

Ch.8 Testing & QA

2017.6

§ 8.0 BUGs



Funny Bugs



Funny Bugs



Costly Bugs



Ariane5
3.7亿美金的大烟花
copy了上一代的代码

Costly Bugs



聪明人留下的错

> 5000 亿!, \$

Fatal Bug



**宰赫兰反导系统拦截侯
赛因飞毛腿导弹失败**

0.33s

**空速达4.2马赫（每秒
1.5公里）**

**炸死28个美国士兵，
炸伤100多人**

目录

- § 8.1 基本认知
- § 8.2 有效的测试过程
- § 8.3 测试工作的专业性

主要概念

- Software Testing
- Quality Assurance

目标

- 找到bug → 检查是否合格的找茬游戏
- 找到bug并且将严重的bug fix掉 → 确保软件“每个角度”都满足要求

Synonym

Defect--缺陷

Variance--偏差

Fault--故障

Failure--失败

Problem--问题

Inconsistency--不一致

Error--错误

Feature--特性

Anomaly--异常（反常）

Exception--异常（不正常）

测试一个水杯



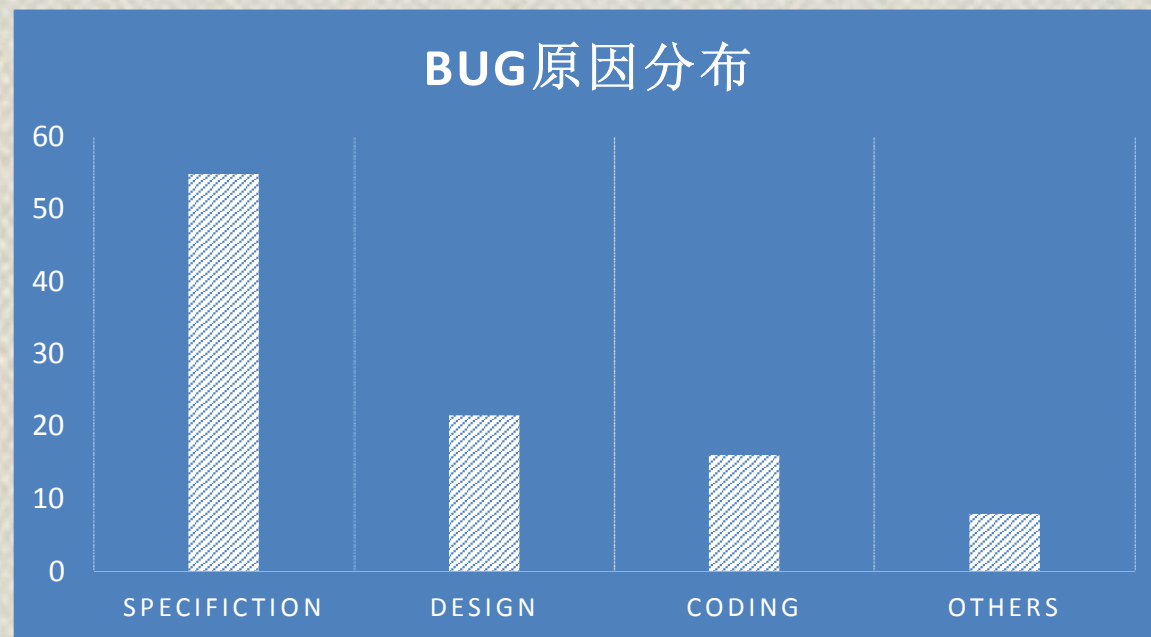
也是水杯



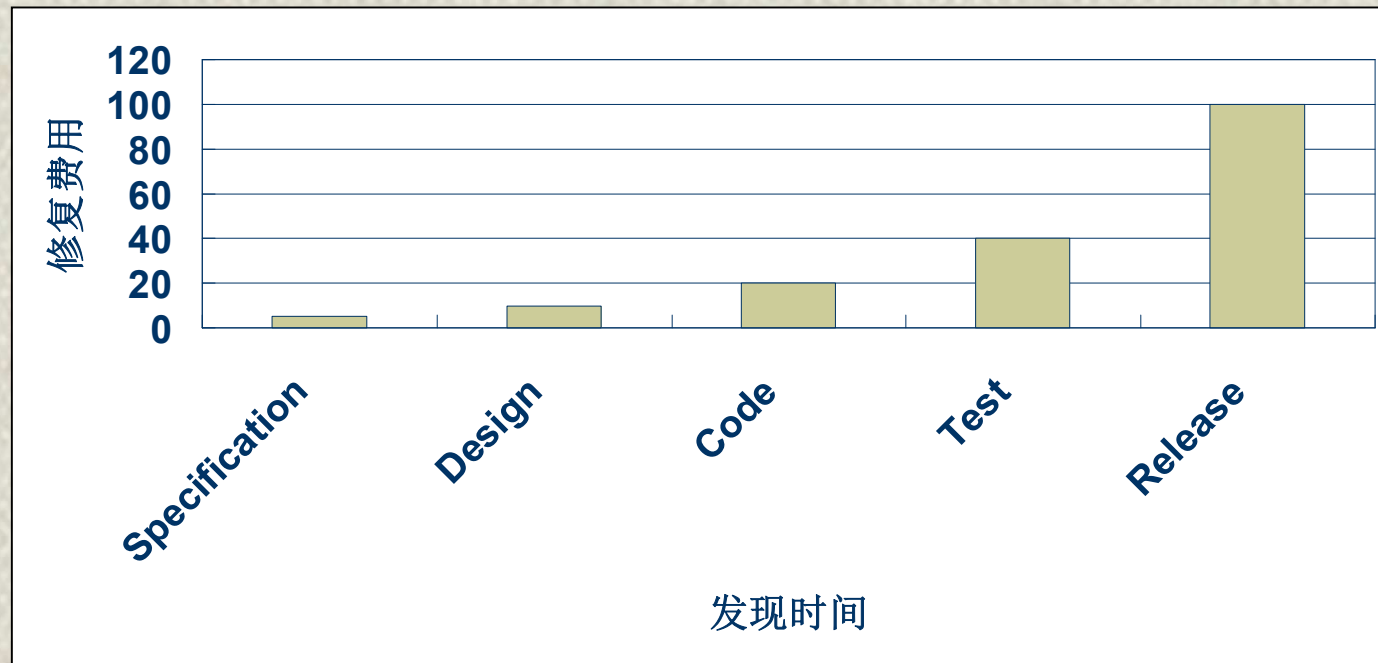
怎么定义软件缺陷？

- 软件未达到产品说明书标明的功能
- 软件出现了产品说明书指明不会出现的错误
- 软件功能超出产品说明书指明范围
- 软件未达到产品说明书虽未指出但应达到的目标
- 软件测试员认为软件难以理解、不易使用、运行速度缓慢、或者最终用户认为不好

BUG怎么产生的？



修复BUG的成本



说出来你可能不信

- **QA和测试的工作从项目一开始就展开了**
 - 文档审核
 - 资源审查
 - 流程检测

- **100%的充分测试是不可能的**
 - 工程量太大（太多的输入/输出）
 - 测试时间有限
 - 软件说明书没有客观标准
 - 总有些信息是无法获取的

- 测试本身具有风险
 - 由于错误地估计BUG的严重性，导致成本增加
 - 过分依赖于测试团队，导致工程质量下降
 - 团队更大，更容易泄漏核心技术

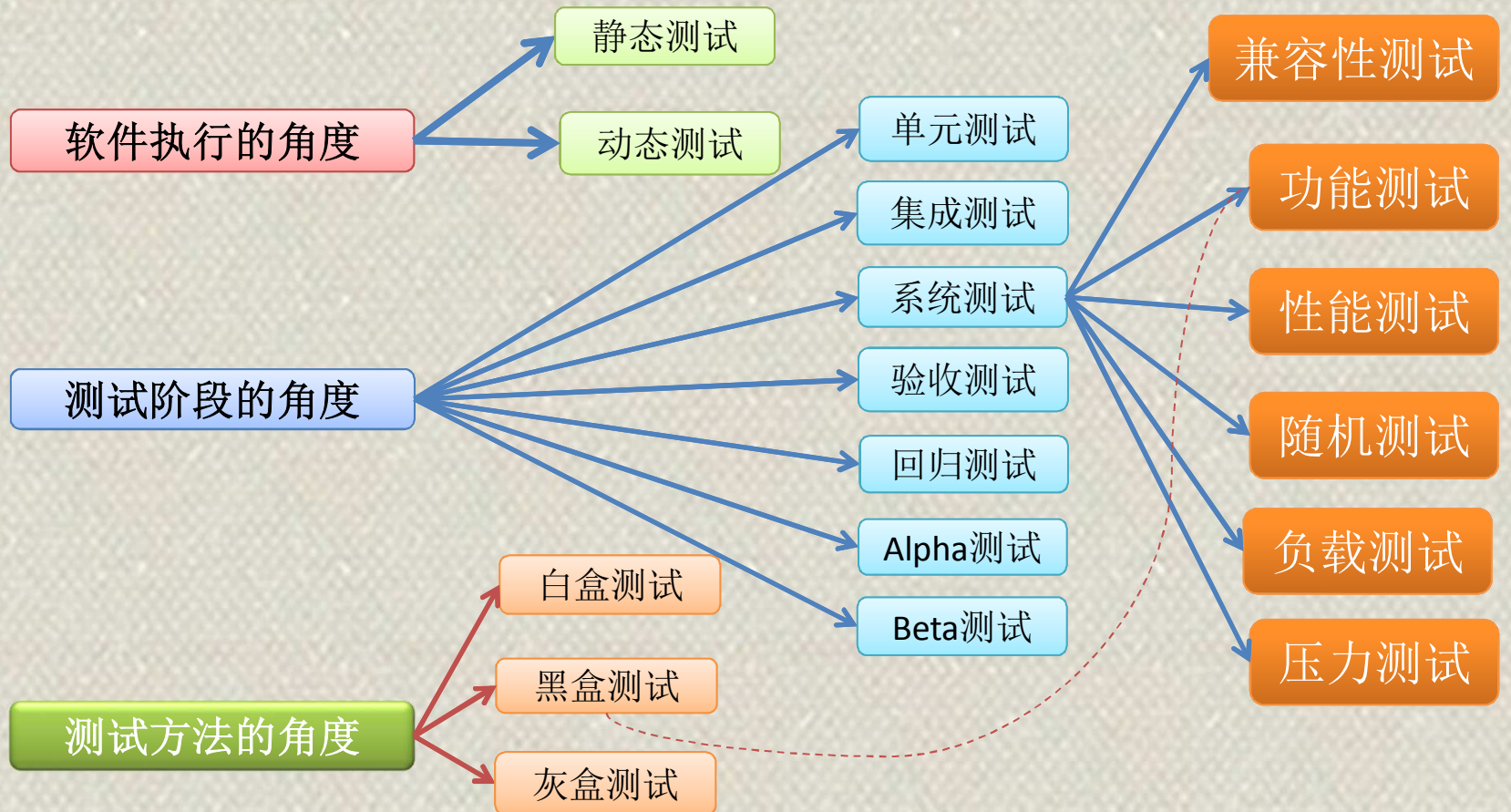
- 有些BUG永远都不能修复
 - 没时间修复，工期太紧张！
 - 没法修复，就是找不出什么原因！
 - 功能太复杂，修复过程和重做没差别！
 - 牵一发动全身，解决一个，带来一百个新的！
 - 不值得修复

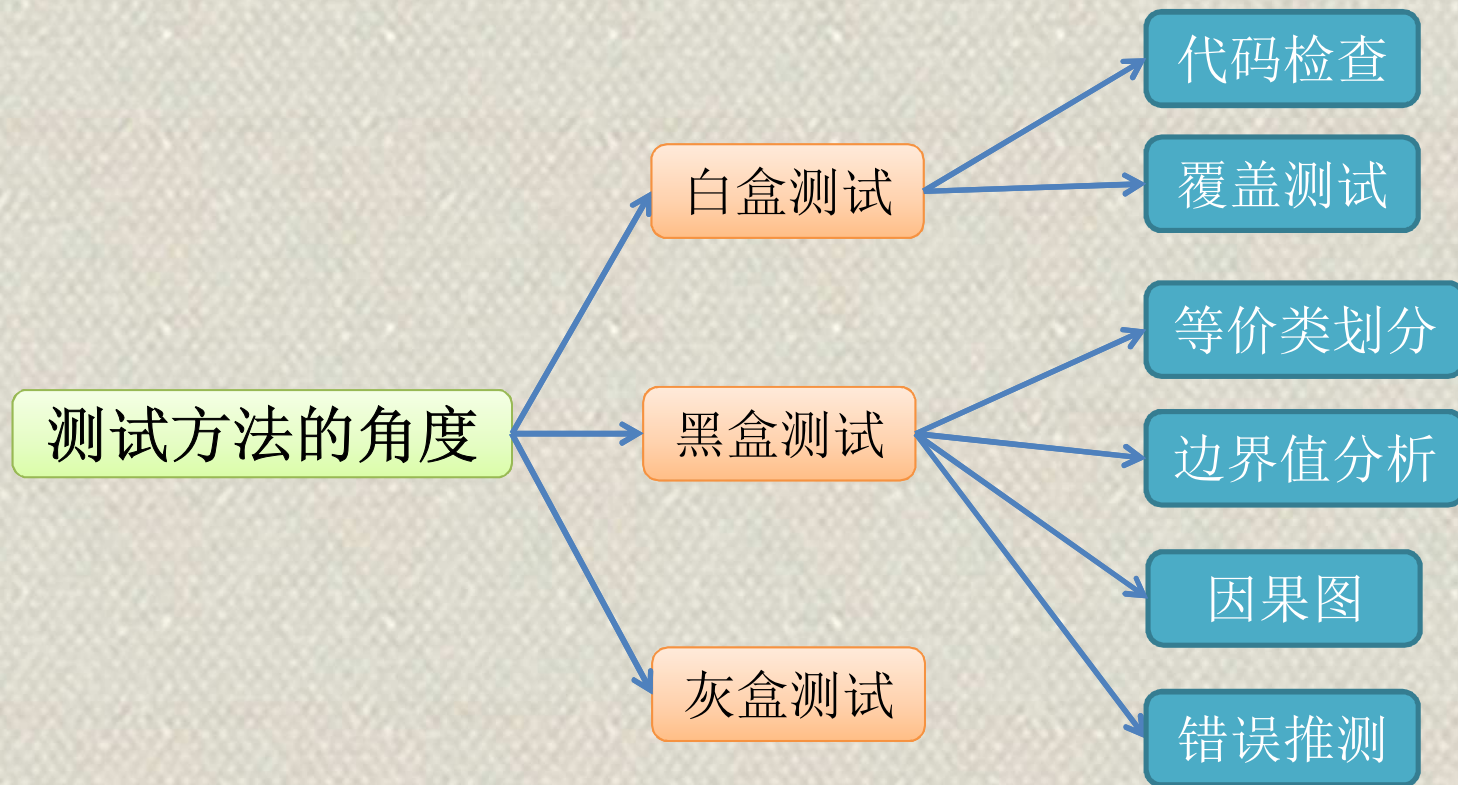
- 找到的**BUG**越多，潜在的**BUG**就越多
- 只能找到一些**BUG**，不可能找到全部

目录

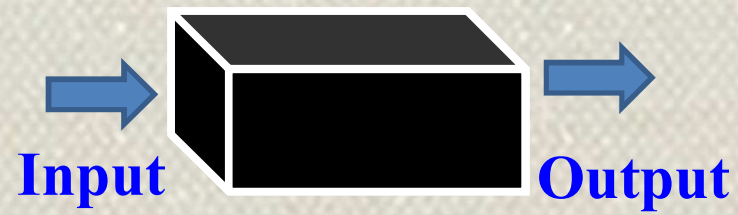
- § 8.1 基本认知
- § 8.2 有效的测试过程
- § 8.3 测试工作的专业性

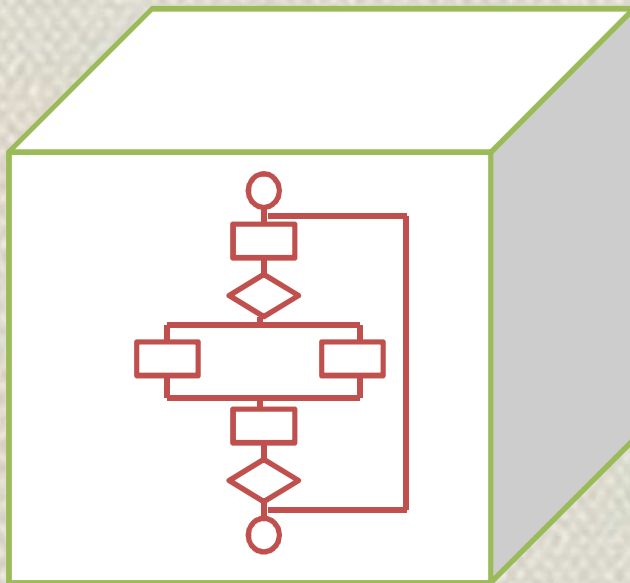
大体的分类方法





Black & White Box





```
int fac(int n)
{
    if (0 == n) return 1;

    return n * fac(n - 1);
}
```

Static & dynamic

- 静态测试：不运行程序, 核对文档/代码
- 动态测试：执行程序, 对“执行过程”和“执行结果”进行分析



Load, Stress & Performance

- 测试产品在不同负载下的表现
- 在各种极限情况下对产品进行测试
- 0.1秒计算出结果/20秒计算出结果



目录

- § 8.1 基本认知
- § 8.2 有效的测试过程
- § 8.3 测试工作的专业性

测试工作的专业性

- 要懂编程和必要的计算机知识
 - if/else/switch的覆盖性测试
 - 边界值分析
 - 数值会不会溢出
 - 硬盘为什么是性能的瓶颈

测试工作的专业性

- 运用分析方法
 - 思路清晰，避免重复无效的工作
 - 正确的方法，增加有效的工作量

分析敌情，围城打援，困守孤城

测试工作的专业性

- 使用工具
 - 可以完成人工无法完成的任务
 - 大量重复的工作让机器去做
 - 工具是方法的实现

攻城车，石砲

测试工作的专业性

- 报告总结
 - 软件测试评价
 - 软件测试报告
 - 数据最具说服力

测试用例

- 测试用例的重要性
- 合理的测试用例
- 等价类划分方法