Baidu 百度

百度工程能力白皮书

——工程标准V1.0

前言

自1968年软件工程被首次提出，到今年刚好是50周年。在这半个世纪的时间里，质量和效率是软件工程领域永恒的主题。人类自第一次工业革命以来，解决质量和效率问题的不二法门就是自动化、自动化、再自动化。智能化更是自动化的一种高级形态。然而在当今企业中，这样的一个根本方法却被人们无意或有意的忽视。说无意忽视，是因为在企业中负责工程能力建设的人可能缺乏工程经验，心中眼中只有流程。仿佛流程顺了，即便大都靠人工执行，也心安理得。说是有意，则是因为软件行业中原有的分工等原因，自动化的导入带来分工模糊、岗位重要性变化等诸多不确定性，缺乏引入的意愿。本人无意贬低流程，而是更推崇自动化和智能化的流程，更无意非议职责分工，而是更推崇自动化和智能化解放人，使每个人都能更多从事创造性的工作。

人们对于人与世界、人与人的关系的认知，在过去的半个世纪里并无显著突破，这些认知又进一步决定了流程优化的上限，仿佛一道透明天花板，人人可感，却无从突破。另一方面，ABC技术（A-AI人工智能、B-Big Data大数据、C-Cloud云计算）突飞猛进、日新月异。如果工程能力的建设无法站在ABC技术进步的巨人肩膀上做创新，那是软件工程领域在这个时代的悲哀。

互联网公司关注快速迭代，天生具有追求卓越工程能力的基因，百度作为技术立身的企业，创立早期便启动了研发流程自动化的工程能力建设，需求流转、文档阅读、代码搜索、程序分析、软件测试、制品部署、服务监控、用户反馈等覆盖研发全流程的工具平台积累，涉及到Server、App、SDK、AI算法模型、自动驾驶等异构产品业务形态。40余个工程实践环节，积累了40余个可插件化接入的平台，在公司内被广泛使用。在这些实践和平台运行的同时，一个新的金矿正在悄然生成，那就是软件工程大数据。如今我们依托百度领先的ABC技术和工程能力建设经验，将公司积累的500多亿条软件工程数据进行处理，其中锤炼出的技术用于代码推荐、缺陷检测、智能测试、智慧运维等领域。今年百度承接了国家重点专项《基于编程现场大数据的软件智能开发方法和环境》，与国内软件工程领域实力顶尖的15所高校一同建设中国未来的智能软件开发平台。

作为百度内部以『工程卓越、毕生所求』为使命的一群人，我们一边支持公司业务打胜仗，一边努力探索工程能力建设的新方向，心中充满着欣喜。并非常渴望能将我们的理解和主张向行业分享，共同进步。然而由于工程能力体系庞大、平台众多，一时无法穷尽，所以我们将这个体系进行分解，首先分享价值高、可迁移的《工程标准V1.0》。后面陆续会向大家汇报我们在AI赋能软件开发、AI开发软件工程等方面的进展。更期待您的反馈和交流。

李涛

百度工程效率总监

百度平台化委员会秘书长

2018年10月10日

编委会成员

顾问：陈尚义、闫艳、高果荣、贺锋、马杰、王龙

主编：李涛

编委：白伟、董海炜、朱华亮、王伟冰、杨斐、彭云鹏、文立、胡飞、刘存良、姜丽芬、杨杨、刘庆、刘玮立

统稿：王一男

视觉：张梦雨

目录

百度软件工程标准概述 1
工程标准制定的背景 1
工程标准目标和意义 1
工程标准的修订规则 1
工程标准的制定思路 1
百度软件工程实践集 2
百度的软件工程类型 2
Server工程类型的工程实践集 2
App工程类型的工程实践集 3
SDK工程类型的工程实践集 4
部分实践名词解释 5
百度软件工程实践标准 6
需求(Server/App/SDK) 6
开发(Server/App/SDK) 7
代码准入(Server/App/SDK) 8
测试（Server） 9
测试（App & SDK） 10
上线&验收（SERVER） 12
灰度 （APP&SDK） 12
发版（App） 12
交付（SDK） 13
流水线/自动化 (Server/App/SDK) 13
百度软件工程标准的实施 14
实施方式 14
验证方式 15
实施效果 16
未来与展望 17

百度软件工程标准概述

工程标准制定的背景

百度是一个技术公司，研发工程能力的高低直接影响公司的持久创新力和公司在市场上的作为。只有不懈追求卓越的工程能力，才能够带来长期的核心竞争力，才能为每个用户、每个企业客户以及整个社会创造价值。长期以来，百度在大量的软件开发经验中总结了许多优秀的工程实践，这些实践来自于公司工程标准和开发工具链的结合。经过长期团队观察和大量研发数据分析，我们证明这些实践可以有效帮助提高软件开发效率和产品质量。

工程标准目标和意义

百度软件工程标准制定的目标是帮助研发团队持续提升工程能力。工程标准可以快速指导团队采用优秀的软件工程实践和研发工具，使其在研发效率或产品质量上获得提升。同时有了标准和规范，也能够更有效地衡量团队工程能力的水平，让各个团队能够更好地了解自身的工程能力现状，进而设定工程能力提升目标，不懈追求工程卓越。

本白皮书希望分享百度在软件工程标准、实践、度量和改进方面的经验，呼吁业界共同加强工程能力建设、研发工具投入、工程标准更新与工程素养提升，共同推进软件工程的发展。

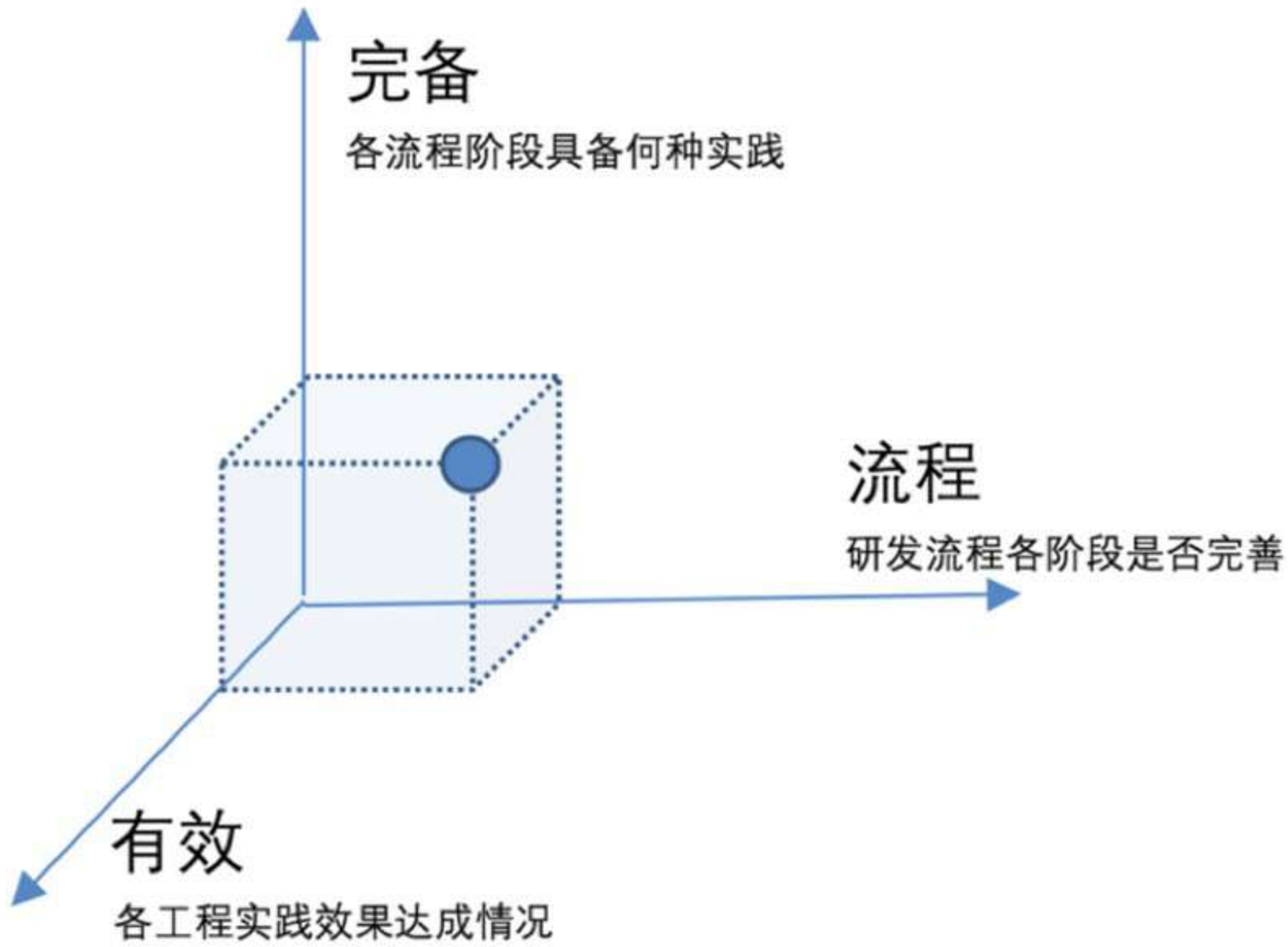
工程标准的修订规则

百度软件工程标准是由百度DevOps TOC（Technical Oversight Committee）制定并发布的，并且随着公司工程技术的发展不断更新。首先，DevOps TOC的委员广泛收集各个研发团队的优秀工程实践，以及工程标准实施的反馈，制定工程标准的初版或修改意见草案。然后TOC委员将草案提交至DevOps TOC进行充分讨论，如果TOC会议通过，就进行标准修改的公示，同步修改研发工具中的对应规则。同时收集各团队反馈和实际研发数据进行分析，来验证标准实施的效果，并继续准备下一轮的规则更新。

本白皮书是百度软件工程标准1.0版本。在此之前，百度软件工程标准已经在百度内部经历了十余次修订，基本包含了百度所有产品团队的优秀工程实践，具备了可靠的权威性，起到了有效的指导作用。

工程标准的制定思路

百度软件工程标准从典型的研发过程中，抽象出研发过程主要看流程是否顺畅、实践是否完备、实践是否有效三个方面，再加上构建系统是否可靠稳定，形成四位一体的工程能力标准，具备流程全面、实践完备、效果驱动、构建合理等特点。



百度软件工程实践集

百度的软件工程类型

百度公司的软件产品形态有多种，例如App，Browser，PC Client，SDK等。不同的产品类型其研发过程及优秀实践也不尽相同。目前我们把研发的工程类型大致分为4类：Server、App、SDK和AD（Autonomous Driving），本白皮书介绍Server、App和SDK三种工程类型的工程标准。百度AD工程类型的工程标准将在白皮书后续版本中增加。

Server工程类型的工程实践集

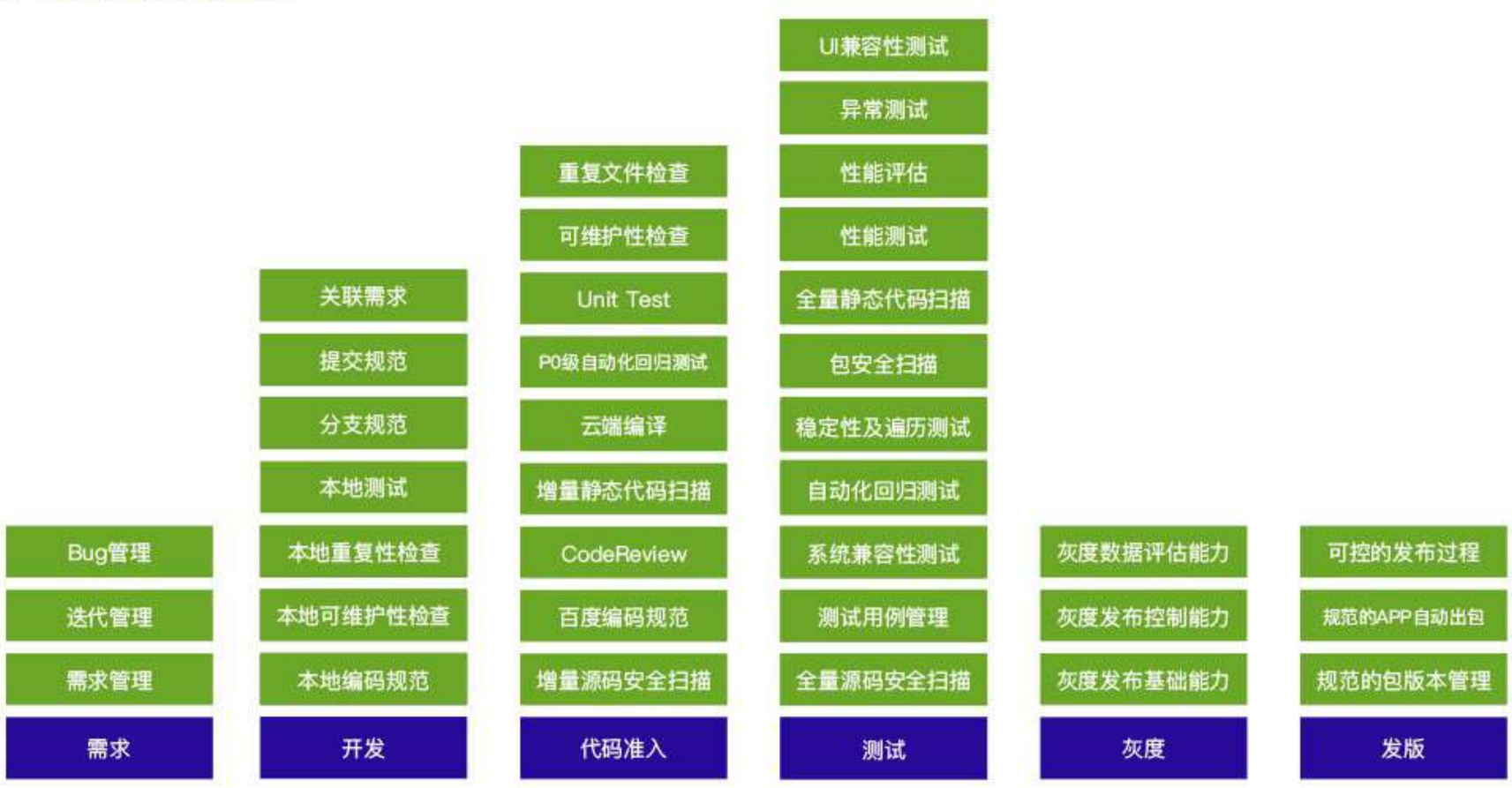
开发一个Server工程，需要经过需求、开发、代码准入、测试、上线&验证等阶段，每个阶段有若干优秀工程实践。下图是百度Server工程类型的优秀实践集合——Server类型工程能力地图。通过工程能力地图，可以指导从开发到上线的流程，建立标准化研发工具链，统一度量工程能力。

SERVER类型工程能力地图

App工程类型的工程实践集

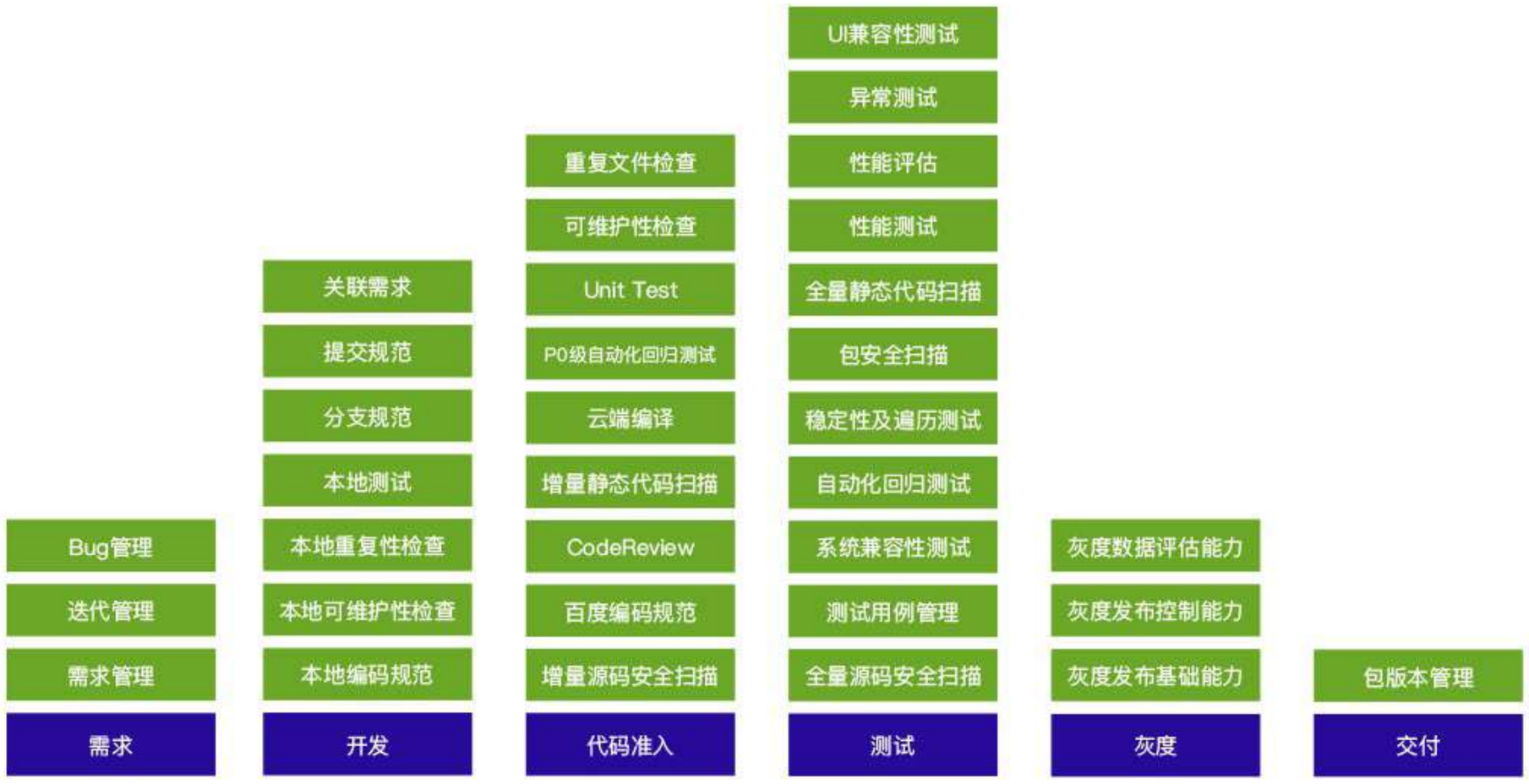
开发一个App工程，需要经过需求、开发、代码准入、测试、灰度、发版等阶段。下图是App类型工程能力地图。

APP类型工程能力地图



SDK工程类型的工程实践集

开发一个SDK工程，需要经过需求、开发、代码准入、测试、灰度、交付等阶段。下图是SDK类型工程能力地图。



部分实践名词解释

容量测试：通过模拟线上真实环境、拓扑、流量等信息，对待上线系统进行测试评估，评估出本次上线预计对线上容量的影响，以便指导运维进行扩容等相关操作。

异常测试：通过模拟系统运行过程中出现的各种异常场景，如网络、硬盘、输入等，测试系统对异常的容错性，提升系统的鲁棒性。

DIFF测试：通过向两个及以上的相同服务（可能不同版本）发送批量请求，分析返回结果，以验证系统可能出现的随机或返回不稳定等问题，或评估新服务策略升级效果等。

性能测试：通过对两个相同服务（不同版本）发送正常qps批量请求，通过分析服务表现，含（日志、网络、CPU、内存）等，用于初步评估升级给服务带来的影响，以便提前发现性能问题。

全量静态代码扫描：通过引用丰富的工具对代码全库进行源码扫描，发现代码可能存在的风险或漏洞。

环境基础能力：需要不断提升被测系统与其线上系统的模拟真实性，含流量、拓扑、机器等。

压力测试：通过发送不同的qps的批量请求，以验证被测系统的抗压能力和稳定性。

服务安全扫描：对web或api类服务进行请求或者服务扫描，以便发现服务中出现的安全漏洞。

测试用例管理：合理的用例管理，可以让同一批人共享用例提升效率。

自动化回归测试：用于对被测系统进行功能逻辑的回归验证，以便提前发现新策略的问题或新代码对历史功能带来的影响。

全量源码安全扫描：通过对代码全库进行扫描，以便发现源码中可能存在的安全漏洞，如显示密码等。

Unit Test：软件中的最小可测试单元进行检查和验证，不用搭建被测系统即可验证代码的逻辑或功能。

P0级自动化回归测试：自动化回归测试的定义如之前所说，P0级是指被测系统的核心功能点。

CodeReview：通过代码走查，发现代码的规范、风险等问题。

百度编码规范：代码符合百度CMC制定的代码规范。

增量源码安全扫描：对当次提交的代码进行源码安全扫描。

增量静态代码扫描：对当次提交的代码进行静态代码扫描。

包管理规范：发布上线包需要规范化、安全存储，也需要符合公司对包描述的管理规范，以便做到可追溯等。

分级部署能力：具备按照流量切分、环境隔离等逐步上线控制能力，供自动检查或快速回滚等场景使用。

自动检查能力：发布上线过程中，具备自动化功能检查、线上系统运行情况监控等能力，在系统最后上线过程中有效发现问题。

本地测试：开发者通过API、脚本等发起功能、单测等相关验证能力，对代码在开发环境中进行验证。

UI兼容性测试：通过各种类型的遍历方法，对遍历过程中的截图进行校验，检查图片可能存在的问题。

性能评估（APP）：通过各类核心case，统计app在case执行过程中的性能表现，以评估升级过程中带来的性能问题。

性能测试（APP）：通过检测代码，发现内存泄露、句柄泄露等风险。

包安全扫描：通过对APP包进行扫描，扫描出包中可能存在的安全性风险。

稳定性及遍历测试：通过对APP页面各类元素进行模拟点击，发现该过程存在的崩溃等问题。

系统兼容性测试：让APP在大批量不同品牌手机、不同系统上进行安装、卸载、执行，发现APP的潜在问题。

灰度发布基础能力：主要是指APP发灰度过程中，有能力根据人群、手机、地域等特点进行灰度。

灰度发布控制能力：主要是指APP发灰度过程中，具备快速止损、控制流量等能力。

灰度发布评估能力：主要是指APP发灰度过程中，能够通过各类驱动收集用户反馈、客户端表现等，全方位快速得到灰度过程中产品问题，最终形成闭环。

百度软件工程实践标准

说明：下面的三级标准里面，数字相关的要求是根据百度历史经验估出来的中位值，会不断持续更新，很多优秀团队的工程实践往往会优于这些值。

需求(Server/App/SDK)

需求管理

- Average　在iCafe中统一管理需求，记录需求内容和相关负责人
- Good　在iCafe中对需求统一定义优先级，并持续更新需求状态
- Excellent　需求拆分粒度大小合适，需求状态停留周期（从离开第一列到到达完成状态）小于5天，并持续更新需求状态

迭代管理

- Average　在iCafe中统一管理计划， 持续通过迭代计划进行有节奏排期，并进行项目进度跟踪
- Good　　迭代周期小于两周，平均计划完成率不低于80%

BUG管理

- Average　Bug记录：在iCafe中统一Bug（含线上）， 记录Bug内容和相关负责人
- Good　　Bug追踪：开启代码提交关联Bug id的开关
- Excellent　Bug跟进：iCafe空间中停留了半年以内的Bug完成率大于80%

开发(Server/App/SDK)

本地可维护性检查

- Average　使用本地检测，提交评审代码之diff部分函数可维护性指数大于50【效果相当于准入阶段的Good，引导前置】
- Good　　使用本地检测，提价评审代码之diff文件部分函数可维护性指数大于80

本地编码规范

- Average　使用本地检测，diff代码部分规范问题全部修复，提交评审之代码diff内容无编码规范问题
- Good　　使用本地检测，diff部分文件规范问题全部修复，提交评审之代码diff文件无编码规范问题

本地重复性检查

- Average　使用本地检测，提交代码不合理文件级重复数量比例低于20%
- Good　　使用本地检测，提交代码不合理文件级重复数量比例低于10%
- Excellent　使用本地检测，提交代码无不合理的文件级代码重复

本地测试

- Average　使用本地代码扫描，diff部分无严重问题
- Good　　进行了两项及以上的本地自动化测试

分支规范

- Average　主干保有完整代码（分支上的修改3个月内须合入主干）

提交规范

- Average 清晰的主干历史（配置了“Push仅含一个commit”或“合入策略=Always Merge” 中的一个）
- Good 线性的主干历史（配置了“Push仅含一个commit”且“合入策略=Rebase if necessary | Fast forward only”，icode开启关联iCafe
- Excellent 95%的commit修改不超过400行

关联需求

- Average 提交值日中包含有效iCafe卡片id（ 配置了"commit message 必须包含 iCafe 卡片 ID"）

代码准入(Server/App/SDK)

百度编码规范

- Average 进行百度编码规范检查，并解决所有检查问题

增量源码安全扫描

- Average 进行增量源码安全扫描，并解决所有安全漏洞(C/C++，OC/OC++删除）

CodeReview

- Average iCode中开启人工评审功能且禁止自评
- Good 过去30天的千行评论数不少于1
- Excellent 过去30天的千行评论数不少于4

增量静态代码扫描

- Average iCode启动检查，千行高危Bug数低于1
- Good 开启阻塞提交，且含Average

云端编译

- Average 正常接入BCLOUD，180天内主线编译成功率大于85%
- Good 依赖使用Stable分支占比不低于70%（C++）

P0级自动化回归测试

- Average 该环节全量覆盖率C/C++：20%；Java、PHP 、Python： 15%
- Good 开启阻塞提交，并且有该环节，含L0
- Excellent 该环节全量覆盖率C/C++：30%；Java、PHP 、Python： 25%

Unit Test

- Average　C/C++：增量行覆盖率30%；Java、PHP ：增量行覆盖率20%
- Good　　开启阻塞提交，并且有UT，含L0
- Excellent 含L1， C/C++:增量行覆盖率60%；Java、PHP ：增量行覆盖率45%

可维护性检查

- Average　增量MI高于20（事后统计不阻碍提交）
- Good　　增量MI高于50
- Excellent 增量MI高于80

重复文件检查

- Average　开启重复文件禁止提交【即使以后做这个拦截，也是经过仔细分析优化之后的一定范围内的拦截，允许合理的重复】

测试（Server）

全量源码安全扫描

- Average　修复全部漏洞（C/C++删除

自动化回归测试(功能、API、schema校验等)

- Average　自动化回归测试+UT，全量分支覆盖率大于40%
- Good　　自动化回归测试+UT，全量分支覆盖率大于65%

测试用例管理

- Average　iCase、ITP或Case代码管理，由工具平台返回
- Good　　具备Case分级手段，由工具平台返回

服务安全扫描

- Average　有安全扫描环节，发布前修复所有漏洞

压力测试

- Average　有压力测试环节，全量分支覆盖率大于30%
- Good　　有压力测试环节，全量分支覆盖率大于40%

环境基础能力

- Average　具备可动态伸缩资源的自动化搭建环境能力
- Good　　具备基于线上硬件环境同质的环境能力，且数据无裁剪
- Excellent 具备部署方式、硬件环境与线上一致的能力

全量静态代码扫描

- Average　进行全量静描，千行高危问题<0.4
- Good　　进行全量静态代码扫描，修复所有高危问题

DIFF 测试

- Average　有DIFF测试环节，并有报告
- Good　　DIFF问题都得到有效跟进，如果有DIFF标记了原因

性能测试

- Average　有性能测试环节，并有报告
- Good　　性能评估指标量，输出至少7个性能结果指标

异常测试

- Average　有异常测试环节，且异常测试场景大于1个
- Good　　异常测试场景大于3个（接口、数据、网络、硬件等）

容量测试

- Average　有容量测试实践环节，并有报告
- Good　　可评估升级对线上的影响和扩容标准

测试（App & SDK）

全量源码安全扫描

- Average　修复全部漏洞（OC/OC++删除)

系统兼容性测试

- Average　有兼容性测试环节，覆盖机型或版本 10%
- Good　　有兼容性测试环节，覆盖机型或版本 40%

自动化回归测试

- Average　自动化回归+UT测试，全量分支覆盖率大于10%
- Good　　自动化回归+UT测试，全量分支覆盖率大于30%

稳定性及遍历测试

- Average　有稳定性测试，activity覆盖大于5%
- Good　　高覆盖稳定性测试，activity覆盖大于20%

测试用例管理

- Average　iCase 、ITP或Case代码管理
- Good　　具备Case分级手段

包安全扫描

- Average　有安全扫描环节，修复全部漏洞

全量静态代码扫描

- Average　进行全量静描，千行高危问题<0.4
- Good　　进行全量静态代码扫描，修复所有高危问题

性能测试

- Average　有返回至少2项测试结果（内存、CPU、耗电、流量）

性能评估

- Average　具备自动化能力，至少返回8项不同场景指标
- Good　　有竞品分析报告

UI兼容性测试

- Average　有UI兼容性测试环节，有至少一款真机任务报告
- Good　　有UI兼容性测试环节，且代码覆盖率或activity覆盖率大于15%

异常测试

- Average　有异常测试环节，且异常测试场景大于1个
- Good　　异常测试场景大于3个（接口、数据、网络、硬件、低电量等）

上线&验收（SERVER）

包管理规范

- Average　使用Agile产品库进行上线
- Good　　产品库含PaaS平台标准描述文件

自动检查能力

- Average　P0 Case检查 + 核心模块监控检查（FATAL，CORE）+核心可用性检查
- Good　　核心业务指标检查+性能检查大于3项
- Excellent　性能退化检查+ 模块级Case检查+端上打点数据检查 + B端客户可用性检查

分级部署能力

- Average　使用了标准的上线变更平台且且模块具备按机房分级发布能力
- Good　　使用公共的PaaS平台部署，具备沙盒环境提供能力
- Excellent　具备按流量比率进行灰度发布能力

灰度（APP&SDK）

灰度发布基础能力

- Average　使用正规的包打包平台和灰度平台进行灰度
- Good　　覆盖人数或覆盖系统占比10%

灰度发布控制能力

- Average　具备灰度发布能力（灵活比例、渠道）
- Good　　具备灰度止损能力

灰度数据评估能力

- Average　发版前新增crash、anr必须修复
- Good　　具备灰度分析报告，指标数不少于6项
- Excellent　用户反馈处理率80%

发版（App）

规范的包版本管理

- Average　使用包管理平台进行产品包管理
- Good　　可以方便的获取历史包

规范APP自动出包

- Average　在包管理平台中制作渠道包
- Good　　在流水线中全自动制作渠道包

可控的发布过程

- Average　具备控制渠道的止损能力
- Good　　具备线上热修复能力

交付（SDK）

包版本管理

- Average　使用Agile产品库进行发布和版本管理

流水线/自动化 (Server/App/SDK)

执行效率

- Average　模块总构建时间/总构建数Average: (1H, 2H]
- Good　(0.5H, 1H]
- Excellent　<=0.5H

失败恢复时间

- Average　　失败转为成功时间间隔　(0.75H, 1.5H]
- Good　(0.5H, 0.75H]
- Excellent　<=0.5H

异常构建率

- Average　　失败构建数 / 总构建数　[0.2,0.3)
- Good　[0.1,0.2)
- Excellent　[0,0.1)

百度软件工程标准的实施

实施方式

上述的工程标准有助于团队了解工程全貌，自主确定工程改进目标，自助改进工程能力。每一个工程实践都有对应的工程工具平台支持，团队只需在研发工具中选择对应的工具插件，即可获得相应的工程能力。例如在“代码准入”阶段中的若干工程实践，都可以从百度代码管理工具iCode中进行选择。

在“测试”阶段中的工程实践，则可以从百度流水线平台iPipe中选择相应插件，下图为百度研发流水线iPipe的部分插件

验证方式

团队在使用工程工具、按照工程标准进行工程实践的过程中，这些工程工具平台能够按照统一规范回传数据，进而可以分析计算每个团队在每个工程实践上已经达到的级别。为此我们用研发工具平台产生的研发现场数据，建立了百度研发数据仓库，并基于这些工程数据，开发了工程能力地图可视化的产品，方便各个团队了解自身的工程能力，验证工程改进的效果。

实施效果

通过不断修订工程标准、迭代工程工具落地优秀实践、收集研发数据并可视化工程能力，可以使研发团队快速实施优秀工程实践，提升工程能力，进而提高团队开发效率，提高产品质量。 我们观察了公司8个样本团队的工程实践落地进程和开发周期时间的变化。其中4个团队在观察期内按照本文的工程标准推进了若干工程实践的落地，另外4个团队工程实践做得很少且在观察期内没有推进新的工程实践落地。把这8个团队在观察期的每个版本的开发周期时间和周期时间的移动平均值用散点图进行可视化，得到下图。

我们发现在选定的时间范围内（横坐标），推进工程实践落地的4个团队（图中高亮的颜色），其开发周期的移动平均值（下图的散点）成稳定下降趋势；没有推进工程实践落地的4个团队（图中背景部分），其开发周期主要成发散上升趋势。同时这8个团队在观察期内的团队人数、开发习惯等团队特征以及产品质量均没有显著变化。这说明工程实践能够对开发周期的缩短产生积极影响。

为了在更大范围内进一步发现工程标准、工程工具对研发效率和产品质量的影响，我们对百度所有团队6个月时间段内的研发数据进行了定量分析。分析结果表明，团队采用的工程实践数量越多，其开发周期越短；工程实践做得程度越深入的团队，其开发周期也越短。同时研究了开发团队人数对上述结果的进一步影响，发现团队人数越多，实施工程实践对缩短开发周期的作用就越大。以上分析结论进一步佐证了优秀的软件工程实践能够缩短开发周期。

未来与展望

对于一个科技公司来说，保持并不断提升工程能力是最重要的。百度一直坚持培养工程师的工程素养，建设高效的工程工具，持续推动工程能力提升，Relentless Pursuit of Engineering Excellence，永无止境的追求工程卓越。本白皮书介绍了目前百度工程的标准和工程实践，希望作为软件工程的实践者为同行或研究机构提供一个参考样本，促进交流，共同提高。百度工程标准也必然会随着公司内外工程技术的不断革新而持续升级。百度软件工程标准希望及众家之所长，合众人之智慧，以实践为检验标准，促工程能力打造科技产品，用科技让复杂的世界更简单。