

# 代码审查规范

## 0.前置说明

原则上来说，审查者只有查找代码问题的义务，没有修复代码问题的义务。审查者有权要求提交者修改相应的代码问题。

在第 2 部分的代码审查点中，并不是每条审查点都必须严格遵守的。当提交者能够给出足够的理由是，可以违反个别规则，但该理由必须有书面的记录（文档或注释）。

## 1.我们应该这样看待代码审查

代码审查能给我们带来什么？

- 尽早发现问题代码。
- 提高团队成员对项目整体的代码熟悉度。
- 提高新手的代码经验。
- 提高团队成员的项目拥有感。

因此，我们应该从以上多个方面评估代码审查执行的成效。

当你认为代码审查成效不佳时，请优先考虑如何能提高团队的代码审查质量，而不要持消极的代码审查态度。

代码提交者即是第一个审查者。

- 开发人员在提交代码之前应先自行审查代码。

审查者应严格对待代码。

- 代码是我们公共的资产，我们不能允许任何人以任何理由弄脏它。

- 代码审查时深入追究代码问题，是负责任的表现。

#### **审查者应宽厚对待代码提交者。**

- 当发现代码问题时，审查者应给予提交者足够的尊重与耐心。
- 当发现代码问题时，与“如何修复问题”同等重要的是“如何帮助提交者进步”。
- 当发现代码问题时，审查者不应直接给出“你这里违反规定了”的结论（除非这样的结论真的很明显）。而应优先询问提交者“你这里如此实现，是否有什么特殊考虑”。
- 审查者在发现代码亮点时,应给予提交者适当的表扬（可直接在 QQ 群中截图发出）。

## **2.代码审查时我们应该关注这些点（按优先级排序）**

#### **有编译器警告吗？**

- 不应出现任何编译器警告。
- 对于警告中无明确提示的信息，应尽量查证相关资料以改善之。

#### **有 Suggestion 级别以上（包含 Suggestion 级别）的 Resharper 警告吗？**

- 不应出现 Suggestion 级别以上（包含 Suggestion 级别）的 Resharper 警告。
- 团队共享一份 Resharper 的配置文件，该文件定义了团队公认的应该遵守的 Resharper 规则。
- 对于 Resharper 无法正确提示的情况，代码审查时可以在得到两位团队成员的认可后，使用 Resharper 注释的手段消除之。

#### **你能读懂代码吗？**

- 当审查者发现无法理解的代码时，应当要求提交者改善代码的可读性。必须保证团队中代码经验最欠缺的成员也能读懂技术能力最高的成员所写的代码。
- 当无法通过优化代码的方式来提高可读性时，审查者应要求提交者添加必要代

码注释。

- 当代码注释难以系统地表达代码的设计思路时，审查者应要求提交者编写相应的技术文档，同时应告知文档组负责人。此后由文档组负责追讨该文档。

### 代码符合团队代码规范吗？

#### 代码出现在恰当的位置吗？

- 项目文件目录应清晰，文件夹命名应合理。
- 对于可以作为公共组件的代码，由审查者向公共组件组负责人提出代码入库申请。之后，由公共组件组负责追讨入库代码。

#### 你见过相似的代码吗？

- 严禁出现完全一致的函数。
- 对于相似的函数，应该通过抽取参数的方式合并之。

#### 引用有可能是 `null` 吗？

- 当某个引用变量可能是 `null` 时，应该在调用它的方法前进行判空处理。
- 当某个引用变量被设计成不可能为 `null` 时，不能为其添加判空处理。

#### `try-catch` 块使用合理吗？

- 代码审查遇到 `try-catch` 块时，着重考虑 `try` 块中是否使用了需要释放的资源，如果有，一定要添加 `finally` 块。
- 避免出现 `catch` 未知异常的情况，除非提交者能给出足够的理由。
- 避免出现 `try` 块的嵌套。

#### 类和函数的体积过大吗？

- 类应控制在 500 行代码以内。
- 函数应控制在 50 行代码以内。

嵌套的 **for**、**foreach**、**while** 以及 **if** 结构可以想办法避免吗？

- 对于复杂嵌套的 **if** 结构，优先尝试合并条件分支以使代码更易维护。
- 如无法避免结果的嵌套，请尝试将嵌套内层的代码抽取为单独的函数。例如：

```
for(var student in allStudents)
{
    for(var book in student.BorrowedBooks)
    {
        Book.SetTimestamp();
    }
}
```

可修改为：

```
for(var student in allStudents)
{
    SetTimestampForBorrowedBooks(student);
}
```

代码符合 **SOLID** 原则吗？

- 审查者发现有违反 **SOLID** 原则的代码时，应要求提交者改正或给出合理解释。