

Open source software analytics开源软件数据分析组

研究方向介绍

周明辉 等

北京大学软件工程研究所

2020 年 1 月

# **目录**

1. [研究领域 2](#_bookmark0)
2. [研究目标 2](#_bookmark1)
3. [研究范围 2](#_bookmark2)
4. 研究方法 [2](#_bookmark3)
   1. 常用的挖掘和分析技术 [2](#_bookmark3)
   2. [数据来源 3](#_bookmark5)
5. [相关资源 3](#_bookmark6)
   1. [数据集 3](#_bookmark7)
   2. [开源相关的资源 5](#_bookmark8)
   3. [数据处理脚本 6](#_bookmark9)
6. [推荐文献](#_bookmark10) 6

[参考文献 7](#_bookmark11)

# **研究领域**

|  |  |
| --- | --- |
| 一级学科：软件工程(software engineering) | |
| 细分领域关键词 | 数据驱动的软件开发(data-driven software development) |
| 软件仓库挖掘(mining software repositories) |
| 软件度量(software measurement) |
| 实证/经验软件工程(empirical software engineering) |
| 软件数字考古学(software digital archeology) |
| 软件数字社会学(software digital sociology) |

# **研究目标**

* + 观察和度量大规模软件项目或生态中人们的开发行为，量化开发者与其环境和软件制品之间的关系，以期理解和掌握控制大型复杂软件系统的方法。
  + 研究和量化影响个体/项目/社区成功的因素，建立推荐工具，对软件开发各种实践活动进行预测和推荐。[1](#_bookmark4)

# **研究范围**

但凡软件工程领域，任何研究问题都需要与提高开发效率和软件质量相关，即相关性（relevant）。另外需要具有的特点：

前沿性(novel)：选题应聚焦于有新意的问题，进行创新和深入的研究。

普遍性(non-trivial)：问题和解决方案需要有一般性，不能只针对极个别群体。

# **研究方法**

## 常用的挖掘和分析技术

* + - 原始数据处理：shell scripts, perl, python
    - 数据分析：R language, machine learning

• 定性分析：interviews, surveys, grounded theory

1鉴于开源项目的广泛存在及其数据的可获取性，我们的研究对象大多是开源项目及社区。我们面向产业项目

大多做的是咨询工作，企业一般不会允许公开其数据，但产业咨询可以丰富我们的洞察力并应用我们的研究结果。

## 数据来源

目前主要是利用软件开发活动数据，例如由下述软件开发支持工具产生的数据：

* + - Version control system: cvs, svn, git, hg
    - Issue tracking system: jira, bugzilla
    - Email archives
    - Forums

# **相关资源**

## 数据集

互联网上成千上万开源项目的数据都是开放的，可获取的软件开发活动数据非常多。一般来说，开源项目都会支持在线问题追踪系统，版本控制系统和邮件列表等，例如：

* + - Mozilla 的在线issue tracking system：

<https://bugzilla.mozilla.org/show_bug.cgi?id=1000000> <https://bugzilla.mozilla.org/show_bug.cgi?id=14500001>

* + - Linux kernel 的在线代码库(mainline code repository)：

<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>

* + - Linux kernel 的在线邮件列表(mailing-list)、<https://lkml.org/>

我们通过下面的例子，来介绍如何对来自上述资源库的数据进行分析和研究：

**示例 5.1 Issue tracking data**, 即项目中的 issue 报告和处理流程数据。

我们有两份由 mozilla 社区官方提供的 Mozilla Bugzilla dump *(*分别在 2013 年和 2016年提供*)*，原始数据如下：

pae:/store1/bug/MozDump/Mozilla-Bugzilla-Public-02-January-2013.sql

pae:/store/bug//mozilladump20160304/research\_bugzilla\_2016-03-04.sql.gz

下载地址：<https://zenodo.org/record/971162#.Wcb12BfadCU>

* 有关数据本身的处理流程，可以参见文献*"*[*Multi-extract and multi-level dataset of mozilla*](#_bookmark12)[*issue tracking history".*](#_bookmark12)
* 有关如何使用这个数据集做研究，可以参见文献[*"Who Will Stay in the FLOSS*](#_bookmark13) *C*[*ommunity?Modeling Participant’s Initial Behavior".*](#_bookmark13)

**示例 5.2 Commits data**, 即项目开发过程中的代码提交日志。

所有开源项目都可以先 clone 到本地然后获取其 commit log。下面以 linux kernel 为例描述日志获取过程：

***while*** *(< STDIN >){ chop ();*

*#* ***catch*** *end of last revision information*

***if*** *(/^ STARTOFTHECOMMIT$/){*

*<→ $ctime\;$f\;$comment\n";*

*% paths = ();*

*($rev, $aname, $alogin , $atime, $cname, $clogin , $ctime, $comment) = ("",*

*<→ "","","","","","","","","");}*

*}*

*sub output {*

*foreach my $f (keys % paths){*

*$comment =~ s/\ r/ /g;*

*$comment =~ s/\;/ SEMICOLON/g;*

*$comment =~ s/\ n/\_\_NEWLINE\_\_/g;*

*print "$rev\;$aname\;$cname\;$alogin\;$clogin\;$paths{$f}\;$atime\;*

*<→ ","","","","","","","","");*

*my ($getHeader, $getPaths) = (0, 0);*

*######## extrgit.perl#######################*

*use strict;*

*# Extract each revision from FreeBSD - release/10.2.0 log output. -zhouminghui use Time:: Local;*

*my % paths = ();*

*my ($rev, $aname, $alogin , $atime, $cname, $clogin , $ctime, $comment) = ("","*

*<→ \;$clogin\;$nadd\;$at\;$ct\;$f\;$cmt"';*

*done > linux.l2*

***do*** *perl -ane 'use Time:: ParseDate qw( parsedate); ($rev,$aname,$cname,$alogin ,*

*<→ $clogin ,$nadd,$atime,$ctime,$f,$cmt)= split(/\;/,$\_,-1); $at= parsedate("*

*<→ $atime");$ct= parsedate("$ctime"); print "$rev\;$aname\;$cname\;$alogin*

*<→ linux$perl extrgit.perl <log.linux >linux.l1*

*linux$cat linux.l1 |* ***while*** *read*

*$git clone git:// git.kernel.org/pub/scm/linux/kernel/git/ torvalds/linux.git*

*$cd linux*

*linux$git log --numstat*

*--pretty= format:" STARTOFTHECOMMIT%n%H;% an;% ae;% ad;% cn;% ce;% cd;% s" > log.linux*

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

*& output ();*

*$getHeader=1;$getPaths=0; next;*

*}*

*# process file header*

***if*** *($getHeader){*

*$\_ =~ s/ \| /\|/g;*

*($rev, $aname, $alogin , $atime, $cname, $clogin , $ctime, $comment) =*

*$getHeader = 0;*

*$getPaths = 1; next;*

*<→ split(/\;/, $\_, -1);*

*# print STDERR " getHeader $rev, $login , $date, $line\n";*

*}*

***if*** *($getPaths && /^$/){ next;*

*}*

***if*** *($getPaths && /^[0 -9]/){*

*/(\ d+)\s+(\ d+)\s+(.\*)$/;*

*my ($nadd, $ndel, $path) = ($1 , $2 , $3);*

*$paths{$path}="$nadd:$ndel";*

*}*

*}*

*& output (); #########################################################*

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

**示例 5.3** Linux kernel的 **patch data**.

数据处理方法参考论文*”*[*A Multi-level Dataset of Linux Kernel Patchwork”.*](#_bookmark14)

项目链接：<https://zenodo.org/record/1165576#.WsGWPkxuKUk>

## 开源相关的资源

* + - 网址：<https://www.ncsu.edu/it/open_source/>(北卡罗来纳大学)

介绍：分门别类地索引了开源操作系统、开源应用软件、开源中间件、开源硬件、开源课程、开源出版等资源，为学生、科研与教育工作者们提供了一个利用开源的入口。

* + - 网址：<https://schoolforge.net/>(Schoolforge 联盟)

介绍：Schoolforge联盟是一个旨在联合开发、推广开源教育资源的组织的联合体，其所搭建的schoolforge.net平台集合了面向教育的各类开源软件、开源课程、开源教材等资源，为开源教育提供了丰富资源。

* + - 网址：Datahouse.trustie.org(软件数据分析组)

介绍：我们提供数据资源的一个平台，以后还会陆续放置相关的开源资源和服务（该url目前不可用，还在寻找可用url中）。

## 数据处理脚本

我们对于数据的前期处理基本采用相同的R语言和Shell脚本。

Git 数据的前期处理可参见：<https://github.com/minghuizhou/maintainerAnalysis>

Issue data 的前期处理可参见：<https://zenodo.org/record/971162#.Wcb12BfadCU>

# **推荐文献**

|  |  |  |
| --- | --- | --- |
| 推荐文献 | 文章名称 | 作者 |
| 领域科普文章 | [基于大数据的软件工程新思维](#_bookmark15) | [周明辉, 郭长国](#_bookmark15) |
| [开源软件的量化分析](#_bookmark16) | [周明辉, 张伟等](#_bookmark16) |
| 领域早期工作 | [Two case studies of open source software](#_bookmark17)  [development: Apache and Mozilla](#_bookmark17) | [A. Mockus,](#_bookmark17)  [R. T. Fielding, et al.](#_bookmark17) |
| [An empirical study of speed and communication](#_bookmark18)  [in globally-distributed software development](#_bookmark18) | [J. D. Herbsleb](#_bookmark18)  [and A. Mockus](#_bookmark18) |
| [Globalization by chunking: a quantitative](#_bookmark19)  [approach](#_bookmark19) | [A. Mockus and](#_bookmark19)  [D. M. Weiss](#_bookmark19) |
| 组内工作  (程序员、制品及环境的关系) | [Developer fluency: Achieving true mastery in](#_bookmark20)  [software projects](#_bookmark20) | [M. Zhou and A. Mockus](#_bookmark20) |
| [What make long term contributors: willingness](#_bookmark21)  [and opportunity in OSS community](#_bookmark21) | [M. Zhou and A. Mockus](#_bookmark21) |
| [Impact of Triage: a Study of Mozilla and Gnome](#_bookmark22) | [J. Xie, M. Zhou, et al.](#_bookmark22) |
| [On the scalability of Linux kernel](#_bookmark23)  [maintainers’ work](#_bookmark23) | [M. Zhou, Q. Chen, et al.](#_bookmark23) |
| 组内工作  （代码研究） | 大规模开源代码复用检测及度量工具的  [设计与实现](#_bookmark24) | [陈峰宏](#_bookmark24) |
| [A Neural Framework for Retrieval](#_bookmark24)  [and Summarization of Source Code](#_bookmark24) | [Q. Chen and M. Zhou](#_bookmark24) |

**注解**

1. 领域前期工作:使用数据探索开源开发和全球分布式开发是本领域的开拓性工作(也是读起来有美感的论文)，需要尤其关注其研究问题和研究方法及其跟软件开发效率和质量的关系。
2. 组内工作(程序员、制品及环境的关系):我们组目前的代表工作在于研究程序员、制品及环境的关系(跟组长个人意趣有关)。研究对象从个体(成熟度、学习途径、环境影响)到群体(项目最佳实践)，到生态(可持续性、可演化性)。
3. 组内工作(代码研究):我们组另外一部分工作侧重于代码研究，涉及代码克隆、代码搜索（代码与自然语言的转换）、代码生成等。

## **参考文献**

1. Jiaxin Zhu, Minghui Zhou, and Hong Mei. Multi-extract and multi-level dataset of mozilla issue tracking history. In Proceedings of the 13th International Conference on Mining Software Repositories (MSR ’16). ACM, New York, NY, USA, 472-475.
2. Minghui Zhou, Audris Mockus: Who Will Stay in the FLOSS Community? Modeling Participant’s Initial Behavior. IEEE Transactions on Software Engineering, vol.41, no.1, pp.82-99, Jan. 1 2015.
3. Yulin Xu and Minghui Zhou. A Multi-level Dataset of Linux Kernel Patchwork. In MSR ’18: MSR ’18: 15th International Conference on Mining Software Repositories , May 28–29, 2018, Gothenburg, Sweden. ACM, New York, NY, USA.
4. 周明辉, 郭长国. 基于大数据的软件工程新思维. 中国计算机学会通讯. 第10卷, 第3期. 2014年3月.
5. 周明辉, 张伟, 尹刚. 开源软件的量化分析. 中国计算机学会通讯. 第12卷, 第2期. 2016年2月.
6. A. Mockus, R. T. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 11(3):1–38, July 2002.
7. J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in
8. globally-distributed software development. IEEE Transactions on Software Engineering, 29(6):481–494, June 2003.
9. A. Mockus and D. M. Weiss. Globalization by chunking: a quantitative approach. IEEE Software, 18(2):30–37, March 2001.
10. M. Zhou and A. Mockus. Developer fluency: Achieving true mastery in software projects. In ACMSIGSOFT/FSE, Nov 7-11, 2010, pp137-146.
11. Minghui Zhou, Audris Mockus. What make long term contributors: willingness and op- portunity in OSS community, 2012 International Conference on Software Engineering (ICSE 2012), 2012/6/2-2012/6/9, pp518-528.
12. 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 2013. pp247-250.
13. Minghui Zhou, Qingying Chen, Audris Mockus, and Fengguang Wu. On the scalability of Linux kernel maintainers’ work. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). pp27-37.
14. 陈峰宏. 大规模开源代码复用检测及度量工具的设计与实现. 北大硕士论文. 2013
15. Qingying Chen and Minghui Zhou. A neural framework for retrieval and summarization of source code. The 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2018). pp826-831.