

万众创新，移动互联网正在吞噬世界，开发者这个职业炙手可热的程度前所未有的，而且只会越来越热门。许多人意识到这股潮流，加入写程序的行列。不过别看程序工程师坐拥高薪，这可是个强者如云、充满挑战的环境。也因如此，开发者质量的优劣判断总是在网络上引发热烈讨论。曾经在程序员客栈www.proginn.com技术圈上就有这么一道热门的问题「糟糕的软件工程师有什么特征」，亚马逊软件开发工程师Nachiket Naik的回答颇为中肯，获得几千名网友赞同。迈向顶尖开发者的道路上，你该避免成为下列十种人。

1.复制粘贴的人

程序设计问答网站Stack Overflow拥有非常丰硕的资源，很多人写程序碰壁了就会上去找解答，Stack Overflow本身并没有错，它是工程师的得力助手。但是如果只是复制粘贴，改个参数，不去了解前因后果，不去弄懂为何这样的解法到底是不是真的适用于现在面临的问题，那当然很难进步。有不少工程师宁可相信他们在网络论坛看到的说法，而不愿意费心思考眼前的代码或系统。

2.懒得测试的人

「我不干测试这种事，那是测试工程师的责任。」即使在敏捷开发方法如此盛行的时代，这种态度依旧层出不穷。工程师不愿测试的惰性还是很普遍。有可能是他们讨厌设定测试环境，也有可能是缺乏测试的连贯性知识。当然，也或许是，测试工程师在开发者社群中总存在着不能说的污名。

3.不写文件人

有些人觉得程序文件（code documentation）应该如诗一般简洁美丽，他们没能力做到这样，就干脆不做了。可我认为这样的心态是软件开发的头号公敌。杰出的软件，不需要有几百万个酷炫的功能，杰出的软件，应该是要提供几个让人「离不开」不断使用的功能，而且这几个功能背后有几千个人阅读、更新、修正。轻视技术沟通、文件精确度、忽略细节的开发者，肯定是公司获得成功最大的绊脚石。

4.代码写的非常丑的人

我的程序能跑，但??

有些变数被命名为x、flag、str、arr??

Most of what I write is in one giant method.

缺乏连贯的程序惯例或风格

把全局变数喷洒得到处都是

对作者来说，这简直是最恼人的事。虽然某段代码不见得差，甚至有可能是写得最好的部分。只是，如果出现上述情况，就像一条钻石项链被埋葬在泰坦尼克号的残骸中，没人找到它，也没人想清理它、佩戴它、使用它。

5.没有爆发力的人

他写程序、他部署、他继续前进，丝毫没有想要学着解决问题的意愿，只要给这家伙一段程式码，他就会没日没夜奋战，隔天就交出成果，你会得到一个修复好、能执行的软件，

除此之外别无所有。有时候，选择开发者的时候你得有些私心，找个不但会在大限之前完成任务，而且也有旺盛的求知欲的人。

6.一天到晚抱怨人

「这不是我干的」、「这不是我的错」、「这跟我修复的部分无关，一定是有其他人搞砸了」、「这东西真的很烦！（无限循环）」、「我不知道怎么修复这边，找个会的人来啦」??

那个犯错的人可能早就修正向前走了，你还在大肆抱怨什么劲呢？抱怨并不能解决问题。

7.不合群顽固人

「不照我的方法做就拉倒」，是这群人的座右铭。在他们心中，这是一场他的「点子」与你的「点子」之间、他的解决方案与你的解决方案之间的竞争，不为整个项目着想。他们会来来回回仔细你植入的代码，即使他们运作正常、经过测试、看来完美无缺，仍让他们觉得芒刺在背。这类家伙是阻碍生产力的大麻烦，在压力来袭时，他们也会是最先落荒而逃的人，就算经验再怎么丰富、技术再怎么厉害，也别轻易尝试找这些人加入团队。

8.畏首畏尾的人

写Java的A开发者一听到他得写一段Python 就愣住了。B开发者一听到设定档里某个部分必须改正就慌了。C开发者一听到他得在数据库里输入东西就畏缩了。这些人倾向趋吉避凶，不愿离开舒适圈。他们有很奇异的迷信，不想接触系统的某些地方。这个现象尤其容易出现在菜鸟开发者身上，出色的开发者或快或慢，都会渴望跳出舒适圈，探索陌生的事物。

9.迷迷糊糊粗心人

忘掉留存备份、快照存盘、一堆未归档的程序目录??这些都是菜鸟容易出的状况，随着你越来越朝专业者迈进，这些漫不经心的状况都应该避免。

10.小聪明吓唬人

这些人能够耍些小技俩，「骗过」系统使之运作，沾沾自喜。面对复杂的问题，他们仿佛变个魔术就能解决，但就作者的经验，10次有9次都只是表面功夫，实则漏洞百出，而且迟早都会出问题，导致后来还要花更多成本处理。