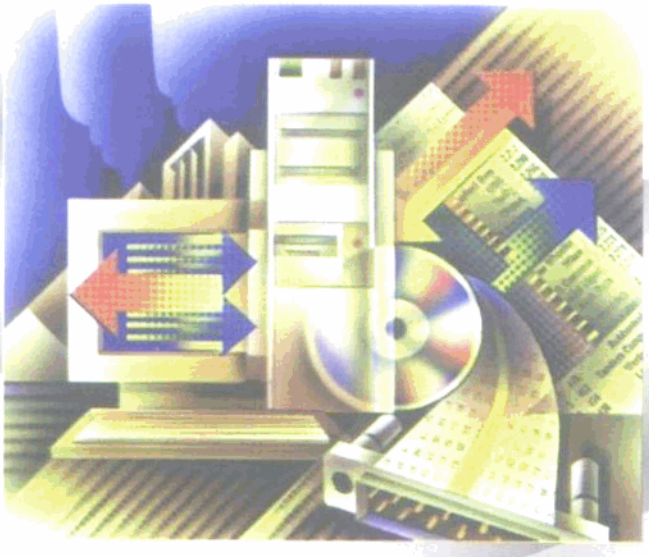
计算机软件质量保证的方法和实践

罗圣仪 编著



科学出版社

计算机软件质量保证的方法和实践

罗圣仪 编著

科学出版社
1999

内 容 简 介

软件质量保证是软件工程学科的一部分，它在软件的生存周期中，有计划地、系统地应用软件工程的方法来处理软件的质量问题。本书着重介绍软件质量保证的基本概念、软件质量保证的工程方法、软件质量的测试与评价、软件质量保证的标准与规范，以及在工程中软件质量保证系统的建立与实践。

本书主要供从事软件开发和维护的计算机专业人员参考，也可供大专院校计算机、自动化、通信等专业的教师和学生学习和参考。对于计算机软件公司的经理及负责软件质量保证部门的领导也是一本十分有用的参考书。

图书在版编目(CIP)数据

计算机软件质量保证的方法和实践/罗圣仪编著.-北京：科学出版社，1999

ISBN 7-03-007425-4

Ⅰ.计… Ⅱ.罗… Ⅲ.软件-质量管理体系 Ⅳ.TP31

中国版本图书馆CIP数据核字(1999)第07248号

科学出版社出版

北京东黄城根北街16号

邮政编码：100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1999年9月第一版 开本：787×1092 1/16

1999年9月第一次印刷 印张：12

印数：1—2 500 字数：267 000

定价：20.00元

（如有印装质量问题，我社负责调换〈环伟〉）

前　言

软件质量保证是软件工程学科的一部分，它在软件的生存周期中，有计划地、系统地应用软件工程的方法来处理软件质量问题。软件质量保证系统是一个完整的工作过程，包括制定策略，选用各种方法、工具、人员和技术资源，对软件生存周期中的各项活动进行调整，以确保和证明软件产品及其维护支持的质量。

自从国际社会进入信息时代以来，计算机软件逐渐发展成为信息产业的核心。计算机正在嵌入到我们生活中的各个角落，其影响不断增长，在各行各业中，没有计算机将是不能想象的。软件的重要作用愈来愈突出，软件主导着信息产品的开发和信息技术市场的开拓，已经成为实现国民经济信息化和社会信息化的战略性产业。对于许多基础研究，工业和国防的关键技术，如：军事系统、运输系统、贸易管理部门、商业部门、工业生产，以及国家的宏观经济管理决策，软件都是必不可少的基础工具。软件产业附加价值高，发展非常迅速，每年约以20%的速度增长。1998年全球软件产值超过千亿美元，预计到2000年，软件的年销售额将达到2 000亿美元。

在当前软件市场已扩展为全球市场的形势下，任何软件生产单位，如果不能生产出高质量的软件产品，将无法取得成功，也不能生存。软件与硬件不同，其开发过程的能见度差，难于测量和控制，因此，软件产业是高风险的高技术产业。

大型的、复杂的软件产品的开发常常不能按照预定的投资和期限完成，尤其是必要的可靠性、安全性得不到保证。因此，在软件系统的重要性日渐增长的同时，由于软件系统失败造成巨大损失和严重后果的例子也愈来愈多。例如，1979年第一颗金星探测器未射中目标，仅仅是由于在FORTRAN程序中错误地用了一个句号代替一个逗号，结果损失了数亿美元。1983年美国F18轰炸机在进行新的飞行器软件实验时，飞机穿越赤道时失事，其原因是程序中的时标发生错误。1984年，法国南部湖峡谷的洪水泛滥，是因为计算机自动锁定系统没有辨认出洪水的危险信息。

从软件工程的观点来看，当前软件开发的现状是不能令人满意的，具体现象如下：

1）经常是缺乏系统的开发过程，而只根据口头下达的指令来执行任务。

2）管理者和软件开发人员往往低估了软件开发的复杂性。

3）在软件的开发过程中，缺乏系统的规划，通常是临时拼凑的计划。

4）软件产品的需求不能完全确定，需求的定义往往是模糊的、混乱的、矛盾的。

5）缺少甚至没有相应的质量保证测试，通常只是采用非系统化的测试。

6）软件项目的组织管理不能满足需要。

7）对于文档这个关系质量的重要部分，通常没有或者质量极差。

8）软件的开发成本和时间也不能满足规定的要求，目前，只有约5%的项目是准时完成的，20%到60%的项目超过规定时间。而时间的拖延经常会导致项目的终止或失败。软件开发成本随着软件的复杂性成指数倍地增长，在很多情况下，60%以上的成本花在产品的维护上。目前，软件产品的错误和缺陷往往发现太晚，大部分是在用户已将系统投入

使用时才发现。而此时维护所需成本要比早期发现高数十倍。

综上所述，过去软件市场只是一种技术交易，成功的关键取决于软件的功能；而今天彼此的竞争对手在软件的功能上可以很快地赶上对方，成功的唯一途径是靠软件质量保证。软件质量保证的重要性可以归结为

1）质量是竞争的需要。

2）质量是生存的基础。

3）质量是进入国际市场的先决条件。

4）质量是降低成本的保障。

5）质量是维护用户和增加利润的保障。

6）质量是世界级企业的标志。

最初的软件质量保证系统是在 70 年代由欧洲首先采用的，其后在美国和世界其他地区也迅速地发展起来。目前，欧洲联合会积极促进软件质量的制度化，提出了如下 ISO-9000 软件标准系列：

ISO-9001；

ISO-9000-3；

ISO-9004-2；

ISO-9004-4；

ISO-9002。

这一系列现已成为全球的软件质量标准。除了 ISO-9000 标准系列外，许多工业部门、国家和国际团体也颁布了特定环境中软件运行和维护的质量标准，如：IEEE 标准 729-1983、730-1984，Euro Norm EN 45012 等。

本书涉及到开发和维护软件产品所需的过程及软件产品的质量，并讨论软件质量保证的各种措施。我们强调的主要论点如下：

1）提高软件生产率的最有效的办法是改善软件质量。

2）软件质量是软件市场竞争中获得成功的关键。

3）改善软件质量可靠性的唯一方法是改善软件开发和维护过程的质量，其中包括人员、设备、仪器、技术、方法和工具的质量。

4）软件质量保证是软件工程学科的一部分，它在软件生存周期中，利用工程技术和实践经验系统地生产和维护高质量的软件。

5）对软件的维护支持与软件产品质量同样重要，维护支持环境必须做到工程化。

6）为了达到软件质量的目标，人的因素和他们的质量意识，即所谓的质量文化，是与技术同样重要的。

7）公司的最高领导者必须认真负责领导软件质量保证工作，否则改善开发和维护过程质量的努力是不会成功的。

8）软件质量及开发和维护过程质量的改善是一个不断努力的过程，应尽可能做到更好、更快和更节约。

9）符合 ISO-9000 标准的软件质量保证系统只是软件公司的初步目标，软件公司的质量保证系统还必须符合本公司的特殊需求和实际情况，否则，该系统就不是有效的和实用的。

有效的软件质量保证系统必须遵守如下软件工程实践的基本原则：

(1) 质量原则

1) 在首次使用时，就应当尽量避免出现错误。

2) 确保尽可能早地检测出存在的缺陷和错误，并加以改正。

3) 根据质量标准和规范审计软件生产。

(2) 管理原则

1) 确定管理的原则和职责。

2) 制定工作计划。

3) 按照计划追踪工作过程，必要时进行修正。

4) 不断地完善计划。

(3) 工程原则

1) 在得出结论之前，要分析所出现的问题。

2) 把复杂的问题划分成许多较容易的问题。

3) 确保能控制各子程序之间的关系，并将它们连接起来。

本书读者的对象是：开发和维护软件产品的计算机专家和软件工程师、软件质量保证专家，从事计算机专业的教师及学生，计算机公司和企业的经理以及负责软件质量保证(QA)的领导。

目　录

第一章　软件质量保证的基本概念

1.1　软件生产率

1.1.1　软件生产率和影响生产率的因素

目前，在很多计算机公司中，新的应用项目常常超过该项目软件开发部门所能承受的能力，这种情况造成了用户应用项目的积压。另外由于软件开发过程是很紧张的劳动，并且需要经过严格培训和有经验的开发人员，而这些人员往往是短缺的。解决这些影响软件生产的问题的途径是提高软件生产率。为了提高软件生产率，可以在开发时采用一些有效的方法、工具和辅助手段，以改进传统的开发过程，或者应用第四代语言(4GLs)生成程序，减轻繁重的编程工作。

软件生产率受到开发时间、投资返回和质量三个因素的影响。提高软件生产率首先要缩短软件的开发时间，其次是开发投资回报较高的系统，即在不改变需求的情况下，降低软件产品的开发和维护成本。第三就是开发具有较高质量的系统，它与开发时间一样，也是开发部门的主要目标之一。

为了达到以上三个目标，必须对软件开发的生产率进行测量。目前，常用的生产率测量包括：每行程序的成本、单位时间的编程行数或单位时间的功能项。功能项只适用于对商业信息系统的评价。在计算软件生产率时要考虑的参数包括用户输入数、用户输出数、用户查询数、文件数或数据库数及与应用有关或与系统有关的接口数。功能项可以用附加复杂性参数和开发过程的经验常数来衡量。

最常用的生产率测量公式如下：

$$P_1=\frac{CSI}{PY}\quad(\text{编程行数/单位时间})$$

$$P_2=\frac{\text{成本}}{CSI}\quad(\text{成本/单位程序行})$$

$$P_3=\frac{FP}{PM}\quad(\text{功能项/单位时间})$$

其中，CSI 是改变的源指令行(包括新的和修改的程序行)；PY 是人·年；PM 是人·月；FP 是功能项。

基于单位时间的编码行(LOC)的生产率的测量是不能令人满意的。首先，由于实际的编程成本只占项目全部开发成本的 20%左右，因此开发过程的生产率也必须考虑到初始阶段的工作；其次，如何对解释行、非执行码和多次共享的编码进行计数，这是编码行的计数问题。现代语言，像 Ada 和 Modula-2，C++和 Visul Basic 等的指令计数也存在问题。第三，依靠 LOC 来鼓励和评价工作人员是很不合理的，这样做的结果将导致无效编码的产生。

基于功能项的生产率测量的缺点是，采用与产生指令数有关的功能项来估算软件生产率时，没有考虑所选择的算法的复杂性；在功能项的计算中所使用的参数和加权因子，

例如，输入、输出、数据库和查询复杂性等，是依靠主观估计和估算来确定的。

生产率测量存在的问题是：怎样定义所采用的值？一个开发小组的 P_1 或 P_3 值是否要与其他组的进行比较？管理者是否用计算值来评价每个开发人员或开发小组？很显然，这种比较是不对的。不同类型的比较必然会得出错误的结论。

另一种测量生产率的公式为

$$P=\frac{\text{开发过程产生的成果}}{\text{成本}}$$

从此式看出，可以依靠增加成果或降低成本来提高开发过程的生产率。开发过程的成本包括工作成本和培训成本、计算机资源和辅助设备的成本。成本可以划分为阶段成本、活动成本、人员费用、培训费用及辅助费用。成本用货币的单位来表示。这个公式的难点和不可靠性在于：

1）怎样确定成果？特别是阶段的成果。怎样看待初期阶段的成果？

2）怎样恰当地评价和测量成果？我们已经讨论过用 LOC 来计算生产率的优缺点。然而，产生的成果也可能是概念性的成果，如系统概念或需求定义等就是不能进行量化来评价的成果，并且有时降低成本反而会导致失败。

3）公式不能适用于不同项目的生产率值的比较，因为它没有考虑影响生产率的各种因素。

因此，准确测量生产率是很困难的。为了能使测量生产率的公式更准确，必须对影响软件生产率的各种因素进行精确的检验。对软件生产率具有重要影响的因素包括：

参加开发的各部门、管理者、开发人员和用户；

开发任务的难度等级；

用来描述开发过程所使用的开发过程模型；

开发过程所使用的方法和工具；

生产的需求及开发资源，特别是它们的利用率等。

达到较高的生产率是管理部门的一项基本任务，他们需要定期采取工作会议的形式与开发人员进行有关提高软件生产率问题的讨论。

有许多研究影响生产率因素的方法，这里我们详细介绍 MIT 实验室的 Packer 研究的在测试开发时影响软件生产率因素的定性方法及 IBM 软件实验室的 Remus 的方法。

1. Packer 的方法

Packer 研究的在软件开发时测量影响生产率因素的定性方法，首先在美国银行的软件生产率分析上使用。Packer 采用让编程人员、分析人员、管理人员和用户四组人员填写调查表的方式。此表是在系统开发过程中出现技术上和组织上的改变前后填写的。调查的问题集中在部门和公司层及用来确定一个机构软件生产率的五个有关的生产率因素：

1）对工作的重要性的认识，对工作是否满意及工作的状态。

2）工作的独立性，是否有足够的可用资源？对成果所负的责任。

3）对工作的改革和创造性及组织机构改善的可能性。

4）工作中的协作关系及领导对工作的支持程度。

5）工作的目标和有关达到目标的反馈数据。

以上五方面是对所调查的四组人员提出的问题。通过多重选择和具有加权的答案，确定每组因素的平均值，并在生产率图上标出上述五点的值。

第一步，标出从管理人员的角度得到的理想位置，即理想的机构环境和工作条件，换句话说就是管理人员必须具有的职责。这种基于加权响应的原始评价图，可以用来分析理想的管理者的位置和单位的实际情况之间的差别。图 1.1 的例子中从理想的管理者的观点看第三组的值大大低于其他组，这表明了组织管理上的弱点，必须提出进一步改进的措施。

第二步，表示出实际管理者的观点与理想值的比较。例如在第二组存在着明显的差别。这是由于从开发到生产，部分计算机资源发生变化产生的。

第三步，当四组人员的实际观点完全得到后，可以进行这四组观点的比较。比较的问题是：这四组人员是否具有相同的组织环境和工作条件。图 1.1 中与其他三组相比，编程人员只有很少的点靠近理想值，其原因可能是年轻的编程人员与年长的之间的关系不好或对编程人员的物质奖励不够所引起的。

第四步，在一个机构的软件生产率图中测试到额外的变化。为此必须在组织变化的前后提出问题，并且将答案画到生产率图上，然后将它们与变化前进行比较。

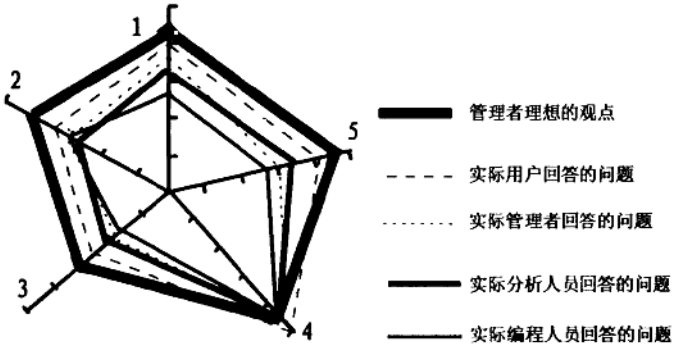


图 1.1　Packer 的部门生产率图

另外，对于计算机领域特殊的生产率的测量，要考虑以下六个影响软件生产率的因素：

1）通信联系。

2）战略的信息系统计划。

3）项目的组织管理。

4）标准化和质量保证措施。

5）工作人员的培训和奖励。

6）技术开发环境。

考虑上述六个因素的理想值与实际情况进行比较的结果如图 1.2 所示。因素 6 最接近，因素 2 偏差最大，因素 3,4,5 偏差相对小些。从图中可以看出需要进行改善的那些方面。

根据两个图中所表示的因素，可以在一个生产率图改变前后对生产率做出判断和评价，然后，对它们进行比较，并讨论它们对四组人员的影响。

Packer 的方法是一个很有用的方法，它可以评价影响生产率的技术的和非技术的因素。它也清楚地表明了高的生产率不仅可以依靠四个组的共同工作来实现，而且也受到各

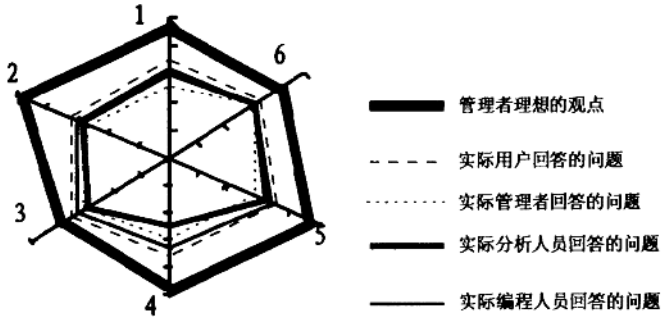


图 1.2　Packer 的计算机领域生产率图

种各样因素的影响。这种系统工程的方法很清楚地表明了计算机技术和组织机构间的复杂关系。

2. Remus 的方法

IBM 软件实验室的 Remus 也进行了软件生产率的分析。他仔细地检验了系统软件的开发和维护，在分析中他分别考虑了生产率受产品、开发者和管理者三方面的影响。

(1) 产品的影响

首先，考虑需求的可靠性，含糊地和频繁地改变需求将延缓开发过程；其次，考虑代表软件产品规模的编码行数的影响，在不同软件产品中，可重复使用的编码数量相同或没有可重复使用的编码时，生产率的提高与产品规模成正比，其原因是小的产品在测试开销上大于大的产品；第三，考虑编码类型的影响，修改编码的生产率低于编写新编码的生产率。最后，考虑任务的复杂性和难度的影响，任务的复杂性和难度明显地影响开发人员的开发效率。

(2) 人的因素影响

人的能力和经验对生产率产生极大的影响。一个具有多年实践经验的编程人员比只接受了培训的程序员的生产率高出近两倍。另外，除了要了解开发过程外，开发人员的工作状态和所使用的方法也是不容忽视的。特别是他们对软件产品质量和生产率的重视程度，是达到较好的质量和较高的生产率的关键所在。通过工作成果的积累和对开发人员的奖励，使他们处于一种积极的状态，通过培训和进一步教育可以改善开发人员对编码和设计过程的适应性，这些对提高软件生产率都是有益的。

(3) 加强开发过程的管理

首先，建立过程模型，可以得到有序的过程流程图和划分过程中必需的各阶段，实现对过程的广泛测量和评价；其次，通过开发人员用常规测量来检查其成果，可以保证软件质量并提高生产率；测量以及对测量值的说明可以使开发过程对所有参与者成为透明的，必须使所有工作人员都了解测量和评价的原因；在开发过程中，由于会议、评审、新员工培训和非计划的事件等造成的工作过程的中断，使计算机停机或对开发人员的支持和帮助不够等都会降低生产率，这是管理者应当特别注意的。

管理者的决策之一是选择合适的软件技术结构。决策不当则将导致开发过程的修改和推迟，建议采用原型和优化的方法。在确定了软件结构之后，使用能够满足需求和用户

友好的工具，特别是能使开发过程每一步实现自动化的工具，如文档和编码生成程序和分析程序等及支持各阶段的软件工程数据库，都会对提高生产率产生重要的影响。采用监测产品状态的工具也是改善生产率的有效的方法。

另外，还必须考虑工作环境和工作场所的设备，如，办公室的工作环境和设备、硬件的合用性、容易存取的计算机终端和工作站等因素对于生产率的影响。

1.1.2 软件质量和软件生产率之间的关系

1. 生产率是效率和质量的函数

人们可能认为，为了改善质量所采取的各种类型的措施的成本太高，因此，要获得较好的质量，就要降低生产率。这些想法错误地理解了质量、生产率和其他开发量之间的关系。

生产率是效率和质量的函数。所谓效率是指在一定的资源下完成开发任务的速度，例如：使用程序发生器完成单位时间编制 100 行的技术规范或源程序或文档，比用相同资源完成单位时间编制 50 行的效率要高。假定在相同的内部和外部条件下，开发过程执行得愈快，产品中产生错误的可能性愈大。这些错误需要在开发过程中和后面的维护阶段进行处理，其结果都要增加成本。如前所述，在产生错误并修改它之间拖延的时间愈长，所增加的成本愈多。其结果是在发现错误和修正错误之间的延误会降低生产率。因此，在测量生产率时必须同时考虑效率和质量。

生产率可以表示为效率和质量的一个线性函数。见图 1.3。从图中可以看出，仅仅改善质量或效率并不能提高生产率 P′，这种情况在实践中常常被忽略。开发管理者经常对开发人员施加压力，以便更快地得到令人满意的成果，其结果是，开发出的软件产品的质量不好，这些产品的错误要在后面花费更大的努力才能改正。因此，总的生产率是不会提高的。

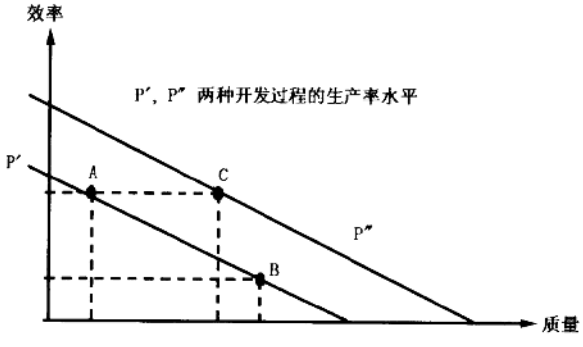


图 1.3 生产率是效率和质量的函数

另一种典型的情况是企图用增加资源的办法改善质量，如增加工作人员，这样做的结果降低了效率，即在图 1.3 中，从 A 点移到 B 点。

开发的管理者的实际目标在于选择一个较高的生产率水平，在图 1.3 中用 P″代替 P′，也就是说效率和质量必须同时改进。质量和生产率是可以同时提高的。生产率的函数是由图 1.2 中所描述的因素所决定的。

通过分析效率和质量与生产率之间的关系，可以得出以下结论：

1）必须测量效率和质量，包括测量在一定的资源和技术条件下执行任务的速度，测量在给定的项目中各阶段发生的错误率等。

2）仅仅提高效率或改善质量不能提高生产率的水平。为了快而不考虑质量，同样得不到较高的生产率。

3）生产率改善的范围必须包括所有的开发阶段，如果编程工作仅占用20%的项目经费，80%编程生产率的改善只带来整个生产率的16%的改善。

2. 生产率模型

根据测量生产率的公式可以推导出生产率模型。图1.4所示为介绍某软件的生产率模型，在模型里用产品的价值代替前面公式中的开发过程产生的成果。

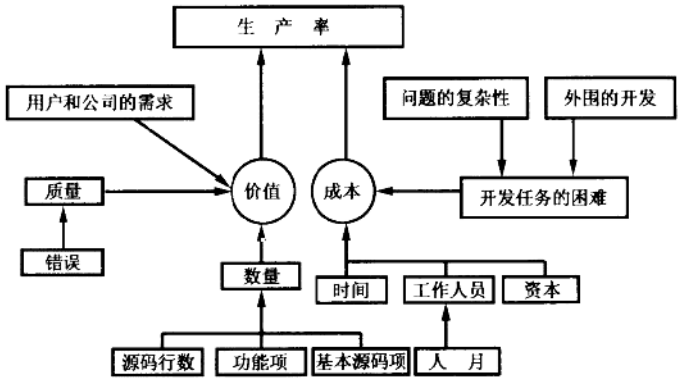


图1.4 生产率模型

因此，得到下面的测量生产率的公式：

$$P(\text{生产率})=\frac{\text{产品的价值}}{\text{生产成本}}$$

在此模型中，将软件生产看成产生价值的过程。软件产品表示为由用户和公司的需求、质量需求、重复使用的部件及产品数量所决定的一个值，此值的大小可以根据产品的规模来确定，而规模是由功能项、源码行数和基本的源码项所决定的。价值也可以用软件产品在满足用户或公司的需求方面的可用性来决定。

在图1.4的生产率模型中，价值和成本是等价的。成本的计算包括工作人员（用人·月计算），考虑所有开发所需要的时间、经济资源和开发任务的难度级，包括：问题的复杂性、开发中的外部接口等。在此模型中值得注意的现象是：

1）降低软件开发成本可以通过减少质量成本来实现，然而其结果是增加了生存周期的成本，它包括40%开发成本，60%维护成本，这是由于修改错误和减少功能所造成的。采用此方式由于降低质量和减少了数量，使产品的价值降低。

2）通过在项目中采用积极的建设性的质量保证措施，如选用合适的方法及工具，使产品成本和软件质量同时得到改善。其结果是产品的价值增加的同时生产率也提高了。

1.1.3 提高软件生产率

本节将讨论提高软件生产率的实际的可能性，同时说明改善生产率的许多措施对开

发过程及最终产品的影响.

1. 提高生产率的途径

提高生产率所采取的措施是由计算机公司的管理者制订的,经常采用“调查表”的方法来寻求改善生产率的途径。

Toshiba 公司每年在它的软件机构中进行两次调查。该公司下属某软件工厂有员工约 2 300 名,每个员工和质量小组都要对怎样要求生产率的目标以及怎样能达到提高生产率的目标写出评议报告。这些评议结果如表 1.1 所示,从表中可以清楚地看到,最大的潜在生产率产生于对软件程序单元的重复使用之中。除此之外,提高生产率的一个值得注意的途径是采用面向目标的编程。

表 1.1 某软件工厂的调查结果

被调查人的百分数	提高生产率的因素
52.0	可重复使用的设计和编码
18.0	改善工作过程、工艺过程中的步骤和工作环境
9.8	使用软件工具
7.2	软件工程的应用
5.7	改善系统的功能划分
5.3	使用第四代语言(VHLL)

2. 提高生产率的措施

图 1.5 表示了提高软件生产率的六大类措施。

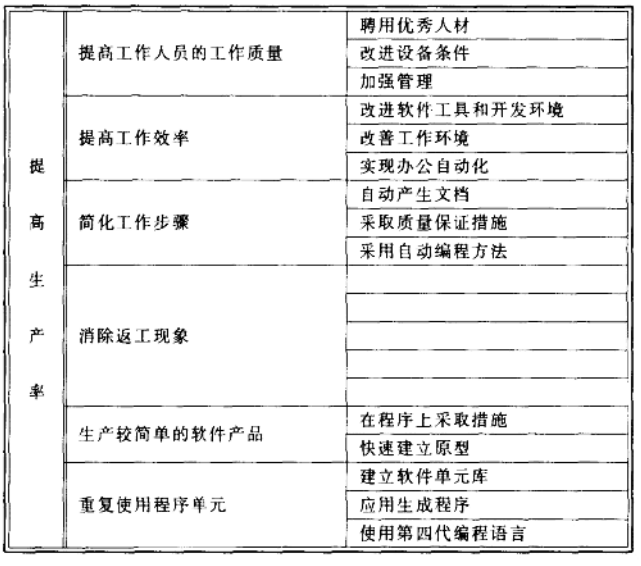


图 1.5 提高软件生产率的措施