2013 年第三届智能系统设计与工程应用国际会议

利用可视化技术挖掘开源软件中开发者的贡献

徐贲、沈北军、杨伟成

上海交通大学软件工程学院，中国 shaci1234@yahoo.com.cn，上海，200240

摘要——开发人员贡献的研究是软件进化领域的重要组成部分。它允许项目所有者更早地找到潜在的长期贡献者，并帮助新来者改善他们的行为。本文基于视觉分析，考察了开源环境中开发人员的贡献特征，并从影响因素、时间特征和区域特征三个方面提出了方法。我们的分析使用了 github 的数据，揭示了一些规律。我们发现，新来者开始与更多人一起参与的代码在某种程度上会导致贡献减少。我们还发现，开发人员的早期贡献和后期贡献之间存在关系。此外，来自不同地区的开发商更有可能拥有主导关系。我们的发现可能为软件进化领域的未来研究提供一些支持。

关键词——开源软件；软件进化；视觉分析；贡献特性

一.导言

软件进化的研究经历了从商业软件到开源软件的转变。从 1972 年到 2002 年，雷曼兄弟和他的团队根据商业研究提出了八条软件进化定律

软件[1]。但是最近的研究表明，这些法律没有得到遵守，特别是在开源软件中

环境[2]。一些研究人员试图通过新的视角改进软件进化的理论模型。例如，姜维·吴已经应用了自我

软件进化的有组织临界性[3]。

除此之外，更多的努力被投入到软件质量、生产力和开发者贡献的研究中。在这三项中，前两项在数量上比后一项具有压倒性优势；同时，开发者贡献研究主要集中在贡献的度量上。

在本文中，我们试图将数据挖掘和视觉分析技术相结合，从全新的角度研究开发者的贡献特征。本文的研究问题和主要贡献如下:

RQ1。与代码相关的哪些因素会影响开发人员的贡献?

本研究采用平行坐标。我们发现，新来者开始与更多人一起参与的代码将在某种程度上减少贡献。

RQ2。投稿人提交的时间分布有什么模式吗?

根据最长的不提交间隔，贡献分为早期和后期。我们发现开发人员的早期贡献和后期贡献之间有关系。

RQ3。开发商的贡献和他们所在的地区有什么关系吗?

本研究采用了优势图，并添加了区域信息。我们发现来自不同地区的开发者更有可能拥有主导关系。

我们的发现一方面让项目所有者能够更早地找到潜在的长期贡献者并给予更多关注，另一方面帮助新来者改善他们的行为并增加他们对项目的贡献。

二.相关著作

在过去的几十年里，许多研究者进行了相关的研究。

北京大学的周明辉提议通过个人的能力、意愿和机会来模拟个人成为有价值的贡献者的机会

在加入[时捐款 4]。利用 Mozilla 和 Gnome 的问题跟踪数据，他们发现新来的人成为长期贡献者的概率与她最初的贡献类型、错误报告风格、是否成功修复了至少一个报告的问题、同行群体等等相关。

[5 号、[6 号、[7 号)使用回归分析方法研究了地理分布对软件生产率和质量的影响。在[7]开发商之间的合作关系也包括在内，实际上他们之间的距离是计算出来的。有趣的是，在分析地理对软件质量的影响时，[和[得出了相反的结论。

进化矩阵[8]，进化分光镜[9)，

时间轴[10]是基于矩阵的可视化，x 轴代表时间，y 轴由软件组件组成，不同的度量可以应用于项目值。

冰柱图是树的变体，结构更紧凑，因此可以在二维中显示多个版本

飞机。它被用于代码流[11]，分级边缘

捆绑[12]来可视化代码结构的演变。

此外，还有基于城市的可视化[10] [13]，如

以及使用动画[14]。

在开发人员活动演变的主题中，小川提出了各种可视化技术，包括

978 - 0 - 7695 - 4923 - 1 / 12 $ 26.00 2012 IEEE DOI 10.1109 / ISDEA 2012.223

故事情节[15)，代码群[16]和星际之门[17]。星际之门也关注代码结构，但是它是静态的，没有考虑进化。

除了周明辉的研究，上述所有研究都没有关注开发者的贡献特征。虽然周明辉的研究主要使用回归分析，但是在本文中，我们借助视觉分析从三个新的角度进行探索。

三.方法学

缴款措施

传统上，最经典、最简单的贡献度量是代码行(LOC)。Gousios 等人提出了一个关于代码和非代码的更复杂的系统

贡献[18]。在我们的研究中，我们只考虑 LOC 和提交次数。第一个开发者的贡献计算如下:

镍

九社区

iCommit

KLOC

iLOC 图标图标

k

无

= +≤≤≤

- =

- =) ( ) (( ) ( ) ( ) (( ) (11

n 是开发人员的总数。

影响因素的研究

我们从开发人员第一次提交的代码中考察了他们贡献的影响因素。我们研究了代码不稳定性、代码结构以及提交特性。不稳定性是用新人第一次承诺(HCC)前一个月别人修改的次数来衡量的。
FanIn、fanOut、LOC、属性数(NOA)和方法数(NOM)用于测量代码结构。提交功能包括 LOC 更改(FCLOC)和第一次提交中的新文件数(NFC)。

我们可以很容易地从提交历史中获得 HCC！FCLOC 和 NFC。至于代码结构，我们通过 git 命令在开发人员初始提交之前和之后导出代码，然后用穆斯工具解析它们。穆斯是一种众所周知的静态分析工具，我们可以从它直接获得所需的度量。

我们将回归分析和视觉分析相结合。在回归分析中，我们使用 Matlab 工具进行一元和多元线性回归。我们在视觉分析中使用平行坐标。平行坐标是可视化高维几何的常用方法。

时间特性的研究

在本节中，我们创建了条形码可视化。它由几条长度相等的垂直平行线段组成。这些片段根据时间从左到右依次排列。线段表示某一天的一个或多个提交。相对数量由线条颜色表示，颜色越深表示提交越多。两条线段之间的间隙是没有提交的期间。

在本文中，条形码可视化被用来研究开发者的惰性期。惰性期被定义为开发人员提交历史的最大间隔。我们将每个开发人员的提交历史可视化为条形码图

然后将整个项目的条形码放在下面，观察两者的一致性。我们还根据惰性期将开发者的贡献分为早期和后期，并进一步研究了它们之间的关系。

我们用回归分析从数学上验证了结果。我们计算了惰性期的长度和这段时间内其他人的提交次数，然后分别与开发者的贡献进行线性回归。同时，我们也使用回归分析来找出早期和后期贡献之间的关系。

区域特征的研究

优势图用于区域特征的研究。这是一个简单的有向图，顶点代表贡献者，边代表某些文件上的主导关系，箭头指向主导者。文件的支配者是对该文件贡献最大的人。当开发者修改一个文件时，他和支配者之间就会产生支配关系。主导者会随着时间的推移而改变，但是主导关系仍然存在，直到开发人员重新对文件做出贡献。我们改进了使用颜色来表示贡献者的区域，并根据贡献者和支配者是否来自同一个区域来可视化两个主导图。

出于各种原因，大多数开发人员没有直接提供他们的区域信息，所以我们使用了各种技术来预测。我们参考了[6]中提到的方法，通过电子邮件分析域名、时区、公司信息，并在网站以及谷歌、LinkedIn、Twitter 上搜索。我们使用国家作为地区单位，因为一方面，国家信息比特定的城市信息更容易收集；另一方面，民族不仅反映了空间距离，也反映了语言和文化的差异。

四.个案研究

数据集

我们从 github 上托管的开源项目 CraftBukkit 收集数据。随着 Git 版本控制系统的广泛应用，越来越多的项目被移植到 github。手工工具包是《我的世界》服务器模块的 API 实现，使用 Java 语言开发。这是一个相对年轻的项目，从 2010 年 12 月开始。我们收集了 2010 年 12 月至 2012 年 7 月期间生成的提交和开发者信息。

影响因素的结果

代码维度的多个度量与贡献之间的回归结果不令人满意(表 1)。从表中可以看出，最大 R 平方出现在新文件数量(NFC)和贡献之间的关系中，大约为 0.3。

表 1 单变量线性回归结果

度量线性系数

HCC - 0.0005 - 0.0115

935

N		
扇入		0.0226
扇出		0.0319
美		0.0104
N		0.0297

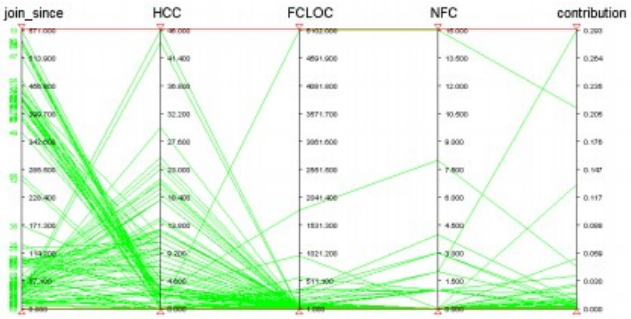

图 1。代码维度上度量的并行坐标

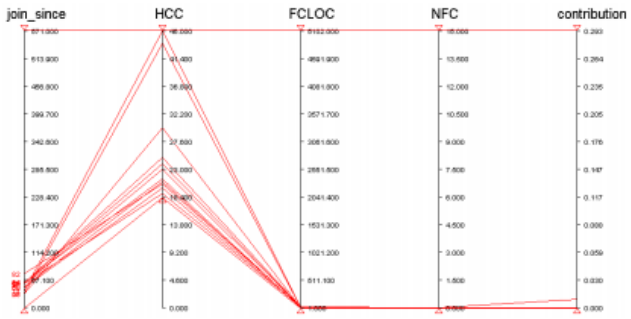

图 2。具有 HCC 下限的平行坐标

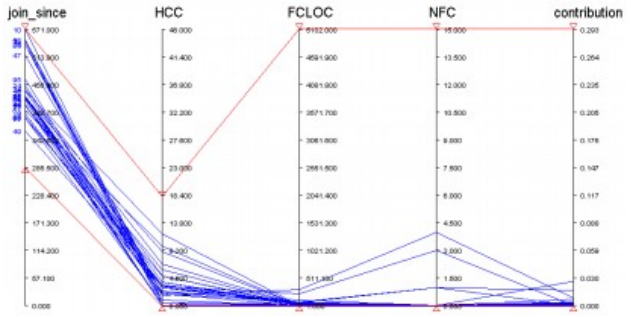

图 3。连接下限为的平行坐标

我们在相同的数据中应用了平行坐标可视化(图 1)。在图 1 中，join _ from 表示自项目首次提交以来开发人员的加入时间。通过调整 HCC 的下限，我们发现 HCC 较高的开发者贡献较小(图 2)。相反，这意味着贡献较大的开发者出现在较低的 HCC 部分。因为 HCC 在首次提交前一个月测量修改的数量，而不是自创建以来测量修改的数量，所以在某种程度上避免了加入时间的干扰。当然，为了更全面地描述这个问题，我们调整了连接的下限，因为在 HCC 较低的开发人员中(图 3)。结果显示，他们的贡献水平仍然高于高肝癌组。通常我们认为修饰的数量代表

代码的不稳定性。不稳定的代码将为开发人员提供更多的贡献机会。但是另一方面，随着更多的人参与进来，个人的平均贡献将会减少。

法律 1 :开发者的贡献受到他或她开始贡献的代码的不稳定性的影响。更多开发人员参与的代码将在某种程度上减少贡献。

时间特性的结果

条形码可视化的结果如图 4 所示。我们只截取了一部分，因为整个图太大了。开发人员条形码和项目条形码的一致性似乎不会影响开发人员的贡献。进一步的线性回归分析证实了我们的想法。此外，可以发现大多数开发人员的早期和后期贡献没有太大差异，线性回归的 R 平方出人意料地达到了 0.7488。我们认为这是一项具有重大意义的发现，因为我们可以通过找到迄今为止的惰性期来预测未来的贡献。

图 4。条形码可视化的一部分

法则 2 :开发者贡献的早期和后期之间存在线性关系。

区域特征的结果

我们分析了工艺品套件的区域组成，并在表 2 中列出了结果。如果不考虑 29 个开发人员，他们的地区仍然未知，剩下的 67 个来自 11 个国家。其中美国占据了 32 家开发商的大多数。显然，他们大多数是欧洲和美国国家。来自英国的正餐和来自荷兰的格鲁姆做出了最大的贡献。

Table 2 region composition of CraftBukkit

region	Developers count
USA	32
UK	7
Germany	6
Canada	4
Netherlands	4
Sweden	3
Australia	2
Estonia	1
Pakistan	1

936

奥地利 1

比利时 1

未知 29

手工艺包里有 40 个主宰。与上述地区构成相比，尽管美国在开发者数量上有优势，但它不是主要的控制国。此外，在分别观察相同区域和跨区域的主导关系图(图 6 - 7)后，可以发现来自不同国家的开发者拥有复杂得多的主导关系。以来自美国的 sk89q 为例，他有八个跨地区的主导关系，而相同地区的图表的相应数量只有一个。我们认为这也可以解释为异国情调。

法律 3 :来自不同地区的开发商更有可能拥有主导关系。

图 5。同区域优势图

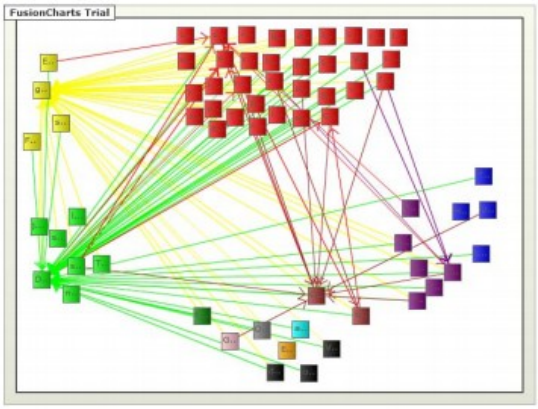

Figure 6. the cross-region dominant diagram

五.结论

在这篇论文文中，我们基于视觉分析技术考察了开发人员对 Cryst 巴克项目的贡献特征。我们从代码的角度研究了开发人员贡献的影响因素，发现新加入的人越多，贡献越少；我们将条形码可视化应用于贡献的时间特征，发现开发者的早期贡献和后期贡献之间存在关系；我们研究了贡献的区域特征

使用优势图，发现不同地区的开发者更有可能拥有优势关系。

关于未来的研究，我们希望对各种开源软件进行更多的案例研究。此外，我们想知道在商业软件领域是否有相似或截然相反的结论。

参考

[1]雷曼兄弟公司，《程序、生命周期和软件进化定律》。IEEE 会议记录，第 68 卷第 9 期，1060 - 1078，1980

[2]迈克尔·W·戈弗雷，强图。开源软件的演变:一个案例研究。2000 年国际软件维护会议记录

[3]姜维·吴，理查德·C·霍尔特，艾哈迈德·哈桑。软件进化中 SOC 动态的经验证据。软件维护，2007 年

[4]周明辉，奥德里斯·莫克斯。长期贡献者是什么:开放源码软件社区的意愿和机会。ICSE'12，第 518 - 528 页

[5]纳拉扬·拉马苏布、马塞洛·加泰罗尼亚、拉杰什·克里希纳·平衡器、詹姆斯·赫布莱布。配置全球软件团队:对项目生产率、质量和利润的多公司分析。ICSE'11，第 261 - 270 页

[6]基督教之鸟，纳齐帕邦·那加帕邦。谁？哪里？什么？检查两个大型开源项目中的分布式开发。MSR 2012，瑞士苏黎世

[7]迪奥米迪斯·斯宾利斯。FreeBSD 项目中的全球软件开发。2006 年从业人员全球软件开发国际研讨会记录，第 73 - 79 页

[8]瑞士伯尔尼的米歇尔·兰萨大学。进化矩阵:使用软件可视化技术恢复软件进化。IWPSE ' 01 第四届软件进化原则国际研讨会记录，2001 年。

[9]吴经纬，理查德·霍尔特，艾哈迈德·哈桑。利用分光镜探索软件进化。第十一届逆向工程工作会议，2004 年。

[10]理查德·维特，米歇尔·兰萨。大规模系统进化的视觉探索。WCRE，2008 年。

[11]亚历山大·泰拉，大卫·奥伯。代码流:可视化源代码的结构演变。欧洲图形学/IEEE-VGTC 可视化研讨会，2008 年。

[12]丹尼·霍尔顿，杰瑞克·范·维克。分层组织数据的视觉比较。计算机图形论坛，2008 年第 27 卷。

[13]弗兰克·史坦布吕克纳，克劳斯·列文茨。代表软件城市的发展历史。SOFTVIS，2008 年。

[14]亚伯兰·辛德勒，蒋珍明，瓦利德·科尔莱特，迈克尔·W·戈弗雷，理查德·c·霍尔特·纱:动画软件进化。可视化理解和分析软件，2007 年。

[15]迈克尔·小川，关宁-刘妈。软件进化故事情节。SoftVis，2010 年。

[16]迈克尔·小川，关宁-刘妈。代码群:有机软件可视化的设计研究。SOFTVIS，2008 年。

[17]迈克尔·小川，关宁-刘妈。星际之门:软件项目的统一交互式可视化。可视化研讨会，2008 年。

[18]乔治奥斯·古西奥斯、伊里尼·卡尔利亚姆瓦库、迪奥米迪斯·斯宾利斯。测量软件储存库数据中开发人员的贡献。采矿软件储存库，2008 年，第 129 - 132 页。