

# 为什么我们程序员写不出好代码？ - 文章 - 伯乐在线



原文出处: [InfoWorld](http://InfoWorld)

译文出处: [CSDN](http://CSDN)

每个开发者在编码时，都希望代码像水管的水一样源源不断的流出，思绪不被打断。但在现实开发中，经常会由于一些事情突然中止或停下。本文就总结了现实工作中，影响编码进度的15个“拦路虎”。



## 1. 开会

停止手头的编码工作，去参加会议。程序员或许不会相信，他们可能已经在会议室花费了数周或数年时间和老板闲聊技术细节。

当你从会议室出来后，大脑可能需要一定的转换时间，才能再重新投入到编码工作上，你很有可能需要一个小时的过渡。

## 2. 回复所有的电子邮件

如果会议已经够糟糕的了，那么没完没了的邮件可能更让人头疼。经过几个小时的来回讨论，最终却没有个结果。

### 3. 衡量生产能力

有些管理团队受到一些书籍启发，对提交到代码库的代码行数或bug修复数进行统计，并且作用一种衡量标准。他们认为，统计是一种衡量，而衡量肯定是有好处的。

然后他们根据此标准为开发者的工作能力进行排名。开发者犹如网络游戏玩家，他们将更加关心自己的排名，而不是如何让代码更好。

开发者的重点变成了统计代码所编写的行数、解决bug或把所统计的提交的仓库里。如果代码行数都计算在内的话，原本一个问题只需10行代码即可解决，程序员有可能编写5000行代码，来让功能更加灵活和兼容，这样，他的代码总量就会增加5000行了。

衡量生产力反而会使代码变的更糟，让项目里充满功能丰富但过度设计的代码。

真正解决这个问题，我们需要跟踪bug，我们需要组织工作流和协调软件开发，这些都是无法准确衡量。

### 4. 爱慕虚荣（Prima donna）的开发者

就开发者而言，最糟糕的莫过于其他开发人员没有按照项目需要进行开发，而是用自己的方式来迭代项目。每一个开发者都能识别出可怕、不可原谅的最后一次迭代行为。

这种不考虑之前已完成编程工作的态度会拖慢项目的进度。傲慢和利己主义会导致程序员扔掉合适的代码，而以他们认为的“正确的方式”重新构建。

### 5. “以后再修复”思维模式即“技术债务”

有时候，我们很难按照需求在数天里完成相应的功能，因此我们可能会偷工减料、补丁代码等。聪明的项目经理在弄清事后必须补上的“债务”后，形象的称它为“技术债务”。

每个项目都会有一定的技术债务，有的可能会快速还清，而有的可能会在下一个版本中初见端倪。

### 6. 非程序员经理

有些程序员很喜欢这样的经理，他们不会对你的代码指手画脚，而且在技术上愚弄他们很容易。而他们也很难给你技术上的指导。

### 7. 程序员经理

虽然程序员可能会抱怨要和完全没有编程经验的项目经理一起工作，但他们也经常私下说如果项目经理具备编程能力可能会更糟糕，甚至有多糟糕就多糟糕。

具有编程能力的项目经理可能会对项目管理的太细，因为他们一旦有新的观点，代码就会大片修改。

### 8. 技术过硬但有些强势的程序员

程序员往往都是因为过硬的技术才被公司赏识，而不是人际交往。但不能每次出现问题都责怪穿西装不自在或销售人员过于热情，有可能问题出现在自己身上。

客户想要一些不同的东西，这对此类程序员来说无关紧要，他们更多的是关注于技术参数。

然而在人际相处中，他们经常会过滤掉彼此的特质，当他们彼此产生争议时，就可能影响到整个团队的进度。

## 9. 自私或莽撞的程序员

自恋狂程序员的工作可以说是非常酷和快，但遗留的问题也会很多，而你的工作就是处理这些琐事，对程序进行测试保证它不会崩溃。

许多团队在发现这一点后，都已经太晚了。在早期的测试中，代码块都可以很好的工作，但在推送一些真实数据后，大家才意识到并没有人检查这一问题。

## 10. 文档不全

有时，这里会有大量的文档，但它可能是几个月前或者一些老版本的记录。我们没有时间继续记录和修复代码，但它对我们来说仍然是有用的。

## 11. 纯粹地编写文档

虽然我们都经历过没有文档的项目，太多的废话和较少的代码常常会导致代码失败。程序员经常会根据需求编写评论，他们很详尽地把每个细节记录在文档里，没有总结或进行深入的理解，但如果没有提供太多的抽象和理解，这很有可能是一份失败的文档。

## 12. 易分散注意力的环境

虽然销售和营销团队能够在具有噪音的环境里很好的工作，而程序员则需要图书馆般那种安静的环境。虽然许多企业给包括程序员在内的员工提供了类似乒乓球这类的运动，但他们常常忘记，程序员需要在安静的环境下办公，否则，嘈杂的环境很容易分散程序员的注意力。

## 13. 办公文化

你想拥有自己的办公室吗？还是你想在可以随时提出你的问题团队里工作？你是喜欢在清晨工作还是熬夜呢？

如果一个团队拥有一个相似的风格，那么这个团队会运营的更好。如果无法找到一个共同点，很有可能会快速失败。

这可能太一概而论了，但你想象下，如果你正在编译或者准备完成项目，而此时团队里的人在互相争吵，你不得不中断下来，这样时间不就浪费了吗？

倘若我创建了一个非常复杂的算法，而中断、谈话、甚至是敲键盘的声音都会使我无法集中精力，这时，我就非常希望有属于自己的办公室。

## 14. 紧随遗留技术

最令人讨厌的莫过于去改写那些尘封已久的旧东西，他们经常会忘记这样做所花费的成本，有些代码是在ASCII之前编写的，意味着你要重新转换输入输出。旧的系统通常会计算空格字符，仅仅是为了弄清其在数据库中是干嘛的，这更要进行转换。

程序员做大量的工作来截图、重新格式化等，而过一段时间后，他们可能会花更多的精力去复制代码，而不是去重新编写逻辑代码。

## 15. 迷恋最新的工具

最新的工具可以给你带来很多乐趣，处于最前沿的程序员总是喜欢修改整个API，并且重写它们，迫使人们不得不修改底层的代码。

当我试图兼顾Python 3.0和Python 2.7两个版本时，尽管Python是一个相对稳定的版本，但我还是感到很烦。

在许多情况下，新的工具都没有得到十足的锻炼。例如，Node.js的确非常快，但只有在你重新学习了关于创建进程时死锁的所有知识后才能做到。利用最新的工具是可以带来很好的结果，但天下没有免费午餐，并且会为此付出足够多的学习成本。

拿高薪，还能扩大业界知名度！优秀的开发工程师看过来 -> 《[高薪招募讲师](#)》

1 赞 3 收藏 [评论](#)



合作联系

Email: [bd@jobbole.com](mailto:bd@jobbole.com)

QQ: 2302462408 (加好友请注明来意)

更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子

[头条](#) - 分享和发现有价值的内容与观点

[相亲](#) - 为IT单身男女服务的征婚传播平台

[资源](#) - 优秀的工具资源导航

[翻译](#) - 翻译传播优秀的外文文章

[文章](#) - 国内外的精选文章

[设计](#) - UI, 网页, 交互和用户体验

[iOS](#) - 专注iOS技术分享

[安卓](#) - 专注Android技术分享

[前端](#) - JavaScript, HTML5, CSS

[Java](#) - 专注Java技术分享

[Python](#) - 专注Python技术分享