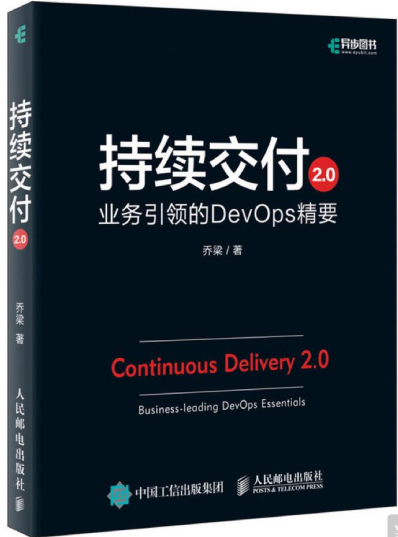




十倍速原则 对工程生产力建设的 方向性影响

乔梁
2019.05 北京



¥ 66.80

签名版 | 持续交付2.0 业务引领的DevOps精要



扫描或长按识别二维码

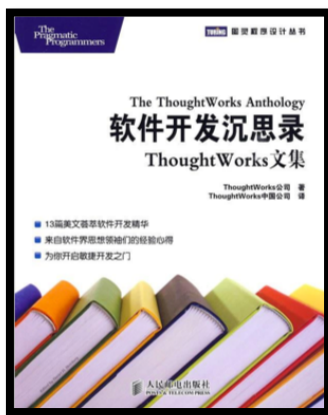
极客邦科技 会议推荐2019



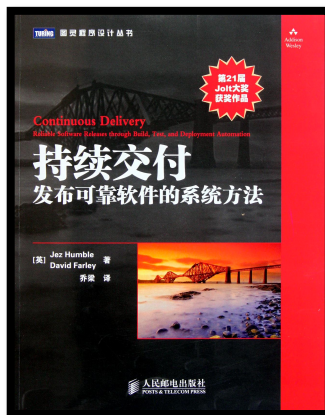


乔梁

敏思特咨询
组织转型 & 研发管理
国内首位持续交付&DevOps布道师



2009



2011



2018

持续交付2.0



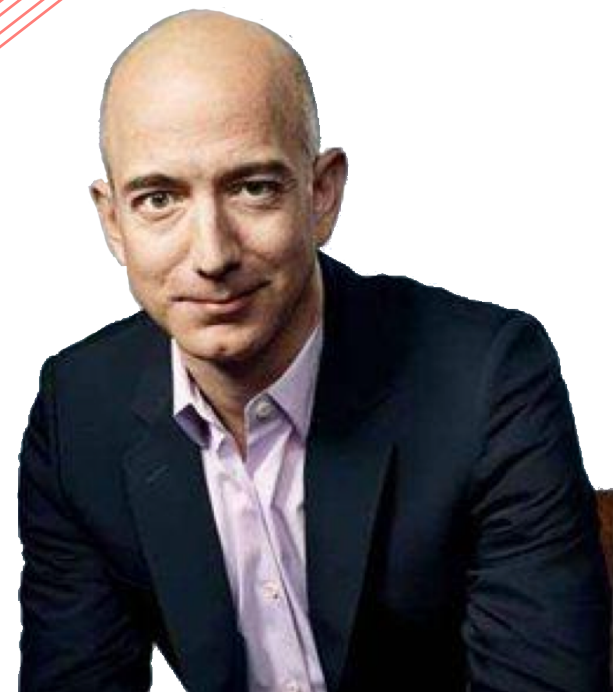


一万次实验法则

“

亚马逊的成功是每年、每月、每周、每天进行多次实验的结果。

”



“

我最为自豪的事情之一是我们成功的关键在于试验框架.....

在任何时候，都不只有一个Facebook版本正在运行，而是一万个左右

”



一万次实验法则

“

亚马逊的成功是每年、每月、每周、每天进行多次实验的结果。

”

Data Inform

“

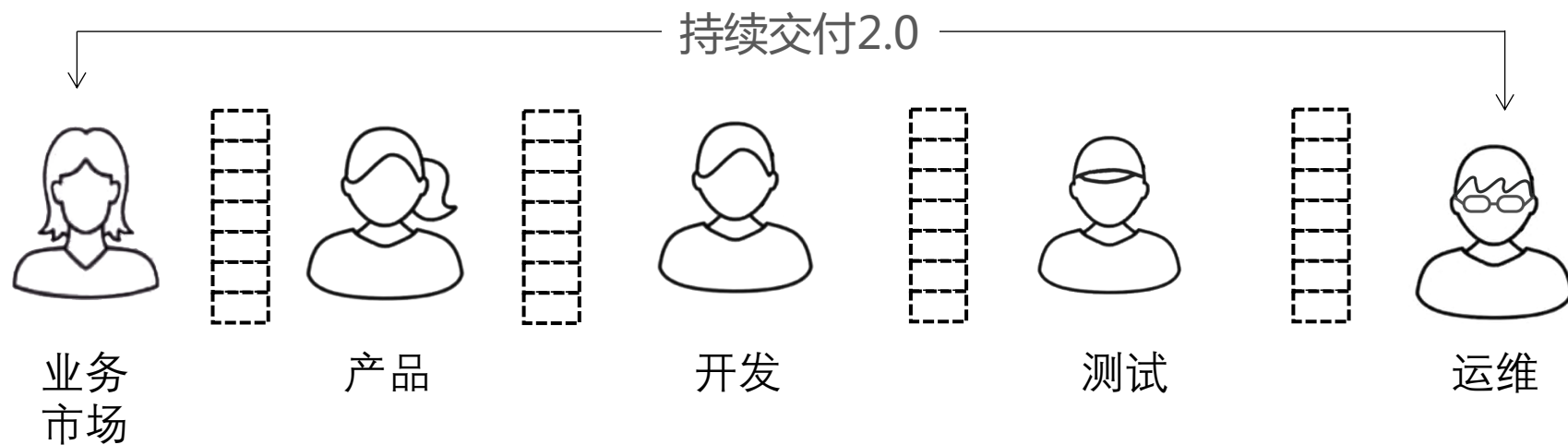
我最为自豪的事情之一是我们成功的关键在于测试框架.....

在任何时候，都不只有一个Facebook版本正在运行，而是一万个左右

”



端到端的业务协作

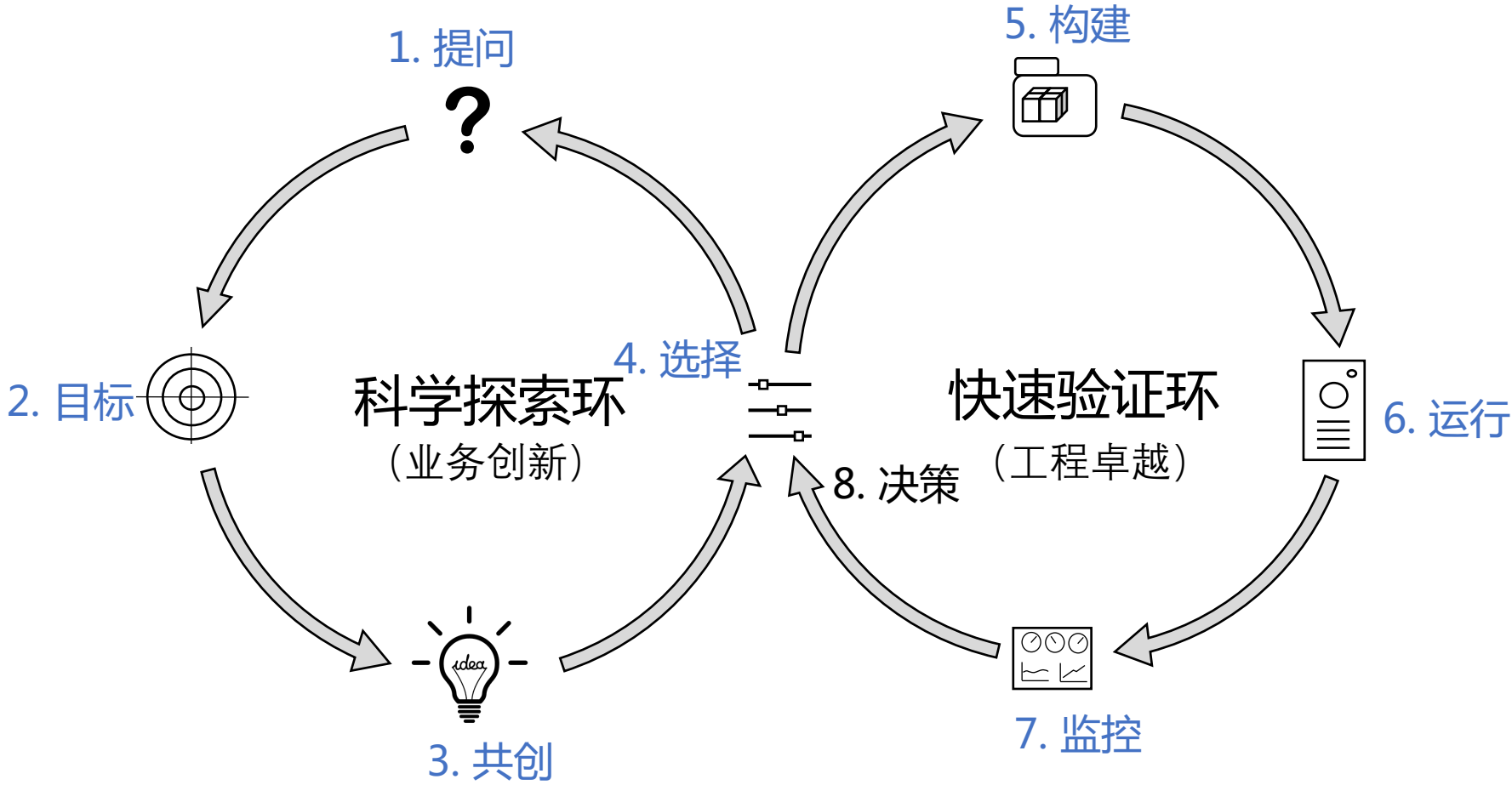


持续交付2.0：
是一种组织能力，应具有可持续性
高质量、低成本、无风险地快速交付服务，提供业务价值



持续交付2.0之双环模型

——硅谷顶级公司的产品研发理念

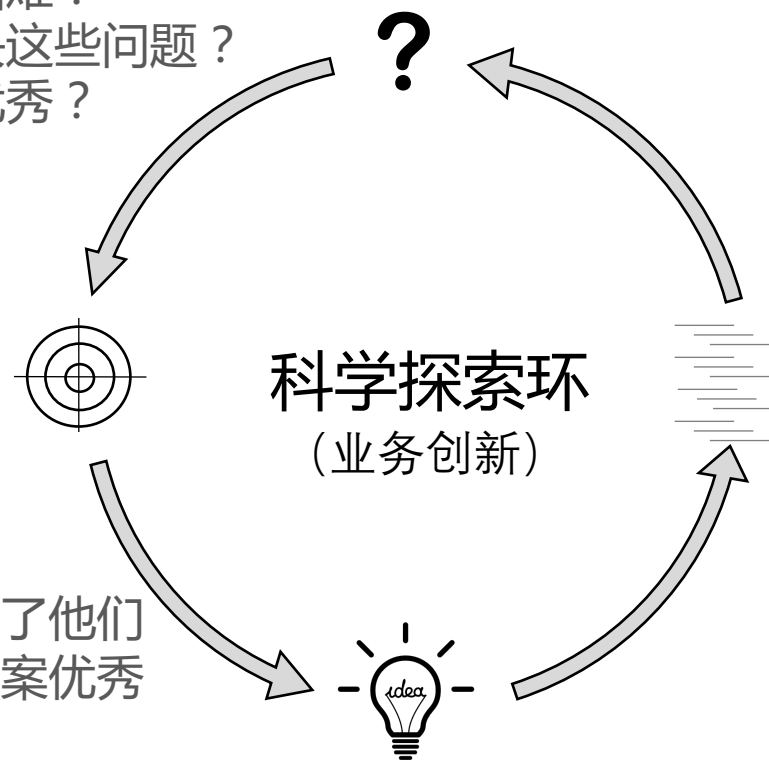




持续交付2.0之双环模型

——硅谷顶级公司的产品研发理念

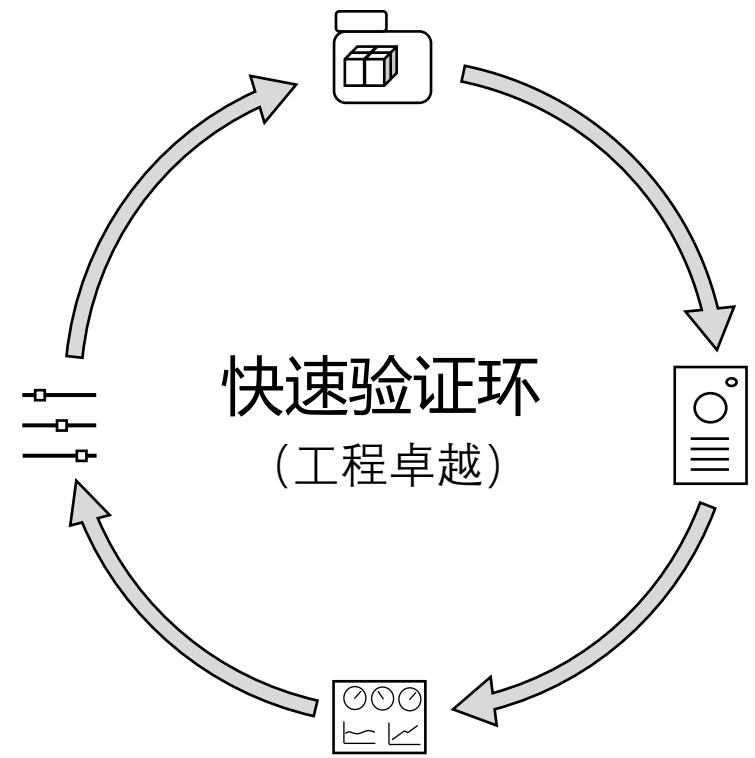
1. 客户在哪里？
2. 他遇到了哪些困难？
3. 他现在怎么解决这些问题？
4. 你的方案有多优秀？



1. 如何衡量你找到了他们
2. 如何衡量你的方案优秀

1. 如何找到他们
2. 你的问题有多少种解决方案
3. 它们的投入和产出如何评估

确定其中一部分，快速验证





工程生产力

1. 客户在哪里？
2. 他遇到了哪些困难？
3. 他现在怎么解决这些问题？
4. 你的方案有多优秀？



工作流程



科学探索环
支撑工具



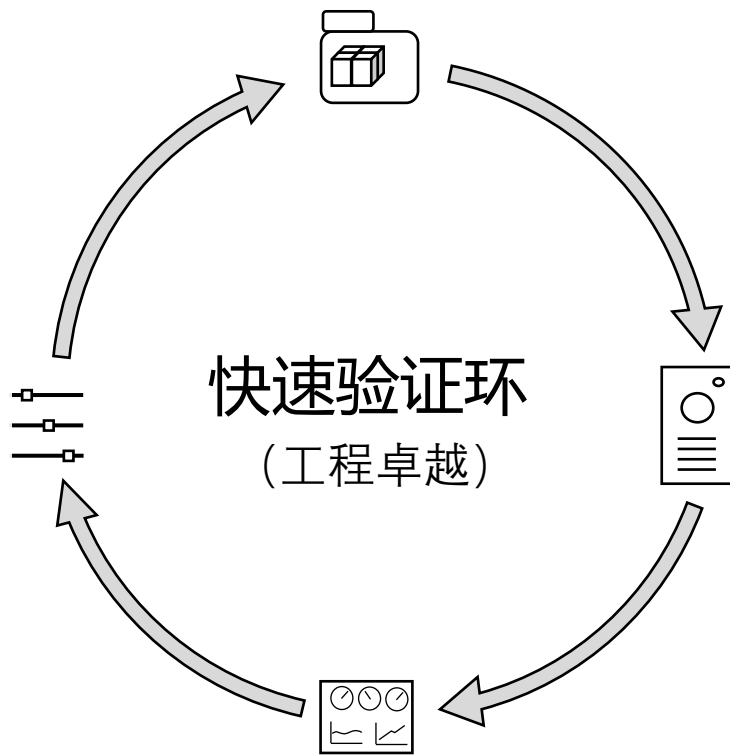
工程素养

确定其中一部分，快速验证

1. 如何衡量你找到了他们
2. 如何衡量你的方案优秀



1. 如何找到他们
2. 你的问题有多少种解决方案
3. 它们的投入和产出如何评估





十倍速原则

如果你希望自己的车子能达到50英里的时速，
可能只要对车子稍加改造即可。

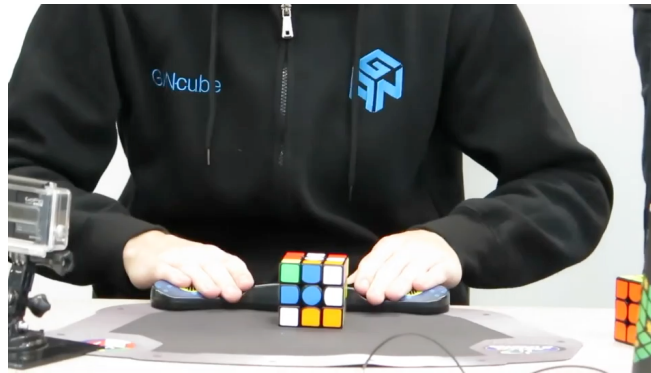
然而，如果让它只用1加仑汽油跑500英里，就要对它重新设计了。

——阿斯特罗·泰勒，Google X



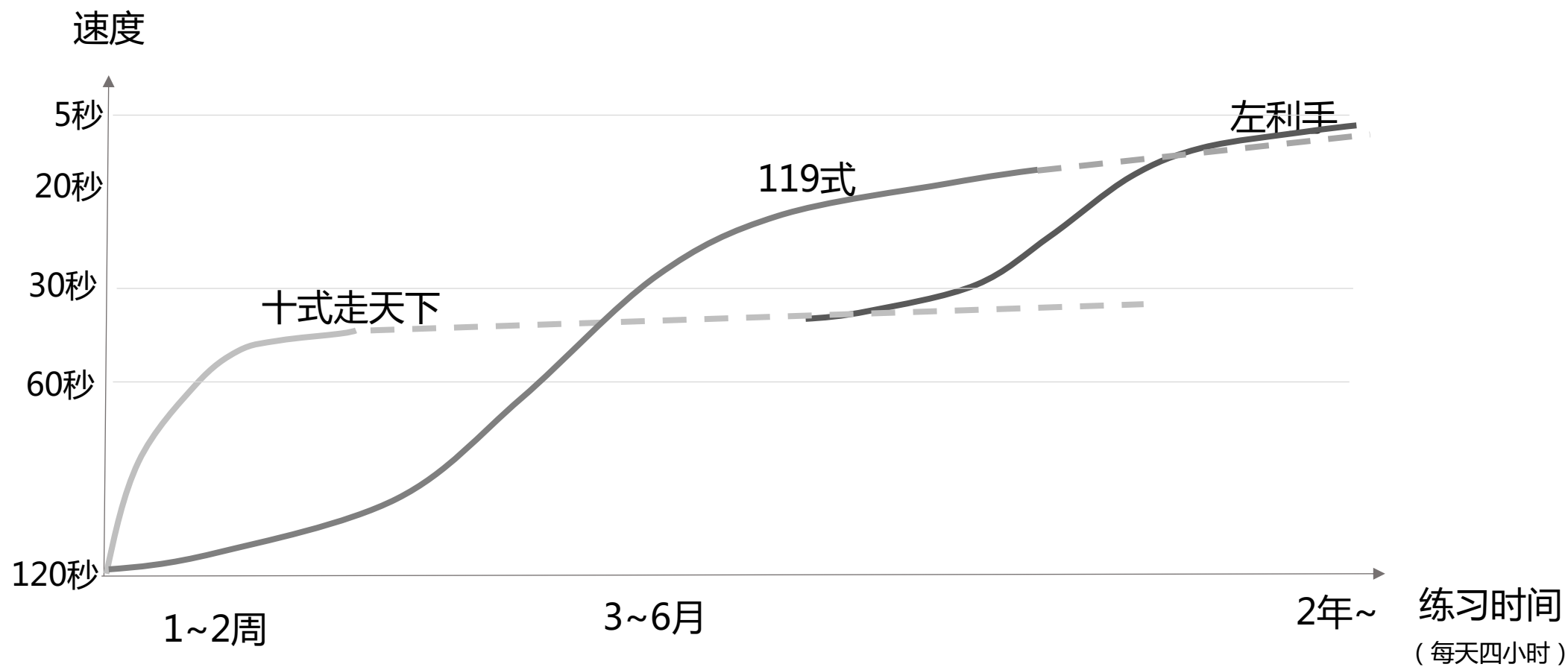


十倍速原则





十倍速原则





十倍速原则

工作流程





每年12 次发布（理想状态）

星期日	星期一	星期二	星期三	星期四	星期五	星期六
					Oncall 任务交接	
	手工	测试...				
	发布	Oncall 任务交接				



每年12 次发布（残酷的现实）

星期日	星期一	星期二	星期三	星期四	星期五	星期六
					匆忙提交 半成品	匆忙提交 半成品
Oncall 任务交接	手工	测试			最后一分钟 提交关键功能	
	延迟发布	Oncall 任务交接	手工	测试		最后一分钟 发现缺陷
紧急修复	发布	Oncall 任务交接				



不断重复你一贯的做法，
你当然只会得到与以前一样的结果！

只有达到远超要求的水平，
才能轻松实现被要求的结果！



提高对自己的要求

1-1-1

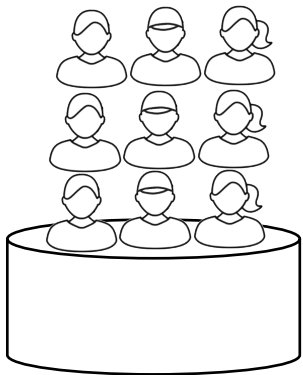
每月1个正式版本

每周1个RC版本

每天1个Alpha版本



流程再造



需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度



发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度



评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

需求评审 > 开发 > 测试 > 灰度

RC 全量

RC

RC

RC

RC

全量





每天两次发布

星期日	星期一	星期二	星期三	星期四	星期五	星期六
	Alpha	Alpha	Alpha	Alpha	RC	
	Alpha	Alpha	Alpha	Alpha	RC	
	Alpha	Alpha	Alpha	Alpha	RC	
	Alpha	Alpha	Alpha	Alpha	PR	
	Alpha	Alpha	Alpha	Alpha	RC	
	Alpha	Alpha	Alpha	Alpha	RC	



工具支持，规则简化

基础设施



管理冲突



管理质量

简单原则



十倍速原则

支撑工具





支撑工具的特点

- 标准化
- 高要求
- 简单易用



工具，更高层次的交互



Bazel

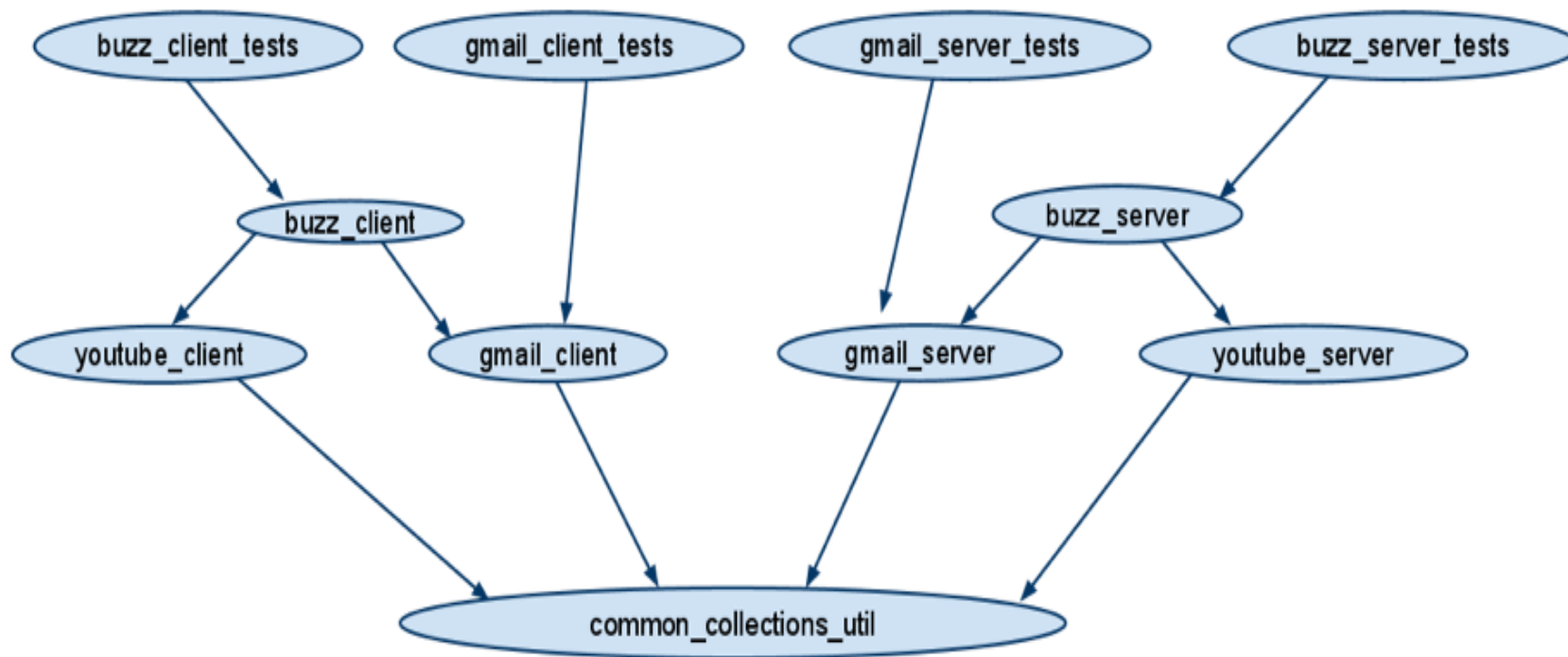
- 支持大多数编程语言
- 统一格式
- 细粒度的项目描述
- 不允许多依赖

the prevailing attitude for a while that the existing open source projects are good enough

Looks like they wized up! Bazel for the JVM has a build language that's readable (unlike Maven's XML) and lightweight (unlike Gradle's Groovy). Too bad there's no Windows edition, tho.

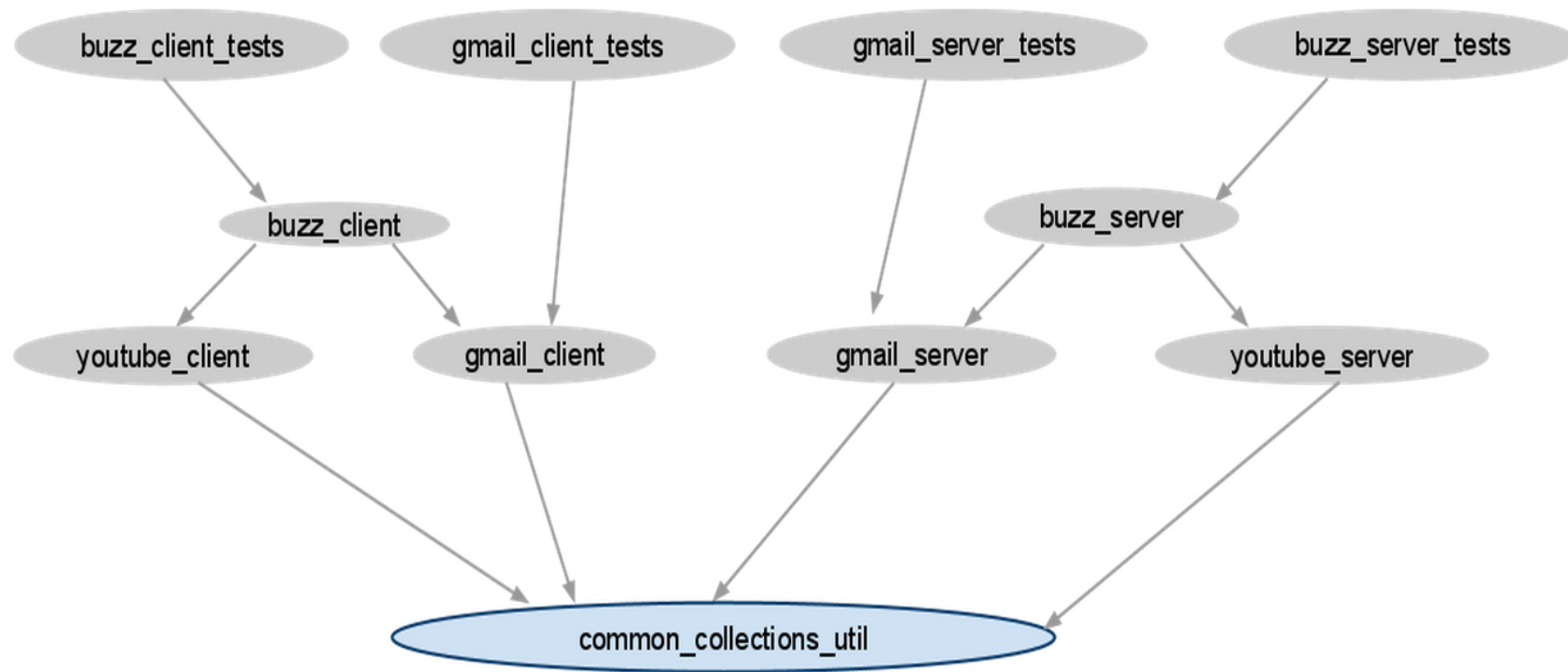


依赖管理



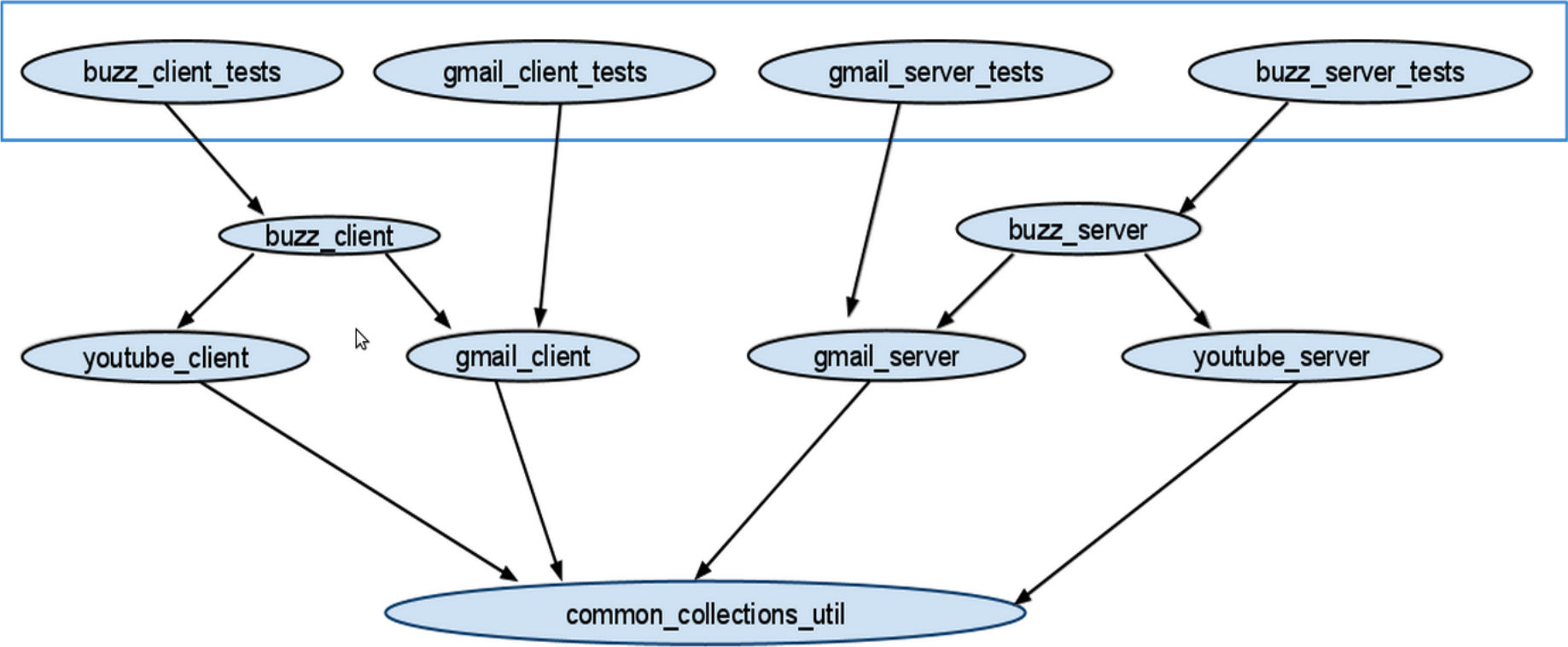


场景一



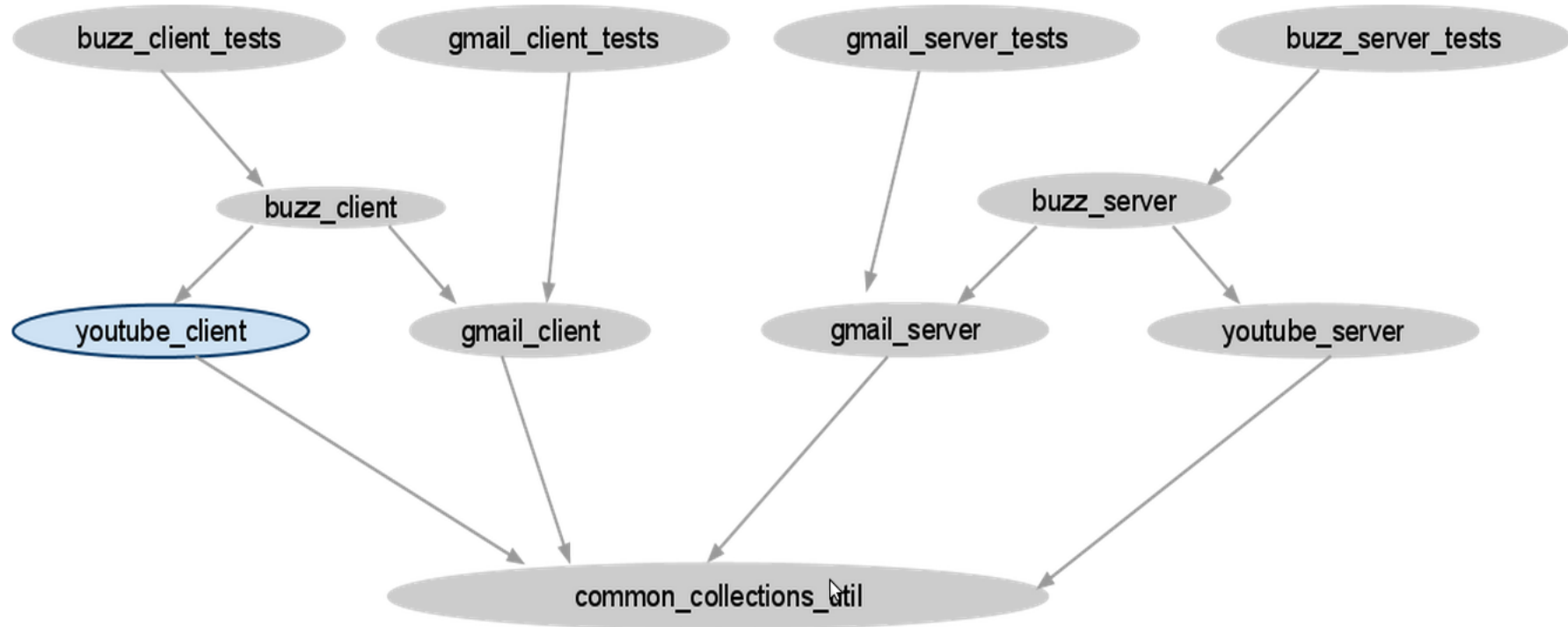


All tests are affected! Both Gmail and Buzz projects need to be updated



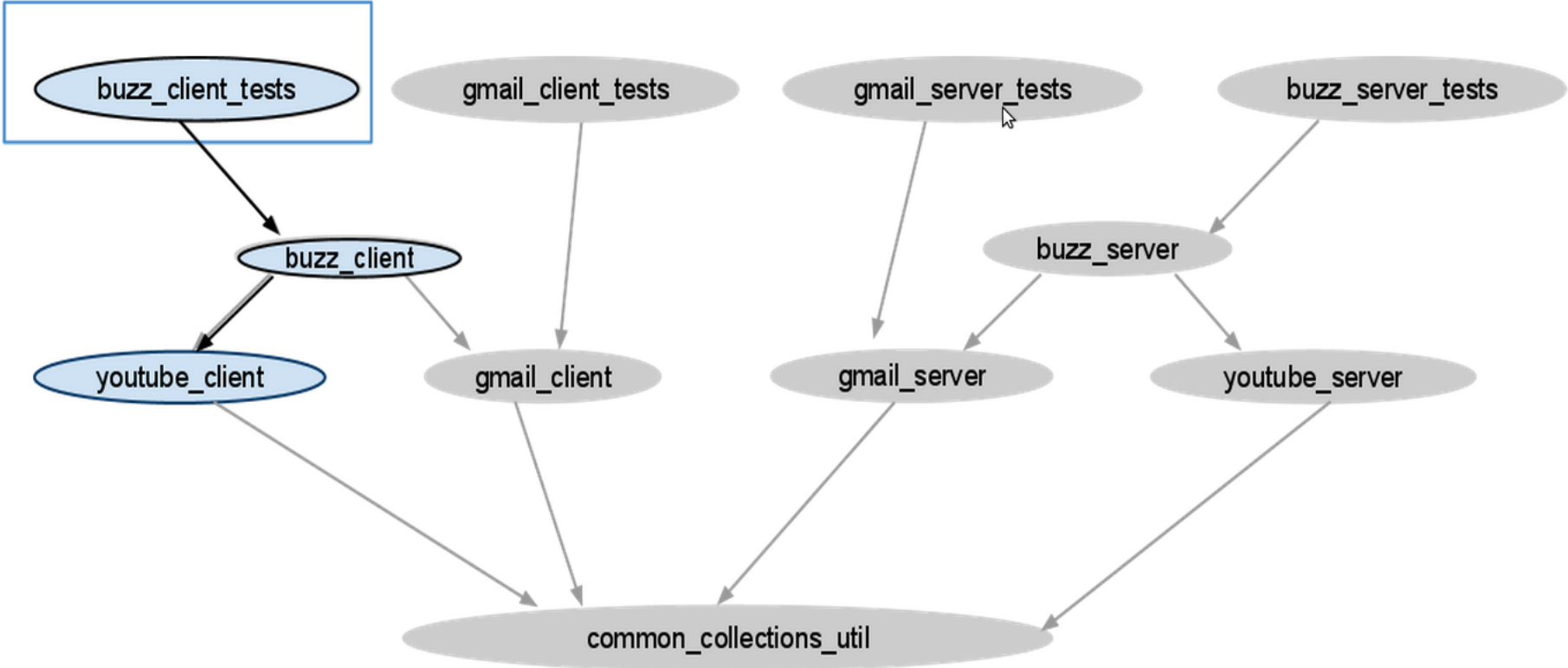


场景二



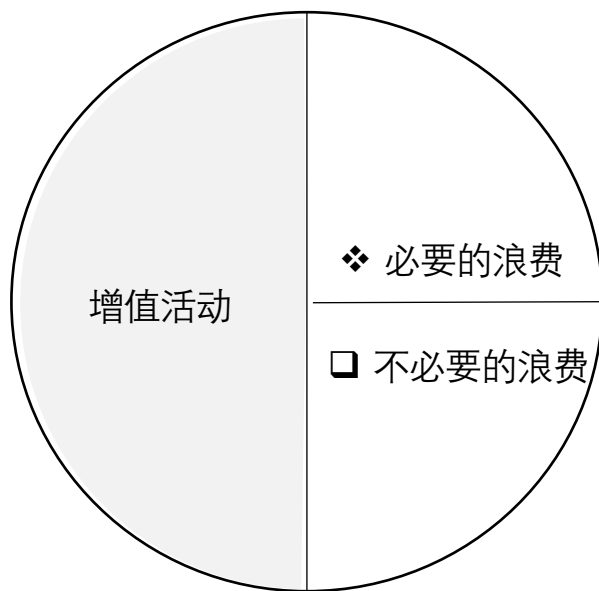


Only `buzz_client_tests` are run and
only Buzz project needs to be updated.

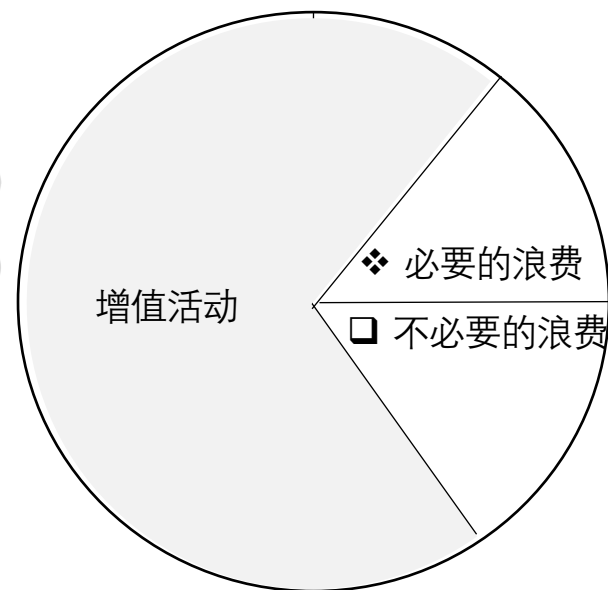




十倍速原则下的收益



- 无用的功能
- 等待（等待其他人的支持或帮助、审批）
- ❖ 工作项的移交(Handoffs)
- 库存（等待开发、等待测试、等待审批）
- ❖ 部分完成的工作（WIP）
- ❖ 寻找信息（上下文切换，反复沟通）
- 软件缺陷（Defects）
- ❖ 测试
- ❖ 项目管理
- ❖





为什么google bazel构建工具流行不起来

作者Jack47

转载请保留作者和原文出处

之前博主写了系列文章Google软件构建工具Bazel原理及使用方法介绍。最近使用了一段时间后，觉得这个东西不是一种通用的构建工具，⁽¹⁾很难对接到情况复杂的大的工程项目里。今天来讲讲为什么博主觉得google bazel构建工具⁽²⁾不会大规模流行起来。本文是参考Gradle--另外一个很出名的构建工具--的人对bazel的看法的这篇文章基础上整理而成的





bazel为什么不适合其他公司

大部分公司都不是Google的这种情况，⁽¹⁾ 不会有强大的构建团队来专门做软件构建相关的工具，⁽²⁾ 也不会有Google那么好的开发文化，⁽³⁾ 没有统一的构建工具，因此大项目想对接bazel，会遇到很多困难。

Google 内部软件构建系统已经开发了超过7年了，所以开发适合Google自己的构建工具是他们的初衷。Google构建工具跟其他工具相比，强调的是“更加结构化”，这样能够提高并发度，拥有更好的可重现性[reproducibility]。但这样对源代码的组织方式和头文件的书写规则，库之间依赖关系的书写要求就



十倍速原则



工程师能力与素养



谷歌工程生产力的三个支柱

The Three Pillars of Hackability

There are a number of tools and practices that foster hackability. When everything is in place, it feels great to work on the product. Basically no time is spent on figuring out why things are broken, and all time is spent on what matters, which is understanding and working with the code. I believe there are three main pillars that support hackability. If one of them is absent, hackability will suffer. They are:



Code Health



Debuggability



Infrastructure



Code Review

我们

- 只有重要需求才做Review
- 紧急情况时不做Review
- 各团队规范不一致

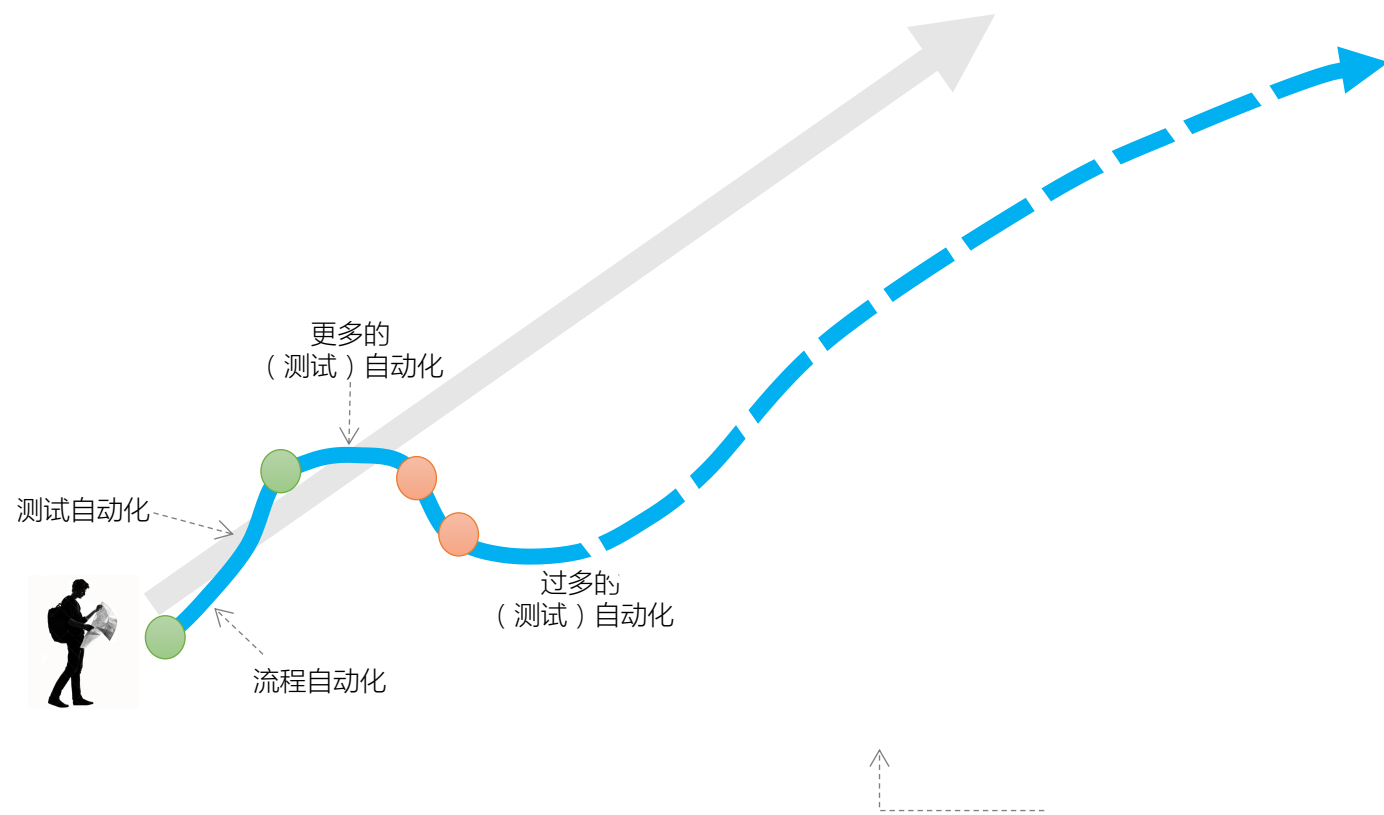
他们

- 每行代码都要Review
- 代码规范全部一致
- 必须具备可读性资质



我们常见的是

- 大量Copy & Paste的代码
- 无法写出自动化测试的代码
- 箭头式的代码
-





十倍速原则

- 流程方面：以终为始，具有颠覆性，才能大飞跃
- 工具方面：高标准下的简单易用，才能创造更大价值
- 能力素养：有了时间，才能学习，提高工程能力与素养

TGO 鲲鹏会

汇聚全球科技领导者的高端社群

📍 全球12大城市

👤 850+ 高端科技领导者

使命
Mission

为社会输送更多优秀的
科技领导者

愿景
Vision

构建全球领先的有技术背景
优秀人才的学习成长平台



扫描二维码，了解更多内容

想做团队的领跑者 需要迈过这些“槛”

成长型企业，易忽视人才体系化培养
企业转型加快，团队能力又跟不上

VS

从基础到进阶，超100+一线实战
技术专家带你系统化学习成长

团队成员技能水平不一，
难以一“敌”百人需求

VS

解决从小白到资深技术人所遇到
80%的问题

寻求外部培训，奈何价更高且
集中式学习

VS

多样、灵活的学习方式，包括
音频、图文 和视频

学习效果难以统计，产生不良循环

VS

获取员工学习报告，查看学习
进度，形成闭环



课程顾问「橘子」

回复「QCon」
免费获取
学习解决方案

极客时间企业账号 # 解决技术人成长路上的学习问题



Q&A

持续交付2.0公众号



签名版 | 持续交付2.0 业务引领的DevOps精要



扫描或长按识别二维码