

如何用好开源软件

willem.jiang@gmail.com

2013-10

开源软件利弊

- 源代码级产品能自由定制
- 对用户的要求比较高
- 可以自己进行客户化工作
- 如何维护相关的代码
- 不需要付版权费用
- 缺乏企业级别的文档和支持

开源软件使用过程中的误区

- 开源软件还需要支持吗?
- 既然代码都有了，多找几个人研究一下就行了!
- 花半天时间修补的一个bug，我得把它好好藏起来。
- 开源软件省了我不少时间，改改程序什么都出来了。
- 感谢写开源软件的活雷锋们，我可以直接在你们的软件基础上写商业软件赚钱。

开源软件如何选型?

开源软件选型

- 项目的软件许可
- 项目的成熟度
- 项目的社区
- 项目的口碑
- 是否有商业支持
- 结合需求做POC测试

开源软件出现问题怎么办?

开源软件出现问题怎么办?

- 选用最小功能集进行测试
- 利用社区资源寻求帮助
- 提交Bug或者patch
- 选用商业产品支持

开源软件与商业支持

没有免费的午餐

- Apache 的项目基本上都是有公司支持的
- 软件不需要付费不等于支持也是免费的
- 开源软件提供的支持
 - 文档，代码，邮件列表，JIRA
 - 不能保证问题得到及时解决
- 开源软件的商业支持
 - 培训，POC，PS

开源软件的本质

- 开源软件不是一个简单的许可
- 社区胜于代码，要融入到开源软件开发生态链中
- 大家在共同的目标下贡献力量，并分享成果
- 时间和金钱之间关系

开源开发和商业开发的区别

- 项目的管理和计划
 - 项目经理，发布时间
- 开发与测试
 - 分离还是合二为一
- 销售与支持

开源软件开发

在家上班

代码大家都看得到

在经营自己的事业

如何工作

Internet

No matter where you are

MacBookPro



工作内容

- JIRA (Company, Apache)
- 邮件列表
- 维护CI稳定
- IRC
- Coding
- Release

如何考核工作情况

- 放羊式管理
- 邮件驱动的开发方式
- 完成商业服务支持
- 自由选择工作内容
- <https://www.ohloh.net/accounts/ningjiang>

融入开源社区

建议大家从今天开始就尝试一下

如何为开源社区做贡献

- 使用开源社区软件
- 提交Bug 报告
- 完善开源社区文档
- 参与邮件列表讨论
- 成为开发者贡献代码
- 参与开源社区建设

提交bug 报告

- 检索是否有类似的错误
- 详细描述软件的版本和使用环境
- 描述错误详细内容
- 给出测试用例
- 登录到JIRA 或者其他的Bug管理工具提交报告或者补丁
- <http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>

完善开源软件文档

- 文档是学习开源软件基础
- 从示例代码入手
- 代码是最详细的文档
- 在抱怨开源软件文档不全的时候有没有想到自己动手帮助完善
- 学会使用邮件列表以及wiki

参与邮件列表讨论

- 提问的艺术
 - 学会使用搜索引擎
 - 不要重复问简单问题
- 互重，真诚，以技术为基础的沟通
- 从提问题到帮助别人解决问题

成为开发者贡献代码

- 熟悉开源项目的代码
- 学会自己构建(build)项目
- 注意运行测试，保证自己修改的代码不
break build
- 多学，多问，多思考
- 参与邮件列表的讨论，帮助软件用户解决问题

查找邮件列表

- 在每个项目的网站上都会有邮件列表信息
- dev@ 开发者之间相关的讨论
- commits@ 自动转发源代码提交
- users@ 用户之间的讨论内容，开发者也会订阅的
- http://mail-archives.apache.org/mod_mbox/

获取最新的代码

- Apache使用Subversion管理代码
- <http://svn.apache.org/repos/asf/>
 - 美国和欧洲各有一个镜像服务器
 - ViewVC <http://svn.apache.org/viewvc>
- 目前Apache项目也在试验使用git
 - <http://wiki.apache.org/general/GitAtApache>

如何创建patch

- 使用diff (svn diff 或者 git diff)
- 为patch创建一个JIRA
 - 说明为什么要创建这个patch
 - 最好能附上重现错误的测试
- 将patch提交到JIRA上
- 在邮件列表上说明这件事情，引起开发者的注意

Apache 之道

- 社区胜于代码 (Community over Code)
- 任人为贤 (Meritocracy)
- 共识决策 (Consensus)
- 透明公开 (Transparency)
- 非隶属关系 (Non-Affiliation)
- 负责监督 (Responsible Oversight)

开源软件的展望

- 社交化开源
- 云端开发环境
- 开源项目是最好的简历
- 开源硬件