

软件维护外包后的缺陷修正流程改进及分析

李玉鹏, 侯红

(西北大学软件工程研究所, 西安 710127)

摘要: 通过研究某公司的大型芯片测试软件维护(纠错性维护和完善性维护)跨国外包后的缺陷修正方法, 总结得到了缺陷修正的基本控制流程。针对基本控制流程的缺点, 从降低维护成本和提高缺陷修正的效率等方面出发, 提出了一系列改进方案及新的缺陷修正流程, 给出了流程改进后的软件质量保证和项目组的工作效率分析。

关键词: 软件外包; 缺陷修正; 缺陷修正流程; 流程改进分析

Process Amelioration and Analysis of Defect Fix of Outsourced Software Maintenance

LI Yupeng, HOU Hong

(Software Engineering Institute, Northwest University, Xi'an 710127)

【Abstract】 By investigating the defect fixing method of large-scale CMOS chip testing software maintenance, including corrective maintenance and perfective maintenance, after outsourced of certain company, this paper concludes and gains basic control process of defect fixing. Aiming at some shortcomings of this basic control process, it gives a series of improved schemes and new defect fixing process. It also analyzes software quality and working efficiency of improved process.

【Key words】 Software outsourced; Defect fix; Defect fix process; Process amelioration and analysis

有数百万行代码的大型软件常使用多种程序语言编写, 可能有多种体系结构, 模块多、应用领域知识复杂, 一般都有十几年以上的进化历史。在当今竞争日益加剧的情况下, 软件的开发不可能停止, 需要增加新的功能, 支持新的操作系统和新的编译器等。软件在某一版本发布后, 不可避免地存在许多缺陷, 这些缺陷必须得到及时的修正, 这是对软件生命力、竞争力的重要保障。

A93k 是 X 公司为测试大型芯片而设计的软件。约有 650 万行代码, 使用 C/C++、Java 等程序设计语言和 Purl、JUnit 等自动回归测试设计工具。软件本身已经进化 25 年以上, 再加上芯片设计技术的发展, 软件架构历经多次变化, 如 HP_UX 的进化, 从 HP_UX 到 Linux 系统平台的代码移植, 以及部分代码的 Java 实现等。由此产生了大量缺陷, 也使软件的缺陷修正(Defect Fix)难度增大。所以 X 公司为控制成本, 将解决该软件缺陷修正问题的项目外包给中国上海某软件公司及其西安子公司, 2 个公司分别在 2002 年初和 2003 年底建立了 CRT (Customer Response Team) 项目组。CRT 项目组要面临软件跨国外包后的一系列问题, 比如缺陷修正过程中软件质量的控制等。

3 年来 CRT 项目得到了该软件客户 CISCO、IBM、Intel 的赞誉。这主要归功于 CRT 项目组和 X 公司之间达成的对缺陷修正流程的严格控制。本文主要通过对 CRT 项目组缺陷控制方法和其取得成功的关键因素进行研究, 定义了 CRT 项目组缺陷修正的控制流程, 并针对其现有流程的弱点给出了改进方案和改进后的控制流程。本文的研究对我国一些正在成长的大型软件的缺陷修正业务和我国的软件外包业务有积极的意义。

1 缺陷修正的控制流程

本文提到的缺陷包括软件纠错性维护中的代码漏洞和完善性维护中一些需要改进和完善的系统功能。需要说明的是, X 公司并没有要求 CRT 修正所有的缺陷, 极难处理的缺陷经认可后可由 X 公司的模块负责人(Module Owner, MO)修正; X 公司任命各个 MO 控制方案设计审查(Design Review)和代码实现审查(Implementing Fix Review)这 2 个缺陷修正里程碑的质量。

软件的缺陷可由手工测试或工具测试获知, 但本文主要关注客户在使用软件过程中发现的缺陷。缺陷修正是软件维护的重要工作之一, 良好的缺陷修正流程控制可以提高缺陷修复的效率, 控制软件的质量, 并可以在缺陷修正后出现问题时, 查找到详细的修改信息和修改目的。CRT 项目组经过大半年的探索、总结, 形成了 CRT 特有的极为详细的工作流程, 使其能够时刻监视每个缺陷的状态, 发现相应问题并采取对策; 使 CRT 内部、CRT 和 X 公司之间能够密切地协作; 使双方能够积累并共享 CRT 特有的缺陷修正知识库。流程设计首先要确定流程的角色, 以及角色的职责和它们的协作关系; 其次要确定流程的主要里程碑, 这是过程控制的关键。CRT 流程的主要角色及其职责如表 1 所示。缺陷跟踪流程概要图如图 1 所示。

CRT 流程的主要里程碑有:

(1) Assign CR (From CRT PM);

作者简介: 李玉鹏(1981 -), 男, 硕士生, 主研方向: 软件度量; 侯红, 博士、高工

收稿日期: 2006-06-15 **E-mail:** lyp800905@sina.com

- (2)Reproducing(From Assigned) ;
- (3)Waiting For Feedback(Reproducing) ;
- (4)Design Fix ;
- (5)Waiting For Feedback(Design) ;
- (6)Implementing Fix ;
- (7)Waiting For Feedback(Implementing) ;
- (8)Set RFD(ready for deliver) ;
- (9)Test and Release(test team and release team).

表 1 CRT 流程的主要角色及其职责

角色	职责
X company Submitter Team	负责 CR/defect 的生成, 提供 CR 详细的重现步骤和必要文件
X company MO Team	负责 HPSmarTest 一定模块的开发, 是该模块功能的权威, 是缺陷修正是否通过 Review 的决定者
X company CR Management Team	协调 MO 和 CRT 项目组之间的工作, 把用户生成的缺陷提交到 CRT 项目组
CRT Junior Engineer	负责 CR/缺陷的修正
CRT Senior Engineer	负责新模块的 transfer, 员工培训, 缺陷修正
CRT Project Manager	分配 CR, 管理团队, 缺陷修正
CRT QA Team	核对 CR 所处的不同状态, 确保修正流程的正确实施
CRT Test Team	对 release 前已实施的缺陷修正进行软件测试
CRT Flow Development Team	维护 CRT Flow, 添加新功能及本软件的缺陷修正

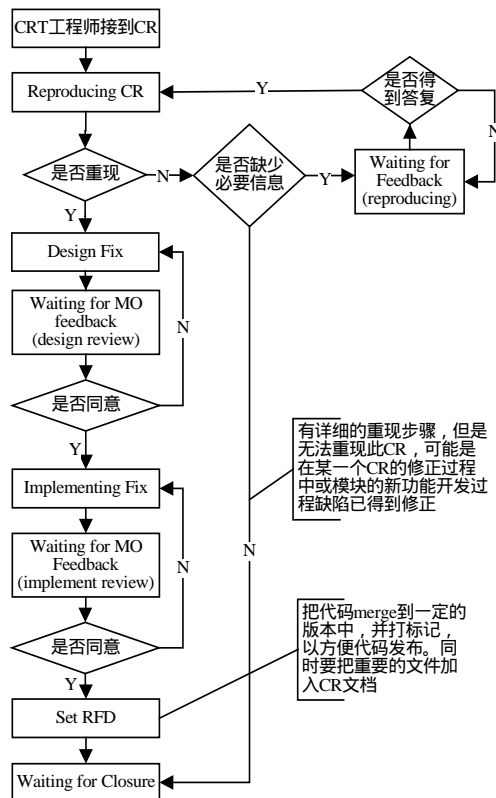


图 1 缺陷修正流程

可以看到, 一个缺陷要得到修正就必须经过重现、设计、实施、发布 4 个重要步骤, 在重现的过程中, 如果能够一次性重现这个缺陷, 那么在 CRT Flow 中更改到 Design Fix 状态, 就可以进入下一步的设计修正工作。然而实际情况通常不那么顺利, 大部分工程师接到一个 CR(缺陷)的时候, 常常缺少必要的重现步骤和相应的重现文件, 这时就要自行尝试重现缺陷, 或者跟公司的 Submitter 进行沟通来获取重现所需

要的信息, 这一步通常需要 5 个工作日左右的等待时间。有时候这些等待后得到的信息也不足以重现缺陷, 这就需要更加漫长的交互和等待时间。

在设计、实施缺陷修正的过程中, 当 CRT 工程师有了一个缺陷修正的设计思路时, 要发送 E-mail (Design Fix Review) 给 X 公司的 MO, 等待对方确认修正方案, 只有 MO 同意之后 (通常是一周), 可以在一定版本上进行代码实现。

CRT 工程师最重要的任务就是代码修改后, 对整个系统 (A93k) 进行编译, 并在相关模块之间进行小范围的测试, 确保在这种情况下没有带来其它的软件缺陷, 此后, CRT 的工程师就可以把具体的修正代码以及是否要添加测试用例等事项通过 E-mail (Implementing Fix Review) 发送给 X 公司的 MO, 在得到其确认的情况下, 进行代码的 merge 及其它相关工作。在以上任何一种交互中, 如果 MO 不同意这个修正方案, 缺陷就要回到 Design Fix 状态, CRT 工程师必须重新设计缺陷修正方案, 并重复以上步骤。

2 缺陷修正流程的改进

图 1 显示, 大量时间被花费在 CRT 工程师跟 X 公司的 MO 和 Submitter 的交互中, 其主要出现在缺陷的重现和修正的设计、实施过程中, 严重制约了 CRT 项目组工作效率的提高。根据对 CRT 项目组近 1 年的工作流程的总结和分析, 本文提出了完善和简化的缺陷修正控制流程, 改进后的缺陷修正流程如图 2 所示。

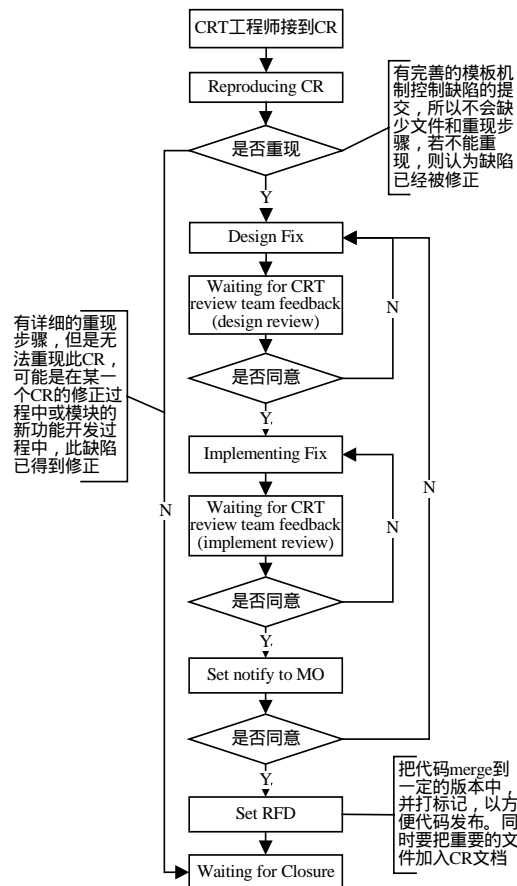


图 2 改进后的缺陷修正流程

其主要特点表现为

- (1) 严格控制缺陷的生成机制

因为在成本控制方面, 完善的重现步骤将直接影响到缺

陷修正的方方面面,所以要求 X 公司的 CR Management Team 严格控制缺陷的生成机制,在缺陷的生成过程中必须有详细的重现步骤和相应的重现文件,用固定的模板机制控制缺陷的提交质量,有效避免仅提出缺陷现象而导致 CRT 工程师无法重现缺陷的情况发生。

(2) 组建 CRT 项目组的复审团队(Review Team)

在缺陷修正被最终实施之前,CRT 工程师不必和 X 公司的 MO 进行交互并得到其确认,MO 这一阶段的工作由 CRT 项目组的复审团队完成,主要包括缺陷修正的设计方案审查和代码实施审查。但是在缺陷的代码最终实施(Set RFD)之前要发送通知(notify)给 X 公司的 MO,若 MO 同意,即可进行代码的实施以及相应的收尾工作,若不同意,则 CRT 工程师在考虑 MO 意见的基础上重新实施修正。

由此在缺陷的重现和设计实施过程中将节省大量的时间,缺陷流程被改进后,CRT 工程师在一个缺陷上花费的时间平均缩短了 5 个工作日,同时 X 公司 MO 的工作量也得到相应缩减,而这并不是以损伤软件的质量为代价的。因为 MO 的工作被 CRT 高级工程师组建的复审团队所替代,CRT 工程师和 CRT 复审团队之间的交流快捷而有效,复审团队能更加充分快捷地理解工程师的设计思路。这样,CRT 项目组在得到更大自主权的同时,提高了工作效率,壮大了团队,最重要的是在技术进步和经济效益提高 2 个方面均有较大的收获。

3 流程改进后的质量、效益分析

3.1 质量分析

为了验证以上流程的可行性,CRT 项目组在同 X 公司协商后,决定在局部范围内采用新的流程。几个月来,CRT 项目组出色的工作受到了 X 公司的赞许,同时也为其赢得了 X 公司新开发的几个模块的维护工作,充分说明了 CRT 项目组的工作质量是稳定而可靠的。CRT 项目组在流程改进后的质量分析方面主要采用缺陷修正后 Reopen 缺陷(修正有严重缺陷要重新修正或者由此修正产生了其它缺陷)的数量为依据。因为 CRT 的月缺陷修正数量基本稳定,所以 Reopen 缺陷应该稳定在一定的范围内,或者说其百分比(Reopen 缺陷/closed 缺陷)应该在一定范围内波动。图 3 反映了 CRT 下半年 Reopen 缺陷的数据。

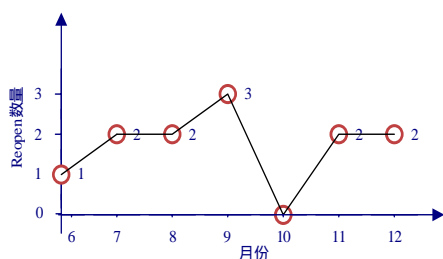


图 3 2005 年下半年 Reopen CR 的数据

据统计,CRT 上半年的月平均 Reopen 缺陷数量为 2 个,在 8 月份采用新的缺陷修正流程后,月平均 Reopen 缺陷数量为 1.86 个,Reopen 缺陷/closed 缺陷一直在 0.04 左右浮动(见图 3、图 4)。而在此后 4 个月的软件使用和测试过程中,也没有发现因为 X 公司 MO 的工作被 CRT 复审团队所替代而产生影响。所以从目前的分析来看,改进后缺陷修正的质量是可靠的,软件质量得到了有效保证。

3.2 效益分析

分析效益时,主要采用单月缺陷修正数量和缺陷修正周期进行评价。根据 2005 年下半年的统计数据,自 8 月份采用新流程后的 10~12 月份的缺陷修正数量明显有所提升。2005 年下半年月修正缺陷数量如图 4。

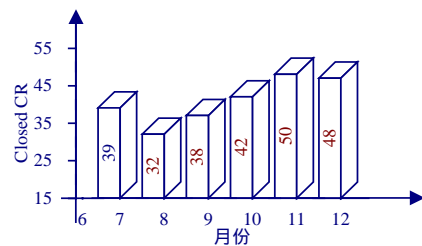


图 4 2005 年下半年月修正缺陷数量

2004 年的统计资料显示,每个缺陷的平均修正周期是 47.5 天,而 2005 年缺陷流程改进后 8~12 月每个缺陷的平均修正周期是 40 天,平均缩短了 5 个以上的工作日。显然 CRT 在降低运行成本的同时增加了利润,而成本的降低不是建立在降低客户满意度的基础上。事实上,缺陷修正流程的改善以及各里程碑控制得力降低了成本,各项指标反而提高了。

4 结论

本文调查研究了大型软件维护跨国外包后的缺陷修正控制流程及其改进方法,提出了以缺陷修正流程跟踪和里程碑式的状态转换为解决方案,根据已有流程的不足提出相应的改进措施,并分析了改进后的软件质量保证和工程师的工作效率。通过研究得出:软件维护外包的缺陷生成机制必须得到完善;软件维护外包的发包方应该在软件质量得到保证的基础上,通过在承包方组建复审团队,把缺陷修正流程控制的一定权限下放到承包方以提高修正效率。

缺陷流程的改进是持续不断的过程,项目团队应鼓励团队成员发现问题,提出进一步的改进方案。对于缺陷修正流程的改进,CRT 项目组下一步从以下几个方面入手:在 CRT 工作流程中增加模块缺陷统计功能,以期找到缺陷多发模块,在此模块加强测试及质量控制;制定完善的缺陷生成模板机制,用模板来控制 Submitter 提交的缺陷描述质量;X 公司的 MO 定期为 CRT 复审团队进行模块知识培训,加强团队成员的专业素质;在 CRT 复审团队内部施行模块责任制,指定类似于 MO 的模块主要负责人以提高复审质量。

参考文献

- Osterweil L. Software Processes are Software tool[C]//Proceedings of the 9th International Conference on Software Engineering, Monterey, CA. 1987.
- Watts S H. Managing the Software Process[M]. Publishing House of Electronics Industry, 2003.
- 胡达风. 大型软件缺陷修正跨国外包后的质量控制[J]. 计算机工程, 2004, 30(增刊).
- Pressman Roger S. 软件工程: 实践者的研究方法[M]. 北京: 机械工业出版社, 1999.
- Cardoso J, Sheth A, Miller J. Workflow Quality of Service[R]. LSDIS Lab, Department of Computer Science, 2001.
- 王小巍, 朱小冬, 王毅刚. 基于 CMM 的软件维护过程管理及其工具研究[J]. 微计算信息, 2005, 21(7).