



不要因为要改变，就放弃

强监管环境下，传统金融企业如何挑战互联网

2018 中国·北京

讲师简介

Speaker Introduction



徐磊

英捷创软 CEO
首席架构师
微软最有价值专家 MVP
Microsoft Regional
Director
Certified ScrumMaster



LEANSOFT首席架构师

徐磊

英捷创软 CEO 首席架构师 | 资深ALM顾问和解决方案专家 | 微软最有价值专家 | 大中华区域社区技术总监 | ScrumMaster | 敏捷教练 | 超过10年的软件研发项目管理经验 | 曾任SSW中国研发中心总经理

项目经验

- 高级ALM/DevOps顾问, 中国农业银行, 互联网金融项目
- 敏捷教练, 兴业银行, 海外网银敏捷试点项目
- 高级管理顾问, 中国移动南方基地, 互联网研发基地外包管理改进
- 高级ALM/DevOps顾问, 上海通用汽车有限公司, 软件研发过程改进
- 高级ALM/DevOps顾问, 上海汇众汽车, 软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问, 中国人民保险公司软件研发中心, 软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问, 斯伦贝谢中国研发中心, 敏捷开发与ALM落地
- 高级ALM/DevOps顾问, 京东商城, PMO管理系统与ALM系统集成 高级ALM/DevOps顾问,
- 华为2012实验室(深圳), rDelta系统选型
- 高级ALM/DevOps顾问, 百威英博, 软件外包管理系统ALM落地实施

我的DevOps之路

My Way to DevOps



TFS 2005 Beta

开始试用TFS 2005 Beta版本，在2年间将公司的50个项目 / 系统代码从VSS迁移到TFS。跨越中国北京和澳大利亚悉尼的多个研发团队



2008



DevOpsDay

多个软件过程改进项目

从2012年开始，包括京东，华为，招商国际，石化盈科，百威英博，中国人民保险(PICC)等多家超过千人的企业实施软件工程管理过程改进。



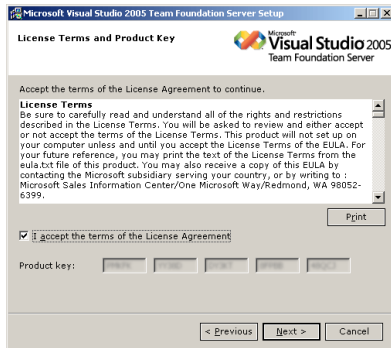
中国农业银行
AGRICULTURAL BANK OF CHINA

2016

2005

VSTS Real world

组织了来自20家大陆最好的软件开发商ISV，组成了虚拟团队，在1周内完成了TFS培训，并开发和发布了“汶川孤儿领养”系统。



2012



中国农业银行互联网金融

中国四大银行之一，传统金融业的代表性大型国有银行。第一次实现跨部门合作，采用类敏捷模式完成重点项目的研发和交付

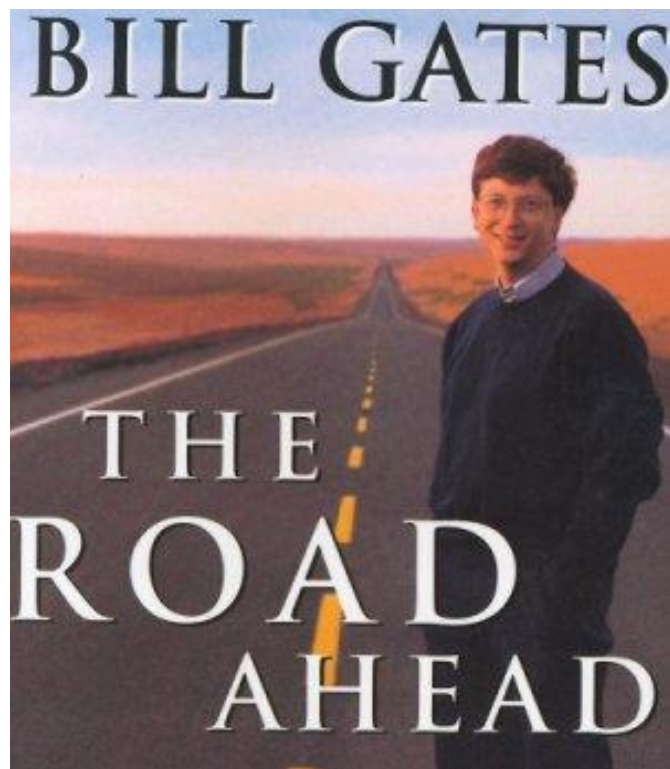


内容



- 要生存就要挑战
- 传统组织中如何激活创新
- 某大型传统金融企业的转型历程





Banking is necessary, Banks are not

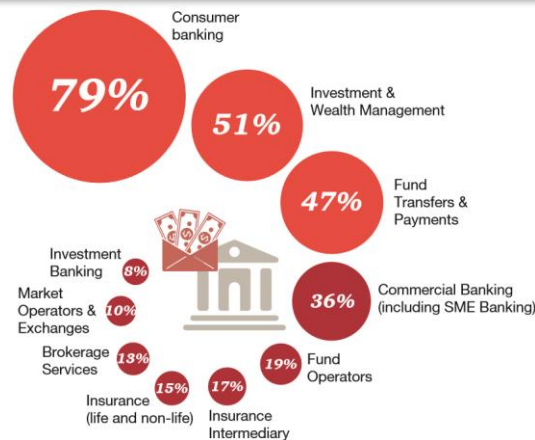
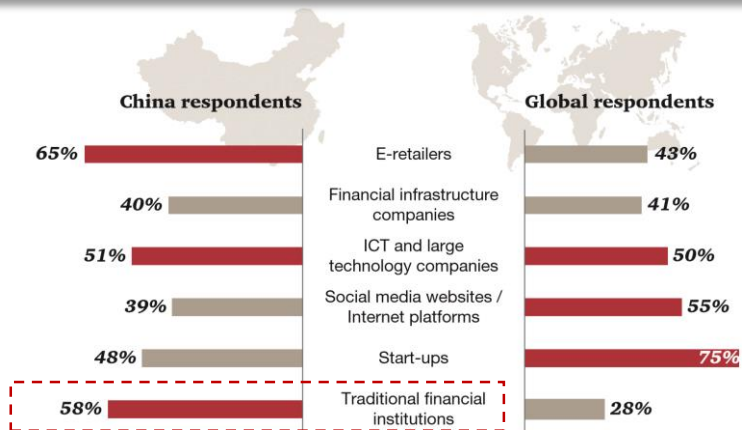
- 1995年《未来之路》比尔盖茨



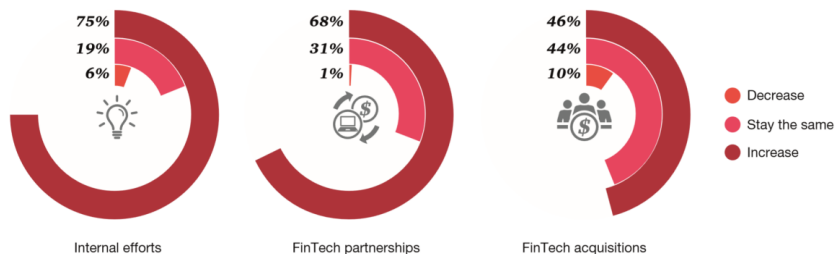
未来五年FinTech对金融行业的影响

除零售业外，传统金融行业最容易被互联网所颠覆

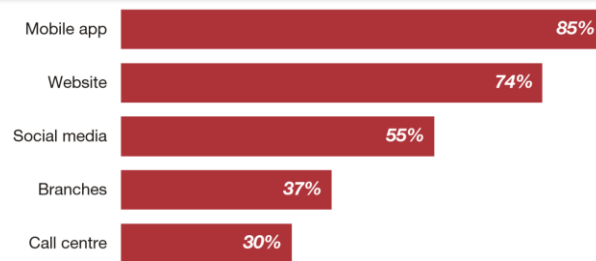
消费金融、财富管理、转账支付业务最可能被Fintech取代的业务



金融创新三种模式的预期：自主创新+合作伙伴+收购兼并



电子渠道是金融行业分布最广的互动渠道



Source: PwC Global Fintech Survey China Summary-Jun2017

2018 DevOpsDays | 北京站



BATJ 在金融行业的布局

	Baidu 百度	蚂蚁金服 + Alibaba Gro 阿里巴巴集团	Tencent 腾讯	JD 京东
借贷	宜人贷 百度小贷		元宝铺 Lufax.com waddle	京保贝
支付	爱贝云计算 百度钱包	支付宝	乐刷 微信支付	京东支付
财富管理	百度	余额宝 蚂蚁聚宝	理财魔方	京东小金库
消费金融	百度金融	趣分期 蚂蚁花呗 网金社	微粒贷	金融1号店
保险	Joint Venture 百安保险	众安保险 国泰产险	众安保险	京东保险
证券基金		德邦基金 数米基金网	HOWbuy 富途证券	京东小白理财
互联网银行	Joint Venture 百信银行	网商银行	WeBank 微众银行	
信用评级		芝麻信用	腾讯征信	Joint Venture zest finance
众筹	百度金融	蚂蚁达客 淘宝众筹 36 Kr	乐捐	京东众筹
区块链	百度外卖 糯米 链家网	ele.me	链家网 美团	3W

注：[虚线框] 为自主研发



传统银行的危机与挑战

生态圈模式 给银行带来的 四大挑战

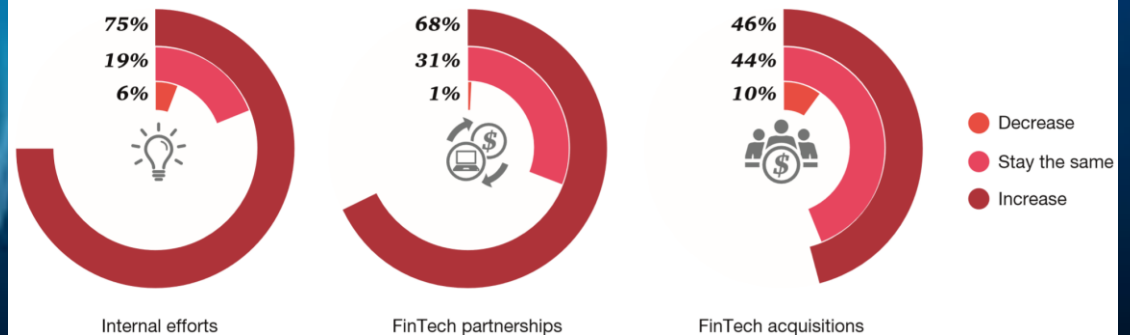
	 客户脱媒	 解绑	 商品化	 隐形化
描述	银行正在失去同顾客的接触机会，因为人们转向非银行渠道/资源	银行提供的众多产品服务正在被肢解，因为顾客可选择更优客户体验的单一服务提供商	银行的差异化越发艰难，因为消费者可以便捷地通过线上比较银行产品	银行正在失去品牌认知度，变得隐形，因为顾客可以在不知晓品牌的情况下使用金融服务
零售进攻企业示例	消费者可以在线上借款而无需寻求银行的服务（如，向Lending Club借贷俱乐部公司线上申请贷款）	存款和支付解除绑定，因为消费者可采用第三方支付平台直接支付（如支付宝）	消费者可以比较来自不同银行的不同存款产品并选择利率最高的产品（如SavingGlobal）	消费者可以通过社交网络获得无品牌的微贷服务（如微信微粒贷），而无需顾及贷款的资金提供方来自哪个银行

Source: 麦肯锡全球银行业报告 (2017)



如何挖掘传统金融的巨大潜力

金融创新三种模式的预期：自主创新+合作伙伴+收购兼并



南航叫停第三方平台值机背后：机票之后的机场场景之争



携程在手，说走就走！

放心的服务 放心的价格

指南针动力 ● 悦旅程



机票新政主推直销，老大南航也不提佣了，携程机票还玩得下去吗？

2016-06-04 20:09 虎嗅网

进入6月，暑期机票销售旺季即将到来，但在线机票预订巨头携程并没有迎来好消息。继航司取消机票基础代理费后，6月2日，南航宣布将自7月1日起执行新的销售政策，国内客票代理手续费将改为按航段定额支付。对于严重依赖机票佣金的携程来说，日子将变得更不好过，机票收入大幅缩水已是“板上钉钉”。可以预见，携程下半年度的财报，在机票交通部分，将一片惨淡。

根据Analysys易观研究显示，2017年第4季度，中国在线机票预订市场交易规模达到1383.7亿元人民币，同比上涨7.3%。其中，携程、去哪儿、飞猪三家OTA的市场份额之和达到74.2%。经过三年的提直降代，国内航司直销比例纷纷接近或者突破50%，但除去航司OTA旗舰店，航司真正的直销占比约10%-15%左右。



建立内部创新机制

-- 传统金融应对互联网的关键要素

- 开放金融是行业转型大势，不是跟风热炒
- 云化生态是业务战略目标，不是阶段任务
- 开放平台是全新业务模式，不只是技术项目

业务变革与转型目标

- 微服务是重新思考架构，不只是旧系统改造
- API是生态粘合的纽带窗口，不只是技术封装
- DevOps是组织和流程的重塑，不只是方法论

技术重塑与战略手段

平台适配与生态定位

- IaaS是旧系统的过渡，不是云的终极形态
- PaaS是业务的解耦重组服务化，不是技术分解
- SaaS由谁造？应是生态伙伴，不是杭研上研广研

体制改革与组织再造

- 思考如何与业务部门形成及时反馈的小团队作战模式，而不是接需求做项目
- 做互联网时代的产品经理，不是开发中心项目经理
- 新业务模式需要的是技术推广部生态伙伴部，不是开发一部测试二部

众多传统行业中谁最需要DevOps?

强烈依赖IT的非IT行业

金融企业的业务人员从来不觉得自己是干IT的
金融企业的IT人员从来不觉得自己是干金融的



其实，有很多次，心里的我都已经在飞身救球了.....当我回过神的时候，自己却呆呆的现在原地，傻傻的笑。



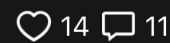
其实，有很多次，心里的我都已经在飞身救球了.....当我回过神的时候，自己却呆呆的现在原地，傻傻的笑。😏



赞



评论



14 11

不要因为要改变，就放弃

外忧内患中，中年油腻大叔如何挑战小鲜肉

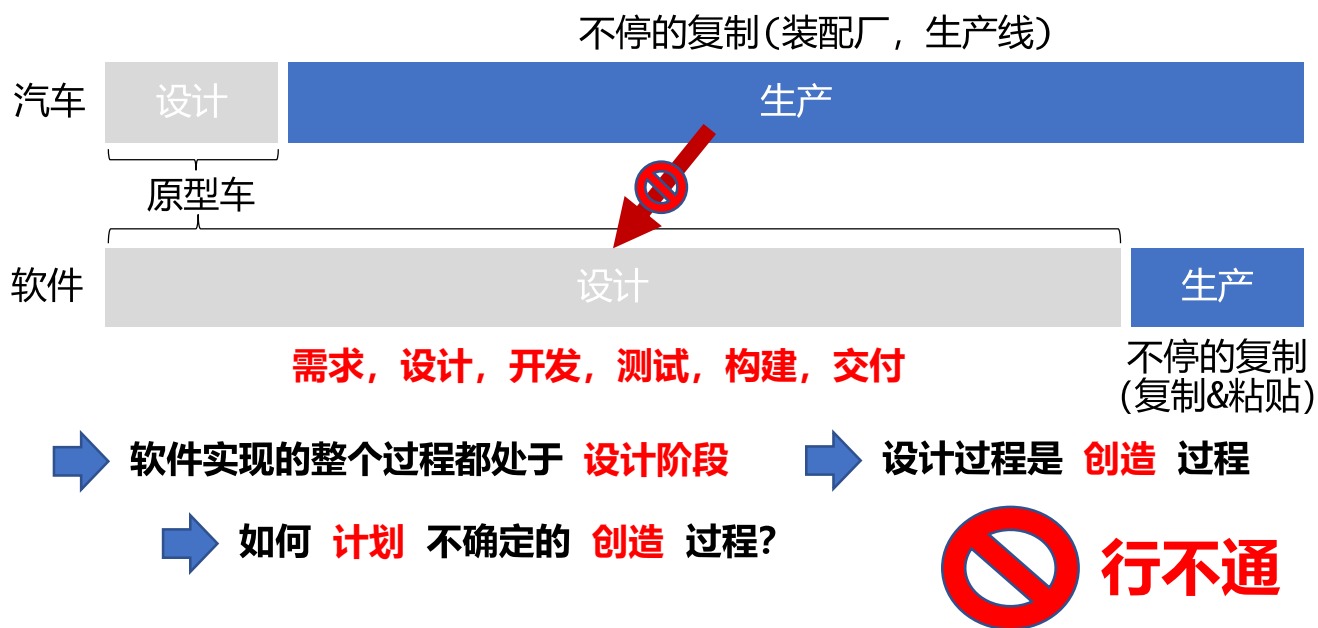
2018 中国·北京



敏捷提供创新的**大脑**，
DevOps提供创新的**肌肉**



接受不确定性



你觉得你看到的公路是这样的 ...



其实 ...



不确定性思维需要新的管理模式

传统管理思路是：自上而下

- 传统管理依赖管理者的主观意识定义管理流程。
- 预定义管理思路

敏捷管理思路是：自下而上

- 敏捷依赖知识工作者的探索和实践形成管理流程。
- 经验型管理思路

某大型传统金融企业的转型历程 #1

- 行级别项目
- 第一次实现端到端应用生命周期管理
- 第一次可以不局限于现有管控手段(满足监管要求)
- 第一次可以打破部门界限组织虚拟团队
- 第一次为跨部门团队提供独立办公场所
- 第一次为了工具变更管控方式
- 实现全部代码集中构建发布

- 没有提出DevOps, 没有提到敏捷; 但是我们尝试了Scrum和Kanban, 引入了持续集成 ...

- 一切都在需要的时候发生

僵化组织中如何激活创新

- 抛弃大计划，大工程
- 现在就开始
- 创新只能被允许，创新不能被计划
 - 接受不确定性让组织细胞变得有意识
 - 反馈机制促进有机体细胞的自我演化
 - 被验证的实践开始在组织内蔓延





世界不只有眼前的苟且，
还有诗与远方

问：现有的流程已经满足业务了？还需要导入DevOps吗？

需要：满足业务是否等同于满足市场？

不需要：没有对手了？60分及格了

问：怎样的团队适合做DevOps？

答：没有不适合做DevOps的团队，只有不愿意离开舒适区的人！



问：为什么团队采用了一些敏捷实践？
生产力反而降低了？

答：评价体系必须跟随实践一同演进，
评价体系存在的价值本来就是促进演进，
而不是保持现状！

问：Scrum一点都不敏捷？ DevOps就是自动化？

答：学会了武功套路就能打败敌人吗？
没有每天挑水砍柴，你哪里来的肌肉记忆？



某大型传统金融企业的转型历程 #2



某大型传统金融企业的数字化转型

#2.1

时间	迭代	活动	实践	收益
2017-10	0	选定试点项目组, 规模30人	建立kanban, 引入Scrum, 借助外部CSM	可视化
2017-12	5	经历2个月, 大致6个迭代, 团队开始形成自己的Kanban拉动式流程	故事拆分, 每日站立会议, 回顾会议, 可视化价值流动	建立流动性 发现瓶颈
2017-12		(其他团队)开始评估使用Git替代现有配置管理工具和发版流程	特性分支, Pull Request, 代码检视	避免工具定制 合规性检查
2018-01	8	引入用户故事地图实践, 调研从tfvc到git配置库迁移的可能性		
2018-02	10	第一轮投产完成, 考虑整合测试团队		
2018-02	10	完成配置库迁移和git库拆分, 开始调研引入CI/CD流水线	代码检视	工程解耦 编码质量 合规性检查
2018-03	12	测试团队开始驻场, 从新评估现有kanban流程和角色定位	测试提前, ScrumBut	消除测试瓶颈
2018-03	12	Git+PR+质量门+CI/CD持续交付流水线验证通过	代码检视, 自动化部署, 版本对齐(故事对齐)	质量左移 按故事交付 生产Ready



某大型传统金融企业的数字化转型

#2.2

时间	迭代	活动	实践	收益
2018-03		(其他团队)开始尝试影响地图和用户故事地图引入产品设计	影响地图, 用户故事地图	改进业务与研发的沟通方式
2018-04	15	启用TFS管理Bug, 尝试使用工作项实现全流程跟踪	电子化研发全生命周期管理	数据分析
2018-04	15	使用TFS工作项与Git PR关联, Kanban卡片开始标注工作项ID/PR ID	衔接物理Kanban和编码工作流	持续交付体系形成



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。



粒度

耦合

管理粒度

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

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”。


DevOps三步工作法

建立全局观

建立反馈

持续改进

- 
- 软件全生命周期管理
 - 识别管理单元
 - 缩减管理粒度
 - 建立流动性

- 配置管理合理化
 - 持续解耦
 - 工艺改进团队建设
 - 内部知识库建设
- 

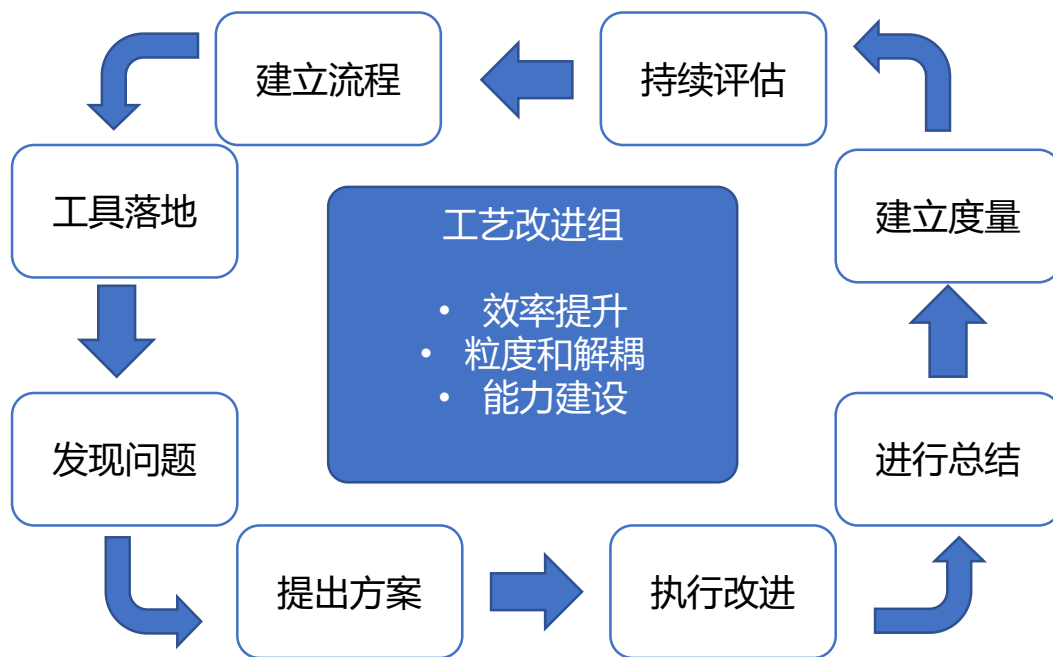
管理单元粒度逐步缩减，工程对象耦合逐渐解除



持续改进的关键：PDCA

Act 被评估证明效率有所提升的，形成流程并在工具中固化

Check 任何的改进都必须能够被评估，评估可以是主观或者客观

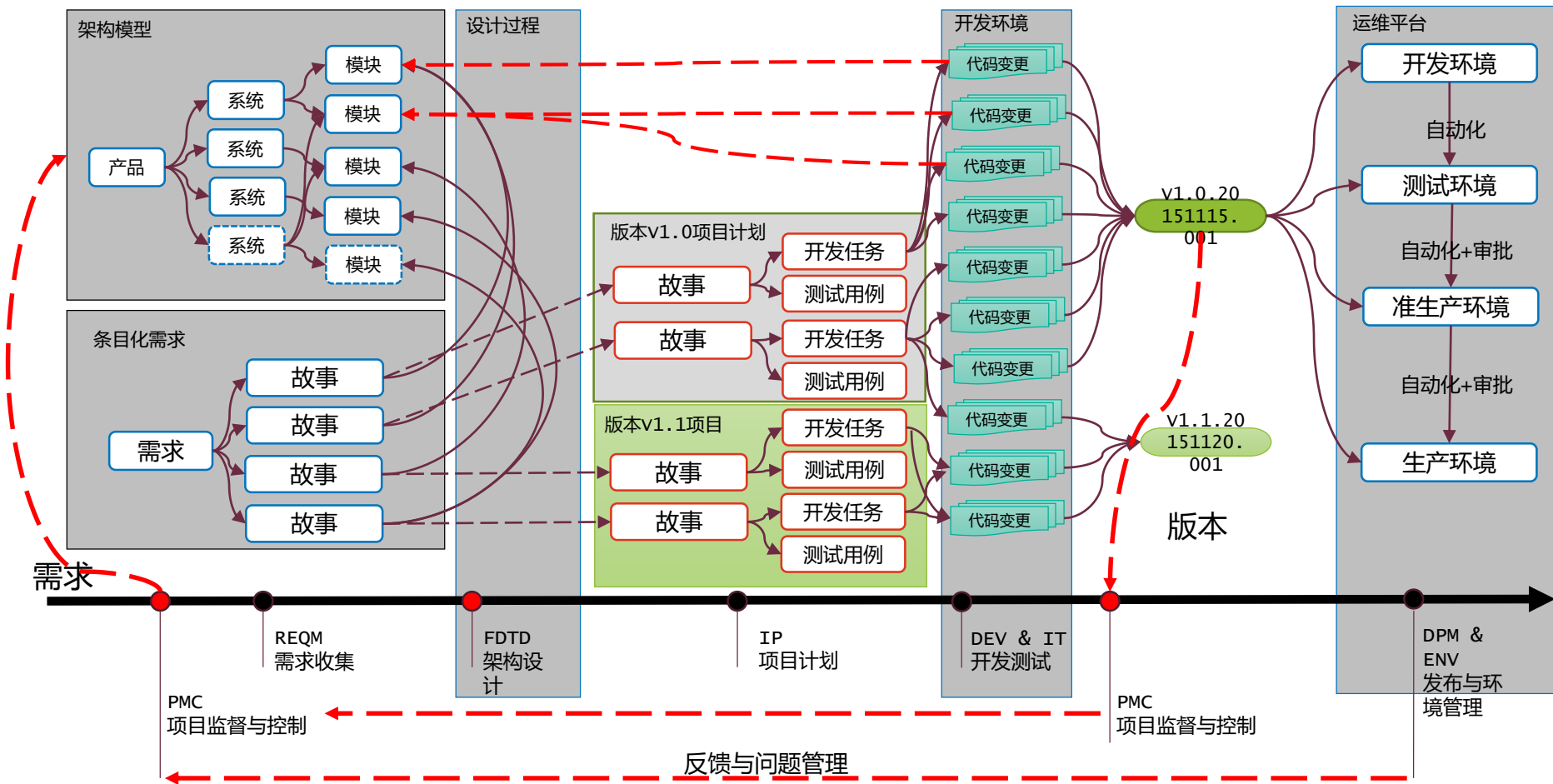


Plan 根据评估结果发现问题，并提出解决方案；给出流程规划和操作指引，与项目组确认，形成最终方案。

Do 实施改进方案，并随时跟踪项目组反馈，对方案有效性进行持续评估。



持续交付体系全景 - 管理对象 (个山又)

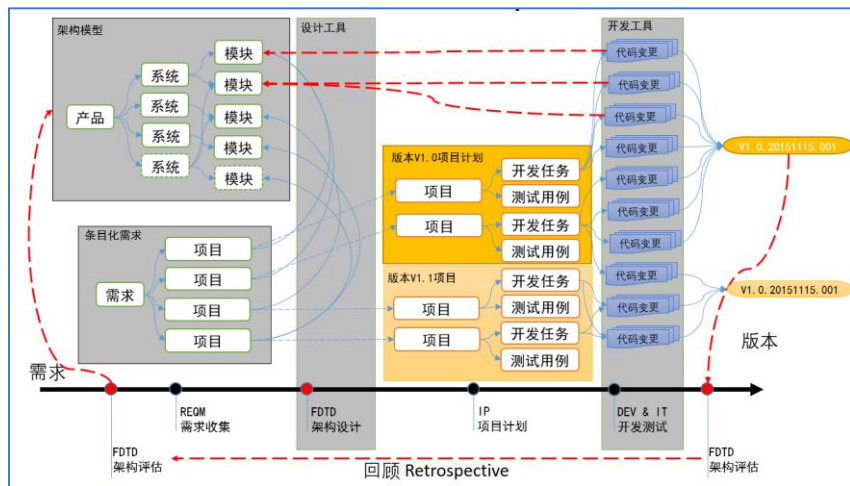


建立持续交付体系

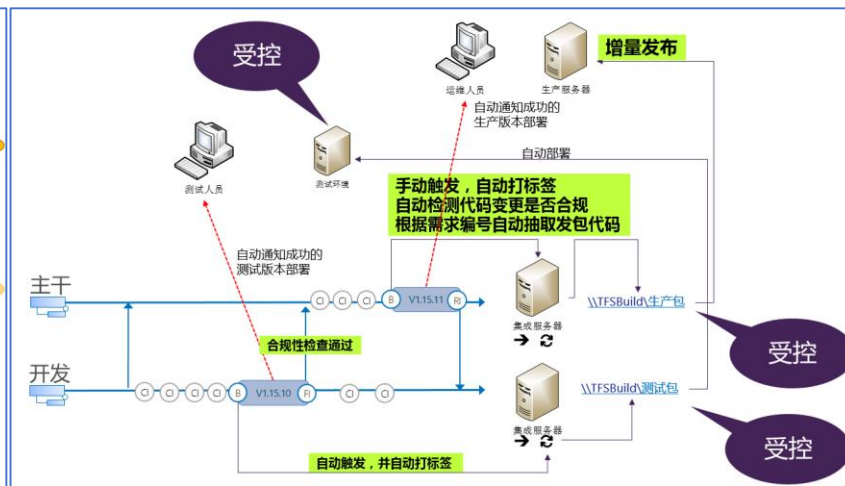
- 通过对上图中的点和线建立对应的**管理单元**的过程；形成**管理体系**。
- 这是软件研发的“行走的骨骼”，任何的实践都是这个骨骼上的肌肉



如何衔接管理与工程：配置管理合理化



管理属性 - 规划版本



工程属性 - 交付版本

规划版本 (希望做什么?)

TFS使用工作项提供端到端的需求版本管理能力, 每个工作项上都可以设置“迭代路径”字段代表需求所属的“规划版本”; 而与这一需求相关的任务/测试用例/缺陷/问题等也都一同归属这一“规划版本”。这样我们就可以用版本号来对开发过程中的所有工件进行查询, 分析和报表展现, 提供了“规划版本”的管控能力。

统一的版本号

根据开发团队及软件产品的本身的复杂程度, 以上“规划版本”和“交付版本”可以使用多级版本号进行标识, 如: 左图中的规划版本 v1.0 提供了2级规划版本 (大版本. 小版本)。而“交付版本”则在“规划版本”基础上继续添加2级, 形成【大版本.小版本.编译日期.序列号】的 4 级版本结构。

交付版本 (实际做了什么?)

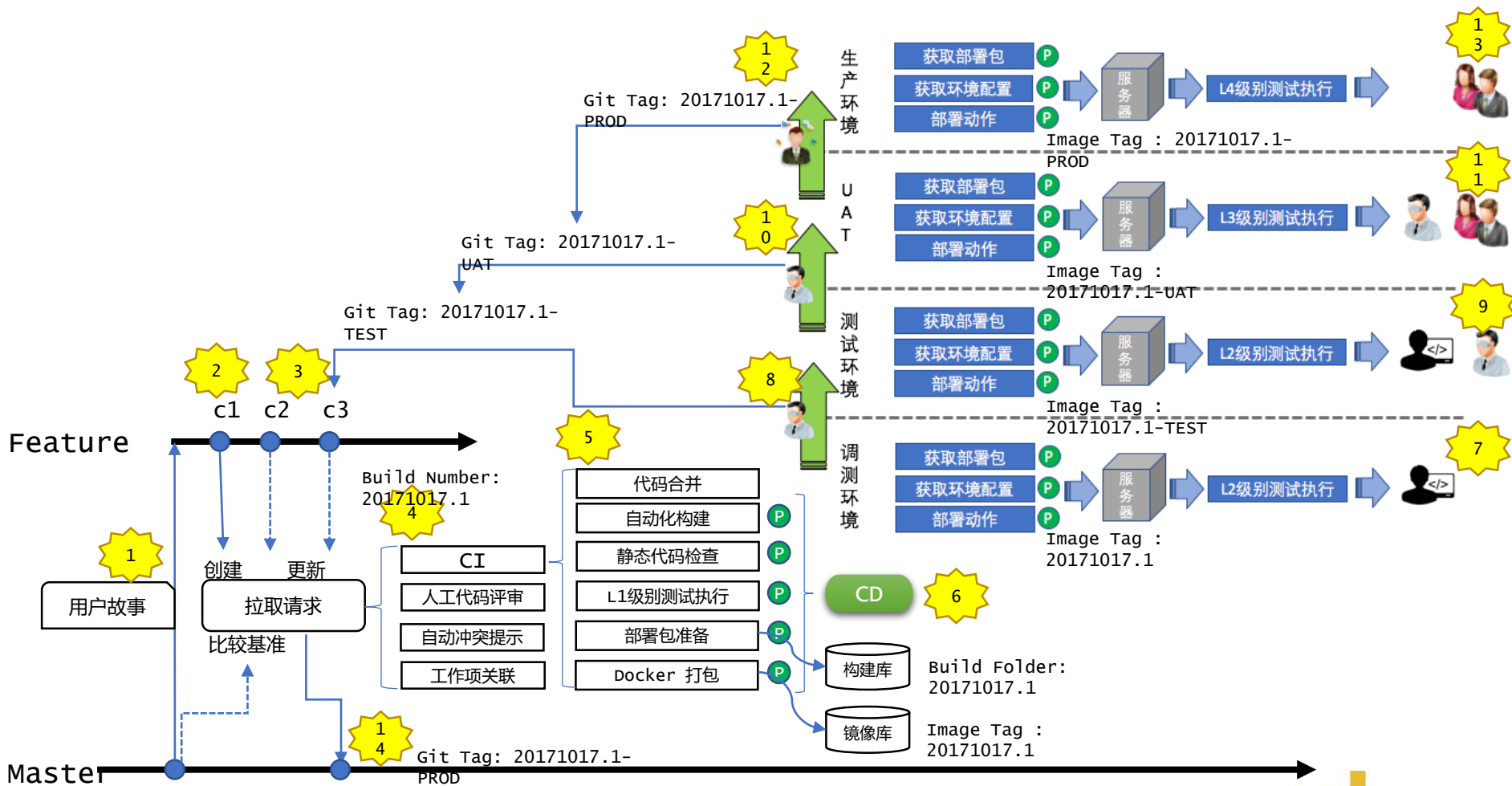
某一规划版本中的任务等工件驱动开发人员完成编码后, 开发人员可以将代码变更与“规划版本”进行关联; 同时TFS构建服务会自动生成“交付版本”号, 并将其所包含的代码变更与之关联, 因而形成了从“规划版本”到“交付版本”的跟踪能力。同时, 使用TFS构建服务还可以保证交付版本完全受控, 确保开发/测试/交付版本的一致性。

配置管理合理化

- 管理属性和工程属性的衔接点，就是**版本管理**
- 没有最好的配置管理流程，只有最合适
- 基本功没练好，懂再多的套路，实战中也发挥不出来



用户故事 + 特性分支 + PR + 质量门



问：物理板还是电子板？

答：转行初期物理板是最优选择，转行后期对板的选择本身也是团队自我选择的结果。

问：好像就是把版本管起来了？就是在做DevOps了吗？

答：不要管敏捷还是DevOps，软件开发最核心的“业务”仍然是写代码，让开发把代码写爽了，把版本发顺了，一切的转型都能发生！

程序员身上投入或节省，都会在生产环境放大n倍

- 只有接受挑战才能生存，最好的防守就是进攻
- 接受不确定性才是创新的本质
- 建立持续交付体系的核心是识别管理单元并缩减粒度
- 软件开发的本源仍然是写代码

所谓DevOps，就是让企业的大脑可以敏捷的指挥每一块肌肉，在需要的时候作出快速的反应，应对任何挑战！





THANKS