

软件质量与评测技术

Software Quality & Evaluation Technology

计算机学院 单纯
sherryshan@bit.edu.cn
2025年11月

软件测试基础

Software Testing Foundation

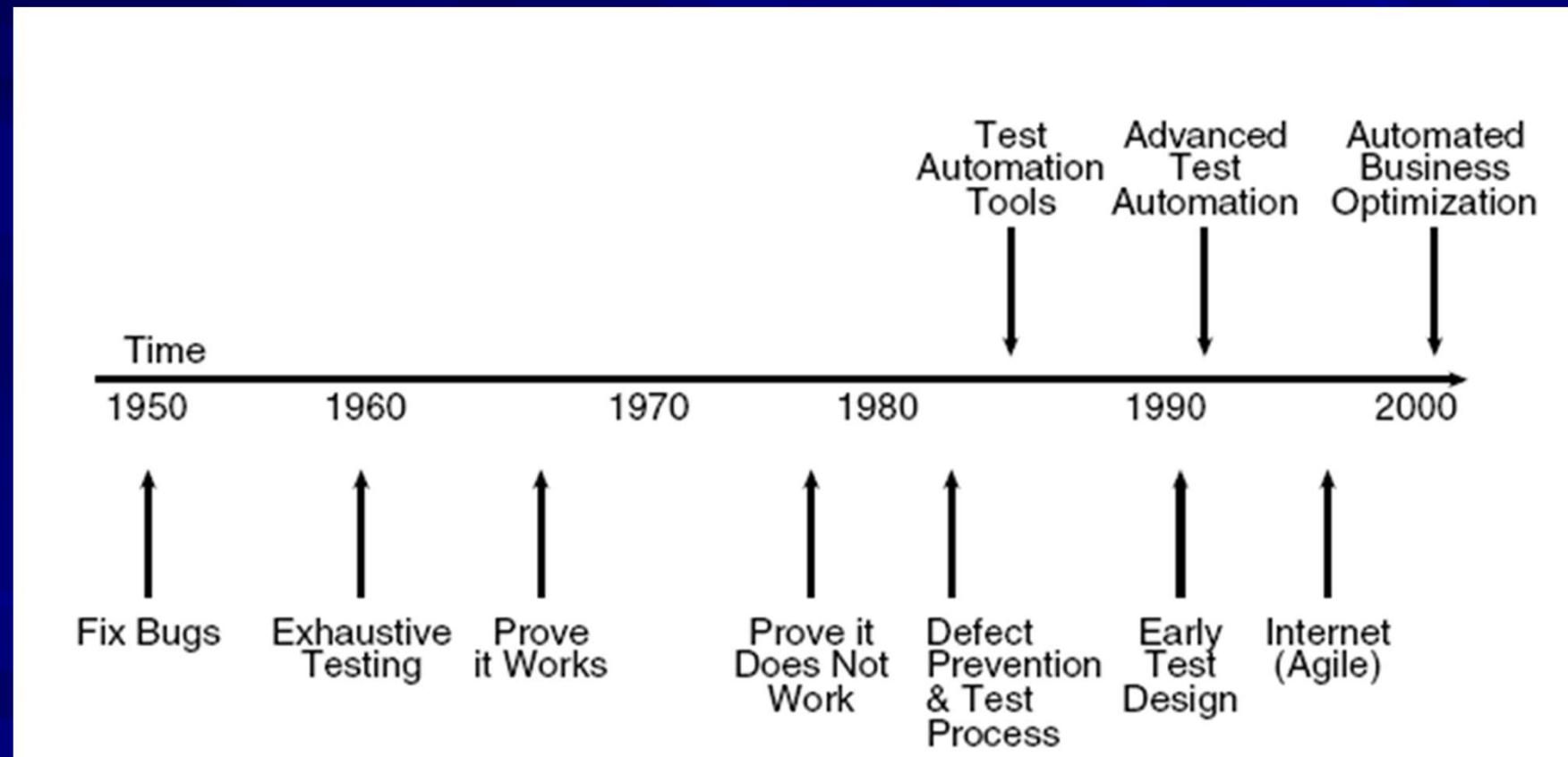
计算机学院 单纯

sherryshan@bit.edu.cn

2025年11月24日

3. 学科发展 (1)

■ A Brief History of Software Testing



3. 学科发展 (2)

- 60年代（软件工程建立前），软件测试最多只是在开发结束后由程序员执行的一些活动，目的是为了表明程序是正确的
 - 没有专职测试工程师
 - 没有测试的专门知识和技能（除了猜测错误）
 - 没有测试工具
 - 没有测试流程
 - 没有多少人能够区分测试和调试

3. 学科发展 (3)

- 1972年，Bill Hetzel博士在 North Carolina 大学举行第一次以测试为主题的正式会议

3. 学科发展 (4)

- 1979年，Glenford Myers出版《The Art of Software Testing》认为测试是Art，介绍了众多的测试经典方法，揭示了测试的目的是证伪，而非证真。同一时期软件测试的先驱们提出：在开发生命周期的前期就应该有基于需求的测试计划

3. 学科发展 (5)

- 1981年，Bill Hetzel博士开设“结构化软件测试”（Structured Software Testing）公开课。随后，出版《The Complete Guide to Software Testing》

3. 学科发展 (6)

- 1988年David Gelperin & Bill Hetzel在Communication of the ACM发表 “The Growth of Software Testing”，文中介绍了系统化的测试和评估流程（Systematic Testing and Evaluation Process即STEP）

3. 学科发展 (7)

■ 70年代后期至80年代中期，很多软件企业成立了QA（Quality Assurance）或者SQA部门。后来QA的职能转变为流程监控（包括监控测试流程），而测试（Testing）则从QA中分离出来成为独立的组织职能

3. 学科发展 (8)

■ 1996年

- 在**Testing Computer Software**会议上，**Rodger Drabick**和**Susan Burgess**提出测试能力成熟度**TCMM** (**Testing Capability Maturity Model**)
- **David Gelperin**和**Aldin Hayashi**在**Application Developmnet Trends**上发表了“**How to Support Better Software Testing**”一文，文中提出了测试成熟度**TMM** (**Testing Maturity Model**)，后来发展为测试支持度**TSM** (**Testability Support Model**)
- **Illene Burnstein**博士、**Robert Carlson**和**Taratip Suwannasart**在**Quality Week**会议上提出了测试成熟度**TMM** (**Testing Maturity Model**)

3. 学科发展 (9)

- TCMM (Testing Capability Maturity Model) 、 TSM (Testability Support Model) 和 TMM (Testing Maturity Model) 是3个各有侧重的有关测试的成熟度模型

3. 学科发展 (10)

■ “成熟度”这一概念来源于美国Carnegie Mellon大学软件工程研究所的软件能力成熟度模型（SW-CMM）。软件能力成熟度模型侧重从管理角度进行流程改进。在工程方面，特别是在软件测试这一领域未做详细的探讨。这3个有关测试的成熟度模型是对软件能力成熟度模型的有益补充

测试支持度模型 (TSM) (1)

- TSM从测试组织的外部而非内部本身来考察测试的成熟度
- TSM有3个级别
 - 第一级：弱 (**weak**)
 - 可测性支持，很少测试问题被提及
 - 第二级：基本 (**basic**)
 - 可测性支持，基本的测试问题被提及
 - 第三级：强 (**strong**)
 - 可测性支持，所有测试问题被提及

测试支持度模型 (TSM) (2)

■ TSM定义了6个KSA (Key Support Areas)

- 测试友好的基础框架 (Test Friendly Infrastructure)
- 顾及测试的项目计划 (Test-aware Project Planning)
- 测试友好的产品信息 (Test Friendly Product Information)
- 顾及测试的软件设计 (Test-aware Software Design)
- 测试件 (Test ware)
- 测试环境设计 (Test Environment Design)

测试成熟度模型 (TMM) (1)

- TMM定义了5个级别

- 第一级：初始级 (Initial)
 - 没有定义成熟度目标 (Maturity Goal)
- 第二级：阶段定义级 (Phase Definition)
 - 目标1：进行测试和调试的目标
 - 目标2：开始一个测试计划过程
- 第三级：集成级 (Integration)
 - 目标1：建立一个软件测试组织
 - 目标2：测试集成进入软件生命周期
 - 目标3：控制和监测测试过程

测试成熟度模型 (TMM) (2)

■ TMM定义了5个级别 (续)

- 第四级：管理和度量级 (**Management and Measurement**)

- 目标1：建立一个面向组织的评价程序
- 目标2：建立一个技术培训程序
- 目标3：建立一个测试度量程序
- 目标4：软件质量评价

- 第五级：优化/缺陷预防和质量控制 (**Optimization/Defect Prevention and Quality Control**)

- 目标1：应用缺陷预防数据过程
- 目标2：质量控制

测试能力成熟度模型（TCMM）

- TCMM定义了5个级别，与CMM的级别完全一样
 - 第一级：初始级（Initial）
 - 第二级：可重复级（Repeatable）
 - 第三级：已定义级（Defined）
 - 第四级：已管理级（Managed）
 - 第五级：优化级（Optimizing）

3. 学科发展 (11)

■ 从测试的思想导向来划分为4个阶段

- 1957~1978年，以功能验证为导向，测试是证明软件是正确的（正向思维）
- 1978~1983年，以破坏性为导向，测试是为了找到软件中的错误（逆向思维）
- 1983~1987年，以质量评估为导向，测试是提供产品的评估和质量度量
- 1988年起，以缺陷预防为导向，测试是为了展示软件符合设计要求，发现缺陷、预防缺陷



更好的阶段划分

- 分为3个阶段——初期阶段、发展阶段和成熟阶段
 - 初级阶段（1957~1971）测试通常被认为是对产品进行事后检验，缺乏有效的测试方法
 - 发展阶段（1972~1982），1972年第一次关于软件测试的正式会议，促进了软件测试的发展
 - 成熟阶段（1983到现在），国际标准Std 829-1983，形成一门独立的学科和专业，成为软件工程学科中的一个重要组成部分



软件测试的正面性



■ Bill Hetzel博士（正向思维的代表）

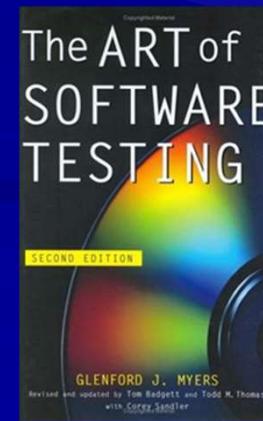
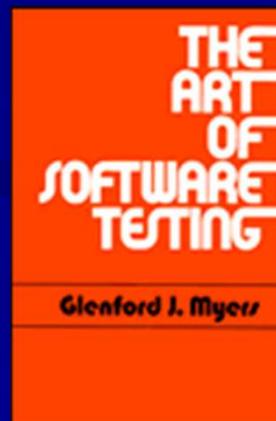
- 软件测试就是为程序能够按预期设想那样运行而建立足够的信心
- “软件测试是一系列活动以评价一个程序或系统的特性或能力并确定是否达到预期的结果”
- 测试是为了验证软件是否符合用户需求，即验证软件产品是否能正常工作



软件测试的反面性

■ Glenford J. Myers (反向思维的代表)

- 测试是为了证明程序有错，而不是证明程序无错误
- 一个好的测试用例是在于它能发现至今未发现的错误
- 一个成功的测试是发现了至今未发现的错误的测试



3. 学科发展 (12)

- 软件测试从一种纯粹的个人行为发展到有计划、有步骤的组织行为，并进一步系统化，融入到整个软件生产过程中，成为软件生产不可或缺的一部分，这充分显示了软件测试正一步步走向成熟

3. 学科发展 (13)

■ 业界的软件测试现状

- 与其他学科门类相比，软件测试的发展水平仍然处于“婴儿期”
 - 软件测试方法和软件测试工具无法完全适应快速应用开发的需要
 - 与软件测试相关的数学模型还有待完善
 - 并不是所有的测试工作都可以自动化
 - 通用测试工具的使用还存在不少局限
 - 对于大型的、涉及多个节点的分布式计算环境的测试，在理论和方法上还远未成熟

3. 学科发展 (14)

■ 业界的软件测试现状（续）

- 国际上每两年召开一次软件测试与分析研讨会，专门就软件测试与软件质量问题进行广泛的交流
- 国际会议：国际测试会议（ITC）、亚洲测试会议（ATS）、欧洲测试会议（ETC）、国际软件工程会议.....
- 我国每两年召开一次的全国软件工程会议和全国测试会议，将软件测试作为一个主要议题
- 全国容错计算会议都设有软件测试专题部分

我国软件测试产业现状（1）

- 国外成熟软件企业，如微软，软件开发人员与测试人员的比例约为1:2，而国内软件企业，平均8个软件开发工程师才对应1个软件测试工程师，比例严重失衡

我国软件测试产业现状（2）

- 随着软件外包行业的逐渐兴起和人们对软件质量保障意识的加强，中国软件企业已经开始认识到，软件测试的广度和深度决定了中国软件企业的前途命运
 - 例如：占中国软件外包总量近85%的对日软件外包企业，业务内容基本都针对测试环节。软件外包中对测试环节的强化，直接导致了软件外包企业对测试人才的大量需求

我国软件测试产业现状（3）

- 近年来，几乎所有的软件企业均存在不同程度的测试人才缺口，软件测试工程师已成为了亟待补充的关键技术工种之一
 - 据统计，国内软件测试工程师的缺口为30万之多
 - IBM、百度、华为、惠普、盛大网络、联想集团等国内外大型IT企业均表现出对成熟软件测试人才的期盼，而微软、三星、西门子、思科、华为3COM等多家国内外IT巨头则相继在全国各大高校招兵买马，并把软件测试人才的招聘放在了突出的位置。国内软件测试人才紧缺的现状已经凸现出来

4. 职业素质 (1)

■ 测试工程师应具备的素质

- 人是测试工作中最有价值也是最重要的资源
- 让那些经验最少的新手，没有效率的开发者或不适合干其他工作人去做测试工作，是一种目光短浅的行为

4. 职业素质 (2)



4. 职业素质（3）

■ 测试工程师应具备的素质（续）

- 沟通能力
- 移情能力
- 技术能力
- 自信心
- 外交能力
- 幽默感

4. 职业素质 (4)

■ 测试工程师应具备的素质（续）

- 很强的记忆力
- 耐心
- 怀疑精神
- 自我监督
- 洞察力
- 分析与表达

4. 职业素质 (5)

■ 测试工程师的技能组成



4. 职业素质 (6)

■ 人员特征

合适的人员特征	不合适的人员特征
是以测试作为自己的职业	轻视测试工作，是以开发工作为职业
职业表现主义者（产品导向）	容易妥协、害怕争论
是能承担签章的测试工作	不能完成艰苦的测试工作
善于观察、能识别细微的差别	粗心大意、忽视一些差别
逻辑性强、思维缜密、表达清晰	思维粗糙、不周全、不能清楚表达
自我管理能力、不偏离工作主线	不能把握工作重点、陷于枝微末节
掌握必要的技术背景及技能	技术背景弱

4. 职业素质 (7)

- 测试工程师的类型（微软公司）
 - 测试组的软件开发工程师 (**Software Development Engineer at Test, SDET**)
 - 具备编写代码的能力和开发工具软件的经验，在测试小组中主要负责自动化测试工具和测试脚本的开发

4. 职业素质 (8)

■ 测试工程师的类型（微软公司）（续）

- 软件测试工程师 (**Software Test Engineer, STE**)：具体负责测试软件产品，保证软件质量的工程师，可以细分为
 - **BVT测试工程师 (BVT Tester)**：负责保证每日生成的软件版本可顺利执行，确认已开发完成的所有功能模块都已连入产品，且主要功能正确无误
 - **功能测试工程师 (Function Tester)**：负责对软件产品的某个特定组件或某一组产品特性进行测试

4. 职业素质 (9)

■ 测试工程师的类型（微软公司）（续）

- 可用性测试工程师（**Usability Tester**）：负责测试产品中有操作流程、用户界面相关的部分，确保产品在最终使用方式上满足用户的需求
- 测试专家（**Ad hoc Tester**）：测试经验丰富，对产品的体系架构和实现方法了如指掌，能够使用各种规范或不规范的方法对产品进行攻击的特殊测试人员

4. 职业素质 (10)

- 软件测试职业的发展
 - 初级软件测试工程师
 - 软件测试工程师
 - 高级软件测试工程师
 - 软件测试组负责人
 - 软件测试负责人
 - 软件测试/质量保证经理

5. 软件测试

- 5.1 软件测试的必要性
- 5.2 软件测试的定义

5.1 软件测试的必要性（1）

- 软件总存在缺陷。只有通过测试，才可以发现软件缺陷。也只有发现了缺陷，才可以将软件缺陷从软件产品或软件系统中清理出去
- 软件中存在的缺陷给我们带来的损失是巨大的，这也说明了软件测试的必要性和重要性

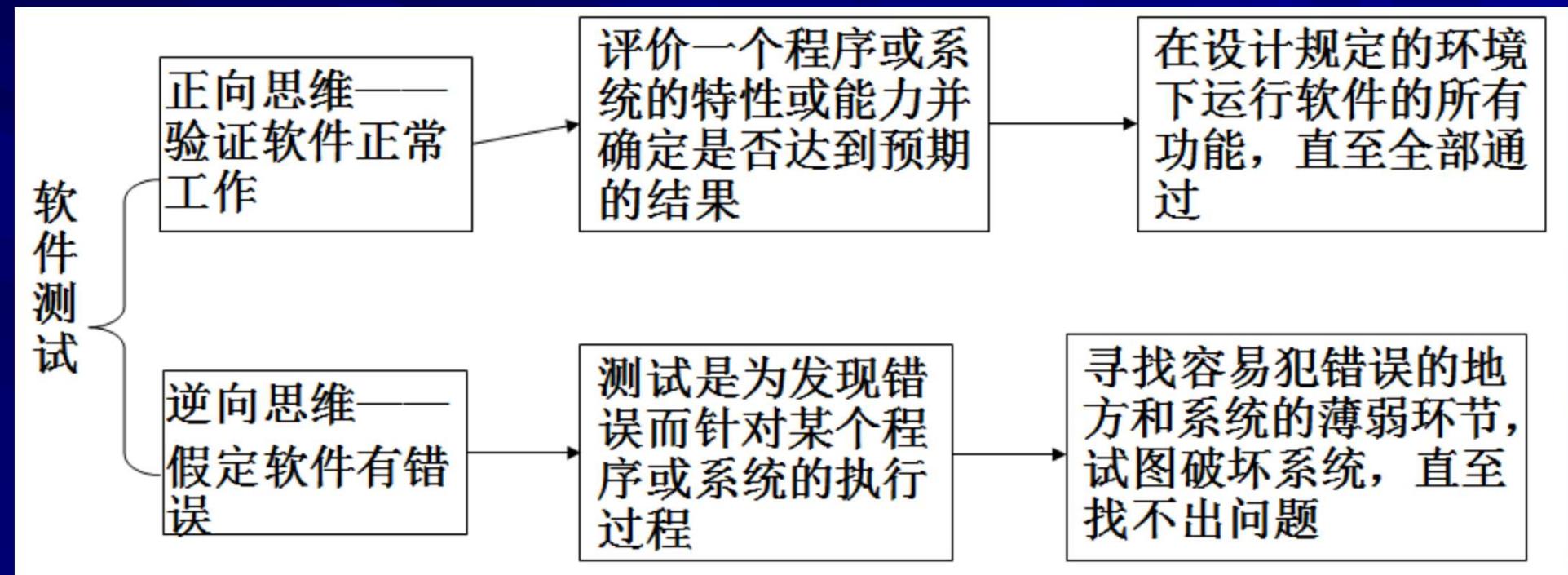
5.1 软件测试的必要性（2）

- 测试是所有工程学科的基本组成单元，自然也是软件开发的重要组成部分
- 测试人员水平越高，找到软件问题的时间就越早，软件就越容易更正，产品发布之后越稳定，公司赚的钱也越多

5.2 软件测试的定义

- 5.2.1 软件测试定义的两面性
- 5.2.2 软件测试的定义
- 5.2.3 软件测试的其他观点

5.2.1 软件测试定义的两面性



5.2.2 软件测试的定义（1）

■ IEEE的定义

- 在特定的条件下运行系统或构件，观察或记录结果，对系统的某个方面做出评价
- 分析某个软件项以发现现存的和要求的条件之差别（即错误）并评价此软件项的特性

5.2.2 软件测试的定义（2）

- 软件测试是由“验证（Verification）”和“有效性确认（Validation）”活动构成的整体
 - “验证”是检验软件是否已正确地实现了产品规格书所定义的系统功能和特性
 - “有效性确认”是确认所开发的软件是否满足用户真正需求的活动

5.2.3 软件测试的其他观点

- 软件测试被认为是对软件系统中潜在的各种风险进行评估的活动。基于风险的软件测试可以被看作是一个动态的监控过程，对软件开发全过程进行检测，随时发现问题、报告问题
- 测试的经济观点就是以最小的代价获得最高的软件产品质量。经济观点也要求软件测试尽早开展工作，发现缺陷越早，返工的工作量就越小，所造成的损失就越小

6. 软件缺陷

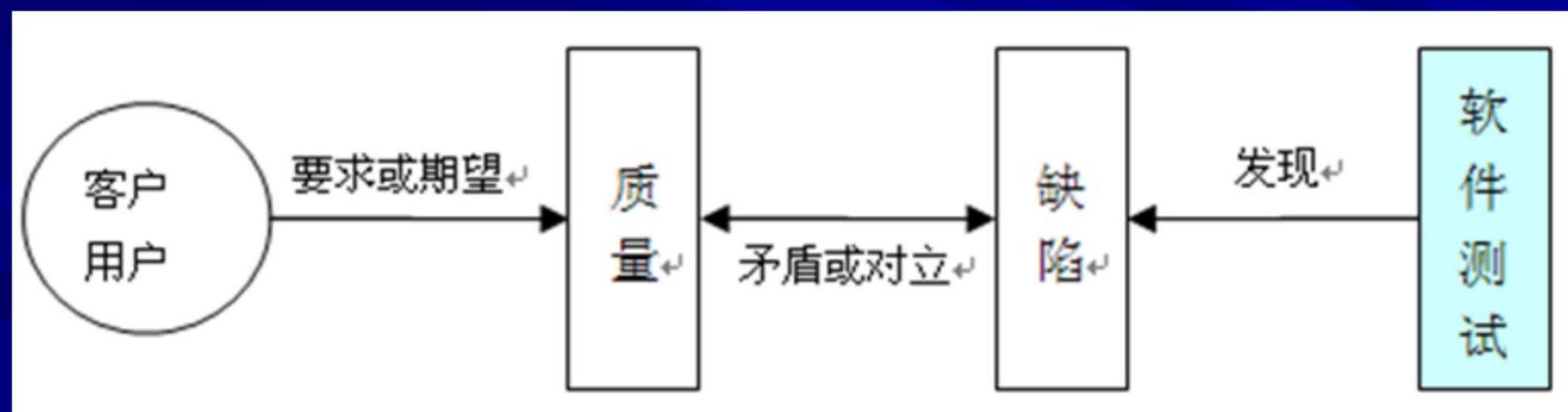
- 6.1 软件质量的内涵
- 6.2 软件缺陷的定义
- 6.3 软件缺陷的描述
- 6.4 软件缺陷的产生
- 6.5 软件缺陷的构成
- 6.6 软件缺陷的修复

6.1 软件质量的内涵

- 6.1.1 缺陷是质量的对立面
- 6.1.2 什么是软件质量
- 6.1.3 高质量软件标准体系
- 6.1.4 产品质量标准
- 6.1.5 软件质量特征

6.1.1 缺陷是质量的对立面

要了解什么是缺陷（defect），就必须清楚“质量（quality）”概念，因为缺陷是相对质量而存在的，违背了质量、违背了客户的意愿，不能满足客户的要求，就会引起缺陷或产生缺陷



6.1.2 什么是软件质量（1）

- 美国电子与电气工程师联合会（IEEE）的软件质量定义
 - 系统、部件或过程满足规定需求的程度
 - 系统、部件或过程满足顾客或用户需求或期望的程度

6.1.2 什么是软件质量（2）

- 由Pressman提出的软件质量定义

- 符合明确陈述的功能和性能需求、明确文档化了的开发标准和所有专业开发软件预期的隐含特性

6.1.2 什么是软件质量 (3)

- Pressman的定义为质量保证提出了要由开发者满足的三个要求
 - 特定功能需求，它主要是指软件系统的输出
 - 在合同中提到的软件质量标准
 - 反映当今水平的专业方法的良好的软件工程方法（Good Software Engineering Practice, GSEP）得以满足，即使在合同中没有明确提到
- Pressman的定义提供了测试满足需求程度的操作方向

6.1.3 高质量软件标准体系（1）

■ 产品质量

- 是人们实践产物的属性和行为，是可以认识，可以科学地描述的。并且可以通过一些方法和人类活动，来改进质量
- 质量模型
 - 鲍曼（Boehm）模型
 - 麦考（McCall）模型

6.1.3 高质量软件标准体系（2）

■ 过程质量

- 软件能力成熟度模型 CMM (Capability Maturity Model)
- 国际标准过程模型 ISO 9000
- 软件过程改进和能力决斷 SPICE (Software Process Improvement and Capability dEtermination)

6.1.3 高质量软件标准体系（3）

- 在商业过程中有关的质量内容
 - 培训、成品制作、宣传、发布日期、客户、风险、成本、业务等

6.1.4 产品质量的标准（1）

- 功能性 **Functionality**
- 可用性 **Usability**（简单安装； 轻松使用；
友好界面）
- 可靠性 **Reliability**（用户使用的根本）
- 性能 **Performance**
- 容量 **Capacity**

6.1.4 产品质量的标准（2）

- 可测量性 Scalability
- 可维护性 Service manageability
- 兼容性 Compatibility
- 可扩展性 Extensibility

6.1.5 软件质量特征（1）

■ ISO 9126

— 功能

- 与一组功能及其指定性质有关的一组属性，这里的功能是满足明确或隐含的需求的那些功能

— 可靠

- 在规定的一段时间和条件下，与软件维持其性能水平的能力有关的一组属性

— 易用

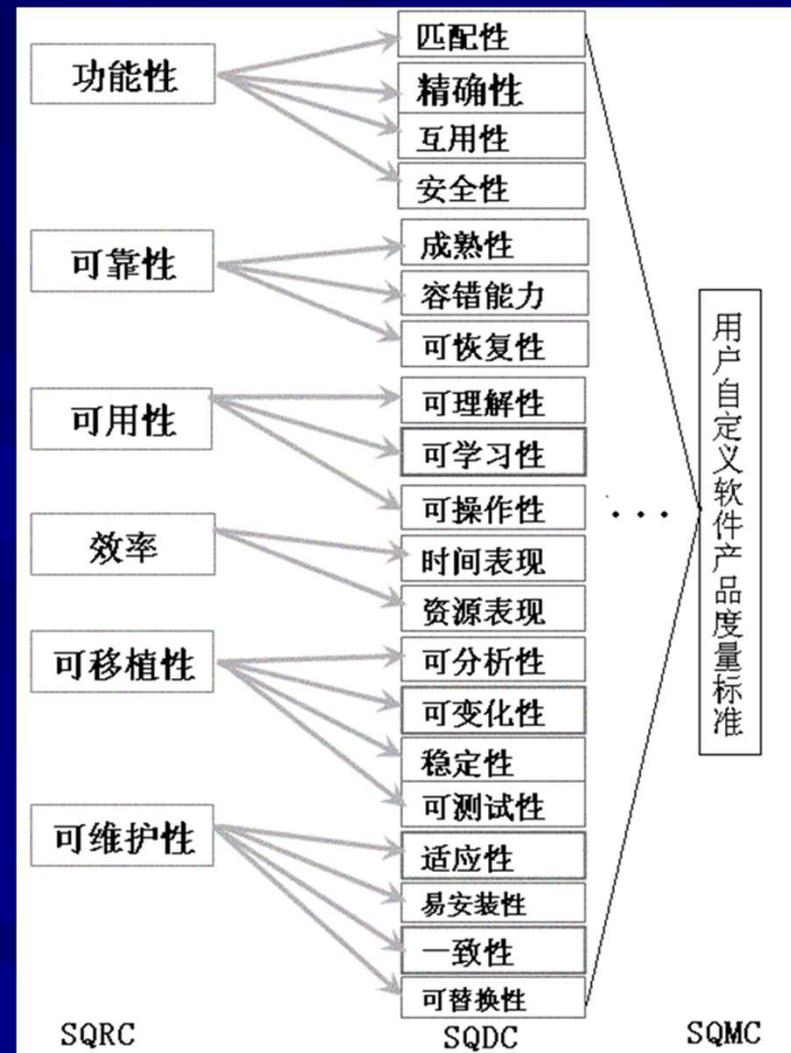
- 由一组规定或潜在的用户为使用软件所需做的努力和所做的评价有关的一组属性

6.1.5 软件质量特征（2）

■ ISO 9126（续）

- 效率
 - 与在规定条件下软件的性能水平与所使用资源量之间关系有关的一组属性
- 可维护
 - 与进行指定的修改所需的努力有关的一组属性
- 可移植
 - 与软件从一个环境转移到另一个环境的能力有关的一组属性
- 其中每一个质量特征都分别与若干子特征相对应

ISO 9126软件质量模型



6.2 软件缺陷的定义（1）

■ 产品说明书

- 有时简称为说明和产品说明，通常是利用文字和图形描述产品的书面文档，是软件开发小组的协定。它对开发的产品进行定义，包括产品有何细节、如何操作、功能如何、有何限制

6.2 软件缺陷的定义（2）

■ 符合下列5个规则叫软件缺陷

1. 软件未达到产品说明书标明的功能
2. 软件出现了产品说明书指明不会出现的错误
3. 软件功能超出产品说明书指明范围
4. 软件未达到产品说明书虽未指出但应达到的目标
5. 软件测试人员认为软件难以理解、不易使用、运行速度慢，或者最终用户认为不好

■ 缺陷是对软件产品预期属性的偏离现象

6.3 软件缺陷的描述 (1)

- 缺点 (**defect**)
- 偏差 (**variance**)
- 谬误 (**fault**)
- 失败 (**failure**)
- 问题 (**problem**)



6.3 软件缺陷的描述 (2)

- 矛盾 (inconsistency)
- 错误 (error)
- 毛病 (incident)
- 缺陷 (bug)
- 异常 (anomaly)



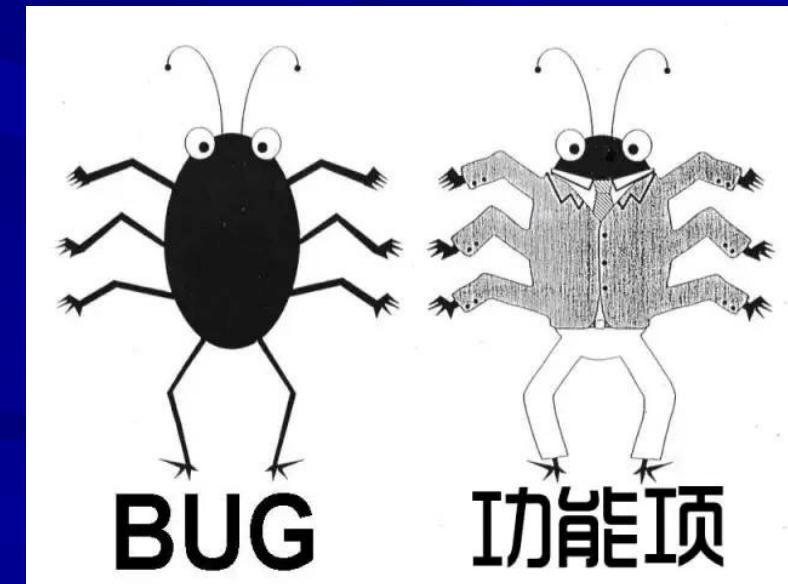
6.3 软件缺陷的描述（3）

■ 英文参考书和认证教材词汇表

- 缺陷（Defect）
- 缺陷率（Defect Rate）
- 故障（Failure）

6.3 软件缺陷的描述（4）

- 描述软件失败的术语与公司的文化和公司用于开发软件的过程相关。使用的术语常常反映了小组的个性，及其寻找、报告问题的方法
- 最常用的描述：Bug
 - Bug的历史渊源



Bug的历史渊源

- 1945年9月9日，著名的女科学家格雷斯·霍珀（Grace M. Hopper）在现代计算机的始祖——Mark I 机器的继电器中找到了一只小蛾子，正是这只小蛾子导致Mark I 机器停止运行。霍珀将小蛾子的标本贴在系统运行日志的纸页上，并在其下注释道：“发现了第一例真正的小虫子（First actual case of bug being found）”。从那时起，Bug的说法开始在程序员中流传并盛行

世界第一只计算机BUG

Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 Anctan started

1000 " stopped - anctan ✓

13' w.c (032) MP - MC

(033) PRO 2

Relays 6-2 in 033 failed special speed test
in relay

1100 Started Cosine Tape (Sine check)

1525 Started Multi Adder Test.

1545 Relay #70 Panel F
(moth) in relay.

1630 Anctan started.

1700 closed down.

First actual case of bug being found.

Relay 2145
Relay 3370



6.4 软件缺陷的产生

■ 技术问题

- 算法错误，语法错误，计算和精度问题，接口参数传递不匹配
- 没有考虑系统的自我恢复或数据的异地备份、灾难性恢复等问题

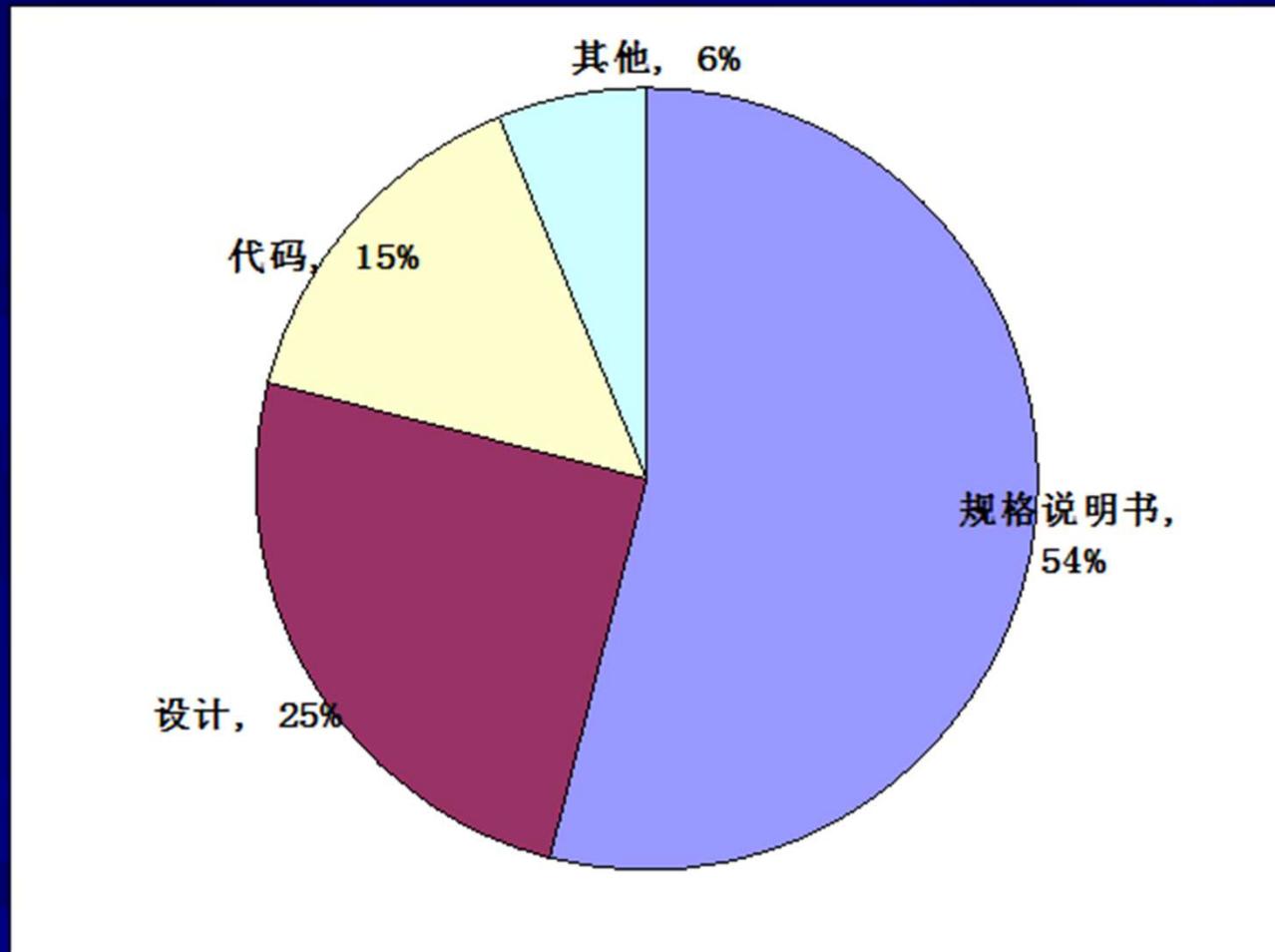
■ 团队工作

- 误解、沟通不充分

■ 软件本身

- 文档错误、用户使用场合（user scenario）
- 时间上不协调、或不一致性所带来的问题

6.5 软件缺陷的构成



6.6 软件缺陷的修复（1）

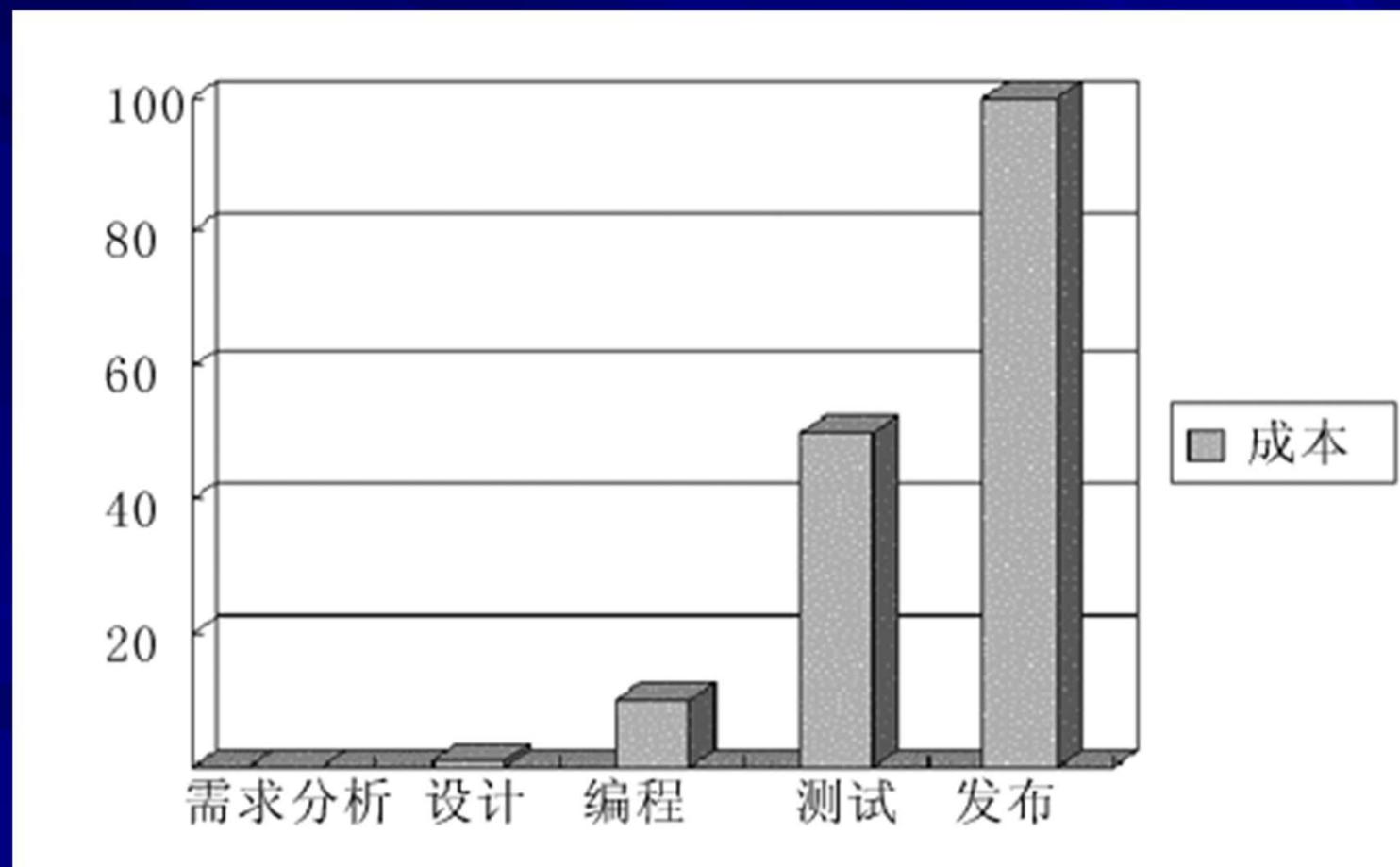
- 从开始到计划、 编制、 测试， 一直到公开使用的过程中， 都有可能发现软件缺陷
- 随着时间推移， 修复软件缺陷的费用呈几何数量级地增长

6.6 软件缺陷的修复（2）

- 在真正的程序测试之前，通过审查、评审会可以发现更多的缺陷
- 规格说明书的缺陷会在需求分析审查、设计、编码、测试等过程中会逐步发现，而不能在需求分析一个阶段发现

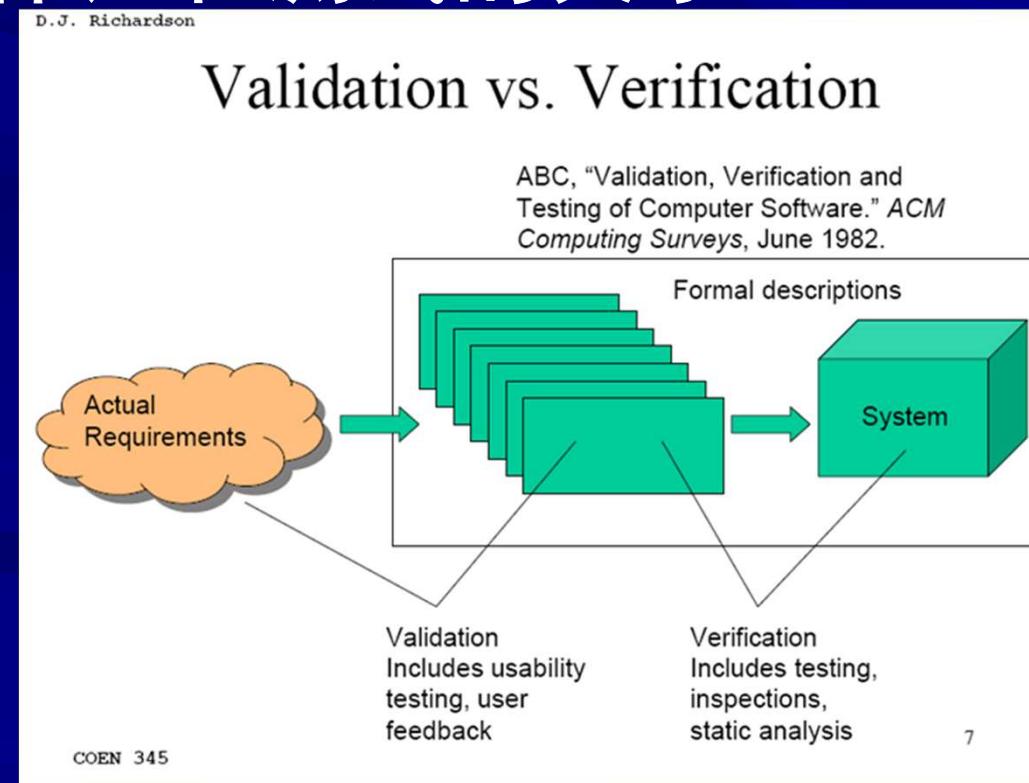
6.6 软件缺陷的修复（3）

■ 缺陷成本



7. 验证确认

- 7.1 验证和确认
- 7.2 评审
- 7.3 质量保证和测试的关系



7.1 验证和确认 (V & V)

■ Verification

- Are we building the product right?
- 是否正确地构造了软件？即是否正确地做事，验证开发过程是否遵守已定义好的内容。验证产品满足规格设计说明书的一致性

■ Validation

- Are we building the right product?
- 是否构造了正是用户所需要的软件？即是否正在做正确的事。验证产品所实现的功能是否满足用户的需求

7.2 评审

- 7.2.1 评审
- 7.2.2 评审分类

7.2.1 评审

- 通过软件评审，可以更早地发现需求工程、软件设计等各个方面的问题，大大减少大量的后期返工，将质量成本从昂贵的后期返工转化为前期的缺陷发现
- 评审是对软件元素或者项目状态的一种评估手段，以确定其是否与计划的结果保持一致，并使其得到改进。检验工作产品是否正确地满足了以往工作产品中建立的规范

7.2.2 评审分类

- 管理评审
- 技术评审



7.3 质量保证和测试的关系

- 7.3.1 软件质量保证的定义和目标
- 7.3.2 质量保证和软件测试的区别
- 7.3.3 质量保证和软件测试的关系

7.3.1 软件质量保证定义和目标 (1)

- 美国电子与电气工程师联合会（IEEE）的软件质量保证（**Software Quality Assurance, SQA**）定义
 - 一种有计划的、系统化的行动模式，它是为项目或产品符合已有技术需求提供充分信任所必需的
 - 设计用来评价开发或制造产品的过程的一组活动。与质量控制有区别

7.3.1 软件质量保证定义和目标 (2)

■ 软件质量保证活动的目标

— 软件开发（面向过程）

- 以可接受的置信度确保软件符合其功能技术需求
- 以可接受的置信度确保软件符合管理上的进度安排与预算要求
- 启动与管理软件开发和软件质量保证活动的改进和更高效率的活动。这意味着在减少进行软件开发和软件质量保证活动的费用的同时增加对达到功能和管理方面需求的预期

7.3.1 软件质量保证定义和目标 (3)

■ 软件质量保证活动的目标（续）

- 软件维护（面向产品）

- 以可接受的置信度确保软件维护活动符合其功能技术需求
- 以可接受的置信度确保软件维护活动符合其管理上的进度安排与预算要求
- 启动与管理改善和提高软件维护和软件质量保证活动效率的活动，这包括在降低费用的同时，增加达到功能和管理方面需求的预期

7.3.2 质量保证和软件测试的区别（1）

- 软件测试人员的一项重要任务是提高软件质量，但不等于说软件测试人员就是软件质量保证人员，因为测试只是质量保证工作中的一个环节。软件质量保证和软件测试是软件质量工程的两个不同层面的工作

7.3.2 质量保证和软件测试的区别（2）

- 质量保证（QA）的重要工作通过预防、检查与改进来保证软件质量。QA采用“全面质量管理”和“过程改进”的原理开展质量保证工作。所关注的是软件质量的检查与测量。虽然QA的活动中也有一些测试活动，但所关注的是软件质量的检查与测量

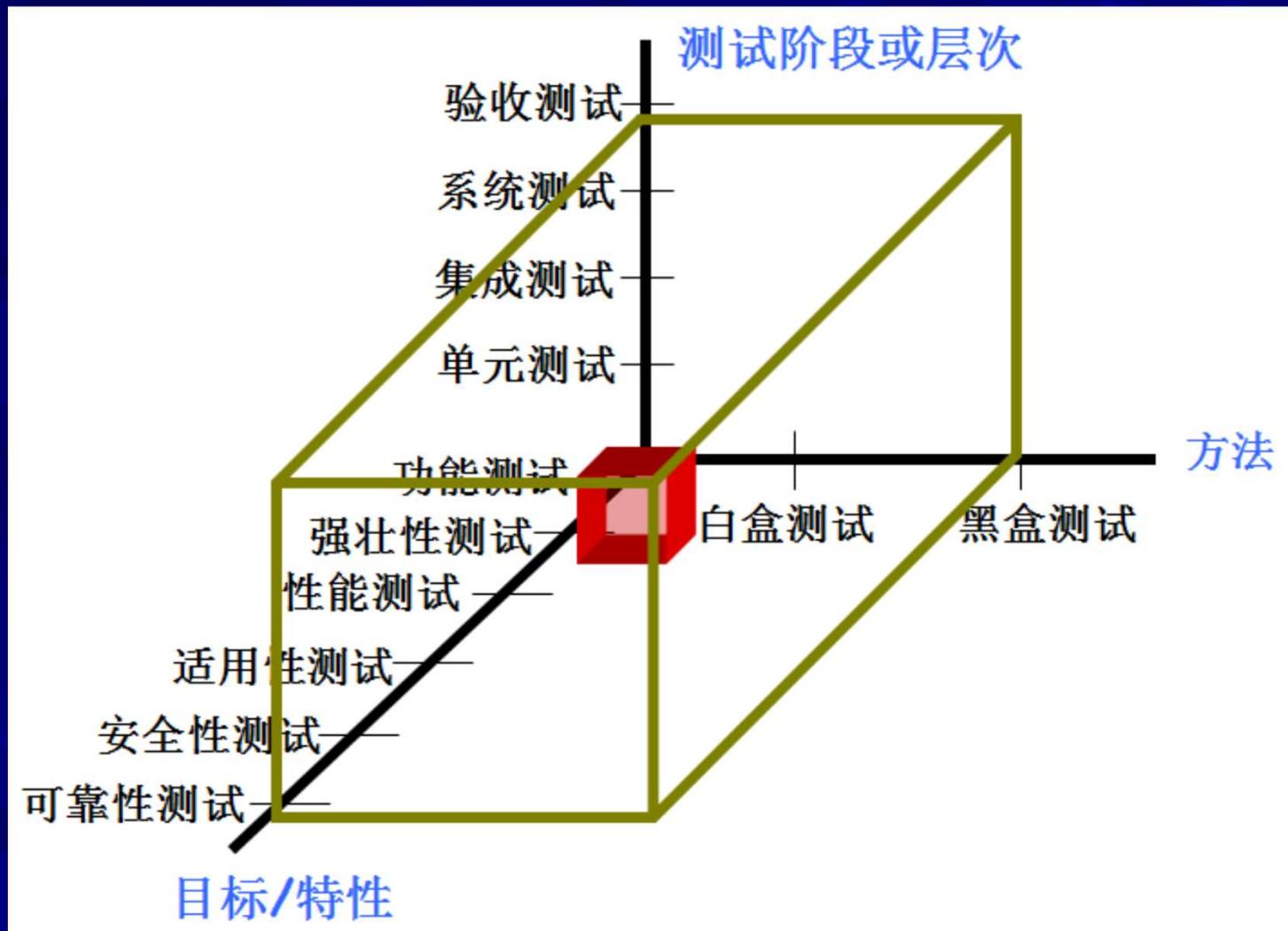
7.3.2 质量保证和软件测试的区别（3）

- 测试虽然也与开发过程紧密相关，但关心的不是过程的活动，而是对过程的产物以及开发出的软件进行剖析。测试人员要“执行”软件，对过程中的产物——开发文档和源代码进行走查，运行软件，以找出问题，报告质量

7.3.3 质量保证和软件测试的关系

- SQA是管理工作、审查对象是流程、强调以预防为主
- 测试是技术工作、测试对象是产品、主要是以事后检查
- SQA指导测试、监控测试
- 测试为SQA提供依据

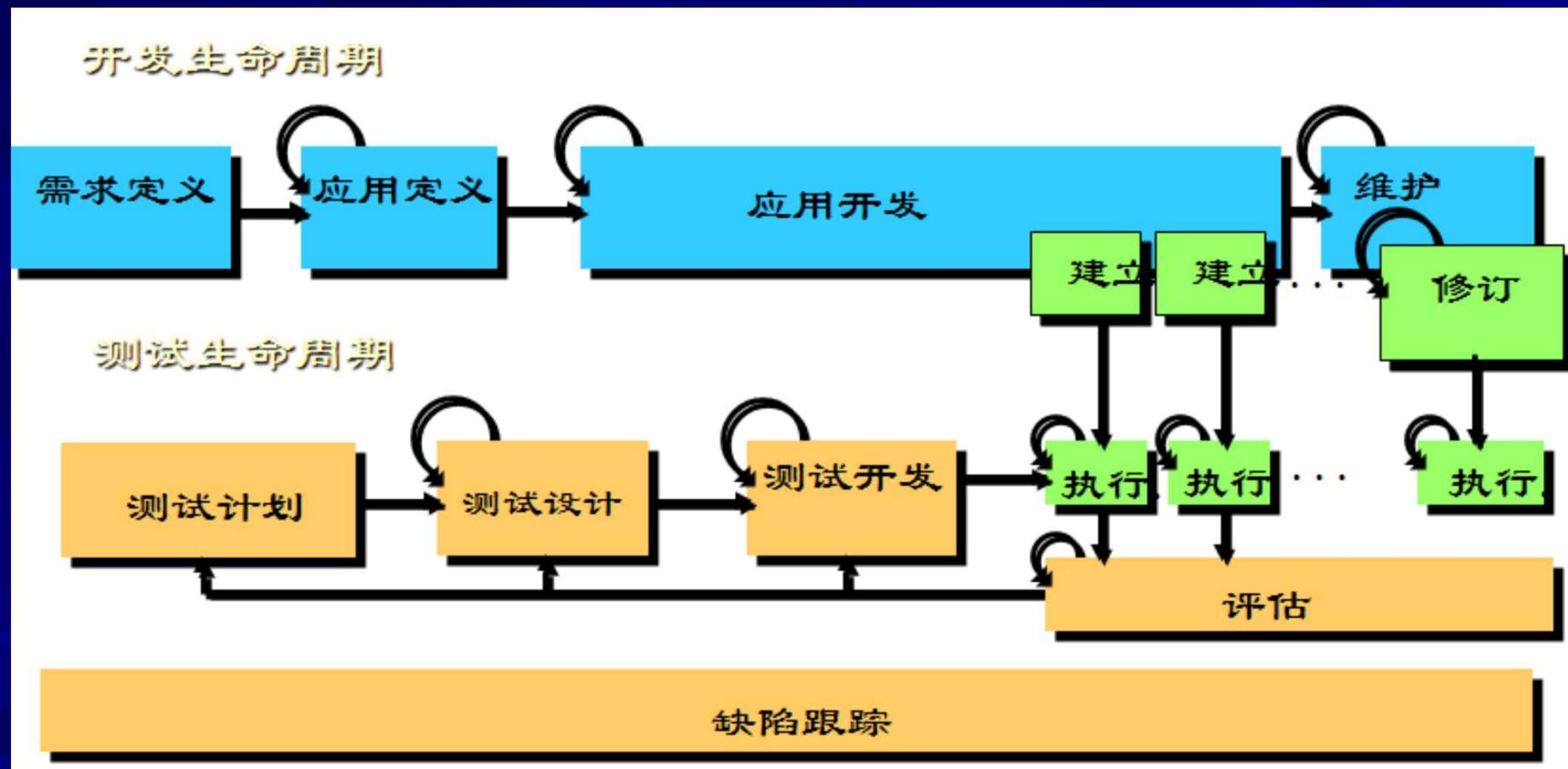
8. 测试分类 (1)



8. 测试分类 (2)

- 按测试的阶段或层次分类，如单元测试、集成、系统测试等
- 按测试目的分类，如功能测试、性能测试、可靠性测试、安全性测试和兼容性测试等
- 根据测试过程中被测软件是否被执行，分为静态测试和动态测试
- 根据是否针对系统的内部结构和具体实现算法来完成测试，可分为白盒测试和黑盒测试

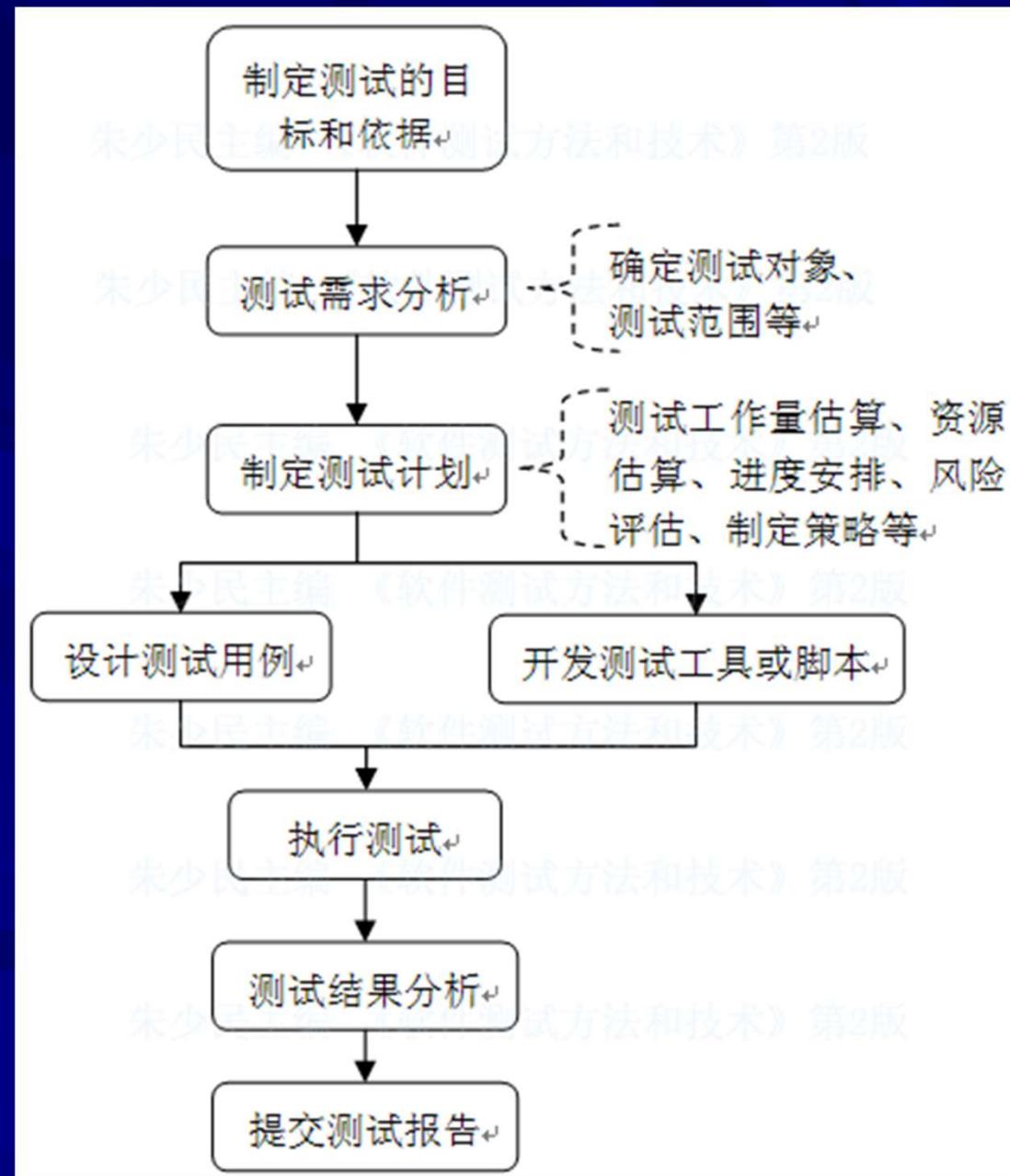
9. 测试阶段



10. 工作范畴

- 软件测试工作的组织与管理：制定测试策略、测试计划，确认所采用的测试方法与规范，控制测试进度，管理测试资源
- 测试工作的实施：编制符合标准的测试文档，搭建测试环境，开发测试脚本、与开发组织协作实现各阶段的测试活动

测试工作流程



参考资料 (1)

- Paul Ammann and Jeff Offutt,
Introduction to Software Testing,
Cambridge University Press, 2008
- (美) 威廉·派瑞 (William E. Perry) 兰
德尔·莱斯 (Randall W. Rice) 著, 周震译
, 软件测试求生法则, 北京: 清华大学出
版社, 2004

参考资料 (2)

- 第1~2章，朱少民 主编，软件测试方法和技术（第4版），北京：清华大学出版社，2022
- 第1章，宫云战主编，软件测试教程（第3版），北京：机械工业出版社，2022
- （美）Ron Patton著，周予滨 姚静等译，软件测试，北京：机械工业出版社，2003

参考资料 (3)

- Ron Patton, **Software Testing (Second Edition)**, Sams Publishing, 2005
- William E. Lewis, **Software Testing and Continuous Quality Improvement (Gunasekaran Veerapillai, Technical Contributor, Second Edition)**, Auerbach Publications, A CRC Press Company, 2005

参考资料 (4)

- 飞思科技产品研发中心编著, 实用软件测试方法与应用, 北京: 电子工业出版社, 2003
- Andreas Spillner, Tilo Linz & Hans Schaefer, **Software Testing Foundations: A Study Guide for the Certified Tester Exam, Second Edition, Posts & Telecom Press, 2008**

参考资料（5）

- 张旸旸 于秀明, 软件评测师教程（第2版），北京: 清华大学出版社，2021
- （以）Daniel Galin著, 王振宇 陈利 王志海等译, 软件质量保证, 北京: 机械工业出版社, 2004
- Aditya P. Mathur, **Foundations of Software Testing: Fundamental Algorithms and Techniques**, Pearson Education, 2008

Thank You