

百度工程能力地图介绍

2018 DevOps Days 深圳站
2018 中国·深圳



个人简介:

朱华亮:

- 百度资深测试工程师
- 百度CI/CD技术委员会负责人
- 负责主导百度凤巢、糯米、百度APP、信息流等产品DevOps实践
- 长期致力于客户端和服务端质量效率提升、AI赋能质量等相关技术领域

2018 China DevOps Days 深圳站



工程能力地图Why:

- 从事DevOps工程实践一般都会遇到以下几个问题;



如何评

如何知道建设的工程能力达到何种水平

全面评估



如何做

全新业务怎么做才能达到

合理指引



如何快速达到

如何低成本快速达到

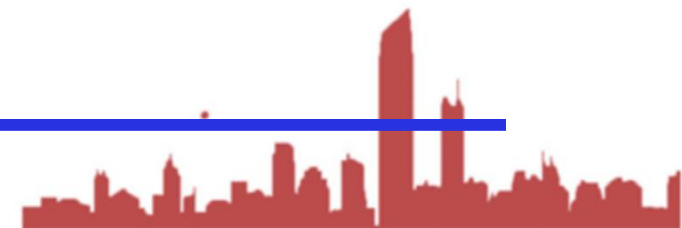
快速达成



如何迭代优化

如何根据现状逐步优化

持续优化



项目目标

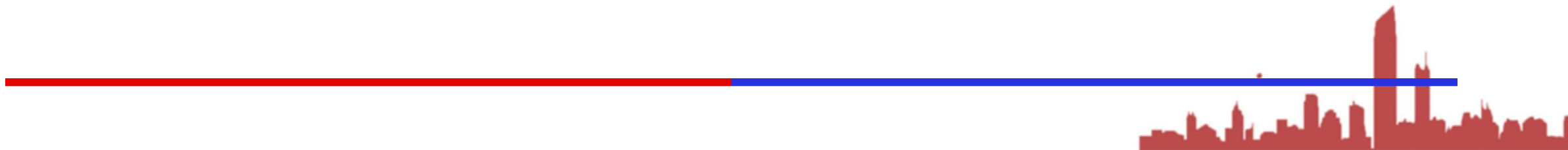
- 基于业界和百度内部实践经验，打造完整的DevOps流水线，为百度工程能力建设提供指导
 - 规范研发与上线流程- 可衡量
 - 形成推荐工具链方案- 可操作
 - 统一的工程能力视图- 可见

2018 DevOps Days·深圳站



- 百度工程能力地图
 - 需求、开发、提交、测试、交付的环节、级别、指标和工具
- 集成的标准化研发工具链
- 工程能力Dashboard，指标数据通过工具自动生成

2018 DevOps Days 深圳站



工程能力地图如何制定：

完 备

各流程环节该具备何种实践

有 效

各个实践效果达到多少

2018 DevOps Days·深圳站

研发流程环节是否完善

流程

CI/CD委员会制定初版

提至技术委员会酝酿

进行公示

抽样分数与实际分析

流 程

- 1、保证权威；
- 2、保证易懂；
- 3、具备指引作用；

适用范围 (大类划分 测试或部署方式有明显区别)

Server研发流水线

- PHP
- Java Web
- 策略(C++)
- APP插件(H5等)
- Python类

APP研发流水线

- Android
- IOS
- APP插件 (SDK)

SDK研发流水线

- 语音、度秘SDK等

APP研发流水线



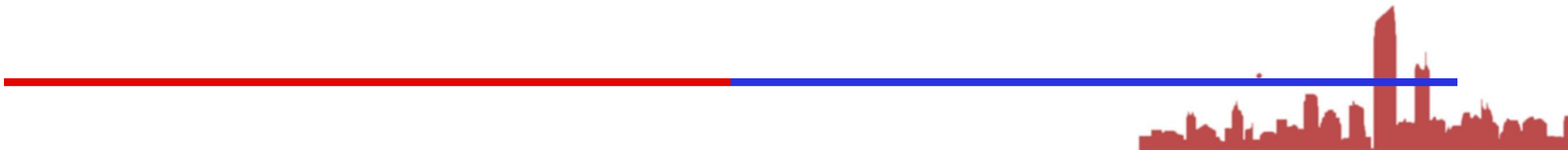
标准&工具 - 测试 (APP&SDK)

UI兼容性测试	L0: 有UI兼容性测试环节, 有至少一款真机任务报告 L1: 有UI兼容性测试环节, 且代码覆盖率或activity覆盖率大于等于15%	luban、mvp
性能评估	L0: 具备自动化能力, 至少返回8项不同场景指标 L1: 有竞品分析报告	luban
性能测试	L0: 有能力判断至少2项测试结果异常 (内存、CPU、耗电、流量等) L1: 有能力判断至少4项测试结果异常 (内存、CPU、耗电、流量等)	leackcanary
全量静态代码扫描	L0: 进行全量静描, 千行高危问题<0.4 L1: 进行全量静态代码扫描, 修复所有高危问题	Bugbye
包安全扫描	L0: 有安全扫描环节, 修复全部漏洞 (误报可标记剔除)	appscan
测试用例管理	L0: iCase、ITP或Case代码管理, 由工具平台返回 L1: 具备Case分级手段, 由工具平台返回	iCase、源码管理
稳定性及遍历测试	L0: 有稳定性测试, activity覆盖大于等于5% L1: 高覆盖稳定性测试, activity覆盖大于等于20%	xmonkey、musi
自动化回归测试 (含UT)	L0: 自动化回归, 全量分支覆盖率大于等于10% L1: 自动化回归, 全量分支覆盖率大于等于20%	MVP、musi
系统兼容性测试	L0: 有兼容性测试环节, 覆盖机型或版本 10% L1: 有兼容性测试环节, 覆盖机型或版本 40%	musi、MVP、MTC
全量源码安全扫描	L0: 修复全部漏洞 (OC/OC++删除)	猫头鹰

流水线/自动化:

- 异常构建率: $\text{失败构建} / \text{总构建数}$, 任务范围主线的merge流水线。
- 失败恢复时间: 失败转为成功的时间间隔, 任务范围主线的merge流水线。
- 执行效率: $\text{模块总构建时间} / \text{总构建数}$, 所有merge流水线。

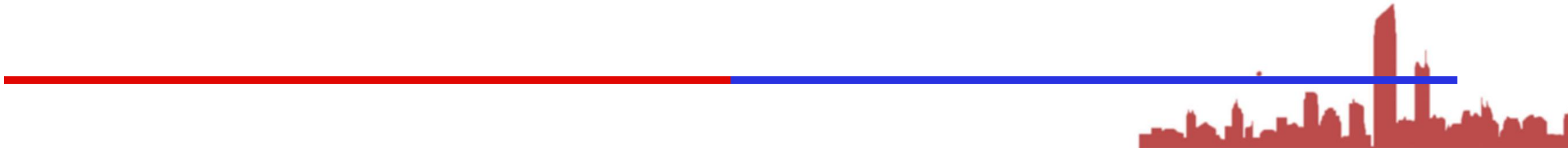
2018 DevOps Days 深圳站



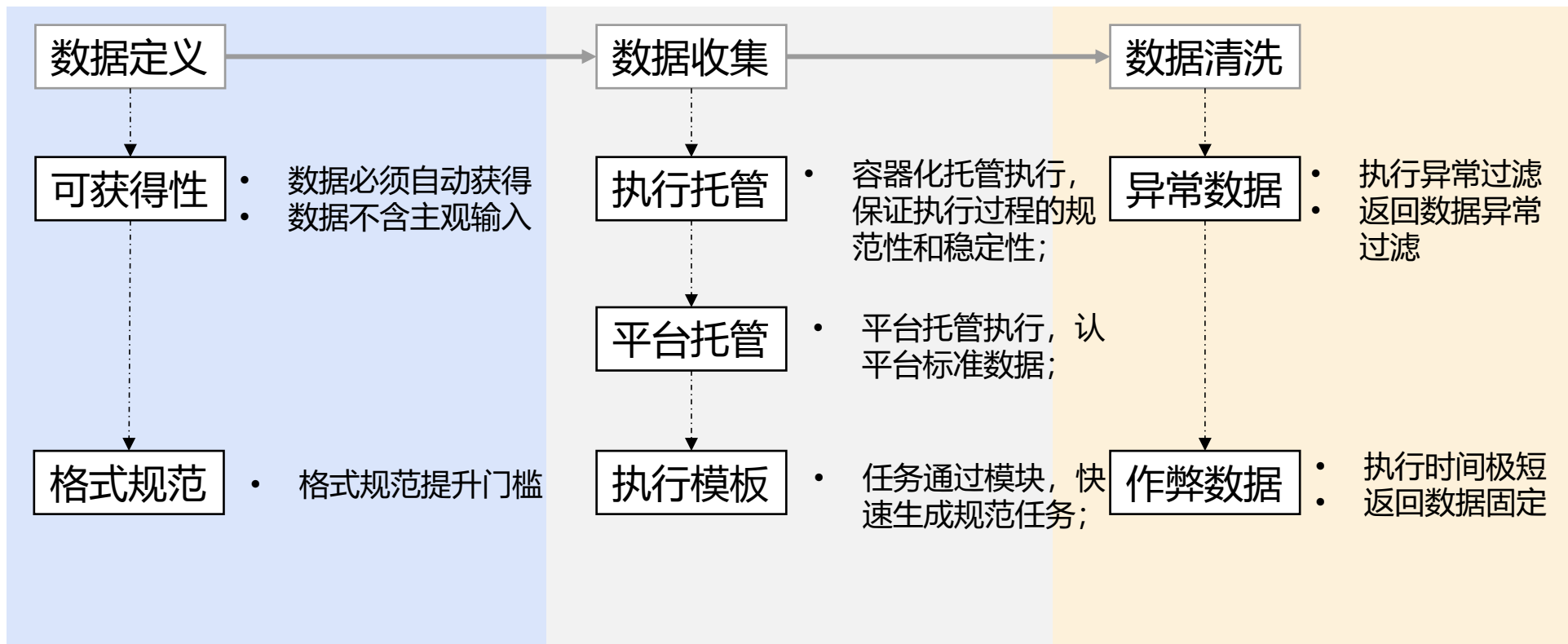
工程能力地图需要的主要特点

- 数据更真实、更全面
 - 数据获取方式、过滤方式、发布范围;
- 算法更具弹性
 - 四种调权方法，让算法更具备适应空间
- 分数更贴近实际
 - 更加重视有效性

2018 DevOps Days 深圳站



如何收集数据:



- 1、保证客观；
- 2、保证真实；
- 3、保证持续；

关于规范：

- 1、数据格式参考agile；
- 2、jenkins数据必须得到合理关联；
- 3、建议都吐原始数据；

算法:

$$T_i = \sum_{j=1}^n T_{ij} = C_i \left(\sum_{j=1}^n V_{ij} P_{ij} + \sum_{j=1}^n V_{ij} Q_{ij} D_{ij} - \sum_k^m V_{ik} \right)$$

T_i : 代表第 i 个阶段的实际得分;

C_i : 代表第 i 个阶段分配的分值;

V_{ij} : 代表第 i 个阶段第 j 个实践的权重;

P_{ij} : 代表第 i 个阶段第 j 个实践的完备性权重;

Q_{ij} : 代表第 i 个阶段第 j 个实践的有效性权重;

D_{ij} : 代表第 i 个阶段第 j 个实践达到的标准层级/该实践的总级数, 比如该实践总共有 3 个级别, 如果达到了第二个级别, $D_{ij}=2/3$;

备注: $P_{ij} + Q_{ij} = 1$ 备注: $\sum V_{ij} = 1$

例子:

APP	需求	开发	代码准入	测试	灰度验证	发布	流水线/自动化
C=	5	10	20	35	10	10	10

实践名称	分级标准			完备性	有效性
	Average	Good	Excellent	10	10
百度编码规范	•			1.25	0.63
增量源码安全扫描	•			1.25	0.63
CodeReview	•	•	•	1.25	1.88
增量静态代码扫描	•	•	•	1.25	1.88
云端编译	•	•		1.25	1.25
P0级自动化回归测试	•	•		1.25	1.25
Unit Test	•	•		1.25	1.25
可维护性检查	•	•		1.25	1.25

- 1、保证权重可调整: 各个阶段、各个实践、完备性和有效性的调权, 不断迭代, 达到最优;
- 2、保证一定弹性: 达到依据业务需要弹性控制;

流水线：泛指DevOps所有环节依赖的工具平台的配置、操作等内容，agile流水线是其核心重要一环

- 0-30： DevOps各个阶段的流水线方面有明显缺陷，流水线几对产品研发过程帮助很少；
- 30-50： DevOps各个阶段的流水线等几个方向建设得相对较完善，流水线在部分方向有突破，主要在风险把控方面有效辅助产品研发；
- 50-70： DevOps各个阶段的流水线等大部分方向建设得比较完善和深入，流水线较为完整和深入，流水线对产品研发的风险把控有明显赋能作用，研发效率赋能逐步开始呈现；
- 70-85： DevOps各个阶段和流水线等大部分方向建设得非常完善和深入，流水线全面且可协同有效运转，流水线对产品研发的风险把控有非常显著赋能作用，研发效率赋能作用明显提升；
- 85-100： 产品研发完全融入在DevOps各个阶段和流水线中，流水线完全实现风险可控，随时发布。

部分产品描述



需求	开发	代码准入	测试	灰度 / 验证	发版 / 上线
需求记录	代码集中托管	增量源码安全扫描	全量源码安全扫描	灰度发布能力	包版本管理
状态更新	关联产品	百度编码规范	手工功能测试	灰度数据自动收集	APP手工打渠道包
迭代计划	关联需求	CodeReview	兼容性测试	灰度数据自动计算	APP自动出包
	分支规范	增量静态代码扫描	自动化回归测试	自动灰度质量报告	
	提交规范	云端编译	稳定性及遍历测试		
		P0级自动化回归测试	测试用例管理		
		Unit Test	插件的自动化准入测试		
		可维护性检查	包安全扫描		
			全量静态代码扫描		
			性能测试		
			全功能UI回归		
			异常测试		

流水线自动化

主线成功率

执行效率

失败恢复时间

我们还在路上。。。。

按照模块特性自动调权

根据模块特性计算重点建设的实践

不同模块调权机制个性化

不断推出其余产品形态标准建设

实践分数与研发效率更强的正相关

有了标准，研发大数据成为可能，AI赋能DevOps时代，
我们来了



THANKS

Website:
chinadevopsdays.org/

Global Website:
www.devopsdays.org/events/2018-shanghai/

Official Email:
organizers-shanghai-2018@devopsdays.org



Official Wechat

