我最开始不是做软件开发的，是一个售后技术支持工程师，你懂的，就是公司卖出的设备坏了，我就到现场去鼓捣两下，换个零件，重启下系统什么的。后来我转行做软件开发，颇经历了一些曲折，遇到了很多实际的困难，再后来我成了程序员，干上了手艺活儿，就这么一路过来了，还出了两本书，《Qt on Android核心编程》和《Qt Quick核心编程》。 后来我也和一些刚入行的开发人员共事过，有的朋友可以很快度过适应期，有的朋友则会花费比较长的时间，遇到各种不那么酸爽的事儿。你知道，我是一个爱瞎琢磨的程序员，根据我的个人经验和我观察别人的结果，我发现，如果一个新手可以坚持并实践这么几点，就能够很快从青涩变成熟，完成最初的角色转变，融入到让你欢喜让你忧的软件开发工作中来。 接纳自己是一张白纸这个事实 我觉得这是一个首要的前提。也许你很优秀，有很强的学习能力，有强大的信念，有超强的小宇宙，有百折不挠的韧性……但是，你没做过，你确实是一张白纸。这是一个客观事实，我们必须要要承认。我们所做的一切，都是在接纳现实的前提下展开的，唯有承认这一点，其它美好的事情才有可能。 我们平静的看待这一点，不要妄自菲薄，也不要好高骛远，从一条线开始，慢慢绘制自己的蓝图，一切可能就会慢慢变成现实。 关注自己能做到什么 我们是一张白纸，但我们还是可以做很多事情，我们通过学习，将能做更多事情。我们会遇到各种各样的问题，会产生自己这个不会、那个不会，可能安装一个IDE都会出问题，可能三行代码都会遇到10个编译错误，可能Run起来你不知道结果是对是错是否符合预期……这都没关系。 我们要看自己今天做到了什么，明天又能做到什么，一周后能做到什么，一月后能做到什么。我们关注积极的方面，不要因为各种挫败而将自己陷在不良情绪的泥潭里，自怨自艾、自我否定、羡慕嫉妒恨、失望、沮丧……这些情绪都是我们的敌人，都是有不良居心的魔鬼，它们只会坏我们的事儿。我们要逃离它们，不给它们机会，我们的对策就是，看我能做什么，看我将来能做什么，看我需要做哪些事情才可以达到明天、下一周、下一个月的目标。这样我们就能积极行动起来，就会把时间花在有价值的地方，我们也会慢慢更有价值。 如饥似渴地学习 我们是一张白纸，就要如饥似渴地学习，抓住一切机会学习，读书，求人，读代码，上班8小时，下班还有更多时间，我们可以把一切可以利用的时间都投入到相关知识和技能的学习中来。 当你连一个DEMO都不知道如何创建，当你连一个Hello World都跑步起来，当你弄不懂变量、操作符、函数、对象、类、实例、控制语句，当你不知道那些被老手们当做常识的递归、链表、单例、观察者、MVC、CRUD、RBAC……你唯一能做的，就是把自己变成一块干燥的海绵，持续不断地去吸收各种知识。 这个过程是快乐的，幸福的，你会发现随着你掌握的知识越来越多，随着你能解决的问题越来越多，你会发现今天的自己和昨天的自己不同，这个月的自己和上个月的自己不同，每一天、每一周、每一月都是一种惊喜。就让自己变成一个小孩，收获简单的快乐。 别怕犯错 很多新手怕犯错，其实不必。犯错不等于失败，犯错也是一种学习。 我们不必担心别人因为自己犯错而看轻自己，要知道，此时此刻我就是一张白纸啊，我没有经验啊，我什么都不会啊，我不犯错才是奇怪的事情。 就像小孩子蹒跚学步，跌倒是很正常的事情。可是它会在跌倒中收获宝贵的经验，摔几次，摔几天，它就学会了。 犯错一点儿也不可怕，可怕的是不总结、不归纳、不吸取教训。 迎难而上 我遇到一些新入行的朋友，面对一个任务时会畏缩，说这个我不会啊，说这个我从来没弄过啊，说我搞不定啊…… 当然会遇到困难！作为新手，你遇不到困难才是见了鬼呢。要把困难当做机会，只有做你眼下搞不定的事情，你才能提升，你明天才能搞定更多的烂事儿。 所以，要迎难而上，把困难当做猎物，当做锻炼的机会，要见猎心喜，用于挑战自己。 记录问题和心得 我觉得这一点非常重要。在初始阶段，记录遇到的问题，记录学习心得，总结解决问题的经验，将会对我们后来的进步提供非常大的帮助。我们要模仿牛的反刍，不断从有限的实践经验中总结、消化，获得更多的营养。 同时这也会帮助我们养成良好的工作习惯，良好的习惯，会让我们终身受益。 适时求助 有些人喜欢闷头黑搞，搞不定也吭声，要么把困难扔在那里不管，自己也停在那里，要么继续黑搞，东搞搞西搞搞始终搞不定，停在那里。 我觉得应该给自己设定一个期限，在期限到达时还搞不定，就向别人求助。 向别人求助是正常的事情，每个人都可能会求助别人，一个人不可能搞定所有事情。不要担心别人太忙没时间帮你，也不要担心欠别人的人情。要知道“帮助别人”这件事本身，就是一件快乐的事。在帮助别人的过程中，提供帮助的人会收获快乐。多数程序员都很乐意帮助别人。真的。你获得了帮助，你在遇到别人的求助时也乐于提供帮助，这就够了。别的不用想太多。 提前告知上级你真的不能搞定 没错。就是这样，一旦你经过了种种努力，确信自己不能搞定手头的事情，一定要尽早告知你的上级和小伙伴。 一个软件项目的周期，通常是由最晚结束的那个任务决定。 假如你是一个新手，碰巧Take了一个任务，你当然不希望自己的任务是最晚结束的那个。可实际情况表明，你有很高几率扮演那个角色。我要说的是，你需要清楚判断自己手上的任务的状况，如果你真的不可能搞定（或者不可能按期搞定），一定要提前告知你的上级，把你的状况、任务的状况都说明白，这样你的上级才可能有时间做必要的调整（比如分配新的资源、延长时间等）来应对可能的项目延期。如果你在最后一刻才告诉别人，那就没有补救的机会了。真的。 别担心别人看轻你，即便经验丰富的优秀程序员，也会碰到搞不定的事情。软件开发的性质就是如此，永远会有超越你能力边界的问题蹦出来。这很正常的，让别人知道，让大家能想办法补救，这就是很好的措施。 向优秀的同伴学习 三人行必有我师。 想让自己变优秀的一个好办法就是和优秀的人走在一起。 你一定要知道（自己观察、别人告诉你、上级告诉你）你所在的团队中的哪些程序员在哪方面比较优秀，比如张三代码简洁能自解释，比如李四设计能力强，比如王五总能搞定技术难题，比如赵六具有卓越的影响力，比如钱八善于分析问题，比如赵十能深入浅出的讲解技术问题，诸如此类，如果你能够清楚，一方面可以在自己遇到问题时向正确的人求助，另一方面，你也可以主动学习。 见贤思齐，自古就是这个道理。有时很多变化是在潜移默化中发生的，你所处的环境、你交往的人，往往会以润物无声的方式影响你。孟母三迁，只为择邻，就是这个道理。 让上级为自己指定导师 如果可能，让你的项目经理或者领导为你指定一个导师（mentor），让导师来帮助你制定你的阶段目标，并且让他来指导你达到目标。 有个实践，叫OJT培训（On the Job Training）。不过不一定每个公司都运用这种机制。如果你所在的公司没有，那可以让上级给你找一个导师，或者自己找一个。 有经验的导师，既能引导你少走一些弯路，又不会越俎代庖直接帮你把事情搞定。这样你就会自己走在正确的方向上，在节省不必要的精力浪费的同时，又能自己收获实作的经验。