



链滴

ThoughtBot 官方给出的代码审查指导原则

作者: [88250](#)

原文链接: <https://ld246.com/article/1361937186068>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>这篇文章的内容由 ThoughtBot 在 GitHub 上官方主页提供，指导你如何在 GitHub 上进行代码审查和如果让别人审查自己的代码。</p>

<h3 id="toc_h3_0">针对所有人的审查</h3>

接受这样的事实：很多编程上的主张都是一种个人观点。应该讨论它们的利与弊，提出你的倾向点，迅速的达成一种解决方案。

提问，而不是命令。（“把这个变量命名成<code>:user_id</code>你觉得怎样？”）

请求说明。（“我不明白。你能解释一下吗？”）

避免代码的归属之争。（“我的”，“不是我的”，“你的”）

避免使用一些会被认为是有关人身特征的词语。（“笨蛋”，“愚蠢”）要把所有人都看作是有魅力的、聪明的、善意的。

要明确。要记着并不是每个人都能理解你的意图。

要谦虚。（“我不能确定——我们来分析一下。”）

不要用夸张修辞语。（“总是”，“从不”，“永远”，“毫无...”）

不要讽刺。

展现真实的你。如果你不是幽默型的人，不喜欢使用一些表情符号或动画gif图，不要勉强。如果你是这种人，请自信的发挥。

如果有太多的“我不理解”或“另一种方案：”的评论，请专门针对这个人进行交流。可以把你线下的交流总结成一个帖子附在后面。

<h3 id="toc_h3_1">让别人审查你的代码</h3>

对审查者的建议表示感激。（“谢谢提醒。我会把它改正。”）

理解审查是对事不对人。审查的是你的代码，而不是你。

解释为什么代码写成这样。（“因为xxx原因我才写成这样。如果我把这个类/文件/方法/变量改名会更清晰些吗？”）

整理所作的改动，在以后的迭代中重构它们。

在做修改的版本上注明代码审查的链接。（“Ready for review:http://github.com/organization/project/pull/1”）

push提交要基于最早的一轮反馈，并形成一个新的分支。等这个分支上的任务完全完成了再并。这让审查者能够根据早先的反馈找到你的单独的更新。

努力站在审查者的立场上理解。

争取回复每个评论。

直到最后一个人退出登录后再合并分支。

直到持续集成测试(TDDium, TravisCI,等)告诉你这个分支的测试套件通过后再合并分支。

<h3 id="toc_h3_2">代码审查的过程</h3>

<p>先要清楚你提交的代码的必要性(是修补bug，提升用户体验，重构...)。然后：</p>

针对你感觉非常好的地方以及不是很好的地方与作者交流。

找出既能解决问题又能简化代码的方法。

如果讨论变得过于哲学或理论，把讨论转到线下，做成一个有规律的每周五下午的讨论会。同时是否采用你提出的实现方案，让作者自己做决定。

提出你的实现方案，但要表现出作者也在考虑这种方案。（“你觉得这里用一个自定义校验如何”）

努力理解作者的立场。

pull请求登出时，加一个:thumbsup:或“可以合并了”的注释。

<h3 id="toc_h3_3">关于程序风格样式的评论注释</h3>

<p>审查者应该对那些不符合样式指导的地方进行注释。例如这样注释：</p>

```
<pre><code>[Style](./style):
```

</code><p><code>> 按名称的字母顺序排列多个路由。

</code></p></pre><p></p>

<p>对上面这个提醒的一个回复的例子: </p>

<pre><code>哦。你眼真尖, 谢谢。已在 a4994ec 修复。

</code></pre>

<p>如果你不同意某个指导原则, 请在指导repo里创建一个问题, 而不要再代码审查中争论它。同时请运用这个指导原则。</p>

<div>

[英文原文:

code review]]

</div>

<div>

]]

</div>

<div>

转自: <http://www.aqee.net/code-review/>

</div>