

# 软件测试之我见

吴群林



什么是软件测试？



“在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。”



# 软件测试方法

- ◆ 黑盒测试
- ◆ 白盒测试



# 黑盒测试

“测试者不了解程序的内部情况，不需具备应用程序的代码、内部结构和编程语言的专门知识。只知道程序的输入、输出和系统的功能，这是从用户的角度针对软件界面、功能及外部结构进行测试，而不考虑程序内部逻辑结构”

话外音： 功能真的跟产品使用手册描述的一样？ 🤨



# 白盒测试

“测试者了解待测试程序的内部结构、算法等信息，这是从程序设计者的角度对程序进行的测试”

话外音：好吧，我承认没有写测试用例😓



# 软件测试类型

- ◆ 功能测试：对软件产品组成的功能模块测试
- ◆ 系统测试：完整的用户体验
- ◆ 极限值测试：软件准入的最大、最小值.....
- ◆ 性能测试：响应速度、内存占用、C10K.....



# 测试阶段

- ◆ 单元测试：面向函数
- ◆ 集成测试：面向接口
- ◆ 系统测试：面向候选产品
- ◆ 回归测试：面向发布产品



# 软件测试方法论看似完美👼

## 如何落地.....





# 开发工程师



- ◆ 哪里有Bug
- ◆ 很快，马上就改完
- ◆ 这个改动很小，不用测试
- ◆ 我测的时候明明好好的
- ◆ ...

李纳斯法则：只要有足够多的眼球关注，就可让所有软件缺陷浮现。



# 测试工程师



- ◆ 改好没有
- ◆ 还是不行呢
- ◆ 改动引入了新bug
- ◆ 接口好慢
- ◆ .....

软件测试二八原则：80%的错误是由20%的模块引起的



# 开发与测试

- ◆ 测试代码是往往是功能代码的好几倍
- ◆ 微服务系统下mock调用链
- ◆ UT/IT/CI
- ◆ 测试覆盖率
- ◆ 流程自动化
- ◆ 工具定制化

话外音：没有谁比谁更重要，协作方能共赢



# 行业乱象

- ◆ 快、糙、猛
- ◆ 70%可用性即可上线
- ◆ 产品需求朝令夕改
- ◆ 拍脑袋定Deadline
- ◆ 部门墙
- ◆ 踢皮球

话外音：本是同根生 相煎何太急😓 最近又掉头发，要聪明就不帅了😂



# 系统稳定性

一个关于9的故事

BEST-IN-CLASS

**99.99%**

AVAILABILITY



# 软件故障

- ◆ 服务不可用（第三方依赖出错.....）
- ◆ 异常退出（软件运行时崩溃.....）
- ◆ 其它不可控因素（如太阳黑子影响硬件物理特性引发软件异常.....）

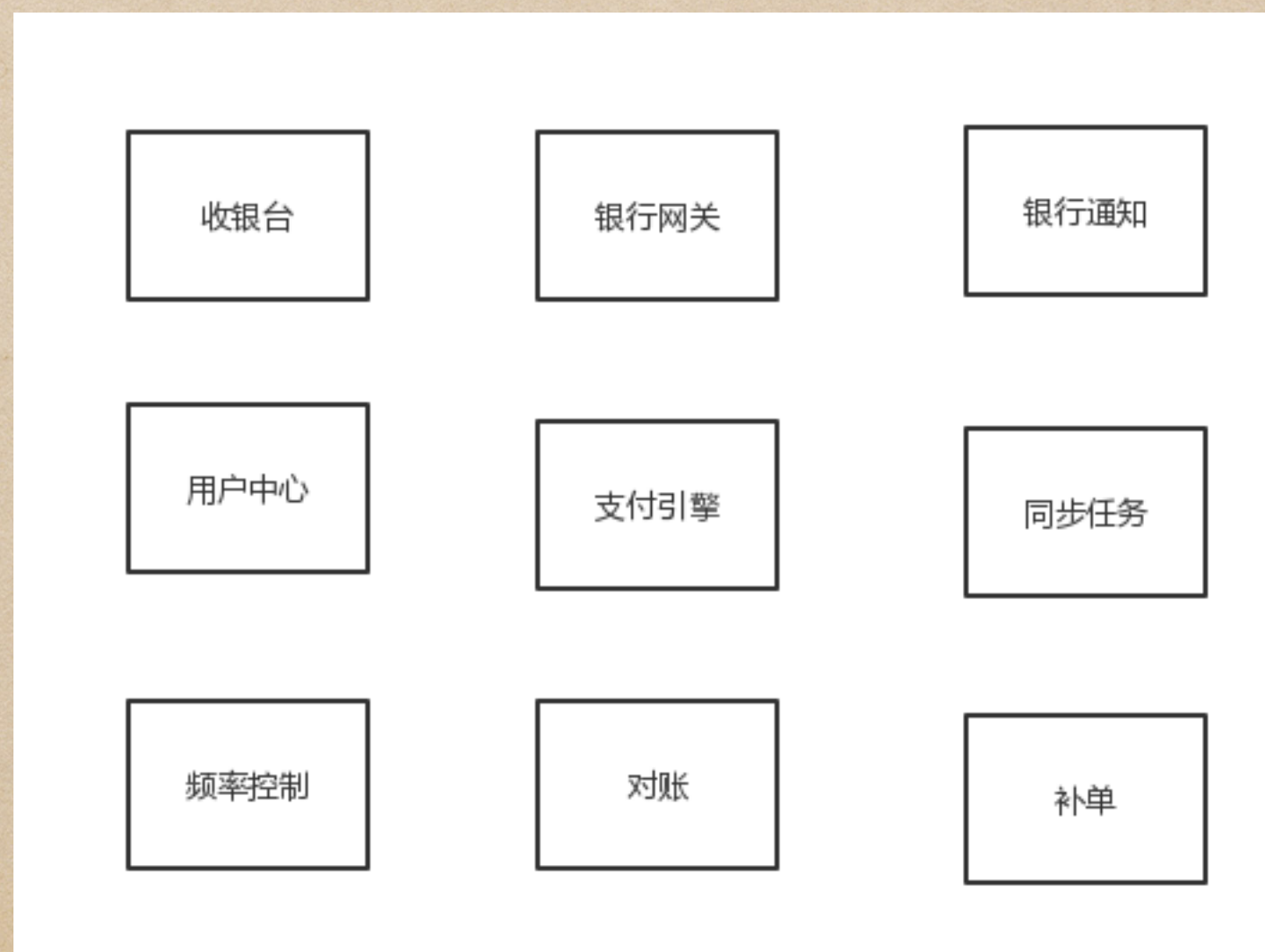


# 故障处理

- ◆ 确定故障特征
- ◆ 确定影响范围
- ◆ 止损



# 支付平台主要子系统





# 系统发生故障

- ◆ 第三方银行故障致网关不可用?
- ◆ 银行通知失效?
- ◆ OOM(Out of memory)?
- ◆ 数据库死锁?
- ◆ 上线版本、配置错误?
- ◆ 微服务容器间失联?



# 故障测试

- ◆ 识别所有内部和外部系统依赖关系。
- ◆ 记录系统故障点及其对系统的影响。
- ◆ 记录用于监视和排查生产问题的“标准操作规程”。
- ◆ 确定系统是否按照设计的方式运行。
- ◆ 确定系统发生故障时对用户体验的影响。



# 减少故障的建议

- ◆ 严格计划和执行故障测试。
- ◆ 理解应用程序的故障点。
- ◆ 记录如何排除故障并从故障中恢复。



# 故障预防

- ◆ 可预见故障~尽可能完善测试集

- ◆ 不可预见故障~随机模拟  
(Chaos Monkey)



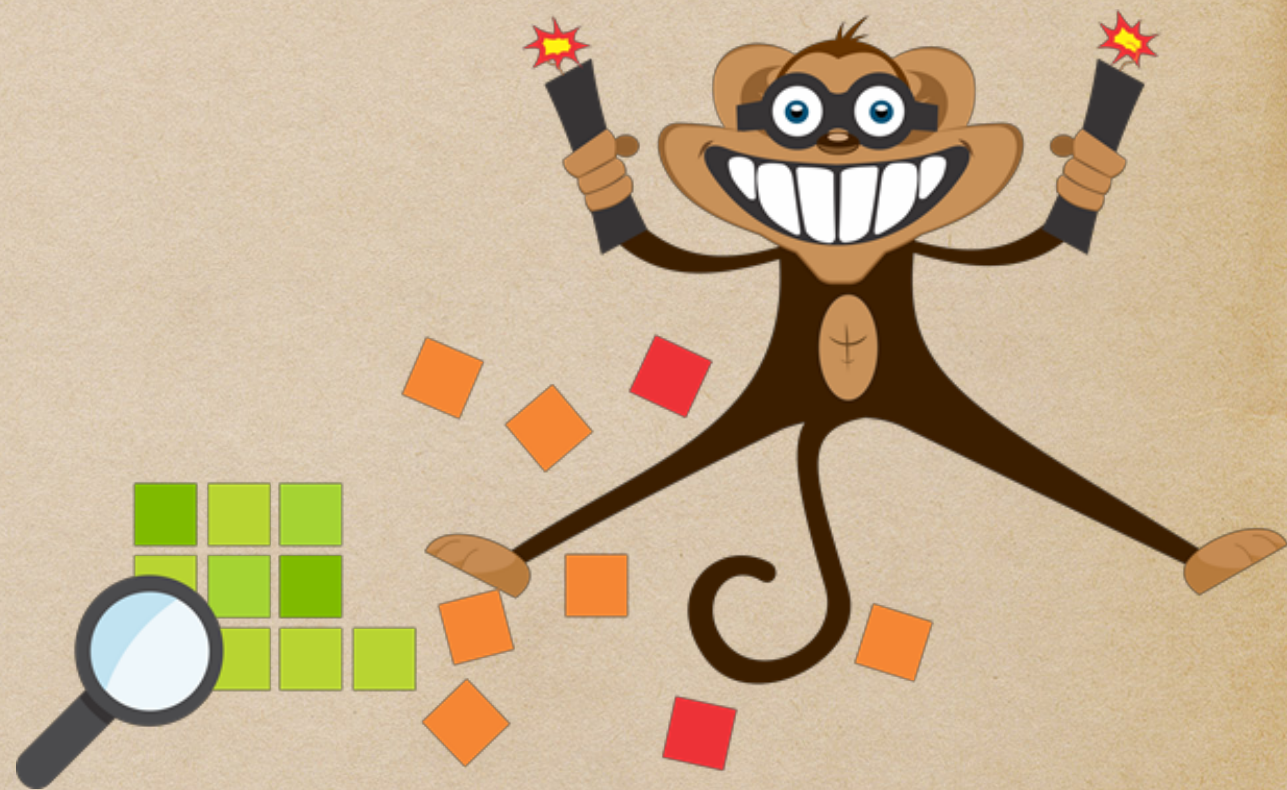
“在他人攻击您的程序前，先自己攻击”

- ◆ Fuzz
- ◆ 故障注入
- ◆ 自动化测试
- ◆ .....



# 混沌工程

- ◆ 混沌工程师通过应用一些经验探索的原则，来学习观察系统是如何反应的





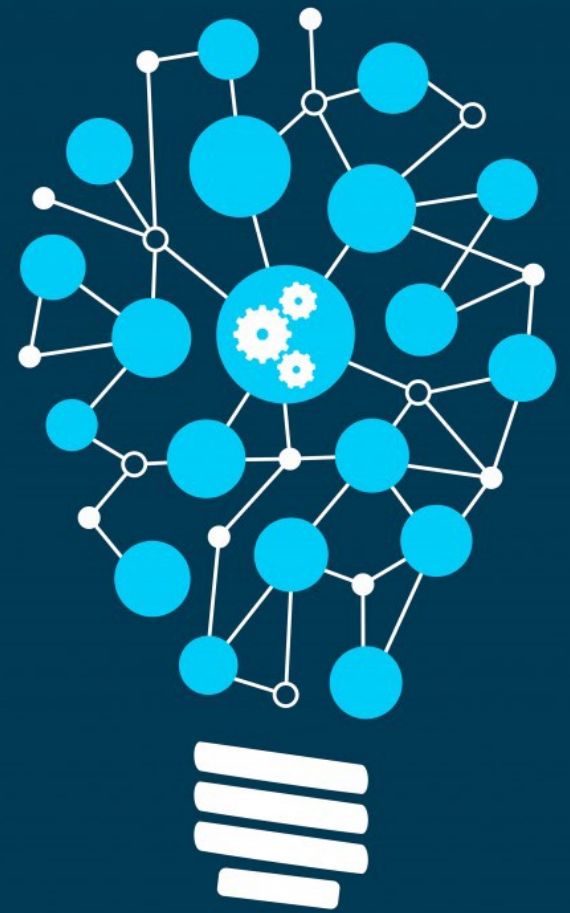
“应用混沌工程能提升整个系统的弹性。通过设计并且进行混沌实验，我们可以了解到系统脆弱的一面，在还没出现对用户造成伤害之前，我们就能主动发现这些问题。”



# 测试未来展望



MACHINE  
LEARNING





- ◆ AI修复补丁.....
- ◆ 机器学习检测软件系统代码故障.....

话外音：开发测试傻傻分不清



# 参考内容

- ◆ <https://zh.wikipedia.org/wiki/软件测试>
- ◆ <http://www.cghuawei.com/archives/8513>
- ◆ <https://techblog.constantcontact.com/software-development/failure-testing-avoid-a-recipe-for-disaster/>
- ◆ <https://www.jianshu.com/p/96df8c4c100e>
- ◆ <https://www.jianshu.com/p/4bd4f88e24e4>
- ◆ <https://github.com/Netflix/chaosmonkey>
- ◆ <https://www.infoq.cn/article/chaos-practice-in-tidb>
- ◆ <https://www.ibm.com/developerworks/cn/java/j-fuzztest.html>
- ◆ <https://github.com/src-d/awesome-machine-learning-on-source-code#machine-learning>
- ◆ <https://mp.weixin.qq.com/s/dpJRequ5eilPKBT-wqA6YQ>
- ◆ <https://www.infoq.cn/article/system-failure-modeling>
- ◆ <https://zhuanlan.zhihu.com/p/30605918>
- ◆ <https://www.infoq.cn/article/2016%2F03%2Fchaos-testing-microservices>



Thanks