开源软件的量化分析

周明辉[1] 张 伟[1] 尹 刚[2]
[1]北京大学
[2]国防科学技术大学

关键词：开源软件 量化分析

20 世纪末以来，开源软件取得了令人瞩目的成就。无论是开发质量，还是开发效率，成功的开源软件都可以与商业软件相媲美；很多开源软件在市场上的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呈现出充分共享、自由协同、无偿贡献、用户创新等特征。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交互与协同，颠覆了经典软件工程的许多基本假设和理论。开源软件的巨大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，吸引了一大批研究者对其展开研究 [1]。基于数据的开源软件量化分析是全面揭示开源开发新机理的重要研究途径，也是近年来非常活跃的一个研究方向。

从定性到定量

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，是软件工程领域采用的基本研究方法。例如，模块化是软件工程的一个重要原则，而这一原则正是帕纳斯 (Parnas) 从开发经验中提炼产生的 [2]。基于经验的软件工程需要解决两个核心问题：一是如何从经验中提取出抽象的原则；二是如何将抽象的原则应用到新的软件开发活动中。从目前来看，能否将定性的抽象原则有效地应用到实践中，取决于实施者的经验和软件项目的具体环境（例如，针对一个具体的软件如何实现模块化，这个问题并不存在兼具一般性和易操作性的答案）。也就是说，将一个项目的成功经验复制到另一个项目中，是一件非常困难的事情。

导致出现这种问题的根本原因是，不同项目的具体环境（如规模、应用领域、实现技术、开发团队等）会在开发实践中产生差异，而我们目前仍然无法对其差异进行准确的刻画。软件开发实践对项目环境具有敏感性，其主要原因是软件开发具有智力密集的特征 [3]，人的因素始终是软件工程无法回避的一个关键因素。传统的定性研究对人的因素给出了很多具有深刻洞察力的结论 [3]。但是，由于缺少量化方法，这些结论很难在软件开发实践中定量实施，进而难以得到有效传播和普及。

目前，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（如版本控制系统和问题追踪系统等）的广泛应用积累了大量数据，记录了软件代码的演变 [4]、开发任务的流程等。截至 2015 年 4 月，GitHub 中的软件仓库数量超过 1600 万（如图 1 所示）。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 [5]。截至 2014 年 9 月，StackOverflow 的问答数量超过 1300 万（如图 2 所示）。

这些数据从不同阶段、不同侧面反映了软件开发和应用的历史和动态，并提供了庞大的项目样本 [6]，为软件工程研究和实践带来了新的思维模式，提出了新的研究问题，例如，如何有效地收集、组织和

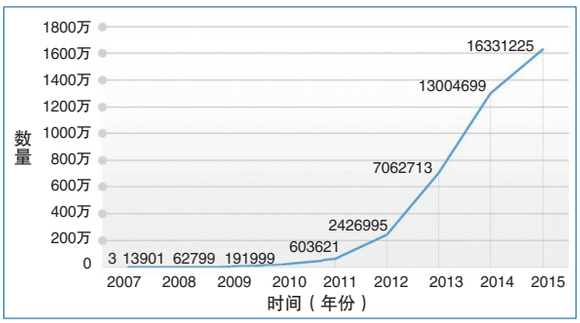

图1　GitHub的代码库/项目增长图

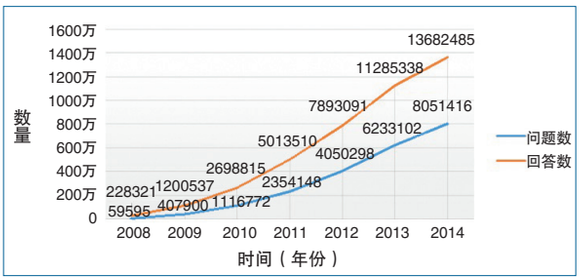

图2　StackOverflow社区的问答数量增长图

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？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具体环境下的开发活动进行观察和分析（我们将这种从微观层次度量到的开发活动模式称为微过程[7]）。这些挑战和机遇对于软件工程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，但相关的研究工作尚处于起步阶段。

数据的处理和量化分析方法

近些年来研究人员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，CVSAnaly实现了版本控制系统数据的处理[8]；SoftChange能够从源代码中推断各种隐含的事实[9]；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[10]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[11]，将迁移学习方法应用到缺陷预测问题[12]等。也有一些工作尝试建立软件开发数据的量化分析方法，例如莫卡斯(Mockus)提出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，最后验证结果并审查检查研究假设以确定是否对上述步骤进行迭代[13]。特别值得一提的是，数据质量对软件数据分析至关重要，正逐步成为一个重要研究方向。例如，我们提出了一种利用数值分布来定位和修正错误数据的方法[31]。

开源量化分析：从理解到改进和推荐

目前，基于数据的开源软件量化分析领域的研究大致分为两个阶段：首先对开源软件现象进行分析和理解；然后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。这些研究的关注点不仅在于开源软件制品本身，还涉及到开源软件开发者及其形成的开源社区。图3呈现了开源数据量化分析的基本脉络。

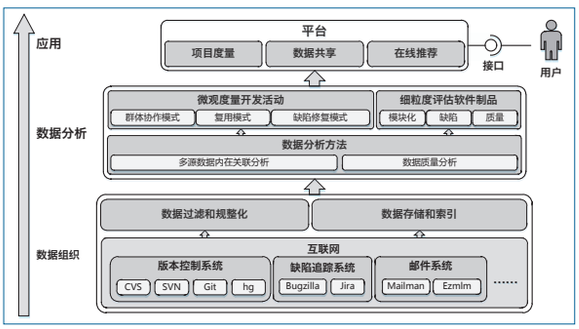

图3　开源数据量化分析的脉络图

理解开源

在理解开源方面最早的代表性工作是莫卡斯等人对Apache和Mozilla这两个开源项目的研究[1]。该研究对核心开发团队的规模、任务量以及组织方式等进行了量化分析。杰尔曼(German)遵循类似原则对Gnome社区所使用的分布式开发实践进行了量

化分析[14]。这些分析使得我们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开发高质量软件有了基本的认识。

开源软件开发如何弥合全球开发者在语言、文化、时区、习俗方面的差异也引起了很多关注。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，卡塔尔多 (Cataldo) 等人通过度量程序员的协作行为，发现许多高产的程序员会随着时间改变他们的电子通信媒介，以更好地满足协作需求[15]。

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、过程说明，甚至没有设计文档[1]；很多成功项目往往依赖于个别天才程序员[10]；许多开源项目的主要推动力来自用户的外部贡献（包括代码贡献[1]和缺陷报告[16]等）。随着 Pull-Request 等新型开发机制的出现，越来越多的外围大众参与到软件开发活动中来，使得大众开发成为当代软件开发的一个标志。

开源软件所蕴含的全新的经济理念和社会化生产方式呈现出社会、经济、组织与管理、技术、实践等方面的重要属性，吸引了社会学、经济学与管理学等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开发出高质量的软件产品？传统上，这些问题的研究更多是采用定性方法。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。

度量和改进开源实践

在开源成为一种现象之后，很多研究工作开始致力于利用开源数据从各方面帮助和改进开源开发。例如，齐默尔曼 (Zimmermann) 等人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[17]；霍尔姆斯 (Holmes) 等人通过启发式规则对正在开发的代码结构和代码库中已有的样本代码进行匹配，进而定位相关代码[18]；米哈伊尔 (Michail) 使用泛化的关联规则挖掘技术发现函数库的复用模式[19]；谢 (Xie) 等人使用频繁序列挖掘技术分析应用程序接口的调用频度和调用顺序，以帮助代码开发人员理解应用程序接口的使用方法[20]。其他工作还包括：探究开源项目中的缺陷分类活动 (bug triage)[21]、寻找重复的问题报告 (duplicate bug)[22]、预测问题被修复的概率[23]等。

还有一类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、其贡献质量如何、社区如何维护其长期贡献[16, 24~26]等多个方面。例如，我们通过度量贡献者进入开源项目后第一个月的行为，发现从统计意义上说，他们进入项目时的初始环境对其是否能够成为一个长期贡献者具有重大影响[16, 24]。这些发现能够帮助我们更好地推荐最佳实践和设计协作工具。

开源数据还为我们对软件本身进行度量和评估提供了一种有效的方法。例如，邹 (Zou) 等人从软件用户的角度提出了一种基于互联网用户评论的软件质量度量方法[6]：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、稳定等方面对软件质量进行综合评估。

开源和产业

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（黑鸭软件）在 2014 年统计，开源软件广受欢迎的十大特点包括：软件质量、

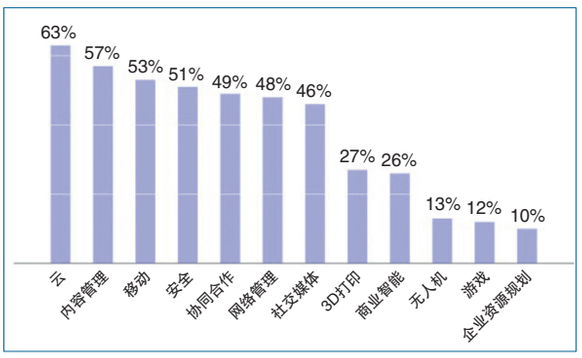

图4　开源引领的主要领域[27]

功能特性、安全性、创新速度、可扩展性、定制化、合作、标准化、前沿性、成本[27]。图4显示了黑鸭软件所调研的1240名软件从业者（软件工程师、首席执行官、IT管理者、教育者、销售等）所认为的开源引领的主要领域。

为了充分利用开源开发的优势，越来越多的公司和团体组织参与到开源运动中，设立起了“商业+开源”的混合项目。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”[28]。

然而，如何在企业中有效利用开源特性及优势并非易事，目前已经出现了一些探索性研究。丁克拉克(Dinkelacker)等人描述了惠普公司采用的渐进式开源策略[29]。该策略提供了三种开放性逐渐增强的模型：内部源码（向公司内部开放）、受控源码（在选择的伙伴间开放）、开放源码（向全世界开放）。沃格斯特伦(Wagstrom)等人度量了开源项目Gnome和Eclipse中的公司行为[30]，总结出两种模式：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；另一种是聚焦产品的公司，通过销售产品获得利润。我们利用问题追踪数据度量了三个同构的开源产品（JavaEE应用服务器）中公司的参与活动，发现公司的投入力度对外部参与者有重大影响[28]。上述策略和模式的效能在何种条件下能够更好地发挥作用，也需要采用开源量化分析手段进行深入研究。

数据和开源的启示

开源软件是人类历史上一次基于群体智慧、利用互联网实现分布式协作的重大实践。开源运动的先行者通过对早期自由软件活动的深刻认识，创造性地解决了开源软件在法律和商业方面遇到的问题，逐步建立起基于群体智慧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。这种“认识世界—改造世界”的循环发展过程还将在开源世界中持续进行。

开源软件的历史发展过程几乎被完整地记录了下来，供我们去理解、反思和改进。大量的开源软件项目在互联网上积累了海量的数据。这些数据覆盖了开源软件开发活动的各个方面。基于这些数据的量化分析，我们对开源软件的开发模式和机理有了更深入的理解，这为开源软件开发实践提供了良好的反馈，对其发展起到了很好的促进作用。开源软件开发过程中形成的数据为我们理解开源世界提供了一个广阔的渠道。数据体现了一种现实存在，其揭示的本质规律通常具有可自证的性质（当然，数据本身可能存在局限性或错误[31]）。如果一个规律适用于成千上万的项目，如同物理定律，那么我们也许就真的找到了软件工程的“银弹”，或者反之则能够证明不存在“银弹”。

开源带给我们的启示不仅体现在开发方面，还体现在更高抽象层次的理念方面。开源社区的协同共享、用户创新等理念，已经扩展到了软件行业之外（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）的萌芽与持续发展。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建设一个更加美好的信息化人类文明。■

周明辉

CCF专业会员。北京大学副教授。主要研究方向为软件数字考古学。

zhmh@pku.edu.cn

张　伟

CCF专业会员。北京大学副教授。主要研究方向为需求工程、需求复用、网络计算环境下的新型需求工程方法。

zhangw@sei.pku.edu.cn

尹 刚

CCF专业会员。国防科学技术大学副研究员。主要研究方向为分布计算与软件工程。jack_nudt@163.com

参考文献

[1] Audris Mockus, Roy T. Fielding, James D. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Methodol*. 2002; 11(3): 309~346.

[2] D. L. Parnas. 1972. On the criteria to be used in decomposing systems into modules. *Commun. ACM* 15, 12 (December 1972), 1053~1058.

[3] P. Robillard, The role of knowledge in software development, *Communications of the ACM*, 1999; 42(1): 87~92.

[4] Audris Mockus: Amassing and indexing a large sample of version control systems: Towards the census of public source code history. MSR 2009: 11~20.

[5] Bogdan Vasilescu, Vladimir Filkov, Alexander Serebrenik: StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. SocialCom 2013: 188~195.

[6] Gang Yin, Tao Wang, Huaimin Wang, et al. OSSEAN: Mining Crowd Wisdom in Open Source Communities, 2015 IEEE Symposium on Service-Oriented System Engineering (SOSE), Page(s): 367~371.

[7] Minghui Zhou and Audris Mockus. Mining micro-practices from operational data. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014)*. Nov 18-20. Hongkong. 845~848.

[8] G. Robles, S. Koch, J. M. Gonzalez-Barahona. Remote analysis and measurement of libre software systems by means of the CVSAnalY tool. RAMSS 2004.

[9] D. M. German, A. Hindle. Measuring fine-grained change in software: towards modification aware change metrics. Metrics Symposium 2005: 28~38.

[10]G. Gousios, D. Spinellis. G. H. Torrent. Github's data from a firehose. MSR 2012: 12~21.

[11]E. Kocaguneli, T. Menzies, J. Keung. On the value of ensemble effort estimation. *IEEE Transactions on Software Engineering*, 38(6), 2012: 1403~1416.

[12]N. Jaechang, J.P. Sinno, S. Kim. Transfer defect learning. ICSE 2013: 382~391.

[13]A. Mockus. Software support tools and experimental work. In V. Basili and et al, editors, Empirical Software Engineering Issues: Critical Assessments and Future Directions, volume LNCS 4336, pages 91~99. Springer, 2007.

[14]D. M. German. The gnome project: a case study of open source, global software development. *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 201~215, 2003.

[15]M. Cataldo, P. Wagstrom, J. Herbsleb, and K. Carley. Identification of coordination requirements: Implications for the design of collaboration and awareness tools. In Conference on Computer Supported Cooperative Work CSCW'06, Banff, Alberta, Canada, 2006.

[16]Minghui Zhou and Audris Mockus. Who Will Stay in the FLOSS Community? Modelling Participant's Initial Behaviour. *IEEE Transactions on Software Engineering*. vol.41, no.1, pp.82~99, Jan. 1 2015.

[17]T. Zimmermann, P. Weisgerber, S. Diehl, A. Zeller. Mining version histories to guide software changes. ICSE 2006: 563~572.

[18]R. Holmes, G. C. Murphy. Using structural context to recommend source code examples. ICSE 2005: 117~125.

[19]A. Michail. Data mining library reuse patterns using generalized association rules. ICSE 2000: 167~176.

[20]T. Xie, J. Pei. MAPO: mining API usage from open source repositories. MSR 2005: 54~57.

[21]J. Anvik, L. Hiew, G. C. Murphy. Who should fix this bug? ICSE 2006: 361~370.

[22]P. Runeson, M. Alexandersson, O. Nyholm. Detection of duplicate defect reports using natural language processing. ICSE 2007: 499~510.

[23]P. Guo, T. Zimmermann, N. Nagappan, B. Murphy. Characterizing and predicting which bugs get fixed: an empirical study of Microsoft windows. ICSE 2010: 495~504.

[24]Minghui Zhou, Audris Mockus, Does the Initial Environment Impact the Future of Developers? the 33rd International Conference on Software Engineering (ICSE 2011), 2011/5/21-2011/5/28, pp 271~280, 2011/5/21.

[25]Jialiang Xie, Minghui Zhou, Audris Mockus. Impact of Triage: a Study of Mozilla and Gnome. ESEM 2013 (Empirical Software Engineering and Measurement). Oct 7~11. Baltimore, Maryland, USA.247~250.

[26]Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative

Work & Social Computing (CSCW '15). ACM, New York, NY, USA, 1379~1392.

[27]Black Duck. 2014. www.slideshare.net/blackducksoftware/2014-future-of-open-source-survey-results.

[28] Xiujuan Ma, Minghui Zhou, Dirk Riehle. How commercial involvement affects open source projects: three case studies on issue reporting. SCIENCE CHINA Information Sciences. August 2013, Vol. 56:153~165.

[29] Dinkelacker, J., Garg, P. K., Miller, R., & Nelson, D. Progressive open source. In Proceedings of the 24th International Conference on Software Engineering (pp. 177~184). ACM. 2002, May.

[30] Patric Wagstrom, James Herbsleb, Robert Kraut, and Audris Mockus. 2010. The Impact of Commercial Organizations on Volunteer Participation in an Online Community. In Academy of Management Annual Meeting. Montreal, CA.

[31]Qimu Zheng, Audris Mockus, Minghui Zhou. A method to identify and correct problematic software activity data: exploiting capacity constraints and data redundancies. ESEC/SIGSOFT FSE 2015: 637~648.