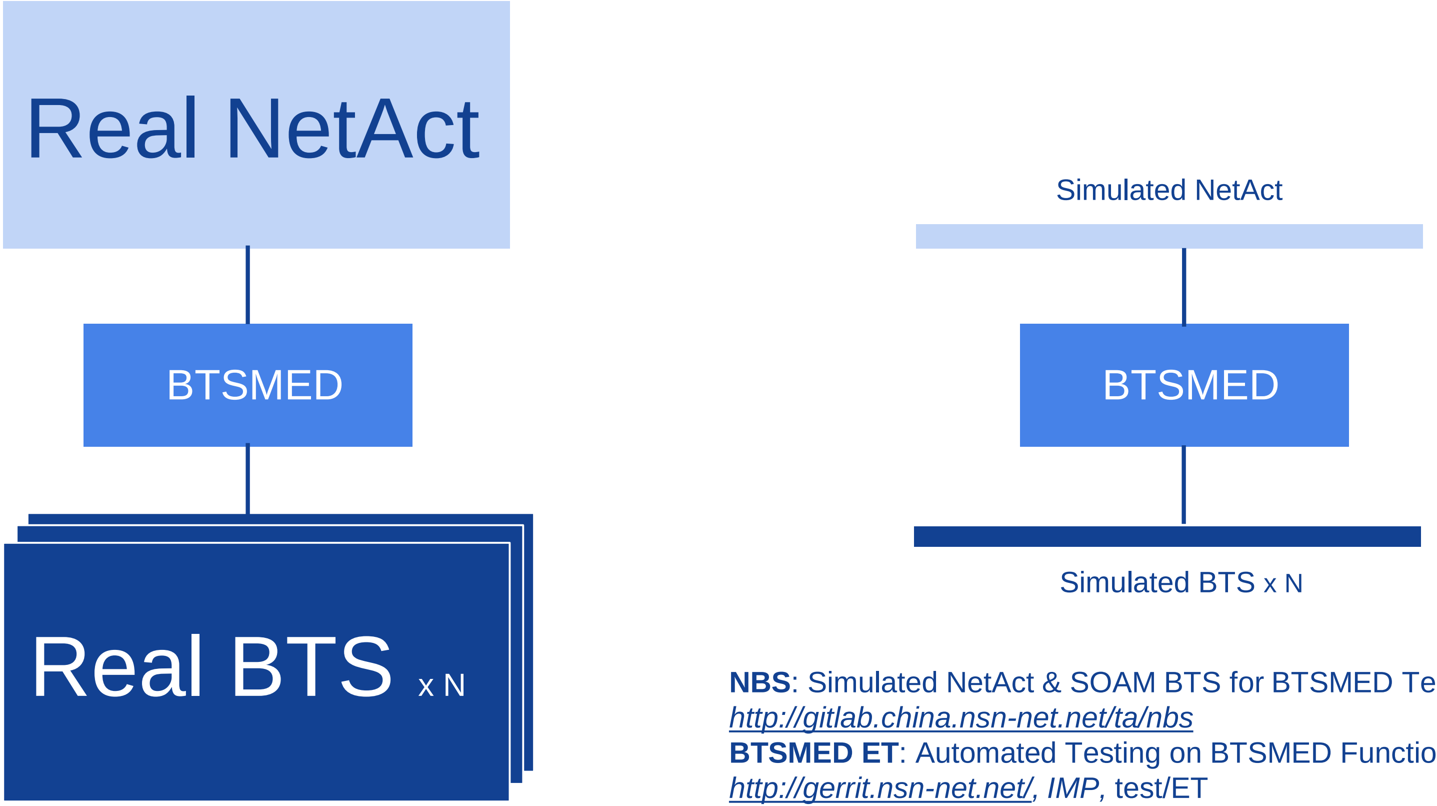


# Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

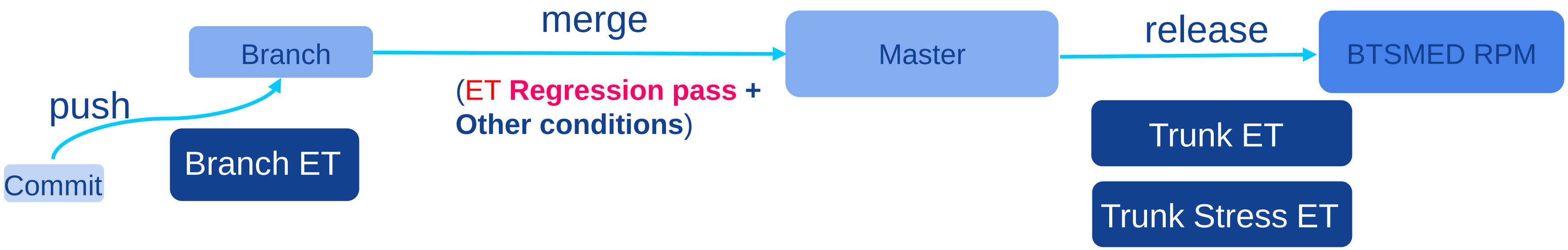
- Lesson: What a good TA is
- Lesson: How to achieve a good TA

# BTSMED ET



**NBS**: Simulated NetAct & SOAM BTS for BTSMED Testing, *http://gitlab.china.nsn-net.net/ta/nbs*
**BTSMED ET**: Automated Testing on BTSMED Functionalities, *http://gerrit.nsn-net.net/*, *IMP*, test/ET

# Change-based BTSMED ET Regression

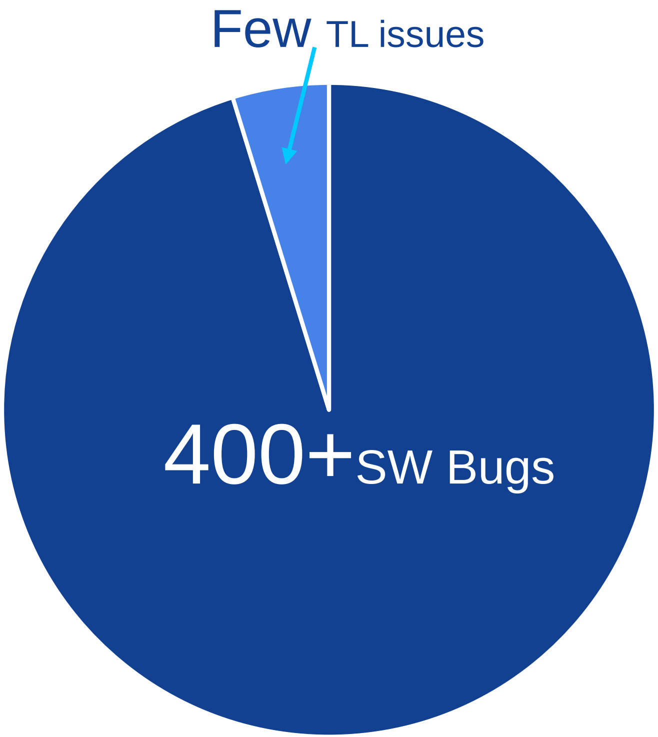


| Job name | Case Num | TL Num | Time | Rounds | Comments |
|---|---|---|---|---|---|
| Branch ET | 163 | 15 | ~7 min | ~250 | Run on every commit **quickly** |
| Trunk ET | 190 | 1 | ~40 min | ~15 | Run on every **released** build |
| Trunk Stress ET | 190 | 1 | ~40 min | ~75 | Run on newest builds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

# Test Summary

| | | | |
|---|---|---|---|
| **340+** | **250** | **5000+** | ***+N*** |
| runs/day | builds/day | runs in 1-months | runs in local ENV |

# Issue Summary



# Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

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

# A Bad TA

Cannot find SW bugs efficiently

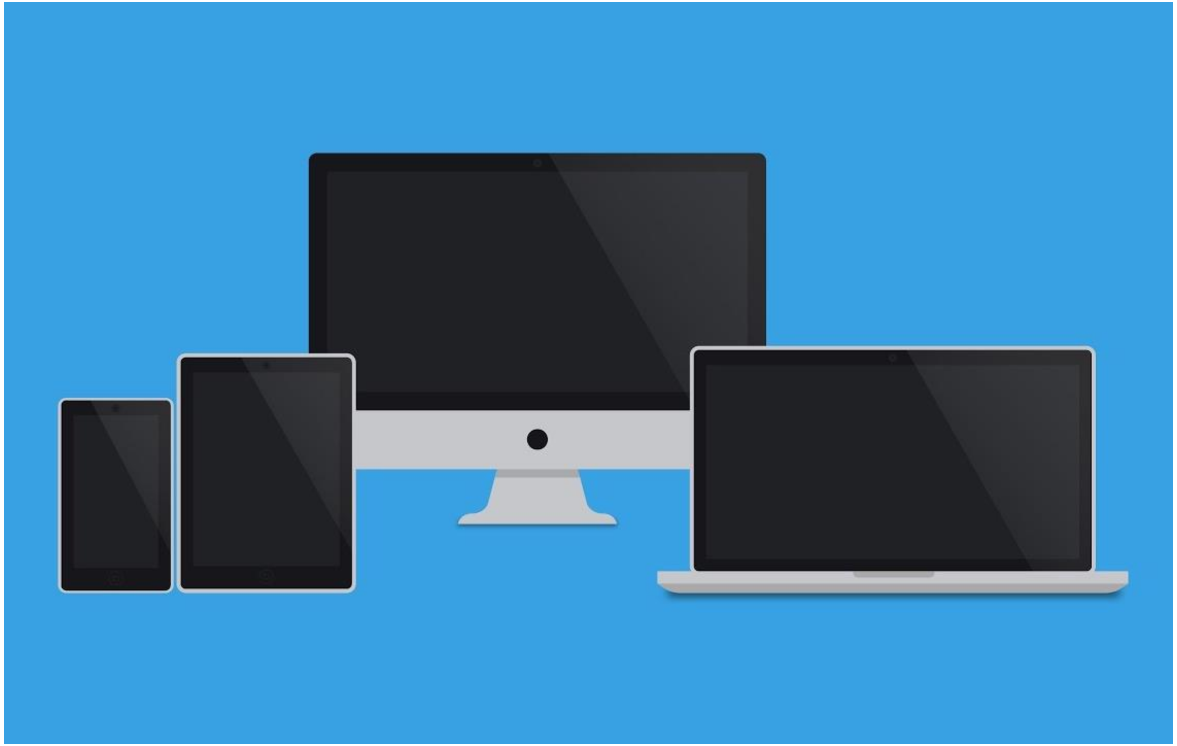
Find many TA issues

Test ENVs are unstable

Test lib/cases are hard to maintain

# A Good TA

*Simple and Reliable*



# TA Simplicity

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

# TA Reliability

*How much confidence do we hold on that case failure is caused by SW bugs, not by TA itself?*

# Content

- Practice: TDLTE BTS CRT
- Practice: BTSMED ET

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

# Achieving A Good TA

## Invest on Automation of *Early* Test Stages

# 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.**

# 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* stage by increasing or modifying one test case.**

# Achieving A Good TA

**Use *Mock* Technique As Much As Possible**

# Benefits of Mock

Focus On

Early

Cheap

# Achieving A Good TA

***Test* of Test Automation**

# A Good Example: NBS

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

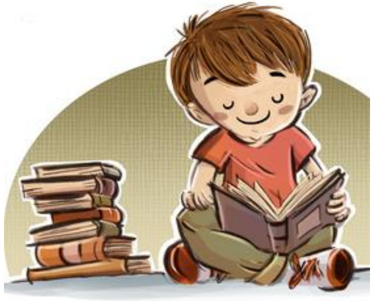
# Achieving A Good TA

## TA Grooming

# Achieving A Good TA

## TA Case Review

# TA Case Review



Cases be as readable as requirement docs

All cases follow common paradigms

Large-screen meeting review

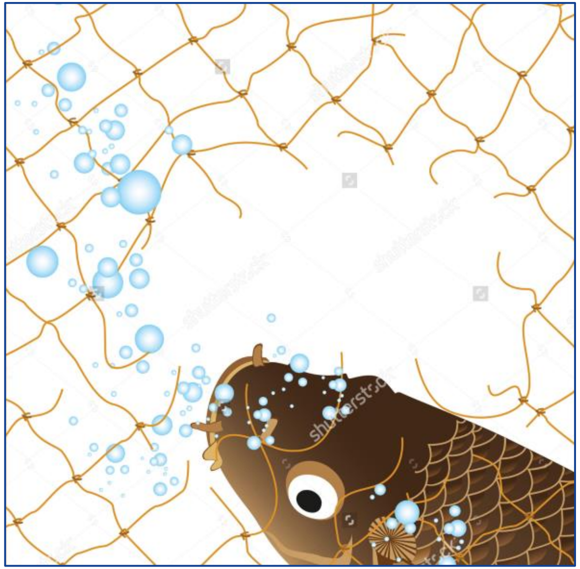
Everybody involved

# Achieving A Good TA

## Improvement by Escaped Defect Analysis

# Escaped Defect Analysis (EDA)

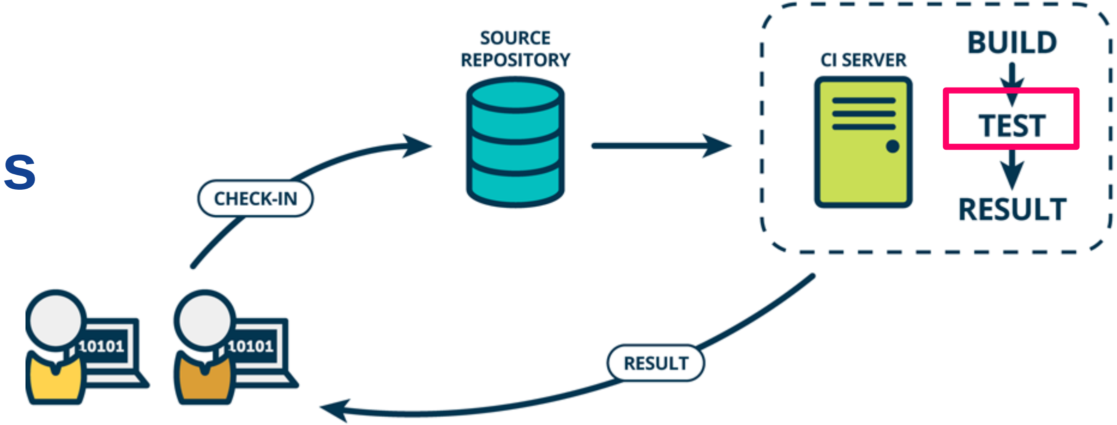
1. For each issue proven to be TA bug, do EDA



2. For each SW bug found by next-stage test, do EDA

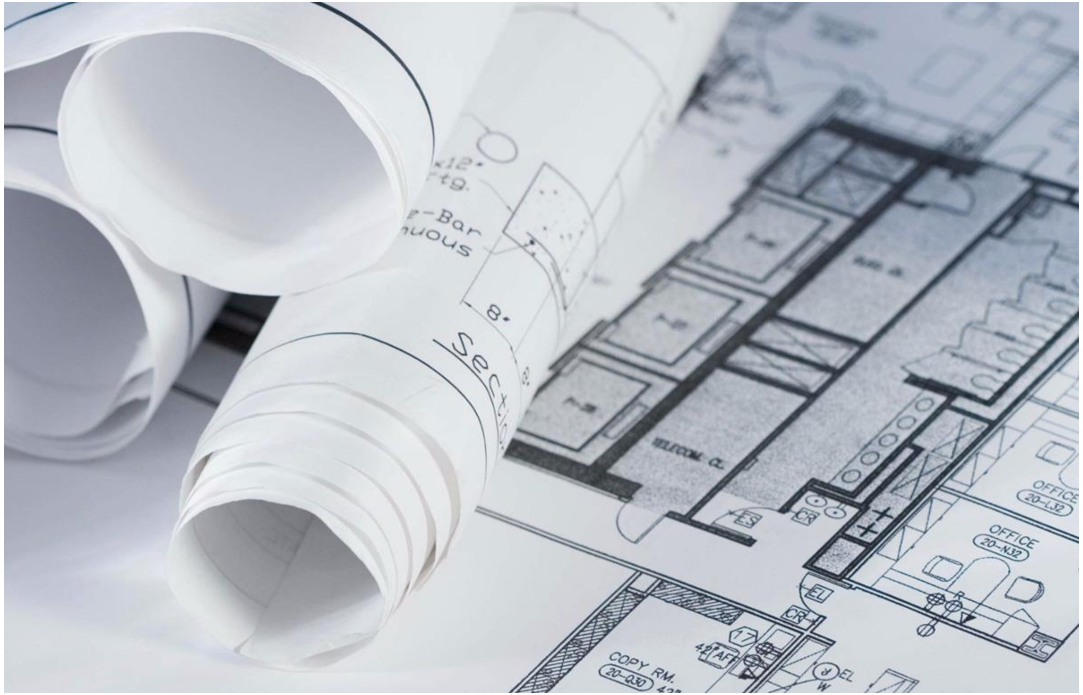
# Achieving A Good TA

## Put *Fully* Automated Testing into Continuous Integration



# Achieving A Good TA

## *Testability* Matters in Software Architecture



# Q & A

# Learn More

TA Forum

My Blog

GTAC