



链滴

你的程序员是在努力工作还是在偷懒？

作者：[Vanessa](#)

原文链接：<https://ld246.com/article/1391743870624>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>说的很有道理，特别喜欢“他说对于复杂的脑力工作来说，一天工作 4 个小时以上是完全没有意义且生产力低下的方式。”这句话。</p>

<p>转自：http://www.infoq.com/cn/news/2013/12/are-your-programmers-working</p>

<p>Mike Hadlow 是一位资深软件开发者，同时也是 EasyNetQ 与 Suteki Shop 的作者，喜爱历史科技，是一个技术极客。近日，Mike 就程序员工作效率、工作表现以及工作成果等主题撰写了一篇博客，谈到了我们该如何看待程序员到底是在努力工作还是在偷懒这个问题。</p>

<p>如果人们从事的是体力劳动，那么我们就很容易能够看出他们工作的努力程度。你会直观、清楚地看到他们频繁移动的步伐，流下的汗珠。还会看到他们工作的成果：逐渐升起的砖墙，地上的洞变得越来越大，等等。观察到并奖励努力工作的人是人类的一种很基本的天性，这也是我们觉得耐力运动人着迷的原因之一。不过对于管理技术创造性的员工来说，这种对努力的体力劳动天然的欣赏之情就出现问题。高效的知识工作者常常看起来并不是那么努力工作。</p>

<p>回到 2004 年，那时我还是一名初级开发者，工作在一个大型团队中，主要从事有线电视公司的单与供应系统。就像所有的大型系统一样，这个系统由大量相对比较独立的组件构成，不同的小组负责不同的组件开发工作。模拟电视与数字电视供应系统几乎是完全分开的，由两个不同的小组分别进行开发。</p>

<p>模拟电视团队决定根据 Microsoft Biztalk 的一个早期版本来构建他们的系统。团队有 4 个人以一个来自于微软的团队共同开发并运行这个系统。他们看起来工作非常努力，常常工作到深夜，周末在加班加点。当遇到生产问题时，团队的每个人都会放下手头的工作，围拢在一起，提供各种建议和意见，以及如何修复问题的见解。你会看到，这个团队中的每个人都有很强的团队凝聚力，而且他们都非常努力。</p>

<p>数字电视供应系统则大相径庭。系统的代码几乎是由一个家伙搞定的，我们暂且称这个家伙为 Dave 吧。我是团队的一名初级维护工程师。一开始，我在理解代码时遇到了很多麻烦。程序中并没有长的过程语句，相反有大量的小体积类和方法，每个方法的代码也只有寥寥数行而已。我有几个同事抱怨 Dave 将事情搞得过于复杂了。不过 Dave 却建议我应该阅读几本关于面向对象编程的图书。他教会我设计模式、SOLID 原则以及单元测试等知识。很快，他所编写的代码开始变得具有现实意义了，随着我不断加深对代码的理解，我也越来越发现它优雅的设计。代码在产品中没有出现问题，只是一直默默地工作。要想对代码做出修改也是轻而易举的事情，因此实现新的特性简直是手到擒来。单元测试意味着进入到生产系统中的 Bug 数量变得微乎其微。</p>

<p>结果就是我们这个团队看起来工作不那么努力。我每天下午 5:30 下班，周末也从来不用工作，我们也不必围聚在一起猜测到底是什么原因导致了生产系统的问题。从外人的角度来看，我们所从事的工作肯定要比模拟电视团队的简单得多。事实上，二者的需求非常相像，只是我们开发出了设计更好的软件、提供了更好的支持基础设施，特别是单元测试。</p>

<p>管理团队宣布要根据绩效给予员工奖励。轮到老板与我谈话时，他说公平的做法是对那些努力工作的员工给予奖励，工作越努力，奖励力度越大，我们的团队看起来对公司并不那么在意，更无法与那放弃晚间与周末时间的英雄相提并论。</p>

<p>这家有线电视公司有一点与众不同，你可以直接比较好的软件设计与差的软件设计之间的差别，可以对团队行为进行比较。大多数组织者都并没有提供这一比较。我们很难说某个员工是不是通宵达旦地工作，甚至周末时间也在工作，频繁充当救火队员的角色，这种做法是不是就是复杂软件系统必须做的呢。除非你可以让几个互相竞争的团队解决同一问题，否则你永远也没法直接比较他们工作上的差别。相反，对于那个坐在角落里，朝九晚五工作的家伙有可能花了很多时间在网上呢？也许他非常善于编写稳定、可靠的代码，也许他的工作要比其他人的简单。对于偶尔过来检查的管理者来说，他们会得第一种人工作很努力，第二种人则不是这样。努力工作就很好，偷懒就很不好，真的是这样么？</p>

<p>我的看法是表面上的努力工作常常是失败的信号。高压、频繁中断的环境常常无法开发出好的软件。长时间的工作也不是一种正确的做法。有时，解决难题最好的方式可能是不再思考这个问题，出去走走，或是睡个好觉，让你的潜意识来解决问题。我最喜欢的一本书是 G. H. Hardy 所编写的 A Mathematician's Apology，他是 20 世纪英国的一位杰出数学家。这本书描绘了他每天的工作：上午工作 4 个小时，下午观看板球比赛。他说对于复杂的脑力工作来说，一天工作 4 个小时以上是完全没有意义且生产力低下的方式。</p>

<p>我想对管理者说的是，以结果为导向，根据员工工作的成果，根据可运行的软件为导向，不要被表面上的努力工作所蒙蔽。另外，最好不要与你的开发者坐在一起，你会得到更好的结果，不受传统、直觉判断所影响的好结果。远程工作的好处是非常多的，你只能以输出来衡量员工的工作情况，而

是看他们是不是每天端坐 8 小时盯着 IDE 在看为标准，或是聚集在一起提出自己的见解为衡量的准。

有读者评论说，文章写的很实在，有时真的很难说服同事设计的简单性与恰当使用 OO 原则所带的好处。我就看到有的人以编写复杂代码并工作到深夜为傲。

还有读者说，我曾经与一个家伙共事过，他说“第一次就将事情做对的困难之处在于没有人认识事情会有多么复杂”。几年过去了，我发现这句话非常正确。我现在都会在项目开始前进行大量的提设计。这常常会让执行变得非常平滑，不过可能会让其他人觉得这件事挺简单的。

手动同步自：<https://ld246.com/forward?goto=http%3A%2F%2Fvanessa.b3log.org%2Fare-your-programmers-working>