

基于风险的测试(RBT)

wangwei1237

一个笑话

QA工程师走进酒吧，要了一杯啤酒，

要了0杯啤酒，

要了999999999杯啤酒，

要了一只蜥蜴，

要了-1杯啤酒，

要了一个noaasdfnasdf,

酒保从容应对，QA工程师很满意。

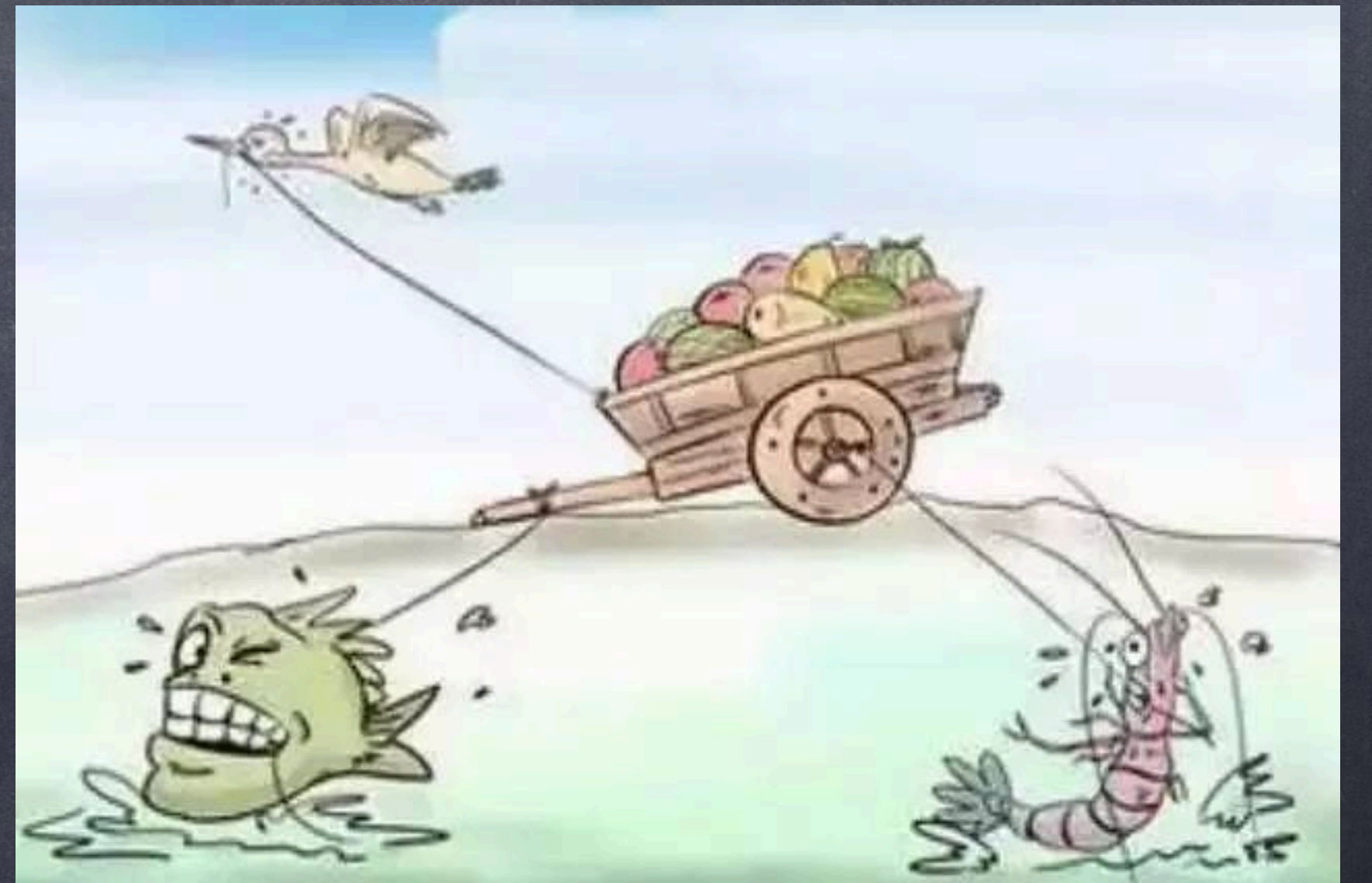
QA工程师走进了同样的酒吧，用999999999个机器人同时要了999999999杯啤酒（一个机器人点一杯），

酒保从容应对，QA工程师也很满意。

一名顾客来到了同样的酒吧，问厕所在哪，酒吧顿时燃起了大火，然后整个建筑坍塌了。

1. 什么是风险

- 风险首先是某些不确定性事件的发生
- 它们会对项目的成功产生正面或者负面的影响



2. 风险类型

- 正向风险 (机会/opportunities)

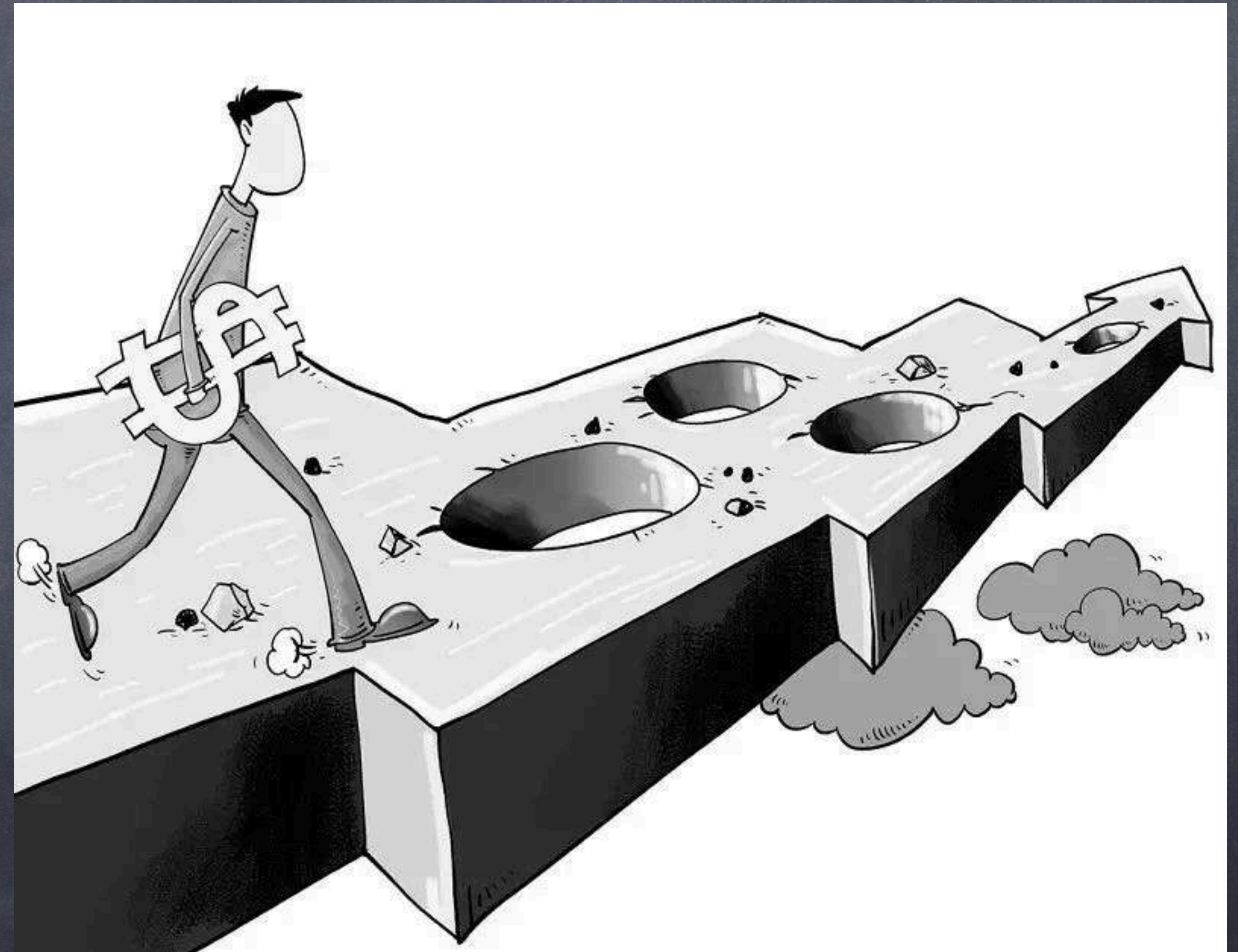
- 投资

- 业务的合并与拆分

- 负向风险 (威胁/threats)

- 资产坏账

- bug太多



3. 基于风险的测试 (RBT)



风险 = 产品某个功能不能工作的概率 * 预期的损失



功能的使用频率 * 出bug的概率

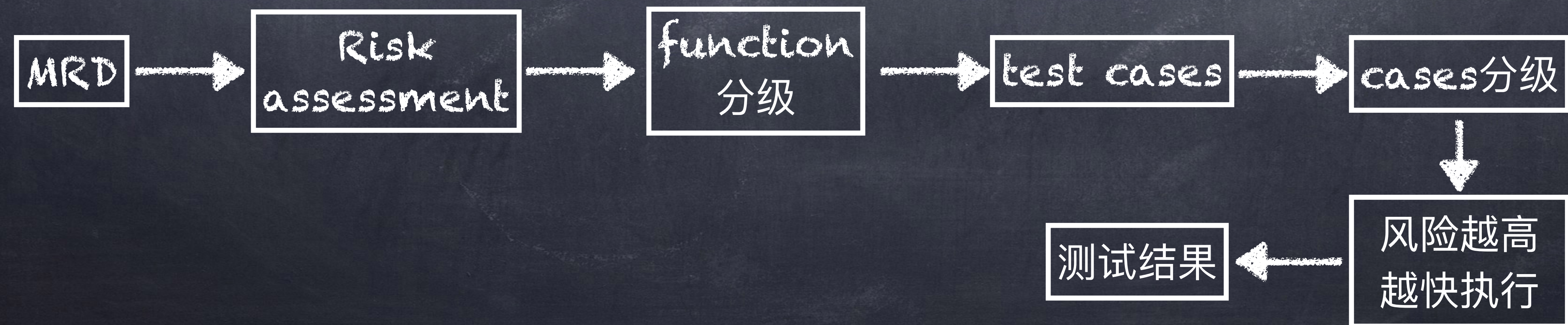


3. 基于风险的测试 (RBT)

- 目前的case设计与case优先级



- RBT的case设计与case优先级



3. 基于风险的测试 (RBT)

QA工程师走进酒吧，要了一杯啤酒，

要了0杯啤酒，

要了999999999杯啤酒，

要了一只蜥蜴，

要了-1杯啤酒，

要了一个noadfnsdf,

酒保从容应对，QA工程师很满意。

QA工程师走进了同样的酒吧，用999999999个机器人同时要了999999999杯啤酒（一个机器人点一杯），

酒保从容应对，QA工程师也很满意。

一名顾客来到了同样的酒吧，问厕所在哪，酒吧顿时燃起了大火，然后整个建筑坍塌了。



3. 基于风险的测试 (RBT)

- RBT强如下的几个点

- 测试过程不是面面俱到，而是在一定的限定条件下，保证项目风险最小

$$\min_{r, \omega, b} \frac{1}{2} \|\omega\|^2 \quad \text{s.t.} \quad y^{(i)}(\omega^T x^{(i)} + b) > 1, \quad i=1 \dots m$$

- 在case编写之前做好风险评估
- 优先执行风险高的case

4. How to do RBT

前期调研（技术调研，功能调研）

- 了解产品功能，用户操作路径
- 了解技术架构，涉及模块，模块交互.....
- 了解功能在架构上的映射和实现

评估风险

- 风险来自变化之中
- 风险来自异常~~与~~边界~~与~~交互
- 风险来自用户使用频率高的功能
- 风险来自影响面大的功能
- 风险来自自己不可控的地方

4. How to do RBT

- As a QA Engineer, you should be familiar with
 - all the touch points around the functions you are responsible for testing,
 - how they interact with each another as well as other applications in your overall architecture,
 - and all the paths the user will take into, through, and out of your area.

This will allow you to easily identify and isolate the important areas for any given change, and allow you to test just around those areas.

4. How to do RBT

• Some examples:

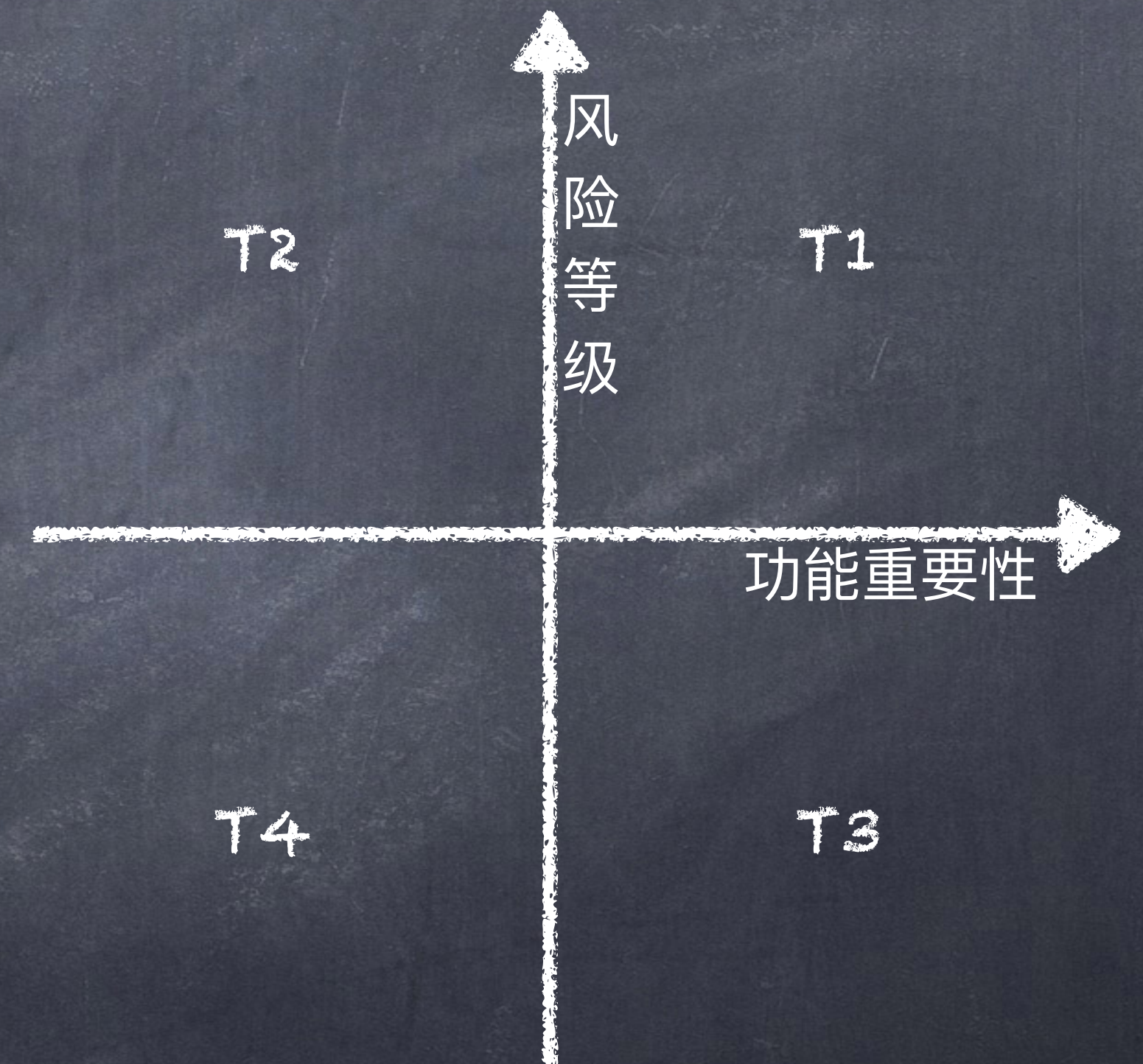
• 接口升级

• 重构类项目

• wise/NA兼容性测试

5. 怎么判断是否做到了RBT

- case的分级是否按照风险分级
- 在测试过程中, p0, p1问题是否在测试前期发现
- 在测试过程中, p0, p1问题是否在测试后期还有出现



6. Q&A

Question?