

《极客与团队》

图书基本信息

书名：《极客与团队》

13位ISBN编号：9787115308443

10位ISBN编号：7115308446

出版时间：2012-3

出版社：人民邮电出版社

作者：[美]Brian W. Fitzpatrick/Ben Collins-Sussman

页数：176

译者：徐旭铭

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

前言

“工程问题都很简单。人际关系才是最难的。”——比尔·库格伦，前Google工程部资深副总裁

生活中总是充满了离奇的转折，就好像我们俩从没想过会合作写一本软件工程的书一样。和大多数电脑狂一样，大学毕业后我们发现自己的兴趣和热情(折腾电脑)居然也是不错的谋生手段。而和那个时代的大多数黑客一样，我们的整个20世纪90年代中期都是在干这些事情，用别人剩下的零件攒机，拿着一大叠软盘安装预览版的Linux，然后学着操纵UNIX机器。我们都是系统管理员出身，然后在互联网泡沫刚起来的时候跑到小公司里去当程序员。泡沫破裂后，我们开始为那些幸存的硅谷企业(比如Apple)工作，后来又跳槽去了一家创业公司(CollabNet)，全心设计开发了一款开源版本控制软件Subversion。然而2000年到2005年期间，一些意想不到的事情发生了。尽管我们创造了Subversion，但是我们每天的任务却渐渐发生了变化。我们不再只是天天坐在那里写代码，而是开始领导一个开源项目了。这意味着我们要整天挂在聊天室和一堆程序员志愿者打交道，关注他们都在做些什么。这还意味着要几乎完全通过邮件列表来协调各种新特性。我们逐渐发现一个项目成功的关键不仅仅是写出漂亮的代码：所有人向着同一个目标一起合作也是同样重要的。2005年的时候我们一起创建了Google芝加哥分部，以程序员的身份继续着我们的职业生涯。这时我们已经完全融入开源世界了——不仅仅是Subversion，还有Apache软件基金会(ASF)。我们把Subversion移植到Google的BigTable架构上，并以Google Code为名发布了一项开源项目托管的服务(类似于SourceForge)。我们开始参加(后来开始做演讲)各种开发者大会，例如OSCON、ApacheCon、PyCon，还有Google I/O。我们发现同时为企业和开源项目工作的经历让我们无意间了解到了软件工程团队运作的奥秘。一开始我们做的演讲都是轻松幽默的，大多是批判无用的开发流程(“Subversion最差实践”)，后来则开始转向探讨如何保护团队不受害群之马的影响(“开源项目如何应付害群之马”)。越来越多的听众聚集到我们的演讲会场，我们称之为程序员的“集体疗法”。每个人都对我们谈论的问题有着切肤之痛，都想要对大家抱怨一下。于是，六年来我们做了一大堆反响热烈的和软件开发过程中人际关系有关的演讲。最后奥莱利公司的编辑玛丽·特莱斯勒建议我们干脆把这些演讲写成一本新书。接下来发生的事情就略过不表了。

本书的读者 本书是专门写给那些想要更上一层楼以及编写出色软件的程序员看的。CEO、心理学家、管理层、计算机理论学家，还有焊电路板的技师等都不算是我们的目标读者(虽然这些朋友或许也能读得津津有味)。在此我们假设本书的读者具备以下两个重要条件。你需要和团队里的其他程序员合作。无论是为公司企业工作，或是某个开源项目或是学校作业的一员都可以。你喜欢软件工程，并且觉得它应该是一件很有成就感、很好玩的事情。如果你只是应付工作混口饭吃的话，那估计你对实现自我价值和获得职业满足感不会有什么兴趣。在讨论程序员怎么才能“通力合作”的时候，我们总结出了一些表面上看起来和程序员这份工作风马牛不相及的经验和准则。我们会在各章里谈到怎么才能更好地领导一支团队和企业，怎么才能和用户建立起良性的关系等话题。乍看起来这些章好像是专门写给所谓的“管人的人”或者说“产品经理”看的，但我们可以保证，你的职业生涯到了某个阶段的时候一定会有意无意地担当起这样的角色。所以先放下疑虑，继续往下读吧！这本书里讲到的东西是每个软件工程师最终都会碰到的。警告：本书不是技术手册 首先我们需要澄清一些事情。好学的程序员通常都喜欢去读一些能将某些个特定问题用纯数学的形式展现出来的书籍，而通常这样的问题都存在完美的流程化的答案。本书并不属于这种类型。本书主要关注的是软件开发中有关人的那个方面，而人是很复杂的。正如在我们自己的演讲中说到的那样，“人基本上就是由一大堆间歇性bug组成的”。这里论及的问题和答案都是很凌乱的，很难用完美逻辑来解释。本书读起来像是论文集，而事实上它的确也就是一本论文集。每一章我们都会讨论一系列相关的问题(通常是用案例故事的方式)，然后针对整个主题再探讨各种应对方案。你可能需要反复阅读很多页，努力把它们联系起来，甚至思考数日才能融会贯通！我们在此还要作一些声明。正如我们在演讲里开过的玩笑，“这里的观点完全是基于我们自己的经验得出的，完全属于我们自己。要是你不同意我们的观点，完全可以去自己做演讲啊”。和我们的现场演讲一样，只要和本书主题相关，任何讨论都是值得鼓励的。我们欢迎各种反馈、修正、新观点，以及反驳：你可以在<http://www.benandntz.com/>上找到我们。本书的所有内容都来自我们的亲身经历和无数错误中得到的教训。你还应该注意，我们在例子中提到的都是化名，以保护无辜(或者犯错)的当事人。本书的内容都是课本里学不到的 我们认识的软件工程师绝大多数都花了4~10年的时间在学校里学习计算机科学和软件工程。截至本书出版为止，我们不知道有任何一门课程是真正教你怎么在团队和公司里与人合作交流的。好吧，大多数学生都会在

《极客与团队》

学校里被要求参与某个项目，但是教一个人怎么好好地和另一个人协作与把他直接丢到一个迫使他协作的环境里完全是两码事。绝大多数学生最后都不怎么喜欢这种经验。 小结 所谓成功的程序员不仅仅是追逐最新的语言或是编写最快的代码。职业程序员几乎都是要参与团队合作的，而且事实上团队直接影响个人生产力和幸福感的程度超出很多人的想象。 本书的目标很简单：编写软件是集体项目，而且我们认为人的因素对结果的影响不亚于技术因素。大多数人虽然在编程技术上耕耘数载，但是在人际关系上却从来没有下过功夫，而学习与人合作也是成功路上不可或缺的重要环节。只要你能在软件工程的“软技能”上下点功夫，就能达到事半功倍的效果。

《极客与团队》

内容概要

《极客与团队》是一本写给程序员看的，教你怎么交朋友，怎么影响团队中的其他人。书中充满了操作性极强的建议和意见，让你在技术团队中过得更开心，变得更有效率，更加如鱼得水。《极客与团队》旨在帮助程序员改进理解他人，与人沟通，以及与人合作的能力，进而在编写软件的过程中变得更有效率。

《极客与团队》适合那些想要更上一层楼并编写出色软件的程序员阅读，也适合软件项目和团队的管理者自己或组织团队成员阅读和参考。

《极客与团队》

作者简介

布莱恩·傅攀勃现在是Google数据解放阵线和透明度工程两个团队的负责人，之前他还领导过Google的项目托管团队以及电子商务伙伴团队。他帮忙建立了Google芝加哥分部，并且为Google在开放数据上提出了很多想法和建议。

本·科林斯-萨斯曼是Subversion的初创成员之一，领导过Google的项目托管团队，现在是Google电子商务伙伴团队的负责人。他帮忙建立了Google芝加哥分部，并将Subversion移植到Google的BigTable平台上。

书籍目录

第一章 天才程序员的传说 1
帮我把代码藏起来 1
天才的传说 2
隐瞒是有害的 6
团队才是王道 11
三支柱 13
hrt实战 17
下一步 26
第二章 培养出色的团队文化 27
什么是文化 27
为什么要关心它 30
文化和人 33
优秀团队文化中的沟通模式 36
高层面同步 38
每日进行的讨论 46
使用bug跟踪系统 50
沟通也是工程的一部分 50
说到底真正重要的还是代码本身 55
第三章 大海航行靠船长 57
自然界没有真空地带 57
@deprecated manager 59
主管才是新的经理 60
唯一要担心的就是.....好吧，所有的事情 62
仆人式领导 64
反模式 65
领袖的处事之道 71
人是植物 89
内部激励和外部激励 91
结语 92
第四章 对付害群之马 94
什么是“害群” 95
保护团队 96
发现威胁 98
第五章 操纵组织的艺术 113
优点、缺点和策略 113
理想的情况：团队在公司里应该是怎么运作的 114
现实的情况：当环境成为成功路上的绊脚石 117
操纵你的组织 124
b计划：走为上 138
不要放弃 140
第六章 用户也是人 141
管理大众的印象 142
管理和用户之间的关系 161
结语 172
附录a 延伸阅读 174
StartFragment第一章 天才程序员的传说 1
帮我把代码藏起来 1

- 天才的传说 2
- 隐瞒是有害的 6
- 团队才是王道 11
- 三支柱 13

- hrt实战 17
- 下一步 26
- 第二章 培养出色的团队文化 27
- 什么是文化 27
- 为什么要关心它 30

- 文化和人 33
- 优秀团队文化中的沟通模式 36
- 高层面同步 38
- 每日进行的讨论 46
- 使用bug跟踪系统 50

- 沟通也是工程的一部分 50
- 说到底真正重要的还是代码本身 55
- 第三章 大海航行靠船长 57
- .自然界没有真空地带 57
- @deprecated manager 59

- 主管才是新的经理 60
- 唯一要担心的就是.....好吧，所有的事情 62
- 仆人式领导 64
- 反模式 65
- 领袖的处事之道 71

- 人是植物 89
- 内部激励和外部激励 91
- 结语 92
- 第四章 对付害群之马 94
- 什么是“害群” 95
- 保护团队
- 96
- 发现威胁 98
- 第五章 操纵组织的艺术 113
- 优点、缺点和策略 113
- 理想的情况：团队在公司里应该是怎么运作的 114

- 现实的情况：当环境成为成功路上的绊脚石 117
- 操纵你的组织 124
- b计划：走为上 138
- 不要放弃 140
- 第六章
- 用户也是人 141
- 管理大众的印象 142
- 管理和用户之间的关系 161

结语 172
附录a 延伸阅读 174
EndFragment
StartFragment第一章 天才程序员的传说 1
帮我把代码藏起来 1
天才的传说 2
隐瞒是有害的 6
团队才是王道 11
三支柱 13

hrt实战 17
下一步 26
第二章 培养出色的团队文化 27
什么是文化 27
为什么要关心它 30

文化和人 33
优秀团队文化中的沟通模式 36
高层面同步 38
每日进行的讨论 46
使用bug跟踪系统 50

沟通也是工程的一部分 50
说到底真正重要的还是代码本身 55
第三章 大海航行靠船长 57
.自然界没有真空地带 57
@deprecated manager 59

主管才是新的经理 60
唯一要担心的就是.....好吧，所有的事情 62
仆人式领导 64
反模式 65
领袖的处事之道 71

人是植物 89
内部激励和外部激励 91
结语 92
第四章 对付害群之马 94
什么是“害群” 95
保护团队
96
发现威胁 98
第五章 操纵组织的艺术 113
优点、缺点和策略 113
理想的情况：团队在公司里应该是怎么运作的 114

现实的情况：当环境成为成功路上的绊脚石 117
操纵你的组织 124
b计划：走为上 138

《极客与团队》

不要放弃 140

第六章

用户也是人 141

管理大众的印象 142

管理和用户之间的关系 161

结语 172

附录a 延伸阅读 174

EndFragment

媒体关注与评论

“我和工程师已经打了三十几年交道了。经验告诉我，人际关系在工作中的重要性并不亚于科学技术，但很多工程师都不愿意尝试了解如何与人合作。如果你想要更有效地进行创新，那绝不能错过本书。”——狄恩·卡门，DEKA研发公司创始人“作者为软件开发团队搜集了一套令人惊艳的模式和反模式。无论你是开发人员本身还是经理，或是任何有一点关系的人，只要你觉得自己难以理解为什么这些东西能让团队更有活力，你就应该读一读这本书。它道出了很多出色的开源项目程序员与生俱来的特质。我当初要是有这么一本书就好了。”——布莱恩·贝伦多夫，世界经济论坛CTO“软件开发是一项团队运动。如果你想要扬名立万，市面上有无数本教你如何磨练技术水平的好书，教你当好经理的书也不少。但这本书却另辟蹊径，教你应该如何与团队合作，以及如何当好合作伙伴。这个领域早就需要这样一本书了。”——彼得·诺维格，Google研发主管“如果你想要组建一支能专注开发优秀软件的团队，那绝不可错过本书。作者将谦虚、尊重和信任等感性的话题漂亮地转变成极富技巧的建议，哪怕是最挑剔的工程师也会欣然接受的。”——埃里克·伦特，BrightTag联合创始人兼CTO“这本书太精彩了。它探讨的是计算机编程里最难的问题，怎样和其他程序员打交道。我打算让Samba团队人手一本。”——杰瑞米·埃里森，Samba作者之一“你可能已经听过所谓的‘十倍程序员’传说了吧，它的意思是顶尖程序员的产力比普通程序员要高一个数量级。但巨大的影响力不仅来自经验和技能，更少不了来自同事和用户的共鸣感，而且无论多少聪明才智都弥补不了后者的缺失。好在这本书可以帮你磨练这项软技能，以期给世界留下更深的烙印。”——鲍勃·李，Square支付CTO“作者设定了一个基本的信条——谦虚、尊重和信任——并围绕它们提供了大量的情景案例。这些宝贵的经验和智慧能够帮助绝大多数需要团队合作的工程师（也就是我们）变得更有效率。”——格雷格·巴罗斯，Facebook产品工程部副总裁“软件是由人创造出来的。只要能运用本书中所列出的准则，这样的团队无论是在思路、代码品质上，还是在产品发布方面，都可以完胜任何单打独斗的黑客。程序员们，好好学吧！”——乔纳森·南丁格尔，Mozilla Firefox工程部高级主管“本书是写给程序员看的，一本教你怎么交朋友、怎么影响他人的书。书中充满了操作性极强的建议和意见，让你在技术团队中过得更开心，变得更有效率，更加如鱼得水。”——艾德利安·霍罗瓦第，Django创始人之一“作者说出了我一直在做但是总结不出来的东西。”——吉多·范·罗苏姆，Python之父“请把本书送至：保罗·海宁·坎普FreeBSD核心团队必须在1994年3月之前送达。”——保罗·海宁·坎普，FreeBSD项目程序员“作者无意为孤独的程序员唱赞歌，相反他们决心要亲手埋葬这个传说。他们撰写了一系列文章来指导那些靠谱的程序员如何对付他们这辈子最复杂的难题：怎么处理好和团队的关系。本书说明了为什么最有人情味的软件往往都是最会合作的人创造出来的，而且它还教你如何同时做到这两点。”——约翰·托尔瓦，芝加哥市政府CTO“这是一本有关软件开发社会学的出色著作，它同时照顾到了开源项目和大公司的需求。对所有新踏入职场的工程师来说，有关管理和应对办公室政治的那个部分绝对是必读的。我的建议是不管你是什么背景的工程师都应该读一读这一章！这是我见过的第一本写给工程师看的、专门有讲到办公室政治的读物，而且可读性非常强。在‘怎么和难以相处的人一起工作’里分享的奇闻异事和实用小贴士都是金玉良言！千金难买哦。”——蓝俊彪，ArEngineer's Guide to Silicon Valley Startups和Startup Engineering Manager作者“本书绝对是书中瑰宝，作者分享的理念能让程序员们更好地为团队作出贡献。我们终于有幸可以开诚布公地探讨这个领域的话题，而且还是以这么平和幽默的方式。要是在21岁的时候有这样一本书让我研读领会就好了。”——布莱恩·奥沙利文，Facebook“这本书为建立健康的软件开发文化提供了基本的蓝图。它应该成为工程经理和技术领导的必读书目，甚至那些想要了解团队动力学是如何留住顶级人才以及影响软件质量的非技术主管也不应该错过本书。”——布鲁斯·约翰逊，Google工程主管“编程技术能让你混口饭吃，但要是能把它和与人合作的能力结合起来，你就可以改变世界。这本书教你的并不仅仅是当一个更好的程序员，它还要你当个了不起的程序员。”——克雷·约翰逊，The Information Diet作者“本书就如何构建成功团队和产品的话题进行了极富洞察力的探索，探讨的是多年来我们程序员在职业生涯里都经历过的痛苦和挣扎。这种轻松愉快的办法不但能同时解决技术团队里的技术问题和人际关系问题，更以一种有趣的方式转化成文字，实在是所有程序员书架上必不可少的书目。”——乔纳森·勒布朗，X.Commerce首席工程师“编程现在涉及的已经不仅仅是代码和机器了，它更像是把已有的组件按照新的方式拼装在一起——而这些组件背后的作者都是活生生的人。本书的作者对此了然于胸，无论给出什么样的建议，他们要传达的信息都是非常简单直观的：只要像在代码上那样在人际关系上狠下功夫，你不但可以变成更快乐的程序员

《极客与团队》

，更可以让其他程序员也变得快乐起来。这本书来的正是时候！”——傅凯，Open Tech Strategies LLC联合创始人“很少有人会谈及和极客合作时该怎么处理好人际关系这方面的东西，所以多年来我一直通过博客记录本和傅攀勃在各种大会上的演讲。现在我很高兴能方便地在一本书里就读到他们所有演讲里的精华，而不用再追着他们满世界跑了。”——罗伯特·凯，MusicBrainz首席极客“精彩至极，直指人心！哪怕你不认为自己是极客，它提供的建议也值得一读。”——文顿·瑟夫，Google首席互联网专家

《极客与团队》

编辑推荐

《极客与团队》编辑推荐：Google研发主管、Google首席互联网专家、Samba核心开发者、Django核心开发者、Python之父等众多业内高端人士和顶级程序员力荐，软件工程师的团队生存秘笈。在童话里，写出最佳程序的软件工程师一定也是最成功的人。但是现实并非如此，成功往往还取决于如何与人合作完成任务。作者通过《极客与团队》和你分享了软件开发中与他人合作、与团队协作以及应对用户的种种模式和反模式。本书的两位作者来自Google，具有丰富的项目开发和管理经验，还曾经是Subversion的初创成员。他们都是盛誉正隆的工程师，他们的一系列谈话节目（包括著名的“如何对付工作中的害群之马”）吸引了成千上万的听众。软件开发是一项团队运动，人的因素对结果的影响完全不亚于技术因素。即使你已经在编程领域里磨练十数年也没关系，这本《极客与团队》教你的将是那些往往被忽略的“人”的那部分。只要能够学着与人合作，在软件工程中的所谓“软技能”上多下点功夫，就一定能在工作中事半功倍。

《极客与团队》

名人推荐

“这个领域早就需要这样一本书了。现在终于有了。”——彼得·诺维格，Google研发主管“人际关系在工作中的重要性不亚于科学技术，但很多工程师都不太愿意投入太多精力去了解如何与人合作。如果你想要更有效地进行创新，那绝不能错过此书。”——狄恩·卡门，DEKA研发公司创始人“精彩至极，直指人心！哪怕你不认为自己是极客，它提供的建议也值得一读。”——文顿·瑟夫，Google首席互联网专家

《极客与团队》

精彩短评

- 1、有些价值观不是很认同
 - 2、讲的很不错，一定要多读几遍。
 - 3、谦虚、尊重与信任，全书一直在诠释这三个概念。知行合一不易，知易行难，在工作中慢慢践行这一原则，也许以后会对本书的内容具有更深的理解。
 - 4、最近书单之一。
- 最想吐槽的是译者对人名的翻译，像人名还有一些专属名词完全没有必要翻
总之就是要淡定！实在不行plan b
不过还是讲了很多同感的大实话。
- 5、1
 - 6、很多项目团队的好建议，尤其是hrt谦虚 尊重 信任三个原则非常经典，尽管有一些小观点不赞同，但大体是受益的
 - 7、10分钟读完 第四章有阅读的价值
 - 8、我们经常把编写开源软件比喻成在弹跳床上用扑克牌搭建房屋。这需要稳健的双手、大量的耐心，还有能大声制止那些看也不看就想往下跳的人的决心。
 - 9、写的都是经验之谈吧，总体不错，值得一读。
 - 10、看看
 - 11、极客境界：（1）记住HRT原则：“谦虚、尊重和信任”。（2）优雅的设计应该保持简洁，化困难为可能。（3）隐藏复杂性通常涉及漂亮的抽象，就像Google和Apple魔法界面一样，深刻关注用户在软件启动后30秒内的体验。
 - 12、说出了我心中的梦之队。怎么才能成为其中的一员呢？不断的提高自己。
好像是在机场看完的
 - 13、一本只谈如何沟通、团队文化、团队领导的书。三个观点让我受益颇大：
 - 1，你的自尊不等同于你的代码！
 - 2，隐瞒公司信息对团队极为不利。
 - 3，提出清晰的工作目标。书中反复强调谦虚、尊重和信任，我觉得这与人自身的性格思维紧密相关，需要更深入的自我剖析和修炼。
 - 14、HRT：谦虚、尊重、信任。如何构建一个高效和谐的团队，推荐一读~
 - 15、谦虚、尊重、信任，知易行难，工作日久，警惕你会变成你曾经很鄙视的那个人。有人的地方就有江湖，有江湖的地方就会有纷争！
 - 16、我觉得这个是团队管理的必读圣经，得反复阅读。
 - 17、HRT原则：谦虚、尊重、信任——又是一本知易行难。
 - 18、本书用大量软件开发过程中的实例，示范如何处理软件开发中人的关系。核心原则是谦虚、尊重、信任。P.S. 作者言简意赅，运用了大量有趣的类比。值得寻味。
 - 19、极为贴合程序员出身的管理者，看似内容松散，其实章节组织上极有条理。第五章关于公司政治，透彻。
 - 20、关于团队管理的理念介绍很实用，深受启发
 - 21、整本書的內容都是圍繞HRT，不過人和人之間，總是能和HRT 扯上關係，每章的獨立性其實不是很強，作者在講很多論點的時候，會穿插其他內容。總之，看完也很難記住，只能記住HRT，好好工作，多實踐。
 - 22、教导技术牛人如何与他人友好相处。团队，领导，公司，用户
 - 23、书主要围绕hrt（谦虚、尊重、信任）三个方面为核心进行讨论。在进行人与人之间的交往过程中HRT原则无时无刻不在起作用。个人觉得书的前几章不仅适合程序员读，也适合其他职业的人读。书的最后介绍了用户的作用，一个软件产品脱离了用户就不能称之为产品。一个公司抛弃了用户，也就是衰败的开始。
 - 24、里面的知识，对团队内部如何一起协作工作，很有用
 - 25、Great book!

《极客与团队》

- 26、其实吧，读过以后感触挺深的，像我这样的码农内心自私欲及占有欲太强，是该洗刷一下了，就拿代码头来说，必然会加上author，就好像宣布这块是我的领地，他人勿动。。编码太久后内心深入总是缺乏信任，缺失安全感，数数HRT（谦虚、尊重、信任）所剩无几，真是病入膏肓了啊。。。
- 27、工作中待人处事的态度：谦虚，尊重，信任，这也是构建稳定，高效团队的基础。

书中给出了很多实用的方法，指导研发团队在项目中保持正确的方向。

借用书中的一句话：有人的地方就有江湖，软件开发圈子里面临的问题和其他行业没有什么不同

- 28、第一章读起来很流畅，后面几章逻辑一般
- 29、如果你想要在团队中或者让一个团队生存，那么这本书必定是你的好朋友。
- 30、推荐所有工程师阅读
- 31、以前我的学弟问我,php和ruby还有其他语言有什么优势?为什么Java是开发web的首选语言?我想我可以跟他解释了。
- 32、权当做为了解，没有特别精彩之处。
- 33、极好。看完我想早几年看到这本书多好，可是又想，只有现在的我才能完全理解和共鸣它的内容吧。在合适的时间看到的一本好书，开心。
- 34、高分是给这书站在开发者角度来诠释管理。但很多想法对小团队很难落实下去。
- 35、谦逊，尊重，信任。工作日久，有可能变为当初你鄙视的人，时时警惕，莫忘初心。
- 36、挺不错的书,里面的一些管理经验可以尝试下
- 37、HRT原则：谦虚、尊重、信任
- 38、额，很短。
- 39、有些启发，需要重读
- 40、细细看，书中会有很多与自己的想法碰撞的地方！
- 41、篇幅短小精悍却可以大幅提高人际交往水平的好书
- 42、我发现我特别适合去帮助刚拿完B轮，因为快速迭代而需要梳理偿还一堆技术债的公司。里面提到的实践基本上都在做，不同的是我是自己思考出来的。
- 43、实际就是一个team building的书。说实话不是做项目经理估计没啥看头。
- 44、喜欢向上管理的观点 ~ ~
- 45、mindset
- 46、全书强调的HRT概念，在实验室性质的团队中比较适合。而当前的高科技或者互联网企业正是这种比较强调人的地方。
- 整书读下来，收获颇丰。但觉得这种书的受众是有限的，如果没有真正经历过各种各样的现实的问题，碰到奇葩的人，感受不会太深。
- 47、程序员办公室政治指南
- 48、里面提到的很多点在现任leader上得到体现。很多观点都很中肯，很适合一些对办公室政治不是很清楚，对自己手头工作不知道在忙啥的小鲜肉阅读。
- 49、1.每周讨论
- 50、从code review，bug report到team building等等面面俱到。作者作为SVN社区的主要维护者，有很多开源项目经验分享。
- 51、主要还是讲团建的，内容还是比较丰富的，有点小收获！
- 52、作者从自身的实践中总结出来的管理之道，具有一定的借鉴意义。说的是软件开发管理，也可以借鉴到其他方面。
- 53、贯彻HRT三大原则（Humility谦逊，Respect尊重，和Trust信任）以你为中心，推及至团队，合作人，组织和用户等，维护和营造一个良好社区。值得参考！
- 54、谦虚、尊重、信任
- 55、花两个小时草草过了一遍，有很多观点挺不错的，以后走技术路线了可以recap一下。
- 56、感觉都是些常识...

- 1、这是一本讲团队的书，当然团队不是一般的团队，而是一帮子Geek组成的团队。这帮人坐在一个屋子里可以当面无话可用IRC聊得热火朝天。书的结构很清晰 1. 团队很重要，为人要HRT 2. 建设团队文化 3. 怎么做个好老大 4. 对付害群之马 5. 公司政治学 6. 关注用户。其实最后一章和团队没太大关系（虽然也很有用），其他5章则很实在。读毕虽无醍醐灌顶的感觉，但感触也很深，里面很多的经验是来自开源社区的（Subversion, ASF），这类东西由于以前接触过且很喜欢，因此读来非常亲切，比如维护邮件列表，用IRC而不是qq/旺旺等IM；比如code review；特别是知名社区大家和和气气，不论自己多牛，都是那么有礼貌（Linus例外），反观国内看到的一些所谓技术讨论，很多华而不实、或者qq群那样乱糟糟、或者牛人一出来大家集体跪拜……书中看待办公室政治的视角也值得考虑，其实要做成一件稍微大点的事情，是逃不了政治的，因为有人的地方就有政治，不妨用积极的眼光去看待这个事情，该出手时就出手，对团队，对自己的职业生涯都是有好处的。如果你是个程序员，工作年限长了其实或多或少要关心团队的事情，或者干脆做主管，这个时候除了考虑技术问题，对非技术问题也要理解并积极参与，这书讲得虽然没有太多新意，但都很实在。
- 2、书中的观点应该是很好的，不过这些推荐评论看得让人真恶心！还有，我一直很怀疑这些真正的技术极客们有时间、精力和兴趣（重点是兴趣！）去写这种文采飞扬、煽动人心的团队管理方面的论述性文章……窃以为多半还是枪手代笔，当然并不妨碍观点的正确性
- 3、“精彩至极，直指人心！哪怕你不觉得自己是极客，它提供的建议也值得一读。”——文顿·瑟夫，Google首席互联网专家“我和工程师已经打了三十几年交道了。经验告诉我，人际关系在工作中的重要性并不亚于科学技术，但很多工程师都不愿意尝试了解如何与人合作。如果你想要更有效地进行创新，那绝不能错过本书。”——狄恩·卡门，DEKA研发公司创始人“作者为软件开发团队搜集了一套令人惊艳的模式和反模式。无论你是开发人员本身还是经理，或是任何有一点关系的人，只要你觉得自己难以理解为什么这些东西能让团队更有活力，你就应该读一读这本书。它道出了很多出色的开源项目程序员与生俱来的特质。我当初要是有这么一本书就好了。”——布莱恩·贝伦多夫，世界经济论坛CTO
- 4、“本书是写给程序员看的，一本教你怎么交朋友、怎么影响他人的书。书中充满了操作性极强的建议和意见，让你在技术团队中过得更开心，变得更有效率，更加如鱼得水。”——艾德利安·霍罗瓦第，Django创始人之一“作者说出了我一直在做但是总结不出来的东西。”——吉多·范·罗苏姆，Python之父“请把本书送至：保罗-海宁·坎普 FreeBSD核心团队 必须在1994年3月之前送达。”——保罗-海宁·坎普，FreeBSD项目程序员“作者无意为孤独的程序员唱赞歌，相反他们决心要亲手埋葬这个传说。他们撰写了一系列文章来指导那些靠谱的程序员如何对付他们这辈子最复杂的难题：怎么处理好和团队的关系。本书说明了为什么最有人情味的软件往往都是最会合作的人创造出来的，而且它还教你如何同时做到这两点。”——约翰·托尔瓦，芝加哥市政府CTO
- 5、“软件开发是一项团队运动。如果你想要扬名立万，市面上有无数本教你如何磨练技术水平的好书，教你当好经理的书也不少。但这本书却另辟蹊径，教你应该如何与团队合作，以及如何当好合作伙伴。这个领域早就需要这样一本书了。”——彼得·诺维格，Google研发主管“如果你想要组建一支能专注开发优秀软件的团队，那绝不可错过本书。作者将谦虚、尊重和信任等感性的话题漂亮地转变成极富技巧的建议，哪怕是最挑剔的工程师也会欣然接受的。”——埃里克·伦特，BrightTag 联合创始人兼CTO“这本书太精彩了。它探讨的是计算机编程里最难的问题，怎样和其他程序员打交道。我打算让Samba团队人手一本。”——杰瑞米·埃里森，Samba作者之一
- 6、第一章：天才程序员的传说现今的代码工作再也不是早年个人英雄主义的工作，没有一项伟大的产品不是团队的结晶，没有一位IT天才不是身后站着一个伟大的团队。所以成功的程序员必须是一个依赖团队、帮助团队、成为团队重要分子的人。而在团队合作中我们却经常会犯致命的错误，比如在工作完成前想尽量隐藏代码或想法防止被其他人嘲笑或否定、团队沟通效果很差、自我中心等等。每个人都会对因为害怕自身能力被否定、idear被剽窃而感到不安感，so很多程序员不愿意在工作完成前向其他人分享idear、同享代码。但是作者在众多案例中得到的结论是早分享就会早发现问题，早分享就会早得到帮助，早分享该来的失败就会来得更快，早分享也不会总是重复造轮子，早分享也会成长的更快。所以作为一个期望项目成功的人、期望快速成长的人必须告诉自己隐瞒是有害的，开放才会高效。要时刻告诉自己尽可能的快速多次体验失败成功才会到来。控制情绪往往是工程师群体的弱项

，但是真正高效的团队成员必须是能够在某些方面控制自己的恶情绪或欲望，在合作中保持谦虚、尊重和信任。说的直白一点所谓谦虚就是放下你的玻璃心，当别人说你代码有问题的时候不是在说你不聪明，再说你真的很聪明从来不会犯错吗？所谓尊重就是学会批评别人前先批评自己，别人批评你的代码要认真思考而不是怒气冲天。所谓信任就是相信别人是和你一样聪明努力的，一样为团队着想，一样批评别人是为别人好。 <http://blog.sina.com.cn/u/1371801224>

7、“这本书为建立健康的软件开发文化提供了基本的蓝图。它应该成为项目经理和技术领导的必读书目，甚至那些想要了解团队动力学是如何留住顶级人才以及影响软件质量的非技术主管也不应该错过本书。”——布鲁斯·约翰逊，Google工程主管“编程技术能让你混口饭吃，但要是能把它和与人合作的能力结合起来，你就可以改变世界。这本书教你的并不仅仅是当一个更好的程序员，它还要你当个了不起的程序员。”——克雷·约翰逊，The Information Diet作者“本书就如何构建成功团队和产品的话题进行了极富洞察力的探索，探讨的是多年来我们程序员在职业生涯里都经历过的痛苦和挣扎。这种轻松愉快的办法不但能同时解决技术团队里的技术问题和人际关系问题，更以一种有趣的方式转化成文字，实在是所有程序员书架上必不可少的书目。”——乔纳森·勒布朗，X.Commerce首席工程师

8、总结来说，这本书写的是关于团队和社区的事。什么是团队，为什么要HRT。如何培养团队文化。如何领导团队。如何对付不良行为。如何应对组织问题。如何看待用户。整体来说，把程序员代码以外，与人沟通的事写了个纲领。

9、其实这本书给自己的收获并不多，原因有二：一是自己看了不少介绍软件工程师团队如何工作的书；二是自己在工作中也在不断思考如何才能更有效率地工作，更好地与同事和领导相处；因此，这本书中介绍的内容对于自己来说没有太多的新意。新意没有太多，但是倒是有一二处启发。一，我对团队文化有了更深一步的认识。书中第31页提到：“每当有新人加入时，她并不是只从团队负责人那里了解团队文化，而是从一起工作的每个成员身上学习。例如，你在仔细检查新同事的代码的时候，会向她解释为什么你的团队是按照某种方式写代码的，这样她就很快明白团队重视的是代码的哪些部分。她还会同感观察团队的工作，交流，以及解决冲突的方式来学习团队文化”。自己的理解是：团队文化不是上嘴唇一碰下嘴唇就有的，她是在公司的各个角落，产生于面试的时候，产生于公司如何对待离职的员工，产生于同事之间的争论，她无时无刻不存在。行动是建立文化的唯一方法，而不是标语和演讲！二，搞清楚哪些事情是不能做的，对于团队也是很重要的。我们常常在想自己能做什么，而忽略了自己不能做什么。其实搞清楚不自己能做什么，剩下的就是我们都能做的了。三，面对用户数量上升的时候，不是提供更复杂的功能，而是应该提供更简介的界面、更友好的交互。因为“当用户的数量上升时，他们的平均技术水平会递减，因为有越来越多的普通大众变成了你的用户。如果此时增加产品的复杂度，用户失望的程度会直线上升”。四，老外那里也有办公室政治啊，而且一点也不必国内的差！但是，不管你接受不接受，它都在那里，自己要做的就是尽可能地保护自己不受其伤害。就像作者说的，也许你对目前的处境非常满意，但是获得更高职位会让自己在灾难来临的时候有更大的机会全身而退。备注：人们的认识都是有局限性的，因为人们总是根据自己身边发生的事情去下结论，妄图进而将其推广到所有人那里。但是人们自己经历的事情总是有限的，事情的发生也都有“上下文”的，因此，根据这些事情得出来的结论也会带有一定的局限性。譬如，作者是根据自己在Google的工作经历以及自己的朋友的工作经历写出了这本书，那么，这本书应该比较适合类似Google文化的公司或团队，至于自己目前所在的团队嘛，只有试过才知道行不行啦！

10、“这是一本有关软件开发社会学的出色著作，它同时照顾到了开源项目和大公司的需求。对所有新踏入职场的工程师来说，有关管理和应对办公室政治的那个部分绝对是必读的。我的建议是不管你是什么背景的工程师都应该读一读这一章！这是我见过的第一本写给工程师看的、专门有讲到办公室政治的读物，而且可读性非常强。在‘怎么和难以相处的人一起工作’里分享的奇闻异事和实用小贴士都是金玉良言！千金难买哦。”——蓝俊彪，ArEngineer's Guide to Silicon Vauer Startups和Startup Engineering Manager作者“本书绝对是书中瑰宝，作者分享的理念能让程序员们更好地为团队作出贡献。我们终于有幸可以开诚布公地探讨这个话题，而且还是以这么平和幽默的方式。要是在21岁的时候有这样一本书让我研读领会就好了。”——布莱恩·奥沙利文，Facebook

11、“这个领域早就需要这样一本书了。现在终于有了。”——彼得·诺维格，Google研发主管“人际关系在工作中的重要性不亚于科学技术，但很多工程师都不太愿意投入太多精力去了解如何与人合作。如果你想要更有效地进行创新，那绝不能错过此书。”——狄恩·卡门，DEKA研发公司创始人

《极客与团队》

“精彩至极，直指人心！哪怕你不认为自己是极客，它提供的建议也值得一读。”——文顿·瑟夫，Google首席互联网专家

12、“你可能已经听过所谓的‘十倍程序员’传说了吧，它的意思是顶尖程序员的生产力比普通程序员要高一个数量级。但巨大的影响力不仅来自经验和技能，更少不了来自同事和用户的共鸣感，而且无论多少聪明才智都弥补不了后者的缺失。好在这本书可以帮你磨练这项软技能，以期给世界留下更深的烙印。”——鲍勃·李，Square支付CTO “作者设定了一个基本的信条——谦虚、尊重和信任——并围绕它们提供了大量的情景案例。这些宝贵的经验和智慧能够帮助绝大多数需要团队合作的工程师（也就是我们）变得更有效率。”——格雷格·巴罗斯，Facebook产品工程部副总裁 “软件是由人创造出来的。只要能运用本书中所列出的准则，这样的团队无论是在思路、代码品质上，还是在产品发布方面，都可以完胜任何单打独斗的黑客。程序员们，好好学吧！”——乔纳森·南丁格尔，Mozilla Firefox工程部高级主管

13、“编程现在涉及的已经不仅仅是代码和机器了，它更像是把已有的组件按照新的方式拼装在一起——而这些组件背后的作者都是活生生的人。本书的作者对此了然于胸，无论给出什么样的建议，他们要传达的信息都是非常简单直观的：只要像在代码上那样在人际关系上狠下功夫，你不但可以变成更快乐的程序员，更可以让其他程序员也变得快乐起来。这本书来的正是时候！”——傅凯，Open Tech Strategies LLC联合创始人 “很少有人会谈及和极客合作时该怎么处理好人际关系这方面的东西，所以多年来我一直通过博客记录本和傅攀勃在各种大会上的演讲。现在我很高兴能方便地在一本书里就读到他们所有演讲里的精华，而不用再追着他们满世界跑了。”——罗伯特·凯，MusicBrainz首席极客

章节试读

1、《极客与团队》的笔记-第1页

明星的号召力是很大的。人类会本能地去找寻领导者和榜样，崇拜他们，然后模仿。

假如你一直都是单打独斗的话，你其实是增加了自己失败的风险，而且浪费了自己成长的可能性。首先，你怎么知道自己选的路是对的？

真正做出产品之前不愿分享好创意实际上是一场很大的赌博，你很容易在一开始就犯下很基本的错误，你有可能是在重新发明轮子。

你需要确定自己在做的事情是对的，方法也是对的，而且不是重复劳动。一开始就踏错步的概率总是很高的，越早征求意见和反馈，就越能把风险降低³。记住这句久经考验的原则——“确保失败尽早发生，尽快发生，经常发生”。

“足够多双眼睛可以确保你的项目保持正确的方向。”闭门造车的结果往往是当实现最初的创意后，却发现世界已经完全改变，原本的产品已经失去意义了。

本质上，单打独斗比合作风险更高。相比担心自己的创意被偷走或是被人笑话，你更应该担心自己不是在错误的方向上浪费了大量时间。

要达到合作无间的境界，你首先要学习理解所谓的社交技巧“三支柱”。这三项原则不但是人际关系中的润滑剂，更是所有良性互动与合作的基本。谦虚 没有人是宇宙中心。谁也不是万能的，谁都会犯错。你必须不断地提高自己。尊重 你必须真心实意地关心同事。他们都是活生生的人，他们的能力和成绩都需要得到肯定。信任 要相信别人的能力和判断力，在适当的时候懂得放权。

基本上所有的社交摩擦最终都是由于缺乏谦虚、尊重，或是信任而造成的。

不要低估社交的力量。社交不是勾心斗角，或是操纵别人，它是通过建立起人与人之间的关系来把事情做成功，而且这种关系延续的时间肯定比项目本身更长。

一份出色的事后检讨应该包含以下内容：? 简要? 事件的时间线，从发现到调查，再到最终结果? 事件发生的主因? 影响和损失评估? 立即修正问题的步骤? 防止事件再次发生的步骤? 得到的教训?

建设性批评是任何工程团队成长发展的基石。接受批评是需要一定的自信心的，而我们觉得建设性批评是最容易接受的一种。另一方面，给出建设性批评要比直接批评嘲笑对方困难得多。而且我们也清

《极客与团队》

楚，请别人批评指正是非常困难的事情，大多数人只会觉得你想听的是赞许和褒扬罢了。如果你身边有朋友和同事能在你需要的时候真正提出建设性批评，千万不要疏远他们，诤友难得。

攻击性性格的人（通常）也能够适应比较平和安静的环境，但是比较内向的人却很少能在激烈的环境里生存（或者开心地工作）——这样的环境下，他们的声音不但很容易被杂音盖过，而且会逐渐影响他们参与的积极性。

沟通的指导原则之一就是在同步沟通的时候（比如开会），人越少越好。而在异步沟通的时候（比如E-mail），涉及的听众越多越好。更重要的是，你必须确保项目文档里的信息要尽可能地让所有人都看到。

如果你不花精力好好沟通，最终一定会浪费更多的精力去做一些没必要的工作，或是团队里别人已经做过的工作。

没用的会议和折磨没什么两样。

如果你正打算做一些新的设计，那么尽量把会议人数控制在五个人以下——除非只有一个人可以拍板，否则在超过五个人的会议室里是做不出任何新设计或者决策的。

有关开会的五条小贴士：

1. 只邀请一定要参加的人；
2. 开会前要决定好议程，而且要事先通知所有人；
3. 达成目的后应提早散会；
4. 注意别跑题；
5. 尽量把会议安排在休息时间前后（比如午饭时间，下班前等）。开会前一定要事先准备好会议议程并且至少提前一天通知所有与会人员，让大家知道要开的是什么会。把会议人数降到最低（别忘了同步交流的成本）。我们认识很多人会无视没有议程的会议邀请，包括工程师、工程经理，甚至主管和副总裁。为了能达成会议的目的，只邀请那些真的需要与会的人。有些人在发现与会者实际上是在看E-mail而没有认真开会的时候，会规定不准带笔记本电脑去开会。可惜这只是治标不治本——人们在开会的时候看E-mail很有可能是因为他们一开始就不需要参加这个会议。主持会议的人应该拿出权威来毫不犹豫地（有礼貌地）打断那些跑题乃至妄图独揽话语权的人。要做到这一点并不容易，但却是值得的。最后，也是最重要的一点就是，如果时间尚早而议程上的议题已经讨论完毕，千万别犹豫，立刻散会吧。

强大的团队不是自发形成的，它们都是由团队的领袖和创始人培育起来的

尽管为团队招募到合适的人才和为团队注入正确的价值观都是非常重要的事情，但最后绝大部分能真

《极客与团队》

正成为文化一部分的努力其实都是来自沟通

千万别新官上任三把火

设置明确的目标，让团队同心协力的最佳方法就是为他们写一份简明扼要的任务宗旨。

只要帮助团队设定好方向和目标，你就可以放手给他们更多的自主性，只要定期检查他们有没有偏离方向就可以了。这样你不但可以有更多的时间来处理其他管理事务，还能大幅提升团队的效率。虽然在缺乏清晰目标的情况下团队也能够取得成功，但是这样通常会浪费很多精力，因为每个人努力的方向会略有差异。这会让人觉得很沮丧，导致团队进展缓慢，还会迫使你耗费越来越多的精力来纠正这些偏差。

在直截了当反馈或者批评的时候，表达的方式是确保别人听得进你的意思，不会发生偏差的关键。如果对方被你弄得很防备，他是不会思考自己怎么改进的，他只会想着怎么和你争辩，证明你错了。

不必事事躬亲，但也不能当甩手掌柜。在从个人贡献的角色转变成领导角色的过程中，要做到平衡是最困难的：一开始的时候，你会倾向于自己包揽一切，但是在担任领导职务一段时间后，又很容易养成凡事自己不动手的习惯。在刚刚担任领导职务的时候，你往往需要努力克制自己，把工作交给其他人去做，即使他们比你花更多时间才能完成。这样你不但自己能保持清醒的头脑，还能让团队里的其他人有机会学习。如果你已经有了领导的经验，然后去领导一支新的团队，那么获得团队尊敬，并且跟上大家步伐最简单的办法就是卷起袖管亲自动手——最好是那些没人愿意做的脏活累活。你的简历上或许有一长串的成就，但是没有什么能比得上你参与进来，真正地去攻克一些难题更能让团队了解你的技术和决心（当然还有谦虚的态度）。寻找接班人。除非这辈子都不打算再换工作，否则你就应该为自己寻找接班人。之前也提到过，这件事要从招聘的那一刻开始做起：如果你打算在团队里挑选接班人，那么在招聘的时候就应该雇佣有能力接替你的人，这就是我们常常挂在嘴边的那句话，你应该“雇佣比自己聪明的人”。而找到这样的工程师之后，还要给他们机会来承担更多的责任，有时候还可以暂代一下领导职务。这样一来，很容易就能在他们中间找到最有潜力，并且有意愿要带领团队的人——记住，有些出色的工程师只想在自己的岗位上发光发热，没有当领导的兴趣，这种选择并没有错。我们总是会吃惊地发现一些公司不顾员工意愿，把最优秀的工程师放到管理职位上去。其实这么做往往只会让你的团队失去一名优秀的工程师，平添一名蹩脚的经理罢了。知道什么时候要做恶人。总会有一些（不可避免，同时又经常发生的）棘手的情况，让你束手无策，只想逃避。或许是某个工程师的技术达不到平均水平；或许是某个工程师不管什么事情都要发表反对意见；又或许是某个工程师每周迟到早退，只干30个小时。“再等一下吧，会好的。”你对自己说。“说不定自己就好了。”你会自己给自己找理由。千万不要掉进这种陷阱——在遇到这种情况的时候，你就应该立刻站出来有所行动。这些问题是不会自动消失的，你拖得越久，对团队的伤害就越大。俗话说得好，夜长梦多。等待只会延缓问题的爆发，只会导致无法言喻的损失，所以还不如坐言起行。

保护团队不受混乱干扰。担任领导职务后，通常第一件让你注意到的事情就是团队之外的世界混乱不堪，充满了不确定性（甚至还有点疯狂），以前还是工程师的时候是看不到这些的。

《极客与团队》

帮团队遮风挡雨。保护团队不受各种不确定性的干扰，帮他们挡下团队之外各种琐碎无意义的需求是非常重要的；同样，保持信息的透明度，让他们知道公司的“高层”有什么新闻的重要性也完全不亚于前者。在条件允许的情况下，你应该尽可能地和团队分享信息，但是也不要那些不太可能会直接影响到他们的事情告诉他们，这种组织性的混乱只会让他们分心。

告诉团队他们干得很好。很多新上任的团队主管都不怎么会处理队员的缺点，结果忘记了要经常表扬他们。别人把事情办砸了的时候你要说，把事情干好了也一样要说出来，还要让他（和团队里所有人）知道他干得有多出色。

如果遇到有冒险精神、常常想要尝试新东西的队员，最好的领导常常会用这样的招数：只要回转的余地足够大，就可以批准。

一般来说，一个人总是让自己沉浸在负面情绪里是不健康的行为——长远来讲，它会侵蚀你的一切，制造更多麻烦。

这一章我们讨论了很多场景，说到最后似乎会产生一种偏执的感觉。但是别忘了，这个世界里混蛋其实并不多。正如罗伯特·J·翰龙所说的：不要把用愚蠢可以解释的行为归结为恶意的。这里我们更倾向用“无知”而不是“愚蠢”，但是基本思想还是一样的。就像我们在一开始提到的，把人简单地分成好和坏是很幼稚的。没有什么坏人处心积虑地想要毁掉你的文化——大多数人只是被误导了而已；又或者只是想要得到认可，同时又不太擅长与人交际罢了。不管怎么样，你的任务不是要培养傲慢的态度，把那些没有那么聪明的普通人赶出项目；你的任务是拒绝容忍毁灭性的行为，明确自己对HRT的期望。有智慧的人才能体会其中的差别，而有能力的人才能真正予以执行。

刚接触一个人的时候很难界定他是不是所谓的办公室政治高手，因为这种人往往左右逢源，很擅长和人打交道——所以一开始的时候他一定让人觉得很好相处。一般这种人很会利用同僚或下属，踩着他们往上爬。他会抓住各种机会推卸责任，甚至比抢别人功劳还迅速。通常这种人都是两面三刀，见人说人话，见鬼说鬼话，就是希望给你留个好印象。但要是发现没办法利用你或是操纵你，他就会当你不存在，或是把你当作是威胁，想方设法地在暗地里中伤你。所以只要相处一段时间，这种人就藏不住狐狸尾巴了：他大多数时间都在想办法让自己表现得有影响力，而不是真正地让自己变得有影响力。我们

坏习惯是停不下来的，你只有用一个好习惯去替换掉它。

《极客与团队》

万一发生什么坏事，你的位置越高，全身而退的希望就越大。

千万不要低估了为产品精心设计的初体验，对情感所产生的影响

关注用户，其他的東西自会随之而来。

不要有优越感

保持耐心

营造信任和愉悦的氛围

信任是最神圣的资源，必须悉心呵护、步步为营。任何举动都要三思而行。决策的时候，眼光要长远，不要只注重眼前利益。同样，愉悦是另一种能大大改善用户关系的感情。它能提升温暖、无法言语的感觉，让团队更加人性化。

营销 了解用户对你的软件的印象是怎么样的，这决定了他们是否愿意尝试你的产品。易用性 如果你的软件不好用，速度不够快，界面不友好，访问不方便的话，用户最后一定会离你而去。客服 主动和老用户保持联系能影响软件的演化，决定你是不是能留住他们。

2、《极客与团队》的笔记-领袖的处事之道

如果你已经有了领导的经验，然后去领导一支新的团队，那么获得团队尊敬，并且跟上大家步伐最简单的办法就是卷起袖管亲自动手——最好是那些没人愿意做的脏活累活。你的简历上或许有一长串的成就，但是没有什么能比得上你参与进来，真正地去攻克一些难题更能让团队了解你的技术和决心（当然还有谦虚的态度）。寻找接班人。除非这辈子都不打算再换工作，否则你就应该为自己寻找接班人。之前也提到过，这件事要从招聘的那一刻开始做起：如果你打算在团队里挑选接班人，那么在招聘的时候就应该雇佣有能力接替你的人，这就是我们常常挂在嘴边的那句话，你应该“雇佣比自己聪明的人”。

3、《极客与团队》的笔记-第17页

HRT实战：

1. 放下自负
2. 学会批评和接受批评
3. 快速失败；学习；迭代：不要等到完美的时候再出来... 正确的记录错误，能让他人避免重复历史
4. 为学习预留时间
5. 学习保持耐心
6. 对影响保持开放的态度。接受意见改变自己没什么大不了，不要随意挑起争斗

《极客与团队》

4、《极客与团队》的笔记-优秀团队文化中的沟通模式

沟通的指导原则之一就是在同步沟通的时候（比如开会），人越少越好。而在异步沟通的时候（比如 Email），涉及的听众越多越好。

5、《极客与团队》的笔记-第70页

相比担心自己的创意被偷走或是被人笑话，你更应该担心自己是不是在错误的方向上浪费了大量时间。对于这句话，非常有体会，在充满复杂性和不确定性的世界上，要快速的应对变化，不但需要以人为本的设计，还需要快速的实施能力，而且最重要的是，从人类本身甚至生命诞生的历史就说明，不断的尝试，试错，快速的，频繁的，多样化的不断尝试，才最终积累并且形成了如此复杂的有机世界。

从另外一个角度来说，开发软件虽然是指挥机器运作，其实真正本质的过程是，人之间的沟通，如何理解复杂的架构，HRT是非常好的核心。

6、《极客与团队》的笔记-用户也是人

如果你想吸引别人来使用你的软件，你就必须去关心他们对你的软件的情感诉求，而试图改变人们做决定的方式则是不现实的。

承诺的时候要谨言，做产品的时候要超出预期不能让营销部门的人口无遮拦地放卫星，要是用户问起新特性或是发表时间，一定要抓住机会给出最保守的估计。

软件好不好用：身为一个工程师，你绝对不是评估软件可用性的最好人选。软件速度很重要

最好的软件之所以成功是因为它解决了一个很特定的问题。

用户可以轻易的从应用程序里导出自己的数据，然后带着他离开。当用户量上升时，他们的平均技术水平会递减，因为有越来越多的普通大众成了你的用户。再加上不断增加的复杂度，用户失望的程度会直线上升。信任是最神圣的资源，必须细心呵护、步步为营。任何举动都要三思而行。决策的时候，眼光要长远，不要只注重眼前利益。

7、《极客与团队》的笔记-第21页

失败是可以接受的，如果没有经历过失败的话，说明你的创新还不够，或者你承担的风险还太小。

8、《极客与团队》的笔记-对付害群之马

转移“完美主义者”的注意力：我们决定以这个设计作为起点开始工作，看看效果如何。希望你能帮助我们解决在这个过程中出现的各种问题。随时欢迎提交补丁——开源社区里让人闭嘴的委婉说法。别去搭理那些挑衅的家伙，别用感情用事，抓住重点，对付挑衅要不卑不亢。知道什么时候应该放弃。

9、《极客与团队》的笔记-现实的情况：当环境成为成功路上的绊脚石

如果你是那种喜欢保持低调、专心干活的人，可偏偏身边就有那么一位高手的话，你最好改变一下这种工作方式。你或者会觉得不想“玩这种勾心斗角的把戏”，懒得去争权夺位，可最后却发现那些小丑变成了你的上司，结果你身边不但有玩弄办公室政治的家伙，又多了一个坏经理。

10、《极客与团队》的笔记-培养出色的团队文化

团队文化不仅仅是成员们编写代码的方式或是成员之间的相处之道，它还包含了所有人都认可的

《极客与团队》

经验、价值观、目标。<每个人都是团队文化的一部分，都要为定义、维护和保护她做出贡献...建立起一种团队文化，让他们可以放心的分享创意，并且在决策过程中用发言权...意味着某种程度上的“共识驱动管理”...共识是指每个人对产品的成功抱有强烈的主人翁精神和责任感，同时团队的领袖也真的愿意倾听团队的意见建设性批评是任何工程团队成长发展的基石。接受批评是需要一定信心的，接受建设性批评更容易一点。另一方面，提出建设性批评比直接嘲笑对方要难。

优秀团队文化中的沟通模式：

沟通的指导原则之一就是在同步沟通的时候（比如开会），人越少越好。而在异步沟通的时候（比如Email），涉及的听众越多越好。

任务宗旨：任务宗旨可以保证项目不会随着进展而偏离目标，但它不是一成不变的...任务宗旨应该与时俱进，及时反映公司或者产品的变化。

关于开会：1. 邀请一定要参加的人

2. 开会前决定好议程，并事先通知所有人

3. 达成目标后提早散会

4. 注意别跑题

5. 尽量吧会议安排在休息时间前后千万不要低估了面对面交流的力量

11、《极客与团队》的笔记-大海航行靠船长

如果你想要主导自己产品而不是当一名随行者的话，就必须学会掌舵，否则只会令自己和项目搁浅。经历若能表明对手下员工的信任，他们就会感受到正面压力，并努力让自己对得起这份信任传统型经理关心的是怎么完成任务，而主管只关心完成了什么任务...量化管理工作比数数要困难的多，不用把团队的成功据为己有，相反，让他们开心有动力，才是你的主要工作。

反模式：雇佣听话的人无视表现不佳的人无视人际关系和谁都是朋友降低招聘标准把团队当做小孩子领袖的处世之道：放下自负做一个禅师称为催化剂当一个导师设置明确的目标坦诚：不要用汉堡包批评法

12、《极客与团队》的笔记-第三章 大海航行靠船长

(Kindle Locations 1097-1101)只要把你公司的组织结构图想象成一系列齿轮，就可以轻松地看到它的效果。写代码的工程师是一边那个小小的齿轮，只有几个轮齿，他上面每一级的经理都是一个更大一点的齿轮，而最末端的大齿轮是CEO，拥有几百个轮齿。这样，工程师之上的“经理齿轮”（可能有十几个轮齿）转一圈，“工程师齿轮”就要转两到三圈。而CEO只要动一点点，就能让六七层之外的工程师齿轮飞速旋转！你转动的齿轮离底端越远，就能让下面的齿轮转得越快，不管是不是本意如此。

组织结构如同齿轮组。老板是大大齿轮而码农是小小齿轮。大齿轮带动小齿轮。大齿轮转一圈，小小齿轮要转很多、很多圈。

13、《极客与团队》的笔记-第11页

工程师往往不愿公开自己做到一半的项目。这样做的原因是缺乏安全感，不喜欢自己做到一半的事情被人指手画脚。

但其实，关于天才的传说，描述的都是共同完成某个伟大项目的团队中的一个榜样或者领袖而已。没有团队的力量，是不可能完成该伟大的项目的。

尽早分享，不仅可以防止在一开始就误入歧途，还可以强化所谓的公车因子。公车因子：一个项目里，需要有多少人被公车撞到才能令其完全瘫痪。总而言之，单打独斗比合作的风险更高。相比担心自己的创意被偷走或是被人笑话，你更应该担心自己是不是在错误的方向上浪费了大量时间。

14、《极客与团队》的笔记-大海航行靠船长

找到对的那个人的成本（不管是付钱给面试官的钱，还是花在广告上的钱，还是寻求推荐的费用）比起招到一个不应该招的员工的代价来绝对是微不足道的。

15、《极客与团队》的笔记-第166页

确实如此，和用户沟通，就是这样，沟通关键不是试图将自己的理解和术语、词汇系统强加给用户，沟通和尊重至关重要。这几年似乎确实学会了很多这方面的方法，理解意味着能够并且愿意努力将对方用自己发明的词汇系统来描述的事物和行为，翻译成自己已经掌握的一套虽然更强大但是却需要花时间来理解的系统。其实这样实现顺畅沟通的过程确实是美妙的。

强大的技术也是，最终一定是要实现更加优雅的、简洁的表现形式，同样是美妙的，就像魔术一样，所以测试完的模块集成的时候，第一感觉总是见证奇迹的时刻到了

16、《极客与团队》的笔记-第14页

软件开发是集体项目。

团队合作才是软件开发成功的捷径。

团队合作和沟通三个支柱是：谦虚、尊重、信任。即HRT原则

17、《极客与团队》的笔记-操纵组织的艺术

学习向上管理：有必要花点时间在向上管理上。这就是说，你要尽可能确保你的经理以及团队以外的人不但知道你在干嘛，还要知道你干的很棒。在做承诺的时候要谨慎，而干工作的时候要尽最大努力。

关于重构

不要花太多时间在这种防御性的工作上（重构），很少有人看重这些，到时候你会发现自己很尴尬，因为花了那么多时间，你却拿不出什么（政治上）看起来很重要的成果。这样你不但得不到别人的认可，还很容易导致自己的项目被取消。这不是说所有的防御性工作无足轻重，不过是这些工作在团队之外的人眼里是如此罢了。不管技术债务有多少，团队也永远不应该花超过三分之一甚至一半的时间和精力去做防御性的工作，否则就等于政治自杀。

运气和互惠经济互惠经济最有意思的地方就在于，哪怕离职了，你的账户也不会被清空——你还是可以在需要的时候找他们帮忙。这也是为什么在离职的时候千万不要自断后路，不管当时有多冲动。晋升到一个安全的位置：在公司里的位置越高，你就越能掌控自己在公司里的命运。在对自己的职位感到满意的前提下，稍微投资一点精力在获得晋升上面是保护自己的好办法...即使晋升是个很主观和不确定性很大的事情，你还是可以取做很多事情来提高概率的...万一发生什么坏事，你的位置越高，全身而退的希望就越大。向忙碌的管理层求助：越短的邮件就越有机会得到回复。申请小马的步骤：*

我们缺少小马

* 缺少小马让大家很郁闷

* 拥有小马可以提高生产力

请给我们一批小马。

只做正确的事情，随时准备被炒：Google的新员工常常会问我，为什么我的工作这么有效率。我总是半开玩笑的答道，很简单，我只做对Google和这个世界来说正确的事情，做完之后我就等着被炒鱿鱼。要是没有被炒，那说明我做的事情对大家都有好处；要是真的被炒了，那说明这个不是我想要服务的雇主。

18、《极客与团队》的笔记-第25页

1113434113412

3

4343

23444

2335553

555654

19、《极客与团队》的笔记-操纵你的组织

创意是非常吸引人的东西，如果你不在意是不是一定要让别人知道是你原创的话，创意往往能变得非常有活力！有时候你会发现人们谈论创意只是想要让人觉得这是他想出来的，这时候你就要作出选择，到底是要原创还是要让创意传播。尽管听到自己的想法从别人（可能这个人还有点让你看不起）的口中说出来会叫人很痛苦，但很多时候这是唯一让创意传播的办法。

《极客与团队》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com