

大教堂与市集- 开放源代码软件

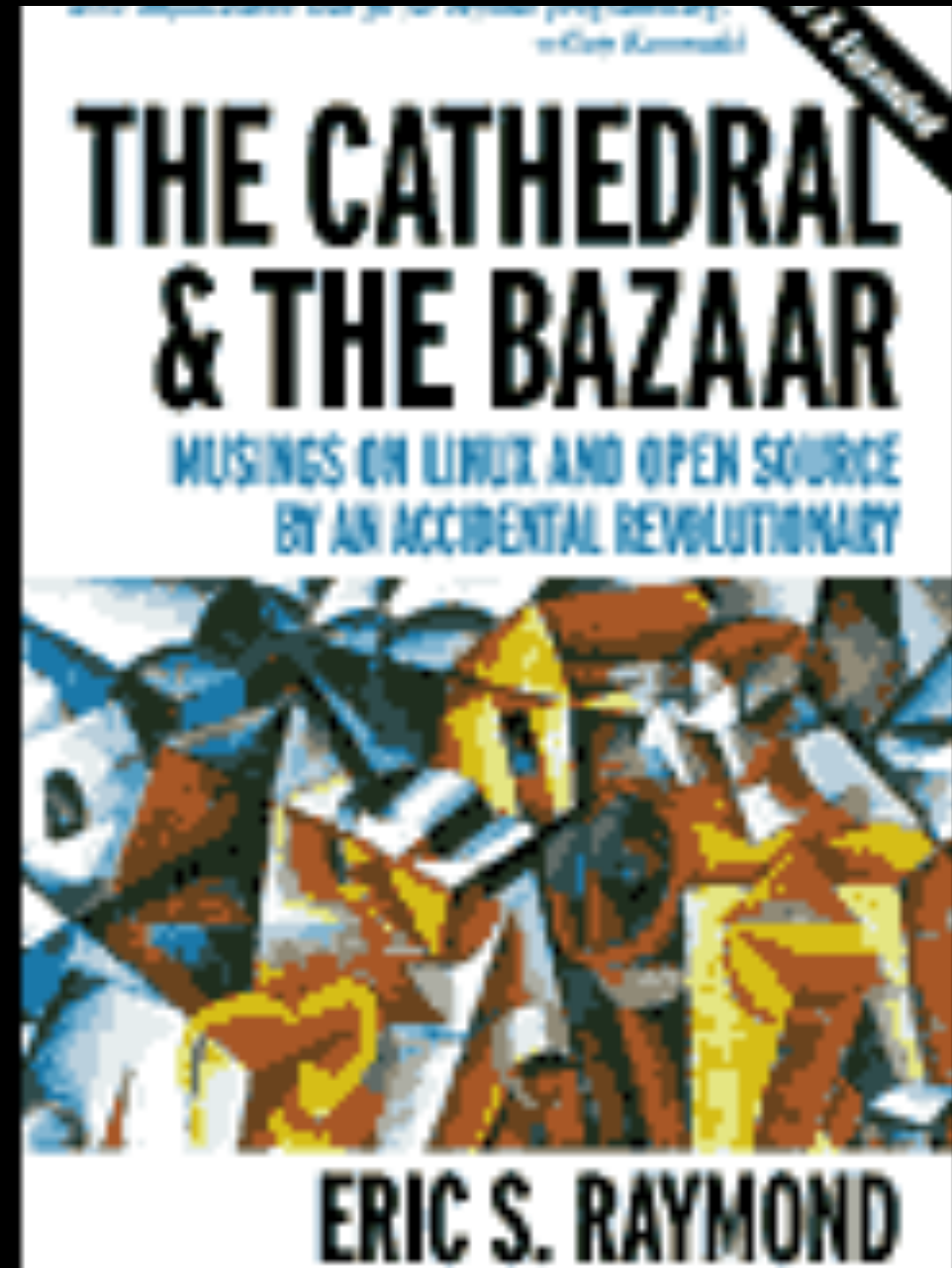
邵栋

dongshao@nju.edu.cn



南京大学 软件学院
NANJING UNIVERSITY SOFTWARE INSTITUTE

《大教堂与市集》（The Cathedral and the Bazaar）是埃里克·斯蒂芬·雷蒙（Eric Steven Raymond）所撰写的软件工程方法论。以Linux的核心开发过程以及作者自己主持开发的开放源代码软件——fetchmail为讨论案例。文章在1997年5月27日发表，并在1999年出版成书。



《大教堂和市集》—— 开放源代码运动的《圣经》

- 《大教堂和市集》第一次以大教堂模式和开放市集模式的比喻形象生动地将自由软件和商业封闭软件区分开来——“一种是封闭的、垂直的、集中式的开发模式，反映一种由权利关系所预先控制的极权制度；而另一种则是并行的、点对点的、动态的开发模式。”他在文中论证了自由软件不仅仅是一种乌托邦的理想，而是在开发模式上真正代表着“先进生产力”，代表着历史发展趋势的必然。
- 1998年2月3日，硅谷的一次会议上，“开放源代码（Open Source）”被正式提出，后来发展成为开源运动。雷蒙德成了领导这场运动的理论家和开放源代码促进会（Open Source Initiative）的主要创办人之一。

Linux

- Linux是一种自由和开放源代码的类UNIX操作系统内核。
- Linux内核最初只是由芬兰人Linus Torvalds在赫尔辛基大学上学时出于个人爱好而编写的。
- Linux的第一个版本在1991年9月被发布在互联网上，随后在10月份第二个版本就发布了。



开放源代码软件- 开源软件

- 开源软件
 - Linux
 - MySQL
 - Eclipse
 - Chrome, Firefox
 -
- 商业软件
 - Windows, MacOS
 - Oracle, DB2, SQL Server
 - Visual Studio
 - IE
 -

大教堂和市集

- Linux的影响是非常巨大的。甚至在5年以前，有谁能够想象一个世界级的操作系统能够仅仅依靠用细细的Internet连接起来的、散布在全球的几千个开发人员以业余时间来创造呢？
- Linux推翻了许多我认为自己明白的事情。
- 我以前相信多数重要的软件（操作系统和象Emacs一样的真正大型的工具）需要象建造大教堂一样来开发，需要一群与世隔绝的奇才的细心工作，在成功之前没有beta版的发布。

- Linus Torvalds的开发风格（尽早尽多的发布，委托所有可以委托的事，对所有的改动和融合开放）令人惊奇的降临了。
- 这里没有安静的、虔诚的大教堂的建造工作——相反，Linux团体看起来像一个巨大的有各种不同议程和方法的乱哄哄的集市（Linux归档站点接受任何人的建议和作品，并聪明的加以管理），一个一致而稳定的系统就象奇迹一般从这个集市中产生了。

Fecthmail

- 1.每个好的软件工作都开始于搔到了开发者本人的痒处。
- 也许这应该是显而易见的（“需要是发明之母”长久以来就被证明是正确的），但是软件开发人员常常把他们的精力放在它们既不需要也不喜欢的程序，但在Linux世界中却不是这样——这解释了为什么从Linux团体中产生的软件质量都如此之高。

- 2.好程序员知道该写什么，伟大的程序员知道该重写（和重用）什么。
- 伟大程序员的一个重要特点是建设性的懒惰。他们知道你是因为成绩而不是努力得到奖赏，而且从一个好的实际的解决方案开始总是要比从头干起容易。
- Linus并不是从头开始写Linux的。相反的它从重用Minix（一个386机型上的类似Unix的微型操作系统）的代码和思想入手。

- 3.“计划好抛弃，无论如何，你会的” (Fred Brooks, The Mythical Man-Month, Chapter 11)
- 你常常在第一次实现一个解决方案之后才能理解问题所在，第二次你也许才足够清楚怎样做好它，因此如果你想做好，准备好推翻重来至少一次。

- 4. 如果你有正确的态度， 有趣的问题会找上你的。
- 你在思考、 审视一些你感兴趣的软件时， 你会有新的更好的想法。

- 5. 当你对于一个程序失去兴趣时，你最后的责任就是把它传给一个能干的后继者。
- 在开源软件的开发中。

拥有用户的重要性

- 6. 把用户当做协作开发者是快速改进代码和高效调试的无可争辩的方式。

早发布、常发布

- 尽量早尽量频繁的发布是Linux开发模式的一个重要部分，多数开发人员(包括我)过去都相信这对大型工程来说是个不好的策略，因为早期版本都是些充满错误的版本，而你不想耗光用户的耐心。
- 这种信仰强化了建造大教堂开发方式的必要性，如果目标是让用户尽可能少的见到错误，那你怎能不会仅仅每六个月发布一次(或更不经常)，而且在发布之间象一只狗一样辛勤“捉虫”呢？

- 在Linux广泛应用之后，很清楚，一些不同的更加健康的东西诞生了，Linux的开发模式正好与建造教堂方式相反，推动了许多发布。所有这些都是在闻所未闻的频繁的内核系统的发布所推动的。

- 7. 早发布、常发布、听取客户的建议
- Linus的创新并不是这个(这在Unix世界中是一个长期传统)，而是把它扩展到和他所开发的东西的复杂程度相匹配的地步，在早期一天一次发布对他来说都不是罕见的!而且因为他培育了他的协作开发者基础，比其他任何人更努力地充分利用了Internet进行合作，所以这确实能行。

- 但是它是怎样进行的呢?它是我能模仿的吗?还是这依赖于Linus的独特天才?
- 我不这样想, 我承认Linus是一个极好的黑客(我们有多少人能够做出一个完整的高质量的操作系统内核?), 但是Linux并不是一个令人敬畏的概念上的飞跃, Linus不是(至少还不曾是)象Richard stallman (开发了Emacs) 或James Gosling (java之父) 一样的创新天才, 在我看来, Linus更象一个工程天才, 具有避免错误和开发失败的第六感觉, 掌握了发现从A点到B点代价最小的路径的诀窍, 确实, Linux的整个设计受益于这个特质, 并反映出Linus的本质保守和简化设计的方法。

- 如果快速的发布和充分利用Internet不是偶然而是Linus的对代价最小的路径的洞察力的工程天才的内在部分，那么他极大增强了什么？他创建了什么样的方法？
- 问题回答了它自己，Linus保持他的黑客用户经常受到激励和奖赏：被行动的自我满足的希望所激励，而奖赏则是经常(甚至每天)都看到工作在进步。

Linus定律

(Linus Law)

- 8. 如果有一个足够大的beta测试人员和协作开发人员的基础，几乎所有的问题都可以被快速的找出并被一些人纠正。
- 或者更不正式的讲：“如果有足够多的眼睛，所有的错误都是浅显的”(群众的眼睛是雪亮的)，我把这称为“Linus定律”。

- 我认为这是建造教堂和集市模式的核心区别，在建造教堂模式的编程模式看来，错误和编程问题是狡猾的、阴险的、隐藏很深的现象，花费几个月的仔细检查，也不能给你多大确保把它们都挑出来的信心，因此很长的发布周期，和在长期等待之后并没有得到完美的版本发布所引起的失望都是不可避免的。

- 以市集模式观点来看，在另一方面，我们认为错误是浅显的现象，或者至少当暴露给上千个热切的协作开发人员，让他们来对每个新发布进行测试的时候，它们很快变得浅显了，所以我们经常发布来获得更多的更正，作为一个有益的副作用，如果你偶尔做了一个笨拙的修改，也不会损失太多。

Delphi 效应

- 社会学家在几年前已经发现一群相同专业的(或相同无知的)观察者的平均观点比在其中随机挑选一个来得更加可靠，他们称此为“Delhpi效应”， Linus所显示的证明在调试一个操作系统时它也适用——Delphi效应甚至可以战胜操作系统内核一级的复杂度。
- Delphi估算法。

两种版本

- Linus也做了一些改进，如果有一些严重的错误，Linux内核的版本在编号上做了些处理，让用户可以自己选择是运行上一个“稳定”的版本，还是冒遇到错误的险而得到新特征，这个战略还没被大多数Linux黑客所仿效，但它应该被仿效，存在两个选择的事实让二者都很吸引人。

- 9. 聪明的数据结构和笨拙的代码要比相反的搭配工作的更好 （自己编程的认识）
- Fred Brooks讲道：“让我看你的 [代码] ，把你的[数据结构]隐藏起来，我还是会迷惑；让我看看你的[数据结构]，那我就不需要你的[代码]了，它是显而易见的”。
- 10. 如果你象对待最宝贵的资源一样对待你的beta测试员，他们就会成为你最宝贵的资源。

- (以下7条针对Fetchmail的开发)
- 11. 想出好主意是好事，从你的用户那里发现好主意也是好事，有时候后者更好。
- 12. 最重要和最有创新的解决方案常常来自于你认识到你对问题的概念是错误的。（你在开发中碰壁了，头破血流，反省后才能得到最好的解决方案）
- 13. “最好的设计不是再也没有什么东西可以添加了，而是再也没有什么东西可以去掉。”

- 14. 任何工具都应该能以预想的方式使用，但是一个伟大的工具提供你没料到的功能。
- 15. 当写任何种类的网关型程序时，多费点力，尽量少干扰数据流，永远不要抛弃信息，除非接收方强迫这么作！

- 16. 如果你的语言一点也不象是图灵完备的，严格的语法会有好处。 (rc file, control file syntax)
- 17. 一个安全系统只能和它的秘密一样安全，当心伪安全。

集市风格的必要的 先决条件

- 不能以市集模式从头开发一个软件，我们可以以市集模式测试、调试和改进，但是以市集模式从头开始一个项目将是非常困难的。
- 当你开始创建社团时，你需要演示的是一个诺言，你的程序不需要工作的很好，它可以很粗糙、很笨拙、不完整和缺少文档、它不能忽略的东西是要吸引大家卷入一个有趣的项目。

市集模式组织者

- 我认为能够提出卓越的原始设计思想对协调人来说不是最关键的，但是对他 / 她来说绝对关键的是要能把从他人那里得到的好的设计重新组织起来。
- 对市集项目来说，我认为还有另一种通常与软件开发无关的技能和设计能力同样重要——或者更加重要，市集项目的协调人或领导人必须有良好的人际和交流能力。

- 这是很显然的，为了建造一个开发社团，你需要吸引人，你所做的东西要让他们感到有趣，而且要保持他们对他们正在做的工作感到有趣，而且要保持他们对他们正在做的工作感到高兴，技术方面对达成这些目标有一定帮助，但这远远不是全部，你的个人素质也有关系。

- 18. 要解决一个有趣的问题，请从发现让你感兴趣的问题开始。
- 最好的开发是从作者解决每天工作中的个人问题开始的，因为它对一大类用户来说是一个典型问题，所以它就推广开来了。

交流与沟通

- 在《人月神话》一书中，Fred Brooks观察到程序员的工作时间是不可替代的：在一个误了工期的软件项目中增加开发人员只会让它拖得更久，他声称项目的复杂度和通讯开销以开发人员的平方增长，而工作成绩只是以线性增长，这个说法被称为“Brooks定律”，被普遍当作真理，但如果Brooks定律就是全部，那Linux就不可能成功。

- 几年之后，Gerald Weinberg的经典之作“The Psychology Of Computer Programming”为我们更正了Brooks的看法，在他的“忘我(egoless)的编程”中，Weinberg观察到在开发人员不顽固保守自己的代码，鼓励其他人寻找错误和发展潜力的地方，软件改进的速度会比其他地方有戏剧性的提高。
- 虽然编码仍是一个人干的活，真正伟大的工作来自于利用整个社团的注意和脑力，在一个封闭的项目中只利用他自己的脑力的人会落在知道怎样创建一个开放的、进化的，成百上千的人在其中查找错误和进行修改的环境的开发人员之后。

Linux 与 Internet

- Linux是第一个有意识的成功的利用整个世界做为它的头脑库的项目，我不认为Linux的孕育和万维网的诞生相一致是一个巧合，而且Linux在1993-1994的一段ISP工业大发展和对Internet的兴趣爆炸式增长的时期中成长起来，Linus是第一个学会怎样利用Internet的新规的人。

- Linux世界的行为更象一个自由市场或生态系统，由一大群自私的个体组成，它们试图取得(自己)最大的实效，在这个过程中产生了比任何一种中央计划都细致和高效的自发的改进的结果。
- Linux黑客取得的最大化的“实际利益”不是经典的经济利益，而是无形的他们的自我满足和在其他黑客中的声望，(有人会说他们的动机是“利他的”，但这忽略了这样的事实：利他主义本身是利他主义者的一种自我满足的形式)。

- 19. 如果开发协调人员有至少和Internet一样好的媒介，而且知道怎样不通过强迫来领导，许多头脑将不可避免地比一个好。
- 我认为自由软件的将来将属于那些知道怎样玩Linux的游戏的人，把大教堂抛之脑后拥抱市集的人，这并不是说个人的观点与才气不再重要，而是，我认为自由软件的前沿将属于从个人观点和才气出发的人，然后通过共同兴趣自愿社团的高效建造来扩展。

- 也许最终自由软件文化将胜利，不是因为协作在道德上是正确的或软件“囤积居奇”在道德上是错的(假设你相信后者，Linus和我都不)，而仅仅是因为商业世界在进化的军备竞赛中不能战胜自由软件社团，因为后者可以把更大更好的开发资源放在解决问题上。

Thank you.

邵栋

dongshao@nju.edu.cn



南京大学 软件学院
NANJING UNIVERSITY SOFTWARE INSTITUTE