毁掉你代码的35个恶习

对于新人程序员来说，良好的工作习惯能够提高自己的工作效率，改善自己在同事心目中的形象。

今天就分享一下web开发专家Christian Maioli总结的导致程序猿效率低下，代码像意大利面条一样难以维护的35条恶习（归为代码组织、团队工作、写代码、测试与维护四大类）。

**代码组织**

1.总是说“一会弄好”，但从来不兑现。（缺乏任务管理和时间管理能力）

2.坚持所谓的高效、优雅的“一行代码流”，事实上，可读性才是最重要的，聪明是第二位的。

3.无意义的优化。（类似网页大小之类的优化最后再做）

4.不注重代码样式和风格的严谨。

5.使用无意义的命名。

6.忽略经过验证的最佳实践（例如代码审核、TDD、QA、自动化部署等，推荐阅读软件开发必读经典著作：Making Software：What Really Works，and Why We Believe It）。

7.给自己埋雷。（例如使用不会报错的库或者忽略例外）

**团队工作**

8.过早放弃计划

9.坚持一个无效的计划

10.总是单打独斗。（必须强迫自己与团队分享进度和想法，避免错觉，提高效率）

11.拒绝书写糟糕的代码。（日程紧迫的时候可以写一些“糟糕”的代码，这是程序员的能力而不是bug，当然，有时间的时候一定要回头偿还“技术债”）

12.抱怨他人。

13.不与团队分享所学。

14.向主管/客户反馈的速度过慢

15.不会充分利用Google

16.看重个人编码风格

17.带着个人情绪看待他人对自己代码的评论和注释

**写代码**

18.不懂优化策略

19.使用错误的工具。

20.不追求对开发工具和IDE的精熟。

21.忽略报错消息。

22.迷恋称手的开发工具。（不同类型的开发任务需要匹配对应的最佳开发工具，例如Sublime适合动态语言，而Eclipse适合Java，如果你喜欢vim或emacs，并不意味着能用这些工具干所有事）

23.不注重代码中赋值的可配置性。（不养成把代码中的活动部件分离出来的习惯，会导致技术债暴增）

24.喜欢重新发明车轮。

25.盲目地剪切/粘贴代码。

26.应付差事，不求甚解，不花时间搞清楚项目运作的机理。

27.对自己写的代码过度自信。

28.不去考虑每一个设计、方案或者代码库的“副作用”。（一个成功的用例并不意味着“万灵药”）

29.在一个地方卡住了但坚持不呼救。

**测试与维护**

30.只去写能通过的测试

31.重要项目中忽略性能测试

32.不去核实代码是否真的可用，没有养成开发中及时快速测试的习惯。

33.重大改动延迟推送

34.抛弃和逃避自己的代码。

35.忽略其他非功能性需求。（例如安全和性能，准备一份这方面的清单，忽略这些会毁掉你的所有成果）

改变以上这些不好的习惯，小白程序员也能变成程序员大牛！