

02 | 以终为始：如何让你的努力不白费？

2018-12-28 郑晔

10x程序员工作法

[进入课程 >](#)



讲述：郑晔

时长 11:57 大小 10.95M



今天内容的开始，我希望你可以先来思考一个问题：**如果让你设计一个登录功能，你会怎么做？**

我曾在公司内部做过这样一个练习，我扮演客户，让大家帮我设计一个登录功能。同事们一听就高兴了，登录不就是用户名加密码嘛，我熟啊，我还可以设计出验证码、找回密码、第三方登录等等功能。

更有个别动作快的同事，甚至已经开始设计数据库表，考虑用 Redis 做缓存了。整个过程下来，大家彼此讨论得热火朝天，唯一没人理会的就是我这个“客户”。

讨论结束，扮演客户的我告诉大家，作为一个“土豪”，我打算做一个打车软件，用户可以通过手机号接收验证码的方式进行登录。你可以想见，同事们一副“被套路了”的表情。是

的，他们设计那套用户名密码登录完全是文不对题。

虽然这是一个简单的练习，但反映的却是我们日常面对的真实工作场景：许多人都是刚刚听到别人要求做的一个功能，就开始脑补接下来的一切。导致的结果，就是付出的努力毫无意义。

那么问题出在哪呢？因为我们欠缺了“以终为始”的思维习惯。

一种反直觉的思维方式

以终为始，就是在做事之前，先想想结果是什么样子的。

说起来很简单，但做到并不容易。因为我们习以为常的思维模式是线性而顺序的，第一步做完，做第二步；第二步做完，做第三步。

这也情有可原。我们人类都是从远古时代演化而来，在那个食不果腹的时代里，倒着思考的用途并不大，人们甚至不确定自己能否见到明天的太阳。几十万年的进化留给我们很多短视的行为和思考习惯，因为这样的做法最为节省能量，把目光放长远是需要额外消耗能量的。

“以终为始”是一种反直觉的思维方式，是大多数人不具备的。所以，日常生活中，我们看到很多有趣的现象。

比如，大学毕业时，有很多人想考研，如果你问他为什么要考研，得到的理由通常是为了找个好工作。但考研真的能帮他找个好工作吗？不一定，因为找工作和考研根本就不是同一棵技能树。

如果真的是想找个好工作，那你就应该了解工作的要求是什么，怎样才能掌握工作要求的技能。

从后面这个角度出发，你会发现考研只是通往工作诸多道路中的一条，其他的路径也是可以到达的。比如，你应该找个实习的地方锻炼一下职业技能。这就是“以终为始”思考问题的方式。

回到前面“设计登录功能”的例子，对比“以终为始”的思维，你也许会替我的同事抱不平，他们或许也有“以终为始”的思路，只不过，他们的“终”和我这个客户的“终”不一样罢了。这就要说到做软件，本质上是在构建一个“集体想象”。

想象的共同体

如果你读过尤瓦尔·赫拉利的《人类简史》或《未来简史》，有一个说法你一定不陌生：想象的共同体。作者认为，人类历史发展的一个重要因素是“集体想象”，无论是国家、宗教，还是法律、习俗，都是人们达成的“集体想象”。人类就是认同了这些“集体想象”的一个共同体。

我们这些做软件的人其实就是一个想象的共同体，这个“集体想象”就是我们要做的软件，任何想象都需要一个载体将其展现出来，我们编写软件的过程就是将这个“集体想象”落实的过程。

既然是“集体想象”，那么在载体将想象呈现出来之前，我们的想象很难统一起来，都或多或少存在差异。

所以，**任何事物都要经过两次创造：一次是在头脑中的创造，也就是智力上的或者第一次创造（Mental/First Creation），然后才是付诸实践，也就是实际的构建或第二次创造（Physical/Second Creation）。**

我们在工作中遇到的很多问题，其实就是在于第一次创造没有做好，就进入到第二次创造。所以，我们在工作中会遇到很多“惊喜”，准确地说，是惊吓。

相比于第一次创造，第二次创造是一件成本很高的事。我们知道，软件开发最费时费力，一旦投入大量精力做出来，却发现与理解偏差甚大，所有人都会欲哭无泪。

所以，在动手做事之前，我们要在第一次创造上多下一些功夫，将相关各方的“集体想象”统一起来。以建筑为例，就是先在图纸上构思各种细节。对应到做软件，我们也可以做很多事，比如：

要给用户看产品的样子，可以用原型工具把它做出来，而不是非得把完整功能开发出来；

要呈现服务接口的样子，可以用模拟服务器搭出一个服务，而不用等后端全部开发完毕；

要让程序员知道要开发产品的细节，可以在任务上描述出软件各种场景给出的各种行为。

再回到前面“设计一个登录功能”的例子，我的同事们在构建的其实是他们自己的想象，而不是我们共同的想象。这其中最大的一个区别就在于，没有人会为他们自己的想象买单的。

所以说，他们看到的“终”不是真正的终，只是一个自我的“终”，至于看到什么样的“终”，这取决于每个人的见识。

对做软件的人来说，我们应该把“终”定位成做一个对用户有价值的软件，能够为别人带来价值，自己的价值才能体现出来。

至此，你对“以终为始”已经有了一个初步的认识，有了这种思维方式，我们可以在工作中怎样运用它呢？

规划和发现

软件行业有很多英雄传说，一个人或者一个团队连续奋战一段时间，写好了一个软件，在上线前夜发现了一个问题，然后冒着“不成功便成仁”的风险，通宵达旦解决了问题，一战成名。

这种故事听起来让人热血沸腾，但仔细想想，为什么总在最后一刻发现问题？除了时间压力确实大的情况以外，大多数情况，他们还是一开始没有想好就动手了。

在团队内部，我一直坚持“以终为始”，让大家在执行任务之前，先倒着想想再动手规划，这样规划出来的工作更能瞄准真正的目标。举一个之前做产品的例子，当年在创业的时候，我们打算做一个物联网开发平台，但具体应该做成什么样子呢？

有了“以终为始”的思维，我们考虑的是别人会怎么用我们的平台。我们设计的方式是，用户到我们的网站，阅读相关文档，然后参考文档一步一步照着做。

这其中的一个关键点是：文档，特别是《起步走》的文档，这是用户接触我们这个平台的第一步，决定了他对我们产品的第一印象。

所以，我们决定从写《起步走》这个文档开始，这个文档描绘了用户怎样一步一步使用我们的开发平台，完成第一个“Hello World”级别的应用。**请注意，这个时候，我们一行代码都没有写。**

写好了这个《起步走》文档，团队的所有人对于我们的平台要做什么样子，已经有了一个比较初步的认识。更重要的是，我们可以拿着这个文档，去和外部的人讨论这个尚未出世的平台。

人类是一个擅长脑补的群体，一旦有人看到了这个文档，他就已经可以构想出这个平台已经存在的样子，进而给出各种各样的反馈：“我认为这个地方可以这样做”“我觉得那个地方可以改改”。

所有这些反馈都是真实的，因为他们已经“看到了”一个真实的东西。正是这些真实的反馈，让我们逐渐地锁定了目标。之后，我们才开始动手写代码。

“以终为始”的方式，不仅仅可以帮我们规划工作，还可以帮我们发现工作中的问题。

有一次，我的团队在开发一个大功能，要将现有的系统改造成支持多租户的系统。也就是说，别的商家可以到我们的平台上发起申请，拥有和我们现有平台一样的能力。

功能来了，各个团队将任务分解，然后就各忙各的去了。但我有着习惯性的不安，总担心丢点什么，于是催着项目经理梳理一下上线流程。

是的，上线流程，虽然我们的代码还没开发完，但是本着“以终为始”的态度，我们就假设各个部分已经开发好了，来想一想上线应该怎么做。

果不其然，一梳理上线流程，我们便发现了问题：怎么识别不同的租户呢？有人给出的方案是设置一个 HTTP 头。但谁来设置这个 HTTP 头呢？没人仔细想过。于是，一个潜在的问题就这样被发现了，至少不用在未来为它加班了。至于解决方案，作为程序员，我们有的是办法。

事实上，在今天的软件开发实践中，已经有很多采用了“以终为始”原则的实践。

比如测试驱动开发。测试是什么？就是你这段代码的“终”，只有通过测试了，我们才有资格说代码完成了。当然，测试驱动开发想做好，并不是先写测试这么简单的。

比如持续集成，我们是要交付一个可运行的软件，倒着来想，最好的做法就是让软件一直处于可运行的状态，那就是持续地做集成。

概括地说，**践行“以终为始”就是在做事之前，先考虑结果，根据结果来确定要做的事情。**

这是“以终为始”这个内容版块的开篇，后面我会给你介绍这个原则在不同场景下的应用，也会引入一些现在行业内的最佳实践进行解析。相信会对你的实际工作有帮助。

总结时刻

有一段时间，网上流传着一个帖子，亚马逊 CTO 介绍亚马逊是如何开发一项产品的，简单来说，他们采用向后工作的方法，开发一项产品的顺序为：

1. 写新闻稿；
2. 写 FAQ（常见问题解答）；
3. 写用户文档；
4. 写代码。

今天我带你了解了“以终为始”的做事思路，回过头再来看这个帖子，相信你不难理解为什么亚马逊要这么做事情了。

人们习惯采用顺序思考的思维方式，几十万年的进化将这种思考模式刻在了我们的基因里。要成为更好的自己，我们要克服自身的不足，而这个做法很简单，那就是“以终为始”，做事倒着想，先考虑结果。

人类是一个想象的共同体，做软件的团队更是如此，而我们写出来的软件是我们将“集体想象”落地的载体。

任何事物都要经过两次创造：一次是在头脑中的创造，也就是智力上的或者第一次创造（Mental/First Creation），然后才是付诸实践，也就是实际的或第二次创造（Physical/Second Creation）。我们应该在第一次创造上多下功夫，统一集体想象，让目标更明确。

“以终为始”的思维可以帮助我们更好地规划我们手头任务，也可以帮助我们发现过程中的问题。

如果今天的内容你只能记住一件事，那请记住：**遇到事情，倒着想。**

最后，我想请你思考一下，在实际的工作或生活中，你有运用“以终为始”的思维方式吗？帮助你解决过哪些问题？欢迎在留言区写下你的想法。

感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给你的朋友。

10x 程序员工作法

掌握主动权，忙到点子上

郑晔

火币网首席架构师
前 ThoughtWorks 首席咨询师
TGO 鲲鹏会会员



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

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

上一篇 01 | 10x程序员是如何思考的？

下一篇 03 | DoD的价值：你完成了工作，为什么他们还不满意？

精选留言 (49)

写留言



每天晒白牙

2018-12-28

22

我们在生活中经常遇到这样的场景:许多人听到别人要做的一个功能，然后自己开始脑补后面的东西。导致的结果，付出的努力没有意义。同样在面试的过程中也会发生类似的事，面试官谈到一个点，还没等面试官说完，就开始吧啦吧啦说一堆，然而并不是面试官想要问的。

这里面的问题是缺少“以终为始”思维习惯。...

展开 ▾



Edward

2018-12-29

13

开头的“套路”确实很生动。做技术的人会有这么样的倾向，听到要做一个功能，第一反

应就不是去梳理功能的细节，而是把熟悉的技术（也有是自己业余时间研究的，想在项目里用上）拿出来，粗略地构想解决方案。也是那句话：“手里拿着锤子，看什么都像钉子。”

展开 ∨



jueyoq

2018-12-28

👍 8

当前处于什么水平？

优势：年轻(今年刚毕业)，没对象(充足学习时间)，善于钻研底层原理，对编程有热情，善于总结方法论，技术表达力强(会吹牛逼)

劣势：完美主义，英语差，算法一般，计算机原理不成体系(非科班)，稳定性差，数学功底一般，缺乏工作经验，工作效率低。...

展开 ∨



pyhhou

2018-12-29

👍 7

感觉“以终为始”这样的思考模式其实在生活中经常出现，例如说要做菜给别人吃，得先考虑别人爱吃什么？南方人还是北方人？是否有饮食顾虑？自己会的菜当中哪道菜做出来更吸引人眼球？先是考虑好要做什么，再去买菜，然后做菜；还有就是买一个东西会先想买回去预期有一个怎样的效果，然后再决定买不买；

...

展开 ∨



Y024

2018-12-28

👍 6

“以终为始”，最常见的一个实践就是计划倒排了。先定时间，然后看功能是不是做不过来砍掉一些，人力是不是不够需要补充一些，提前预知规避风险。

展开 ∨

作者回复: 你说得对，倒排时间表不是错，倒排时间表却不调整需求范围和资源配置就是问题了。



yu

2019-02-14

👍 3

延伸出来的思考。

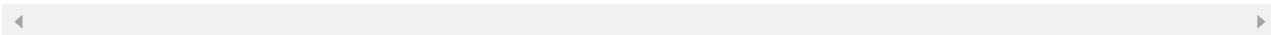
拿到一个需求先写接口文档，评审过了再开发代码。争取做到接口文档就是最终的实现方案，不需要做着做着再去沟通方案。而不是拿了需求文档，直接就写代码。

好处：

1. 写接口文档，可以更好的规划工作量和时间。...

展开 ▾

作者回复: 很好的延伸思考。



王宇泽

2019-01-13

👍 3

程序员版的高效人士

展开 ▾



Alexdown

2018-12-28

👍 3

前段时间遇到一个事，领导派给我一个任务——做一个XXXX功能，没太细说就告诉我：你自己去想，自己去看别人怎么实现的，把他弄出来

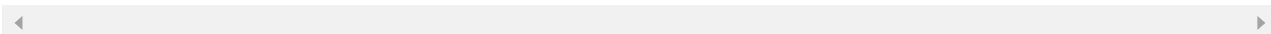
我经过一番调研，这个大功能可以拆分为8个小功能。于是我将4个必备小功能（编号1~4）实现并提交；...

展开 ▾

作者回复: 这个问题很有趣，决定产品特性的究竟是谁，这是需要先明确下来的。如果这是你要做的事，你就在扮演产品经理的角色。你需要做的就是分析产品特性，找用户了解使用场景，而不是闷在办公室里空想。

即便你只是为领导在做产品，你需要把你们会谈的结果详细地写出来，然后，发邮件让领导确认。一旦确认，你领导就不好拿这事说事了，因为你有证据。

如果一个领导连自己的话都不认，你需要考虑一下与他合作的前景了。



David Mao

2018-12-28

👍 3

1.开发采用“以终为始”的原则有利于认清目标，提高开发效率。

2. 根据自己的经历，在产品的整个生命周期，业务的部分如销售在前期需求评审及开发阶段介入，会更好的支撑后续的销售工作。实际遇到的情形，产品开发完成后，由于产品没有市场竞争力，导致销售人员销售时遇到很大的压力，站在全局的角度，销售也应提早介入，销售也可以提产品的需求。目前提的持续交付业务（销售）也应是里面的一个环节...

展开 ∨



helloworld

2019-02-15

👍 2

这就是倒逼思维吧！在事情开始前先在脑海中模拟下整个过程，等到真的做起来时就游刃有余了！



喜悦

2019-01-01

👍 2

今日概念：

1. 想象共同体：集体对同一个目标达成的共同想象；
2. 任何事物都需要经过两次创造：一次在头脑中，一次是付出实践；
3. 遇到问题倒着想，再用工具模拟“想象共同体”发现问题；

...

展开 ∨



长满鱼的树

2018-12-29

👍 2

我个人觉得以终为始的意思就是从结果开始倒排开发，先确认明确清晰不含任何二义性的需求和结果，然后再倒推软件的设计架构，再动手编码，在开发过程中不停调整和反馈，最终达成目标

展开 ∨



王维

2019-01-08

👍 1

这确实是一种思维方式，从结果反推前导，这条原则不仅适用于工作，而且还适用于生活。在具体的开发工作中，我也使用过这个原则，例如在做一个功能之前，我会画一个原型图，先让要使用的人确认，如果ok我们就动手开发。

其实我们在生活或者工作中有意或者无意都用到了这个方法。

作者回复: 赞！



geeeee

2019-05-31



工作中以终为始的例子很多呀，例如：新需求要在一周后上线，但是正常情况下5个工作日肯定完不成，为了完成这个目标怎么办，只能要么加班、要么加人。。。

展开 ▾



大力

2019-05-14



以终为始，令我想到了“先订家具再装修”的套路，这样可以避免供电位不合适要返工等情况。



大力

2019-05-12



最近在复习10x程序员，并且在将每一课都做成思维导图，不知道郑老师有没有兴趣读读呢？实在不知该如何上传至极客时间。



春之绿野

2019-04-30



就在看完这篇文章之后，本来有个需求是log上传到s3，我就准备根据这几个字的需求展开工作了，看了这篇文章之后又去跟同事详细讨论需求，讨论为什么会有这个需求，是为了解决什么问题。有很多发现。感谢老师。不过日常生活中想想还真的很少以终为始，大部分时候都是走到哪说到哪。想学习大数据方面的技能，准备去了解大数据都有哪些应用，普及情况以及有哪些要求，贯彻一下以终为始。

展开 ▾

作者回复: 很高兴看到你能把所学运用在工作中。



周孟

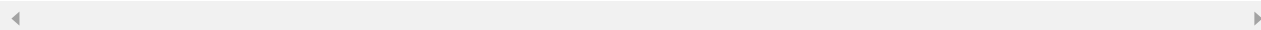
2019-04-24



"以终为始"是不是就是所谓的"结果导向"。以前经常听到说"结果导向"，但不是很明白，现

在感觉有点明白了：无论是哪项工作最好的结果应该都是最终达到相关“共同体”的“集体想象”，所以工作中最优先的就应该去搞清楚这个“集体想象”

作者回复: 抓住重点了！



晓峰

2019-04-07



先搞清楚要做什么，不是想清楚，如果只是想，也许就真正是我们自己的想象，和客户需求南辕北辙，

1.首先看文档，把文档里面不清楚的、模糊的、模棱两可的东西记录下来，看完以后再找客户沟通，把需求理顺最重要。

2.把需要实现的东西想清楚，有哪些细节需要实现？如何实现？有哪些难点？该怎么解...

展开 ∨



enjoylear...

2019-03-27



以终为始这种思维，比如要做用户故事，我们就应该针对最后的界面效果，如何跟第三方交互，涉及的场景如何测试达成共识，经常被某些同事或客户不理解，他们会讽刺说按某网站做就行，或者说跟咱们做过的某某产品哪个功能类似，这还用问和讨论嘛？其实往往最后的结局是变化频繁，改改减减，耗费精力不说，对方还觉得我们效率低下，合作的不愉快。

展开 ∨