

开源软件与开源软件生态:现状与趋势

金芝,周明辉,张宇霞

高可信软件技术教育部重点实验室(北京大学),北京 100871

摘要 开源正在侵吞软件世界。目前已有80%的软件声称使用了开源的方式进行开发。开源技术占据主导地位领域包括云、内容管理、安全、移动和社交媒体等。本文回顾了开源软件(OSS)的历史发展脉络,探讨了开源软件的研究进展及开源软件对技术、行业和社会的影响,展望了开源软件生态系统的演变趋势。

关键词 软件技术;开源软件;开源软件生态

开源软件,从字面上说,是指开放源代码的软件。但开源软件不是简单的源代码开放,而是代码创作者在遵循相关开源协议(如MPL许可证等)的基础上,将自己的源代码全部或部分向世界公开,允许用户进行自主学习、报错、修改等活动,以共同提高软件的质量。

开源运动的先驱者通过对早期自由软件活动的深刻认识,有效地解决了开源软件在法律和商业方面遇到的问题,建立起一种群体参与的软件开发方法和生态环境,将分布在全球的个体智慧汇集到开源软件中,把用户对高品质软件的需求、企业商业战略、抑制技术垄断、产业良性循环等诸多目标有效地集成到开源活动中,实现了对软件产业的重大变革,使开源软件开发成为一种重要的软件开发形式和研究热点。

20世纪末,开源软件取得的很大成功,为软件开发提供了一种用户创新驱动、成本低、质量高的新思路。由于认识到开源软件的优势,越来越多的公司和组织参与到开源运动中,建立起了商业-开源混合项目,并驱动搭建围绕开源软件技术和平台的各种业务模型,促进项目参与者(Stakeholders)之间的协作和利益关联,形成了“开源软件生态系统”。

开源软件是人类历史上一次利用

互联网实现群体参与、分布协作,进行软件创作活动的重大实践,在开发模式上展现出无偿贡献、用户创新、充分共享、自由协同、持续演化的新特征^[1]。本文介绍开源软件及其开源生态的发展历程和趋势

1 黑客文化、自由软件到开源软件

自由/开源软件起源于“黑客文化”^[2]和对软件私有化的抗争。早期,在计算机程序设计领域,“黑客”是一个褒义词,指那些有才干、喜欢解决难题并乐于贡献的程序员。当时的硬件价格非常昂贵,软件却不需付费购买。当需要软件来满足某种特定需求时,通常由使用者自己编写代码来实现。20世纪60—70年代,大部分软件都是由科研人员自己编写。当时,人们普遍认为软件应该在研究环境中自由分享、交换,可以被他人修改或者重新编译,并应该与他人分享这些修改。这种自由分享软件源代码的行为就成为“黑客文化”的一个重要特征。这种文化在麻省理工(MIT)人工智能实验室软件开发小组中非常盛行^[2,3]。

20世纪70年代末,微软开始倡导软件私有化,声明软件拷贝需要付费,并且只提供可执行程序而非源代码。

20世纪80年代,MIT将由黑客编写的代码授权给一个商业公司,但该公司却严格限制对这些源代码的访问,阻止MIT的黑客们继续学习和开发这些软件。MIT人工智能实验室的Richard Stallman^[4]对源代码访问权限受限和当时软件私有化(特别是对广泛使用的操作系统Unix的商业私有化)的趋势非常不满,提出了自由软件的想法。1984年,Richard Stallman辞去了MIT的工作,成立了自由软件基金会(Free Software Foundation, FSF),希望通过自由创作的方式开发出一套操作系统。该计划称为GNU计划,允许任何人自由地下载、使用、修改和发布该软件。为从法律上保证这种自由的权利,经过斟酌,Stallman提出GNU通用公共授权许可证(GNU General Public License, GPL)。那些有意将自己的软件作为自由软件的作者都可以简单地附上这样一份标准许可证,来保证未来用户的权利。这种软件被称为“自由软件”。

对于自由软件,基本的权利包括免费使用、学习、修改以及免费分发修改过的或未修改的版本等。但自由软件的想法并未立即成为主流,按照GPL的要求,使用GPL授权的软件(即自由软件的衍生软件)也必须是自由的,不能转化成私有软件。并且,如果在某个程

收稿日期:2016-05-30;修回日期:2016-06-22

基金项目:国家重点基础研究发展计划(973计划)项目(2015CB352201)

作者简介:金芝,教授,研究方向为需求工程、知识工程和基于知识工程的软件工程等,电子信箱:zhijin@pku.edu.cn

引用格式:金芝,周明辉,张宇霞. 开源软件与开源软件生态:现状与趋势[J]. 科技导报, 2016, 34(14): 42-48; doi: 10.3981/j.issn.1000-7857.2016.14.005

序中使用了 GPL 授权的代码,那么整个组合软件也必须用 GPL 发布为自由软件,这一点被认为是 GPL 的“传染性”。工业界很难接受这种想法。

1998 年, Bruce Perens 和 Eric Raymond 指出,在 Stallman 的“自由”软件中存在这一问题,他们与其他优秀的黑客一起发起了“开放源代码(Open Source)”的号召,它包含了与之前的自由软件运动类似的许可证内容,但倾向对软件许可进行更少的限制。例如,关于衍生软件,开源运动提出,许可证必须允许修改和继续衍生,允许(并不强制)它们按照和原始软件相同的许可条款进行发布。

从这个意义上说,开源在理念上与自由软件运动不同,开源运动不纠结“道德公正性”和“赋予用户在使用软件中真正的自由”等问题,而是强调许可证带来实际利益,如人们对软件源代码的免费访问、学习和使用等。这样,开源不再与商业公司对立,随着越来越多商业友好的开源许可证的出现,如 Apache 许可证等,开源软件逐步进入商业软件生产环境,同时也成为软件工程领域一种重要的现象和学术研究的一个重要课题。

目前,开源软件广泛应用于各个领域,著名开源软件公司 Black Duck Software 和投资公司 North Bridge Venture Partners 联合公布的《The Ninth Annual Future of Open Source

Survey》报告显示,目前世界上独立的开源软件项目已达到百万量级(图 1),即使在 IT 设施更新迟缓的领域,如政府、医疗和媒体等部门,开源软件项目的数量也在逐年快速增长,相当多的商业公司都以各种形式参与到开源软件的生产应用中。

2 开源项目和开源社区:研究关注点与研究内容

自 20 世纪末以来,开源软件取得了令人瞩目的成就,无论在开发质量或效率上,成功的开源软件都达到了与商业软件相媲美的程度。很多开源软件的市场占有率已经远超同类商业软件,对全球软件产业的格局产生了重大影响。

与传统商业软件相比,开源软件在开发模式上展现出充分共享、自由协同、无偿贡献、用户创新等新特征。分布在全球的开源软件开发者在基于互联网的虚拟社区中进行交互与协同,颠覆了一些经典软件工程的基本假设和理论。开源软件项目并没有传统软件工程项目中显式的组织结构和过程安排,但所开发出的软件却可以与传统商业软件相抗衡,这引起了学术界和企业界的广泛关注^[5-7]。

在过去十几年的时间里,众多研究者尝试从不同的角度来理解开源软件的开发实践,总结其成功因素,以促进开源软件的发展。这些研究主要围绕以下几个方面的问题展开:

1) 探究开源软件的开发过程与传统软件开发过程的区别。例如,开源软件的开发是否也有传统公司环境下软件工程中的需求获取与分析、设计、实现、测试、发布等环节?

2) 寻找开源项目吸引、保留贡

献者,进而围绕开源软件形成了开源社区的原因。例如,为什么这些软件开发者会为并不会支付任何报酬的项目做贡献?

3) 扩展现开源软件生态系统的概念范畴,探索开源软件生态系统的规律,构建成功的开源软件生态系统等。

学术界和企业界对开源软件研究的关注点不仅在于开源软件制品本身,还涉及到整个开源软件生态系统。目前针对以上研究关注点,对开源软件的研究内容大致分为 2 类:第 1,对开源软件现象进行分析和理解;第 2,基于这些分析和理解对开源软件乃至商业软件的开发实践进行改进。

2.1 理解开源软件项目

理解开源软件方面最早的代表性工作 Mockus 等对 Apache 和 Mozilla 这 2 个开源项目的研究^[8],主要对核心开发团队的规模和任务量,以及团队的组织方式等进行了量化分析。German 遵循类似原则量化了 Gnome 社区的分布式开发实践^[9]。使得对由大量志愿者参与、组织松散的分布式开源软件开发模式如何能够高效生产高质量软件有了一些基本的认识。

在开源软件的开发过程中,如何弥合全球开发者在语言、文化、时区、习俗方面的差异也吸引了很多关注。有很多研究致力于理解开源分布式协作开发中的沟通,以及开发者协作方式对产品质量的影响等问题。例如, Cataldo 等通过度量程序员的协作行为,发现许多高产的程序员会随时间改变他们的电子通信媒介,以更好地满足协作需求^[9]。

开源软件开发与商业软件开发的巨大差别,使得开源开发活动的组织、管理和协同一直是热点议题。例如,开源项目的管理相当精简,通常没有规范的开发计划和过程说明,甚至没有设计文档^[9];许多开源项目的主要推动力来自用户的外部贡献。特别地,随着 Pull-Request 等新型开发机制的发展,越来越多的外围大众参与到软件生产活动中来,使得大众开发成为现代软

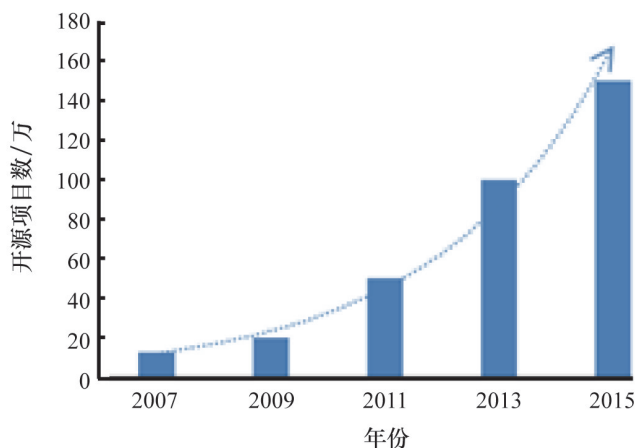


图 1 Black Duck 全球开源项目增长趋势

件开发的一个标志。

2.2 度量和改进开源实践

基于经验理解软件开发活动,进而制定相应的开发原则和方法,是软件工程采用的基本研究方法。但是由于缺少量化方法,基于经验理解所获得的软件工程相关结论很难在软件开发实践中定量实施,进而难以得到有效的传播和普及。

在开源成为一种流行现象后,互联网上积累了海量开放的软件开发和应用数据,为软件开发活动的量化分析提供了良好条件。软件开发支持工具(如版本控制系统,问题追踪系统,邮件系统等)的广泛应用也积累了大量数据。很多研究工作采用量化分析的方法利用开源数据帮助和改进开源开发的各个方面。

Zimmermann 等挖掘版本历史以指导程序员在源代码发生变更的时候进行其他相应变更^[10]; Michail 使用泛化的关联规则挖掘来发现函数库的复用模式^[11]; Xie 等使用频繁序列挖掘技术来分析 API 的调用频度和调用顺序,以帮助代码开发人员理解 API 的使用方法^[12]。

其他工作还包括:探究开源项目中的缺陷分类活动(bug triage)^[13]、寻找重复的问题报告(duplicate bug)^[14]、预测问题被修复的概率^[15]等。还有一部分研究关注开源社区中的贡献者本身,涉及贡献者如何进入社区、如何学习、贡献质量如何度量、社区如何维护长期贡献者^[11,15-18]等多个方面。

开源数据还为软件本身进行度量和评估提供了一种有效的方法。例如, Yin 等从软件用户的角度入手提出了一种基于互联网用户评论的软件质量度量方法^[9],对互联网上存在的海量用户评论的正面和负面倾向进行情感分析,并基于评论数据从轻量、易用、稳定等6个方面对软件质量形成综合评估。

基于数据的开源软件量化分析是揭示开源开发机理的重要研究途径,也是近年来一个非常活跃的研究方向^[20]。图2给出了开源数据量化分析的一个基本脉络图。

开源软件的这些不同阶段、不同侧面的数据反映了软件开发和应用的历史和动态,并提供了数量庞大的项目样本^[20],为软件工程研究和实践带来了新的思维模式和研究问题,例如:如何有效地收集、组织和运用这些数据来重新审视软件工程的本质复杂性问题?如何在新的开发模式下探寻新的软件过程规律,建立可复制的软件开发最佳实践?特别地,大数据的存在使得人们可以突破传统宏观层次的软件过程,在微观层次上对不同上下文的开发活动观察和分析,使得从微观层次度量开发活动模式^[21]成为可能。这些挑战和机会对软件的发展乃至整个信息技术领域的发展都具有重要的理论价值和现实意义。

3 开源软件的发展和产业投入

开源已经被认为是软件技术创新和产业发展的主要模式,据 Black Duck

在 2015 年统计^[22],全球目前共有大约 180 万个开源软件,且其数量依然呈现快速上升态势。全球排名前 10 的 IT 企业所开发的新产品和服务,有 80% 的软件创新来自企业外部的开源软件,而企业内部的自创成果只占 20%。全球所有 IT 企业中 78% 的企业其业务主要建立在开源软件之上,只有不到 3% 的企业没有使用任何开源软件,相比 2010 年,开源软件的使用量已经增加了 2 倍。全球开源软件项目风险投资持续增长(图 3),截止到 2014 年,已经达到了 1300 百万美元。

开源软件的蓬勃发展,其所具有的大众化协同、开放式共享、用户持续创新等产品开发新模式和快速发展、颇具潜力的商业模式使其得到世界各国政府的大力推广和发展。各国政府采取的策略主要包括:制定策略大力推广对开源的采用;为开源机制的建立和开源技术的使用提供指导 Governance, IP

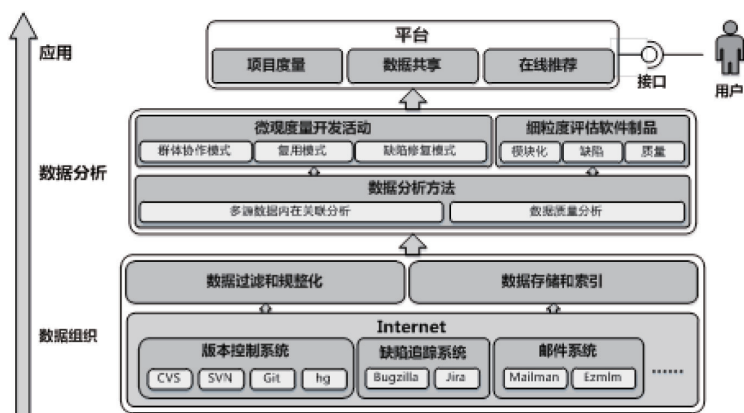


图2 开源数据量化分析的脉络

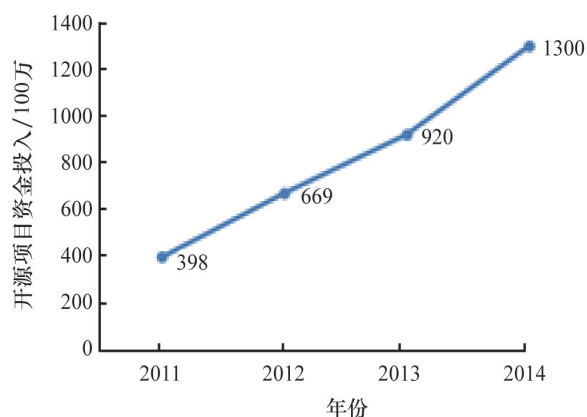


图3 Black Duck全球开源项目资金投入统计

等;基于公共部门建立开源基础设施和协同平台;围绕自有技术积极促进开源社区的建立和发展;面向创新技术建立 initiative 等。

开源已经成为软件领域技术、产品创新的重要模式,也是驱动信息产业变革、强化信息产业基础的关键要素,根据 Gartner 的预测,到 2016 年,至少 95% 的主流 IT 企业和组织将会直接或间接地在其关键任务方案中使用开源软件。企业使用和参与开发开源软件的动机不仅包括促进本身的创新和减少开发成本,更在于能够紧密地跟随技术发展趋势,甚至对行业的发展方向形成影响,保持和提升企业竞争力。全球如 Google、FaceBook 等互联网巨头纷纷基于开源提升其软件技术能力、开发新的软件功能及产品服务。就连传统商业软件巨头微软也加大了对开源世界的贡献,全面投入了开源的怀抱。

在云计算、大数据等新兴领域,基于开源模式的技术创新对产业发展的影响力日益提升。移动网络、云计算、大数据等技术的不断创新,带动了 OpenStack、Hadoop 等开源软件的迅速发展。近年来发展最为成功的开源项目要数 Apache Hadoop,海量数据革命掀起的浪潮带动了数据分析行业的增长。要在网络上处理如此大规模的资料,目前比较经常被使用的平台是 Hadoop,而 Facebook 使用的即是 Hadoop 下的 HBase 数据库。未来几年内,开源 PaaS 框架、开源数据管理工具等开源技术将被越来越多的企业关注和使用,推动开源技术从基础层逐渐向上渗透。值得一提的是,在成千上万的开源项目中,成功的项目为数很少,例如,GitHub 上 98% 的项目在 1 年之后就不再活跃。Crowston 等提出早期开源软件项目的成功包括 3 个因素:软件开发者的贡献、软件的使用反馈和软件质量^[23],但企业的参与和市场的选择也许是成功的关键性因素,例如,相比 FreeBSD 等,Linux 发布版更为成功,也许在很大程度上借力了 Android 的蓬勃发展及其形成的软硬件生态。

国内开源软件兴起于 1997 年前

后,经过近 20 年的发展,其参与群体正发生着很大的改变。最早主要以中科红旗和中软网络为代表的本土 Linux 企业为代表。从 2005 年起,其他行业的一些重量级企业也加入到开源软件的开发及应用中来,进入了以大型企业为主导的阶段。阿里作为 Linux 内核 ext4 文件系统核心贡献者之一,不仅为 Hadoop、MySQL、OpenJDK 等多个开源项目贡献大量代码,而且采用开源软件构建了全球第一大电商平台,并开源了 100 余个自主开发的软件,如服务框架 Dubbo、分布式文件系统 FastDFS、开源项目托管平台 TaoCode 等。腾讯和 CSDN 于近期共同宣布,由腾讯公益慈善基金会出资成立 CSDN 腾讯开源公益社区,逐步将腾讯开源项目通过该社区发布出去,目前腾讯内部开发者社区已初成规模,积累项目达 500 余个,开放源代码项目 200 余个。OSChina 也逐步建立了开源软件库、项目托管、代码分享、翻译等平台化社区工具,吸引了超过 10 万个社区项目。

中国对国际开源社区贡献逐年增多,其中华人对 Linux 内核贡献在最新 4.2 版中位列国籍排行榜的第一位,在 OpenStack、Hadoop、MySQL 等项目社区中也越来越活跃。值得一提的是,华为公司已经成为国内参与开源软件项目的佼佼者^[24]。华为从 2011 年开始加入开源活动,5 年的发展历程大致经历了 3 个阶段:第 1 阶段只是把开源单纯看作一个外购件;第 2 阶段则把开源看做外部协作的一种方式,研究如何让与社区的技术人员、社区本身以及其他公司协作;第 3 阶段则把开源软件当作自己的代码,开源被提升到战略高度来看待和管理,开源软件成为研发的重要来源。近年来,华为公司设立了专门的开源软件中心,积极支持参与外部社区,成为 Apache 基金会、Linux 基金会、Linaro、OpenStack 等开源社区的主要成员。以 OpenStack 为例,华为已经成为其金牌会员,可以参与到整个 OpenStack 架构层面的研究和创新层面的变革。公司每月向各大开源社区回馈约 1500 个补丁,仅次于 RedHat、Intel

和 IBM,社区活跃程度大致与 Google、Samsung、Oracle、Microsoft 等公司相当。

然而,总起来说,国内对开源的参与在贡献力度和领导力方面跟国外相比还有较大差距,例如自有项目一般规模较小,缺乏其他地区的参与者等。并且,中国软件开发者群体具有自己的特点、需求和文化习惯,因此,寻求发展时不仅要吸纳国外经验,在构建中文开源技术和社区方面也需要有所创新。

4 当前研究与技术发展趋势

与传统商业软件相比,开源软件在开发模式上呈现出充分共享、自由协同、无偿贡献、用户创新等新特性。分布在全球的开源软件开发者在基于互联网的虚拟社区中交互与协同,颠覆了经典软件工程的许多经典假设和理论。开源软件的成功及其颠覆传统的开发模式,对软件研究领域产生了巨大的冲击,吸引了一大批来自企业和学术届的研究者对其展开研究。当前较为流行的研究工作整体可以分为开发策略与项目管理、软件生态系统建设、开源软件量化分析及其工具研究、社会多角度研究等层面。

在软件项目开发策略上,为利用开源软件开发的优势,越来越多的公司和组织参与到开源软件的创建、开发和维护中,建立起很多商业-开源混合开发的项目,如 Mozilla,JBoss, Eclipse 等^[25,26]。这些项目由软件工业界驱动,围绕开源软件技术或平台搭建各种业务模型;利益相关者之间相互协作、利益彼此关联,形成了各种“开源软件生态系统”^[27]。然而,由于开源软件与传统商业软件的开发方法存在多方面的冲突,两者在实际的结合中会遇到诸多问题,如众多志愿者参与的大型软件开发如何保证软件产品或服务的质量?如何对待开源软件对其生态系统中参与者的活动产生负面影响,例如,开源软件开发强调开放、免费、贡献者自由参与。

另一方面,商业软件开发强调以盈利为目的、可控、统一进行任务分工。在开源软件生态系统中,商业公司始终

需要在“开放”与“盈利”之间权衡^[26]。如何能够在企业中有效利用开源特性及优势并非是一个容易回答的问题。Dinkelacker 等描述了 Hewlett Packard's (HP) 采用的渐进式开源策略^[28]。该策略提供了 3 种开放性逐渐增强的模型: 内部源码(在公司内部开放)、受控源码(在选择伙伴间开放)、开放源码(在全世界开放)。Wagstrom 等度量了开源项目 Gnome 和 Eclipse 中的公司行为, 总结出 2 种模式: 一种是聚焦社区的公司, 通过建立一个活跃社区来提供付费服务^[29]; 另一种是聚焦产品的公司, 通过销售产品获得利润。Ma^[27]和 Zhou^[30]等利用项目开发数据度量了 3 个同构的开源产品(JavaEE 应用服务器)中公司的参与活动, 发现公司的不同投入力度对外部参与者有重大影响^[27,30]。

海量开源软件相关数据的存在使得对开源软件的量化分析成为一种重要的研究方法。首先, 软件开发支持工具(如版本控制系统和问题追踪系统等)的广泛应用积累了大量数据, 记录着软件代码的演变^[4]、开发任务的流程等。截至 2015 年 4 月, GitHub 中的软件仓库(software repository)数量超过 1600 万。其次, 近些年兴起的软件知识分享社区(如 StackOverFlow)记录了大量的软件问答和软件评价等数据^[7]。截至 2014 年 9 月, StackOverflow 的问答数量超过 1300 万^[31]。对这些数据深层次分析将对软件工程的发展乃整个信息技术领域的发展都具有重要的理论价值和现实意义, 但相关的工作尚处于起步期。

另一部分研究工作关注度量工具所需的数据支撑机制。这是因为, 尽管利用项目开发过程数据来度量开源软件项目的健康状态是一种常用且有效的途径, 但研究者发现, 在使用这些数据的过程中会遇到很多问题, 导致影响度量方法的效率和度量结果的质量。例如, 各种类型的软件开发过程资源库规模庞大, 项目之间、资源库之间的数据格式不统一, 将这些数据收集、过滤并处理为度量可用信息的工作十分繁

重。近些年来研究人员逐渐研发了各种技术和工具来获取和处理开源数据并加以利用。例如, CVSAAnaly 实现了版本控制系统数据的处理^[32]; SofiChange 能够从源代码中推断各种隐含的事实^[8]; GHTorrent 将 Github 站点上项目的相关数据存储在一个数据库中, 供研究人员离线使用^[33]等。同时, 许多研究者尝试引入经典的统计分析和机器学习方法来利用软件开发活动数据。例如, 将传统的集成学习方法应用于软件工作量估算^[34]; 将迁移学习方法引入到缺陷预测问题^[35]等。也有一些工作尝试建立软件开发数据的量化分析方法, 例如一个量化研究的经典步骤包括: 获取原始数据并进行清洁和处理, 针对特定的研究问题建立相应的量度, 然后围绕量度展开分析, 最后验证结果并检查研究假设以确定是否对上述步骤进行迭代^[36]。数据质量对软件数据分析至关重要, 对数据质量的分析也正逐步成为一个重要研究方向。

在软件项目管理与开源生态系统建设层面, 开源软件开发与商业软件开发的差别, 使得开源开发活动的组织、管理和协同一直是热点议题, 软件产业和开源社区普遍关注的问题有: 在商业和开源 2 种开发方式进行融合的环境中, 如何建立一个健康的开源软件生态系统? 在开源项目贡献者遍布全球的情况下, 如何有效的进行项目管理? 特别地, 是否可以度量开源软件生态系统的健康状态, 从定量的角度提供决策支持? 研究者受自然生态系统的启发, 尝试从生态系统的角度探讨开源软件项目开发以及社区组成, 正在逐步形成软件生态系统理论体系。例如, Koch 和 Kerschbaum 调研了在智能手机领域自由软件开发者选择参与到生态系统中的动机^[37]; Dittrich 基于 4 个典型的软件生态系统探讨了软件生态系统的出现对管理和维护软件产品的传统观点带来的挑战^[38]; Jansen 提出了一个可以帮助决策者判断是否加入一个软件生态系统的概念, 叫做软件生态系统健康, 并且分析了多个软件生态系统对这一概念进行发展和完善^[39]。

开源软件生态作为一个研究领域在近 10 多年的发展中得到了越来越多的关注。由北京大学金芝教授牵头于 2015 年立项的国家重点基础研究发展计划(973 计划)项目, 提出研究基于开源生态的网构化软件开发原理和方法, 探索从新型互联网软件范型(网构软件)出发, 从基于群体智慧的软件开发、应用嵌入的软件自适应和演化, 以及开源场景下的软件可信性保证等 3 个方面, 研究网构化软件的生态构建的基本机制和支撑方法。

开源软件所蕴含的全新的经济理念和社会化生产方式, 展现出社会、经济、组织与管理、技术、实践等方面的重要属性, 吸引了包括社会学、经济学与管理学在内的诸多领域研究者的共同关注。从经济学角度, 研究者关心的核心问题是: 为什么会有大量的开发者个体自愿且无偿地参与到开源软件开发中? 从组织学的角度, 研究者则关注: 为什么一个看似松散无序的群体能够生产出高质量的软件制品? 传统上, 这些领域更多地采用定性方法进行研究。开放数据的存在正在为这些工作提供可行的定量研究途径。

5 结论

开源软件所代表的新的理念和独特的生产方式对创新模式、互联网发展、软件开发技术等产生了深远而巨大的影响。首先, 在创新模式上, 开源使得全球化的群智聚集, 为开源社区带来了巨大的创造力和生命力, 已经成为软件技术创新和产业发展的主要模式, 并创立了新的软件商业模式。其次, 在促进互联网发展上, 开源的共享创新与互联网的繁荣休戚相关, 互联网的每一次进化都伴随着开源软件技术的演进。而开源软件运动对网络经济大爆发起了至关重要的作用, 互联网反之也扮演了开源运动向前演进的巨大推手。最后, 在群体化开发方法上, 以开源软件为代表的互联网大规模群体创作活动则取得巨大成功, 为经典的软件开发方法带来新的重大启示, 其开源实践中蕴含的大众化协同、开放式共享、

持续性创新 3 类互联网大规模协同机理,为解决经典软件开发方法面临的难题提供了新的途径和机遇。

开源带来的的启示不仅体现在以上方面,还体现在更高抽象层析的理念方面。开源社区的协同共享,用户创新

等理念,已经扩展到软件行业之外,如开源硬件运动,促进了一种新的经济现象即协同共享经济的萌芽和持续发展。在某种意义上,开源理念正在对整个人类文明的发展产生深远的影响。对开源软件开发模式和机理的量化分

析的深刻意义是,通过对历史的量化理解,指导我们如何通过大规模的社会化协作去创建一个更加美好的信息化人类文明。

参考文献(References)

- [1] 梅宏, 金芝, 周明辉. 开源软件生态: 研究与实践[J]. 中国计算机学会通讯, 2016, 12(2): 22-23.
Mei Hong, Jin Zhi, Zhou Minghui. Open-source software ecosystems: research and practice[J]. Communication of the China Computer Federation, 2016, 12(2): 22-23.
- [2] Levy S. Hackers: Heroes of the computer revolution[M]. New York: Penguin Books, 2001.
- [3] Kirkpatrick G. The Hacker Ethic and the Spirit of the Information Age[J]. Max Weber Studies, 2002, 2(2): 163-185.
- [4] Richard Stallman. Richard stallman's personal site[EB/OL]. [2016-06-21]. <http://stallman.org>.
- [5] Mockus A, Fielding R T, Herbsleb J D. Two case studies of open source software development: apache and mozilla[J]. ACM Transactions on Software Engineering & Methodology, 2002, 11(3): 309-346.
- [6] Krogh G V, Hippel E V. Special issue on open source software development[J]. Research Policy, 2003, 32(7): 1149-1157.
- [7] Yamauchi Y, Yokozawa M, Shinohara T, et al. Collaboration with Lean Media: how open-source software succeeds[C]//Proceedings of the 2000 ACM conference on Computer supported cooperative work. New York: ACM, 2000: 329-338.
- [8] German D M, Hindle A. Measuring fine-grained change in software: towards modification-aware change metrics[C]//11th IEEE International Software Metrics Symposium (METRICS'05). Como: IEEE, 2005: 10-28.
- [9] Cataldo M, Wagstrom P A, Herbsleb J D, et al. Identification of coordination requirements: implications for the Design of collaboration and awareness tools [C]//Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work. New York: ACM, 2006: 353-362.
- [10] Zimmermann T, Weibgerber P, Diehl S, et al. Mining version histories to guide software changes[J]. IEEE Transactions on Software Engineering, 2005, 31(6): 429-445.
- [11] Michail A. Data mining library reuse patterns using generalized association rules[C]//Proceedings of the 22nd International Conference on Software Engineering. Limerick: IEEE, 2000: 167-176.
- [12] Xie J, Zhou M, Mockus A. Impact of triage: A study of mozilla and gnome[C]//2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. Baltimore, MD: IEEE, 2013: 247-250.
- [13] Anvik J, Hiew L, Murphy G C. Who should fix this bug?[C]//Proceedings of the 28th International Conference on Software Engineering. Shanghai: ACM, 2006: 361-370.
- [14] Runeson P, Alexandersson M, Nyholm O. Detection of duplicate defect reports using natural language processing[C]//29th International Conference on Software Engineering (ICSE'07). Washington, DC: IEEE, 2007: 499-510.
- [15] Guo P J, Zimmermann T, Nagappan N, et al. Characterizing and predicting which bugs get fixed: an empirical study of Microsoft Windows[C]//2010 ACM/IEEE 32nd International Conference on Software Engineering. New York: IEEE, 2010, 1: 495-504.
- [16] Zhou M, Mockus A. Who will stay in the floss community? modeling participant's initial behavior[J]. IEEE Transactions on Software Engineering, 2015, 41(1): 82-99.
- [17] Zhou M, Mockus A. Does the initial environment impact the future of developers [C]//Proceedings of the 33rd International Conference on Software Engineering. Honolulu, HI: ACM, 2011: 271-280.
- [18] Issman S, Steinmacher I, Conte T, Gerosa M A, et al. Social barriers faced by newcomers placing their first contribution in open source software projects [C]//Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. New York: ACM, 2015: 1379-1392.
- [19] Yin G, Wang T, Wang H, et al. OSSEAN: Mining crowd wisdom in open source communities[C]//Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on. San Francisco Bay, CA: IEEE, 2015: 367-371.
- [20] 周明辉, 张伟, 尹刚. 开源软件的量化分析[J]. 中国计算机学会通讯, 2016, 12(2): 24-27.
Zhou Minghui, Zhang Wei, Yin Gang. Quantification of open-source software[J]. Communication of the China Computer Federation, 2016, 12(2): 24-27.
- [21] Zhou M, Mockus A. Mining micro-practices from operational data[C]//Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM, 2014: 845-848.

- [22] Black Duck. Future of open source survey results[EB/OL]. 2015-04-16 [2016-06-13]. www.slideshare.net/~blackducksoftware/2015-future-of-open-source-survey-results.
- [23] Crowston K, Annabi H, Howison J. Defining open source software project success[J]. ICIS 2003 Proceedings, 2003, 12(31): 28.
- [24] 欧建深. 企业视角看到的开源-华为开源5年实践经验[J]. 中国计算机学会通讯, 2016, 12(2): 40-43.
- Ou Jianshen. Open source in enterprise-Huawei's best practice[J]. Communication of the China Computer Federation, 2016, 12(2): 40-43.
- [25] Riehle D. The economic motivation of open source software: Stakeholder perspectives[J]. Computer, 2007, 40(4): 25-32.
- [26] Andersen-Gott M, Ghinea G, Bygstad B. Why do commercial companies contribute to open source software?[J]. International Journal of Information Management, 2012, 32(2): 106-117.
- [27] Ma X, Zhou M, Riehle D. How commercial involvement affects open source projects: Three case studies on issue reporting[J]. Science China Information Sciences, 2013, 56(8): 1-13.
- [28] Dinkelacker J, Garg P K, Miller R, et al. Progressive open source[R]. Palo Alto: HP Laboratories, HPL-2001-233, 2001.
- [29] Wagstrom P, Herbsleb J D, Kraut R E, et al. The impact of commercial organizations on volunteer participation in an online community[C]. Academy of Management Annual Meeting. Montreal, Canada, August 7, 2010.
- [30] Zhou M, Mockus A, Ma X, et al. Inflow and retention in OSS communities with commercial involvement: a case study of three hybrid projects[J]. ACM Transactions on Software Engineering and Methodology, 2016, 25(2): 1-29.
- [31] Vasilescu B, Filkov V, Serebrenik A. StackOverflow and GitHub: Associations between software development and crowd-sourced knowledge[C]// Proceedings of 2013 IEEE International Conference on Social Computing (SocialCom). New York: IEEE, 2013: 188-195.
- [32] Robles G, Koch S, Wien W, et al. Remote analysis and measurement of libre software systems by means of the CVSAnalY tool[J]. Journal of Clinical Microbiology, 2004, 42(11): 5264-5269.
- [33] Gousios G, Spinellis D. GHTorrent: GitHub's data from a firehose[C]//Mining Software Repositories (msr), 2012 9th IEEE Working Conference on. Zurich: IEEE, 2012: 12-21.
- [34] Kocaguneli E, Menzies T, Keung J W. On the value of ensemble effort estimation[J]. IEEE Transactions on Software Engineering, 2012, 38(6): 1403-1416.
- [35] Nam J, Pan S J, Kim S. Transfer defect learning[C]//Proceedings of the 2013 International Conference on Software Engineering. San Francisco: IEEE, 2013: 382-391.
- [36] Mockus A. Software support tools and experimental work[M]//Basili V, Rombach D, Schneider K. Empirical Software Engineering Issues. Critical Assessment and Future Directions. Berlin Heidelberg: Springer, 2007.
- [37] Koch S, Kerschbaum M. Joining a smartphone ecosystem: Application developers' motivations and decision criteria[J]. Information & Software Technology, 2014, 56(11): 1423-1435.
- [38] Dittrich Y. Software engineering beyond the project-Sustaining software ecosystems[J]. Information & Software Technology, 2014, 56(11):1436-1456.
- [39] Lucassen G, van Rooij K, Jansen S. Ecosystem health of cloud Paas providers[C]//International Conference of Software Business. Berlin Heidelberg: Springer, 2013: 183-194.

Open source software and its eco-systems: Today and tomorrow

JIN Zhi, ZHOU Minghui, ZHANG Yuxia

Key Laboratory of High Confidence Software Technologies(Peking University), Ministry of Education, Beijing 100871, China

Abstract Open source becomes ubiquitous in the software world. Eighty percent of software is claimed to be developed using open source approach. Open source technologies dominate such domains as cloud, content management, security, mobile, social media, etc. In this article we review the history of open source software (OSS) development, the research on OSS, and the impact that OSS has on technologies, industry, and society. We conclude with the evolution trend that we picture of the future OSS ecosystem.

Keywords software; open source software; open source software ecosystem

(编辑 韩丹岫)