

程序员的职业素养

笔记本： 编程与实践

创建时间： 2016/2/3 9:02

更新时间： 2016/2/5 9:10

作者： Barret Ren

1. 就我所见，技术人员经常太容易说“是”，旺旺在没有明确目标和期限的情况下，就草率的给出了确认的答复，而且并不将其视为自己的一种承诺
2. 花三分的力气去抵制无力的需求，可以节省十分甚至二十分的开发时间；相反，自欺欺人地说服自己凑合接受了无理需求，往往会非常被动乃至无法脱身。
3. 代码中难免会出现bug，但这并不意味着你不用对它们负责；没人能写出完美的软件，但这并不代表你不用对不完美负责。
4. 成熟的专业开发人员知道，聪明人不会为了发布新功能而破坏结构。结构良好的代码更灵活。以牺牲结构为代价，得不偿失，将来必追悔莫及。
5. 要想证明软件易于修改，唯一办法就是做些实际的修改。如果发现这些改动并不像你预想的那么简单，你便应该改进设计，使后续修改变简单。
6. 一般认为对科工作软件不断地做一系列修改是危险的。错！让软件保持固定不变才是危险的！如果一直不重构代码，等到最好不得不重构时，你就会发现代码已经“僵化”了。
7. 将自己的职业发展寄希望于雇主的软件开发人员将会很惨
8. **一周有168小时，给你的雇主40小时，为自己的职业发展留20小时，剩下的108小时再留56小时给睡眠，那么还剩52小时可以做其他的事。或许你会觉得这样做只会让人精力枯竭。恰恰相反，这样做其实能让你免于枯竭匮乏。**
9. 不写代码的结构式必然遭殃，它们很快会发现自己跟不上时代了；不学习新语言的程序员同样会遭殃，他们只能眼睁睁的看着软件业径直向前，把自己抛在后面；学不会新原则和技术的开发人员必将沦落，他们身边的人都日益卓越。
10. 专业人士敢于说明真相而不屈从于权势。专业人士有勇气对它们的经理说“不”。
11. 许诺“尝试”，就意味着你承认自己之前未尽全力，承认自己还有余力可施；许诺“尝试”，意味着只要你再加把劲还是可以达成目标的。从本质上讲，承认“尝试”就是一种不诚实的表现。你在说谎，你这么做的原因，可能是为了护住面子和避免冲突。
12. 有些事情先前你可能没预料到，这很现实。但如果你仍然希望自己可以不负众望，那就赶紧去调整别人对你的预期，越快越好。
13. 我发现，要精熟掌握每项技艺，关键都是要具备“信心”和“出错感知”的能力
14. 疲劳的时候，千万不要写代码。奉献精神和职业素养，更多意义上指要遵循纪律原则而非成为长时间工作的工作狂。要确保自己已经将睡眠、健康和生活方式调整到最佳状态，这样才能做到在每天的8小时工作时间内全力以赴
15. **我知道许多人在写代码时喜欢戴着耳机，但愿音乐真的能够帮到他们。但同时我也怀疑，真是的情况是，音乐正将它们带入流态区。**

16. 粗暴相对的反应方式通常都是因为流态区所致。被他们从流态区中拉出来，或者当你正努力进入流态区却被其他人干扰时，你可能都会十分生气。不管哪种情况，粗暴方式都与你如何看待流态区有关。
17. 如果磅数唾手可得却让自己一个人堵在那儿，是很不专业的表现
18. 先行编写的测试时进攻，时候编写测试的作者已经受制于已有代码，他已经知道问题是如何解决的。与采用测试先行的方式编写的测试代码比起来，后写的测试在深度和捕获错误的灵敏度上要逊色很多。
19. 既然你用自己的事件联系，就不必限制在老板规定的语言和平台。可以选择你喜欢的语言，练习你喜欢的技术。
20. 参加的会议太多，只能证明你不够专业。邀请你参加会议的人并不负责管理你的时间，为时间负责的只有你。
21. **凡是不能再5分钟内解决的争论，都不能靠辩说解决。争论之所以花这么长的事件，是因为各方都拿不出足够有力的证据。所以这类争论依据的不是事实，而是信念。**
22. 优先级错乱是自我麻醉的谎言，因为不能面对真正需要做的事情，所以我们告诉自己，其他事情更重要。我们知道这不是真的，但还是用他来欺骗自己。
23. 让你的团队和主观知道你正深陷困境中。告诉他们你所指定的走出困境的最佳计划。请求他们的额支援和指引。避免制造意料之外的诧异。没有东西比诧异更令人愤怒和失去理性。诧异会让你的压力增大十倍。
24. 当任务琐碎且无足轻重、和另一个人一起工作显得浪费时，可以一个人工作。但一般来说，和其他人紧密协作，在大部分时间段中结对工作，是最好的做法。