

# 注释是恶魔，请不要再写一行注释 - 文章



原文出处: [leotsai](http://leotsai.com)

你可以从你们现在项目里面随便找几处注释，看看写注释的代码是不是存在如下两种毛病之一：

- 1. 命名不准确；
- 2. 方法太长（超过50行）。

如果你找到的代码没有出现上面两种毛病而注释依然存在，那你再看看这个注释是否有实际意义，是不是这个注释不要也无所谓呢。

## 注释是恶魔

这个观点可能你第一次看到，你可能很难接受，因为写了这么多年的注释，你从未想过注释居然是恶魔，所以，你看到这个观点的时候可能就会本能的找出1000种理由反对（绝对不可能实现啊什么的），但是，这个观点并不是今天才出现，相信很多年前就有人提出，现在已被越来越多的人认可。

我第一次接受到这个观点还是从一个美国客户（十几年编程经验的技术大牛）那里，2011年，他让我们不要写注释。他当时主要意思是我们写的中式英语他猜起来太费劲，所以他后面又安慰我们说“好的代码是不需要注释的”，而我从此就将他后面那半句话奉为至宝。

注释是恶魔，它将我们的代码变得很难理解。就像本文开篇说的，你可以找找你们项目中出现注释的地方，要么命名不准确，要么方法太长。你可以随机找10处注释，看看有几处是恶魔，欢迎贴到评论中。

举一个以前项目中的例子吧，命名不准确的例子：

```
1
2
3
4
/// <summary>/// 管理员是否可以审核该申请/// </summary>
public bool IsAudit { get; set; }
```

在这个例子中，其实将”is”换成”can”就不需要注释了。

写注释让代码更难读。

首先，如果一个程序员可以随便写注释，那么他对命名准确性和方法长度的控制就不会那么在意，写代码更随意，代码质量比不能写注释的程序员更大几率低下。

其次，代码注释只是在写代码的时候提供说明，如果读代码都依靠注释的话，那一个类被另一个类引用

来引用去的就根本没法阅读了。

所以，“写注释是为了让代码更易读”本身就是站不住脚的。

不写一行注释？根本就做不到！

这句话可能从你阅读本文开始在心里面重复了无数遍，这也是大多数人的心声。

其实前面说的，写注释让代码更难读的观点很多朋友从内心上是认可的。因为确实没有办法啊，有的方法业务逻辑复杂，不知不觉方法已上百行，有的命名还是中西结合的，不写注释自己第二天就读不懂了。所以，真是纠结，内心承受百般折磨。

写到这里，突然想起在园子里看到的一个笑话，说一个公司的产品每年都在更新换代，因为每年新招的程序员都要把程序重新写一遍。

“零注释”根本做不到？如果你从刻开始怀疑自己的这个观点，那你就可能做得到。

如何做到不写一行注释

- 1. 从现在开始，强迫自己不要写注释。
- 2. 控制每个方法不超过50行，用方法定义来描述方法的实现逻辑。
- 3. 变量命名不要太过随便。

本文想要告诉大家的是，零注释一点都不难。我们团队大约从2012年开始全面执行零注释，后面经历2个产品项目，多个外包项目，积累的经验越来越多，获得的质量效果越来越好，零注释越来越深入人心。

零注释这个编码规则也是我们团队近些年质量建设非常重要的里程碑之一，再此分享给大家。如果能够影响你一点点，那都足够了。

附2个我们的代码片段

虽没有注释，大家不妨猜猜这两个方法做什么用的。

#### 1. 查询的例子

```
public void Submit(Guid userId, string content, string text, double lng, double lat, string address)
{
    using (var db = base.NewDB())
    {
        var issue = new Issue(userId, content, text, lng, lat, address);
        db.Issues.Add(issue);
        db.AddIssueLog(IssueLog.CreateOnSubmit(issue.Id,
        db.Users.GetNickName(userId)));
        db.SaveChanges();
        issue.GenerateSerialNumber();
        if (!string.IsNullOrEmpty(SettingContext.Instance.AdminOpenIds))
```

```
{  
    var name = db.Users.GetName(userId);  
    var totalPendings = db.Issues.Count(x => x.Status ==  
IssueStatus.None && x.IsDeleted == false);  
    var adminOpenIds = SettingContext.Instance.AdminOpenIds.Split(',');  
    foreach (var openId in adminOpenIds)  
    {  
        var message = new PendingProcessTemplateMessage(openId,  
issue, name, totalPendings);  
        db.WeixinScheduledMessages.Add(message.ToWeixinScheduledMes  
sage());  
    }  
}  
db.SaveChanges();  
}
```

拿高薪，还能扩大业界知名度！优秀的开发工程师看过来 -> 《[高薪招募讲师](#)》

3 赞 7 收藏 [23 评论](#)



合作联系

Email: [bd@jobbole.com](mailto:bd@jobbole.com)

QQ: 2302462408 （加好友请注明来意）

更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子

[头条](#) - 分享和发现有价值的内容与观点

[相亲](#) - 为IT单身男女服务的征婚传播平台

[资源](#) - 优秀的工具资源导航

[翻译](#) - 翻译传播优秀的外文文章

[文章](#) - 国内外的精选文章

[设计](#) - UI, 网页, 交互和用户体验

[iOS](#) - 专注iOS技术分享

[安卓](#) - 专注Android技术分享

[前端](#) - JavaScript, HTML5, CSS

[Java](#) - 专注Java技术分享

[Python](#) - 专注Python技术分享