



Devops统一研发体系建设方案

技术创新 变革未来

1

背景介绍

2

DevOps实践及研发平台建设

3

经验和教训



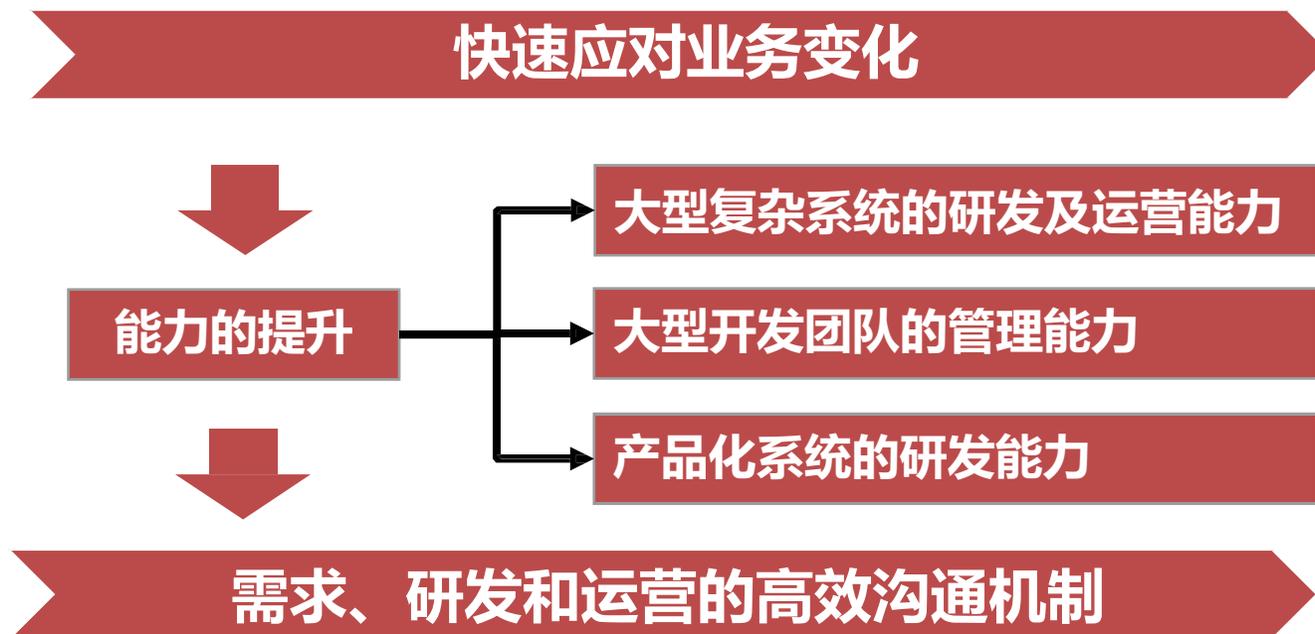
行业自主研发的要求及面临的问题

- 行业信息系统服务商尽管有很强的能力，但已经不能满足行业的个性化和快速响应业务变化的需求。
- 互联网金融和金融科技背景下，对自主研发能力提出了更高的要求。
- 信息系统的自主研发能力，在一定程度上决定了业务能开展的广度和深度，长远看决定了传统金融行业在互联网时代的生存能力和创新发展能力。

开发人员	项目管理	应对业务变化	产品化开发	开发能力
<ul style="list-style-type: none"> • 成本高、招聘难、流动大 • 大型团队管理难以沿用和小型团队方法 • 外包人员难管理 	<ul style="list-style-type: none"> • 代码未妥善管理，文档、代码注解不全，缺少开发管理历史资料 • 和服务商合作开发过程，不能在甲方管理 	<ul style="list-style-type: none"> • 低价值重复性工作多，交付效率低 • 业务部门不了解需求开发进度 • 管理层不知道IT在干什么 	<ul style="list-style-type: none"> • 跟着零碎业务需求跑，不重视分析和设计，没有统一的产品设计和架构把控 • 欠缺版本管理 	<ul style="list-style-type: none"> • 面对大型复杂系统开发，无从下手 • 系统更新换代，兼顾新系统开发和老系统维护，困难重重



项目及平台目标



项目总体介绍

项目简介

- 采用敏捷和DevOps理念
- 整合软件研发工具、容器化技术、运营监控工具
- 解决了研发中的代码分支管理、环境获取、自动化、微服务化、持续交付等突出难题
- 支持博时基金IT团队实现敏捷和DevOps转型
- 在团队规模和核心能力上实现跨越式提升，适应互联网时代对自主研发能力的要求

实施历程

- 2016年1月开始调研、7月一期上线，支持全部50多个项目的生命周期管理
- 持续进行敏捷和DevOps研发管理的转型，团队能力跨越门槛
- 支持新一代大型业务复杂的项目、产品化开发、按计划交付

观念：从传统向敏捷的转变



敏捷宣言

- > 个体和互动高于流程和工具, Individuals and interactions over processes and tools
- > 工作的软件高于详尽的文档, Working software over comprehensive documentation
- > 客户合作高于合同谈判, Customer collaboration over contract negotiation
- > 响应变化高于遵循计划, Responding to change over following a plan



DevOps三步工作法

建立全局观

建立反馈

持续改进

- 
- 应用生命周期系统建设
 - 量化管理能力建设
 - 指标体系建设
 - 报表体系建设

- 配置管理能力
 - 自动化能力
 - 工艺改进团队建设
 - 内部知识库建设
- 

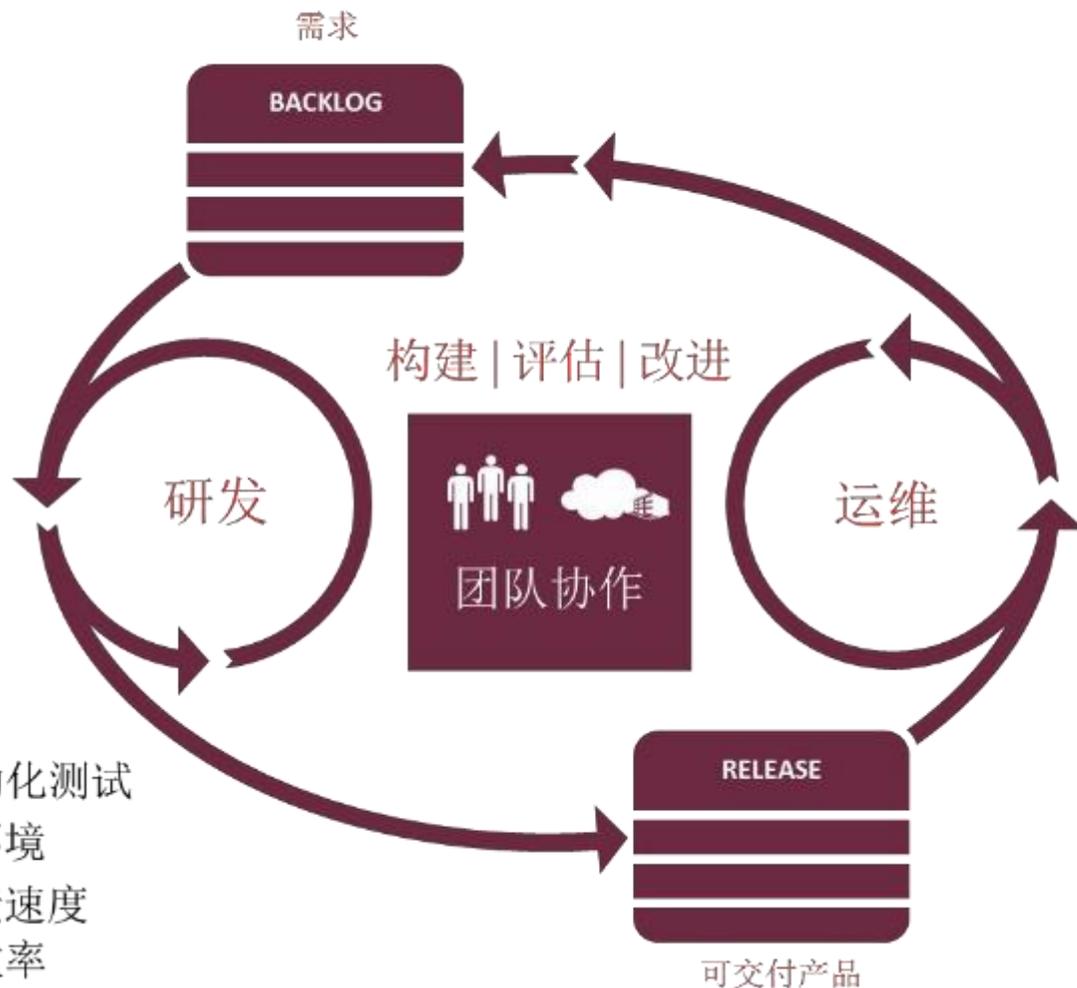


建立研发端到端协作能力



规划

- ✓ 需求管理流程
- ✓ 项目管理流程
- ✓ 迭代规划流程
- ✓ 报表和监控反馈



运维

- ✓ 引入容器编排平台
- ✓ 实现应用无停机更新
- ✓ 启动微服务架构改进试点
- ✓ 容器监控能力建设
- ✓ 持续反馈能力建设



研发

- ✓ 完善配置管理流程
- ✓ 引入单元测试和自动化测试
- ✓ 持续集成提供调测环境
- ✓ 加速开发和测试反馈速度
- ✓ 提升测试环境获取效率



发布

- ✓ 全面启用CI/CD流程
- ✓ 引入Docker容器化技术
- ✓ 环境标准化统一管理
- ✓ 完善应用发布审批流程
- ✓ 减少人工参与

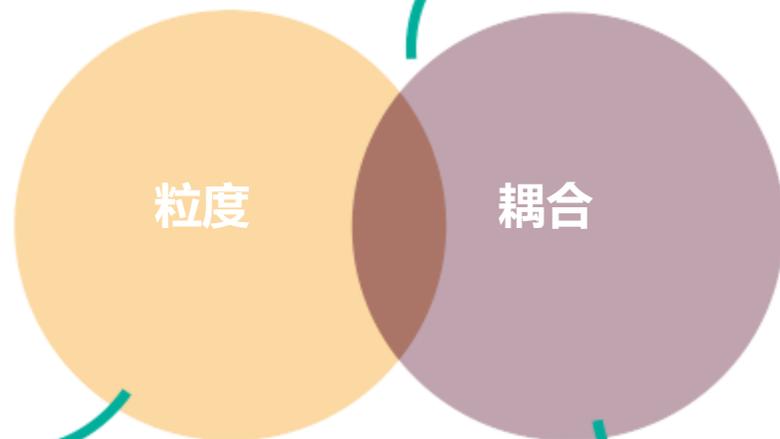


实施落地 – 粒度&解耦

提升效率

无论是敏捷，精益或者持续交付，其最终目的都是为了提升效率。

所谓“效率”，就是单位投入的产出量。



管理粒度

DevOps从管理角度的优化永远是在通过控制“管理单元”的粒度来完成的。

所谓的“管理单元”可能是团队，需求，任务，测试，交付物等任何研发中的被管理对象

工程解耦

DevOps 从技术角度的优化永远是在通过解除“工程对象”之间的耦合实现的。

所谓“工程对象”可能是系统，工具，代码，模块，服务，平台，云或者任何在研发过程中存在或者交付的“技术对象”。

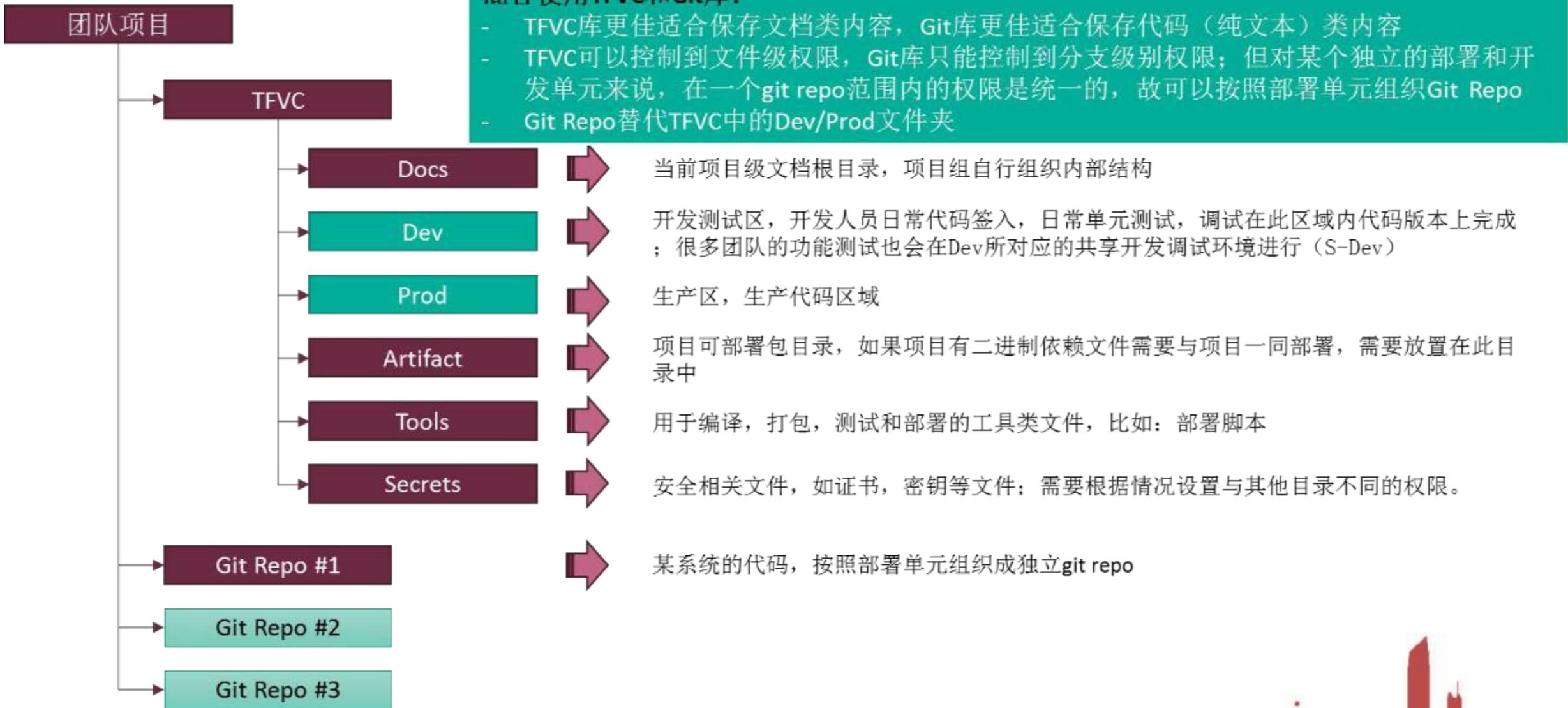


实施框架

业务系统	原业务系统	新一代业务系统
	统一技术架构 (BOFA)	
管理	分析、设计、开发、测试、构建、集成	环境、部署、发布、运行
方法	项目管理、配置管理、持续集成、自动化测试	持续部署、容器化、运行监控、日志管理
平台	TFS, 生命周期管理、数据仓库、知识库	
工具集成	EA、IDE、Maven、Loadrunner、Sonar、Junit、Selenium、Postman、自研工具.....	Docker、Harbor、NetGain、APM、DPM、EM、日志审计、自研工具.....
基础设施	基础设施即代码	



基础:配置库文件结构

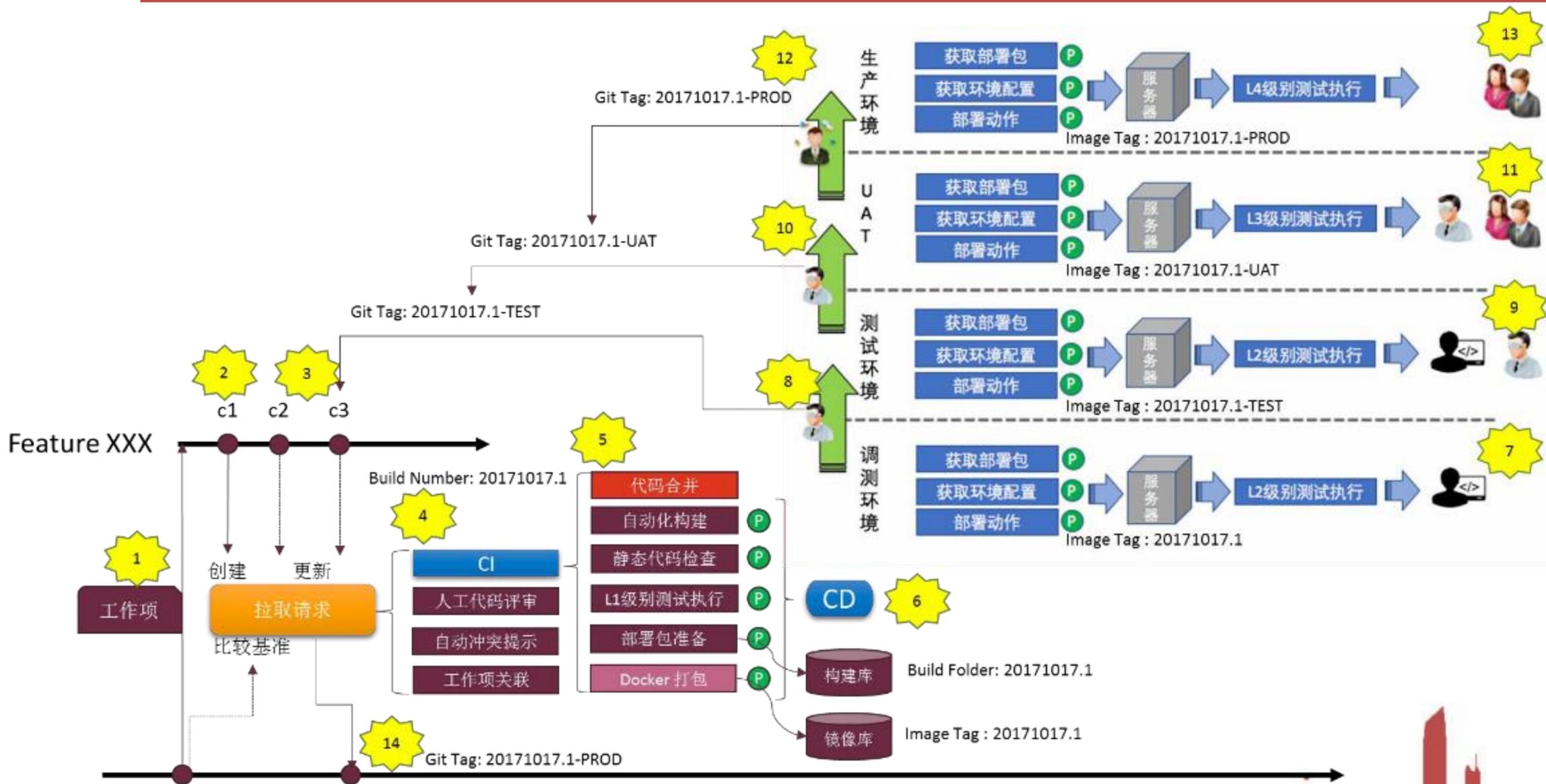


混合使用TFVC和Git库:

- TFVC库更佳适合保存文档类内容，Git库更佳适合保存代码（纯文本）类内容
- TFVC可以控制到文件级权限，Git库只能控制到分支级别权限；但对某个独立的部署和开发单元来说，在一个git repo范围内的权限是统一的，故可以按照部署单元组织Git Repo
- Git Repo替代TFVC中的Dev/Prod文件夹



Git分支模型与CI/CD流水线



基于TFS的CI配置

Visual Studio Team Foundation Server 2015 / 新版门户

定义 / portal | 生成

生成 选项 存储库 变量 触发器 常规 保留 历史记录

添加生成步骤...

Docker Compose - Maven Package

Docker Registry Connection harbor-test

Docker Compose File **docker/docker-compose-ci.yml**

Additional Docker Compose Files

Environment Variables

Project Name \$(Build.Repository.Name)

Quality Image Names @

Action Run a Docker Compose command

Command up

Advanced Options

控制选项

已启用

出错时继续

始终运行

More Information [g](#)

Visual Studio Team Foundation Server 2015 / 新版门户

portal-service | 资源管理器 历史记录 分支 拉取请求

master | docker-compose-ci.yml 成功

内容 历史记录 比较

路径 (/) /docker/docker-compose-ci.yml

编辑 下载

```

1 version: '2'
2 services:
3   ci-build:
4     image: harbor-test.boshi.com.cn/library/maven:3.3.9-jdk-8
5     volumes:
6       - ../src
7     working_dir: /src
8     command: /bin/bash -c "mvn clean package"

```

使用docker容器完成构建环境获取

Visual Studio Team Foundation Server 2015 / 新版门户

portal-service | 资源管理器 历史记录 分支 拉取请求

master | docker-compose-prod.yml 成功

内容 历史记录 比较

路径 (/) /docker/docker-compose-prod.yml

编辑 下载

```

1 version: '2'
2 services:
3   service:
4     image: harbor.boshi.com.cn/portal-prod-service:#{Build.BuildId}#
5     volumes:
6       - /var/portal-service/logs:/var/portal-service/logs
7     restart: always
8     ports:
9       - "81:8080"
10    environment:
11      - SPRING_PROFILES_ACTIVE=prod

```

使用Build.Id跟踪容器镜像版本

基于TFS的CD配置

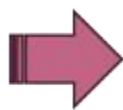
The screenshot shows the Visual Studio Team Foundation Server 2015 interface. On the left, the '发布定义' (Release Definition) pane shows a definition named 'service' with three tasks: '发布测试环境' (3/3 tasks), '测试环境-自动化...' (0/2 tasks), and '推送到生产镜像仓库...' (2/2 tasks). The main area shows the configuration for the '发布' (Release) step, including Docker commands and advanced options. On the right, the '服务 / Release-86' (service / Release-86) details pane shows the execution history, including a successful test run in the '测试环境-自动化...' environment.

在部署好的测试环境自动运行UI自动化测试

审批后推送到
生产镜像仓库,
满足测试/生产
隔离需求



Harbor-test



Harbor-prod



4级测试体系建设

L4 只能在生产环境执行的测试

L3 可以在完整系统内使用测试数据执行的测试

L2 可以仅依赖当前部署单元运行的测试

L1 不依赖任何外部环境，当前代码文件存在就可以执行的测试

测试类型	测试环境	测试方式
UI自动化/性能测试 探索测试	生产环境 预生产环境	手动 自动化
UI自动化/性能测试 探索测试/用例测试 API 测试	UAT环境 测试环境	手动 自动化
UI自动化/性能测试 探索测试/用例测试 API 测试	测试环境 调测环境	手动 自动化
单元测试 静态代码检查	持续集成环境	自动化

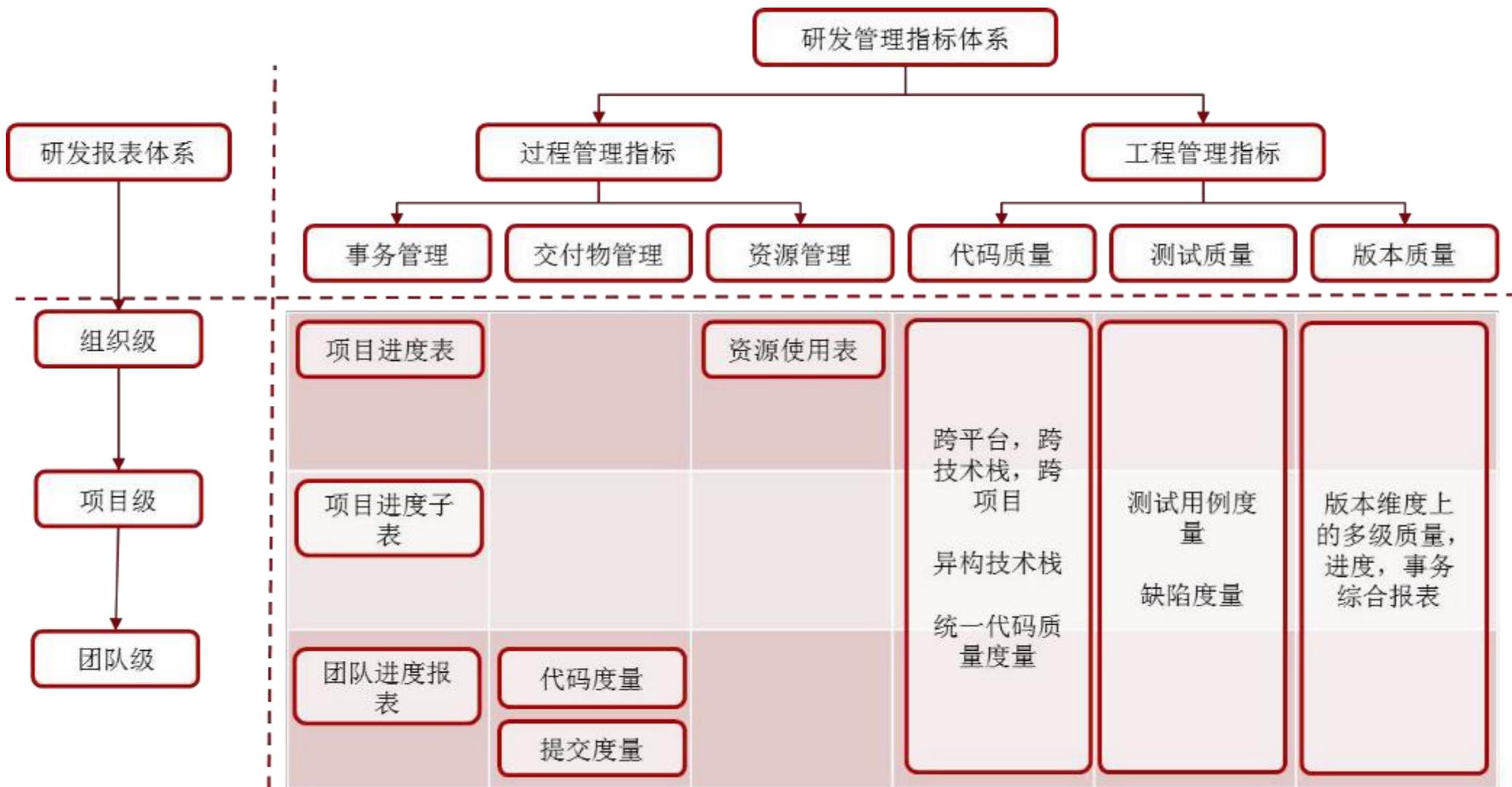


持续交付能力提升

领域	DevOps之前	DevOps之后
版本管理	开发、测试、生产相互独立	多分支开发、主干发布
环境权限	开发人员权限极大	各环境权限隔离
构建	构建环境不统一且分散，每天有效构建不超过10次	构建环境标准且统一，目前每天有效构建次数62次
部署	依赖开发人员，每天各环境发布次数不超过10次	基于自动化发布，目前每天各环境有效部署次数超过80次
发布回退	过度依赖开发人员手工回退，低效且风险巨大	提前制定变更风险等级策略，基于版本进行回退，高效而可靠



研发管理指标体系与报表体系建设



项目进度总览表

博时基金项目进度总览表

完成比: 计划完成百分比 (蓝色), 实际完成百分比 (黄色)
测试结果: 已通过 (绿色), 失败 (红色), 未运行 (灰色)
Bugs: 活动 (蓝色), 已解决 (黄色)

提示:
*1 点击表格各列 (+/- 特性) 可以查看相关子报表
*2 超出几天, 超出几天计划的变更情况(红色字体)

工作类型	编号	需求列表	状态	分析阶段			设计阶段			开发阶段			测试阶段			测试结果	Bugs
				完成比	计划	剩余	完成比	计划	剩余	完成比	计划	剩余	完成比	计划	剩余		
特性	18019	公用模块	活动	N/A	0	0	N/A	0	0	100%	105	2	100%	14	0		
特性	18024	产品系统一期	活动	N/A	0	0	100%	2	0	100%	524	18	96%	238	100%		
特性	18411	账户管理一期	活动	N/A	0	0	N/A	0	0	100%	234	5	98%	50	87%		
需求	18066	1 开户流程管理-开户流程管理	已关闭	N/A	0	0	N/A	0	0	100%	16	0	72%	10	3	77%	2%
需求	18067	2 开户流程管理-绑定任务	已关闭	N/A	0	0	N/A	0	0	0%	8	0	80%	5	1	100%	
需求	18068	3 开户流程管理-绑定	已关闭	N/A	0	0	N/A	0	0	0%	16	0	63%	8	3	100%	
需求	18069	4 开户流程管理-开户申请、受理	已关闭	N/A	0	0	N/A	0	0	100%	16	0	62%	15	5	100%	
需求	18071	5 开户流程管理-材料准备	已关闭	N/A	0	0	N/A	0	0	100%	24	0	91%	32	7	100%	
需求	18072	6 开户流程管理-材料用印	已关闭	N/A	0	0	N/A	0	0	100%	8	0	60%	8	3	100%	
需求	18073																
需求	18074																
需求	18075																

- 特性到需求的属性分解
- 各阶段工时进度
- 测试用例执行情况

博时基金项目进度总览表

完成比: 计划完成百分比 (蓝色), 实际完成百分比 (黄色)
测试结果: 已通过 (绿色), 失败 (红色), 未运行 (灰色)
Bugs: 活动 (蓝色), 已解决 (黄色)

提示:
*1 点击表格各列 (+/- 特性) 可以查看相关子报表
*2 超出几天, 超出几天计划的变更情况(红色字体)

工作类型	编号	需求列表	状态	分析阶段			设计阶段			开发阶段			测试阶段			测试结果	Bugs
				完成比	计划	剩余	完成比	计划	剩余	完成比	计划	剩余	完成比	计划	剩余		
特性	29561	M版改版的需求	活动	100%	12	0	100%	90%	527	92	77%	34%	1115	+130	5		
需求	29551	【M版2017】后台接口(新增)开户需求-基金信息查询关联(6)															
需求	29552	【M版2017】后台接口(交易)开户(5)															
需求	29553	【M版2017】后台接口(交易)开户(4)															
需求	29554	【M版2017】后台接口(交易)开户(3)															
需求	29555	【M版2017】后台接口(交易)开户需求(5)															
需求	29556	【M版2017】后台接口(开户)开户-银行卡、开户、绑定手机关联(6)															
需求	29557	【M版2017】后台接口(开户)开户-银行卡关联(6)	已解决	N/A	0	0	N/A	0	0	56%	48	0	100%	0	N/A	0	0
需求	29558	【M版2017】后台接口(开户)开户-风险评估关联(2)	已解决	N/A	0	0	N/A	0	0	100%	16	1	92%	0	N/A	0	0
需求	29559	【M版2017】后台接口(新增)开户-交易信息查询(9)	已解决	N/A	0	0	N/A	0	0	100%	87	0	100%	0	N/A	0	0
需求	29560	【M版2017】后台接口(新增)开户-持仓及收益关联(5)	活动	N/A	0	0	N/A	0	0	99%	36	4	99%	0	N/A	0	0

- 实际进度与计划工时的偏离情况提示

资源使用情况报表

博时基金项目任务进度表

完成比
■ 计划完成百分比
■ 实际完成百分比

提示：
 #1 点击表格各列 (e.g. 特性) 可以查看相关子报表

工作类型	工作项编号	需求列表	状态	指派给	计划工时	已投入工时	按计划工时	是否按计划完成	完成百分比
特性	29561	M版改版的需求	活动	杨晓黎	46	45	1	✓ 0	97.83%
需求	29551	【M版2017】后台接口 (查询) 开发需求-基金信息	已关闭	杨晓黎	46	45	1	✓ 0	97.83%
			已关闭	周运波	6	6	0	✓ 0	100.00%
			已关闭	周运波	6	6	0	✓ 0	100.00%
			已解决	周运波	6	6	0	✓ 0	100.00%
			已关闭	周运波	6	6	0	✓ 0	100.00%
			已关闭	周运波	6	6	0	✓ 0	100.00%

博时基金项目进度资源子表

提示：
 #1 点击表格各列 (e.g. 特性) 可以查看相关子报表

编号	工作类型	任务标题	指派给	任务状态	初始估计	已完成	剩余
29561	特性	M版改版的需求	杨晓黎	活动			
29610	需求	【M版2017】中后台需求分析设计	杨晓黎	活动			
29617	设计任务	【M版2017】接口文档设计	杨晓黎	已解决	8	0	
29618	设计任务	【M版2017】项目框架搭建	杨晓黎	已解决	8	0	
29619	设计任务	【M版2017】开发规范及迭代流程拟定	杨晓黎	已解决	8	0	
29620	设计任务	【M版2017】中后台逻辑详细设计	杨晓黎	已解决	8	0	
29611	需求	【M版2017】后台接口 (账户类) 开发需求-登录及用户信息相关 (5)	杨晓黎	已解决			
29674	开发任务	【M版2017】后台接口 (账户类) 开发任务-登录	杨晓黎	已解决	4	4	0
29613	需求	【M版2017】数据库开发需求	杨晓黎	已建议			
29643	开发任务	【M版2017】数据库开发任务	杨晓黎	已建议	24	0	
30016	需求	【M版2017】文档阅读、项目搭建、一期 xmind 评审	杨晓黎	已解决			
30026	设计任务	【M版2017】文档阅读、项目搭建、一期 xmind 评审	杨晓黎	已解决	18	18	0
30307	需求	【M版2017】微服务	杨晓黎	活动			
22420	开发任务	【M版2017】微服务-工资理财接口代码复	杨晓黎	已建议	8	0	



直接效益&间接效益

直接效益

- 全部50多个系统、260名技术人员在该平台上进行统一管理，其中最大系统的开发人员规模达120人。整体研发效率提升20%以上

间接效益

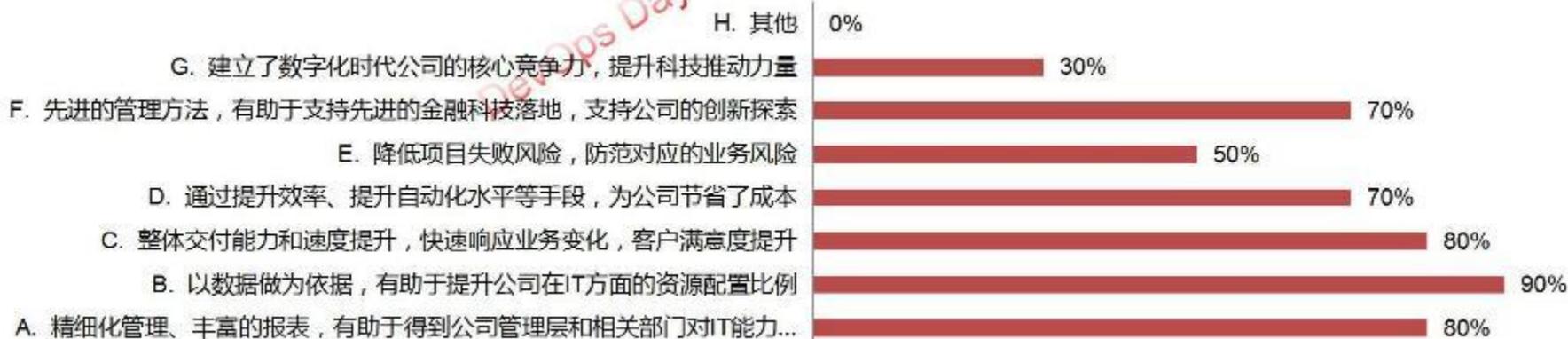
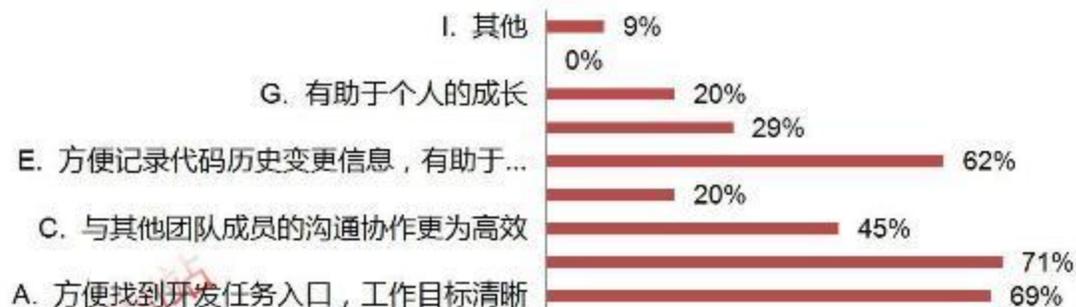
- IT治理水平提升：提升资源配置比例
- 能力提升：大型团队，大型项目，复杂项目，产品化
- 快速应对业务变化：快速交付
- 价值和品质提升：更关注价值创造工作，敏捷方法提升价值和品质，减少技术债
- 成本控制：项目进度、资源可控，精益IT避免浪费
- 提升效率：顺畅的流水线，自动化，管理视图
- 降低风险：项目管理关键因素的风险管理，全部代码纳入管理

内部调查问卷

1. 如何影响您的工作效率



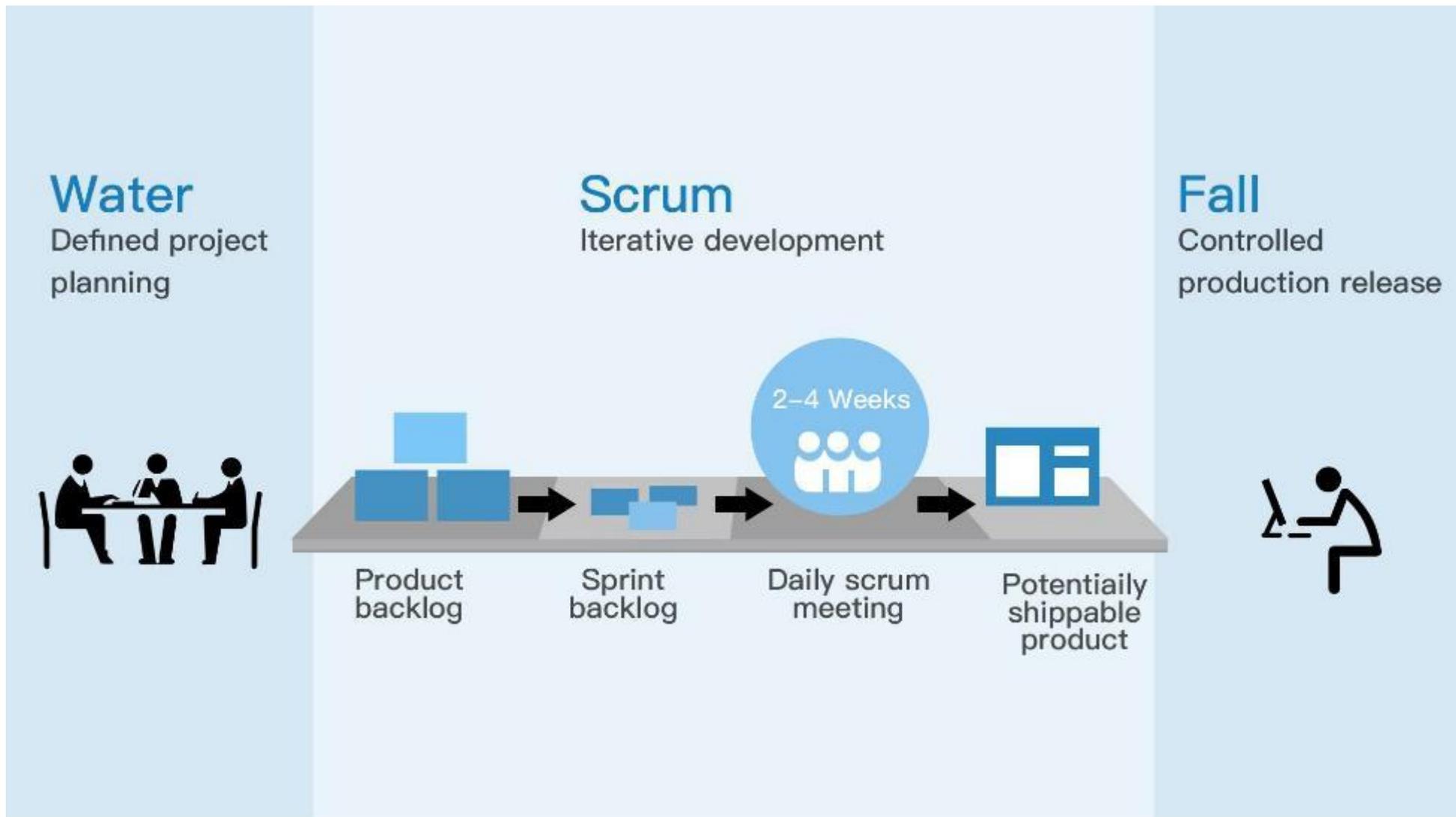
2. 在哪些方面对您有明显帮助或改进？



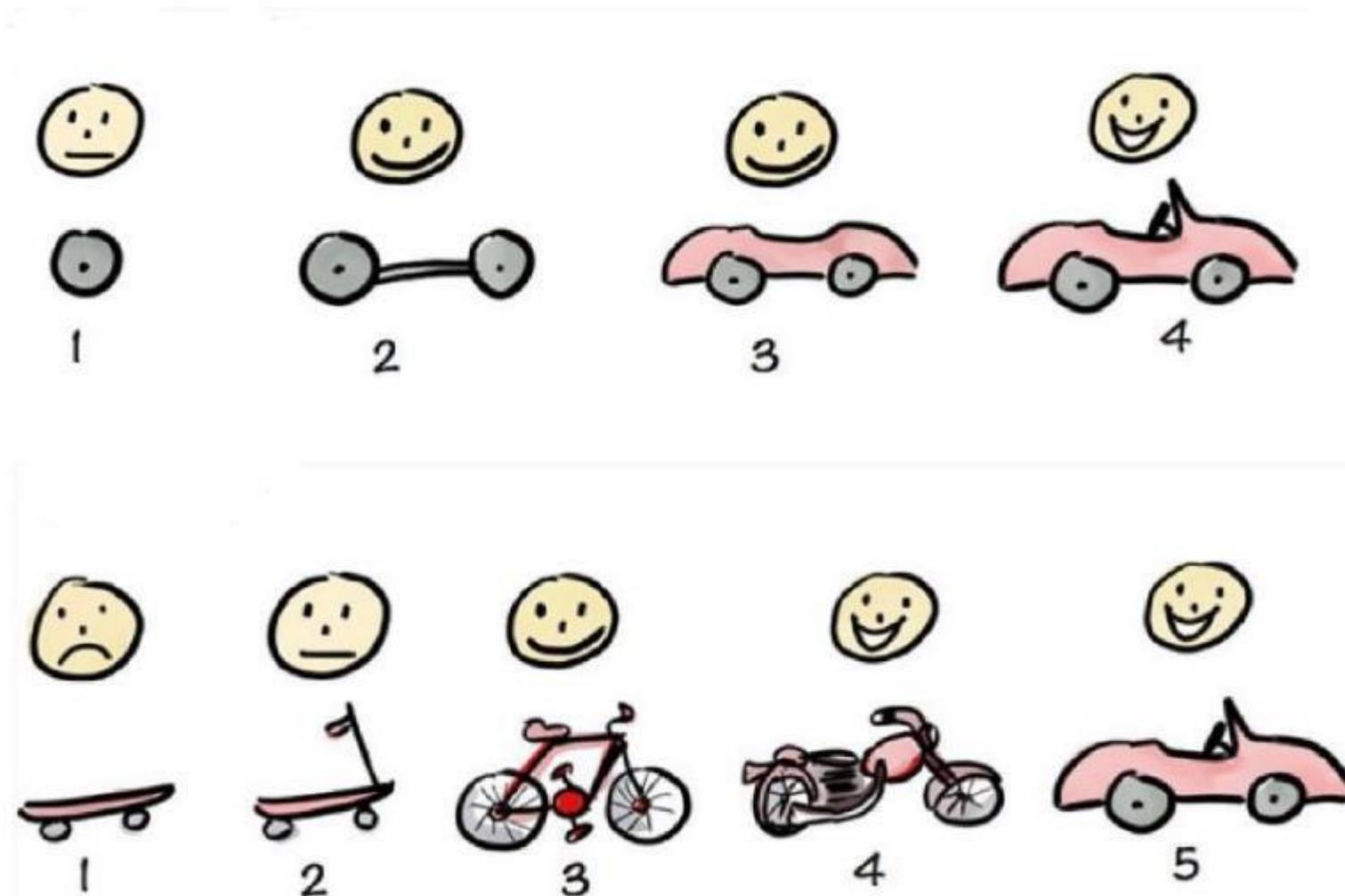
项目实施的策略



瀑布式敏捷Water-Scrum-Fall



对于敏捷的理解



转型和实践的焦点应该回归到人的身上



利用转型和实践打造一个更有战斗力的团队

- 一群行动目标一致的人
- 一群技能可以不全栈，但利益和责任全栈的人
- 一群积极沟通的人
- 一群不断精进的人

用更看重团队成长的视角，来看待转型和实践过程遇到的难题，
很多时候也许就不是问题，而是成长过程中必不可少的学费。

细心、耐心，没有最好的，只有当前最合适的。



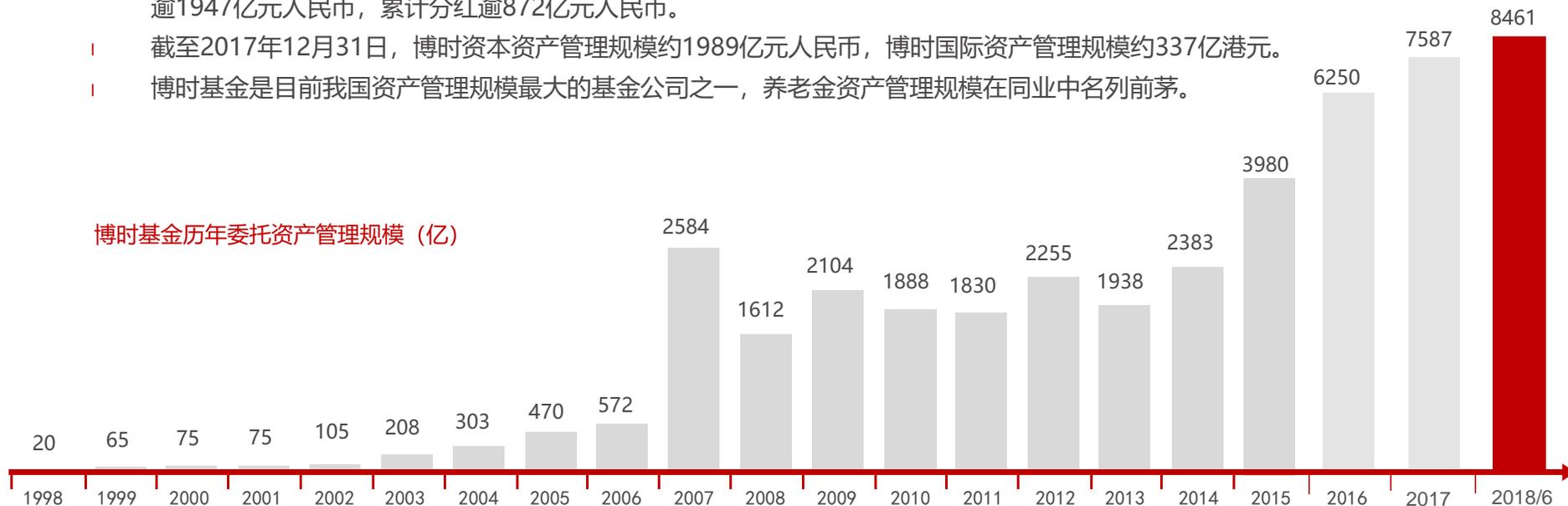
博时基金：中国基金行业的开拓者

- 博时基金成立于1998年7月13日，是首批成立的五家基金公司之一，致力于为海内外各类机构和个人提供专业、全面的资产管理服务。经过20年努力奋斗，已发展成全牌照经营的大型基金管理公司
- 公司目前注册资本2.5亿元人民币，总部位于深圳，在北京、上海等地设有分公司，在香港和前海分别设有子公司



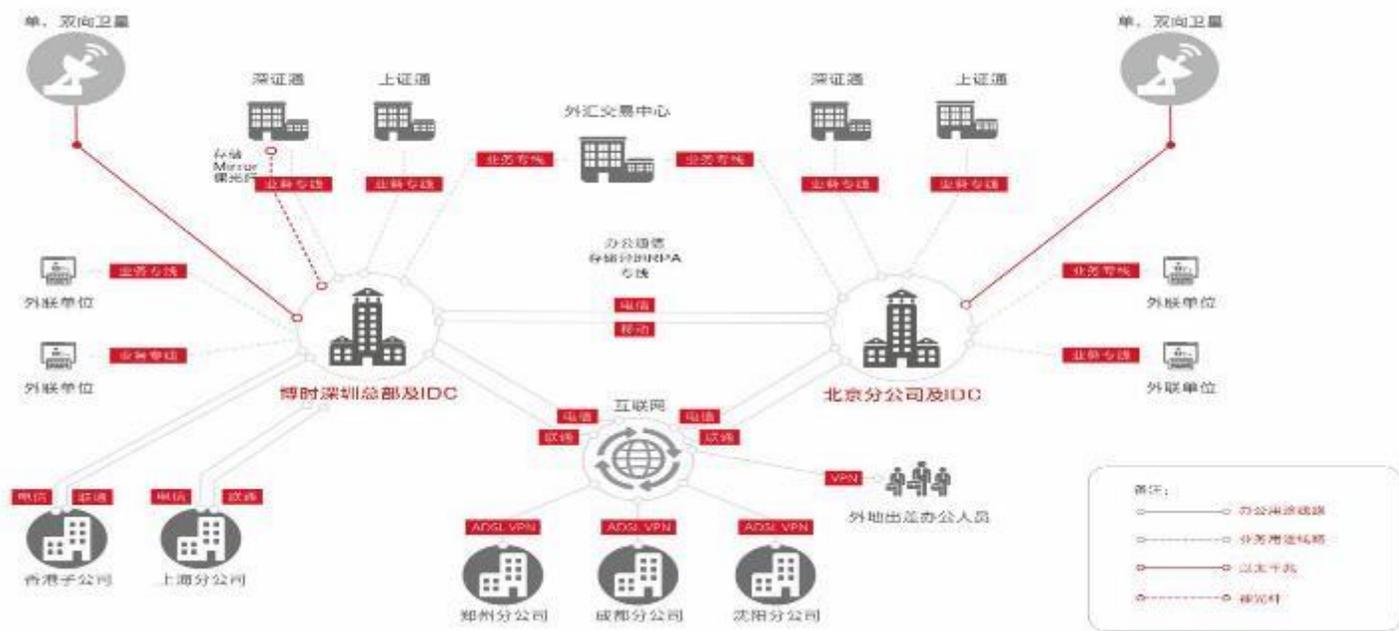
管理规模：各类委托资产规模位列行业前列

- 截至2018年6月30日，博时基金资产管理规模逾8461亿元人民币(不含子公司博时资本、博时国际)，其中非货币公募基金规模逾1947亿元人民币，累计分红逾872亿元人民币。
- 截至2017年12月31日，博时资本资产管理规模约1989亿元人民币，博时国际资产管理规模约337亿港元。
- 博时基金是目前我国资产管理规模最大的基金公司之一，养老金资产管理规模在同业中名列前茅。



IT建设：业务IT化与IT业务化

- 博时拥有强大的IT研发队伍，被誉为国内公募基金IT人才的“黄埔军校”
- 多个系统在行业内率先投入使用：知识管理系统、投资决策支持系统、CRM系统、管理会计、独立账户管理系统、统一沟通平台、全球投资数据平台
- 获多项大奖：2008年荣获证券期货行业科学技术奖的“最佳创新奖”、2006年、2012年、2013年多次荣获深圳市金融创新奖、2018年，DevOps统一研发平台获第六届证券业协会科技进步奖二等奖



THANKS

