



GOPS2017
Shenzhen



全球运维大会

2017



深圳站

指导单位： 数据中心联盟
Data Center Alliance

主办单位： 高效运维社区
GreenOps Community

 开放运维联盟
Open OPS Alliance



企业如何迈出DevOps第一步

普元·刘相

目录



1

IT精益运营与DevOps

2

对DevOps的认知

3

企业实施DevOps的建设难点

4

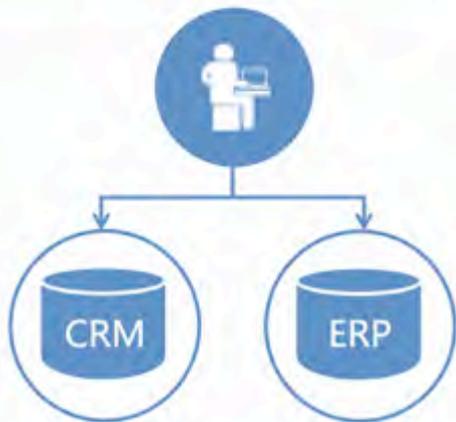
DevOps平台关键设计

5

如何迈出DevOps第一步

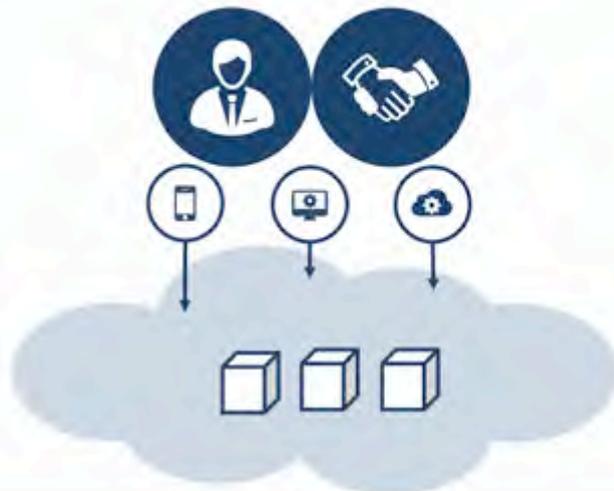
企业数字化要求从对内服务到互联网服务转变

企业数字化要求 IT 从对内服务向内外兼修转变



传统应用

- 服务内部用户为主
- 需求明确、功能全，覆盖广，大集成，中央控制，适合稳定发展阶段
- 刚性强，难以快速变化，维护成本高，快速变革的新业态无法支持



新兴互联网应用

- 服务外部客户和合作伙伴
- 需求变动快，功能简单，独立和分散，分布式进化，一切都从零开始，业务与IT无法分开，需要快速创新
- 应用规模变化大，大范围广泛的尝试，易失败（淘汰），对业务弹性、快速发布要求高

商业场景 航班延误



阻碍企业数字化转型的挑战

失真的信息传递

- 落后的协作手段
- 信息传递形变
- 技术和业务语言不通
- 点对点逐层衰减



落后的协作方式
让IT与客户渐行渐远

未知的技术欠债

- 技术短视
- 选型风险
- 架构能力弱
- 跨界复杂
- 遗留和创新的矛盾



隐形的技术成本让IT行动迟缓
不适应数字化时代快速创新

过高的隐性成本

- 无价值活动
- 重复劳动
- 束缚创造力
- 过程黑盒
- 产品缺陷



低效的生产方式
让知识工作者沦为码农

缺乏架构升级的支持

- 终端移动化
- 部署内容增多
- 依赖关系复杂
- 多次快速部署
- 灰度发布



移动化 微服务 分布式到来面
临的挑战

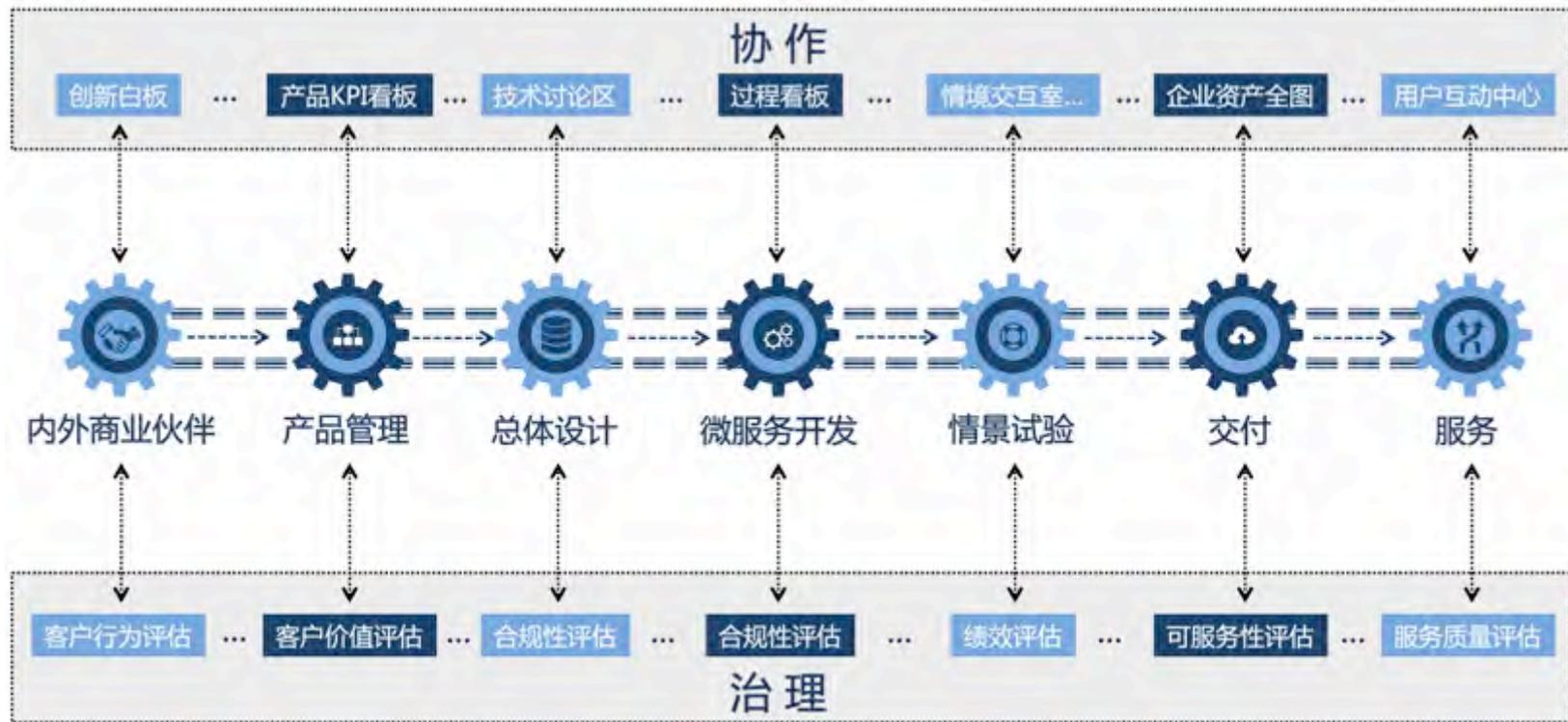
隔离在虚拟社会之外

- 缺乏时效性数据
- 机械的统计报表
- 缺少端到端数据
- 无反馈闭环机制



失控的IT系统
让IT缺少驱动创新的能力

企业数字化转型需要IT精益运营 (DevOps)



目录

- 1 IT精益运营与DevOps
- ➔ 2 对DevOps的认知
- 3 企业实施DevOps的建设难点
- 4 DevOps平台关键设计
- 5 如何迈出DevOps第一步

原因是什么？没有明晰业务价值

提升研发运营效率50%

两周迭代上线

开发测试环境100%自动化部署

整合****等开源工具

.....

某集团DevOps平台业务目标

工程效率提升50%

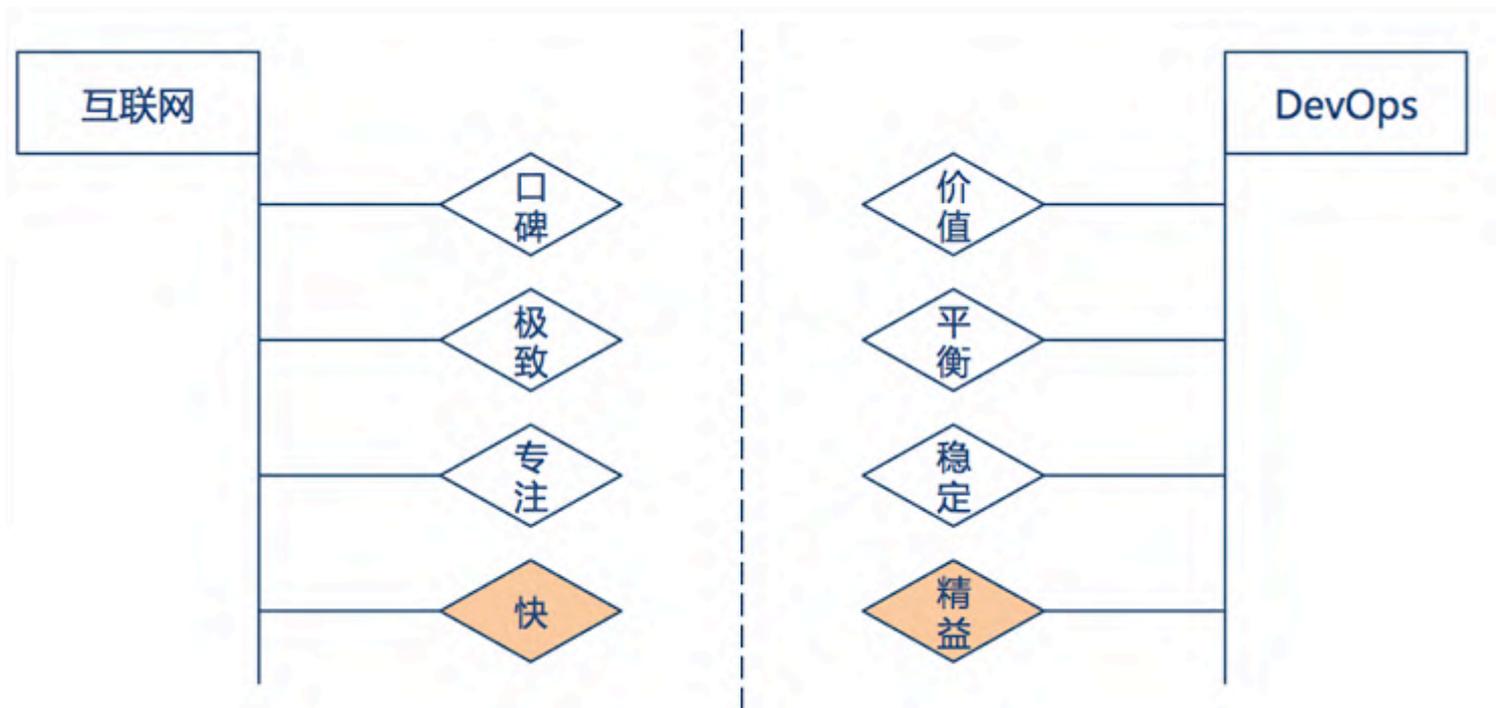
6个月时间

40多套核心产品

周交付

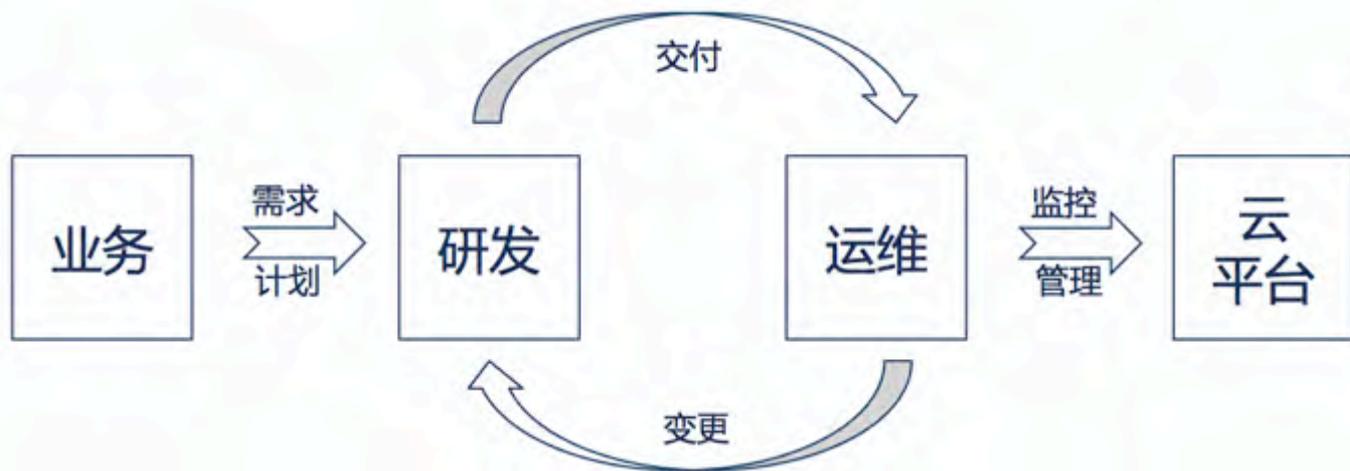
- 验证测试环境和生产环境部署发布的100%自动化执行，最大限度规避人为误操作
- 轻松应对超过日平均TPS访问量20倍以上的瞬间高峰访问压力
- 让产品、项目能统一管理，对团队、对人能数字化考核

企业DevOps思维 VS 互联网思维

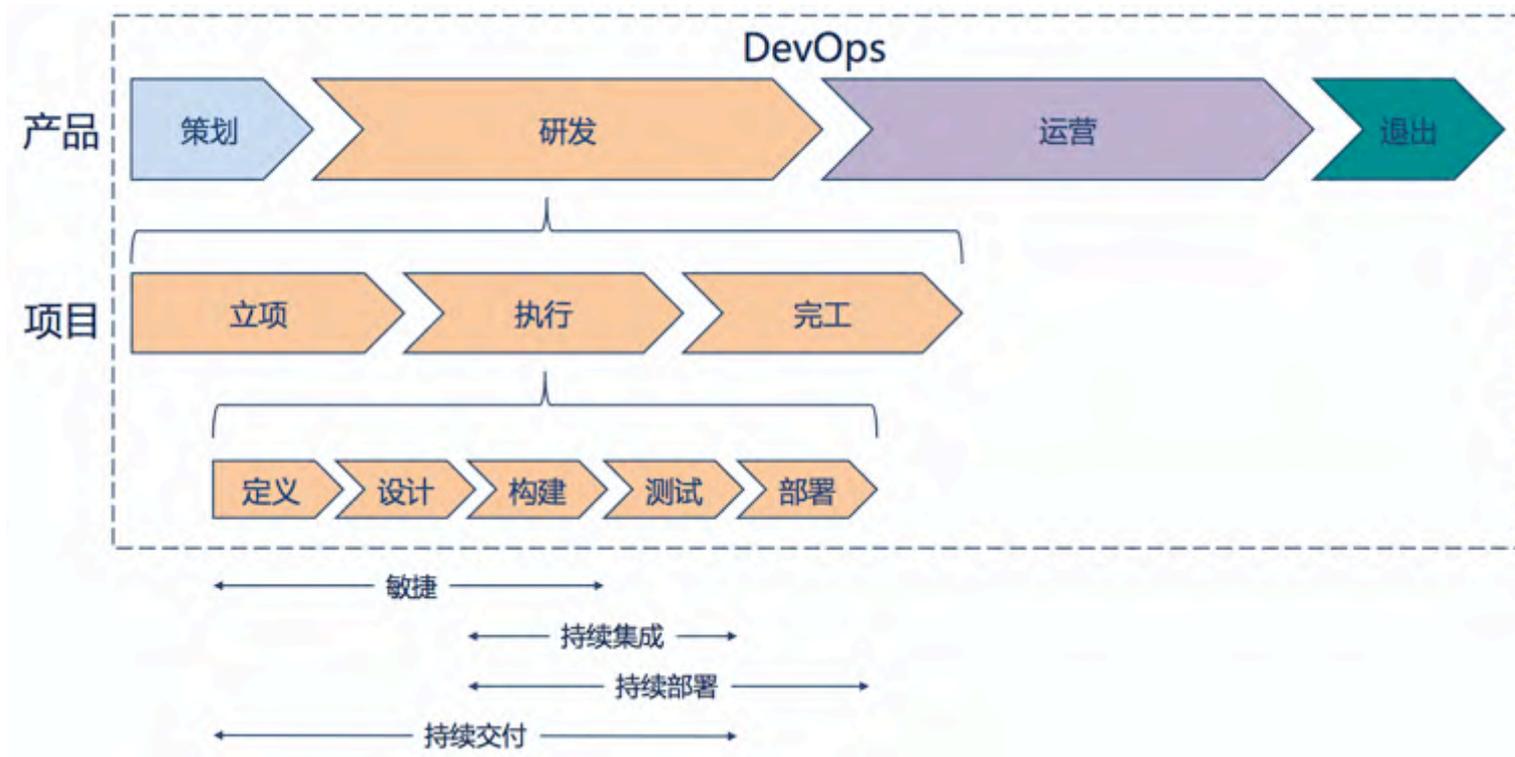


DevOps核心定位—IT交付生产线

支撑全生命周期的『IT生产线』



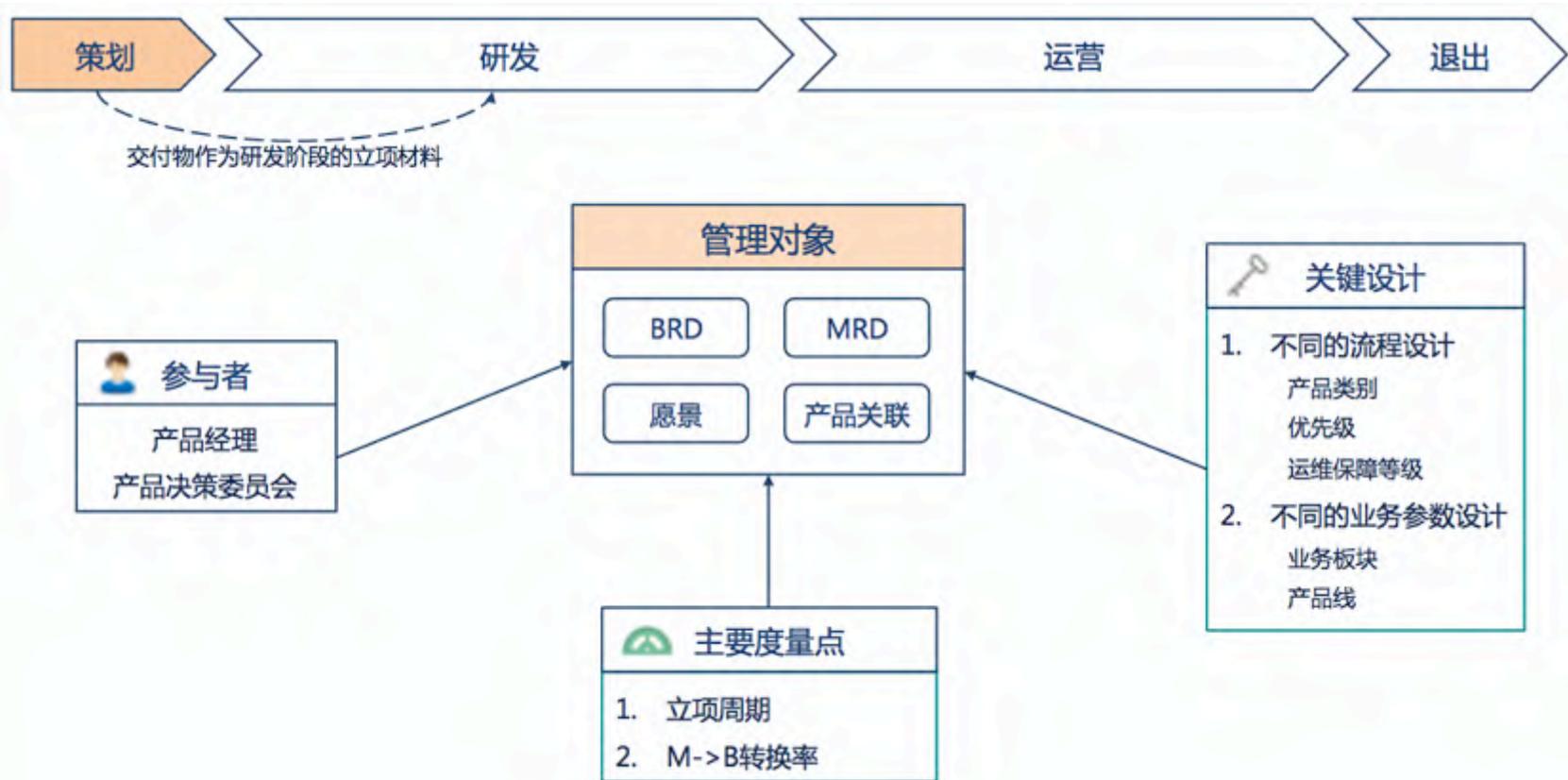
DevOps是对全生命周期的新拆解



核心观点

*DevOps*平台需覆盖产品、项目的全周期

产品策划阶段



产品研发阶段



产品研发阶段—版本号示例

产品版本：1.0

版本别名：shark

修订号：一般引入第3位，
如1.0.1，代表第一次修订

构建号：一般引入第4位

1. 有体现在jar包名称上
2. 有体现在manifest文件中

代码版本（POM）

1. snapshot 1.0.0-snapshot
2. release 1.0.0

发布版本

1. 往往与修订版本一致
2. 也有独立声明的

假设再引入了微服务？

1. API版本
2.

产品研发阶段—版本看板

1、产品规划版本 → 1.0

2、产品修订版本 →

3、产品发布版本 →

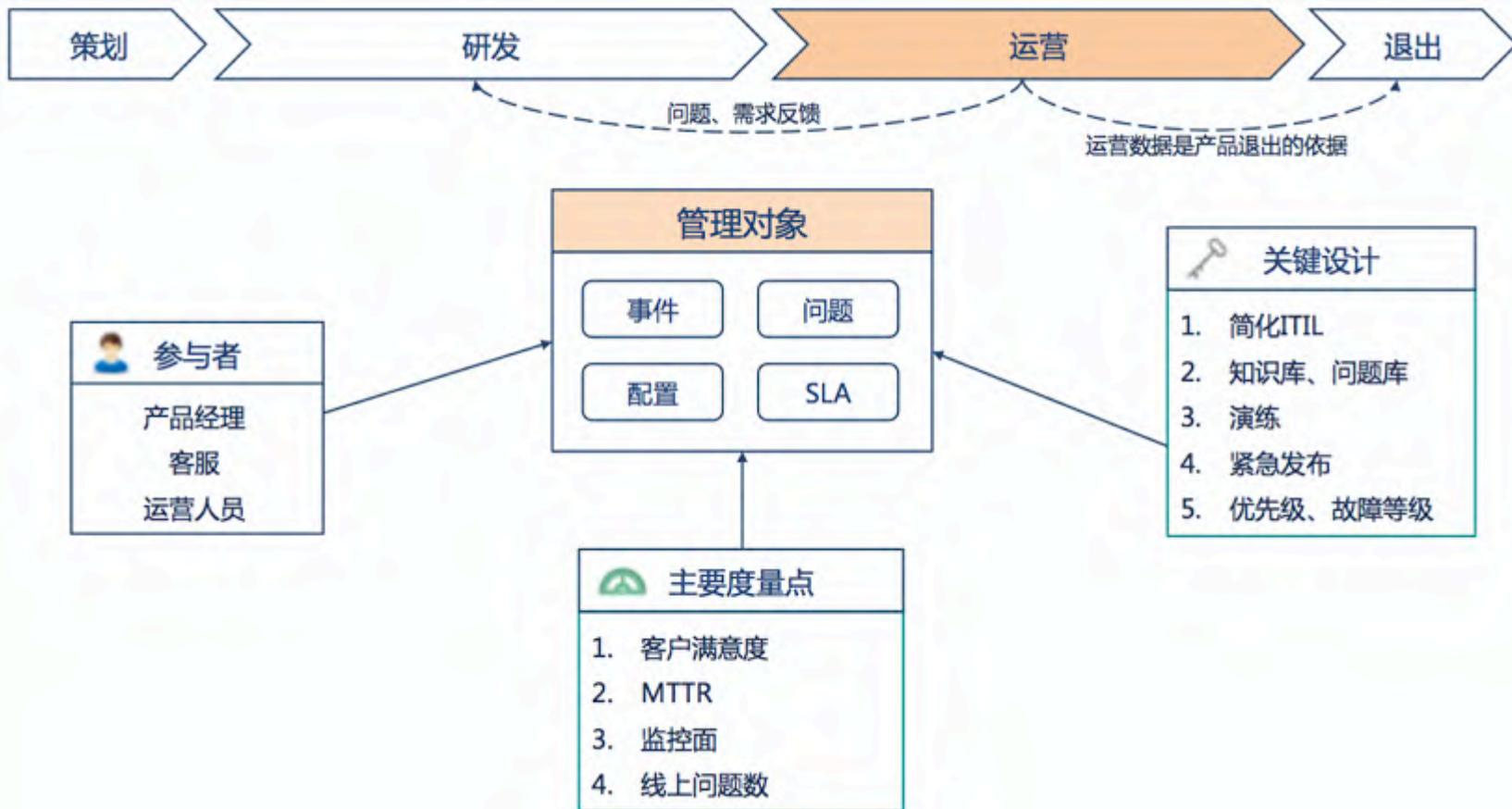
4、内部服务版本 →

发布版本	Sit	Lab	Uat	Staging	Prod	操作
1.0.0						
1.0.0.1	✓通过	✓通过	✗未通过			执行
CMDB-API 1.0.0.041501						
CMDB-Service 1.0.0.041501						
1.0.0.2	✓通过	✓通过	⚙️执行中			执行
CMDB-API 1.0.0.041501						
CMDB-Service 1.0.0.050102						
2.0.0						

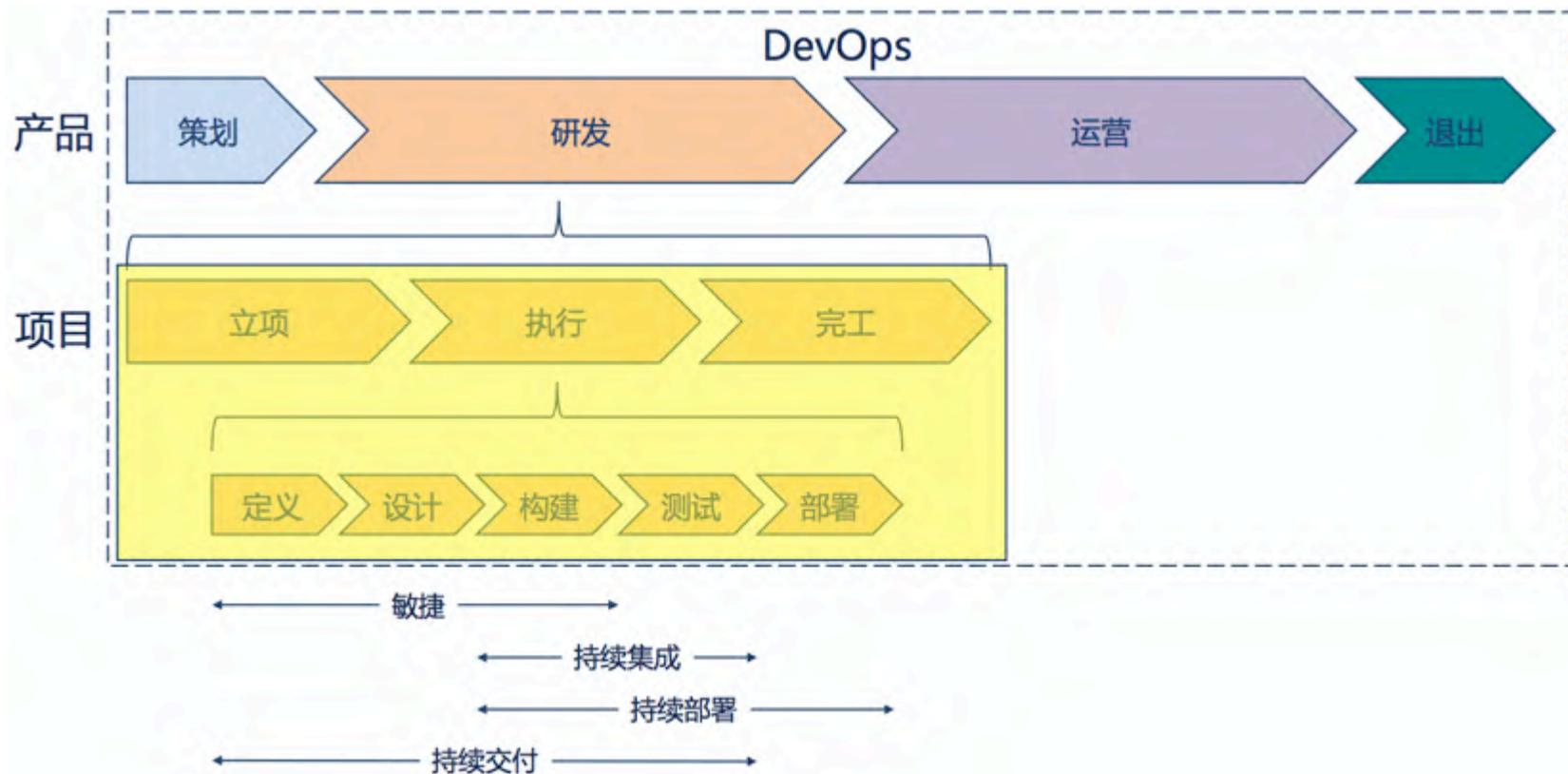
核心观点

*DevOps*平台更重要的是提供**最佳实践**

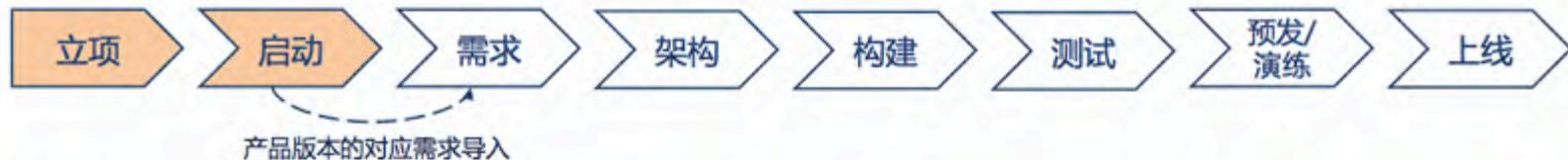
产品运营阶段



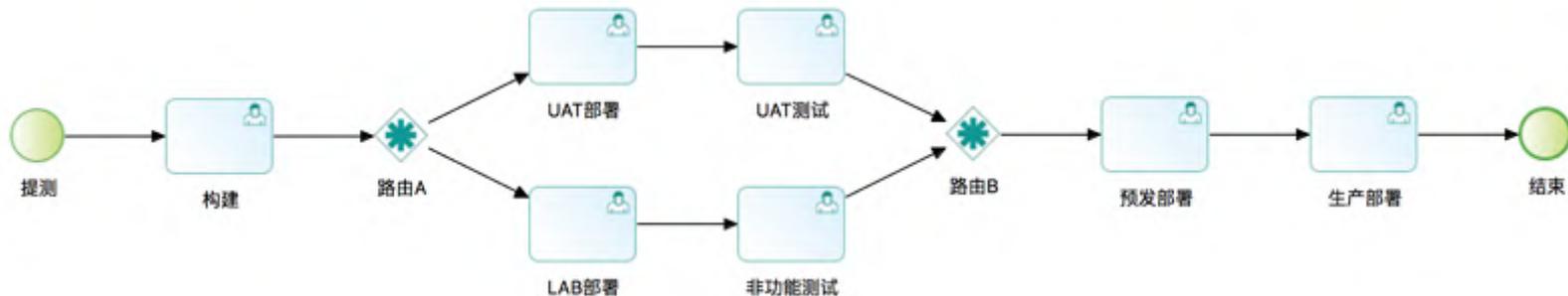
DevOps是对全生命周期的新拆解



项目立项与启动阶段



项目启动阶段—流水线示例



构建 编辑

构建定义: 进入 责任人: 自动执行

增加构建前动作

增加构建后动作

SIT部署 编辑

部署环境: 进入 责任人: 自动执行

部署容器: javaapp-platform mysql-platform nginx-platform

增加部署前动作

增加部署后动作

SIT测试 编辑

测试定义: 进入 责任人: 自动执行

- 关键一：环节必选与可选
- 关键二：自动与人工配合
- 关键三：主数据，buildNumber
- 关键四：参与者权限

活动耗时	自动化执行时长	自动化执行进度	操作
1h	10s	<div style="width: 100%; background-color: #007bff; height: 10px;"></div> 100%	查看
20m	/	<div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0%	
/	/	<div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0%	

核心观点

*DevOps*平台，
让不同角色更好的在流水线上协作



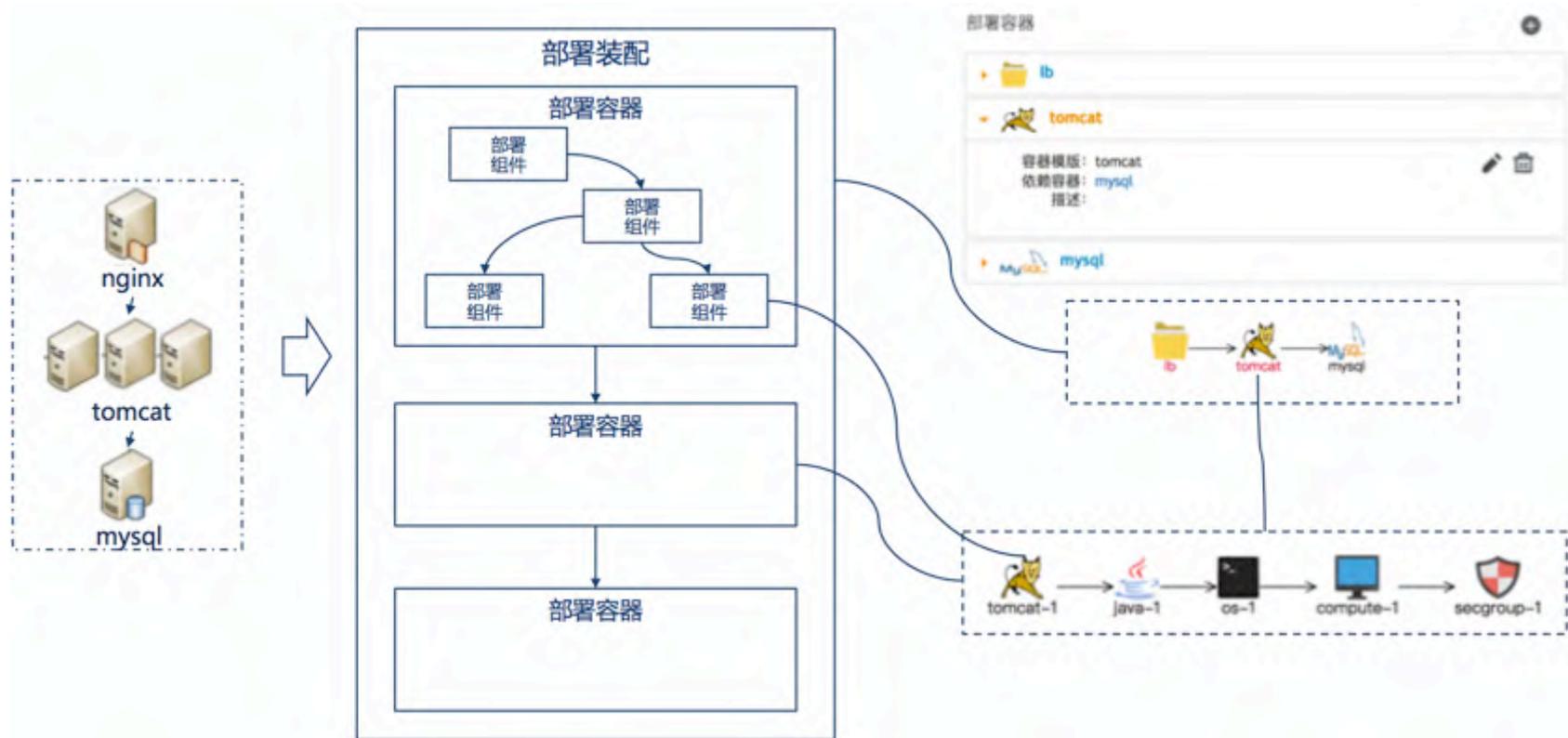
项目需求阶段



项目架构阶段



项目架构阶段-部署设计示例



核心观点

*DevOps*平台，**管理前移**，有效指导和约束
后续工作

项目构建阶段



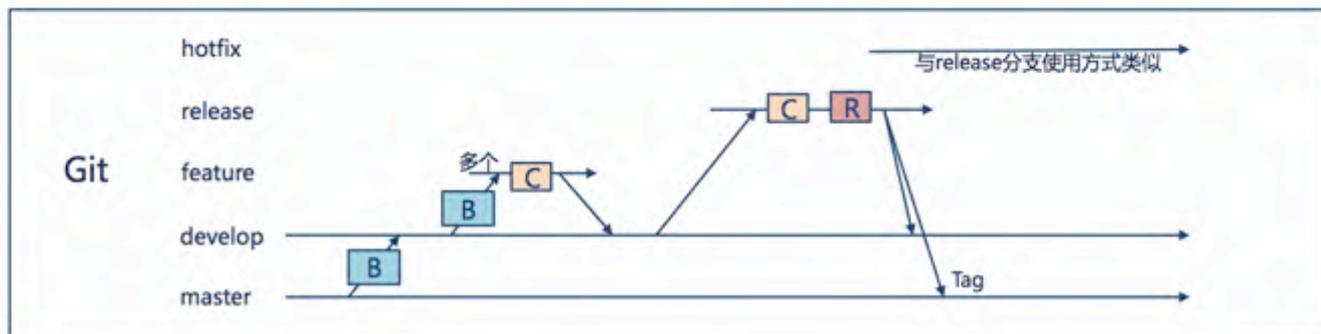
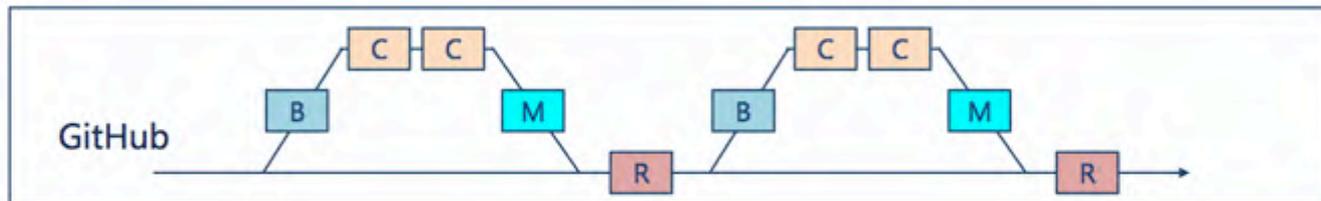
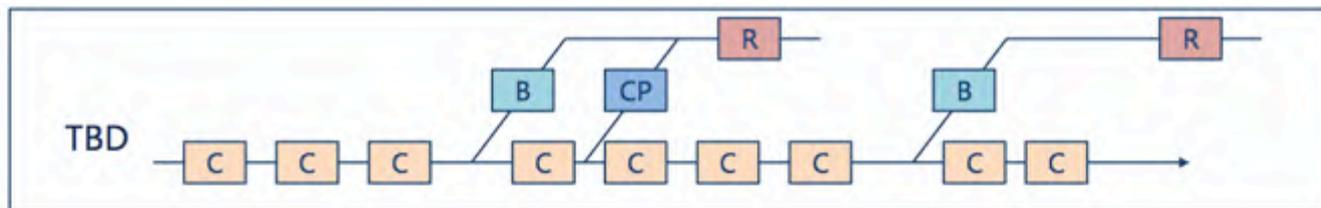
与任务关联，细粒度考核开发、测试人员效率

开发、review等规范保证质量，进入集测阶段的标准



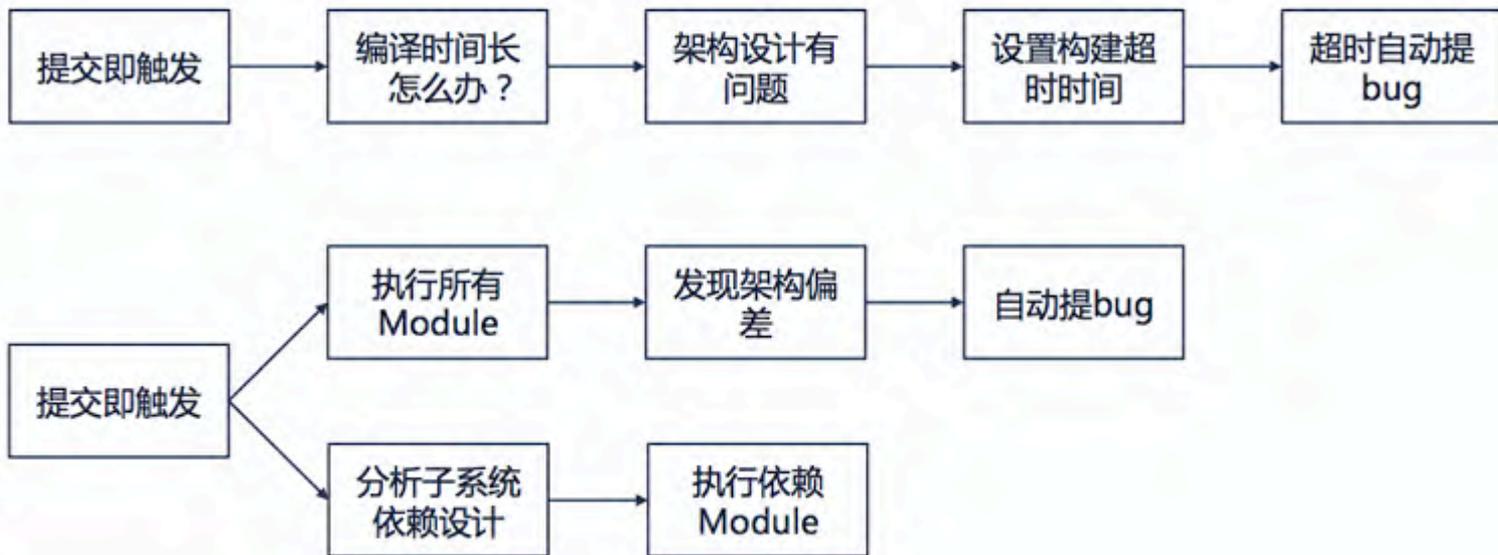
项目构建阶段—代码库使用示例

- C commit
- B branch
- R release
- M merge
- CP cherry-pick



项目构建阶段—构建策略示例

提交即触发，我的一个产品，构建一次要20多分钟，怎么办？



核心观点

对于**已有系统**，DevOps平台不仅仅是
通过新的工具链实现快速交付，
更是一种驱动**优化**的变革

项目测试阶段



集测通过是进入发布流水线的前提
UAT通过是进入预发或上线的前提



项目演练与上线阶段



验证测试成果、缺陷逃逸等



项目上线阶段—滚动升级下的部署转换示例

部署转换：部署设计是逻辑部署架构，在实际部署时，需要结合目标环境类型、部署策略、部署模式等，将其转换为可执行的物理部署架构（执行计划）。



核心观点

*DevOps*平台，并不是自动化一切，
而是在可控中有选择的自动化

对DevOps认知总结

- 一条IT生产线
- 覆盖产品、项目领域
- 体现出最佳实践
- 加强协作
- 优化架构
- 可控范围的自动化

核心价值：提升IT工程效率

目录

1 IT精益运营与DevOps

2 对DevOps的认知

➔ 3 企业实施DevOps的建设难点

4 DevOps平台关键设计

5 如何迈出DevOps第一步

DevOps平台建设难点

业务

一定的
方法论

技术

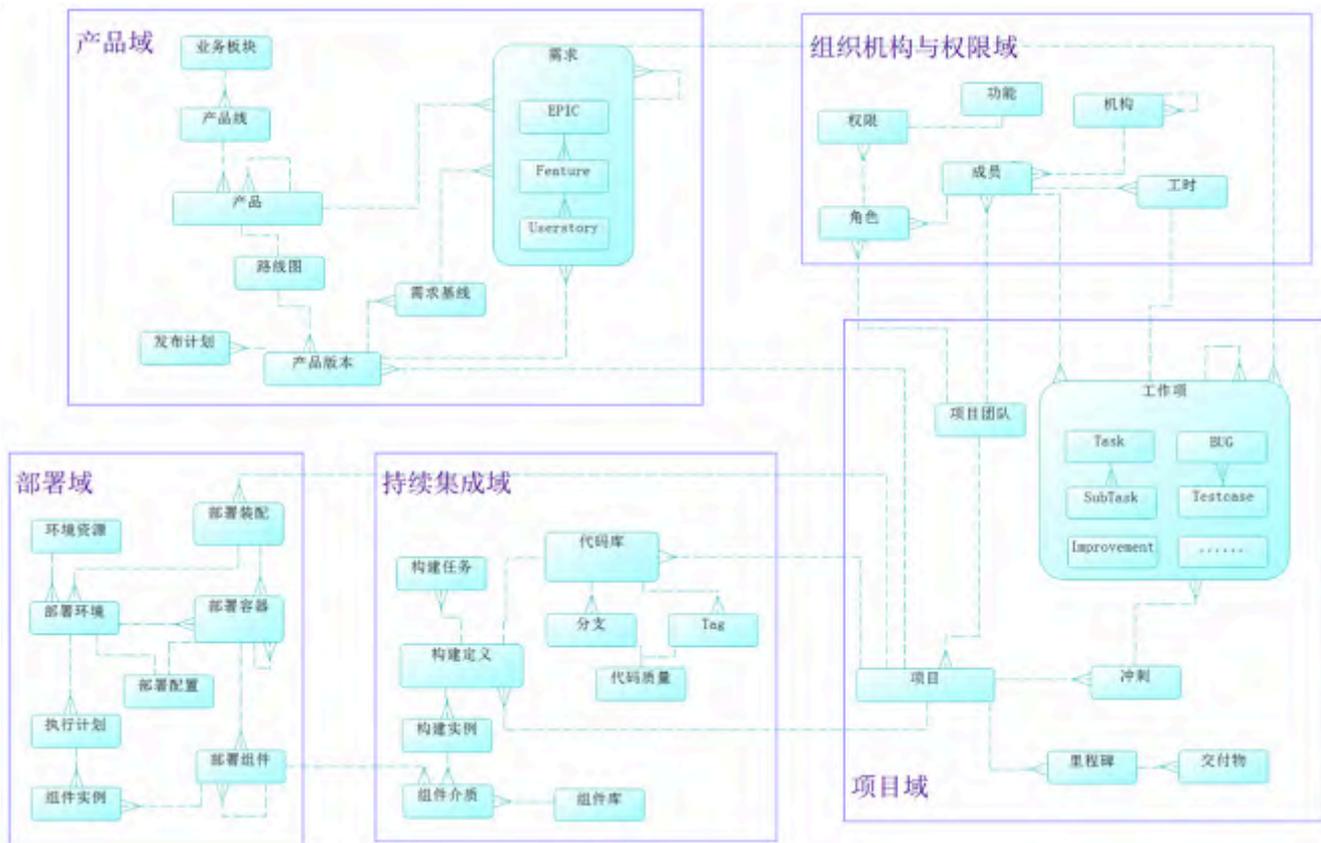
- 我们有很多系统，哪些应该纳入？
- 概念模型上，DevOps怎么划分？
- 无规矩不成方圆，规范该怎么定？
- 度量指标，是否有些方法论的指导？
- DevOps建设过程，是否有些建议？
- 各类工具链怎么集成？注意些什么？
- 微服务架构的必要性与可行性？

业务：切面定义—基础设施举例

1. DevOps无法让每个人的工作都在上面，高级能力还是专人在专业系统上完成；
2. 如果专业系统足够自动和自助化，可考虑逐步纳入DevOps平台
3. DevOps是工程效率平台，不是给多个系统做个统一门户



业务：梳理领域模型



业务：流程 + 标准

产品管理规范

比如，版本号怎么定？需求分几层？

项目管理规范

比如，项目中的issue类型有哪些，一个团队的组成是什么样子？

开发与构建规范

比如，snapshot、staging和release怎么分？主干、分支怎么用？

部署规范

比如，有几套环境？CMDB在哪儿？哪种脚本工具？

迁移规范

比如，现有代码是否需要同步？现有项目数据怎么导入？备份工作怎么做？

质量规范

.....

方法：站在工程效率、工程质量角度定度量

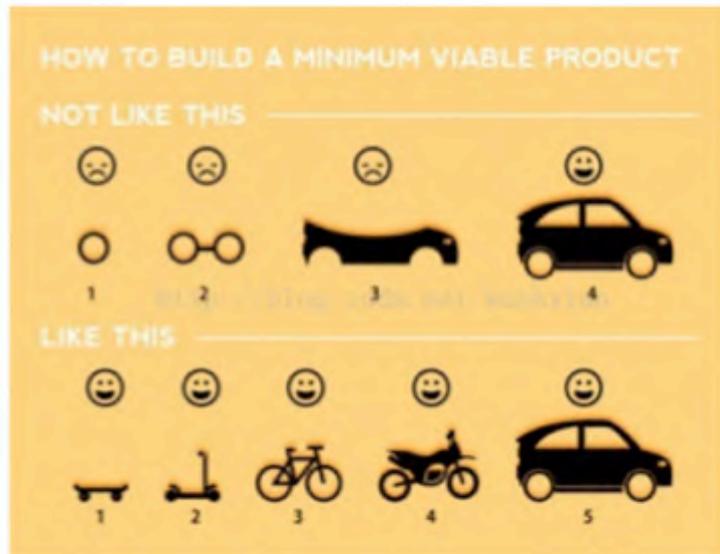


	员工	团队	部门
质量	代码问题数	构建成功率	构建成功率
	严重Bug数	缺陷移除率	部署成功率
	千行Bug率	需求覆盖率	周期内生产事故数量
	问题打回率	文档过审	版本上下线周期
效率	编码效率	构建频率、时长	事故恢复时间
	Bug修复时长	部署时长	周期内产品数
	用例执行时长	需求交付时长	周期内项目数
进度	任务偏差	项目偏差	项目偏差
	任务滞留	需求积压	

方法：可迭代式的思路建设



迭代不是简单的版本更新

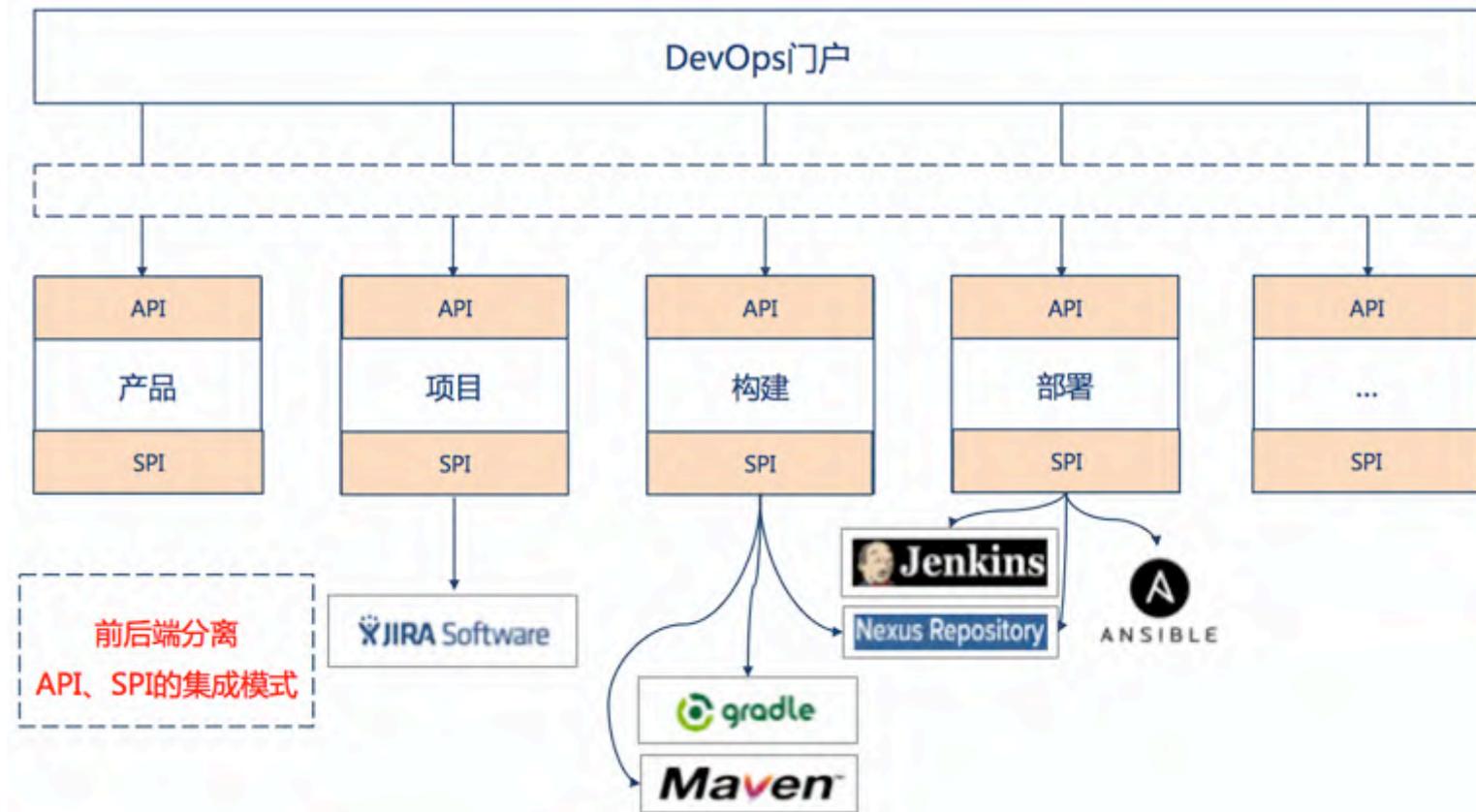


产品功能有浪费，但更为市场接受

技术：工具链的统一集成模式



技术：微服务切分



目录

1 IT精益运营与DevOps

2 对DevOps的认知

3 企业实施DevOps的建设难点

➔ 4 DevOps平台关键设计

5 如何迈出DevOps第一步

DevOps平台功能



关键设计：支持多套开发架构

后端服务组件(standalone jar)

```
/
|--- app
|   |--- conf
|       |--- autoconfig.properties
|       |--- application.yaml
|   |--- bin
|       |--- entrypoint.sh
|       |--- autoconfig.sh
|   |--- lib
|       |--- application.jar
|   |--- data
|--- jdk
```

前端服务组件(html)

```
/
|--- app
|   |--- conf
|       |--- autoconfig.properties
|       |--- nginx.conf
|       |--- conf.d
|       |--- *.conf
|   |--- bin
|       |--- entrypoint.sh
|       |--- autoconfig.sh
|   |--- html
|       |--- *.html
|       |--- images
|       |--- *.png
|   |--- data
|--- usr
|   |--- lib
|       |--- nginx
```

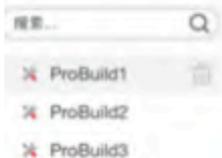
JEE组件

```
/
|--- app (tomcat)
|   |--- conf
|       |--- autoconfig.properties
|       |--- *.xml
|   |--- bin
|       |--- entrypoint.sh
|       |--- autoconfig.sh
|   |--- webapps
|       |--- ROOT
|       |--- WEB-INF
|   |--- data
|--- jdk
```

关键设计：编排式的构建定义和可视化执行

在项目中创建构建定义

构建定义



选择模板

- ① JavaEE
- ② Android
- ③ Html
- ④

增加多个构建任务（步骤）

任务管理



设置触发规则，如日构建、每次变更；
对编译历史设置保留策略；
对一些公共变量可设置为公用参数；

触发规则



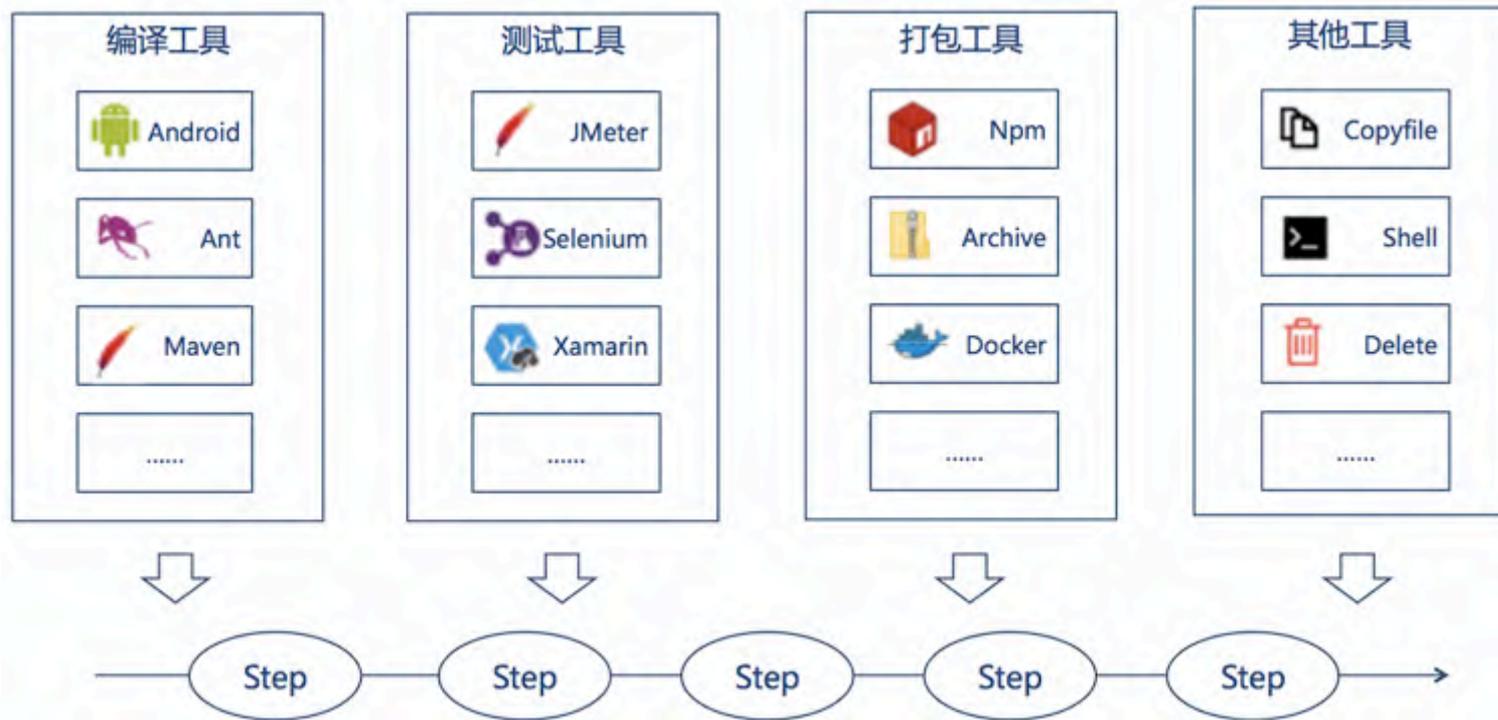
保留策略



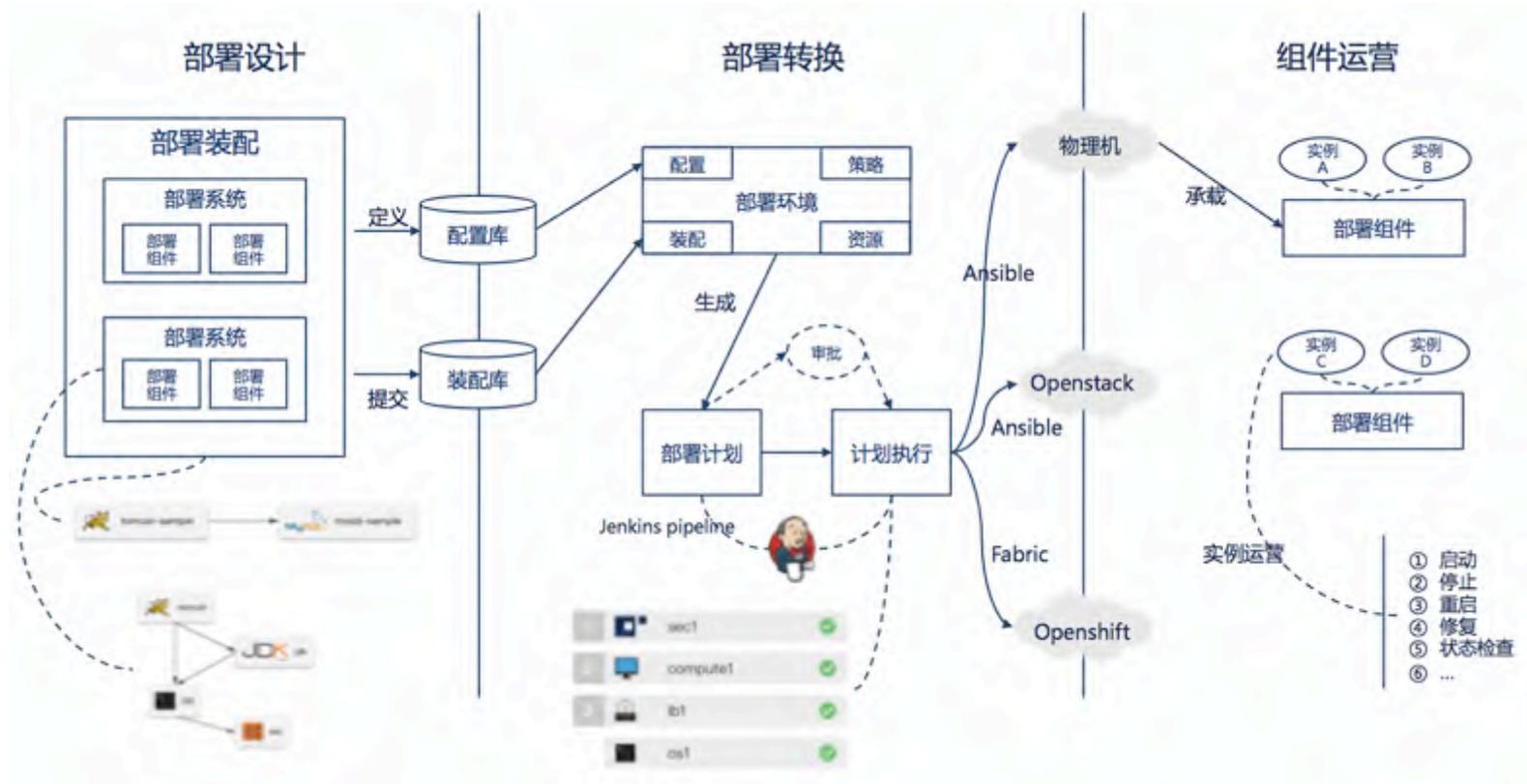
参数设置



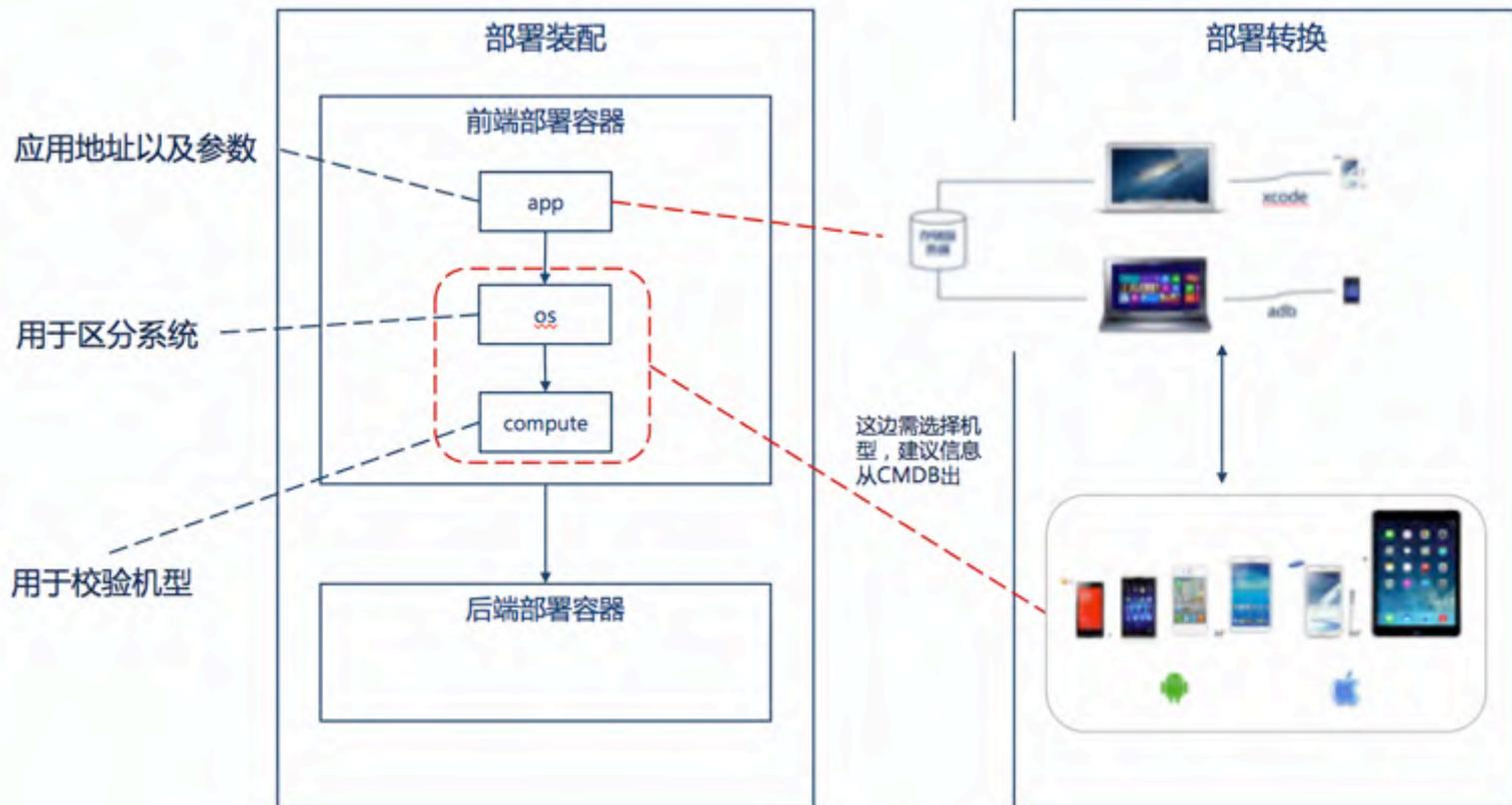
关键设计：持续集成-原子步骤编排



关键设计：支持异构设施的多策略发布



关键设计：兼容多种部署架构示例（移动应用）



关键设计：打上企业生产线最佳实践的标签

🔑 日报与周报

1. 与分配任务关联，工时对应check
2. 基于模板自动出个人及项目周报

🔑 沟通集成

1. 风险、变更、失败等自动通知
2. RTX、Slack等

🔑 产品与项目

1. 产品类型、优先级、运维保障等级
2. 项目类型、项目类别驱动交付流程

🔑 版本管理

1. 串接产品版本、构建版本、修订版本、快照版本、发布版本等

🔑 模板

1. 内置任务模板、bug模板、用例模板等
2. 自定义构建任务模板、部署架构模板、执行计划模板等

🔑 其他

1. 各类应用架构支撑
2. 跨中心基础设置支撑
3. 高效的团队组成（如铁三角等）

目录

1 IT精益运营与DevOps

2 对DevOps的认知

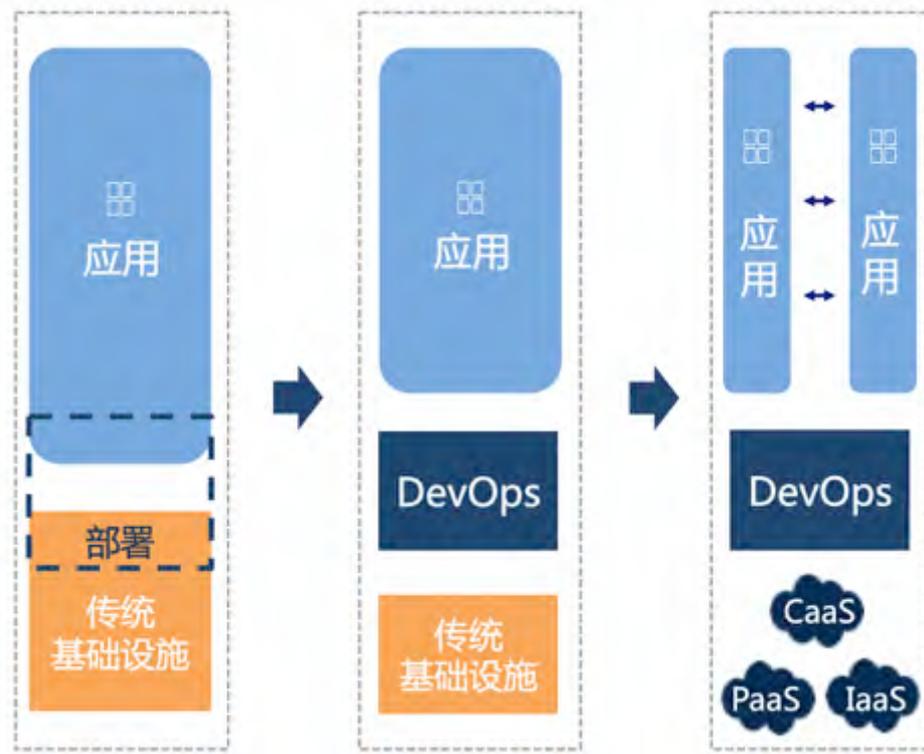
3 企业实施DevOps的建设难点

4 DevOps平台关键设计

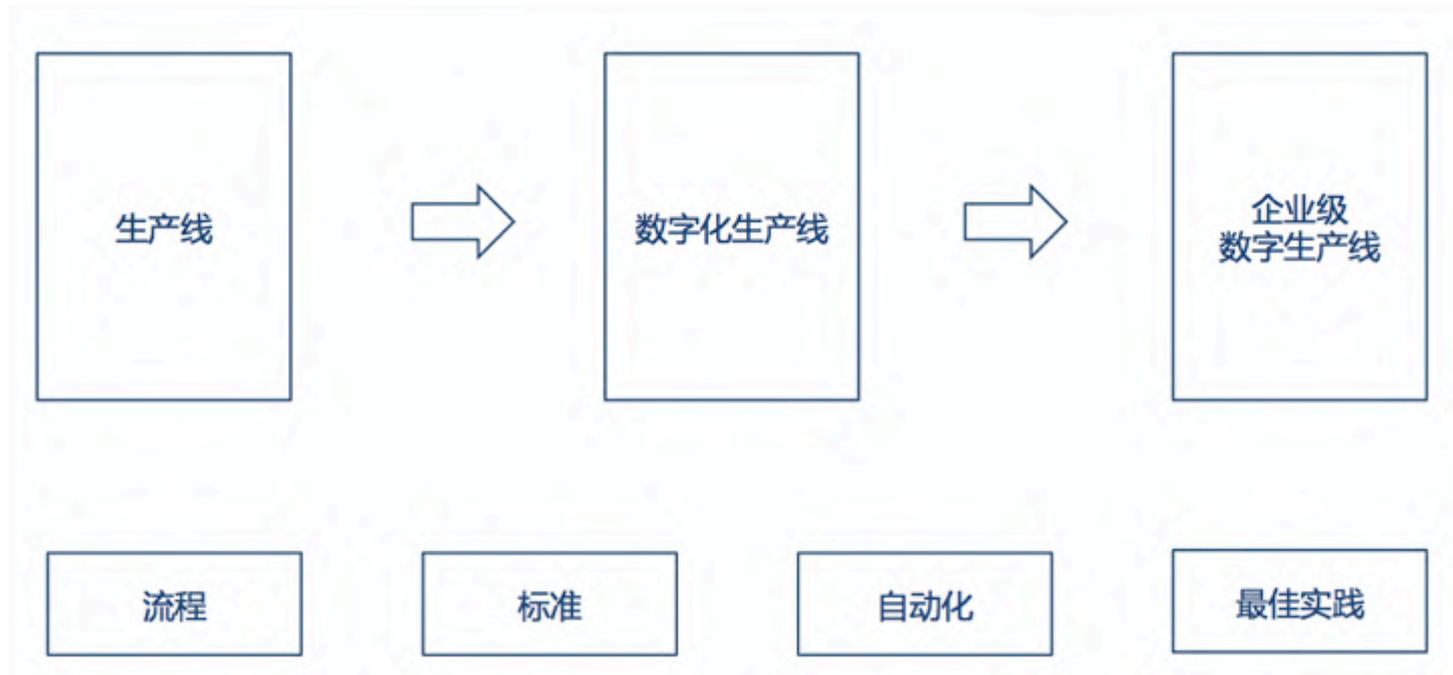
➔ 5 如何迈出DevOps第一步

实施DevOps从三方面开始

1. 统一组织架构、认证、SSO
1. 持续集成与一定的自动化部署
2. 企业研发与运维的过程梳理



总结：DevOps是IT生产线、提升IT工程效率



普元云计算研发开放共享

EAll公众号





高效运维社区
GreatOPS Community

会议

- 3月18日 DevOpsDays 北京
- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 4月21日 GOPS深圳
- 11月17日 DevOps金融上海

培训

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训

咨询

- 企业DevOps 实践咨询
- 企业运维咨询



商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com





Thanks

高效运维社区
开放运维联盟

荣誉出品



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

