

<<开源技术选型手册>>

图书基本信息

书名：<<开源技术选型手册>>

13位ISBN编号：9787121068034

10位ISBN编号：7121068036

出版时间：2008-1

出版时间：电子工业出版社

作者：开源技术选型手册委会

页数：525

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<开源技术选型手册>>

内容概要

包括的领域有开发平台、Web框架、应用服务器、数据库、版本控制、项目管理、富客户端、面向服务架构、工作流等。

如果这个世界没有了选择，那将会是什么样子？

不可想象。

如果这个世界选择太多，那将是什么样子？

不可想象。

相信很多朋友置身于开源技术领域时，都会遇到选择的困扰。

经过多年的发展，开源技术越来越成熟，一个稍有经验的开发人员经过学习都能用开源的产品和技术构建出一套可用的系统。

但是，很多时候我们需要的不仅仅是一个可用的系统，而是希望这个系统开发更简易，性能更高，扩展性更好等。

那么在这个时候，专家的经验就变得弥足珍贵。

<<开源技术选型手册>>

书籍目录

第1章 闲话开源社区篇第2章 Web框架篇 5 2.1 Struts 7 2.2 Spring 14 2.3 Seam 26 第3章 开源Web服务器 41
3.1 Apache 43 3.2 Lighttpd 46 3.3 Nginx 49 第4章 应用服务器篇 51 4.1 JBoss 53 4.2 Geronimo 60 4.3 JFox 66
第5章 开发平台篇 (IDE) 81 5.1 Eclipse 85 5.2 NetBeans 96 第6章 动态语言篇 105 6.1 Python 107 6.2 Ruby
112 6.3 Groovy第7章 Ajax开发篇第8章 富互联网应用篇 (RIA) 第9章 版本控制篇第10章 项目管理篇
第11章 面向方面编程篇 (AOP) 第12章 反向控制容器篇 (IOC) 第13章 用户界面篇 (UI) 第14章 商业
智能篇 (BI) 第15章 建模工具篇第16章 对象关系映射篇 (ORM) 第17章 数据库篇 (DB) 第18章 软件
测试篇第19章 面向服务架构篇 (SOA) 第20章 工作流篇 (Workflow) c

章节摘录

第1章 闲话开源社区篇 对于从事软件开发的人员，尤其是对Java或动态语言相关领域的人来说，“开源”也许是他们最喜爱的单词。

每天，当他们坐在电脑前开始编程时，从所使用的Linux操作系统，到Eclipse集成开发环境，到编程所用的Java或Ruby、Python语言，“开源”烙印在每件工具身上。

如果按照我们通常所说的，对待件事情“不仅要知其然，还要知其所以然”，那么我想很多开发人员在这个问题上应该是不合格的，而这也是本文所要着重解决的问题。

本文试图从开源的历史、开源运动中相关的一些重量级人物，以及国内外的主要开源社区等方面展开讨论，对“开源”做简要分析。

本文标题之所以加了“闲话”二字，初衷是因为开源话题庞杂，非几本专著所能阐述清楚，更不用说一篇小文了。

本文很多地方都是点到为止，要深入了解相关信息的读者请借助Google和文末的参考资料、网站等自行挖掘。

开源软件诞生小史 要谈开源，首先需要了解开源之前的世界是什么样子。

在软件产业高速发展的背后，知识产权有着功不可没的作用。

也就是有着它的保护，软件领域很快涌现出了像微软、IBM、Oracle、BEA等一大批知名软件公司，分析这些公司的特点，我们可以发现它们对其旗下的核心产品都有着占有权，也就是它们所生产的软件的源代码对外界来说是严格保密的，其所造成的后果就是在带给用户便利的同时也送来了诸多不便。

举个例子来说，在使用这些代码非开源的软件时如果遇到问题，客户是没有办法通过自行检测代码来解决的，他们只能求助于产品提供商，而这一过程可能就会给客户带来难以估量的损失。

另外我们也不难看出，对源代码的保密还带来了大量重复的劳动，阻碍了创新脚步。

上面我们所提到的这种非开源的软件通常被我们称为“商业软件”。

随着人们对软件的认识越来越深入，商业软件的垄断也引起了很多软件用户和计算机专家的不满。

麻省理工学院在1984年率先发起了自由软件运动，而且成立了自由软件基金会。

其代表人物是Richard Stallman，他在《为什么软件不应有所有者?》一文中提到，“一个软件并不是单纯的工具，一旦软件的编写者将其传播出去，就不单单是在提供‘工具’，而是在传播‘思想’”，同日寸，他也对现有的版权体系进行了批判。

事情还不仅仅如此，Stallman还试图创造一种有别于传统的知识产权模式，避免垄断，加强软件技术交流与合作的软件开发模式。

这种模式要求软件开发者开放源代码，允许任何使用者阅读源代码并可以进行修改，同时也不需要向软件开发者支付使用费。

Stallman还表示用户之间批次拷贝软件不但不是盗版，还体现了人类天性的互助美德。

对Stallman而言，自由是根本。

从用户角度来看，自由软件好处多多，但对于开发者或生产厂商而言，这不啻是一个噩耗。

因为很多人会将自由软件理解成“免费”，针对这种情况，自由软件基金会在版权声明中明确强调“当提到自由软件时，我们指的是自由而不是价格……”。

尽管如此，商业公司还是将这种模式视为一种威胁，有可能会使他们无利可图。

为了解除这种威胁，Stallman创造了GNU通用公共许可证，即GPL。

GPL的基本意思是你可以任意拷贝和散发根据GPL许可授权的软件，并且你不能阻止他人也做同样的事情。

另外，GPL还要求从GPL作品衍生出来的作品也必须同样遵循此条款的约束。

由此可见，自由软件显然是违背商业信条的，这也导致了人们对自由软件越来越敬而远之。

在这个背景之下，1997年春天中的一天，自由软件社团中的一部分人，如Eric S.Raymond（《大教堂与集市》一文作者）等决定寻找另外一种可以赢得更多人认同的新模式，这也是今天和我们的开发生涯息息相关的开源软件。

与GPL相比，开源软件的定义允许更大的授权自由，特别是当软件中混合了商业软件和开源软件时，

<<开源技术选型手册>>

开源软件被允许在这些软件基础上做更自由的创作。

开源软件定义 下面我们简单地了解一下关于开源软件的定义，官方的定义版本写得很全面，这里我们仅仅挑选一些重要的条款来解释。

开源软件需要遵循的首要标准就是要能够“自由地重新发行”，被认证为开源软件的产品不能限制任何参与者销售，也不能限制将该软件作为其他集合软件的一部分发行，也不能要求使用该开源软件的人支付版税或其他费用。

从这一点可以看出开源软件较自由软件的概念更为自由。

对于开源软件中“源代码”的定义，是指在提供软件的同时也要提供程序源代码，也必须允许其他人对代码进行编辑或修改。

如果发行软件时没有附带源代码，那么源代码必须能够很容易且低成本地获得，比如可以从网络上免费下载等。

为了便于程序员阅读源代码，它们必须要以比较友好的格式下载，以便容易地编辑。

那些故意写得很模糊的源代码是不被允许的。

从上面这一点来看，“开源”更大意义上来说是一种文化上的分享。

其他限制条款，还包括开源软件必须允许修改和衍生作品，并且必须允许他们在原软件的许可证下进行发行。

而且开源软件也不能限制任何人将程序使用于任何目的，包括不能限制程序用于商业用途等。

也许正是这一点，让包括IBM、BEA、Oracle，甚至连微软在内的公司开始染指开源软件领域的原因吧。

<<开源技术选型手册>>

编辑推荐

《开源技术选型手册》特别邀请了多位国内技术社区的专家，请他们结合自己多年的开发经验，对当前开源技术领域流行的技术或者产品进行比较分析，以期为中高级技术人员在工作中做技术选型时参考使用。

<<开源技术选型手册>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>