分享书名《开发者思维》完整内容（部分章节内容）的简介【必须】

开发者思维，不只是一种让开发者感到被重视的管理方式，更是企业制胜数字化转型的全新方法论。 从苹果、谷歌到亚马逊，将开发者视为真正的战略人才和合作伙伴的公司正在取得成功。 如今数字化革命浪潮正在重塑企业管理格局，软件开发者成为现代数字经济竞争中急需的人才。

然而，大多数公司仅把他们当成数字工厂的打工人，却没有真正了解如何充分解放他们的潜力。我们需要充分理解和激励开发人员，用软件的力量驱动企业增长与创新。

Twilio联合创始人、CEO 杰夫·劳森，将十几年的开发者管理经验汇集成《开发者思维》一书，为读者提供了一种全新的思维方式：应对数字化转型，一家企业需要所有职能部门之间通力协作和思维转变，再加上实际编写代码的软件开发者的贡献，才能共同为公司打造数字化未来。

他还用诸多的行业案例为读者展示了，那些创造出改变行业的产品的商业领袖是如何持续做好三件事的：

第一，他们明白为什么开发者比以往任何时候都重要；

第二，他们真正了解开发者，并知道如何激励他们；

第三，他们愿意并擅于为开发者的成功投资。

既是开发者又是CEO的独特身份使杰夫·劳森能够将企业高管使用的管理语言与高绩效、有创造力的极客文化联系起来，构建“开发者思维”的思考模式，帮助商业领袖、产品经理、技术领袖和管理人员更好地理解技术人才并与之合作，以实现他们的共同目标——创造优秀的数字产品和体验，在数字化时代中快速取胜。

作者和出版年份，版本等等基本信息【推荐填写】

作者 杰夫·劳森出

版年份2022年

出版信息第一版

吸引你的地方【可选】

语言通俗易懂 作者成绩卓越，他的成就已经证明了他的思想

你推荐的理由【可选】

1 程序员朋友可以了解下

2 获得创业灵感和对SAAS的认识

你准备展开的几个话题及你的主要观点【推荐填写】

这些话题可以是书里书外都行，但是应该跟这本书有明确的关系。

1 软件和互联网公司以及创业公司，怎么管理人才

2 持续创业者怎么发现靠谱的点子

介绍

一个月前，我聊起了这本书，但是当时时间仓促，有些事耽误了。

我已经忘记了我当时是怎么介绍这本书的，我也忘记了我大概要分享的内容。

所以这次，我决定使用讲稿的形式来记录我大致要分享的内容。

反复阅读

如果要我今年推荐一本书给其他人，我要推荐这本书。

这本书，塑造了很大一部分我的关于企业数字化的知识。

而我，反复的阅读，每一次阅读，总能感觉有新的收获。

关于技术崇拜

说实话，现代的社会已经多少陷入技术崇拜的范畴了。人们对技术的意识空前的高，因为人们明白一个技术的出现会深刻的影响社会的形态。

技术的进步能够带来社会的变革已经是妇孺皆知了。

在200年前可能还没有这么高的认可。

今天这本书也不例外，他聊到了技术崇拜，尤其是对开发人员的崇拜。也许有些人看来有点过了，但是我认为刚刚好。

这本书的名字叫做《Ask your developer- How to harness the power of software developers and win in the 21st Century》。中文名 《开发者思维 技术如何驱动企业的未来》

这本书的作者是一个开发人员，也是现在 Twilio 公司的CEO和创始人。

他在书中强调对于一线开发人员的权力下放和给予充分的自由。

在我看来这是一种非常强列的技术崇拜的风格。

在传统的行业，可能销售是主导的，很奇怪，有无一家中国的工程师文化很强势的，有人说，百度不错，也许需要调查一下，但是百度的名声不太好。

不全是技术崇拜

这本书里也探讨了几个问题，比如

1. 在数字化转型的过程中，如何激励和激发开发人员你的积极性。
2. 如何了解用户的需求。
3. 数字化转型与数字化供应链（这本书比较宏观的一些观点）。
4. 作者如何发现并且做成Twilio这个生意的。

担忧-中西两种文化

书中说的很多东西，不能完全移植。

比如，公司管理中的集中统管与充分授权之争。

还是要结合具体情况来看待。

在书中提到的很多例子，我必须说明，尽管我们被它们振奋到，但是我们不能不加思考，尤其是需要注意到中美巨大的国情和文化差异带来的机遇，文化，认知的差异，除去这些，我们再来挖掘其中有意思的东西。

第一部分-构建和购买的选择

核心观点：

1. 软件部门已经从成本中心转向利润中心。
2. 全行业都在做数字化转型。比如教育，汽车，房地产销售和租赁，银行和其他金融市场，家居。

   > 优步是一家打车公司，我们也可以认为出行也在数字化。
   >
   > 票务更是
   >
   > 零售就不提了，电子商务，最早的数字化。
   >
   > 亚马逊创始人贝索斯很早就认为，亚马逊是一个软件公司，而不是一个在线书店。

3. 采购存在的问题，包括

成为一家软件公司

构建自己的软件是否是一个必要的事情？是的。也许以前不一定，但是现在是这样的，各行各业，但凡有未来长久发展规划和宏大愿景的企业，必须构建自己的软件。

软件的发展带来了新的机遇，即每个企业都有必要变成一家“软件公司”。

从一个宏大的视角来看，新的技术，带来交通和通讯情况的变化的时候，比如能显著提升人的影响范围，那么市场可能就会重新被搅动。比如飞机，火车，高铁的出现。软件行业也不例外，软件行业削减了企业管理中的信息获取和存储、检索成本，缩短了客户和生产商之间的距离，由此引发了很多意想不到的变化，催生了很多新的企业和商业模式。所以，现在来说，数字化转型是很多企业的热门方向。

假如，我们说，我们这么看我们自己的公司，我们是一家软件公司，只不过它不生产软件，只是用软件来服务客户。也许这样想有点牵强，但是有必要这么想，我们的软件是作为服务的一部分来让客户满意的，虽然我们没有直接交付给客户所谓的软件。

看看现在的很多银行，都推出了手机客户端，很多重要的业务入口都可以从手机上进去办理。这说明银行已经在大范围的提供信息化服务。现在银行的金融科技部门正在加班加点的研发新的系统，改造旧的业务模式。

在过去，很多传统的公司都把IT部门作为一个后台支撑部门，他们负责运维外购的ERP软件，以及很多财务系统软件，人力系统软件等等，即对公司内部服务而不能赚钱。

当一个踌躇满志的首席信息官（假如有这么类似的职务）想要改造一个传统的企业，ta面临的问题就是，采购还是构建的问题，即公司所用的软件是从外部采买还是自己内部团队研发的问题。

外购的好处无疑是很多的，比如成熟的软件，现成的通用的方案，很多系统只是为了内部使用，只要能用就行了，何必自研呢？何况自研也不一定专业。而且自研的风险也是相当的大，包括巨额的开发成本和后期使用中的不确定因素。也就是说，很多软件只要能用就行了，客户并不关注你到底使用了什么软件，只关注你的产品。

然而，现在情况慢慢发生了变化，即我们的产品有必要通过软件来向用户提供服务了，越好的软件越能提高用户的满意度。如果你的产品提供软件界面交互，则会拥有更好的竞争力，这样的例子比比皆是，比如在线看房，在线试穿，在线看车，远程系统诊断，在线店铺等等。

软件行业，准确的说，因为互联网加持的软件行业发生了巨大的变化。更多的竞争对手以更低的成本涌入了客户的视野，更容易因为某些特质而获得关注，其成本更容易被降下来，比如其生产基地被放在偏僻的地方而丝毫不影响其被知道，其成本被控制的更低，其人员更少，其决策更快，其人均盈利能力更强。

所以，软件已经从有从成本中心转移为利润中心的趋势。

在这种趋势下，外购已经不能满足所有场景的需求了，尤其是直接对用户提供服务的场景，更别说通用的某些软件实在不怎么好用。自我构建软件不是一个可选项，而是一个必选项，而且因为越来越多的玩家的选择加入而增加了反应缓慢企业被淘汰的风险，就好比达尔文所说的优胜略汰。

我们看到所有的银行都有自己的软件了，即我们手机上装的客户端，然而如果所有的银行如果提供的软件都是由一家公司提供的，那么必然没有竞争优势，因为大家提供的在线服务都是一样的，所以提供差异化的、更优质的软件服务是赢得客户的关键之一，所以必须自我构建软件。

内部工具的开发

听说一家软件公司80%的开支都是用于内部软件的开发和支撑，这虽然无法印证，但是说明软件公司花了很多钱在构建自己内部的支撑平台。

虽然没有查证，但是相当的符合直觉。

类似于台上一分钟，台下十年功。

软件的优势

1. 快速迭代。
2. 再生产成本几乎为0
3. 可以设计成非常方便的通道来满足人们的各种需求。

软件的思维模式在与提供最少的不可变性和最多的可变性和不断升级的可能性。比如遥控器的进化，比如电动汽车的控制台的进化，很多按钮消失了，取而代之的是大屏幕和更多的操作方式，持续不断的软件升级。

软件的思维模式契合不断升级的需要，降低了优化和重新发布的成本和时间周期，这正是现代社会竞争力的重要组成部分。

构建和购买的决策选择

购买通用软件存在着些许问题。

很多外购软件（包括请咨询公司进场定制化开发）的流程设计并不能很好的契合客户的实际情况，并且其开发周期慢，多个系统之间沟通存在着不少的问题，逐渐形成多个数据孤岛和缺乏协作。

在外购软件的时候，漫长的流程也非常的消耗人，包括需求征集，招标评标，听证会，诸如此类，一通劳心劳神的操作也未必能满足人意，甚至结果是矮子里面拔将军。更不用说后面实施的时候有非常多的不可控的因素导致工期缓慢和效果差强人意等。

很多外购的软件还存在着很多不确定的因素，比如停止更新，或者昂贵的升级费用，不断膨胀的解决方案来解决旧的问题，臃肿不堪的反应速度，链式的复杂依赖等等，旧的系统往往因为诸如此类的问题而难以升级改造，或者在艰难的升级改造的路上，使得维护人员苦不堪言而不知道何日能够结束这种痛苦的日子。

于此同时，有很多竞争对手已经开始自研软件了，不论是新贵还是旧的对手。这才是重点，如果大家都外购的话，也许没有问题，但是现在我们有可能已经被自研核心系统的竞争对手抛在后面了，所以努力追赶，努力自研可以提高我们竞争力的软件才是一个好的答案。

现在也有很多的机遇，包括相对比较完善的数字化模块供应链，能够避免完全从零开始构建系统，而是快速的利用外部的软件模块来构建属于自己的系统。相比于完全交给外部开发，自己内部的团队利用数字化供应链来定制和开发属于自己的系统，完全可控的多。

老牌的企业不愿意构建的一个原因就是害怕失败，但是当我们已经充分认识到外购回不断的加重问题的严重性的时候，我们就不能再回避问题。与其把问题抛给成套软件商，不如自己开始反思和做出坚决的决定，重点不是外购的软件不够好用，而是我们自己没有自主权和不懂这些东西，如果一旦我们有勇气迈出这一步，那么在数字化转型的路上，我们将会有一个新的起点。这一点认识，需要大家达成共识，至少需要高层中最有权力的一群人认识到这个问题的所在，否则各个部门之间互相指责和推诿必然会导致数字化转型的失败。

所以，如果你想成为一个软件构建者，需要从改变整个组织的思维方式开始。

这句话是《数字化思维》作者说的，我深感振聋发聩。

想要彻底的走上自我构建之路，需要自上而下给与关注和重视，需要一种真正鼓励尝试和创新、不怕失败的文化，需要大规模的宣传，让变革的思维像春风吹过冬天的原野一般，否则数字化的尝试便只能像一滴水滴在沙子里，一棵树种在戈壁滩，无助而没有前途。

到底是构建还是购买

单纯说构建还是购买没有意义，因为没有上下文。

有的东西需要构建，有的东西需要购买。

扮演乔布斯的演员和硅谷著名风投，阿仕顿·库彻说，你选择自己不构建的东西和你选择构建的东西同样重要。公司应该唯一选择自行构建的，是其业务的核心。很多时候，人们最终选择的系统在市场上已经有了现成的产品，并且可以以较低的购买成本获得使用权。

作者说，如何构建的东西能够使得你面对客户实现差异化并且获得竞争优势，你就应该自己构建。

但是自己构建的话，也是需要花费巨大的时间和成本、人力的，所以我们需要借助数字化供应链来提高自己的开发效率。

和传统的行业一样，软件开发也需要数字化供应链。我们无法想象自己从0开始编程，这样将无法形成我们真正的东西和优势，所以我们需要尽量的采购。

自己采购

我们到底需要干啥?

我现在受到了极大的鼓舞，我想云计算中的SAAS可能远远超过我的理解，它能做的事情更多，它让我们关注于我们真正唯一想关注的东西。

SAAS应该提供尽可能简单的操作模式。

体验saas

按量付费是非常好的建议，我觉得真的很棒。我应该体验一下。

API经济听上去是一个非常振奋人心的事情，比如作者的公司特维利奥的通信API，比如微信支付的支付API。

如果一个团队对API的更改会破坏其他团队正在处理的代码，显然他们无法真正独立的工作，解决办法就是将代码和团队打包在一起，这就是微服务。微服务的根源是协调的复杂性导致增长的缓慢变化。

小团队服务的价值衡量问题引来了下一个问题，即内部结算或者转移支付。如果你的服务让下游不舒服，那么你的服务质量会被通过使用次数、评价等暴露出来。

小团队服务引发的另一个质变是外部分包，即可以选择外部更加有优势的社会团队和服务、资源来满足自己的需要，而非只能在内部自选。所以，基于此种逻辑，把自己的服务做到极致并且出售，也是一门生意。

数字化供应链的明星

SalesForce和亚马逊云是数字化供应链的明星公司。

他们的共同特点就是成为了数字化时代的电力公司类型的基础设施提供者。他们赌对了未来。

亚马逊云的出现，与他们公司业务和公司治理有关，从电商业务多余机器的对外租用，到一个披萨团队，微服务，内部结算，最后到外部服务，即API生意。

SAAS和API经济

API经济带来了采购权的一个问题，即一线开发人员能不能自主采购自己最喜欢的工具，而非经过公司采购批准之类的?

这是个问题。

这本书的作者认为，应该直接去问开发人员该买什么服务。这个我非常支持。

实际上，现实是，可能我们一线生产者没有采购决定权，但是最好我们要尊重一线开发人员的意见。

市场调研-私有云市场的进展

不知道私有云怎么样了?

以客户为中心-发掘市场与需求

以客户为中心的困难

以“客户为中心”这句话如此的流行，以至于大家都会经常提起，并且标榜自己是这么做的，然而实际上，这个愿望常常难以实现。

仔细分析这句话，这句话毫无疑问是一句非常主观的话，你可以把这句话挂在墙上，也可以刻在什么其他地方等等，人们每天都会看到，然而却因为经常听经常见而选择性的忽略。也就是说，“以客户为中心”最大的问题在于，没有具体的指导规范告诉我们如何落地。

调查客户

做一个信息化项目最重要的是要弄清楚客户的需求。一般的模式就是，把一个项目分成若干个阶段，在开始的某一个阶段，就是收集用户的需求。

收集客户的需求存在若干个困难。比如，收集用户需求时，向哪些用户收集需求？这个圈的划分存在一定的难处。愿意参与配合需求调研的用户不一定是典型用户，反而有可能是乐意帮忙的老朋友用户。于此同时，需求调查的结果往往与最后一次互动息息相关，即某一个用户某一时刻的需求。用户需求的收集还往往受制于办公室政治的影响，即权力结构和实际使用的用户之间的矛盾。还有比如，收集用户需求往往仅限正式的几次，或者简单的通过过于正式的电子邮件等等形式，用户真正的诉求难以被发掘。

真正的以客户为中心

以客户为中心的真正含义是，创建一个不断自我修正的组织，让客户成为我们决策的中心。实际运作过程中，会有各种各样的干扰因素降低客户为中心决策的优先级，因此需要这个组织的运作方式充分考虑这些因素。

以客户为中心重点在于，让用户感觉到有温度。即从各个方面提供无微不至的关怀，让用户感觉到，我们是替他们考虑的，着想的，也就是，让用户感受到被以自己为中心非常重要，而不是仅仅提供一些冰冷的流程，比如服务时的热情、即时响应，但是不解决问题，这个不是以客户为中心（所以真正的落实以客户为中心不容易）。

以客户为中心就是要认真的倾听客户的诉求。比如海底捞的服务是除了名的好，它们会安排专门的服务员在桌旁听任你传唤，会主动问你需要什么，或者主动为你提供什么。在软件开发中，这种模式难以落地，因为我们没有那么多人，但是我们依然要想办法随时倾听用户的需求，比如能够快速看到用户需求并且和用户能够快速互动的论坛系统。如果没有办法建立论坛系统，那么提供了简陋的邮件联系方式的时候，在用户发出问题的时候，尽快详细的回复也是提升用户积极性的方式之一。

不同角色和客户的距离

在一个小的创业团队里，所有的员工都能够与客户保持密切的联系，因此上，创业公司可以快速的响应市场和讨得用户的欢心。

随着公司的壮大，或者事情的增多以及管理方式的演进，往往与客户沟通成了部分人的专职工作，比如项目经理或者产品经理，以及销售等等，然后后台人员负责落实用户的需求。

这种方式有好的一面，即专人专事，可以提升效率。但是也有不好的一面，即用户的需求会被过滤，评审，屏蔽或者扭曲。比如，产品经理可以评判某个需求不合适，然后拒绝这个需求，但是实际上，这个需求可能很容易实现，而且对于用户的体验有巨大的正反馈。

如果让所有的用户都能直接接触用户的需求，有一种方式是开会。然而开会也有非常大的弊端，比如开会的时候，对某一个事项的讨论花费过多的时间而无法取得进展，比如开会的时候没法真正的讨论参与，只能旁听，比如突然被邀请参会而不知道上下文，有些事情靠我们主动可以解决，有的事情则需要大家一起努力。直接听取用户的需求效果很好，但是会议的效率和效果具体如何，也非常需要在实际过程中精心设计才能取得最佳。毕竟时间就是金钱，如果参会了却没有收获也会打击参与者的积极性。

具体措施

以客户为中心过于主观和空泛，因此具象化它可以促进它。

比如，近距离观察客户的工作，感受客户到底怎么在使用我们的产品，而不是靠我们自己去想象。有的时候，近距离观察用户怎么使用我们的产品或者交付品，得到的结论会大到让我们意想不到，从而让我们重新考虑和设计我们的产品，真正做到以客户为中心。

比如，想法子去获取并且快速满足用户的小需求，赢得用户的认可和喜爱。小的需求容易满足，虽然看上去不重要，实际上，被满足之后，用户往往会得到被尊重的感觉和快速被回应的感觉，比起等待大的需求漫长的被满足的过程，时不时的快速满足一个小的需求可以消解用户的不耐烦。想要获取用户的小需求往往需要提供给客户一种方便快捷，低成本的途径来联系到我们，甚至帮助客户之间彼此联系，交流和讨论，形成一种社区，从而增加认同感，这是很多成功的产品所做的。

直接获取用户需求还可以省去计划管理上的成本。有的时候过度的筛选，评审需求，反而不如直接立刻把需求做出来，当然，这个需要我们有优秀的技术团队和长期的积淀。

直接获取用户需求并且快速落实用户的需求，我们还可以获得免费而威力巨大的活体移动广告牌，毕竟一个认可我们的客户是非常乐意帮我们宣传的产品的。

使用我们自己的产品至关重要。我们必须亲自作为用户去使用我们自己的产品。只有自己使用自己的产品，才能切身的体会到自己的产品是不是如自己所期望的那样工作，才能与客户共情，内心明白，这东西到底好不好用等等的。因为我们的客户不知道产品背后是怎么实现的，他们只关注于实际使用体验和价值，所以完整的真正的当一次用户，从用户的视角体验自己的作品，必然会让自己足够重视和重新反思。

如果“人人都是销售”，不一定真的每个人都要做销售的全部工作，但是应该做一部分，比如亲自去拜访客户，包括像销售一样定期去走访客户，询问其使用效果和体验，这样做可以显著带来以下好处：1.用心对待自己的作品，因为你面对客户是有可能被提问的，因此越了解自己的作品显得尤为必要，包括从全局到局部，2.促进和客户的关系，人和人的实际接触是提升亲密度最直接有效的方式之一，当面拜访可以表明，“我是重视你的”。3. 见证自己的作品并且与客户共情，只有真正的见到客户，看到它产生实际价值了，帮助客户解决问题了，才会内心更加热爱自己的工作和产品，防止在过细过长的分工链条中迷失自我，对工作麻木和失去兴趣。在与客户的接触过程中，正如上文所说，才能真正知道客户遇到了哪些问题，是怎么用我们交付的产品的等等。

产品的两个关键设计

便利性设计

软件的优势之一在于便利性设计，在【第一部分构建和购买的选择】中已经提到。

很多人喜欢看短视频，不是因为他们觉得短视频有多有营养，只是因为短视频方便。

想要让用户对我们的软件上瘾，就必须设计的足够好用。

虽然让用户上瘾的动机非常的不高尚，但是很多公司这么做了，并且赚了很多钱。

我们做一个平衡，就是让我们的设计为用户提供最大的便利，为用户创造真正的价值，比如节省时间，提高效率等等，简直正向有效的工作，也能创造价值。

基于反馈的设计

用户要什么，用户喜欢什么，用户抱怨什么，用户可能喜欢什么，我们都要认真的研究。

程序员的素养

书中提到了 编程马拉松，也许多数程序员应该参与一下。

作者的成长经历

非常朴素的针对生活周围的需求的挖掘，持续的创业，非常强的执行力等等。

关键词有，创投，互联网泡沫，因为和主题不绝对紧密，我不展开讲了，但是线上读书会的时候可以多说说。

作者有一句话很打动我，我想把我的时间花在建设自己发自内心热爱的东西，而不是销售门票。

加入亚马逊是作者的一个让他印象深刻的经历，他加入亚马逊云的时候，那只有30个人左右。

作者所赞赏的"问你的开发人员"理念源自于亚马逊的工作，这是他受到的管理启蒙。

专注

是的，写代码，你必须尽量努力专注。

对创造充满热情

作者的成长经历，即他和一个商务人士的无间合作让他如鱼得水。

遇到需求，他总会动脑子去实现，而不是说，哦，这个实现不了，你想办法吧。我觉得大家都会喜欢这个这种肯动脑子的人。

激励开发人员

了解开发人员，才能激励开发人员。

其实开发人员不是书呆子，他们充满了创意，他们能创造很多有意思的东西。

只要环境允许，他们的创意就会发芽。

这些环境包括足够的自由和责任，包括工作时间上的自由，穿衣打扮的自由，上班地点的自由等等。

提出问题而非指定解决方法

这样做，能够真正的解决问题，即让解决问题的人完整的去负责这件事。