

• 计算机软件与算法 •

论软件评审在军用软件质量控制中的作用

刘从越, 张洪霞

(中国电子科技集团公司第七研究所, 广东 广州 510310)

摘要: 为了说明软件评审在军用软件质量控制中所起的重要作用, 对评审和测试的有效性进行了比较, 并对目前我国军用软件领域中的评审实施不利的原因进行了深入的剖析, 最终得出结论, 软件评审对军用软件产品的质量有至关重要的影响, 只有采用正确的方法进行评审, 并将软件评审和软件测试相结合, 才能真正提高我国军用软件的质量。

关键词: 评审; 审查; 测试; 质量; 缺陷清除率

中图分类号: TP311.5 **文献标识码:** A **文章编号:** 1000-7024 (2009) 08-1900-03

Roles of software review in military software quality control

LIU Cong-yue, ZHANG Hong-xia

(China Electronics Technology Group Corporation No.7 Research Institute, Guangzhou 510310, China)

Abstract: The goal is to illustrate the important roles of software review in military software quality control. The effectivity of software review is compared with the effectivity of software testing, and the reasons why review did work well in the area of military software in China in depth is analyzed. The conclusion is: Software review plays important role in quality of military software product. Quality of military software product can be truly improved only when reviews are adopted correctly and when review and testing is combined together.

Key words: review; inspection; testing; quality; rate of defect removing

0 引言

高可靠性和高安全性是军用软件最显著的特征。为了保证软件的可靠性和安全性, 就需要采用一些方法对军用软件的质量进行有效的控制。软件评审和软件测试是最常用的软件质量控制方法。

广义的测试也包括评审, 但通常所说的测试是狭义的测试, 指编码完成后所进行的发现缺陷的活动。本文中的测试均指这种狭义的测试。软件测试早已在军用软件开发过程中得到广泛应用, 而且测试技术也日渐成熟。在 GJB 2786 中, 已经将单元测试、集成测试、配置项测试、系统测试作为标准的软件开发阶段。可以说, 软件测试已成为军用软件工程活动中不可缺少的活动。然而, 大量数据表明, 软件测试清除缺陷的能力是非常有限的, 远远不能满足军用软件对质量的要求。

软件评审指对软件工作产品进行人工检查以发现错误的过程。Karl Wieggers 将软件评审的形式分为 6 种, 按正式程度的高低依次为: 审查、团队评审、走查、结对编程、轮查、非正式评审。其中, 审查是最正式的评审形式。已经证实, 审查是目前最有效、最强大的软件缺陷清除方法。软件评审, 特别是软件审查在美国军用软件领域早已引起高度重视并得到了广泛

应用。在美国, 除军用软件外, 航空软件、电信软件、医疗设备软件及系统软件和操作系统等所有高可靠性软件的开发商都会在测试前对软件进行审查, 并将审查作为首选的软件缺陷清除方法。

但是, 和软件测试的成功应用相比, 评审在我国军用软件的开发过程中还远未发挥出应有的作用。为了改进目前这种状况, 推进评审在军用软件中的应用, 本文先对软件评审和软件测试的有效性进行了比较, 然后分析了软件评审目前在我国的军用软件开发中实施不利的主要原因, 最后指出只有采用正确的方法进行评审, 并将软件评审和软件测试相结合, 才能真正提高我国军用软件的质量。

1 软件评审和软件测试的有效性比较

我们从缺陷清除率和效率两个方面来比较软件评审和软件测试的有效性。

1.1 缺陷清除率的比较

软件评审的缺陷清除率也随着评审的正式程度的提高而增加。在所有评审形式中, 正式审查的缺陷清除率是最高的。

20 世纪 70 年代, IBM 的 Michael Fagan 及其同事发明了正式审查的方法。35 年来, 审查过程积累了大量的、连续的数据, 已被证实是到目前为止效率最高且最有效的一种质量控

收稿日期: 2008-04-07; 修订日期: 2008-06-03。

作者简介: 刘从越 (1971—), 女, 黑龙江牡丹江人, 高级工程师, CCF 会员, 研究方向为软件过程改进、嵌入式软件分析设计; 张洪霞 (1957—), 女, 山东阳谷人, 高级工程师, 研究方向为项目管理、质量管理。E-mail: congryue_liu@hotmail.com

制方法。通常情况下, 正式审查发现缺陷的效率是已知的任何测试的两倍。大多数测试只能发现不到 30% 的缺陷, 而正式审查通常可发现约 60% 以上的缺陷, 有时甚至高达 95%。此外, 审查还可以提高测试的效率, 并且是非常有效的缺陷预防方法。

基于大量软件项目的数据, Capers Jones 总结出与软件测试和软件评审有关的两条经验法则:

- (1) 每一次软件测试将发现并清除目前存在的 30% 的错误;
- (2) 每一次正式设计审查将发现并清除目前存在的 65% 的 bug。

由此可以看出, 正式设计审查和代码审查的缺陷清除率比测试要高得多。正因为每一次测试的缺陷清除率比较低, 所以, 在软件生存周期中, 才需要进行从单元测试到系统测试的多级测试。这也充分说明, 美国软件业的平均缺陷清除率仅为 85% 的原因是因为绝大多数软件组织没有采用正式审查。而 IBM、HP、Motorola 等成熟的软件组织的缺陷清除率之所以能高达 95%, 甚至 99%, 最主要的原因就是他们都采用了正式审查。

Ron Radice、Karl Wiegers 及其它研究人员在软件审查方面的近期研究结果也进一步证实了早期结论的正确性, 即证实了人是发现并解决源于需求、设计和其它非代码工作产品中的复杂问题的最佳“工具”。由于人脑具有很强的适应性, 并同时具备逻辑归纳和逻辑推理的能力, 所以审查也是一种最通用的缺陷清除方法。

从本质上来说, 审查可用于任何软件工作产品。实际上, 正式代码审查在发现源代码中的深层问题方面的效率也远远高于测试。

相比较而言, 多数测试仅能发现低于 30% 的错误, 发现需求和设计缺陷的效率就更低。而 IBM 公司的统计数据显示: 大多数企业的产品开发中, 2/3 的缺陷是在需求和设计阶段引入的。因此, 对于那些在需求和设计等早期阶段引入的缺陷来说, 通过评审尽早发现的修复成本远低于在产品开发后期测试中才发现的修复成本。如果在需求阶段和设计阶段没有发现这些错误, 那么测试很可能根本无法发现这些缺陷, 因为测试用例是以需求规格说明和设计规格说明为依据而设计的。

跨世纪时出现的众所周知的千年虫问题就是一个非常典型的软件需求缺陷的例子。多年以来, 软件测试一直没有发现该问题的主要原因是: 软件需求中规定用两位数表示年份, 而且该需求通过了评审。但问题的关键是: 软件测试用例是以需求规格说明为依据来设计的, 所以测试根本就无法发现问题。就是这样一个小小的需求问题, 在全球造成了数以千万亿计的损失。

我们用图 1 和图 2 来说明评审和测试的缺陷清除率之间的差异。图中将发现软件缺陷的时间点与软件开发中引入缺陷的时间点用直线相连。引入缺陷的点和发现缺陷的点之间的连线角度越小, 说明软件质量控制的问题越严重。因为引入错误和发现错误之间的时间间隔可能已经长达几个月, 而且已经以有缺陷的工作产品为依据生成了其它的工作产品, 这样就使缺陷也随之进入其它工作产品中。

缺陷清除的目标是使引入缺陷的点和发现缺陷的点之间的连线角度接近 90 度。尽管 90 度是不可能的, 但正式审查至少可以将角度由 30 度增加到 60 度 (如图 1 和图 2 所示)。

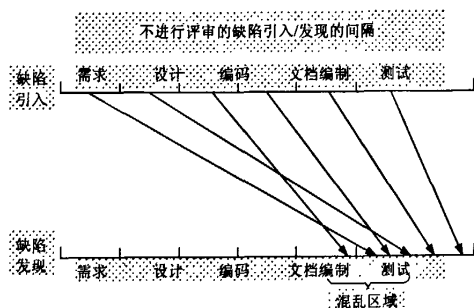


图 1 没有进行正式审查的缺陷引入点和发现点

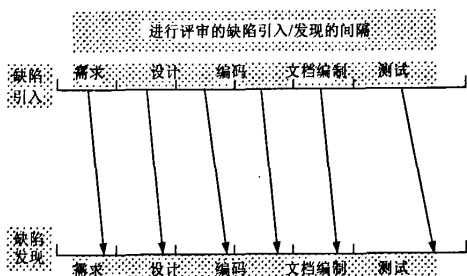


图 2 进行正式审查的缺陷引入点和发现点

从图 1 很容易可以看出, 没有进行正式审查的软件项目从测试阶段开始进入混乱区。这是因为需求和规格说明中的深层次问题会突然之间暴露出来, 需要进行大量的、代价高昂的修改和返工。评审的最大贡献之一就是比测试在更早期发现错误, 从而可以在早期以最低的成本消除错误。

而在图 2 中, 引入缺陷的点和发现缺陷的点之间的连线角度比较大。更重要的是要注意到缺陷是在其引入阶段即被清除, 而没有进入后续阶段的软件活动。

1.2 效率的比较

评审不仅成本非常高, 也非常耗时。3 至 6 名人员以低于 10 页/小时的速率审查软件文档意味着审查的生产率仅为 1.25 页/人时。如果被审查的软件文档的规模很大, 如有 500 页, 则审查的总工作量将高达 400 个人时。

对审查进行估计时通常采用页等物理度量。例如, 以软件需求文档的页数和用例数为单位的典型经验法则如表 1 所示。

评审的目的是发现并且修正错误, 因此, 软件工作产品中错误的数量和严重级别会对评审的时间和成本产生非常大的影响。

但是, 从整个软件生存周期来看, 审查既可降低项目成本, 又可缩短项目进度, 是一个名符其实的可以一举多得的质量控制方法。Capers Jones 用一个示例说明了审查效率对测试

表1 软件需求文档审查的经验法则

评审活动	生产率
准备	10 到 15 页/小时
准备	5 到 10 个用例/小时
审查会议	5 到 10 页/小时
审查会议	3 到 10 个用例/小时

效率的影响,该示例建立在软件生产力研究所收集的数以千计的软件项目的数据的基础之上。

假设软件项目的规模为1000个功能点,规格说明的规模为500页,那么规格说明的审查准备和审查共需花费500个人时。但是,审查中共发现了250个缺陷,其中50个是严重缺陷。对该项目的测试也将花费约500个人时。

现在假设在开发同样的1000个功能点的项目时没有进行设计审查,而且在开发之前没有发现这250个缺陷,其中50个是严重缺陷,那么这些缺陷就进入了编码阶段。在这种情况下,测试将花费2000个人时,因为在测试阶段发现并清除严重的设计缺陷是非常困难的。

因此,正式设计审查所投入的500个人时可减少测试的1500个人时,从而缩短整个项目的交付进度。

这是基于对大量经验数据的观察而得出的结论。尽管不同项目的实际结果会有一些的差异,但是,通常情况下,采用了审查方法的项目的进度周期会缩短15%,工作量会减少20%,而应用程序发布前可能会多清除200%的缺陷。进行审查的项目在前期进度会比较慢,成本也比较高,但当测试开始时,情况就会发生变化:与未经过审查的部分相比,审查过的部分通常在测试进行到1/3或1/4时就完成了测试。而且和未经审查时的缺陷修复的工作量相比,经过审查后的缺陷修复的工作量也只有不到1/3。

如前所述,软件评审有多种方式。总的来说,其它评审方式的有效性低于审查的有效性,但至少可以保证一部分缺陷在其引入的阶段即被发现,从而在一定程度上降低了软件的质量成本,同时提高了测试的效率。

总的缺陷清除率高于99%的项目通常都会将评审和审查与软件测试结合起来使用。

2 软件评审实施不利的原因分析

如前所述,评审是业界公认的最高效的质量控制手段。但奇怪的是,在我国的军用软件开发过程中,评审并没有得到很好的实施,甚至有些军用软件研制单位根本就没有进行评审。

造成这种现象的主要原因如下:

- (1)评审过程流于形式,缺乏可操作性(如没有提供检查表或检查表不具备可操作性及针对性);
- (2)既未对员工进行评审流程的培训,也未在评审过程中提供适当的指导和监督;
- (3)对评审的重要性和严肃性认识不足,没有在项目计划中考虑评审的时间,评审前的策划和准备也不够充分。这样,评审就变成一种临时性的行为;

(4)评审人员的评审技能或专业知识技能不够;

(5)评审会偏离主题,不是重在发现问题,而是讨论如何解决,使评审会变成技术攻关会,从而降低了评审效率;

(6)没有对评审发现的问题进行跟踪,使评审功亏一篑;

(7)没有收集、分析评审的测量数据如评审工作量、发现的问题数、解决的问题数、缺陷清除率等,无法使管理层和技术人员看到评审带来的效益。因此,也就无法说服他们积极组织并参与评审。

找到了评审失败的原因之后,就可以有针对性地采取措施,提高评审的质量和效率。Karl Wiegers、Ron Radice等人在软件评审和审查方面做了大量的研究,从而摸索出了很多有效、实用的评审和审查方法,有兴趣的读者可参考相关的文献。

3 结束语

一言以蔽之,软件评审对军用软件产品的质量有至关重要的影响。尽管目前评审在我国的军用软件开发中并没有发挥应有的作用,但这并不等于说评审技术是无效的,只能说明我们目前所采用的评审方法还存在很多问题,从而导致大部分评审是在走形式,走过场。

软件评审即是技术活动,也是管理活动,因此会受到国家文化、企业文化的影响。为改变目前的评审现状,当务之急是开展软件评审方面的研究,摸索出适合我国国情的软件评审方法和技术。

但是,只做评审仍不足以保证军用软件所要求的高质量。要使我国军用软件的质量有质的飞跃,必须大力推广使用各种高效的软件评审技术,并将软件评审和软件测试有机地结合起来。

参考文献:

[1] Jalote P.CMM 实践应用-Infosys 公司的软件项目执行过程[M].北京:电子工业出版社,2002:150-165.

[2] JONES C. Estimating Software Costs [M]. 2nd Ed. New York: McGraw-Hill,2007:421-432.

[3] Kan S H. Metrics and models in software quality engineering [M]. 2nd Ed. 北京:清华大学出版社,2004:271-308.

[4] Pressman R. 软件工程-实践者的研究方法[M]. 6 版. 北京:机械工业出版社,2005:563-569.

[5] Radice R S. High-quality low-cost software inspections[M]. Andover:Paradoxican Publishing,2002:74-88.

[6] Schulmeyer G. 软件质量保证[M]. 3 版. 北京:机械工业出版社,2003:153-177.

[7] Wiegers K E. Peer reviews in software-A practical guide[M]. 北京:科学出版社,2004:31-41.

[8] Wiegers K E. 软件需求[M]. 2 版. 北京:清华大学出版社,2004:181-192.

[9] 石柱. 军用软件能力成熟度模型及其应用[M]. 北京:中国标准出版社,2003:103-105.