

# “为什么有些大公司技术弱爆了？” 我的一些不同看法 - 文章



小编转注：《[为什么有些大公司技术弱爆了？](#)》话题讨论得激烈，前些天小编转的几个热门回复，是教育题主的。今天发一篇支持题主的。

知乎上看到一个热帖，我觉得很有意思，叫做“[为什么有些大公司技术弱爆了？](#)”。我刚看到标题的时候，先入为主和刻板偏见了一下，正如同第一个回答一样，我皱了皱眉头，产生了对题主的鄙视之情；但是很快，读完帖子以后，我却立场明确地站到题主一边了。正如同里面有位回答：

看题目以为是题主傻逼，看了正文发现真的是公司傻逼。

上面这种情况其实发生的概率挺低的，但是我觉得这回是真的发生了。

但是令我感到遗憾的是，各式各样的回答里面，大部分居然都跳出来“教育”题主，表态这个世界就不是完美的，表态要妥协要接受这样的事实，要无奈地咽下这个现实的苦果。这个大面积出现的观点，太不正常了吧？

比如这样的话：

写好代码是程序员的节操。抱歉，节操多少钱一斤，北京三环商品房多少钱一平？

有的问题实在不是能够三言两语回答的，尤其是当面对整个行业的现实的时候，但是，可以很明确的是，当一个项目代码写烂掉的时候，这个项目也活不久了。

好吧，还是就事论事，下面就贴过来，然后逐一分析其中的内容。说说为什么大致上题主没问题，有问题的是这家公司，这个项目组。

=====

今年年初，到一家互联网公司实习，该公司是国内行业龙头。

不过技术和管理方面，却弱爆了。

那里的程序员，每天都在看邮件，查问题工单。

这些问题，多半是他们设计不当，造成的。

>> 这就是所谓的operation的工作，大多情况下很无趣。这通常也意味着系统复杂，负载高，问题不容易轻易定位和解决，往往是历史遗留下来的大型系统。这并不奇怪，就像外人见到风光的AWS一样，之后内部的工程师才知道其中有多少维护性的工作，压力有多巨大。我记得不久前，看过一个principal talk，讲oncall的折磨使他成长，我想说法不假，只是很遗憾我还到不了这个层次。题主能看到问题多

半出自设计不当，不错。而问题居然多半是因为设计不当，这个工程的初始架构人员，以及后来架构看护的骨干工程师，是要挨批评的。对于一个“国内行业龙头”，这样的事情是不该发生的。

代码写的一团糟，全是复制粘贴，连作者都没改，大家普遍不写注释，也不格式化，代码歪歪扭扭。

>> 这又是一件不该发生的事情。对于代码质量的追求各有说法，但是“复制粘贴”、“作者都没改”、“不写注释”、“不格式化”等等这样的字眼，我不相信一般的“龙头”公司能够接受。这些东西就像饭要一口一口吃一样，纵然有再大的野心，这些最最基本的细节，始终是不能忽略的。我觉得公司在招人的时候，既然是双向选择，就可以互亮代码，这样的代码摆出来看到以后，大家就不用浪费时间了。

一个项目里，httpclient竟然出现了四种。

一种是该公司研发部写的，

一种是老版本的开源项目，

一种是新版本的开源项目，

还有一种是开发人员造的轮子。

>> 最理想的情况当然是统一成一种。遇到这样多种实现，并且有自造轮子的情况，很多都是源于“历史原因”。当然，都能看到代码简单地“复制粘贴”了，不看以往代码实现，按照自己的理解来写也就并不奇怪了。当然，我可以接受因为某某特殊原因而导致一个httpclient有多于一种的实现方式（我在这还写过[造轮子的好处](#)），但是居然有四种之多，我觉得凶多吉少了。

打接口请求响应日志，竟然不知道用拦截器。

打错误日志竟然不打上下文信息，每个人一种日志风格，千奇百怪。

许多重要的中间流程，居然不打日志。

>> 拦截器是个好东西，简化代码，避免啰嗦的日志影响业务逻辑的阅读。当然也有不好的地方，比如不直观、不好调试，以及有时候可能发生的性能问题等等。因此这个也不强求，根据项目实际情况而定。日志风格千奇百怪的问题，多是由于缺乏项目内部的管理造成的，各就各业，缺少沟通。开发人员不怎么样不说，这个项目经理更是弱爆了。重要流程不打日志，这一条只能帮助证明这群开发人员的工程意识还欠缺。

idea、eclipse、myeclipse的配置文件竟然全部传到项目里去了。

>> IDE用的不一样没事儿，但是这些IDE的配置文件也传上去了？这样的低级问题都出现……难道代码不用review么？

该公司混了两年的程序员，跟快递公司做查询接口，竟然不知道加密运单号。

>> 这样的信息是否要加密通常取决于调用两边的协议是怎么规定的，但是凡是涉及到隐私等重要信息，都需要加密以减少信息泄露的风险。

所有服务间通讯，都没有设requestId，导致跟踪会话很困难。

>> 如果只是牵涉到服务之间的通讯，而通讯又只是简单的查询的话，没有requestId我觉得是可以接受的。至于会话的跟踪，如果这里指的是整个系统在一个request到达和处理的过程中，能够跟踪全部的或者重要的行为，比如调用了哪些接口，得到了哪些结果，做了哪些操作等等，这个功能确实是很必要的，但是这个跟踪是可以在服务间通讯没有requestId做到的。比如在主系统中使用一个线程变量，在每次打印这些信息的时候把线程变量放置在前面，后续的日志分析工具就可以捕捉到这次会话交互的所有日志。

一个没什么qps的边缘接口，居然做消费者生产者+阻塞队列的异步模式。

显得你技术少是不是。

不知道异步会增加维护成本，提高测试难度吗？

而且，任务队里没有考虑持久化，赶上发布，丢了好多任务。

>> 没什么qps的边缘接口，做成这样的异步模式，看起来是有点杀鸡用牛刀了。但是这个事情要结合背景去分析，比如有可能是为了未来的扩展需要，有的接口可以预见到请求量会大幅增加。当然，结合整个上下文来看，我更倾向于是这个项目组疏于管理，然后来了一个自恃牛逼的“大拿”，整了一套高大上唬住大伙儿；或者是一个很想在项目中尝试新东西的小哥，就拿这东西练手了。至于任务队里没有持久化这个一条，依然要看具体的问题，不过通常情况下，如果要设计一个通用的任务队列，持久化是一个必选项。当然话也不能说死，你是要搞完全不在乎任务丢失的，或者任务调度者可以不断地重试那些挂掉的任务，于是你不在乎他们丢失的问题——不过想想好像这样的case挺少的。

读取一个小小的xml和exc配置文件，居然用流式解析，没见过这么二逼的，真是醉了。

>> 这和上面那个杀鸡用牛刀是同一个问题，已经阐述过了。

做优化全靠拍脑门拍大腿，难道不会用excel分析日志，用jprofile扫项目？

一个100以内的常数集合遍历，他也要写个优化算法进去，算法跟业务还搅在一起，一团乱麻。

每个人都在嚷嚷性能、算法、分布式计算……

>> 看起来这里指的是性能方面的优化，那么做优化至少包括两部分，一部分是在设计阶段就要分析需求层面的数据推出要“优化”到什么程度，另一部分才是题主说的根据日志和已有项目运行的数据“反推”（几年前写过一点[这方面的东西](#)）。当然，无论哪个，都比拍脑门和拍大腿靠谱得多。至于100以内常数集合遍历，也要写优化算法，这有时未必是件坏事，比如大家都在遵循最佳实践，不过结合上下文看（包括“算法和业务搅在一起”），我更倾向于是属于前面已经阐述过的问题。就这种状况下，每个人都还嚷嚷“性能、算法、分布式计算”就显得有点没抓到主要矛盾了，主要矛盾应该是把这些代码最基本的问题给解决了。我记得小时候练习书法的时候，老师批评过我：先不要尝试那些技巧，先把最基本的横平竖直给练好了。

几乎没有文档，全靠从代码反推逻辑。

>> 代码和文档经常是对立面，这样的状况并不稀奇。我觉得比较可行的做法是，有概要的文档，但是详细文档往往不现实，即使写了也难免过时。公司内部文档太多太多是一坨浆糊。

有枚举他不用，非要在每个页面上，把枚举值挨个儿写死，知道后面改代码多么费劲吗？

>> 欠缺基本的程序员素质。

欺骗性的变量名，里面存储的是AES加密的，变量名后缀却写成了DES；里面存的是小写字母，却写成upperStr。

一个方法十几个参数，有三分之一是极其简略的缩写，注释肯定也没有的。

一个类写到三四千行是常事。

>> 看到这里我已经产生无力吐槽的感觉了。

开发自测，居然要把代码全丢到公共机器上，而且都是走svn，他们把svn当ftp用。

svn里面大量的无意义提交，一多半的提交连都编译不过去。

我看到有个应届生，改了两句话，马上提交，说是怕代码丢失。

>> 自测走svn其实不稀奇。就像自己开发一个新功能在git下可以cut一个新的branch，然后开发了，提交了，部署到各种机器上去。不过当ftp用显然是不对的。“一半多编译不过去的提交”，这个项目没有项目管理吗？开发机上不单元测试吗？至于“改了两句话，马上提交”，没看出有什么不妥，只要提交前的测试、review等等通过。

一个运行了两年的项目，spring的包扫描明显配错了，有些bean根本扫不进来，居然没有人发现。一半的bean在spring管理下，另一半的bean他们自己写单例模式来实例化。

>> 第一条依然是项目组疏于管理的佐证。即便是主力程序员，也缺少对项目整体的责任感，或者是代码烂得让人难以提起兴致。第二条则是一个典型的不好的实践。有时候可能会需要这样的妥协，但是居然有一半的bean脱离Spring的管理，那最初引入Spring干嘛？

他们用mysql来做审计系统，出报表，有个报表要跑8分钟。

原来是有人用字符串来存多值（逗号分隔），sql里写了like，导致没有利用到索引。

为什么不用pg，pg在sql编程方面，功能更丰富，更适合做统计，它本身就支持数组。

>> 报表跑8分钟很正常。Sql用字符串存多值这个，没有利用索引，还是要分析具体问题，原则上我不觉得有什么问题。要想完美解决这个问题，还是在mysql里面，就得把多值拆解成多行，放到一张新表里面去。另外，也有一些NoSQL系统天然支持value多值，比如DynamoDB，不过这是题外话。至于为什么不用pg，这涉及到最初的技术选型，后人看的时候只是说说“如果用xxx就yyy了”当然容易，但是不清楚最初是否有技术层面的考量。当然，这个项目那么烂，也许是一开始图方便搞了mysql的prototype就上了。无论如何，有质疑的想法总是值得鼓励的。

程序员们都是得过且过的态度，怎么把代码灌进去，跑的通测试，就算交差了。

为什么大型互联网公司，技术和管理这么差劲，是怎么形成的？

加入伯乐在线专栏作者。扩大知名度，还能得赞赏！详见《[招募专栏作者](#)》

1 赞 1 收藏 [3 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