教育部职业教育与成人教育司推荐教材
国外优秀教材精选

软件的质量

软件的分析、测试与验证

Software-Qualität

Testen , Analysieren und Verifizieren Von Software

（德）Peter Liggesmeyer 著
于芳 译

机械工业出版社
CHINA MACHINE PRESS

教育部职业教育与成人教育司推荐教材

国外优秀教材精选

软件的质量

软件的分析、测试与验证

Software-Qualität Testen, Analysieren und Verifizieren Von Software

（德）Peter Liggesmeyer 著

于芳 译

机械工业出版社

本书详细地叙述了当前软件质量保证的技术、方法、原理和构成方面的最新知识，每一章的开始是本章的简介，使读者了解本章要讲述的内容；每一章的结束部分都有评价，对从业人员来说具有实践指导意义。本书在写作上，注重了将理论和实践、软件和硬件、经验知识和教学知识领域的牢固结合，使读者更贴近软件开发这个重要的领域。

本书适合作为高等教育计算机相关专业的教材和教学参考书，也可用作从业人员的参考书。

Liggesmeyer, Peter:
Software-Qualität: Testen, Analysieren und Verifizieren Von Software
ISBN 3-8274-1118-1

图书在版编目（CIP）数据

软件的质量 软件的分析、测试与验证/（德）里格斯麦尔（Liggesmeyer, P.）著；于芳译. —北京：机械工业出版社，2009.6
教育部职业教育与成人教育司推荐教材. 国外优秀教材精选
ISBN 978-7-111-26981-6

Ⅰ. 软… Ⅱ. ①里…②于… Ⅲ. ①软件质量-质量管理-成人教育：高等教育-教材②软件-测试-成人教育：高等教育-教材 Ⅳ. TP311.5

中国版本图书馆 CIP 数据核字（2009）第 068025 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：孔熹峻 蔡 岩 责任编辑：蔡 岩 责任校对：姜 婷
封面设计：鞠 杨 责任印制：杨 曦
北京富生印刷厂印刷
2009 年 7 月第 1 版第 1 次印刷
184mm×260mm · 18.5 印张 446 千字
0001—3000 册
标准书号：ISBN 978-7-111-26981-6
定价：42.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
本社服务邮箱：marketing@mail.machineinfo.gov.cn
投稿热线电话：（010）88379194
编辑热线电话：（010）88379934
投稿邮箱：Kongxijun@163.com
封面无防伪标均为盗版

出版说明

随着计算机和软件深入到生活的各个领域，软件的质量问题始终是需求客户满意与否的关键所在，同时它也是产品是否成功的一个关键性因素。软件的质量在社会生活中已涉及到多个方面。目前，我国各高校均开设有软件方面的课程，因此软件技术也已成为高等教育重要的组成部分。基于此，本着借鉴国外教育的先进经验，为我所用，为我所学，尽快缩小与国外在软件领域技术和教学方面的差距的目的，我们引进并翻译了这本国外优秀教材并推荐给广大读者。

本书根据德国教授 Peter Liggesmeyer 的著作《Softeware-Qualität》第 1 版译出。原书共 14 章，详细介绍了当前软件质量方面的最新技术方法和最新知识状况，适合作为高等教育计算机相关专业的教材和教学参考书。全书本着方便读者阅读的目的，进行了结构编排，便于读者学习和理解所学知识。

全书由北京外国语大学的于芳老师翻译，长安大学的席晓慧老师对本书进行了认真的审阅和修改，并提出了很多的宝贵意见。在此对译者、审阅者的辛勤劳动表示由衷的感谢。限于在专业及文字方面的理解偏差，书中错漏之处在所难免，恳请读者批评指正。

机械工业出版社

前　言

计算机和软件深入到生活的各个领域，比如进行汇款、控制汽车发动机、监控病人、控制飞机飞行，并在网络上提供人们所需的信息。如果软件质量有问题，在某些应用范围会造成危害。软件质量始终是使用者满意与否的关键所在，对产品成功也十分重要。但软件质量也是一个很难的课题：既不是指“特定的”软件质量，也不是指“特定的”软件产品。软件的质量涉及多个方面。对于一种产品质量特别重要的质量性能组，可能对另一种产品毫无意义。控制飞机飞行的软件必须保证运行安全，排除一切危险的可能，可这对银行所需的记账软件来说微不足道，因为银行软件不会带来人身危险。不过，记账软件必须确保记账过程中不能造假，又要防止资料被窃取。

本书详细地叙述了当前软件质量保证的技术、方法、原理和构成方面的最新知识状况。编写此书的目的是为了让读者更贴近软件开发这个重要的领域。本书适合作为讲师和大学生的教学材料，也可作为从业人员的参考书。每一章的开始都有该章的简介，为读者指明方向。每一章的结束部分附有评价和核对表。核对表对从业人员来说具有实践指导意义。

完成本书耗费了很多时间，也占用了很多业余时间。我对妻子 Petra 和儿子 Alexander 表示歉意，并感谢他们的理解。特别要感谢 Katrin Augustin 女士，她认真专注地撰写了书中文章、绘制了图形、补充校正，并进行了书刊排版。对于书中的注解和校正说明，我要感谢巴黎 Alcatel 的 Christof Ebert 博士，多特蒙德大学信息科学的 Eike Hagen Riedemann 教授，柏林理工大学软件技术专业的 Thomas Santen 博士，哈根函授大学实用信息学的 Mario Winter 博士，以及斯图加特大学软件工程系的主任、教授 Jochen Ludewig 博士。我也感谢我的合作者 Joerg Gericke 工程学士、Lars Grunske 工程学士、Bernhard Kaiser 工程学士、Roland Neumann 工程学士和 Christopher Robinson-Mallett 工程学士，他们进行了本书的校对工作。尤其要感谢出版社的领导兼海德堡光谱出版社的信息程序设计人员 Andreas Ruedinger 先生，感谢他为了将我的想法变成此书而积极的工作，以及对本书内容和设计上的建议。

本书的每一章都由不同人士多次阅读和修改。然而，这样的质量保证也无法完全排除软件与书籍的一点共性：尽管付出辛勤劳动，仍旧有一些错误存在，而这些错误只能在“使用中”才会被发现。如果您在阅读本书时发现错误，请通知我，以便在 http://www.liggesmeyer.de/集中。同样，也欢迎您随时反馈鼓励、批评和表扬的意见。

我希望您在阅读本书后能获取新的知识，并对实践有所帮助。如果我能够把通常很“枯燥的”课题“软件的质量保证”以吸引人的、有趣的内容呈现在您面前，能将理论和实践、软件和硬件、经验知识和数学知识领域牢固结合，将不胜荣幸。

Peter Liggesmeyer

目　录

第1章 引 言

每个开发软件的企业都努力提供最佳质量的软件。只有精确定义了一个目标，我们才能切实地实现这个目标，但这个道理却并不适用于“最佳质量”的概念。软件质量是涉及多方面的。一个软件的多种性能共同构成软件的质量。这些性能对使用者和制造商而言并非同等重要。一些性能对特定的软件产品特别重要，其他性能对相同的软件产品则毫无关系。一些性能相互发生负面作用。有人说，我们要做出质量最好的软件，显然是没有理解软件质量的含义。开发软件的目的不在于实现最好的质量，而是最合适的质量。为此，需要确定所谓的质量目标来制定所需的软件质量。随后，人们可以决定，采用何种方式来达到确定的质量。一般来说，必须采用设计上有前瞻性、经过分析检验的技术与组织管理方面的手段相结合。在实现质量的过程中，重要的是考虑到经济性，即一定不要忘记时间和成本两个要素。不同软件产品的多样性，造成了对软件质量的不同要求，加上时间和成本，导致人们无法找出万能的解决方案。本章将分门别类地为您介绍软件质量管理和软件质量保证的各种组织上和技术上的解决方案。

1.1 动机

随着计算机越来越深入人类生活的各个应用领域，其软件功能的正确性、可靠性也越来越重要。成本的发展表明，和硬件成本相比，软件成本明显呈上升趋势，同时，软件使用寿命也明显长于硬件。所以，在软件开发领域缩减成本是十分经济的做法。如果根据软件生命周期来分析软件开发的成本，则其结果是市场上某个软件产品的大部分成本在维护阶段产生，也就表明软件质量不够理想。在制作软件时产生的错误，和使用软件时发现的错误是导致软件质量不完善的原因。如果一个软件产品结构不清晰，内容不够简洁明了，要修正错误则是一件耗费时间的重任。软件产品日渐增强的复杂性同样也增加了软件质量不完善的可能性。要将错误本地化，并消除错误很难，特别是结构上有欠缺的软件，修正的错误可能会带来其他错误，因为对某一处的修改会与软件的其他部分相互作用。如果某个错误的形成原因已经在软件开发的早期阶段出现，比如在定义要求的时候或者设计软件的时候，那么必须进行大量的变动。

超高的维护成本以及与之紧密相关的人员培训，可以用适当的手段解决。人员培训不是在所有地方都要进行。有一些方法可以制作更可靠、更易懂、更容易更改的软件。这些方法是设计性的或者分析性的。在实践中流传最广的主要是功能分解性的、面向对象型的软件开发方式。例如结构化分析（Structured Analysis)/DeMarco 85/、结构化设计（Structured Design）和统一建模语言（UML)/Booch et al. 98/。在/Balzert 00/中有对软件开发方式的详细介绍。

没有任何设计过程能保证制作出完美无瑕的产品。所以，要用分析手段来测试产品质量。如果软件开发过程被分为几个阶段，那么在每个阶段都能采用设计性方法，并在每个阶段末尾，用分析手段对已形成的中间产品质量进行评价，如图 1-1 所示。人们把这种做法称为整体质量保证原则。一个质量尚不完善的中间产品，将无法进入下一开发阶段，直到达到已定义的、充分的质量程度为止。

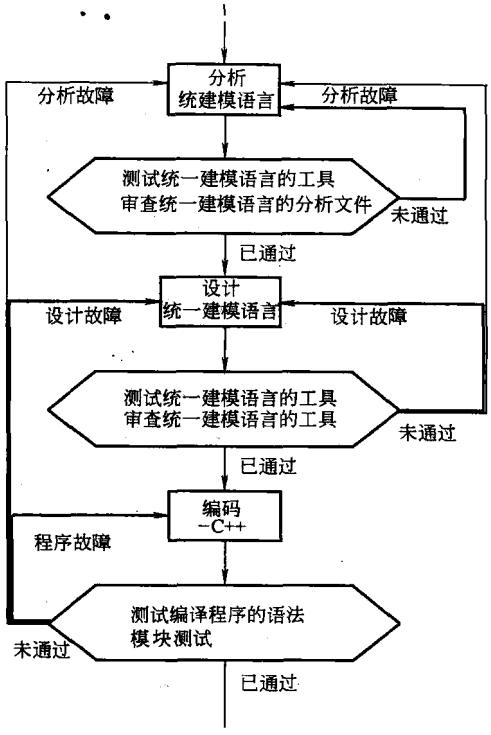

图 1-1　整体质量保证的原则

这样的操作方式能够尽早识别错误，并能花费较少的时段间来正错误。其目的在于，在测试中就尽量发现设计步骤中形成的错误。设计步骤在测试之前进行。分散在各个步骤中产生的错误，应尽量减少。质量保证和第一道中间产品共同作用，以发现早期的质量偏差，并为消除偏差采取措施。仅仅测试一个“已制作完成的”程序远远不够。在设计性质量保证和分析性质量保证之间存在着相互作用。在理想状态下，一个设计性活动之后应该跟随一个相应的分析性活动。在软件开发过程中也能将设计步骤和检验步骤紧密结合，但没有任何单独的过程规定要结合（比如说极限编程）。

在初期确定要实现的质量有至关重要的意义。在开发软件的早期阶段确定质量参数并确定所需的参数值，以此来确定质量目标。软件的测试在软件开发过程中处于中心地位。每个程序员打开一个已制成的程序，输入一些内容，查看产生的结果，并将其与自己的预期进行比较。

程序员还会仔细查看程序编码，以便发现错误。基本上，第一个操作步骤是应用一种非系统化的动态测试流程；第二个步骤是人工执行非系统化的静态分析。一方面，测试技术以简单的形式得到应用。另一方面，对不同的流程、流程的系统性及其功能仍旧存在不清楚的地方。测试的目标必须是在清晰定义了执行步骤之后，制造出能复制的产品。但很遗憾，在实践中无法实现这个目标。

在本书中讨论了关于解析软件质量保证的技术、工具和测试流程，其中的重点为测试技术。除了动态测试以外，书中也介绍了静态分析和符号技术。面向对象型分析技术、设计技术和编程技术日益深入生活实际，随着其重要性的增强，面向对象型软件测试过程也清晰地表现出来。测试嵌入式软件时必须注意经常存在的安全要求和稳定性要求，该应用领域的技术将在某一章节中详细描述。

由于并非所有的技术都需要工具的支持。检查技术、复审技术明显不需要利用工具。而许多技术如果没有有效工具的支持，又根本无法采用。要让技术和工具各司其职，需要

能够调节流程步骤和责任性的结构框架。在这个意义上出现了各种处理方法。关于测试的组织，书中也进行了讨论。

随着软件不断深入技术系统内部。不管是对硬件而言，还是对软件而言，都存在系统化的开发技术和质量保证技术。硬件质量保证技术通常带来量化的结果，而软件无法有同等程度的表现。开发技术软件密集型的系统时，一般地区分硬件质量保证和软件部分质量保证还不够，因为系统制造商必须给定并保证整个系统的属性。但有一点适用：整体的各个部分之和大于整体本身。硬件质量保证和软件密集型系统的质量保证，在书中也有相关讨论内容。

1.2 数据和概念的定义

下面将定义几个非常重要的基础概念。本书还包括大量的其他定义，分别出现于各个章节中。此处定义如下几个概念。

- 质量，质量要求，质量参数，质量测度
- 故障，缺陷，错误
- 正确性
- 完整性
- 安全性
- 稳定性
- 可用性
- 坚固性

（1）质量　质量的概念已在/德国工业标准 DIN 55350-11 95/中定义为某个用户满足需求或隐含需求（质量要求）的特性和适用性。

（2）质量要求　质量要求的概念指对某个用户的各个要求的总和，与该用户的特性有关。

（3）质量参数　通过所谓的质量参数来具体确定质量，并以此判断质量。质量参数是一个功能单位的各种属性，有了这些属性，质量才得以被描述、被评估，但这些参数并不包含与参数值相关的内容。一个质量参数可以在几个阶段中细分为局部参数。对于质量参数存在许多建议，分别由不同出版物的作者们提出（比如/Balzert 98/，/Pagel, Six 94/）。

例如，质量属性安全性、稳定性、可用性、坚固性、保存效率和运行时间效率、可维护、可移植、可测试、可使用。其中质量参数可修改、可移动、可测试对产品制造商很重要，而安全性、稳定性、适用性、保存效率和运行时间效率对使用者比较重要。所以，对制造商来说，后者也与客户的满意程度相关。ISO/IEC-标准9126 /ISO IEC 9126 91/详细描述了几个质量参数。一般说来，控制质量参数正确性的技术，与检查质量参数如安全性或者可靠性所采用的技术相互不同。在一些出版物中可以看到对功能性参数和非功能性参数的划分。各个质量参数之间也存在相互作用和联系。安全性和可用性常常是相互矛盾的优化目标。比如，安全的系统能通过冗余识别组成部分的故障，并能过渡到一种安全状态。此时，该系统的普通功能未产生，也降低了系统的可用性。安全的系统（如双信道的 2-aus-2 系统）一般比传统的单信道结构可用性低。软件可以从保存效率方面，即尽量

少的保存需求或者从运行时间效率方面，即尽量少的运行时间得到优化。一个质量参数的改善经常会导致其他质量参数的恶化。所以，在每个方面都期求获得最佳质量毫无意义，而且，由于质量参数之间的相互作用，面面俱到的最佳质量也是不可能实现的，必须在开发阶段初期就确定各个质量参数要达到的值，即确定质量目标。

（4）质量测度　一个质量参数的具体值通过质量测度来确定。这些测度能推断出某个质量参数的值。例如，质量参数稳定性的质量测度可以是 MTTF（平均失败时间）。

（5）故障　在使用产品过程中频繁出现错误（Failure）。在软件的动态测试中，人们识别的不是错误，而是故障。故障是程序中的错误带来的结果。

（6）缺陷　软件的缺陷是静态存在于软件编码中的故障原因。

（7）错误　缺陷的原因可能是程序员的错误，也可能是一个简单的输入错误。错误地理解当前使用的编程语言的某种结构，也可能构成错误。错误、缺陷和故障经常相互关联。因此，错误→错误理解编程语言的某种结构→导致程序的指令有误，在执行该指令时出现另一个结果→故障，运行的结果不是我们想要的结果。

（8）正确性　/Endres 77/对质量参数正确性的定义是毫无错误的同意词，我们观察到的结果和所预期的结果完全一致：……一个程序是正确的，如果它完全没有逻辑错误，即执行该程序的结果是该程序应有的结果。IEEE 建议/ANSI 729 83/定义中的正确性是具体说明程序之间的持续性及该程序满足使用者期待值的程度。此类定义都是以实践为导向的，但是要确定正确性的测试标准，它们并不是最理想的方法，因为它们的定义本身就比较模糊。比如，程序满足使用者期待值的程度很大程度上取决于选择正确的受访人群。

偶尔使用一个系统的使用者，提出的要求和期望与经常使用该系统的专业使用者不同。专业领域对正确性这个概念的定义如下。

1）正确性没有程度深浅的特征，也就是说，一个软件要么是正确的，要么是错误的。

2）一个没有错误的软件是正确的。

3）如果一个软件与自己的说明保持一致，那么这个软件是正确的。

4）如果一个软件没有详细说明，那么便无法判断该软件是否正确。

根据这个定义，只要程序与其说明有丝毫不符，那就是不正确的，一方面，可以由此得出，在某种程度以上几乎不存在正确的程序。另一方面，在实际中人们很普遍地将缺陷级别作为标准，区分可容忍的和不可容忍的缺陷。由此，可以将正确性的概念应用到不同的缺陷级别，例如：如果程序不再包含最高级别的缺陷，该程序就是正确的。

（9）完整性　如果一个软件实现了其说明中的所有功能，那么它就是完整的。不管是处理普通情况还是有缺陷的情况都适用这个概念。

（10）安全性　德国标准 EN ISO 8402/德国标准 EN ISO 8402 95/定义安全性为一种状态。人员伤亡或者物品损坏在这种状态里都被限定在一个可以接受的值上。/Birolini 97/定义安全性是一种能力度量，这种被观察单元既不伤害人或物、也不破坏环境。我们应区分无故障系统的安全性（防止意外）和有故障系统的技术安全。系统不存在任何损害和损坏情况，称之为安全。

此外，还有一个特征，可以翻译为数据安全，包括系统属性、阻止信息流失、只允许被授权的人进入数据。

（11）稳定性　稳定性有多个标准化的定义，在专业文献中也有大量定义。德国标准40041/德国标准40041 90/对稳定性的定义是，在规定的条件下，在规定的时间内或者之后，单位地表现质量的一部分。德国标准EN ISO 8402/德国标准EN ISO 8402 95/的定义是，稳定性是描述可用性及其影响因素等性能的总称：可用性、可维护和维护支持。在德国标准ISO 9000第四部分/德国标准ISO 9000—4 94/中，稳定性是一个单位在给定的应用条件下，在规定的时间范围内或者之后，满足稳定性要求的属性，该定义符合德国标准40041/德国标准40041 90/的定义。在英语国家，对应广义的稳定性概念的词是相依性Dependability。/Birolini 97/使用的是受到限定的稳定性概念。这个稳定性在狭义上是可靠度Reliability的意思，指一个被观察的单元保持操作功能的能力，其表现就是，在特定的应用条件下，在规定的时间范围之内，能毫无故障的运行。这个定义允许人们借助随机技术来描述稳定性。这样，到系统发生故障之间的时间，即所谓的平均失败时间（MTTF），可以给定预期值，并判断稳定程度。可维护的系统中，人们使用两次相邻的故障之间的预期值，即所谓的平均故障间隔时间（MTBF）。另外，还可以选用生存几率$R(t)$，它指一个系统从时间点零开始运行，到时间点t仍然完好无损。下文中出现的稳定性都是以最后一个狭义的定义为标准的。稳定性指被观察的单元在一定时间范围内以一定的方式运行，并且保持无故障运行的可能性。它是可以随机描述的属性。稳定性一般来说不是常数，而是在软件修正故障、稳固性增强时增大；在软件功能扩展后，因为新故障的出现，稳定性变小。在硬件的各个部件中，需要注意的是磨损带来的后果。

另外，观察得到的稳定性由系统的使用方式确定。我们经常可以用静态流程做出稳定性诊断。

（12）可用性　可用性是对能力的度量，度量一个观察单元在某个给定时间保持操作功能的能力。可用性表现为，观察单元在特定时间、特定的工作条件下执行所要求的功能的几率。可用的度量可以用

$$\frac{\text{平均故障间隔时间}}{\text{平均故障间隔时间}+\text{平均修复时间}}$$

来表示。商的值可以通过两种方式增加。可以增大平均故障间隔时间的值，也就是增加稳定性，或者减少平均修复时间，即缩短故障后的停顿时间。增加平均故障间隔时间同时影响稳定性时，减少平均修复时间只对可用性有影响。

（13）坚固性　坚固性在/Liggesmeyer，Rothfelder 98/中的定义是，一个观察单元在异样的环境下有选择的工作并执行重要反应的属性。根据详细说明，一个正确的观察单位拥有的坚固性可能很弱。如果一个系统能正确实现详细说明中描述的各种功能，那么这个系统根据上面的定义就应该是正确的。如果发生了详细说明没有涉及到的情况，则此时系统故障是被允许的，该系统仍旧是正确的。典型的就是类似数据输入错误或者数据相互矛盾的运行情况。坚固性与其说是详细说明的实现，不如说是详细说明的属性。一个坚固的系统是正确执行详细说明的结果，这些详细说明即使是在异常运行条件下也要求系统执行特定动作。一般来说，人们不可能在制定详细说明时就能考虑到所有方面，因为坚固性也是有程度的，所以提高系统的坚固性会增加开发成本，意识到这一点很重要。制定详细说明、实现详细说明以及测试的成本都会额外增加。

出于这个原因，应该在开发阶段的早期确定所需的坚固性，通过制定详细说明来确定。举一个例子，如果在软件开发中没有这样做，程序员将会对故障处理规定相应的程序编码。有时候被视作是良好的、防御性的编程风格绝不是受欢迎的操作方式。一方面，这种工作方式从经验上讲会产生大量冗余的程序编码，因为故障询问被多次执行。另一个方面，无法保证，真正重要的、需要捕捉到的情况是否真的被截取到了。

1.3 技术水平

1.3.1 质量管理

质量管理指的是达到并证明高级质量的组织措施。

德国标准 ISO 8402/德国标准 EN ISO 8402 95/对全面质量管理（TQM）的定义是一种组织以全体成员共同作用为基础的领导方法。这个组织以质量为中心，并通过客户的满意度追求长期的商业成功，追求各个成员及社会谋求的利益。全面质量管理和传统的质量保证明显不同，见表 1-1，拥有下列先锋代表。

（1）零缺陷的观念（Zero Defects Concept） 此观念是由 P. B. Crosby/Crosby 79/提出的观念，其出发点是只有无缺陷的产品才能被接受。目标是没有不合格品、不需要补充工作的无缺陷产品："产生成本的不是因为质量被制造出，而是其要求不能被满足。"

（2）持续改善过程（CIP） 持续改善是由 W. E. Deming/Deming 86/五十年代引入日本工业的观念，革新了生产率和质量。它包括了持续改善（Kaizen）和一个十四点观念（管理原则），持续改善借助德明周期来实现（PDCA 循环）。

表 1-1 全面质量管理和传统质量保证的比较

	传统的质量保证	全面质量管理
目标	更好的产品 更少的成本	更好的企业 客户满意 灵活性
导向	产品	市场 过程
组织	质量保证的强势	一切都以质量为中心
质量责任	质量专员	线性管理 每个工作人员
方法	测量 控制 记录并评估缺陷	用机制化的方案来减少缺陷 监督过程并优化过程 在自己的领域内优化

全面质量管理（Feigenbaum，1961）是集开发、维护和改进营销质量、开发质量、生产质量和客户服务质量为一体的系统/Feigenbaum 83/。

全公司的质量管理（石川馨 Ishikawa）的概念是对全面质量管理的扩充，扩充了工作人员定向这个组成部分。石川馨是质量循环和鱼骨图（石川馨图）的发明者。

J. M. Juran/Juran 64/中的质量三部曲分三个级别，有系统的持续提升质量（计划过

程，执行和保证，过程改进）。

全面质量管理是文献中管理技术的重点。此外，全面质量管理/缺陷 93/，/Kaminske，Brauer 93/还包括正确执行质量保证的大量技术，比如：

- 品质机能展开（QFD）
- 统计过程控制（SPC）
- 稳定性建模
- 复审，验证
- 质量循环
- 分析故障可能和影响
- 鱼骨图（石川馨图）
- 佩瑞多分析
- 相互关图

1. 品质机能展开（QFD）

品质机能展开是集中和权衡客户要求的一项技术。其作用是全面突出客户的要求，并根据这些要求对客户利益和商业成功的意义大小进行权衡。另外，可以通过开发过程追踪客户要求。因此，哪些活动与哪些客户要求相关，都一清二楚。所以，可以从瓶颈状态到满足重要要求的情况中都能利用资源。品质机能展开符合全面质量管理强烈的客户导向。基本上，所有的决定都可以追溯到按客户要求的开发过程。

2. 统计过程控制（SPC）

统计过程控制能区分一个稳定过程中的偶然控制过程标记数字与系统性的更改过程的做法/Braverman 81/，/Wheeler，Chambers 92/。偶然控制的例子就是一个铣切部件的制造公差，而系统性的更改是指工具磨损或者夹紧错误的铣头造成度量的缓慢偏移。统计过程控制是一种制造技术，在开发中测量方面的应用能力有限。

统计过程控制可以利用统计学的一些辅助手段。它借助所谓的质量管制卡来进行，如图 1-2 所示。通过测量值得到干预的上限（UCL：管制上限）和下限（LCL：管制下限），用来评估质量管制卡中的测量值。超过上限和不及下限的测量值可能会给系统造成不良影响。全面质量管理中，统计过程控制是评价过程中的变化的一种控制手段。

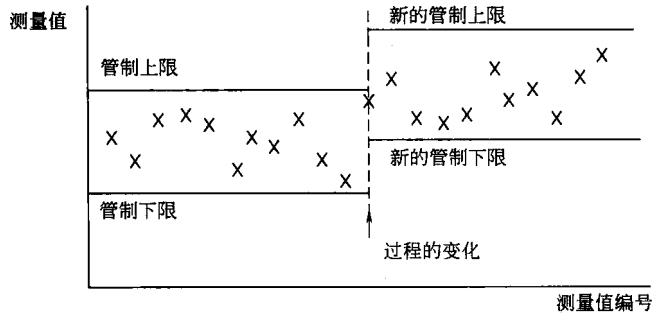

图 1-2　质量管制卡

3. 质量循环

质量循环是所有工作人员都参与到全面质量管理中的一个重要措施。质量管理循环指一个工作小组的成员定期实现目标，解决工作上质量问题和采用积极改进措施。典型做法是每周在工作时间内抽一小时开会，通常要小组批准后才进行改善，结果管制也是一样。质量循环应该采用合适的流程来识别、分析和解决问题（比如佩瑞多图、石川馨分析、

头脑风暴）。来自上级管理部门的支持和参与也至关重要。质量循环的典型工作方式由下列步骤组成。

1）识别问题，选择问题。

① 选择要研究的问题。

② 使用创新技术识别问题。

③ 设置位置的优先地位（比如用佩瑞多图）。

2）处理问题。

① 决策批准。

② 与其他质量循环配合。

③ 分开主要原因和次要原因（比如用石川馨分析）。

④ 确立目标。

⑤ 寻找解决方案（比如通过头脑风暴）。

⑥ 评估替代性方案，选择解决办法。

3）展示结果。

向决策人员展示解决办法，并准备着手解决。

4）执行和结果管制。

① 执行解决办法。

② 记录问题，解决办法和结果。

③ 归纳（转到其他组织部门）。

4. 因果图（鱼骨图，石川馨图）

石川馨的因果图是分析因果联系的图像技术，如图 1-3 所示。对于要研究的问题（效应），首先找出主要原因，然后再将主要原因细化为次要原因。这个流程是石川馨最先应用到质量循环中的（石川馨图）。效应置于鱼骨头的头部。鱼脊上记录主要原因。应用 6M 方法很有帮助，能检查原因是否属于下列某个类别：人、机器、方法、材料、环境、测量。更进一步的原因记录到从鱼脊衍生的鱼骨头上。这个图保留了 6 问（6W）方法：什么、为什么、怎么样、谁、什么时候、哪里。集中了可能存在的原因以后必须查明确切的原因。我们必须评估可选择的解决方法，选出最佳方案来解决问题。

5. 佩瑞多分析

佩瑞多原则表明，20% 的故障原因造成了 80% 的故障。佩瑞多分析利用条形图（直方图）进行，柱形图将各个故障情况根据其重要程度（比如成本，后果的严重性）从左到右排列出来。此外，直方图上还可以添上数字曲线。根据图 1-4 的示范，每个条形都代表一种故障类型。

条形的高度代表消除故障所需的成本总值。在这里，佩瑞多原则就意味着“20% 的故障造成 80% 的成本”必须首先避免这些占 20% 的故障。它们是质量改善的目标。

6. 相关图

相关图是分析两个参数之间相互依赖性的简单工具，它以大量参数对为基础。关联系数是相关图的统计学基础。图 1-5 展示的图例为软件模型数量和开发成本之间的关系、测试数量和产品投入使用第一年中已报告的故障数量之间的关系。