麦哲思科技 Measures

DevOps量化目标驱动的改进

简介：陈正思

软件工程师 ➡ 项目经理

项目管理岗 ➡ 咨询顾问

DevOps：DevOps Master/Professional授权讲师

敏捷：PMI-ACP、 SAFe SPC、LeSS-CLP

CMMI：CMMI-ATM、CMMI 2.0 Associate

项目管理：项目管理师

规模度量：COSMIC-FFP认证

目录

CONTENTS

驱动改进的两种力量

■ 解决当前遇到的问题

■ 达成未来更高的目标

开发人员习惯于用技术手段解决问题

- 开发人员是研发工作的核心
- 拿到需求后开发人员首先考虑的是使用哪种技术实现（自己熟悉的、同业使用的、最先进的）
- 当项目推进过程中出现问题时，通常会在技术上探索新的可能性

解决问题的各种手段

改变思维方式是驱动改进效果的关键

- 看似人的问题，实则情境问题。
- 看似懒于改变，其实精疲力尽。
- 看似生心抗拒，实则方向不明。

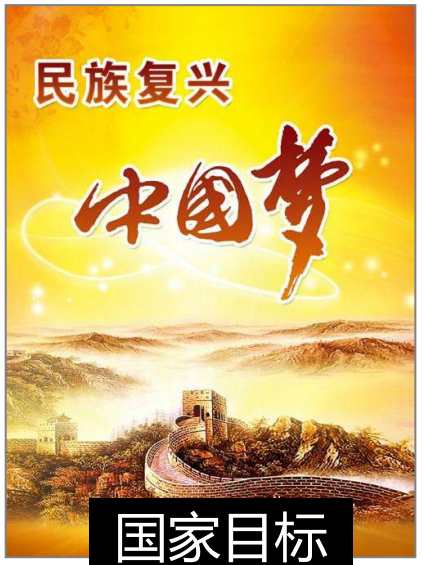
- 系统一：直觉、本能、潜意识
- 系统二：深思熟虑，意识

不同规模的团队都能建立共同目标

个人目标

团队目标

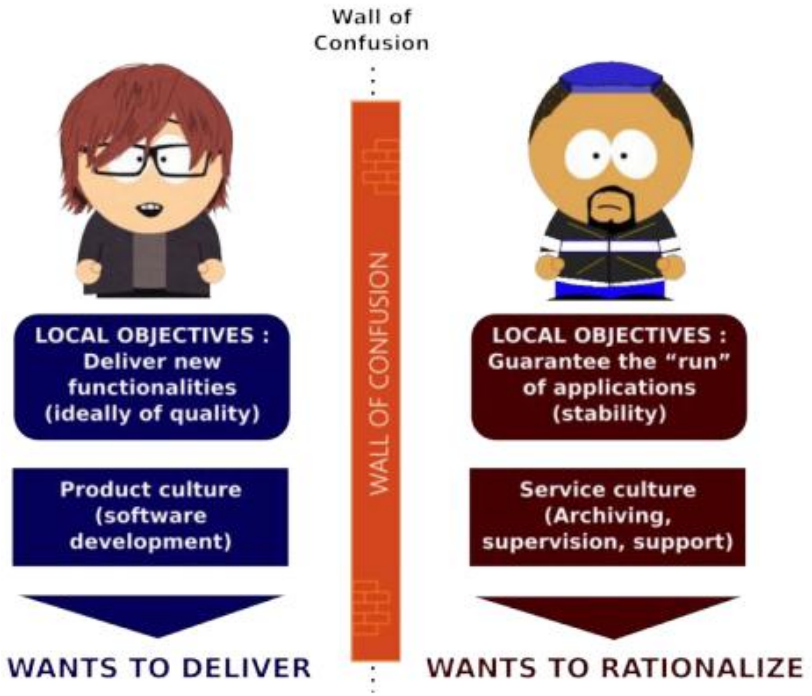
企业目标



国家目标

全球目标

研发交付目标逐渐被过程目标侵蚀



整体目标的分解引发冲突

整体目标应该变清晰

有效驱动改进面临的几个障碍

- 容易将解决问题的手段限制在单一领域
- 目标本身定义不清晰或逐渐被侵蚀
- 总体目标分解后引发对立冲突、影响力被削弱
- 目标未被执行主体认同

目录

CONTENTS

DevOps年度报告中的统计数据

软件交付效能的度量指标	精英[a]	高效能	中等效能	低效能
部署频率 对于你负责的主要应用或服务，你的组织部署代码的频率？	按需（每天多次部署）	介于每小时1次和每天1次之间	介于每周1次和每月1次之间	介于每周1次和每月1次之间
变更前置时间 对于你负责的主要应用或服务，变更的前置时间是多长（即从代码提交到代码成功运行在生产环境需要多长时间）？	小于1小时	介于1天和1周之间	介于1周和1个月之间[b]	介于1个月和6个月之间[b]
服务恢复时间 对于你负责的主要应用或服务，发生服务故障时（例如：计划外中断、服务受损），恢复服务通常需要多长时间？	小于1小时	小于1天	小于1天	介于1周和1个月之间
变更失败率 对于你负责的主要应用或服务，有多大比例的变更会导致服务降级或需要事后补救？（例如：导致服务受损或服务中断，需要热修复、回滚、前向修复、补丁）	0-15%	0-15%	0-15%	46-60%

IT企业的交付能力差异巨大

Company	Deploy Frequency	Deploy Lead Time	Reliability	Customer Responsiveness
Amazon	23,000 / day	minutes	high	high
Google	5,500 / day	minutes	high	high
Netflix	500 / day	minutes	high	high
Facebook	1 / day	hours	high	high
Twitter[2]	3 / week	hours	high	high
typical enterprise	once every 9 months	months or quarters	low/medium	low/medium

部署频率在驱动什么？

驱动团队致力于快速响应变化的环境

- 探索如何快速响应变化
- 按需设置的迭代长度
- 松耦合架构
- 持续交付流水线
- 赋能团队

变更前置时间在驱动什么？

自动化、内建质量

- 基础设施即代码
- 自动化部署
- 持续集成
- 自动化测试

服务恢复时间在驱动什么？

服务的可用性、服务稳定性和修复能力

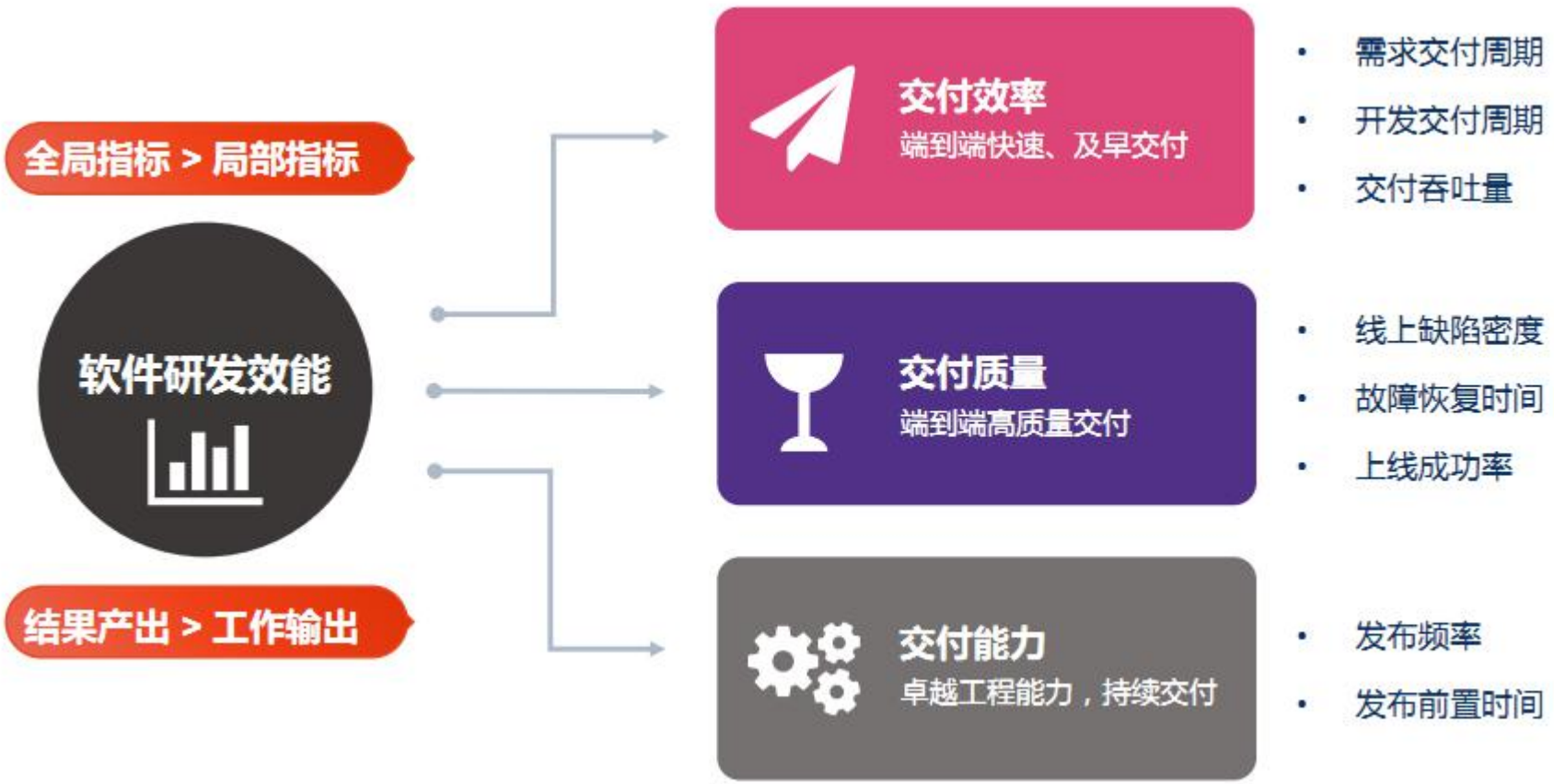
- 应用可扩展性、资源可伸缩性
- 自动化部署
- 自动化测试
- 实时监控与自动修复
- 前瞻性预警通知

变更失败率在驱动什么？

交付质量、系统健壮性

- 可视化与透明性
- 自动化部署与回退
- 破坏性试验
- 自动化测试

交付目标组成



交付效能指标 Vs 研发过程指标

研发过程度量的一些局限性

- 代码行不准、功能点太难，且不能等同于业务产出价值（科学度量除外）
- 用工作量投入或功能规模产出评价知识工作者难以被认同（软件冰山类比法除外）
- 研发包含的创造性工作过程本身的不稳定性和可视化程度不高
- 研发工作职责边界和范围难以严格区分，工作切分也没有唯一标准
- 局部工作输出对全局目标的贡献难以界定，局部目标从全局来看可能被部分抵消

定性目标 Vs 定量目标

定量目标能有效支持持续改进

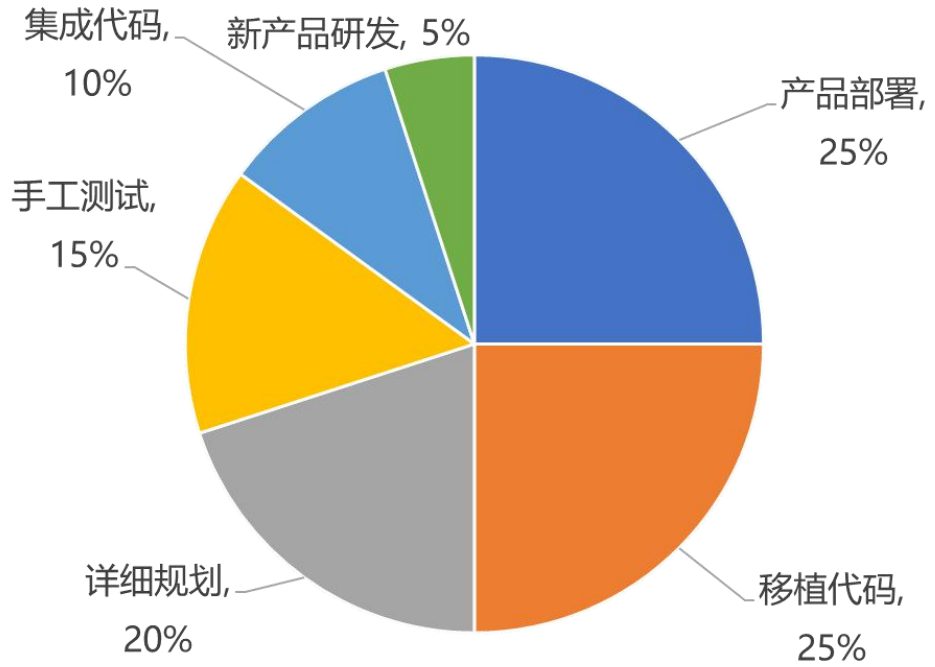
- 定量目标更容易衡量
- 可为团队提供更明确的方向引领与指导
- 可将量化目标进行分解，最终映射到具体的行动项
- 可以用可视化的方式呈现研发过程与结果

目录

CONTENTS

量化管理驱动改进的例子

HP激光打印机固件部门（400名开发人员、分布式团队）



量化管理驱动改进的例子

改进推进路线

【**A**rchitecture adjustment】架构调整（功能开关）

【**C**ontinuous Integration】持续集成与自动化分级测试

【**D**evelopment iteration】迭代开发（替代瀑布）

【**P**lanning agile】敏捷计划（业务敏捷）

量化管理驱动改进的例子

采用的一些改进措施

- 使用持续集成
- 基于源代码控制，将代码提交到共享分支(基于主干的开发)
- 加大自动化测试投入
- 创建硬件模拟器，以便在虚拟平台上运行测试
- 使用统一的构建和发布方式，支持所有打印机产品

量化管理驱动改进的例子

研发过程的改进效果

- 最初每天执行1次构建，最终实现每天执行10-15次构建；
- 最初每天由总构建人进行大约20次代码提交，到每天所有开发人员有超过100次的代码提交
- 开发人员每天更改和添加的代码行数达到了7.5-10万行（400人团队）
- 回归测试的周期从6周缩短为1天

量化管理驱动改进的例子

最终带来的改进收益

- 开发人员用于创新和新特性开发的时间从5%增加到40%
- 总开发成本降低约40%
- 处于开发状态的项目增加了约140%
- 每个项目的开发成本降低了78%

在网上向所有人公布的里程碑交付目标

小里程碑目标（30个迭代）

优先级	主题	完成条件（目标完成或没有完成）
0	**质量准入条件**	测试失败修复时间<1周；L2测试失败24小时内给出响应
1	**季度性小发布**	完成所有P1变更；稳定性错误率达到发布条件
2	新平台稳定性和测试覆盖率	客户验收测试100%通过；所有L2测试通过率98%以上 满足L4测试准入条件；覆盖所有已打开功能的L4测试 新产品L4测试通过率100%
3	产品的前后依赖和关键功能	持续打印一小时，打印速度与装订速度一样 全速复印测试一小时；可以设置省电模式 通宵进行生产线测试集测试；公共测试库对4条控制面板的显示进行测试
4	下一个版本的产品编译	端到端的新处理器编译；对新处理器进行高级性能分析
5	系列集成计划	根据内容和日程，在系统测试实验中对架构薄片进行端到端的敏捷测试

用过程度量数据支撑的反馈信息

集成队列/自动化编译状态 每日度量结果 05/24/2011							
编译器类型	**编译类型**	**编译执行次数**	**编译通过总数**	**编译失败总数**	**分支测试执行总数**	**分支测试通过总数**	**分支测试失败总数**
Stage 2 Builder	Stage2 Combined	10	4(40%)	6(60%)	133	105(79%)	28(21%)
Stage 2 Builder	Stage 2 CEQbar	36	31(86%)	5(14%)	36	31(86%)	5(14%)
Stage 1 Builder	Stage1 Qbar	50	42(64%)	8(16%)	50	42(84%)	8(16%)
Stage 1 Builder	Stage1 XPQbar	23	21(91%)	2(9%)	23	21(91%)	2(9%)
Totals for all Stage 2 builds		10	9(90%)	1(10%)	133	105(79%)	28(21%)
Totals for all Stage 1 builds		109	94(86%)	15(14%)	109	94(86%)	15(14%)

目录

CONTENTS

总结

1. 随着待解决问题的复杂度提高，除了技术手段和管理手段，我们还需要遵循认知规律来改变思维方式，通过迭代的方式来促进个体意识与潜意识对话，从内部驱动因素来优化我们的行为模式。
2. 驱动改进的力量，自下而上是解决问题，自上而下是目标引领，解决问题需要考虑多层次共同作用，目标引领要量化且面向交付。
3. 当我们开始转向思考一些典型的交付指标时，其实我们已经切换到了全局优化和系统思考的模式，能使我们的改进能针对瓶颈。

麦哲思科技 Measures

感谢聆听！