

1

统一开发环境/协作工具

为什么要统一开发环境/协作工具

- 你知道开发环境指的是什么吗？
- 你所了解的协作工具有哪些？
- 你们之前开发环境都选用过哪些呢？
- 为什么要对它们进行统一呢？



开发环境
Development

测试环境
Testing

“仿真预演”环境
Staging

生产环境
Production

软件环境

往往创建在开发者的设备上
软件在这个环境下开发
一个沙盒环境

开发环境
Development

“仿真预演”环境
Staging

生产环境前最终验证
这一环境尽可能的仿真了真实的生产环境
另一个关键是用用于性能测试，特别是加载时间

允许人类测试者采用自动化或非自动方式进行测试
如果测试失败，测试环境会将这部分
并通知对应开发者

测试环境
Testing

生产环境
Production

也被称为线上环境
向生产环境部署是最为敏感的步骤
生产环境出现问题会导致严重的后果

开发环境

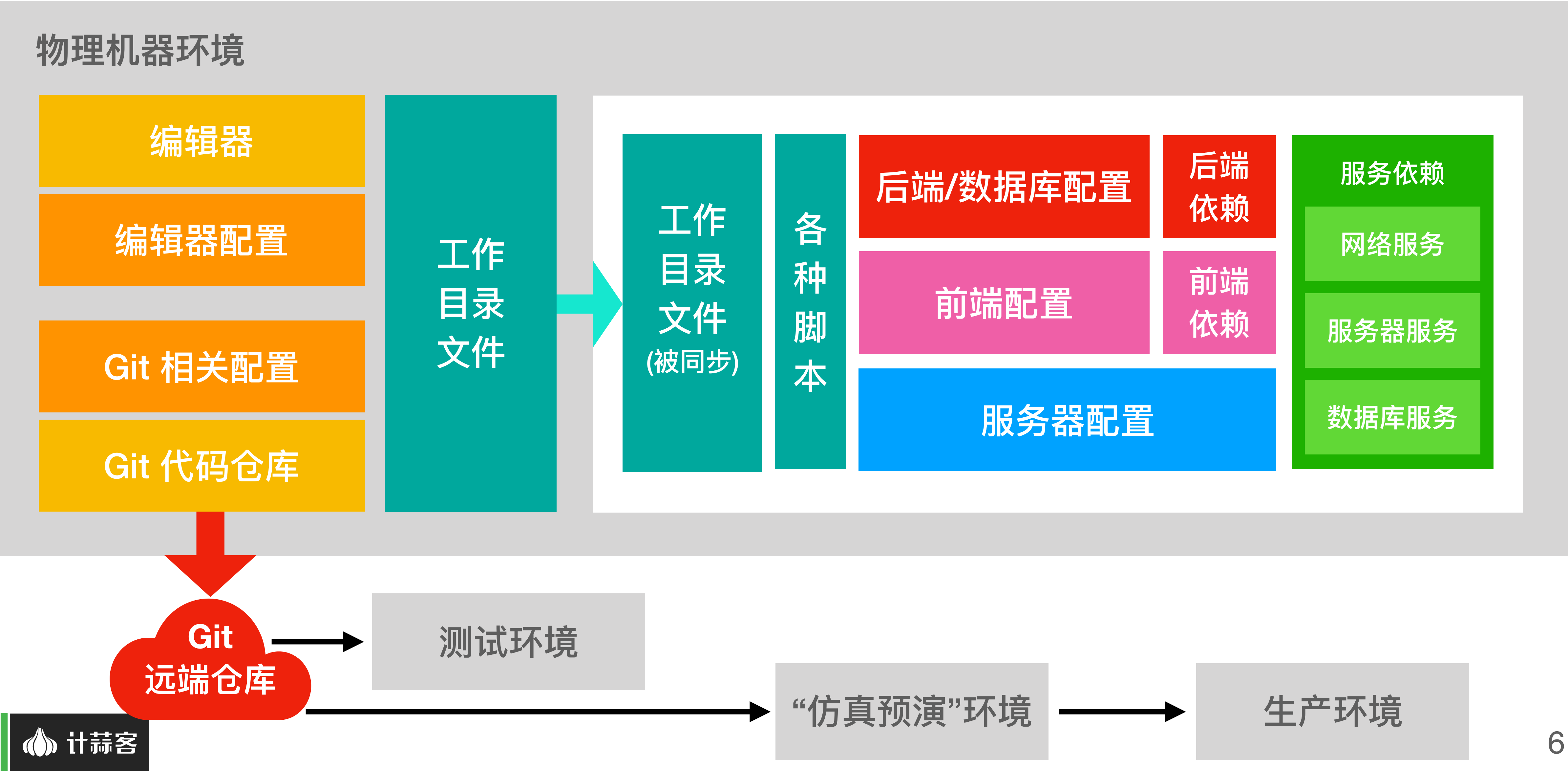
01 | 工程运行环境

02 | 开发依赖环境

03 | 开发工具/编辑器

04 | 配置文件

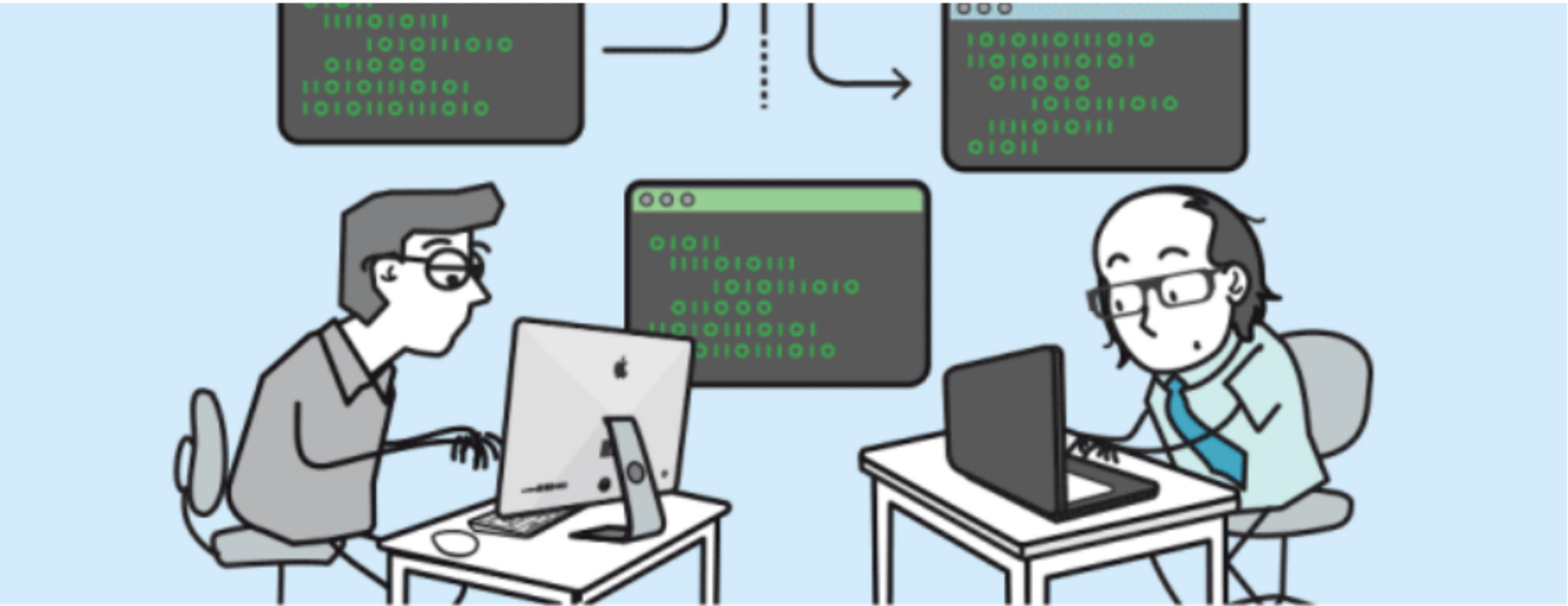
开发环境 - 以我们的配置为例



为什么我们要写脚本

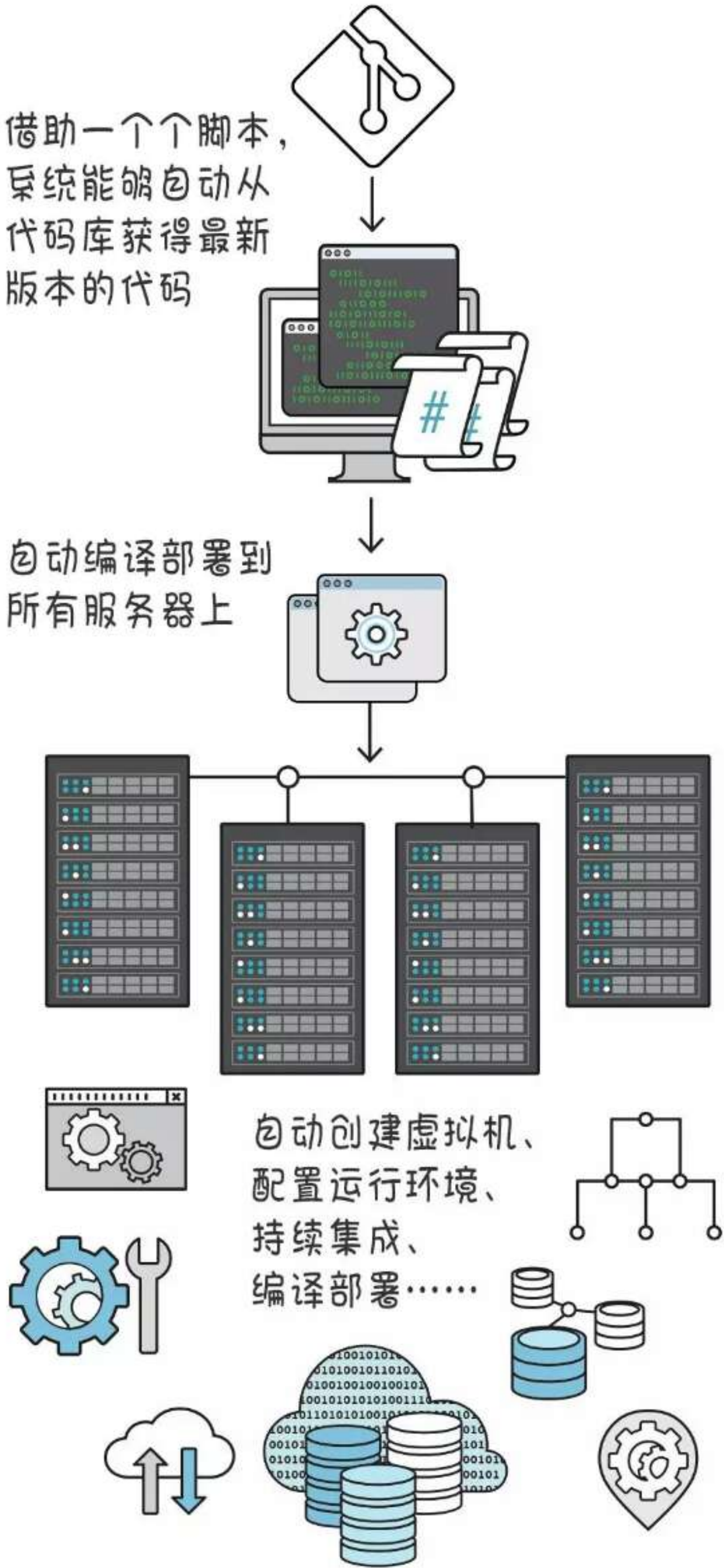
为什么程序员们喜欢写脚本？

9月23日



《神秘的程序员们》

重复200次只需花2小时的事，程序员会花5个小时写个脚本来解决。这是一种病吗？为什么程序员们爱写脚本？我们要不要消灭这些“破坏世界公平”的“坏码农”？



就这样，在技术世界里，只需更少的人力就能支持更大规模的应用。

让更多人有机会以更低的成本和门槛来实现自己的产品和梦想

现实世界因此被推动和改变……

嘿嘿，这就是为什么我们程序员喜欢写脚本啦~~



协作工具

01

版本管理工具



02

工单追踪

03

提交审计



04

开会工具



05

依赖管理工具



06

提交检查



07

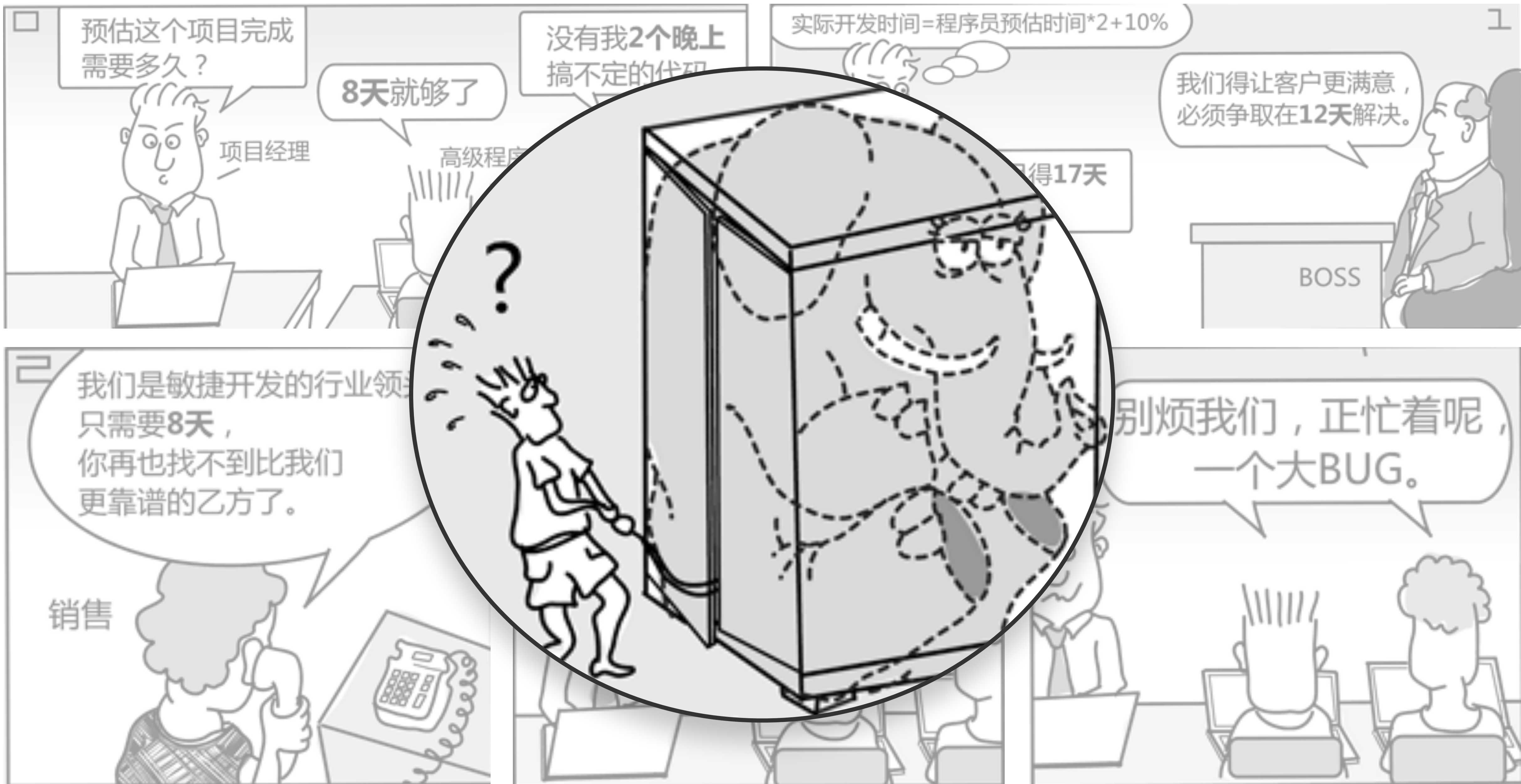
文档工具



2

工程管理与 workflow





“软件人员太像皇帝新衣故事中的裁缝了。当我来检查软件开发工作时，所得到的回答好象对我说：我们正忙于编织这件带有魔法的织物。只要等一会儿，你就会看到这件织物是极其美丽的。但是我什么也看不到，什么也摸不到，也说不出任何一个有关的数字，没有任何办法得到一些信息说明事情确实进行得非常顺利，而且我已经知道许多人最终已经编织了一大堆昂贵的废物而离去，还有不少人最终什么也没有做出来。”

-- F.P. Brooks, Manager of OS/360 “The Mythical Man-Month”, 1974

“...没有别的场景比巨兽在焦油坑中垂死挣扎的场面更令人震撼...它们挣扎得越是猛烈，焦油纠缠得越紧，没有任何猛兽足够强壮或具有足够的技巧，能够挣脱束缚，它们最后都沉到了坑底。

过去几十年的大型系统开发就犹如这样一个焦油坑，很多大型和强壮的动物在其中剧烈地挣扎...各种团队，大型的和小型的，庞杂的和精干的，一个接一个淹没在了焦油坑中...”

-- F.P. Brooks, Manager of OS/360 “The Mythical Man-Month”, 1974

在软件项目管理中我们关心什么

- 需要关注钱的问题吗?
- 需要关注时间吗?
- 还需要关注什么呢?



在软件项目管理中我们关心什么

- 01 | 在预算之内
- 02 | 在计划时间之内
- 03 | 工程质量良好

- 这个功能为什么要被开发？
- 什么时候做，什么时候完成？
- 谁负责来实现？
- 这个功能被用于哪里？
- 从技术和管理上这个功能可以怎么被完成？
- 有哪些资源应该被需要到？

- Umbrella activities
 - 测量 & 数据矩阵
 - 预估
 - 风险
 - 排期
 - 追踪与控制



“to enhance the readiness of software organizations to undertake increasingly complex applications by helping attract, grow, motivate, deploy, and retain the talent needed to improve their software development capability”

B. Curtis 1994

关键：寻找、筛选、效率管理、训练、职业发展、组织设计、团队与文化发展

- “并非每一群人都是一个团队；也并不是每个团队都很有效率”
- 如何高效率的工作：
 - 互相之间要信任
 - 大家的能力要互补，满足解决问题所需
 - 对于一些与团队不太相合的人，要排除在团队之外

- 解决问题
 - 找到最为紧迫的技术与管理问题
 - 系统性的找出一个解
 - 从过去的经历中总结出经验
 - 灵活的调整解
- 管理角色： 在项目中起到管理作用
- 赏罚分明： 给予合适的奖励与成就感
- 团队内影响力：“读懂人”、对团队成员的需求进行响应

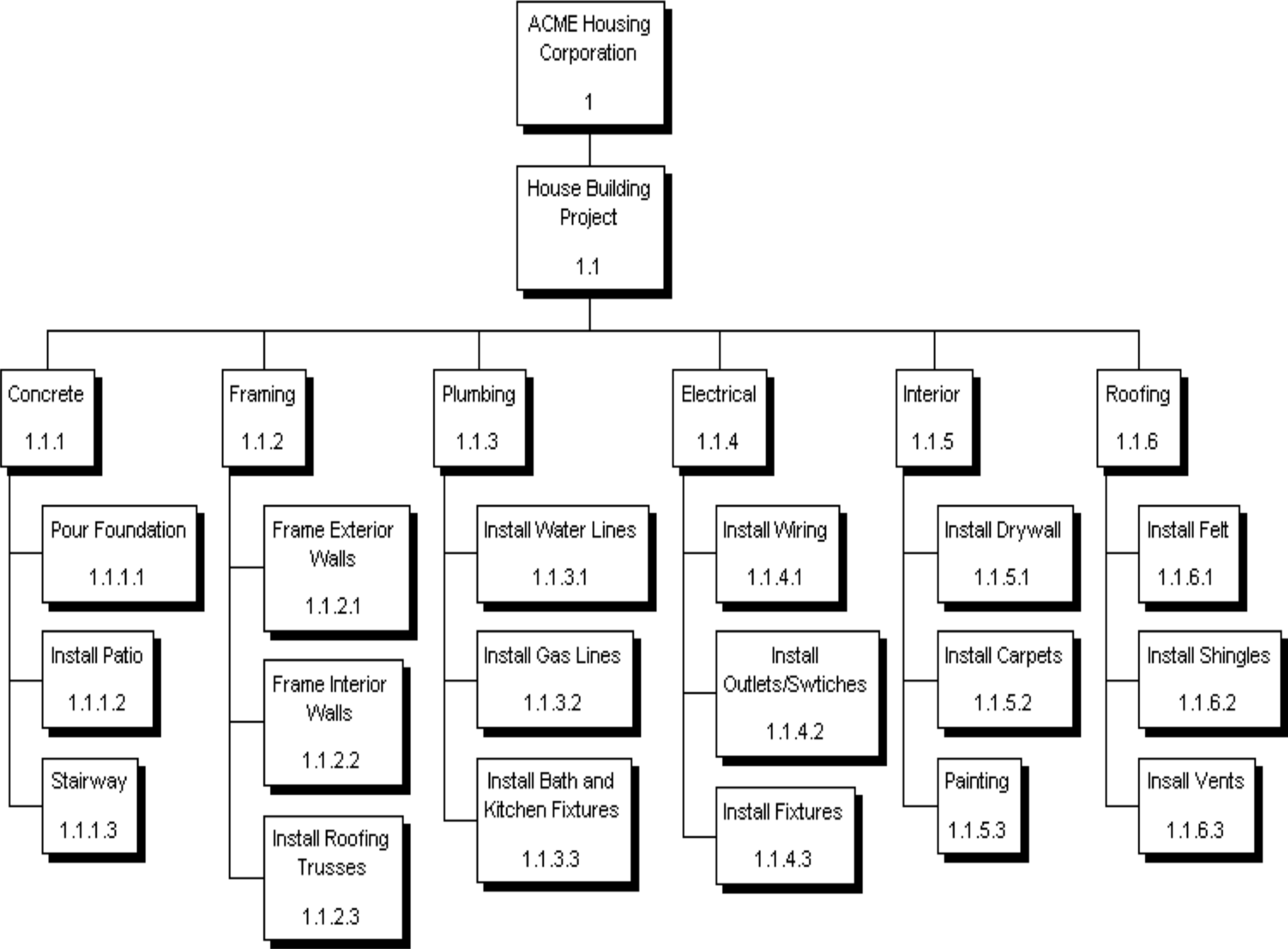


- 最大化团队的产出
- 好的团队带头人的特征：
 - 动机：激励团队成员主动做到最好
 - 组织：能将一个想法推成最终的成果
 - 创新：让团队成员去创造，并且有意愿创新

- 项目的体量很大、很复杂，团队成员之间可能很难协调。
- 建立有效的协调方式需要处理好以下问题：
 - 客观的沟通
 - 涉及到多人的流程
 - 不见面的沟通表达
 - 人际网络的长期维护

- 将任务拆解为多个部分
 - 产品
 - 过程
 - 组织
- 自上而下的拆解
 - 考虑全局的分析与组织
 - 确保没有遗漏
 - 考虑现实性

一个例子

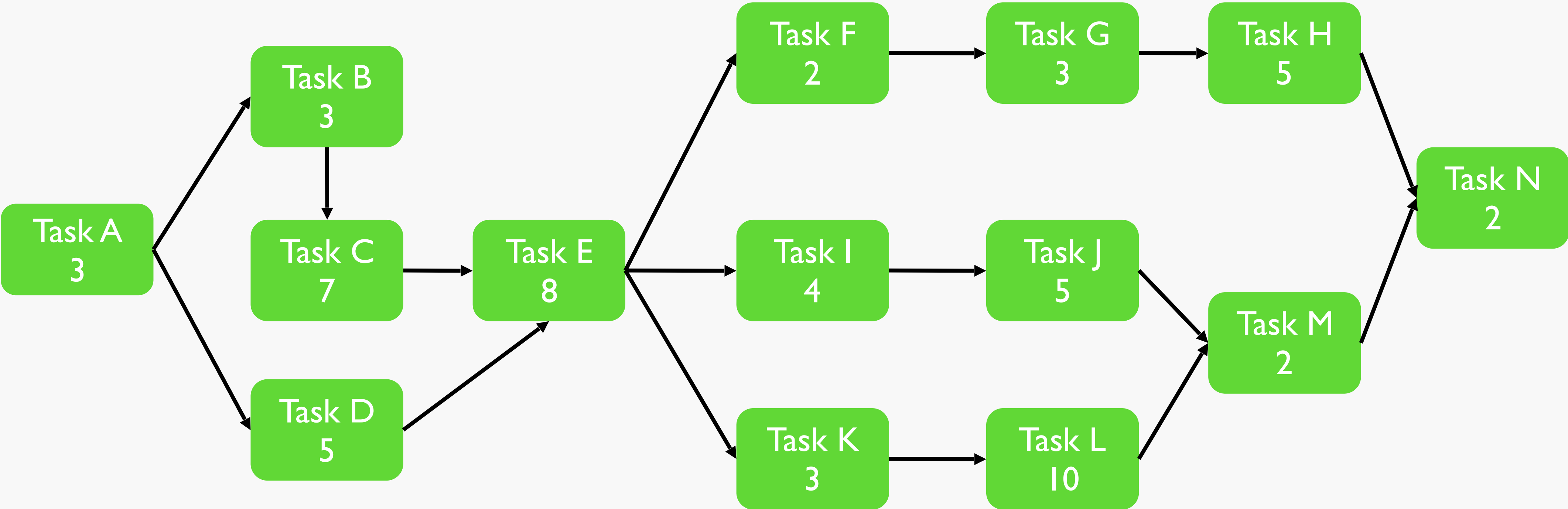


问：一个项目是怎么会延期了一年的呢？

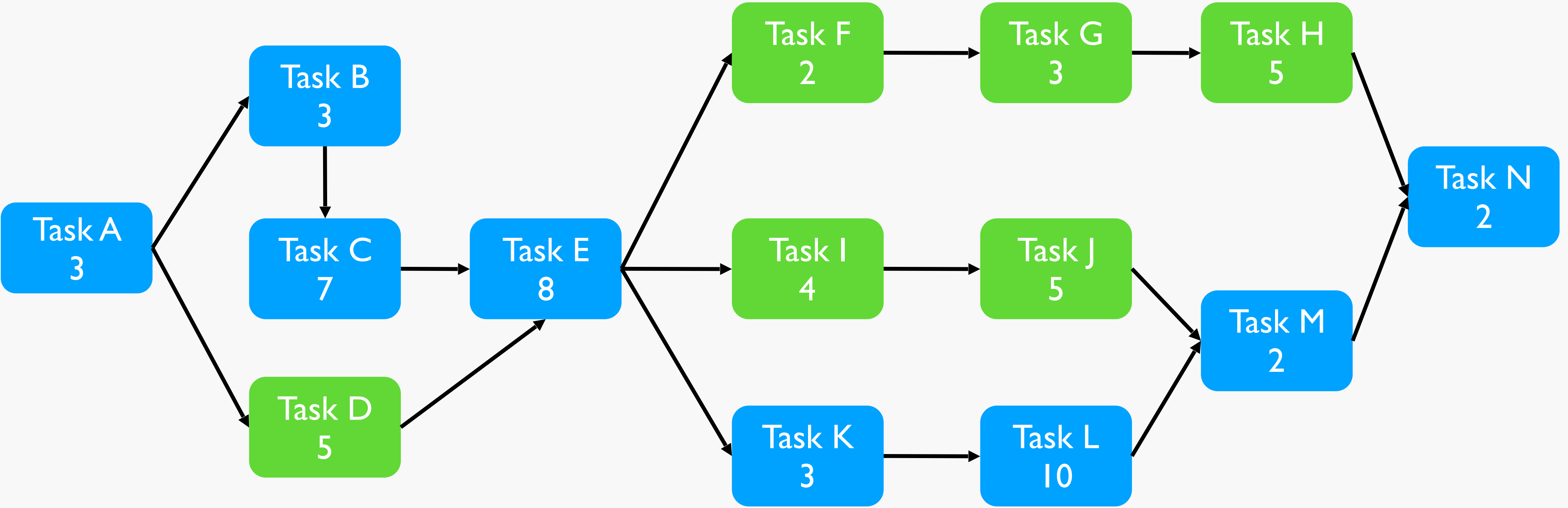
问：一个项目是怎么会延期了一年的呢？

答：每次遇到问题延迟一天就足以导致这个结果

了解任务依赖



做好项目时间预估



workflow 是什么？

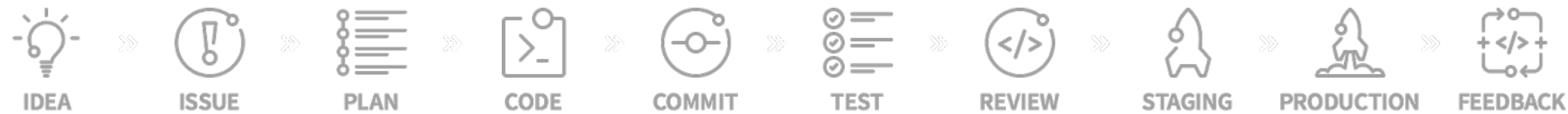
- 我们做开发是怎么做的呢？
- 需要按哪些步骤进行？
- 每个步骤之间关系是什么？



项目级的工作流



需求的工作流



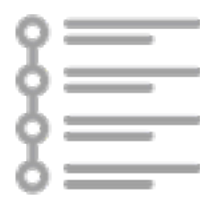
评审的工作流



IDEA



ISSUE



PLAN



CODE



COMMIT



TEST



REVIEW



STAGING



PRODUCTION

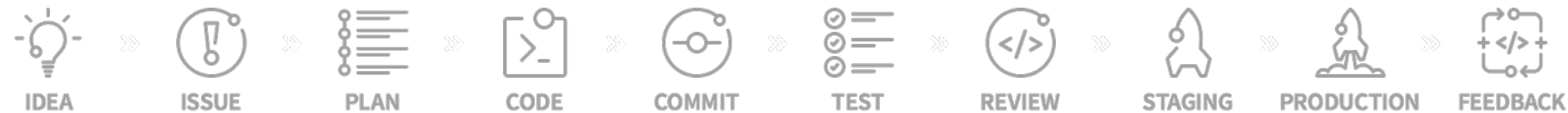


FEEDBACK

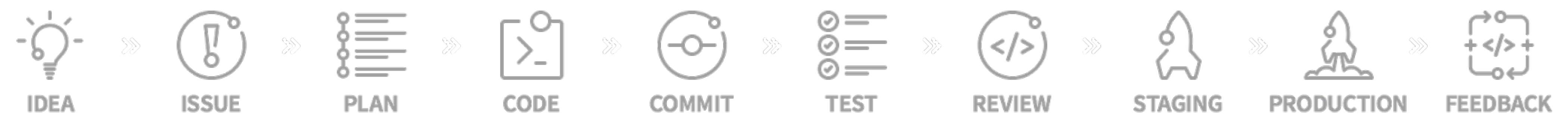
设计的工作流



开发的工作流



审计与测试的工作流



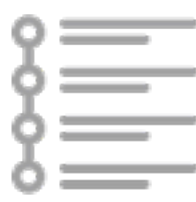
部署与监控的工作流



IDEA



ISSUE



PLAN



CODE



COMMIT



TEST



REVIEW



STAGING



PRODUCTION



FEEDBACK



有了工作流之后

01

切忌偷工减料

02

工作流是提升效率的，而不是降低效率的

03

在执行前要有约定好的惩罚性、奖励性措施