

# 协作软件开发文化的最佳实践

协作、透明和代码复用不仅仅存在于开源社区中。成功的软件开发人员以同样的协作方式构建软件——这种方法被称为“内部开源”。内部开源的最佳实践将帮助您的团队创建一种高效协作的开发人员文化。



## 透明

透明性是一切内部开源文化的基准，就如同更广泛的开源社区一样。

- 尽可能广泛地在整个组织中共享项目。对于需要更高安全性等级的项目，可以根据具体情况进行例外处理。
- 确保您的团队拥有可以公开交流和一致构建的工具和流程——并且正在使用这些工具和流程。如果没有，找出原因并进行调整，直至建立一个可靠的例行程序。
- 确认您的工程计划有充足的资源和领导的支持。
- 给予开发人员足够的自主权，以便他们为其直属团队之外的项目做贡献。



## 分配工作量

为了有效地采用内部开源实践，贡献者需要能够轻松地跨多个团队和组织部门工作。

- 通过将控制权分配给人数较少的参与者小组，可以让审批和审查更加高效。一支跨职能的小型决策者团队还有助于团队坚持质量标准并获得领导支持。
- 将繁重的工作量分散到异地分布的开发人员，甚至是非开发团队成员。
- 切记，非开发人员是内部开源社区的宝贵财富。

项目经理可以快速评估项目状态，对项目时间表所做的更改可以直接与讨论中的代码联系起来。

应该鼓励技术文档撰写人与项目工程师一起工作，精确地记录项目工程师创作的代码特性。

安全工程师可以审查代码的合规性和与安全性相关的缺点，从而在工作流早期阶段预防问题发生。



## 鼓励沟通和持续审查

公开分享工作和反馈可以给内部开源社区带来更多的构想——和更多的代码。

- 充分利用版本控制系统的注释、标记和审查功能。
- 使用复刻和分支，这样开发人员在尝试不同的更改时不会影响原始项目。这些工具还能让开发人员方便从现有解决方案开始，而不是每次都编写新的代码。
- 鼓励开发人员通过在代码提交中标记其他人尽早并经常共享他们的最新工作。
- 如果有人发现了一个错误，则应该创建一个议题进行描述。当您更新代码时，标记给相同的贡献者，这样他们就知道错误已经修复了。
- 让软件开发生命周期的每个步骤（如测试和部署）对所有人都可见。



## 不要忘记文档编制

随着内部开源项目变得越来越复杂，译码文档编制保持了工作的连续性和一致性。

- 文档是默认选项，而不是事后补记。有效的文档编制可以使外部贡献者清楚地了解您的决策过程，并避免浪费资源和重复。
- 将自述文件和贡献指南添加到所有存储库中，以帮助所有人理解基本用法和项目最佳实践。
- 使用文档来帮助新的贡献者和其他参与者加入和适应您的流程。
- 定期更新译码文档，使用户指南和手册与最新版本的代码保持同步。

想了解GitHub如何帮助您企业吗？  
告诉我们您的更多需求。

联系GitHub:

[github.com/enterprise](https://github.com/enterprise)  
[sales@github.com](mailto:sales@github.com)