

浙江移动敏捷与DevOps尝试

中国移动浙江公司
李海传
2016.7.19





目录

I 背景介绍

II 敏捷转型-管理实践

III DevOps实践-工程效率提升

IV 小结

从价值创造的角度，软件研发活动可以分为几类：

- 创造价值的活动：写代码
- 不创造价值但是价值实现必须的活动：代码编译、部署
- 不创造价值也不是价值实现必须的活动：写BUG、等待

$$\text{过程的有效性} = \frac{\text{创造价值活动所花费的时间}}{\text{整个过程所花费的时间}}$$

测试 **40%** 在等待（环境、数据）

开发规范有 **10+** 种

1次上线要提交 **33+类** 文档种

软件开发的“最后一公里”，是指软件满足了功能需求之后，尚未投入实际运行并创造业务价值的阶段。软件开发者——尤其是面对交付压力的软件开发者——常常对“最后一公里”视而不见。但它确实正在成为业务软件交付中最大的压力点。

改一行代码的上线

要花费 **5** 人天工作

上线前手工提交的物件超过

10+ 种

上线当晚重启加

验证 > **5** 个小时

故障恢复时间 (MTTR) >

20 分钟

精益生产定义的七大浪费，很不幸地我们基本都有了！！！！

七大浪费

过剩

等待

库存

缺陷

任务
切换

移动

额外
处理

工程代码庞大、结构复杂，难以快速集成和发布。

单系统工程代码

700W

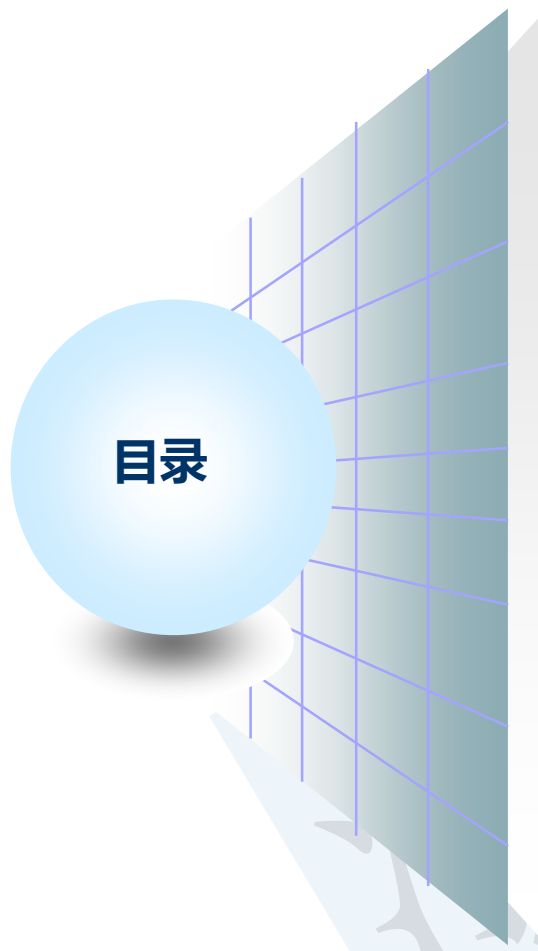
+ 行

编译时间>

30 分钟

生产发布>

2 小时



目录

I 背景介绍

II 敏捷转型-管理实践

III DevOps实践-工程效率提升

IV 小结

敏捷Scrum知识体系

业界实践

公司目前现有实践

定义

敏捷实践集

辅导

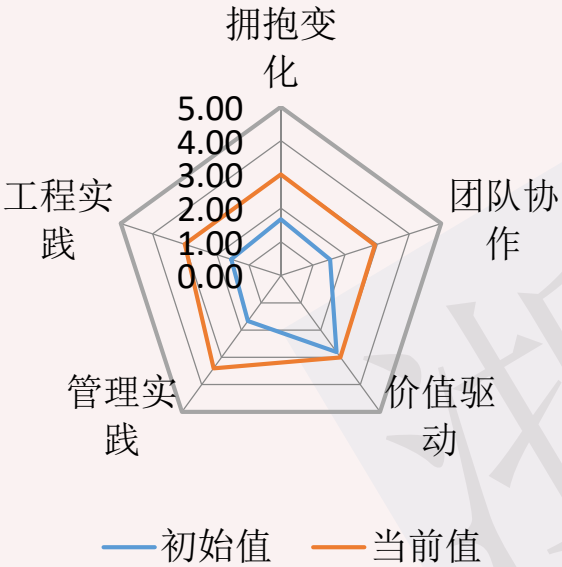
ETC 成员组成		
角色	姓名	职责
项目总代表	沈健、万炜、项建晨	1. 定义转型大方向，把握转型重点和方向； 2. 参与ETC Easeling和计划制定和工作评审；
ETC SCRM PM/TL	何宗富	1. 协调和组织ETC团队的敏捷转型推进工作； 2. 发现组织敏捷转型障碍，推动解决；
产品总负责人	Chen, 田建	1. 协助管理层定义转型方向、目标； 2. 制定转型方案； 3. 对组织管理层、试点团队进行转型辅导； 4. 配合ETC的Scrum Master进行组织敏捷转型；
内部敏捷教练小组	王平、郭龙彪、李一冰	1. 参与ETC Easeling和计划制定和工作评审，参与各种敏捷培训会议； 2. 负责组织敏捷的推广实施工作；
产品总代表	王作、陈松、林晓斌	1. 参与ETC Easeling和计划制定和工作评审，参与各种敏捷培训会议； 2. 从产品、业务分析领域为ETC Backlog提供输入； 3. 推进产品敏捷开发业务分析人员相关技能落地实施；
开发团队成员代表	王云天、刘晨、李刚	1. 参与ETC Easeling和计划制定和工作评审，参与各种敏捷培训会议； 2. 从开发/测试/运维成员的角度为ETC Backlog提供输入；

- ETC工作时间的节奏
 - 迭代、例会时间/周期
- 决策方式
 - 冲突处理
 - 方案选择
- 沟通方式
 - ETC内部、外部、团队
 - 交付物管理方式
- 成员进入和退出机制

建立敏捷实施成熟度标准，以测试作为运营手段，定期评测在需求、开发、测试、交付、运维各环节的执行质量，通过ETC组织驱动敏捷方法持续执行。

系统的敏捷实施质量测试

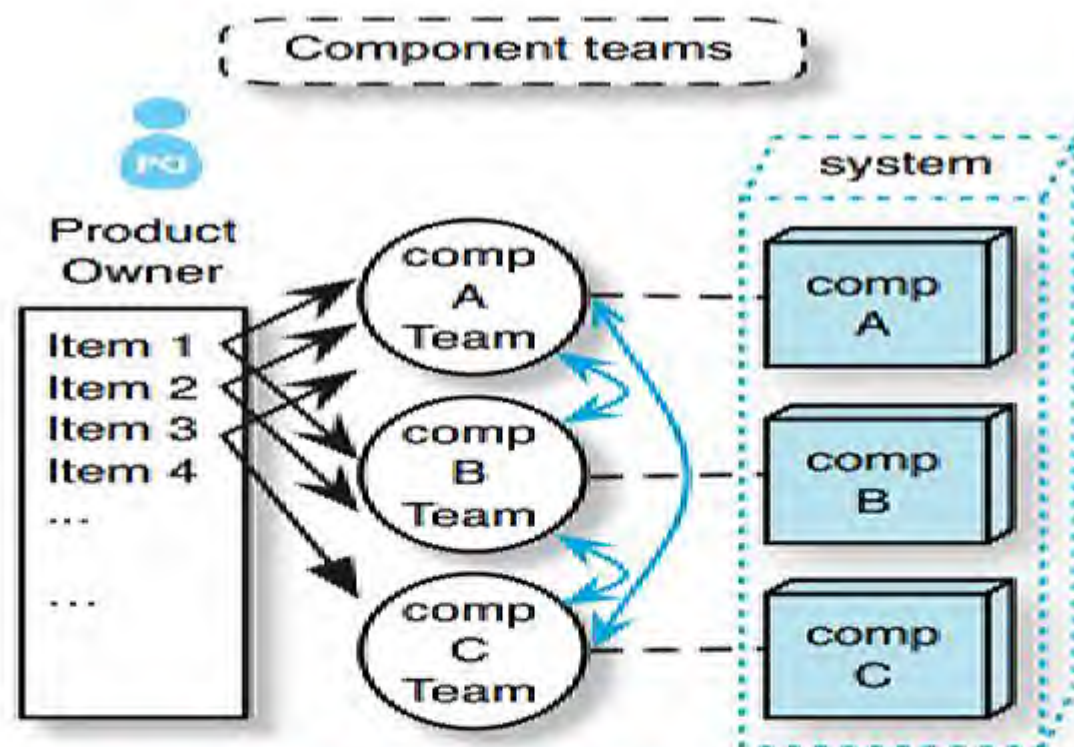
敏捷实施成熟度测试



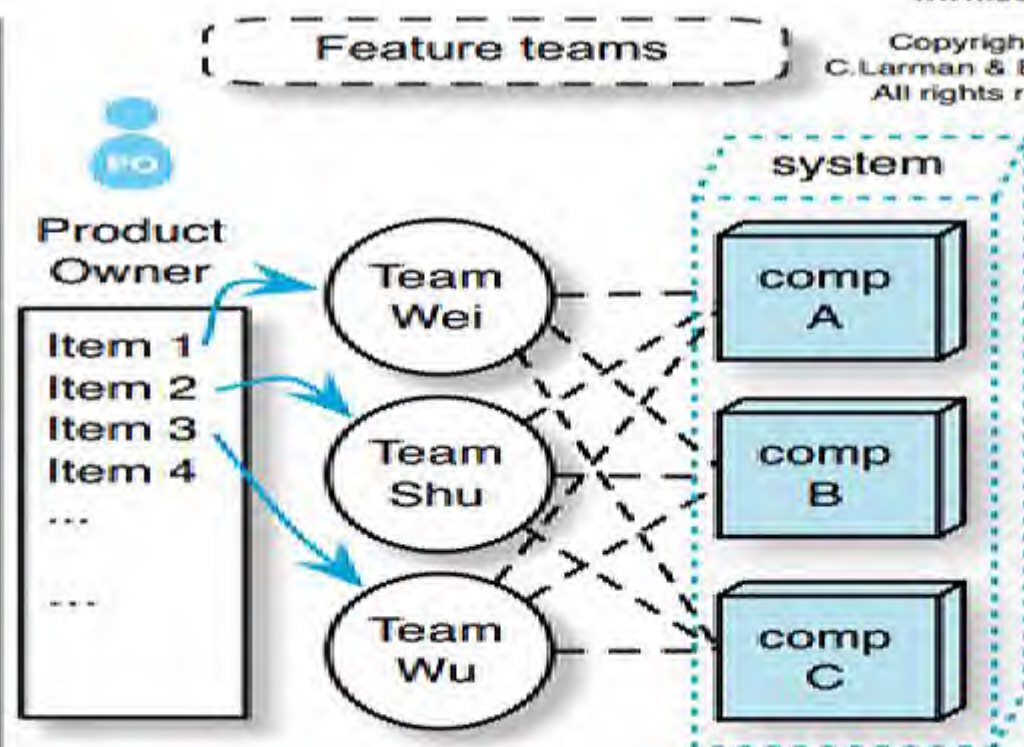
•从拥抱变化、团队协作、价值驱动、管理实践、工程实践五大方面对敏捷实施成熟度进行全方位评分。

敏捷转型前得分为1.55分，当前得分3.5分。





Work from multiple teams is required to finish a customer-centric feature. These dependencies cause waste such as additional planning and coordination work, hand-offs between teams, and delivery of low-value items. Work scope is narrow.

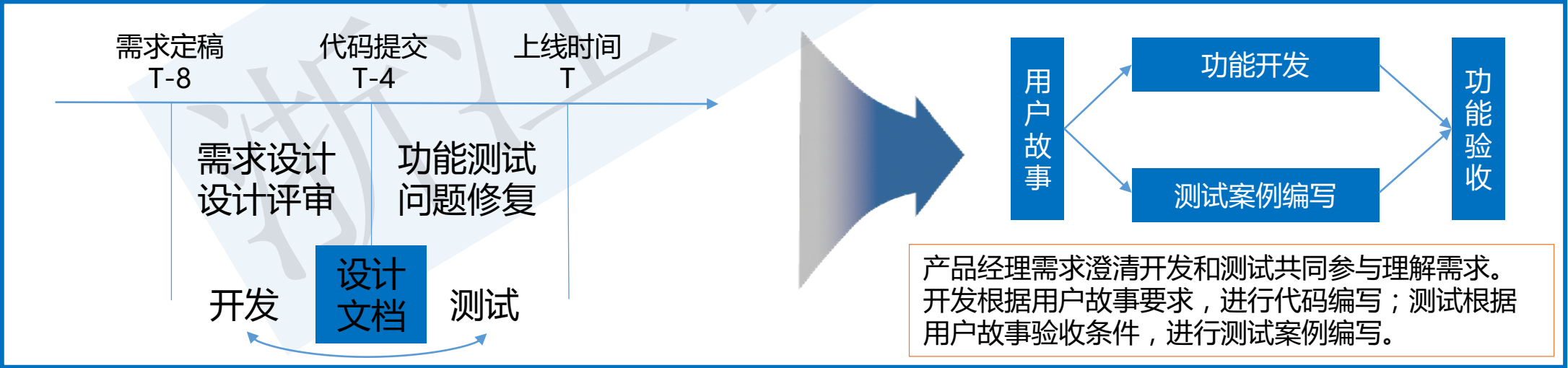


Every team completes customer-centric items. The dependencies between teams are related to shared code. This simplifies planning but causes a need for frequent integration, modern engineering practices, and additional learning. Work scope is broad.

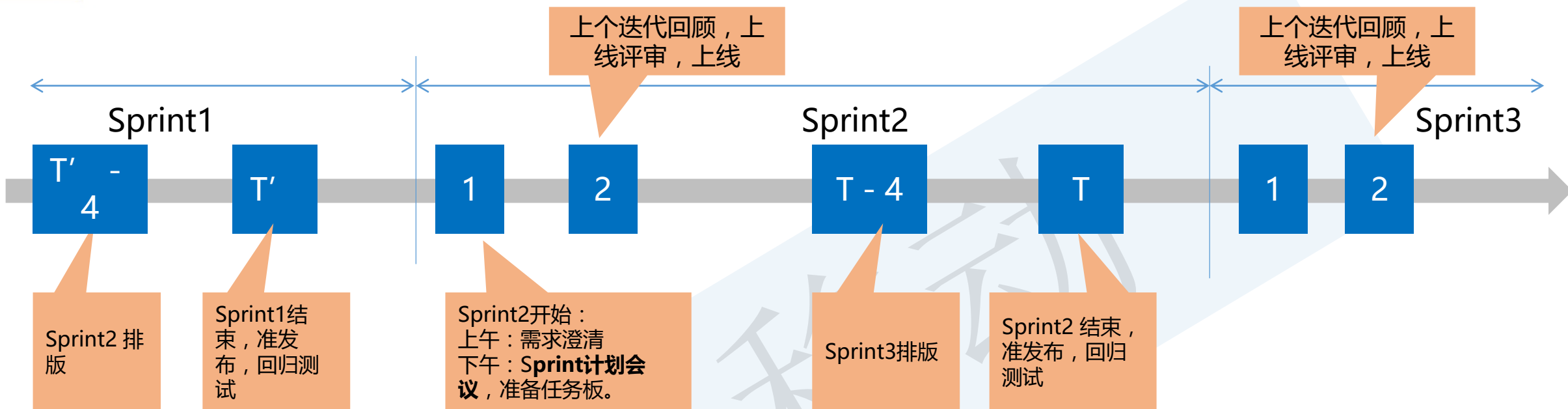
团队融合



工作方式转变







SCRUM的4+1活动

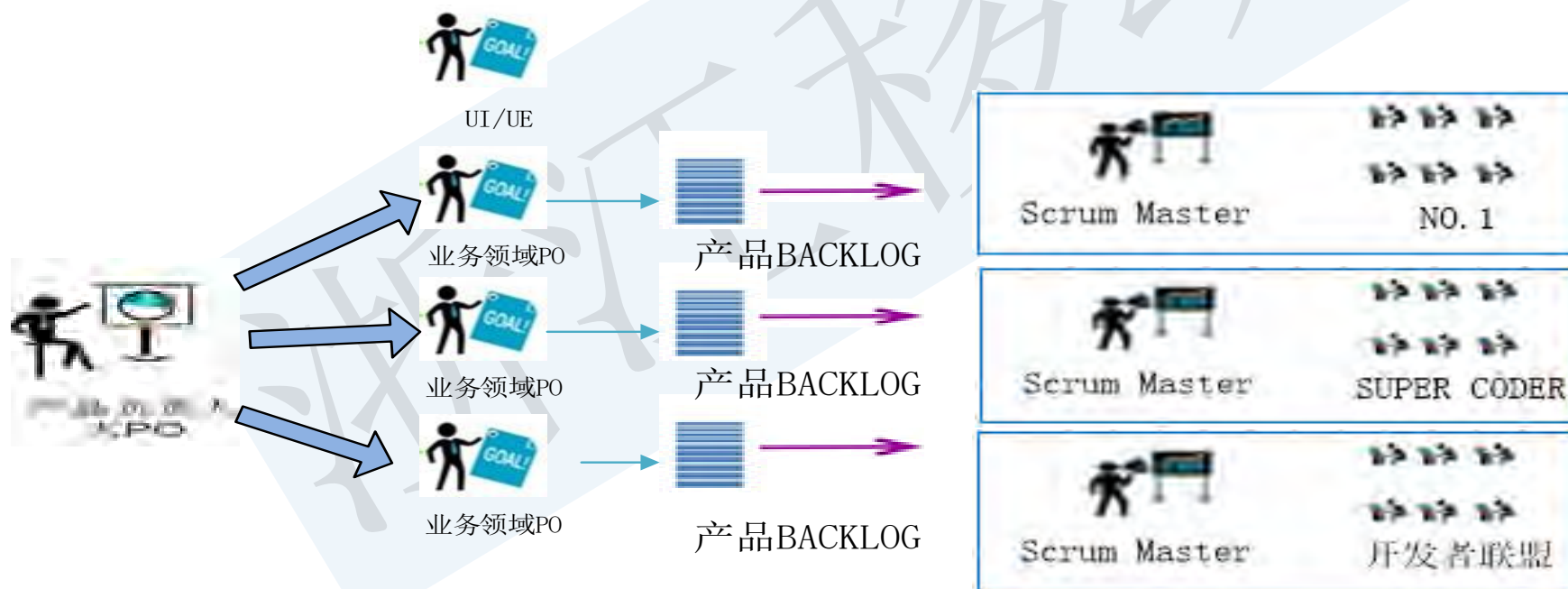
- 4个会议：
 - Sprint计划会议
 - 每日例会
 - Sprint评审会议
 - Sprint回顾会议
- 1个活动：
 - 需求澄清会

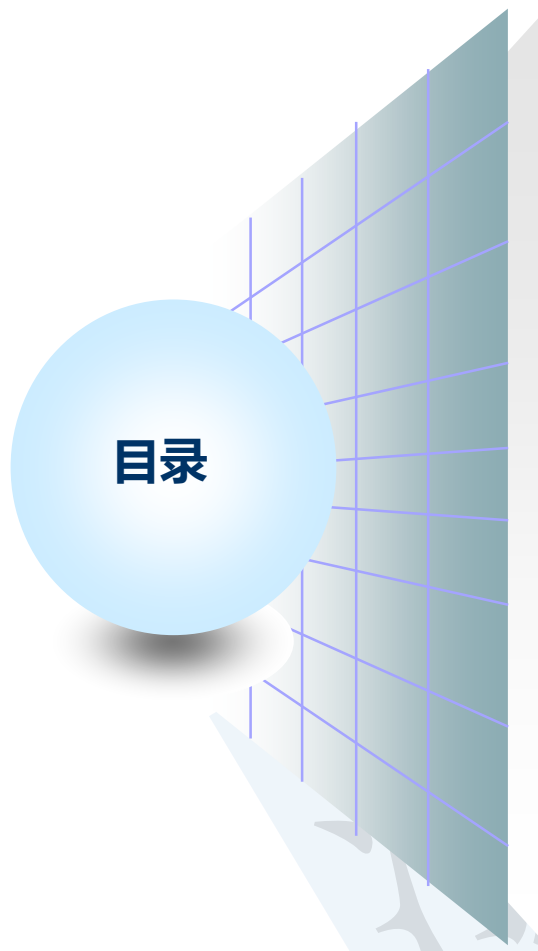
Sprint的每天：
召开**每日例会**，回顾昨天的工作、规划今天的工作，发现问题。
更新Sprint看板和燃尽图。**如果燃尽图2天没有下降，预警。**

T-4到下个迭代的第一天PO要将已经排版的需求梳理和细节确认完毕（需求管理员 + 系分及PO + 团队PO助理）

在Sprint开始第一天上午召开**需求澄清会议**，为下个Sprint做准备。（大约占0.5天），**Sprint中开展Show Case实践**，开发成员完成某个功能（自测完毕）向PO演示所完成的功能。

- 产品负责人PO**：作为产品的owner，代表客户决定产品需求以及优先级，是Team对于需求的唯一诉求人，帮助团队屏蔽传统的需求纠纷干扰。
- 业务领域PO**：负责具体功能的用户故事、验收标准编写，把握故事的优先级，保证团队瞄准正确的目标。每个团队都有属于自己的
- Scrum Master**：确保scrum被理解和遵循；确保scrum运用的有效性；确保team内外不受干扰，使团队专注在开发生产中。
- 开发团队**：是一个自组织、跨职能的团队，一般规模控制在5-9个人。





目录

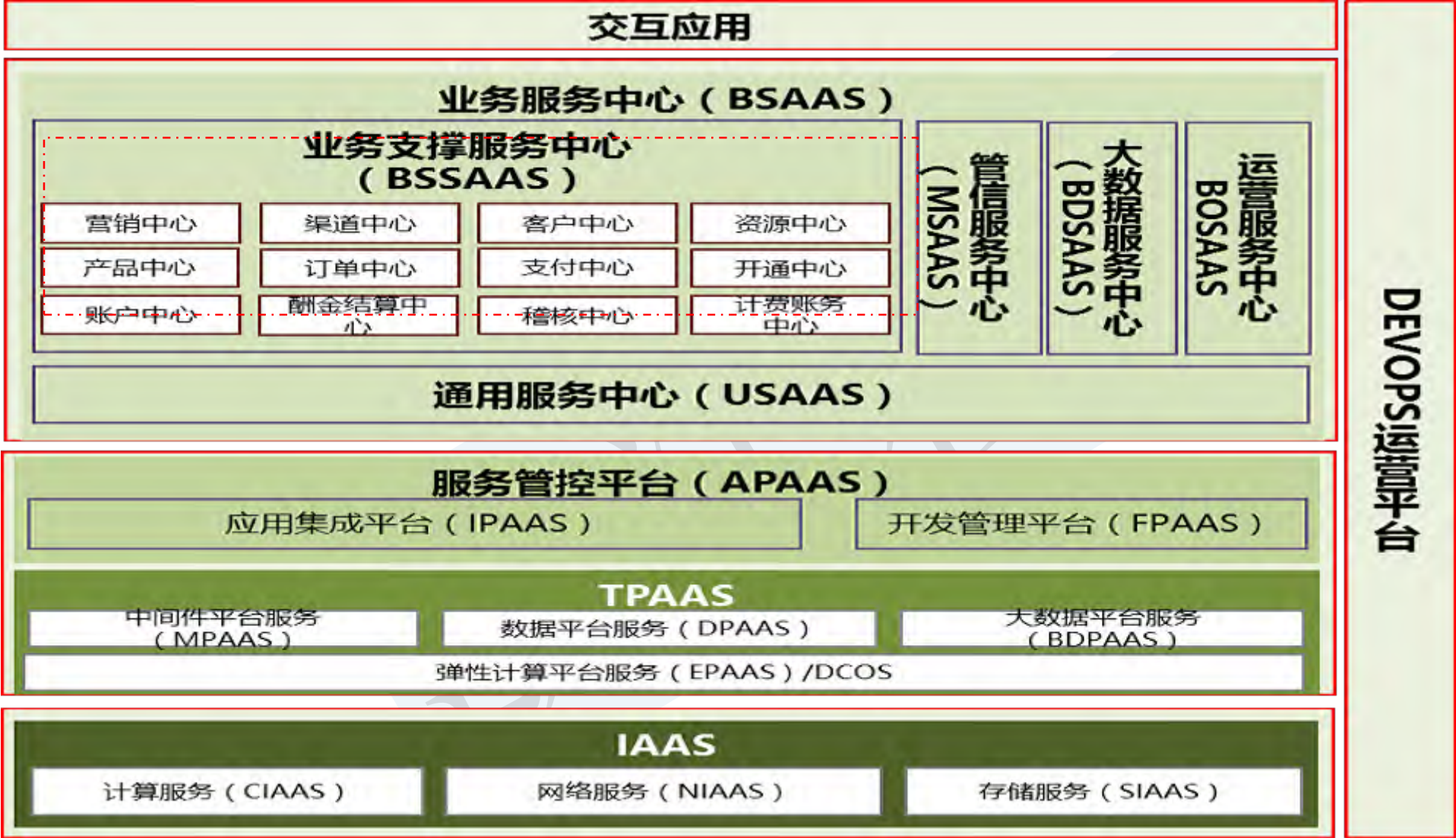
I 背景介绍

II 敏捷转型-管理实践

III DevOps实践-工程效率提升

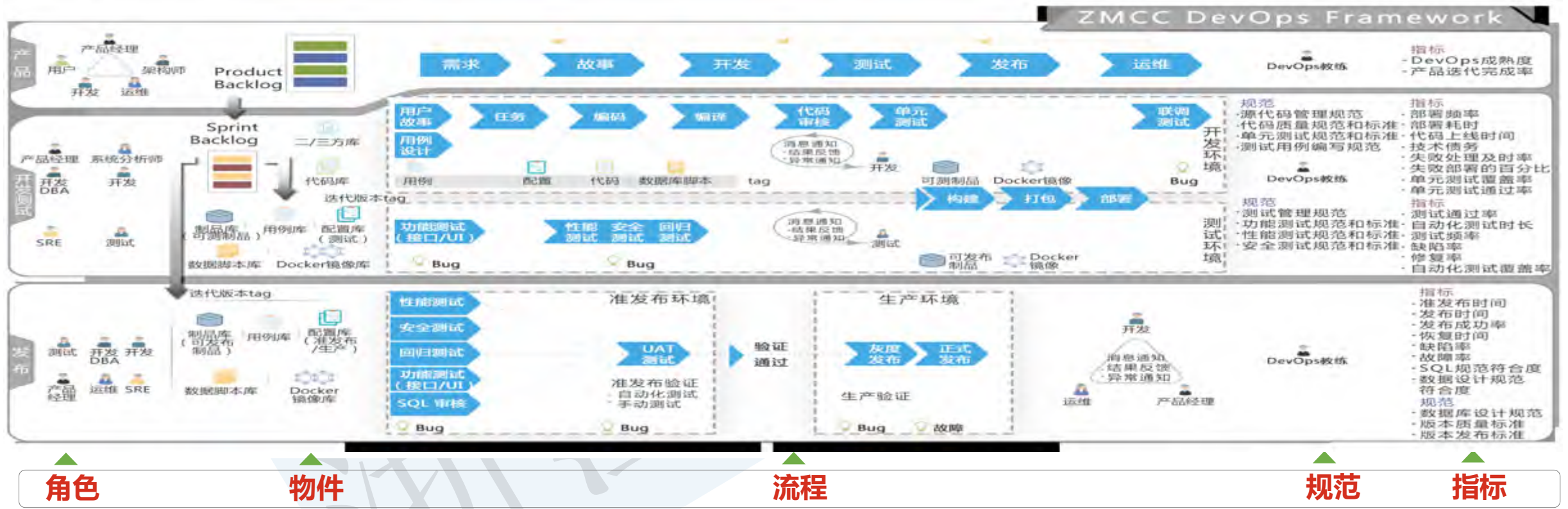
IV 小结

代码重构，拆解原CRM单体700W+行代码为12个业务中心。



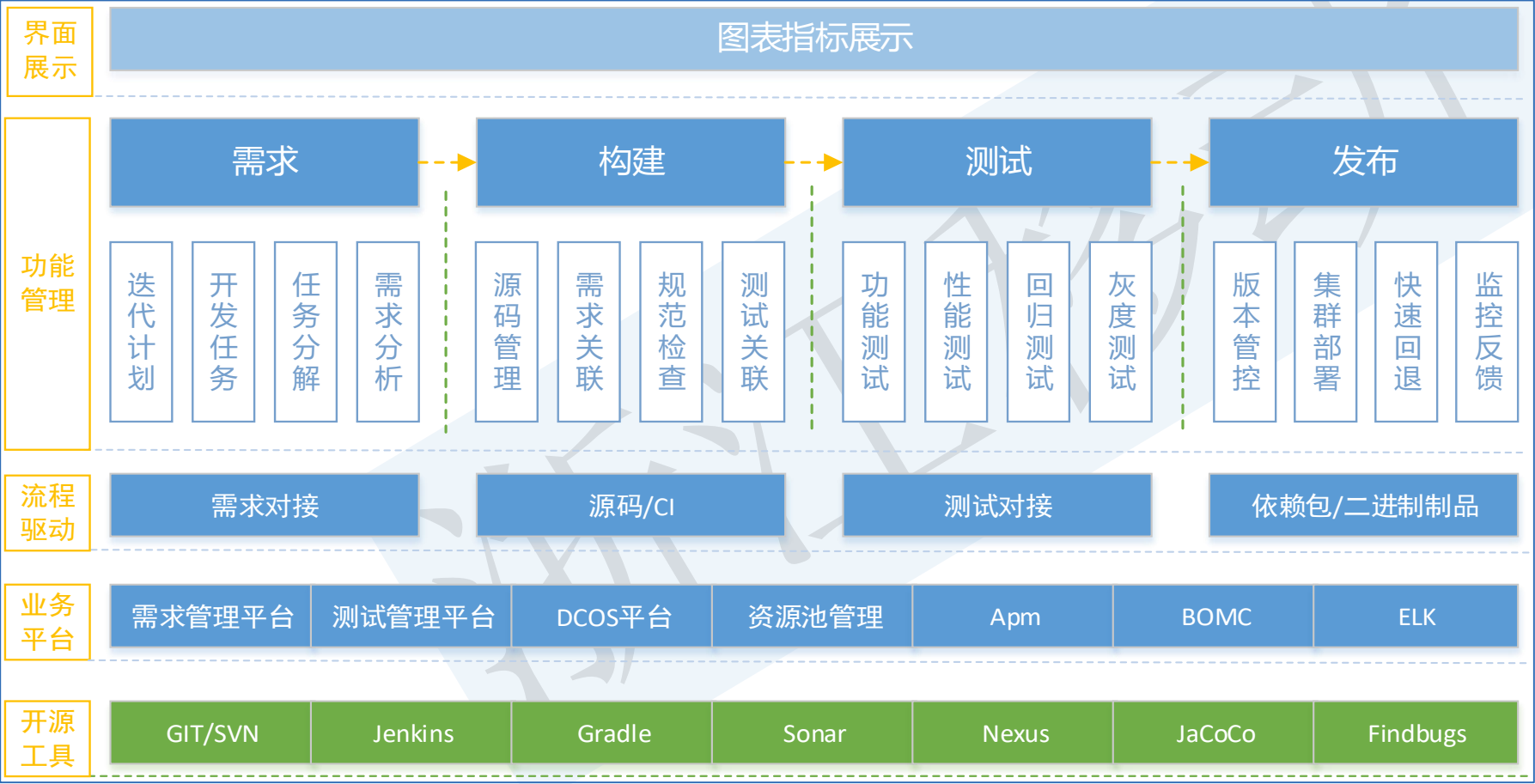
- **厚平台，轻应用，组件标准化**：标准的技术组件和业务组件；
- **开放的服务化架构**：服务原子化解耦和高性能云化服务调度框架，服务能力可对外开放；
- **高可靠和高容错**：任何节点和链路故障能够自动检测和优化，确保高可靠性；
- **DEVOPS**：快速迭代，持续交付；
- **数据化运营**：对服务运行全面监控，监控数据积累可视化；
- **应用级线性扩展**：随着资源加入，能进行微服务级线性容量和性能的扩展。

基于敏捷DevOps方法，结合已有的开发实践，建立DevOps Framework方法框架。该框架从全局上贯通了产品研发的整个流程，从细节上定义了流程中的角色、物件、流程、规范、指标等5个方面落地标准规范化。



- 5个方面规范化
- 角色：明确9种角色在各个阶段的职责。
 - 物件：明确研发过程中的4类输入物、5类输出物的规范。
 - 流程：明确研发过程中需要经过的4个环境20+个步骤，并要求实现工具自动化。
 - 规范：明确和质量相关的11个规范，并要求实现工具化落地。
 - 指标：明确3级指标体系，1级是成熟度指标、2级是效率和成功率指标，3级是具体质量指标。

建立敏捷交付(ADCloud)平台，打通从需求、代码构建、系统测试、上线发布全流程，精益管理软件开发过程。实现持续交付流程的全自动化。整合Sonar、Findbugs等质量分析管理工具，进一步提升代码质量。通过平台展示软件开发完整过程。



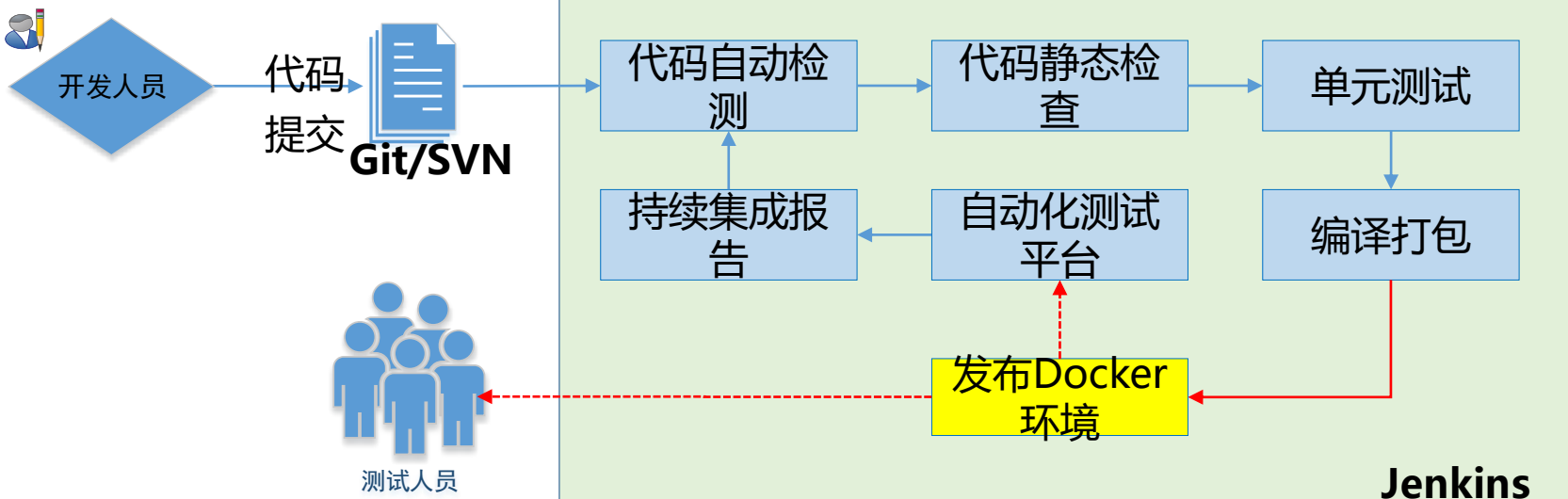
流程全贯通，打通从需求到运维的流程，建立需求、开发测试、部署上线的闭环流程。

全面自动化，通过持续集成平台完成代码自动编译、静态代码检查、单元测试、覆盖率统计、编译发布、集成部署等内容。提升开发测试流转速度，实现集成测试环境按天部署，代码提交和编译包、单元测试联动。

过程可视化，通过持续集成平台，集成代码扫描结果、编译结果、单元测试等报告，提供展示功能。打通集成测试平台测试报告和持续集成平台接口，使得从代码提交到集成部署报告的全界面统一展现。实现全流程、全过程可视化。

指标固化，平台上固化源代码、代码质量、单元测试、版本发布等11个质量相关规范。

通过Jenkins持续集成平台，实现代码自动化编译，单元测试自动化，集成代码扫描工具，自动打包发布，打通自动化功能测试平台。



主要工具：

代码仓库：Git/SVN进行管理；静态检查：Sonar集成。

单元测试：JUnit；单测覆盖率：JaCoCo插件；编译打包：Gradle。

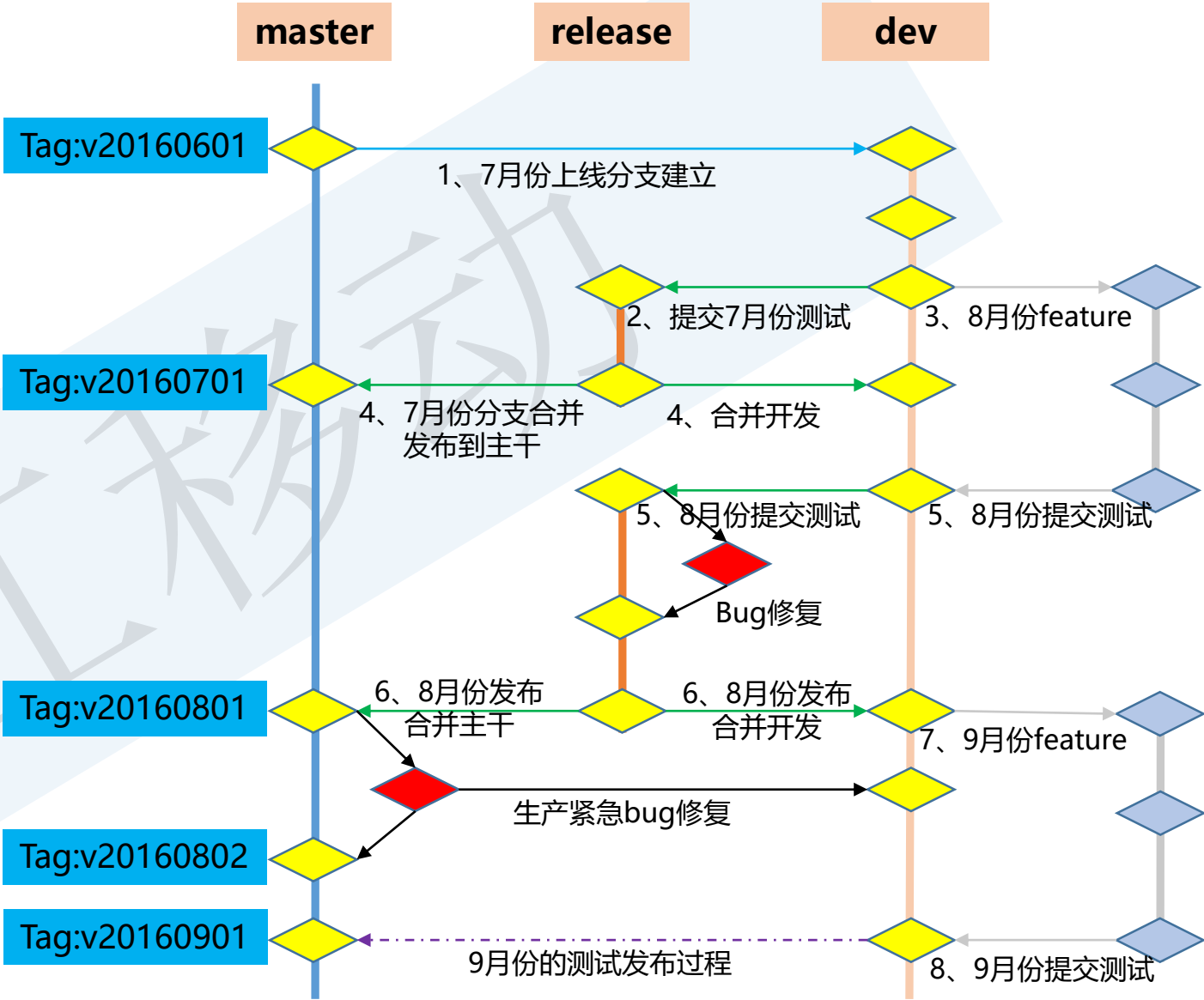
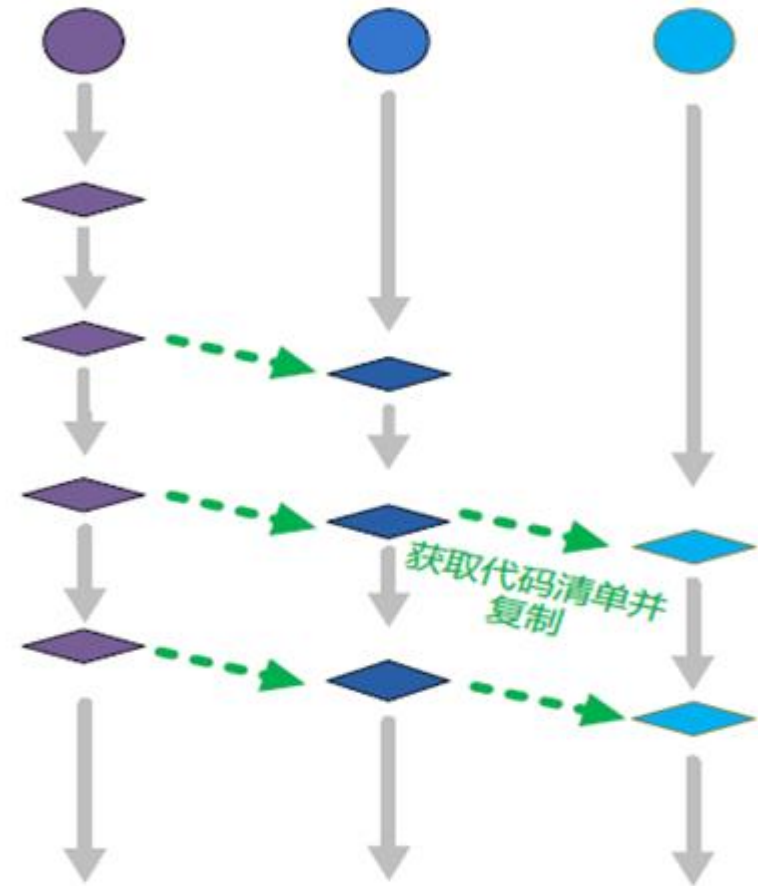
集成部署：shell脚本/服务接口；

自动化测试：Ruby，自动化测试平台；CI报告：Sonar，自动化测试平台。



移动研发源码管理过程

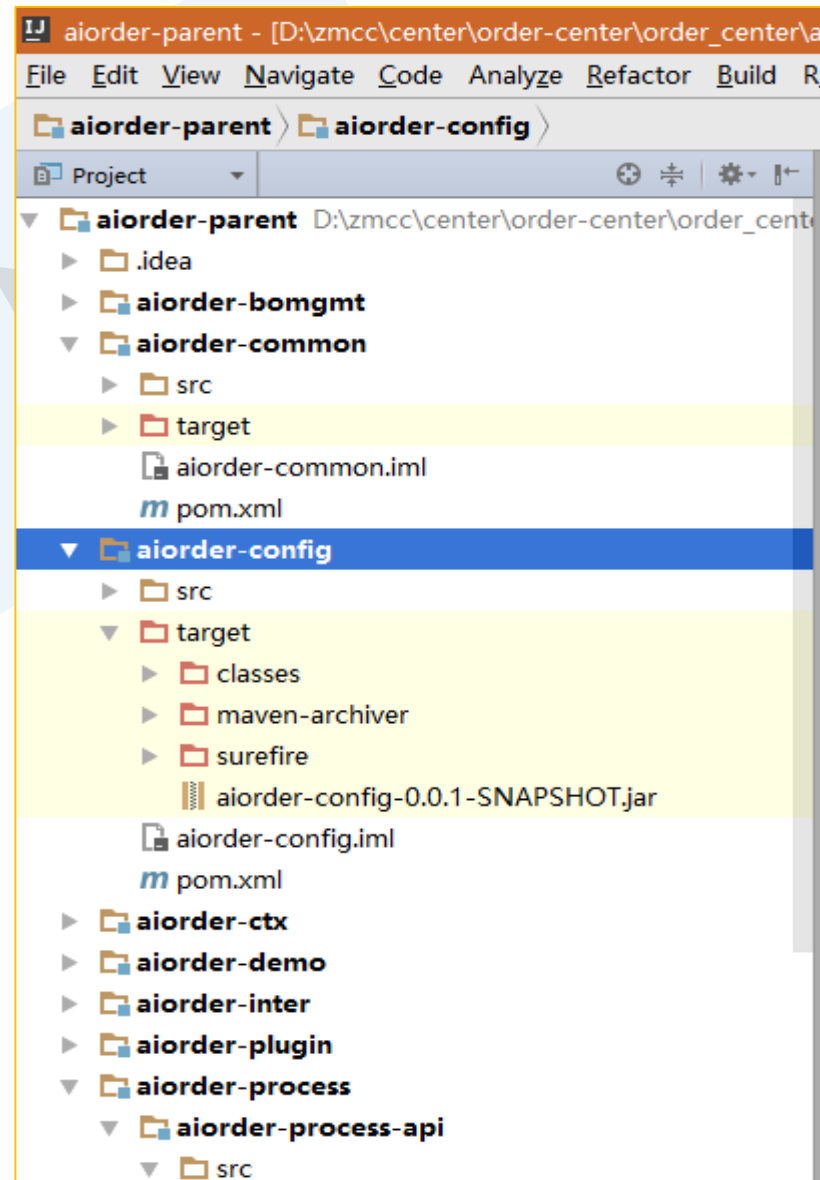
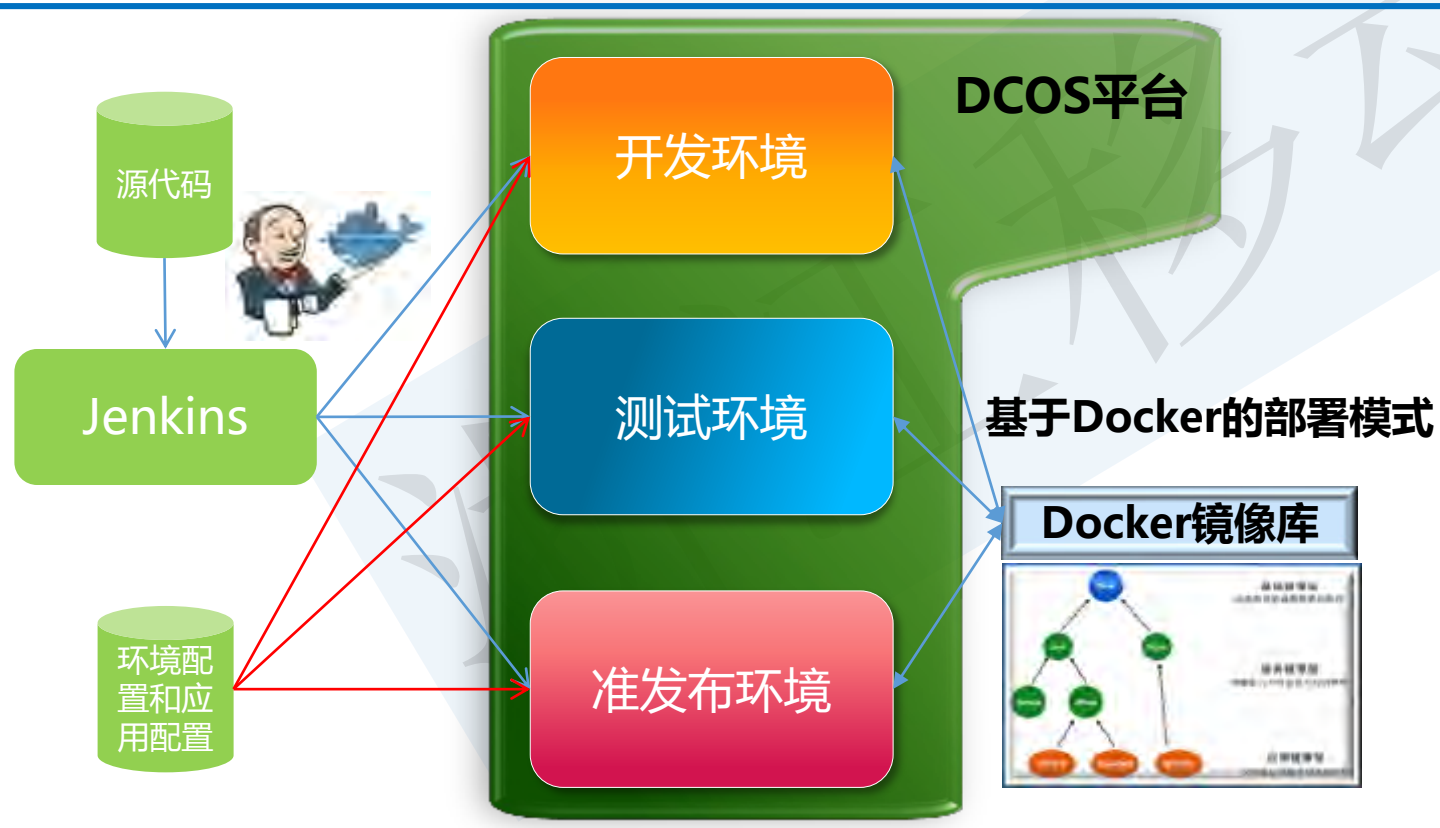
开发主线 联调主线 测试主线



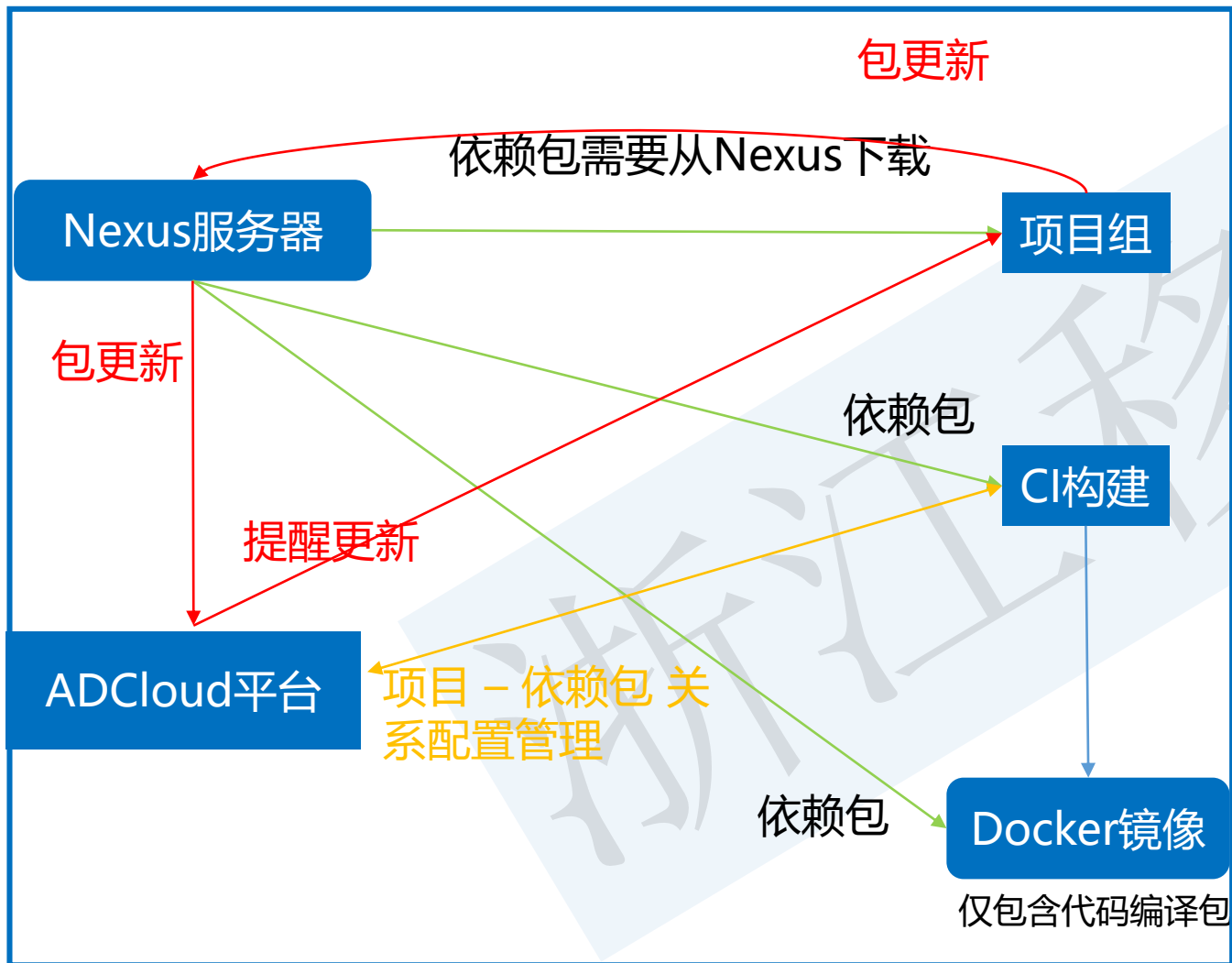
静态
打包



动态
运行



建立项目依赖包管理服务，统一管理依赖包



Nexus

Welcome Repositories

Refresh User Managed Repositories

Repository	Type	Health Check	Format	Policy
Public Repositories				
3rd party	group	ANALYZE	maven2	Release
center-release	hosted	ANALYZE	maven2	Release

Public Repositories

Browse Index Browse Storage

Refresh Path Lookup:

- org
- apache
- commons
- com.springsource.org.apache.commons.lang
- 2.4.0
 - com.springsource.org.apache.commons.lang-2.4.0.jar
 - com.springsource.org.apache.commons.lang-2.4.0.jar.md5
 - com.springsource.org.apache.commons.lang-2.4.0.jar.sha1
 - com.springsource.org.apache.commons.lang-2.4.0.pom
 - com.springsource.org.apache.commons.lang-2.4.0.pom.md5

```
<dependency>
    <groupId>com.asiainfo.crm.other:hamcrest-core</groupId>
    <artifactId>hamcrest-core</artifactId>
    <version>1.1.1</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:jmockit</groupId>
    <artifactId>jmockit</artifactId>
    <version>1.23</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:jmockit-coverage</groupId>
    <artifactId>jmockit-coverage</artifactId>
    <version>1.23</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:bcmail-jdk15on</groupId>
    <artifactId>bcmail-jdk15on</artifactId>
    <version>1.5.0</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:bcprov-ext-jdk15on</groupId>
    <artifactId>bcprov-ext-jdk15on</artifactId>
    <version>1.5.0</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:axiom-api</groupId>
    <artifactId>axiom-api</artifactId>
    <version>1.2.12</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:axiom-impl</groupId>
    <artifactId>axiom-impl</artifactId>
    <version>1.2.12</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.other:3des</groupId>
    <artifactId>3des</artifactId>
    <version>1.0</version>
    <scope>test</scope>
</dependency>

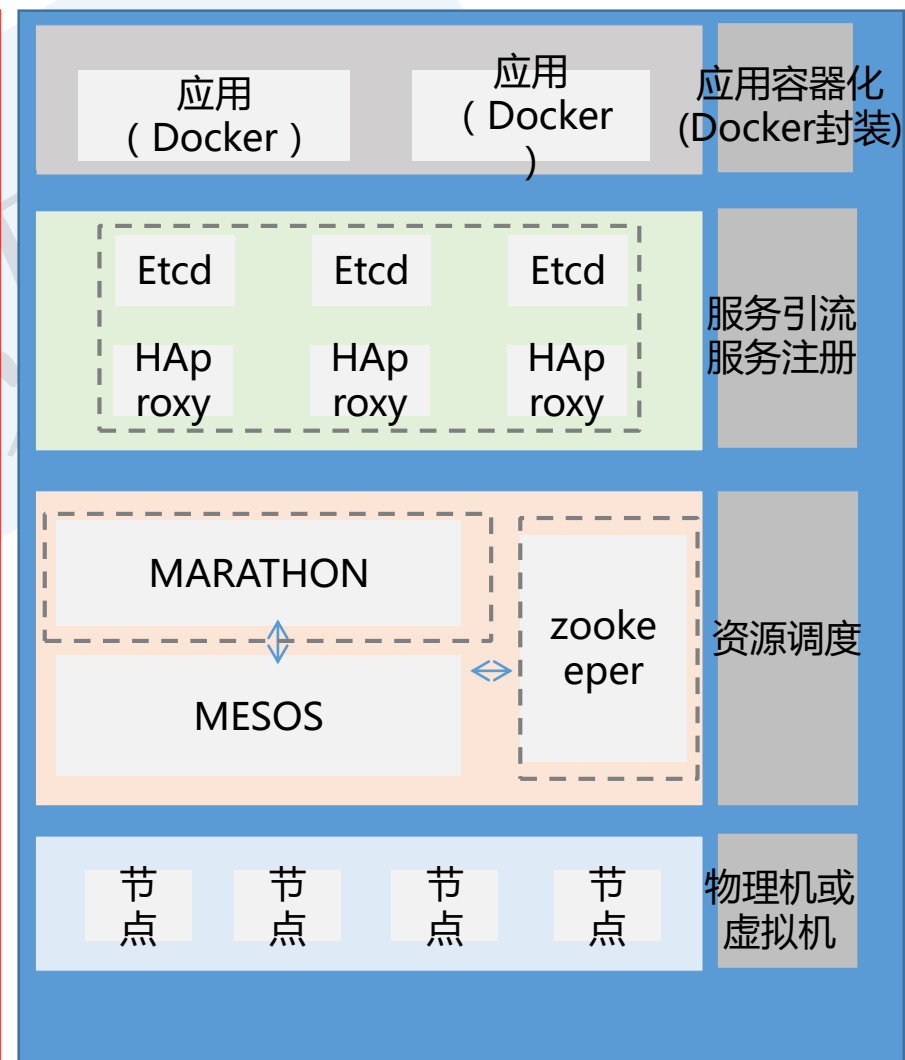
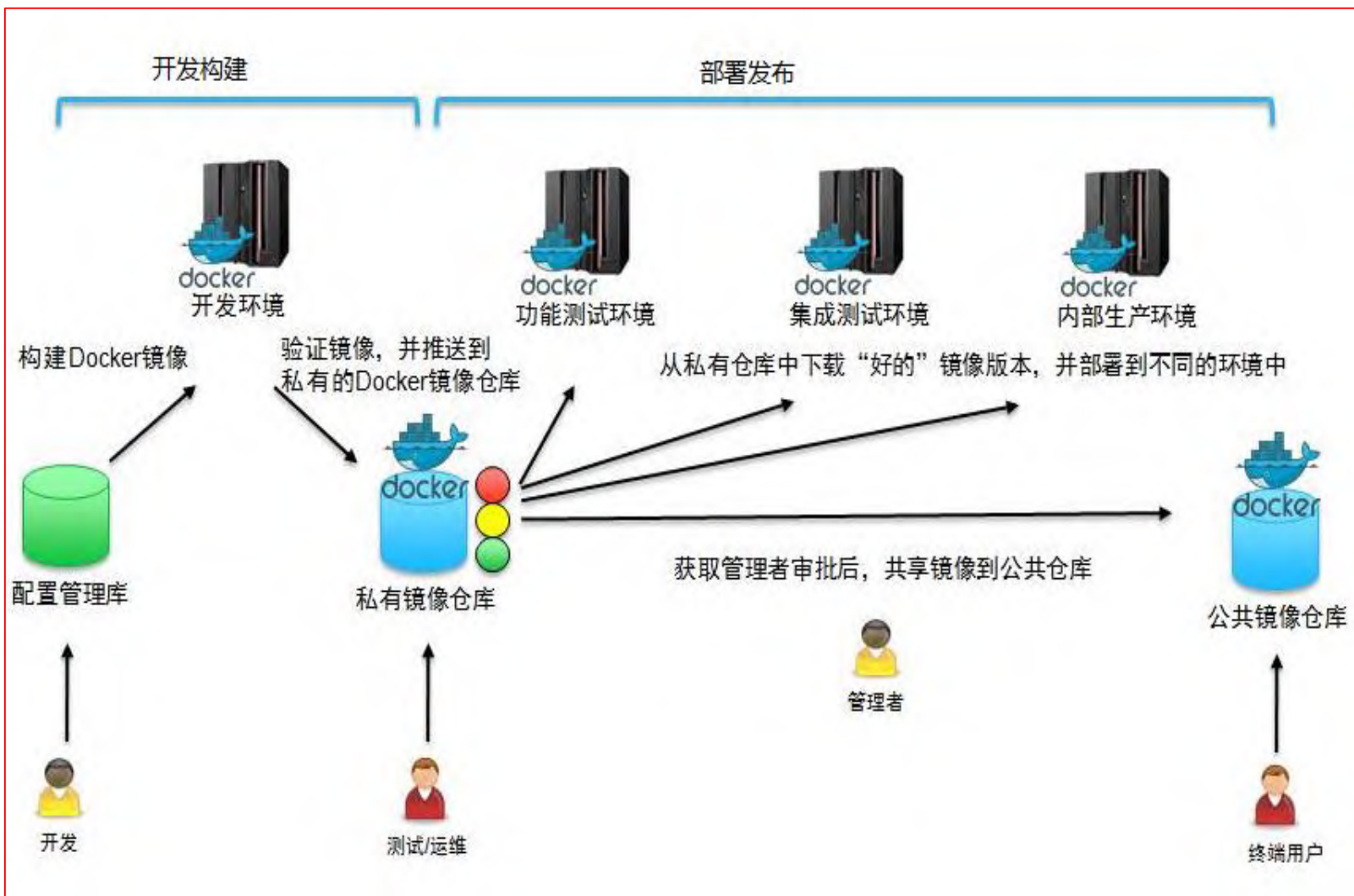
<dependency>
    <groupId>com.asiainfo.crm.apache:activemq-client</groupId>
    <artifactId>activemq-client</artifactId>
    <version>5.10.2</version>
    <scope>test</scope>
</dependency>

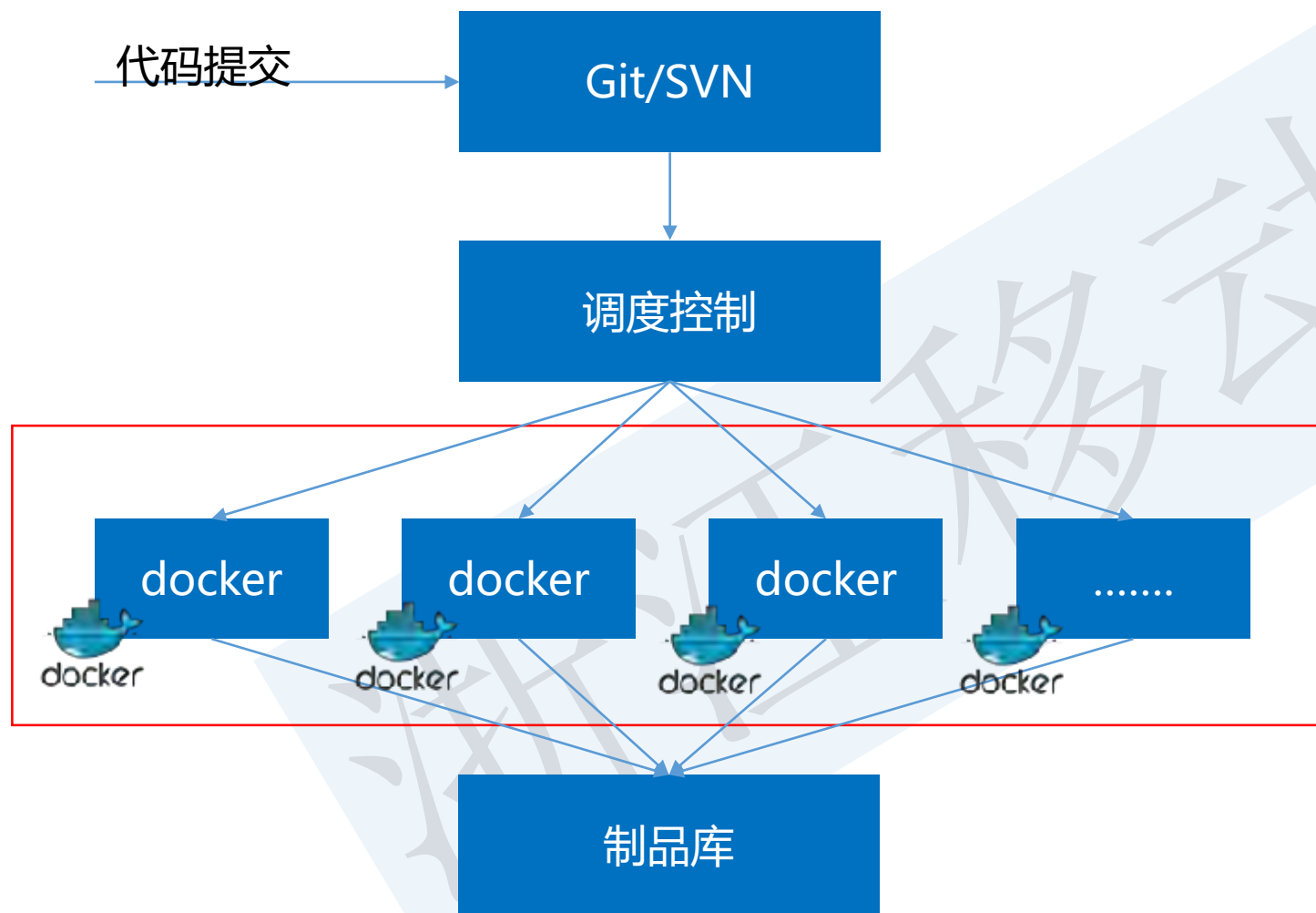
<dependency>
    <groupId>com.asiainfo.crm.appframe:ailogger</groupId>
    <artifactId>ailogger</artifactId>
    <version>1.0</version>
    <scope>test</scope>
</dependency>

<dependency>
    <groupId>com.asiainfo.crm.appframe:aischeduler</groupId>
    <artifactId>aischeduler</artifactId>
    <version>1.0</version>
    <scope>test</scope>
</dependency>

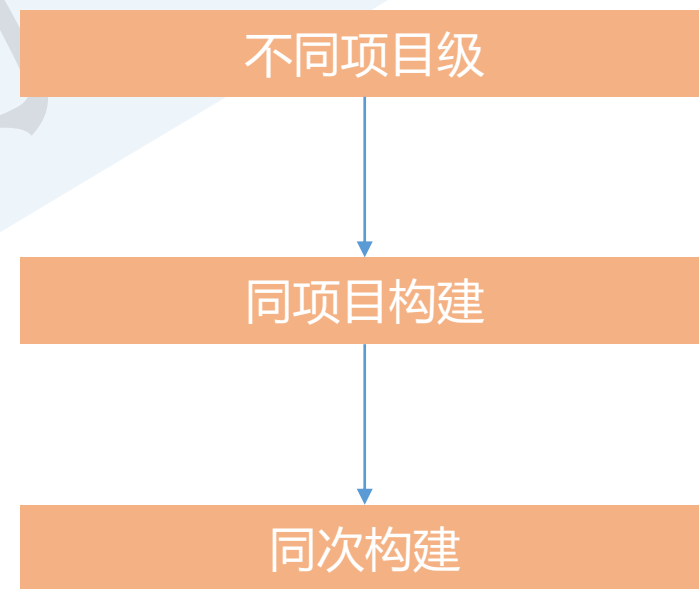
<dependency>
    <groupId>com.asiainfo.crm.appframe:appframe</groupId>
    <artifactId>appframe</artifactId>
    <version>1.3</version>
    <scope>test</scope>
</dependency>
```

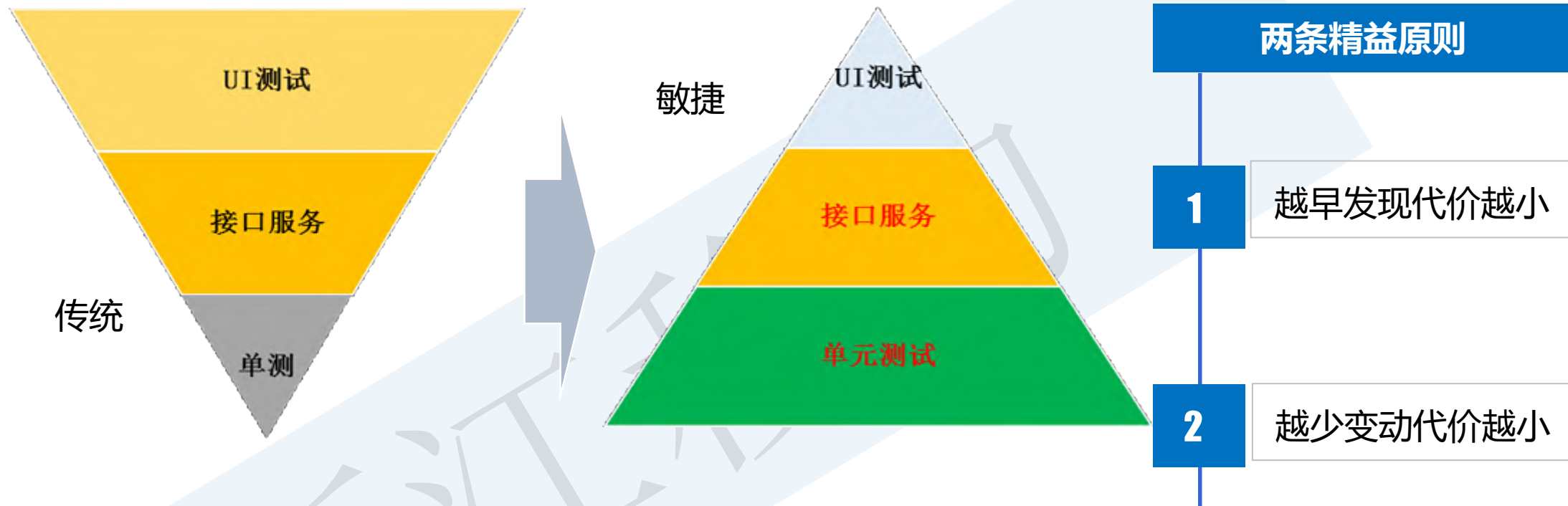
基于DCOS (数据中心操作系统) 的持续部署





通过对编译环境docker化来，
实现代码分布式编译，解决多项目
编译资源争用的问题。





传统企业执行自动化测试困扰：

- 1.大量的自动化用例覆盖在前端UI界面发起的业务，而前面界面变化快，执行失败率高，维护量大；
- 2.接口服务为后端服务，覆盖量不大，而实际上接口自动化实现效率高，稳定性也高，适当高密度覆盖；
- 3.单元测试几乎未覆盖，原因所耗费开发工作量大。

明晰方向，找准位置，清楚差距，确定发展思路。

实践	闭环管理	协同管理	构建管理和持续集成	环境和部署	发布管理和合规	测试	数据管理	配置管理	监控管理
5级-可优化级： 聚焦于流程改进	团队从需求到开发测试、部署、运维所有工具都是在一个平台、可管理、可流转、可追溯。	需求、开发测试、运维团队协作提高流程中的等待和浪费	团队经常碰面讨论集成问题和自动化解决方案，更快速的反馈和更高的透明度	所有环境都被高效地管理起来，自动化地配置管理，虚拟化应用程度高	运维和发布团队协作管理风险和减少周转时间	几乎没有生产回退，缺陷能及时发现和修复	数据库性能和部署过程在两次发布之间构成反馈循环	经常性地检验CM策略能支持高效的协作快速开发和可审计的变更管理流程	系统监控和业务反馈及时，快速形成优化提升。
4级-可量化级： 过程度量和控制	从平台上可以看到整个流程的信息，覆盖面和数据充分	在团队协作的过程中，各个环节的移动平滑，统一的协作过程	收集构建度量信息、数据可视化并采取行动，确保构建不被破坏	各类部署过程和编排都被管理起来，发布和回退过程被充分测试	主动管理环境和应用监控，周转时间被收集记录起来	质量度量和趋势跟踪，非功能需求被定义和度量	每次部署都验证数据库更新和回退，数据库性能被监控和优化	开发人员每天至少签入一次代码到主线，分支只被用于发布	所有的系统都纳入统一的运维监控平台，开发运维通过监控信息提升系统可靠性管理。
3级-可持续级： 过程自动化应用到整个应用生命周期中	各个环节流程管理是闭环，并在平台中管理，环节之间流转具备自动化能力。	团队协同通过契约方式，各个团队之间分工明确，过程可控可追溯	每次变更提交都触发自动化构建和测试，依赖包得以管理，重用脚本和工具	每个环境的部署都共用同一个过程	部署被定义和执行，合规审核条件得以满足	自动化单元和验收测试，后者由测试人员编写，部分开发过程伴有测试活动	作为部署过程的一个环节，数据库变更自动化执行	库和依赖被管理起来，版本控制的使用策略由变更管理流程决定	监控平台能反映出系统当前状态，具备数据分析能力
2级-可重复级： 流程文档化，部分自动化完成	需求、开发测试、运维之间切换有规范，不存在信息遗漏	有明确的规范确定各团队的工作，团队间协同有记录。	自动化构建和测试，任何一次构建都可以使用自动化流程从源代码控制开始重新创建	自动化部署到某些环境，创建一套新环境比较简单快捷，所有配置都版本化外在化	发布过程比较痛苦，但发布过程本身还是比较可靠的，从需求到发布仅有限的可追溯性	部分自动化测试	数据库变更由版本化的自动化脚本执行完成	所有重建软件系统所需的东西都纳入了版本控制：源代码、配置、构建和部署脚本、数据迁移	监控作为常态化运维保障的工作，能在故障发生时发出通知
1级-手工级： 过程不可重复，缺乏控制	流程流转靠人的经验管理，人的能力决定需要交接的内容	没有团队分工，靠人工经验完成事件	纯手工过程构建软件，工件缺乏管理和报告	纯手工过程部署软件系统，二进制工件跟环境相关，环境配置手工进行	不能频繁发布，发布过程不可靠	手工测试	数据迁移手工进行且未版本化	未使用版本控制，或签入代码不频繁	以人工监控位置，自动监控为辅

总体来说处于可持续级，需要往量化级和优化级发展！

阶段划分

阶段一：持续集成

阶段二：持续交付

阶段三： DevOps PaaS云

关键能力

- 持续构建
- 持续检查
- 持续单元测试
- 持续自动化测试
- 持续部署

- 交付流水线
- 仪表盘
- 自动化监控
- 配置管理
- 部署环境管理
- 制品管理
- 全景报表

- 部署即服务
- 监控即服务
- 日志即服务
- 开发测试服务水平管理
- 弹性扩容变更
- 流水线即服务

敏捷效果

- 团队级DevOps
- 实现对包括部分新业务和CRM试点敏捷项目的持续集成能力支撑

- 部门级DevOps
- 实现对信息技术部各个域采用持续交付支撑
- 实现需求、开发测试、运营的一体化平台管理

- 公司级DevOps
- 可以对各个部门应用开放开发测试服务
- 完成DevOps PaaS云平台的支撑落地实施

◆敏捷转型

- 组织转型
- 团队试点
- 文化活动

◆DevOps实践

- 平台建设
- 工作实践

浙江移动

感谢聆听！

身处传统行业，积极变革，踊跃创新。
欢迎加入我们，三墩IT人敏捷俱乐部！

三墩IT人
浙江移动IT技术团队

IT人敏捷工坊
浙江移动敏捷大本营



IT人敏捷工坊



IT人敏捷工坊

