

今天我们说到的Facebook黑客精神，并非是指Facebook培养了一堆黑客，有很强的黑科技，并在全世界各地研究各种安全漏洞，做出各种黑客应该不应该做的事情。在这里我们讲的黑客精神其实是Facebook对待软件开发的态度和方式。

Facebook的创始人马克·扎克伯格有一句名言：“迅猛而动，突破前行”（Move Fast and Break Things）。这句话也成为了Facebook的座右铭。在Facebook的公司里，到处都贴着这句话。

这句话非常有名，它甚至深深地影响了整个Facebook对待软件开发的態度。那么这句话到底是什么意思，它带给Facebook的到底又是什么结果呢？

“迅猛而动，突破前行”的意思在扎克伯格看来，就是软件开发不要想太多，写出来的东西就发布出去，哪怕写出来的东西不太对，这里那里可能会有问题。如果出大问题的话，赶紧修好就可以了。

扎克伯格认为，让市场上见到东西的速度是很重要的。一个东西如果拖着，等久了可能就晚了，所以，他并不认同传统企业，乃至某些类似谷歌这样的互联网公司对待产品和代码的态度。

他觉得大家每个人就应该像黑客一样，在代码里乱改顺便把新功能实现了。然后可以快速地给用户用，这是一条正确的道路。

这是公司创始人的态度，并且是在公开场合多次强调的态度，所以它在内部的影响力可想而知。

于是Facebook内部有“Hack Everything”的传统，代码很多时候是怎么快怎么来，东西做出去如果有Bug再修，如果把已有的功能搞坏了，修回来就可以了。这基本上在很长一段时间里面是整个Facebook公司的传统了。

这个传统是不是好，从软件开发的角度上来说，一个软件开发的团队，如果一直靠着“Hack Everything”的态度，不好好做架构，不好好测试一下软件，就直接把产品放出去，我个人是很难相信这样的做法是可以持久发展的。

因为软件代码的质量本身也是软件可以持续发展下去的基础。如果不好好维护，慢慢的，加一个新功能就会越来越难了。为了加一个功能付出的代价也越来越高了，有可能这个代价高到无法估计。

当然很长时间内，尽管业界很多人都认为这个想法做法是不对的，在Facebook内部这个做法一直大行其道。而这种做法也反映到了Facebook开源的产品的质量上。基本上，Facebook开源出来的很多产品的代码质量是堪忧的。

比如说著名的Hive，这是Facebook早年投了很大力气开发的SQL on Hadoop的产品，它的代码我看过，“快糙猛”绝非是一个谎言。Hive这个开源项目的代码质量参差不齐，很多地方感觉像是从来没有在工业界正经写过代码的人写的，看起来真的有一种让人不舒服的感觉。

当然除了Hive以外，Facebook的另外一些开源项目的代码质量也同样受到了质疑。这些代码质量差的开源项目，和谷歌极少数虽然开源，但是代码质量犹如艺术一般的项目比起来，我只能说，真的很难想象为什么这是两家齐名的公司，代码质量却大相径庭。

在很长时间内，我们其实并没有注意到Facebook内部是不是真的为这种黑客精神付出了代价。当然我个人是一直坚定地相信这个代价是迟早要付出的，而且等到发觉的时候，代价可能已经很大了，大到需要付出难以承受的成本才能够修复了。

不过，2014年在Facebook的F8开发者大会上，做Keynote的扎克伯格，把自己说过的这句话改了。新的版本变成了：“迅猛而动，稳定架构”（Move Fast With Stable Infra）。简单一点来讲，整个基础架构需要足够稳定的前提下Move Fast。好了，扎克伯格终于改变主意了，再也不说突破前行了。

当整个代码被无休止的“Hack”，天天“突破前行”以后，现在，整个Facebook面对的东西是没有什么不能被突破的。怎么办呢？老老实实回头该补的补，该修的修吧。不补不修，房屋天天漏水，还怎么装点门面迎接客人啊。

这种修补的代价是非常巨大的。我在微软的时候，见过一些开始写得乱七八糟赶时间赶进度赶出来的代码，代价就是这些代码要一个模块一个模块推倒重来，在上面修修补补是没有办法修好的，这需要很多的人力物力。我参与过的一个项目投入了10个人，做了18个月，做完之后总算是看起来能够看了。

我一直有一个困惑，到底是什么让扎克伯格相信他自己的黑客理念呢。只要发布产品足够快就不用付出代价吗？扎克伯格是一个非常聪明的人，而且他周围应该也不缺人告诉他，这个想法的问题。那么在这个背后，扎克伯格想的到底是什么？这个问题我一直没有太多的答案。

一个可能是早年创业的时候，刚开始运转的话，扎克伯格的想的是对的。一个公司如果连生存问题都不能解决好，那么代码质量到底有多重要，也只能是以后的事情了。

虽然说谷歌从一开始连生存问题都没想好怎么解决的时候，就对代码质量要求很高，但是谷歌毕竟是个例。无论如何，有可能是扎克伯格的创业公司的经验和梦想在Facebook变大以后依然在，并未与时俱进。

另外一个可能是扎克伯格从来没有在大公司实践过，所以他固然是一个天才，却不知道在什么样的时候，这种黑客做法会对公司造成伤害。

这种伤害可能还是非常巨大的，所以在2014年，他作为公司领袖，才必须再一次通过他自己的嘴巴告诉大家，他错了。

但是不管怎么样，我们必须看到，榜样的力量是无穷的。因为扎克伯格的观点，很多硅谷更新的创业公司做法都是快糙猛，当年的谷歌那样对待代码的精神，对于现在的公司来说，已经荡然无存了，这在一定程度上造成了开发人员平均水平的严重下降。

扎克伯格在Facebook倡导的黑客精神，如果只是影响了自己公司，那么破坏力还小一点。如果影响了周围后来的很多公司，这种破坏力，我有点不好估量了。所以，虽然Facebook今天已经强调一个稳定的架构是很重要的，但是很多后来的公司其实并没有听进去。

而且，从Facebook内部看，这个黑客精神其实仍然根深蒂固。比如说在Facebook的很多组里进行绩效考核的时候，会先看一个人到底写了多少行代码，行数多的人比写的行数少的人，绩效考核就会更高。这种纯粹以代码量来考核的做法，无疑是当年“迅猛而动”的遗留问题。

代码质量问题是每个大公司有效率分工合作开发的基础。很多时候，代码质量问题，也是从一个创业公司到一个成熟大公司的过渡过程中，一个巨大的挑战。不少公司都会选择严格对待代码质量问题。

但是Facebook却选择了一条截然不同的道路，它们提倡代码的黑客精神。这给Facebook自身的发展带来了许多负面影响，同时，也因为Facebook的影响力，这给整个互联网行业都带来了不好的风气。Facebook的黑客精神所造成的负面影响，不容易消除。