

# 行为学软件度量的研究

罗光春, 孙殿余, 赖晓龙, 李 炯

(电子科技大学计算机科学与工程学院 成都 610054)

**【摘要】**针对目前软件度量方法的工作量大、度量困难问题,以行为学研究和软件度量为依据,提出了操作简单、度量简单的行为学度量方法来进行软件进度度量。通过实践项目的数据统计,发现改进后的软件进度度量方法有效地降低了软件度量的工作量,减轻了工程管理人员进行软件度量的工作压力。同时偏差分析也有效地识别出了软件度量中的虚假数据。新的软件度量方法提高了项目管理的效率。

**关键词** 行为学研究; 虚假数据; 项目管理; 软件度量

中图分类号 TP311.5

文献标识码 A

doi:10.3969/j.issn.1001-0548.2009.03.025

## Research of Software Measurement on Behavior

LUO Guang-chun, SUN Dian-yu, LAI Xiao-Long, and LI Jiong

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

**Abstract** In order to reduce the burden of measurement, a new simple and convenient software measurement method on behavioral research and software measurement research is given in this paper. The data of practical projects are analyzed. Results show that the presented method can effectively reduce the workload of software measurement and alleviate the press of software measurement technicians. At the same time, the deviation analysis can identify fake data in software measurement.

**Key words** behavioral research; fake data; project management; software measurement

软件度量学在20世纪90年代获得了空前的发展,成为软件工程方面研究的热点方向之一<sup>[1]</sup>。本文针对当前软件度量方法的工作量大、度量困难,实现了统计击键和代码量的方式来进行软件度量的方法。

### 1 软件度量概述

当前软件度量的主要研究领域<sup>[2-3]</sup>有度量的理论基础研究、软件产品度量、软件过程度量等。它们又可以分为两类:一类为帮助编制成功的项目计划的度量体系;另一类为确保项目成功,协助项目日常管理的度量体系<sup>[4]</sup>。

软件度量到目前为止取得了一定的成功,但是软件度量的问题仍依然存在<sup>[5]</sup>,所以传统的度量往往由于度量计划目标不明确、度量定义不合理等情况而出现各种问题<sup>[6-7]</sup>:

- (1) 度量的定义不精确。
- (2) 度量了不正确的事物或属性。
- (3) 度量得太多,太频繁。

### 2 软件度量的理论基础

依据概率论的大数定理可以计算出一个软件度量周期的时间。大数定理公式如下:

$$\lim_{n \rightarrow \infty} p \left( \left| \frac{u_n}{n} - p \right| < \varepsilon \right) = 1 \quad (1)$$

用符号 $P_m$ 表示软件度量周期, $D_i$ 表示一周时间, $D_w$ 表示一周的工作日,这样式(1)里的 $n$ 可以用软件度量周期表示为:

$$n = P_m \times \frac{D_w}{D_i} \quad (2)$$

假如规定管理人员要有95%的把握保证编程人员5天工作日有1天会出现报告数据和软件统计数据不一致的情况的概率不大于5%,那么由大数定律可得到:

$$p \left( \left| \frac{u_n}{n} - p \right| < \varepsilon \right) \geq 1 - \frac{p(1-p)}{n\varepsilon^2} = 95\% \quad (3)$$

由式(3)得到软件度量周期 $P_m=95$  d。

假设由于使用了基于行为学的软件度量方法,

编程人员报告数据和统计数据不一致的概率 $E_x$ 有所下降,异常情况和度量周期的关系如表1所示。

表1 异常情况和度量周期的关系

$E_x$	1/5	1/6	1/7	1/8
$P_m/d$	95	82	73	64

由表中可以看出采用了新的度量方法会使软件度量周期减少,周期的减少会使得对项目进度的把握更加准确<sup>[8-10]</sup>。

### 3 项目数据度量的改进

为了避免上述软件度量中的部分问题,本文提出一种基于程序员行为学的项目跟踪的方法。该方法分为3个部分:(1)记录程序员击键行为。(2)监控文件系统变化的程序。(3)记录工作任务。下面将分别给予介绍。

#### 3.1 记录程序员击键行为的程序

核心代码在于拦截键盘输入,在user32.dll中,WindowsAPI包含了3个方法来实现该目的:(1)SetWindows-HookEx,负责建立键盘钩子。(2)Unhook-WindowsHookEx,负责移去键盘钩子。(3)CallNext-HookEx,负责把击键信息传递到下一个监听键盘事件的应用程序。创建一个能够拦截键盘的应用程序的关键是,实现前面两个方法,而“放弃”第三个。结果是,任何击键都只能传递到这个应用程序中。其流程如图1所示。

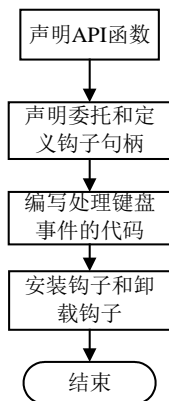


图1 击键程序的流程图

通过图1的程序可以得到项目组成员击键的记录和统计。

根据图2可以看到,一天工作时间内,10:00~12:00点、14:00~18:00点是正常工作,击键次数1800次左右;9:00~10:00点、13:00~14:00点击键次数较少,原因可能是刚刚开始工作;而12:00~13:00点则是午休时间,所以击键次数较少。

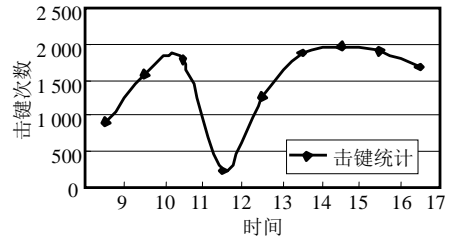


图2 某程序员一天的击键行为统计

#### 3.2 监控文件系统变化

该程序是用于统计程序员编码工作的规模的参考。实现该功能首先在系统中制定要监视的文件夹,并注册文件监视器。其核心代码如下:

```

///<summary>
/// 注册文件删除事件
///</summary>
this.fileWatcher.Deleted +=
new FileSystemEventHandler(this.OnDelete);
///<summary>
/// 注册文件创建事件
///</summary>
this.fileWatcher.Created +=
new FileSystemEventHandler(this.OnCreated);
//查找路径文件
foreach(FileInfo tempfile in Dir.GetFiles(searchPath))
{
//得到文件信息
FileInfo fileInfo =new FileInfo(filePath);
//得到文件大小
int len=fileInfo.Length;
}
    
```

图3为程序统计的某工程师一周内每天撰写的代码量的统计,代码大小的单位是Byte。可以看出该工程师在周一的工作量为2 968 Byte代码的工作量,周二为2 718 Byte代码的工作量,而周五只有1 023 Byte代码的工作量。

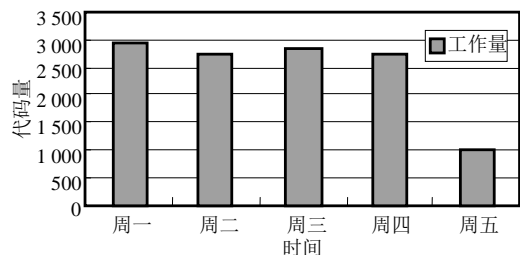


图3 代码量变化统计

#### 3.3 记录工作任务的程序

图4为工作量统计程序界面。该程序自动记录每项工作的工作时间,在系统运行时自动运行。程序

员在执行某项任务时, 首先在选择框中选择当前正在进行的任务, 然后点击计时, 系统会自动开始计算该任务的执行时间。如果程序员切换为另一项任务, 可以在下拉框中选择另一项任务名称, 系统会自动切换到该任务进行计时。

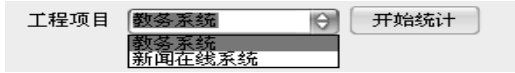


图4 工作量统计程序界面

### 4 软件度量方法的比较

本文对使用和不使用行为学工具的项目进行了比较, 使用上述行为学工具的项目, 在软件度量上花费的工作量明显降低, 并且从工程师的感受来说, 行为学工具很大程度上帮助工程师完成了工作量很多的自我管理的工作, 抵触情绪降低很多。图5是软件度量时间的比较图。

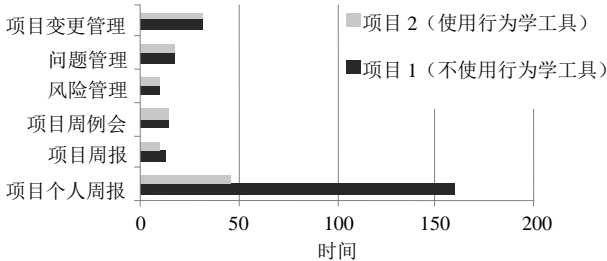


图5 软件度量工作量月报比较

可以看出, 在不使用行为学工具的项目1中, 每月花费160个工时用于收集项目数据、填写项目个人周报; 而在使用行为学工具的项目2中, 花费在项目周报上的时间减少为45工时。这为项目组节约了大量的项目管理时间。

另外, 行为学工具在识别假数据上的功能, 根据统计数据, 可以得出某工程师在某项目中一周5天的工作时间、击键次数和文件大小改变的统计数据及其离散系数。

表2 工作量度量和行为学统计标准差比较

时间	报告的工作时间		击键统计		文件大小改变				
	值	离散系数	值	离散系数	值	离散系数			
周一	8.0	0.975	609 756	13 164	1.090	854 851	2 968	1.202	008 748
周二	8.5	1.036	585 366	11 792	0.977	161 987	2 718	1.100	761 380
周三	8.0	0.975	609 756	12 285	1.018	015 181	2 848	1.153	410 011
周四	8.5	1.036	585 366	13 260	1.098	810 037	2 789	1.129	515 633
周五	8.0	0.975	609 756	9 837	0.815	157 944	1 023	0.414	304 228

根据上述的标准差计算, 可以得出如图6所示的数据度量异常统计图。

由图6可知, 周一、周二、周三、周四和周五的报告工作时间和击键行为、文件大小改变的统计标准差基本吻合, 但周五的报告工作时间和击键行为、文件大小有明显偏差, 调查发现是项目组成员在该天

的工作上报告了不正确的数据。

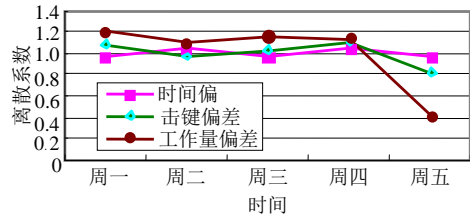


图6 数据度量偏差统计

### 5 总结

本文在软件度量方法的基础上使用3个行为学的工具, 有效地改进了软件度量的方法。帮助项目组更准确、更方便地收集数据。在实际项目中, 通过工程实践得到了统计比较的数据, 证明了统计学工具能有效地降低软件度量的工作量, 并可以有效地识别出不准确的项目数据。

#### 参 考 文 献

- [1] 邢大红, 曹佳冬, 汪和才, 等. 软件度量学综述[J]. 计算机工程与应用, 2001, 37(1): 17-19.  
XING Da-hong, CAO Jia-dong, WANG He-cai, et al. Introduction to software metrics[J]. Computer Engineering and Applications, 2001, 37(1): 17-19.
- [2] NORMAN E F, SHARI L P. Software metrics: a rigorous and practical approach[M]. Beijing: Tsinghua University Press, 2003: 97-98.
- [3] VIVANCO R A, PIZZI N J. Identifying effective software metrics using genetic algorithms[J]. IEEE Electrical and Computer Engineering, 2003, 3: 1305-1308.
- [4] 侯红. 基于度量的软件过程管理方法与分析技术的研究[D]. 西安: 西北大学, 2006.  
HOU Hong. Research on measurement-based software process management and analysis[D]. Xi'an: Northwest University, 2006.
- [5] NORMAN E F, MARTIN N. Software metrics: successes, failures and new directions[J]. Journal of Systems and Software, 1999, 47: 149-157.
- [6] KARL E W. Software metrics: ten traps to avoid[J]. Software Development, 1997: 10-35.
- [7] ANANDASIVAM G, KRISHNAN M. S, TRIDAS M, et al. Measurement programs in software development: determinants of success[J]. IEEE Transaction on Software Engineering, 2002, 28(9): 863-875.
- [8] ROGER S P. 软件工程实践者的研究方法[M]. 梅宏译. 北京: 机械工业出版社, 2003: 14-17.  
ROGER S P. Software engineering: a practitioner's approach[M]. Translated by MEI Hong. Beijing: China Machine Press, 2003: 14-17.
- [9] 李征. 基于CMM的软件项目过程度量及优化控制研究CMM在我国中小型软件企业中应用的研究[D]. 青岛: 山东科技大学, 2006.  
LI Zheng. Study of CMM-based software projects process metrics and control optimization[D]. Qingdao: Shandong University of Science and Technology, 2006.
- [10] PUTNAM L H. Trends in measurement, estimation, and control [software engineering][J]. IEEE Software, 2002, 8(2): 105-107.

编辑 张俊