



云推送技术实现与 敏捷开发

Powering Mobile Apps

@Tiger_张虎

Agenda

- Push是什么
- 极光推送原理
- C2000K问题
- 敏捷开发实践

如何更新应用的数据？

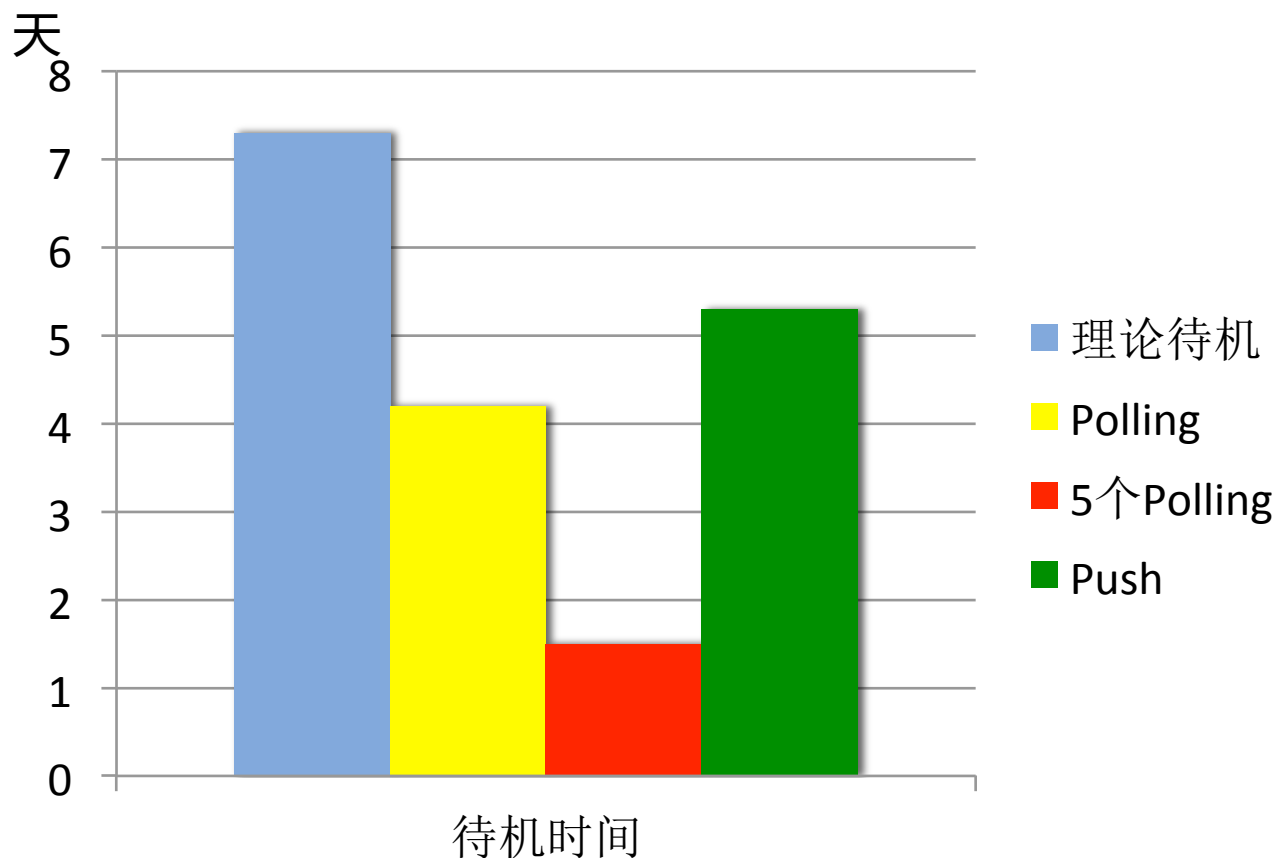
- 新数据更新到应用
- 实现的技术
 - Push: 服务器实时推送
 - Polling: 定时查询

Polling的问题

- 无论有没有数据，都需查询
- 额外的电量和带宽

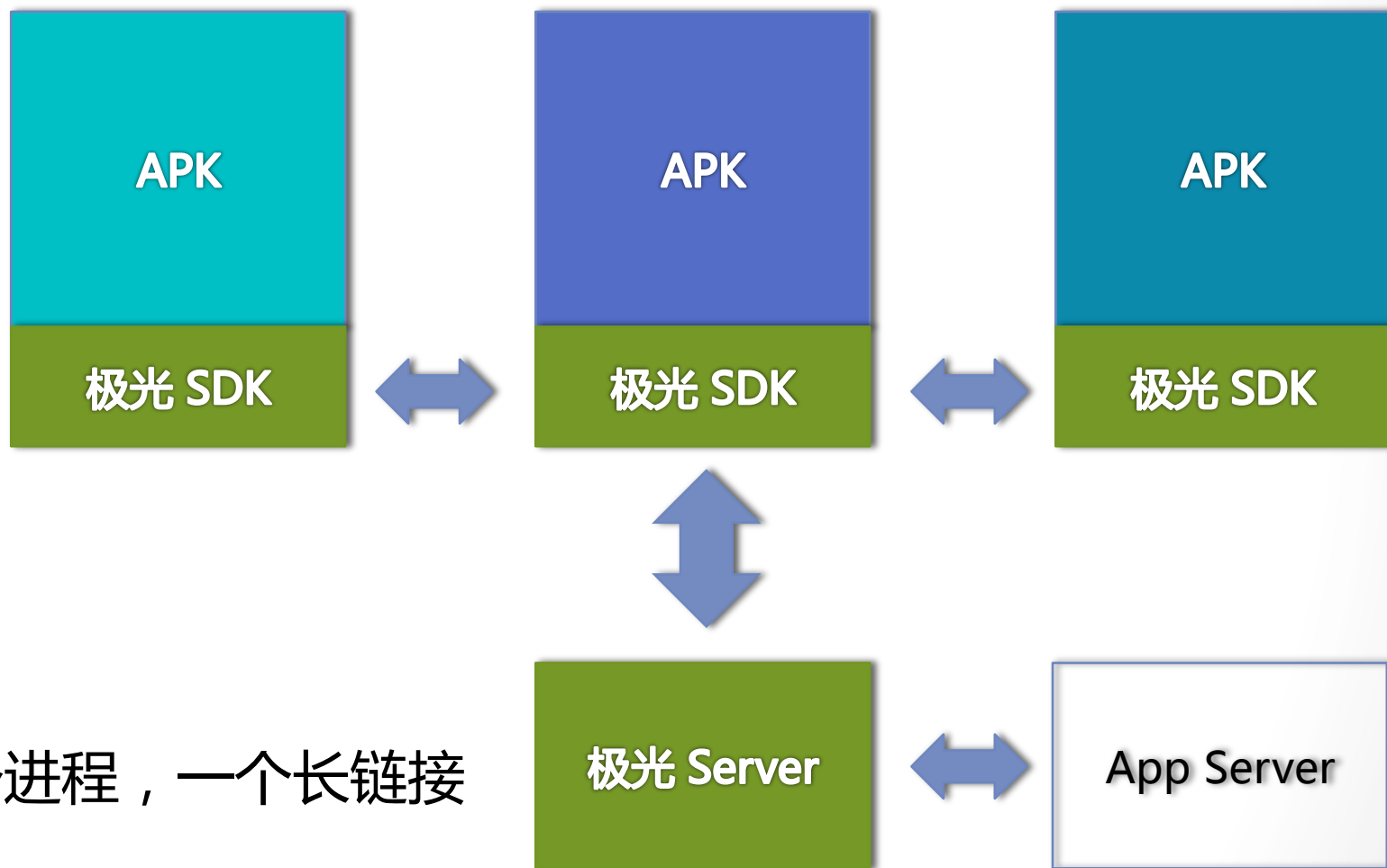
	Push	Polling
延迟	低	高
流量	低	高
电量	低	高
实现的复杂度	高	低

手机耗电情况



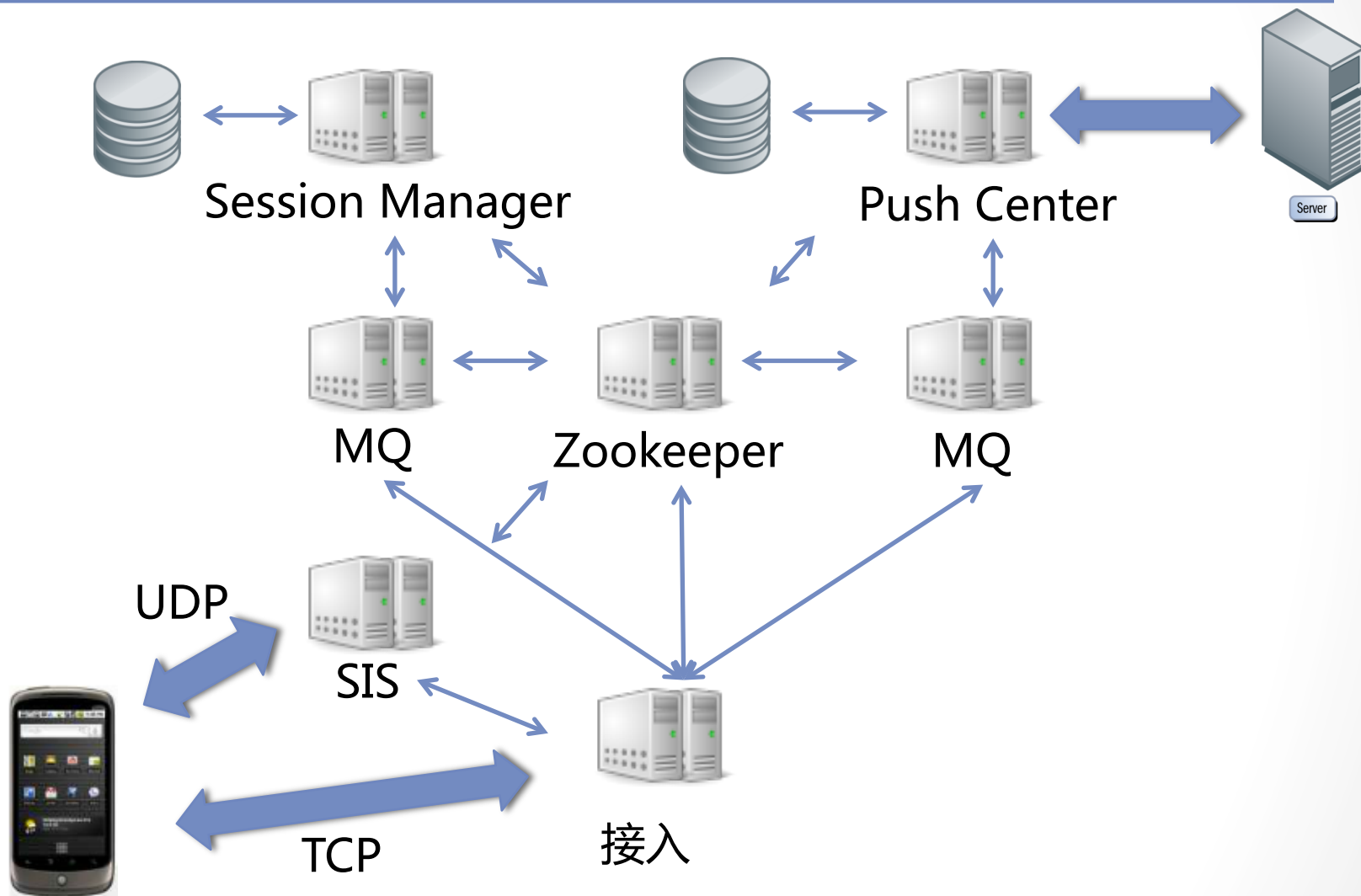
- 电池1400 mAh的手机，底电流8mA

极光 Android SDK



一个进程，一个长链接

极光 服务器集群架构



中间件

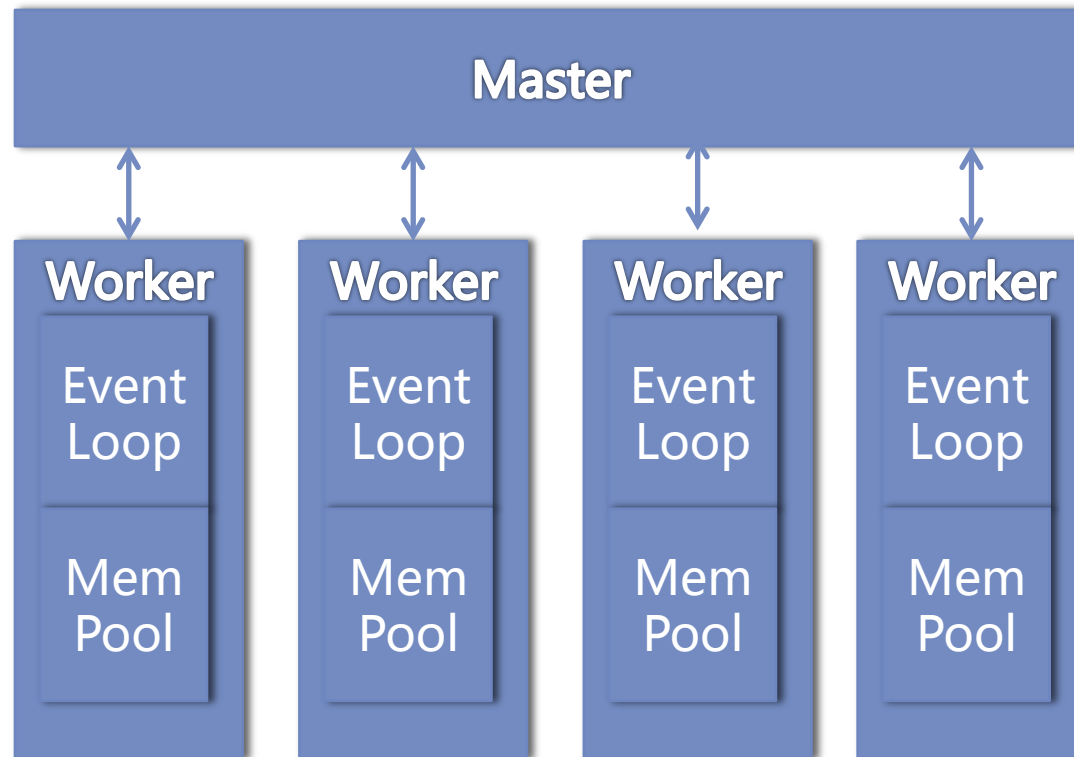
- Cache
 - Memcached/Couchbase
 - 100%读命中
- Message Queue
 - RabbitMQ

如何维持大量长连接?

C2000K: 服务器设计

- Libev: 事件驱动、异步、非阻塞
- Memory Pool: 省内存、加快分配和回收
- 多核
 - 一个核一个进程
 - 共享监听一个socket
- 每个子进程有自己的event loop, mem pool

C2000K: 架构



C2000K: Linux Kernel参数

- `fs.file-max` = 10485760
- `net.ipv4.tcp_wmem` = 4096 4096
16777216
- `net.ipv4.tcp_rmem` = 4096 4096
16777216
- `net.ipv4.tcp_tw_reuse` = 1

C2000K: 长链接曲线



-
- 目前为止业界最大单机维持连接数

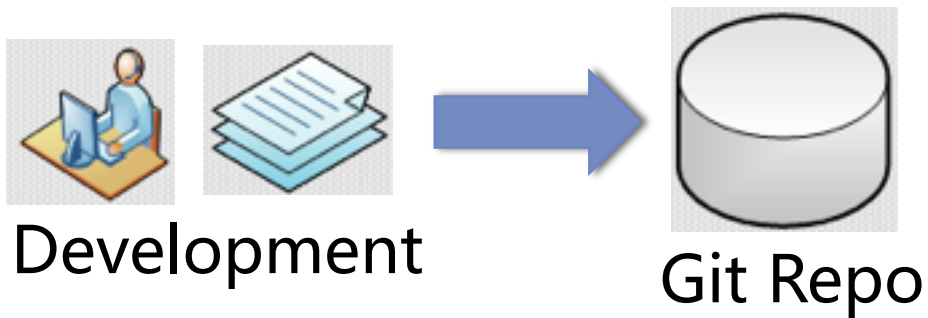
敏捷开发实践

- 持续集成(Continuous Integration)
 - 提交间隔时间越长，风险越大
 - 尽早集成、经常集成
 - 每次集成后做完整的回归测试

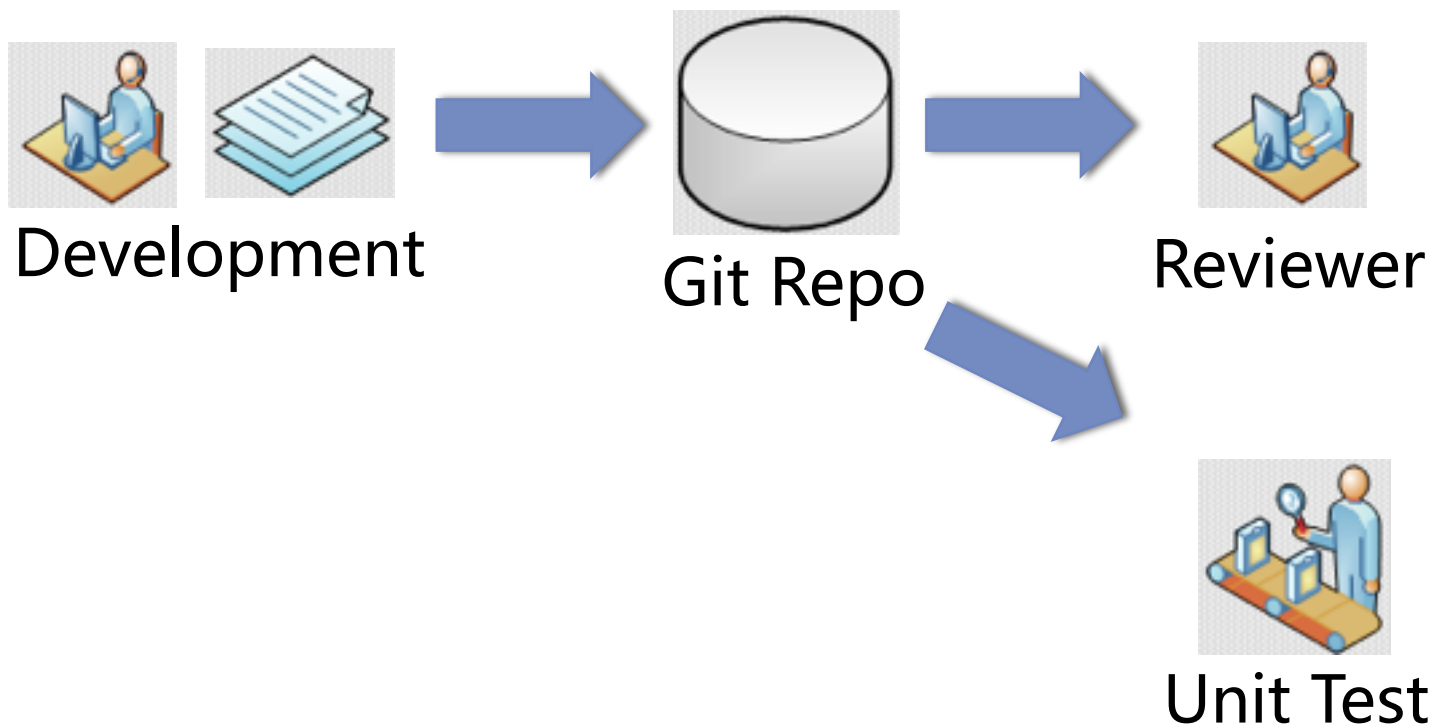
持续集成：工具

- 代码管理：Git
- 任务管理: Pivotal Tracker
- 代码review: Review Board
- 自动编译：Buildbot
- 自动测试：Unit Test
- 自动部署：Xen, Yum

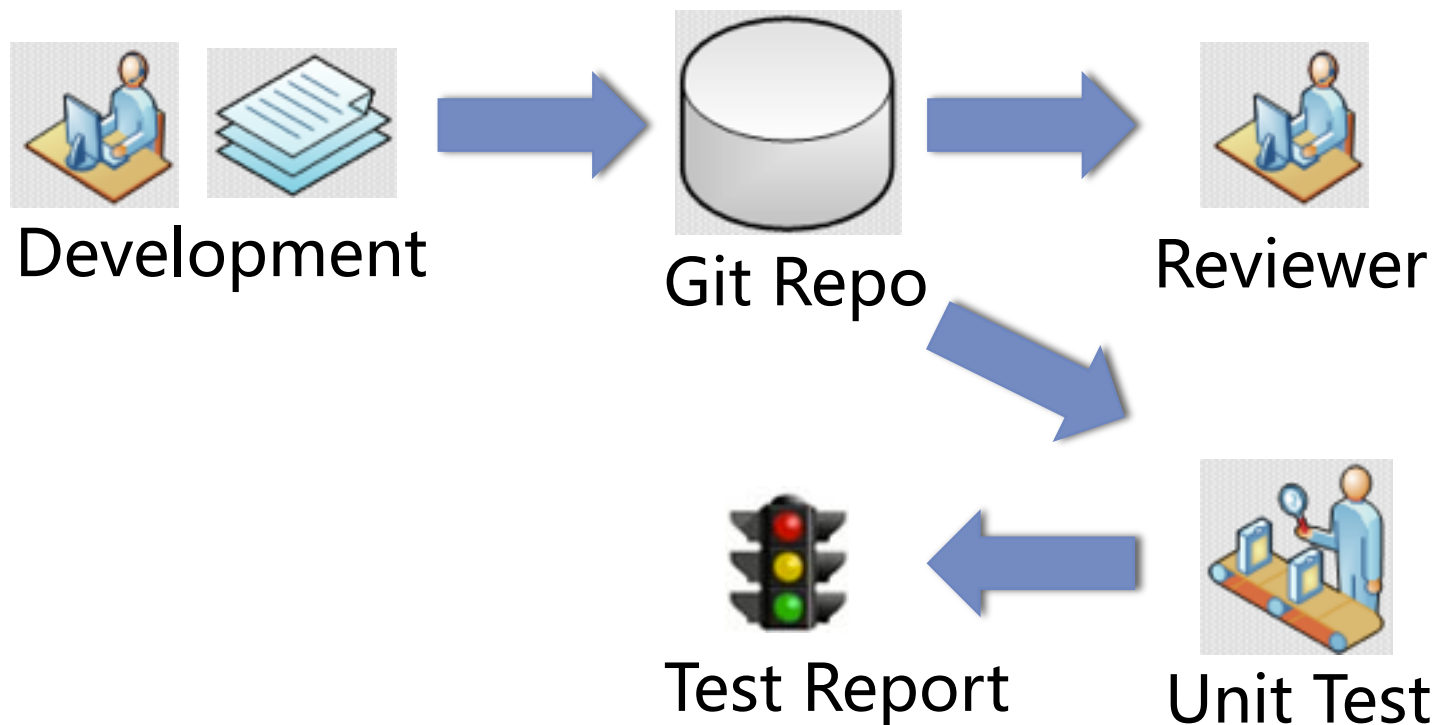
持续集成：流程



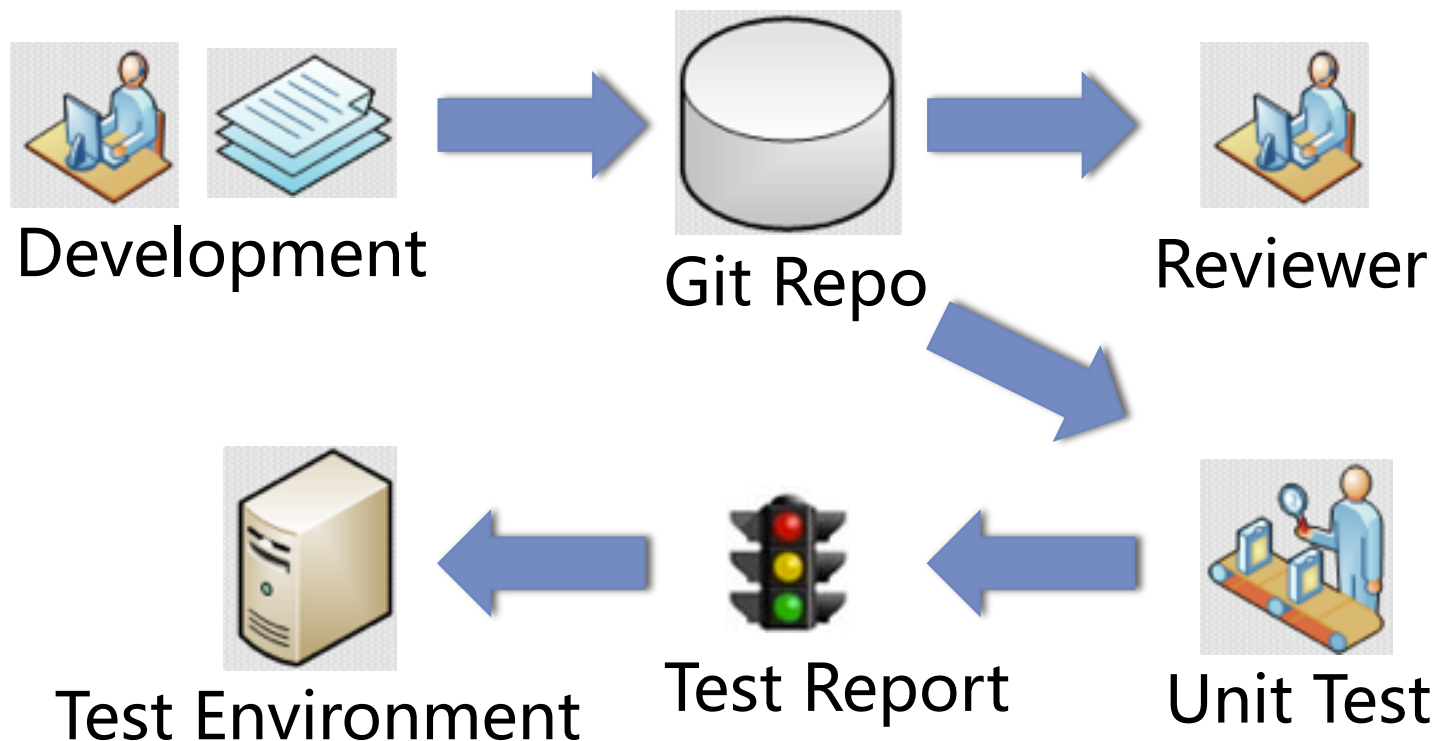
持续集成：流程



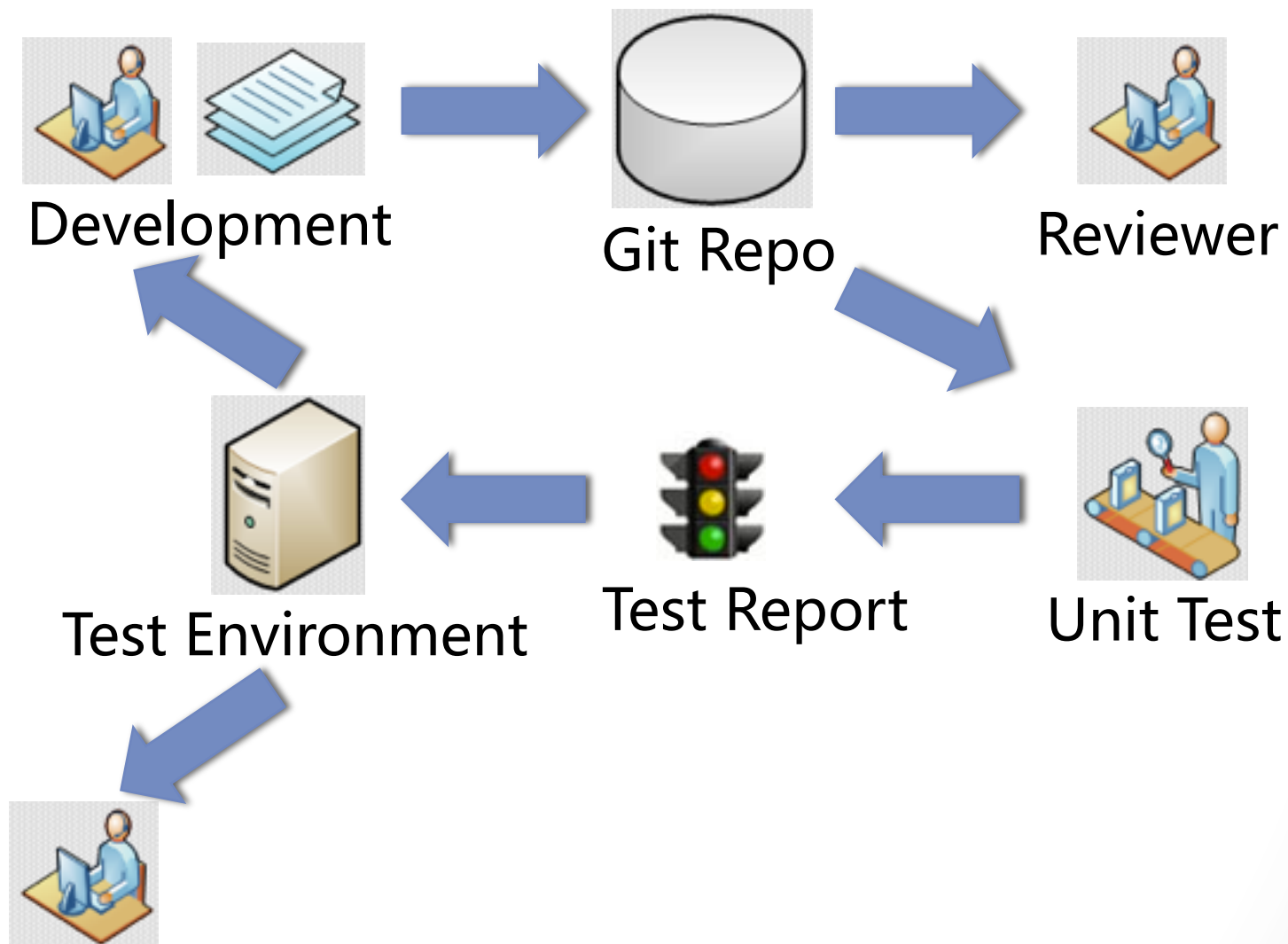
持续集成：流程



持续集成：流程



持续集成：流程



持续集成：代价与优点

- 代价
 - 前期系统搭建工作量大
 - 需要编写优质的测试用例
- 好处
 - 及早发现问题，大大降低debug成本
 - 一直有一个可用的版本
 - 开发人员能更专注于实现优质可用代码

持续的持续集成

- 持续集成本身也是一个持续的过程
 - Test Case 的不断累积
 - 自动化工具的优化
 - 管理方法的优化
- 持续的自我优化，一个团队核心竞争力

Q&A

@极光推送
@Tiger_张虎



THANK YOU