

• 软件与算法 •

# 高效的小型软件项目的质量保障体系

左旭辉<sup>1</sup>, 李文泽<sup>2</sup>

(1. 西南科技大学 计算机科学与技术学院, 四川 绵阳 621010; 2. 开封教育学院 计算机系, 河南 开封 475000)

**摘 要:** 小型软件项目具有可用成本低, 预算周期短的特点, 给质量保障带来一定困难。为了从可靠性建模、缺陷预测、缺陷管理等方面研究小型软件项目的质量保障体系, 基于小型软件项目的实际特征, 为小型软件系统建立了可靠性模型。然后根据小型软件规模小的特点, 建立了改进的捕捉模型作为缺陷预测模型。设计了无需借助工具即可完成的小型软件缺陷管理流程。实践结果证明, 质量保障体系可以在不提高成本和周期的情况下保证小型软件项目的质量。

**关键词:** 软件工程; 可靠性模型; 缺陷预测; 缺陷管理

**中图法分类号:** TP311 **文献标识码:** A **文章编号:** 1000-7024 (2009) 09-2211-02

## Efficient QA system for small software project

ZUO Xu-hui<sup>1</sup>, LI Wen-ze<sup>2</sup>

(1. School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China; 2. Department of Computer, Kaifeng Institute of Education, Kaifeng 475000, China)

**Abstract:** A small software project (SSP) has limited costs and short cycle, which brings some difficulties for quality assurance (QA). QA is researched from reliability modeling (RM), defect prediction (DP), and defect management (DM). At first reliability model is built up for SSP, based on the logic features of small it. Then an improved capture-recapture model is developed for DP model according to the small scale of SSP. At last process of management for SSP defect without help of external tools. The new system can achieve QA of SSP without additional costs or cycle, which is proved in practice.

**Key words:** software engineer; reliability modeling; defect prediction; defect management

## 0 引 言

根据RitaHadden在1998年的关于CMM和小项目的软件工程过程小组会议上的定义, 型项目为周期为3~4个月并且员工数不超过5的项目<sup>[1]</sup>。目前, 国内绝大多数项目为小型项目或被拆分为小型项目实施, 因此小型项目成败关系到软件业的发展。由于软件系统组成复杂, 软件开发过程很难进行预测和控制, 软件质量也因此无法保障。为提高软件质量, 改善软件开发过程, 便于评估, 很多公司都利用能力成熟度模型(capability maturity model for software, CMM)进行过程管理。CMM对软件工程模式与标准规范进行了系统的规定, 但这些模式和规范过于庞大和复杂, 在小型软件开发过程中会极大地增加成本和降低效率, 甚至导致开发过程的失败。因此, 必须研究针对小型系统的质量保障体系。

软件系统的质量问题突出表现在可靠性与缺陷两方面。传统的可靠性理论使用任务阶段性进行建模, 主要考虑了系统的组成结构变化对可靠性的影响。由于软件结构的多样性和复杂性, 没有通用的软件结构分解方法, 也很难将传统的可靠性理论直接应用于软件可靠性建模。必须评估软件

功能结构变化对可靠性的影响, 才能完成软件的可靠性建模。解决这一问题的思路可分为基于时域和基于输入域两类。基于时域的可靠性模型即考虑时间软件系统随开发时间变化的特性, 可称为“白盒法”, 它无法描述软件开发过程中结构动态变化产生的影响。基于输入域的可靠性模型只考虑软件的外部输入而不考虑软件本身的内部结构, 可称为“黑盒法”, 也不能体现软件结构对可靠性的影响。吴超<sup>[2]</sup>等人基于软件开发逻辑思维正确性的问题提出了新的可靠性建模思想, 并给出基于失效过程的软件可靠性定义, 以及一种基于“任务-事件-功能”的软件体系结构分解方法, 进行了形式化描述和理论证明。

根据ISO 9000的定义, 所谓缺陷, 是指系统中未满足与预期或者规定用途有关的要求的地方, 说明缺陷是软件中客观存在并可以通过修改软件而消除的使人不满意的部分。20世纪70年代, 为提高和保证软件质量, 出现了缺陷预测技术, 通过学习历史数据以及已经发现的缺陷等软件度量数据, 预测软件系统的缺陷数目及类型。然而, 目前大多数缺陷预测技术都是基于对历史数据的统计<sup>[3-5]</sup>, 对小型软件项目与小型团队并不适用。消除缺陷的有效手段是进行缺陷生命周期管理,

收稿日期: 2008-10-09; 修订日期: 2009-02-09。

基金项目: 四川省教育厅自然科学基金项目 (2006C076)。

作者简介: 左旭辉 (1974—), 男, 四川人, 硕士研究生, 讲师, 研究方向为软件工程、信号处理等; 李文泽 (1979—), 男, 河南开封人, 硕士, 讲师, 研究方向为语义网络、软件工程。E-mail: clwswust@163.com

缺陷管理方面研究较多,如鞠秀娟<sup>[9]</sup>等人利用 CMM 二级模型与缺陷生命周期管理相结合设计了缺陷管理系统,这种方法在使用 CMM 的环境下效果较好,但目前尚无针对小型软件系统的缺陷管理研究。

小型软件系统由于成本较小、周期短、人员不充足等限制,在质量控制过程中具有许多特殊要求与条件,难以实现一般软件系统所使用的常规方法。在可靠性建模方面,小型系统由于功能较少、规模较小,难以表现出各种统计特性,但在系统内部的关系却更紧凑,具有更强的逻辑关联。在软件缺陷方面,小型系统可依赖的历史数据通常不多,统计学习与预测的手段效果不明显,其开发过程不成熟,导致缺陷管理困难,但小型系统的可能缺陷数远远小于大中型项目,这也降低了缺陷管理的复杂度。针对小型软件系统的这些特点,本文设计了相应的质量保障体系,并通过实际案例的总结和分析,证实了新方法的有效性。

## 1 小型软件的可靠性建模

根据吴超<sup>[10]</sup>等人的定义,可将软件的可靠性被定义如下:令  $O$  为允许用户使用的功能,  $O_c$  为禁止用户使用的功能,  $B_f$  是  $(O, O_c)$  上的关于  $P$  的功能规约函数,即  $B_f = P(O, O_c)$ ,  $P_f$  为  $(O, O_c)$  上的一个给定的概率分布,则  $S$  在概率分布  $P_f$  下对应于  $B_f$  的可靠性  $R(S, B_f)$  定义为  $R(S, B_f) = P(\{f \in O, O_c \mid P(O, O_c) = B(O, O_c)\})$ 。其中  $f_1, f_2, \dots, f_i$  表示一个完成特定任务的软件  $S$  所具备的有限功能(点)的集合。根据此定义,可以设计出软件可靠性结构(如图 1, 图 2 所示)。

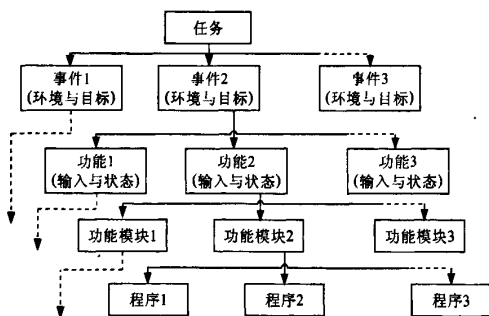


图 1 基于任务剖面的软件系统结构框架

小型软件的功能较少,但主要结构与上述模型仍然相符,因此可以沿用以上结构,所不同的是,图 2 中的  $n$  变为一个较小的数字。

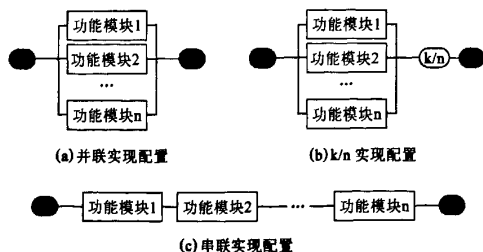


图 2 可靠性结构

## 2 小型软件的缺陷预测技术

现有的软件缺陷预测技术很多,如基于软件复杂度的缺陷预测技术、基于软件规模的缺陷预测技术、缺陷分布预测技术、基于多维软件度量元的缺陷预测技术、缺陷预测模型技术等,根据小型软件规模小、可统计量少的特点,采用 Capture-Recapture 模型为基础作为小型软件缺陷预测的基础。捕捉模型源于生物学实验中对生物种群总数的估算问题。生物种群数目庞大,流动性强,只能通过估算得出其总数。生物学家通过对部分生物的捕捉、标记、释放、再捕捉等过程,根据再捕捉中有标记生物与无标记生物的比例,利用后验概率估算出生物种群的总数。与之相似,可以利用捕捉模型来估计软件设计中的缺陷总数,即将软件缺陷视为一个生物种群,事先对设计中的某些缺陷进行记录,当软件开发测试等过程结束时根据已记录缺陷被发现的数量估计缺陷总数及软件可靠性等参数。

当然,在软件工程中的缺陷估计与生物学中的种群估计并不完全一致,最主要区别的就是在生物种群数量往往很大,并不需要精确的估算结果,而在软件缺陷估算过程中,总缺陷数的数量较低,因此,通过捕捉模型估算出的结果与实际情况会有一定出入,在小型软件项目中这一矛盾更加突出。针对小型软件项目评审人员少、可能发现的缺陷数也少的特点,对捕捉模型加入种群有限的条件,即假设每个缺陷被发现的概率与发现的顺序相关,越后发现的缺陷越难发现,即评审人员  $j$  发现缺陷  $i$  的概率为  $P_i P_j f(i, j)$ , 其中  $f(i, j)$  为  $i, j$  的减函数。

## 3 小型软件的缺陷管理技术

缺陷管理是软件质量的最终保障,为满足缺陷管理的需要很多公司开发了专业的测试管理软件,如 Mield Soft 的 RAI-DS, IBM Ratio 的 Test Studio 等。这些软件可以帮助一般的大型项目进行测试管理,但使用成本较高,小型项目无法承担这些费用,同时软件功能复杂,很难在短期内完成对员工的培训。另外,也有一些免费的测试管理软件,如 Mantis, Bugzilla, Bugfee 等,这些软件功能过于简单,很难满足实际应用的需要。因此,必须设计满足小型软件项目需要的缺陷管理体系。

缺陷管理通常需要经过若干环节,包括:①缺陷收集过程,收集包括质量评估、综合测试、客户试运行等项目活动中产生的各种缺陷;②缺陷责任分配,项目负责人将各个缺陷指派给相关人员,并通知责任人;③缺陷集中分拣,缺陷责任人对缺陷进行分类处理;④缺陷改进与修改,根据项目实际情况,责任人根据要求对缺陷进行相应修正;⑤缺陷部署,修改人员完成缺陷的修改后,提交修改后的版本给部署人员;⑥缺陷回归,测试人员受到部署人员的通知,对缺陷进行功能与性能验证。如果验证过程未能通过,缺陷将重新进入下一个版本循环。根据小型软件项目的实质情况,缺陷分拣与缺陷部署均可以由修改人员完成,而修改人也基本可以确定为责任人,因此缺陷管理过程可以简化为缺陷收集、缺陷分配、缺陷修改、缺陷回归 4 个步骤,简化后的流程如图 3 所示。

图 3 流程简单易行,无需借助工具都可以完成,既能保证软件质量的要求,又能满足小型软件项目对成本和时间上的控制。

(下转第 2318 页)

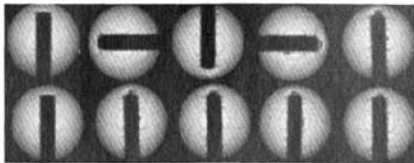


图6 旋转3维 Zalesak 圆盘界面追踪测试结果

表1 两种存储结构性能比较

存储结构	解析度	耗费空间/G	平均处理速度/(t/s)	追踪效果
三维数组	100×100×100	15.26	1.98	精确度高
H_RLE 水平集	100×100×100	0.38	2.51	精确度高

别。但如表1所示,在解析度为100×100×100时,本文方法耗费空间仅为三维数组结构的2.5%,平均处理速度提高了22.11%。

5 结束语

本文采用的H\_RLE水平集结构是一种高度压缩的存储结构,以该结构作为界面存储结构可使空间复杂度由O(n³)降低至O(D)。本文改进了重构窄带方法,使得重新初始化时间复杂度由O(n³)降低至O(D)。随着解析度的不断提高,三维数组将不能满足应用需求,而本文提出的追踪方案可以满足计算机应用存储空间小、运行速度快的要求,它在视频运动物体追踪、流体模拟、三维图形分割等领域,有着广泛的应用前景。

(上接第2212页)

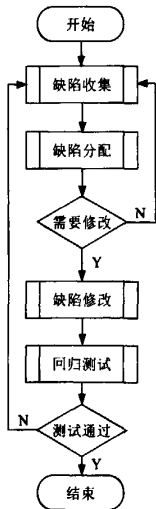


图3 小型软件项目缺陷管理流程

4 结束语

本文结合小型软件项目的实际需要与客观条件,设计了相关的可靠性建模、缺陷预测和缺陷管理等方法,为小型软件的开发提供了切实可行的质量保障方法。通过某公司进销存管理软件、某省水文局历史水文数据管理系统、某厂生产信息管理系统、某通信公司彩信阻断系统等项目实践,本文所设计的各种方法都已通过验证。事实证明,小型软件项目可以在

参考文献:

[1] 柳有权,刘学慧,朱红斌,等.基于物理的流体模拟动画综述[J].计算机辅助设计与图形学学报,2005,17(12):2581-2589.

[2] Bridson R.Computational aspects of dynamic surfaces[D].Stanford,CA: Stanford University, 2003.

[3] Guendelman E, Bridson R, Fedkiw R. Nonconvex rigid bodies with stacking [J]. ACM Transactions on Graphics, 2003,22 (3): 871-878.

[4] Losasso F,Gibou F,Fedkiw R.Simulating water and smoke with an octree data structure [J]. ACM Transactions on Graphics, 2004,23(3):457-462.

[5] Houston B,Nilson M B,Batty C,et al. Hierarchical RLE level set: A compact and versatile deformable surface representation[J]. ACM Transactions on Graphics,2006,25(1):151-175.

[6] 谷汉斌,李炎保,李绍武,等.界面追踪的Level Set和Particle Level Set方法[J].水动力学研究与进展,2005,20(2):152-160.

[7] Enright D,Marschner S,Fedkiw R.Animation and rendering of complex water surfaces [J]. ACM Transactions on Graphics, 2002,21(3):736-744.

[8] Enright D,Lossasso F,Fedkiw R.A fast and accurate semi-Lagrangian particle level set method[J].Computer and Structures, 2005,83(7):479-490.

保证进度和成本控制的同时保证质量。

参考文献:

[1] Berger A,Della Pietra S,Della Pietra V.A maximum entropy approach to natural language processing [J]. Computational Linguistics,1996,22(1):39-71.

[2] 吴超,林家骏,俞岭,等.软件可靠性建模研究[J].计算机工程, 2008,33(11):52-54.

[3] 王青,伍书剑,李明树.软件缺陷预测技术[J].软件学报,2008,19 (7):1565-1580.

[4] 朱永春,徐红.一种基于历史数据的软件缺陷预测方法改进[J].北京航空航天大学学报,2003,29(10):947-950.

[5] 刘旸.基于机器学习的软件缺陷预测研究[J].计算机工程与应用,2006,28(1):49-53.

[4] 鞠秀娟,赵明.基于CMM的缺陷管理系统的设计及应用[J].四川大学学报,2007,39(51):238-241.

[5] 刘英博,王建民.面向缺陷分析的软件库挖掘方法综述[J].计算机科学,2007,34(9):1-4.

[6] Mark C Paulk.The capability maturity model(CMM L2) guidelines for Lmproving the software process[M].Addison-Wesley, 1995:18-42.

[7] 张祖勇,孙龙清.基于CMMI的软件过程管理系统的研究[J].微计算机信息,2008,24(3):25-27.

[8] 朱连章,李妍琛.用于软件可靠性分析的分解方法[J].计算机工程与设计,2007,28(24):5835-5840.