

1975年，弗雷德里克·布鲁克斯（Frederick Brooks）出版了软件行业的名著《人月神话》，他给出了一个统计结果，优秀程序员的开发效率是普通程序员的10倍。40多年过去了，这个数字得到了行业的普遍认同。

在接下来的这段时间里，我希望通过这个专栏和你一起探讨，作为一个程序员，该如何更高效地工作，怎样才能把时间和精力尽可能地放在处理本质复杂度的事情上，减少在偶然复杂度上的消耗。

一个思考框架

- 我现在是个什么水平？
- 我想达到一个什么水平？
- 我将怎样到达那个目标？

更新请加微信1182316662 众筹更多课程162

处于什么水平？和有经验的人相比，他们大多自认为比较“菜”。但对于后两个问题的讨论，却可以切实看出人和人之间处理问题的能力差异。

有人通过之前的资料搜集，已经对自己的未来有了一个打算。比如想成为一个研发大牛，或者想做一个开源软件等，也就是说，对于第二个问题，他有明确的答案。

而有的人则是一脸茫然，他很可能根本没有考虑过这个问题。而从题目本身来看，**目标相对清晰的同学，才会进入到第三个问题，而茫然的同学，则完全无从下手。**

那么我为什么会问这几个问题呢？我是想让大家跳出现有的思考模式，摆脱仅凭直觉“闷头做事”的习惯方式，把低着的头抬起来，看一眼未来，给自己找一个方向。

否则，如果你对未来没有定位，是茫然的，尽管你也知道要努力，但劲儿该往哪里使呢？如果使劲的方向不对，那么你越使劲儿，可能会在错误的路上跑得越远。南辕北辙的道理大家都懂，但具体到自己的工作和发展上，真正能体会并实践的却是少数。

其实，这三个问题来自一个思考框架。在给其他公司团队做咨询时，我也经常会运用到它，原来的问题是：

- Where are we?（我们现在在哪？）
- Where are we going?（我们要到哪儿去？）
- How can we get there?（我们如何到达那里？）

这三个问题实际上是帮我们确定：

- 现状；
- 目标；
- 实现路径。

如果一个人能够清晰地回答出这三个问题，通常意味着他对要做的事有着清晰的认识。这个框架虽然看似简单，但却非常有效，它已经成为我工具箱里一件非常称手的思考工具。

在我的职业生涯里，与很多人讨论不同的事时，我都会用到这个思考框架的不同变体，而在这个专栏里，我也会用它来帮助回答“怎样高效工作、怎样做好软件”这件事。

四个思考原则

在实际的工作中，这个思考框架会帮助我更好地了解自己的工作。比如，当一个产品经理给我交代一个要开发的功能特性时，我通常会问他这样一些问题：

- 为什么要做这个特性，它会给用户带来怎样的价值？
- 什么样的用户会用到这个特性，他们在什么场景下使用，他们又会怎样使用它？
- 达成这个目的是否有其它手段？是不是一定要开发一个系统？
- 这个特性上线之后，怎么衡量它的有效性？

如果产品经理能够回答好这些问题，说明他基本上已经把这个工作想得比较清楚了，这个时候，我才会放心地去了解后续的细节。

我们用思考框架对照一下，为什么我会问这些问题。一般来说，一个新特性要开发时，现状我是知道的。所以，我更关心目标，这里“为什么要做这个特性？”就是在问目标，“给用户带来怎样的价值”是在确定这个目标的有效性。

接下来，我会关注实现路径，用户会怎么用，是否有其他的替代手段，我需要了解产品经理的设计是经过思考的，还是“拍着

脑袋”给出的。衡量有效性，则是要保证我的工作不会被浪费。

通过这个例子，我给你展示了怎么用这个思考框架提出问题。但我估计你更想了解的是，我怎么会想到问这些问题。**给出思考框架是为了让你明白为什么要提出问题，而具体问题要怎么问，就可以遵循下面这四项原则：**

- 以终为始；
- 任务分解；
- 沟通反馈；
- 自动化。

这是我从思考框架延伸出来的。在这个专栏里，我会围绕这四项原则和你详细讨论。

解释一下，**以终为始**就是在工作的一开始就确定好自己的目标。我们需要看到的是真正的目标，而不是把别人交代给我们的工作当作目标。你可以看出这个原则是在帮助我们回答思考框架中，Where are we going?（我们要到哪儿去？）这个问题。

任务分解是将大目标拆分成一个一个可行的执行任务，工作分解得越细致，我们便越能更好地掌控工作，它是帮助我们回答思维框架中，How can we get there?（我们如何到达那里？）的问题。

如果说前两个原则是要在动手之前做的分析，那后面两个原则就是在通往目标的道路上，为我们保驾护航，因为在**实际工作中，我们少不了与人和机器打交道。**

沟通反馈是为了疏通与其他人交互的渠道。一方面，我们保证信息能够传达出去，减少因为理解偏差造成的工作疏漏；另一方面，也要保证我们能够准确接收外部信息，以免因为自我感觉良好，阻碍了进步。

自动化就是将繁琐的工作通过自动化的方式交给机器执行，这是我们程序员本职工作的一部分，我们擅长的是为其他人打造自动化的服务，但自己的工作却应用得不够，这也是我们工作中最值得优化的部分。

这四个原则互相配合，形成了一个对事情的衡量标准。总体上可以保证我的工作是有有效的，在明确目标和完成目标的过程中，都可以尽量减少偶然复杂度。

我们现在在哪？	我们要到哪去？	我们如何达到那里？
你很清楚	以终为始	任务分解
		沟通反馈
		自动化

怎么把这四个原则用在工作中呢？我们回过头来看一下前面的场景，产品经理把要做的功能特性摆在我面前。站在**以终为始**的角度，我需要了解真正的目标是什么，所以，我会关心**为什么要做这个特性**。为了保证目标是有效的，我会关心**它给用户带来**

的价值。

有了**任务分解**的视角，我需要将一个大的目标进行拆解，如果我要达成这个目标，整体解决方案是远远不够的，我需要把任务分解成一个一个小的部分。所以，我会关心一个一个**具体的使用场景**。

一方面，我会了解到更多的细节，另一方面，当时间紧迫的时候，我会和产品经理来谈谈究竟优先实现哪个场景。

为什么要学会**沟通反馈**？因为我需要明确，自己是否真正理解了产品经理提交的需求。所以，我要不断地问问题，确保自己的理解和产品经理交代的内容一致。

另外，我也需要保证我的产品做出来确实能够达到目标。所以，我会关心它**上线后的衡量手段**。因为我知道，这个行业里有太多代码上线后，从来没有运行过。

自动化的角度很有意思，**我们做的方案通常是一个自动化方案，但我们需要了解这个方案没有自动化之前是怎么做的**。如果不自动化，用户会怎么用。所以，我会关心是不是还有其它替换方案，比如，买一个现成的服务。因为很多需求的提出，只是因为我们有了一个开发团队而已。

好，现在你已经对这四个原则在工作中的应用有了一个直观的认识。但你也会发现，我问的这些问题似乎已经“超纲”了，超过了一个普通程序员应该关注的范围。但这就是真实世界，它不像考试一样，有一个标准答案。

我们不是一个人孤独地在工作，而是与其他人在协作，想要做到高效工作，我们就要“抬起头”来，跳出写代码这件事本身。所以，我在开篇词里说，程序员解决的问题，大多不是程序问题。

可能你对这些原则的了解还没过瘾，没关系，这篇文章只是让大家清晰地了解思考框架和原则的背后逻辑。接下来，我会结合行业里的最佳实践，给你进一步讲解这些原则和具体应用。

总结时刻

总结一下今天的内容。大多数人工作低效是由于工作中偶然复杂度太多造成的，只要能够更多地将注意力放到本质复杂度上，减少偶然复杂度造成的消耗，我们“真实”的工作效率自然会得到大幅度提升。

而想要减少偶然复杂度的消耗，就要了解一些高效的工作方式和行业的最佳实践，而这一切是可以用统一的框架进行思考的。

运用这个思考框架，我们需要问自己一些问题：

- Where are we?（我们现在在哪？）
- Where are we going?（我们要到哪儿去？）
- How can we get there?（我们如何到达那里？）

为了把这个框架应用在我们程序员的工作中，我给了你四个思考原则：

- 以终为始，确定好真实目标；
- 任务分解，找到实施路径；
- 沟通反馈，解决与人打交道出现的问题；
- 自动化，解决与机器打交道出现的问题。

如果今天的内容你只能记住一件事，那请记住：**面对问题时，用思考框架问问自己，现状、目标和路径。**

最后，我想请你思考一下，如果把这个思考框架运用在你的职业发展规划上，你会如何回答这三个问题呢？

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。

精选留言



此方彼方Francis

这个方法论恐怕要建立在开发和产品双方地位比较平等的基础上。

以我们目前的现状，很多时候产品要做这需求的理由就一个：老板要的！

2018-12-26 09:19

作者回复

后面还真有一篇怎么对付顶着老板名义做事的产品经理，敬请期待。

2018-12-26 10:30



李

现状：勉强的架构师、技术管理

目标：架构师、cto

途径：1.理论加强学习：togaf学习理论

2.实践技能应用：netty、redis、mysql、nginx、微服务架构、分布式架构

3.做微信公众号，多思考多积累经验，形成自己的套路，同时形成影响力。

2018-12-26 09:47

codeparser

以终为始的理念还是非常有启发的，具体工作中往往会陷入细节中而看不到目标或者忽视目标，例如把实现路径当成了目标。同时，有了目标还需要不断的反思，孔子日三省吾身，我们起码按季度需要重新总结和考虑一下目标和路径的合理性。另外，文中讲到的沟通和自动化如果更加深入些就好了

2018-12-26 09:29

作者回复

后续内容会针对这四项原则进行深入探讨的，加油

2018-12-26 10:07



David Mao

结合工作中每年都要基于现状制定KPI，但往往忽视了最重要的环节，实现途径或方法。老师讲的逻辑很清晰，简单明了。虽是简单的三个环节，但每个环节要想认识清楚，还是需要很深的功力。总结起来：

1. 认清现状

2. 制定切合实际的目标

3. 切实可行的实现方法。

2018-12-26 08:24

妮可

既是产品经理也是项目经理也是程序员，所以全部是我本职工作，IT部部长要的结果就是我的目标。

2018-12-26 18:43



苏忆

我觉得程序员的优秀程度最能体现在自动化这个环节中，高手首先对自己要到达的目的地的实现细节了如指掌，同时可以用较短的时间转化为自动化工具或者程序。同时超前的视野和丰富的开发经验也不是一个新手能很快拥有的。这两个都可以有效的缩短开发时间。那么问题来了，这些经验视野和高手所拥有的熟练度转化度怎么能更系统的来培养？

2018-12-26 09:04



Demi

计划赶不上变化，怎么办？

2018-12-26 08:47

作者回复

计划太大，变化就可怕，后面很多内容就是在讲拥抱变化。

2018-12-26 09:34



white_fisher

在哪？到哪去？怎么去？

答案：勉强程序员；架构师；知识储备深度广度，应用实践

问题：目的地变更了甚至不存在了怎么办？到达路径走不通或者出现了更好的路径怎么办？

总结：即时反馈，实时调整

2018-12-26 08:08



WTF

老师举得针对拿到产品经理新特性要求之后的灵魂四连问，刨根究底，其实也是为了避免 X-Y Problem，免得浪费无谓的人力和精力。关于 X-Y problem，可以看下隔壁专栏作者耗叔的 <https://coolshell.cn/articles/10804.html>（我保证不是耗叔派来的，哈哈

2018-12-27 23:21



One day

老师的建议和指导都是非常有效，而且“感觉非常可行”，可是实际现实的情况，却有太多意外，我个人曾经在某银行的外包团队中，协同开发一个商户-收银以及集成了各种第三方的团队的接口等等，这个大项目大概有300人以上，由此我得leader不在，我代表开会，足足开了3-4个小时(约30个leader)。杂乱无章，从我个人开发角度无法去高屋建瓴的把握全局应用场景，使用价值之类的。但是每个模块之间互相调用联系，系统升级需要协作。首先，这里沟通成本很高，最要命的是需求变动，无法预测，有的时候新增需求，技术leader直接告诉我在什么地方写一个action，调用一个方法之类的。基本就是码农搬砖之类的，我想提的建议是：希望老师在有丰富理论知识，能给我们有一些实践中的例子，或者列举一些举一反三的例子，理论+实践，我觉得才是最佳法则(没有说老师只是有实际，没有实践，莫怪)，第一次这么长的留言，望采纳，谢谢您了

2018-12-26 14:17

作者回复

你的情况就是工作的上下文不足，这种情况你会很纠结，需要扩大自己的工作上下文，了解更多东西。这个话题后面会讲到。

2018-12-26 18:51



alan

谢谢老师！听君一席话，胜读十年书。

现在：可能还不达标的“五级工程师”(五级工程师理论by吴军老师)

目标：四级工程师

路径：勤学苦练！主动寻求机会！

2018-12-26 10:03



王维

有点不懂自动化，为什么说这一块我们优化得不够呢？

2018-12-26 08:35

作者回复

在“自动化”版块中会进行详细讲解，可以关注后续更新～

2018-12-26 11:49



blueguitar

更新请加微信1182316662 众筹更多课程167



现状:一个合格的member

目标:project manager

实施:具备项目管理技能, 精通业务, 能够提案和打单

2019-01-06 15:25



Xunqf

我们的产品经理的回答是:“你是在拷问我吗? 这个你不需要知道, 做出来就好了。”或者:“先做出来上线看看用户的反馈, 不行了再下掉。”又或者:“这个是老板提的需求, 优先级最高”

2019-01-05 23:16

作者回复

不靠谱的产品经理, 总拿老板说事, 刚好是后面的两个主题, 欢迎关注!

补充一点, 不允许问问题的产品经理一看就是虚的, 如果他不让提问, 那就把问题带到老板面前, 让老板判断一下, 自己问的问题是不是合理, 扩大上下文是解决问题的一个好办法, 对, 这也是后面的一个主题, 敬请期待!

2019-01-06 10:12



张维元

思考框架是道, 原则是演化下的术, 我们从 $A \rightarrow B$, 有无穷无尽的路径, 最有效的唯有那条直线。本质上, 各个维度、原则 (不限于作者提到的四项原则) 都是帮助我们更好地定位 A 在哪里, B 在哪里, 那条直线在哪里。

这个思考工具算是吸收了, 四项原则也是有借鉴意义, 谢谢

2019-01-03 23:10

作者回复

这个理解真棒!

2019-01-04 14:21



超

现状: 中级Java程序员

目标: 架构师

实现路径: 主要从技术和管理两个方面进行修炼提高, 针对每个方面的能力要求, 先制定长中期目标, 然后拆分中长期目标为短期目标, 一个个击破, 同时要检验目标达成效果, 不断优化, 从而实现自己的最终目标

2019-01-01 21:22



阿狸爱JAVA

遇到问题: 运用我目前的现状如何, 最终要达到怎样的目标, 如何达到这个目标这个思考框架。

运用这个框架遵循以终为始, 任务分解, 沟通反馈, 自动化四个原则。

和产品讨论需求时, 多问一句为什么要做这个功能, 这个功能上线要达到什么效果, 是那些客户群体在用, 如何评判这个功能的价值所在

2018-12-31 15:33



MarksGui

目前我们团队产品经理要求做很多功能点, 然而最终跟客户沟通客户关注的根本不是这些点。这让我很苦恼。

2018-12-30 16:55



朱小白RUN

我们存在很多功能都是顶着领导的要求做的, 根本就不会考虑实际的可行性以及使用情况, 导致花了时间精力开发出来的功能可能用过一次以后就再也不用了

2018-12-29 09:57



pyhhou

想问一下老师这里说的这些原则主要运用在自身长期的职业发展, 还是说解决工作中的大小项目, 看三个思考框架可以用在自我的评价以及期望目标上面, 但是四个思考原则像是针对工作中的任务来的, 对如何提高自身能力没有直接的对应, 还请老师指明

2018-12-28 13:24

作者回复

文中的问题就是让你用思考框架中的三个问题回答自己长期的职业发展。

2018-12-28 21:30

