



# 2021 China DevOpsDays

— 4月17日 · 上海 —

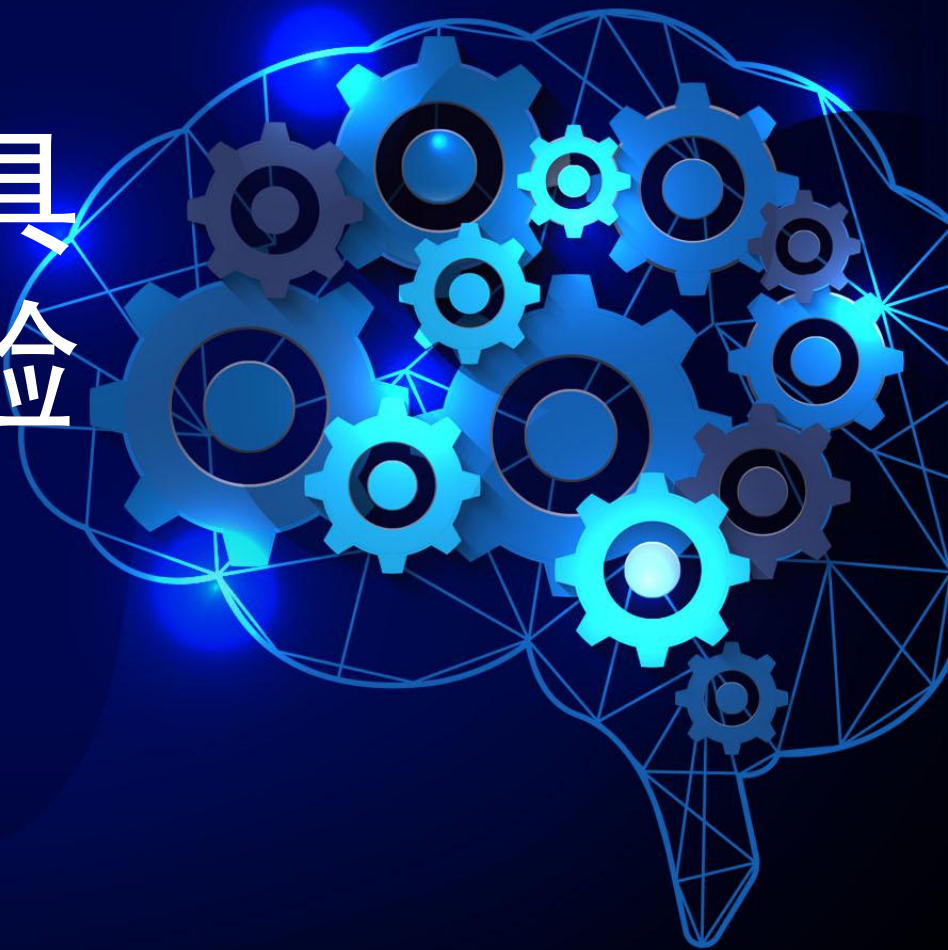
## 互联网研发效能方法工具 落地金融行业的实践经验

王一男

主办方:



BEST PRACTICE



## 自我介绍

王一男



腾讯 — DevOps产品专家

前

百度 — 资深产品经理

开源中国 — 产品总监

# 2020 China DevOpsDays





互联网行业涌现出大量敏捷、DevOps优秀实践，显著提升了研发效能

“如何将前沿的研发管理、软件工程实践与工具与不断变化的业务需求快速结合，提供具有市场竞争力的金融服务”已成为银行科技部门面临的一大挑战。

和互联网行业相比，金融科技在研发管理，工程效能方面的难、关、痛在哪里？

我们发现了相同点，和更多不同点。感受到了“隔行如隔山”般的难度和挑战



# 目的

深入了解两个行业在研发管理、工程效能方面的特点，以及背后的深层原因

发现敏捷、DevOps跨行业“赋能”过程中遇到的问题与解决方案

展望研发效能领域在两个行业中未来发展趋势





- 互联网行业与金融科技的研究特点
- 互联网行业与金融行业研发主要环节的比较
- 互联网行业与金融行业研发管理、工程效能发展趋势



# 互联网行业与金融科技的研发特点

在研发管理和工程效能领域，互联网和金融科技越走越近

- 业务和产品遇到的挑战相似
- 产品所使用的技术相近
- 研发人员规模相近
- 研发管理、工程效能遇到的问题和挑战相近

两个行业在软件研发管理的方式方法、软件工程实践方面越走越近

我们也发现了两个行业在研发效能领域有很深的“误解”与“鸿沟”



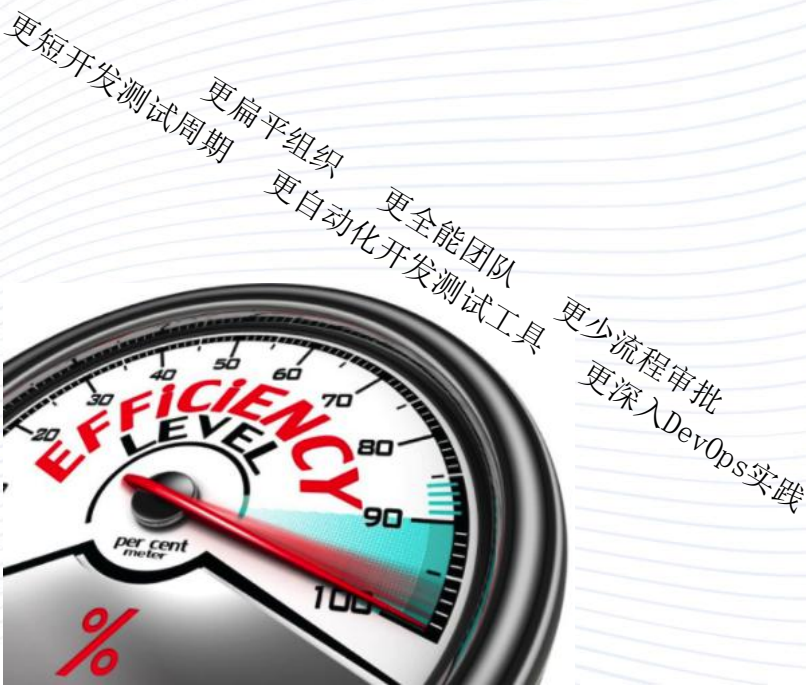


# 互联网行业与金融科技的研究特点

互联网研发不懈追求“更高效率”，金融科技更加注重“风险管理”

互  
联  
网

天下武功,唯**快**不破  
专注 极致 口碑 **快**  
**拒绝** me too me **later**  
.....



“更高效率” VS “风险管理”

两个行业在研发效能目标上的不同  
导致了从研发方法理念、研发工具平台建设思路、  
到具体管理实践、工程实践上的显著不同

《银行业金融机构重要信息系统投产及变更管理办法》  
《商业银行业务连续性监管指引》  
《商业银行信息科技**风险管理**指引》  
《银行业金融机构信息科技外包**风险监管**指引》  
《金融行业信息系统信息安全等级保护实施指引》  
《网上银行系统**信息安全**通用规范》  
《银行业金融机构信息系统**风险管理**指引》  
《关于进一步加强银行业金融机构信息科技**风险保障**工作的指导意见》  
《电子银行**安全评估**指引》  
《关于促进互联网金融**健康发展**的指导意见》  
.....

金  
融  
科  
技



# 互联网行业与金融科技的研究特点

互联网公司要解决研发管理“新问题”，金融科技要迎接工程效能“新挑战”

新问题

互  
联  
网



在互联网公司普遍落地的“敏捷精益”的管理实践与工具，以及以“自动化一切”为目标的DevOps工程实践与工具，在“赋能”传统行业时遇到“水土不服”的问题。

- 无人车
- AI硬件
- toB项目

新挑战

金  
融  
科  
技



传统银行怎样在现有体制下既不断加强风险管理能力，又要更快更好地满足业务发展需要，是金融科技面临的“新挑战”

- 互联网金融







# 研发环节比较 —— 需求/项目/产品管理

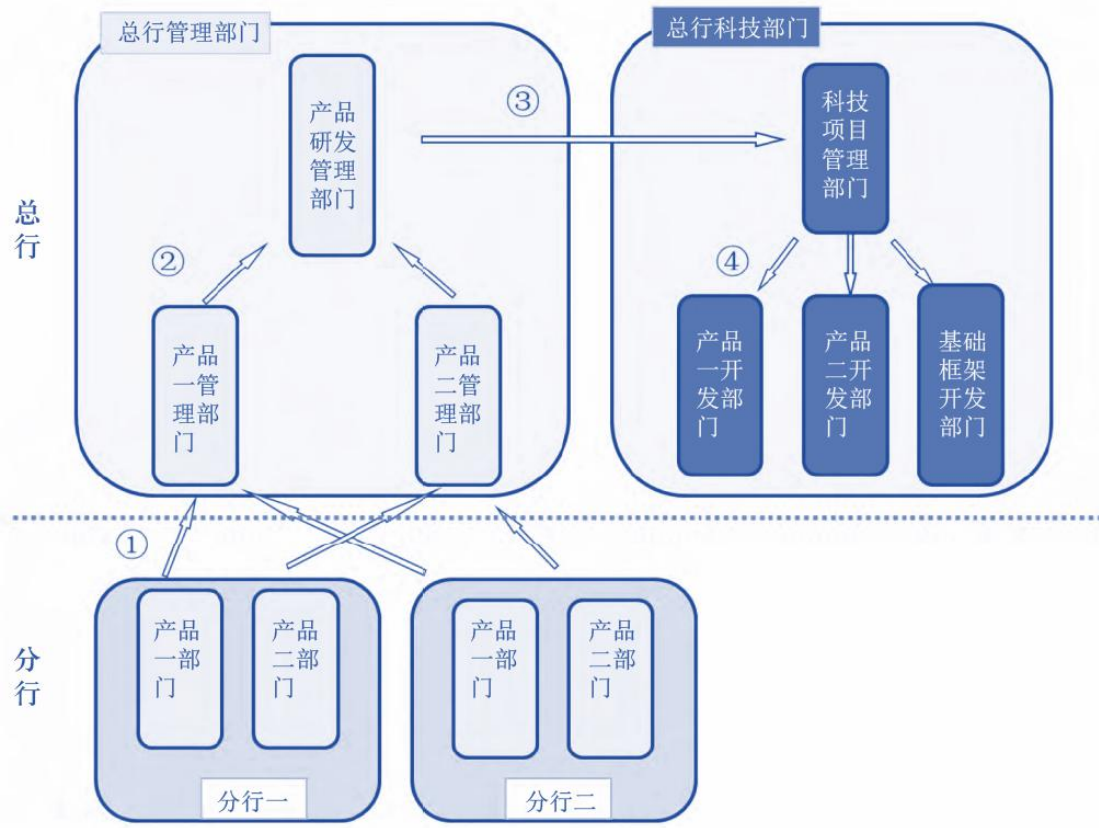
互联网公司



- 一般把“业务”和“研发”放在同一个BU/BG下面
- 基础技术独立一个部门，横向提供基础技术能力
- 团队目标一致，结果导向

组织结构

银行



- 一般分为“业务部门”和“科技部门”
- 科技部门既要研发新产品，又要维护基础架构
- 研发链条长，环节多

# 研发环节比较 —— 需求/项目/产品管 理 管理方式

## 产品管理

产品全生命周期

团队持久性地负责产品

通常以小团队的形式出现，产品经理更接近终端用户

多功能团队，密切融合，部分彼此

需要业务和技术的“多面手”

和经营利润密切相关，共背业务目标

需要尽可能解耦关联性

不断优化产品特性，不断优化管理流程，不断优化团队效能

## 实施内容

团队持久性

团队组织模式

与业务部门关系

人员能力

与企业经营关系

对技术架构的要求

组织可持续发展

## 项目管理

时间、资源、质量要求内完成项目范围

临时组织，共同完成阶段性项目目标

在金融IT领域通常表现为弱矩阵，各职能单位负责职责内工作

需要相对明确的输入（需求文档、立项申请）

需要管理和协作能力强的人员

与业务经营距离较远

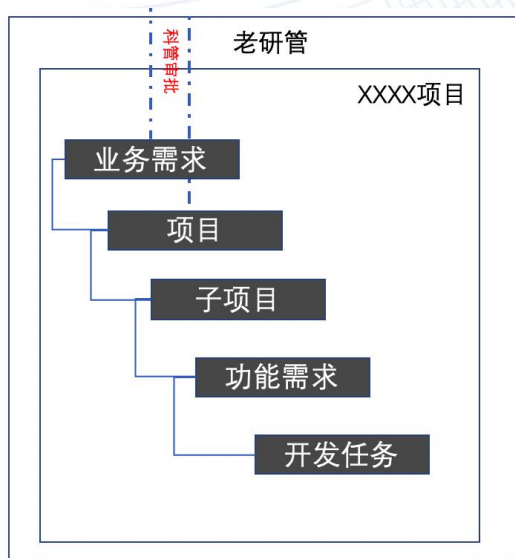
技术架构交织复杂情况下，由项目经理统筹协调各方

关注短期项目目标，不利于沉淀技术和业务组件



## 面临问题

- **互联网：缺乏项目管理管控实践经验和必要的工具支撑**
  - 例如：无人车、AI、公有云
  - 方法：项目管理 PMBOK CMMI TMMI
  - 工具：在已有研发管理基础上，增加文档协同、基线管理、增加项目管理等功能
- **金融科技：研发效率赶不上业务需求增长速度**
  - 科技管理流程和工具：项目管理、固化流程、控制风险
  - 研发管理流程和工具：统一需求、开发、测试管理、向产品管理、敏捷管理方面建设

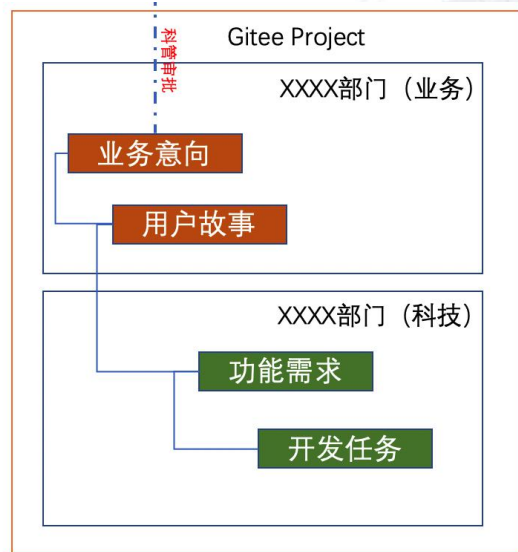


卡片类型减少

工作空间分开

审批过程简化

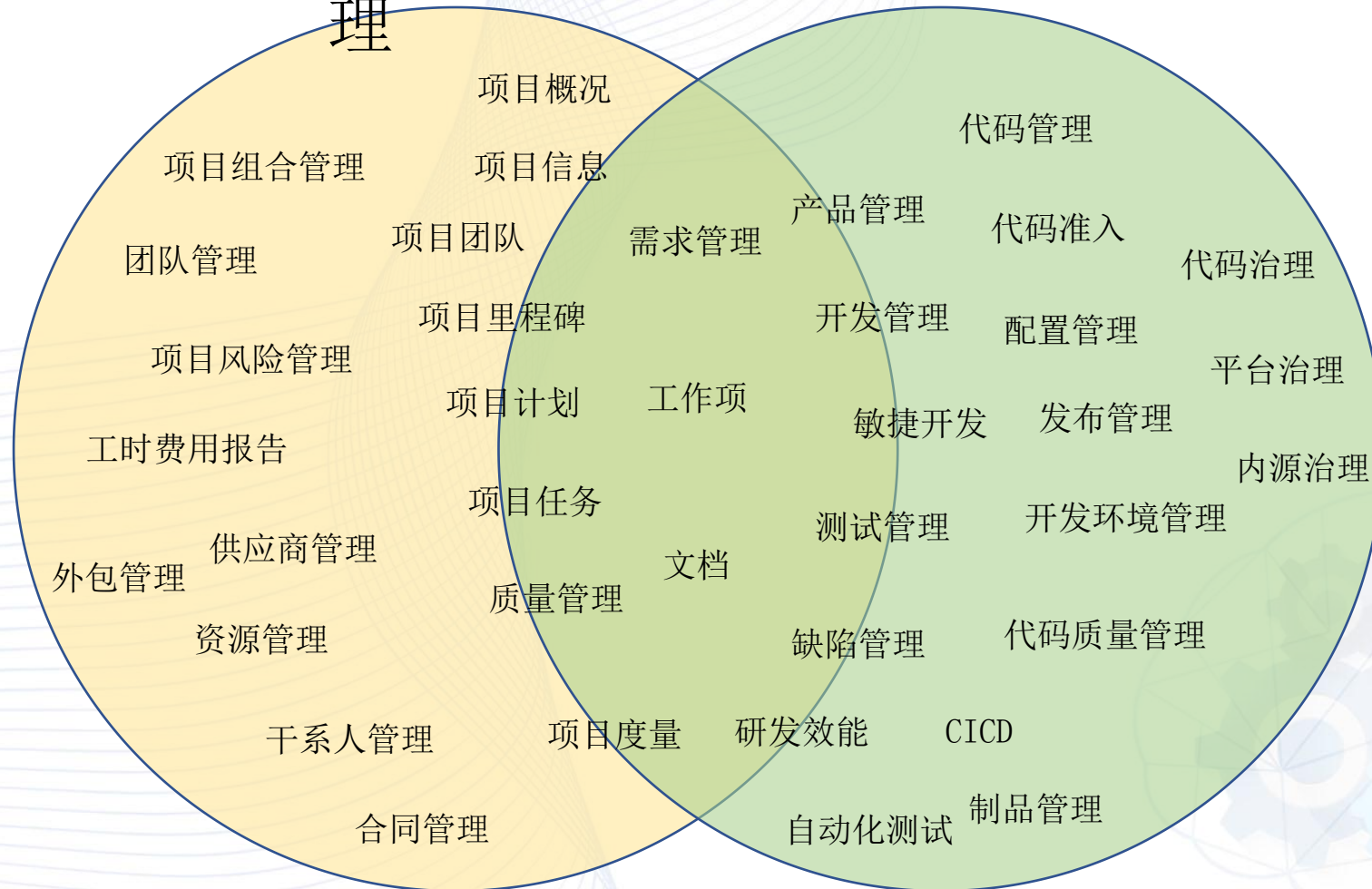
流程状态精简





发展趋势

金融科技研发管理



互联网研发管理



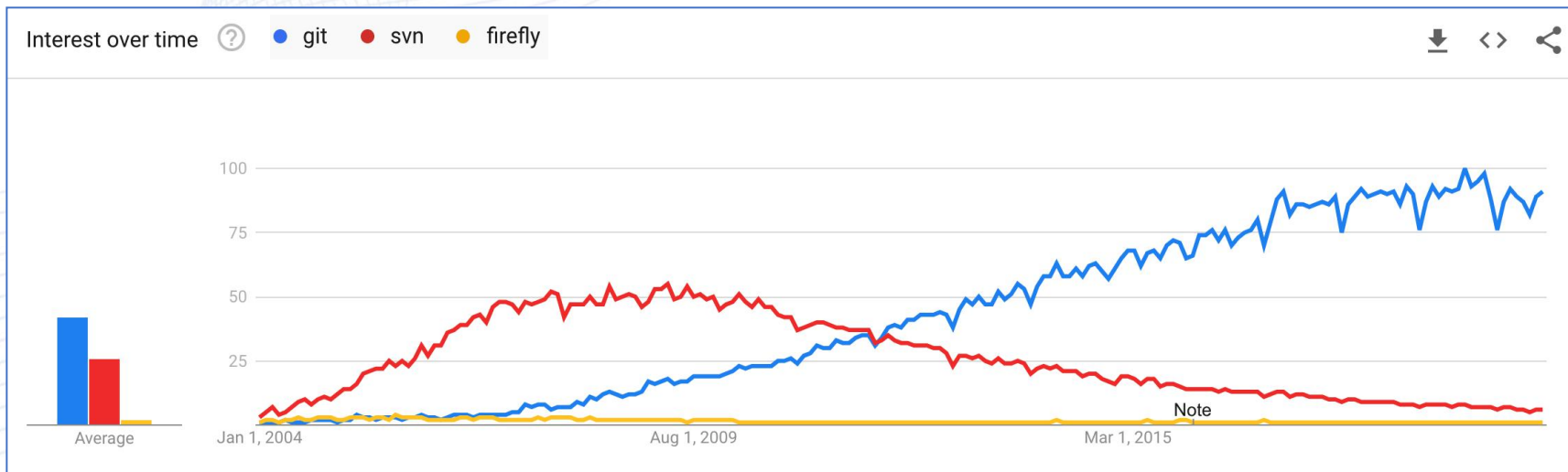
我们在服务银行客户的过程中，在项目管理产品上加强了字段、权限、工作流的能力  
打通测试管理、对接科技管理系统以适应银行客户需求

对版本管理工具认知的不同

### 版本控制工具使用趋势

#### 谷歌指数 - 全球

2012年以后 Git 超过 SVN



#### 百度指数 - 中国

2016年以后 Git 超过 SVN

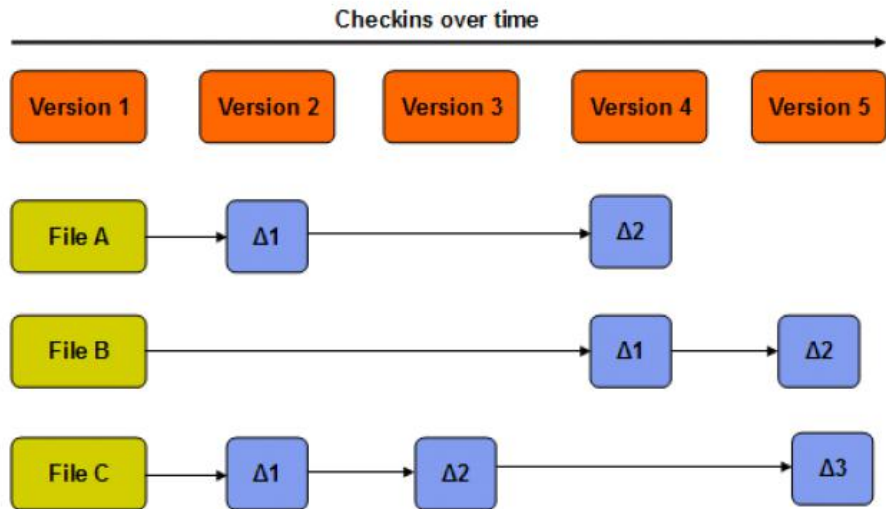


## 对版本管理工具认知的不同

Git 对文件版本管理和其他VCS系统对文件版本的实现理念完全不一样

**集中式****速度慢、存储大****分支“昂贵”****svn、firefly、clearcase**

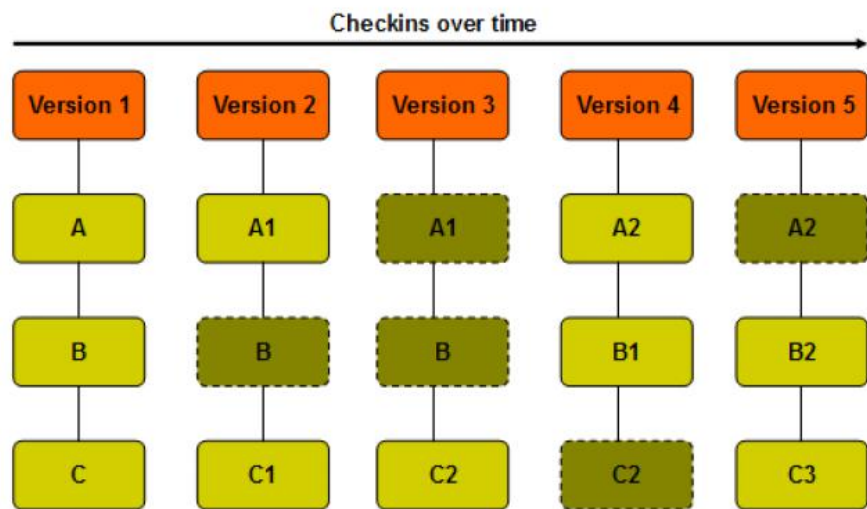
SVN: 记录每个文件在每个版本下的delta $\Delta$ 改变



版本V1到版本V2的更迭过程中，会复制版本V1的没有改的文件FileB，同时算出新的FileA和FileC，覆盖原来的FileA和FileC。V2到V3到Vn同理

**分布式****速度快，存储小****分支非常“廉价”**

Git: 每次提交都是全库的快照

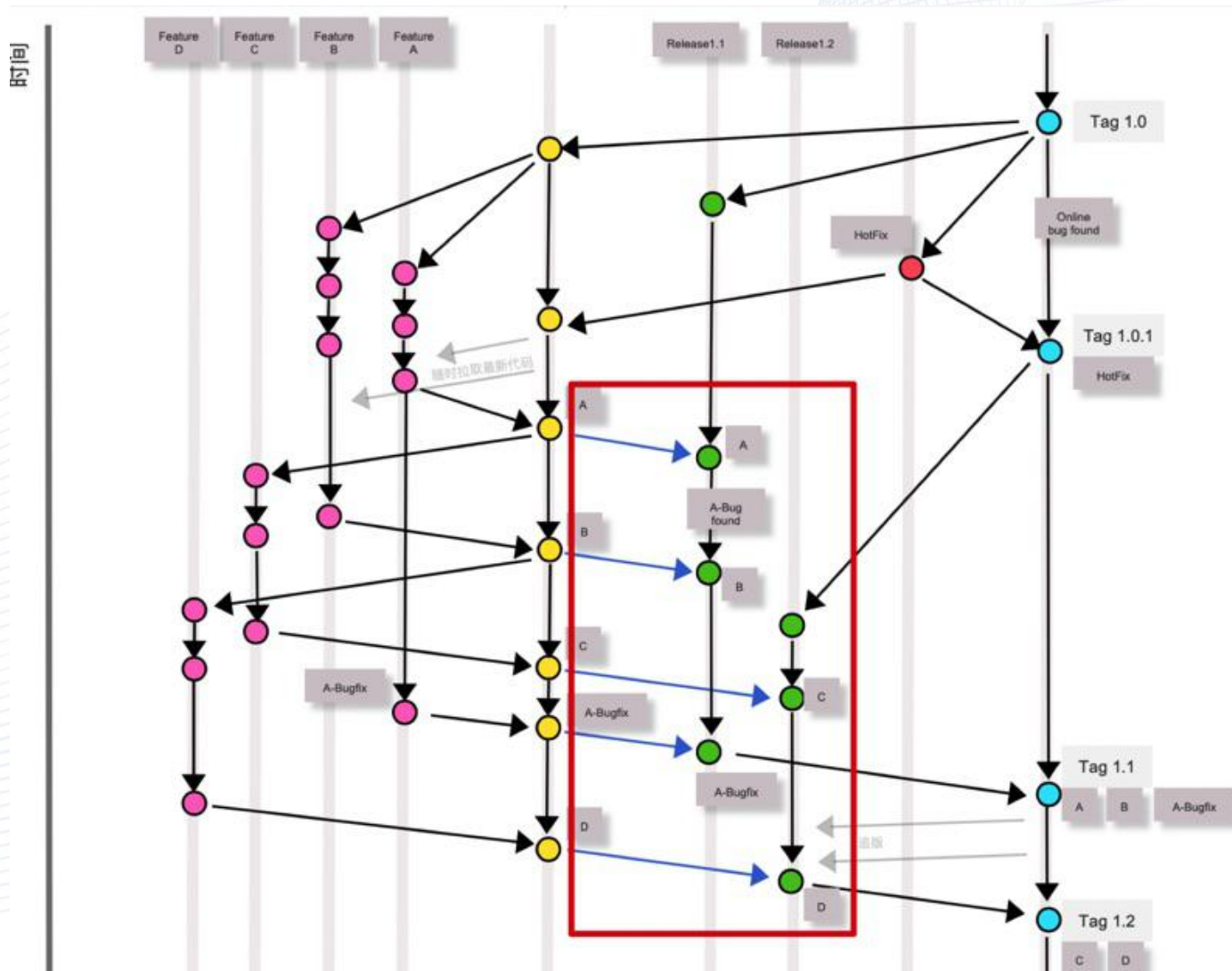


Git 对文件版本的管理理念却是以每次提交为一次快照，提交时对所有文件做一次全量快照，然后存储快照引用。Git 在存储层，如果文件数据没有改变的文件，Git 只是存储指向源文件的一个引用，并不会直接多次存储文件。



对版本管理工具认知的不同

## Cherry-Pick 还是 Merge ?



熟练掌握Git的特点，设置适应Git以及银行研发流程的分支管理策略

我们在产品中，根据银行客户需求，增加新功能

- 文件、文件夹权限
- 分支权限
- 限制文件类型提交
- 在线Cherry-Pick
- 统计分析

## 配置管理角色、流程和工具不同

互联网：

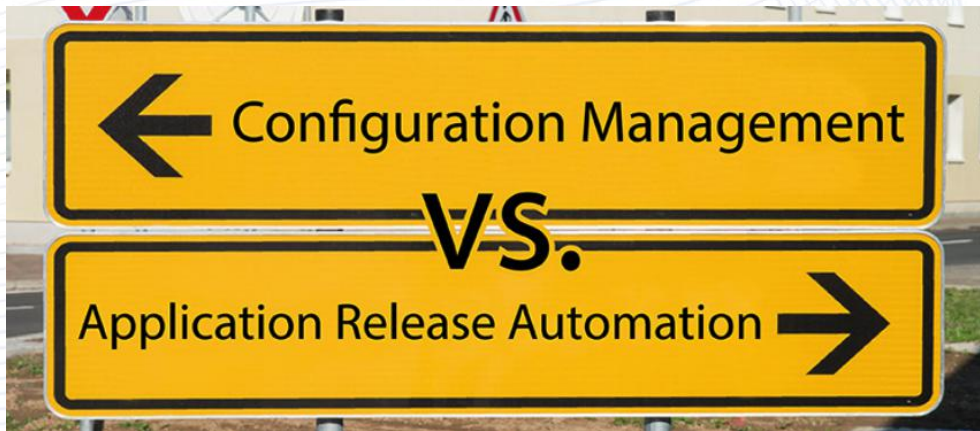
配置管理工程师职位人数越来越少

- 团队较小，自己定义版本，与其他系统（团队）解耦。
- 资源环境较丰富
- 监管审核较少
- 自动化变更内容分别记录在多个系统中

金融科技：

一般都有配置管理岗位以及配置管理工程师

- 团队规模较大，系统之间耦合较多
- 资源环境比较紧张
- 受监管审核
- 变更过程要进行变更单流转审批，变更内容，审批过程要记录在同一个系统中。



根据客户需求我们设计并开发了ARM产品，在Git基础上实现了统一版本管理和版本变更管理

- 变更项管理（代码Commit 和PR）
- 变更单管理
- 变更管理（变更单、变更项批量操作）



## 对代码质量的认识不同

互联网：质量是开发人员保证的，这种思想落地在代码工具中，增加了许多自动化检测工具

质检前置，代码提交合入前：

冲突检查

编码规范

缺陷检查

MI指数（圈复杂度、注释...）

重复文件

提交前构建流水线（编译、**UT..**）

**Review App**

人工评审

代码库 > hui-service > 评审 > 10060484 > 文件变更

搜索全部

Mario-15332 [Bug] 头图负责人条件不影响总数据量 ...

2019-08-06 10:29:03 更新代码 10060484 (下载) → master

失败 3 项检查失败，其中 2 项阻碍合入

- 评审状态 当前评审基于已废弃的评审，无法合入，解决方法 block
- 缺陷检查
- 编码规范
- 可维护性 检测到 1 个函数发生变更/新增，其中有 1 个函数的 MI<20，详情：MI报告
- 安全检查
- 持续集成-ChangePipeline block

待评价 人工 0条评论

缺少评审人+2评分

文件变更 提交记录 讨论动态 4

问题类型 <<

可维护性 2

变更文件 1/3 ☐ 问题代码

BASE 1

展开全部 63 行...  
向上展开 10 行...

```
64 // 匹配AND OR 但是不匹配中括号[*AND*OR*]，用到了java正则负向前瞻
65 public static Pattern MIA_QUERY_MISD = Pattern.compile("(?!(AND|OR))([*AND*OR*])");
66
```



### 对代码质量的认识不同

金融科技：质量是测试人员保证的，代码扫描工具仅个别团队自己搭建使用

这么多自动化检查，是否会降低效率？

质量风险前置是提高整体研发效能、提高产品质量的最有效手段

解法：成立整体负责研发效能的组织 或成立“准入处”



**DevOps and  
Agile  
Fail Fast  
Fail Early  
Fail Often  
Fail Safe**

互联网：

- 统一环境（线下线上）
- 云原生流水线调度资源和环境
- 测试、部署平台 -> 流水线插件
- 一条流水线实现自动化 + DevOps流程管控

金融科技：

- 正在环境统一的过程中
- 使用Jenkins，已经遇到瓶颈，并寻找解决方案
- 脚本 -> 流水线插件
- 自动化 与 流程分开

### 流水线落地实践

1. 云原生资源调度执行，大幅提升资源利用率；
2. 兼顾流程管理，不同角色的门禁管理在同一条流水线上实现





互联网：重点做效能改进

金融科技：重点做外包团队管理

互联网：

- 以工程数据为主（需求、开发、准入、测试、部署），实时度量团队的功能能力
- 数据量大，准确度高，应用范围广

金融科技：

- 以项目管理的数据，工时的数据为基础进行度量改进
- 想要更多数据，就要更多字段，流程节点，效能就越低
- 数据都是人工填写，准确度不高

度量改进落地实践：通过工程数据客观度量外包团队能力

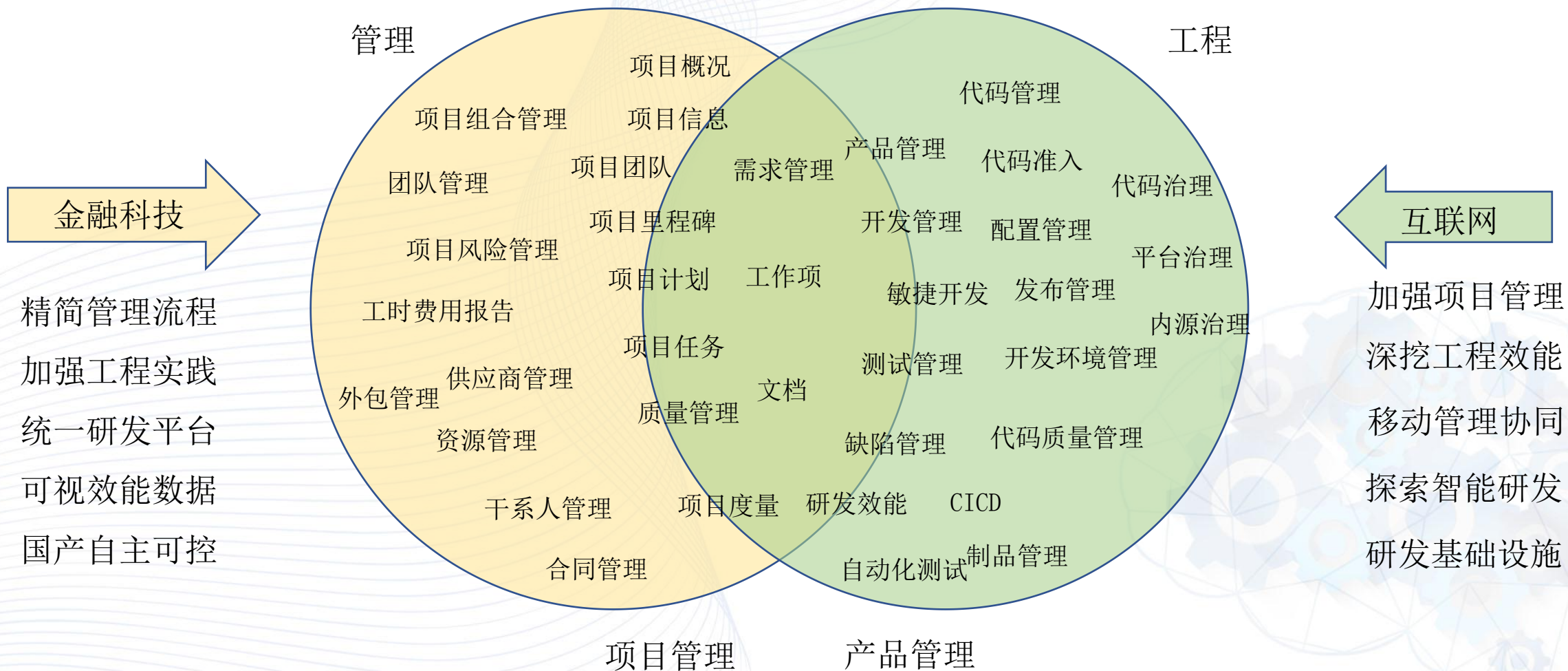


工程能力地图								
工程能力素质 技术债务 效率与质量								
PDB-L1	代码准入							查看详情
	CR-千行评论数	CR-打回率	有效CR占比 (代码库)	增量单元测试占比 (代码库)	增量单元测试覆盖率	机检高危BUG修复率 (增量)	机检高危量BUG行BUG数	
需求管理	6.09	0.00%	100.00%	12.50%	0.00%	100.00%	0.099	详情
开发管理	2.45 ↑	0.50%	94.68%	9.57%	0.47%	80.30% ↑	0.12	详情
代码准入管理	2.08 ↑	0.80%	100.00%	83.33%	10.33%	0.00%	0.00	详情
测试管理	2.53 ↑	0.00%	100.00%	30.00%	0.57%	0.00% ↓	0.15	详情
部署变更管理	0.62	0.22%	100.00%	25.00%	0.011%	72.73% ↑	0.16	详情
流水线管理	8.12	0.16%	100.00%	71.88%	2.14%	53.33%	0.10	详情
需求	1.02	0.00%	95.74%	4.28%	0.0024%	45.45%	0.27	详情
开发	1.37	1.00%	100.00%	88.00%	12.06% ↓	63.64% ↓	0.23	详情
代码准入	2.75 ↑	0.30%	100.00%	4.35%	0.87%	100.00%	0.15	详情
测试	0.51	0.58%	98.48% ↓	13.84%	2.13%	94.74% ↑	0.012	详情
部署变更	2.88	0.30% ↓	100.00%	0.00%	0.00%	100.00%	0.15	详情
流水线	4.97	2.34%	100.00%	84.62%	3.72% ↑	87.10% ↑	0.45 ↓	详情
公司平均值	2.52	0.87%	93.87%	47.56%	3.33%	58.28%	0.80	-
Q4目标值	2.00	5.00%	100.00%	50.00%	30.00%	80.00%	0.32	-





## 发展趋势





- 互联网应该更多学习金融科技在项目管理、风险管理方面的实践经验，以适应未来toB业务的发展
- 金融科技在正在积极学习互联网研发管理（敏捷）、代码管理、CICD、效能度量的整套研发方法和体系，从而在产品研发方面更能适应业务快速发展的需要

## 研发工具产品方面

- 我们国内产品与国外领先产品仍有非常巨大的差距
- 更多投入到研究和产品设计中来，使国内软件研发管理、工程工具平台更快、更好建设







# 2021 China DevOpsDays

— 4月17日 • 上海 —

# Thanks

## 感谢聆听

