

大数定律重塑软件测试基础

Re-Define the base of Software Testing by “Law Of Large Numbers”

陈耿



- 软件测试的基础问题
 1. 软件测试的覆盖率-无根之木
 2. 由此产生多种问题
- 核心问题：
 1. 结果有多准
 2. 该测多少才准
- Hoeffding不等式
- 运用Hoeffding不等式解决上述问题的方法
- 为何可以应用此方法：
 - 软件黑盒测试与随机抽样结果
 - 软件黑盒测试是二元结果
 - 置信区间，置信水平，采样量
 - Hoeffding不等式应用问题
 - 局部采样与整体采样
 - N过大问题
 - 结论

软件测试的基础问题



白盒测试
覆盖率

要求覆盖的代码/分支/
决策

总的代码/分支/决策

%

100%覆盖了能保证没
有问题吗？

软件测试的基础问题



黑盒测试
覆盖率

被测试到的测试覆盖项

%

总的测试覆盖项

100%覆盖了能保证没有
问题吗？

软件测试的基础问题



- 软件测试的原理：
 - 测试不可穷尽
 - 测试只能告诉你有缺陷，而不能告诉你无缺陷

白盒测试
覆盖率

黑盒测试
覆盖率

100% 覆盖说明不了问题！

测试覆盖率是“无根之木”

软件测试的基础问题



- 由此产生的各种问题

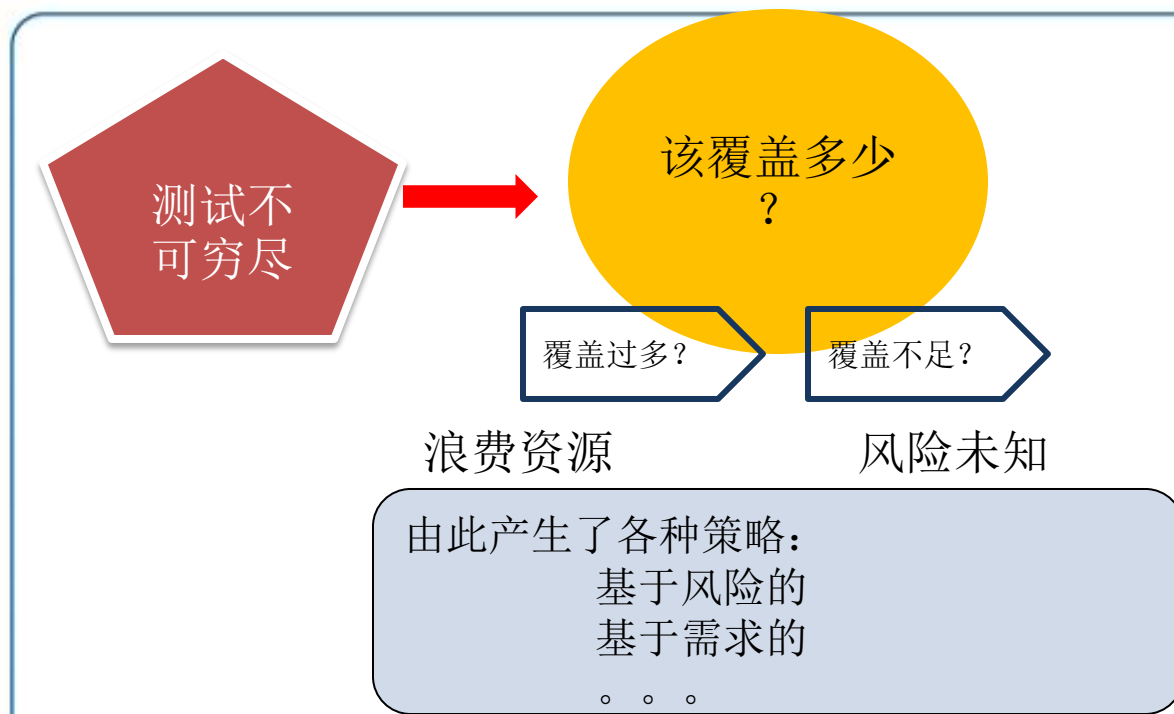
测试就是找bug，只有找到bug才能体现测试的价值

测试预算可大可小，有钱多测点没钱少测点。实在没钱不测也行

测试完成了，但上市/线后遇到了缺陷，都是测试不足

目前的测试理论回答了如何进行测试的问题，但无法回答测试效果的问题，以及测试投入与测试效果的关系问题

核心问题



- 但都无法回答核心问题

- 结果有多可信（风险发生的概率）

测试效果的问题

- 要投入多少才能获得可信的结果（要花多少测试预算）

测试投入与测试效果的关系问题

解决核心问题的办法-Hoeffding不等式



概率论->大数定律-> Hoeffding不等式（二元结果）

$$\mathbb{P}(|\overline{X} - \mathbb{E}[\overline{X}]| \geq t) \leq 2e^{-2nt^2}$$

$\overline{X}$ 采样结果的平均（采样结果仅2个，1或0）

$\mathbb{E}[\overline{X}]$ 期望结果的平均-即真实的结果

t 采样结果平均与真实结果的偏差

n 采样量

含义：采样结果与真实结果之间的偏差超过某个阈值的概率小于右边的这个**上限**，此上限**仅**与偏差和采样量相关

置信区间，置信水平，采样量



- 置信区间 $\pm t$ (Confidence Interval)



- 置信水平 $1-\alpha$ (Confidence Level), 真实结果落在置信区间内的概率
- 采样量-在黑盒测试中即检查点的数量

运用Hoeffding不等式解决上述问题的方法



- 结果有多可信（风险发生的概率）测试效果的问题
- 要投入多少才能获得可信的结果（要花多少测试预算）测试投入与测试效果的关系问题

$$\text{令 } \alpha = \mathbb{P}(\bar{X} \notin [\mathbb{E}[\bar{X}] - t, \mathbb{E}[\bar{X}] + t]) \leq 2e^{-2nt^2}$$

$$n \geq -\frac{\ln(\alpha/2)}{2t^2}$$

- 要更可信—即 t 要小，则 n 会更大
- 关键概念

1-a 置信水平(Confidence Level)

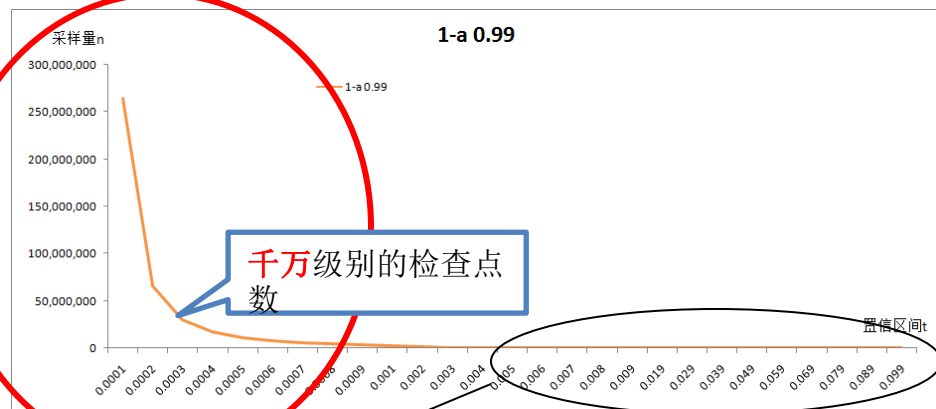
t 置信区间 (Confidence Interval)

©2015 CSTQB www.cstqb.cn

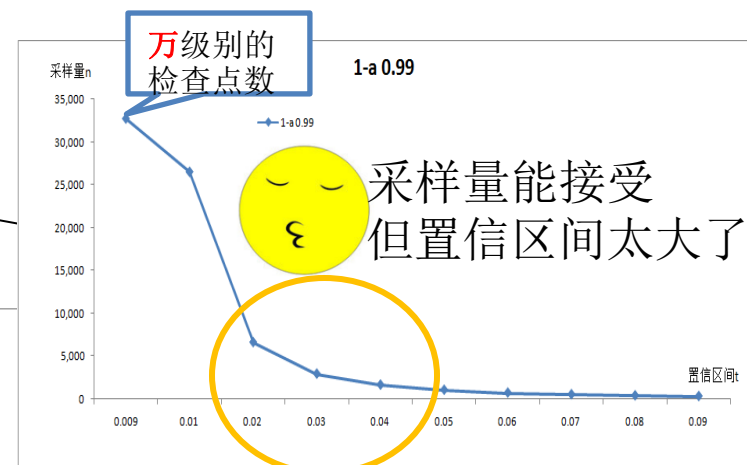
运用Hoeffding不等式解决上述问题的方法-a, n与t



- 置信水平为99%时，采样量与置信区间的关系

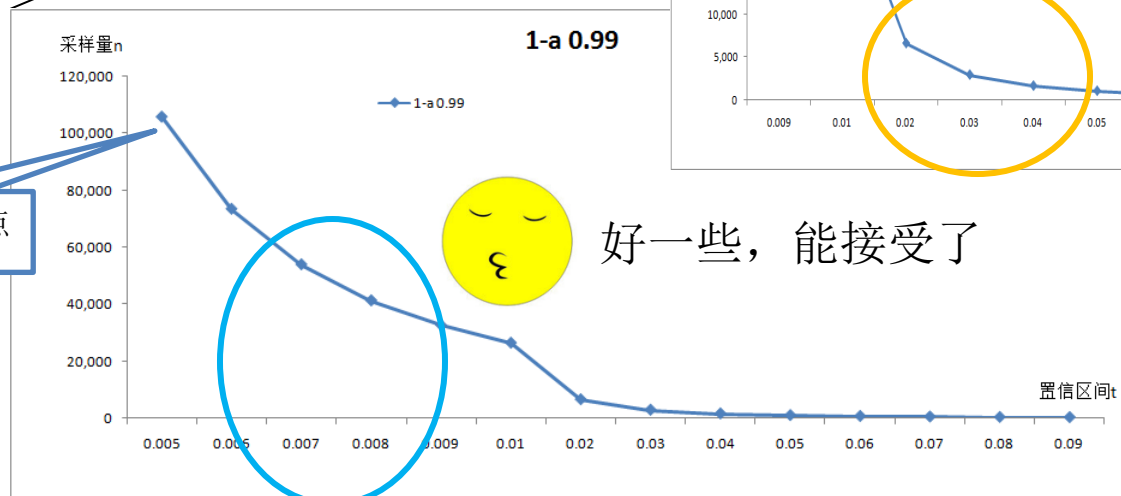


软件测试中不能直接套用！



太多了！

十万级别的检查点数



运用Hoeffding不等式解决上述问题的方法



- 软件测试中能运用Hoeffding不等式的关键要素：
 - 软件测试，特别是黑盒测试约等于随机抽样（后文论证）
 - 软件测试的结果是二元结果（通过/失败）
 - 软件测试中的“等价类现象”即一个检查点可代表多个检查点
 - 引入“平均等价数(m)”概念和“非等价概率(p_m)”概念

运用Hoeffding不等式解决上述问题的方法



- 利用平均等价数和不等价概率来修正 hoeffding不等式:

$$n_s \geq - \frac{\ln(\alpha / 2)}{2t^2 m p_m}$$

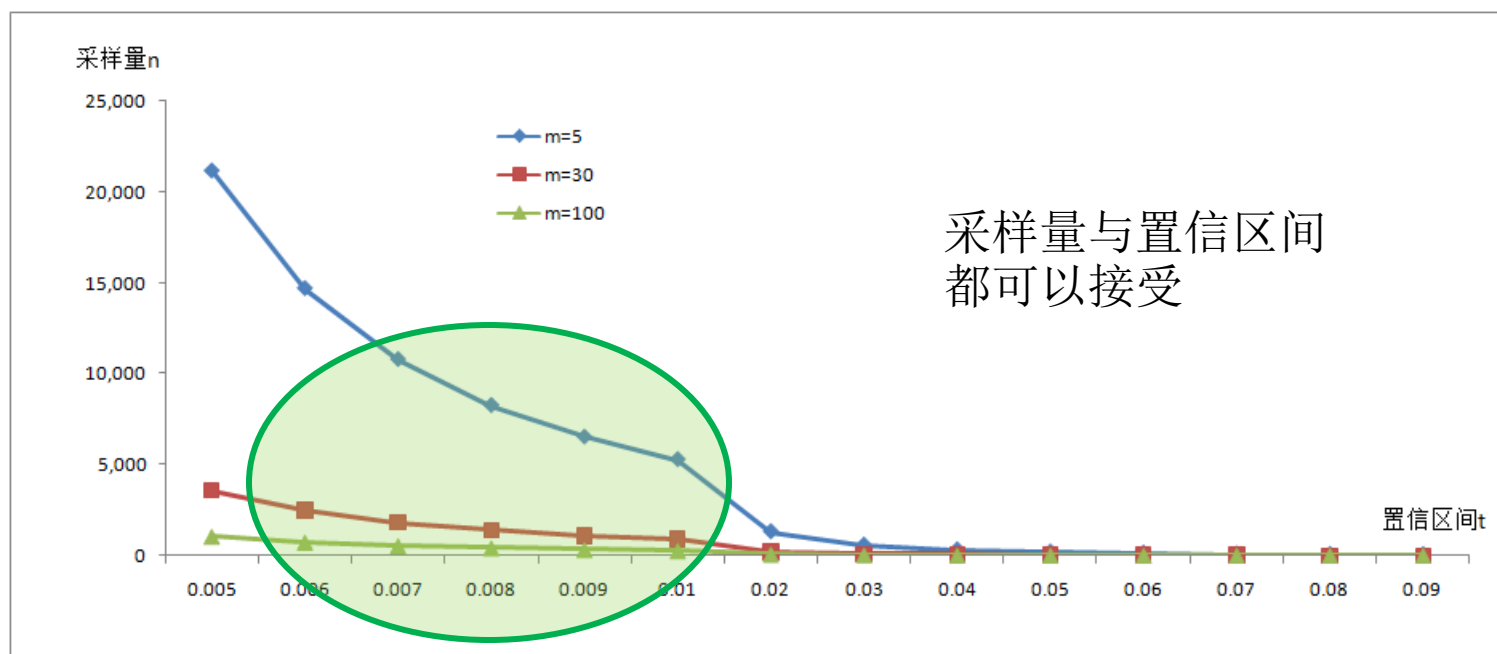
(软件测试中的采样量)

- 平均等价数(m), 是根据系统的基本情况, 估算的“安全”的等价的量。即一个检查点能够相当于多少个检查点
- P_m 是这样的等价情况成立的概率。实际软件来说, m设置为1~10甚至是30都可以认为 P_m 非常大接近99.99%

运用Hoeffding不等式解决上述问题的方法



- 运用上面公式可得, m 为不同值时需要的采样量（测试检查点的量）



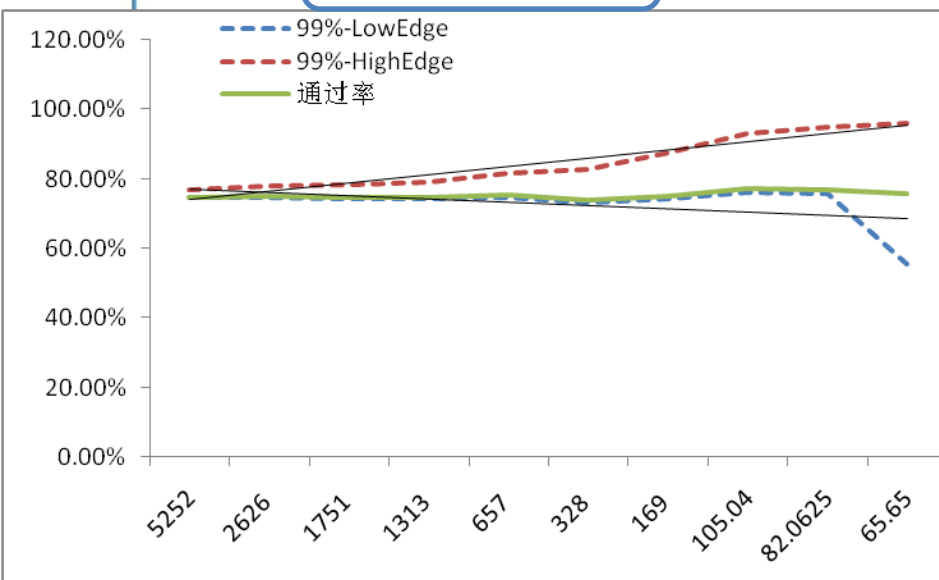
一般软件测试中，一个检查点可以认为等同于至少5个相邻的检查点，应该不是大问题，即 P_m 99%以上

真实项目数据

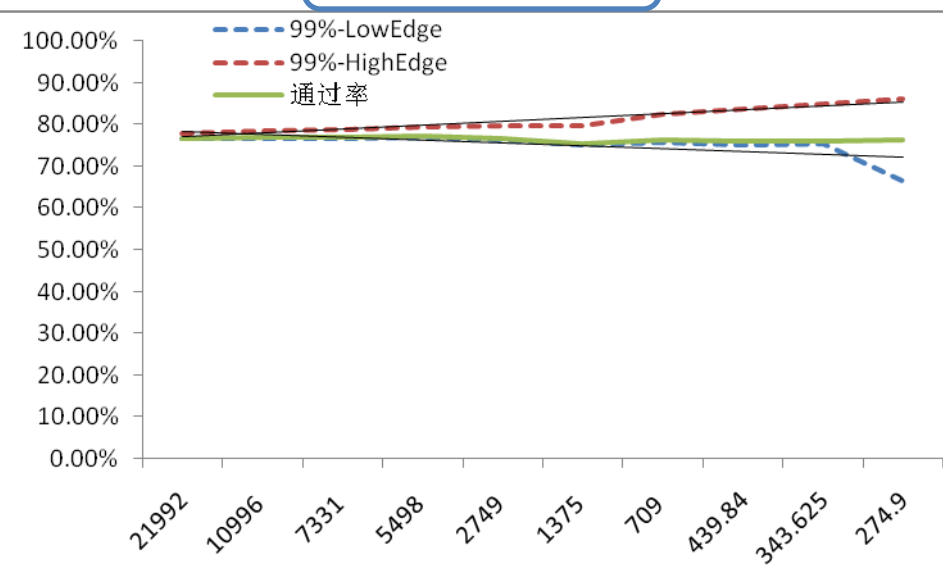


- 18个release
- 多个产品
- 都在Hoeffding不等式给出的上限内

五千个用例



两万个用例

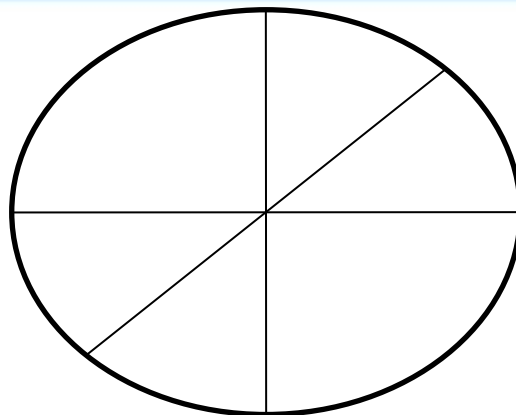


真实项目数据



验证方法:

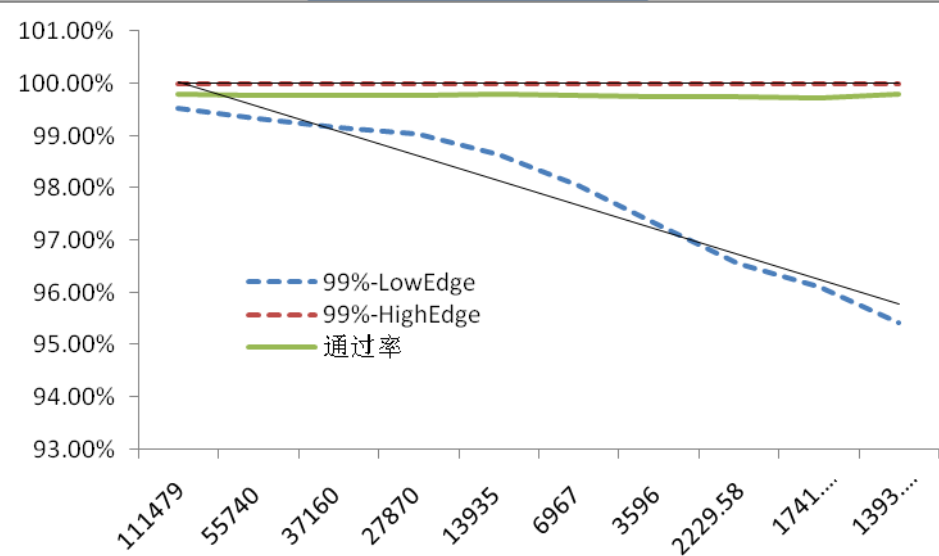
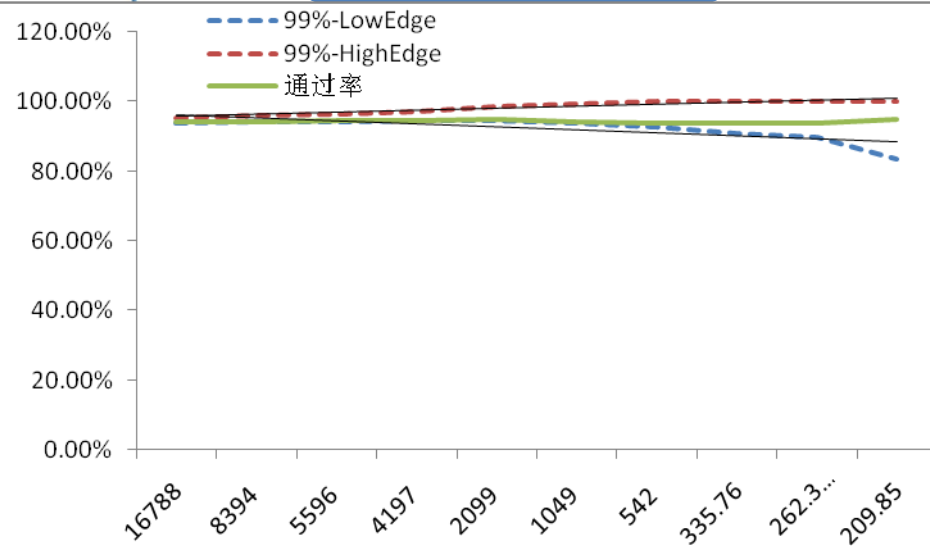
- 1.完整测试集合的通过率
- 2.随机取半数检查点, 计算通过率, 置信区间
- 3.重复第2步



减半随机取测试点相当于当时测试只测试了减半数量的用例的得到的结果
观察增加检查点时, 通过率是否仍在上下限之内

一万六千个用例

十一万个用例



有用的推论



- 测试早期通过率低，通过增加覆盖面来提供更精确的信息意义不大
->主要力量应放在发现缺陷上
- 测试晚期通过率高，应该通过增加覆盖面来提供更精确的信息以支撑发布/上线决策
- 发布决策应该平衡风险发生的概率与继续测试的代价

运用Hoeffding不等式解决上述问题的方法-例1



- Web 服务, 无人身伤害风险, 一定经济风险, 有补救, 重做机制
 - 置信区间=0.02; 置信水平99% 可计算出检查点的量为六千左右
含义是: 当检查点99.9%通过时, 其真实系统最低的正常概率为97.9%, 并且发生低于这个最低正常概率的可能性仅1%。
 - 如果置信区间=0.008; 置信水平为99%, 可计算出检查点的量为4万多也在可承受范围内

采样量 (千用例)	测试成本 (万元)	置信水平	置信区间	测试通过率	可能最低通过率	风险概率	风险影响 (万元)	风险折现 (万元)
6	0.2*6 ≈ 1.2	99%	0.02	75.9%	73.9%	$\leq 0.241 + 0.261 * 0.01 = 0.2436$	1000 该继续测试否?	243
6	0.2*6 ≈ 1.2	99%	0.02	99.9%	97.9%	$\leq 0.001 + 0.021 * 0.01 = 0.00121$	1000 该继续测试否?	1.21
40	0.2* 40 ≈ 20	99%	0.008	99.9%	99.1%	$\leq 0.001 + 0.009 * 0.01 = 0.00109$	1000 该继续测试否?	1.09

运用Hoeffding不等式解决上述问题的方法-例2



- 网络影音系统，第一版开发完毕,内测,准备了500个用例,1500个检查点,其中通过率为75%
 - 置信水平为99%，可计算出置信区间为0.042。即真实系统正常工作的概率下限是70.8%，非常低不能发布；而到后期，通过率为99%，则正常工作的概率下限是95.7%，按风险来平衡的话，可以发布。

采样量 (千用例)	测试成本 (万元)	置信 水平	置信 区间	测试通 过率	可能最低 通过率	风险概率	风险影响 (万元)	风险折现 (万元)
1.5	0.2×1.5 ≈ 0.3	99%	0.042	75%	70.8%	$\leq 0.25 + 0.292 \times 0.01 = 0.25292$	500 该继续测试否?	127
1.5	0.2×6 ≈ 0.3	99%	0.042	99.9%	95.7%	$\leq 0.001 + 0.043 \times 0.01 = 0.00143$	500 该继续测试否?	0.71
26	0.2×26 ≈ 13	99%	0.01	99.9%	99.1%	$\leq 0.001 + 0.009 \times 0.01 = 0.00109$	500 该继续测试否?	0.55

结论



- 软件测试覆盖率应该用置信区间+置信水平来度量
 - 即测试用例的覆盖使得有多大的可能性，测试结果能落在什么范围内
- 度量的关系应该依据Hoeffding不等式

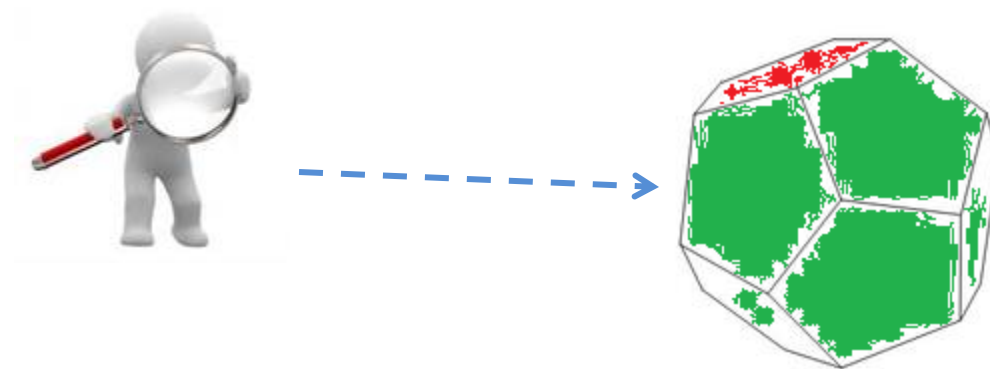
- 软件质量信息与测试
- 核心问题：
 1. 结果有多准
 2. 该测多少才准
- Hoeffding不等式
- 运用Hoeffding不等式解决上述问题的方法
- 为何可以应用此方法：
 - 软件黑盒测试与随机抽样结果
 - 软件黑盒测试是二元结果
 - 置信区间，置信水平，采样量
 - Hoeffding不等式应用问题
 - 局部采样与整体采样
 - N过大问题
 - 结论

软件黑盒测试与随机抽样结果



- 黑盒测试:

依据分析测试基础文档来选择测试条件、测试用例或测试数据的技术。
不需要使用任何关于被测组件或系统的内部结构信息。



- 面-相当于“测试基础文档”，因为正是“测试基础文档”勾勒了系统的边界
- 而根据此边界来选择测试条件等即设置检查点来观察系统运行的结果

软件黑盒测试与随机抽样结果

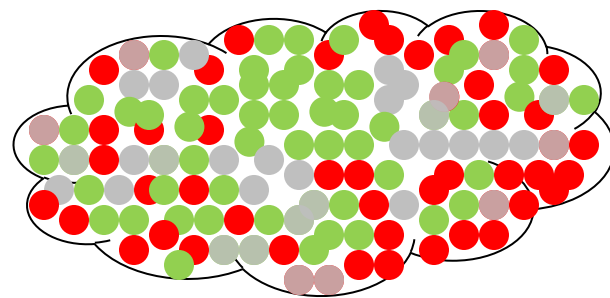


- 随机过程：
 - 任意扔骰子

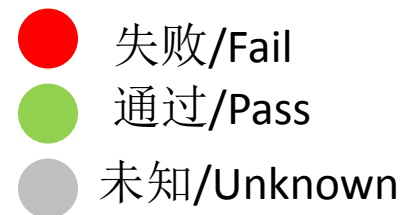


- 不关心骰子内部结构（虽然内部结构会影响结果）
- 反复试验，观察结果

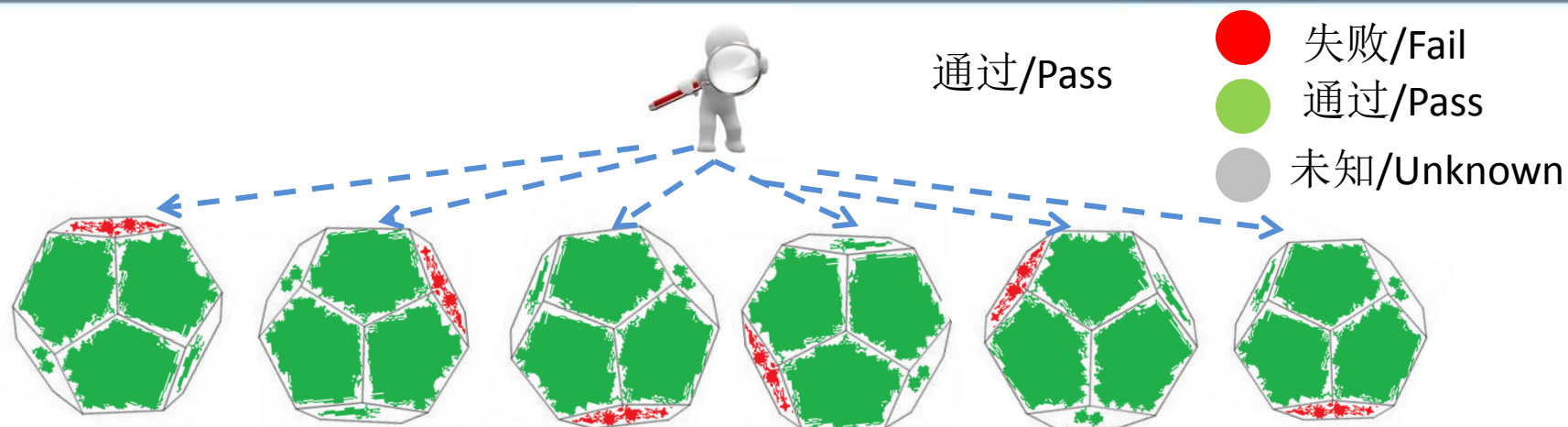
- 黑盒测试：
 - 从系统界面采样检查



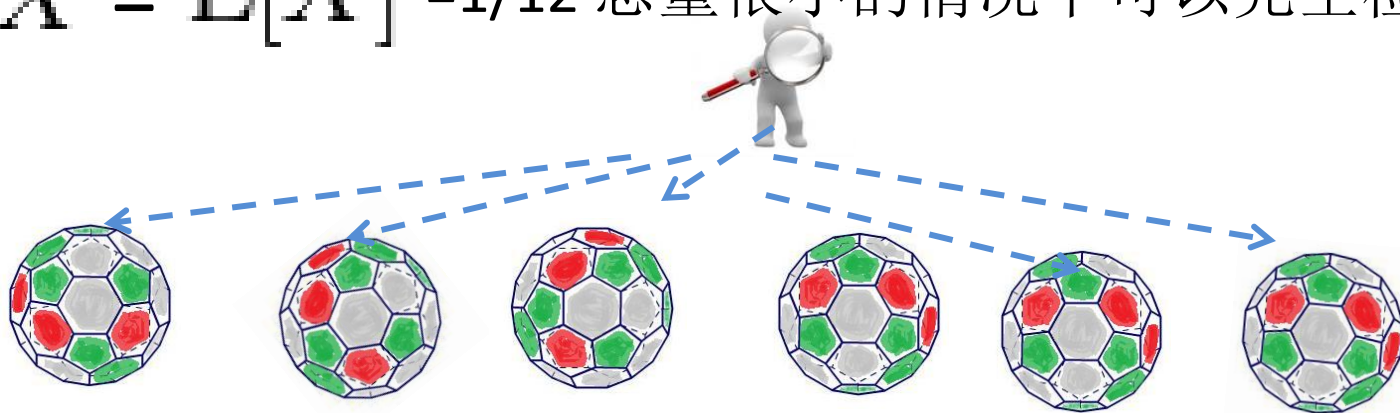
- 不关心系统内部结构
- 反复试验，观察检查点的结果



软件黑盒测试与随机抽样结果



$\bar{X} = E[\bar{X}] = 1/12$ 总量很小的情况下可以完全检查



$\bar{X} \rightarrow E[\bar{X}]$ 总量很大时是采样,采样越多,结果越趋近真实

软件黑盒测试是二元结果



- 软件测试结果：
 - 通过/Pass—符合预期
 - 失败/Fail—任何其它情况

- 失败/Fail
- 通过/Pass
- 未知/Unknown

- 二元结果：
 - 非0即1 (fail=0, pass=1)

置信区间，置信水平，采样量



- 置信区间 (Confidence Interval)

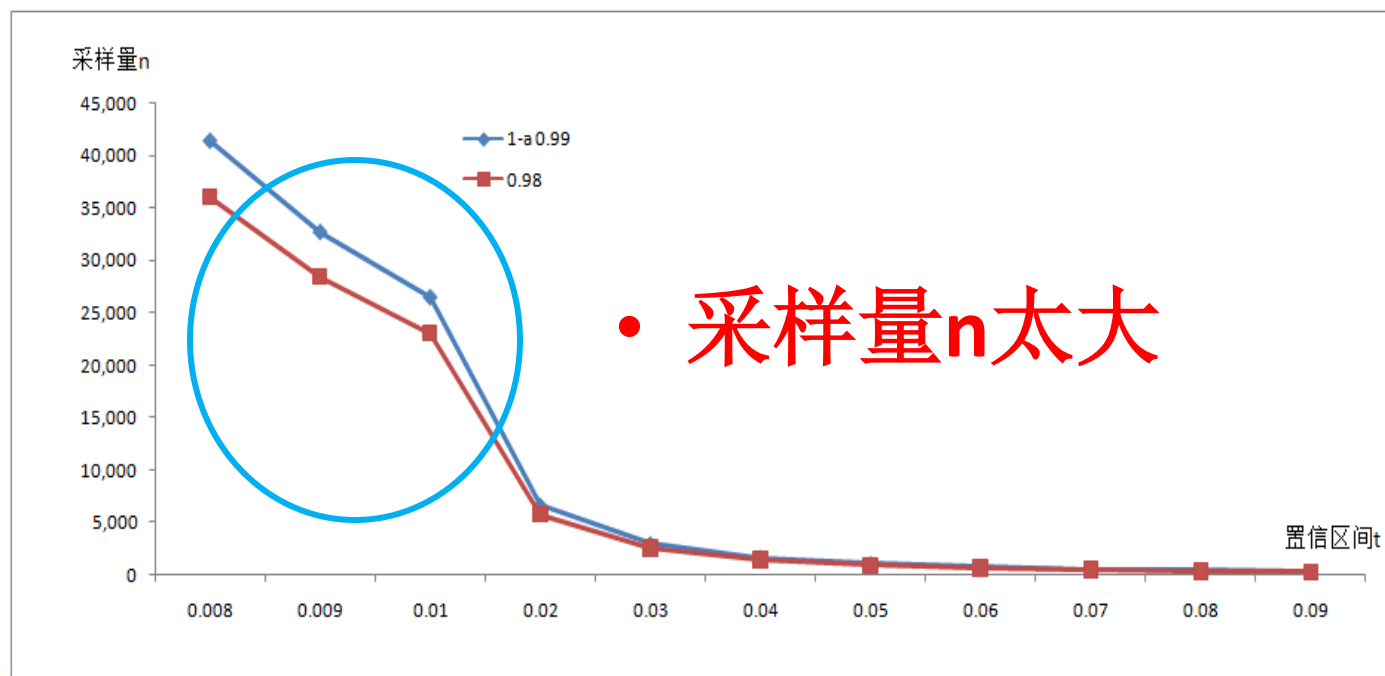


- 置信水平(Confidence Level)，真实结果落在置信区间内的概率
- 采样量-在黑盒测试中即检查点的数量

Hoeffding不等式应用问题



- 根据“测试基础”文档采样能等于“系统采样”吗？



- 采样量n太大

局部采样与整体采样

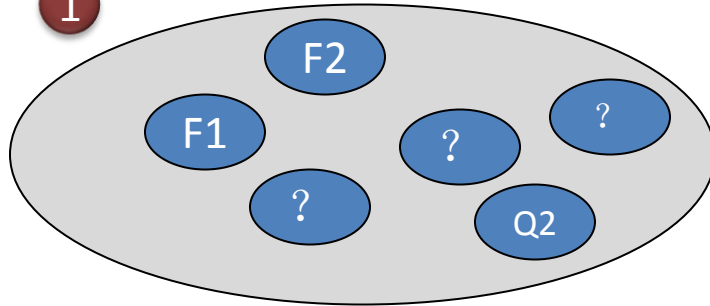


- 即：根据“测试基础”文档采样能等于“系统采样”吗？
 - 软件系统看不见，摸不着
 - 对软件系统的检查有很多不同的视点，如功能视点，质量属性视点等。
 - 这些视点互有不同，但又互相交错
 - 形成N个纬度的实体。
 - 测试实践是对这些“视点”进行观察
- 多个视点观察结果的总合“等于”系统整体吗？

局部采样与整体采样-分析



1



系统已知部分为蓝圈部分
其他部分未知
对F1,F2,Q2进行抽样检查

2

F1	F2	F3
F4	F5	F6

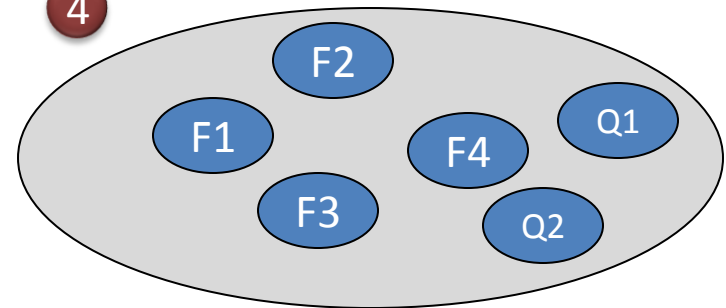
系统完全由蓝框组成
无未知
对F和Q进行均匀采样

3

F1	F2	F3
F4	F5	F6

系统完全由蓝框组成
无未知
对F和Q进行非均匀采样

4



系统已知部分为F1...
其他部分未知
对F1 ...进行抽样检查

局部采样与整体采样-分析



1

对整体无保证
只能对被采样部分得出结论
即“部分系统”

2

F1	F2	F3
F4	F5	F6

$$F_s = \sum_{i=1}^6 F_i$$

要获得 F_s ,得逐个对 F_i 进行采样获得,符合“分步骤”获得的特征。

所以 $P_s = P_1 * P_2 * P_3 * P_4 * P_5 * P_6$, 即:

$$P_s \leq \prod_{i=1}^6 e^{-2t^2n} = e^{-6 \bullet 2t^2n}$$

情形2时, 系统概率上限绑定在Hoeffding上限

局部采样与整体采样-分析



3

F1	F2	F3
F4	F5	F6

$$F_s = \sum_{i=1}^6 F_i$$

要获得 F_s ,得逐个对 F_i 进行采样获得,仍符合“分步骤”获得的特征。

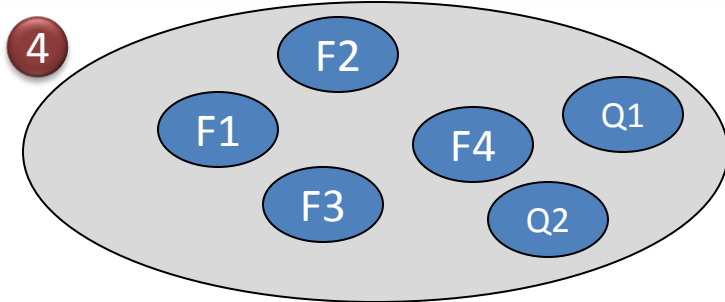
所以 $P_s = P_1 * P_2 * P_3 * P_4 * P_5 * P_6$, 即:

$$P_s \leq \prod_{i=1}^6 e^{-2t^2 n_i} = e^{-2t^2 \sum_{j=1}^6 n_j} = e^{-2t^2 N}$$

N为各个方面采样数的总和，即系统总采样数

情形3时，系统概率上限仍绑定在Hoeffding上限

局部采样与整体采样-分析



$$F_s \approx \sum_{i=1}^6 F_i \quad ?$$

关键问题是是否能认为F整体近似等于各Fi之和。如果可以，则可以利用2,3的结论。

事实上，对系统整体来说，能够被外界感知，或者说对外界产生作用和影响的是系统的边界。可以概括性的认为系统的边界由功能，交互界面，质量属性组成。所以可以用功能和质量属性的矩阵来组成系统，使其类似于的情况

这样的话就解决了4的问题

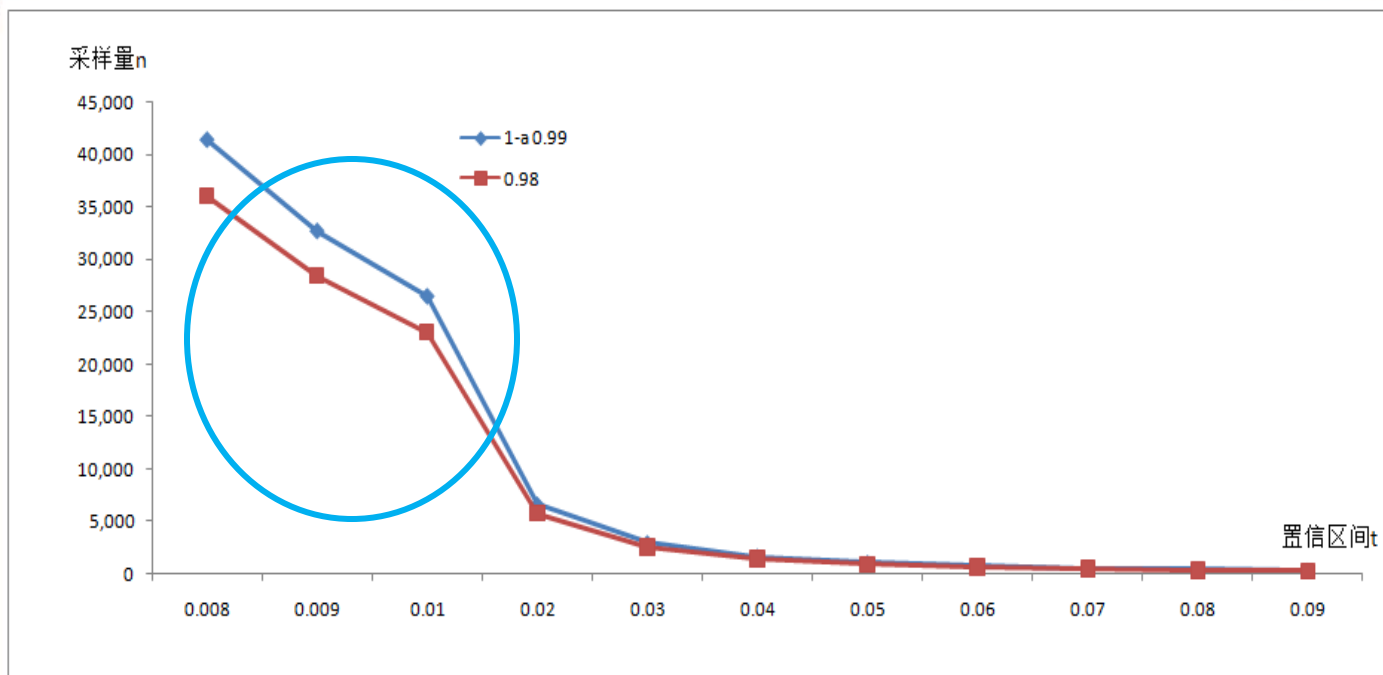
$$P_s \leq e^{-2t^2 N}$$

N为各个方面采样数的总和，即系统总采样数

当对“能感知，能影响”的主要方面都进行覆盖时，可以近似认为F整体等于各Fi之和

情形4时，系统概率上限仍可绑定在Hoeffding上限

N过大问题



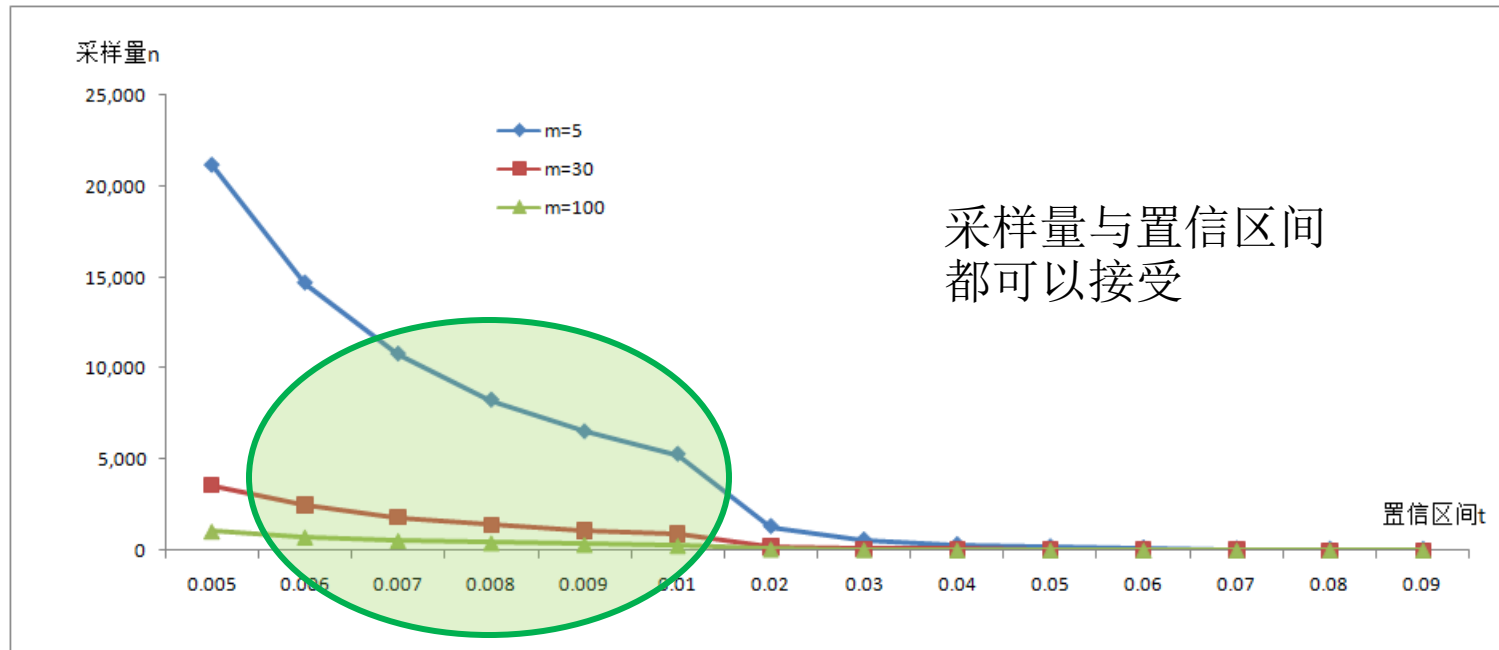
- 软件系统的特殊性：
 - 逻辑和计算是确定的
 - 所以有“等价类”的概念

N过大问题



- 利用“等价类”的概念减少采样量
- 可从上限逐步下推：
 - 能否二分？
 - 能否五分？
 - 能否十分？
 - 能否百分？
- 实际上从实践经验出发，一般软件系统，平均等价数设置为3~5是非常安全的。
- 因为大多数输入/参数，环境等，在软件中的处理是走同样逻辑的

N过大问题



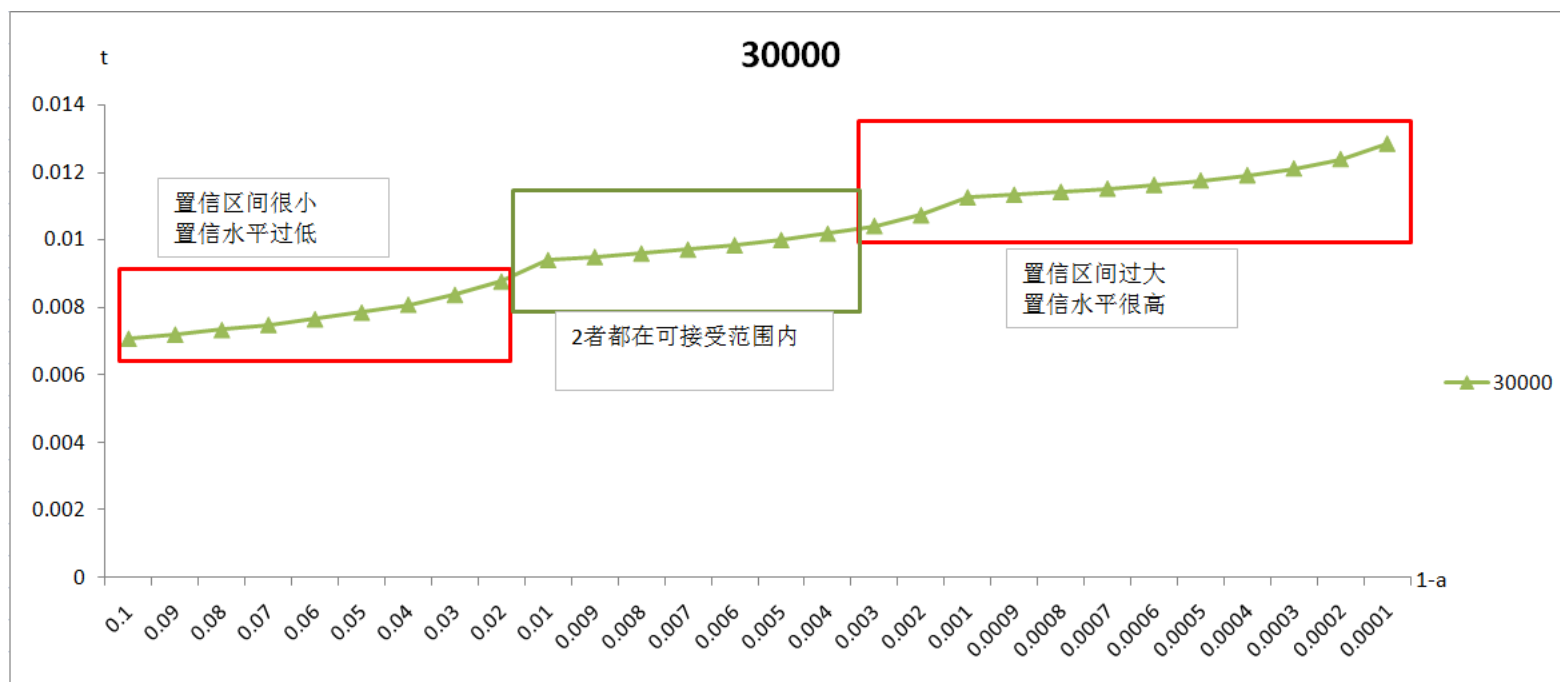
一般软件测试中，一个检查点可以认为等同于至少5个相邻的检查点，应该不是大问题，即 P_m 99%以上

可根据项目实践积累经验参数

结论



- 通过“平均等价数”修正，可利用Hoeffding不等式计算采样量
- 科学的得到黑盒测试覆盖的指标（黑盒测试覆盖率的数学基础）
- 科学的度量工作量，预算以及与风险进行平衡
- 对应一个采样总数，概率和偏差曲线的含义和作用





Thank you
ISTQB®让测试更专业

软件质量信息与测试

- ISTQB 基础级大纲:

- 测试可以有如下目标:

- 发现缺陷;
 - 增加对质量的信心;
 - 为决策提供信息;
 - 预防缺陷。



为了达到这些目标
软件测试的活动有

测试策略
/计划

测试分析/
设计

测试执行

结果汇总
与报告

确定覆盖要求
覆盖要求与资源投入平衡

依据覆盖要求，完成各项活动