

# 中国DevOps社区峰会 2024 · 上海

10.19 | 上海龙之梦大酒店



# 目录

1 前言

2 持续交付X.0

3 工业DevOps建设挑战及实践

4 Q&A





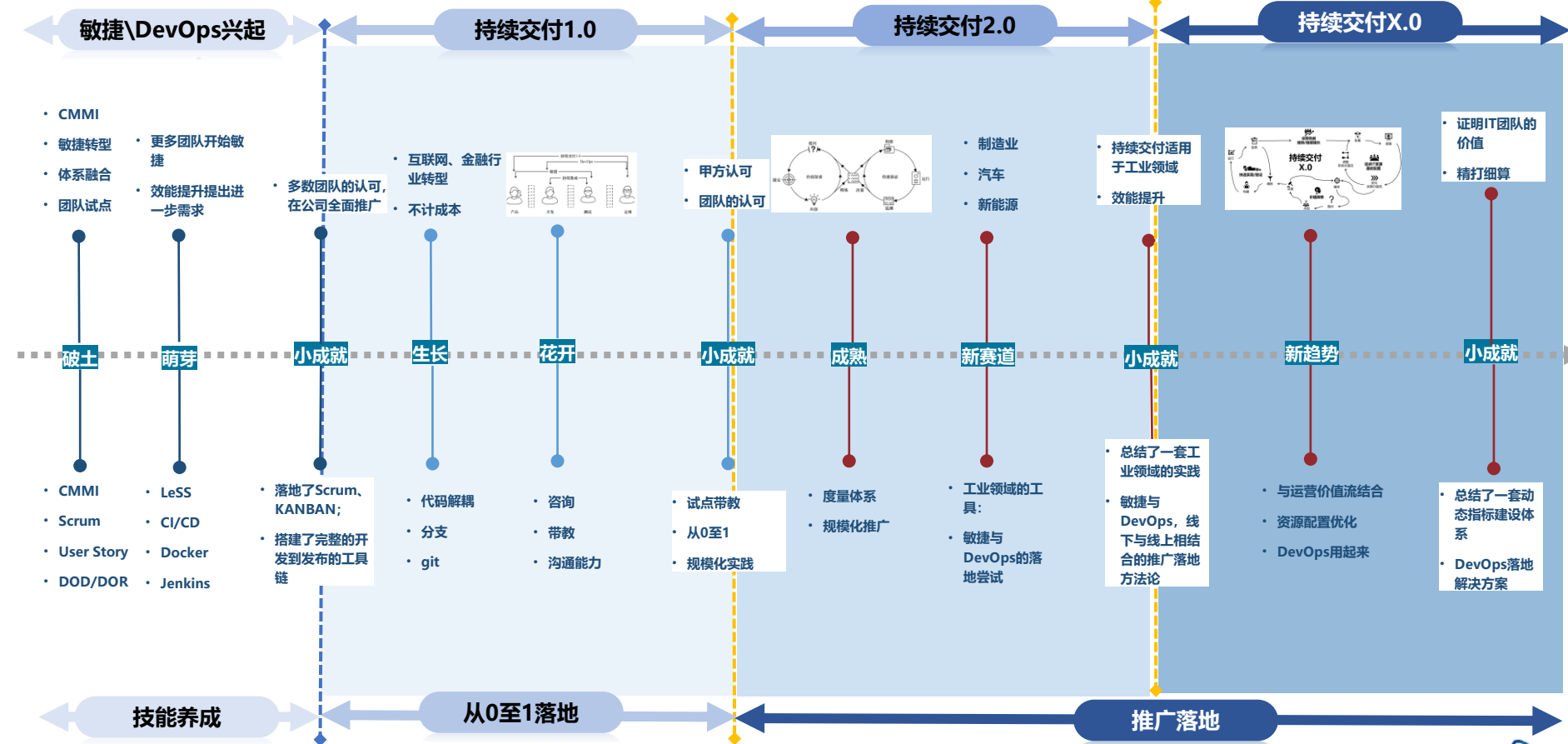
## 简介:

- 14+敏捷相关经验, 资深咨询师
- 覆盖业务敏捷、团队敏捷、DevOps
- 精通SAFe、Scrum、KANBAN、Design thinking、LeanUX、LeanStartup等
- 熟悉金融业、制造业、游戏行业的业务模式及IT结构
- 作为专家级咨询顾问主导大型机构敏捷转型、DevOps建设

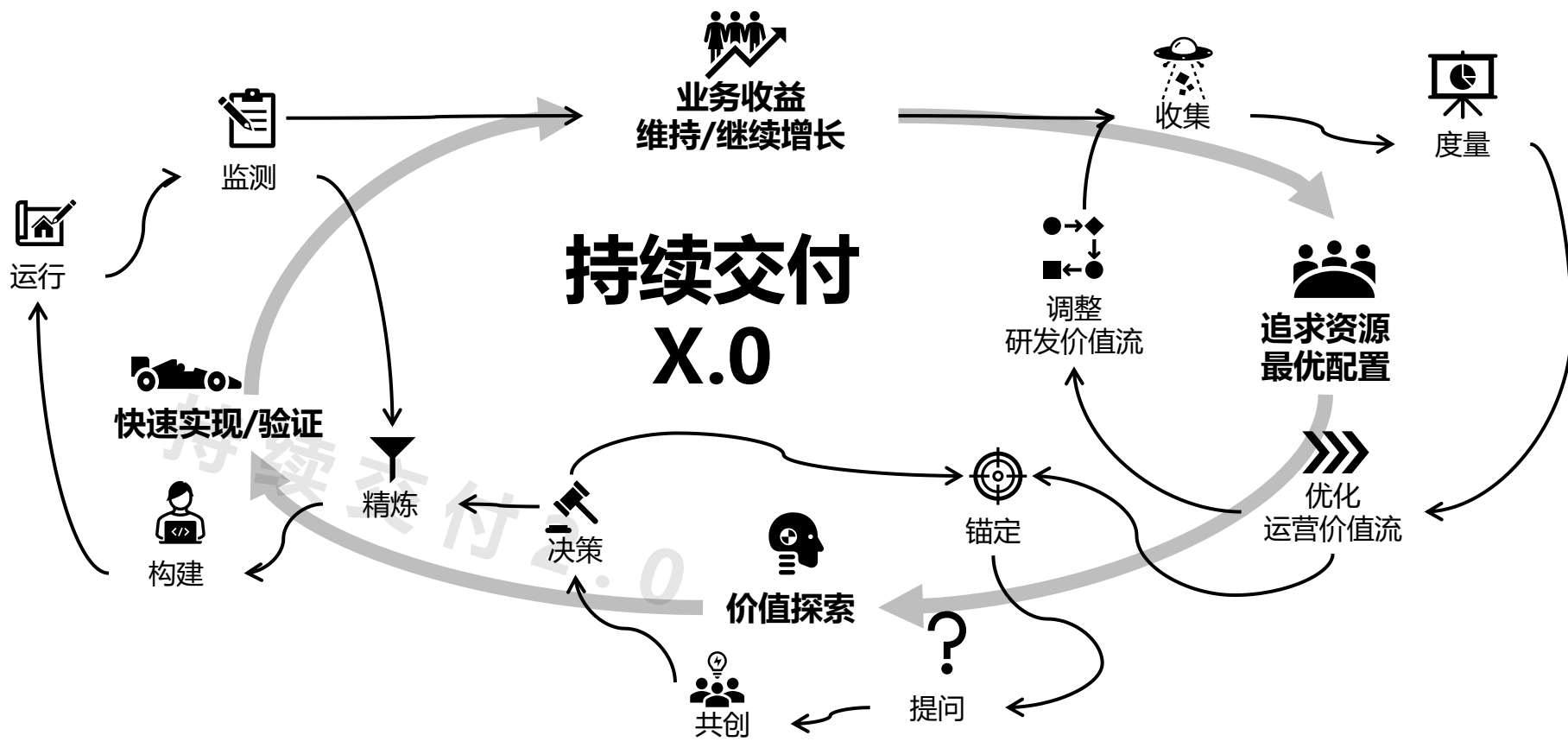
## 咨询与实施案例 (汽车):



# 前言：DevOps发展趋势



持续交付X.0：企业增长飞轮





# • 所谓工业



第一次工业革命  
蒸汽技术革命

18世纪60年代-19世纪40年代



第二次工业革命  
电力技术革命

19世纪60年代后  
期



第三次工业革命  
计算机及信息技术、  
新能源技术等革命

20世纪四五十年代开  
始



第四次工业革命  
人工智能、虚拟现实、  
量子通信等

21世纪-22世纪中  
叶



- 所谓工业





# 造车4.0



## 第一次革命

### 机械化生产-蒸汽动力

对应于工业1.0时代。在这个阶段，汽车制造主要依赖机械化生产，通过蒸汽动力驱动机器取代人力，从而实现了手工业从农业中的分离，正式进化为工业。



## 第二次革命

### 电气化生产-燃油动力

对应于工业2.0时代。在这个阶段，电力开始广泛应用于汽车制造中，取代了蒸汽动力，使得零部件生产与产品装配实现了分工，汽车工业也因此进入了大规模生产的时代。



## 第三次革命

### 自动化生产-电池动力

在这个阶段，自动化技术开始被引入到汽车生产中，大大提高了生产效率和制造精度。机器人和自动化设备取代了部分人力，使得汽车制造更加高效和质量可控。



## 第四次革命

### 规模化自动生产-智能化

在这个阶段，互联网化和智能化成为了汽车发展的主要趋势。通过引入物联网、大数据、人工智能等先进技术，汽车制造实现了高度数字化和智能化。







## • 小米汽车



# 汽车



## 车外系统泛互联网类业务

### 互联网营销与品牌管理:

汽车厂商通过监测互联网上的媒体动态，了解消费者对汽车产品的关注点和需求，以优化营销策略。

### 互联网金融服务:

包括汽车贷款、保险、融资租赁等金融服务，通过互联网平台实现线上申请、审批和放款，提高服务效率和用户体验。

### 智慧出行与共享服务:

如网约车、共享单车等，通过互联网平台实现车辆资源的共享和优化配置，满足用户多样化的出行需求。



## 车内应用层软件开发: 主要涉及车辆的高级软件部分

### 车联网 (V2X) 系统:

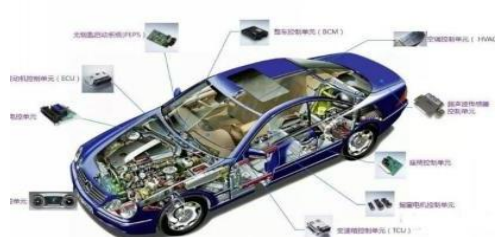
传统软件开发还可以用于开发车联网系统，这些系统通常基于互联网协议和标准，可以实现车辆与周围环境和和其他车辆的通信和协作。

### 车载娱乐系统:

传统软件开发可以用于开发车载娱乐系统，如音频播放器、导航系统等。这些系统通常基于通用的操作系统和开发工具，并与用户进行交互。

### 智能驾驶辅助系统:

传统软件开发也可以用于开发智能驾驶辅助系统，如碰撞预警、自动泊车等。这些系统通常基于机器学习和人工智能技术，可以进一步提高车辆的安全性和舒适性



## 嵌入式开发: 主要涉及车辆的硬件和底层软件部分

### 实时操作系统 (RTOS)

RTOS用于管理嵌入式系统的任务和资源，以确保系统的实时性和可靠性。在汽车领域，RTOS通常用于控制车辆的各种功能和系统

### 传感器和执行器

传感器用于监测车辆的状态和环境，而执行器则用于控制车辆的各种动作。嵌入式开发需要编写软件来驱动这些传感器和执行器，并确保它们能够正确地工作。

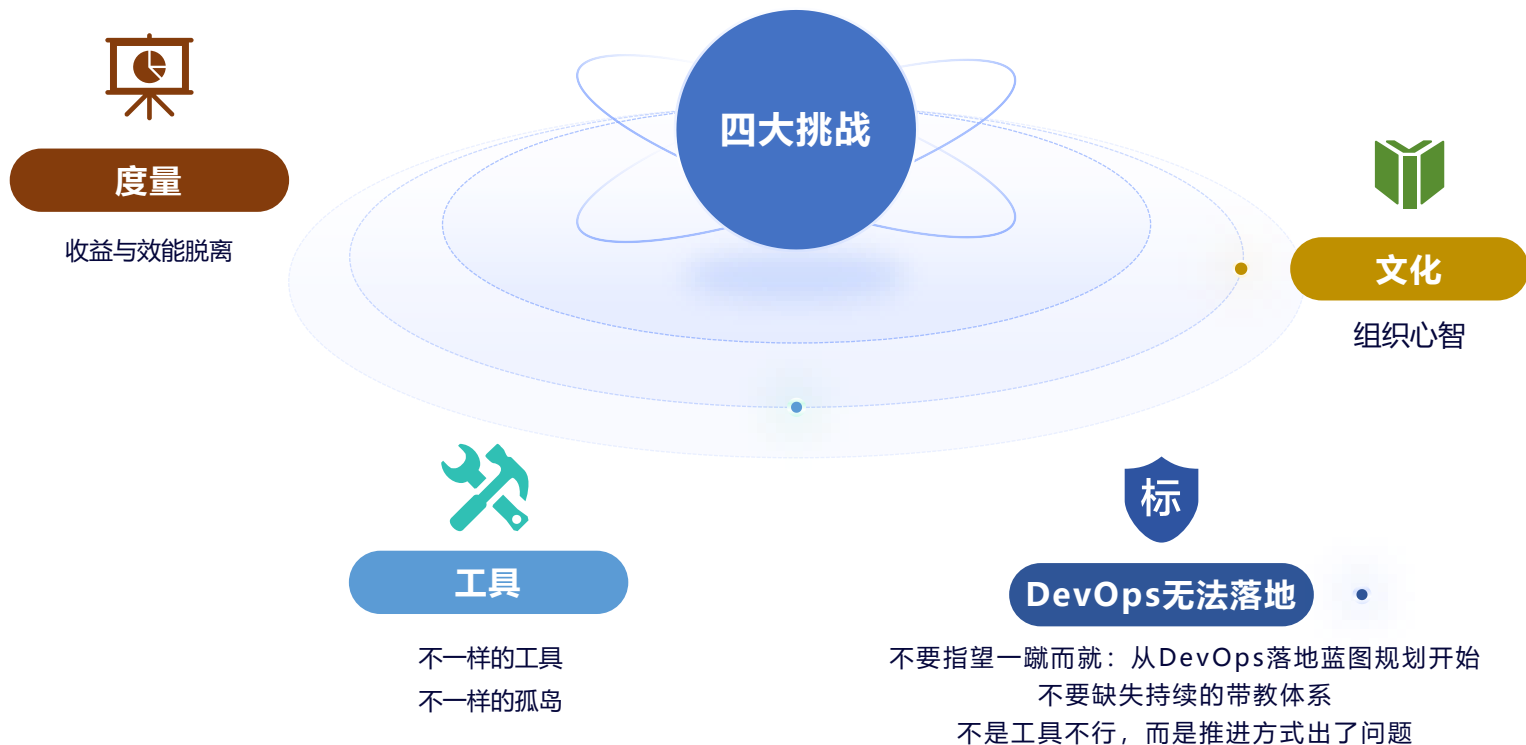
### 汽车电子控制单元 (ECU)

ECU是汽车的大脑，负责控制车辆的各种功能和系统，如发动机控制、悬挂系统、刹车系统等。嵌入式开发需要对硬件进行设计和编程，以确保ECU能够正确地控制这些系统。





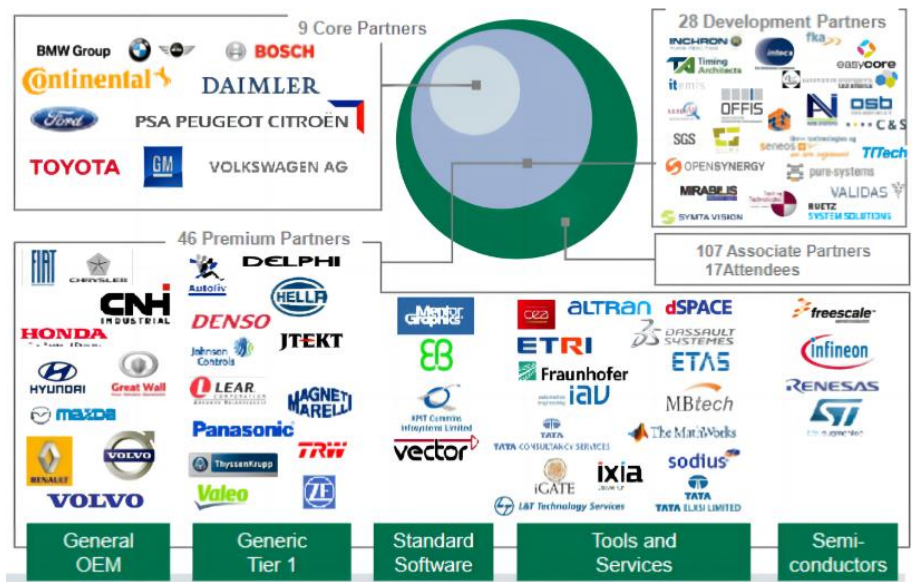
# 工业领域DevOps转型挑战及实践





## 工具孤岛：工业化的工具繁多

基于AUTOSAR标准的工具数量是相当多的，但具体数量可能因不同的供应商、开发商和应用领域而有所变化。一些典型的基于AUTOSAR标准的工具包括**Vector CANoe**、**ETAS INCA**、**dSPACE ControlDesk**等





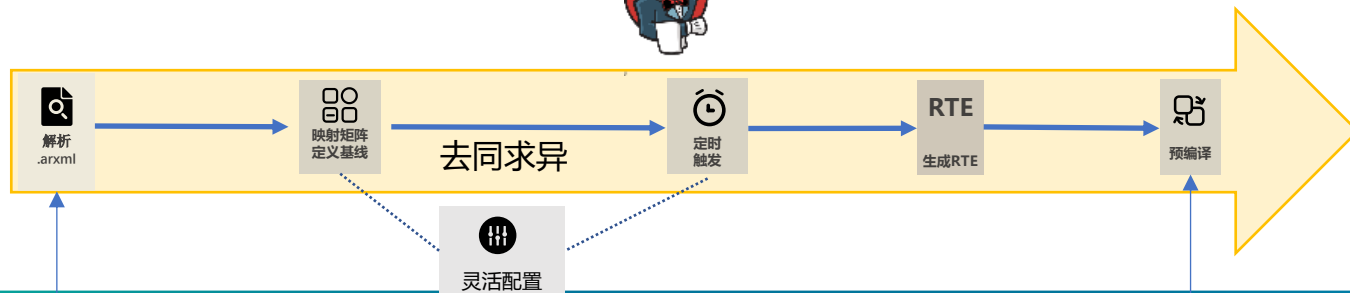
# 工具挑战：落地过程——局部自动化



1.0: 用最快的速度获取反馈, 即使失败, 也能快速发现并定位问题。

自动:

- 1、分析ISOLAR-AB 中的.arxml文件存放路径, 并添加到配置库管理工具中管理
  - 2、建立组织级流水线, 从配置文件修改即可触发至预编译的job, 提交即触发
- 手工: 连线配置



2.0: RTE环境配置自动化。

自动:

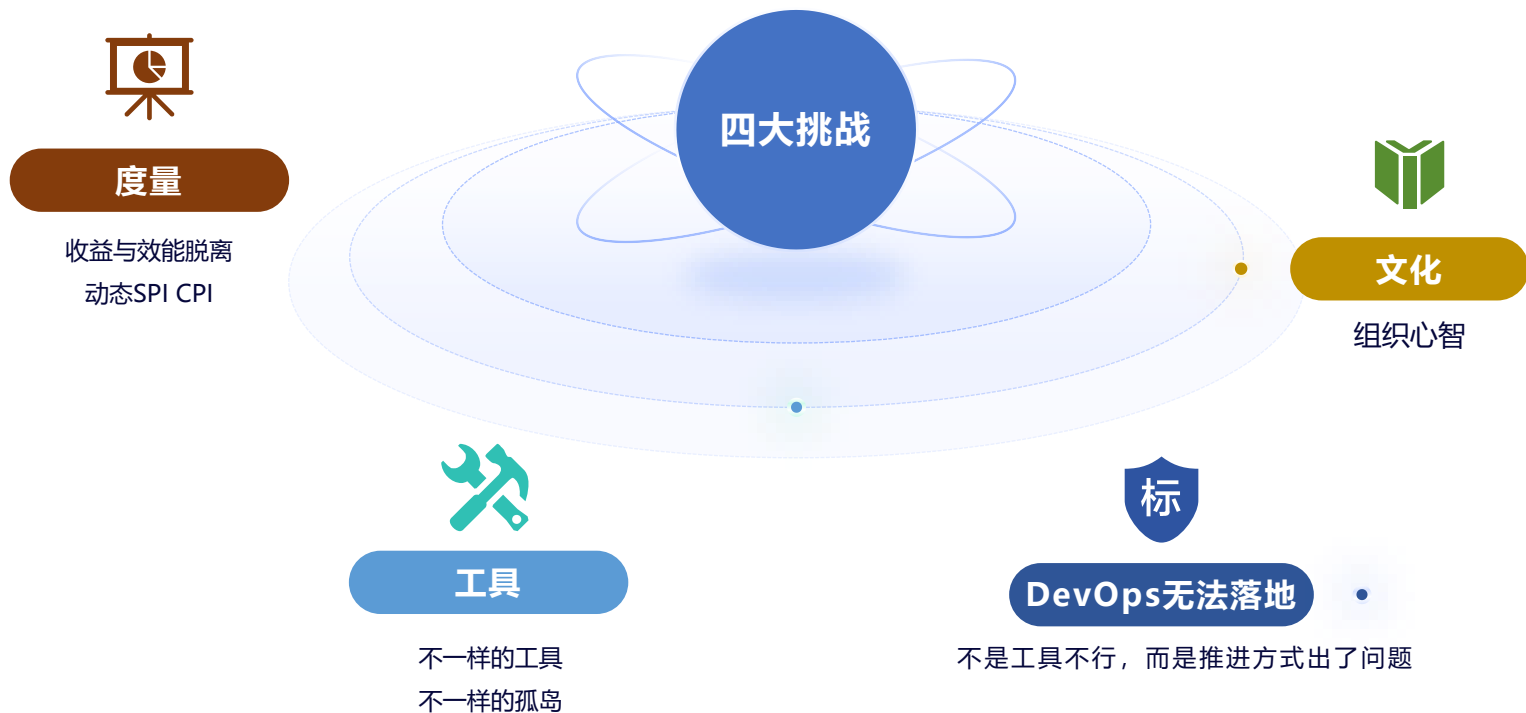
- 1、解析.arxml; 2、结构化字段, 形成映射矩阵; 3、识别稳定要素, 定义基线; 4、去同求异, 灵活配置; 5、即时构建, 快速反馈



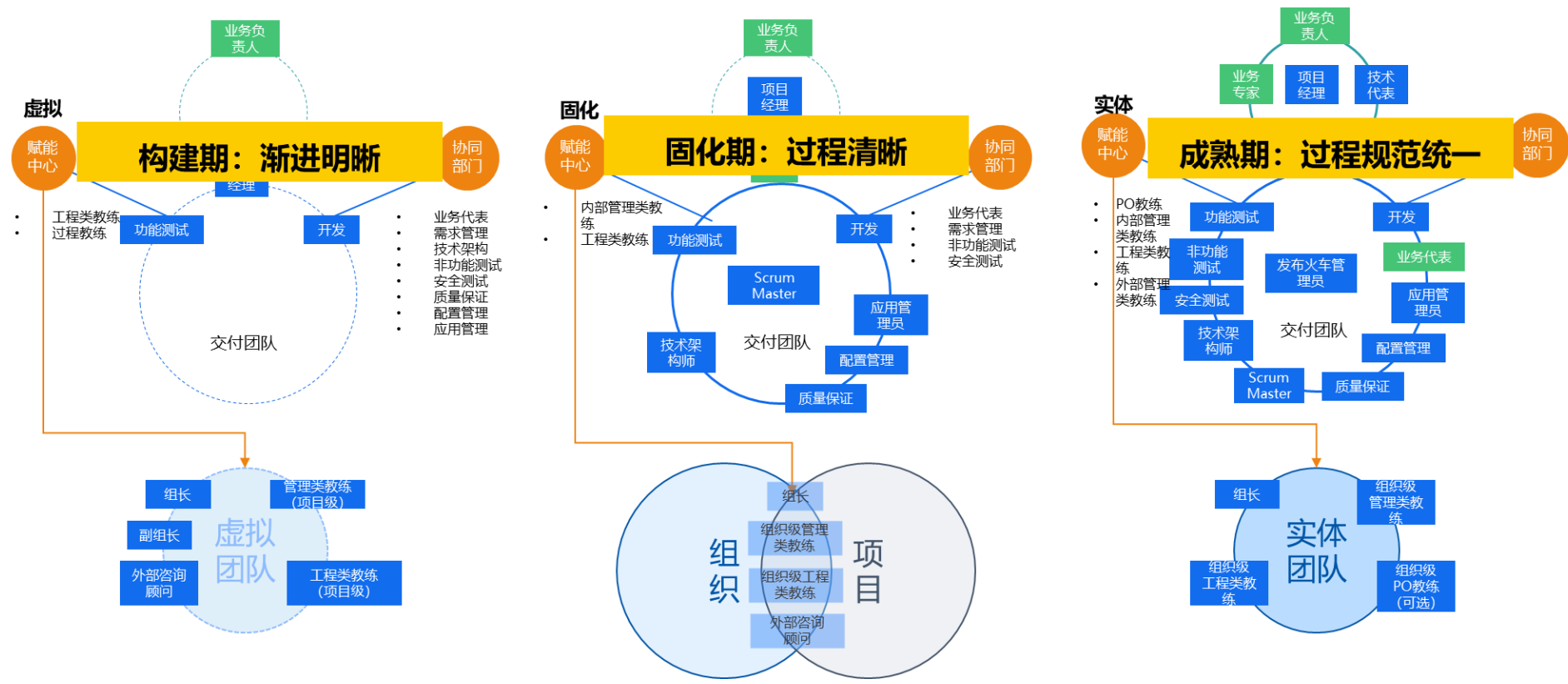




# 工业领域DevOps转型挑战及实践



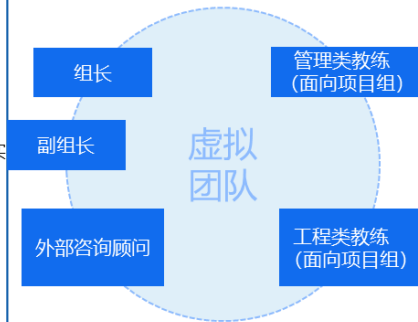
# 推进方式：组建赋能中心



# 推进方式：组建赋能中心

## 【职责】

- **组长：**
  - 负责整个赋能中心的战略规划和管理
  - 确保中心的目标与企业的战略目标一致
  - 负责协调各个部门和团队，确保中心的高效运作
- **副组长**
  - 负责整个devops项目的执行，战略的拆分
  - 在外部咨询顾问的指导下，安排和协调必要的资源、活动
  - 为外部咨询顾问带教扫除阻碍
- **外部咨询顾问：**
  - 负责为企业提供咨询服务，帮助企业了DevOps理念、方法论和实践
  - 制定适合企业的落地策略和实施计划
  - 跟进和指导团队树立标杆
  - 指导赋能团队提升能力
- **管理类教练（面向项目组）：**
  - 积极响应和支持外部咨询顾问的工作安排
  - 在外部咨询顾问的指导下，落地与DevOps平台相结合的管理类实践，例如：精益敏捷
  - 参与项目组活动，发现和识别阻碍并持续改进
  - 收集和提炼团队内部优秀实践
- **工程类教练（面向项目组）：**
  - 积极响应和支持外部咨询顾问的工作安排
  - 在外部咨询顾问的指导下，落地与DevOps平台相结合的工程类实践，例如：CI/CD
  - 参与项目组活动，发现和识别阻碍并持续改进
  - 收集和提炼团队内部优秀实践



## 【所需能力】

- **管理类教练：**精益/敏捷实践，过程建设，流程管理，团队管理等能力。较匹配岗位：项目经理、团队Leader。
- **工程类教练：**配置管理、CI/CD、静态代码扫描等能力。

**说明：**现阶段并不是所有候选人都具体上述能力，可找到比较适合候选人，扮演这样的角色，咨询顾问会根据《工艺沙盘》帮助其提升能力。

- **来源：**各个试点团队，大部分工作本来就要做，会存在学习成本

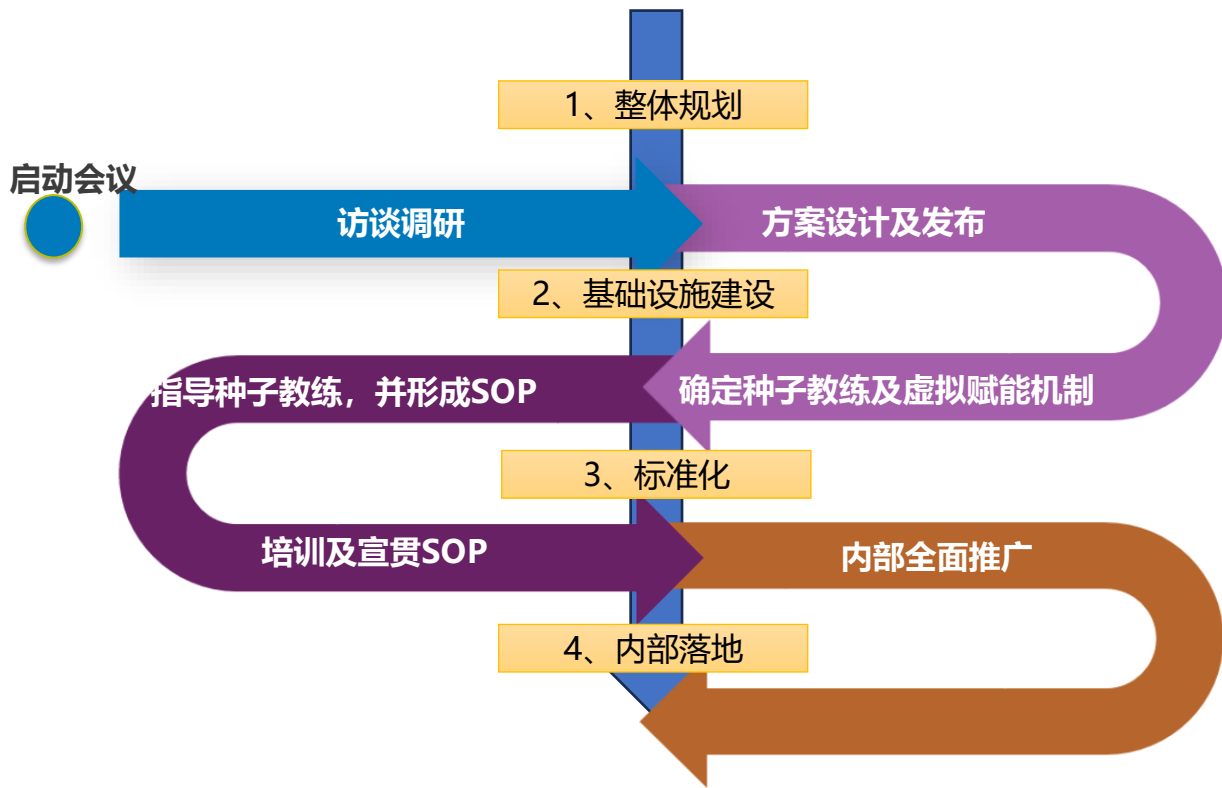
## 【工作机制】

- 1、外部咨询顾问根据调研，制定整体推进计划及带教沙盘
- 2、外部咨询顾问与教练沟通确定《工作日历》
- 3、外部咨询顾问根据工作日历和沙盘带教团队，内部教练全程跟进学习。
- 4、内部教练负责提炼归纳实践和过程框架，并分享
- 5、内部教练需要独立实践，确保能力提升，有问题及时与外部咨询顾问





## • 推进方式：试点成果标准化后直接推广/直接放开推广

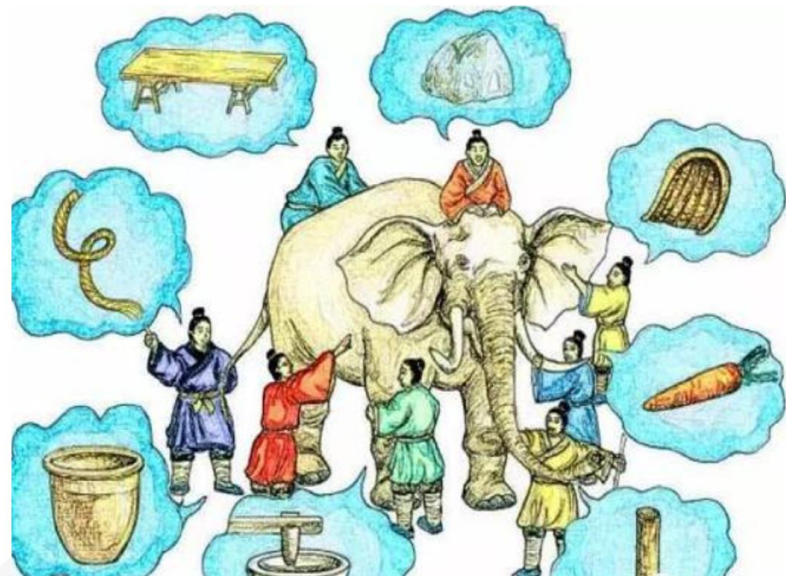


## • 推进方式：试点成果标准化后直接推广/直接放开推广

**缺乏全局视角：**各个团队可能只关注自己部门的利益和需求，而忽视了整个组织或项目的全局视角。这可能导致在推进标准规范落地的过程中出现局部优化而全局受损的情况。



效率竖井







## • 推进方式：标准规范用得不好，就是“虚假繁荣”

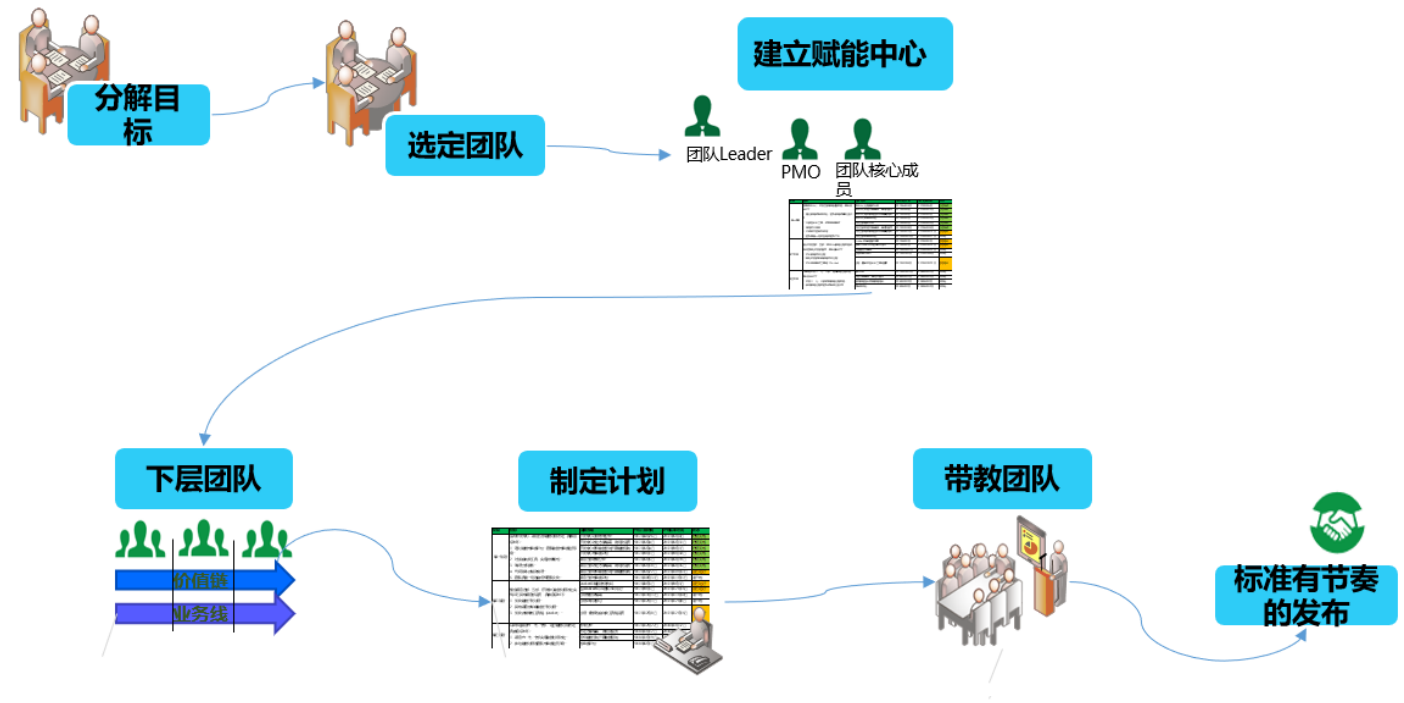
- 标准是规范不了团队所有人员的操作，
- 标准是基线，识别偏差
- 偏差很多时候是好事，取其精华，去其糟粕。
- 糟粕需要治理，精华需要汲取
- 如果一味强推，会是什么后果，如下图：



# • DevOps要成功落地并不容易，需要一整套科学推进的方法论落地

其他因素：

激励机制  
组织结构  
高层支持  
...





# 工业领域DevOps转型挑战及实践





## • 度量：动态化SPI、CPI、EV

在复杂环境下，肯定会做预测，传统的燃尽图可以跟踪Sprint的中进度变化，但是整体项目进度偏差？成本偏差？如何度量，始终是个难题，其实在PMP中本来就有答复，但是传统SPI\CPI指标，是基于不变的初始整体计划作为基线，导致数据不敏感，失真等问题，在DevOps中，我们落地了SPI、CPI、EV。

**成本绩效指数 (Cost Performance Index, CPI)**，截止到某时点衡量成本绩效的一种指标，也就是实际每花一元钱，完成做了多少钱的事（花钱的效率）， $CPI = EV/AC$   
**进度绩效指数 (Schedule Performance Index, SPI)**，截止到某时点衡量进度绩效的一种指标，也就是实际完成的工作量与计划完成工作量之比， $SPI = EV/PV$

**计划价值 (Planned Value, PV)**，截止到某时间点计划要完成工作量的价值，也就是计划要做多少事；

**挣值 (Earned Value, EV)**，截止到某时间点实际已经完成工作量的价值，也就是实际做了多少事；

**实际成本 (Actual Cost, AC)**，截止到某时间点实际已经发生的成本，也就是实际花了多少钱；

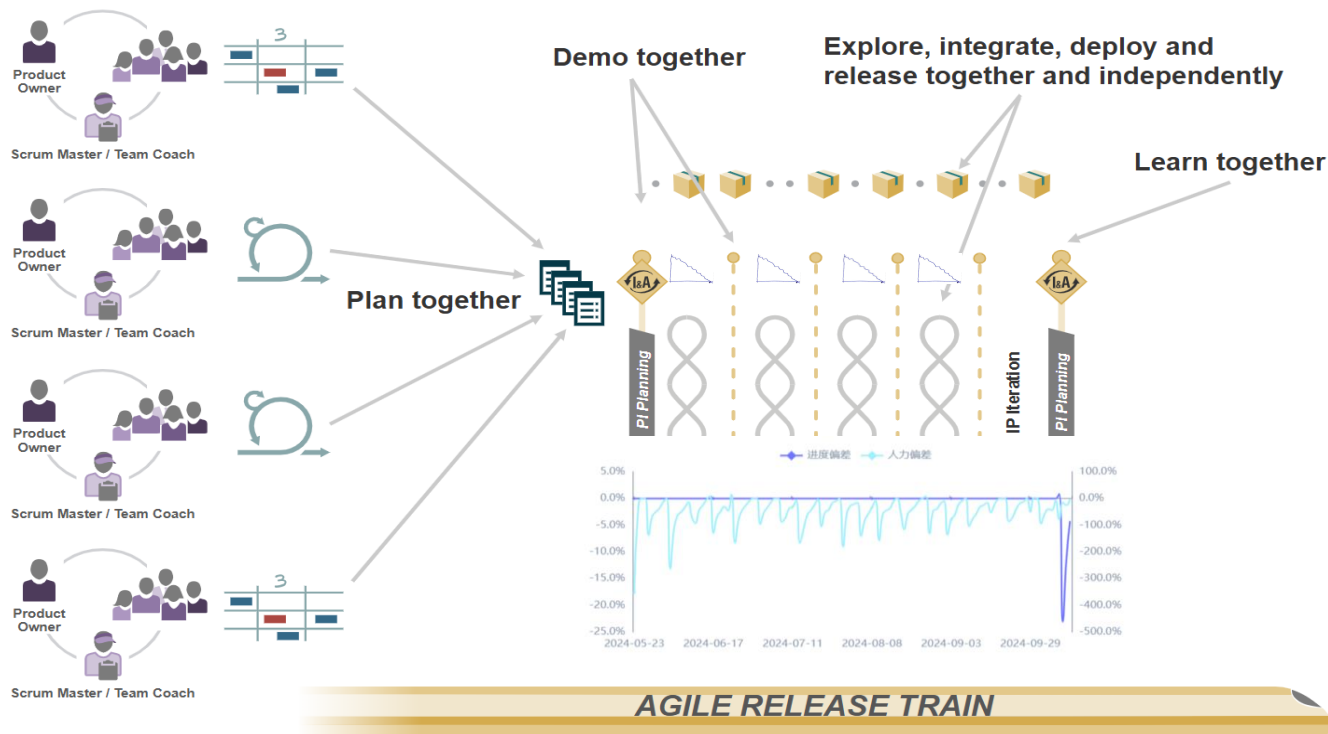
重点项目事项-整体进展情况

重点项目名称	项目状态	预计结束时间	进度偏差	人力偏差	预计人天	已投入人天
DevOps体系建设	进行中	2024-12-31	62.8%	0.0%	200	52
业务中台二期	进行中	2024-12-31	58.0%	0.0%	3918	1179
风控体系建设	进行中	2024-12-31	32.0%	0.0%	1654	668
全民营销及客户管...	进行中	2024-12-31	16.7%	0.0%	366	235
风险管理门户	进行中	2024-12-31	1.8%	0.0%	125	63
供应商管理和租赁...	未开始	2024-10-18	0.0%	0.0%	0	0
征信上报系统	进行中	2024-12-31	-16.4%	0.0%	60	27
资产管理二期	已完成	2024-09-30	-17.9%	-0.6%	130	65
大运营平台建设	进行中	2025-03-03	-20.4%	0.0%	3875	2454
物联网平台创新	进行中	2024-12-31	-31.2%	4.4%	103	6
客户一本账	已完成	2024-05-24	-32.6%	0.0%	6	6



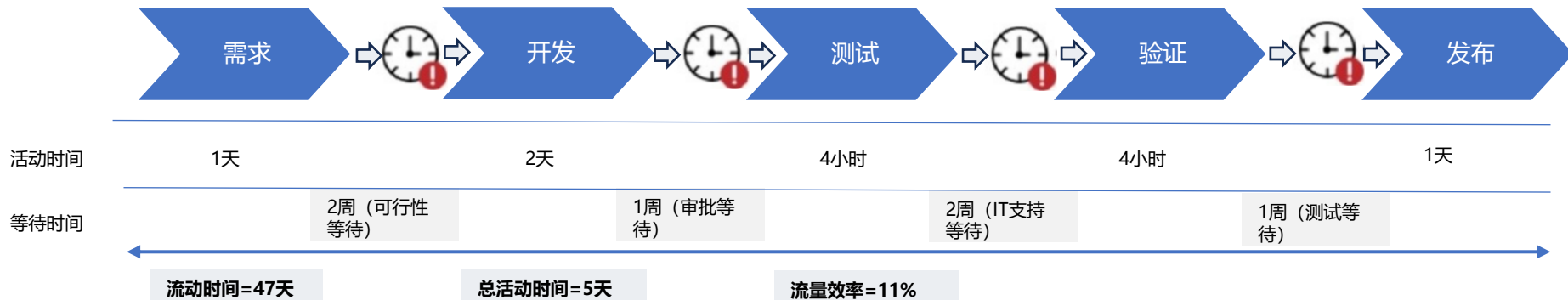


# 度量：动态化SPI、CPI、EV





# 度量：研发价值流



DevOps工具链，优化了哪些步骤？是否真正的端到端？ 互联网和传统企业区别在那里？

## ❌ 问题

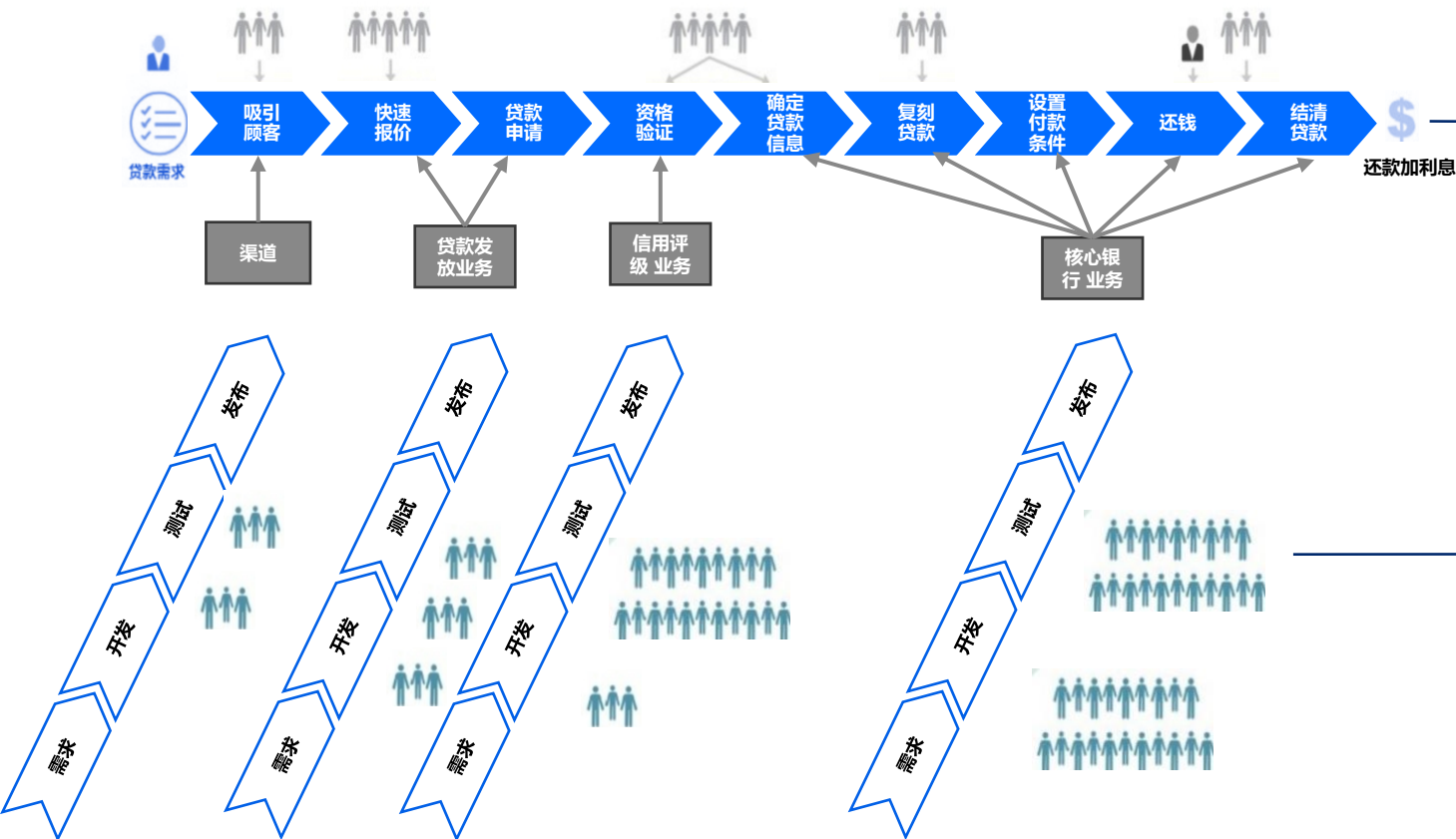
- ❶ 需求追溯
- ❷ 产品需求
- ❸ 产品版本计划
- ❹ 代码审核墙
- ❺ 质量门禁
- ❻ 测试报告
- ❼ 制品发布

## ➡ 可改进点

- Ⓐ 需求处理效率
- Ⓑ 变更电子化
- Ⓒ 变更
- Ⓓ 开发分支
- Ⓔ 白盒自动化
- Ⓕ 黑盒自动化测试
- Ⓖ 缺陷



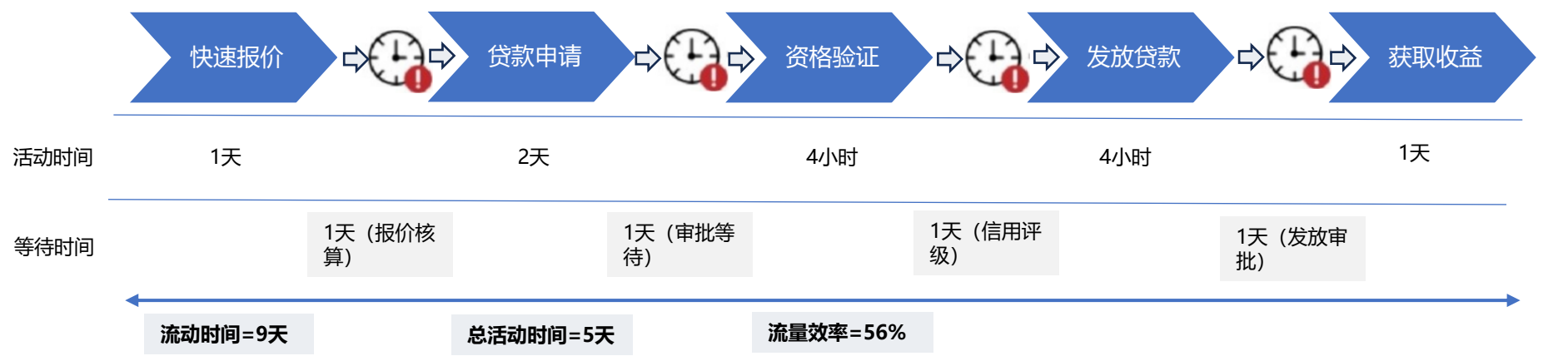
# 度量：运营价值流与研发价值流



**OVS**  
是向客户交付  
产品或  
服务所需的一系列活动

**DVS**  
基于业务假设  
转换为提供客户价值的活动  
序列，需要的  
技术解决方案  
步骤

# 度量：运营价值流



## DevOps工具链，识别运营价值流并改进

### ❌ 问题

- 1 资源配置
- 2 业务流程
- 3 收益可视化
- 4 用户满意度

### ➡ 可改进点

- ① 资源的合理配置
- ② 客户旅程地图
- ③ 闭环反馈
- ④ 持续改进

# 运营价值流

运营价值流（OVS）是向客户交付产品或服务所需的一系列活动



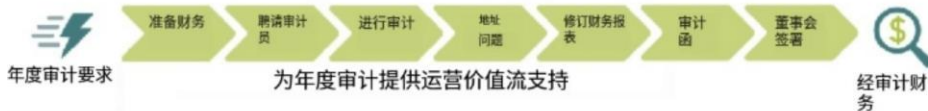
消费者提供保险产品或履行电子商务销售订单，电商行业。



制造将原材料转化为客户购买的产品，制造业。  
示例包括消费类产品、医疗设备和复杂的信息物理系统。



软件产品提供和支持软件应用程序和解决方案，以出售给最终用户或企业，软件行业。示例包括ERP 系统、SaaS 以及桌面和移动应用程序。



支持价值流是各种重复性和内部支持活动的端到端 workflow，支撑类行业。示例包括员工招聘、建立和执行供应商合同、执行年度审计以及完成企业销售周期





# 工业领域DevOps转型挑战及实践







## 文化变革：组织心智

75%的传统企业转型为什么会失败，不是工具的问题，不是流程的问题，不是组织结构的问题，企业转型最难的从来不是用什么招式，而在于看不见、摸不着的组织心智。一旦形成，难以改变。

# 源于社区 服务社区

 中国DevOps社区峰会 2024 · 上海

