

图灵电子书

大道至易

实践者的思想



“你是66%的从众者吗？”

第二版

周爱民 著

本书经作者授权，由北京图灵文化发展有限公司发行数字版。

本电子书许可第三方（含个人）在对本作品不作任何修改（含水印、增页等形式）的前提下自由分发，并许可在保留有关版权和著作权信息的前提下分发作品的部分内容。

本电子书不可印刷成纸质书籍或用于商业行为。

本电子书封面用图来自 [OPENCLIPART.ORG](https://openclipart.org/)，并遵循 CCZ 1.0 开放协议。

作者保留本作品以非电子书形式出版发行的权利。

版权所有，侵权必究。

内容提要

本书可以看成是《大道至简》的姊妹篇，是以软件工程为体系、以组织结构为视角的全新著述。全书围绕对人的理解、对项目目标的理解，以及对工程整体性的理解三个大的方向，构划了一个相当宏大的论述结构。在具体的讨论中，作者仍然采用其早期著作《大道至简》中“着笔于章句之末”的写作手法，置喙于文中而留意于言外。并且延续了《大道至简》一书有关广义工程和具体工程的讨论，进一步地引导读者回到“工程的本质问题是组织”这样的视角。

本书适合各类工程管理人员阅读。

序 1：大道至易，落地为要

和爱民兄打交道已经十年有余，在几年前有幸阅读到他的《大道至简》时，就有一种醍醐灌顶的感觉，还向很多搞技术的朋友做了推荐。

后来因为工作关系，我从技术编辑往管理方向转，对技术文章看得少了，对组织构建、团队建设有了更多认识。回头来看，其实组织构建和软件架构设计一样，都强调“原则”，在一个大的方向指导下，精雕细琢、每日精进就好，并没有那么多大道理可言。爱民兄这本新书的名字“大道至易”，真是一语道破天机。

但正如韩寒在他的电影《后会无期》中说，“我们听过无数的道理，却仍旧过不好这一生”。估计每一个团队领导者都知道做公司不外乎招人、留人、育人，做业务也不外乎为用户、为客户创造价值。而每一个架构师也都懂得做架构，就是要安全、要稳定，要有效支撑业务的发展。可是即使这样，我们依然每日为团队建设所累，为架构的各种不完善所累。

我想问题就出现在抵达“易”之前，我们能否抽丝剥茧，看透问题的

实质，找到事物发展的“第一原理”，并毫不犹豫地扎进去，踏踏实实地去打造每一个细节。如果让我给“大道至易”来一个对仗词，我会选择“落地为要”。就如同经常有人问我，“极客邦科技现在做得很棒，和很多企业建立了合作，用户也很认可，做什么事情几乎都是一呼百应”。岂不知这家公司到现在已经走过了十年，在过去的漫长岁月中，它匍匐前进，做了太多不为人知的“脏活累活”，被无数专家和企业拒绝。但就是在这样每天“乐此不疲”的努力中，才有了前面所说的“一呼百应”，才有了“易”。

大道至简，大道至易，落地为要，与诸君共勉。不论你是一个软件工程师、架构师、团队领导者，还是一个创业者，相信都能从本书有所收获，共同成长。

极客邦科技创始人兼 CEO、微信号“精进学堂”作者

霍泰稳

2017.05.26

序 2：个中无我

镜之用在鉴，但若相信个中有我，便是愚人了。

周爱民

2012.01.27

关于本书

1. 从第一版说起

对于本书的第一版，我不满意处颇多，这有很多的原因，而首当其冲的问题就在于整本书的结构。

我对软件工程的早期思考带来的结论是：作为组织角色之一，工程管理者对工程本身其实影响有限。这促使我重新反思我的整个职业生涯，于是在《大道至易》中采用了“实践者的思想”这样的副标题——这“实践者”为读者设定了三种角色：工程管理、软件研发和架构设计。它们其实是我在书中的一个影像：我的确在多年的工作中反复经历这些角色。所以《大道至易》对工程的讨论本质上是“基于组织视角”的，并且会有一篇名为“领域角色的关注”的总论，以讨论这一视角下的工程模型。

换个说法：我把自己写进了书里。

镜之用在鉴，书之用在读。当我执著于书中有没有我的思想，以及是否完整地体现了那些思想时，我就忘却了读者在这本书中的位置——当他们在读这本书的时候，是有各自的背景、视角、观见与所

得的。而我一己私念地要求他们同时理解三种角色，要求他们有着对这些角色总体的、系统化的设问与思考，这在大多数情况下是做不到的。为此我甚至建议一部分读者把书“撕开来读”。

然而这些除了证明这本书“匪夷所思地难读”之外，还有什么意义呢？

2. 第二版

从现在开始，原书（《大道至易》第一版）被分成了三本书，以电子书的形式由图灵出版公司发行。这些书采用了免费共享的授权（是的，你可以很自由地分发它），分别面向各自的读者群体。它们不但在内容上独立清晰，而且在行文上各持风格，再也不受原书的影响——例如我恢复了所有章节、小节的标题，分别设计了它们的封面等等。

这三本书分别是：

- 《大道至易——实践者的思想》（第二版）

本书，约只有原第一版的 1/4 内容。面向软件工程相关角色。

- 《程序原本》

面向程序员、软件设计师。

- 《我的架构思想——基本模型、理论与原则》

面向架构师。

如果你仍然期望获得第一版的纸质书，可以考虑使用网上书店中的“按需印刷”服务。我确信这是可行的，但是除了收藏价值之外，它的内容与组织形式并不比本电子书更好。当然，真诚地感谢你对我，对本书，以及对出版业的支持。

3. 标题

尽管从这一版中开始恢复了各章节与小节的标题，但仍请读者小心这些标题中的陷阱。正如第一版中所提到，“我只是拿这些标题来搭了个架子”。这样的架子，很大程度上可以帮助读者探索那些真实的内容，但并非这些内容的梗概或主旨。

例如本书的最后一个小节，标题取自陆游诗句“纸上得来终觉浅，绝知此事要躬行”的后半句。这是提醒读者书上内容总是皮表，还需要在具体工程中去实践发掘，不可将书中观点奉为圭臬——这确实是学习之道，而非谦辞。这标题用在本书最后一节当然是不错，但是它并不能表现该小节的其它观点：

- 管理不是技术方法，是所谓“真正的管理者，是不看‘管理的技术书’的”。
- 管理要一以贯之（甚至是贯彻貌似错误的东西，这样才有机会找到问题真相并着力修正）。
- 管理与组织的趋同性、不变性本身，正是求变的机会。
- 找到组织自己的目标。
- 多问“你要什么”，而不是追随某种学术定义、组织指令，或他人的需求。
- 所谓“具体工程”，是“每一个”在探索过程中的工程（它没有在学术上的统而一之，而是你正在做的、为之负责的、具体而微的那个案例）。

如同这个段落，书中很多小节都是在“把自己的思维过程和喃喃自语很原生态地记录下来”（孟岩语）。因此，要找到一个确切的标题来概述其要，难免挂一漏万。我一再细述这关节，只期望读者能视之为线索，深思有得，浅知无益。

4. 历史

撰写本书的最初冲动来自于对它的前作《大道至简：软件工程实践者的思想》的进一步反思。在与众多工程界的专家学者进行广泛的讨论之后，《大道至简》一书的问题被锁定在“什么是组织问题”上面。因此，在 2007 年之后，我一面在《大道至简》的新版中提出它面向软件工程的最终、最完整的概念“具体工程”，另一方面，也开始思考“领域角色的关注”。

所谓“领域角色的关注”，也就是“你在哪里？你是谁？在做什么？”的问题。

对于我来说，这个问题的答案在 2007 年稍晚的时候就已经有了。是时，我已经开始使用“工程的组织视角”模型来分析问题（这个模型请参见《我的架构思想》中的“附三：超越软件架构——组织与架构”）。然而，我当时只有能力得到与使用这个模型，却无法解释它。

解释这个模型，正是《大道至简：实践者的思想》一书的最初设想与动力。但最开始，我只是画出了这个模型，并把它的原始稿投进一个箱子里，封存了起来。

从 2009 年开始，我在阿里巴巴旗下的支付宝任业务架构师工作。这使我得以接触到与之前在盛大完全不一样的组织模型与策略。因此，上面这个问题及其答案就因为有了不同侧面的观照，瞬时变得清晰起来。于是最迟至 2010 年底，我就开启了撰写本书的计划。不过它是从撰写《程序原本》这个部分开始的，亦即是开始于我对程序生涯的彻底反思，解决的是上述“领域角色的关注”中“在做什么”的那个问题。

到了 2011 年年中，我从阿里巴巴离职的时候，这本《大道至易》已经完成了它的 1/3。接下来，我离职在家，用了一年多的时间，完成了它剩下的部分。——到了 2012 年 4 月，我与 Joy 在长达一月的旅途中结束了这本书的全部内容。包括《大道至易》第一版的封面用图，就来自于这个旅途中在稻城亚丁拍摄的一张照片。

如同这张照片所展示的，这一切不过是最初那些点滴思想的涟漪。

5. 其它

本书同时发行了多种格式的电子版，包括 ePub、eBook、PDF、mobi 等。如果你使用桌面电脑，我推荐你阅读 PDF 格式的电子版。PDF 格式的电子版采用了标准纸质书的排版方法，包括页眉、页脚、奇偶页设置，以及目录等等，因此建议阅读时使用“双页”版式并设置“首页不同”（在 Macbook 的预览中是默认的）。另外，本电子书也针对 Kindle 进行过优化，全部插图采用了.svg 矢量格式，确保在 Kindle 中阅读效果最佳。

这本电子书的制作过程是：

- 使用 Markdown 制作全书正文，使用 PowerPoint 制作全部插图并转为 PDF；
 - 使用专门工具将 PDF 转换为 PNG 和 SVG 格式
- 使用 Pandoc 将 Markdown 转为 ePub 格式
 - 使用专门工具对.epub 文档进行优化处理，包括脚注处理和高精度与矢量图片重置等
 - 使用 KindleGen 将优化后的.epub 转换成 ebook(azw3/kf8)电子书格式

- 使用 Calibre 将优化后的.epub 转换成 mobi(azw/kf7)电子书格式
- 使用 Pandoc 将 Markdown 转为 Microsoft Word 格式
 - 重置所有图片为高精度、重新排版，以及生成目录等
 - 导出成 PDF 格式
- 制作封面图片等。

我推荐使用 Typora 来写 Markdown 文件，它有着优异的书写体验。

最后，如果没有严重的错误，我不会发布本书的修订版。——如果确实需要的话，它们会出现在我的 Github 仓库中，而不会有公开的发行。此外，由于使用 kf7 格式的.mobi 电子书实在效果太差，所以它也只是出现在我的 Github 仓库中，而并不公开发行。

致谢

感谢所有的读者、编者以及一直以来支持我的朋友们。

感谢霍泰稳先生为本书作序。

感谢图灵出版公司。

感谢我的中学语文老师。

感谢 Joy。En，我最爱的妻。

目录

- 序 1: 大道至易，落地为要..... - 1 -
- 序 2: 个中无我..... - 3 -
- 关于本书..... - 4 -
- 致谢..... - 10 -
- 目录..... - 1 -
- 引言：管理中的逻辑..... - 1 -
- 篇一：任人治事，组织行为的基本认知..... - 5 -
 - 第 1 章 刺秦与灭秦..... - 6 -
 - 1.1 求活而不是志死.....-6 -
 - 1.2 集体目标对某些个人来说是无意义的.....-7 -
 - 1.3 张良是失败的产品经理.....-8 -
 - 1.4 力士之死.....-10 -
 - 1.5 “成功”与“忠诚”.....-11 -
 - 1.6 信任是组织的基本需求.....-12 -
 - 1.7 吴起爱兵是宣传策略.....-14 -
 - 1.8 专业 = 做该做的 + 该做的都做得成.....-15 -

第 2 章 看到别人能做什么	- 17 -
2.1 田忌赛马不仅仅是个娱乐项目	- 17 -
2.2 马性	- 17 -
2.3 在不讨论整体时，某些具体行为可能是逻辑正确却没有存在的必要性	- 19 -
2.4 架构师的职责：将“切实有效的架构”解构为具体可行的“系统化方法”	- 21 -
2.5 团队是“管”不好的	- 24 -
2.6 组织行为学中的 leader 与 owner	- 25 -
2.7 不同的利益追求是组织角色的基本差异	- 28 -
第 3 章 要做事，不要管理	- 30 -
3.1 从团队的视角来看问题	- 30 -
3.2 要知道国共也曾互称“友军”	- 30 -
3.3 你的作为，是否对得起你的权力	- 33 -
3.4 要么用对人，要么就不用	- 33 -
3.5 再解“跟随蚂蚁”	- 36 -
3.6 “理性利己主义者”的行事逻辑	- 38 -
3.7 君子和而不同	- 39 -
3.8 尺长寸短，人尽其用	- 41 -
第 4 章 伯夷与叔齐是怎么死的	- 43 -
4.1 求活的逻辑	- 43 -
4.2 没有产品或业务，还要做项目干嘛？	- 44 -
4.3 项目的核心是事	- 47 -
4.4 求死之道	- 48 -
4.5 不但要听 BOSS 的，还要听明白	- 49 -
4.6 别抱怨	- 50 -
4.7 现实	- 51 -
4.8 更多的现实	- 52 -

4.9 工程只是一个可选手段.....	- 52 -
篇二：谋定后动，项目的存在权.....	- 55 -
第 5 章 试错通常是无能的托辞.....	- 56 -
5.1 知州孙觉的计谋.....	- 56 -
5.2 道德目标通常是障眼法.....	- 56 -
5.3 被撤职的 CEO.....	- 57 -
5.4 势者，利之所趋.....	- 58 -
5.5 随势就利、投其所好皆不过是手法而已.....	- 59 -
5.6 你无法讨好所有人.....	- 61 -
5.7 错了就是错了，哪有什么“试错”.....	- 63 -
5.8 再论“做正确的事”.....	- 66 -
5.9 “虽有智慧，不如乘势”.....	- 68 -
第 6 章 合法的山大王为什么没能成功.....	- 71 -
6.1 “这个团队大概是没什么希望了”.....	- 71 -
6.2 是温伯格解答了我的疑问.....	- 71 -
6.3 你是“66%的从众者”吗？.....	- 72 -
6.4 给“1%的人”责权.....	- 73 -
6.5 领导者的眼光.....	- 75 -
6.6 山大王的诞生.....	- 76 -
6.7 方向明确，制度有效，知道怎么做，以及知道做什么.....	- 78 -
6.8 最后：别让兵掉到水里.....	- 81 -
第 7 章 自己想办法.....	- 83 -
7.1 三块贝壳.....	- 83 -
7.2 佛祖的“过程论”.....	- 84 -
7.3 将“最佳实践”工程化，本质上去掉了它“立可时用”的特性.....	- 85 -
7.4 先拉完屎再说.....	- 86 -

7.5再谈《人月神话》	-86 -
7.6敏捷工程的成功之处	- 89 -
7.7拉屎就算是一项工程，具体来说，也得自己拉	- 90 -
篇三：具体而微，工程是系统而不是事	- 93 -
第 8 章 做事的选择	- 94 -
8.1开诚布公地谈“权”	-94 -
8.2没有“凭什么”	-95 -
8.3修身的智慧与做事的素质	-96 -
8.4你可能只是建功者，而非创业者	-98 -
8.5马援之死，以及死透了之后	-99 -
8.6谋事、成事，以及忠其事	- 101 -
8.7组织：在形式上的、团队的构成	- 104 -
第 9 章 你要什么	- 106 -
9.1长期的组织战略	- 106 -
9.2为 1 万人做布局	- 108 -
9.3组织机会	- 108 -
9.4棋子之所以为棋子，在不见棋眼	- 110 -
9.5时者，机之所发	- 111 -
9.6任免	- 112 -
9.7此事要躬行	- 113 -
附一：行在道上，从局部到全局	- 117 -
附二：本来面目——大教堂、集市，与作坊	- 120 -
附三：杀不死的人狼——我读《人月神话》	- 123 -
附四：工程的本质问题是组织	- 125 -

引言：管理中的逻辑

“不太逻辑的思维”，是不是我们用以逃避问题的一贯策略？

不同角色对人的思维方式的要求是不一样的，例如你不能要求音乐家如同数学家一样地去思维——这当然存在特例，但追问与研究它通常是科学家的事情。对于大多数人来说，意识到“自己可能不擅长从事某类工作”，是相当令人沮丧的。

没有人承认自己是笨蛋，于是“思维缺乏逻辑”便成了一种逃避软件开发（或其他某些类型）角色的一种托辞。然而就本质上来讲，逻辑思维事实上是每个正常人的基本能力，例如《大道至简》中谈到过的起床穿衣刷牙等这类日常行为中就包含了大量的逻辑与思维活动。

但是真正深入地讨论“逻辑”却远非易事。西式逻辑学在最早传入中国时被译作“西方名学”，这是从古典的“名实学”继承过来的一种说法。但是“名实”并不能完整地表达这一学科的内涵，于是在后来的翻译中便渐渐地采用了 **Logic** 一词的音译，亦即是“逻辑”。所以逻辑二字，在中文的文字学上其实是无解的。反观“名实学”，它其实只是部分地表达了逻辑的涵义：所谓名，就是一个物的名字或指称；所谓实，就是一个物的本身。

在逻辑学的概念中，“桌子”既能指代现在你眼前的办公桌，也能指

代我家的餐桌。那么当我们都用“桌子”这个词的时候，究竟是在“谈什么”就成了大问题。“名正则言顺”讲的就是这么个道理，亦即是说我们要在讨论“东西”之前，必须先说清楚这些东西究竟“是什么”。“是什么”在逻辑学中，就是概念定义。

清晰的逻辑有助于我们分析事理，但并不见得能让我们发现“事、情”的端倪。逻辑中的“事”必有序，“情”必有势，然则我们现实中的事情大多是处于一个无秩序、无态势的境况下，以及处于这境况下的一个过程片段之中。因此往往地，“析理”只是经营管理者的一个手段或工具罢了，它能解决一些问题，但远非全部。

这也带来一种“状况”，如我常常说的：真正的管理者，是不会看管理的技术书的。但是，正如把管理当“技术”来讲是一个强调逻辑的极端，在另一方向上的极端，则是把管理当成权谋术来讲。“管理学”总的来说分成相当多的流派或风格，不可否认的是，“技术工程”与“权谋政治”都是其中的典型方法。但——尤为重要的一——这样的管理学派本身，也只是把管理当学问来讲，求学者也只是当成技术来学。

这仍然是逻辑思维的问题。我们总得找个确指的东西——例如流派或风格——才能讲清楚我们“怎么管理”。然而事实上的管理中，我们面对的是一个确指的背景、一群确指的人，以及一些确指的事。因此具体到管理者，无论是什么流派风格，亦无论是什么方法工具，常常是拿来就用，听人家说好便是好的。若是按照“逻辑一些”的做法，大概应该先搞个样本，做个试验，出个统计数据，并证明一下有效性，然后再实施。但如果真的这样“很逻辑”地去做，大概是什么结果也不会有。

具体做事，毕竟不是做学术讲原理，而只是在“解决问题”。谁都明

白，管理学跟数学是不同的。但不同处究竟在哪里呢？在我看来，对于数学来说，特定的过程总会产生特定的结果，函数的可靠性就在这里，计算的可靠性也就在这里。但对于管理学来说，即使“特定的过程会产生特定的结果”，这个过程本身，也只是可选的。

“特定的过程”是一个工程问题，工程是一种可选的、实施的、做事的方法。

如上，我们用“还算有点逻辑”的方法来说明了管理本身的逻辑缺失，或者说管理实践中“不太科学”的某些方面。然而，更进一步地，事实上的管理者并不会关注这些推论与逻辑叙述的过程——大多数情况下，他们是“意识冲动”型的。

行动可以去规划，决策可能会慎细，但“事、情”的开端呢？

篇一：任人治事，组织行为的基本认知

孙臏是一个策士，献计给田忌将军，一场赛马赢得了齐威王的千金赌注。孙臏最后成了兵法家，田忌逃楚，齐国名列战国七雄之首，故事讲到这里便成了终局。

马呢？

一场竞技，成事败事的都在那几匹马，但他们没有在历史上留下任何东西。除了有一个人曾经认真地观察过它们——司马迁的《史记》在写孙臏献计之前，还写道：“孙臏发现他们的马脚力都差不多，可以分成上、中、下三等。”而正是因为这种观察，孙臏胜了，成了诸子之一。

工程中的各个角色，在工程或成或败之后纷纷散去，成为下一局赛事中的、终将继续被遗忘的行动者。那些指挥行动的人，拍着脑袋，说着畅想，用着败了局的失败手法，或学着胜了局的得胜策略，试图挽救一个又一个所谓伟大的工程项目。

但是，马呢？

田忌不是因为一个策略而得胜，孙臏不是因为一个策略而得名。一场赛事或者一个项目成败的关键之一，在于对角色的观察——了解他们，知道他们该做什么，能做什么，以及能怎么做。但像孙臏这样了解他们的人，却寥寥无几。

第 1 章 刺秦与灭秦

1.1 求活而不是志死

张良是位儒生，手无缚鸡之力。他找来一个大力士，做了一个百余斤重的大铁椎去刺杀秦始皇。结果因为力士无法分出秦始皇坐在哪辆车中，把铁椎扔到副车上，失败了。后来力士被缉杀，张良逃到下邳，得了侠士之名。再后来，张良追随了刘邦，成了一代名相和谋臣。

张良活着，这很重要。

先求活而不是志死，是面临工程失败的第一选择。无论工程做得怎样，人还在，团队还在，就总还是有机会的。即使力士死了，再找一个，又死了，还可以再找；即使团队散了，再建，再散了，还可以再建。只要工程还在，就还可以有张良，也还可做张良。做张良的第一要素不是聪明，而是活得长，活得长就有机会；做工程的第一要素，也不是做得多好多漂亮多有成就，而是这个项目还在，这个产品还活着。

所以张良的目标一直都在。只要有这个目标在，今天可以找力士去刺秦以推翻暴政，明天可以随刘邦夺天下以立汉朝。总而言之，只要目标在，做什么事、用谁来做以及怎么做，都是方法问题。工程中也是如此，不要以为只有“程序员写代码”能做出产品，买来的也一样是产品，抄袭的也一样是产品，挖了人家的整个团队来，也一样拥有了产品。这些事例，在我们这个行业中全都是有的，是真实的成功案例。所以具体到一件工程决策，只要不违道德，就没有可

耻；只要不违法律，就不算错误。

工程的目标明确，做事的法子就有权变；目标不明确，做得再多，力量再大，团队再好，也一样是把大铁椎扔错了地方。

1.2 集体目标对某些个人来说是无意义的

出主意的是张良，做事的是力士；主意出错了，力士死了，张良活着。

力士为什么死？

不是笨死的。你周围没有一个人是笨人，大家都聪明。力士大概是肌肉发达许多，但这并不代表他头脑简单。也就是说，力士可以选择跳楼摔死，但绝不是因为他笨才跳楼。力士这样选择，总是因为他“认为这种选择正确”。正确与错误——亦即是非观，是一个人的基本性格组成。若一个人是非不分，那么，大概他也不会成为公司一员、坐在你旁边或进入你的团队。

你看起来“是非不分”的那个人，可能只是他判断“正确与否”的依据与你不同。

程序员所认为的正确，总是一个问题的最优解——这与程序员接受的职业训练有关。若程序员认为系统“有了”一个目标，就必将认为这是“唯一的一个目标”。若不是这样，他就会再努力寻找、一直寻找，因为他认为若不唯一，则必不是“最终”目标，而追求最终目标而不是阶段目标才是程序员的兴趣之所在。这一点倒是很像当年一挥手，便让三千童男童女入了海的秦始皇。对于秦始皇来说，唯一正确的解就是长生不死。

秦始皇很一根筋啊。这没有不好，这反过来其实也很像程序员：程序员总是在刻意的追求中“无意间”创建了一片新大陆。如果没有这样的刻意追求，程序员可能过不了今晚就决心改行做市场人员、管理人员或是产品经理了；如果没有这种无意的收获，那么我们就看不到 Ajax 或者 Linux 这样的东西了。

有些东西是从无意间开始的，只是努力的追寻让它变得有了意义。有些东西，则是还没有开始追寻就夭亡了的，例如力士的刺秦。

力士相信刺秦是正确的。这是张良唯一能让力士相信的事情。张良设立了一个目标，并成功地让力士相信了这一目标是唯一并最终正确的，这也是力士志以赴死而非跳楼的原因。而力士，在这个问题上并不聪明：需知“刺秦≠灭秦”，这两件事所指的其实是两个不同的“秦”的抽象啊。所以这至少证明他缺乏一个好程序员的核心素质：抽象概念都分不清，当然做不好程序员。

不过，也许“刺秦”的确是力士的目标也说不定——力士嘛，怎么会想着灭秦呢？没必要的事情嘛。

力士死了，没有名字；秦灭了，不是因为刺秦。

1.3 张良是失败的产品经理

灭秦是一项工程。若只以结果论，这项工程的总监是张良，老板是刘邦。

张良刺秦王，是用错了法子；张良找力士，是用错了人；张良用力士刺秦王，是设计错了产品。总之，张良最初是错得一塌糊涂的。他比其他的糊涂人要好——至少运气要好一点：他活了下来。

所以成功还是多少要有点运气的。

不过接下来就不仅是运气了。当“灭秦”成为了产品目标时，天地就宽阔了。比如说，秦始皇死不死其实是不要紧的，只要秦国灭了就好；又比如说，秦国灭不灭其实也不要紧，只要不是秦国统治就好。前者是西汉之所以建了朝代，后者是南宋之所以亡国的原因——所谓一个工程的成败，于当时而言，就是定义的不同；于长期而言，就是一个收效的问题。

当然，张良还得跟对了人。据说张良见刘邦之前，是打算追随楚王景驹的，但正巧路上遇见了刘邦。但仅仅“巧遇”还是不够的——就好像说你在公司里被“碰巧”分到了某个项目团队，这样的偶然事件其实是不能决定你能否成功的。

张良之所以追随了刘邦，成了“刘邦的张良”，是因为刘邦对张良“言听、计从”。首先，对于一个团队——哪怕是仅仅两个人的团队——来说，“听别人说话”是第一等级的待遇。我发现许多项目经理失去了威信，倒不是因为他多笨多傻，或者太聪明掩盖了别人的光芒，反倒是他太过刚愎自用，不听别人的意见。于是团队里的声音就渐渐少了，腹诽渐渐多了，隔阂也就渐渐出来了。所以，“听别人说话”倒是学做刘邦的第一要务。

其次，张良是谋臣，谋臣的“谋”是他的生存之本，所以“计从”是他愿意呆在刘邦身边的根本原因。若刘邦只听他说话，而不采信他的计谋，那么张良也呆不久，因为这样的张良便只是一个言官而不是一个谋臣。但是，“计从”这件事情有很大的风险，谁能做到“算无遗策”呢？

所以做好刘邦，也不是件容易的事。

1.4 力士之死

刘邦的手边大概有两类人，一类是张良、韩信这样的谋臣与将帅，第二类则是许多的力士。谋臣与将帅大多都留有名字，当然刘邦也留下了。力士却没有——这一点算是遗憾，不过整体上不要紧，毕竟力士很多，做个大一些的纪念碑，能写多少名字就写下多少，这就好了。

力士们要用得好，最重要的是给他们一个目标，例如刺秦；而且要给他们一个靠得住的理由，比如刺了秦大家就有饭吃。当然，仅仅给个理由是不足够的，比如“有饭吃”这个理由就并不充分，因为别家“刺秦公司”或“刺 X 公司”也可能给饭吃。所以当有很多家公司用这个理由时，“目标”就不紧要了：目标因为理由而存在，如果理由不充分，则目标就不是那么必要，这是一个力士们都可以推论得出来的逻辑。

所以有两种公司层面的选择，一种是给出更多的理由，比如更高的薪酬待遇；另一种呢，则是改变他们的目标。后一种选择涉及到两个关键问题：其一，人是利己的还是利他的；其二，人是先利己的还是先利他的。本质上来说，这是对人性的质问——无论是程序员，还是力士^①。马斯洛提出的“需求五层次”理论对于这一问题的回应很简单：人是既利己也利他的，但总的来说是先利己后利他的，终究来说还是利己的^②。

简单地说，要吃饭、要活命是最基本的**生理需求与安全需求**，这是

^① 所以这是一个普遍性而非道德性的问题，切勿站在道德的至高点上来给出答案。我们可以要求一个力士做圣人，但如果所有力士都超凡入圣了，也就没有“圣人”了。

^② 马斯洛是从人的“自我需求”出发来讨论的，因此所谓“利他”的结果必是“利己”的一种附加收益。

薪酬待遇可以解决的。接下来是在此二者之上的**社会需求**，这是一个社会角色——首先是职业角色的需求，所以在公司层面的选择就是职务的提升，以及给出职业发展空间，等等。其四，职业上的**尊重感**一定程度上来自于专业性和必要性，亦即是说，公司强调职业价值与专业价值是对大家的一种基本肯定，这与“言听计从”是等效的。最后，大家——整个公司、团队中的所有人——都追求的、自我的、发自利己性的需求，便是剩下的**成就感**了。

无论力士之路，抑或张良之道，其实都是如此。力士将成就感定位在刺秦的成败，张良将成就感定位在灭秦的成败，这仅是目标的不同，其背后的需要——成就感——其实是一样的。当然，张良的道路走得慎重得多：藏身下邳，是求生存；得刘邦重用，是求尊重；灭秦兴汉，是谓成就。

力士连生存问题也考虑得不甚了了，所以刺秦一失，便身亡了。

1.5 “成功”与“忠诚”

刘邦毕竟是用“灭秦”统一了力士们的目标，无论如何，他做到了这一点。

但是谋臣与将帅，对于刘邦来说，就不是设定“一个目标”就可以的了。这其实也是许多公司留不住中层的原因：目标若只是一事一物，那么该事物被替代之日，便是去留取舍之时。大公司死的产品多，那中层也就换得勤：要么换产品线，要么就换公司。总之，如果将“一个产品”作为目标，那么中层就总在失望中煎熬：熬得住的像油条一样在不同的产品线上翻滚，熬不住的换一家公司继续翻滚。大体如此。

反过来说，刘邦若是大公司的老板，就决不会将“一个产品的生死”作为中层的目标。更多的情况下，这样的产品只是老板权衡属下的砝码。这便是所谓“重结果也重过程”的管理思路。从公司层面上来说，中层所必须确保的是对“企业目标”的认可，其次才是“做好当下”的态度与方法。有了这一前提，不因一时一事之得失而奖惩——这样的管理行为才是可以理解的。

这是“有趣的”公司管理制度：成功了十次的人，可能比不上对公司“一贯忠诚”的人。正确吗？答案是：正确。

因为在中层管理以上，最大的成本其实不是一两个产品的得失，而是信任成本。比如说：刘邦为什么信任张良的一个“计谋”？如果刘邦需要花许许多多的方法去考察，又花许许多多的战役去评估，再花许许多多的时间去推演张良的“计谋”，那么张良对刘邦也就失去了价值。张良之于刘邦，大多数时候的核心价值在于“军师是可以信赖的”。只有有了这一信任基础，高层的决策才能迅速而卓有成效，其结果才能显著，其实施方才可行。

一个扯皮会议的背后，不是所谓的缺乏决策，本质上是缺乏信任。

你能说清楚你在会议上的一个建议的所有“企图”吗？

解释很累、很花成本。

1.6 信任是组织的基本需求

所以，做好张良与韩信的诀窍在于：取得信任。

很多中层天天要授权，但却没有反过来想想：我真的能放心让老板授权吗？嗯，大多数中层要是反思一下这个问题，大概就得脸红，

就得吃不下饭，就得挂靴退场。反过来说呢，大多数没有挂靴的人，主要还是吃得下饭、脸不会红，以及或者不会反过来想。

建立信任的方式有很多种，例如亲近感。我第一次为“亲近”二字感到恐惧的，大概是来自一次淘宝交易——当我打开“旺旺”与卖家初次交谈时，对方的一句“亲，有什么可以帮到您的？”立时就让我惊愣当场。多么廉价、直白而又脆弱的亲近感啊！你千万不能指望如此亲昵而又随意的称呼会给你带来何种信任。如同今晚请你吃饭的老板或同事拍着你的肩膀说“兄弟，我们就这么干了”——这一类的亲近大抵都是靠不住的。

用利益构成的信任也靠不住，例如薪酬。但有种情况是一个特例，即是说“如果某种利益是特有的”，那么这时信任可能要强得多。诸葛亮之所以跟随了刘备而不是曹操，更可能的原因是刘备能给的“利益”更为特有一些：曹操帐下人才济济，相比之下刘备更需倚重诸葛先生。比较来看，薪酬这种利益就没什么“特殊性”。因此你开出再高的薪水，员工该辞职的还是要辞职，对手能挖角的还是照样挖。靠利益构筑起来的团队关系，要多薄弱就有多薄弱——但是老实说，这也比“亲近感”带来的信任要好一些。

薪酬解决的是人的第一、二类需求，即**生理需求与安全需求**。在一个成熟公司的人力结构中，这是应当被基本满足的——也就是说，它应当不成为主要问题^①。但是如果它被有意地“放大”成为主要问题，比如说公司就非得宣称“我们具有业界一流的薪酬”，那么无异于把信任建筑在利益之上，这一利益崩溃之日，便是信任瓦解之时。

^① 在创业公司，这可能是第二位的，大家勒紧了裤腰带过日子是因为大家都接受了“创业”这样的原始目标，对这一背景的“认可”决定了人的价值取向。

许多公司在第一、二类需要能基本被满足时，将信任关系“转移”到其他方面，例如认同感。譬如认同公司的目标前景、认同公司的价值观，以及认同“这群人”——这些的确相对明智得多。但是，这事实上正好是违背人性的——我的意思是人的利己性。人的自利性决定了“被认同才是需求”，而此时所谓“认同”却是付出。当一个信任关系建筑于个人的付出时，它也不是那么牢靠的。例如让员工“认同公司的目标前景”，那么授予期权股票吧，而当上市无望或股票贬值时，这种认同就不存在了；又例如“认同公司的价值观”，那么当个人利益与公司价值观冲突时，这种认同就被质疑了。

所以作为中层，你向老板表示亲近，或表示对薪酬的满意，或对他们认为正确的东西表示认同等，都不足以取得他们的信任。作为你的老板，他比你更清楚：将信任构筑在这些东西之上，是靠不住的。反过来说，如果他们因为这些而向你表示信任，那么你也要小心了：无论老板真是猪头，还是他们别有所图，你都得快点换个东家。

1.7 吴起爱兵是宣传策略

战国。魏国。一天，一个战士身上长了脓疮，吴起将军前来探病，亲自用嘴为他吸出了脓血。这个战士的母亲听说后，伤心地哭了起来。邻人问她说：“吴将军关心你的儿子，你应该高兴才是，为什么要哭呢？”战士的母亲说：“过去，吴将军曾为他父亲吸脓血。他父亲感动不已，打仗时拼命冲杀，死在战场上。如今吴将军又为他吸脓血，我真不知道他又会死在哪里！”

战士死没死，我不知道。

吴起的法子偶尔用用是可以的，但想想伤病士兵的数量，就知道这个法子没有普适性。所以如果士兵也这样来想问题，就知道“等着吴

将军来为我吸脓血”大概无异于等天上掉馅饼。这个法子之所以“偶尔地有用，却又被常常地提及”，是因为这是用人术的典型示例。但这并不具有团队价值，简单地说：无法推及到整个团队的方法，都只是技术技巧而已。

吴起要爱兵，好的法子为爱部下，并让部下爱兵；吴起要用兵，好的法子也是用部下，并让部下用兵。“吴起→部下→兵卒”是一个组织结构，找到让这个组织结构活起来的法子，才是将帅之本。若非如此，脓血是可以一个个地吸过去，但临到战场上，却还是只会让当兵的送命。

兵之于将，在于求活；将之于兵，在于求胜。常胜，大家才有活路，队伍才不会散了。所以会不会做产品，会不会让项目活下去，会不会让工程或产品线成为公司关注的焦点，才是兄弟们跟着你的缘由，才是他们求活的普遍法子。

吴起爱兵，并不是他成为将军的原因。

这一点许多人误读了。

1.8 专业 = 做该做的 + 该做的都做得成

信任要么来自口碑，要么就来自于实事；所谓口碑，追究起来，也是实事的效应。所以总的来说，所谓信任，是做人的境界，亦是做事的效应。于一个职业角色而言，做到“信任”并不复杂：其一，取得老板的信任，关键在于“做该做的”；其二，取得下属的信任，关键在于“该做的都做得成”。

我常常说要“专业”，便是这个意思。

我必须先摒弃一种所谓“人浮于事”的组织。在这种组织中，“求做事”并不是第一位的，其第一位的是在于“求利”，第二位的则在于“求活”。所谓求利，就是大多数组织角色都在谋求**自己的利益**，而非**组织整体目标的利益**；所谓求活，就是大多数组织角色都在为自己能呆在这个组织里的、组织角色上的时间长短而担忧——其一，呆得更长则获利更多，所以求活也是求利的一部分；其二，组织的，以及组织整体目标的价值与“我”是无关的，因为我的目标便是从组织中牟利，或与组织谋利。

总的来看，在这样的组织中，“做事，或不做事”都是次之又次的事情。“专业”在其中是没有价值的，因为没有事可做；即使有事要做，也只是为了延续组织角色的时间；即使要做好，也只是为了“做好这件事本身”之外的利益。所以，人浮于事，因而不求事业之专。若再换言之，便是“人浮于利”。

所谓良禽择木而栖，便是选一个组织以容身的事情。要么你可以改变组织，要么你可以容身于组织，要么你可以远离组织，这些都是你的选择^①。你的“专业性”本身，同组织的形态与既定构成是无关的。但你选择了怎样的组织，决定了你的“专业”是否适用于它，这也进一步的决定了你的、团队的、项目的前途与未来。

^① 这也是本编的关键命题：不讨论用人术，不讨论成功学，也不讨论办公室政治。我们讨论的基点是“一个做事的组织”，这才是团队的根本，也才是一个项目所能依赖的基础。

第 2 章 看到别人能做什么

2.1 田忌赛马不仅仅是个娱乐项目

齐威王用了三匹马，以及一个田忌将军。田忌用了三匹马，以及一个孙臧。孙臧用了三匹马，以及一个计谋。齐威王想得的是快乐，赛马的胜败是无所谓的，开开心心地与大家同乐就好；田忌将军想要的是齐威王开心，如果只能败不能胜，齐威王的游戏就乏味了；孙臧想得的是胜，因为只有得胜，才会得重用。

齐威王事实上连自己有多少马都不见得清楚，何况它们的脚力；他只是想着该与田忌将军赛一回马了。田忌将军不是因为不聪明而不关注马力，而是因为他更多地关注齐威王的心情。孙臧关注于马力，是因为马的成功就是他的成功；他只能胜，只有胜可以选。

齐威王要用好田忌将军，他做的事情是与将帅同乐。田忌将军要用好孙臧，他做的事情是言听计从。孙臧要用好马，他做的事情是观察马的脚力。

大家都做对事，就成功了。

2.2 马性

作为马，了解自己的脚力是很重要的。

大多数程序员“一贫如洗”，所能倚仗的就是他们手中的技术，如同马倚仗它们的脚力一样。但这种“贫”只是指种类的多少，而不是数量上规模。举例来说，一个很资深的程序员与一个刚入门的程序员，

可能在这个问题上是一样“贫”的——唯技术可恃。

与世界上所有职业角色一样，程序员可能会认为自己是“优势物种”。一方面，这一认识可以细化到具体的分工，例如前端认为后端简单，知识结构单一；后端认为前端只会玩玩脚本，数据结构与系统性能等都全然不懂。另一方面，这一认识也会被推及到整个组织中的所有角色，例如程序员可能贬低产品经理，认为他们根本就不知道产品有多么复杂多么难以实现；也可能贬低项目经理，认为他们除了会看墙上的钟，就只会看手上的表，再或是就只剩下看桌面日历上弹出的提醒；也可能贬低各个部门的总监或职员，理由更加简单——他们什么也不懂。

支撑程序员做这件事的，正是他们唯一剩下而又唯一倚仗的东西——技术：一方面，技术是他们信心的来源；另一方面，技术也是他们实现一切的手段。所以，当公司提出“我们能不能减少程序员”的时候，程序员的想法是：开发一个用于减少程序员的程序——当然，前提是增加几个程序员。类似地，程序员对于新产品的想法是“我们可以做一个”，对于大系统的想法是“更多的人手”，对于稳定的想法是“重构系统”，对于性能的想法是“优化代码”……这些都可以归结为一个简单的描述：更强的脚力。

程序员超乎寻常地追求逻辑性。例如，他会在会议上纠正你的一个措辞，即使这个措辞并不影响大家的理解——但是就程序的逻辑的必要性而言，不正确的概念会产生歧义。程序员愿意在会议中牺牲掉十分钟来解释一个名词概念，因为他并不在乎“达成会议的原始目标的必要性”，他更多地在乎“逻辑的必要性”。

程序员热衷于确定一件事物，而反对随意性以及“看起来不必要的”尝试。无论如何，只要一事情有着可度量、可预知、可确定的结

果，那么程序员就对此信心满满——即使这些仅仅是基于他“专业性的预估”。更深一层地分析，之所以程序员敢于拍着胸脯说“这个功能我们三天就可以做完”，其本质原因在于“他希望这一开发过程是确定的”，并主观地加以了确定。确定性是程序员进行“分析活动”的基本需求，他不得不先满足这一点，进而才能论及其他。有趣的是，如果你告诉程序员“这件事一辈子也完成不了”，那么通常会有一个结果：他什么也不做，直到他证明了上述观点的对错。

逻辑与正确是程序的基本思维方法，这些思维方法影响了（大多数的）程序员的性格形成。例如他们不愿意做多余的、重复性的、无意义的工作，也对冒险性的尝试充满了敌意。最特别的一种情况是，他们会说：这不是错误，而是正确的设计。但是如果从基于人性——而非逻辑——的角度上来讨论一下正确性问题，那么：

不能用的，就一定是错的。

2.3 在不讨论整体时，某些具体行为可能是逻辑正确却没有存在的必要性

逻辑是因为其必要性的存在而存在的。

田忌赛马中的胜负并不一定是由马的脚力决定的。即使田忌的马都比齐威王的好，他也不能场场都胜；就算是胜了，最好的法子也是把马献给齐王。所以，“赛马以取胜”这一逻辑，是因“若与齐王赛马取乐，则不能常输”而存在的。一旦田忌将军是常胜，那上面的逻辑必要性也就不存在了，那时“赛马求败”反倒是可能的逻辑。

架构角色往往追求系统整体的必要性，他因此必须预见到“何谓系统整体”，并判断出系统整体的、核心的目的。以田忌赛马而言，系统

整体是田忌与齐王双方，目的是使君臣齐乐，最后才决断出“取胜”的必要性。以具体软件系统的架构而言，架构师必须判断的是哪些地方该考虑性能高低，哪些地方该做功能增补；哪些地方宜先实施而后优化，哪些地方又必须先找出最优解再做实施；哪些地方可以通过产品选型来购置，哪些地方必须自主研发.....这些都是架构在系统的组织结构上要考虑的事情。但如果相同的事情交由程序员来判断，则答案可能就缺乏了轻重缓急的权衡；交由项目经理来判断，则答案又可能偏离系统的事实而难于实施。

架构是最没有安全感的角色。在架构师的眼中一切都可能存在，也都可能湮灭于一霎：千日之功，亏于一簣。之所以架构师对任何一个选择都没有绝对的信心，是因为架构这一角色对系统提供的总是“可能性”，而非必然性。从这一点出发的架构师有三种。第一种是**殚精竭虑型**的，这种人总会认为系统的一切与自己都有关，也将一切的安危系于自己一身；第二种是**置身事外型**的，这种人总认为系统的成败其实只是系统自我发展的结果，而架构师只是系统的观察者、考量者与守护者。以上两种架构角色其实都过于极端，对系统的发展都是趋害的。第三种架构师则会在这二者之间**觅求平衡**，他应知轻重缓急，熟稔何时把握什么，争取什么，以及推动什么；他还应知刚柔曲直，攻守并济，把生死看成系统的第一要务，把时间看成调节生死的阀门。

好的架构师会在系统出现急难之前就对你发出警告，更好的则已经提出过解决方案，再好一些情况下，架构角色已经要求你与团队一起预演过这样的方案了。但是一旦急难发生，你需要的是技术高手，是实施团队，是那些在代码中摸爬滚打的程序员。因为这时候再谈架构，无异于房子着火了却仍在埋怨“我们没有安消防栓”。若“着火了，且没安消防栓”已是既成事实，那就应该加大水的输送或鼓动消

防队员奋勇争先，尽一切可能去灭火，这才是上策。我们可以追究一个架构师在“如何部署消防栓”和“没有提供着火时的应急措施”这两件事情上的失职，但这种追究或是在事前，或是在事后，切不可事中。

架构的危险性也正在于此：架构师的失职是无法补偿的。架构师只有尽可能全面地评估系统中“所有可能的”问题与方案，并尽可能使团队与系统处于安全边界以内。退而求其次，危机一旦发生，架构也要保证提供可能的应对之策。

架构师要做的事情，任何一个角色都可以去做，但架构师所需的素质，却在“做”那些具体事务的能力之外。这并不表明架构师眼高于顶、手低于足。须知架构这一角色必备的是眼界与敏锐：放开眼界，才能看得到危机；切中利弊，才能控制系统代价。我们可以要求架构师是行动派，但架构师必须控制自己不要变成冲动派。

行成于思，毁于随。

2.4 架构师的职责：将“切实有效的架构”解构为具体可行的“系统化方法”

事实上，程序员讨厌架构师的未雨绸缪。而且同时，管理者也不喜欢这样。

有趣的事情是，在大多数情况下，在危机出现之前是没有人相信危险的，并且也是没有人愿意为此投入资源的。因此大多数架构师提出的“方案”都可能面临实施者的质疑：那个架构师的想法是多余的，是没用的空中楼阁。没人愿意为这些“可能的异常”付出代码、时间与防护性的子系统。

我至今所知道的唯一一件能让程序员——以及包括项目经理在内的整个团队——所接受的预防性实施技术，大概就是“结构化异常”(try...catch...finally)。但仔细地探究也会发现，其实越是在项目中面临过工期拖延、系统崩溃以及未知 BUG 折磨的开发人员与团队，就越能接受在代码中出现更多的、貌似无用的异常捕获。所以看起来，真正的问题并不是程序员以及管理者不愿意推进“预防性的”架构，而是在于：一是他们无法感受到反复的系统危机所带来的痛苦折磨，二是他们无法找到有效的、可实施的方法来应对上述危机。

层次系统是极有可能符合这一要求的架构方法。但是就程序员与管理层所能接受的理念来讲，层次系统是过于复杂而随意的——只要会说“增加一层抽象”的，就都成了架构师。但正确的层次抽象是确实可以隔离复杂性的^①，而且通过接口设计可以进一步减少、屏蔽或确指层间的依赖。然而这些工作必然带来程序员的不满，因为接口的抽象与使用是多余的工作；也必然带来管理层的不满，因为“层次”这一系统抽象在事实上也约定了组织结构的规划。

测试驱动开发过程是另一个视角下的解决方法。从工程过程的角度上来讲，它通过明确系统的“问题”、“关键点”与“指标”等来将系统参数化，并将这样的参数作为过程中贯彻始终的标准加以推进。尤其重要的是，它事实上也内在的要求了系统构件间的“接口化”与“复杂性隔离”，因此它可以与层次系统有着相当良性的配合。

迄今为止，我没有看到过一个更好的、更简单的“架构”，能切实地应用于项目之中，将可能发生的危机阻隔在一定的范围之内。我在这里所说的“架构”是指两个方面，其一是架构师对系统的抽象方法，

^① 在《我的架构思想》一书中会更加详细地说明这一事实，并探究层次系统隔离“可变性”的本质。

例如层次架构方法；其二是指由“软件架构”与“软件开发方法”配合所产生的整体效果，例如层次系统与驱动开发过程的配合。因而，我所指的“切实有效的架构”可以简单地描述为：

目标系统、架构方法与开发过程协调统一的整体。

事实上，我认为只有这样（包括组织的、过程的、实施的等在内）的架构才是对“预防性”的整体求解^①。

然而，这带来了很高的组织成本。管理者喜欢简化问题，喜欢尽快投入行动并看到结果——在这些方面，管理者能很快与实施者达成一致，但与架构师则很难。追究这一问题，事实上症结在于架构与管理二者的相互脱责。首先，架构师所要做的事情，就是推进这些“看起来多余”的事情的实施。架构师若一力担之，便出现了架构师领导下的应急小组；架构师若置身事外，整个团队失去了在这件事情上的基本动力。所以，架构师必须在“方法”问题上迎合管理者，必须使架构的具体实施方法简化、可行、有效。

因此，真正有效的架构方法，就是让整个团队都把这些“看起来多余”的事情视为系统实施所必须的一部分，即“系统化方法”，亦即是团队可以应用的法子。若架构师的主意总是“我们需要一两个高手来做这件事”，或“我们需要一两个突击队”，那这并不会影响团队整体，对管理者来说也必是“可选的”。大多数情况下，管理者只会在不影响团队整体资源的配给时，才会同意这些想法。

而反过来，如果一个方法是“系统化方法”，它影响到团队整体的投

^① 例如，正是因为开发过程是测试驱动的，所以可以将测试视为“catch (...)”，而将“catch() {...} finally {...}”视为包含于一个迭代中的补偿。又例如，正是因为系统是层次化的，因此“try ...”才是有确定范围的，且“catch (...)”的捕获才是有效的。

入与产出，并且将深刻地影响到系统的推进，这时管理者才会真正地重视起来。这也才是考验管理者的团队决策能力的地方，即我们——整体地——要使用什么样的方法来工作。因为一旦管理者认为“这不可行”时，这个系统化方法便失去了价值。所谓不可行，本质上来讲仍然是“对于当前的系统（团队、项目与产品目标）来说，是难于实施的”。

架构师必须与管理者在“系统化方法”上达成一致，从而在团队、方向、目标以及整体策略上，对系统施加约束。

2.5 团队是“管”不好的

所谓“管”，就是“官”头上的帽子。换言之，一个管理者若想体现自己的“官性”，那么必然提出种种的管理方法。然而，管理人员的真正价值并不在于“设定做事的方法”，而在于帮助团队找到他们做事的方法。

我仍然记得我在团队中推进“代码格式化”时采用的方法。那时我所在团队有着从不同语言、不同团队转来的程序员，其中不乏有着种种开发怪癖的能人异士（相信我，你也会碰到这样的成员的），于是整个团队所提交的代码风格混乱不堪。尽管大家都提出“统一代码风格”的要求，尽管几乎每个人都认为应该这样做，但是具体决定要采用何种风格时，整个会议室吵翻了天：每个人都有要求采用自己熟悉的风格的充足理由，甚至一对大括号应该占据一行、两行或是四行这样的问题，都成了涉及哲学、美学以及基于某种学派理论的争端。为了这个问题，整个团队整整两天都无法提交任何有效的代码。然后，终于地，在一个会议上我提出了最终的解决方案，我告诉大家：

- (1) 我决定，我可以接受任何人的、任何风格的代码。但是，
- (2) 我决不接受同一份代码中有第二种代码风格。所以，
- (3) 如果你修改别人的代码文件，那么你必须让该文件中的所有代码使用你或别人的风格；你可以选择任何一种，但结果必须保证该文件中的代码风格唯一。

很快，整个团队的代码风格统一了。对于有美学追求的人，我告诉他们，同一个代码文件中有两种风格，肯定是不美的；对于有哲学追求的人，我告诉他们，你若要修改别人的代码，就有责任使代码变得更好，而不是更乱。于是，所有人都在“改代码风格”上付出着努力，适应别人或是改变自己。一段时间之后，他们便找到了最简洁和有效的法子：

能保证最快修改代码的风格，就是好风格。

2.6 组织行为学中的 leader 与 owner

有两种与“做一个官”不同的角色定义，可以更好地实现管理价值。

领袖（Leader）是一个极需个人魅力的角色。与大多数人设想的并不一样，领袖其实是最需要那种事无巨细亲历亲为的人来担任。领袖角色最富于行动性，当组织确定一个目标时，最先听到的声音应该来自领袖：

“我来做！”

这种声音具有强烈的号召力，自信果决，有着一往无前的勇气。同时，领袖角色并不是一个孤高自负的人，他深知团队力量远远强于个人力量。但是他并不乐意与平庸合作，因此得到他首肯的唯一方

法，只能是“超乎寻常地优秀”。领袖角色最怕拖泥带水的人、事与环境，他从心底里拒绝自己陷入泥沼，从而避免自己成为悲剧英雄。领袖角色有“不成功，则成仁”的勇气与心态，但他会把这个“成仁”的时间放大到自己可控的范围内，一旦时间失控，他最可能的选择是黯然离场——唯一能击倒他的方式只有“时间”二字。

争取领袖角色的合作是有条件的。其一，他需要正确的方向或是明确的目标，对于前者，他有探索的勇气；对于后者，他有实做的能力。其二，他需要最聪明的人，他可以驾驭聪明。其三，他需要精练的决策团队。他的个性特点，决定了这三个条件是在团队管理中所必须的补充。首先，领袖角色是行动派的，他是推动系统向前发展的核心力量。但是如果系统的方向或目标并不明确，那么他的行动就失去了理由，他的判断就失去了依据，他的勇气、信心、能力以及种种努力就都失去了存在与施展的舞台。一旦领袖角色失去了方向，那么整个团队将迷茫停滞，时时处于溃散的边缘。其次，领袖角色对于驾驭聪明有着浓厚的兴趣，这是领袖角色管理艺术的核心：领袖角色是彻头彻尾的实干家，因此他事实上是在寻找“有用的聪明”——对“聪明”真正正确的判断应落脚在对于系统的影响上。最后，办公室政治、人际纷争之类是领袖角色会尽力避免的，在这些边角上的聪明只会令他反感。凡是与系统目标无关的决策者、决策过程，都是领袖角色大刀阔斧砍削的对象。

上述条件也反映了另一个事实：领袖角色将组织“适合自己的团队”。因为上述条件中有目标、有人、有做事的方法，这些构成了一个独特团队的基因，也是这个团队持续的、长久的价值核心所在。这也表明，领袖角色在“组织授权”方面是有条件的——他需要更高层次组织（例如公司、事业部等）的明确授权。

第二种角色称为 **Owner**（主管）^①。对于一件事情，一个 **Owner** 的正确反馈是^②：

“我负责！”

Owner 最需要的是清醒的头脑、务实的作风，以及敢于牺牲的精神。首先，他需要足够清醒以确保在合适的问题上找到合适的人、合适的方法，以及判断什么是“合适的解”。例如在技术专长方面，他可以借助团队的力量；在方向决策方面，他可以借助架构的力量；在团队激励方面，他可以借助领袖的力量，等等。**Owner** 的协调能力是他对整个团队的最大价值，因此他将尽力避免与任何一个角色的冲突——事实上他是尽力与所有人合作，并通过这一协调能力来消化团队内部的矛盾。其次，**Owner** 角色对“可能的意见”的判断趋于务实，因此可能是“不那么聪明的”，也可能是“不那么完美的”，但对于此时此事、于当时的选择必是最妥贴的。最后，**Owner** 视系统为自己的一部分，因此尽管他可以接受架构角色关于方向与策略的观点，但他总是更优先地考虑“系统当前的生死”——即使有人承诺给出“更长的时间、更大的空间”，也不能使 **Owner** 忽略系统当下的现实问题。他总是清醒地面对这样的“切肤之痛”：若系统出了问题，是 **Owner** 的问题；若系统死了，**Owner** 便失去了他存在的唯一价值。

Owner 眼中只有系统，“系统是我的”，激发着 **Owner** 的自利性；“我只有系统”，使得 **Owner** 舍无可舍。无论从概念上、心态上，还是从组织角色上 **Owner** 都与系统绑在一起，他担负着系统责任，其

^① 这里采用了一个较特别的翻译。**Owner** 在词义上有“所有者”含义，一般译为属主、物主来表明这种关系。但在职场中，**Owner** 通常用于指代其责权，而不是确指职务，例如“这件事的 **Owner** 是谁”。

^② 但对于许多人——尤其对于“官”字当头的管理者来说，第一反应往往是：“（这件事）我管”。**我管与我负责**体现了两种不同的角色定位，也决定了二者用以实现管理价值的方法的不同。

存在就证明系统是有价值的，而这也同时就是他的管理价值。

所谓系统，并不一定是软件，也可以是组织。

2.7 不同的利益追求是组织角色的基本差异

我注意到一个事实：即使是最无能的管理者，在争取部门利益时也是最积极的。

一个人——我是指职业角色——的底限，在于其求活的根本。程序员求活，在于技术，所以苦练技术是每一个程序员都愿意投入的事情。新项目、新产品、新技术，凡此种种，都是让程序员听一下就“连后脑勺都乐”的事情。架构师求活，在于眼光，所以放开眼目是每一个架构师都愿意立即着手的事情。分析领域、跟进系统、参与决策，凡此种种，架构师尽管貌似烦不胜烦，但实则不遗余力。Leader 与 Owner 的求活在于他们所依赖的环境，所以砍掉系统，Leader 就扯旗走人；砍掉部门，Owner 就跳脚骂娘。反之，Leader 总是试图使项目更受重视或优先级更高，Owner 则总在强调资源与规模。

每种管理培训都在谈“成功学”、讲“价值观”，仿佛人人都在追求理想、道德，有着崇高的目标。但现实中往往并非如此，我们的组织并没有这么复杂，组织中的人也并没有这么复杂。大家都只是在守着自己的底限，并“尽量地做些事情”而已。

“尽量地做些事情”，如果正好是组织给他们——这些具体的角色——的目标，那么大家就可以庆幸了，因为这意味着系统必将向良性发展。否则，大家就必然面临个体与团队的利益冲突。而在面临这样的问题时，员工能守着“职业道德的底线”就已经不错了，怎能奢

求系统的发展呢？因此只有当他们视这一目标为自己的目标，其利益是自己的利益时，才会一往无前。在这个问题上，人，首先是自利的——每个人都在追求自己在团队中的位置、价值与认可。

但是，正确的做法是先看到别人的位置与价值。因为认可是别人给的——欲先取之，必先予之，若你都不认可别人，那别人自然也就变尽了法子来排斥你。所以若以团队为背景，则重要的不是先看到自己，而是先看到别人；重要的不是先看到别人什么都不能，而是要看到别人能做什么。

用与不用，在你。

第3章 要做事，不要管理

3.1 从团队的视角来看问题

“我凭什么要帮他？”

A 公司很成功地处理了这个问题。一个新员工在 A 公司入职之后，总是会被分派一个师傅。师傅与这名新员工之间，被公司从组织层面上施加了一个名义上的师徒关系。并且，在此之后，这一师徒关系还是一荣俱荣的：在季度考核中，有最佳学生、最佳师徒，以及最佳人师之类的奖项。

建立并维系一种关系，是一个很好的“帮他”的理由。类似地，还可以有“家族”、“帮派”、“兴趣小组”以及“（同学）会、（同领导）门、（同乡）宗”等繁复的社会关系系统。事实上，我们的组织管理思想，一面在骂着被称为“国粹”的人情关系，一边又在利用着人情关系来维持组织的有效性。作为组织者，我们自己都没有从心底里放下那些自己鄙弃的东西，却又信誓旦旦地要将这些东西踩在脚底。

我们真的找到了那个问题的答案吗？

当我们问对了问题，事实上也就找到了解：

“既然在一个团队中，你为什么不能帮他？”

3.2 要知道国共也曾互称“友军”

总说“物以类聚、人以群分”，人们若只扎堆在一群里，是没有“分类”

这样的属性的。所以若真把项目团队说成是“一群人”或“几群人组成的一大群人”，其实是无法考察他们的共性与差别的。真要达到这一目标，还是得把团队中的人分作几类。

不同分类方法的着眼点是不同的。一种是阶层化的，例如“下者用器，中者用人，上者用势”，是根据不同人在组织中的行为方式来区分的。事实上这也决定了他们的技术、能力与素质的不同。所谓下者用器，一定是“（工）欲善其事，必先利其器”的。——所以我常说“如果你认为这句话对，说明你是‘工’”，因为这原本就是你的工作方法、状态以及你构建知识、能力的基本模式。所谓中者用人，一定是举贤任能、政治教化的，辨的是人情关系，用的是人力才智，察的是人事浮沉，能力多强调在情商方面。所谓上者用势，那么就必然要能深悉事理事态，对形势方向尤其敏锐，在战略宏观上有着超常的判断力与决策力才行。这样的团队构建自然地形成了等级，形成了不同层级之间不同的行事风格与判断标准，因此带来的冲突是不可避免的。在这其中，我们通常谈及的凝聚力向心力之类，多是由人情世故或规章制度来形成与强化的——前者，是中层用人的法子，后者则是上层用势的法子。

另一种分类法基于对人的社会性的考察，根据人们在群体中的自我分工来进行，例如“领袖与追随者”。这一般取决于对自我的认知，以及对他人的认同。自我认知与社会认同是人的基本社会属性，如果一个人缺乏自我认知（缺乏自信或过于自信），或者缺乏社会认同（不信任他人或无法获得他人的信任），那么他都无法在群体中找到合适的位置。类似于“领袖与追随者”这样的定义建立了基本的“人与我”之间的社会角色关系，这种关系既构建于认知与认同之上，也随着它的变化而变化。例如，也许有一天追随者会提出“那些人、那些方向是不是值得我追随”这样的问题，或者领袖基于他对自身在

团队中的价值的认知产生了“没有我就没有这一切”的态度。可见，人的社会性（例如自我分工等）并非一成不变，它给整个团队组织带来了不稳定因素。

再举一个分类法的例子，比如朋友与敌人的区别。我们的团队很容易由“朋友”构成，尤其是小到三五个人的时候。这样的团队看起来应该能持久，但真正做下去，却发现矛盾重重，渐渐地朋友做不成了，队伍也就散了。究其原因，大家都只是冲着朋友的情面儿凑在了一起，却并没有人深刻思考过“如何构建与维护团队”的问题。事实上，所谓“朋友”有两层含义，同门为“朋”，同志为“友”。如果大家原本就是出自一门，比如同宗、同学、同事等，那么就可以算是“朋”。而这只是基本的、能快速抹去人际隔阂的一种亲近关系，这种关系能让大家快速聚起来并以最少的成本达成共识。创业之易便易在这里——我们总可以找到一群“看起来相同”的人。

但是接下来呢？如果大家在团队中的目的不一致，各自打着自己的小算盘，那么就不是有着共同志向的“友”。所谓朋党、友党的区别也就在这里。朋党有变成友党的基础，但不一定有“成为友党”的事实，因为后者取决于大家的“利益点、目的、方针与蓝图构画”是否相同。而一旦——理想情况下——整个团队中有了一个领袖人物，能在“利益、目的、方针、蓝图”等方面统一大家的思想，那么这个团队才能成其为团队。更有甚者，领袖人物也能敏锐地发现对手的利益取向，通过共同目的或方针蓝图之类的工具来化敌为友。所谓“没有永远的敌人，只有永远的利益”，说的便是这个意思。

团队之中，朋友与敌人是没有那么明显的界限的。这种矛盾的真正出处在于：大家在图什么？统一了图谋（亦即是“共同的志向”），才谈得上谁跟你是友，你跟谁做友。至于“朋”者，倒是简单了，刚

从小酒馆出来那几位，不还一边走，一边唱着臧天朔的《朋友》吗？

由所谓“朋友”构成的团队，最终的溃散通常是出自利益的侵蚀。

3.3 你的作为，是否对得起你的权力

团队是什么？这是一个似有解而又无解的问题，比如我们可以答“一个人以上的群体”，但这样的答案并不能反映团队的根本属性。

换个问题，例如“团队从哪里来”，就会很好回答了。因为通常“你”面对的团队只有两个来源，其一是自建，其二是接手。这里，我们先谈第二种，即“接手一个既有的团队”。注意，在这种情况下，你首先必然、必须——将自己的角色定义为：**Manager**，而这也确实是最棘手和最容易出错的。而且，老实地说，作为 **Manager**，你的确是一个“官”，因而你也必须有着自己的管理风格与手法，并行使你的管理责权。

你必须明确一个事实：**Manager** 的权力可以看做是天生的——因为在接手型的团队中，你是被“任命”的。无论你是“有能者居之”或是“有德者居之”，你以前的能力才干与德行声望，对于现今的团队来说都没什么意义。——你接手之前的东西都化作了**权力**，接手之后的东西都被称为**作为**。作为管理者，你（对上或于下）能取得的所有信任，将集中于一点：

■ 你的作为，是否对得起你的权力。

3.4 要么用对人，要么就不用

你必须先观察/了解这个团队的组成，尤其是其中是否存在两种关键角色，即 **Leader** 与 **Owner** 型的人物。我们说过，这两类角色不一

定是“官”，但他们都能很好地（甚至是更好地）实现管理价值。而 **Manager** 对于他们的了解与运用，很大程度上也决定了自己的成败。

Leader 决定了一个团队的风格与气质，他引领团队的方向并将一大批骨干吸引在自己的周围。对于一个既有的团队来说，**Leader** 对 **Manager** 的阻碍主要会来自于两个方面。第一，**Leader** 确实存在管理欲望。但本质上来说，**Leader** 仍然是实干型的人物，因此大多数情况下，他的管理欲望起源于对整体环境的不满。也就是说，他既知道“碰触到官僚的、管理性的事务”将是一个泥潭，但他又担心如果不这样做，自己的目标与理想是不能实现的。对于这种情况，**Manager** 应斟酌考量，如果 **Leader** 偏重于事务推进，则应该努力使他重掌团队事务；如果偏向于管理协调，则应该考虑是否给他管理角色和责权。但是无论如何，**Manager** 应该清楚，**Leader** 的去留对于团队整体来说是生死攸关的。

第二，**Leader** 努力的方向或试图达到的目标，与团队接下来要做的事情并不能一致。这是一个相当复杂的问题，因为团队更换管理层，多是因为不能达到而放弃旧的目标，或旧的管理层无能。但无论如何，总之是旧的事务已经无法正常推进了。这种情况下，**Leader** 在“方向问题”上其实已经受到一次重创，现在支撑他的主要理由——甚至是唯一理由可能就是“做，或是不做这件事情”。一旦团队原定的目标改变，而又不能使 **Leader** 接受新的目标，那么这个团队离实质性的溃散就不远了。对于这种情况，**Manager** 应该反思自己接受任命后的职责，究竟是从事新的项目还是持续旧的工作。一旦确实要“选择新的方向”，那么除了争取 **Leader** 的认同之外，可能就只好重建团队了。

Owner 最有可能对你的到来愤愤不平，但也最有可能衷心期待。但

这两种完全相反的心理却都出自同一个缘由：**Owner** 认为自己是系统的所有者。作为 **Manager**，你必须注意的是：对“系统”的评价，在 **Owner** 看来其实就是你对他的评价。例如，假设上任伊始，你就说“看看你们这个烂系统烂摊子，瞧瞧你们做的这堆无法收拾的烂事情吧”，那么基本上就打破了 **Owner** 对你的一切幻想。**Manager** 对于系统的态度，很大程度上决定了 **Owner** 对于“合作或排斥”的选择，如果 **Manager** 对系统表现出热心、关切并且努力推进，则 **Owner** 会认为“这起码是个来帮忙的”。

Owner 的另一种排斥心理，则来自于个人职业发展中的瓶颈，例如他会想：“为什么不让我来当这个系统的 **Manager**？”或者会干脆认为：“就是这个人挡住了我成为 **Manager** 的路！”但这种组织角色上的问题，其实是任何一个 **Manager** 的职业生涯中必须也是必然要面临的事情。一般来说，你必须表现得比别人更优秀、更强势、更有能力和资源（但是我一定程度上反对“更有手腕”这种技巧性的“处理”）。而且你必须明确一点事实，即：你的权力来自于“任命”，而非“实有”。因此在“自上而下”的管理组织中，你已得到认可；但在“自下而上”的行事过程中，你还远远未被认同。所以，如何应付“**Owner** 的排斥”，可能是你在团队中表现出管理风格的第一个契机。而且，这也决定了你是否能真正掌握这个团队。一般来说，你可能会有如下几种选择。

(1) 如果 **Owner** 确实有团队管理能力，那么可以将团队的权力还给 **Owner**，例如保留原有的结构、做事方法以及汇报与审核体系。

(2) 如果 **Owner** 只是团队中的黏合剂而缺乏必要的管理能力，那么你应该表现出强力而又强势的一面来构建团队，并在这个过程中带领 **Owner**——如果他接受你的领导。在这一选择中，如何保证

Owner 对团队的归属感和积极性是一个很重要的命题。一般来说，信任他，更好的沟通、商谈，以及将更多的事务与责任交付给他，可能会是一个不错的开端。

(3) 如果 **Owner** 在团队中只是一个老好人的角色而缺乏必要的素质与能力，那么必须先把他放在一边，重新任命一个角色来替代他。注意观察他的情绪变化，帮助他重新认识并找到自己在团队中的合适位置。如果（但这常常发生）他腹诽不已并给团队带来负面的、难于控制的影响，那么你可能必须放弃他。

对于接手型的团队，用人治事，是作为 **Manager** 的第一要务。

3.5 再解“跟随蚂蚁”

我们还得谈谈既有团队成员的普遍心理。事实上大多数不成功的团队是既没有 **Owner** 也没有 **Leader** 的，因为如果有（哪怕其中一个），那么这个团队也很难走到现在这种困境，以至于需要你来做他们的 **Manager**。

第一种心理因素来自于对你的了解与认同。你是 **Manager**，是官，那么大家首先看到的是你的职务高低——这并不一定是官性与奴性之类的问题，而通常是你的职务高低决定你能把握的资源，进而决定这个团队的未来。你在多大范围内有发言权，有影响力，以及是否有过成功的管理实践，这些都可能是团队成员在你到来前后会多方打听、收集并相互交流的“情报”。

接下来是对你的直观而又主观的认识，例如你所组织的第一个会议，或第一次团队活动，或者第一次约谈。包括时间计划，内容上的组织，你对现有团队、系统所了解的程度，你说话的语气、精神状态

以及条理逻辑等，这些都会成为团队成员对你的认识的关键组成部分——一些人甚至会把你的约谈次序都视为具有某种象征意义。必须强调的是，这些主观印象一旦形成，就很难在短时间内改变。因此，除非你真的想形成某种印象，否则你应该特别留意这些在“新官上任”时期的安排。

事实上你对团队的了解并不完全取决于你的观察，在很大程度上是来自团队“展现给你的那一部分”——这基于团队对你的认识，进而取决于“团队会告诉你什么”。所以作为 **Manager**，你必须知道并预设了你之于他们的基本印象，这既是管理方法的一部分，也是管理风格、素质修养的一部分。再接下来，才谈得上你对他们的了解。这包括对人和对事两个方面。除非你真的了解团队在做的事情，又或者你就是这方面的专家，再或者你此前负责的就是相同或类似事务的团队，否则你千万不要在团队面前“不懂装懂”——就我所知，这是在技术性项目团队中最让人绝望的管理行为。

如果你真的不了解这些“具体事务”，那么你可以找到关键角色（包括技术专家、**Leader**、**Owner** 或者某几件事情的“头儿”），征求他们的意见、建议与问题。你必须能足够耐心地充分利用自己的逻辑能力来梳理这些事情。即使你不懂这些事情的细节，你也必须将事情的骨架（这很类似于架构）整理出来，并尽量多地参与团队的沟通而又尽量少地干涉事务的细节。这一切，便如同《大道至简》中说的“跟随蚂蚁，但别掉进蚂蚁洞”——你的目的只是要让自己变得“懂一些”，而不是变成专家。你必须了解“团队在做的事情”，才能对这些事情有所评价，进而有发言权。又如果，你被派来的目标是“放弃现在的事情”，那么上述的过程既是你寻找“给大家一个放弃的理由”的过程，也是你“了解这群人的特点”的过程。所以，无论如何，你都需要先经历这个过程。

这之后，才是管理。

3.6 “理性利己主义者”的行事逻辑

团队中不一定是做事的人。

从统计学上来看^①，团队中积极做事的成员大约只有 8%，而 40%~66%的成员都是随大流的。基本上来说，积极或可能积极的人占到 2/3，而近 1/3（20%~30%）的人其实是“理性利己主义者”。

对于一般人来说，“做事”是必要的。这缘于两个方面的因素，其一是“大家都在做事”，其二是“我确实想做成些事情”。但对于理性利己主义者来说，“做事”是迫不得已的行为：一方面，不做事就会被一个做事的团队赶走；另一方面，做事可能是必需的谋生手段。但正好因为他们是“理性自利”的，所以这种“迫不得已”最终会变成两个对自我的、理性的拷问——进而争取利己价值的最大化：

(1) 如果只是为了“不被赶走”，那有没有比“做事”能投入更少成本的方法？

(2) 是不是只有“做这件事”才能谋生？

所以滥竽充数的南郭先生其实是典型的理性利己主义者——既然齐宣王喜欢听合奏，那么在他所在的团队中他就可以什么都不做而不被赶走。同样，如果“赶走别人”比拼命做事（以便让自己留下）代价更少，那么南郭先生也会想办法挤兑走那个发现自己没吹竽的人。但细究起来，这样的南郭先生并不是没有做事，而是通过更简单的

^① 参考《经济学家茶座》2002年第3期，“奥尔森学术思想介绍”一文，作者陈抗。主要观点与数据出自黎格曼的社会懈怠理论，以及奥尔森的理性自利的群体行为基础。

法子来达成上述留在团队中的基本目的。而且一旦“留在这个团队中”变得毫无必要，比如说有了更好的位置，或有了更好的谋生手段，那么他们就会毫不留恋地离开。

对于理性利己主义者来说，不但“是否需要做事”以及“是否可以投入更少的成本”是问题，而且即便“努力做事”确实是唯一之选，他们仍然会踟躇不已。因为还有两个与利己、与价值相关的问题：

(1) 为什么要“给你”做？

(2) 为什么要“在这里”做？

所以对于这一类成员来说，价值取向最终必然是偏向于私己的。但是从统计学上来说，三个人中就有一个是利己的。因此：

你面临的团队，并不会比这个状况好多少。

3.7 君子和而不同

那么，我们能通过团队教育来改变这一状况吗？

不能。

我并非是在否定团队教育的作用。事实上，我认为团队教育很有意义，但这只是体现在改变技术、协作以及形成一些基本认识方面。例如，我认为团队技术培训很有用，因为如同军队一样，相同或相配合的作战技能总是能得到最佳的效果。又例如，我认为企业文化培训很有价值，但价值并不在于“让大家变成同一种人”，而是让他们形成对这个企业的“共同认识”——基于这种认识，新员工就能有机会来调整自己以适应这个公司的文化与行事风格。

不要忘了，团队中有超过半数的人是从众者。团队教育对这一类成员的影响是良性的，既易于形成好的团队风格，又能使团队文化得到有效的延续——所以团队教育是“自建团队”中相当重要的方法与过程。但这却很难作用于前面所讨论的约占团队 $1/3$ 的理性利己主义者。他们相对理性，因此具有独立的思考与行事风格；他们选择有条件地合作，并期望合作从整体上来说总是利己的；他们算计着与他人交往的收益，并期望投入更少而又获得更多。所以，公允地说，有这种思维的成员更为理性、更有自主思考的能力。

这些成员并非“不可救药”——准确地说，是他们不接受你所给予的“药”，例如“被教育”。解决理性自利问题的法子是降低收益。确切地说，就是在同一层面上、更为公平的收益分配。在一个做事的团队中，如果收益总是以做事的多少、收效来衡量的，那么自然没空子可钻。

所以我常说：一个好的团队是不需要 **Manager** 的。因为 **Manager** 的收益并非来源于做事，而是来源于管理。反之，如果将这一收益方式置入团队中，例如，让“管理”或“在管理上的投资”成为衡量团队收益的标准时，那么围绕团队目标的、项目目标的“做事”就显得不那么要紧。而一旦实施这种做法，那 $1/3$ 的成员可能会选择跟随管理层，并通过感情投资或夸大功绩等投机方法来获得收益。由于 **Manager** 离具体做事的层面通常很远，因此判断一个人的方法往往来自于组织性的考量，例如是否听话，是否好沟通，是否能信任，以及是否看起来“不错”。

可见，搞坏团队的，正好是管理行为本身。

3.8 尺长寸短，人尽其用

管理行为应该从组织层面获得收益（而非从团队中）。例如，我在“愚公移山”中说到过的两个故事，当端木氏献上黄金车乘给晋王时，晋王赐他爵位；当端木氏展现军事政治才能时，晋王拜为高官。爵与官都是两种组织层面的收益，效用却各是不同。赐爵，是与端木氏的利益互换，有爵位并不一定参政；而拜官则是授了权，是要主理国事的。

从组织层面的行为来说，作为 **Manager**，你确实有机会将自己变成 **Owner** 或 **Leader**——这两种角色才真正地融入了团队。不过，在这种情况下，**Manager** 还需要考虑组织上的必要性。

温柏格的法子——“让他们签约”是一个简单的能让你变成 **Owner** 的办法；而他又说“否则离开”，则是典型的 **Manager** 的失败。“签约”在西方文化背景下很可能是一个好办法，并且从团队构建的方法来说——“约谈”并在沟通中达成共识，也是所谓签约的一个具体的可操作手法。所以总的来说，这是可行的。

但是如同前面所讲的，若团队有 **Owner** 或 **Leader**，那么“你从 **Manager** 转变成 **Owner** 或 **Leader**”的必要性就很成问题，并且这势必会给团队内部带来权力冲突与组织结构上的震荡。反过来，即使团队中还没有 **Owner** 或 **Leader**，那么“提拔与任命”仍然是你作为 **Manager** 的第一选择，而“亲历亲为”则是在你对整个团队及其目标、方向和流程方法有足够的驾驭能力之后的第二位的选择。

如果你不得不选择变成 **Owner** 或 **Leader**，那么我必须告诫你：你将不再是“官”。事实上，官性与奴性是我能找到的、对团队伤害最大的两种人性。因此，至少对于这个团队来说，你得收拾好官派的

做法与逻辑，准备好以 Owner 与 Leader 心态与能力来应付接下来的一切。

这个团队是做事的。你既然决定给自己一个角色，那么也就必然是一个做事的角色。

第 4 章 伯夷与叔齐是怎么死的

4.1 求活的逻辑

历史上，有两个人很有骨气地被饿死了，这就是伯夷与叔齐。这两兄弟先是相让而不做孤竹国的国君，后是相携而不食周朝的粮食。再后来，他们饿死了。

饿死了就饿死了，有人说“这死得很好”，那是跟“死得不怎么好”的情况去比较的。大体上，如果能好好地活着，是没有人会选择去死的。那么，他们又是怎么会死的呢？

首先，伯夷与叔齐认为周武王还没来得及埋葬父亲便要打仗，是谓不孝；作为诸侯要去讨伐君主，是谓不仁；于是这样出兵是不讲道义的，而这样得来的天下也是令人不齿的。所以伯夷与叔齐并非不想活着，而是周朝没有给他们一个——他们认为合适存活的环境。其次，伯夷与叔齐跑去首阳山上采野菜吃，但是有人给他们说：周朝统一了天下，所以粟米固然是周朝的，那野菜也是周朝的啊。于是二人无法可想，便都齐刷刷地饿死了。所以伯夷与叔齐并非一心向死，而是他们自己无法给出自己一个可以求活的逻辑。

既得不到一个可以存活的环境，也寻不到一个可以求活的逻辑，因此伯夷与叔齐便只能这样死去。这种舍身取义地死，当然必将传为美谈——我们需要这样的美，也需要基于这样的美来谈。我们的道德观念、价值取向需要以一些东西作为事例，以丰富我们在这些问题上的种种规约。

然而，没有一个项目会因为死掉而传为美谈。

4.2 没有产品或业务，还要做项目干嘛？

所以公司要给项目一个生存的环境，项目要给自己一个生存的逻辑。

如何构建项目的生存环境呢？

项目的生存环境与公司的项目管理体系有关，例如公司决定如何进行项目立项、审核、推动、考评等流程。这样的生存环境是一系列管理行为的总和，它与具体项目内部的运作方法没有直接的关系，但却实实在在地影响了这个项目的生存与状态。事实上，它也反映了公司对于产品、项目、团队三者的整体态度。

首先，公司是否基于产品建立了一系列的外围组织（或更为细致地说，“产品”作为一个市场化的基础条件是如何被定义、被评价以及是以何种形式作为公司运作的基础构件的），这是后续问题的基本前提。表 1-1 列出了三种需求类型^①。

表 1-1 对三种软件产品定义及其需求类型的简单考察

^① 这是对软件产品定义与其需求类型的一个不完整考察，其主要目的是形成相互参照以便后续讨论。

类型	简称	主要需求提出者	运行环境
A	定制生产系统	生产型企业	面向内部网（Intranet）或互联网上的特定用户
B	在线通用网站	网络用户业务部门或管理部门（公司内）	基于互联网（Internet），面向一般用户。通常开放注册，可能使用邀请制等方式来限制注册用户量，但不限制用户类型与可能的用户行为
C	离线桌面软件	桌面用户产品部门（公司内）	基于桌面操作系统或具体的功能运行环境，用户需求的功能明确，也包括移动平台或手持设备上的应用

下面，我们来分析这三种需求类型。

(1) 若需求方是生产型企业（以下称为 A 型），需要的是业务生产系统，那么产品通常是一个软件产品包，开发方需要现场安装、部署、调试以及技术培训；从长期来看，开发方还需要提供版本计划、维护手册、作业限制以及应急方案等。这事实上也包括开发方与需求方是同一公司或公司的不同层次的情况。

(2) 若需求方是一般的互联网用户（以下称为 B 型），需要的是一个在线功能，那么产品可能是一个线上网站，所有的产品构件都在开发方所能支配的服务器、机房等环境中运作，而使用功能的客户端可能是网络用户的 Web 浏览器，或者其他第三方网站通过 API 进行的访问。

(3) 若需求方是离线的桌面软件用户（以下称为 C 型），需要的是一个能解决具体问题的功能性软件——这包括桌面操作系统、移动设备、手持终端等环境下的软件产品，那么产品的形式可能就是一

个离线或在线分发的安装包，以及一些附属的构件（例如在线手册、在线升级或配套的工具软件）。

上述三种类型的产品定义、实现过程和交付方式都是根本不同的。例如对于 A 型，公司对产品的要求就是客户需用、能用以及符合客户的生产运行环境。如果客户现有的应用环境是 Java+Oracle DB，那么公司就必然是基于 Java+Oracle DB 来开发，项目所需的人力及与技术储备就都受到这一前提的限定，项目的评价自然就是产品订单，项目的立项依据就必然是业务人员接触到的市场信息与客户反馈——这些基本设定都依赖客户需求，而与开发方的技术能力与产品设计几乎完全无关。从公司整体上来看，应当是业务驱动型的组织结构最为合理，技术研发上的创新会少，技术实现上也会保守很多。

假定与 B 型来做一个比较，那么 B 型就并不一定适宜业务驱动。因为 B 型的业务部门可能会碰触不到确切的用户需求，以用户需求为核心依据的业务模式就会发生变化。反之，应从更大范围内来考虑以互联网用户行为为背景的产品推演，并关注基于业务链上下游需求的、功能性（或制约性的）产品设计。于是，在这样的背景下，面对需求问题，业务部门的形态将变成“观察—规划”型，而非（与 A 型类似的、传统含义上的）“收集—反馈”型。所以通常在 B 型产品的公司中，业务部门是以构建产品线为其核心工作方法的。进一步地，整个模式就会变成产品与产品线驱动，例如“产品”的定义就可能是子网站、多级域名或功能模块，项目的评价依据就会是上线下线、运行流量以及用户数量等因素。

所以，项目的生存环境首先依赖于产品定义与产品管理方法。进而，只有在公司内形成与**产品过程模型**相匹配的**项目过程模型**，项目的

生存问题才会有解。但不幸的是，大多数公司在对待这两个模型的做法上是有冲突的，尤其是在企业转型阶段——请尝试着考虑一下微软与谷歌在产品形态上的不同，以及微软公司在向后者转型中一些可能的障碍。所以，项目组并不一定总在“舒适”的环境中生存。也因此，当公司转型时既可能“砍掉”一些项目，也可能有一些项目自然地就消亡掉了。

产品与业务的规划，是公司为项目准备的生存环境。当我们人性化地来看待项目的生存问题时，这一环境正好决定了项目的价值观、归属感以及项目的基本道德伦理。例如，一个项目是否更好、更值得立项、更有效益，可以看成是项目价值观的问题；而一个项目是否觉得自己重要、受重视以及有前途，则就是一个归属感的问题。至于两个项目间的资源之争，是“掠夺”还是“奉献”，抑或是“有条件地相互合作”，那么就是一个基本道德伦理问题了。

4.3 项目的核心是事

对于“铁打的营盘，流水的兵”这句话，很多人认为说的是一个团队问题。

但事实上不是。因为团队总是以人为基础的，人——比如说兵——没了，团队也就没了，所以一旦“流水的兵”是事实，那么所谓“铁打的营盘”就一定不能归为团队问题，也不可能从团队管理中找到解。然而这一点往往会被忽视，于是我们会看到，为了寻求持续、长久的发展，管理者总是将项目与团队关联起来，试图通过“更好的团队建设”来得到更好的、更稳定的项目。然而最终，团队该散的还是会散，项目该死的还是会死。

到底是什么决定了“铁打的营盘”？

军营的体制不是为了单个士兵或将帅而设立的，部队可以拿“雷锋班”、“杨根思连”这样的荣誉称号来激励大家，但并不意味着这个班或连就只围绕这两个英雄人物来建设。军营的核心在于战争，训练、操演、技能、口号、人事、后勤、奖惩等都是围绕**战争这件事**来进行的。因此只要部队的目标还是保家卫国，那么不管士兵换了多少轮，也不管部队迁了多少地方，营盘都是那个营盘，军号定时地响起，连节奏都不曾改变。

项目的确离不开人，但项目的核心并不是人。项目的核心是事。事在、目标在、环境在、实现目标的可能性还有，那么项目就还有机会，项目就还没死——兵，可以再招；团队，可以重建；项目经理，可以换个人来做嘛。所以，根本上决定项目生死的，是决策层说**“No”**：产品不做了，产品线没了，业务方向放弃了，市场计划失败了.....诸如此类。

我所见过的项目，通常会把自己的生存逻辑设计为“资源丰富+需求明确”。这样的逻辑当然是好的，是要去尽量满足的。但是，如同伯夷与叔齐一样，这其实是给了项目一个寻死的逻辑：要人、要需求、要计划、要配合、要.....所有需要的一切，要不来就饿死，饿死了大家都玩儿完。所以在公司看来，所有的项目都在喊饿，所有的项目都岌岌可危。

活着的那个项目可能只有一个人，也可能一个人也没有。

今天做不了，明天做嘛。

4.4 求死之道

一个项目要做——当然我们不讨论根本不需要做的项目——那么也

通常只有两个“相当废话”的理由：一是因为当前必须要做，二是因为将来势必要做。在这许多个项目之中，甲领到了必做的项目，乙领到了要做的项目。于是甲、乙的生存问题便马上对立起来了。甲说“我要人”，乙说“我也要人”，然后齐齐地跑来问公司：怎么办？公司说：你们都很重要，所以一边分一点儿人吧。于是，甲乙两边人都不够，甲很快饿死了，乙慢慢饿死了。

于是，有人说：我们把要做的事分成重要而紧急的、重要而不紧急的、紧急而不重要的、不紧急也不重要的.....但只要这样一分，所有的人都立即跑来说：“我们项目是又重要又紧急的，而是还是所有项目中最重要而又最紧急的。”所以，更多人就说：管理要的是平衡；要平衡人，平衡事，平衡.....

平衡是求死之道。

4.5 不但要听 BOSS 的，还要听明白

与其如此，不如告诉大家真相：这个世界没有所谓的平衡。

“要平衡”或“要公平”只是你跑去找老板要资源的一个说辞罢了，无论你做了什么努力，或者老板因此做了什么样的决定，平衡事实上都是不存在的。你得到了资源，本质上就是掠夺了另一个项目的资源。你能谈平衡，只是因为你有资本站到老板面前去谈而已。

项目不是平衡出来的。正确的情况下应该是：该做的就做，不该做的就不做。于是你又会问：凭什么来分“该做或不该做”呢？答案是：不凭什么，那是决策^①。决策？是的，决策！哪怕只是你看起来拍拍

^① 其实并不是大家不懂得这个道理，只是大家觉得自己可以通过“种种努力”去影响决策，故而有了上下奔走，有了哭穷叫急，有了各种各样的组织活动。这些活动也的确增加了决策者在某些方面的砝码。但我的

脑袋就出来的决策，总好过没有决策——没有决策力的组织，乌合之众而已。

当然，如果你非得认为做决策的那帮人都是猪头，那么：

你还呆在这里做什么？

4.6 别抱怨

总的来说，我们需要明确的产品定义，并据此来明确我们到底要做什么事，以及基于怎样的组织体系来做。并且，我们假定那些决策者都不是猪头，因而项目明确，决策可行。那么我们还缺什么？

我们——在整体上——还缺一种做事的心态。《南都娱乐周刊》采访《金陵十三钗》时，张艺谋说^①：

当年抓这个剧本，为此付出了很多心血，四五年的时间去做一个好作品，希望能成功，希望是个好电影，我从心里来说，真没有具体目标。卖到10亿，要得奥斯卡，我没这么想过，我要这么想，就不是导演所应该有的心态，我就是平常心去做，尽管耳边有很多这样的话。我知道这些豪言壮语，但是我知道自己的工作是什么，我专心致志于把这个电影拍好了，就行了。

一个团队有一些不同的声音，有一些豪言壮语都是可以的。但做事的人要时时知道自己在做什么，该做到什么样子，并努力地去完成既定的目标。这种心态，就是整个团队应有的、对内的一种环境。当外在的、人为的、商业的、市场化的声音影响着团队的时候，唯

问题有两个方面，其一，这是否必需？其二，即使这一目的是必需的，那么除了这些方法之外，有没有其他方法来达到目的？

^① 引自《南都娱乐周刊》2012年1期，文《张艺谋：“我尽量保持平常心”》，采写记者谢晓、曾明辉。

一能让团队清醒面对的，只有团队的目标，以及团队中务实的工作态度。

公司季度大奖不是颁给想拿奖的团队的，也不是颁给能拿奖的团队的，而是颁给真正合适拿到这个奖的团队的。合适与否，不仅仅取决于你现在正努力着的东西，例如业绩，又例如你现在所认为的那些唯一衡量标准。

合适与否，是另一个决策。

4.7 现实

最后，作为组织中的一个个体，我们需要知道：并没有人会关注“你的价值”。

对于公司，他们关注的是你对企业目标的价值；对于团队，他们关注的是你对团队目标的价值.....类似于此的，只有在确定的目标下，才有确定的价值观，才有确定的价值。因此，“你的目标”才是“你的价值”的考核依据。而“你的目标”多数情况下是别人不关心的，所以也不要期望别人看到你的价值。

你应该感激那些关注“你的价值”的公司，这是不可多得的。但你尽可以感激这些得之不易的东西（甚至这也可能是假象），却不必期望可能得到它。因为，大多数情况下，HR（人力资源官）来找你只是因为入职、调薪或离职。

他们并不会问问“你的目标”以及“你期望的价值”。

如果真的问了，他们也只是希望得到一个标准答案：你的目标就是实现公司的价值。

这是现实。

4.8 更多的现实

没有人是傻子，爱公司是一个玩笑。

因为公司是一个组织，项目是一件事。只有团队是一群人——我们可以爱这群人，爱这个团队；我们也可以在团队中找到归属感，找到自己的位置，找到实现自我价值的方法与途径，并为此付出努力。然而，大多数团队的可笑之处在于：如果它认为“你在为自己努力”，那么它会鄙夷你的利己。因为我们的教育总是在说：要为了公司、为了产品、为了团队、为了项目.....

所以他们要么鄙夷你“为了自己”，要么无视你这样做。

其实，用人无过于四种境界，其一，我以己为器，认为一己之力便足够了，别人不堪一用；其二，我以人为器，就是把别人当成工具来用；其三，人以己为器，就是别人愿意把自己当成工具，粉身以报；其四，人以我为器，就是别人觉得做什么事都是在为他自己，而最终成就了你。

大多数个人，只是以己为器；大多数公司，只是以人为器。仅此而已。

这也是现实。

4.9 工程只是一个可选手段

团队的核心是人，项目的核心是事，产品的核心是物。三者是企业经营中的三个不同视角，不同的管理者从不同的视角入手，所持有

的观点、采取的做法以及关心的重点也都不同。

工程是什么？工程是实现（上述三者中的以“事”为核心的）项目的一个可选手段。因为在本质上说，工程讨论的也是事，以及事的做法。所以传统工程，也包括其他领域中的工程实施所讨论的三要素，便是“（做事的）过程、方法与工具”。三种要素都集中反映了“所谓工程，就是通过‘工程化的法子’来实现一个具体的目标”这一事实。

“立一个项”是我们常见的对开始某个事务的一般性说法。但就某一件事而论，具体做这件事的方法是什么，才决定了这一方法是否可以工程化。如果一个方法原本就不能工程化——我们举例来说，它过于依赖于一人一技——那么非要讨论“这个项目是不是一个工程”，在本质上就不成立。所以，公司大可以为种种事务而建立项目，但事事强调工程却并不见得可行（例如事事强调过程、流程、方法、工具化等）。

我们讨论的“人、事、物”三个方面都可以存在管理行为。仅其中的项目管理这一视角出发就有许多种不同的学派^①，它们各自的关注点是并不相同的^②。这些不同的学派，抑或说是不同的管理风格，也是适应不同的管理对象的。例如，一个管理者偏向于人际关系的梳理，那么他可能擅长团队建设与组织管理，但不见得能在生产型的项目管理中干得顺手；又或者把他放在一个偏向“做事”的项目/团队中去

^① 参阅《项目管理技术》2006.10，以及2007.01~11期“项目管理学派”系列文章，作者 Rodney Turner，师冬平翻译。

^② 泛义的项目管理对“项目是‘事’”这个观点是并不完全认同的。相反，上述这些学派将工程化（做事的具体方法）、团队（做事的组织）、产品（做事的结果与效果）等全部纳入了“项目管理”这个概念集中。举例来说，权变学派事实是将关注点放在了“人—组织”方面，而营销学派则可能将关注点放在“物—市场”方面，如此等等。

的时候，他就往往会显得过于官僚和权变。

“基于软件工程化的项目管理”是我们在本书中的一个重要命题。就上面的讨论而言，这个命题总体是有许多背景限制的。例如：

- (1) 是否有面向产品的组织结构；
- (2) 是否是做事的团队；
- (3) 项目对工程化是否有着本质的需要

等等。最后，管理者在项目的不同阶段是否有适当的管理风格，例如将自己设定为 **Manager**、**Owner** 或 **Leader** 等角色，也是“做事的环境”所讨论的因素。但在这个问题上，既可以通过任人治事来解决，也可以作为对管理者自身素质的要求。

篇二： 谋定后动，项目的 存在权

势者，利之所趋也。

第 5 章 试错通常是无能的托辞

5.1 知州孙觉的计谋

宋朝。福州。有富人跑去请教当时的知州孙觉，说他们打算捐几百万钱去修缮佛殿。孙觉是个慢性子的人，也就慢吞吞地问道：你说说，你们捐佛殿的目的是什么呢？大家回答道：求福报嘛。

这个故事的上半部分是说：富人有钱，富人求福报。

接下来孙觉就说了：你们看那佛殿，坏也没怎么坏，破也没怎么破，修了它能有什么福报呢？话说回来，我们福州有不少的穷人，因欠了官府的银子而被关了监，受着牢狱之苦。若你们把钱捐给他们，便可以使几百人解开枷锁。你们说，这救人于难的福报岂不是更大？

这个故事的下半部分是说：穷人没钱，穷人得活命。

说来说去，这个事情好像跟孙觉没什么关系。然而当时，福州的监狱人满为患：行政开销是问题，国计民生是问题，官府形象也是问题。总的来说，这些是孙觉这个知州的问题。

孙觉此计一出，监狱空了不少。

5.2 道德目标通常是障眼法

看明白了这个故事的人，大概会说我不厚道，又或者他们会说孙觉不厚道：孙老先生算计来算计去，银子倒变成了官府的了！

眼里只有银子的人，眼界便不过如此。

富人们要捐钱到佛殿，那便去庙堂里捐就好了，请问为啥非得跑去问知州呢？所以这个故事的底子里还有一层意思：富人在表面上是要捐钱给佛殿，以求得福报；面子下，却还是觉得自己这个行为很公义，很值得赞许，应当受官府的褒奖——好的话，给派个小官或任个会长之类的当当；即使差那么一些，也得立个牌坊授个匾额什么的。

面子上下的两个目的，哪一个才是真的目的呢？

我的不厚道在于说了一个权谋的故事，孙觉的不厚道在于看明白了富人们的用心。

做事情如果厚道是会受人赞许的，不厚道就会被人鄙夷。

当大家都有了道德目标，便没人考虑事情做得好不好了。

5.3 被撤职的 CEO

孙觉，字莘老。他与苏轼是好友，原本都是皇帝身边的重臣。莘老著《春秋经解》十五册，当时王安石一看便知道自己之上。所以莘老其实是儒学大家，放在古代，就是在道德的至高点上了。这样的人的作为，我们（以当时而论）又怎能说他不厚道呢？所以我们拿着现代人——或者仅仅说是现在的某些人——的价值标准，亦即“金钱物质的得失”，来评判孙莘老是行不通的。我们得放在当时的环境中去看，那样才能明白这个故事的有趣之处。

首先，于官而言，这些穷人犯了法而被收监，孙莘老就算体恤民情，也不能违法犯纪。表面上大家都知道得民心者得天下，但是把这些

穷人都放了，民心看来是得了，法纪却乱了，天下还是要丢。所以孙莘老让富人捐钱，是在法律边界内解决掉“穷人坐监”这一问题的不错的主意。

其次，如果孙莘老逼着富人们捐钱，那富人也是“民”——所以到头来，看着是偏向穷人，却是害了富人，总的来说还是鱼肉了百姓。且不说富人们有没有路子去跟他打官司，单就情理上也是讲不通的。

再次，富人一方面要捐佛殿，另一方面又打着小算盘要求个功名。这功名原本就是官府给的，给功名或不给功名都是官府的份内之事。既然捐佛殿是求福报，捐百姓也是福报；既然求功名是求利，那便认可了这一行为，给些官府份内可给之利，又有何不可？

富人原本是不想把钱捐给官府的。但道理上，捐给官府和捐给佛殿的所得利益是一样的——更何况，就情势来说，如果富人们非要捐给佛殿，那就不一定能得官府的褒奖了。所以，既然所得利益一样甚至更多一些，富人把钱捐给官府也就合情合理了。

种瓜得豆是乐事，因为得瓜还是得豆是无所谓的事；事与愿违就并不可乐，因为当有某个特定的“意愿”时，事做得怎样就不打紧了。我记得 B 公司的一个故事：年终的时候，CEO 的业绩还算不错，但是大老板却把他撤了职。问之，大老板答曰：业绩是你们定的。

他要的是目标，亦即那种“特定的意愿”。

5.4 势者，利之所趋

当然，这也要不违道德。

大老板是这样解释的：我们公司的价值观的第一条就是客户第一，

你们做了一年两年，看起来业绩是有，但是我们的客户不满意，有那么多投诉，连我自己——作为客户去用你们的产品都不满意，那你们还怎么能说你们做得好？

所以一个 CEO 当然可能纵容下属去投机取巧——或是做好了业绩而做坏了行业，或是干脆失了道德去伪造了业绩的假象。然而放在一个道德问题上去炙烤的时候，这样的做法就是经不得考验的。至于大老板，他之所以忽略业绩，是因为业绩已经有了；之所以追问价值观，是因为价值观失了。

得与失，只是一瞬间的转变。这就是所谓形势。

势，就是利之所趋。看到利益是什么，看到利益的趋向是什么，也就是看到了势。所以孙莘老是看得到形势的人，大老板也是看得到形势的人。

大多数埋头做事的人，都忘了看看形势。

5.5 随势就利、投其所好皆不过是手法而已

形势是两个东西，我常说“看山见形，看水见势”。在 B 公司的故事里，一时的业绩便是那可以看见的“山之形貌”，公司内在的追求便是那难于明辨的“水之势向”。

作为一个项目经理，所做产品便是确定的“形”；采用软件工程来做，便是具体的“法子”；而所谓客户需求，便是基本的“利益”；更进一步，用户的利益取向就是这个事情的“趋势”。

在一个产品开始做的时候，用户问的是“能不能用”，所以只要功能满足，“多快好省”的东西大抵都是满足用户要求的。这种情况下，

快点做产品，快点圈用户，快点树口碑，都是当务之急，因为这些都是该阶段下用户所追求的。但到了往后一些的日子，用户积累起来了，需求分支变得混乱而复杂了，系统整体有了性能瓶颈了，这个时候再强调“快做快上”就事与愿违了——因为用户开始讲“好不好用”的问题了。再到了第三个阶段，基本的市场格局已定，用户已经达到了相当的基数，而竞争对手的关系以及领域细分变得微妙起来，这个时候再讨论“某一个用户能不能用或好不好用”的问题，便是吹毛求疵的事情——这种情况下要讨论的，通常会领域扩张的问题以及新领域中用户的需求问题。

然而在第三个阶段中还有两个有趣的问题。第一个有趣的问题，我们既然说“用户的利益取向”是势，那为什么我们看起来却是忽视了“当前领域中的”用户的利益取向呢？

这取决于“产品的用户群体”这一定义是否扩大了。如果没有扩大，那么我们必以这个产品的当前领域为利益核心；如果扩大了，我们就得将整个领域作为利益核心。以 Adobe 为例，Photoshop 是为平面工作室而定义的产品，在这么多年的发展中从未改变过；与此同时，Adobe 通过产品线定义，将 Premiere、Flash Catalyst 等放在 Creative Suite 套件之中，来完成以整个“图形、图像艺术设计师”为对象的产品设定。与此相对应的，ACDsee 早期的产品定位为“阅图者”（viewer），在后期却扩大到“阅图者+制图者+数字图片管理者”这样一个复杂的用户群，并且仍旧使用“单一产品”这一概念来应对用户需求，以至于 ACDsee 5.0 Powerpack/Suite 之后的产品一直都无法满足各类用户的“集体意愿”。

形势在变，观念不变是不行的。一方面，单个用户既需要单一产品的便捷，又需要组合产品的强大功能；另一方面，企业用户既存在

特定业务为核心的产品需求，又存在以部门与角色分割的功能选择。因此我们要么拆散了用户群，以不同的产品去面对不同的用户；要么以（可以自由组合的）套件的形式来满足用户对产品定制与搭配的需求。

我们自己也是用户。

5.6 你无法讨好所有人

第二个有趣的问题是，在这个阶段中，竞争对手可能还处于一个较小的规模之下，这时他们便会强调“能用、好用”的问题，而且还会以此为优势来打击你的产品。于是便会形成一种局面：大公司的东西“似乎”越来越不好用。

产品与产品线总会走入这样的困局。用户的口味越来越趋向于差异化，但产品要规模化生产与供应，却只能趋向于同质。这才是困局背后的根本原因。所以，一个工业化产品无论如何都只能在“一定程度上”保证品质，而很难超越这个边界来使它变得跟一个小规模生产的（如手工生产的）产品有同样的性质^①，或保证会受到某些特指用户的欢迎。

大老板是特指用户——注意，任何一个个体都首先是特指用户，这一点非常重要。正因为大老板也确知自己必然难于站在“典型用户”的角度来对产品性质提出意见和建议，所以大老板通常不会对产品的性质提出质疑。但是如果这一点能被确实——我的意思是说，某个管理者、决策者或管理与决策层能确实将自己放在典型用户的角度上，那么他们也只会在两个时机指出“产品性质问题”。

^① 这里说到的是“性质”，而非品质或质量。手工生产或精细生产并不是品质或质量的唯一保障条件。

- 第一个时机是管理层的变化。因为事实上产品性质必然存在问题，至少（我举个例子来说）在一段时间中总会有 5%上下的用户对产品持续不满，因此任何时间追问这个问题都必然是管理层权利变更中的一个权谋之术。
- 第二个时机是产品的转型。产品转型并不一定是“变成另一个产品”，也可能是在产品面对的用户或面对的用户品味方面的调整，因此当大老板“放大”了问题，则必然是试图通过对这一问题的锁定，来将产品调整到一个新方向。

如果大老板“不择时机”地、随意地对产品性质提出问题，那么他不但浪费了自己的一个管理工具，而且还使得管理、经营与决策三者在产品问题上辗转碾压、互为制约，进而产生大量的内耗。

“大公司的东西”总会对某一部分用户不好用，知道对某些用户以及某些需求说“No”是明智的，但学会“如何说 No”则是一门艺术。例如，公司可以宣布“我们只做高端消费者市场”，或者在招牌旁边贴上“五星级酒店”的标志，那么本质上就是在对某些普通用户说“No”。这种情况下，当某一个用户走进店内，面对种种价格乍舌的时候，只需要有服务生走近他，无比诚恳地说：先生，请问我有什么能帮到你的吗？

当然没有。事实上作为用户的我们总在被拒绝，只是因为被拒绝的形式不同，故而我们的反应也就有不同罢了。在软件产品上与此是类似的。“承诺产品特性”与“拒绝用户需求”是我们面对市场的两张王牌，但大多数的市场人员、产品设计者与经营者只懂得运用前者。这也就是 ACDsee 产品特性爆炸与麦当劳千店一味这两种风格在产品设计与经营理念上的本质差异。当然，这并不是说 ACDsee 就该“放弃产品特性”，而是说，ACDsee 在“做什么”这个问题上或许是对的，但在“怎么做”这个问题中却确凿无疑地错了。

5.7 错了就是错了，哪有什么“试错”

“更多的产品特性”并不一定是正确的事情，所以看起来人山人海的开发活动中，各个小组在做的事情也就不一定“都”那么正确。总有团队、总有项目、总有产品会是牺牲品。切近形势的大体上会好一些、保险一些，但一旦形势逆转，又福祸两不知了。这也就是我们所谓的“大.....规模开发”中的现实状况——所有人都在战战兢兢，为项目的生死而担忧，为“明天做什么”以及“我能不能做得了”而焦虑。

我在“大规模开发”中间加上了省略号来表示“无限”。因为在我看来，这是一种由“产品多寡”带来的虚假的繁荣。这种虚假繁荣背后是有理论支撑的，亦即“快速试错”与“短平快项目”。前者是一种产品策略，后者是应对产品策略的项目方法——与此还多少有一定关系的，是敏捷开发方法。“快速试错”真的是可行的产品策略吗？答案是肯定的，但同时必须明确：所谓试错总是要付出代价的。这种代价要么是时间，要么就是组织。“较为明智的”做法大体上是牺牲二者中最小的一个。例如创业团队，组织规模是牺牲不起的——总的来说就三五个人十来条枪，所以就只好牺牲时间，于是创业迟迟不成通常是因为产品投入得不对，最后试错成本越来越高以至于不得不放弃^①。又如大公司的产品线，在组织规模上“通常”是牺牲得起的，因而往往试图同时开发多个产品来及时响应不同的市场需求，于是组织规模越来越庞大复杂，最终组织成本大于产品成本，又变成得不偿失。

试错的定义终是基于成功二字的——成王败寇就是这么个意思。但是对于我们要讨论的对象来说，试错的对象不同，成功的意义也就

^① 很多人庆幸在这一过程中“获得了不少的学习经验”。但这其实已经失去了“创业”的初衷，是最无可奈何的下下限而已。

不同。在团队、项目与产品这三个企业经营的关键视角中：

- 如果试错以团队为目标，则显而易见地变成了组织重构这样的事情；又若以团队为试错基础，即团队可以自行选择要做的事情以及其价值判断，则是各自为战的战术方法。
- 又如果试错以项目为目标，则容易多劳而无功，渐成组织内耗，例如，一个产品开发完就被扔掉便可能是运营部门、生产部门所谈的“项目”在内容上互不相同；但若以项目为试错基础呢，则项目是否作为基础单元以包含对所涉及全部资源的把控，便成必然要求。
- 因此，第三个视角便成了更为常见的选择，即以产品为试错对象。这既在于产品的成败有明确的定义，又在于“面向产品的项目”的管理与控制过程在企业的组织形式上已经较为成熟。

总的来说，在这些视角下存在着职能部门、事业部与产品线这些不同的组织形态，以及各自不同的关于成败与价值取向的定义^①。

试错的可行总的来说取决于两个方面的前提。第一个方面的前提是组织的可行性。在一个不适宜的组织结构上“试错”，其代价必将是惨烈的——所谓船大难掉头，倒不真的是“船太大”，而可能是船上的拖网太多，要先砍掉拖网才能掉头。所以，这些试错产品便是随时要砍掉的拖网，于是这些拉着拖网的人，便随时可能因为碍事而被推下水。进而，即使船再大，船上的人也都无不战战兢兢了。

一个组织，例如公司或部门，对于试错的结果是否有预期？对于试错的“错”是否能承受？对于试错的收益如何评价？对于试错团队的牺牲持何种态度？这些都是组织层面要考虑的事情。例如，如果我

^① 例如，职能部门以部门的、人的利益为核心，事业部以一类事务为方向，而产品线关心某类用户的需求等。

们要有尖刀班尖刀连，那就得有让他们“虽死尤荣”的体制与背景^①。此外，“短平快”带来的通常是项目主体完整而缺乏细节。因而如何将一个“缺乏细节”的产品推到用户面前而不会招致反感，是一个组织化的行为，也就是说，需要市场、营销与产品的配合。因此，“短平快”如果缺乏组织力，与“死得快”是没区别的，开发完就扔掉、还没上线就说 No、用户三分钟热度等这些问题，基本上都是由此导致的。这也是大公司通常会做“小规模内测”，以及划分高端用户、体验用户、概念用户等特殊用户群体的原因——当我们把“试错”当成一种稀缺资源时，试错就变成了“试鲜”，成为用户有品味的、有 VIP 身份的或者有特权的一种表征了。

第二个方面的前提是试错的必要性。如果一个组织中的大多数项目都是在试错，这既对组织整体的积极性、荣誉感是一种打击，对经营收益来说也是巨大的负担。所以总体趋势上，我们是要减少（而绝不是增多）试错项目的。因此试错绝非“正常的产品决策”活动中的首选手段，它只有在当产品——必须再次强调，这里讨论的“试错”是以产品为目标的——面临类似如下无法直接抵达用户或抵达成本过高的困境时，才适宜作为一种产品的阶段性决策工具来使用：

- (1) 无法获得用户反馈来支撑它的价值（例如产品线中规划的关键产品环节）；
- (2) 无法与产品的直接用户沟通需求（例如通过市场调查报告来决策的产品）；
- (3) 难于获得用户使用产品的细节等。

^① 壮烈牺牲是与国家的烈士抚恤，牺牲、病故抚恤和伤残抚恤政策等一体的，例如民政部门就分管革命烈士褒扬，指导优抚事业的管理等事务。否则，烈则烈了，壮则无闻矣。

永远不要试图生产一个“不知道用户是谁”的产品来试错，这绝对是系统性的灾难。反之，若是已经明确了或可以通过用户接触来逐渐明确了用户是谁，要什么，用什么，怎么用等问题，那么这样的产品就不是“试错”，而是“试对”了^①。

真正有效的、可实施的大规模开发，既是组织上适宜的，也是方向上适宜的。在组织与方向不匹配的情况下的“规模庞大”，既是假象也是负担。而在这其中，“敏捷团队+测试驱动”只是适宜作为一种试错的方法而已^②。然而无论投入产品的数量，还是实施方法的合理性，实质上都是无助于解决上述问题的。之所以我常常不讨论方法问题，而偏向于讨论方法的前提，其原因就在于此。我们往往在“技术上”准备好了试错，但试错的前提（组织的与方向的，可行性与必要性的）却常常不具备。这种情况下，看起来不错的项目或产品，其最终的失利自是必然，而其成功也多是机遇罢了。

5.8 再论“做正确的事”

做公司总是要挣钱的，不挣钱大家就得散伙；但是如果把挣钱这件事情说得那么直白，一定程度上来说，大家还是得散伙。这个问题我们在前面的章节里也讲到过，我们如果鼓励公司以“经济价值”为先，事实上也是在鼓励“经济价值”为先的道德模式；如果以此为核心，则员工的利己行为便倾向于对“经济价值”的考量，倾向于“如何投入更少而获得更多”。

^① 许多类似“成功学”的励志访谈中，创业者的经历总是被解说成充满机遇、运气与挑战的试错过程。但细细分析，大多并不是因为有不惧试错的好运气，而是他们往往在选择性地“试对”。

^② 试错的基本保障在于成本可控，而这一模式在本质上有两点是适宜试错的：其一是团队规模，其二是开发方法。

一种可能的做法是“在商言商，在民言民”。也就是说，道德标准对内和对外是不一样的，是一种选择性结果。例如，我们的市场人员去跟客户代表谈时会说：你看，我们做企业的总得挣钱，对吧？而对组织内却强调：我们都是一家人，所以应该互帮互助，对吧？

对吧？

其实，不怎么对。因为这是做人的做法，不是做事的做法。仅做事而论，如果是去跟客户谈判，那自然是如下这样的目标：第一，为公司争取最大利益；第二，为公司争取最大的合作空间；第三，在为公司争取最大利益与合作空间的基础上，让客户最大程度地满意.....至于这样的目标，你告诉或不告诉客户都是不要紧的，因为他们必然知道。当然，如果你收受客户的好处，那也就相当于告诉客户你并不看重公司的利益与目标了。

再讨论“对内”的问题。无论如何，事该怎么做、该谁做都是和职业角色与责任相关的，因此第一等的话题是：

|(1) 该，或不该你做。

如果舍了这层关系，例如在创业团队中，大家都是有着共同的目标与期望的，也有着自己的能力与特长，因此第一等的话题是：

|(2) 我能做，我立即动手去做。

再舍掉这层关系，例如在一个松散的组织结构里，可能还没有既定的责权关系，那么第一等的话题就应该是：

|(3) 如果我合适做 **Leader** 或 **Owner**，那就让我来做。

这三种做事的法子（或态度），可以一言以蔽之：或尽职尽责，或各展所长，或当仁不让。

所以，所谓做事，也就是个能力与才干的问题。做事但求有尽心尽力的态度，这样一来，即使是看起来圆滑世故一些，也是受人尊重和堪能任用的。所以要鼓励员工做事，要形成“做事”的形势，那么公司也得是一个任贤用能、实事求是的公司。尽管从组织结构上来说，人的考察、调度与任用等，不免会涉及“做人”的问题，但这仅限于组织层面上就好了——我说过，管理者的利益得失要从组织层面上去找，也就是这个意思。

“先做人，后做事”是一种道德要求，而并不是做事的“法子”或前提。以做人论，宋朝的丞相宋庠^①就曾经开了一个不太好的先例。他任守洛阳的时候，有一个仆人来检举主人行李中有漏税的东西。对于这件事，宋庠认为主人故然有错，罚缴税款即可；仆人控告主人，也是过错，所以打了一顿板子并赶了出去。在这个例子中，宋庠就是把法理判案这样的“做事”，套上了“做人”这样的道德规矩^②。

结果呢？规章制度乱了，事情也就渐渐地乱掉。

人做成了好人，事做成了坏事。

5.9 “虽有智慧，不如乘势”

当以道德准则来看一件事时，事做得好与不好是不要紧的；当以宏观趋势来看一件事时，事做得好与不好也是不要紧的。所以，做事的评价，不在于事做得好与坏。

^① 音“祥”。

^② 这个故事的背景是在讨论儒家的“尊卑有序，下不欺上”的道德观。这样的道德观本身带来的是类似“欺下谄上”的道德败局。但我们这里并不讨论道德观本身的对错，而是讨论治人与治事之间的冲突。

而在于正确与不正确。

事可以做得漂亮，例如工程化可以细致到程序代码的行数统计，又例如流程的规定可以设计到一个人的左右手的不同分工等。总之，我们可以把一件事做成无可挑剔的范本，但是放到一个观察者的角度上去看时，它可能仍然是一件坏事。

我们多有做事的激情，而少有观察者的冷静。这让我想起在中式教育中的一个非常有趣的观念，叫“有眼色”。小的时候，我的父母亲便会教育道：家里来客人了要倒茶，大人在午休的时候不要吵等等，诸如此类。总的来说，有眼色的意思大体就是“知道在什么时候、什么情况下该做什么事”。

察情、知时、应势，那是很好的品质啊！哪个管理者、哪个公司不希望自己手边都是“眼里有活儿”的员工呢？然而为什么“有眼色”现在变成了乖巧狡佞的代名词了呢？进一步讲，我们的教育只强调培养埋头读书的孩子，以及埋头做事的员工。我们一面叹息大家眼里没活儿，另一面又把员工的颈子摁得死死的，不教他们有举头四顾的一丝丝机会。

眼色这个东西，用在人情便是察己识人，用在析理便是真知灼见，用在决策便是远见卓识，用在奉承便是奸滑谄媚.....但是，这些都只是用的问题，而并不是“有眼色”本身的问题啊。做事与观察做事是两个角度。于前者，努力是必需的，认真是必需的，细致也是必需的.....但这些都是“做事的必需”。如果你是管理者，那么你正好应该是站在后者的角度——观察做事。

所以你的评价依据以及常常反省的，应该是这件事的正确性。

第 6 章 合法的山大王为什么没能成功

6.1 “这个团队大概是没什么希望了”

我们谈到过：团队中积极或可能积极的成员约占到 $2/3$ ，而约 $1/3$ （20%~30%）是“理性利己主义者”。如果单单看到这样两笔模糊的数据，那么团队大概是没有希望的了。

在大多数不太成功的团队中， $2/3$ 的成员挣扎于这种“失去希望”的境地。他们或努力地避免自己沦为那些理性利己主义者中的一员（这事实上并不容易），或对这些人的存在视而不见。根本上来说，这是因为，对于组织整体而言，他们既没有权利也没有责任来处理这些人带来的团队问题。

这是谁的权利与责任呢？

6.2 是温伯格解答了我的疑问

我喜欢开会。这是很少人知道的一件事情，因为大多数同事都将我视为“会议敌视者”。事实上他们并不知道，在数年之前，当我的工作中有着大量的管理事务时，我总会“快速”地组织起一场会议。

——然后又快速地结束它。

之所以我们大多数人变得“敌视会议”，是因为那些会议组织者的无能，而非我们对会议这种形式怀有敌意——只可惜大多数“喜欢开会”的人并不能意识到这一点。他们的喜欢开会总是体现在：

- 准备好记录工具，例如记事本或录音笔；

- 准备好食物饮料，如果有两颗老年健身球之类的就更好了；
- 找个相当适合开会的地方，例如茶楼或会务中心；
- 让会议开始于自我介绍，而不是对问题的陈述；
- 不设定讨论的范围，而寄期望于会议中“产生”点儿什么出来；
-

而我则一直在努力地理解为什么我会敌视这一类的会议。后来的后来，我终于知道^①：

希望自己独立解决问题的人，与希望延长解决问题的会议的那些人之间存在着矛盾。

6.3 你是“66%的从众者”吗？

“独立解决问题”是一种很特别的能力，它既决定了你在团队中的位置，也决定了你的组织风格——例如开会。

我能快速地组织并结束一场会议的基本原因在于，会议对我来说，主要是起到两个作用：其一，能快速统一一群人的行为；其二，能快速得到一群人反馈。我组织的大多数会议是发布决议或（统一地）调整工作方法、步骤与组织责权——简而言之，就是要让大家去做，或要让大家知道的事。例如，我可以为了让大家写一份好的 Word 文档而组织一次学习，但不愿意为讨论“要不要统一文档格式”而开一次会议。

我绝不会为一个会议而“设立一个问题”。这个问题若提出了，我必

^① 引自《成为技术领导者——解决问题的有机方法》，温伯格著。

是在得到解的时候再开会^①；若它是在会议中提出的，我会质问“为什么不在会前提出”并将这个问题直接纳入会后讨论。我并不习惯“用会议来解决问题”，或“在会议中解决问题”。然而，温伯格提到的“希望延长解决问题的会议的那些人”正与此相反，他们开会的目的就是为了解决问题，以及“部分或全部地解决问题”。

习惯于“独立解决问题”是领导者的一种基本素质。这也是那 66% 的团队成员从众而你出众的基础条件之一。需要说明的是，在思维法上的“解决问题”是关注于问题的求解方法，而不强调它的实施——你应当努力找到问题的解，而通过团队力量（而非独自）去实施它。然而，66% 的从众者要么并不能发现这些问题，要么发现了问题而不主动求解，要么有了求解的思路而无法推动实施。

前两种情况是从众者本身“之所以从众”的本质特性，亦即他们性格构成的一部分；后一种情况则是组织的责权问题。

6.4 给“1%的人”责权

组织的“可笑”之处在于：并不是有能力发现问题、求解和推动实施的人，就一定是拥有相应责权的人。组织的授权过程是渐进的，他首先取决于组织决策者对你的信任，其次是你的能力，最后但也是最理想、最复杂的是二者的组合。

当团队陷入“从众+集体自利”的泥潭时，组织行为是唯一有效的解。当这种组织责权未出现明确授予时，就出现了两种情况：一是等待

^① 例如，“要不要统一文档格式”，若它是一个问题，则我所组织的会议就是用于通告“统一，或不统一”这样的一个决议的。若是前者，则可能还包括一系列用于培训“统一格式”的学习课程，并且在开会之前，这些课程就已经定下了。为了达到这一目的，我可能已经走访了一些人、一些部门，并组织了培训讲师、制定了时间计划。

组织觉醒，二是团队内的利他角色的出现。基于奥尔森的观点，经济学家发现：不同文化的社会中都存在一些利他主义者——为数少于 1%；模型模拟的结果显示，若这个比例提升到 3% 以上，则整个社会的风气和行为规范就会出现可喜的变化^①。例如，三个和尚没水喝，但如果有人出来制定“轮流挑水”的规则，那情况就立即改善了。问题在于：

(1) “制定规则”这一责权是三个人中谁都不具备的，且这件事对这个人自身并没有任何特殊好处。所以他必是首先出于“有利于（包括其他人在内的、全体的）组织”的思想，才会着手制定与推动上述规则。

(2) 这个人可能必须身先士卒地挑起第一担水，并且他又必然因此面临“从明天开始”没有人服从这一规则，而白挑了这担水的风险。

无论是谁来挑第一担水，整个的解决问题的方法都取决于“轮流挑水”制度是否能有效保障。而这只能依赖强权，或群众运动——这也可能演变为政治，例如争取一个从众者，形成 2:1 的局面以强行推动这一制度，并实施惩罚措施。

“轮流挑水”作为三个和尚问题的求解方案之一，将要求“规则制定者、维护者”这类角色的出现。这本质上是组织行为，且在现实中也通常是由公司通过组织委派来解决的。但作为一个有自省性的、有自发力量的团队，这个问题也可能被潜在的“利他角色”来解决——不过在大多数情况下，这个利他者将因为对责权的需求，而会表现得强权、强势和政治化^②。

^① 引自《经济学家茶座》2002 年第 3 期，“奥尔森学术思想介绍”一文，作者陈抗。

^② 也许所有的和尚都知道“轮流挑水”能解决问题，但谁跳出来制定和维护这条规则谁就是众矢之的。真正的“领导者”敢于把自己置于焦点，并同时有具备让自己不被射杀或灼伤的能力——在这个过程中，“为谁

我们的教育中是反感这种角色的，但往往团队却渴望这种角色的出现。我们一面焦渴着没水喝，一面憎恶着团队的堕落，而又同时对那个跳起来试图解决问题的人投以白眼。渐渐地，一种坏味道漫散于整个团队：既愤愤于没有人解决问题，又愤愤于自己的想法不被重视，还愤愤于那个出头者抢占了自己的机会。

持尊重强者与不畏强权二者之一的，都是弱者；二者兼备的，才是强者。

6.5 领导者的眼光

领导者是那个能独立解决问题并拥有相应责权的人。一部分人正在成为领导者的过程之中，所以他们表现得聪明、果敢，富于决断，并且处于权力的上升期；另一部分人已经成为事实领导者，因此在权力方面处于衡定期，而权利问题就变得明显起来。

“因权而谋利”并不是一件坏事。事的好与坏是有角度的，就如同我们前面举的美国民众的例子一样。权利——作为一种利益——是有视角与立场的，当为民众谋利时大家就都支持都赞成，而当大家发现你在谋取小集团利益时，民众就开始反对了。

群众的眼睛总是雪亮的。

这就回到了我们对问题的原始设定上：团队的普遍利益是什么？我提到过以下几点。

(1) 项目利益：项目是面向“事”的，所以必然要求这是一个做事的团

谋利”是所有强权、强势和政治手段合法化的基础条件。例如，美国民众就向来是“国家在国际事务中应起到领导作用”的支持者。

队，所以有事做、做对、做好是基本前提与利益核心。

(2) 个体利益：这个团队中的成员是各有私利的，各角色有其在利益上的基本底线。

(3) 团体利益：可以参考马斯洛“需求五层次”来考察**团队整体**的利益。

而领导者的眼光，或者说他对大势的主要判断，便在于他如何认识什么是真正应当寻求的“利益”。

6.6 山大王的诞生

那么我们来讨论一下“组织的革命”。

随着新任 CEO 的上任，C 公司面临着一次重大的内部组织变革。这次变革概括起来有如下几点：其一，从产品管理走向产品线管理，在全公司范围内，有生产（包括内外部的软件开发）活动的都称为一个产品线，名曰“山头”；其二，权力下放，山头设有“山大王”，其内部组织由山大王在全公司内选择，产品线内部决策可以山大王一言而决；其三，资源放开，山头经营所需资源作为预算，由山大王提出并报批，按预算计划拨给。

这次变革的核心动力来自于两个方面。其一，原有产品管理过于碎片化，资源使用上的冲突面临失控；其二，新的管理高层对于既有业务的体系和方向并不明确，寄希望于此次改革能产生新的团队、业务、产品线与经营模式。大体上来说，这样的一次变革涉及很多部门的重新组织，也涉及许多的产品与产品线的启动与中止。当然，不可避免地也涉及了许多的权力、资源与组织关系上的冲突——但这些，是在这一变革的决策中可预期并计划承担的代价。

但这样的一次变革的结果如何呢？

事实上这次变革除了在形式上的轰轰烈烈之外，其他的几乎什么也没有改变：

- 原有产品一个没少，还产生了更多的产品，并因为产品线的阻隔而需要更多的资源；
- 原有的大多数职能性、管理性的部门并没有撤销，并按着原定的规则继续要求评审、考核、组织会议和配给资源等；
- 原有企业的经营目标被暂时性打破或置疑，但新的目标尚未明确，因此各个产品线处于“自我规划”的状态，完全缺乏通盘的考虑；
- 由于资源存在竞争，因此在各个山头的规划中普遍出现了“过高地设定目标”的情况，这一方面可以使规划更有吸引力，另一方面也有利于争取更多的资源；
- 原有的软件开发过程没有变化，仍然基于“产品需求文档”（PRD）并基本采用瀑布模型来推进工程；
- 部分项目组尝试新的工程过程模型，却因为公司组织机构上既有流程规范上的限制，以及公共流程环节（例如产品部门）存在资源竞争而无法有效推进——通常情况下，为避免此类资源竞争而设定了计划、优先级与流程，而这正好又是体制性的；
-

如上所述，整个组织与经营活动在短时间内乱象纷呈。但事实上管理高层并没有为之所动（这也是决策力中相当重要的部分），因此这一决策仍然被忠实地贯彻了下去。这些浮现于表面的问题点，既是一种战术需要，也被视为是战略推进的必然阶段。也就是说，如果乱象是暂时的，而其收益是长期有益、有效的，那么这些代价是可以接受的。

什么样的收益呢？

前面提到过这两项重要的收益。其一，是否可能在这个过程中产生新的经营模式？其二，是否可能顺利过渡到产品线管理？如果这两项目标达成，且这个“乱”的过程（在资源和时间上）是公司可以承受的，那么这将会是一场成功的改革。

但事实是：这次改革彻底地失败了。

6.7 方向明确，制度有效，知道怎么做，以及知道做什么

看起来，这个策略本身存在很多的问题。例如，产品线过多，并将因此导致跨产品线的决策效率低下；又或者，产品线为了安全起见，完全可能在新产品的思路趋于保守；又或者组织层面并没有解决资源竞争问题，因而增加了失败的风险.....但这些都并非这次改革失败的核心原因。因为我们定义一个战略的失败与成功，并不是看细节上的代价与冲突，而是关注它在战略目标上的关键设定是否达到，或是否失去了达到关键设定的**基本条件**。

对于这次改革来说，其策略的关键点（即其**基本条件**）在于“产品线上是否有真正的领导角色”。这既可以通过组织任命来达到，也可以寄期望于在经营过程中有人能脱颖而出，或者还可以通过刻意地战术侧重来形成。但这三种情况（或更多种可能）都必然有明确的前提，也就是我们需要清楚地知道：一个合格的山大王应该是怎样的？

这个问题却正好在改革之初就没有明确。回顾这次改革，C 公司设定了“新的经营模式”与“产品线管理”这样两个目标，但却没有仔细观察在组织中是否真的有具备相应能力与素质的管理者。也就是说，我们并不确知组织中是否有这样的人；又或者，即使我们先有了某种确定的组织，我们也不确定有人能胜任所需的组织角色。然而即便如此，这一革命性的战略仍然是相当值得尝试的。如果我们确知

“我们要找的人是什么样子的”，那我们可以先改变组织，以便等着那样的人“跳将出来”——尽管这已经是接近太公钓鱼似的妄念了，但它毕竟仍然是一种可能的选择；而且事实上在创业团队中，自发的角色产生仍然是一种主要的模式。

如果我们真的要做这种尝试，那么我们究竟要以什么样的“山大王”为标准呢？按毛泽东的说法就是：就算当山大王，也要当“有主义、有政策、有办法、闹革命”的山大王^①。

首先是“有办法”。这是个方法问题，如果一个人在山头中是明确的探索者、实践者，有着层出不穷的“鬼点子”并努力付诸实践，那么他便有这样的基础条件。大多数从众者愿意跟着“有办法”的人走，这是简单的趋利模式。也就是说，看起有“有办法”的人胜率总是要高些。“有办法”是一个团队的基础条件，但不是领导素质，也非领袖气质。举例来说，团队中可以有智囊，可以有军师，因而领导可以通过决策力来解决“办法”问题。

“有政策”通常很有可能是外部——更高层级的组织给予这个山头的限制，这更多地出现在既有的（而非新创的）组织结构中。例如，红军的“不拿群众一针一线”就是一条很有效的政策，它约束了山大王的行为，是红军之所以成为革命之师，而不会变成军阀流寇的政策保障^②。另一方面，“有政策”也是对这个山头经营方法的细化和规则化，是对所谓“有办法”的一个补充、概括和提炼。领导者对于方法有着天生的敏感，但又非学问家似的书面化，他必以简洁可行的

^① 引自《历史选择了毛泽东》，叶永烈著。

^② 按照我的惯例，便在这里留一个彩蛋吧。试问：“不拿群众一针一线”到底概括了哪些管理方法以及对管理效果的评价？

方法来形成可实践的、可集团化的规则，这是“有方法”之于团队的价值。而这一点，也是C公司的“形成新的经营模式”这一战略目标所需求的高度概括能力。

战术与其规则化，都是可以学来的。比如，当时井岗山上的山大王、土匪朱孔阳的战术就是“不要会打仗，只要会打圈”——只要部队还在，就可以打仗，就还可以继续做山大王，所以朱孔阳的办法归纳起来还就是“山头策略”。而毛泽东的高明处之一在于将“打得赢就打，打不赢就走”、“必要时拖队伍上山”^①等类似山头策略的办法，变成了著名的《论游击战》。但是，远远不仅限于此，毛泽东非常明确地指出^②：“游击战争一刻不能离开民众，这是最基本的原则。”游击战并不是简单的打打跑跑，它在指导思想上就被毛泽东分解成了“游”和“击”二字：“打开以发动群众，收拢以应付敌人。”这样一来，当“游”是有目的地发动群众时，“游”就不是东躲西藏了；当“击”是收拢而战时，“击”就表现为以强击弱了——游击战整体既是有目的的“游”，又是有目标的“击”。

这样的战略、战术与战法，是有系统性的。这既有系统的关系，又有系统的背景，这一背景即是游击战的基础：只有在“闹革命”的地方，游击战才有战的必要性；只有在“有主义”的地方，民众才能配合共产党打好游击。“闹革命”与“有主义”既互成方法，亦互为思想，还互作背景。这事实上也区别了“山大王”与“领袖”的不同。毛泽东等老一辈革命家自言上井岗是当上了“山大王”，但他们走得进井岗山，也出得了井岗山。从进去到出来，他们自始至终都是“有主义”与“闹

^① 引自《朱德选集》，文《在编写红一军团史座谈会上的讲话》（1944）。

^② 引自《毛泽东文集》，文《论抗日游击战争的基本战术——袭击》（1938）。

革命”的，这就是他们与历史上众多的山大王不同的地方。

纵观整个反围剿与抗战，共产党一直有战，一直有胜，也就一直有着最终胜利的希望。而你坐镇一方山头，到底要给团队什么，才能让团队跟着你走，或按着你的路线走？看来做个山大王也不是容易的事。这在本质上是对领导角色的要求，即方向明确，制度有效，知道怎么做，知道做什么。

6.8 最后：别让兵掉到水里

有合法的山大王却仍然“群龙无首”，这就是领导职缺位的结果。有了“职”而不确定其“务”，不确定其“权”，不确定其所谋之“利”，这个领导的所有作为都将失去支撑。反过来，势者，利之所趋也！我们明确地知道：

由个体利益、团队利益与项目利益所构成的大势是项目的整个背景。

而一个有团队价值、为项目负责的领导者，其核心在于：如何正确认识与统一这些利益？

“合法的山大王”们，无论是因为何种原因得了山头，也无论他们在具体的行事当中是如何地“有方法与有政策”，若失了行事的方向（闹革命）与成事的背景（有主义），那便也只不过是闹腾一时的山大王。C 公司在这次变革中的策略不见得不好，但这一策略的两个主要背景，亦即形势，亦即基于形势的主要判断——**行事的方向与成事的背景**都并不清晰。前者，亦即是**行事的方向**的不明确，导致产品线划了等于没划，原来试图通过产品线来归拢并进一步削减产品，却因为方向不明确而“不知道怎么划”，削减也就无从谈起；进一步地，产品线的做法与产品的做法没有不同，新的流程与方法

又受限于旧的体制与流程，体与制互成掣肘；再进一步，各个产品线又“貌似”获得了授权，故而时时有“自作的主张”而无视（也无法视见到）全局的利益。后者，亦即**成事的背景**的不明朗，使得“形成新模式”这一构想变成了赌博，寄期望于十几个山头的相互竞争中会有异军突起。

果然，这些产品线折腾了一年之后，终于有一条产品线“似乎”杀出了血路。而当大家纷纷看齐，准备学学他们的“方法与制度”时，这条产品线却一夜之间崩溃了：在业绩上存在着不道德的竞争行为。大家这才发现，这些山头事实上并没有出现过真正的“产品线”，那些貌似“产品线”和“新模式”的结果正是虚假繁荣的一种。这才恍然大悟，缺乏一个真正的领导者，缺乏领导职能中最重要的“方向设定”与“背景判断”，团队的成功多只是暂时的现象，或某些特殊因素导致的表象。

如同我们开始讲到的，当团队处于这种困境中的时候，才是彰显一个企业的组织力的时机。若“找到合格的山大王”是关键点与基本条件，那么这场变革在三个月内就可以结束了；然而在山头立起来了之后，却不太可能三个月就削平。这样的兵行险着，是在我们无法对组织背景进行精确判读时最选无可选的策略。

真要这样做，也总得有人守在岸边看着，别让兵掉到水里。

第 7 章 自己想办法

7.1 三块贝壳

如果说马桶“是 20 世纪最伟大的发明”，那么粪坑便可以说是整个人类从愚昧走向文明的标志了。动物除了少数的几种会掩埋粪便的之外，实在没有谁会像人类这样在地上挖个坑来解决腹中污秽了。上面飞的鸟类会把人类的头顶当做粪池，下面跑的兽类会在地上设下“陷阱”，水中的鱼类则更简单、直接、正大光明地污染着水源。

所以人类学会了挖粪坑。自从有了粪坑这个伟大的事物，原始人类才开始摆脱通过气味来寻找居所的“本能行为”，从而深度刺激了脑部记忆功能的进化，脑容量随之增加。进而人们开始了共同利用粪坑的群居生活，有了原始部落，以及群体、社会等组织形态。

粪坑对于原始人的重要性远比今天的马桶重要。马桶隔离了人们之间的交往，各家专用的马桶使得人们的沟通日益艰难，甚至成为奢望。而远古的人们与马桶时代之前的人们一样，可以在两个粪坑之间，在一道小小的、类似于篱笆的障碍物之间语言交流、学习和沟通，甚至可以互通有无，解决类似于没有厕纸这样的尴尬小事。

说到厕纸，这也是另一个令我充满敬畏的发明。因为自从看过《超级战警》^①这部科幻电影之后，我就一直被一个问题深深地困扰：如果没有了厕纸，该怎么办？在这部科幻片中，到 2032 年，那时的厕所里就不再有厕纸了。

^① 英文片名 *Demolition Man*，马考·布莱姆比拉导演，西尔维斯特·史泰龙主演，1993 年上映。

在相同的位置上，放着三块贝壳。

原始人可能从粪坑上传递木棍这样的事件中，理解了互助与友谊。而马桶则逐渐地加深了人与人之间的防范与私隐。马桶与厕纸带来了文明的如厕方法，而我们却再也无法让自己回归原始：贝壳与木棍，只是材质的不同。

在 2032 年的未来，什么才是可行的如厕方法？

7.2 佛祖的“过程论”

古人是很讲程序、方法与规则的，这其中就包括如厕这件事。而且，这件事的程序性相当重要，器具制式和用法也有一定之规，以至于需要佛祖释迦牟尼来详加解释。这件事情记在《毗尼母经》第六卷里，要求如厕者常自备“厕筹（木片）”；如果没有，那也不能拭在墙上、厕板上，也不能用石头、青草、土块软木之类代替；而应该现找木、竹、苇，临时用作厕筹。不单单如此，佛祖还解释了厕筹的长短制式，以及用完了之后不能用弹、振、甩这样的法子来弄干净，也不能跟干净的厕筹混放在一起。这个，便是佛祖他老人家传下来的“上厕用厕筹法”。

佛经原文大抵便是如此。所以那个时代的僧侣们，大概是要随身备厕筹一枚的。倘若一时忘掉，那便只能折了木、竹、苇来替用。再如果这些东西都没有，可能就必须蹲在原地，等着其他人来，或是借用，或是求人家去取。但是，顺着这个思路想下去，我便发现：如果僧侣在四野无人的地方遇上了这个问题，既没有人又没有替代物，那么岂不是要活活地“蹲到死”？

几千年了，从没听说过有佛教徒这样蹲到死的。即便这个范围放得

宽大一些，我也没听说过有哪个人是会这样蹲到死的。

没有人会机械地遵循某种“如厕方法”所约定的过程，一旦有“标准方法”所未顾及的、不太周全的情况发生，我们总是自己想办法，去把这些问题解决掉。我从来没听过会有人自己提着裤子出来，备好了厕筹再蹲回去完成程序的。

而在我们的工程中，“过程论”的专家总会要求你这么做。同样的，这样的专家在面临三块贝壳时，宁可触墙而死也不会上厕所，因为没有任何可用的过程、方法与工具。

7.3 将“最佳实践”工程化，本质上去掉了它“立可时用”的特性

佛祖其实是个好人，他老人家介绍自己的如厕最佳实践，包括厕筹最短也不要短过四指，要不用着不方便；又包括不能把染着污秽的厕筹甩来甩去，要不沾到别人，起了纠纷，打得头破血流；还包括用过的厕筹不应当乱放，要不传染疾病或是影响别人使用.....这些方法的确是很好的，体现了丰富的如厕经验，并总结得条理清楚，可行性相当强。

但无论如何，你得先有厕筹。因此工具论也就大行其道，以至于南唐后主——就是写“春花秋月何时了”的那位——也要亲自去削竹片木片来做，并且还要放在脸上试用，若有锐利不适的地方再加以修正。所以厕筹这个东西，也是可以做得相当考究的，例如在外面拿香囊之类来包裹，既款款有型又不失雅趣，这样的事情在东晋也是有的。

先有了方法，而后其细节的步骤被规则化了，就变成了过程论的论

调；其细节所依赖的那些东西，就构成了工具论的基础。如此一来，“过程+方法+工具”，就构成了我们将一件事工程化的全部。

当一件事情的做法被工程化之后，事就不再是具体的事了，成了一类工程。或是统一工程，或是敏捷工程，总之都失去了“具体”二字。

不再具体，也不再立时可用，也不再（仅仅）是一个法子。

7.4 先拉完屎再说

我们原始的目的仅仅是“方便”而已，最终却纠结于“怎么擦干净”。

还是先拉完屎再说吧。

这件事得自己努力。

7.5 再谈《人月神话》

软件工程原本是只讨论“过程+方法+工具”的，这是纯粹工程学的视角^①。这其中即使有一些看起来与人相关的话题，也不会被过深地讨论到。例如，过程中显然涉及人，但不会因此而讨论到“过程中的产品环节是否可以由开发人员来兼任”这样的问题。事实上，在“纯粹的”软件工程的讨论中，过程的各个环节都是由一些“角色”来推动的，这些角色如何存在于一个具体组织的部门中，其职业定位、能力结构等都并不算一个严格意义上的工程话题。

^① 事实上，还会讨论到“目标”问题。但在传统工程中的“工程目标”并不是确指，而是使用“需求”来指代的一系列技术方法。传统工程基于“需求”而非基于“具体目标”，并且——与此相关的——它与架构的“面向问题”也是互相背离的。因此传统工程中的工程师视角下的思考，对“架构师”这一角色在团队中的融入帮助并不大。

但是这样的“纯粹”显然遇到了挑战。当我们试图将工程完全地理论化之后，它就必然面临实践的困境。“人”的问题在这方面首当其冲。

《人月神话》并不是一个“人/月”的度量理论那样简单——事实上那本书很少讨论工程度量的问题。《人月神话》在工程上的重要性在于，它严肃地提出了“人，是不是工程学问题的一部分”这样的话题。这本书的答案将这个话题引向了一个极其巨大的迷局，即如果需要更多的人来实现工程，则应付由人带来的复杂性可能将超过工程本身的技术复杂性。《人月神话》一方面有先见地对技术复杂性提出了一些可靠、可行的方案，另一方面也悲观地认为由人带来的复杂性必然导致“巴比伦塔”最终的倒掉。“没有银弹”的论证过程将所有的焦点集中于：通过（较小规模的）程序实现的过程，无助于求解（包含大规模工程在内的、普遍含义的）根本任务。

作者 Brooks 把三个问题从《人月神话》的讨论中摒除了出去。其一，项目必是有始终的；其二，哪怕是一个不成功的产品，也是需要交付的；其三，团队是人的问题，并不是事的问题。《人月神话》将这三个问题的前题隐设为：项目需要承担大量的附加责任，产品必须是尽善尽美的，以及人是私利的。而这正体现了“工程学”的学术本质，即如果我们可以的话，应当基于工程来解决一切问题；因而它必须首先是能解决一切问题的。

在这些年的实践中，这是我所见的、对“一件事”具有最大伤害的但又貌似义正辞严的、跨越行业与组织的普遍观点了。我之所以用这样复杂的定语来说明这一“普遍观点”，是因为它确实那样“自然而然”地存在着，就像我们每个行业都受了同一个神灵的启迪一般。事实正是如此。任何一个有道德、正义而又富有职业责任感的产品经理都会跳出来，说“这个产品没做好是我们的责任”，然后历数种种设计细节，再加以细化，并立时加入你的需求库；任何一个有同样品

质的项目经理也都会跳出来，说“这个项目没做好是因为我们对流程的贯彻不够”，然后开始制定更详细的管理措施；然后开发人员也跳出来，说出超过 200 个的技术改进点.....

但每个人都将问题揽在自己头上，这难道不是一种好的品质吗？如果是，那么我又怎么会用一种调侃的语气来说出上面的假设？

这样的假设，在于警示我们一个被忘掉的事实：这是一个系统，并不在于一人一己或者一个角色的过失，也不取决于某一个角色的单方面的努力。这些“积极、冲动而又充满道德责任感”的人，事实上是有助于找到系统问题的答案的。我们总是可以现实中找到许许多多具有这种性质的工程、产品与过程模型，并且在这样的模型中，（提出模型者所代表的）中心角色也总是认为自己应当能够并且也必须能够去承担更多——因而也就更重要，因而其他所有角色都应该围着自己打转转。

然而这样一来，每一个角色都站在一个自己认为“更合适”的角度来看我们在做的事情，而全然不管那件事情本身为何。例如，开发人员可能认为自己的某项技术发明有着相当“巨大”的前景，于是期望着有一个团队来配合自己将它实现为产品。至少在他看来，由技术推动产品和市场是相当伟大的成就。但是，一旦他提的这个想法如同扔进了泥坑一样丝毫得不到反馈，那么他又立即开始着手“发明”下一个技术与想法了。

你会发现，他原来根本不需要对产品和市场负上任何责任的！他只管提出，而不需要负责任，那么又怎么可能站在别的角色上去思考问题呢？

所以我再读《软件工程》这本教材的时候^①，就只会看到一些“管理的技术”了。然而我从不指望，把“管理”当成技术手段和工程方法的人能真正地做好项目。

我的意思是那个具体的、眼前的项目。

7.6 敏捷工程的成功之处

如上所讨论的，在发现“人的问题”是一个关键因素之后，传统工程就将它作为一个学术话题，用一些角色、规格与方法来限制这些人的行为，试图将管理变成一个“技术所需的条件”以及“可以基于技术手段来改善的群体工件”——“人件”这个名词其实就翻译得很好。

《人件》本质上也是基于这样的立论来讨论工程话题的一本书。简单地说就是：总有那么一些方法可以让管理成为工程的一部分。

但是敏捷工程却在“人的问题”上探索另一个“解”。关于敏捷工程，我从来都是把它与传统工程“对立”起来讨论的。但在根本事实上，它们是不对立的——我的意思是说，它与传统工程在“关注人的问题”这一点上并不对立，但在二者的求解方法上却大相径庭。

敏捷工程直接将问题指向“人的私利”与“合作”之间的矛盾，即人们“究竟是为什么要合作”的。敏捷工程无比强调团队内部的合作，并将这一合作设定为一种“道德需要”。由于有了这个前提，不合作者将被套上道德枷锁，或被拷问，或被驱逐。总而言之，敏捷团队必将因此变成一个（尽可能地）倾向合作的团队。因为这一核心前设

^① 早些时候的《软件工程》教材是不讲团队、管理等内容的，但后来它们都逐渐地加入了相应内容。但老实说，这些书——例如《软件工程——实践者的研究方法》与《软件工程（Ian Sommerville 著）》等——中有关于人的讨论都是隔靴搔痒、无济于事的。这些经典的、学术化的工程方法仍是以自己为中心的，它们试图把管理学、组织学或社会学“拿来一用”，其思想离“看到一个项目自身的系统性”还相当之远。

的不同，在四条敏捷宣言中就有两条涉及合作（另外两条则是针对产品定义的），而在十二条敏捷原则中则有超过半数都是在强调团队成员的道德认可，而非强调某种技术方法。

从这个角度上来说，敏捷工程的成功之处在于形成益于合作的团队氛围，以及在道德与行为准则上实现人们的“强制合作”。

7.7 拉屎就算是一项工程，具体来说，也得自己拉

无论敏捷工程还是传统工程，都是无法为你正在做的这件事提供直接答案的。

这件事得自己努力。

一个具体化的工程涉及三个方面的问题：人与团队，产品与用户，项目与方向。如果这些东西要被“系统化地”组织在一起，必然是需要将它视为“一件事”的，这也是它被“立项”的目的。如果这一前提不存在，则产品只是用户调研、产品设计或产品线上的一个规划，而团队则可以简单地视作“一群人”。只有要做一件事，这三个方面的东西才发生了关系，也才交织在一起成了“问题”。

要开始做“这件事”，会有不少的前提条件。例如，作为一个项目经理，你或许要求“先做三个月技术培训，再开始开发”。但事实上这些都算不上“前提”，因为你显然也可以不这么做就匆匆上马——只要你有机会在团队行进中调整大家的步伐。唯一能作为“开始这件事”的充要条件的，事实上只有一个决策性的判断：我们是否需要用工程化的方法来实现一个产品目标。这里涉及下面三点细节。

(1) 是软件产品吗？如果这不是一个需要“软件生产”的产品，则它可能不需要“软件开发”这一过程，固而也就不需要我们的软件工程。

例如，市场部门需要了解客户反馈，那么它既可以实现为一个在线聊天的客服机器人，也可以实现为包含大量职员客服部门。而后者显然是不需要“软件生产”的。

(2) 需要开发吗？即使是一个软件产品，获得它的方法也可能不是“软件开发方法”。例如，上述的“在线聊天客服机器人”，如果我们将它作为一个产品外包，那么“外包”这件事对于我们来说就是一个供需管理，而不是“工程化”的软件开发活动。

(3) 需要工程化方法吗？一个研发性软件产品的开发过程，可能不是“工程化”的。例如，我们真的要在公司内立项来做上面这个产品，但我们只分配了一个开发人员或一个临时小组，产出的产品也不用于市场销售，对时间的控制也可以放得很宽轻，而且最为重要的事情是，假定这个项目是基于技术研究的，因而当它失败了也不会有什么负担，那么我们便不需要考虑“是不是需要一个工程化的软件开发方法”。

当决策判断发生了，当“应作为一个工程化的软件产品来予以立项”成为事实的时候^①，产品、团队与项目三者就必须立即成为“具有相同目标”的一个合作群体。这个相同目标，就是“做一件事”；这个相同目标中的道德原则也只有一个，就是“做完”。

我知道有一个项目，一个做了整整五年的项目。在项目坚持了这么久之后，是什么在支持着这个团队还保持着追求成功的激情？就这个问题，一个普通成员的回答典型、普遍而又发人深省：

^① 事实上三种情况的反例都是可以“立项”的，譬如组织建设或人力规划的项目，企业资产购置与管理的项目，实验性项目。它们作为项目是满足“时间、质量与成本”这样的项目要素的设定的，但却都不是工程化的。

“在我们的头脑里，完成一个项目是最最重要的事情。”一位成员说：“即使我们不喜欢这里，我们也不会让项目流产，我们宁愿先完成项目，然后在第二天辞职。”

这个团队就是 **Windows NT 4.0** 的开发团队。在读《观止——微软创建 NT 和未来的夺命狂奔》这本书的过程中，这名普通的甚至没有在这本书中留下名字的成员，说出了让我如此印象深刻的话。这样的普通成员，正是这个团队整体的职业精神的缩影。

回顾这本《观止——微软创建 NT 和未来的夺命狂奔》，我们是无法孤立地站到团队、项目与产品三个视角之任一来思考的。从根本上，具体工程最重要的话题就是这三个视角必须是统一的、系统的以及面向具体的“一件事”的。

篇三：具体而微，工程是系统而不是事

我看到的一种可能的答案，就是“组织”。它远远地在 EHM 图的最外层——那个层面约束了工程、实施的形式，并适时调整着这种形式；那个层面看得到工程的目标方向，也适时地调整着这种目标方向——真正决定着工程生死的，在工程之外。

当然，工程本体的病患是有的，例如“牛屎图”中的工具与方法不匹配，或者质量模型中的时间与质量起冲突等。这些病患并不会因组织的调适而消亡，但是组织的调适能给出治疗这些病患所需的时间、空间和基础。所有的病患其实都是长在组织这个“体”之内外的，当组织环侧与开放自身，毛病就显露了出来。这如同人的屁股上长了疮，总是要抬抬屁股才能抹得上药的。

从这一点来看，“从局部看来，向全局看去”，既是我这些年来工程实践的所为，也是我将来必行的方向。就软件开发而言，跳出具体角色的限制，看到“一个组织”和“工程的实施”的全局，也许正是破局的机要？

第8章 做事的选择

8.1 开诚布公地谈“权”

大多数讨论组织及其管理的书籍总把一个问题谈不好，这就是“权”。中国人善权畏权，而又避免谈权，这是一种普遍的处世哲学。然而因此绕开了这个字去谈组织与管理，总是戳不到痛处！

此前我们绕了很大的圈子来谈“因权而谋利”，本质上就是指出了“权”的一种出处，即由于它代表了一部分人的利益，因此得到了支持，而这种支持就变成了“权”。例如，共产党代表全民利益，其出处就在于毛泽东等老一辈革命家对“天下穷人要闹革命”的时局判断，这显然不是出自某种权力机构的授予。而“合法的山大王”则是第二种权的出处，即授权。尽管两者在由来上不同，但都表现为一种“权”。

C公司中的“合法的山大王”最终没有取得成功，从根本上来说，是获得授权的管理者并没有做好准备。管理者并没有意识到，在组织变化时，最大的、最本质的变化产生于“权”的变更；或者，大家已经意识到了，却也并没有做好“有权”之后加以正确运用的准备。

组织活动的核心，在于权；权的核心，在于利。一部分人越来越怕谈权，是因为“利益”已经被越来越狭隘地理解为“钱”，因而谈谋利就变成了谈图钱。而权的责任跟钱一旦发生关系，就越来越说不出口了。

权之责，在于谋利。开公司总是要挣钱的，做一个挣钱的公司没有什么不好，但公司的利益却并不仅仅是钱。谈钱有谈钱的部门，例

如销售部门如果也不谈钱，反倒不正常了。在整个公司的组织活动中，公司所追求的种种利益，都必将通过组织授权行为来明确。

这就是权、权责与权利的关系。

8.2 没有“凭什么”

很多时候追问“凭什么”是一件相当无意义的事情。例如一个新员工可能会对部门领导充满置疑：他何德何能，凭什么来管我们？在这个新员工眼里，领导几乎在方方面面都差得很远：或比自己差，或比某些他觉得更尊崇的人差。这种追问显出理性的光辉，仿佛只要某人“有德有能”就应当居其位、行其权，似乎这才是公平合理而不受置疑的。

事实上他忽略了组织形成的代价，以及领导角色的成本。当把这些因素考虑进去之后，“有德有能”便居之，其实才是真正的不公平。举例来说，你可知那个“看起来平庸的人”事实上经历了多少轮组织变更与重建才留到现在？他在那些重大的利益选择面前，多少次坚定地组织站在了一边？他又是在多少次外界诱惑面前，与组织不离不弃？事实上正是对“权、利、责”的清醒认识，才使得他现在站在你的面前，成为看起来能力平平而又是一个“事实上的领导者”。

即便我们可以无视这些“看似是术”的东西，那么“机会与机遇”可能是唯一能解释大多数组织现象的理由。这主要是指，组织在早期形成时存有相当多的机会，例如职务空缺就更多；而在组织成型之后，这样的机会也就渐渐趋少了。然而参与组织早期建设的那些人，比如创业团队，他们承担了多少风险与责任，面临了多少困难与冲突，又岂是这些后来者可以想见的呢？

所以这个新员工要追问“凭什么”是既不公平也不合理的。但反过来说，组织内部如何解决权力与能力的问题，也是一个相当关键的话题。大多数情况下，“何德何能”也是组织内部任免时会提出的一个关键问题。但我需要说明的是，这是组织行为。组织行为是基于“组织的核心利益”而提出“何德何能”的问题。若无法看清这些利益，就无法判断组织行为；若无法左右这些利益，也就无法左右组织行为。这也是我说上面的“追问”无意义的原因：我们大多数的时候是在利益集团的外面，而非里面。

8.3 修身的智慧与做事的素质

不过，你仍然可以有选择。

蔡学镛先生是我的前同事兼好友，他曾在博客中写道：

公司为我们做绩效考核的同时，我们也要为公司做绩效考核。要考核公司的哪些地方呢？薪资、股票、学习成长的机会、地位、成就感、稳定.....这些都是评分点。你可以根据自己对这些条件的重视程度，做权重的调整，得到一个绩效值。比较各家公司的绩效值，做出去留决定。

如果你打算留在公司内，你也必须对你的主管做绩效考核。他是不是能充分发挥你的特长？他是不是答应给你的资源都有做到？他是不是有给你适当的奖励？他是不是赏罚分明？他是不是有在培养你？.....如果你给这位主管打的绩效不好，你可以考虑换部门，或建议公司换主管。

所以你一旦选择了“在其位”，则应当抱有对“从整个公司到具体领导”、对整体组织系统的认可。这种情况下对于“凭什么”的追问，表面看起来是对别人的腹诽，实际上是对你自己选择的怀疑。你或是未能清醒地做出最初选择，或是难于坚定不移地支持这个原始选择，前

者是谓“盲”，后者是谓“浮”。

首先要能责己、省身，这是**修身的智慧**。有了这种心态，我们再来谈谈这种选择背后的东西。回顾《大道至简》这本书，它在谈“画虎类狗”时就讲到过马援的家书。其实关于马援，我们还可以讲到他更为著名的一些故事。

西汉自刘邦始，历时两百多年之后被王莽夺了政权。而马援刚刚被王莽任为新成大尹（今陕西安康、汉中一带的太守）不久，王莽就被起义军杀了。一时天下大乱，四方兵起，于是马援就跑到凉州避难去了。这时地方军阀隗嚣占据了天水这个地方，却没有称王，而自封西州大将军。当时，有公孙述在四川称帝，成都由之得名“白帝城”；又有刘秀在洛阳立了东汉朝，史称汉光武帝。于是在归附哪个朝廷的问题上，隗嚣犹豫不决，遂求计于马援。马援详查细思，认为公孙述功业未成却先已奢靡，不是王者；而刘秀恢廓大度有汉高祖刘邦的气象。因此他建议隗嚣归附刘秀，并随后随众前往洛阳，臣于刘秀。然而过了几年，隗嚣又有了反叛的意思，而马援则坚持自己对刘秀的看法，认为隗嚣想要称王是自挟奸心，遂帮助刘秀灭了隗嚣。此后，马援在刘秀一朝中西伐南征，战功无数。他在后世被追封为忠成侯，“马革裹尸”即是语出于马援。

从王莽、隗嚣、公孙述，再到刘秀，在得到刘秀起用之前，马援一直在选择，既有被动的，也有主动的。人的禀性不同，因此或建功，或立业，总是有着志向的不同。如同马援一生择主而事、守疆拓土，成就一代名将，这就是做事的人。

而在我们的现实中，往往一些人一时不得其志，或一时有所触动便谈“创业”。在遇到这样的人和这样的机会时，我们需要慎重选择。对于这样的人，小事尚且不能做好便谈大业宏图，不足与谋。如果

不能与他“共其事”，又如何能与他“同其谋”呢？对于这样的机会，即便你身怀成事之大才，也应当先择业而事；即便你胸罗大志，欲成就一番伟业，也需知“要学会用人，先学会被用”这个简单的道理。所以我们要先谈事、后谈业，这才是行在实处；若是反过来，就是唱高调了。

反思马援的几次选择，有两处是值得回味与借鉴的。其一在于归附刘秀，是有眼光；其二在于平叛隗嚣，是会坚持。归结起来，明辨、坚持，便是从他身上可见的**做事的素质**。

8.4 你可能只是建功者，而非创业者

马援志在建功，而不在立业。

但什么是“业”呢？现在一谈“创业”就是开公司，仿佛只要有个公司的名字，只要有个总经理的名头，就是创了业。归根结底，这还是在“什么是利”这个人生思考上随了大流的结果。“什么是利”是你对人生价值取向的发问，你追求什么，想要什么，然后想要通过什么方式来得到，这个思维过程决定了你的价值观与事业观。举例来说，你追求经济上的满足，想要更多的财富，并且认为得到它的方式就是开个挣钱的公司。这样一来，你创业的目标和方向就基本成形了。

我的确是在用一种粗浅的方式谈论“创业”、“财富”与“价值观”等问题。但这也是大多数人具有的一种思维方式，是普遍存在的，而并非少数的、精英化的思维、道德、价值与人生取向。我这样谈“业”的目的是要将这个话题赤裸裸地抛出来，并且强调：这没有什么不好，没有什么不正常，也没有什么不高尚。这是可行的、善的、正确的选择。

但这不是唯一的选择。“创业”其实是你的的人生图谋，“一生所事者，为业”。因此，“求幸福”可以是你一生的事业，“图开心”也可以是你一生的事业，“谋财富”也可以，“得安心”也可以.....历史上种种开“创业公司”的人物不少，例如张良开的“刺秦公司”，本以为杀了秦始皇天下就太平了；又例如刘邦开的“灭秦公司”，认为灭了秦朝，创建新的制度才可以给天下人带来幸福。当张良发现“刺秦”行不通时，就转而投入了“灭秦公司”。而我们在这一章谈到的刘秀的“兴汉公司”，就是觉得天下要太平，就得四方安定，所以马援便有了施展才能的空间。

在**创业**这个问题上，你的选择多多，例如张良就创过业，当然也失败过。但若是单单谈**建功**，要么出自机遇，要么就出自能力。前者我们暂且不谈。而后者，以能力论，需知有能力建功的人并不一定能创业，比如张良。建功者，既需要身具大能，又必须全力以赴。能与力，前者是内底里的素养，后者是外在上的施为。

这就是我们谈“事”的基础了。也就是说，你得有做事的才能，并且有做事的作为。马援两者都有，因而终成大器。以能而言，他首创了军事上的沙盘推演，是历史上最有名的伏波将军^①；以力而言，他一生征战，最后病死军中。

8.5 马援之死，以及死透了之后

马援之可叹处，在于他虽然成就了“马革裹尸还”的壮志，却也几乎落下了“死无葬身之所”的凄凉。前面说他“后世被追封为忠成侯”，讲的便是这个。

^① “伏波将军”是一个名号，意思是“降伏波涛”，因此历史上有很多将领被任命或被称为“伏波将军”。

晚年，马援效廉颇请战，主动请兵去征讨武陵（属荆州，今长沙）的蛮夷。稍有失利时，刘秀便委派了素与马援不和的梁松去问责，并代为监管。但是梁松到部队的时候，恰好马援刚刚病死。梁松旧恨难消，便乘机诬陷马援。刘秀因此免了马援的侯爵爵位。马援的尸体运回，家人都不敢葬在原来的坟地，只草草埋了。然后家人用草索绑了自己，到朝廷请罪。刘秀拿出梁松的奏章给他们看，马援的家人这才知道蒙受了天大的冤枉。马援夫人知道事情原委后，先后六次向皇帝上书，申诉冤情，言辞凄切。光武帝这才命令安葬马援。

刘秀是一时失察？未必。但这件事就这样搁下了。过了一朝，到了汉明帝刘庄时期。这时刘庄评定开国功臣，但选出的“云台二十八将”中也没有马援。当大臣惊疑时，刘庄却只是笑而不答。难道又是刘庄一时失察？未必未必。

有趣的是，刘庄的皇后却正是马援的女儿。当年，刘秀一方面不再深究马援案的细节，另一方面又作主把马援的女儿聘为太子妃。后来，当刘庄为“云台二十八将”画像时，也正是他册封马援的女儿为皇后之时。每每两事并举，用意十分微妙。于是，这事情又再过了一朝。到了公元75年，刘庄死后，刘烜^①即位。在他的支持下，大权在握的马皇后才终于为自己的父亲平反昭雪，追封为忠成侯。

汉朝的天下，究底里还是个“家天下”。所以事有亲疏，“家里”与“家外”的事是分别论的。梁松此人，其实是刘秀的女婿，因而梁松品行好坏，是否构陷马援可以暂不细论，此其一；他既是家里人，便是可以用来平衡日渐坐大的马援在军中的势力，此其二。马皇后在刘

^① 音“达”。

庄一朝中也不谈为父亲恢复名誉的事情，因为她深知这权力若失了平衡，就容易让刘庄心疑。再等到马皇后可得权柄，且军中再无马援旧部势力之时，这名誉恢复也就恢复了，再也无妨大体。

马援的得失往深底里谈，就成了一些人的权力政治。就好像我们再往深底里谈“凭什么”的问题，就成了办公室政治一样，这是我极力想回避的一件事。因此虽然我们谈到了“权”，但我却要就此打住，把权的问题放在一个组织的视角上去，而把接下来的问题简单地归为“职”。于是可能就有人说了，权职权职，不都是一个东西吗？

其实不尽然。权谋的是利，职谋的是事，所以权有权益，职的利益却最多只谈到薪酬。

8.6 谋事、成事，以及忠其事

我们大多数人“求一职”，无非是“求一事”，求个做事的环境，有个做事的目标。但这个目标并不见得是公司的目标。开公司的目的是谋利，这是创业的基本思维：如果不谋利，你找一堆人来做什么？国家大事与私企小事，其区别只在于所谋的“利”的不同，谋“谁的利”也有差异；由此，进而有权，更进而有责。这些是组织形成的普遍逻辑。然而“职的利益”却非常明确，他只是一个人求生存的基本方法。

很多组织会将“组织利益”加于成员的“职责”，这是站在组织者视角上的愚民之术。我们谈到过，传统软件工程将根本任务与次要任务放反了，因为它将工程的根本任务设定为“抽象软件构成的复杂概念结构”，这终究只是学术的观点；而将它反过来，我们就会看到，“完成具体的一个工程目标”才是我们真实面对的问题。与此相同，组织者说大家的“职责”就是与公司一体，爱公司就像爱家，爱客户就像爱自己，这也不过是一种“教给你的”的追求；如果也将它反过来看，

你就会明白，你的职责究底里不过是“做好手边的事”，以此作为自己讨生活的资本。

你做这件事的目的，首先是私人的，其次才是为公司的；而公司所希望的，正好是你做每件事都是为公司的，其次才是为自己的。我之所以一再提及这个话题，是因为公司的利益原本另有出处，公司与你谈利益，所站的都首先是公司的视角，而非你的视角。因此公司将你的职责，也预设成“为公司的利益”就是可以想见的事情了。

那么，你还为不为公司谋利呢？

职的业，在于谋事；职的责，在于成事；职的道德，在于忠其事。这才是职业的视角。你的“利”在于你决定“做这件事”时的选择（例如薪酬，又例如人生理想），这是首要目标；若因这件事而成就了公司的利益，却是次要目标，或者我常常称之为附加收益。

但我仍然必须强调，这一问题在创业与守业阶段是略有不同的。你如果加入一个创业团队，你加入时所设定、所承诺的这件事的目标便是创业，是将自己的利益与“所创之业的利益”放在一起的。若非如此，你便仍然称不上创业，只不过是打着创业的名头，求着自己的私利罢了。创业与守业之不同，就如同攻城与守城的不同。你看攻城，所有的人都朝着一个目标，撞着城门，爬着城墙垛子，前仆后继总为着相同的目标：进城去，就什么都有了。而你再看守城，各自守着各自的城门，管垛口的只管垛口，管马棚的只管马棚，每个人都看着自己的职责；若管马棚的不管马棚，而跑去守垛口，马群散了在城里，撒开花儿地跑，既乱了军心也挡了他人的事。于是管马棚的就被砍了头，失了职也失了命。

守城之重，在于各司其职，有条不紊。尽管头儿会告诉你说，丢了

城我们大家都玩完，但是你也知道：人家打进来不一定会杀了你，但如果你现在失了职去守垛口，那就可能立即被正了军法。所以从守城之职的视角来看问题，无论于私于公，其最优解是看着垛口一个个失掉，又或者被夺回来，而自己仍只管好自己的马棚。所以管好马棚是你的首要目标，如果这促成了守城的胜利，那是附加的收益。

再说“忠其事”。其实，在我看来，这已然是非常高的道德标准了。

西周时期，齐国的崔杼杀死了国君，于是齐国的史官就写道：“崔杼弑其君。”这是说崔杼不忠，以下犯上地杀了他的君主。于是崔杼不满，就把这个史官给杀了。钱穆先生曾在这里点评说^①：那时周王室派出很多史官，他们虽在各（诸侯）国，而其身份则仍属王室，不属诸侯。因此，崔杼可以把这个史官杀了，但不能以齐国的名义另派一人来做。于是史官的两个弟弟便接任兄长的职务，再来照写“崔杼弑其君”，崔杼再把他俩杀了。又再换第三个弟弟，还是写成这样，崔杼没有办法了，只好住手，不杀了。

当时的史官受命于周天子，在齐国称为“大史”，而在齐国以南的其他诸侯国，则泛称为“南史”。所以当时有“南史氏”听说齐国的大史都因此死光了，于是便自备了录史用的竹简，捧笔前往，以志赴死。

因为受命于“录史以存真”的职务，便舍了性命也不改一字，便舍了性命也要前往记录史实，这便是我所说的“忠其事”。

^① 引自《中国史学名著》，钱穆著。

8.7 组织：在形式上的、团队的构成

你需要一个做事的团队，但事实上这非常难得。

我们这里讨论的团队，并不是公司（或我们统一地称为“组织”）中的一个部门，至少并不简单地是这样一个组织结点。组织中部门的规划取决于许多因素，有些时候与企业的历史背景有关，例如国企或事业单位下属的一个软件公司；有些时候与企业领导者的管理风格有关，例如某些采用家长式管理的私营企业；有些时候又与企业正在经历的变革有关，例如业务转型阶段中的企业。在这些种种不同的背景或时间点中，一个“具体做事的”团队有可能保证它的相对独立吗？

答案是：几乎不可能。也就是说，在不同的组织结构、风格与背景下，我们事实上难于得到具有相同或相似性质的“做事的团队”。但是出于两个目的，我们不得不假定：

所谓“团队”，是具有相对独立的组织特点，有着自我目标和道德环境的一个特殊群体。

这两个目的是：其一，我们在讨论中需要一个相对纯粹的模型；其二，构建新的团队，可以作为“立一个项”的普遍条件，即我们有望针对项目来构建团队，而在一定程度上忽略既有组织的影响。

事实上现实中的大多数公司都已经踏出了这关键的一步。大家似乎都已经意识到，没有哪家公司一开始就是围绕“工程化的软件开发”来构建组织的，也没有哪家公司是“先建设好组织模型”再来生产与经营的。在现实中，大家都是边干、边扩张、边调整，组织是在动态发展的过程中渐渐感到“工程化”的必要性。但这种必要性又与现有的组织形态存有一定的冲突，例如：

- 其一，敏捷工程通常要求团队的规模较小，但现有组织中的任何一个部门都可能有上百人；
- 其二，如果以“小组”为单位来应对敏捷工程，就会发现小组中存在许多的角色缺失，例如测试角色可能无法由直接的团队成员来担任，因为这些专职人员可能归属于公司内的另一个独立部门；
- 其三，当一个工程团队所需的资源零散地分布在公司的既有组织中时，“团队是什么”就成了一个非常实际的问题，而更实际的问题可能还包括“团队到底有哪些责权”。

所以上述的假定是有必要的，这有助于通过组织授权来建立“具有明确责任的团队”。基于这样的假定，可以大略地将“围绕项目目标而组织起来的一群人”归纳为三种模型：产品线团队，对应于企业业务；产品团队，对应于客户产品^①；创新团队，对应于自发需求。

三种模型的根本相同点在于：一群人，一个目标，一种方法。^②

^① 这里的“客户产品”与前一种的“业务”是不同的。这里用“产品”来表明“有明确的特性内容的需求”。所谓“客户”，既可以是以企业自身为目标，也可以是以特定用户或用户群为目标，或类似外包中的出包方等。

^② 这让我想起《集装箱改变世界》中提到的著名的“3C 原则”：一只集装箱（container），一个客户（customer），一种商品（commodity）。

第9章 你要什么

9.1 长期的组织战略

我一直有一个问题：既然是“前人栽树、后人乘凉”，那么前人该不该栽树呢？

栽树事实上是一个布局的事情。有两种方法来完成这个布局：第一种是直接移植一批树来；第二种则是新栽一批树。这两个布局的结果可以是完全相同的：一片可以乘凉的树。但是，两者在本质与操作手法上都存着极大的差距。

移植的方法是管理层在组织问题上最愿意采用的，但考察收益，其风险却又是最大的。简单地说，有冒险精神的管理层往往采用这一策略。所谓“冒险”，意味着这样一些事实：

- (1) 管理层对移植目标并不充分了解；
- (2) 移植目标的可能收益相当大，从而能构成诱惑；
- (3) 可能发生的移植失败在一定程度上来说是可以承受或有所预期的。

例如，我们可能要在开发活动中引入“敏捷方法”，于是我们要考虑建设“敏捷团队”。但管理者可能并不清楚什么是“敏捷团队”，只是听说这样做开发快、质量好，而且对技术人员的提升很明显，于是便有意让现有组织“引入”这一组织方式。

如果敢于冒险而又不敢于坚持，这种移植方式往往是收效很差的。

但正是因为管理层原本就是期望“快速地”解决问题，所以他们在这件事上并没有足够的毅力。一旦“移植组织”面临团队内部矛盾，或团队与现有组织结构产生冲突，整件事情就会渐渐陷入泥沼。接下来，管理层如果在“继续与否”这个问题上稍一犹豫，则整件事情就要被搁置了。这种“稍一犹豫”的根本原因在于，组织管理者原本并没有做好“管理这样一个移植过来的组织模型”的准备。也就是说，他们事实上根本不知道如何去管理。但这样的管理层与“山大王”还是有着不同，这一区别在于：前者缺乏对组织的认识，后者缺乏对权力的意识。

敏捷团队实践中最大的问题，并不在于其内部运作——这通常在短期的训练之后即可解决，而在于这个团队与既有组织模式的整合。更明确地说，就是让“旧的管理层”如何接纳新的事物的问题，而这并不是决定“移植”就可以解决得了的。因此，在诸多的尝试之后，（保守而又不思上进的）管理层往往认为他们已经“迫不得已”要选择第二种方法了，即栽一片树，这总好过弄一堆树来却不知道怎样让它们成活。

栽树是相对慢得多的一种布局，因此“为什么栽树”就成了很实际的问题。急着乘凉摘果的组织是没时间栽树的，这在逻辑上也成立：人都快饿死了，还谈栽树做什么？所以一个创业型的组织通常对这个问题缺乏思考，而当整个组织的规模达到成百上千人时，这个问题就凸显了出来。而且它一露出端倪，就将立即伴随着大量的组织变更，例如部门调整、人事任命、责权划分、奖惩措施、问责制度等。

这些都只栽树的手法而已。树可以拔了再栽，只要地盘还是你的，就无碍大局。这就是布局的根本思路。时、势造英雄，栽树就是一

个让中层成长起来的、长期的组织战略。

9.2 为 1 万人做布局

“木秀于林，风必摧之”，只要不倒，长成歪脖子树也是景观。在这个过程中，大多数人只着眼于歪脖子树“成长的奇迹”，或感叹左一阵右一阵的风，少有人观察到那片“林”。

我在 D 公司的时候，例行有三个月一次的组织管理大会。某次，CEO 宣布了一个大胆的组织计划：

- (1) 建立产品专员制度，产品专员可自荐或推荐，由 CEO 亲自挑选与培训，适当的时候由总裁办公室委任到各生产部门；
- (2) 扩大管理规模，在原有三大中心的基础上，增加两个中心和相应的部门组织结构；
- (3) 宣布成才计划，包括购置与投资第三方开发组、研究室或小型软件公司，并成立相应的监管部门与监理机制。

大会结束时，许多人在私底下表示了对这件事情的不理解：我们才两千人，怎么公司就有了这么多管理层级？干事的全变成了当官的，这活还怎么干？当一个同事向我提出这个问题的时候，我回答道：如果你把这看成一万人的公司的管理规模，你就觉得合适了。

CEO 是在为一万人的公司做组织布局。

9.3 组织机会

为什么别人做这样的事情就是乱来，而 CEO 做，看起来就是“组织布局”了呢？先撇开“该谁做”的责权问题不说，所谓布局是“有象无形”

的事情，即我们大体上看得到一种趋势，并在这种趋势下采取了一些作为，但这时确指“做成了什么样”还言之过早。

这次组织布局概括来说，就是三点：

- (1) 权力下放的核心方法；
- (2) 组织扩大的规模与路线；
- (3) 新组织的内容来源。

在组织上，“人与事”是两个核心路线，而“权利”则是决策依据。正如我们前面所讨论过的：“权利”围绕着“利”的确认而构建，“目标”是对“利”的阶段性的设定，“人与事”是实现目标的组织要素。

在 CEO 的这个组织布局中，你看得到“基于产品的专员制”，其核心在于“权力”由 CEO 给出。因此产品的定义权、决策权和认可权等，都归“CEO + 总裁办公室”的组织结点。换言之，对于今后上万人规模的公司来说，其核心利益以及与此相关的产品生产决策，将仍然是由 CEO 来把握。

你也看得到“新中心与新部门”，这意味着“项目中心管理制”将是后续组织制度的基本模型。你还看得到“购置与投资”，这意味着新的生产力量主要来自于外部，并且公司为这些新的组织结点配置了“监管部门与监理机制”。这些都必然意味着“大量的、非生产性质的管理岗位”的产生。

那么现在来看，这个“组织布局”还有冗余的问题吗？

再深一层地发问：你看得到其中的组织机会吗？

9.4 棋子之所以为棋子，在不见棋眼

但这与“具体工程”有什么关系？

工程，或用来指代“其具体事务”的项目，是中层管理语境下的一个讨论对象。对系统，它依赖于具体的组织结构形态；对外，它依赖于管理层授权；对内，它依赖于团队协作并逐渐形成独立的行事风格；对局部，它依赖于产品或产品线对于“事务”的定义、限制以及相关的决策过程。

具体工程，是一个无法脱离环境约束的、对工程的定义。大多数对具体工程的成败构成显著影响的因素，都是组织行为而非具体事务的做法。

因此，如果组织在布局，那么项目多数是“求势”而不是“求实”。我曾经举过一个例子，你的主管可能会要求你^①：

在 Lucene 这个搜索引擎的基础上做几个示例程序。

但他不会告诉你：

公司打算选择一个开源的搜索引擎。

或者更进一步的问题是：

公司必须在使用商业产品、开源代码和自主研发一个搜索引擎之间做出选择。

又或者导致这一问题的根本原因是：

某某搜索引擎公司正在与高层探讨战略合作，而我们必须“尝试

^① 引自《程序员修炼之道——从小工到专家（评注版）》，第7章。

合作”的同时，通过支持某一个开源引擎来保持战略选择上的灵活性。

如果主管所要求的“做几个示例程序”是一个项目，那么项目中的实施者又如何能看到组织在“布局”上的全像呢？这个例子的不恰当处在于：也许有人会认为“做几个示例程序”是可以放弃的棋子，而布局是那个弃无可弃的棋眼。

但除了布局者，谁又知道棋子与棋眼的区别呢？

9.5 时者，机之所发

组织的契机，便在于棋子与棋眼的选择与变换之间。

在 E 公司，员工 A 在跨部门调动时，新部门的主管 B 找他谈话，大体上是讲讲今后的工作内容、工作方法以及相互之间的协调配合等。但他们的谈话，事实上已经快速地切入到今后工作的责权分配问题。当时员工 A 便提出了他的要求：你可以给我一个目标，我来决定怎么做并确保它的推进。但是有两点：第一，不要怕我闯祸；第二，我做事可以，不背黑锅。

主管 B 对这个“赤裸裸的要求”感到有些震惊。然后他从职业精神与工作态度的层面对此加以了高度赞扬，并接受了员工 A 的调动。在随后一年多的时间里，员工 A 既没有闯祸，也没有背黑锅。

但问题是，员工 A 基本也没什么事可做。

不做事当然不闯祸，当然也不会扔出黑锅或背上黑锅。主管 B 的用人与治事是否出了什么问题，这并不好说，毕竟这与“部门要做什么事”以及“员工能力范围内适合做什么事”还有关系。但是对于员工 A

提出的“条件”，主管当时的反馈并不适当。

当我们展开一个合作关系（而非简单的雇佣关系）时，总是有一个相互试探过程的。员工 A 的条件看似随意或只是为了彰显个性，但实底里是对管理责权的要求——如果他与主管 B 之间的确可以形成权力关系的话。因此大体上来看，这个要求的目的是在合作、任用、使用等多种相互关系上定一个基调，例如期望借此了解“如果要让我当排头兵去做事，那么限制的边界在哪里”。

对这一问题，我认为可期望的、也可能是最佳的答复是：你要做任何事，我都表示支持并完全不发言；但是如果我要发言，那么你一定要听。

棋子与棋眼的关系，便在于可控与不可控。如果主管不能控制一枚棋子，那便将他只用作了棋子，而绝不用作棋眼；若要将他用作棋眼，那便要放得下去、留得住、看得到。

主管 B 对员工 A 的反馈只着眼于“可否成事”，而不在于是否“以之成势”，前者是用作棋子，而后者可作棋眼。但在他们所讨论的组织层面，仅仅谈“事”是不够的，信任、责权、方向以及对形势的分析与把握等要达成一致，或形成一体，才能谈得上合作，否则便难成大用了。

9.6 任免

一个九岁的小学生说她“幸亏没有要当总统的梦想”，于是妈妈问她为什么。她说：

当总统也就是刚刚当上那会儿会很开心，好像是赢了。你看奥巴马，选举的时候许多人支持他，那么风光，而一当上总统，反对

他的人马上就出来了。^①

没有人天生就是管理者。不管是老板、部门负责人或项目经理，大家只存在管理对象的不同，在管理能力上其实是一样地“先天性缺乏”。所以多数情况下，我们都处在一个“并不太合适”的组织位置上，至少我的所见是如此。因此我们必须清楚地认识到什么才是“合适的位置”，或与这个位置匹配的“合适的能力”。然而在这一点上，绝不能以“别人的反对与支持”作为判断标准。

对于管理层来说，一个管理者是否“被反对或被支持”都不是组织任免的核心原因。大多数有成就的管理者，往往都面临爱恨交织、毁誉参半的局面。通常一个组织内的各个角色，以各自的视角与利益取向来看这个“管理者”，损己则责之，利己则爱之，反而很少去思考“如果自己也在那个角色上，又能如何”的问题。

很少有人会设身处地地去“为别人思考”，所以大多数的言论貌似“诚实、诚恳、由衷”之言，实则是无关痛痒的“观感”；即使这种观感有一点点进步，也不过是因为触到了发言者自己的利益而变得带有些切肤之痛的味道来。

九岁的小学生还不知道那些被代表的利益，因此只看到了支持与反对之间的矛盾。

9.7 此事要躬行

工程做不做得了，是不是技术问题？

这是一个关键的设问，涉及很多方面。例如其一，工程在时间与空

^① 引自《羊城晚报》（2011.07.28）的文章，《日本孩子长大后最想做什么》，作者唐辛子。

间上可能达到的规模，是否可以通过组织团队或项目来满足；其二，工程中可能出现的问题，是否可以在技术上解决；其三，工程所需要的资源，是否可以持续供给。第一个问题，是工程在系统架构领域中的技术问题；第二个问题，是工程在产品/业务领域的技术问题；第三个问题，则是工程背景下的商业环境问题。

工程自身的技术问题，是类似于“过程、方法、工具”这样的、传统而学术的工程视角下的问题，是把工程作为技术并对其中所涉及的干系者（人和物）的统一管理。但是这种管理又具体与上述三个方面（架构、技术、环境）的问题无关。例如，我们在工程中所讨论的过程是以瀑布模型（分析—设计—开发—测试等环节）为基础的，它根本上与“怎样一个具体的架构、技术与环境”无关。这也是《人月神话》一书中“银弹”的核心，即我们是否可以找到一个与具体情况无关的工程方法。

这在根本上，就是“唯技术论”的思维。而一个具体工程下所需的管理与组织，却并不能基于“技术思维”而得到。正如我一再强调的：

真正的管理者，是不会看“管理的技术书”的。

管理的总的诀窍，在于一以贯之。无论是什么，吐啊吐啊就习惯了。^①就如一个著名的组织行为观点所说：改变不了思想，就先改变行为；行为改变了，慢慢地思想也就改变了。对于大公司的体制来说，往往需要“我们”是被改变的那批人，并且往往是“先从行为改起”。在本质上，这是形成企业发展的一种势态，当大家的行为模式、方法、习惯以及思想趋同时，要做的事情便可以顺势而为了。

^① 这既是温水青蛙式的风险，却又是组织行为的精义。

真正的组织管理者，既在寻求组织的不变性，又在这种不变性中看到变化的机会。组织中的大多数人只是被左右挪动的一枚职务角色，甚至像刺秦的力士一样连名字都不为组织所知。正因为如此，探索并形成有利于“工程实务”的组织结点，是大多数公司在“软件研发管理”上的痛中之痛。

在这样“探索并形成的某一个组织结点”下的一个工程（或项目），便是我所谓的“具体工程”^①。这个工程具体而微：有人、有事、有目标，也有可资借鉴的过程、方法与工具。这个工程借鉴但并不照搬任何一个既已存在的模型或过程方法论。这个工程的首要任务是完成具体的“某件事情”，这既包括让大家清楚地认识到这个事情的目标（可能表达为产品或产品线上的阶段等）、收益以及风险，也包括构建让所有成员都能有效投入的职业氛围。不过，它并不为“明天公司是否以此为范本”而负责。

那是别人要的，不是你要的。

^① 我们此前的许多讨论并不适用于“开源工程”，但这句话是绝对的例外。具体工程是一种从对“工程的结构元素”的认识出发，试图在特定背景下形成适宜的工程方法的思维方式，本书的讨论可作为这一思维的一个示例。

附一：行在道上，从局部到全局

节选自《程序员》杂志 2009 年第 10 期同名文章。

EHM 不是一个用于实作的工程模型，它只是从某个角度分清了工程的一些环节而已。如同“牛屎图”一样，EHM 是一个思考模型。相较而言，EHM 更易于发现问题，而“牛屎图”更适于理清思路。如果非要从中找出个“更核心”或“更重要”的，那么在“牛屎图”中要以哪一个作为最底层的圈子也成了问题。所以不妨抛开“谁更核心”的问题，把赵善中先生^①的模型、EHM 图以及“牛屎图”作为可以相互补益的图来看，这样就所得多多了。

一个软件产品，究竟是被“开发”出来的，还是被“架构”出来的，抑或是被“管理”出来的？这是一个争执不休的问题。一般人，尤其是技术出身者，会直接否定掉第三种答案。他们认为“一个软件产品不可能被管理出来”，它只能是被开发出来，而管理不过是这个过程中的

^① 赵善中先生是“结构行为合一”架构方法的创始人，人称“SBC 架构之父”。阿拉巴马大学计算机与信息科学博士，先后任职于中华电信公司、GE 实验室、中山大学资管系。时任国际企业架构师协会台湾分会理事长。

官僚角色罢了。但什么是管理？管理真的就是今天命令你工作、明天要你汇报进度的那个人吗？是不是我们把一些个人的私怨——对某个拙劣的管理人员的不满或轻视——带到了我们讨论问题的语境之中？

开发、管理和架构三种角色，站在自身角度，都会认为自己应该主导软件产品产出。这种观点我向来持以理解而又质疑的态度。更渐渐地，我对所有类似“某个单一角色主导了软件产品产出”这样的观点都表示怀疑。正如高焕堂先生^①所提到的，这更像是一个“三足鼎立”的局面。这种“鼎立”的局面是一个衡势，这意味着它的平衡是瞬态的，且总在平衡与不平衡之间。所以我的疑问是：这种局面对于软件的研发、项目的过程来说，是适宜的吗？应成为常态吗？

从组织学（而非单纯含义的管理学）的角度上来说，鼎立是组织的一种形式，是管理的一个施为目标，而不是管理本身。虽然我认为高老师所指出的“鼎立”的局面可能是将来的方向，但如何去组织这样一个组织，管理它，并在这一局面上产出软件产品，是更进一步的学问。

在《大道至简》一书中，我基本否定了对软件开发过程的管理：

“开发团队并不需要管理。或者说，在你没有弄清楚状态之前，不要去管它。”

同样地，我也否定了传统的软件产品的“产出”观念：“**工程不是做的，是组织的。**”既然我们不能“管理”一个团队，又不能去“做”工程，那么我们该怎么办呢？在我跳出上述的三个角色之后，我得到的答案

^① 高焕堂先生是台湾软件架构设计大师，从事 IT 行业近 30 年，被称为“台湾 OO 技术教父级代表人物”。曾创办 MISOO 对象教室及《SoSE 杂志》，时任 MISOO 软件开发与管理顾问公司 CSA 职务。

是：组织。

进一步地说，管理角色的任命、项目团队的结构等，都是在“形成组织”——这一过程中的产出和阶段性的决策。对于（泛义的）软件项目来说，没有一成不变的组织，也没有一成不变的模式。更进一步地说，在（具体的）某个软件项目中，组织的行为也处在变化之中。

我推崇高老师“以序容易”的架构思想，而这也意味着我上面的言论会有一个推论：必然以及必须要存在一个“无序”到“有序”的过程，即我们最终仍然会得到一些“组织模型”（以及管理、过程等模型），用以规划和指导实施同类性质的项目。我不否认这一状态和阶段性结果的必然存在，但我怀疑现在是否已经存在，例如某些工程界吹嘘的“模型”是否就是终极的银弹。

因此，回到赵老师的话题，在“工程实施”的语境下，究竟（现在）有没有确定的模型呢？我不置可否，唯只置疑。架构是工程中的推手吗？是推动的原力吗？我从架构这一角色的位置及权威性上看得到希望，也从这个角色所必备的能力涵养上看得到希望，但是推及到组织及具体的项目，架构角色真的有这样的能量吗？谁赋予了他调适组织形态、降低组织内耗的责任？如果他没有这样的责任，那么组织如何生存？如果组织不存在了，岂不是仍然回到了“皮之不存，毛将焉附”的问题上了？

附二：本来面目——大教堂、集市，与作坊

节选自《程序员》杂志 2009 年第 3 期同名文章。

不管是理想化的，还是神化的方法与论断，在《梦断代码》里都用到过。当然这些方法、论断也确如乩言一样灵验或失灵过。正因其灵验，以及偶尔的失灵，才让那些崇信者拜服与生畏（更何况一些崇信者原本就是这些方法与论断的始作俑者）。

《梦断代码》中的团队有一个绝对集市的名字——OSAF，也有绝对集市的原则：开源且接受大众的眼球（眼睛足够多，缺陷无处躲）。但是这个项目启动了一年之后，这个团队还没有一个真正意义上的领导者：他们（包括那些从来不出现在团队中的志愿者们）总是在不停地开会、讨论，以及推翻上一次的开会与讨论。

团队和产品的精神人物，是又出钱又做事的卡普尔。他声称要做一个能打败一切 PIM 软件的 PIM 软件。简单地说，直到这个软件只剩下以自己为对手，这就是目标。类似的目标也出现过、成功过，例如 Internet Explorer。因此，看起来卡普尔是要在开源界打造一个类似的神器。正是在这样的精神感召下，一批热血之士“混成”（混

然天成的混成）了这个团队。

卡普尔看着大家积极地讨论着他的“原始需求”，并接受了他们制造出来的更多的需求。在这些目标堆积成山之后，卡普尔开心地聆听了第一任项目经理基尔默（或只是一个技术首领）的预期：我们将在年底发布该软件的一个早期版本，预计明年年底或稍迟的时间即可以发布 1.0。

“他的预估太过乐观了”，Rosenberg 在后来说。

在《大道至简》一书中，我写过：做了这么多年项目，当我一听到“那我们就开始开发吧”这句话，就哆嗦。因为大多数人在说这句话的时候，连组织结构是什么都还没搞清楚呢。

果然，接下来 OSAF 团队陷入了“与组织作斗争”的阶段。没有决策者，这意味着任何一个“看起来像决策的东西”都可能在你背转身去的时候被打破（P.154）；没有设计者，这意味着任何“看起来应该已有雏形的东西”其实都未曾现身（P.132）；没有跟踪与限制，这意味着任何人都可以停下手中的工作，去做另一件“看起来更要紧的工作”（所以他们制造了大把大把的工具并不停地废弃它们，P.129）；没有.....是的，OSAF 团队没有任何可以依赖的组织结构，直到一年之后，他们把项目组分成了三个小组，并设立了正式的团队经理，由卡普尔兼任（P.157、P.158）。

“别再继续发疯了！在书里，他们该错的都错了！”Rosenberg 在后来说，“我自己也在怀疑，他们何时才能起步？要花多长时间？障碍是什么？”（P.158）在这个时候，《梦断代码》这本书正好写到一半。你可以猜测另一半的结果，或者干脆直接给它判死刑。

但真正的问题是在哪里呢？是集市工程的失败么？问题是，《梦断

代码》在后面还记叙了微软 Longhorn 项目的失败，那里有着最具规模的教堂队伍，从神职人员到扫地的，一个不缺（据说连扫地的都至少是硕士以上学历，以备将来发展为扫地老僧）。

具体工程也会失败，给一个工程冠以“具体”的名字，并不代表它就成功了。《梦断代码》讲述了一个具体工程的失败，失败的过程闪耀着光辉：集市的、教堂的，以及牛人们的、有钱人们的.....无限光辉。

但是他们想做什么来着？

附三：杀不死的人狼—— 我读《人月神话》

节选自同名博客文章（2007 年 3 月），该文章是关于《人月神话》的读后感。

总的来说，我事实上并不反对某种具体的方法，而是在努力学习种种方法。但我并不认为有什么单一的方法能解决所有问题。每一种解决方案都有其前提和背景，精通一种或全部工程工具、方法和过程，都无助于掌握这些前提、背景以及潜在的关系。理解工程的适用性，涉及工程专家的整体素养和实践经验，也涉及具体的工程环境、目标和实施者（团队）。而这些课题，正好是目前的软件工程甚少谈论的。

在银弹的话题上，答案不外是：有、没有和将来一定会有。我的答案与这些都不一样。我认为“有没有”的话题没有意义，不值得讨论。因为 Brooks 在人狼的设定上是一个伪命题，而在银弹的假设上则是学术的。我进一步的观点是，广义工程对首要任务（人狼）和完美方案（银弹）的假设，不应该成为狭义工程所追求的终极与利器。

我们要分清狭义工程与广义工程。正是由于他们对本质需求的设定

完全不同，因而也有各自不同的主要与次要任务。一切工程活动的历史告诉我们，曾经的经验、失败和成功，以及据此在广义工程中归纳推演的理论，都只是我们在具体的、狭义的工程中的参考，而非无往不利的银弹。

曾经，我们如同走在一片沙漠，看不到沙漠的边缘，也不知道如何行走。Brooks 的伟大贡献，在于他用《人月神话》指出了一条道路并教给我们基本的行走方法，让我们从中学会了辨识流沙，懂得了沙暴的征兆；同时“没有银弹”的假设激发了我们的斗志，如同有人在说“你要证明沙漠有边界，你就拿出一束青草来看看”。

于是有了几种喋喋不休于“有没有青草”的人。一种是已经在灵魂上上升到另一个层次，故而无需行走于沙漠。另一种则只是站在沙盘边上，用长杆推动卒子，而鞋子上根本不会沾上一点点沙土。还有一种人则不时地宣称他们已经找到了青草。他们总能确保你在付费的时候看得到青草，而你转身时才会发现那里并不是沙漠的边缘。

剩下的人一边喃喃着“沙漠边缘的青草”，一边在焦躁的沙海里缓步前行。不同的是喃喃而不自觉者掉进了沙坑，心怀憧憬而盯紧脚步的人则走出了沙漠。

走出来，你才会觉得：原来有没有青草，并不是那么重要。

附四：工程的本质问题是组织

节选自《大道至易》第1版原书总论。

工程不是“做”的，是“组织”的。这个“组织”，既有名词的含义，也有动词的含义。除了这个层面上的意思，它还意味着一旦没有确定的组织模式，那么一个具体的工程也就难以落足——即使是个体工程，也有一个确定的组织模式。

组织的源起及其发展的过程，是一个核心问题。

组织的背景

然而在实践中，并没有人注意到工程的组织问题，而只看到了工程实施的结果。这些结果包括 RUP、UML、XP 以及未知的种种模型、框架和方法论。所有这些看起来可以一统天下的，或致力于一统天下的方法，都最终走出了它们原本栖身的岩洞，走入莽莽苍苍的草原或森林；而它们还在认为可以通过敲击岩壁去听发出的声音，并

以之判断树洞、沙穴或草坑的居住年限。

它们的共同问题是，没有考虑到具体方法实施时的组织背景。

组织是一个项目的原始背景，组织的构成是人，是人群。工程中最终要解决的，不是具体方法的问题，而是将不同性格、性质的人组织成群体并实质推动的问题。从这个角度上来说，《集体行动的逻辑》所带来的思考远比《人件》、《人月神话》来得更加深刻和直接。

无论是以人为工件，还是以“人/月”为度量衡，根本上还是把人作为个体来思考的。虽然这些直指个体的思考是必要的，但也无助于组织与该组织下的集体活动。

契约社会与人情关系之间的区别

中国的传统社会模型是基于人情关系的，这与儒家的社会阶层化有着密不可分的关系。儒家从根本上来确定了天地人伦的关系之后，人们的社会行为、族群关系以及家庭婚姻等都是在这一背景下的自然发展。因此，我们讨论的同学、朋友、党派、宗族以及更简单的门当户对等，都是人情关系的基础单元。并且这样的基础单元深深地植根于我们的民族文化中，是文化内涵的一部分，也是构成我们民族化的人性的一部分。

人情作为社会成本的一部分（如果将规模缩小到一个组织，则是组织成本的一部分），是我们这个社会的现实状况。这个成本在不同的领域与不同的事务之间，并不是均匀分布的。有些领域例如产品生产，人情成本会少一些，而另一些领域，例如组织构建，则会多

一些。粗略地分析这一分布的模式，大体上（并且与后面我们要讨论的契约社会比较来看）可以认为：所需要决策的事物对**人与人之间的信任**的依赖程度，决定了人情关系的比重。

我们将人情社会中的人情成本转嫁为信任成本的缘由在于，很大程度上来说，人情是构成社会信任（以及组织信任）的要素。对于任何一个事物的决策，信任所带来的收益往往是别的努力所无法达到的，例如开上 100 小时的会议抵不上老总的一句口头传达。这看起来是政治与权力在起作用，事实上也可以看着对权威的普遍信任。

我们讨论的“组织”，既是社会背景下的一个局部群体，也是社会行为模式下的一个普遍模式。简单地说，我们的大多数组织都是人情组织，而这样的组织下也都是人情成本转嫁为信任成本的。正如某天，一个开发人员对我说他想转职去做管理，我问他的第一个问题是：你认为去做管理，有多少人会听你的？同样的问题是：你去做产品，有多少人听你的”，或“你要去创业，又能拉上几个兄弟的人马？”

当这样的问题提出来时，我们的大多数开发人员都哑口无言。因为多数工匠思维下的工程师是关注做事，而不关注人情的。在我们的社会与组织背景下，事做得好不好，只是整件事情中的一个比重较大，但不十分关键的因素。同样地，作为一种“集体行动”的组织方法，与“在做的事情”关系也并不大。例如我常常让别人思考：“为什么外行可以当领导？”我的意思是说：领导作为一门技术，显然也是有其方法的。所以，反过来也可以问：一个“很有领导能力”的领导，为什么不能做（这件事的）领导？然而，这样的逻辑——至少在我们的背景下——掩盖了基于人情与信任的那些逻辑，即我们的组织基础以及由此进化而来的行事逻辑。

然而我并不是要将这样的组织与逻辑引入到我们的“工程”中来。正好与此相反，我是要将这些因素从工程中排除出去。我是想彻底地问清楚：工程到底是什么呢？在我们的讨论语境下，所谓工程，指的是一种行事方法。如果我们不能清醒地认识到“方法是依赖背景的”，则我们不会转而去看一看这些背景，也就不会把这些背景摒弃在我们的工程之外，也不会善用之，使其变成有益于工程方法的一部分。然而这仍然存有巨大的风险：工程究其底里仍旧是事，它必然有着方法的背景。那么如果我们把“这样的组织与逻辑”推出去了，又能把什么样的东西拿进来呢？

我的答案并不是“契约”。

只有**基于契约的利益得失**严重影响到社会成本时，“契约”才能够作为组织手段实施。这是基于“自利”这一思维模式的、必然的、必须的组织结论。换个简单的说法，就是如果毁约代价大，则大家都守约；否则，大家该人情还人情，该无耻还无耻。就如同没有极端代价的、公义的法律约束，大家就会犯法一样——这看起来很是好笑，但我们的社会在底子里就存在着这样的一个模式^①。

有没有可能换一个视角来看这个问题呢？比如信仰，又比如文明。但如果没有与法律这种“人与社会的契约”类似的东西的话，又何以判断行为是否“文明”，或是否是“信仰所确定的一部分”呢？看起来，“道德”可能是一个很好的解，也就是“讲道德”大概是文明的，或符合某种信仰的规义的。但“道德”又是什么呢？

我一直说我们的思维模式是理性而又逻辑的，因此我们必须先说清楚“是什么”。然而在我们的实践中，是不是要这样才没有“错”呢？看

^① 至于这一模式诸多的内在与外在的驱动力，还可以有相当多的讨论，但并不是我们这里的话题。

起来，我们一方面在花极大的时间去讨论这样的“正确与否的问题”与“问题的正确与否”，却在一些现实中要去解决的“事”上寸步不前。

换言之，我们是不是可以把大家普遍看起来“道德一些”的东西拿来就用，而暂且无视它学术的、逻辑的或严谨的定义呢？

回到契约的本质上来。即使我们承认一个人的选择是自利的，承认这个选择的背景是阶层化而非自主平等的^①，但为什么不能将选择的结果视为契约的呢？“诺”于一职，“诺”于一事，于“人情社会”真的就是挑战么？我认为不是的。所以在我看来，工程在具体上于我们的解，便在于大家对事的负责与不负责，而不在于对组织“是否契约”的讨论。若变了法子要去改变组织，或要去理解组织的全体，既无益于我们的一个具体的工程，亦无益于我们自己的选择——如果是后者，那你的选择应该不是“做一事”，而是组织的革命^②。

对此，承认还是漠视？

学术的讨论通常会将契约与人情二元对立起来，试图非黑即白地讨论这二者。大多数情况下，一个论者必须表明他对两个观点的态度，并且义正辞严地批判另一种。出于学术的讨论与真相的挖掘，这样极端的思维——在一时之间——是必要的。因为若不如此，我们便

^① 维护契约及其平等性才有必要，因为这就是社会信任的全部出处。然而，如果我们以“既存的人情”与“基于人情的组织”来讨论这一问题，那么人情关系对信任的影响就相当大了，而“维护契约及其平等性”的必要性则会相应地变小。

^② 做好一职，做好一事，是我对事的求解，而非对组织的求解，亦非对软件工程的求解（它可以视为从组织视角出发，在具体工程下的一个可能的解）。需要明确的是，所谓“一诺”是基于道德的，而非基于契约的，只是在形式上用到了契约的封皮罢了。

难于看到一面或另一面的真相。有了这种学术讨论的氛围，若有人说“契约社会不错，人情社会也很好”，那么就会被斥为和稀泥，变成两种观点同时讨伐的对象。大家争来讨去，把第三者打倒了，又开始双方的互殴，这是市井间打架的惯常路数。

改革家也会跳出来。他们往往关注旧事物的败亡，以及新事物全面占领阵地的那一刻。这种情况下的局面通常是：旧的东西被没有了，即使有一点点可能存在的痕迹也要用黑布挡了起来；新的东西看来已经存在了，至少每个人的口号中都这样地呼喊。然而这时，这些“改革家”（例如那些来向你宣传的布道者们）立即就消失了踪迹，因为他们也在瞬间失去了主意，不知道应该怎么办才好了。——他们的义务与职责，原本只是改革，无对象可革了，这样的角色也就不复存了。

所以真正有意义的改革是革新，而不是革命。革新是容旧渐新的，革命是推翻了事的。然而既然如此，我们便会推知一个重要的事实：旧的，必须与新的共存着非常长的时间。这意味着，这一过程中旧的必须渐渐“懂得”新的，而新的也必须渐渐“容得”旧的。于是旧的不再旧，而新的也必然不再是最初设定的那个“新”。因为，“新”的自身也必然是在发展着的，如我一再说的，即使那些“（看起来更正确的）新东西”是真理，也只是一时暂存的“真理的某个表象”，我们必是在追求真理的过程中揭示它的各个方面，而最终——以人类的终极来说，得到一个所谓的真理的。

看起来我是把这个问题变成了一个哲学的讨论。但问题在于，对于“承认还是漠视”这个设问来说，观点无非是“承认”、“漠视”以及“一边承认着，又一边漠视”。这样的观点在二元论的、一元论的思维体系中，总是要去打倒一个两个的。于是我们大家就变成了一群聒噪的

村妇，为着地头上的几株桑树占了谁家的地盘而厮打不休。

然而又过了些年，无论吴、楚都归作了秦，这些厮打也就全然失去了价值。

学术与实践的区别就在于此。能用就用是践行的第一原则，小平同志的话更简单直接：不管黑猫白猫，能抓老鼠的就是好猫。而在我们的工程实践中，我们的背景是人情关系，是阶层管理，是不完善的规章制度……我们若是无视这些而去实施某种“标准化的”工程，那是有革命的胆量却没有改革的眼光，勇而无能；反过来讲，若是受了这些的牵绊，便又陷入了人事的、章程的困局，因循守旧故而难有破局。

然而我的问题一直不在于这样的——一个“看似合理而又毫无意义”的结论。亦即是说，偏左亦死偏右亦亡，是通常学术所谓“骑墙派”的观点。——这些骑墙看戏的，通常置身事外，将看戏视作一种乐趣亦或是一种必须。尽管这样的“结论”无助于任何的实践，然而这一“过程”——

“停下来，看看你们在干什么，在什么样的背景下干”。

却是实践所必需的。

所以我认为，工程的问题，尤其是一个具体工程的问题，还是要回到工程的本体——这个组织与系统的背景上，重新地审视才对。无论那些争吵不休的学术家，或是那些埋头苦干的践行者，都应该仔细看看我们的工程环境，看看那些“正在做”的做法，看看那些在组织中被记着“人头数”的工程师，以及被记着“大头数”的管理者们^①。

^① 在大多数的公司中，一个部门或团队只是“你有多少人”，以及有多少个“高管”或“大 V、大 P”这样的概念。

我想若是停了下来，哪怕一刻，也是有益于我们前行的。

组织与系统

如果“你”已经是组织角色中的一员——例如架构师、项目经理或程序员，那么当跳到系统之外去思考时，你会提出的有关组织的问题是：那个人是谁、在哪里以及在做什么？人总是能看清别人而无法认识自己，因此当你看到“那个人”其实就是“你”的时候，你就已经看到了整个体系、整个体系中的每一个人与每一个“我”——当看到“你”的时候，你已经游离于自我之外。

于是，你的问题是不是在另一个体系上，或在你当前的体系上，只是你的视角问题。例如，当你辨别出了所谓的“另一个体系”以及“当前的体系”，那么只是你因为观察的时候站在了两个体系之间或者前后。如果，你能够站在由二者构成的“整个的体系”之上，那么你的问题就被投放到了这“整个的体系”之中，变成了一个确定的问题。

而所谓的“整个的体系”，就是整体。

你看得到“整体”吗？

所谓“架构师要有大局观”，其实就是指架构师对系统全局的思维能力。例如，当你把需求方纳入你的思考范畴之后，你就来到了更为复杂的工程背景设定之中。这时你需要考虑的问题，将不单单包括一个项目的具体实施过程，也包括一个项目如何产生与交付。影响产生或交付一个项目的因素极其复杂，从社会的产能，到各国首脑

组织管理者对“人/角色的性质”的、不正常的忽视，确是由来已久。

的多方会谈，甚至到跟你握手的那个客户经理能否在楼下顺利地找到停车位.....这所有的一切，构成了一个项目过程的外部因素。而你的组织，只是飘摇在这些外因之下的、微不足道的一把木片而已。

只有站在组织之外，才能看到组织的整体。无论那把木片中的某一块（例如架构师这个角色）的材质是何等的优良，也不能决定自己以何种形式、能否足够优美地飘摇。因为局部不能决定系统，它只能影响后者，或受后者影响。最简单的实例是，你的项目过程再完美，但老板说客户已经破产了，于是你的项目就终结了。在这个例子中，“公司+客户”之间的关系，决定了项目的成败，与团队、组织、过程这些因素全然无关。