<https://blog.csdn.net/youanyyou/article/details/105926213?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.nonecase>

**在滴滴和头条干了 2 年后端开发，太真实…**



[Java技术栈](https://me.csdn.net/youanyyou) 2020-05-04 13:35:00  616  收藏

版权



先简单交代一下背景吧，某不知名985的本硕，17年毕业加入滴滴，今年下半年跳槽到了头条，一直从事后端研发相关的工作。

之前没有实习经历，算是两年半的工作经验吧。这两年半之间完成了一次晋升，换了一家公司，有过开心满足的时光，也有过迷茫挣扎的日子，不过还算顺利地从一只职场小菜鸟转变为了一名资深划水员。

在这个过程中，总结出了一些还算实用的划水经验，有些是自己领悟到的，有些是跟别人交流学到的，在这里跟大家分享一下。

**学会深入思考，总结沉淀**

我想说的第一条就是要学会深入思考，总结沉淀，这是我觉得最重要也是最有意义的一件事。

‌先来说深入思考。在程序员这个圈子里，常能听到一些言论：我这个工作一点技术含量都没有，每天就CRUD，再写写if-else，这TM能让我学到什么东西？

抛开一部分调侃和戏谑的论调不谈，这可能确实是一部分同学的真实想法，至少曾经的我，就这么认为过。

后来随着工作经验的积累，加上和一些高level的同学交流探讨之后，我发现这个想法其实是非常错误的。之所以出现没什么可学的这样的看法，基本上是思维懒惰的结果。

任何一件看起来很不起眼的小事，只要进行深入思考，稍微纵向挖深或者横向拓宽一下，都是足以让人沉溺的知识海洋。

‌举一个例子。某次有个同学跟我说，这周有个服务[OOM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw%3D%3D&chksm=eb539286dc241b90529e3aa5efee03dc2cf41c9884c2e015ea0123ead6c02dda605e607d4c15&idx=1&mid=2247489456&scene=21&sn=f24823b901bf4a697b7593737546e548#wechat_redirect)了，查了一周发现有个地方defer写的有问题，改了几行代码上线修复了，周报都没法写。

可能大家也遇到过这样的场景，还算是有一定的代表性。其实就查bug这件事来说，是一个发现问题，排查问题，**解决问题的过程，包含了触发、定位、复现、根因、修复、复盘等诸多步骤。**

花了一周来做这件事，一定有不断尝试与纠错的过程，这里面其实就有很多思考的空间。**比如说定位，如何缩小范围的？走了哪些弯路？用了哪些分析工具？**

**比如说根因，**可以研究的点起码有[linux](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw%3D%3D&chksm=eb539b4edc2412587424703bdf6d717a4143c62d2186ecda3d4379656a2a797c6508c0f397fd&idx=3&mid=2247491192&scene=21&sn=2b9e0173b07ce4e5cdd50813b01bfa8a#wechat_redirect)的OOM，k8s的OOM，go的内存管理，defer机制，函数闭包的原理等等。如果这些真的都不涉及，仍然花了一周时间做这件事，那复盘应该会有很多思考，提出来几十个WHY没问题吧…

**‌再来说下总结沉淀。**这个我觉得也是大多数程序员比较欠缺的地方，只顾埋头干活，可以把一件事做的很好。但是几乎从来不做抽象总结，以至于工作好几年了，所掌握的知识还是零星的几点，不成体系，不仅容易遗忘，而且造成自己视野比较窄，看问题比较局限。

**适时地做一些总结沉淀是很重要的，这是一个从术到道的过程，会让自己看问题的角度更广，层次更高。**遇到同类型的问题，可以按照总结好的方法论，系统化、层次化地推进和解决。

‌还是举一个例子。**做后台服务，今天优化了1G内存，明天优化了50%的读写耗时，是不是可以做一下性能优化的总结？**

比如说在应用层，可以管理服务对接的应用方，梳理他们访问的合理性；在架构层，可以做缓存、预处理、读写分离、异步、并行等等；在代码层，可以做的事情更多了，资源池化、对象复用、无锁化设计、大key拆分、延迟处理、编码压缩、gc调优还有各种语言相关的高性能实践…

等下次再遇到需要性能优化的场景，一整套思路立马就能套用过来了，剩下的就是工具和实操的事儿了。大家也可以关注微信公众号：Java技术栈，在后台回复：架构，可以获取我整理的 N 篇 Java 架构教程，都是干货。

‌还有的同学说了，我就每天跟PM撕撕逼，做做需求，也不做性能优化啊。先不讨论是否可以搞性能优化，单就做业务需求来讲，也有可以总结的地方。**比如说，如何做系统建设？系统核心能力，系统边界，系统瓶颈，服务分层拆分，服务治理这些问题有思考过吗？**

每天跟PM讨论需求，那作为技术同学该如何培养产品思维，引导产品走向，如何做到架构先行于业务，这些问题也是可以思考和总结的吧。就想一下，连接手维护别人烂代码这种蛋疼的事情，都能让Martin Fowler整出来一套重构理论，还显得那么高大上，我们确实也没啥必要对自己的工作妄自菲薄…

‌所以说学习和成长是一个自驱的过程，如果觉得没什么可学的，大概率并不是真的没什么可学的，而是因为自己太懒了，不仅是行动上太懒了，思维上也太懒了。

‌可以多写技术文章，多分享，强迫自己去思考和总结，毕竟如果文章深度不够，大家也不好意思公开分享。说到分享，大家可以关注微信公众号：Java技术栈，在后台回复：Java，可以获取我写的过的 N 篇 Java 技术教程，都是干货。



**积极学习，保持技术热情**

‌最近两年在互联网圈里广泛传播的一种焦虑论叫做35岁程序员现象，大意是说程序员这个行业干到35岁就基本等着被裁员了。[程序员究竟能干多少年？](https://blog.csdn.net/youanyyou/article/details/105764401)这篇推荐大家看下。

不可否认，互联网行业在这一点上确实不如公务员等体制内职业。但是这个问题里35岁程序员并不是绝对生理意义上的35岁，应该是指那些工作十几年和工作两三年没什么太大区别的程序员。

后面的工作基本是在吃老本，没有主动学习与充电，35岁和25岁差不多，而且没有了25岁时对学习成长的渴望，反而添了家庭生活的诸多琐事，薪资要求往往也较高，在企业看来这确实是没什么竞争力。

**‌而如果我们积极学习，保持技术能力、知识储备与工作年限成正比，这到了35岁哪还有什么焦虑呢，这样的大牛我觉得应该也是各大公司抢着要吧？**但是学习这件事，其实是一个反人类的过程，这就需要我们强迫自己跳出自己的安逸区，主动学习，保持技术热情。

在滴滴时有一句话大概是，主动跳出自己的舒适区，感到挣扎与压力的时候，往往是黎明前的黑暗，那才是成长最快的时候。相反如果感觉自己每天都过得很安逸，工作只是在混时长，那可能真的是温水煮青蛙了。

‌刚毕业的这段时间，往往空闲时间还比较多，正是努力学习技术的好时候。借助这段时间夯实基础，培养出良好的学习习惯，保持积极的学习态度，应该是受益终身的。至于如何高效率学习，网上有很多大牛写这样的帖子，到了公司后内网也能找到很多这样的分享，我就不多谈了。

‌可以加入学习小组和技术社区，公司内和公司外的都可以，关注前沿技术。

**‌主动承担，及时交流反馈**

‌前两条还是从个人的角度出发来说的，希望大家可以提升个人能力，保持核心竞争力，但从公司角度来讲，公司招聘员工入职，最重要的是让员工创造出业务价值，为公司服务。

虽然对于校招生一般都会有一定的培养体系，但实际上公司确实没有帮助我们成长的义务。而在能为公司办成事，创造价值这一点上，我觉得最重要的两个字就是主动，主动承担任务，主动沟通交流，主动推动项目进展，主动协调资源，主动向上反馈，主动创造影响力等等。

我当初刚入职的时候，基本就是leader给分配什么任务就把本职工作做好，然后就干自己的事了，几乎从来不主动去跟别人交流或者主动去思考些能帮助项目发展的点子。自以为把本职工作保质保量完成就行了，后来发现这么做其实是非常不够的，这只是最基本的要求。

而有些同学的做法则是leader只需要同步一下最近要做什么方向，下面的一系列事情基本不需要leader操心了 ，这样的同学我是leader我也喜欢啊。入职后经常会听到的一个词叫owner意识，大概就是这个意思吧。

‌在这个过程中，另外很重要的一点就是及时向上沟通反馈。项目进展不顺利，遇到什么问题，及时跟leader同步，技术方案拿捏不准可以跟leader探讨，一些资源协调不了可以找leader帮忙，不要有太多顾忌，认为这些会太麻烦，leader其实就是干这个事的。。

如果项目进展比较顺利，确实也不需要leader介入，那也需要及时把项目的进度，取得的收益及时反馈，自己有什么想法也提出来探讨，问问leader对当前进展的建议，还有哪些地方需要改进，消除信息误差。

做这些事一方面是合理利用leader的各种资源，另一方面也可以让leader了解到自己的工作量，对项目整体有所把控，毕竟leader也有leader，也是要汇报的。可能算是大家比较反感的向上管理吧，有内味了，这个其实我也做得不好。但是最基本的一点，不要接了一个任务闷着头干活甚至与世隔绝了，一个月了也没跟leader同步过，想着憋个大招之类的，那基本凉凉。

‌**一定要主动，可以先从强迫自己在各种公开场合发言开始，有问题或想法及时one-one**



‌除了以上几点，还有一些小点我觉得也是比较重要的，列在下面：‌

**第一件事建立信任**

‌无论是校招还是社招，刚入职的第一件事是非常重要的，直接决定了leader和同事对自己的第一印象。入职后要做的第一件事一定要做好，最起码的要顺利完成而且不能出线上事故。这件事的目的就是为了建立信任，让团队觉得自己起码是靠谱的。

如果这件事做得比较好，后面一路都会比较顺利。如果这件事就搞杂了，可能有的leader还会给第二次机会，再搞不好，后面就很难了，这一条对于社招来说更为重要。

‌而刚入职，公司技术栈不熟练，业务繁杂很难理清什么头绪，压力确实比较大。这时候一方面需要自己投入更多的精力，另一方面要多跟组内的同学交流，不懂就问。

最有效率的学习方式，我觉得不是什么看书啊学习视频啊，而是直接去找对应的人聊，让别人讲一遍自己基本就全懂了，这效率比看文档看代码快多了，不仅省去了过滤无用信息的过程，还了解到了业务的演变历史。当然，这需要一定的沟通技巧，毕竟同事们也都很忙。

‌**脸皮要厚一点，多找人聊，快速融入，最忌讳有问题也不说，自己把自己孤立起来。**

**超出预期**

‌超出预期这个词的外延范围很广，比如leader让去做个值周，解答用户群里大家的问题，结果不仅解答了大家的问题，还收集了这些问题进行分类，进而做了一个智能问答机器人解放了值周的人力，这可以算超出预期。比如leader让给运营做一个小工具，结果建设了一系列的工具甚至发展成了一个平台，成为了一个完整的项目，这也算超出预期。

超出预期要求我们有把事情做大的能力，也就是想到了leader没想到的地方，并且创造了实际价值，拿到了业务收益。这个能力其实也比较重要，在工作中发现，有的人能把一个小盘子越做越大，而有的人恰好反之，那么那些有创新能力，经常超出预期的同学发展空间显然就更大一点。

‌**这块其实比较看个人能力，暂时没想到什么太好的捷径，多想一步吧。**



**体系化思考，系统化建设**

‌这句话是晋升时候总结出来的，大意就是做系统建设要有全局视野，不要局限于某一个小点，应该有良好的规划能力和清晰的演进蓝图。比如，今天加了一个监控，明天加一个报警，这些事不应该成为一个个孤岛，而是属于稳定性建设一期其中的一小步。

这一期稳定性建设要做的工作是报警配置和监控梳理，包括机器监控、系统监控、业务监控、数据监控等，预期能拿到XXX的收益。

这个工作还有后续的roadmap，稳定性建设二期要做容量规划，接入压测，三期要做降级演练，多活容灾，四期要做…给人的感觉就是这个人思考非常全面，办事有体系有规划。

‌**平时积极总结沉淀，多跟别人交流，形成方法论。**‌

**提升自己的软素质能力**

‌这里的软素质能力其实想说的就是PPT、沟通、表达、时间管理、设计、文档等方面的能力。说实话，我觉得我当时能晋升就是因为PPT做的好了一点…可能大家平时对这些能力都不怎么关注，以前我也不重视，觉得比较简单，用时候直接上就行了，但事实可能并不像想象得那样简单。

比如晋升时候PPT+演讲+答辩这个工作，其实有很多细节的思考在里面，内容如何选取，排版怎么设计，怎样引导听众的情绪，如何回答评委的问题等等。

晋升时候我见过很多同学PPT内容编排杂乱无章，演讲过程也不流畅自然，虽然确实做了很多实际工作，但在表达上欠缺了很多，属于会做不会说，如果再遇到不了解实际情况的外部门评委，吃亏是可以预见的。

‌**公司内网一般都会有一些软素质培训课程，可以找一些场合刻意训练。**

‌以上都是这些分享还都算比较伟光正，但是社会吧也不全是那么美好的。。下面这些内容有负能量倾向，三观特别正的同学以及观感不适者建议跳过。

**拍马屁是真的香**

‌拍马屁这东西入职前我是很反感的，我最初想加入互联网公司的原因就是觉得互联网公司的人情世故没那么多，事实证明，我错了…

入职前几天，部门群里大leader发了一条消息，后面几十条带着大拇指的消息立马跟上，学习了，点赞，真不错，优秀，那场面，说是红旗招展锣鼓喧天鞭炮齐鸣一点也不过分。

除了惊叹大家超强的信息接收能力和处理速度外，更进一步我还发现，连拍马屁都是有队形的，一级部门leader发消息，几个二级部门leader跟上，后面各组长跟上，最后是大家的狂欢，让我一度怀疑拍马屁的速度就决定了职业生涯的发展前景（没错，现在我已经不怀疑了）。

‌坦诚地说，我到现在也没习惯在群里拍马屁，但也不反感了，可以说把这个事当成一乐了。倒不是说我没有那个口才和能力（事实上也不需要什么口才，大家都简单直接），在某些场合，为活跃气氛的需要，我也能小嘴儿抹了蜜，甚至能把古诗文彩虹屁给leader安排上。而是我发现我的直属leader也不怎么在群里拍马屁，所以我表面上不公开拍马屁其实属于暗地里事实上迎合了leader的喜好…

‌但是拍马屁这个事只要掌握好度，整体来说还是香的，最多是没用，至少不会有什么坏处嘛。大家能力都差不多，每一次在群里拍马屁的机会就是一次露脸的机会，按某个同事的说法，这就叫打造个人技术影响力…

**‌想舔就舔，不想舔也没必要酸别人，Respect Greatness。**

‌永不缺席的撕逼甩锅实战 ‌ 有人的地方，就有江湖。虽然搞技术的大多城府也不深，但撕逼甩锅邀功抢活这些闹心的事儿基本也不会缺席，甚至我还见到过公开群发邮件撕逼的…

这部分话题涉及到一些敏感信息就不多说了，而且我们低职级的遇到这些事儿的机会也不会太多。只是给大家提个醒，在工作的时候迟早都会吃到这方面的瓜，到时候留个心眼。

‌稍微注意一下，咱不会去欺负别人，但也不能轻易让别人给欺负了。



**不要被画饼蒙蔽了双眼**

‌说实话，我个人是比较反感灌鸡汤、打鸡血、谈梦想、讲奋斗这一类行为的，这一套\*\*\*治还在大行其道，真不知道是该可笑还是可悲。当然，这些词本身并没有什么问题，但是这些东西应该是自驱的，而不应该成为外界的一种强push。

『**我必须努力奋斗**』这个句式我觉得是正常的，但是『你必须努力奋斗』这种话多少感觉有点诡异，努力奋斗所以让公司的股东们发家致富？尤其在钱没给够的情况下，这些行为无异于耍流氓。

我们需要对leader的这些画饼操作保持清醒的认知，理性分析，作出决策。比如感觉钱没给够（或者职级太低，同理）的时候，可能有以下几种情况：

•leader并没有注意到你薪资较低这一事实  
•leader知道这个事实，但是不知道你有多强烈的涨薪需求  
•leader知道你有涨薪的需求，但他觉得你能力还不够  
•eader知道你有涨薪的需求，能力也够，但是他不想给你涨  
•leader想给你涨，也向上反馈和争取了，但是没有资源

‌这时候我们需要做的是向上反馈，跟leader沟通确认。如果是1和2，那么通过沟通可以消除信息误差。如果是3，需要分情况讨论。如果是4和5，已经可以考虑撤退了。

对于这些事儿，也没必要抱怨，抱怨解决不了任何问题。我们要做的就是努力提升好个人能力，保持个人竞争力，等一个合适的时机，跳槽就完事了。

‌时刻准备着，技术在手就没什么可怕的，哪天干得不爽了直接跳槽。‌

**学会包装**

‌这一条说白了就是，要会吹。忘了从哪儿看到的了，能说、会写、善做是对职场人的三大要求。能说是很重要的，能说才能要来项目，拉来资源，招来人。

同样一件事，不同的人能说出来完全不一样的效果。比如我做了个小工具上线了，我就只能说出来基本事实，而让leader描述一下，这就成了，打造了XXX的工具抓手，改进了XXX的完整生态，形成了XXX的业务闭环。老哥，我服了，硬币全给你还不行嘛。

据我的观察，每个互联网公司都有这么几个词，抓手、生态、闭环、拉齐、梳理、迭代、owner意识等等等等，我们需要做的就是熟读并背诵全文，啊不，是牢记并熟练使用。

**‌这是对事情的包装，对人的包装也是一样的，尤其是在晋升和面试这样的应试型场合，特点是流程短一锤子买卖，包装显得尤为重要。**

晋升和面试这里就不展开说了，这里面的道和术太多了。。关注微信公众号：Java技术栈，在后台回复：面试，可以获取我整理的 N 篇 Java 面试题干货。

**下面的场景提炼自面试过程中和某公司面试官的谈话，大家可以感受一下：**

我们背后是一个四五百亿美金的市场…

我负责过每天千亿级别访问量的系统…

工作两年能达到这个程度挺不错的…

贵司技术氛围挺好的，业务发展前景也很广阔…

啊，彼此彼此…

嗯，久仰久仰…

**人生如戏，全靠演技。**

‌可以多看leader的PPT，多听老板的向上汇报和宣讲会。

**选择和努力哪个更重要？**

‌这还用问么，当然是选择。在完美的选择面前，努力显得一文不值，我有个多年没联系的高中同学今年已经在时代广场敲钟了…

但是这样的案例太少了，做出完美选择的随机成本太高，不确定性太大。对于大多数刚毕业的同学，对行业的判断力还不够成熟，对自身能力和创业难度把握得也不够精准，此时拉几个人去创业，显得风险太高。

**我觉得更为稳妥的一条路是，先加入规模稍大一点的公司，找一个好leader，抱好大腿，提升自己的个人能力。好平台加上大腿，再加上个人努力，这个起飞速度已经可以了。**等后面积累了一定人脉和资金，深刻理解了市场和需求，对自己有信心了，可以再去考虑创业的事。

本来还想分享一些生活方面的故事，发现已经这么长了，那就先这样叭。上面写的一些总结和建议我自己做的也不是很好，还需要继续加油，和大家共勉。

另外，其中某些观点，由于个人视角的局限性也不保证是普适和正确的，可能再工作几年这些观点也会发生改变，欢迎大家跟我交流~（甩锅成功）

最后祝大家都能找到心仪的工作，快乐工作，幸福生活，广阔天地，大有作为。

作者：syntax\_error  
https://www.nowcoder.com/discuss/351805

END