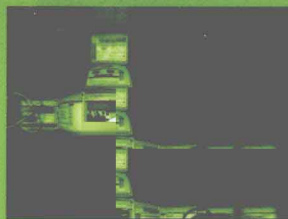




职业技术·职业资格培训教材



软件测试人员(.NET)(高级)

人力资源和社会保障部教材办公室
中国就业培训技术指导中心上海分中心 组织编写
上海市职业培训研究发展中心



中国劳动社会保障出版社

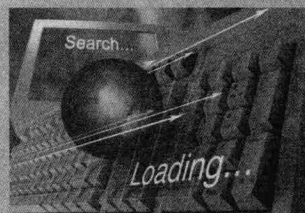


职业技术·职业资格培训教材

主 编 张伟平 徐龙章

编 者 崔海霞 栾东庆 彭华林

主 审 罗晓丹



软件测试人员(.NET)(高级)



中国劳动社会保障出版社

图书在版编目(CIP)数据

软件测试人员(.NET):高级/上海市职业培训研究发展中心组织编写. —北京:中国劳动社会保障出版社, 2011

1 + X 职业技术·职业资格培训教材

ISBN 978 - 7 - 5045 - 9071 - 8

I. ①软… II. ①上… III. ①软件-测试-技术培训-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 118918 号

中国劳动社会保障出版社出版发行

(北京市惠新东街 1 号 邮政编码: 100029)

出版人: 张梦欣

*

北京宏伟双华印刷有限公司印刷装订 新华书店经销

787 毫米×1092 毫米 16 开本 14 印张 257 千字

2011 年 7 月第 1 版 2011 年 7 月第 1 次印刷

定价: 28.00 元

读者服务部电话: 010 - 64929211/64921644/84643933

发行部电话: 010 - 64961894

出版社网址: <http://www.class.com.cn>

版权专有 侵权必究

举报电话: 010 - 64954652

如有印装差错, 请与本社联系调换: 010 - 80497374

内 容 简 介

本教材由人力资源和社会保障部教材办公室、中国就业培训技术指导中心上海分中心、上海市职业培训研究发展中心依据上海 1 + X 软件测试人员（. NET）三级职业技能鉴定细目组织编写。教材从强化培养操作技能，掌握实用技术的角度出发，较好地体现了当前最新的实用知识与操作技术，对于提高从业人员基本素质、掌握高级软件测试人员的核心知识与技能有直接的帮助和指导作用。

本教材在编写中根据本职业的工作特点，以能力培养为根本出发点，采用模块化的编写方式。本教材内容共分为 4 章，主要包括：软件测试技术应用、VSTS 及测试技术、自动化测试工具以及软件测试管理与应用。

本教材可作为软件测试人员（. NET）三级职业技能培训与鉴定考核教材，也可供全国中、高等职业技术学院相关专业师生参考使用，以及本职业从业人员培训使用。

前 言

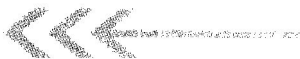
职业培训制度的积极推进，尤其是职业资格证书制度的推行，为广大劳动者系统地学习相关职业的知识和技能，提高就业能力、工作能力和职业转换能力提供了可能，同时也为企业选择适应生产需要的合格劳动者提供了依据。

随着我国科学技术的飞速发展和产业结构的不断调整，各种新兴职业应运而生，传统职业中也愈来愈多、愈来愈快地融进了各种新知识、新技术和新工艺。因此，加快培养合格的、适应现代化建设要求的高技能人才就显得尤为迫切。近年来，上海市在加快高技能人才建设方面进行了有益的探索，积累了丰富而宝贵的经验。为优化人力资源结构，加快高技能人才队伍建设，上海市人力资源和社会保障局在提升职业标准、完善技能鉴定方面做了积极的探索和尝试，推出了1+X培训与鉴定模式。1+X中的1代表国家职业标准，X是为适应上海市经济发展的需要，对职业的部分知识和技能要求进行的扩充和更新。随着经济发展和技术进步，X将不断被赋予新的内涵，不断得到深化和提升。

上海市1+X培训与鉴定模式，得到了国家人力资源和社会保障部的支持和肯定。为配合上海市开展的1+X培训与鉴定的需要，人力资源和社会保障部教材办公室、中国就业培训技术指导中心上海分中心、上海市职业培训研究发展中心联合组织有关方面的专家、技术人员共同编写了职业技术·职业资格培训系列教材。

职业技术·职业资格培训教材严格按照1+X鉴定考核细目进行编写，教材内容充分反映了当前从事职业活动所需要的核心知识与技能，较好地体现了适用性、先进性与前瞻性。聘请编写1+X鉴定考核细目的专家，以及相关行业的专家参与教材的编审工作，保证了教材内容的科学性及与鉴定考核细目以及题库的紧密衔接。

职业技术·职业资格培训教材突出了适应职业技能培训的特色，使读者通



过学习与培训，不仅有助于通过鉴定考核，而且能够真正掌握本职业的核心技术与操作技能，从而实现从懂得了什么到会做什么的飞跃。

职业技术·职业资格培训教材立足于国家职业标准，也可为全国其他省市开展新职业、新技术职业培训和鉴定考核，以及高技能人才培养提供借鉴或参考。

本教材在编写过程中得到了上海市信息服务外包发展中心、上海立达职业技术学院的大力支持与协助，在此一并表示衷心的感谢！

新教材的编写是一项探索性工作，由于时间紧迫，不足之处在所难免，欢迎各使用单位及个人对教材提出宝贵意见和建议，以便教材修订时补充更正。

人力资源和社会保障部教材办公室
中国就业培训技术指导中心上海分中心
上海市职业培训研究发展中心

目 录

● 第1章 软件测试技术应用	1
第1节 软件测试技术基础	2
第2节 常见软件测试技术	13
第3节 白盒测试及用例设计	36
● 第2章 VSTS 及测试技术	49
第1节 面向对象与 C#语言基础	50
第2节 面向对象软件测试	114
● 第3章 自动化测试工具	123
第1节 NUnit 安装与应用	124
第2节 FxCop 安装与应用	140
● 第4章 软件测试管理与应用	153
第1节 测试计划与测试用例	154
第2节 缺陷管理与分析	160
第3节 缺陷管理工具及应用	166
第4节 软件测试项目应用	180
附录 FxCop 基本规则	202



1

第 1 章

软件测试技术应用

- 第 1 节 软件测试技术基础 /2
- 第 2 节 常见软件测试技术 /13
- 第 3 节 白盒测试及用例设计 /36

第 1 节 软件测试技术基础

一、软件工程

1. 软件工程目标

软件工程是研究和应用如何以系统、规范化、可量化的过程化方法去维护软件，以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来的工程。狭义上说，软件工程的目标在于生产出满足预算、按期交付、用户满意的无缺陷的软件，当用户需求改变时，生产的软件必须易于修改。

图 1—1 为一个精简的软件工程概念模型。

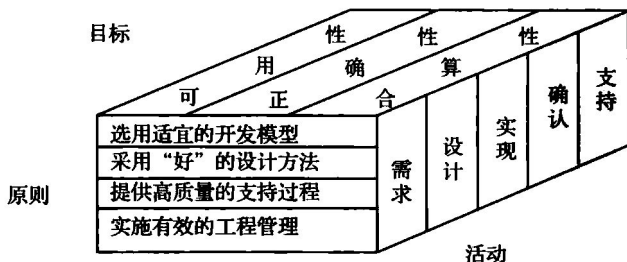


图 1—1 软件工程

该模型可以用一句话概括：软件工程是目标、原则、活动的集合，体现了目标—原则—活动的三维正交关系：

- (1) 任何目标，都要依照特定方法，由特定活动实现。
- (2) 任何原则，都要指导特定活动，来完成某种目标。
- (3) 任何活动，都由特定方法指导，来完成某种目标。

2. 软件工程的基本原理

自从 1968 年提出“软件工程”这一术语以来，研究软件工程的专家学者陆续提出了上百条关于软件工程的原理和准则；美国著名软件工程专家 Barry Boehm 综合各方面专家的意见，并总结了 TRW 公司多年软件开发的经验，于 1983 年提出了软件工程的七条基本原理。

Boehm 认为，七条原理是确保软件产品质量和开发效率的最小原理集合，它们是相互独立的，且缺一不可，是相当完备的原理。虽然人们不能用数学的严格方法证明它们是一

个完备的集合，但是可以证明，在此之前专家学者提出的上百条软件工程原理都可以由这七条原理的任意组合蕴含或派生，具体如下：

(1) 用分阶段的生命周期计划严格管理。有数据统计发现，在不成功的软件项目中，有一半左右是由于计划不周造成的，把建立完善的计划作为第一条基本原理是在吸取前人教训的基础上提出的。在软件开发与维护的漫长生命周期中，需要完成许多性质各异的工作，这条基本原理意味着，应该把软件生命周期划分成若干个阶段，并相应地制订出切实可行的计划，然后严格按照制定的计划对软件开发与维护工作进行管理。Boehm 认为，在软件的整个生命周期中应该制订并严格执行六类计划，它们是项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划。不同层次的管理人员都必须严格按照计划各司其职、各尽其能地管理软件开发和维护工作，绝不能受客户或上级人员的影响而擅自背离预定计划。

(2) 坚持进行阶段评审。在提出软件工程的七条基本原理之前，人们当时就已经认识到，软件的质量保证工作不能等到编码阶段结束之后再行进行，支持这种观点的理由至少有两个：其一，大部分错误是在编码之前造成的，例如：根据 Boehm 等人的统计，设计错误占软件错误的 63%，编码仅占 37%；其二，越晚发现和改正错误，所需付出的代价也越高。因此，在每个阶段都进行严格的评审，以便尽早发现在软件开发过程中所犯的错误，是一条必须严格遵循的重要原则。

(3) 实行严格的产品控制。在软件开发过程中不应随意改变需求，因为改变一项需求往往需要付出高昂的代价，但是，在软件开发过程中改变需求又是难免的，由于外部环境的变化，相应地改变用户需求是一种客观需要，显然不能硬性禁止客户提出改变需求的要求，而只能依靠科学的产品控制技术来顺应这种要求。即当需求改变时，为了保持软件各个配置成分的一致性，必须实行严格的产品控制，其中主要是实行基准配置管理。

所谓基准配置又称基线配置，它们是经过阶段评审后的各个阶段产生的文档或程序代码，一切关于修改软件的意见和建议，特别是涉及对基准配置的修改建议，都必须按照严格的流程进行评审，只有获得批准后才能实施修改。绝对不允许未经批准随意修改（包括尚处于开发过程中的软件）。

(4) 采用现代程序设计技术。从提出软件工程的观念开始，人们就一直把主要精力用于研究各种新的程序设计技术。60 年代末提出的结构程序设计技术，已经成为绝大多数人公认的先进的程序设计技术，后来又进一步发展出各种结构分析、结构设计、面向对象设计等新技术。实践证明，采用先进的技术既可以提高软件开发的效率，又可以提高软件维护的效率。

(5) 结果应能清楚地审查。软件产品不同于一般的物理产品，它是看不见、摸不着的逻辑产品。软件开发人员的工作进展情况可见性差，难于准确度量，从而使得软件产品的

开发过程比一般产品的开发过程更难于评价和管理。为了提高软件开发过程的可见性,更好地进行过程控制和管理,应该根据软件开发项目的总目标及完成期限,规定开发组织的责任和标准,从而使所得到的结果能够清楚地审查。

(6) 开发小组的人员应该少而精。该条基本原理的含义是,软件开发小组的组成人员应具备良好的素质,人数不宜过多,因为开发小组人员的素质和数量是影响软件产品质量和开发效率的重要因素。素质低的人员的开发效率与素质高的人员的开发效率可能相差几倍甚至几十倍,而且素质高的人员所开发的软件,错误明显少于素质低的人员所开发的软件。此外,随着开发小组人员数量的增加,因人员交流情况、讨论问题而造成的通信开销会急剧增加。当开发小组为 N 人时,可能的通信路径有 $N(N-1)/2$ 条,可见随着数 N 的增大,通信开销将急剧增加。因此,组建少而精的开发小组是软件工程的一条基本原理。

(7) 不断改进软件工程实践的必要性。遵循上述六条基本原理,就能够按照当代软件工程的基本原理实现软件的工程化生产,但是,仅有上述六条原理并不能保证软件开发与维护的过程,能赶上时代前进的步伐和技术的不断更新。因此,Boehm 提出,应把不断改进软件工程实践的必要性作为软件工程的第七条基本原理。按照这条原理,不仅要积极主动地采纳新的软件技术,而且要注意不断总结经验。例如:收集资源耗费数据,收集出错类型和问题报告数据等,这些数据不仅可以用来评价新的软件技术效果,而且可以用来指明必须着重开发的软件工具和优先研究的新技术。

3. 软件测试与软件质量的关系

软件测试是为软件项目服务的,在整个项目中,要强调测试服务的概念,虽然软件测试的目的是为了发现软件中存在的错误,但是,其根本目的是为了₄提高软件质量,降低软件项目的风险。

软件的质量风险表现在内部风险和外部风险两个方面。内部风险是在即将销售的时候发现有重要的错误,从而延迟软件产品的发布日期,失去市场的机会;外部风险是用户发现了不能容忍的错误,引起索赔、法律纠纷以及用于客户支持的费用,甚至失去客户的风险。

软件测试只能证明软件存在错误,而不能证明软件没有错误。软件公司对软件项目的期望是在预计的时间、合理的预算下,提交一个可以交付使用的软件产品;测试的目的就是把软件的错误控制在一个可以进行产品发布的程度上,发布的产品并不是没有错误的产品,因此软件测试不可能无休止地进行下去,而是要把错误控制在一个合理的范围之内,因为软件测试也是需要花费巨大成本的。

有资料表明,波音 777 飞机整体设计费的 25% 都花在测试上,而且随着测试时间的延

伸，发现错误的成本会越来越大，这就需要测试有度，而这个度并不能由项目计划的时间来判断，而是要根据测试出现错误的概率来判断。这也要求在项目计划时，要给测试留出足够的时间和经费，仓促的测试或者由于项目提交计划的压力而终止测试，会对整个项目造成无法估量的损失。

4. 软件测试与软件开发的关系

(1) 测试与开发各阶段的关系。软件测试与开发各阶段的关系如图 1—2 所示。

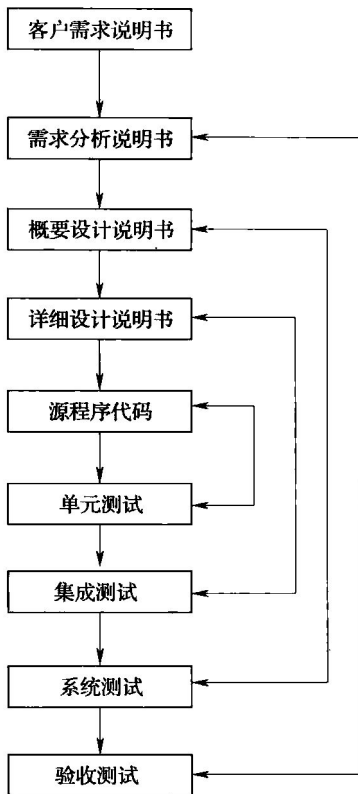


图 1—2 软件测试与开发关系图

从图中可以发现，测试在开发阶段的作用如下：

1) 需求分析阶段：确定测试需求分析、系统测试计划的制订、评审后的管理。其中，测试需求分析是对产品生命周期中，测试所需要的资源进行深入分析；系统测试计划是依据软件的需求规格说明书，设计测试用例和测试的实施等。

2) 概要设计与详细设计阶段：确保完成单元测试、集成测试和系统测试计划。

3) 编码阶段：该阶段主要由开发人员测试自己负责部分的代码。在较大的项目中，

则由专人进行编码阶段的测试任务。

4) 测试阶段：实施单元测试、集成测试、系统测试、验收测试，并提交相应的测试报告。

(2) 测试与开发的并行性。在确定软件的需求并通过评审后，概要设计工作与测试计划的制订和设计工作就要并行。如果系统模块已经建立，对各个模块的详细设计、编码、单元测试等工作可以并行实施。待每个模块完成后，可以进行集成测试、系统测试、验收测试。并行工作流程如图 1—3 所示。

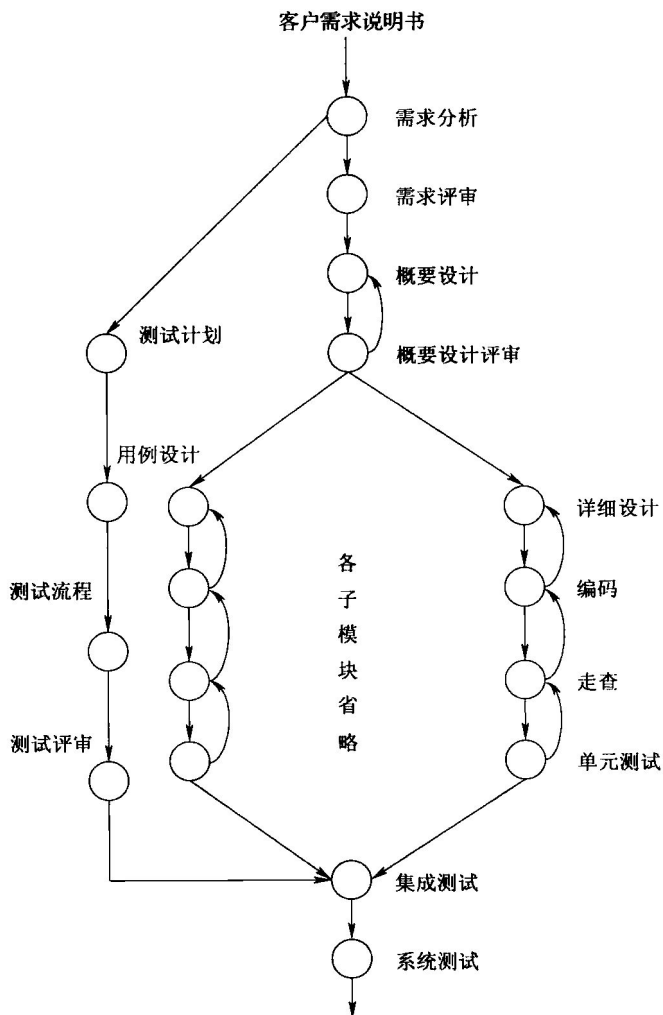


图 1—3 测试与开发并行图

二、软件质量模型

在软件行业领域中，比较流行的质量模型主要包括：McCall 质量模型、ISO 质量模型（软件部分）、CMM - SW 质量模型、6 - Sigma 质量模型。

1. McCall 标准简介

1976 年 Boehm 等人提出了软件质量模型的分层方案，1979 年 McCall 等人通过改进 Boehm 质量模型又提出了一种改进型软件质量模型，即 McCall 质量模型。模型的三层次框架如图 1—4 所示，表 1—1 列出了 McCall 模型对象的作用。

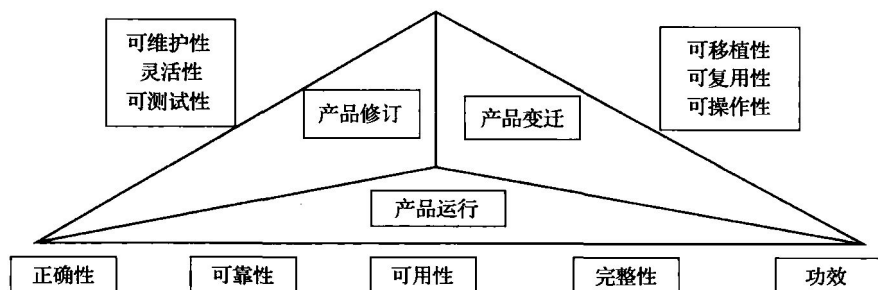


图 1—4 模型的三层次框架

表 1—1

McCall 模型对象的作用

作用	说明
正确性	通常指严格按产品规格说明书的要求进行评估，保证软件产品没有缺陷
可靠性	通常指软件产品持续运行或产品在特定的条件下（负载、大数据量）运行时不会出现错误、异常
可用性	通常指软件产品的易用程度，客户体验
完整性	通常指数据、功能的完整性，同时对重要数据和功能进行权限保护
功效	通常指运行软件产品时所需消耗的计算机资源
可维护性	通常指客户提出新的要求或软件产品在特殊环境下运行时出现问题的维护成本
灵活性	通常指软件产品在需求变更时所需修改的程序
可测试性	通常指提供的软件产品功能完善
可移植性	通常指软件产品在移植过程中的难易程度
可复用性	通常指软件产品中各模块的可重复利用性，复用程度越高，软件产品越好
可操作性	通常指软件产品操作是否简便、上手快

2. ISO9000 标准简介

20 世纪 90 年代初期, 软件工程组织试图将诸多的软件质量模型统一到一个模型中, 并把这个模型作为度量软件质量的一个国际标准。国际标准化组织 1991 年颁布了 ISO 9126—1991 标准《软件产品评价 - 质量特性及其使用指南》, 与此同时, 我国 1996 年发布了类似的软件产品质量评价标准 GB/T 16260—1996。它是一个分层质量模型, 有 6 个影响质量的特性, 图 1—5 为质量特性和质量子特性的层次结构。

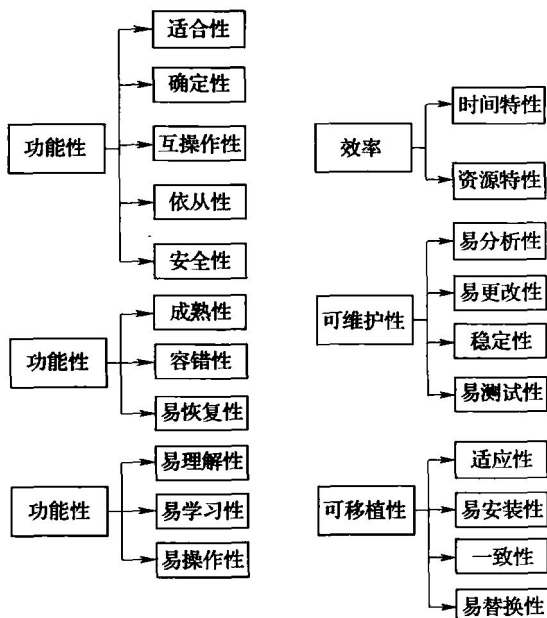


图 1—5 质量特性

3. CMMI 标准简介

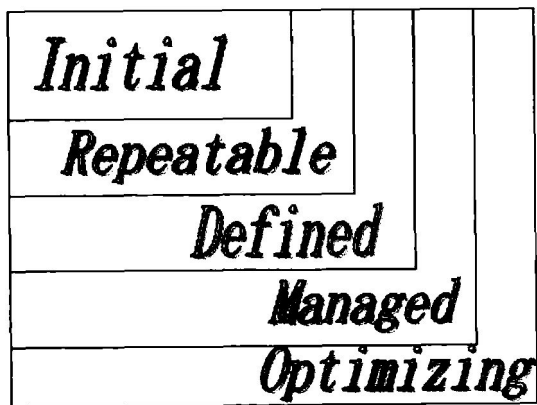


图 1—6 软件能力成熟度模型等级

在介绍 CMMI 标准之前，有必要先介绍一下 CMM，CMM (Capability Maturity Model) 是能力成熟度模型的英文缩写，该模型是基于多年对产品质量的研究成果而建立的。随着计算机的发展，1991 年美国制定了最初的标准 CMM1.0，该标准最初的主要目的是为了评价美国国防部的软件合同承包组织的能力，后因为软件企业应用 CMM 模型实施过程改进取得了较好的收益，该标准很快被推广到全美乃至世界范围，随着 CMM 的实施，SEI 同时建立了主任评估师评估制度，为全世界的软件企业进行认证评估，CMM 的评估方法为 CBA - IPI。

在 1992 年 4 月，SEI 举行了一个 CMM 的研讨会，在广泛听取了众多软件工程专家的意见之后，于 1993 年推出了 CMM1.1 版，直到 2003 年 SEI 宣布不再更新和支持 CMM；同时启用新的标准 CMMI (Capability Maturity Model Integration，即能力成熟度模型集成) 进行替代。CMMI 标准最为人所熟知的就是将标准中的若干个过程区域分成了 5 个成熟度级别，帮助实施 CMMI 的组织建议一条比较容易实现的过程改进发展道路，以下分别介绍 CMMI 的 5 个成熟度级别。

(1) 初始级 (Initial)。初始级，作为 CMMI 的最低的等级，其项目在软件开发过程中没有提供一整套完成的项目流程；尽管公司对项目的目标很清晰，但项目的成功往往依赖于项目组成员的个人能力，而且没有监督、评估，项目的成功具有非常大的偶然性，同时企业也无法保证在实施同类项目时能够及时完成任务，作为仅达到初始级的软件企业，在实现软件产品功能上是非常努力的，但最后基本上都无法控制项目的进度、成本、软件产品的稳定性、易用性等问题。

(2) 可重复级 (Repeatable)。项目在软件开发过程中已建立了基本的项目管理过程，可用于对成本、进度和功能特性进行跟踪和管理，当再次研发类似应用开发项目时可以重复利用以往成功的案例。

(3) 已定义级 (Defined)。项目在软件开发过程中用于管理、工程的软件过程均已实现文档化、标准化，并形成了整个软件组织的标准软件过程，在展开测试之前，均会采用适当修改并力争与实际情况相适应的标准过程。

(4) 可管理级 (Managed)。项目在软件开发过程和产品质量方面均有详细的度量标准，在实施过程中全程进行监督和控制，使项目按计划进行。

(5) 持续优化级 (Optimizing)。通过对已有过程的改进，同时引进新概念、技术和方法进行各种定量分析，能够不断地、持续性地对过程进行改进。

4.6 - Sigma 质量模型

6 - Sigma 是质量标准领域的一个新兴标准，其标准概念最早由摩托罗拉公司提出，该公司在推行 6 - Sigma 质量管理标准之后，产品在品质方面产生巨大的提升，6 - Sigma 质

量管理标准随即受到瞩目并获得广泛的应用。

Sigma (中文译为: 西格玛) 是希腊字母 σ 的译音, 在统计学上用来表示“标准偏差”, 即数据的分散程序, 6-Sigma 即意为: 6 位标准偏差。Sigma 质量标准通过应用数量统计来协助衡量价值流的每一过程、每一工序, 以及每一个发送过程与结果, 在质量领域, 6-Sigma 表示每百万个产品的不良品率不大于 3.4, 即每一百万个产品最多只有 3.4 个不合格产品。

6-Sigma 质量模型通常划分了五个阶段, 分别是: D 代表定义; M 代表测量; A 代表分析; I 代表改进; C 代表控制。企业通过实施 6-Sigma 质量标准可大大提高软件产品质量, 降低产品的失误和故障率, 并提高客户的满意度和市场占有率, 从而增加销售额。另一方面, 通过提高效率、减少故障, 企业还可节省大量因质量问题和效率低下而损失的成本。

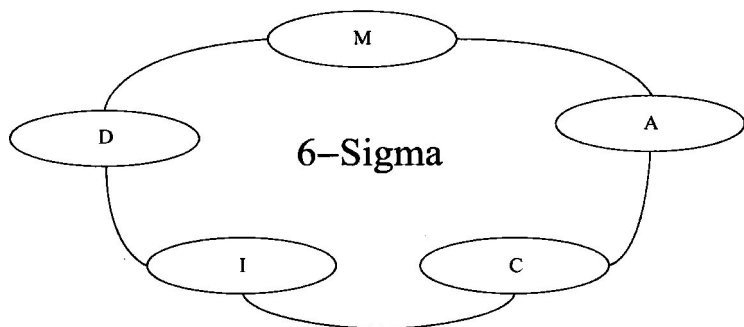


图 1—7 6-Sigma 质量模型

三、软件测试的策略

为了实现软件测试活动的总体方法和目标, 保证开发的软件符合客户和软件需求规格说明书的要求, 同时确保软件产品质量, 软件测试策略必须按照标准规范首先构造测试环境和测试数据资源, 并从不同的角度、采用不同的测试设计方法, 并借助测试工具, 考虑测试策略中可能隐藏的风险等进行测试。

其中:

- 测试的结束标准依据事先定义好的标准执行, 例如: 当测试用例成功执行 95% 以上时, 可以认定软件测试通过验收; 或者在现有的缺陷库中没有非常严重的 bug, 如: 系统崩溃、数据丢失等现象的发生; 还可以通过白盒方法测试代码覆盖率, 当达到一定的标准时, 可以认定测试结束。

- 测试类型, 如功能测试、性能测试、容量测试、负载测试、安装/卸载测试、接口