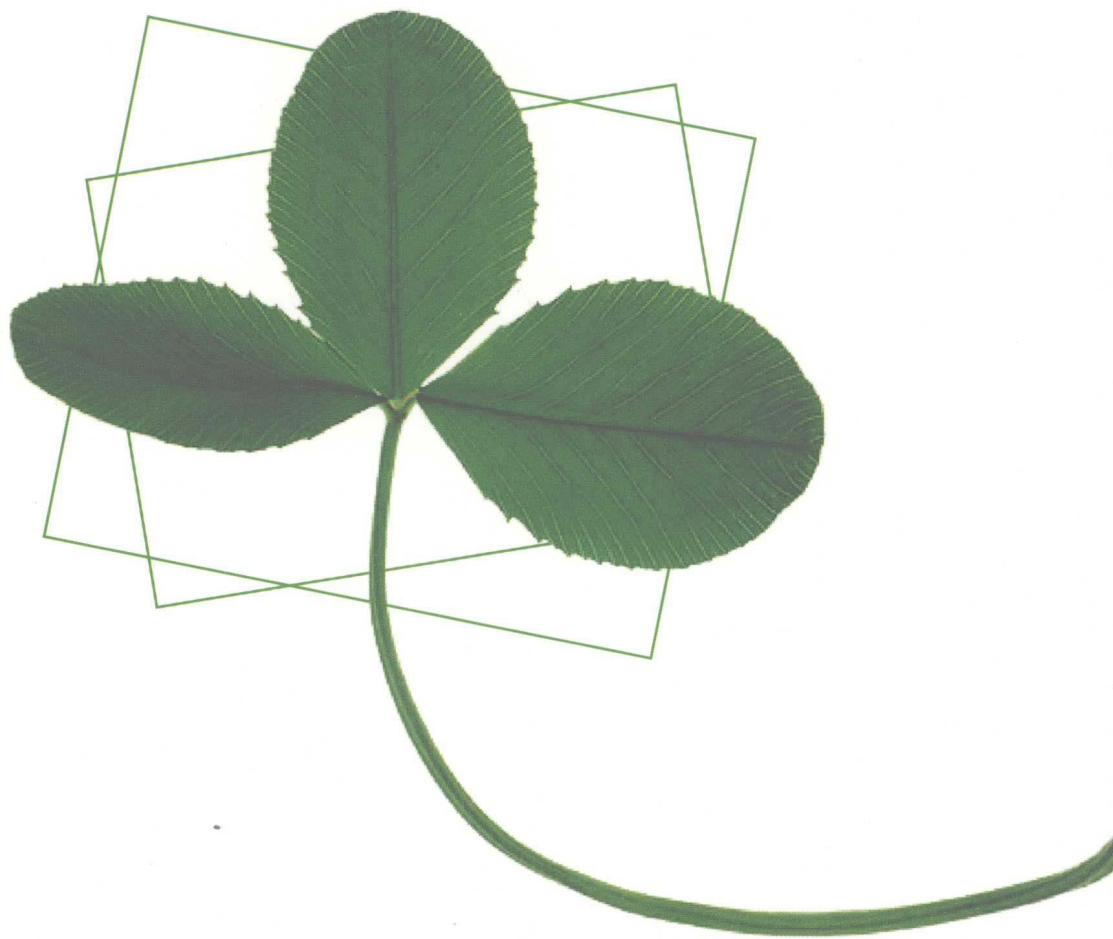


Broadview[®]
www.broadview.com.cn

测试实践丛书



Practice of Software Quality Management

软件质量管理实践

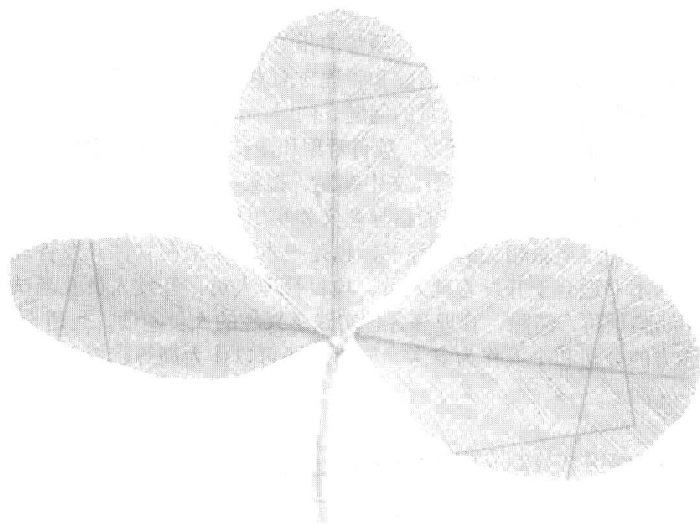
——软件缺陷预防、清除、管理实用方法

于波 姜艳 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

测试实践丛书



Practice of Software Quality Management

软件质量管理实践

——软件缺陷预防、清除、管理实用方法

于波 姜艳 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从过程管理角度，分析了影响软件质量的相关因素，分享了可行易操作的实施与管理方法。本书涉及软件缺陷分类、预防、发现、清除和管理方面内容，结合多个耐人寻味的小故事，浅显易懂地揭示了开发中面临的各种影响软件质量的问题。同时，结合软件开发以及管理实践，给出了简单、实用的模板和例子，有助于提高软件开发、项目管理和测试水平，从而达到提高开发产品质量的目的。

本书系统性、实用性和可读性较强，文中编制、搜集、列举的模板及数据对软件公司日常开发、过程改进、CMM/CMMI 评估等有很强的指导意义和实用价值。

这是一本与软件开发质量相关的指导书，也是一本多年来在软件开发管理实践中总结出来的一套系统经验的书。本书适合于开发管理人员、项目管理人员、开发人员及测试人员等任何对过程管理、软件开发和缺陷预防、清除、管理等各种实践感兴趣的人员阅读，也适合希望通过 CMMI 评估提高整体开发能力的公司和个人作为参考。同时，还可以作为高等院校计算机软件工程课程的参考教材使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

软件质量管理实践：软件缺陷预防、清除、管理实用方法 / 于波，姜艳编著. —北京：电子工业出版社，2008.11

（测试实践丛书）

ISBN 978-7-121-07420-2

I. 软… II. ①于… ②姜… III. 软件质量—质量管理 IV. TP311.5

中国版本图书馆 CIP 数据核字（2008）第 146207 号

责任编辑：葛 娜

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：28.5 字数：465 千字

印 次：2008 年 11 月第 1 次印刷

印 数：4000 册 定价：55.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

序

当今软件质量日益受到人们的高度关注。其原因有以下几方面：

首先，近年来软件逐渐成为人们社会经济活动和日常生活不可缺少的元素，所有计算机应用领域对软件质量都提出了全方位的要求，包括功能、性能、方便灵活、稳定可靠以及安全等，特别是在一些关键领域。其次，我们正面临着软件质量问题的严重挑战，包括软件质量问题引发的系统事故屡见不鲜；解决好软件质量问题存在着一些实际困难，这不仅因为软件及相关系统日益庞大和复杂，而且开发过程和软件产品不可见，需求又往往易变、多变。再者，软件测试技术至今仍不能理想地清除隐蔽的所有缺陷。

我们已经认识到软件质量问题无法只凭技术手段得以解决，重视开发的管理，加强软件过程，特别是加强对软件缺陷的管理，是取得高质量产品的必由之路。这就需要在大量的软件实践中总结一切有益的经验教训，让软件人员有所了解，进而体会其内涵和实质，并且在工作中贯彻。

于波长期从事软件开发和质量管理工作，积累了丰富的软件质量管理经验，他曾在著名软件工程专家周伯生教授指导下，负责组织和推动所在企业的软件过程改进工作，取得了显著的效果，特别是他把学习和掌握的软件质量管理知识与管理的工作实践结合起来，做了进一步的分析和总结，在此基础上编写的这本专著较为全面、系统，书中除了阐明软件缺陷及其发现、预防外，还涉及软件变更及配置管理、同行评审、软件测试及质量保证等重要问题，具有前瞻性的内容体现在软件度量等相关章节。所有这些都是当前软件企业经常遇到而又十分棘手的重要问题。

在此，我谨向从事软件管理与软件开发工作的同行们以及学习软件工程的高校学生积极推荐此书。我相信，认真地阅读它，结合自己的工作深入地思考其中的一些问题，一定会令你受益匪浅。

中国软件行业协会系统与软件过程改进分会常务副会长
清华大学教授
郑人杰
2008年10月

前 言

故事发生在 2001 年初，周一早晨，新上任的质量保证部经理召集部门的全体人员开会。当时该部门有 8 个测试人员和 5 个文档、3 个美工。对于一个刚刚成立的拥有员工 70 多人的软件公司来说，该部门人员占比达 23% 之多，其投入的规模和同行小公司相比不算小。与会过程中，就软件质量问题，大家各抒己见。

经理说：“公司自从转向软件开发以来，开发工作开展很顺利，我们质量保证部主要工作是测试，尽快、尽多、尽早地发现程序中的错误，保证开发出来的产品质量，其责任重大。”

测试工程师甲说：“缺陷就是 Bug，程序员开发完产品后，咱们测试人员尽可能多地发现其中的 Bug，遗留在产品中的 Bug 就会越少，产品质量就会越高。”

测试工程师乙说：“只要我们发现了所有的 Bug，让程序员改正过来，我们就可以放心地对用户说，我测过的软件你放心，绝对不会再有质量问题。”

……

会议在经理的主持与分配任务中告一段落。

上述对软件产品和软件产品质量的说法，从事软件研发的人可能都听到过。相隔 7 年之后，回首往事，仔细思考一下，其中类似说法有多少是完全正确的呢？软件质量真的像他们说的那样吗？

在 20 世纪 90 年代中期，几次著名的分析都得到了相同的具有普遍性的结论，即软件项目的成功率非常低。著名的观点如，软件开发仍然具有高度的不可预知性，只有大约 10% 的软件项目在最初估计的预算和进度内成功地交付；与其说管理规范是技术进步，还不如说是成功和失败的鉴别器；软件废品和返工的程度是不成熟过程的象征。

目前，随着软件工程技术的不断发展，人们已经逐渐认识到，软件质量是多方面的活动协力构建起来的，由软件开发整个过程的质量所决定，质量问题不是仅仅通过测试就能发现并解决的。大量实践证明，软件产品与传统产品有着不同的特征，如不可见性、灵活性、复杂性等，所以软件缺陷的预防自始至终是重要的，广义的软件开发质量保证活动，更多地强调软件缺陷的预防、及时发现与剔除。

因此，不像建筑工程那样精确地衡量、计算出每一元钱在软件产品中是如何花费的，软件开发过程的控制和软件质量也要远比其他工程制品的管理复杂得多。

那么，什么是“软件产品质量”呢？

Crosby 将质量定义为“符合需求”，而 Juran 认为是“适于使用”。质量的这两个定义相互关联，已经被人们广泛采纳和使用。

“符合需求”隐含着需求必须明确陈述出来，使之不被误解。在开发和生产过程中，不断进行度量以确定对这些需求的符合性。不符合需求就被视为缺陷。

“适于使用”更多地考虑了客户需求和预期的作用。由于不同的客户会以不同的方式使用产品，产品必须具有适合使用的多种要素，这些要素的每一个都包含有质量特性（适于使用的参数）。最重要的两个质量特性参数就是设计质量和符合性质量。

由于生产者坚持将“符合需求”以及将客户满意度作为质量的最终验证标准，质量的定义实际上分成了两级：狭义的质量（“小 q”），即内在产品质量，包括了缺陷率和可靠性；广义的质量（“大 Q”），则包括了产品质量、过程质量和客户满意度。

总之，软件产品质量一般都指利用适当的资源为用户提供能满足要求的产品，达到用户满意。软件质量评价要素主要包括：① 功能性；② 可靠性；③ 易用性；④ 效率；⑤ 维护性；⑥ 可移植性；⑦ 正确性；⑧ 健壮性；⑨ 完整性；⑩ 安全性；⑪ 可用性；⑫ 可理解性；⑬ 可测试性；⑭ 可再用性等。

软件质量始于开发人员本身，如果任何程序模块都存在大量缺陷的话，就很难去测试且需要花很多时间才可能集成到较大的系统中，甚至还会给用户带来很多麻烦。不良质量也会导致进度问题，有缺陷的产品交付期会较长。经过测算，大多数的软件专业人员在开发和最终测试期间，花费近一半的时间来测试并修复他们的产品。

当写一些小程序时，大多数人都会有很高的生产率；而一旦开发较大的程序时，生产率会急剧下降。虽然开发较大的系统还涉及一些额外的架构和设计工作，但大多数增加的工作量还是由缺陷引起的。随着程序的增大，平均花费在发现和修复每个缺陷上的时间会以指数级增加。然而，如果我们能够自始至终地编写高质量的模块化程序，就会生产出更好的产品并且提高生产率和组织效率。

因此，为了保证软件产品质量，需要进行以下一系列的工作。

(1) 定义质量需求，明确本产品的质量目标。要明确影响本产品质量的缺陷

可能有哪些，如何避免。

(2) 定义质量管理过程，采用软件工程的方法管理开发过程，可通过审查/评审/测试/验证等手段来贯彻落实“早预防、早检查、早控制”的保证质量策略。

(3) 建立一个能尽早面对风险的迭代式的生命周期过程。

(4) 定义每个过程成果的质量要求/质量标准，加强软件的测试。

(5) 签订合同，确定客户方相关负责人、责任以及权利，并加强和客户的沟通，了解客户真实的要求和潜在需求。

(6) 将过程建立在构架优先方法的基础上，同时设计方法要转向基于构件的开发。

(7) 编码过程中，要让员工以最佳的状态投入到工作中，最好不要加班。这是因为员工状态不佳时，编写的代码低级错误较多，虽然不致命，但浪费很多测试和调试的时间，所以加班得不偿失。

(8) 测试时要由有经验的高级程序员对业务逻辑部分进行白盒测试。

(9) 建立一个经济上具有伸缩性的可配置的过程，做好需求管理、配置管理及变更管理工作。

(10) 建立一个规范的软件工程过程，包括有效的缺陷管理、全面的计划和及时、准确的项目跟踪及报告。

本书主要内容

全书共分为四个部分：

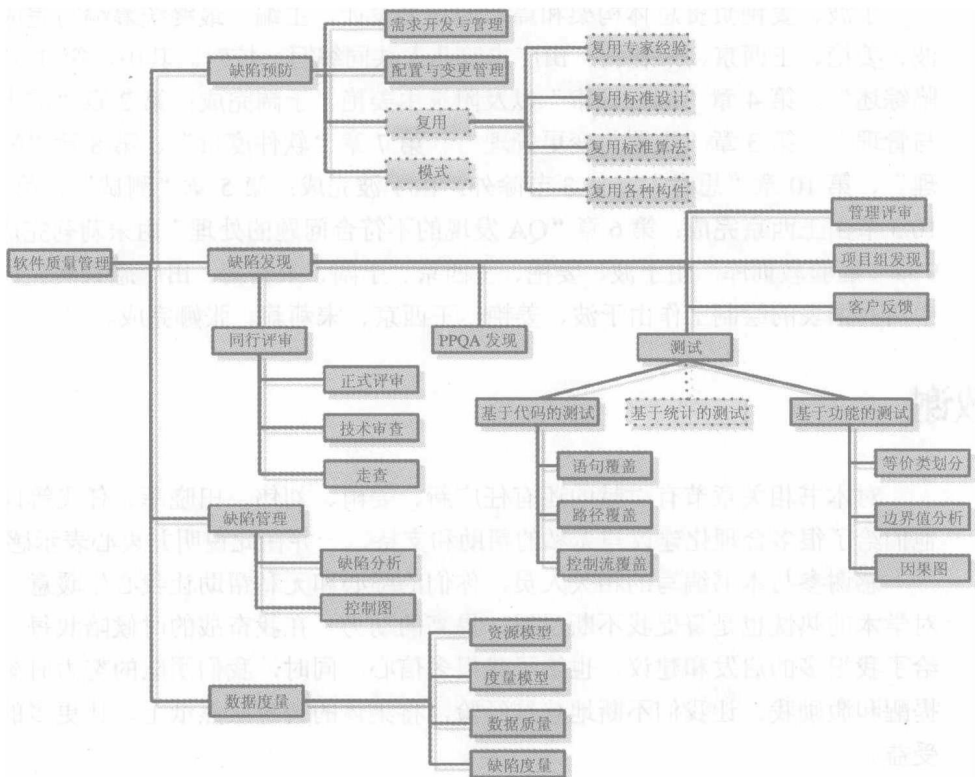
第一部分——包括缺陷综述、需求开发与管理、配置与变更管理 3 章。介绍了什么是缺陷，着重阐述了影响软件质量、造成缺陷的各种因素。

第二部分——包括同行评审、软件测试、QA 发现的不符合问题的处理 3 章。描述了发现和清除缺陷的 7 种手段中最有效的 3 种手段。

第三部分——包括软件度量和缺陷管理 2 章。阐述了缺陷的度量、分析、控制以及预防，给出了具体操作的例子。

第四部分——包括经验教训库、思考和附录。

本书的总体逻辑体现在下图中，通过缺陷预防、缺陷发现、缺陷管理和数据度量 4 个大的方面进行论述，其中的虚线部分是本书中没有描述的内容。



软件质量管理范围

本书将从软件缺陷的产生、预防、清除、管理等方面，理论结合实践地进行阐述，希望能给大家在软件开发和管理上提供借鉴。

本书内容是相关理论与实践的结晶，是我们利用业余时间总结 CMMI 过程改进中方法与经验基础上编写的，也是真实项目与产品研发过程实施的一些心得。当然，书中的内容远不止于此，适度增加了对相关经验的升华与前瞻，大大提高了知识性、丰富性、实用性、可读性。

本书编者

本书编写过程中分工如下：

于波、姜艳负责总体构架和篇章的布局设计、主编，最终统筹编写完成。于波、姜艳、王西京、宋莉莉、田广志等几人共同编写、校对。其中，第1章“缺陷综述”、第4章“同行评审”以及附录由姜艳、于阔完成；第2章“需求开发与管理”、第3章“配置与变更管理”、第7章“软件度量”、第8章“缺陷管理”、第10章“思考”（10.3节除外）由于波完成；第5章“测试”、第10章10.3节由王西京完成；第6章“QA发现的不符合问题的处理”由宋莉莉完成；第9章“经验教训库”由于波、姜艳、王西京、于阔、宋莉莉、田广志共同完成。书中所有图表的绘制工作由于波、姜艳、王西京、宋莉莉、张娜完成。

致谢

对本书相关章节有贡献的还有任广新、姜雨、刘伟、田晓菲、任飞等同志，他们给了很多合理化建议与无私的帮助和支持，一并在此说明并衷心表示感谢。

感谢参与本书编写的相关人员，你们的细心和无私帮助让我心生暖意，你们对学术的热忱也是督促我不断反思、提高的动力。在我奋战的时候陪我每一天，给了我很多的启发和建议，也给了我很多信心。同时，我们团队的努力时刻都在提醒和激励我。让我们不断地总结经验，将集体的智慧跃然纸上，让更多的读者受益。

特别感谢国际软件工程权威、北京航空航天大学教授、SEI主任评估师、北京航空航天大学软件工程研究所创始人、赛柏科技的CTO周伯生教授阅读了本书初稿，并对修改提出了专家性的意见，感谢伯生教授严谨的治学态度和渊博的理论、实践知识。

在本书涉及的内容中，我们研究、学习和借鉴了网络资料、相关博客以及很多专业书籍，特别是周伯生教授及其带领的团队提供的资料，因此，编写过程中仍然会有部分内容借鉴了同行们的思想，可能在本书中没有明确标明来源及不知您的相关声明，冒犯了您的版权，表示歉意，同时请直接与我们联系，谢谢。

于波 frankyu@21cn.com yubo@thtf.com.cn

姜艳 swallowjiang@21cn.com

目 录

第 1 章 缺陷综述..... 1

本章概括地介绍了缺陷的定义、分类、产生原因、预防以及发现手段。大量实践证明，软件产品具有不可见性、灵活性和复杂性的特征。因此，软件产品比传统产品更容易出现缺陷，提高软件产品质量的途径在于加强预防，及时发现与清除软件缺陷。

1.1 软件缺陷定义.....	2
1.2 软件缺陷生命周期.....	6
1.3 缺陷信息.....	8
1.4 软件缺陷分类.....	9
1.4.1 缺陷类型.....	9
1.4.2 缺陷严重程度.....	10
1.4.3 缺陷优先级.....	11
1.4.4 缺陷状态.....	12
1.5 缺陷产生的原因.....	12
1.5.1 缺陷是谁“生产”的.....	13
1.5.2 缺陷来源.....	14
1.5.3 缺陷根源.....	15
1.6 缺陷预防.....	16
1.6.1 缺陷预防的目的.....	17
1.6.2 缺陷预防的目标.....	18
1.6.3 缺陷预防的策略.....	18

1.6.4	缺陷预防的活动	19
1.6.5	缺陷预防的验证	21
1.6.6	软件质量特性的提高	23
1.7	缺陷发现手段	24
1.7.1	同行评审	25
1.7.2	测试	26
1.7.3	管理评审	27
1.7.4	QA 发现	27
1.7.5	项目组内部发现	28
1.7.6	客户反馈	28
1.8	缺陷修复和沟通策略	29
1.9	人员培训	32
1.10	小结	32
第 2 章	需求开发与管理	34

需求开发过程包括从情况收集、分析和评价到编写文档等一系列产生需求的复杂活动，主要有用户需求获取、需求分析、编写需求规格说明书等过程，这些活动经常是相互穿插和反复交互的。同时，该过程还与需求管理、需求验证及测试等过程密切相关。需求管理过程控制和维持了需求的约定，管理着产品和产品构件的需求，识别需求与项目计划及工作产品的不一致，从而保证了项目开发过程的一致性，这个过程使用户能够得到所需的产品。该过程从需求获取开始贯穿于整个项目生命周期，一般包括需求变更控制、需求配置管理、需求跟踪、需求状态跟踪等工作。实践中，最有效的需求管理方法是使用需求追踪矩阵（RTM）。

2.1	需求的概念和层次	38
2.2	需求开发	40
2.2.1	需求获取	40
2.2.2	需求分析	44
2.2.3	编制软件需求文档	46
2.3	需求管理	47
2.3.1	需求管理方法	47
2.3.2	需求追踪矩阵	48

2.3.3	需求变更	61
2.4	需求验证	65
2.4.1	评审需求	65
2.4.2	测试需求	69
2.4.3	需求评价标准	69
2.5	小结	70
第3章	配置与变更管理	72

软件极易发生变更，现代软件开发过程中一个巨大的挑战就是在复杂的协作环境下，通过配置管理和变更控制，保持各部分工作产品的完整性和正确性。从这点上看，缺陷的预防不再仅仅是一个配置管理的内容，更发展为高度灵活和可扩展的缺陷及变更管理系统。配置管理和变更控制的有效实践可以作为评估组织软件管理能力的标尺之一，也可以是实施软件配置管理活动的指南。通过这些有效实践，可以在很大程度上避免、预防软件缺陷的发生。

3.1	相关概念	75
3.1.1	配置项	78
3.1.2	版本	80
3.1.3	基线	81
3.1.4	配置库	83
3.2	配置管理活动	83
3.2.1	制定配置管理计划	84
3.2.2	建立三库	84
3.2.3	确定配置标识规则	87
3.2.4	版本控制	89
3.2.5	构建和发行管理	89
3.2.6	变更控制	90
3.2.7	配置审计	92
3.2.8	配置管理报告	96
3.3	变更管理活动	100
3.3.1	变更申请	101
3.3.2	变更评审	104

3.3.3	变更执行	105
3.3.4	变更验证	105
3.3.5	入库及发布	105
3.4	配置与变更管理相关问题	106
3.4.1	有关的角色和对应职责	106
3.4.2	有效实践	107
3.4.3	日构建与冒烟测试	112
3.4.4	工作环境标准	116
3.4.5	配置管理常见误区	117
3.5	小结	119
第4章	同行评审	121
<p>同行评审是由软件工作产品生产者的同行遵循已定义的规程对产品进行的技术评审。通过同行评审，开发人员能够及时得到专家的帮助和指导，加深对软件产品的理解，有利于及早和高效地从软件工作产品中识别并消除缺陷，让软件变得更易维护，同时减少最终遗漏到产品发布时的缺陷。其主要工作第一是发现工作产品中的具体错误，第二是通过对这些错误的分类和统计，发现共同的缺陷类型和修改这类缺陷的方法，避免今后类似的缺陷发生。</p>		
4.1	同行评审与测试的关系	123
4.2	同行评审的种类和对象	124
4.2.1	同行评审的种类	125
4.2.2	同行评审的对象	125
4.3	同行评审过程	126
4.3.1	正式评审流程	127
4.3.2	技术审查流程	128
4.3.3	走查流程	129
4.4	同行评审方式的选择	129
4.4.1	三种同行评审方式的比较	130
4.4.2	同行评审的结果	130
4.4.3	正式评审的特征	131
4.4.4	工作产品的同行评审方式	132

4.5	迭代生命周期的审查	133
4.6	同行评审的注意事项	134
4.6.1	同行评审遵循的原则	135
4.6.2	同行评审关注的问题	136
4.6.3	同行评审通过的准则	137
4.6.4	同行评审的经验共享	138
4.6.5	文档审查重点	139
4.7	同行评审的度量	140
4.7.1	常用度量元	140
4.7.2	同行评审的质量准则	141
4.7.3	建议的同行评审效率	142
4.7.4	同行评审覆盖率	143
4.8	评审常见问题	143
4.8.1	文化问题	144
4.8.2	准备问题	145
4.8.3	焦点问题	147
4.8.4	人员问题	148
4.8.5	效率问题	149
4.8.6	效果问题	150
4.9	小结	150
第5章 软件测试		152
<p>软件测试就是为了发现缺陷而运行程序的过程。广义的软件测试包含验证和确认，验证就是要用数据证明是否在正确地制造产品，强调的是过程的正确性；确认就是要用数据证明是否制造了正确的产品，强调结果的正确性。所以广义的软件测试指软件生命周期内所有的检查、评审和确认活动。狭义的软件测试指检查代码和文档的质量问题，努力发现问题，进行客观质量评价，测试的对象包括源程序/目标代码、各开发阶段的文档。一个好的测试用例在于能发现至今未发现的缺陷；一个成功的测试是发现了至今未发现的缺陷的测试。</p>		
5.1	软件测试的基本问题	154
5.1.1	软件测试概念	155

5.1.2	软件测试对象	155
5.1.3	软件测试目的	155
5.1.4	软件测试原则	156
5.1.5	测试过程的两个重要里程碑	156
5.1.6	测试可以发现的缺陷	157
5.1.7	软件测试的基本方法	158
5.1.8	测试工程师的技能	158
5.2	软件测试过程	159
5.2.1	单元测试	160
5.2.2	集成测试	164
5.2.3	验收测试	173
5.3	软件测试方法	173
5.3.1	功能测试	175
5.3.2	回归测试	176
5.3.3	性能评测	179
5.3.4	用户界面测试	180
5.3.5	安全性测试	180
5.3.6	安装性测试	181
5.4	测试技术专题	182
5.4.1	测试策略	182
5.4.2	手工/自动测试时机	183
5.4.3	通过二八定理寻找薄弱环节	184
5.4.4	测试用例复审	185
5.4.5	何时终止测试	186
5.4.6	Web 性能测试	188
5.4.7	内存泄漏测试	195
5.4.8	测试风险的管理	198
5.4.9	代码移交过程测试	200
5.4.10	处理不可重复出现的 Bug	202
5.5	测试的度量	204
5.6	小结	206

第 6 章 QA 发现的不符合问题的处理..... 208

QA 是指 PPQA (前置的产品和过程质量保证,侧重于事前教育,从过程上进行缺陷预防工作),与传统意义上的 QC (后置的质量控制,侧重于检查和事后发现)有着很大的不同。QA 的内容主要包括过程评价、产品和服务评价、过程指导工作。QA 工作有效性的衡量一般通过“投入的工作量占项目总体工作量的比例”、“发现的不符合问题数量”、“提供经验数量”、“对公司的有关过程和标准提出改进建议数量”等指标进行度量。

6.1 QA 流程概述.....	209
6.2 QA 的工作内容.....	211
6.2.1 QA 的角色.....	211
6.2.2 QA 工作详述.....	212
6.2.3 对 QA 职责的要求.....	216
6.3 QA 发现的问题.....	216
6.4 QA 工作机制.....	219
6.4.1 不符合项处理机制.....	219
6.4.2 QA 工作报告机制.....	220
6.4.3 问题跟踪和验证.....	223
6.4.4 QA 应遵循的原则.....	224
6.5 QA 的组织形式.....	225
6.6 对 QA 的误解.....	226
6.7 QA 工作的度量.....	228
6.8 小结.....	229

第 7 章 软件度量..... 231

软件度量是针对软件开发项目、过程及产品进行数据定义、收集以及分析的持续性量化的过程。有效度量的作用在于能够帮助软件组织认清自身的能力,理解、评价、控制、预测和改进软件工作产品或软件过程。根据对度量数据结果的分析,进一步为它们的生产和服务制订出可行的计划;及时找到变化趋势,预测问题,发现或者采取有效手段预防缺陷;不断改进软件开发过程。软件度量活动一般从项目级开始,逐步向上扩展为过程度量和产品度量,向下扩展为个体行为度量。软件度量中关键的内容是度量模型的建立和资源模型曲线的绘制及应用。

7.1 软件度量及其方针.....	233
-------------------	-----

7.2	度量活动	235
7.2.1	度量目标	236
7.2.2	度量元	238
7.2.3	度量模型	240
7.2.4	基本过程	251
7.2.5	度量方法与采集	252
7.3	资源模型	257
7.3.1	资源模型的定义	258
7.3.2	项目级资源模型	260
7.3.3	组织级资源模型	262
7.3.4	软件质量度量	263
7.4	数据质量	265
7.4.1	数据的真实性	265
7.4.2	数据的同步性	266
7.4.3	数据的有效性	266
7.4.4	数据的一致性	266
7.5	软件度量相关问题	267
7.5.1	增加度量正确性的措施	268
7.5.2	软件过程性能	268
7.5.3	度量过程的常见问题	271
7.6	缺陷度量	272
7.6.1	什么是缺陷度量	272
7.6.2	缺陷度量元	273
7.6.3	缺陷密度的定义	274
7.6.4	缺陷密度的用途	275
7.6.5	缺陷管理库	277
7.7	缺陷分析	278
7.7.1	缺陷种类分析	279
7.7.2	缺陷根源分析	281
7.7.3	缺陷注入-发现矩阵	281
7.7.4	收敛趋势分析	283