开篇词讲程序员解决的问题，大多不是程序问题

你好！我是郑晔，一个程序员。

很多人都说，程序员很辛苦，与这个角色联系在一起的词儿，通常是忙碌、加班、熬夜等。

作为程序员，我们将其看作一个值得全情投入的职业，希望能够把精力放在设计算法、改进设计、优化系统这些具有创造性与成就感的本职工作上。

但现实情况却是，许多人因为一些“意外”，陷入了无休止的忙碌，比如：

- 你辛辛苦苦写的代码还没上线，产品经理就告诉你需求变了；
- 你拼命加班只因错估了工作量，自己造的“孽”，含着泪也要搞定；
- 你累死累活做出来的东西和要求不符，只能从头再来；
- 你大面积地修改代码只是因为设计糟糕，无法适应新的需求变化；
- ……

诸如此类，不胜枚举。我们很辛苦，但耗费我们大量时间和精力去应付的工作，并不是技术工作，反而是这些看似很“不值当”的事儿。

为什么会这样？

软件行业里有一本名著叫《人月神话》，其中提到两个非常重要的概念：**本质复杂度（Essential Complexity）和偶然复杂度（Accident Complexity）。**

简单来说，本质复杂度就是解决一个问题时，无论怎么做都必须要做的事，而偶然复杂度是因为选用的做事方法不当，而导致要多做的事。

比如你要做一个网站，网站的内容是你无论如何都要写的，这就是“本质复杂度”。而如果今天你还在用汇编写一个网站，效率

是不可能高起来的，因为你选错了工具。这类选错方法或工具而引发的问题就是“偶然复杂度”。

作为一个在软件行业奋斗了近二十年的程序员，我深刻意识到一个遗憾的事实：**大部分程序员忙碌解决的问题，都不是程序问题，而是由偶然复杂度导致的问题。**

换句话说，只要选择了正确的做事方法，减少偶然复杂度带来的工作量，软件开发是可以有条不紊进行的。

如何减少偶然复杂度引发的问题，让软件开发工作有序、高效地进行，这正是我希望通过这个专栏帮你解决的问题。

许多人工作做事主要依靠直觉，在这个科学越发昌明的时代，我们清楚地看到，人类的直觉常常是错的，就像古人凭直觉认为大地是平的一样。

软件开发也不例外，如果你不曾在做软件这件事上有过学习和思考，形成一套高效的工作方法，只是凭直觉行事，在真实世界中往往会举步维艰。

幸运的是，总会有不同的人在不同的方向上探索不同的做法，一旦通过真实世界的验证，就会沉淀出可供行业直接应用的最佳实践（Best Practice）。

在软件行业中，这样能够提升工作效率的最佳实践已经有很多，但是，学习掌握这些最佳实践是有难度的，其根源就在于，很难找到这些实践彼此间的内在联系。

直觉大多是错误的，最佳实践又多而琐碎，所以在这个专栏中，**我会尝试给你提供一个思考框架，帮你在遇到问题时梳理自己真正要做的事情。围绕着这个框架，我还会给你一些原则。**

这些原则，是我从软件行业的诸多软件开发最佳实践中总结出来的，也是我如今在工作中所坚持的。这些原则就是一条主线，将各种最佳实践贯穿起来。

这些原则不多，总结起来就四个：

- 以终为始；
- 任务分解；
- 沟通反馈；
- 自动化。

也许看到这四个原则的名字，你会不以为然，这些说法你在很多地方都看到过，但我想与你分享的内容可能与你想的并不完全一致。

比如：你以为的“终”可能不是终，因为你只是站在自己的角度；你以为自己做了任务分解，在我看来，可能还不够，因为我希望你能够做到微操作；你以为的沟通反馈就是说话聊天，我想告诉你很多技术实践的存在也是为了沟通反馈；你以为自动化就是写代码，我会告诉你，有时候不写代码而解决问题，可能才是一个好方案。

在我看来，想要将精力聚焦在本质复杂度上，提高工作效率，摆脱直觉的束缚，只要掌握上面的四个原则就可以了。

或许你此时会问，这些原则很难吧？其实并不难，在探讨这个专栏的内容时，我的编辑作为软件开发的局外人，经常发出感叹：“这事真的就这么简单吗？这不就是正常做事应该有的逻辑吗？”

是的，就是这样简单，但大多数人没有这样做，因为这些原则在实际工作中很可能是反直觉的。只要打破思维误区，你的整个人都会变得不一样。

下面是整个专栏的目录，我希望能帮助你回答，或者厘清一些开发过程中，曾经遇到，又未曾深入的问题。

《10x 程序员工作法》专栏目录

任务分解	13. 先写测试，就是测试驱动开发吗？ ——测试驱动开发：一种设计挑战
	14. 大师级程序员的工作秘笈 ——任务分解：按部就班工作的前提
	15. 一起练习：手把手带你拆任务 ——任务分解实战：每一步都要可提交
	16. 为什么测试很难写？ ——测试的属性：A-TRIP
	17. 程序员也可以“砍”需求吗？ ——需求的拆分：用户故事
	18. 太多人给你安排任务，怎么办？ ——优先级管理：做重要的事
	19. 如何用最小的代价做产品？ ——最小可行产品：找到一条可行路径
	答疑｜任务分解模块热点问题答疑
沟通反馈	20. 为什么世界和你的理解不一样？ ——信息论的视角看沟通反馈
	21. 你的代码为谁而写？ ——用业务的语言写代码
	22. 你总是在开会吗？ ——团队的沟通：轻量级沟通
	23. 可视化：一种更为直观的沟通方式 ——谈可视化沟通的关键点
	24. 为什么你们公司总是做不好持续集成？ ——持续集成的关键：快速反馈
	25. 开发中的问题一再出现，应该怎么办？ ——回顾会议：复盘与改善
	26. 作为程序员，你也应该了解用户 ——用户思维，聆听来自用户的声音

	27. 为什么被指责的总是你？ ——把事情做在前面：变被动为主动
	28. 写文档、做分享，也是一种学习方式 ——让自己理顺思路
	答疑丨沟通反馈模块热点问题答疑
自动化	29. “懒惰”应该是所有程序员的骄傲 ——想懒惰先勤快
	30. 一个好的项目应该是什么样？ ——构建脚本：让日常开发变得更简单
	31. 程序员怎么学习运维知识？ ——一个思考 DevOps 的框架
	32. 有了持续集成就够了吗？ ——持续交付：一种延伸的“持续集成”
	33. 如何做好验收测试？ ——站在用户的角度看测试
	34. 你们的代码是怎么变混乱的？ ——单一职责：划分界限
	35. 总是在说MVC分层架构，但你真的理解分层吗？ ——分层思维，是计算机的核心理念
	36. 为什么总有人觉得5万块钱可以做一个淘宝 ——不同量级的东西不是一回事
	37. 先做好DDD再谈微服务吧，那只是一种部署形式 ——领域驱动设计：限界上下文
	答疑丨自动化模块热点问题答疑
	38. 新入职一家公司，怎么快速进入工作状态？ ——找到关键点，快速上手

综合运用	39. 面对遗留系统，你应该这样做 ——用思考框架应对遗留系统
	40. 我们应该如何保持竞争力？ ——不断提升自己的核心优势
	答疑｜综合运用模块热点问题答疑
结语	少做事，才能更有效地工作

当我们详谈这些原则时，我会给你讲述一些最佳实践，让你看到这些原则是如何应用于不同的实践中的。希望我对这些实践的理解成为你的知识地图，让你拥有继续探索的方向。

我做这个专栏的原则是“授人以鱼，不如授人以渔”。我希望你很好地理解这些原则，掌握高效工作的方法。至于最佳实践，你可以自行决定，是直接采纳还是曲线救国更为合适。

介绍一下我自己，我是郑晔，目前在火币网担任首席架构师，写过代码、带过团队、做过咨询，创过业，还维护着一个拿过 Oracle Duke 选择奖的开源项目 Moco，至今仍然在编程一线写着代码。

很长时间里，我一直对**如何做好软件**充满了好奇，了解过各种技术以及开发方法。做咨询的经历让我有机会见识到不同公司面临的问题；带团队的时候，我也看到很多小兄弟因为不会工作，虽然很努力却收效甚微；而我自己菜鸟时期的笨拙依然是历历在目。

在我看来，所有做软件的人能力都很强，这些问题都只是因为不会工作造成的，但更可怕的是，许多人深陷泥潭而不自知。

在这些年的工作里，我一遍又一遍给别人讲如何工作，逐渐总结出一套自己的工作原则，如今呈现在你面前的就是我这些年思考的总结。

我不指望所有人都能从这个专栏受益，我只想把这个专栏写给那些愿意成长的人。我只是来做一次信息分享，分享一些思考，分享一些做法，希望可以将你从常见的思维误区中带出来。

也许在这个专栏的最后，你发现自己并不认同我的原则，却能够用自己的原则来与我探讨，那么，恭喜你，因为那是最美妙的事情！

精选留言

风含叶
老师：
您好！
已经购买了这堂课程。同时想咨询一下：
我是一只，刚刚工作一年多的本科生。现在在一家500人左右的公司工作。工作中充当研发+技术支持的角色。事情总是被突然打断。请问您可以简单说一下 您的一些建议吗？谢谢。
2018-12-24 19:58

水有罔象
期待
2018-12-24 17:23

felix
这个专栏太棒了，不客气地说，跟我十年工作之后的体会一模一样。平常我也是跟团队教导这四个原则。
努力很重要，方向和方法更重要。我们要付出至少90%的努力，同时达到120%的效果。
2018-12-26 22:52

作者回复
欢迎你把自己的经验也分享给大家
2018-12-27 09:32

AlphaGao
直觉在某种程度上也是很重要的吧，不是很多专家都是很依赖直觉的么
2018-12-25 22:10

作者回复
这是一个好问题，这种说法混淆了直觉和洞见两件事。直觉是本能的，无需积累，而洞见是溯因推理（Abductive Reasoning），需要前期进行大量的积累之后，从中发现模式，方能形成洞见。

打个比方，同样的思念，可以说，衣带渐宽终不悔，为伊消得人憔悴，也可以说，我想死你了。二者看上去差不多，但境界有差异。
2018-12-26 08:03

爬虫也疯狂

感觉我才大一好像买早了，不过还是希望大佬们多多指教，以后少走弯路

2018-12-24 21:58

妮可

作为开发，要谈需求，写需求，以及日常业务数据处理，有的时候还要充当客服，确实有点迷茫。希望能通过学习大佬的经验，回到开发本质上，提升工作效率。期待(๑˙ー˙๑)

2018-12-24 20:30

作者回复

跨角色是件好事，会让你有更多的视角，这是我鼓励很多人去尝试的。只要把握住自己的核心能力，不断提升就好。

2018-12-24 21:19

唐堂@贝壳找房

期待后续文章~解放自己的劳动生产力

2018-12-24 18:56

二木又土

请教下，对于一个明确的技术点，优秀的程序员仿佛能更快的找到解决方案，而且往往就是最佳实践，这是什么原因？从技术点角度讲，它并不需要掌握其他相关知识…

2019-01-08 07:12

作者回复

你把结果当成了原因，优秀的程序员能够快速解决问题，是因为它已经做了大量积累，有自己的知识体系，任何领域想做到一定的水准都需要刻意练习，而且是大量的刻意练习。

刻意练习，是一个重要的概念，我本打算在这个专栏里讲一下，后来由于主题的关系，暂时拿掉了，看后面是不是有机会专门加餐讲一次。

2019-01-08 10:09

王小勃

打卡

2018-12-28 00:51

杨溢

已购买，坐等大佬更新

2018-12-24 19:35

Panda

码出高效

2018-12-24 19:00

davidce

t.cn/EGeYIAA，一个讲本质复杂度和偶然复杂度的视频

2019-01-09 22:03

雷小鸿

软件开发自己的定势想象和集体想象的矛盾。

2019-01-07 12:38

雷小鸿

很多定势思维和集体想象学到啦。

2019-01-07 12:36

休息一下马上回来

跟着我们的郑老师好好学习

2019-01-02 12:34

超

聚焦本质复杂度，降低偶然复杂度，可通过以下几个原则降低偶然复杂度：

1.以终为始

2.任务分解

3.沟通反馈

4.自动化

2019-01-01 20:58

阿狸爱JAVA

大部分程序员忙碌解决的问题，大部分都是由偶然复杂度导致的问题，因此提出

* 以终为始
* 任务分解
* 沟通反馈
* 自动化

四个原则来减少偶然复杂度引发的问题，提高软件开发效率

2018-12-31 14:32

Monday

我目前就是工作效率很低的0.15*程序员。

做事经常确实方法论，蒙着头一顿猛搞，为做而做。

希望通过本课程至少达到5*程序员目标，希望自己能够有清晰的思维，谢谢老师

2018-12-31 08:52

Ruhm

很有价值的专栏。之前的工作中或多或少对工作方法做了一些思考，但是都不成体系，没有形成自己的原则，希望通过这个专栏的学习，总结出自己的工作方法论。

2018-12-29 13:05

作者回复

希望在过程之中，看到你的方法论逐渐成型。

2018-12-29 20:15

leeheol

嗯，当你判断某人的建议没有的时候，对方的建议其实在起着作用。

2018-12-29 09:48