

舰船在行动

——大型企业开发部门敏捷转型实践分享

中国银行软件中心 阿奎

下一代
软件研发
SOFTWARE
DEVELOPMENT

自我介绍


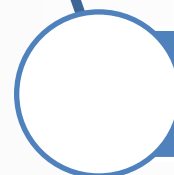
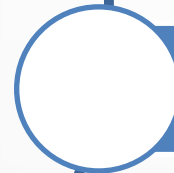

- 现岗位为中国银行软件中心系统分析师，应用基础团队负责人，内部敏捷教练。
- 在金融领域从事软件开发工作13年，拥有8年团队管理经验，带过70+人以上自有软件开发人员的团队，熟悉金融核心系统架构、业务分析。
- 最近4年来从事互联网金融软件开发管理和技术研究，同时，致力于组织级敏捷开发方法的推广和实施，对大型组织的敏捷转型有实际的经验和体会。
- CCEP、CSM、系统分析师



阿奎

自我定位为一名技术型管理者，在进行管理工作的同时，多年来一直保持对技术工作的热爱。自接触敏捷后，对于大型机构的敏捷转型和敏捷转型下的管理者应如何自处，多有实践和思考。

目录

-  我们的敏捷
-  我们的组织
-  我们的实践
-  我们的思考

中国银行软件中心敏捷概述

2016年，专题组继续打磨技术工艺，提升管理水平。

中心于2013年开始引入敏捷，经过三年多的建设，形成了一套纵横两条线的发展规划和试点转型三段论。

- 纵向研究：试点团队深入研究敏捷工艺，探索全流程敏捷的快速交付流程；
- 横向推广：借鉴敏捷的优秀管理和技术实践，加强交流，提高开发效率，降低管理风险。



针对软件中心的特点，敏捷试点工作分为如下三个阶段：



条线管理
常态化阶段

管理部门按照各条线管理要求，形成规范文件，稳定后纳入质量管理体系常态化运行。敏捷专题工作组将继续推动敏捷项目的过程改进工作。

条线管理机制
制建立阶段

管理部门派人进入试点项目，作为项目组的一员参与项目工作，通过实践提出初步的组织级条线管理要点。

试点项目
摸索阶段

管理部门和试点开发部门成立敏捷专题工作组，通过试点项目组的自主实践、探索和磨合来建立敏捷流程框架体系，关注新旧交界点的处理，积累方法和经验。

实施策略方面，基于“以点带面、风险可控”的原则，总结出一套敏捷转型实施策略，即试点产品先行先试，总结新工艺改进的技术和管理方法；试点部门基于试点产品的试点成果进一步巩固技术能力和管理方法；中心所有部门尝试敏捷先进实践，提高开发效率，为后续的转型奠定基础。



开发敏捷 vs 全流程敏捷



开发敏捷

产品在开发阶段进行敏捷迭代，具体为需求分析、总体设计、详细设计、程序实现（编码、单元测试、组装测试）**集成测试**阶段采用迭代方式，迭代结束后仍需按批次要求进行功能测试。



开发敏捷

自主开发，子产品
产品关系单一，松耦合
测试人员提前介入
敏捷相关知识和技能，符合人员配置要求
入门级持续集成和自动化测试基础



全流程敏捷

满足开发敏捷产品的准入条件
主线开发，需求串行
无独立功能测试阶段
所有测试均在迭代内完成，迭代完成后交付正式版本
较为成熟的持续集成和自动化测试水平

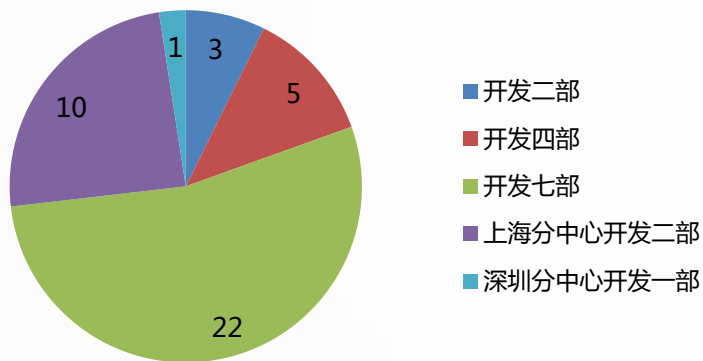
全流程敏捷

产品在开发敏捷的基础上，要求在迭代内完成所有测试、迭代结束后提交正式版本，**无独立的功能测试阶段。**

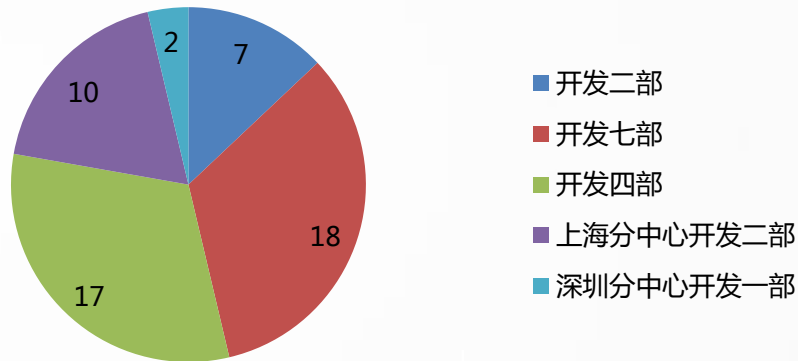


敏捷开发实施成效

敏捷项目按部门分布



敏捷产品按部门分布



瀑布类项目：446个。

敏捷类项目：**41个**，占比**8.4%**。

角色资质

- ◆ CSM：38人（覆盖五地、全领域）
- ◆ CSPO：6人（开发七部5人，上海开发二部1人）
- ◆ 内部教练：9人（开发七部7人，上海开发二部2人）

以子产品维度，共计55个，占比XX%。

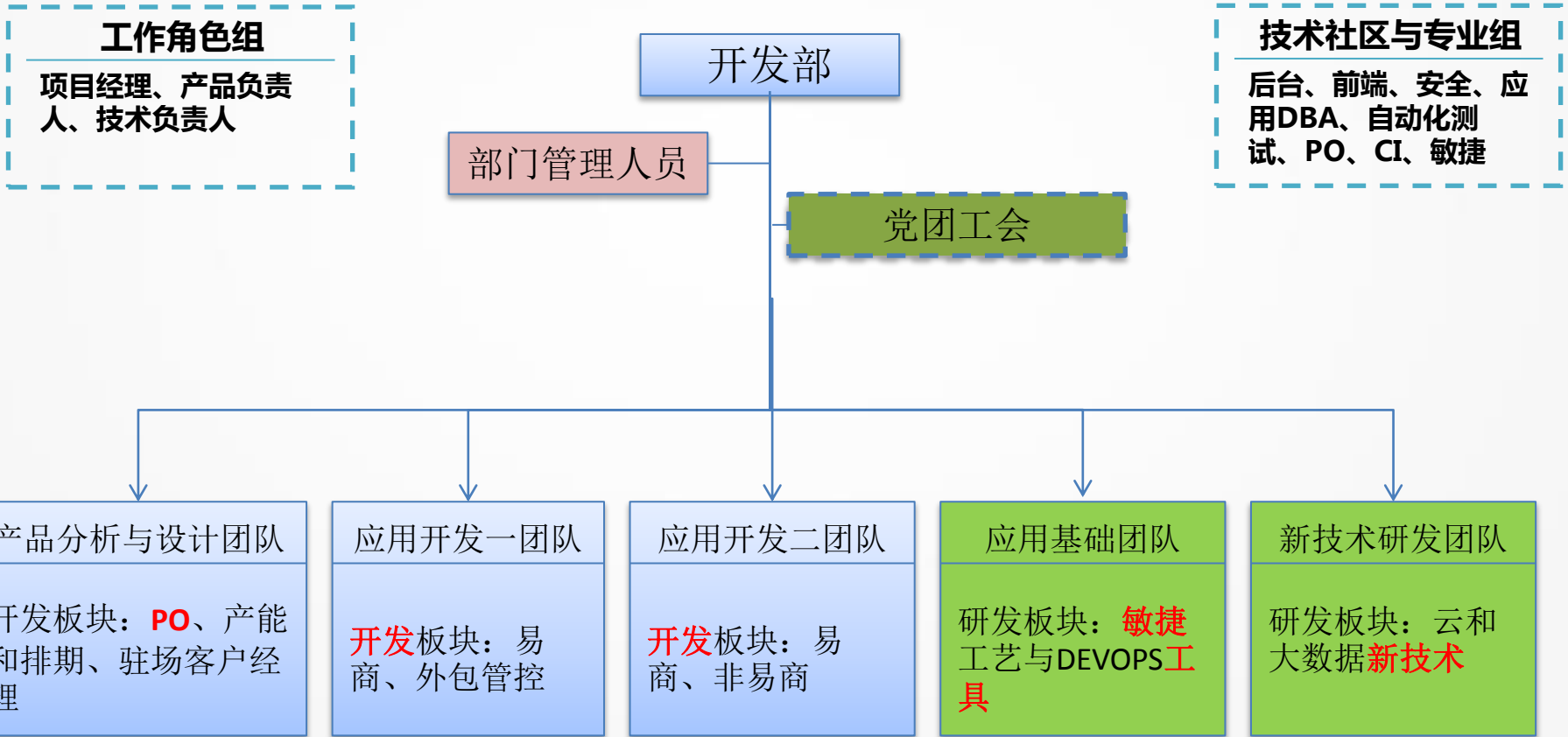
参与敏捷实施人员达500余人，占比XX%。

数据截止到2016年11月

目录

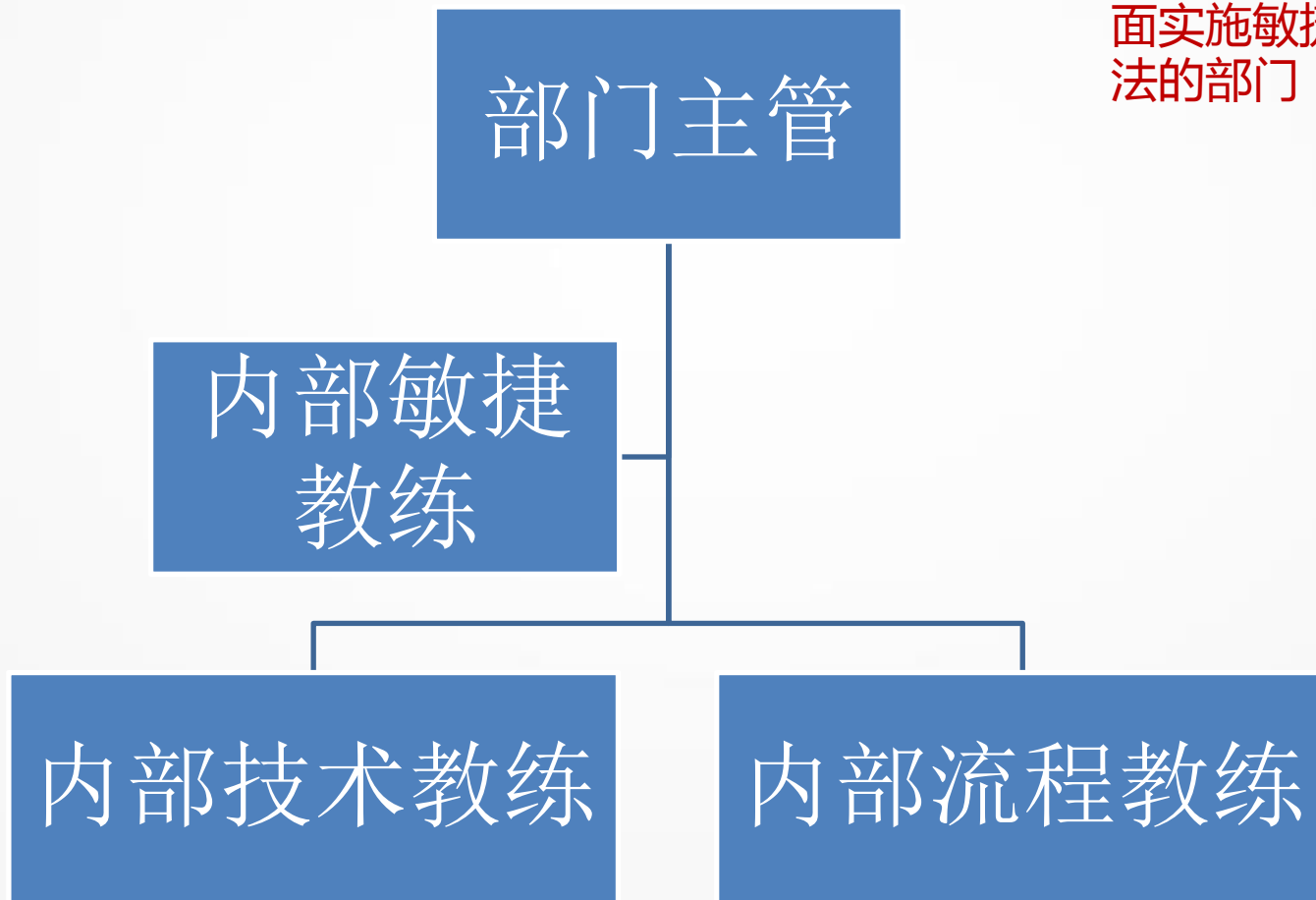
-  我们的敏捷
-  我们的组织
-  我们的实践
-  我们的思考

部门组织结构



部门内敏捷实施组织结构

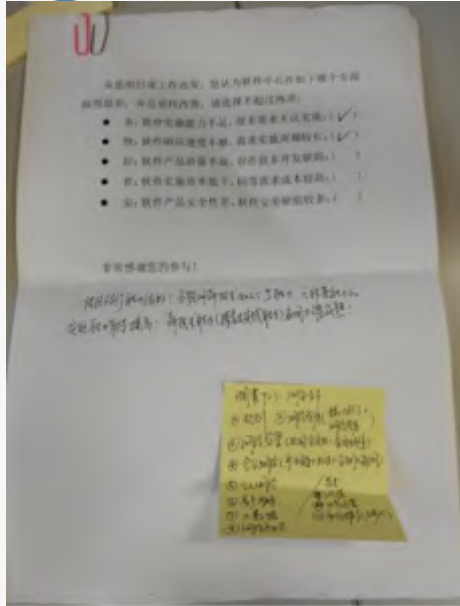
开发七部是全中心唯一全面实施敏捷Scrum开发方法的部门



目录

-  我们的敏捷
-  我们的组织
-  我们的实践
-  我们的思考

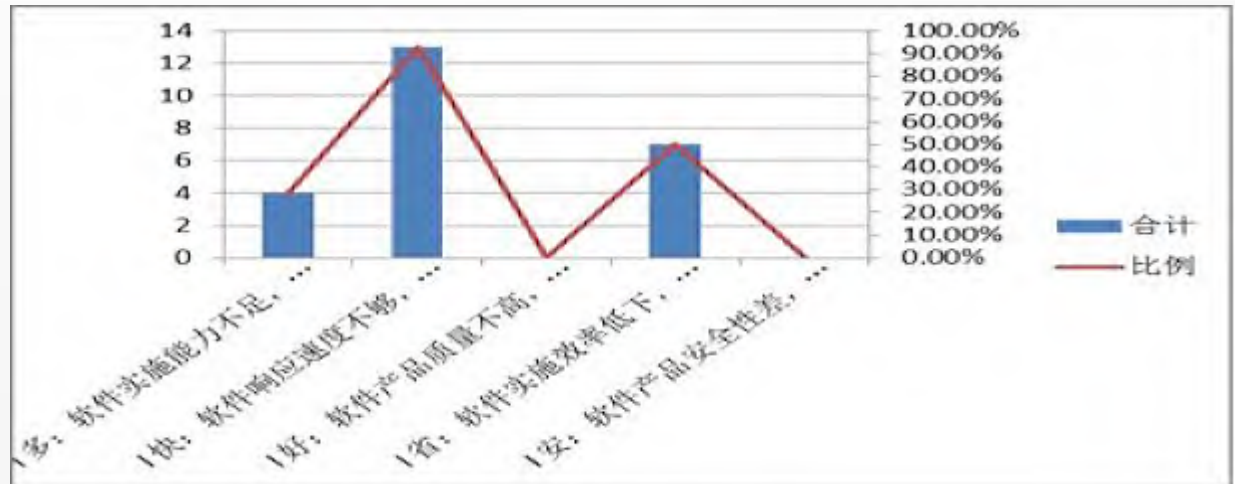
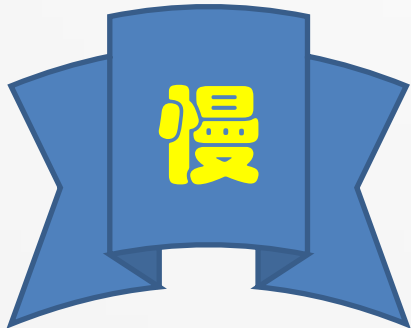
实践案例一：评估、排期、实施与投产



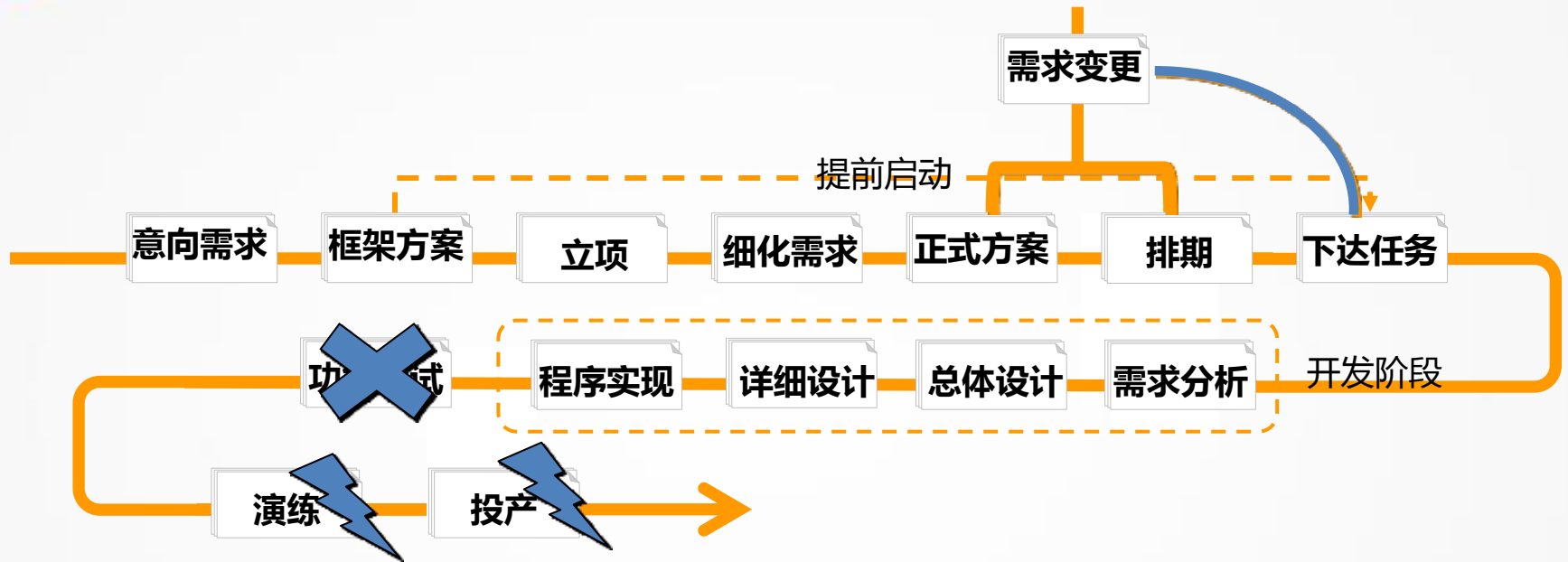
从您的日常工作出发，您认为软件中心在如下哪个方面做得最差，并且亟待改善，请选择不超过两项：

- 多：软件实施能力不足，很多需求无法实施； ()
- 快：软件响应速度不够，需求实施周期较长； ()
- 好：软件产品质量不高，存在很多开发缺陷； ()
- 省：软件实施效率低下，同等需求成本较高； ()
- 安：软件产品安全性差，软件安全缺陷较多； ()

非常感谢您的参与！



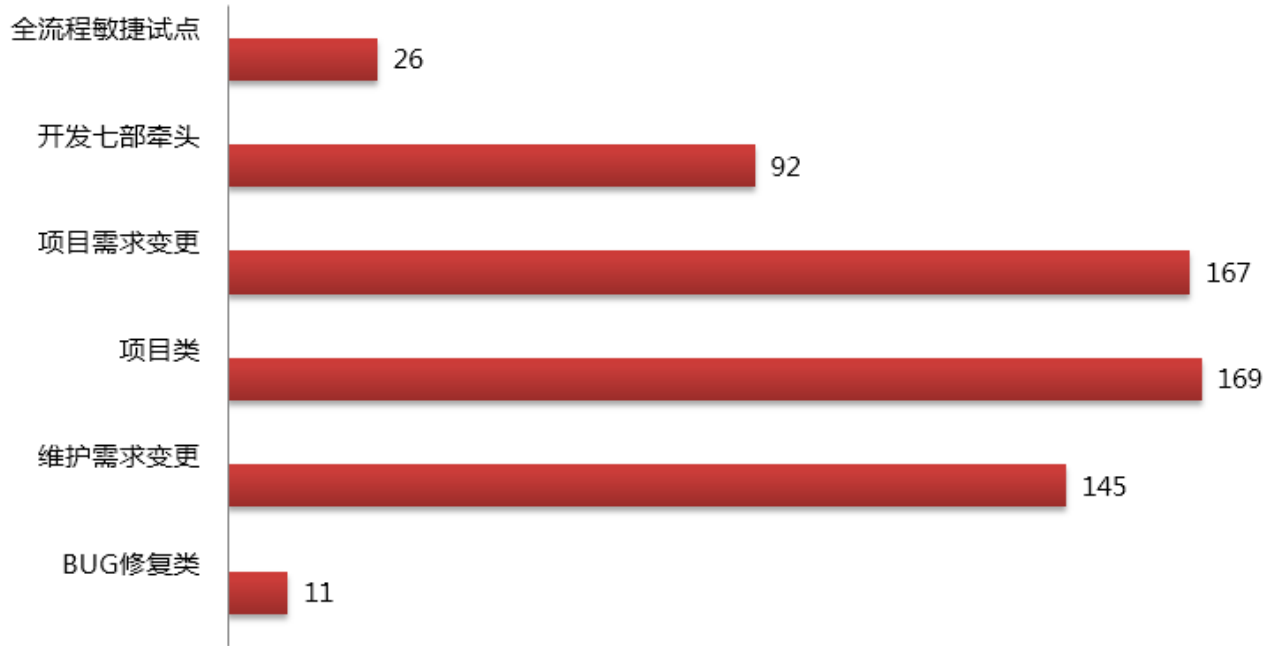
流程突破：排期投产、任务下达



- 无独立的功能测试阶段
- 按需申请生产任务
- 任务后排期（报备制）
- 投产轻量化管理

效果：天下武功为快不破

交付周期（自然日）
（生产任务下达——投产）



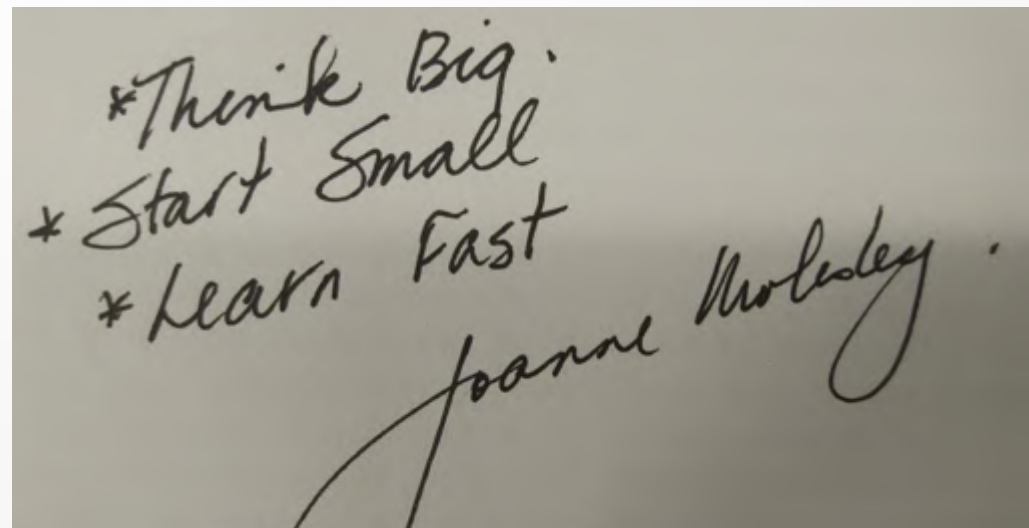
出国金融全流程试点实现了平均26天一次投产。

Poppendieck夫妇问道：“在你的公司里，仅涉及一行代码的改动需要花多长时间才能部署上线？你的处理方式是否可重复且可靠呢？”

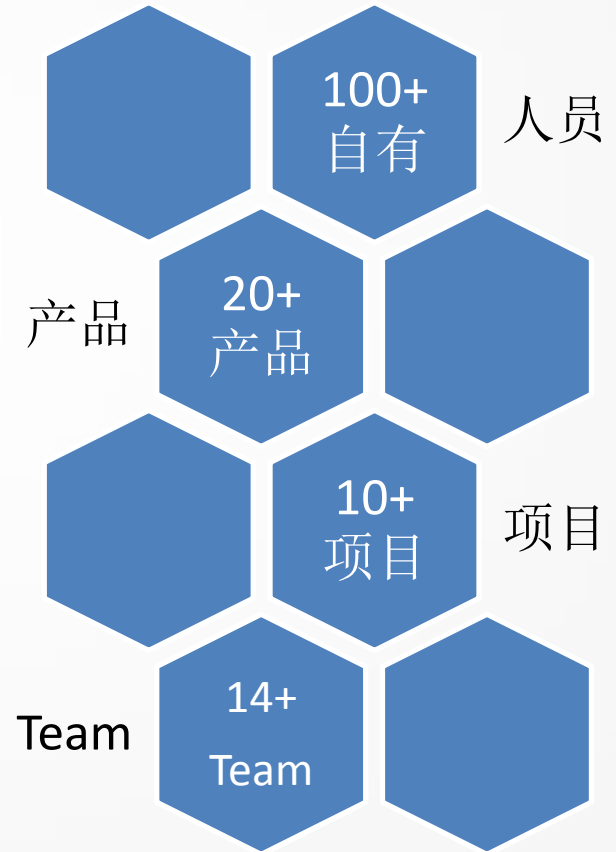
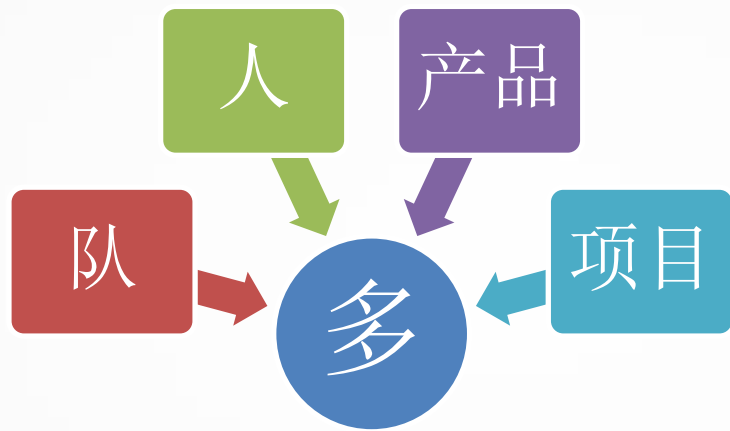
思考

敏捷不仅仅是快，但是，首先要快！

快是变小之后的快，不是加班后的快！

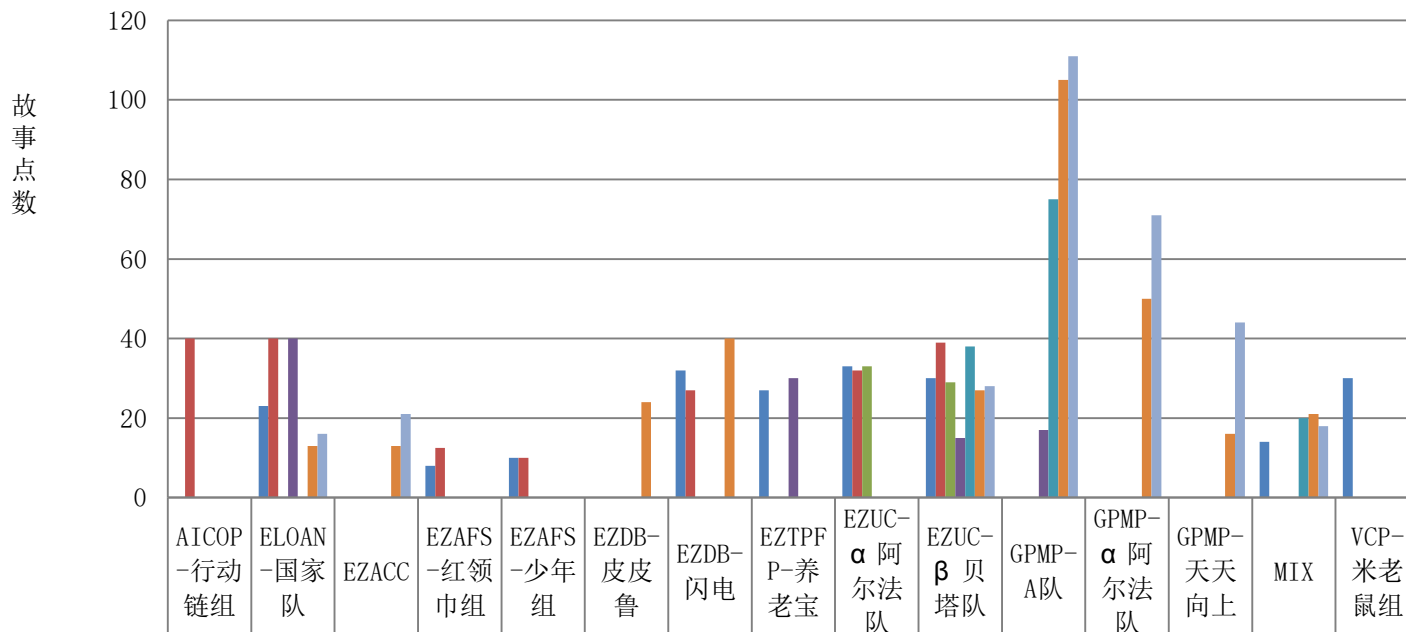


实践案例二：整体敏捷成熟度提升



度量——迭代速率

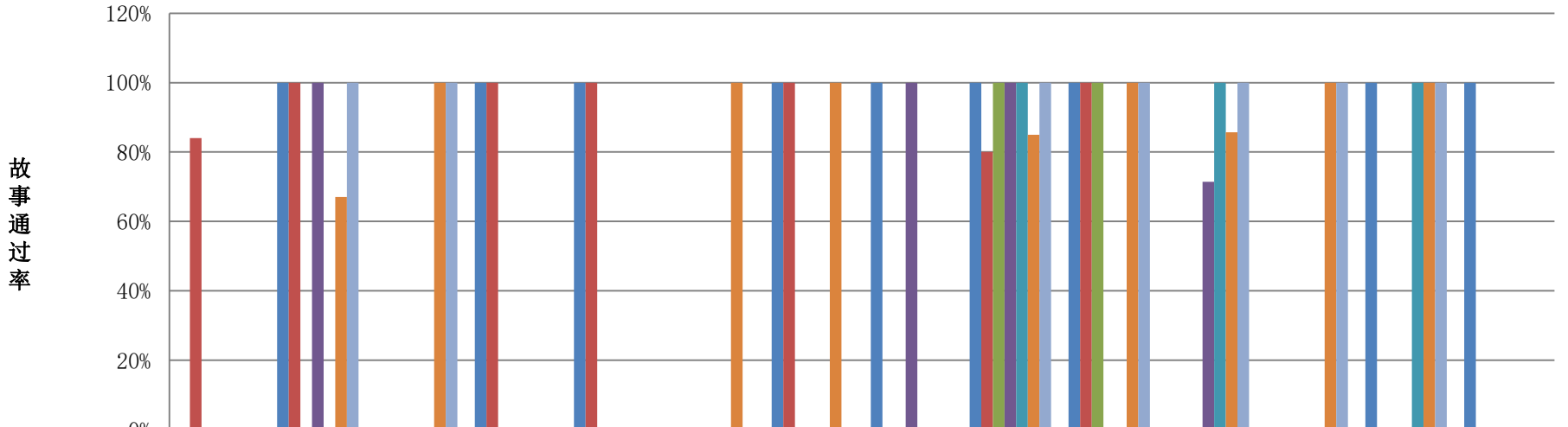
部门1季度迭代速率



	AICOP-行动链组	ELOAN-国家队	EZACC	EZAFS-红领巾组	EZAFS-少年组	EZDB-皮皮鲁	EZDB-闪电	EZTPF-P-养老宝	EZUC-α阿尔法队	EZUC-β贝塔队	GPMP-A队	GPMP-α阿尔法队	GPMP-天天向上	MIX	VCP-米老鼠组
■ 1月第一迭代		23		8	10		32	27	33	30				14	30
■ 1月第二迭代	40	40		12.5	10		27		32	39					
■ 1月第三迭代									33	29					
■ 2月第一迭代		40						30		15	17				
■ 2月第二迭代										38	75			20	
■ 3月第一迭代		13	13			24	40			27	105	50	16	21	
■ 3月第二迭代		16	21							28	111	71	44	18	

度量——故事通过率

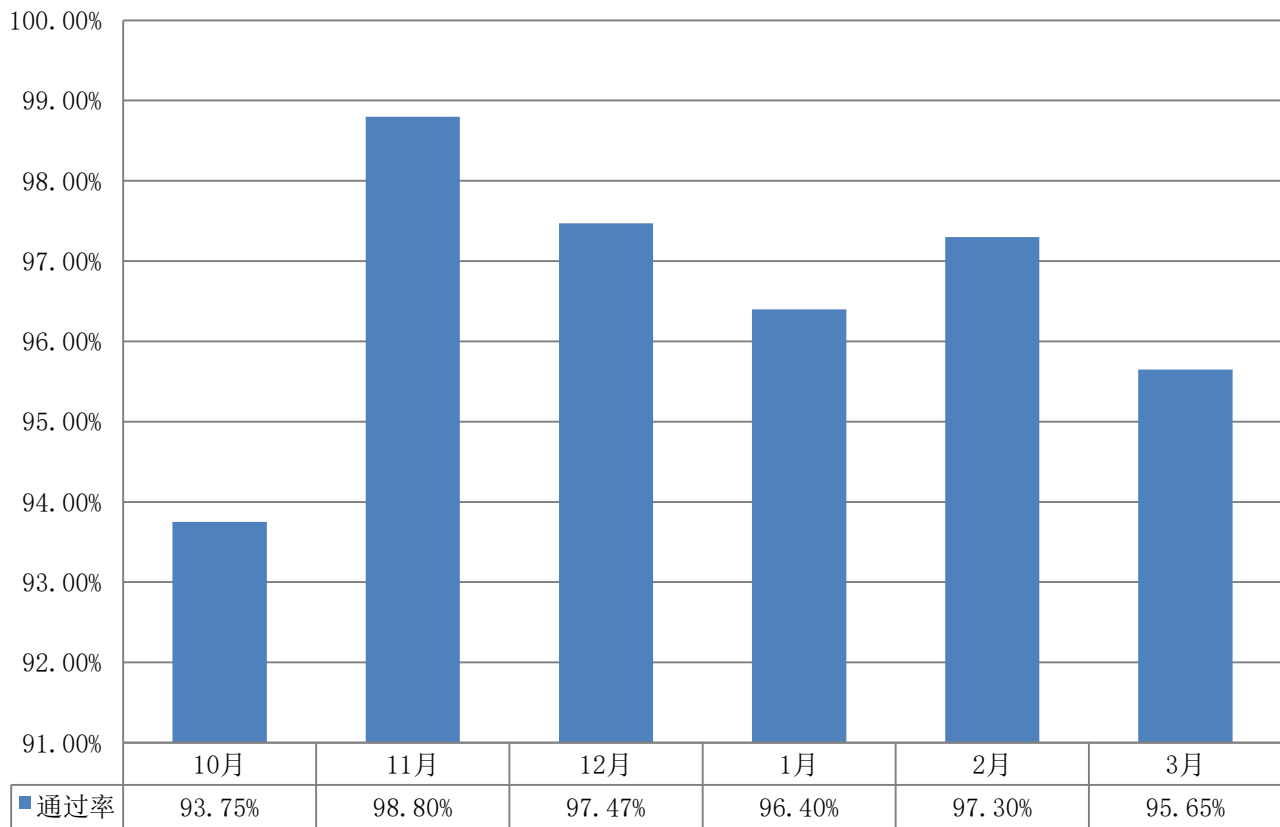
部门1季度迭代故事通过率



	AICOP-行动链组	ELOAN-国家队	EZACC	EZAFS-红领巾组	EZAFS-少年组	EZDB-皮皮鲁	EZDB-闪电	EZTPFP-养老宝	EZUC-α阿尔法队	EZUC-β贝塔队	GPMP-A队	GPMP-天天向上	MIX	VCP-米老鼠组
■ 1月第一迭代		100%		100%	100%		100%	100%	100%	100%			100%	100%
■ 1月第二迭代	84%	100%		100%	100%		100%		80%	100%				
■ 1月第三迭代									100%	100%				
■ 2月第一迭代		100%						100%	100%		71.40%			
■ 2月第二迭代									100%		100%		100%	
■ 3月第一迭代		67%	100%			100%	100%		85%	100%	85.71%	100%	100%	
■ 3月第二迭代		100%	100%						100%	100%	100%	100%	100%	

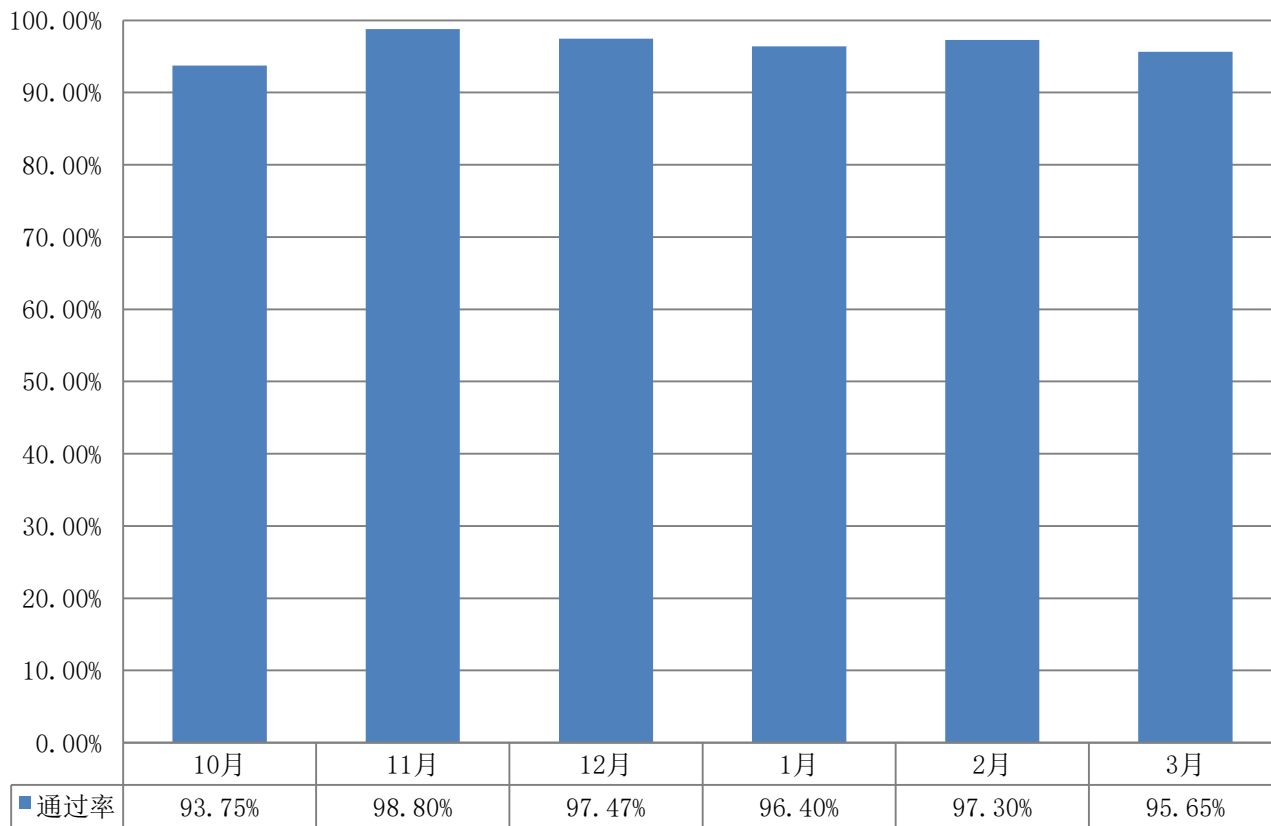
度量——故事通过率

部门2016-2017年平均故事通过率



度量——故事通过率

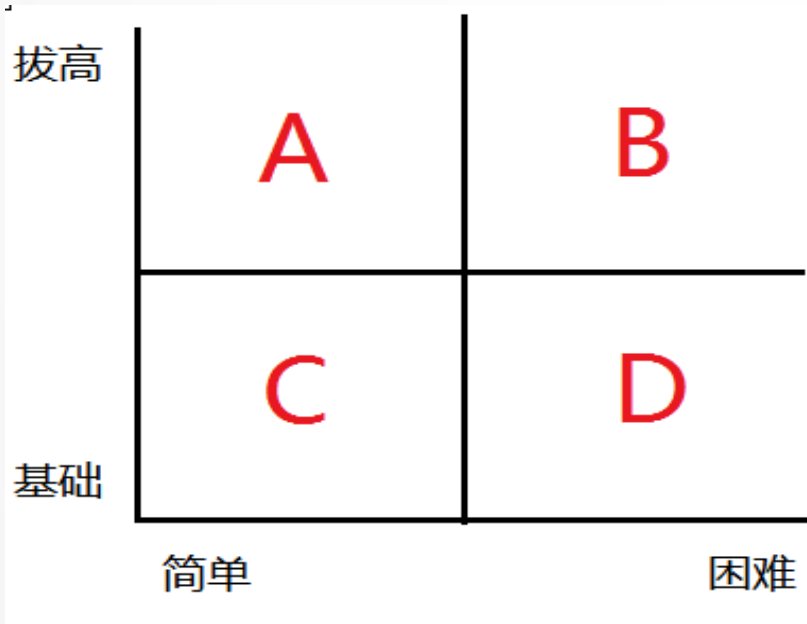
部门2016-2017年平均故事通过率



方案：找到可以引发改变的度量

敏捷成熟度调查表:

分为团队协作、敏捷流程、CI纪律、质量保证4个维度，共44个调查问题。针对团队中不同的角色，通过采样调查表的方式进行问卷调查。



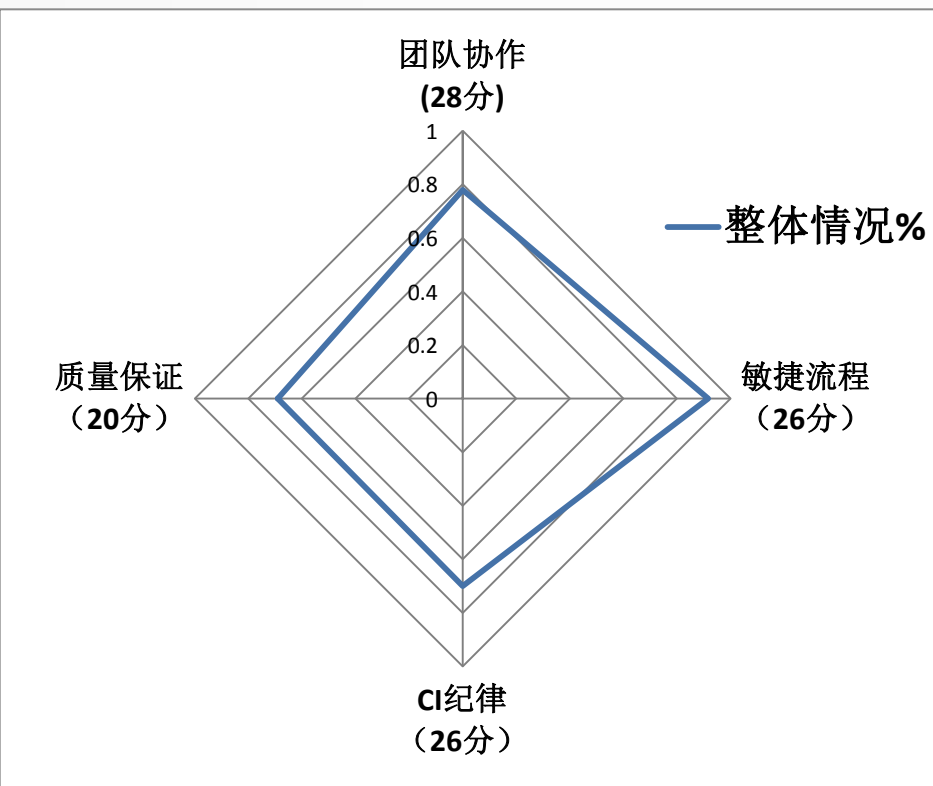
调查后的问题，按照图中的四象限（横轴代表，Team做到该问题描述的情况的困难程度，纵轴代表，该问题描述的是敏捷的基本要求还是拔高要求。）对所有问题进行了分类，以CDAB的顺序，设置分值，分值为C-3, D-2, A-1, B-1, 总分为100分。

其中：C类题的个数为22，D类题的个数为12，A类题的个数为6，B类题的个数为4

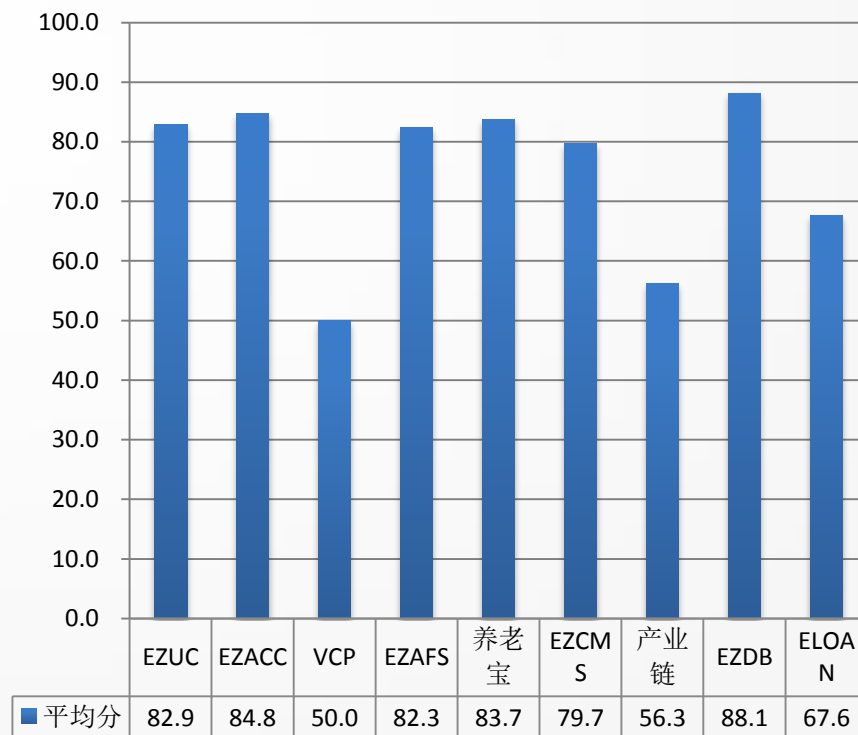
内部教练组织的敏捷成熟度评估

敏捷成熟度评估-201608

开发七部整体的敏捷成熟度分数为：76.31

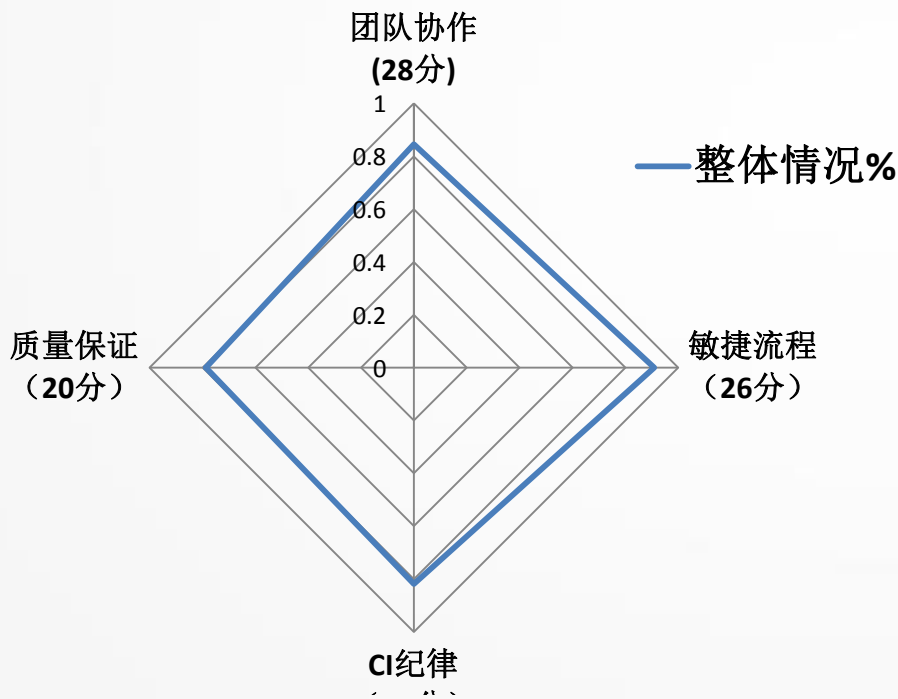


各产品打分情况

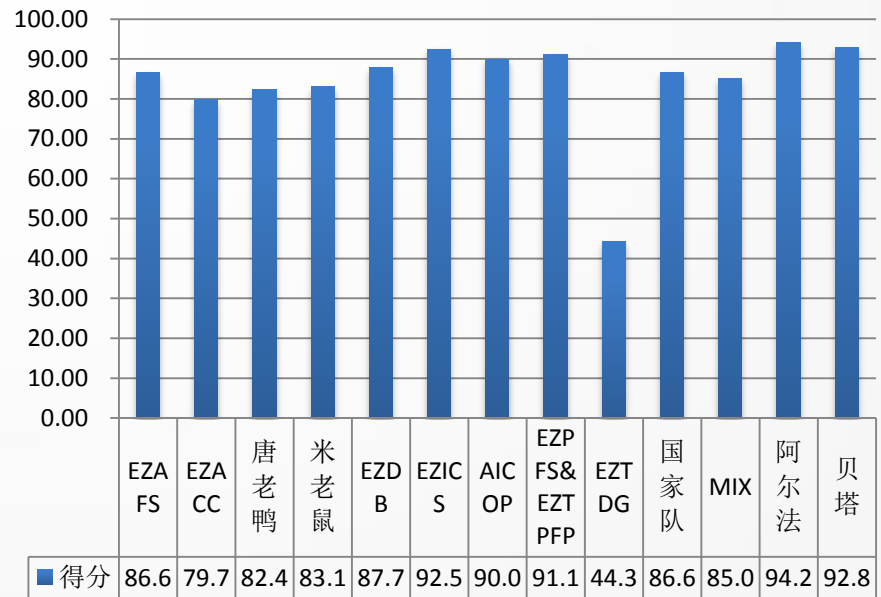


敏捷成熟度评估-201701

开发七部整体的敏捷成熟度分数为：84.33



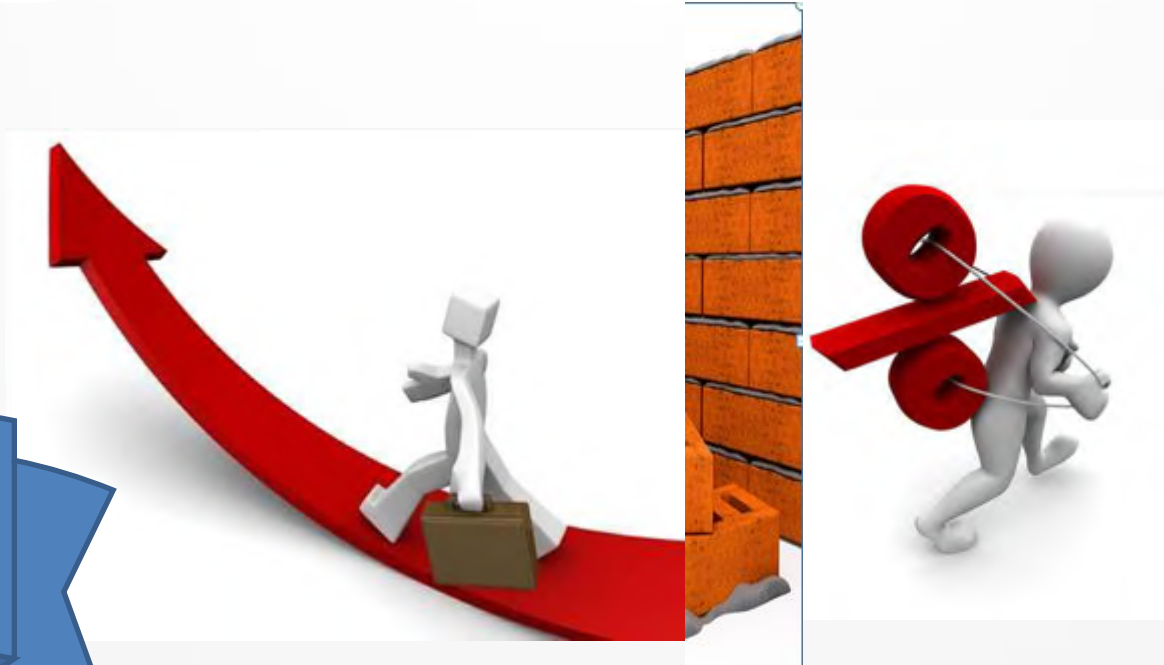
各Team成熟度打分



思考

- 在大型组织中“度量”是必要之恶！
 - 所谓“恶”，就是有明显的副作用，会导致员工工作上的不适，并产生非预期的影响。
 - 所谓“必要”，就是缺一不可。大型组织中没有好的度量，很难找到重点和确认效果。
- 度量的模型很重要，度量的姿势更重要。

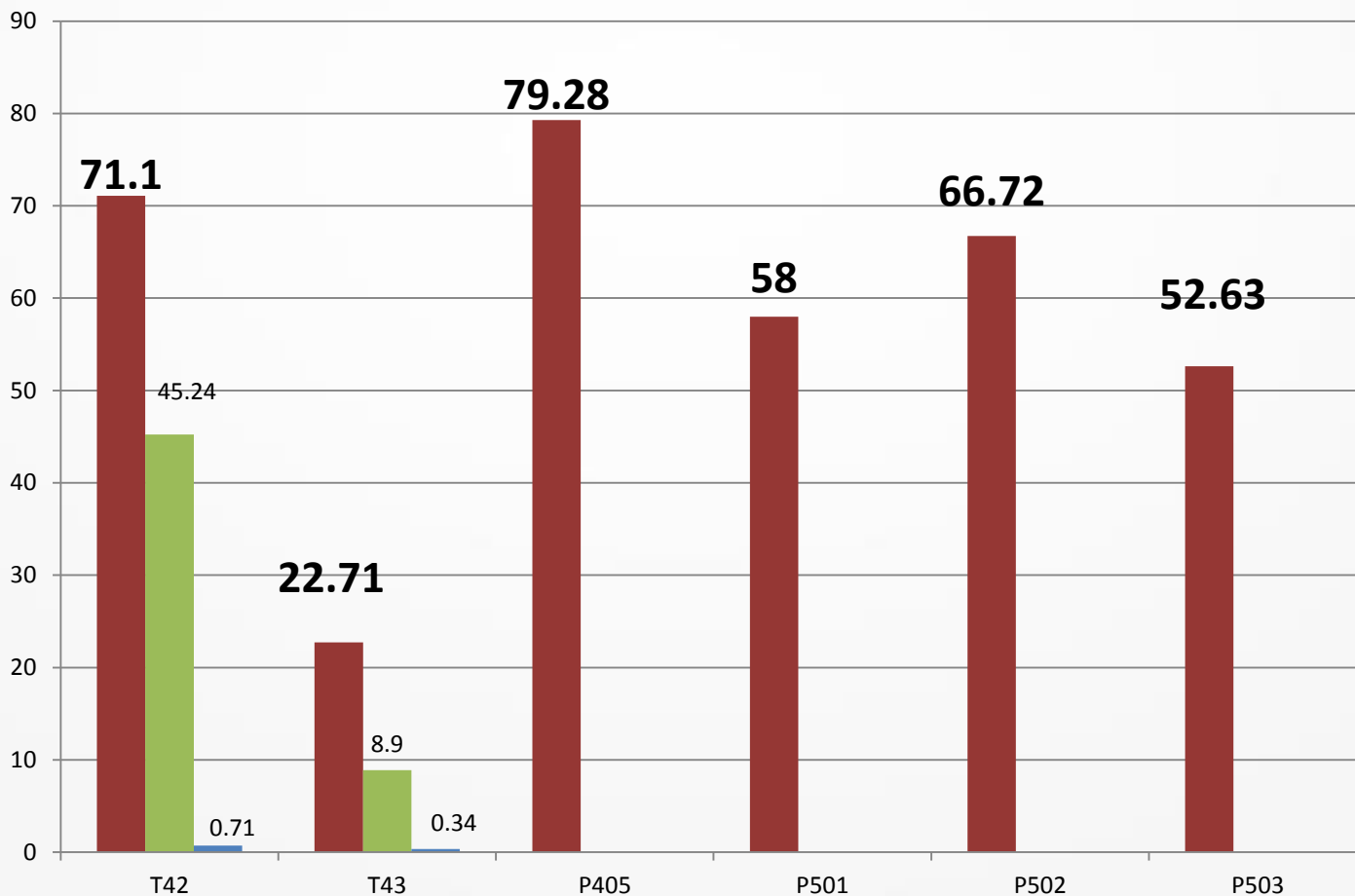
实践案例三：测试问题多，周期长



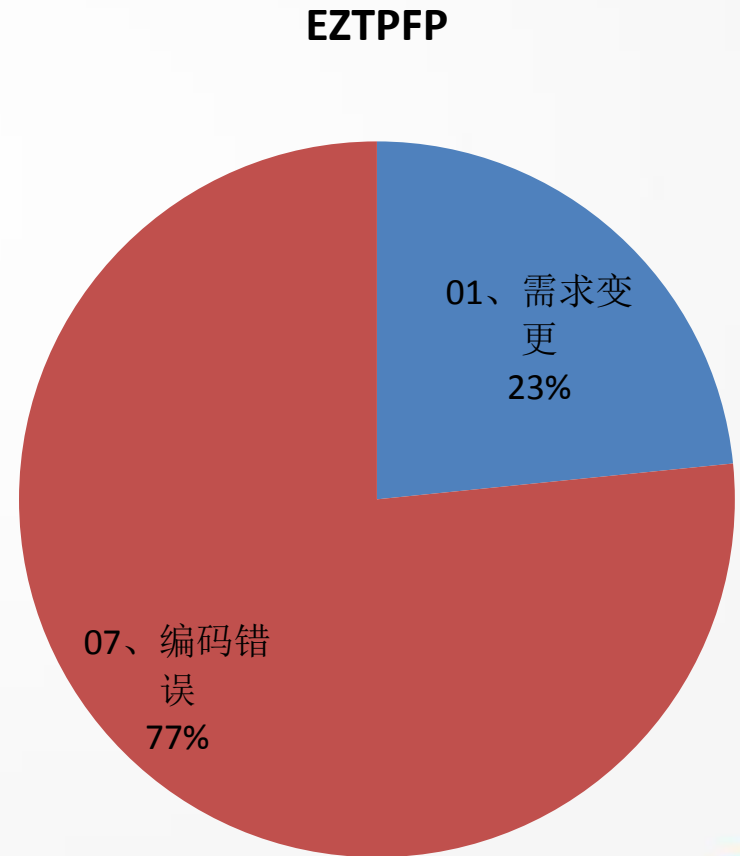
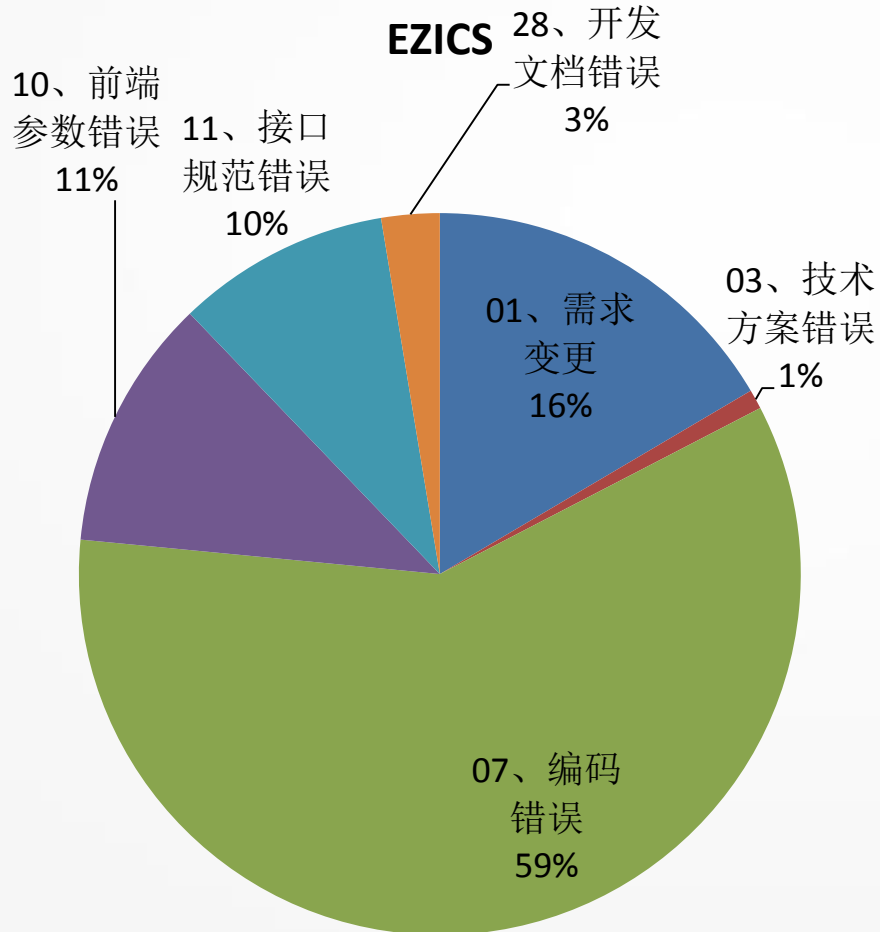
测试问题多

敏捷开发验收BUG率（按功能点统计/千功能点）

■ 开发七部合计 ■ 中心敏捷产品合计 ■ 中心传统瀑布型产品合计



问题分类分析

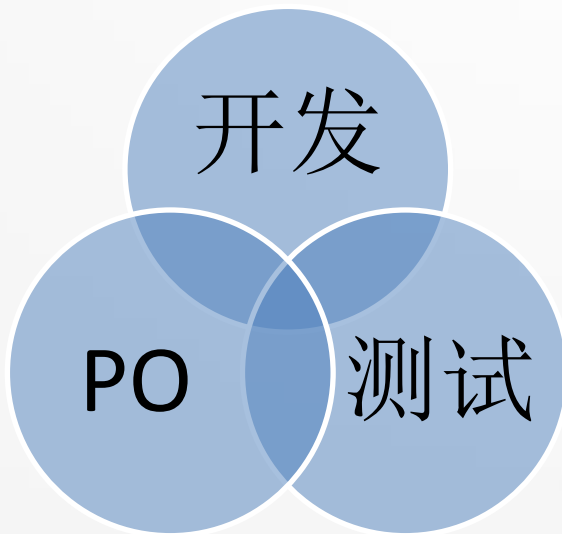


重点问题类型分析

- 编码问题: 6
- 需求变更: 6
- 文档错误: 3
- 前端参数错误: 3
- 接口错误: 2
- 版本错误: 2
- 技术方案错误: 1

理解错误；
理解正确，编写
错误，且没有被
及时发现。

PO未考虑到；
PO考虑了没说清
楚；PO考虑了，
说清楚了，开发
人员理解错了，
且没有被及时发
现。



开发 vs PO

敏捷团队开发工作要点更新

DOR of Sprint Planning

迭代计划会准入标准

候选故事已经估点

候选故事有唯一优先级

物理故事卡已经准备好

故事的主要 AC 点已经明确体现在故事卡上

故事的业务背景、范围清晰明确

故事具备开发条件，如接口已经明确约定

故事有界面的，已经有效果图

故事界面要素说明清晰，已经有界面元素说明文档

DOD of PB Grooming

产品清单梳理完成定义

参会人员角色齐备

完成了下一个迭代的候选故事选定

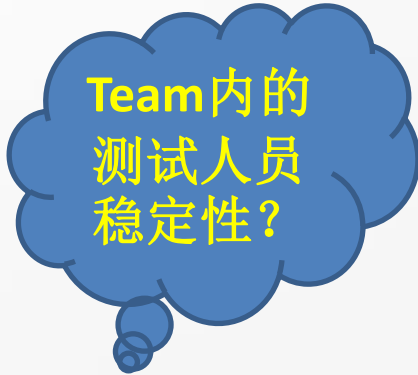
候选故事已经估点并指定了唯一的优先级

候选故事涉及的系统已经明确标出

候选故事涉及的接口已经明确约定

开发 vs 测试

- 一开始就参与（第一个迭代）
- 一地办公（坐在一起）
- 一个Team
 - 一起参与各个敏捷仪式
 - 一起领任务
 - 一起对迭代目标负责



Team内的
测试人员
稳定性？

开发 vs 自测

- 持续提升对单元测试的相关要求

产品整体情况(后台)

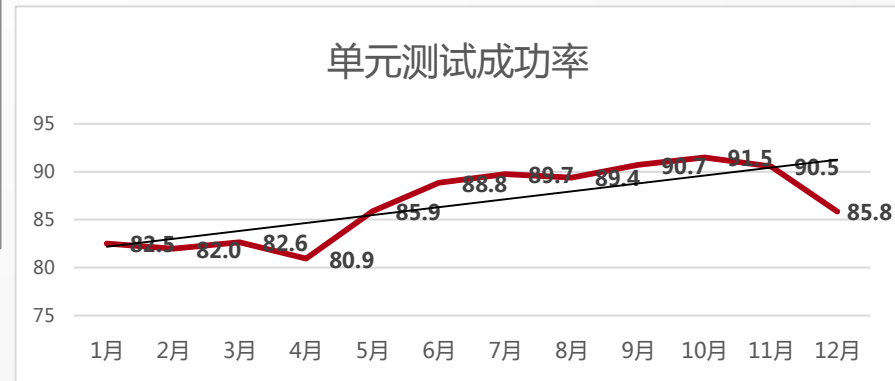
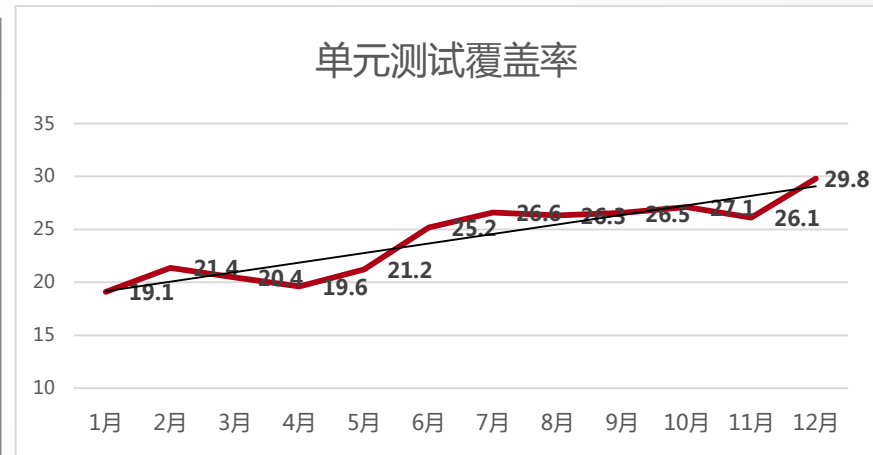
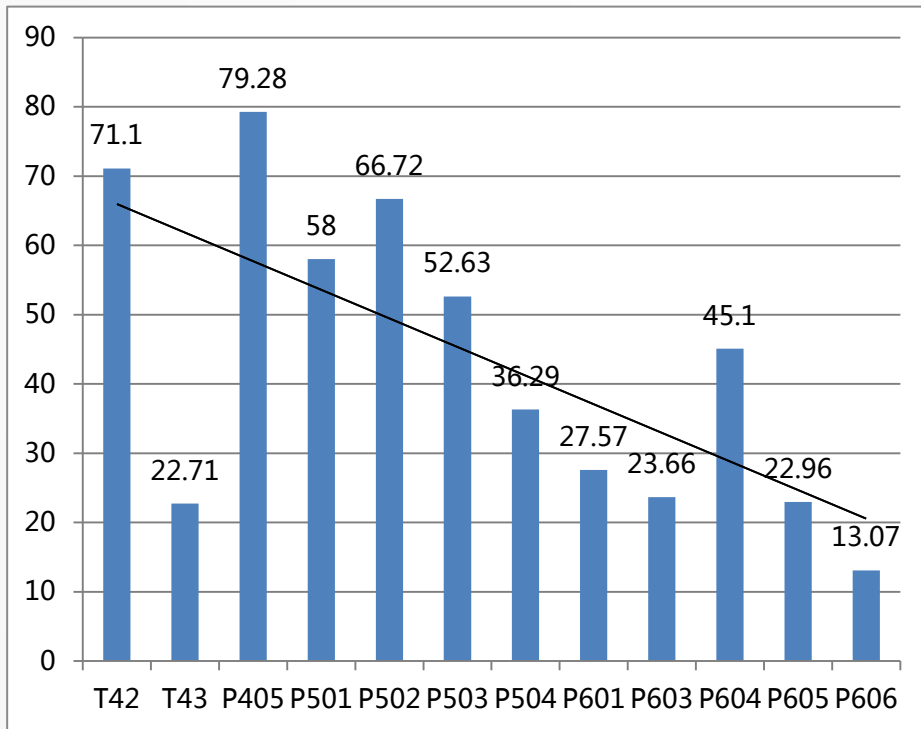
持续集成——软件质量改进和风险降低之道

CI目标:单元测试成功率 100%,覆盖率 50%+(现有指标不能下降);功能测试成功率 100%;功能测试案例数不少于15个(覆盖主要业务流程);问题数

产品名称	单元测试成功率	单元测试覆盖率	代码复杂度	代码重复度	功能测试成功率	Sonar问题数	安全问题数	性能波动
gpmp	534(100.00%)	54.32%	2.97	3%	53(100%)	308(0, 0, 308)	13(0,12,1)	----
ezcms	569(100.00%)	35.67%	2.92	11%	25(100%)	533(0, 0, 533)	0(0,0,0)	(0,0)
ezuc	3033(100.00%)	42.51%	1.83	21%	27(22.22%)	1269(0, 0, 1269)	17(0,15,2)	(0,0)
ezafs	2361(100.00%)	43.80%	4.75	15%	47(97.87%)	1135(0, 0, 1135)	10(0,5,5)	(0,0)
ezdb	1401(100.00%)	28.97%	1.69	19%	15(0%)	1236(0, 2, 1234)	47(7,38,2)	(0,0)
eloan	1166(100.00%)	19.50%	1.19	34%	43(9.3%)	1291(0, 0, 1291)	50(11,26,13)	(0,0)

成效

验收测试Bug率（个数/千功能点）



思考

- 措施总结
 - 增加交接点的DOR和DOD
 - “物理集中”进行筒仓破壁
 - 开发阶段质量内建



- 筒仓效应

谷仓是一种文化现象，成因是社会团体与组织具备特定的分工惯例。有些分工界定得具体清楚，例如纽约市政府有正式的组织架构，各个部门的内部组织与合作模式经过明确规定，采取阶层制度。然而，我们对世界的分类方式通常没有正式定义或明白说出，只源自我们往往无意间从周遭环境吸收到的规则、传统与惯例。

——吉莲·邵蒂，《谷仓效应》

实践案例四：敏捷与瀑布双速配合



敏捷开发与传统开发都是针对产品级的，当同一个项目同时涉及敏捷和传统产品的时候，在评审、接口联调、功能测试等方面存在双速配合的问题。

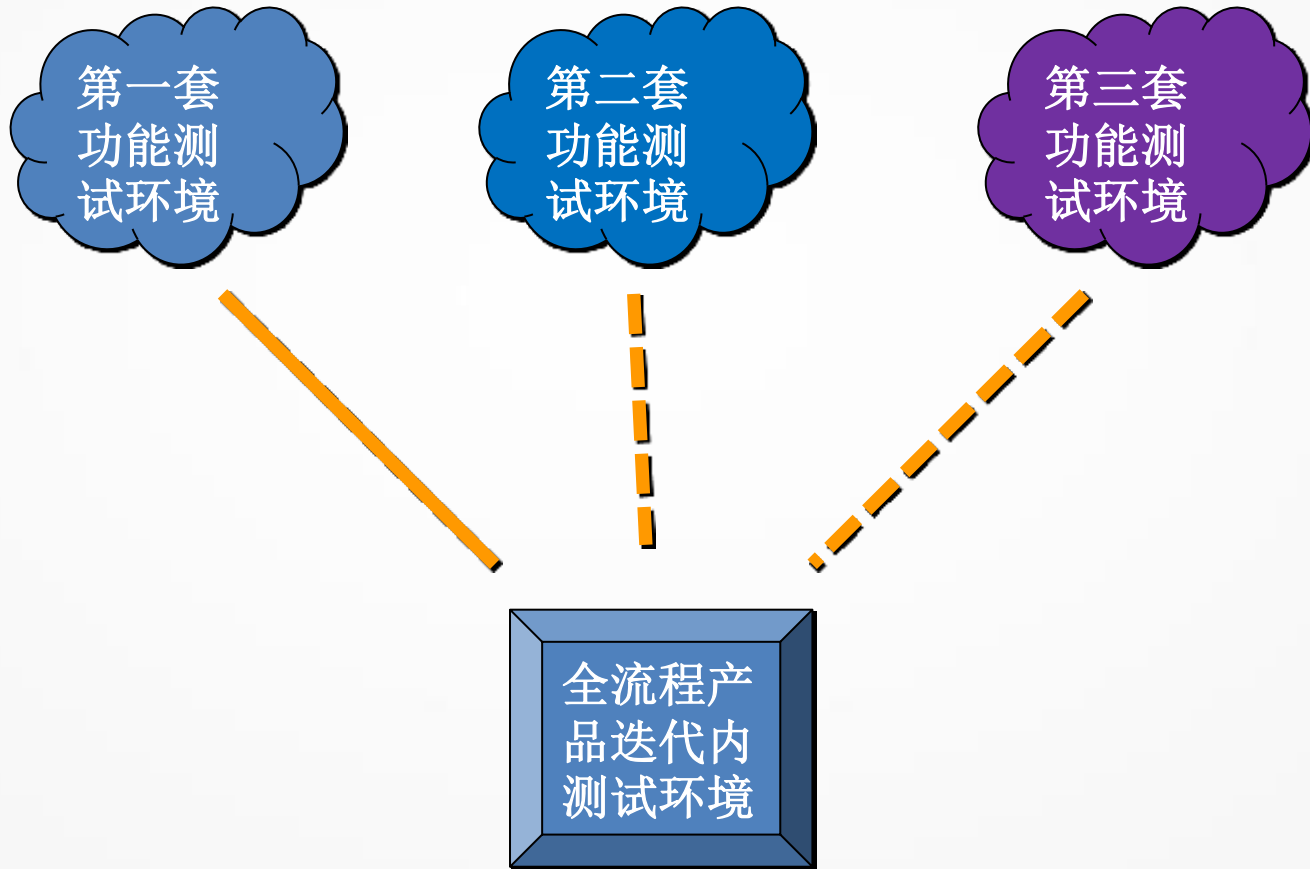


问题分析

目前在软件开发中两个产品之间最主要的配合有三个：

- 1、整体方案上的配合，即同一个需求，不同的产品各自需要做哪些修改；
- 2、功能测试或者叫做集成测试的配合，同一个项目的多个产品，需要真实的联调、联通，以及经过功能性测试后方可投产。
- 3、投产点配合，目前大部分产品的投产基本上是部署和发布的合体，这直接导致每一次投产都如临大敌。

管理要求：测试环境



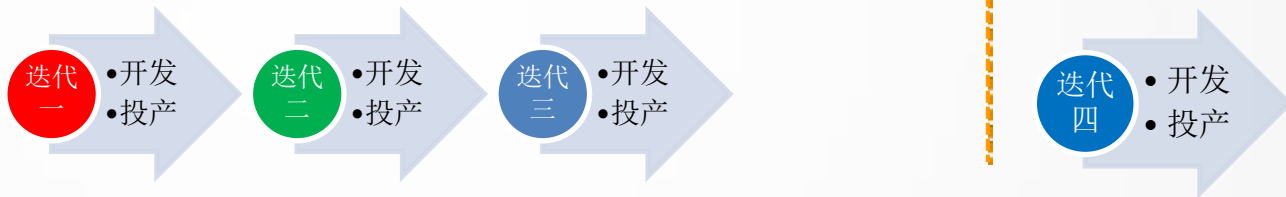
为了做到全流程敏捷：测试环境资源配置类生产，测试系统全联通，并且考虑成本和效益，我们采取了我们称之为“Mx+T”的全流程敏捷测试环境管理策略。

以快打慢，灵活的解耦策略！

前提：敏捷开发的Batch Size较小，或者产出速度足够快。



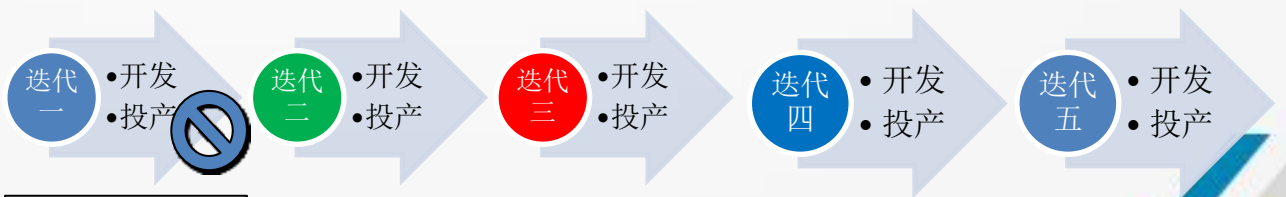
方法一：对齐测试



提交安装手册进行功能测试和投产



方法二：特性开关



通过特性开关关闭特性

开关清理

效果

行得通，行的痛！

思考

- 双模IT理念（Gartner 2014）
- 双模IT谈何容易
 - 测试环境的切换成本
 - 测试时间的协调成本
 - 特性开关的清理成本
- 双模IT的关键在解耦
 - 收益 > 成本 才有意义
 - 解耦的关键不仅仅在技术！



目录

-  我们的敏捷
-  我们的组织
-  我们的实践
-  我们的思考

我们的思考

银行业信息科技 监管制度汇编

大象之舞，监管之链

- 中国银行业监督管理委员会
- 银行业金融机构重要信息系统投产及变更管理办法（银监办发[2009]437号）

第十四条：银行业金融机构应统一组织协调重要信息系统投产及变更工作，制定年度投产及变更规则，编制实施计划和方案，确定实施策略和步骤，明确岗位职责，确保关键岗位职责分离。

第十六条：银行金融机构应建立重要信息系统投产及变更内容评审和审批、授权机制。

第二十条：银行业金融机构应建立与生产环境相隔离的测试环境，测试环境应模拟生产环境的真实情况。

舰船在行动，转型需耐心！

ONE MORE THING



敏捷实施情况调查



AgileDoer

