



巴克莱银行实现DevOps 的挑战和关键

2018中国·北京

周纪海

目录

1 自我介绍

2 巴克莱银行实现DevOps的困难和挑战

3 巴克莱银行实现DevOps的关键和案例分析



自我介绍

- 英国伦敦帝国理工学院博士毕业
- 2012年起开始在巴克莱银行从事DevOps工作
- 曾就职于：
 - 苏格兰皇家银行
 - 瑞银集团
 - 巴克莱银行
 - 汇丰银行



巴克莱银行实现DevOps的困难和挑战



金融行业中IT特有的文化

- 金融行业中的IT部门并不是直接创造价值的部门
- IT高层更关心如何完成项目，而非IT本身如何改进
- 过高安全性的要求造成了繁琐冗长的审批流程，效率低下
- 很多高层和业务人员非IT出身，难以了解下面具体工作
- 在大型银行，职责分工过细，造成工作局限性和效率低下
- 技术更新慢

固有开发习惯难以改变，效率低

- 瀑布（Waterfall）流程居多
- 开发流程不透明
- 代码质量不关心
- 手动配置多
- 工具选择太多，不方便互相支持
- 反馈机制没有建立起来，无法及时修正
- 相关设备和工具利用率低下

算了，我习惯了



相关部门的协调难以进行

- 相关部门太多，分工太细
- 团队各自为战
- 地理位置
- 语言
- 管理不善
- 开发人员水平参差不齐
- 利益 😊





巴克莱银行实现DevOps的关键和案例分析



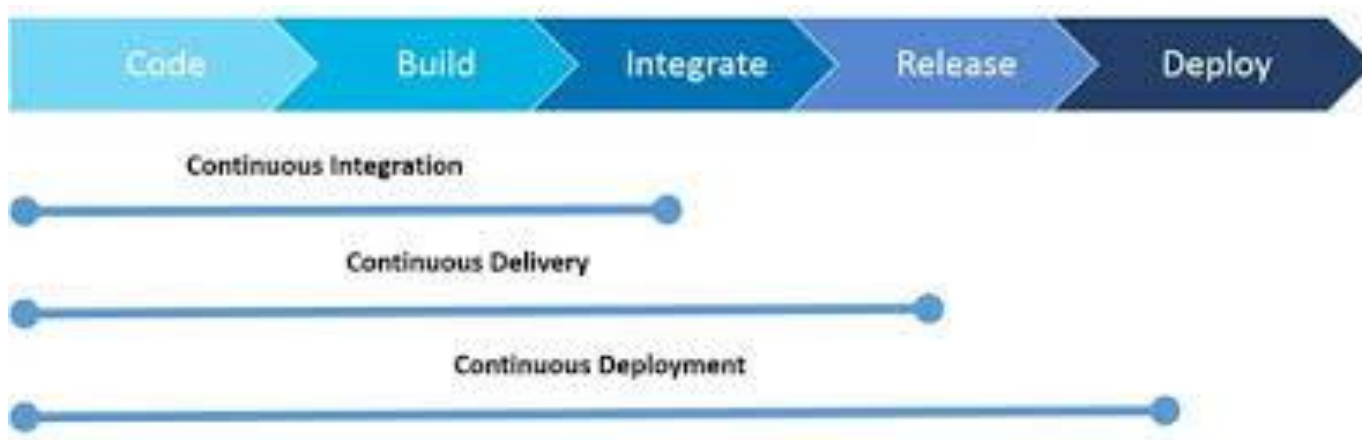
DevOps 成熟度标准

| | Base | Beginner | Intermediate | Advanced | Expert |
|-------------------------|---|---|--|--|---|
| Culture & Organization | <ul style="list-style-type: none"> Geprioriteerd werk Proces is vastgesteld en gedocumenteerd Veelvuldige commits | <ul style="list-style-type: none"> Een backlog per team Gedeelde pijn Stabiele teams Adoptie van basis Agile methoden Verwijderen van grenzen tussen development en test | <ul style="list-style-type: none"> Uitgebreide samenwerking tussen teams Eigendom van componenten Reageren op metrics Verwijderen van grenzen tussen Dev en Ops Zelfde proces voor alle wijzigingen Decentralisatie van besluitvorming | <ul style="list-style-type: none"> Specialistische tools-teams Team is volledig verantwoordelijk tot aan productie Deployment losgekoppeld van release Continuous improvement (Kaizen) | <ul style="list-style-type: none"> Cross-functionele teams Niet meer terug..altijd vooruit. |
| Design & Architecture | <ul style="list-style-type: none"> Geconsolideerd platform en techniek | <ul style="list-style-type: none"> Systeem organiseren in modules API management Library management Versiebeheer op DB wijzigingen | <ul style="list-style-type: none"> Geen (of minimaal) branching Splits op abstractie niveau Configuratie als code Verbergen van features Modules ombouwen tot componenten | <ul style="list-style-type: none"> Volledige component gebaseerde architectuur Push business metrics | <ul style="list-style-type: none"> Infrastructuur als code |
| Build & Deploy | <ul style="list-style-type: none"> Code in versiebeheer Builds zijn gescript Basis build scheduling (CI) Dedicated build server Deployment handmatig en gedocumenteerd Beperkt aantal deployment script | <ul style="list-style-type: none"> Frequenter bouwen Artefacten worden opgeslagen Handmatige taggen en versionering Eerste stap naar gestandaardiseerde deployment | <ul style="list-style-type: none"> Automatische build, na commit Automatisch tag en versionering Buidl once, deploy anywhere Automatische DB Scripts ook voor bulk wijzigingen Basis pipeline naar productie Gescripte configuratie wijzigingen Standaard proces voor alle omgevingen | <ul style="list-style-type: none"> Zero downtime deployment Meerdere build machines Volledige automatische DB deploy | <ul style="list-style-type: none"> Bakken van infra en deployable in één Zero touch continuous deployment |
| Test & Validation | <ul style="list-style-type: none"> Automatische unit tests Aparte testomgeving | <ul style="list-style-type: none"> Automatische integratietest | <ul style="list-style-type: none"> Automatische component test Enige automatische acceptatie test | <ul style="list-style-type: none"> Volledige automatische acceptatie test Automatische performance test Automatische security test Risk-based handmatig testen | <ul style="list-style-type: none"> Validatie op verwachtingen aan business value |
| Information & reporting | <ul style="list-style-type: none"> Basis proces monitoring Handmatige rapportages | <ul style="list-style-type: none"> Meet het proces Statische code analyse Geplande automatische rapporten | <ul style="list-style-type: none"> Gemeenschappelijk informatiemodel Traceerbaarheid in pipeline Rapportage historie beschikbaar | <ul style="list-style-type: none"> Grafische weergave Dynamische analyse van testdekking Rapportage en trend analyse | <ul style="list-style-type: none"> Dynamisch grafisch dashboard Cross silo analyse |



开发流程自动化 - CICD

- 开发流程的自动化是实现DevOps 基础和先决条件
- CICD
 - 持续集成 (Continuous Integration)
 - 持续交付 (Continuous Delivery)
 - 持续发布 (Continuous Deployment)

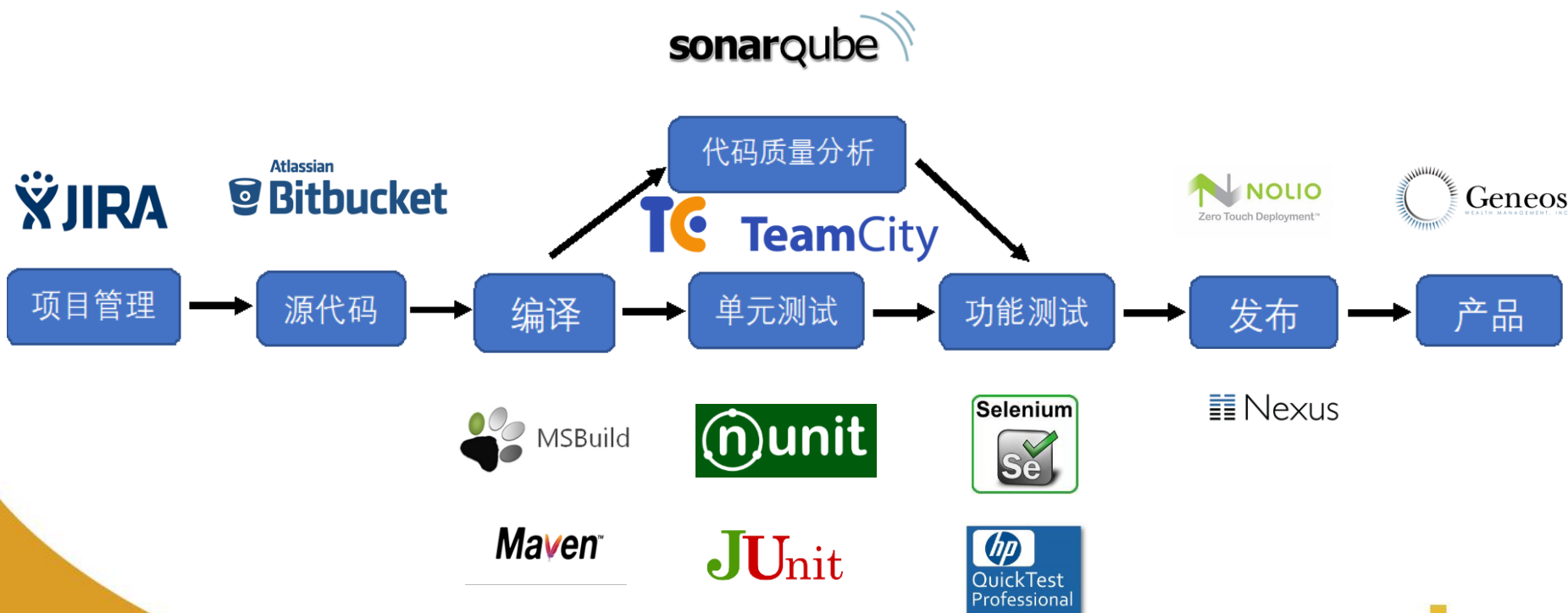


标准化DevOps工具栈

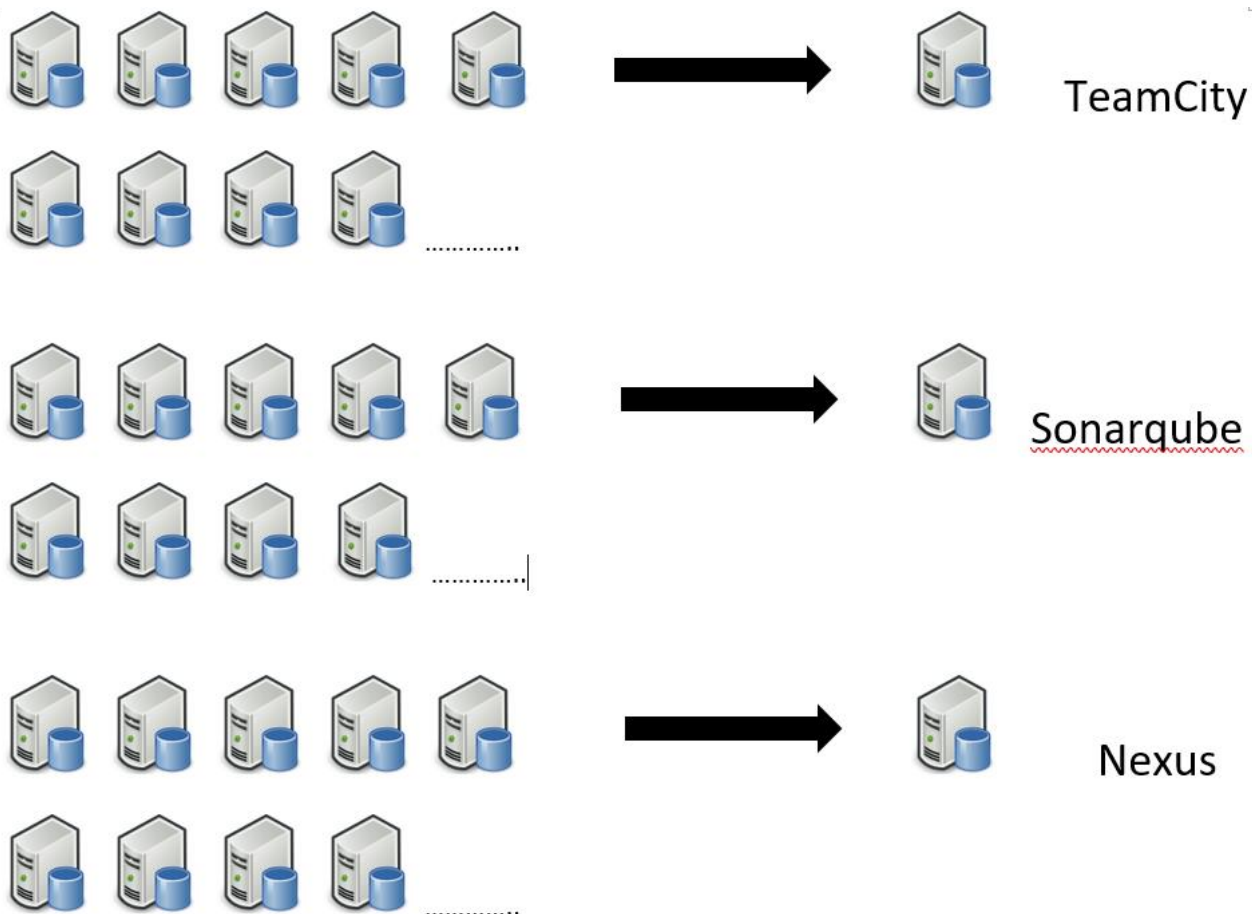
- 项目和任务管理工具: **Jira**
- 源代码管理工具: Github, **Bitbucket**, Gitlab, SVN, Perforce, Clearcase
- 代码质量管理工具: **Sonarqube**, Findbug, Resharper, FXCop, Cppcheck
- 持续集成工具: Jenkins, **TeamCity**, Bamboo
- 编译工具: **Maven**, Ants, **MSBuild**, Nants, Gradle
- 单元测试工具: **Junit**, **Nunit**
- 仓库管理工具: **Nexus**, Artifactory
- 自动测试工具: **Selenium**, **QTP**
- 自动发布和配置工具: Chef, Puppet, Ansible, **Nolio**
- 产品监控工具: **Geneos**, Grafana, Sensu



标准化DevOps自动化流水线



团队级到部门级的工具中心化



开发流程数据的中心化收集和分析



部门级 TeamCity 服务器



部门级 Sonarqube 服务器



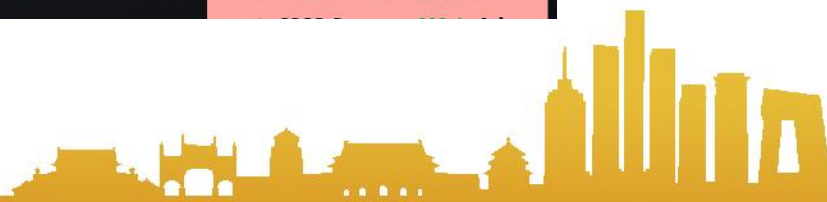
中心 Jira



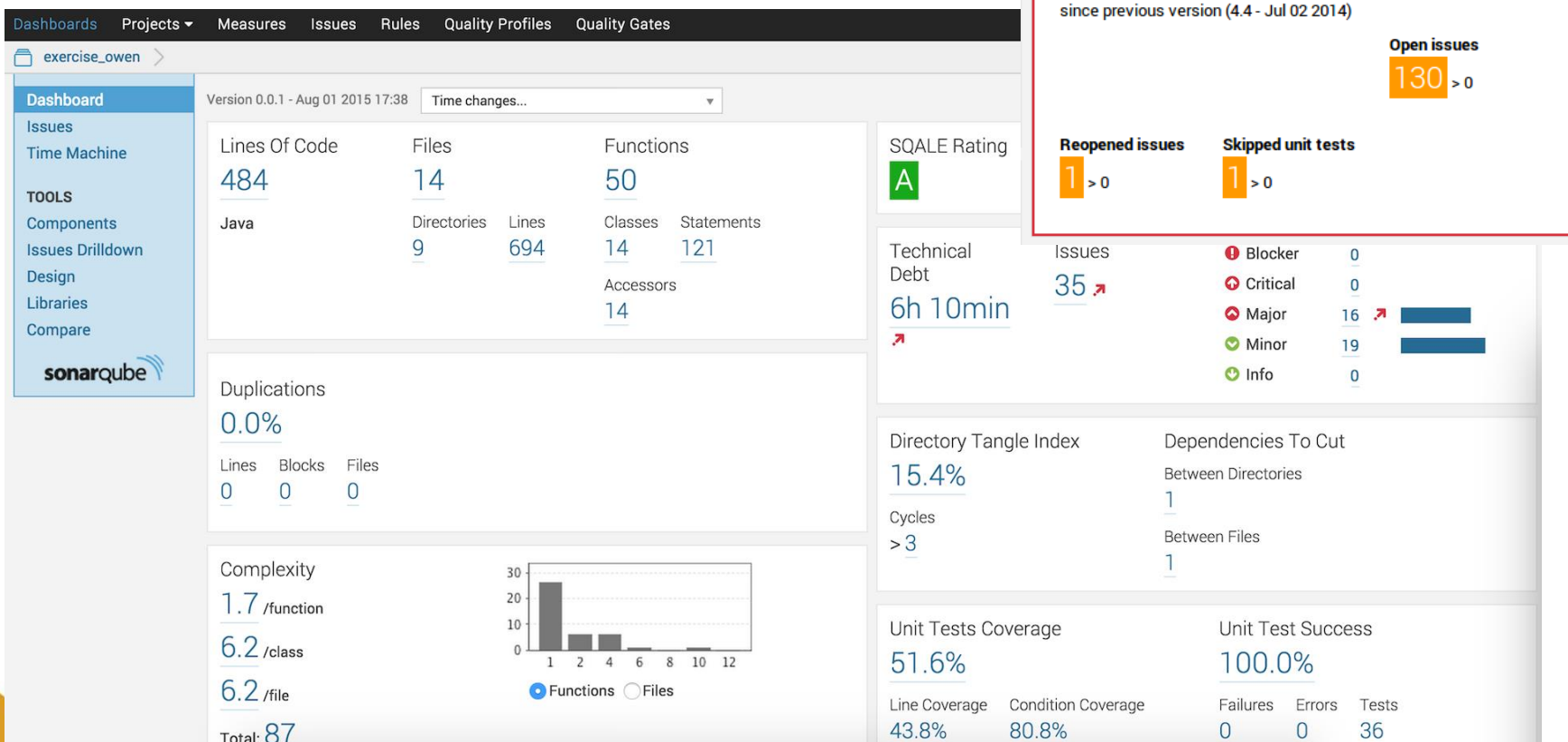
中心 Nolio



开发流程数据的收集，分析和反馈



推动提高代码质量文化



与业务部门紧密协作

Be a good sales man to
convince business to buy it

➤ 案例

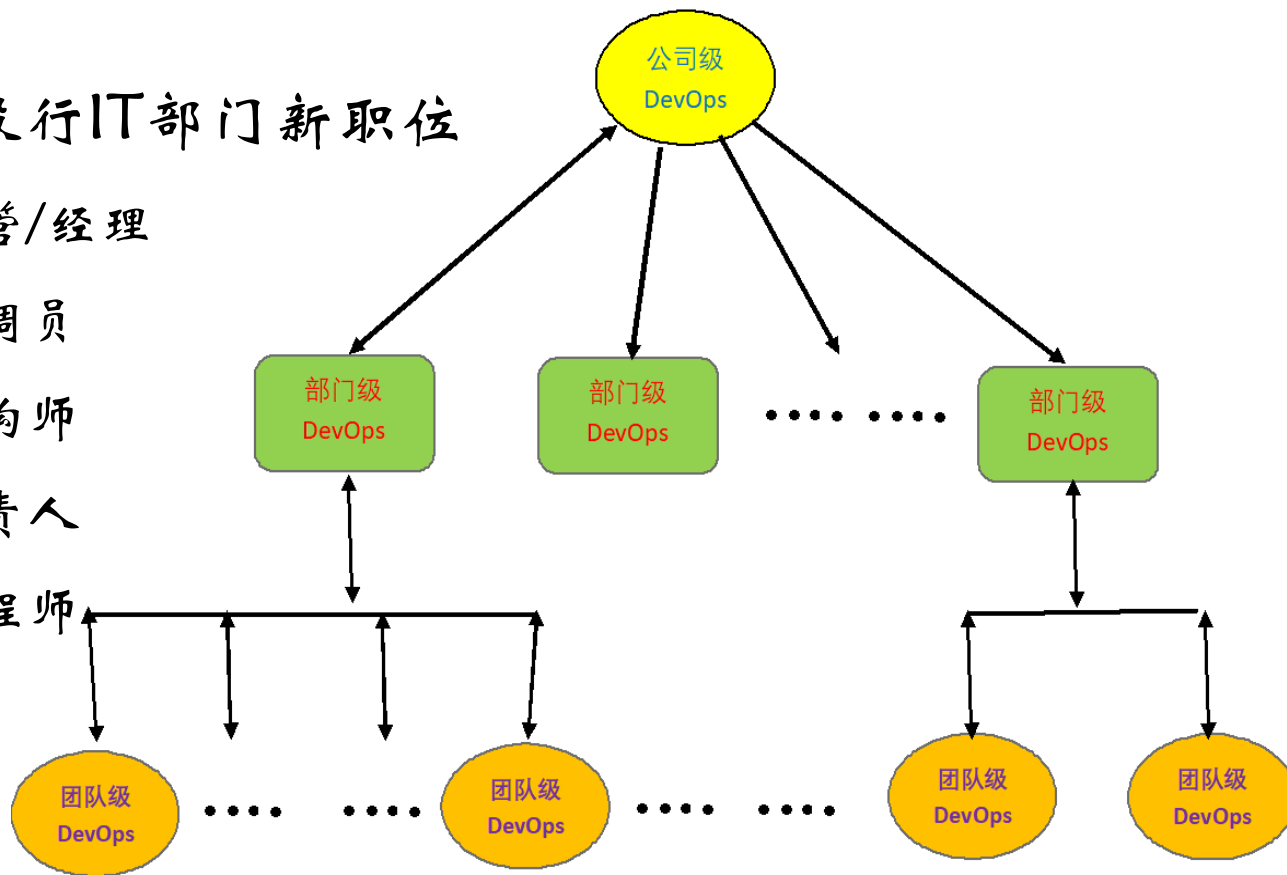
- 巴克莱银行投行外汇IT部门



完善DevOps工作模型，结构和职责

➤ 巴克莱银行投行IT部门新职位

- DevOps主管/经理
- DevOps协调员
- DevOps架构师
- DevOps负责人
- DevOps工程师



Mindset的建立和培养



- DevOps的mindset 是一种
 - 新的思维模式
 - 好的职业素养
 - 拥有更多技能的个体
- DevOps的最终目的是实现开发人员，开发团队以及整个开发部门的思维的转变，技能的提升，并且形成新的或者更好的职业素养
- 难点 - 技术易实现，人难改变
- 案例
 - 巴克莱银行伦敦投行IT部门 - 购买大屏显示器
 - 巴克莱银行伦敦大宗经济IT部门 - 代码质量分析



DevOps社区和资源共享

➤ 目的

- 共享资源和经验
- 总结知识和最佳实践
- 互相帮助

➤ 巴克莱DevOps社区

- 持续集成社区
- 测试社区
- 发布管理社区
- 监控管理社区





THANKS