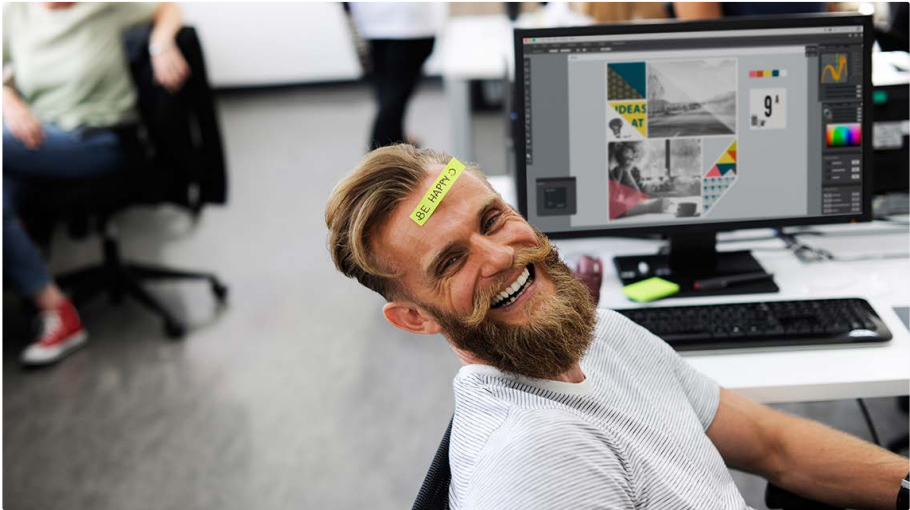


说说硅谷互联网公司的开发流程

2018-01-17 朱赞



说说硅谷互联网公司的开发流程

朱赞

- 00:05 / 09:49

今天，我和你聊一聊硅谷互联网公司的开发流程。之前我的很多文章就或多或少涉及过这一方面的内容，最近我又全程参与并负责了两个大项目，对流程有了更深入的理解，今天就在专栏里系统地和你探讨一下。

总的说来，我们的开发流程包括这么几个阶段：

1. OKR 的设置；
2. 主项目及其子项目的确立；
3. 每个子项目的生命周期；
4. 主项目的生命周期；
5. 收尾、维护、复盘。

第一点，OKR 的设置

所有项目的起始，都应该从 Roadmap 做起。硅谷公司的 OKR (Objectives and Key Results) 一般都是自顶而下的。也就是说，先有整个公司的 OKR，然后有每个部门的 OKR，继而是大组的 OKR，再到小组的 OKR，确保整个公司有一致的目标。在这个过程里面，技术驱动反映在哪些方面呢：

首先，确定 Roadmap 的过程中，我们会采用调查 (Survey) 模式，确保工程师的声音可以准确地触达管理层。比如：工程师们觉得基础架构比较薄弱，公司就会加大这一块的支持力度。如果大家觉得开发环境很低效，就会把这个因素也放到 OKR 的考虑。硅谷的公司一般会分为产品组和系统架构组。总的说来，系统架构组的 OKR 里，工程师的声音会很大。

其次，项目怎么做，怎么规划，一般是由工程师来决定。OKR 只确立目标，是不是要构建新的服务，是不是要沿用现有的架构，如何进行技术选型等等，这些不是 OKR 的组成部分。

最后，估算 OKR 里的目标工期的时候，我们会除去一些用来做技术创新和支持的时间，比如编程马拉松，开源支持等的事务。谷歌的员工会给自己留 20% 的自由项目时间，这些都是时间缓冲区。

(注：OKR 是企业进行目标管理的一个简单有效的系统，能够将目标管理自上而下贯穿到基层。具体概念可以参考 <http://wiki.mbalib.com/wiki/OKR>。)

第二点，主项目及其子项目的确立

一旦确立了 OKR，下一步就是确立主项目和子项目了。主项目是主要的技术或商业产品，一般由产品经理、技术经理和一些技术骨干经过产品需求和技术讨论之后，确定要做什么 (Scope)，不做什么 (Non Scope) 和大的里程碑 (Milestone)；后面我会在“工程师、产品经理、数据工程师是如何一起工作的”一文中更详细地介绍不同角色之间的合作细节。

一旦主项目确定了，就需要安排不同的人做不同的模块，也就是子项目。一般团队协作有两种方式：一种是每个人负责一个子项目，从始至终；另一种是大家先一起完成基本框架，然后逐个需求、逐个模块推进，最终一起完成整个项目。

下面，我来谈谈两种协作方式在实践中的优缺点对比。

第一种协作方法：每人完成一个子项目。

优点：责任清晰，每个人都知道自己的职责，工程师们也有更多的拥有感，他们可以独立负责产品的设计、实现、测试和维护，工作贯穿整个项目过程。

缺点：如果负责某个子项目的工程师设计或者实现能力不足，由于比较独立，这个子项目很容易成为路障或者瓶颈，工程师之间也缺乏互相学习的机会。

另外，因为是按人并行推进项目，需要根据每个人设置里程碑，管理的时候，技术管理者需要常常跟进每个人的进度，管理代价更高。代码审核往往也只是有限的几个人参与。

第二种协作方法：所有人一起逐次完成每个模块或需求。

- 优点：工程师之间合作最大化，可以彼此协调、彼此学习、在对方有事的时候相互补位。项目管理有明确的统一的里程碑，每个工程师都有机会接触更多的工作，每个人的代码可以有更多人参与审核。
- 缺点：每个工程师的责任不是那么明显，容易出现能者多劳、勤者多劳的现象。一些新人总是做一些执行或打杂的事，得不到锻炼。

这两种模式我都曾亲身经历过，感觉两者各有利弊。现实中可以根据情况组合使用。比如，两到三个人合作负责一个模块，也可以在每人一个模块的基础上，将小模块组合成大模块。然后每个大模块有个技术负责人（Tech Lead），对一些能力不足的工程师给予指导和支持等。

第三点，每个子项目的生命周期

子项目一旦确认，它的生命周期就融入到工程师们的日常工作中，内容如下。

1 开发初期的设计文档。一般使用可以共享的谷歌文档（Google Docs），Quip 等。不同的人可以编辑或者评论、阅读。一般设计文档会先由组内工程师和产品经理审核，然后到大组评审（包括Legal, Compliance, Finance 等等）。

如果涉及公司的整体架构，还需要发给全公司审核。参与审核的人员是所有的工程师。很多人会有选择的参与一些设计的审核，通常技术骨干会预留时间审核所有的技术设计文档。设计文档不仅包括怎么实现，还有选型的理由、考虑的因素、支持和不支持的属性、时间线等等。

2 设计测试实验，这是可选的，如果针对某个产品需求我们想知道用户的反馈，就需要数据工程师参与设计实验，也就是 A/B 测试。实验中的数据埋点也会在下一步的实现中完成。

3 一旦设计文档锁定，就可以开始实现了。不论是单人负责还是多人合作，实现都是按照多次代码提交（Pull Requests）来迭代的。每次代码提交要写清除代码改动的摘要和测试。并通知不同的工程师审核。

4 所有的实现都要加入监控、日志、预警代码。

5 所有实现都是隐藏在一个开关后。当代码都就位后，就开始灰度发布。通常是先发布给几个开发人员测试，然后到项目组，然后到其他员工（Google 称之为 Dog Food，因为他们可以大量使用自己的产品），最后按照百分比推给用户。

推送的过程中会结合 A/B 测试，只有测试结果显示对用户体验、公司主要的指标（Metrics）没有明显的负面影响，才会正式发布给所有用户使用。

6 对一些需要重构的关键产品链路，有时候也会使用双重写入（Dual Write），就是新特性和旧特性都写入数据库，并通过不同方式比较两个实现的结果。只有验证结果一致时，才会将交易（Traffic）从旧实现切换到新实现。

7 最后是一些扫尾工作，包括移除用来做 A/B 测试和灰度发布的代码开关等，有时候还会有一些次要需求的实现。

第四点，主项目的生命周期

主项目的生命周期根据子项目的实现方式会有所不同，但有一些特点是共有的。

1. 项目开始都有一个整体设计文档，界定所有子项目的范围和相关性、时间线等。
2. 在所有子项目进行的过程中，有时候会发现一些共同需要的架构或者服务，可以单独提取成公共服务或库，比如一个调度服务，或者一个幂等实现等等。
3. 给相关人员做进度报告，包括主项目的里程碑。
4. 由于子项目完成时间可能不一样，需要进行人员的重新配置。
5. 在开发过程中不断更新文档。
6. 因为不确定的需求变动，会取消或者生成新的子项目。
7. 有时候，也会因为公司的方向变化或战略调整，对主项目做比较大的变更，同时对应调整相关的子项目。
8. 在项目开始和结束的时候，需要做好对外的交流和沟通。一来确保自己的项目改动不会影响到其他组的项目，二来让将来会依赖这个项目的产品组了解相关信息，确定计划。

第五点，收尾、维护、复盘

整个项目结束后，一般都会做一些代码清理和文档的更新和整理，有时还需要写新的用户手册或 Wiki 等。一些基本的错误和异常处理要写到运维手册（Oncall Playbook）里，便于以后运维的人知道怎么处理一些已知的问题。

每个项目结束都会进行复盘，总结整个项目的教训和经验。有时候还需要在组内做些演讲，让更多的人了解这个项目。

总结一下，今天我主要和你探讨了软件产品和服务的开发流程，一般硅谷里稍具规模的互联网公司都会遵循类似的流程，他们就是通过这样的流程开发出了创新性的产品。

这些流程包含什么内容呢：首先在公司内部确认 OKR，然后确定主项目和子项目，开始进行产品实现，也就是完成主项目和子项目的生命周期，最后进行项目的收尾、维护和复盘，一个大项目就开发完成了。

你的项目开发流程有什么异同吗？可以在留言中告诉我，大家一起进步。感谢你的收听，我们下期再见。

Hi，亲爱的订阅读者

每邀请一位好友订阅

你可获得18元现金

快来获取你的专属海报吧！

| | |
|--|------------|
| 戳此获取你的专属海报 | |
| bluze | 2018-01-17 |
| 我去，硅谷的流程貌似更科学一些。我司就5个开发，没那么讲究... | |
| William李梓峰 | 2018-01-21 |
| 国内根本就没有流程可言，比较捷开发还要敏捷。 | |
| 云飞扬 | 2018-01-17 |
| 国内没有这么好的流程，大公司开个kick off，然后就是编码了，小公司就是直接干，中间想改就改，有流程的不多 | |
| ajodfaj | 2018-01-17 |
| 在高校，基本原则就是直接干，先有个能动的版本再说，然后就没有然后了 | |
| 伪装的架构师 | 2018-05-12 |
| 针对项目开发流程我想谈谈在现公司中的一些共同和差异。1、OKR的管理。项目设立时，OKR自上而下进行确立，总体上能保持一致，但是随着项目进行会出现偏差，这个偏差往往是在部分功能投产后，用户实际的反馈会导致需要对项目初设立的OKR进一步补充才能完善；2、技术创新和支持，考虑到工作安排的时效性和在一般公司技术创新的可行性和依赖，国内公司很少有这部分的工作安排。但是这部分时间对技术人员的成长和价值体现又往往很重要，这不知道是不是雇主和雇员利害最后博弈的结果；3、重构关键产品链路，进行双重写入的实现，验证数据结果，再进行切换。这个实现确实对保持服务的稳定性方面起到至关重要的作用，但是实现上需要花费比较大的时间和精力，项目管理者往往就愿意用测试的覆盖率和案例量去替代了这个实现方案；4、项目收尾，基本错误和异常处理写入到运维手册。这个也是对项目运维比较好的支持和积累，值得推荐。 | |
| pjd | 2018-02-03 |
| 4 所有的实现都要加入监控、日志、预警代码。 | |
| 5 所有实现都是隐藏在一个开关后。灰色发布 | |
| 能不能讲讲这两点的具体实现？ | |
| 刘剑 | 2018-01-23 |
| 朱老师想问下。 1.一个版本的开发周期是多久？有延期的情况吗？有哪些原因在这种开发流程下还会导致延期？ 2.中间是否允许需求变更？如果允许变更流程又是什么样子的呢？ 3.在开发流程中用到了哪些项目管理工具能介绍一下吗？ | |
| 张荣 | 2018-03-16 |
| 安姐！ 监控、日志、预警这块能讲讲吗？ | |
| 王逵Tyrion | 2018-01-19 |
| 硅谷的流程就是完整，学习了。 | |
| jerry | 2018-06-02 |
| 一个完整的持续性的项目基本都会遇到上述流程点要解决的问题，有些时候解决问题的方案是多变的，主要是在效率和抗风险能力之间做权衡，公司资源有限。朱姐分享的流程都值得借鉴学习，在资源有限的情况下尽量带领自己得团队拥抱先进的流程，核心是让团队更高效，资源使用更合理，搞清楚每一步关节背后要解决的核心问题是什么，感谢朱姐的分享！ | |
| whhbbq | 2018-05-28 |
| 没有专职测试攻城狮？ | |
| whhbbq | 2018-05-28 |
| 没有专职测试攻城狮？ | |
| hsy | 2018-02-22 |
| 我们在前期还有设计文档评审之类的一堆会议 ♦♦ | |

