

那些会阻碍程序员成长的细节[3]

原创 2018-03-07 Guooo 歪脖贰点零

前两篇文间几乎是想到那里就写到那里，没有分门别类的加以阐述。本篇延续以上两篇文章的思路，在之前的基础再追加。没有看过前两篇文章的同学可通过这两个链接回顾一下：

《[那些会阻碍程序员成长的细节（一）](#)》

《[那些会阻碍程序员成长的细节（二）](#)》

不能主动推动事物前进。 主动做一件事跟被动接受去做事，心情都是不一样的，做事效率更是千差万别。主动的人有更多的成长机会，反之在被动中不断的响应别人的任务，这与处于那个层次高低无关。如果处于高层级却不能主动推进任务前进，相信在这个岗位上也不会呆太久。

技术交流分享会仅仅是听听而已。 在一线城市这种机会特别多，如果你愿意，一天可以赶好几场。虽说是别人分享，如果没有好好利用，仅仅参加完就离开的话，效果几乎为零。可以事先熟悉讲师履历、分享的内容，带着自己的疑问听，不懂处记录在册，答疑环节将疑问点抛出，多认识些分享会上的人，加入分享会微信群会后继续讨论等等，都能更好的发挥一次分享会的效能。不要让参加技术分享会流于形式。

无法多场景切换角色。 刚入门时，醉心于开发还是有可能的，随着年限的增加，个人职责会发生改变，再有整块整块的时间去专心某一事务的机会不多，更多的是碎片化的时间切片，面对不同同事、客户、领导、合作伙伴等等，如果不能快速响应各种变化，只能用疲于奔命应付来形容了，工作的条理性更无从谈起。你需要快速切换角色，融入不同环境，能抽象，也能具化，能全局把握，也能细节深入，这样处理起问题才能得心应手，迅速成长。

容易陷入细节。 或者说很纠结细节问题，相对的另一面会导致全局观缺失，偶尔还会钻牛角尖。过于追求细节完美的人，也就是完美主义者，事实完美在现实生活中不可能存在。陷入细节而不自知轻则项目延期，重则错过市场窗口期。把握一个度很重要，需要以大局为重时，就需要牺牲局部的不完善。

不插手开发任务外的其他事务。 临时任务在所难免，特别是创业团队，更是没有明确界线的岗位分工，需要时，直接顶上，目标就是把事情做成。仅关注开发任务那点事，显然不足以为更高的岗位做技能储备，如果力所能及又不占用过多时间的话，建议去锻炼一下。经历都算自己的，成长过，谁也拿不走。

不注重关键节点的记录留存。 人员流动、业务变更对软件后续的维护都是挑战，仅靠脑袋记，显然不现实。很多朋友说，代码里都有啊，看代码就行了。如果必须拿代码才能说事的话，只能说你不成熟，设计文档、流程图、数据表格这些辅助理解的东西简直难道是摆设？特别是一些复杂的业务流程设计、复杂算法设计、疑难BUG解决方案等等。虽然敏捷开发不主张凡事以文档必备为准，但必要的文档留存很重要，形式可以不限。

不经意间，这些小细节慢慢写成了一个系列。俗语讲：魔鬼在细节中，小处做到极致，养成的好习惯，也是让自己未来的技术晋升之路更加坚实。



扩展阅读：

