「译」追赶时髦的技术

2017.05.25 Scarletsky technology 热度 212℃

在 Web 开发领域，我们会经常看到有人讨论当前最好的框架或编程语言。Scribd 的联合创始人 Jared Friedman 在 2015 年的时候专门写了一篇文章推荐创业公司使用 Node.js 代替 Rails。

他有以下观点：

> Rails 很慢。
> Hack academy 的毕业生都在用 Rails，让高级工程师贬值，并减少了它们的未来前景。
> 一个创业公司应该使用工程师想用的技术，这样能保证应用程序的质量。
> 在 Scribd，他们这几年已经把技术栈从 Prototype 过渡到 jQuery，再到 CoffeeScript，再到 Angular，再到 React。

Node.js 对于创业公司来说是一个很好的选择，但他提到的两点让我很困惑。
首先，一个创业公司的工程师要明白哪个技术会在几年内流行并且让技术栈不会过时。
然后，优秀的软件工程师会被一个创业公司的技术栈吸引，而不是它们要解决的技术难题。
在过去，我听说过一些不好的传闻：

> 创业公司里的开发者不会接受用 ES5 的 Offer(那时候 CoffeeScript 刚出来)。
> 在 Mongo 发布不久，工程师倾向在生产环境下用 Mongo 代替 Postgres。
> 工程师们都渴望用最新的前端框架来不断重构他们的项目。

我担心有些程序员(和他们的雇主)有这种倾向，也就是把注意力都放在迁移技术栈上。他们选择公司的时候会基于框架，想在工作中用上最先的技术，而不是最适合的工具。他们把时间都花在新的库和框架上，而不是用来提升自己的核心技术能力。我把他们称为「技术栈追逐者」—— 那些把新技术(或者他们最喜欢的技术)都用在创业公司的技术栈上，但对核心输出(用户体验，团队生产力)提升有限的人。

「现代化」的 Web 开发

已经把全站切换到 MEAN 技术栈了，但要试验 Koa 和 Go(在 Express 的核心开发者跑路)。在 2015 年，你们在前端开始用 Gulp/ES2015/React，后端用 Express/Go，用 React Native 来替代原生开发语言，并且用 Docker 慢慢地把系统迁移到微服务架构。不久之后，你将会迁移到 Phoenix，如果 Angular 2 更好的话，你们会迁移过去。或许以后 Go 能用来开发安卓，Swift 可能会适合你们的技术栈(虽然我显然是夸张了，但在 Hacker News 的头条上讨论什么更流行是很正常的表现)。

如果考虑下面提到的这些原因，那么这种行为是可以被理解的，现代的 Web 开发工程师要获得一份更好的工作，他们要更加「时尚」。雇主用框架或者编程语言来筛选雇员，而不是测试他们的思考方式和技能。然而，雇主并没有意识到，优秀的开发者可以在几周，甚至几天就能掌握很多语言或者框架。有时候趋势是无法阻止的：Swift 正在替代 Objective C，世界向着轻量的方向发展，后端需要变得更轻量，前端要更多的响应式。通常，这种变化都会带来巨大的好处：生产力有本质上的提升，用户功能更加容易实现。另外，对于中小型企业来说，这些改变会关系到企业的存亡。

我们可以用在创业公司里的现代的 Web 或移动开发者与我们的计算机科学家作为对比。我有一个朋友在一家顶级的科技公司做计算机神经系统科学家，跟大部分技术人一样，他的世界几个月就会发生变化 —— 得益于计算能力，脑成像和深度学习算法的快速发展。但他的编程工具几乎没发生什么变化。公平来讲，只有 C++ 从 11 迁移到 14 引起了一点担心。其他的变化还有分布式系统，键值存储和其他外部服务，但这些用的都是稳定的 API。他大部分的时间都花在架构和算法上，不会花时间去重写功能相似的代码或学习类库。

选择工具

人们可能会建议创业公司去选择现代化的技术栈，因为这是招聘优秀工程师的一个很关键的工具。根据我自己的观察，优秀的工程师关注的是其他方面，他们最关注的是解决感兴趣的问题，和有趣的同事一起解决问题。吸引优秀的人(工程师或者其他岗位)的附加条件是要引领时代发展和肩负伟大的使命。

了解各种语言间的差异，不要只关注当前流行的那个(同样，不要盲目地选择框架)。

对于创业公司，Paul Graham 在 2013 年的时候被问到关于理想的编程语言：「我们有些创业公司用 PHP 来写他们的产品 —— 那让我有点担忧。但这并不像其他事情那么值得我担忧。」
Github 的技术人员 Sam Lambert 在一次的采访中说第一次面试时，CTO 对他说 Github 大量使用 Rails，C，和 Bash 脚本，他非常惊讶：「随着面试进行，他向我透露那实际上是一群非常务实的黑客在 hack Ruby，hack C，把他们的时间都花在感兴趣的事情上，使用一个稳定的技术栈，好过追逐最新最酷炫的技术。」
Github 的方式在我看来是 Web 和移动开发者的一个合理的平衡：广泛地探索工具，但按照实际需求来选择工具来解决你们要面临的问题。

我担心的是某些开发者，特别是在他们早期的职业生涯当中，有这么一种想法：一个创业公司的工程师并不是专门解决问题的，也不是计算机科学家，只是一个荣誉查找表 —— 他们的任务是每几个月就去研究新的类库/框架/编程语言，来获取有限的提升。这使我们这些早期的工程师贬值，因为我们的工作是创造人们想要的东西，关注感兴趣的技术难题，快速的用代码实现。

因此，务必在你的业余时间中才去实验新技术。只有当切换技术栈有非常巨大的提升的情况下，才在生产环境下切换编程语言/框架，同时要好好想想切换获得了什么优势。警惕那些不考虑新技术对团队的影响，却不断为新技术欢呼的人。你应该把时间花在学习概念，解决技术难题或用户反馈上。如果你有合适的应用程序，而你现在又在选择框架来实现它，那么你的技术选型会有一定的灵活性，但要经过持久的努力才能适应市场。

任何一天打开 Hacker News，你都能看到有帖子诱惑你去用某个框架、语言、类库或者服务来构建应用程序。有些工具拥有改变游戏规则的能力，其他的却只有细微的差异，但要精通它们都是需要时间的。有些工具会大力宣传它们的特点，并嘲笑你所学的技能 —— 但他们却需要你的技能和注意力，这样才能与现在有的技术竞争。你会怎么选择?