

Test Design Problems Investigation

邵晓梅

testcomt@gmail.com

新浪微博: MiniStarClub001-邵晓梅

博客: www.taixiaomei.com

Background

- Test design Problems Investigation Report
- MFQ&PPDCS Test Analysis and Test Design Framework

2-day's course

<http://www.taixiaomei.com/archives/31>

CONTENT

Part I The Core Problem

The problem of “Test Analysis”

Part II The Problems existed in test design

List and analyze some problems out of investigation

Part III An example

Show a test design example

Part I The Core Problem

The Deveopment of Test Analysis

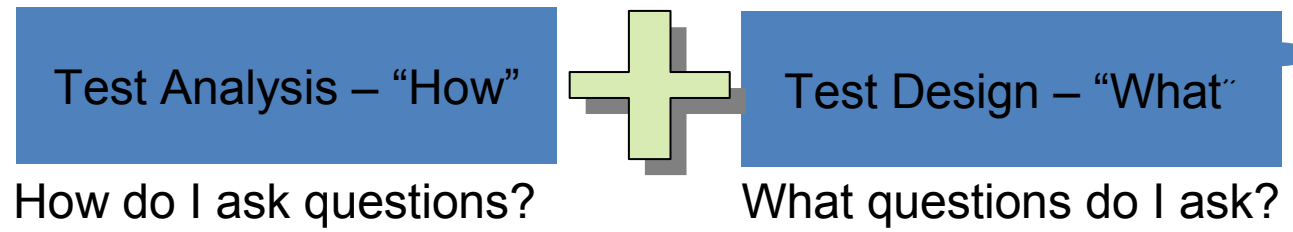
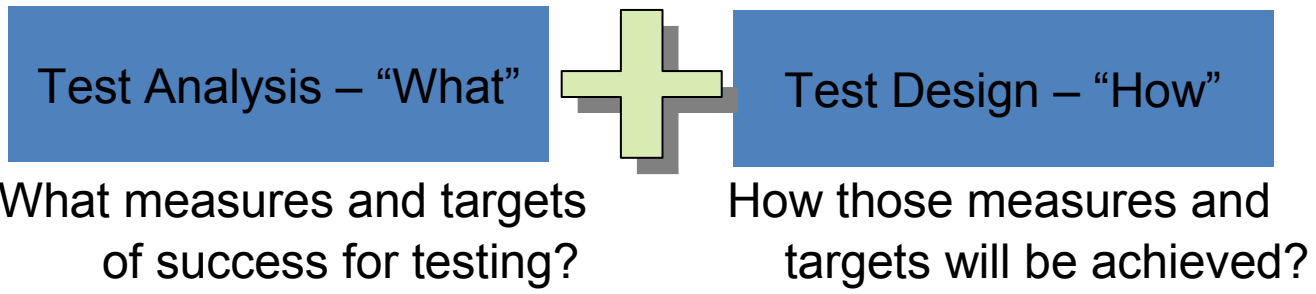
2001, Requirement - test design

2003, Requirement - test analysis - test design -----HLT

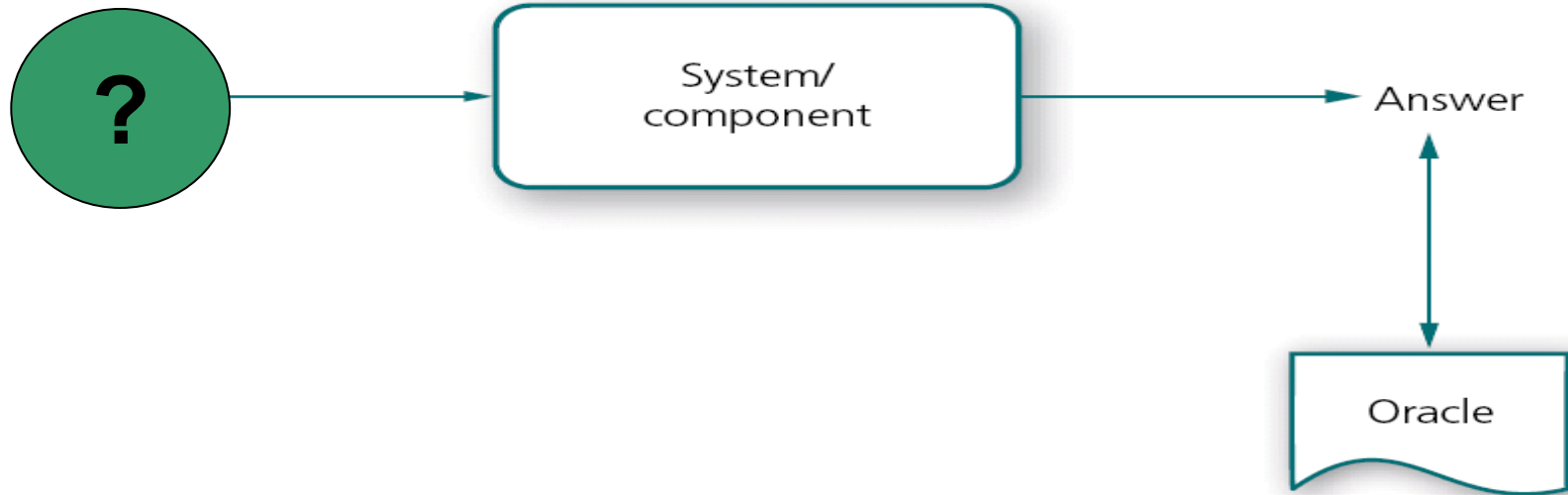
There is still a blank for LLT test analysis!

This blank can be filled by “Models”.

Test Analysis & Test Design(1)



Test Analysis & Test Design(2)



(Copyright) This page is from <Essential software test design> by Torbjorn Ryber

Test design process for one feature

Single Function Test Design

	F1	F2	F3	F4	F5	...
TR1		TS11			TS12	
TR2				TS21		TS22
...	X	X			X	

Function Interaction Analysis

LLT test design

HLT function test design

HLT test design

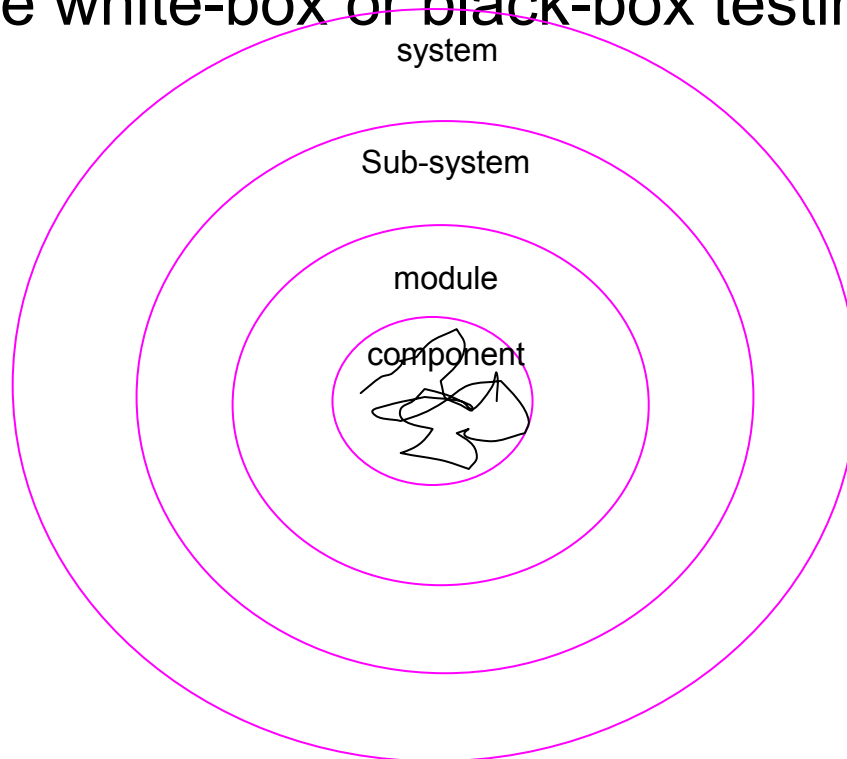
	FUN	comp	CFG	USE	PERF	...
TR1		X			X	
TR2				X		X
...	X	X			X	

Test Type Analysis

LLT Test Design(1)

What should LLT focus on, from the whole software life cycle point of view ? What is its emphasis on?

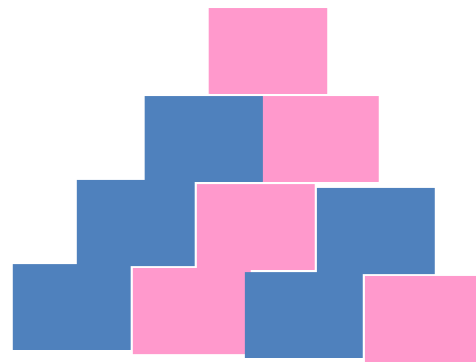
Compared with other test design processes, one of single function test design's most obvious features is “detailed” design, which can be white-box or black-box testing.



LLT Test Design(2)

What's the biggest difference between a designer and a tester?

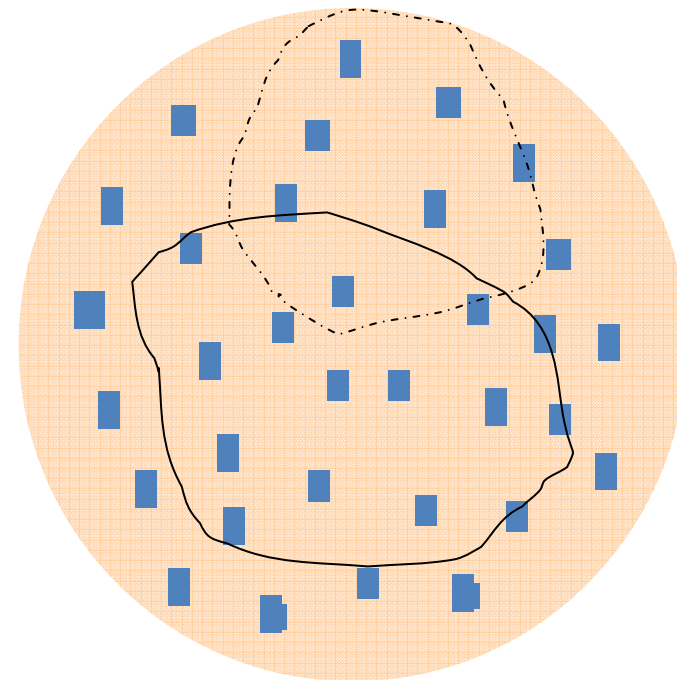
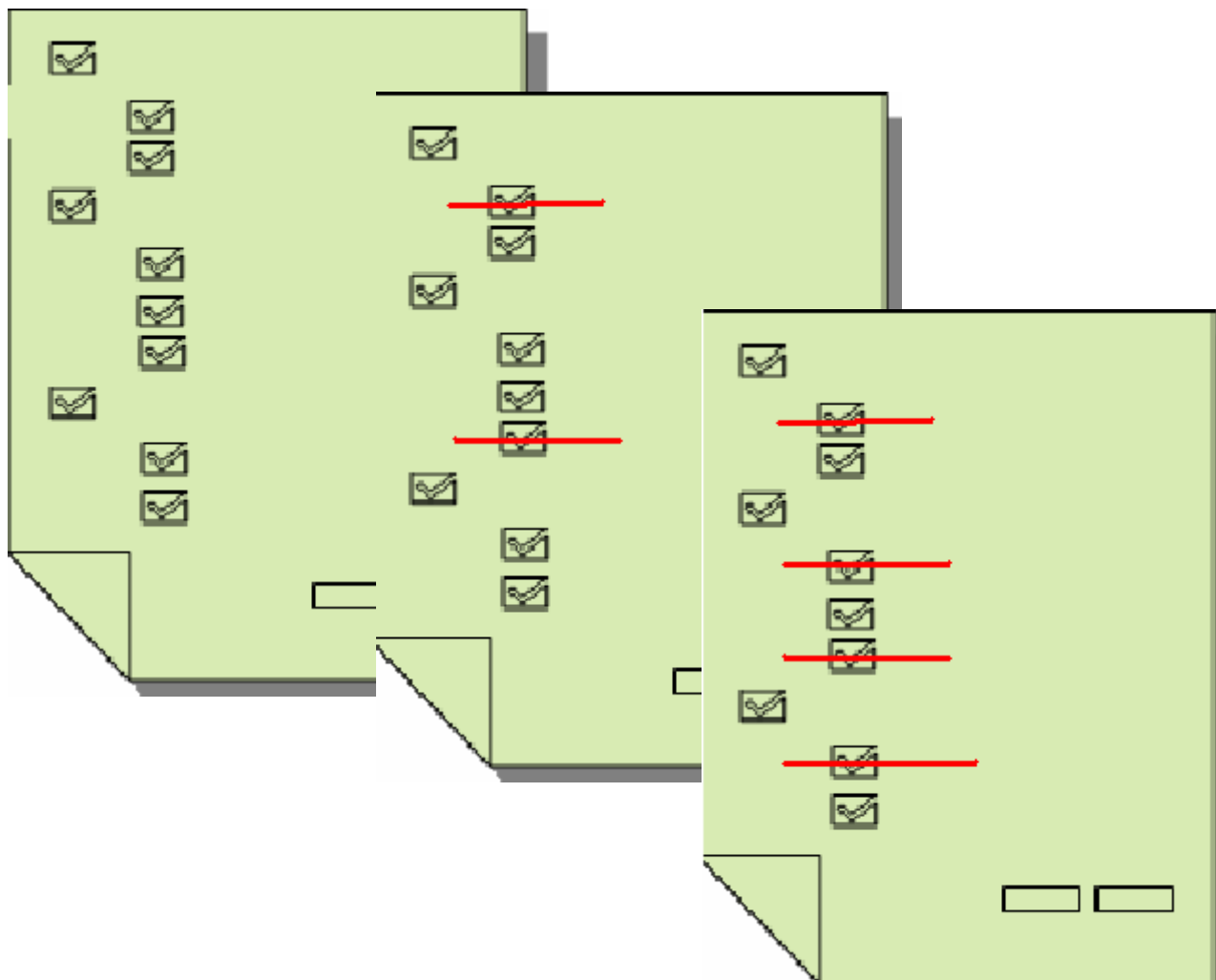
As a designer, when you do test design based on specification docs, you must use totally different viewpoint from that when you design these docs.



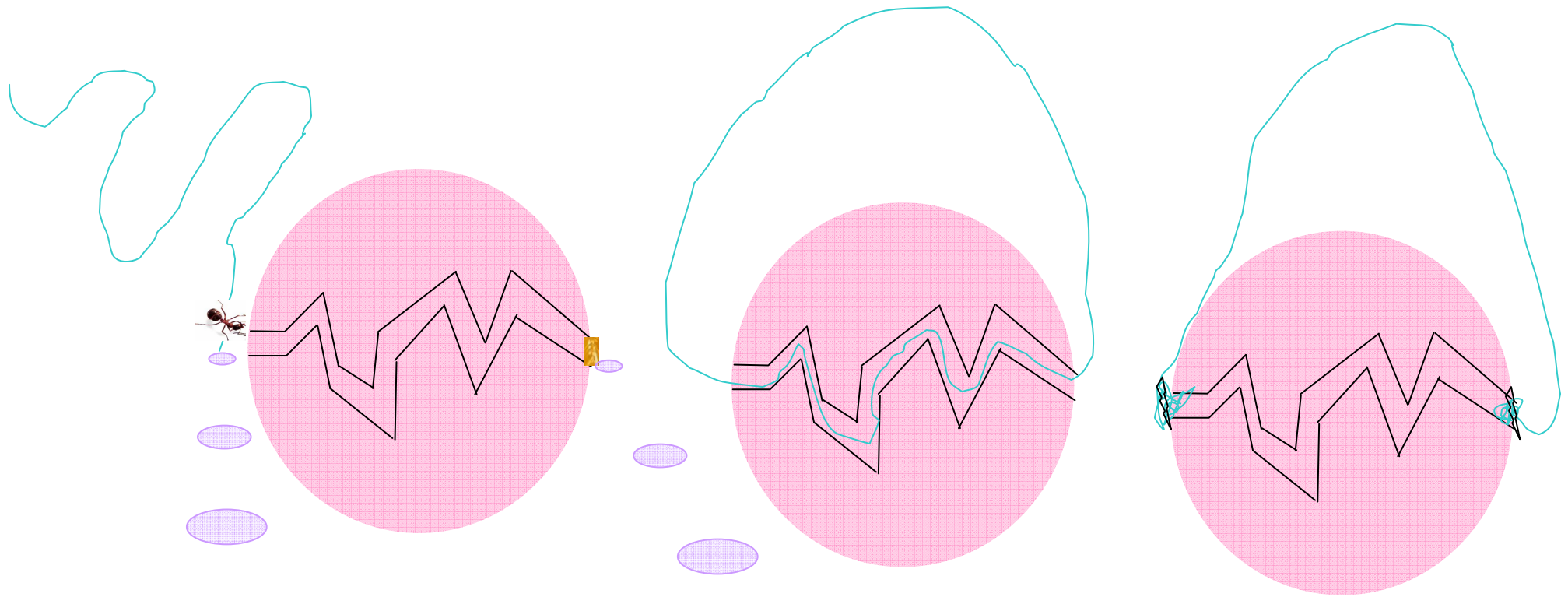
Constructive

Desstructive

The selection of test cases



Inspired by the story of “Confucius strings through a bead”

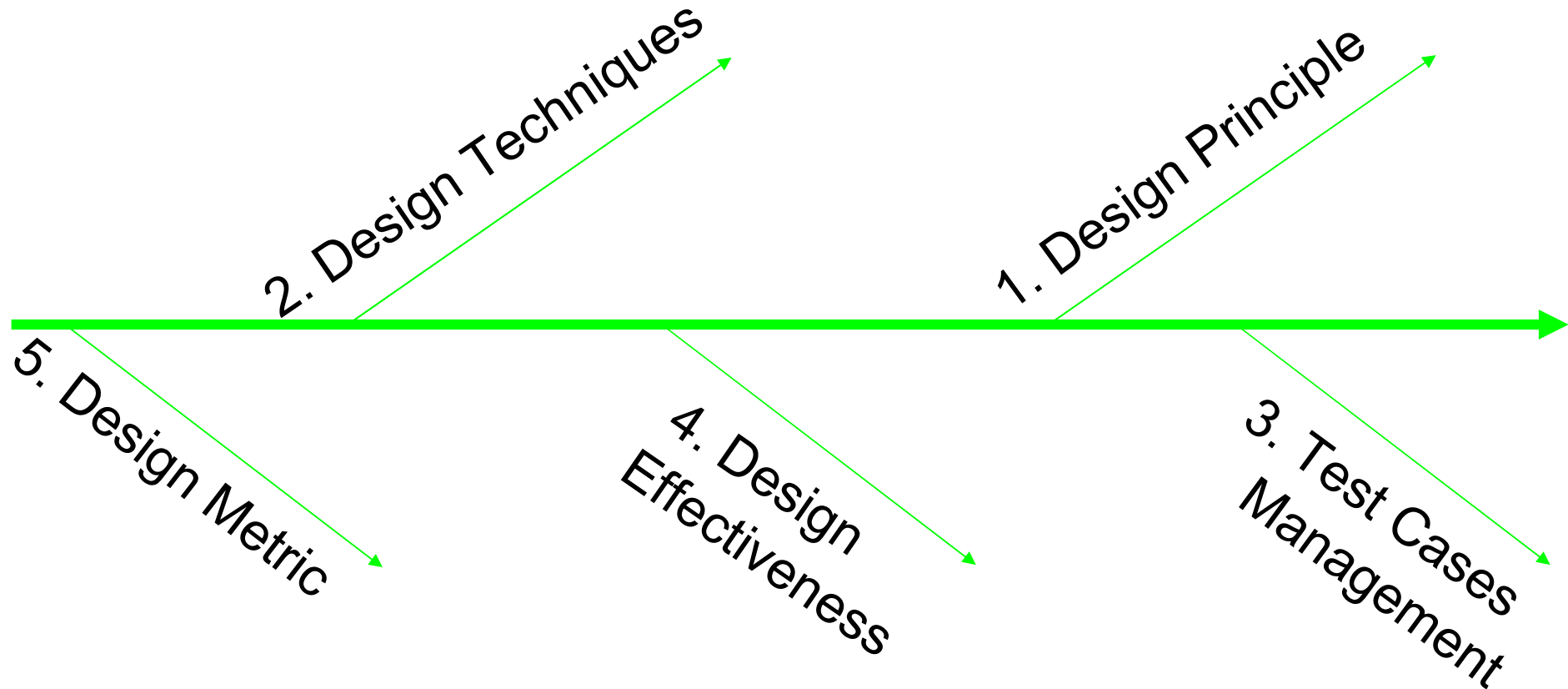


Test Design
Techniques

Effective test
cases

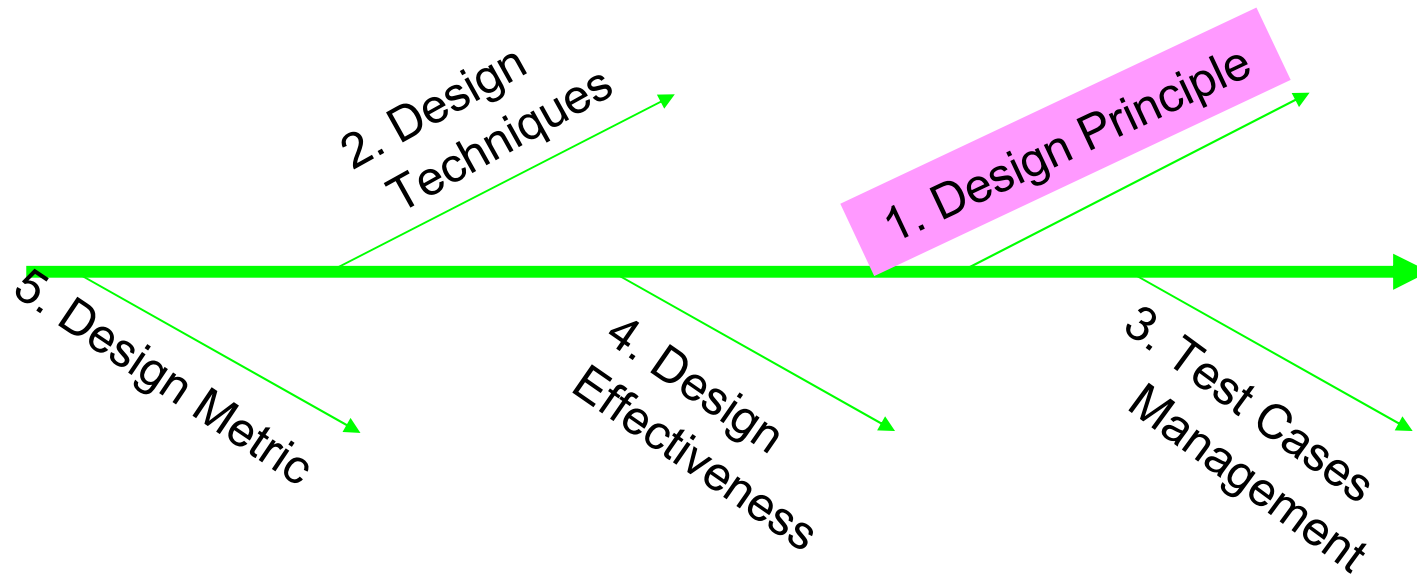
Part II Problems existed in test design

The existed test design problems can be shown in a Fish-Bone Graph. This slides will analyze but **not try to solve** all the problems.



1. Test design principle

Some test design principles have been overlooked or misunderstood.
some other principles people can not make an agreement on.



1.1 How detailed a test case should be?

- Testers often ask this question:

Whether the goal of test design is that, the more detailed a test case is, the better the test case is, so the easier it is for the test case to be executed, and the less dependant on executers the test result is.

- The answer differs on different persons. So the current situation is that some TCs are rather detailed, while most of TCs are rather simplified.
- I don't think all TCs should have the same extent of detailness, but we do need a set of principle, which will be understood and followed by most of people. e.g. some key fields of TC should exist, and more detailed info is needed for high prio TCs, etc.

1.2 How to deal with abnormal test cases?

- People who believe test is a process of finding errors are prone to design many abnormal test cases, but there are some other viewpoints.
- At least, many people believe that there should not be too many abnormal TCs in system test due to the high cost of triggering abnormal TCs in system test level.
- Test designers are doubting about whether or not to write such a TC which is extremely difficult to trigger and demands a complex test environment.

1.3 How many test cases are enough?

- Designers are facing the dilemma of design leakage and design redundancy for a long term. To avoid leakage, people use many test techniques and design a great number of TCs with small granularity, which creates surplus. To reduce redundancy, people cut down TCs which results in test design leakage again.
- The current situation is that, at one hand, the number of designed TCs is 2 or 3 times of the ability of executing them; at the other hand, the lack of effectiveness of TCs is a common phenomenon.
- In fact, we are not left alone. One of experts expressed the same situation in another company: “We have too many test cases and too little time to execute them all”.
- I believe choosing right test techniques combined with risk-based test strategy will help to alleviate this dilemma.

1.4 Where to lay test design emphasis on?

- We can never know when we'll get to a complete test, however, what we can know is what risks our customers can accept.
- Good test design should have a high probability to find important errors.
- Good test design always try to focus on the most risky part of the product.
- Testers always want to lengthen test time more and more to make test as complete as possible, while stakeholders always want to curtail the cost on test activities. Cost, time and quality need to be balanced all the time.
- The reality is that no obvious test emphasis is laid on during test design process.

1.5 How to look on black-box testing and white-box testing?(1)

- As we know black-box testing is focused on functions in HLT while white-box testing is focused on structures in LLT. But in reality, they should not be separated so much.
- The view of “Unit test or component test belongs to the range of white-box testing” and “System test belongs to the range of black-box testing” are not accurate. When we say black or white-box, we refer to the way of testing. In fact, both unit test and system test can be executed in both white-box and black-box way.
- Regarding UT, if the unit or component can be regarded as a relatively independent function (which is called “design functions” by W.E. Howden), the black-box test specification techniques such as BV, EC, CE(Cause-Effect), Combinatorial testing can be used to derive TCs, at the same time supplemented with white-box test techniques such as all kinds of test coverage methods.

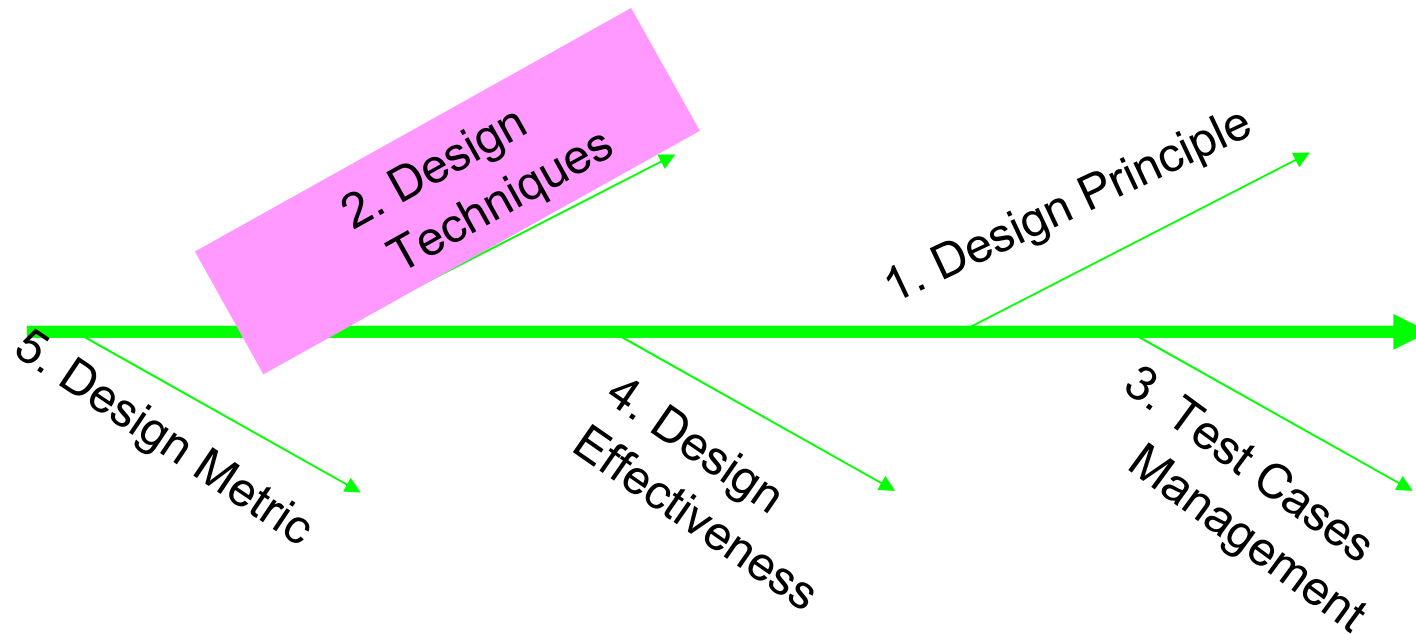
1.5 How to look on black-box testing and white-box testing?(2)

- Regarding system test, black-box method should be used if taking the system as a whole. However, white-box testing should also be taken seriously for at least the following two reasons: 1. The system to be tested like is often so complex that having some knowledge of the structure of the product is very helpful for finding some specific errors. 2. Developers often lack of such an ability of using white-box testing methods very well, so many faults will be leaked to HLT phase, while white-box testing method is often more capable and economical to find such errors.

2. Design Techniques

The usage of test specification techniques are lacking, including those basic test techniques commonly used worldwide.

Regarding the area of test specification techniques, people have too little knowledge to be able to do a good test design.



2.1 Common test design techniques

- Common test design techniques are widely used in many fields. These include EP (equivalence partitions), BV (boundary values), state based testing, decision tables and decision trees, combinatorial testing, orthogonal matrix (or orthogonal experimental design), ET (exploratory testing), etc.
- People don't grasp the usage of these techniques.
- May we ask ---

Will these techniques be suitable for our product?

Can test design be more effective if these techniques are used?

Is it possible to improve people's design ability if we think these techniques are useful to us?

2.2 Existing test design techniques

- The most popular test techniques used now are FI (function interaction) and TT (test type analysis).
- The problems of using FI and TT are that:
 - A large number of TCs are designed, which is far more beyond our test execution ability.
 - All TSs (test specifications) or all features are evenly treated when using FI and TT, making it difficult to apply RBT idea.

2.3 How to choose suitable test techniques?(1)

- No test technique is omnipotent. For each product, there exist some test techniques which are just suitable for it.
- The following questions addressed in <Software Testing Techniques> need to be answered:
 - Which techniques are more valuable and not?
 - Which are easy to apply, which are not?
 - Which find genuine faults, which are not?
- Besides, we also need to consider what test techniques are more suitable in each testing level (UT/IT/ST/BBIT/SDV/SIT/SVT/Beta), and on what granularity the TC should be designed. Do we really need to derive TCs based on each TS (TR -- TS --TC), which results in a large number of TCs ? Or can it be based on function level?

2.3 How to choose suitable test techniques?(2)

- Test design is one of the most intelligent work in test field. But it doesn't mean test techniques should be very complex to use. On the contrary, I believe that good test specification techniques should have good usability.
- Different test specification techniques have different effects in eliminating or avoiding risks of projects. But to put the theory into practice, we still have a long way to go.

2.4 The “Analyst” Role in Testing

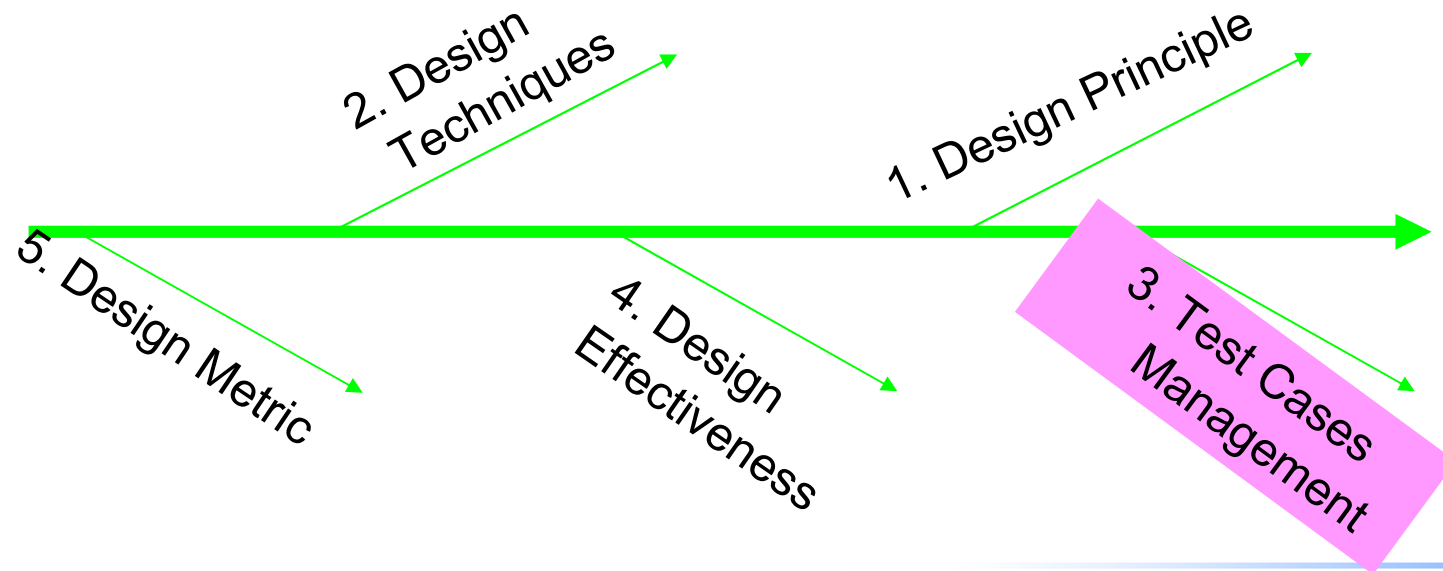
- Mike Smith in his <*Putting the ‘Analysis’ in a Test Analyst*> worried about that testers nowadays lack of analysis ability, which includes structural programming, system analysis, data analysis and process analysis. These abilities are essential for a good test design work.
- Mike also thought generally testers lack of training of above analysis abilities and even though some training is offered, the training often narrowly focused on simple separate test technique, an integrated approach of using test techniques can seldom be offered.
- Another very interesting viewpoint of Mike is that he prefers to call “test analysis” and “test design” rather than “test analysis & design”. The reason for this is that “test analysis” focuses on “What” issue (what measures and targets of success for testing) and “test design ” focuses on “How” issue (how those measures and targets will be achieved).

2.5 Informal testing

- Good test = formal test + informal test = good test techniques + people's experience
- At one hand, we need to learn how to select good, suitable test specification techniques; at the other hand, we need to learn how to exert people's experience to do a good informal testing.
- Although almost all testers are using informal testing method, they never use it in a good and efficient way.
- Informal testing doesn't mean the testing process should be informal and random. However, a formal and structured way of using informal testing techniques such as ET (exploratory testing) is very helpful for improving testing effectiveness.

3. Test cases management

The maintenance and update of test cases is always a headache for test managers. With the development of automatic generation of TC and test cases management tools, some people predict that test cases management will be an easy thing to do in the future, but not now at least. We have too many test cases designed yet not used, which put more stress to test cases management.



3.1 test tracing

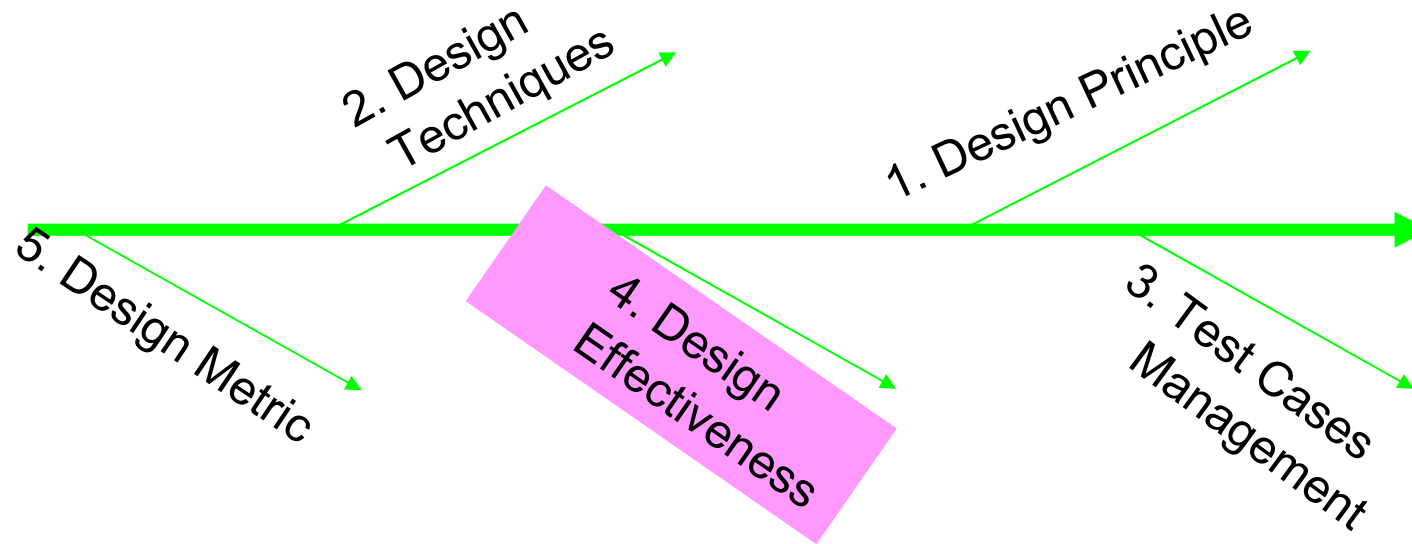
- Good test tracing system should easily trace the relationship of requirements, test cases and test results for different versions, so that all kinds of statistics about testing can be easily obtained.

3.2 test cases update

- < *Mythical Man-month* > mentioned that in software the only thing never changed is change. How does test design adapt to continuous changes?
- After test execution, how to guarantee test cases are rightly updated and supplemented?
- The storing method for TCs are various, including WORD, EXCEL, should we use the same way to store TCs?
- Manual TCs need to be converted to Automatic TCs, which is no good to TCs management.

4. Design Effectiveness

Are test cases effective in finding errors or in proving product's functions? How much effect do TCs take in testing process?



4.1 Ineffective TCs?

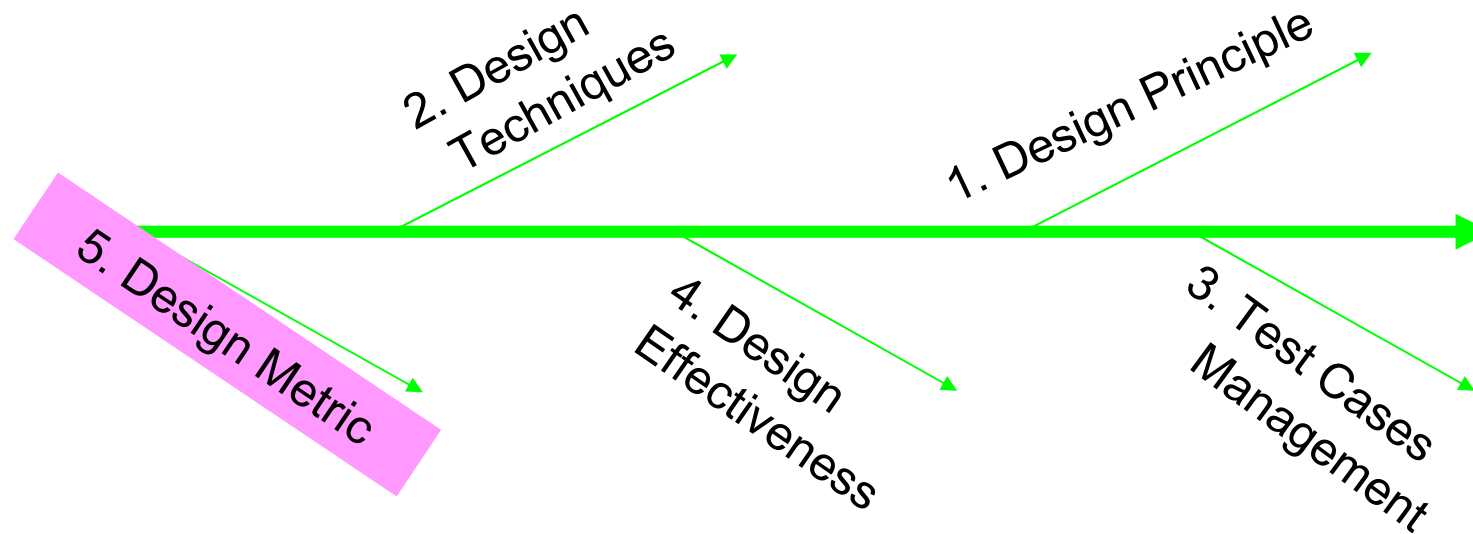
- Both LLT TCs and HLT TCs cover most of normal functions and scenarios, considering little of abnormal situations.
- The incompleteness of TCs results in low efficiency of TCs, which means many of defects are not found by TCs, but by testers' experience.

4.2 TC's writing

- Test cases are the carrier to transfer the knowledge and idea of test designers to test executors. The accuracy of test case's writing has much influence on the understanding of the TC. People may have different understanding of the same TC due to its natural writing language, accordingly, different test results can be produced for the same TC.
- Usually, review of TCs are less effective than other design documents, due to the large number of TCs and non-logical format of TCs.
- How many test executors are following TCs completely in their testing process? Do they need to do so?

5. Design Metric

The metric of test design is another hard thing to do.



5.1 Measurable test design?

- The number of TCs is not enough to measure the size of test design, because of the differentia of TC's granularity. Similarly, the number of executed TCs is not enough to measure the work by testers.
- Unless testing design result can be expressed in such a way that reviewers can point out the writer's thinking at the first glance, can it be easy to evaluate the quality of test designer's work.
- Another way to evaluate test design is to see how much test coverage can be reached by TCs. Among them, statement coverage is the weakest method to measure coverage, while some still demands statement coverage of 100%. In fact, to reach 100% statement coverage, people often spend unnecessary time on coding and test designing, which is not always meaningful to test effectiveness.

Part III An Example

The Pilot

- I have chosen one algorithm, one of the new features, to do a test design pilot – using MFQ&PPDCS method.
- First, I design test cases for LLT and HLT.
- Second, I compare the result with the test cases designed by the real designers and testers.

Comparison

Item	Test design by real designer and tester	Test design by me
1. Test cases	62UTC + 12 ITC + 12 STC (ITC and STC are actually same.); Less than 10% negative TCs; No high level black-box scenario.	>95 normal TCs + n abnormal TCs + More than 10 black-box test scenarios.
2. Test coverage	Mainly focus two of the components. Only considering basic function verification.	Cover 4 components. Considering each test point for each component. 2-way combinatorial testing.
3. Test techniques	No obviously test techniques used, mainly based on experience.	Decision table, decision tree, EP, BV, Pairwise, flow graph testing (branch coverage/MCDC) + Experience
4. Test result	0 defect after UT.	5 defects before any execution.

The importance of test design

- As we all know, complete testing is impossible, so testers must do some choices, which is just the essential part of test design. “Testing, however creative and seemingly complete, cannot guarantee the absence of all errors.” Good test design just like what Myers pointed out in *<The Art of Software Testing>* that “A good test case is one that has a high probability of detecting an as yet undiscovered error.”
- As Hans Österberg said, **“The most urgent need is a couple of skilled test designers!”**

欢迎交流！

邵晓梅

testcomt@gmail.com

新浪微博：MiniStarClub001-邵晓梅

博客：www.taixiaomei.com