

中国开源软件系列丛书 ➤

开源软件 成熟度评估 及选型指南



工业和信息化部软件与集成电路促进中心 编著

- ◆ **经验分享** 分享软件成熟度评估、开源软件选型、项目管理等方面的宝贵经验
- ◆ **专家观点** 视角独特，视野广阔，观点深刻



中国水利水电出版社
www.waterpub.com.cn

中国开源软件系列丛书

开源软件成熟度评估及选型指南

工业和信息化部软件与集成电路促进中心 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书内容主要来自近几年我们对开源软件评估与应用选型的研究成果,以及对优秀的开源软件的筛选整理。内容主要面向那些希望将开源软件部署在其应用环境中,或利用开源软件进行二次开发的中小企业或开源爱好者。本书对于那些利用开源软件的网络社区建设者也有一定的参考价值。

全书内容共分为四部分:第一部分主要讲解开源软件的相关概念,开源运动在国际和国内发展的历史,及开源软件应用普及中遇到的问题;第二部分主要讲解开源软件选型中成熟度评估模型在国际、国内发展的情况,并依据近几年我们在相关领域的研究、探索,结合国内外经验,提出一个成熟度评估模型;第三部分着重讲解在开源软件选型中非常重要的环节——开源软件许可,通过问答的方式向大家讲解开源许可相关的知识产权问题对开源软件选型的影响,并对开源许可中最重要的 GPL 协议进行了分析;第四部分向大家推荐一系列互联网开发、应用相关的开源软件,也作为我们对开源软件选型方法的实践。此外,在附录中给出了一个软件评估规范的参考范本和一些开源软件相关知识点的详细介绍。

本书的一些内容来自相关项目或软件的官方信息;同时,本书的内容也获得了开源中国社区和中日韩东北亚开源合作项目的大力协助,在此对他们深表感谢。

图书在版编目(CIP)数据

开源软件成熟度评估及选型指南 / 工业和信息化部
软件与集成电路促进中心编著. — 北京: 中国水利水电
出版社, 2011.8

(中国开源软件系列丛书)

ISBN 978-7-5084-8893-6

I. ①开… II. ①工… III. ①软件开发—评估—指南
IV. ①TP311.52-62

中国版本图书馆CIP数据核字(2011)第167470号

策划编辑: 周春元

责任编辑: 杨元泓

加工编辑: 陈 洁

书 名	中国开源软件系列丛书 开源软件成熟度评估及选型指南
作 者	工业和信息化部软件与集成电路促进中心 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 18.25印张 491千字
版 次	2011年9月第1版 2011年9月第1次印刷
印 数	0001—2000册
定 价	45.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

中国开源软件系列丛书

编审委员会

特别顾问：

陆首群 中国开源软件推进联盟名誉主席
倪光南 中国工程院院士

主任委员：

邱善勤 工业和信息化部软件与集成电路促进中心主任

副主任委员：

宋茂强 北京邮电大学软件学院院长
韩乃平 中标软件有限公司总经理
刘 澎 中国开源软件推进联盟秘书长
高松涛 工业和信息化部软件与集成电路促进中心副主任

委员：

刘龙庚 工业和信息化部软件与集成电路促进中心 主任助理
陈 钟 北京大学软件与微电子学院 教授
吴中海 北京大学软件与微电子学院副院长
高 林 工业和信息化部电子标准化研究所信息技术研究中心主任
宫 敏 Linux 内核专家
刘 明 工业和信息化部软件与集成电路促进中心软件处副处长
金友兵 北京红旗中文贰仟软件技术有限公司副总经理
郑忠源 北京中科红旗软件技术有限公司副总裁
陈 渝 清华大学计算机科学与技术系 副教授
杨春燕 共创软件联盟常务副秘书长
王威信 工业和信息化部软件与集成电路促进中心战略研究部主任
杜 娇 工业和信息化部软件与集成电路促进中心知识产权部主任
唐仕武 工业和信息化部软件与集成电路促进中心软件处副处长
张 靖 北京赛普信科技有限公司教育培训部副主任
冯晓焰 英特尔亚太研发有限公司首席开源科学家
胡昌军 中国开源软件中心技术总监
陶 品 清华大学计算机科学与技术系 副教授
张 俊 北京共创开源软件有限公司产品总监
张永军 中标软件有限公司产品经理
陈 绪 英特尔中国有限公司总监
杨 松 北京红旗中文贰仟软件技术有限公司产品部总监
李 恒 北京红旗中文贰仟软件技术有限公司技术总监

II

编委会

本书主编：刘龙庚

本书编委：

陈 越	胡昌军	吴 桐	唐仕武
杨玲玲	陶莹莹	王少华	张 阳
刘 帅	刘 风		

计算机软件的发展就像社会发展的某种制度轮回。它出身草莽，从最初小众间自由研究、修改和分享，到比尔·盖茨举起商业软件的大旗，软件产业开始飞跃发展，不仅造就了微软这样一个庞大的软件帝国，也成就了近二十年来软件产业的经济奇迹。然而在软件产业商业堡垒的内部，一些人重新开始怀念那个呼吸着自由空气的计算机时代，为什么我们要将自己的创造力和奉献锁在一个暗不见天日的专有制度下；为什么我们不能自由地交流和分享；为什么我们付了费却仅仅得到软件的使用权。其中一些人最早觉醒起来，Richard Stallman 作为之后自由软件的领袖开始了某种抗争进而发起所谓“浪漫启蒙”的尝试，开源软件运动与开源产业相继诞生，自由与分享又重新回到人们身边。但是这种开源的回归绝不仅仅是重复几十年前的故事，从开源理论到组织形式再到商业哲学，开源运动都发展到一个崭新的阶段，有效地推动了软件产业的发展。可以说，开源软件正好符合了近几年来软件产业转型的需要，促进了以互联网应用为中心的、自下而上的整个软件产业链的发展，推动计算机应用深入到人们生活的每个角落。

作为一种新的软件产业发展机制，除了软件自身开发和商业模式的转变外，与开源软件发展、应用相关的方方面面都值得人们关注，需要我们不断地探索、完善。本书正是关注开源软件成熟度评估、选型方面的理论与实践，并向大家推荐与互联网应用相关的优秀开源软件。

本书内容主要来自近几年我们对开源软件评估与应用选型的研究成果，以及对优秀的开源软件的筛选整理。内容主要面向那些希望将开源软件部署在其应用环境中，或利用开源软件进行二次开发的中小企业或开源爱好者。本书对于那些利用开源软件的网络社区建设者也有一定的参考价值。

全书内容共分为四部分：第一部分主要讲解开源软件的相关概念，开源运动在国际和国内发展的历史，及开源软件应用普及中遇到的问题；第二部分主要讲解开源软件选型中成熟度评估模型在国际、国内发展的情况，并依据近几年我们在相关领域的研究、探索，结合国内外经验，提出了一个成熟度评估模型；第三部分着重讲解在开源软件选型中非常重要的环节——开源软件许可，通过问答的方式向大家讲解开源许可相关的知识产权问题对开源软件选型的影响，并对开源许可中最重要的 GPL 协议进行了分析；第四部分向大家推荐了一系列互联网开发、应用相关的开源软件，也作为我们对开源软件选型方法的实践。此外，在附录中给出了一个软件评估规范的参考范本和一些开源软件相关知识的详细介绍。

本书在各章节的编写过程中，相关参与人员都付出了宝贵的时间和精力。他们包括：开源介绍章节的作者胡昌军和陈越，成熟度评估章节的作者陈越和陶莹莹，开源软件许可基本知识章节的作者胡昌军，Web 服务和数据库章节的作者王少华，PHP 框架和 AJAX 章节的作者杨玲玲，项目管理章节的作者吴桐，以及 Web 应用章节的作者张阳。

本书的一些内容来自相关项目或软件的官方信息；同时，本书的内容也获得了开源中国社区 (<http://oss.org.cn>) 和中日韩东北亚开源合作项目的大力协助，在此对他们深表感谢。由于时间有限，本书错误在所难免，敬请广大读者批评指正。

中国开源软件中心

2011 年 6 月

IV

目录

前言

第 1 章 开源软件现象综述	1
1.1 开源软件相关定义和概念	1
1.1.1 什么是开源软件?	1
1.1.2 什么是自由软件 (FS)?	1
1.2 开放源代码运动	5
1.2.1 开放源码运动与 UNIX	5
1.2.2 开放源码运动与 BSD	6
1.2.3 开放源码运动与 FSF、GNU	6
1.2.4 开放源码运动与 Linux 的诞生	6
1.2.5 开放源代码运动与 OSI	7
1.2.6 开放源码运动与 LF	8
1.2.7 开放源码运动在中国	8
1.3 开源软件对软件生态系统的影响	10
1.3.1 开源软件已成为主流	10
1.3.2 开源软件是一种诱人的商机	10
1.3.3 开放源码促进竞争	11
1.4 开源软件给我们带来的好处和风险	11
1.4.1 我们为什么需要开源软件	11
1.4.2 使用开源软件的好处	12
1.4.3 使用开源软件与专有软件的风险无本质区别	14
1.4.4 使用开源软件的一大挑战	15
第 2 章 开放源代码软件成熟度评估的理论与方法	17
2.1 成熟度评估助力开源软件发展	17
2.2 开源软件成熟度评估的目标	18
2.3 开放源代码软件成熟度评估发展状况	18
2.3.1 国际上主要开源软件质量/成熟度评估模型介绍	19
2.3.2 国内开源软件质量/成熟度研究的情况	28
2.4 开放源代码软件成熟度评估体系模型	28
2.4.1 开源软件成熟度评估模型是什么	28
2.4.2 开放源代码软件成熟度评估环境	29
2.4.3 开源软件成熟度评估规范的定义	30
2.4.4 开放源代码软件成熟度评估中的属性	30
2.4.5 开源软件成熟度属性权重定义	32
2.4.6 开放源代码软件成熟度评分与计算	32
2.5 开源软件成熟度属性定义参考	33
2.6 开源软件成熟度属性评估方法	35
2.6.1 开源软件成熟度评估取值与评分约定	35
2.6.2 软件成熟度评估的评测取值、计算、分析	37
2.7 软件成熟度评估报告内容	37
第 3 章 开源软件许可的基本概念及问题解答	39
3.1 开源软件许可的基本知识	39
3.1.1 为什么需要了解软件的知识产权?	39
3.1.2 软件相关知识产权主要有哪几类?	39
3.1.3 没有登记的软件受著作权法保护吗?	40
3.1.4 在中国有没有软件专利, 商业方法软件专利又是什么?	40
3.1.5 软件产品为什么需要许可协议?	40
3.1.6 发布开源软件需要许可协议吗?	40
3.1.7 使用开源软件需要获得许可吗?	41
3.1.8 为什么存在诸多开源软件许可证?	41

3.1.9 开发者是否可以自己创建一个许可证?	41	3.3.2 原始许可“不担保”, 开源软件就一直得不到担保吗?	47
3.1.10 一个软件是否可以适用多个许可?	41	3.3.3 为什么开源软件的商业应用归根结底还是需要有人提供担保?	47
3.1.11 开源软件许可证带来的纠纷可能有哪些?	42	3.3.4 原始许可“不担保”, 用户怎么办?	47
3.1.12 开源软件许可证有什么共同特点?	42	3.3.5 再分发人对不享有知识产权的开源软件收费是否合法?	47
3.1.13 不同开源软件许可证之间差异在哪里?	42	3.3.6 为什么开源软件本身可以收费, 但通过单纯对软件进行收费实现商业目的的模式注定不成功?	47
3.1.14 主要许可证的之间有哪些关键差别?	42	3.3.7 许可人使用特定许可证发布的软件还能不能撤销许可?	48
3.1.15 许可证之间的规定存在冲突怎么办?	43	3.3.8 已经免费接受许可的一方会不会在日后面临被收费的危险?	48
3.1.16 OSI 与开源许可证是什么关系?	43	3.3.9 开源软件的商标侵权风险存在于哪些方面?	48
3.1.17 许可证经 OSI 认证有何益处?	43	3.3.10 我国企业如何应对开源软件相关的商标侵权风险?	48
3.2 详解 GPL 许可	43	3.3.11 利用开放源码启动专有软件企业应重点关注哪些事项?	49
3.2.1 什么是 Copyright?	43	3.3.12 为什么国外用户大量使用开源软件但看似是并不关心知识产权风险?	49
3.2.2 什么是 Copyleft?	44	第 4 章 Web 服务	50
3.2.3 Copyright 与 Copyleft 是什么关系?	44	4.1 Apache	50
3.2.4 什么是 GPL?	44	4.1.1 软件简介	50
3.2.5 GPL 的核心要求有哪些?	44	4.1.2 功能特性	51
3.2.6 GPL 就是“免费获得”和“免费贡献”吗?	44	4.1.3 成熟度评估	52
3.2.7 GPL 的传染性是什么意思?	45	4.1.4 主要参考资料目录	54
3.2.8 “独立作品”可以对“传染性”免疫吗, 怎样判断作品是不是“独立作品”?	45	4.2 Lighttpd	55
3.2.9 GPL 对作品的哪些知识产权做了规定?	45	4.2.1 软件简介	55
3.2.10 怎样的行为算做接受了 GPL (或缔结了 GPL 许可协议)?	46	4.2.2 功能特性	55
3.2.11 使用遵循 GPL 的软件的被许可人必须遵守哪些义务?	46	4.2.3 成熟度评估	57
3.2.12 GPL 后的软件是不是进了“公有领域”?	46	4.2.4 主要参考资料目录	59
3.3 利用开源软件实现商业应用需注意的问题	46	4.3 Nginx	60
3.3.1 为什么开源软件强调“No warranty”?	46	4.3.1 软件简介	60
		4.3.2 功能特性	60
		4.3.3 成熟度评估	62
		4.3.4 主要参考资料目录	65

第5章 数据库	66	7.1.1 软件简介	105
5.1 MySQL	66	7.1.2 功能特性	106
5.1.1 软件简介	66	7.1.3 成熟度评估	107
5.1.2 功能特性	67	7.1.4 主要参考资料目录	110
5.1.3 成熟度评估	68	7.2 Ext JS	110
5.1.4 主要参考资料目录	71	7.2.1 软件简介	110
5.2 PostgreSQL	71	7.2.2 功能特性	111
5.2.1 软件简介	71	7.2.3 成熟度评估	112
5.2.2 功能特性	72	7.2.4 主要参考资料目录	115
5.2.3 成熟度评估	75	7.3 Dojo	115
5.2.4 主要参考资料目录	78	7.3.1 软件简介	115
5.3 Firebird	78	7.3.2 功能特性	116
5.3.1 软件简介	78	7.3.3 成熟度评估	118
5.3.2 功能特性	79	7.3.4 主要参考资料目录	120
5.3.3 成熟度评估	80	7.4 MooTools	121
5.3.4 主要参考资料目录	83	7.4.1 软件简介	121
第6章 PHP 框架	84	7.4.2 功能特性	121
6.1 QeePHP	84	7.4.3 成熟度评估	123
6.1.1 软件简介	84	7.4.4 主要参考资料目录	125
6.1.2 功能特性	85	第8章 项目管理	126
6.1.3 成熟度评估	86	8.1 项目管理	126
6.1.4 主要参考资料目录	89	8.1.1 Redmine	126
6.2 ThinkPHP	89	8.1.2 禅道项目管理软件	132
6.2.1 软件简介	89	8.2 测试管理	137
6.2.2 功能特性	90	8.2.1 Bugzilla	138
6.2.3 成熟度评估	92	8.2.2 Trac	144
6.2.4 主要参考资料目录	95	8.3 版本管理工具	150
6.3 Zend Framework	95	8.3.1 Git	150
6.3.1 软件简介	95	8.3.2 Subversion	155
6.3.2 功能特性	96	第9章 Web 应用	161
6.3.3 成熟度评估	97	9.1 CMS 内容管理系统	161
6.3.4 主要参考资料目录	100	9.1.1 Drupal	161
6.4 CodeIgniter	100	9.1.2 Joomla	168
6.4.1 软件简介	100	9.1.3 XOOPS	174
6.4.2 功能特性	101	9.1.4 DotNetNuke	181
6.4.3 成熟度评估	102	9.2 BBS 论坛	186
6.4.4 主要参考资料目录	104	9.2.1 Discuz!	187
第7章 AJAX 框架	105	9.2.2 PHPWind	194
7.1 jQuery	105	9.3 Blog 博客	202

9.3.1	WordPress	202	9.5.3	eGroupware	233
9.3.2	StatusNet	207	9.6	电子商务	239
9.3.3	Typecho	212	9.6.1	Zen Cart	239
9.4	Wiki 维基	216	9.6.2	osCommerce	246
9.5	在线 OA/协同	222	附录 A	开源概念定义	252
9.5.1	Zimbra	222	附录 B	开源软件成熟度属性定义参考	261
9.5.2	eyeOS	228	参考文献		281

开源软件现象综述

1.1 开源软件相关定义和概念

对于当今世界的 IT 产业，开放源代码已渐成主流，它的影响遍及行业的每个角落。这不仅体现在软件研发模式和商业模式的改变上，甚至也深入我们的社会生活，改变着人们的一些旧有观念。开放源代码软件（简称：开源软件）在许多情况下已经被认为是封闭源码软件的一种可行替代方案。它的出现促进了软件新技术的应用，加强了市场竞争，扩大了用户对软件的选择范围，避免出现用户被卖方“套牢”的尴尬情况。

1.1.1 什么是开源软件？

我们给出开放源代码促进会（Open Source Initiative, OSI）标准定义的概述：

开源软件即指源代码开放的软件产品。但开源软件的内涵又不只限于源代码的可访问；它还需遵循以下几项标准：开放源码、软件产品可自由发行、允许衍生、保护原始代码完整性、对用户和使用领域的无差别对待，以及技术中立等原则。

开源软件诞生于两种历史渊源（学术界的研究产物和黑客现象的产物），随软件信息产业的发展逐渐演变而来。而随着开放源码运动的不断发展，企业和主流软件社区也参与其中，并且发挥着越来越重要的作用。目前，相当多的开源软件都希望作为传统专有软件的可行替代手段和一种有吸引力的商业模式而获得业界承认。在过去几年中，开源软件市场逐步扩大，引起了更多 IT 工程师强烈的兴趣，并吸收了从私营实体到公共部门的客户群；同时开源软件的鼓吹者和传统的专有软件的支持者之间就到底哪种模式更能促进产业的发展，最终更有利于用户的应用等问题上的辩论从未停止。

接下来，我们通过对开源语境相关的一些常见概念进行介绍，从而有助于读者更好地理解开源软件的内涵。

1.1.2 什么是自由软件（FS）？

自由软件是指允许任何人使用、拷贝、修改、分发（免费/少许收费）的软件，尤其是这种软件的源代码必须是可得到的。从某种意义上说，“没有源代码，就称不上是（自由）软件。”自由软件很少标价出售，可以理解为一定程度的免费。但有时软件公司在提及价格时使用“Free Software”这个字眼，他们的意思可能是你可以免费获得一份二进制代码的拷贝，也可能是在你所购买的机器中已经包含了一份拷贝。但是，这种说法与在 GNU 计划中“Free Software”的意义是不同的。

由于在操作中极易混淆,因此一旦软件公司说他们的软件是自由软件,一定要检查您是否得到了自由软件所应包括的一切。有时,您得到的是自由软件,有时不是。

1. 开源软件和自由软件

“开源软件”和“自由软件”的完整定义可在各自的网站上找到:开放源码促进会(OSI)和自由软件基金会(FSF)。

许多人将开放源代码软件(Open Source Software)与自由软件(Free Software)视为相同,但若以定义条件而言,自由软件仅是开放源代码的一种,也就是自由软件的定义较开放源代码更为严格,并非开放源代码的软件就可称为自由软件,要视该软件的授权条件是否合乎自由软件基金会对于自由软件所下的定义。

严格地说来,开放源代码软件与自由软件是两个不同的概念,只要符合开源软件定义的软件就能被称为开源软件。自由软件是一个比开源软件更严格的观念,因此所有自由软件都是开放源代码的,但不是所有的开源软件都能被称为“自由”。但在现实中,绝大多数开源软件也都符合自由软件的定义。比如,遵守GPL和BSD许可的软件都是开放的并且是自由的。

自由软件又称为“自由和开源软件”(FOSS),有时又称为“自由/自主开源软件”(FLOSS)。虽然开源软件与自由软件在伦理和观念上有差别(对于自由软件基金会来说,“Free”一词是指自由而不是价格),在这两种情况下,开发人员对他们的作品均享有版权,他们所使用的许可条件可能会有不同,但并不排除软件的商业化。在许多评论中这两种表述方式交替使用,因为它们具有若干重要的共同属性,即

(1) 以群体为基础的合作开发模式,允许同行通过检索源代码作出审评。

(2) 任何人、在任何地方都有酌情使用、研究和修改软件的永久和不可撤消的权利。

(3) 许可证允许使用者有权在某些条件下(通常与任何形式的金钱补偿无关),重新发布软件或派生作品。在这方面,自由软件要求其他人重新发布软件时赋予接受者以最初许可证的规定完全相同的权利,任何违反构成对版权的侵犯。自由软件基金会使用“复制权”更好地表达“自由软件”所附带的“版权”有别于传统的版权。

大多数开源软件解决方案没有许可证费用,可自由从互联网上下载。这并不意味着其使用必然不付出成本费用。这类费用涉及培训、支持或如有需要对程序的优化编译等。开源软件的许可证证明已超过 70 个,且数量在继续增长。这些许可协议分为两大类,其中包括:

(1) GNU 普通公共许可证(GPL)和 GPL 兼容许可证,后者提供开源软件所附带的主要自由,但额外要求软件和派生作品局限于 GPL 或 GPL 的兼容许可证范围内。

(2) 赋予不同权利的其他许可证,根据这类许可证,开放源码可与专利代码相结合,或者成为开源软件,或者成为专有软件。

2. 封闭源码软件

封闭源码软件(CSS)或专有软件是创作者持有独享权利(版权和(或)专利)的软件,并将源码视为一种商业机密。其使用、发布或修改的规定受严格条件的限制,主要目的在于保护版权所有者的知识产权,它可以选择免费提供软件,或准许有限检索源码。封闭源码软件还被称为商业软件或商业成品软件(COTS),但这两种表述方法可能令人产生误解,因为市场上目前也可以获得开源软件的商业发布(他们也是商业软件)。尽管全球封闭源码软件市场五花八门,微软公司占有操作系统和办公产品端的 90%以上,而由此也引发了若干项反垄断诉讼。

3. 商业软件

“商业软件”(Business Software)是指通过交易或贸易方式面向社会公众发行的各种商品化的

软件。作为商业软件，其功能、性能通常已经过严格测试因而相当可靠，其使用方法应该相当便利。供应商不仅应该向用户提供程序（通常是目标码文本）和使用说明（通常是手册的形式），而且应该向用户提供包括版本更新在内的技术服务。

4. 开源软件与商业软件是什么关系？

商业软件由商业公司开发，通过收取使用费而牟利。“商业”和“私有”不是一回事。大多数商业软件是“私有的”，但也有商业自由软件，也有非商业、非自由的软件。

例如，GNU Ada 永远在 GNU GPL 条款下进行分发，而且，每一个拷贝都是免费的，但它的开发者却有付费支持合同。当推销员对用户这样说，有时用户会说，“我们觉得商业编译器比较安全一些。”推销员则回答，“GNU Ada 是商业软件，只不过碰巧它也是一个自由软件。”

对“商业”的含义可能存在不同理解，但可以确定的虽然开源软件本身可能不会带来很大的收益，但是利用开源软件的商业模式的确可以实现商业利益，提供了源代码以外的服务等客户所需的要素就可以获得持续、稳定的商业利益。

5. 什么是私有软件？

“私有软件”（即 Proprietary Software，或称“专有软件”），是 FSF 将其作为“自由软件”和“非自由软件”的对立面而对软件作出的分类。FSF 对私有软件的解释为：

私有软件不是自由及非自由软件，对它的使用、传播、或修改是禁止的，要么需要你申请许可，要么它限制你不能充分自由地使用它。

自由软件基金遵守规定，不在计算机中安装任何私有软件，除非为了编写那个程序的替代品而暂时安装到计算机上。除此之外，我们找不到其他借口要安装私有程序。

例如，在 80 年代，我们认为在计算机安装 UNIX 是正当的，因为我们要编写一个 UNIX 的免费替代品。现在，由于有了自由操作系统，因此这个借口就不再使用了；我们消除了所有的非自由操作系统，任何新计算机上安装的都是自由操作系统。我们不能强迫 GNU 的用户或 GNU 的贡献者必须依此规则行事；它只是给我们自己制定的规则，但我们希望你能下决心遵从它。

6. 什么是公有软件？

所谓“公有软件”（Public Domain Software，也可译成“公有领域软件”）是指著作权中的经济权利（包括复制权、修改改编权、发行权等专有权利）有效期已经届满的软件，以及权利人由于不准备使之商品化而已经明确声明放弃著作权的软件。需要指出，公有软件不同于开放源代码软件，因为对于开放源代码软件而言，其著作权仍然由其权利人享有，受到著作权法规保护。

对于公有软件，其特点在于：①人们可以任意复制、传播；②人们也可以对公有软件进行反汇编或者反编译，可以进行反向工程；③人们可以使用、改编、甚至将之纳入为赢利目的而开发的商业软件，但不得限制其他人也这样做。这类软件的知识产权被用尽的或放弃任何知识产权的软件，因其进入了公有领域（Public Domain），没有任何人可以对其请求任何的权利。

7. 什么是共享软件？

试用软件（Share Ware，国内也往往译成“共享软件”）。它比普通的商业软件要便宜得多。所谓“试用软件”，是指其复制品也可以通过网络在线服务、电子公告板（BBS）或者从一个用户的手里传给另一个用户等途径自由地传播。这种软件的使用说明通常也以文本文件的形式同程序一起提供。这种试用软件通常附有一个用户注意事项，其内容是说明权利人保留对该软件的权利，因此试用软件受著作权保护；本软件属于试用软件，用户通过两个阶段试用之后，如果希望继续使用，就应该向供应者办理使用注册手续。用户注意事项中通常包含一份注册表格，并说明如何以及向何处办理注册手续，在办理注册手续时可能会要求用户交纳一定费用，实际上就是使用许可费，需要

交付的使用许可费的款额通常是很低的（几美元或者几十美元）。

8. 开源软件与免费软件的关系如何？

对免费软件通常的理解是不必支付费用就可以免费获得使用权的软件。而恰巧开源软件在多数场合被称作“Free Software”。在许多语言中，“Free”一词有两种意义：一是自由，二是免费。如法语的“libre”和“gratuit”，英语的“gratis”都明白无误地指价格（免费），但没有明白无误地说明“自由”的意义。这是不幸的，因为如果有这个词，在此处它将非常有用。

开源软件通常以免费方式发布，大部分开源软件可以免费获得。但是只要开源软件的许可人或再许可人愿意，可以收取费用。开源是不排斥收费行为的，而免费软件一定是免费获得的。同时免费软件可能是专有软件，只授权免费使用，但不开放源代码。因此，开源软件与免费软件是两个不同层面的概念，需要区别对待。

9. 如何理解“Free Software”的“Free”？

这里的 Free 有两个含义：

第一层含义是自由。自由软件模式为开发者提供了将个人智力成果以不同传统的方式贡献给公众的自由，并保证这种自由符合法律的要求；同时自由软件没有妨碍并且为公众提供了额外的、选择的自由。开发者自愿地将其对于软件的自由权利授权给其他人，保证了更多人享受到使用软件的自由。

第二层含义是一定条件下的免费。由于自由软件的授权模式不强制免费许可也不强制收费许可，因此，对于已经进入社区（流通领域）的开源软件而言，获得这些软件大都不必付费。

了解哲学观念中“没有绝对的自由”是理解这个概念的一把钥匙，开发者和用户的“自由”都是建立在符合法律和协议要求的前提下的自由；“天下没有不要钱的午餐”这句话的道理不在于这个午餐是不是一定要付钱，而是说是不是要有代价，“免费”获得开源软件也是一样道理。

总之，“自由”和“免费”都是附有条件的，这个条件就是协议（合同）规定的“对价”，自愿地履行开源协议的“对价”不存在对合同任何一方造成损害，不过可以确定的是，如果不履行“对价”，合同就可能不成立，这显然对合同双方都没有好处，不能实现订立合同的宗旨。

10. 什么是“开放源代码”？

OSI 给出开放源代码的定义的目的是：把所确信的、由软件开发团体所公认的“开放源代码”的含义作为一组具体的准则写下来。该准则确保按照开放源代码许可证发布的软件可以得到与其他软件同样认真的评审、使软件可以不断地得到改良和遴选，从而提供非开放软件所难以提供的可靠性与能力。

OSI 认为本开放源代码定义涵盖了由绝大多数软件团体使用的术语“开放源代码”的最初含义和当前含义。然而，该术语被广泛地应用，并且它的含义变得不精确了。

11. “开放”源代码就是“公开”源代码吗？

以微软为代表的“共享源代码”（Shared Source）的实质是仅仅公开源代码（且非全部公开）。这是微软为应对“开放源代码运动”而提出的新策略，它与 Open Source 最大的不同在于，微软向他的的大客户拷贝其绝对机密的 Windows 源代码，但用户只能看或提出修改意见，而不能对它有任何修改。

开放源代码与公开源代码的相同之处是两者都保证源代码被公开，即源代码以某种方式为人所知悉。但“开源”与“公开”源代码存在根本的差别。开源软件的许可人不仅仅提供源代码以供其他人参考，同时还许可其他人可以将软件运行、修改和再授权。

更直白的解释就是“公开”源代码只能“看”，“不能用、不能改、更不能再发布”；而“开放”

源代码不仅能看，还“能用、能改、能再发布”。

12. 开源软件也关注知识产权

开源软件虽然拥有开放的软件开发和软件发布模式，但它对待软件知识产权的保护仍然有完整的诉求。软件通常受版权保护，它赋予其创作者在规定时间内复制或发布其作品的专有权利。第三方须得到创作者的许可，才能合法地复制和重新发布这一作品。软件版权在国际一级受到承认并具有约束力，对此世界贸易组织（世贸组织）通过的《与贸易有关的知识产权协定》第 10 条第 1 款规定，“计算机程序，无论是源代码还是目标代码，应作为《伯尔尼公约》（1971）内的文字作品加以保护”。

在一些国家，软件还有可能受专利保护，由政府颁发的专利给予发明人对其发明的使用、复制和发布以临时垄断权，以换取其出版。专利的使用者须征得发明者的许可才能合法使用、复制、发布或重新发布由上述发明派生出的某一产品。对于软件的专利性引发了争议，而对《世贸知识产权协定》第 27 条第 1 款大相径庭的诠释更激化了这种争议，该款规定，“……专利可授予所有技术领域的任何发明，无论是产品还是方法，只要它们具有新颖性、包含发明性步骤并可供工业应用”。

软件许可证是创作者和其产品接受者之间的一项合同，它对接受者针对软件可做什么或不能做什么订出限制。大多数软件配有许可证，公共领域的软件除外。如果软件没有提供许可证（而软件也不属于公共领域的软件），那么只允许接受者享有版权法和（或）专利法为创作者规定的权利以外的权利。

1.2 开放源代码运动

开源软件运动是计算机科学领域的一种文化现象，源自黑客对智慧成果共享、自由的追求。它是一个主要由程序工程师及其他电脑用户共同参与的声势浩大的社区运动，是自由软件运动的一个分支，但两者的差别并不明显；一般而言，自由软件运动是基于政治及哲学思想（有时被称为所谓黑客文化）的理想主义运动，而开放源代码运动则主要注重程序本身的质量提升。开源运动发展到现在，这种能够积极促进人类文明发展的文化已经渗透到信息、教育、健康等领域，融入了哲学范畴。

1.2.1 开放源码运动与 UNIX

在 1969~1970 年，肯尼思·汤普森（Kenneth Thompson）、丹尼·斯里奇（Dennis Ritchie）等人在 AT&T 贝尔实验室开始开发一个很少使用的 PDP-7 上一个小的操作系统。这个小的操作系统很快就被命名为 UNIX，这个名字是对 1969 年贝尔实验室推出的一个较大项目 MULTICS（Multiplexed Information and Computing Service）的戏称。也就是这一年，Linux 之父 Linus Torvalds 在芬兰出生了。

在 1972~1973 年，肯尼思·汤普森（Kenneth Thompson）、丹尼·斯里奇（Dennis Ritchie）成功地用 C 语言重写了 UNIX 的第三版内核，至此，UNIX 这个操作系统修改、移植相当便利，为 UNIX 日后的普及打下了坚实的基础。而 UNIX 和 C 完美地结合成为一个统一体，C 与 UNIX 很快成为世界的主导。

1979 年发布的 UNIX 第七版被称为是“最后一个真正的 UNIX”，这个版本的 UNIX 内核只有 40KB。它是所有现存的 UNIX 系统的祖父。

1982 年，AT&T 基于版本 7 开发了 UNIX System III 的第一个版本，这是一个商业版本仅供出售。而加州大学 Berkeley 分校继续开发 BSD UNIX，作为 UNIX System III 和 V 的替代选择。

Berkeley Software Design, Inc (BSDI) 很快就与 AT&T 的 UNIX Systems Laboratories (USL) 附属公司产生了法律纠纷, USL 是 AT&T 注册的公司。AT&T 拥有 System V 版权, 以及 UNIX 商标。

由于 UNIX 是由 C 语言写的, 所以修改和移植都很容易, 因此, 很多商业公司及学术机构均加入这个操作系统的研发, 各个不同版本的 UNIX 也开始蓬勃发展。这才产生了今天这么多的各式各样的 UNIX 衍生产品, 如 AIX、Solaris、HP-UX、IRIX、OSF、Ultron 等 (这些商业化的 UNIX 基本上都是源于 AT&T 授权的 UNIX System V)。

1.2.2 开放源码运动与 BSD

1978 年, 对 UNIX 而言是革命性的一年; 因为学术界的老大柏克利大学 (UC Berkeley), 推出了一份以第六版为基础, 加上一些改进和新功能而成的 UNIX。这就是著名的 “1 BSD (1st Berkeley Software Distribution)”, 开创了 UNIX 的另一个分支: BSD 系列。同时期, AT&T 成立 USG (UNIX Support Group), 将 UNIX 变成商业化的产品。从此, BSD 的 UNIX 便和 AT&T 的 UNIX 分庭抗礼, 各自蓬勃发展。

1991 年, 一群 BSD 开发者 (Donn Seeley、Mike Karels、Bill Jolitz 和 Trent Hein) 离开了加州大学, 创办了 Berkeley Software Design, Inc (BSDI)。BSDI 是第一家在便宜常见的 Intel 平台上提供全功能商业 BSD UNIX 的厂商。

1992 年, 为了垄断 UNIX, USL 正式对 BSDI 提起诉讼, 说 BSD 剽窃他的源码, 而最终了结了好评如潮的 BSD 系统。这桩诉讼最终在 1994 年 1 月了结, 更多地满足了 BSDI 的利益。伯克利套件的 18,000 个文件中, 只有 3 个文件要求删除, 另有 70 个文件要求修改, 并显示 USL 的版权说明。这项调解另外要求, USL 不得对 4.4BSD 提起诉讼, 不管是用户还是 BSDI 代码的分发者。于是, BSD UNIX 走上了复兴的道路。BSD 的开发也走向了几个不同的方向, 并最终导致了 FreeBSD、OpenBSD 和 NetBSD 的出现。

而 FreeBSD、OpenBSD 和 NetBSD 最终以开源而结束。FreeBSD (被称为当时最容易安装到 PC 类型的计算机上的操作系统), NetBSD (被称为支持最多不同 CPU 架构的操作系统), OpenBSD (被称为最安全的操作系统)。

1.2.3 开放源码运动与 FSF、GNU

自由软件是一个旧概念, 当计算机第一次来到大学里时, 它们是研究工具。软件源代码被自由的传来传去, 为程序员支付劳务费, 而不是程序。只有后来, 当计算机进入商业世界, 程序员和商人为了生存, 通过对他们拥有的软件的每个拷贝收取费用。

1984 年, Richard Stallman 成立了自由软件基金会 FSF 和创建了 GNU 项目, 来贯彻它的自由软件理念。Stallman 的前提是人们应该有更多的自由, 他设计了他认为所有用户应该具有的一系列的权利, 把他们写入 GPL 协议中, 并戏称 GPL 为 “Copyleft”。

Stallman 自己开发了一些自由软件, 如 GNU C 编译器、GNU Emacs 等, 他的作品启发了许多人在 GPL 下促进自由软件发展。尽管开源软件与自由软件的定义不同, 但它也包含了许多 Stallman 的理念, 也可以被看作是他工作的演进。

1.2.4 开放源码运动与 Linux 的诞生

1991 年 8 月, Linux Torvalds 宣布: 计划在 Minix 用户新闻组上创建一个免费的操作系统, 并

称只是一种爱好。不会像 GNU 一样是一个大而专业的项目, 1991 年 10 月他在网上以 GPL 协议发布了 Linux 的 0.01 版本。在 Minix 新闻组里, Torvalds 请一些其他程序员帮忙。在更多人的帮助下, 1991 年 12 月份, 他发布了 0.10 版。在接下来的几年里, Linux 的开发者达到数十万, 并努力使 Linux 与 GNU 的程序兼容。

他的 kernel 与 FSF 组织里其他组件(尤其是一些 BSD 组件和 MIT 的 X Window 软件)被整合成为一个可自由修改和非常有用的操作系统: Linux 或者 GNU/Linux 操作系统。

在 Linux 社区中, 不同的组织都已经整合了不同的有用的组件。每次整合都被称为一次“发行版”, 这样的开发发行版的组织被称为“发行者”或“发行版提供者”, 常见的发行版为 Red Hat、Mandrake、SuSE、Caldera、Debian、Ubuntu、Redflag、Neashine 等。这些发行版之间都有些不同, 但是他们都有共同的基础: Linux 内核和 GNU glibc 库。由于他们都是以“copyleft”类型的协议发布, 这些公共部分的变化可为所有发行版公用, Linux 发行版之间有一种共同的凝聚力, 不像 BSD 与 AT&T-Unix 派生系统之间那样。

1996 年 6 月, Linux 2.0 发布。当时, 它已经被看作是一个成熟的操作系统。1999 年 1 月发布 2.2 了版本, 以及 2001 年 1 月发布了 2.4 版本, 后来的版本也更加成熟和强大, 扩展到支持 4~8 核处理器, 支持更多硬件平台(Intel, Alpha, PPC, Sparc, MIPS, ARM, 680xx, S/390……)。

1.2.5 开放源代码运动与 OSI

开放源代码促进会(Open Source Initiative, OSI), 又译作开放原始码组织, 是一个旨在推动开源软件发展的非盈利组织。

1998 年 2 月, OSI 由 Bruce Perens 及埃里克·斯蒂芬·雷蒙等人创立, 当时网景公司为了与微软的 IE 浏览器竞争, 将他们的旗舰产品网景浏览器发布成自由软件。从 OSI 创立起, 雷蒙就担任主席, 一直到 2005 年 2 月; 之后 Russ Nelson 继任了一个月, 但因为一些争论, 他很快辞职; 随后 Michael Tiemann 担任过渡期的主席。

OSI 的史前史包括整个 UNIX 历史, 自由软件和黑客文化。OSI 作为一个教育, 宣传和管理组织而成立。与 OSI 成立同时相关联的事件就是 Eric Raymond 的论文《大教堂与集市》的出版。在这篇文章中, Eric Raymond 开创性的分析和描述了黑客社会的风俗习惯。他的以分布式同行评审的理念为中心的分析, 迅速在黑客文化内部和外部引起强烈的反响。Raymond 在 1997 年 9 月 O'Reilly Perl 大会上关于大教堂与集市的演讲, 帮助触发了 Netscape 公司在 1998 年 1 月 22 日宣布, 计划发布自由软件 Web 浏览器的源代码。当时, 这给黑客社区和商业界带来了不小的震惊。这预示着一个长期持有的规则将被打破。

“开放源码”一词来源于 1997 年春天在加州的 Palo Alto 召开的一个所谓“纯粹程序员”参与的战略研讨会。参加会议的有 Todd Anderson, 来自 Foresight 研究所 Chris Peterson, 来自 Linux 国际协会的 John Hall 和 Larry Augustin, 有硅谷 Linux 用户协会的 Sam Ockman, 以及 Eric Raymond。它们关系的是寻找一种方式, 来像以前躲避自由软件的人们来推广这种思想, 自由软件和自由软件基金会的反商业信条让很多人对自由软件敬而远之。在 Eric Raymond 的坚持下, 他们一致通过了用新的术语: OpenSource(开源软件)来描述他们所推进的软件。一些对自由软件以及 GNU/Linux 感兴趣的人, 打算用更符合市场口味的方式来介绍自由软件, 试图在商业中找到合适的位置, 减少意识形态上的沟壑。这导致了开放源代码(Open Source, 简称开源)这个称谓的产生, 同时也导致与理查德·斯托曼及其自由软件基金会的分道扬镳。