

## 第54讲 | 打造高速运转的迭代机器：现代研发流程体系打造（一）

2018-07-17 爱范儿CTO兼知晓云负责人何世友

技术领导力300讲

[进入课程 >](#)



讲述：黄洲君

时长 11:59 大小 5.49M



你好，我是爱范儿 CTO 兼知晓云负责人何世友，今天想跟大家分享的话题是“打造高速运转的迭代机器”。

### 现代移动互联网对研发流程提出的新问题

移动互联网时代的产品研发，基本上是两个特征，分别对应老板和用户的要求：

1. 迭代速度再快一点，一分钟上线一个特性；
2. 交付质量更高一点，半个 bug 也不能有。

而且没有上限。

而速度和质量，是一个肉眼可见的相悖的存在——速度快了自然要牺牲质量，质量高了自然要牺牲速度。但在移动互联网公司的意识形态里，再相悖，那也是把大象塞进冰箱的问题，因此也就对研发流程提出了新的挑战。本文试图就该问题分享我的经验所得。

正式解决前，我们伪随机地“采访”几位围观群众。

### **问：如何搞定更快迭代？**

开发工程师：给我精确可实现的需求，我一人能干三人的活。

运维工程师：给我更多一点机器干活，我让机器随时随地跑构建、测试、部署。

其他各种师：大家高效沟通快速复盘，地毯式推进。

### **问：如何产出更高质量？**

项目经理：迭代中各工序完整，涵盖各职责。

其他各种师：沟通全面无疏漏。

运维工程师：更多一点机器背锅。（滚.....

所有人：产品、研发、测试、构建、部署各环节紧密关联，随时可回溯，随时可加入新人。

项目规模或大或小，周期或长或短，都是由一个又一个版本迭代构筑起来的。当我们在谈研发流程时，实际上在谈的，就是这不断重复的迭代过程。因此，打造研发流程，就是要打造一台高速运转的迭代机器。

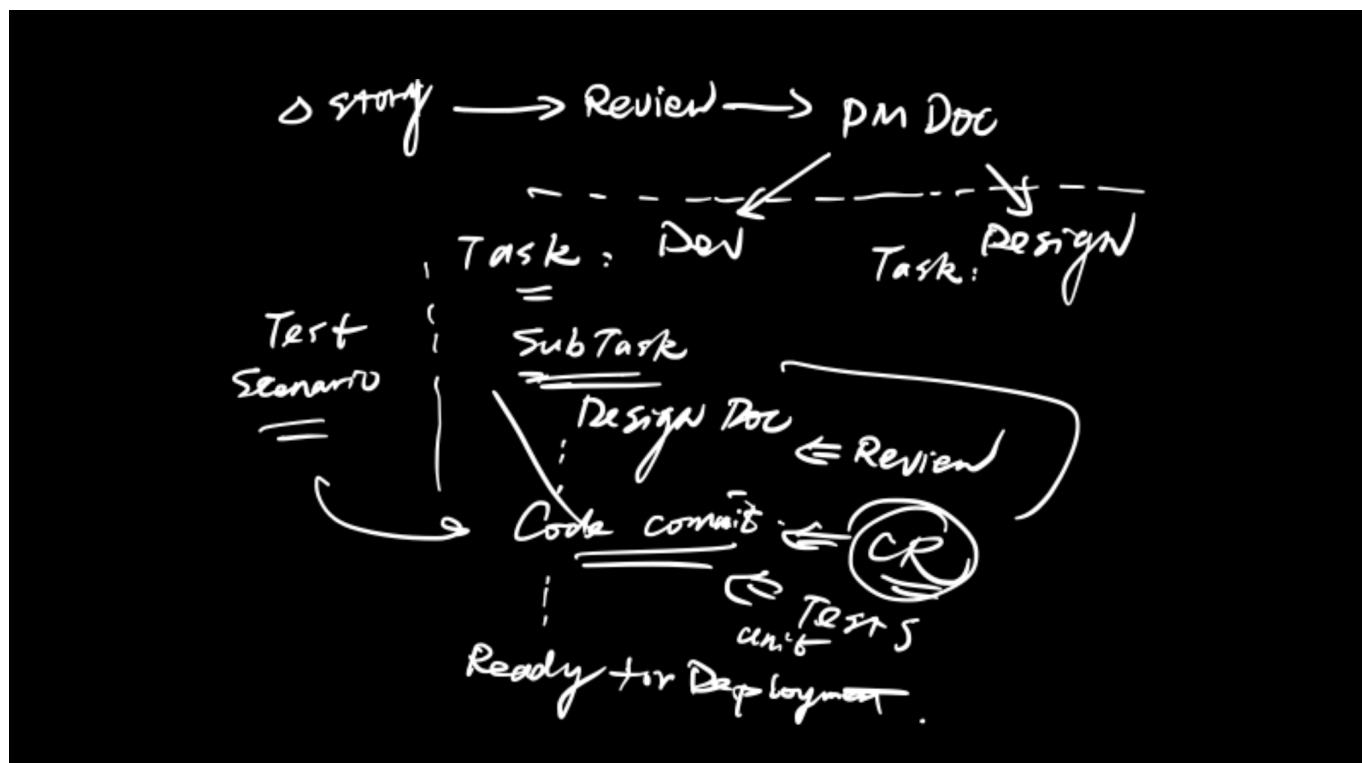
## **打造迭代机器**

工业时代的崛起，运转在工厂车间永不停歇的传送带是成功的基石。这条传送带贯穿了各个环节，配合精确的时间控制，最终形成产出效率最大化的流水线。纵观整个产品迭代周期，我们需要立即着手做的，恰恰就是找到这一条传送带。

那什么是研发流程里边的传送带呢？

## **迭代过程总览**

一个标准的互联网产品的一次迭代大概长这个样子：



这个过程中，参与角色有：

1. PO (项目负责人)
2. PM (产品经理 / 项目经理)
3. Designer (设计师)
4. Developer (泛指各种研发岗位, 开发者、架构师、运维等)
5. Tester (测试工程师)

各环节的交付物有：

1. Story (一段需求描述)
2. Product Document (产品文档、原型)
3. Task (任务, 由各岗位在项目管理平台创建并维护)
4. Design (设计稿、交互稿、切图素材)
5. Test Case (测试用例, 由测试工程师编写并发起评审)
6. Tech Design Document (技术设计文档, 由开发者在开始编码前编写)
7. Code Commit (一次代码提交)
8. CI Build (持续集成系统自动化构建)
9. Test Run (一次测试)

## 10. Deployment（一次部署）

各环节与环节之间的转场动作有：

1. Review（审核、讨论、修改）
2. Acceptance（验收）

需要注意的是，这里的角色、交付物并非限定因素，各团队可以根据自身因素和项目规模应用不同的角色设定和交付物设定，大可不必纠结。

可以看到，一次可短至一天内完成的迭代，涉及到 5 种角色、10 种交付物，同时，每个角色之间、交付物之间，都会由各种审批和验收的动作填充。一眼看上去复杂到爆炸。

而我们作为技术管理者，需要将这些角色和交付物和动作管理起来，这是打造研发流程的基础。

### 现代化的项目管理工具

要做管理，必然离不开管理工具。项目管理工具在近 10 年里得到了很多厂商的青睐，纷纷投入研发力量，于是我们如今可以用上 Redmine、Jira、Trello、Tower、Teambition、Worktile、禅道等各种工具。

其中，受 Trello 的影响，近些年的项目管理工具多实现的是简洁的“看板”管理方法，成品往往具有简单好用的特征。老牌的 Jira 则同时支持 Scrum、看板（Kanban）管理方法。Scrum、看板均是敏捷开发流程的实现，并不存在特别大的区别，各团队可以根据自身的喜好选择，这并不是重点。

但如果项目管理工具还只是让项目中的各角色在上边手工创建、更新任务卡片，那一定会成为项目推进中的阻碍。现代项目管理工具一定要将交付物管理、串联起来。管理工具的目的是为了维护任务状态，而任务状态事实上是实际任务完成与否的反映。也就是说，管理工具需要具备从交付物所在平台获取动作事件，将其转换成任务的流转，并完成任务状态更新的功能。

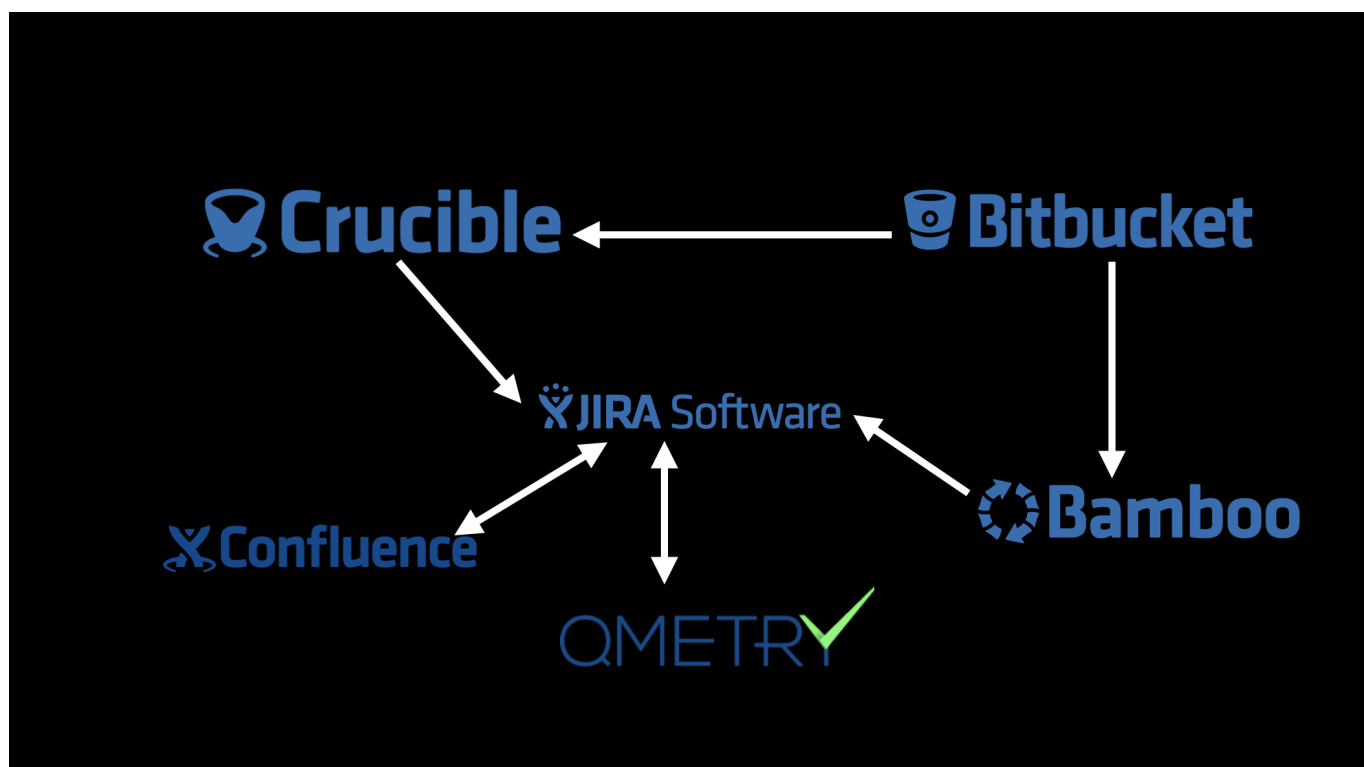
比如说，工程师的任务是写代码，当他做了一次代码提交，就意味着这个任务完成了，如果还需要他手动去任务管理平台更新任务状态，就比较多余。因此，任务管理平台需要具备从代码仓库获取代码提交事件的能力，并自动更新对应的任务状态。

可以说，项目管理工具就是上文提到的研发流程里的传送带。

这条传送带承载了任务的流转，更需要承载一个任务所有交付物的关联。一个任务，包含产品原型文档、设计稿、技术设计文档、代码、测试用例、测试结果、CI 构建、部署等诸多交付物，把这一切串起来，可以解决什么问题？

回溯。

## 可回溯的迭代



“得益于”不断提速的迭代流程，互联网项目的代码腐化速度，早就从年提升到月。一个“正常”迭代的互联网产品，即便立项时的架构设计文档写得再细致出色，每次迭代时的代码写得再易读注释清晰，3 个月后，来一个新人，一定搞不懂为什么代码会写成这个鬼样子。

为什么？

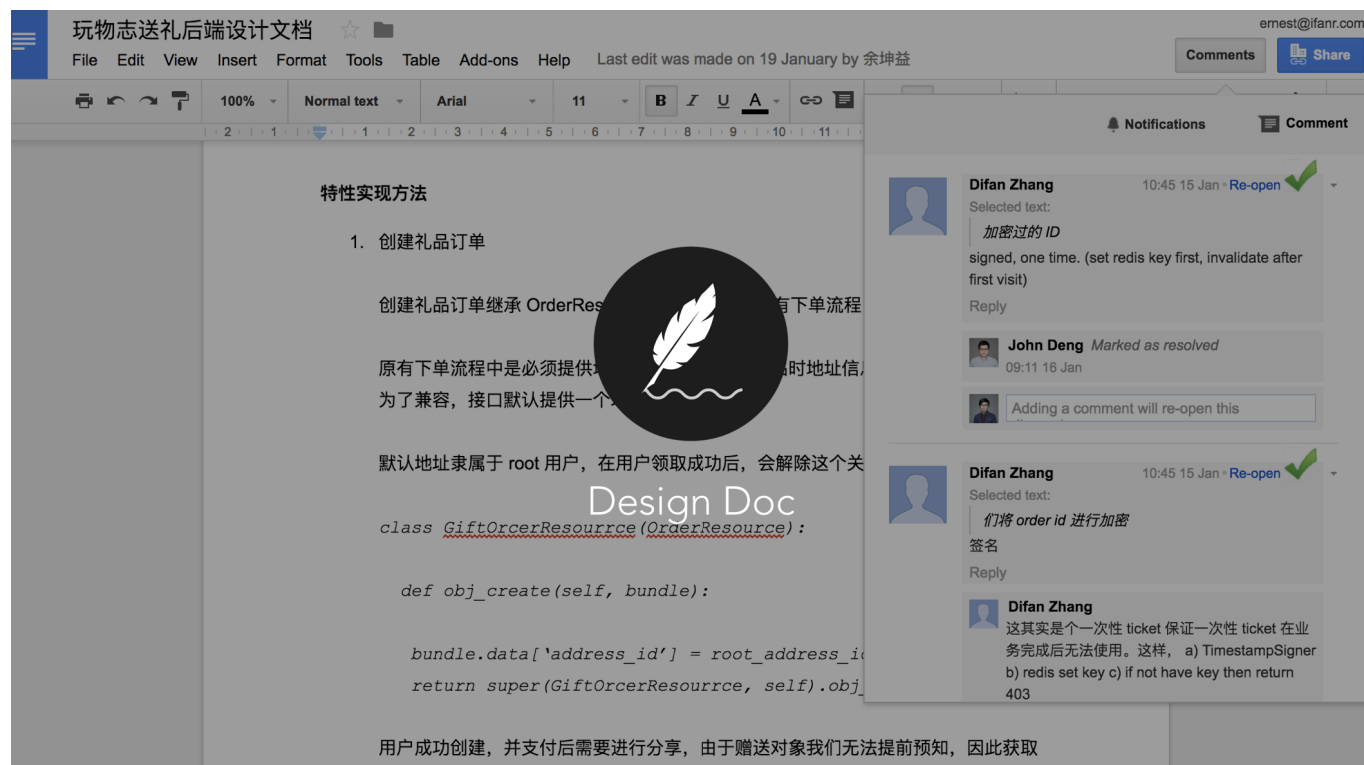
因为无法追溯。

代码的每一次更改，都是在响应当次迭代的产品变更。虽然每一次产品变更都可以通过产品原型的更迭得到留存，但因此导致的代码更改乃至架构微调，却无法产生对应关系。

那通过任务管理平台来关联产品变更和代码变更呢？这样每一个产品变化导致的代码变化以及当次的测试、构建、部署结果都是串起来的，当需要回顾事故现场时可以将这些关键交付物拿起来看。应该可以解决所有问题了吧？

还不够。

## 不断迭代的技术架构设计



虽然可以通过这些元素的串联还原当时的迭代详情，但依然可能出现搞不懂代码为什么写成这个鬼样子的困惑。因为已有元素的提供，可以搞清楚代码变更的罪魁祸首，但无法还原现场。在此，需要的是完善的、不断迭代的技术架构设计文档。

举个例子：电商业务产品想增加送礼功能，用户 A 下了一个订单付完款，用户 B 填写收货地址。产品变更简单，增加一个用户 B 更新地址的流程就可以。但对应的代码变更则大了去了，这一个礼品功能的增加，可能直接导致订单架构的变化。这时候如果没有技术架构设计文档的补充，对当时的需求进行分析和架构变更设计进行记录，产出的代码一定是不好理解的。

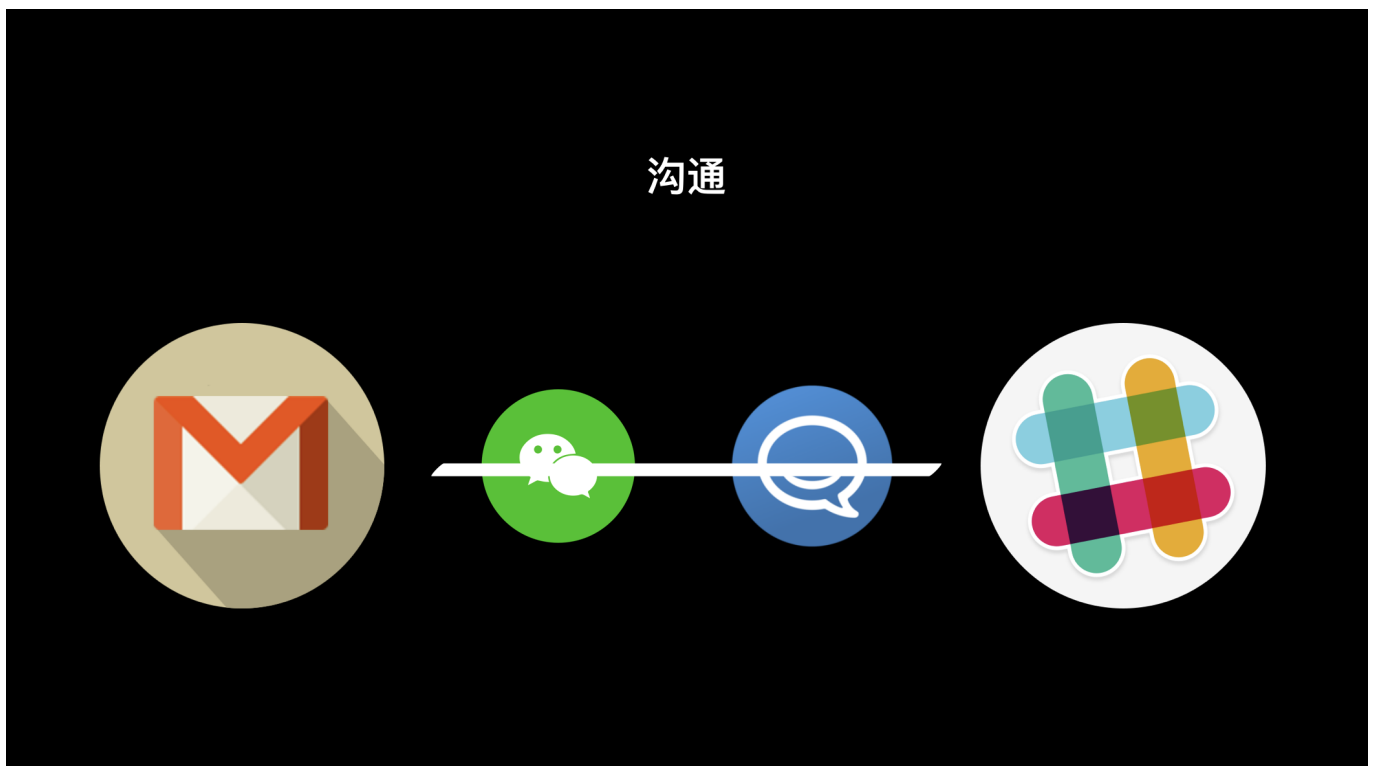
现代的软件工程构建，再也不是一座桥梁的构建模式——蓝图画好了就打死不改。代码可以不断迭代，同样的，蓝图，也就是技术架构设计也应该不断迭代。而设计文档，就是技术架构设计的迭代交付物，在整个迭代过程中，至关重要。



至此，迭代过程可以做到完全可追溯，这对保证项目质量起到非常大的作用。一个新人刚加入项目，不管是从代码提交记录，还是从设计文档的修改记录，还是从每一次构建部署的记录，都可以追溯到关键节点。如此一来，项目的生长轨迹，再也不用依靠老同事的口耳相传得以维系，其腐化速度也能得以缓解。

一个快速迭代成长的项目，就需要这样的基础。在这个基础之上，如何快速推进？须知，参与项目的不仅有人类角色，还有各种交付物所在的系统——在线文档协作平台、代码仓库、测试用例管理平台、CI 构建平台、自动化部署平台等。这些角色 / 机器之间，也需要沟通。

## 沟通，不仅是人和人



迭代中的沟通，一般有这么几种需求：

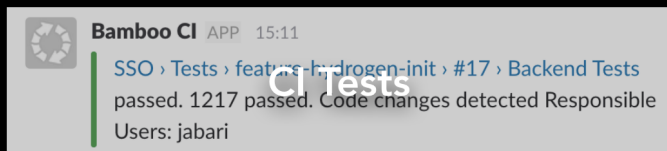
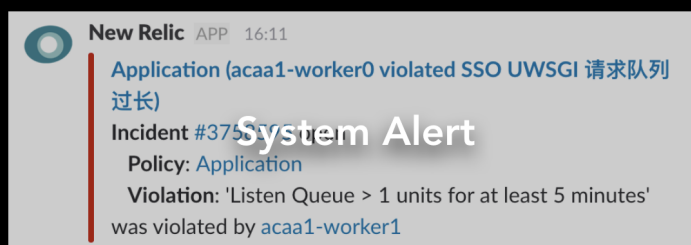
1. 角色之间的沟通，讨论、约定、解决冲突；
2. 群组内的沟通，头脑风暴、计划宣布等；
3. 就某一话题进行多人沟通。

沟通工具要满足的条件之一，就是所有沟通结果可留底用于未来追溯。因此重要的信息，邮件依然是最佳选择。通常在项目启动、验收等关键时刻，邮件确认是最合适不过的。

平日的沟通交流，则以效率优先，大多数是发生在即时聊天工具上的。微信提供了非常便捷的讨论组功能，外部联络时非常高效，可以瞬间圈起公司内外相关人士进行畅聊。但日常工作中，很多时候会有临时的话题需要进行多人沟通，沟通完就散。

举个例子，#codereview 群组里，CI 构建系统发了一条最新的测试失败消息，相关的开发者需要就这一个失败进行讨论。直接在 #codereview 群里讨论吧，消息一多就冲垮了，有噪音；拉一个新的群组吧，又太费劲了。这种情况下，Slack 提供的 Thread 功能就能派上用场。

## 通知器



此外，研发流程里参与的角色一半以上都是机器 / 系统，因此，沟通工具优先要满足的，其实是如何让机器和人对上话——这就是现代沟通工具的特征。在同一个沟通工具里，人类角色可以像和其他人类角色沟通一样和机器沟通，如接收机器的通知（如上图所示，有代码提交的通知、告警通知等）、向机器发送指令等。

## 用机器打造迭代机器

项目管理平台这条传送带，让一次迭代里的各个角色、任务可以快速进行流转，并从流程上充分利用时间、减少损耗。一个需求，粒度划分越细，就越可以做到精确可控。然而，一次需求从代码到上线，这中间的过程是漫长的——

编码⇒静态检查⇒单元测试⇒测试⇒代码审查⇒构建⇒部署⇒运维.....



我们要做的，就是尽可能的自动化，让机器包揽所有能干的活。这样一来，迭代过程，不仅可以无误差运转，甚至连运维的工作都可以更加高效——

编码【人类才智】⇒静态检查【机器】⇒单元测试【机器】⇒测试【机器】⇒代码审查【人类才智】⇒构建【机器】⇒部署【机器】⇒监控【机器】⇒自动扩（缩）容【机器】。

除了两个必须要人类参与的步骤，其他环节机器均可以胜任。

可能有人会问：研发团队写这一套自动化流程，得花多大成本（时间、金钱）？是不是和研发产品缘木求鱼了？答案是磨刀不误砍柴工。

至于成本嘛，感谢开源社区和相关服务提供商，几乎都是现成的、可以花合理价钱买到的。篇幅有限，这里我就不展开了，下一篇文章我将跟大家简单说说各环节的现有解决方案及选型，敬请期待。

研发流程是一个团队打造产品的过程，这里边一个最基本的性质就是生长、不断的生长。而产品快速度、高质量的生长离不开高速运转的迭代机器，本文就迭代机器的打造给出了我自己的经验与答案，存在疏漏在所难免，欢迎大家在评论区提问讨论。

## 作者简介

何世友，爱范儿 CTO [TGO 鲲鹏会](#) 广州分会董事会成员，学习委员。从校园创业到跨国团队技术顾问再到如今，专注于高并发网络、机器学习、移动 APP（部分硬件）、团队管理。

---

# 技术领导力 300讲

每个技术人都应该知道的管理心经

梁宁 / 著名产品人

张雪峰 / 饿了么CTO

陈皓 左耳朵耗子 / 知名创业者

许式伟 / 七牛云创始人兼CEO

李大学 / 前京东CTO

汤峥嵘 / turtorABC COO

右军 / 蚂蚁金服

程浩 / 迅雷创始人

郭炜 / 易观CTO

新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 第53讲 | 如何打造高效且敏捷的组织文化？谈高效研发流程那些事（五）

下一篇 第55讲 | 用机器打造迭代机器：现代研发流程体系打造（二）

## 精选留言 (4)

 写留言



黑米

2018-11-19

 1

很巧，我们正准备自研一个开发团队管理的全链路平台，跟你的理念差不多诶。



大晨\_Richa...

2018-07-17

 1

期待下期，快点放出工具

展开 



Ernest 何...

2018-07-18



@石头，选用可以接受 repo commit event webhook 的管理平台就可以解决，这个市面上可以选择的很多了，做的比较好的是 jira smart commit

---



石头

2018-07-17



"工程师的任务是写代码，当他做了一次代码提交，就意味着这个任务完成了，如果还需要他手动去任务管理平台更新任务状态，就比较多余。"

想法很赞。但这需要将任务管理和代码管理工具进行打通，这个有什么好的工具或者建议？还有一个任务其实需要多次代码提交的，并不是一次可以代码提交就可以搞定。

展开 ∨