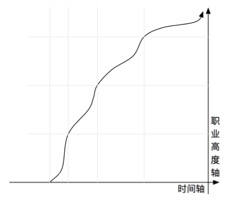
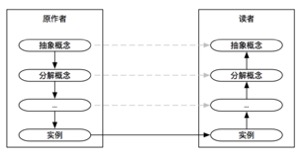
**序**  
  
2013年03月02日, 我写了一篇博客<从1.5K到18K, 一个程序员的5年成长之路>(http://blog.csdn.net/lgg201/article/details/8637763), 被CSDN全站推荐. 后来CSDN社区之星栏目对我做了专访(http://www.csdn.net/article/2013-05-13/2815252).  
  
因为当年CSDN仍然具有较大的影响力, 所以, 这篇文章及专访, 当时比较热门. 有非常多的评论, 截止目前(2016.09.28), 两个链接下, 总共有1262条评论.  
  
有很多支持的朋友, 有很多反对的朋友, 也有很多只看到"薪资"的朋友, 我并没有做过正面的回应, 因为, 我相信, 那篇文章, 一定会帮助到一些人, 使他们找到正确的方向.  
  
现在, 我工作满7年. 原本的计划, 是在工作满10年的时候, 有更多积累之后, 再来写一些能够帮助刚入行的年轻工程师的东西.  
  
不过, 在今年04月29日和09月08日, 我在CSDN私信中收到了两条感谢的私信.  
三年前的这篇文章, 帮助两位年轻的工程师坚定了在这一行继续努力的信念, 三年后的他们有所成, 建立起了属于自己的自信, 给我发私信表示感谢.  
  
并且, 我从其中一位的博客中, 看到了非常正面的价值观.(http://blog.csdn.net/SOLDIER\_BOSSJIE/article/details/52467877)  
  
我深受感触.  
  
正好, 最近一年多时间中, 我主要的工作精力放在团队构建上, 更多的我是从工程师成长角度思考这个问题.  
在这方面, 有一些收获. 所以, 就想把这方面的感悟, 分享出来, 希望可以帮助到更多的人.  
  
Ps. 我觉得, 这篇文章也应该适合其他职业的职业发展参考, 但因为不了解其他职业, 所以, 仅以工程师职业发展命题.  
  
**工程师职业发展的四个阶段**  
  
在我看来, 一个工程师的养成, 可以分为四个阶段:  
新人期  
    硬技能: 理论转化实践  
    软素质: 养成基本的职业观  
成熟期  
    硬技能: 积累业务知识, 积累技术知识  
    软素质: 固化职业观  
发展期  
    硬技能: 明确细化的职业发展方向, 深入研究[可能技术, 可能管理, 可能两者兼具]  
    软素质: 逐步总结并形成自己的方法论  
事业期  
    硬技能: 释放自己, 创造价值  
    软素质: 输出方法论, 影响其他人  
  
我自己目前徘徊在发展期, 能看到的问题有限. 另一方面, 俗话说"三岁看大, 七岁看老", 很大程度上, 职业发展的前面两个阶段, 已经决定了未来的职业发展路径.  
  
所以, 这篇文章, 我主要针对新人期/成熟期的一些问题, 进行阐述分析.  
  
**现象: 成长越来越慢**  
  
在我接触过的工程师中, 这是一个非常普遍的现象. 并且, 这个现象多发于工作3年以上的同学身上.  
  
  
这张图, 是我理解的职业发展模型.  
  
首先, 职业发展过程, 是阶梯式的, 一个台阶一个台阶往上走, 而不是线性提升.  
其次, 职业发展前期, 上台阶所需时间更短, 即: 成长速度更快.  
  
是什么造就了这种现象?  
   
我认为, 造成这种现象, 主要有两方面的原因:  
随着职业等级的提升, 所需的基本素质越来越抽象  
没有及时从"积累具体知识"调整到"提升抽象思维"  
随着职业等级的提升, 所需的驱动力, 越来越转向内在  
没有及时从"被动成长"调整到"主动成长"  
   
首先, 没有及时从"积累具体指示"调整到"提升抽象思维"  
  
回到职业发展的四个阶段, 看看各阶段职级都在做什么样的事情:  
  
新人期: 完成一项具体的任务.  
成熟期: 完成一个项目的整体把控.  
发展期: 引领一个专业方向的发展.  
事业期: 引领一个或多个专业方向的团队, 将技能转化为实际生产力.  
  
很容易看出, 随着职业的发展, 我们需要具备的基本素质越来越抽象.  
  
结合上面的职业发展模型图, 一个人的成长, 分为两个方向:  
一方面是横向的同级别的知识/经验积累  
另一方面, 是更高层级的抽象思考  
  
我认为, 这就是造成这个现象的第一个关键点.  
很多人都只关注了同级别的知识积累, 而一个人能Hold的同级别知识总量是有限的.  
  
所以, 工作两三年后, 相关的东西就积累的差不多了, 你很难找到一些自己"不会"的东西.  
   
那么, 问题就来了:  
感觉成长越来越慢了  
不知道该学什么了  
是不是该考虑做做管理了  
......  
  
建议: 提前用下一个职业等级的思路(或者说超越当下的视角), 去思考问题.  
  
其次, 没有及时从"被动成长"调整到"主动成长"  
  
还是从职业发展的四个阶段来看, 每个阶段是什么在驱动你做事情?  
  
新人期: 按照领导的安排, 1, 2, 3个步骤, 完成一件事情.  
成熟期: 按照领导的安排, 把一件大的事情, 拆解到具体的事情, 独立完成, 或者和其他人合作完成.  
发展期: 在完成领导安排的既定事务的同时, 还要从专业角度主动的发现并解决问题.  
事业期: 需要自主的从公司或产品战略出发, 全方位找到要解决的问题.  
  
可以看出, 随着职业发展, 驱动力越来越收敛到自身.  
  
前两个阶段, 更多的是公司的事情驱动着你的前进, 通过公司的驱动力, 被动的获取成长.  
后两个阶段, 更多则需要自己主动的驱动事情往前走, 通过这种方式, 主动的获取成长.  
  
所以, 从这个角度讲, 一个人要想获得更好的职业发展, 一定要很好的去培养自己的主动性, 一方面主动获取成长, 另一方面, 还要在公司中, 主动的承担更多的问题, 这样才能够获取到更多思考的机会.  
  
**如何避免走向平庸?**  
  
上面所提的现象, 如果不注意去克服, 就很容易走向平庸.  
在上面的分析中, 有一些建议, 这里, 要分享的是3点具体的总结.  
  
**学习方法**  
  
我对学习方法有一个抽象的总结, 是在阅读了<如何阅读一本书>这本书之后, 在一次和团队成员的OneOnOne中总结出来的.  
  
  
  
我认为, 知识的传递, 和网络通信模型有异曲同工之妙.  
  
从知识的分享者角度来看, 比如, 一本结构良好的书, 应该有几个明确的, 抽象的基本观点. 其他所有的内容, 都是围绕着这些基本观点, 一层一层的将问题具象化, 帮助读者逐层的理解最终那几个抽象的观点.  
(Ps. 并不适用所有书籍, 比如小说类书籍, 可能抽象程度更高, 更加隐晦)  
  
从读者角度来看, 一定是先理解了那些具体的实例, 然后才能逐步的Get到原作者想要表达的抽象观点. 通过这个过程, 吸取原作者分享的知识.  
  
所以, 在技术领域的学习方面, 我觉得有两点特别关键:  
从实践出发, 因为有效的信息传递大部分都是通过这层完成. 所以, 大量的实践才能让你Get到原作者直接传递的更多信息.  
以抽象的思想为目标, 所有实践的目的, 都是为了尽可能多的Get到原作者表达的抽象思想, 只有理解了抽象的思想, 才能用它指导自己解决大量同类问题, 甚至对原作者提出的概念进行延伸, 形成自己的方法论.  
  
**工作方法**  
  
从工作的角度去看一个问题, 我认为分3个阶段:  
接收需求  
执行需求  
完成需求  
  
我见过相当一部分工程师, 其实只有"执行"这一个阶段.  
  
拿到一个需求, 草草看几眼MRD, 然后就开始编码, 这是对自己不负责, 对公司不负责, 对其他同事不负责.  
  
对一个系统的设计和实现之间的权衡, 我有这样一个观点: 思考1-3年的变数, 设计6-12月的架构, 只实现当下需要的.  
  
所以, 我认为, 作为一名优秀的工程师, 接到一份需求的时候, 首先应该做的就是认真的阅读需求, 从产品角度思考这个需求背后的逻辑. 只有真正理解一个需求之后, 才能够考虑到未来1-3年可能存在的变数, 才可能设计出符合6-12月业务发展所需的架构.  
  
再来看经常被忽视的"完成需求"阶段. 通常, 工程师在整个项目的研发过程中, 只是其中一环, 要使得整个流程能够顺畅的运转起来, 每一环就必须都及时的通知到下一个环节.  
  
上面这是站在公司角度考虑的.  
  
另一方面, 站在个人角度而言. 对自己做过的事情, 复盘总结, 往往是最佳的成长机会. 因为, 当你完成一件事情之后, 你就对事情的全盘有了了解. 这个时候, 回过头去看这件事情, 就是现成的"站在更高视角看问题"的机会.  
  
**技能是什么?**  
  
技能 = 技术 + 能力.  
  
技术, 是由你的知识体系, 外加你的经验构成的. 它能够通过量的积累直接获得提升. 不过, 它只能用来解决已知的问题.  
  
能力, 是由你的抽象思维能力, 你已有的方法论构成的. 它不能直接通过量的积累获得. 一方面, 它会由天生的智商情商等决定一部分; 另一方面, 要在量化积累的基础上, 经过深度的思考, 找寻问题的本质, 引发质变才能获得. 能力可以被用来解决未知问题.  
  
技术和能力是相辅相成的. 分享这一点, 是因为在我看来, 如果能够认清两者的区别, 对两方面的应用和提升就会更加得心应手.  
  
**新人期常见问题分析**  
  
下面列举三个在我平时和团队成员OneOnOne过程中, 经常碰到的软素质方面的问题, 以及我认为比较好的解决方案, 供参考.  
  
**如何处理并行任务?**  
  
有没有碰到过你的Leader在你上一件事情没有完成的时候, 给你就分配了下一件事情的情况? 同时, 又来了几个来自其他同事的需求.  
  
在新人期的工程师, 主动性方面通常并不会差, 他们希望快点把所有问题解决, 但又无法同时都解决掉.  
  
所以, "好烦啊", 焦虑感就产生了.  
  
一般而言, 很多新人都会使用下面两种方式之一进行处理:  
FIFO: 谁先找我, 就先把谁的问题处理完, 然后再去看后面的事情.  
LIFO: 谁现在来找我, 我就处理谁的问题.  
  
哪种方式好呢? 都不好!  
  
首选方案, 是建议大家查阅时间管理相关的资料, 找到适合自己的解决方案.  
  
下面是我建议的一种解决方案(GTD工作法):  
全身心投入当下正在处理的事情  
如果有新的需求过来, 2分钟内, 判断出这件事情的重要性.  
重要且十万火急的事情, 立即处理  
否则, 扔进自己的TODO List, 设定一个时间提醒自己再进行关注.  
完成手头的事情后, 从你的TODO List整理, 找出下一件最重要的事情.  
如果有多项重要紧急的事情, 无法独自完成, 将问题暴露给Leader.  
每天早晨看看自己的TODO List, 对当天一定要处理的事情有明确的认知.  
每周review回顾自己做过的事情.  
  
看看上面方法的核心是什么? 让所有的事情, 掌握在你的TODO List中, 对它们形成一种明确的管理. 当一切都了然于胸的时候, 你自然知道该怎么为这些事情分配自己的时间片.  
  
**无法找到当下最重要的事情?**  
  
上一个问题的建议方案中提到, 我们需要找到最重要的事情.  
  
那如何找到最重要的事情呢?  
  
从我以往的经验来看, 在新人期和成熟期碰到的事情中, 只需要问一个问题, 就能够判断大部分事情是否重要.  
  
"如果这件事现在不做, 会有什么后果?"  
  
比如:  
会导致这个版本不能如期发布  
会导致在线服务故障  
...  
  
不过, 通常我们得到的答案都是"没什么影响".  
  
新人很容易陷入无法判断优先级的困境, 这是人类天性的弱点. 我们会对未知产生恐惧, 进而产生焦虑, 进入一个恶性的循环状态.  
  
解除这种状态的思路, 就是让未知变成已知.  
  
**如何准确评估排期?**  
  
经理安排下来一个项目, 询问排期, 这个时候, 新人很容易慌乱, 不知道该如何评估.  
  
这个问题的产生, 和上面两个问题, 是一样的. 因为对新人而言, 并不了解一个项目到底需要多少时间才能完成.  
  
那就束手无策了吗?  
  
试想, "读完<钢铁是怎样炼成的>需要多少时间?", 面对这个问题, 你怎么解答?  
  
思路是不是这样?  
看这本书有多少页  
经验告诉自己, 读一页需要3分钟.  
然后计算出需要的总时间.  
  
软件开发的排期预估, 是同样的思想.  
  
大事化小.  
  
大事是不明确的, 无法直接给出工作量预估, 那么, 把它拆解到你能预估的小的事情, 就可以了.  
  
同时, 在你评估过工作量的事情, 最终完成之后, 去复盘, 看自己的评估是否准确, 如果有偏差, 思考问题出在哪里, 长此以往, 一些大事, 你也就可以直接评估出工作量了.  
  
**总结**  
  
通过上面三个问题, 可以看出一些共性, 三个问题, 都是通过把抽象的, 不明确的事务, 拆分成具体的, 明确的事务, 使我们心里更加有谱.  
  
如果你有这三个问题之外的其他问题, 不妨也试试这种思路.  
  
**成熟期常见问题分析**  
  
成熟期, 是工程师成长中的一个关键时期, 迈过去这个坎儿, 更多需要的是精神层面的东西. 所以, 这里提到的成熟期的三个常见问题, 都和"心"有关.  
  
**如何保持高速成长?**  
  
首先, 请再思考下"现象: 成长越来越慢"中, 提到的造成这个现象的原因:  
没有及时从"积累具体知识"调整到"提升抽象思维";  
没有及时从"被动成长"调整到"主动成长"  
  
仔细研究这两个原因, 都可以看做是在没有挑战性的状态, 形成了舒适区造成的.  
  
对大部分工程师而言, 参加工作, 是第一次正式的和社会接触, 基本正式脱离了"学习压力"所带来的约束. 而新人期又可以"轻松的"(主要指心里压力方面)通过公司获取被动性的成长.  
  
温水煮青蛙. 大部分人都会被煮熟的.  
  
请思考一个问题, "是不是你所学的20%就足以解决工作中80%的问题".  
请再思考一个问题, "是不是工作中剩下20%的问题, Google和咨询别人可以全部解决".  
  
进入成熟期后, 很快, 工作就很少, 甚至不能给你的成长产生驱动力了.  
  
剩下的, 就得靠自己. 告诉自己两件事, "主动提升", "主动思考下一职级的问题".  
  
我想通过这篇文章传递的核心价值, 就是这一点, 希望能帮助一些工程师(尤其工作3年内的), 意识到舒适区, 并跳出舒适区.  
  
**如何使自己更自信?**  
  
这个问题, 源自一周前和一位团队成员的沟通.  
  
我仔细的思考了自信的来源, 然后我认为, 自信是成就感驱动的.  
  
那成就感来自哪里呢? 我认为可以从内部/外部两个角度去看:  
内部: 来自自己的认可  
自己的技术或能力得到提升, 让自己觉得自己牛逼  
将技术或能力, 应用到实际项目中, 让自己相信自己真牛逼  
外部: 来自别人的认可  
将自己的实践, 讲出来让别人认可, 让别人觉得自己牛逼  
将自己的实践, 抽象总结出来, 帮助别人变得牛逼  
  
每个人因为性格的不同, 成就感获取的主要渠道可能不尽相同, 但结合自己的情况, 找到适合自己的成就感获取渠道, 通过成就感建立自信, 是一种有效的方法.  
  
不过, 自信和自大一线之隔, 谨慎.  
  
**XX同学已经月薪30K了, 赶紧跳槽求加薪吧**  
  
现实的诱惑, 也是容易让人迷茫的一个点. 不过, 如果对一些基本面有正确的认识, 可以帮助我们做出更加合理的决定.  
  
首先, 一个人的薪资, 是由专业能力, 软素质, 行业经验, 公司内经验, 稀缺度, 人脉, 运气等等方方面面的因素共同决定的.  
  
千万不要因为薪资的攀比而鲁莽的跳槽.  
  
跳槽的确会大概率增加薪资. 因为, 总有一些公司, 是到非常着急用人的时候才去招人的, 他们被迫给出更高的工资. 同时, 你上一家公司的薪水, 会给你提供一定的背书. 所以, 如果沟通表达能力好, 理论上短期是可以通过跳槽, 拿到超出当前能力的薪资水平.  
  
但是, 频繁跳槽会让阻碍工程师的长线发展. 一般而言, 公司是不会把核心职位交给刚入职半年一年的人的. 如果你频繁跳槽, 就注定长期处于一线. 现在的市场行情下, 天花板也就是20多K不超过30K.  
  
且不说钱的天花板, 关键在于长期处于一线, 你的斗志就会被消磨殆尽, 从而缩短你的职业生涯.  
  
因而, 从长线发展来看, 通过跳槽加薪, 并不是一种可取的选择.  
  
如果你想要高薪, 首先让自己变得优秀, 那么就应该静下心来, 在一家公司, 给自己定几个目标, 达成之后再去考虑离开的事情. 这样, 对大家都好.  
  
**总结**  
  
成熟期是整个职业生涯中, 最关键的时期. 快的, 可能一两年就能走过去, 慢的, 可能整个职业生涯都定格在成熟期.  
从上面分析中, 可以看到, 这些问题, 最终都和"心"相关:  
要想高速成长, 就得克服掉温水中的舒适.  
要想提高自信, 还得思考成就感获取渠道这么虚头巴脑的问题.  
干这么多, 还不能跳槽加薪迎娶白富美.  
所以, 要想迈过去这个坎儿, 必须有一颗非常强大的内心.  
  
**后记**  
  
最后, 希望看到这篇分享的同学们, 能够帮助你们延长奋斗的激情, 在这条路上, 走的更远.  
最后的最后.  
感谢和我一起进行各种管理实践的我的团队.  
感谢在我管理能力发展中给我帮助的Alex, 瀚哥, 海仪姐, 宁姐.

感谢一如既往支持我的家人.