Facebook

- Continuous Deployment at Facebook is Not Continuous Deployment
  - 内部代码或者bug fixes一天发布两次，新的生产代码每周发布一次
  - 独立、细小的改变频繁的发布
    - 归功于其架构，实际上可以独立且廉价地进行许多小的独立更改
  - Gatekeeper工具控制哪些客户可以使用哪些功能以支持dark launching
  - 代码增量发布
    - staging -> 部分用户->整个代码的发布
    - 如有必要，可以轻松回滚
- Code ownership
  - 开发人员需要支持运维
    - 使得他们更重视质量
    - 工具和方法论还可能被误用
    - 这种个人责任文化至关重要
  - 大多数源文件仅有少数工程师修改
    - 三分之一的代码有一位工程师修改
    - 四分之一的两位
    - 10%的文件有7位工程师以上处理
  - 文件的工程师分布情况是重尾分布
    - 少量共享文件由至少870位工程师处理
      - 库文件
      - 主要配置
      - 顶级PHP文件
- Testing
  - 没有独立的测试人员
    - 主要靠自动化测试
  - 重视code review，每一行代码都有除作者外的一名工程师review
    - 新的思想帮助发现defect，提供建议
    - 编码相关knowledge得以流传
  - 开发人员负责单元测试、回归测试和自动化的性能测试
  - 开发人员使用开发版本自测 （用于个人用途）
  - 依赖用户测试
    - A/B测试
- Security
  - 很重视性能，但是并不是很重视安全

BY林子版权所有

BY林子网站

BY林子公众号

质量三人行