



软件测试概念

崔义

第1章回顾

① 什么是软件测试

② 软件测试的正反两面性

- 验证软件
- 发现缺陷
- V&V

① 软件测试和开发的关系

② **TDD**



目 录

C O N T E N T S

01

软件项目

02

软件过程

03

软件缺陷

04

软件测试行业

05

测试认知误区

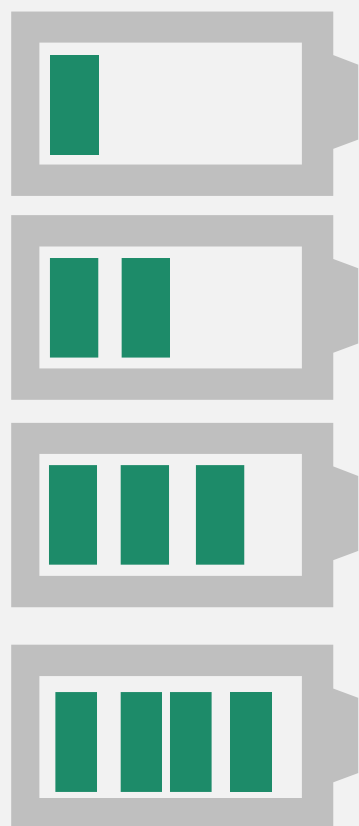
01

软件项目



01

软件



编程软件

系统软件

应用软件

系统软件和应用软件的
中间件

软件的定义

一系列按照特定顺序组织的计算机
数据和指令的集合

1

运行时，能够提供所要求功能和性能的指令或计算机程序集合

2

程序能够满意地处理信息的数据结构

3

描述程序功能需求以及程序如何操作和使用所要求的文档

升级

系统更新换代难度大

规范

开发人员之间管理不规范，
约定不严密，文档书写不完整，
使得软件维护费用高，某些系
统甚至无法进行修改

质量检查

缺乏严密有效的质量检测手段，
交付给用户的软件质量差，运行中
出现许多问题，甚至带来严重的后
果。

工期

软件项目无法按期完成，超
出经费预算，软件质量难以控制



01

项目特性

项目具有临时性。

项目是逐步完善。

项目具有一定的不确定性

项目具有明确的目标。

项目具有独特性。

项目使用的资源是受到限制的



软件项目的特性

知识密集型，技术含量高

客户化程度高

涉及多个专业领域，多种技术综合应用

项目范围和目标的灵活性

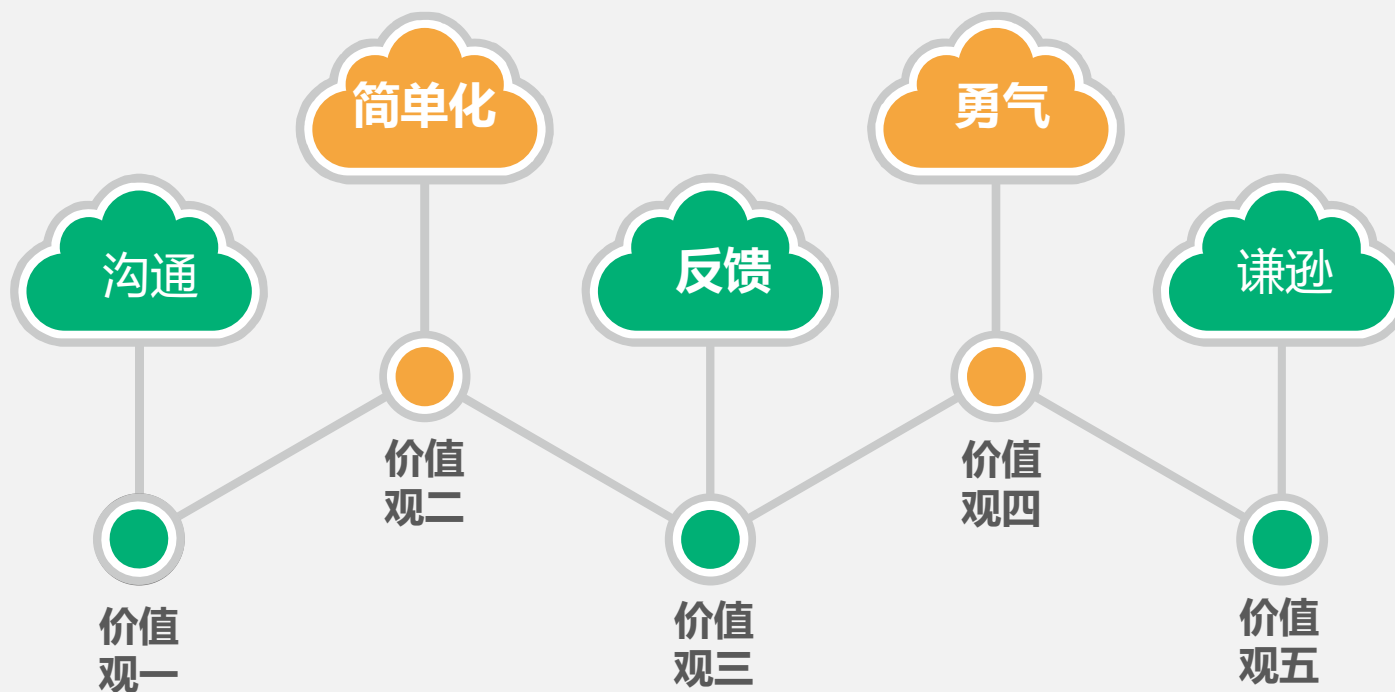
风险大，收益大

过程管理的重要性

02

软件过程





统一软件过程（RUP）：一种典型的软件过程模式，采用迭代增量式，以架构为中心和用例驱动软件开发方法，以统一建模语言（UML）来描述软件开发过程。

*RUp*的核心 workflow



6

部署

8

项目管理

9

环境

7

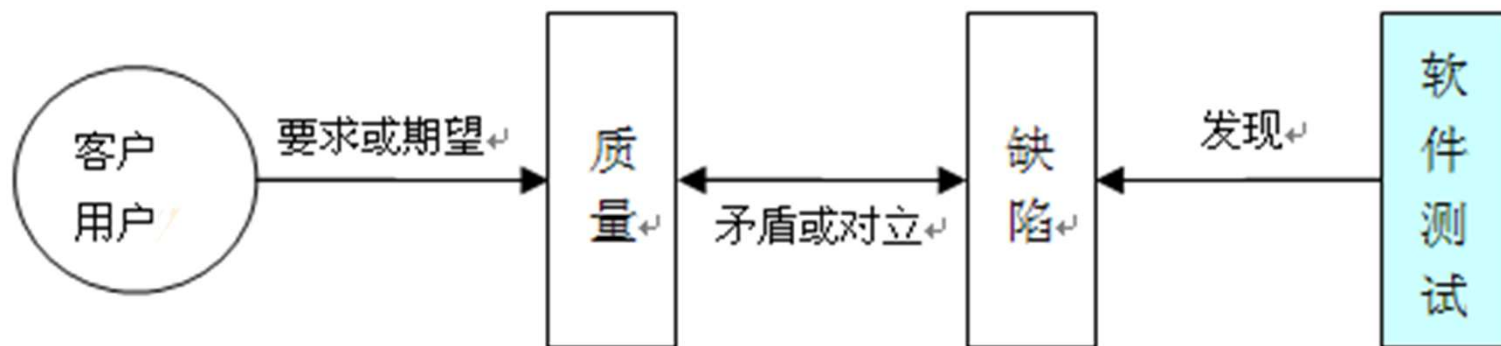
配置与变更管理

03

软件缺陷



缺陷是质量的对立面



什么是质量？



质量 = 品牌 = 客户满意度

软件质量 的内涵

IEEE: 质量是系统、部件或过程满足

① 明确需求

② 客户或用户需要或期望的程度不同

- 软件质量：软件产品具有**满足规定的**或**隐含**要求能力要求有关的特征与特征总和 (ISO 8492)
- 软件质量：软件产品满足使用要求的程度

高质量软件标准体系

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：McCall 模型, Boehm 模型, ISO 9126 模型

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断 SPICE (Software Process Improvement and Capability dEtermination)

在商业过程中有关的质量内容:

培训、成品制作、宣传、发布日起、客户、风险、成本、业务等

产品质量的标准

- 功能性 Functionality
- 可用性 Usability
- 可靠性 Reliability
- 性能 Performance
- 容量 Capacity
- 可伸缩性 Scalability
- 可维护性 Service manageability
- 兼容性 Compatibility
- 可扩展性 Extensibility

非功能特性



软件质量特征 (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作的努力和所作的评价有关的一组属性。

软件质量特征 (ISO 9126)

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:

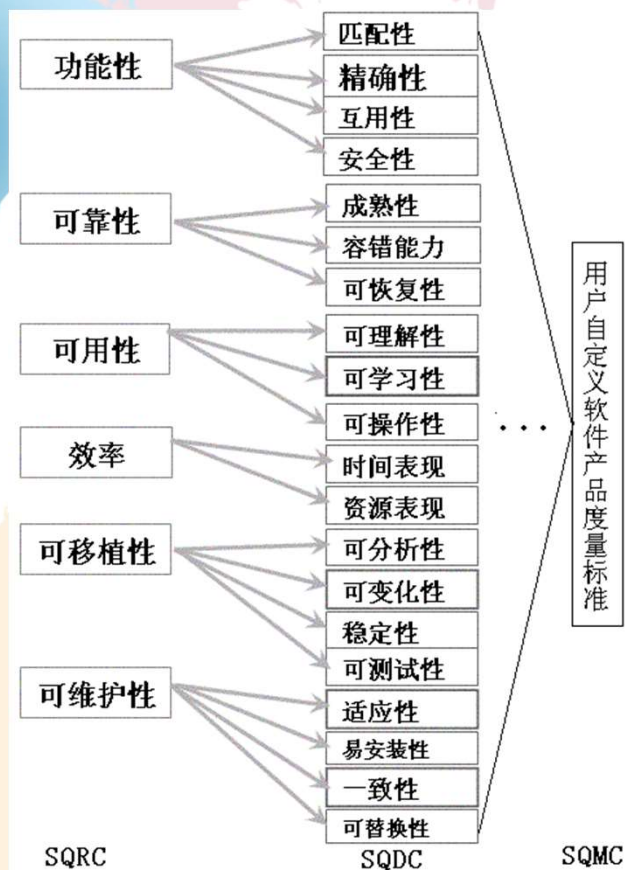
与进行指定的修改所需的努力有关的一组属性。

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软件质量三层模型



1.高层：软件质量需求评价准则（SQRC）

2.中层：软件质量设计评价准则（SQDC）

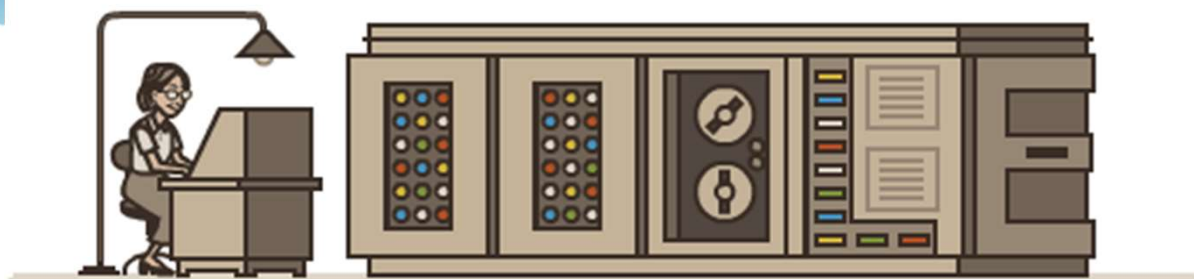
3.底层：软件质量年度评价准则

软件缺陷的定义

任何程序、系统中的问题，和产品设计书的不一致性，
不能满足用户的需求



第一个BUG



领域的表现

随着计算机技术的进步，计算机在多个领域的应用越来越广泛。

领域的表现和应用

“第一个计算机 bug”竟是一只飞蛾

缺陷 – Defect, Bug

与缺陷相近的词：

缺点（defect）

谬误（fault）

问题（problem）

错误（error）

异常（anomaly）

偏差（variance）

失败（failure）

矛盾（inconsistency）

毛病（incident）

软件缺陷



软件未达到软件规格说明书中规定的功能



软件出现了产品说明书中指明不会出现的错误;



软件功能超出了产品说明书中指定的范围



软件测试人员认为软件难于理解，不易使用，运行速度慢，或者最终用户认为软件使用效果不好

软件缺陷的现象

- ❑ 功能、特性没有实现或部分实现
- ❑ 设计不合理，存在缺陷
- ❑ 实际结果和预期结果不一致
- ❑ 运行出错，包括运行中断、系统崩溃、界面混乱
- ❑ 数据结果不正确、精度不够
- ❑ 用户不能接受的其他问题，如存取时间过长、界面不美观

软件缺陷的产生

1、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

2、团队工作

沟通不充分，误解

3、项目管理

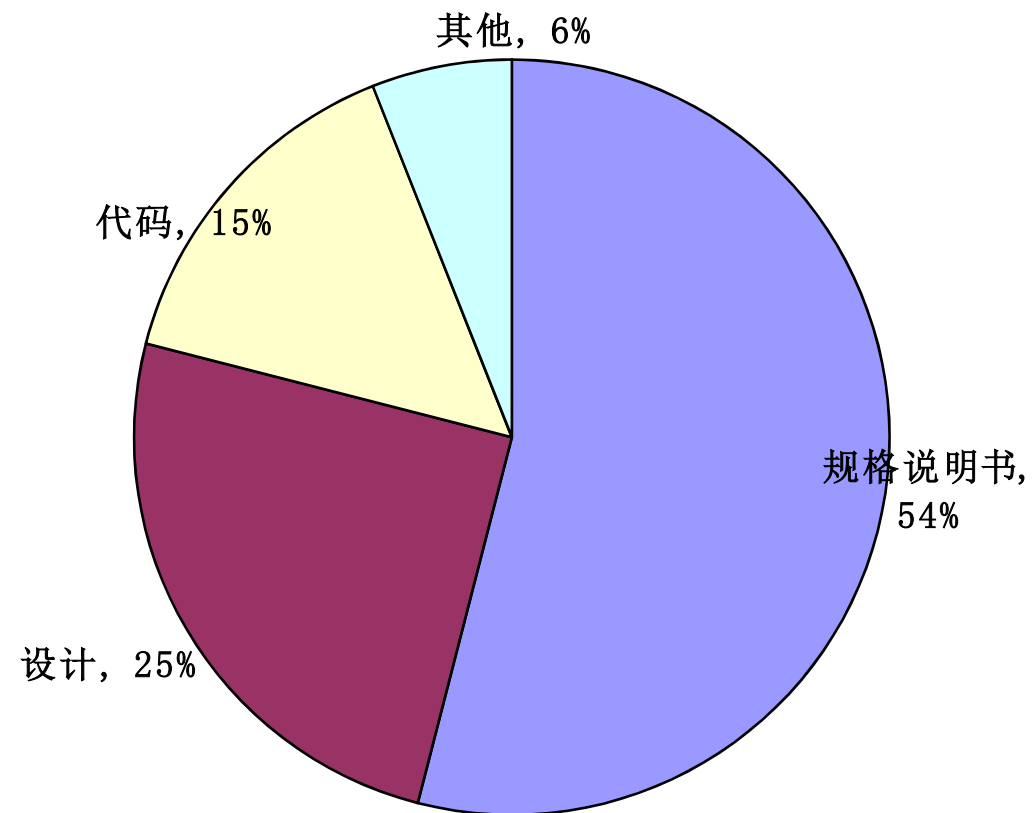
文档错误、用户使用场合(user scenario)，

时间上不协调、或不一致性所带来的问题

系统的自我恢复或数据的异地备份、灾难性恢复等问题

4、复审阶段：不合理或没有

软件缺陷构成



软件缺陷在不同阶段的分布

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

维护缺陷的成本与之成反比

软件缺陷类型列表

缺陷类型	描述
功能	影响了各种系统功能、逻辑的缺陷
用户界面	影响了用户界面、人机交互特性，包括屏幕格式、用户输入灵活性、结果输出格式等方面的缺陷
文档	影响发布和维护，包括注释，用户手册，设计文档
软件包	由于软件配置库、变更管理或版本控制引起的错误
性能	不满足系统可测量的属性值，如执行时间，事务处理速率等。
系统/模块接口	与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表等不匹配、冲突。

软件缺陷严重等级列表

缺陷严重等级	描述
致命	系统任何一个主要功能完全丧失、用户数据受到破坏、系统崩溃、悬挂、死机，或者危及人身安全
严重	系统的主要功能部分丧失、数据不能保存，系统的次要功能完全丧失，系统所提供的功能或服务受到明显的影响
一般	系统的次要功能没有完全实现，但不影响用户的正常使用。例如：提示信息不太准确；或用户界面差、操作时间长等一些问題。
较小	使操作者不方便或遇到麻烦，但它不影响功能的操作和执行，如个别的不影响产品理解的错别字、文字排列不对齐等一些小问题。

缺陷产生可能性列表

缺陷产生可能性	描述
总是	总是产生这个软件缺陷，其产生的概率是 100%
通常	通常情况下会产生这个软件缺陷，其产生的概率大概是 80-90%
有时	有的时候产生这个软件缺陷，其产生的概率大概是 30-50%
很少	很少产生这个软件缺陷，其产生的概率大概是 1-5%

软件缺陷优先级列表

缺陷优先级	描述
立即解决	缺陷导致系统几乎不能使用或测试不能继续，需立即修复
高优先级	缺陷严重，影响测试，需要优先考虑
正常排队	缺陷需要正常排队等待修复
低优先级	缺陷可以在开发人员有时间的时候被纠正。

软件缺陷状态列表

缺陷状态	描述
激活或打开	问题还没有解决，存在源代码中，确认“提交的缺陷”，等待处理，如新报的缺陷。
已修正或修复	已被开发人员检查、修复过的缺陷，通过单元测试，认为已解决但还没有被测试人员验证
关闭或非激活	测试人员验证后，确认缺陷不存在之后的状态。
重新打开	测试人员验证后，还依然存在的缺陷，等待开发人员进一步修复
推迟	这个软件缺陷可以在下一个版本中解决
保留	由于技术原因或第三者软件的缺陷，开发人员不能修复的缺陷
不能重现	开发不能复现这个软件缺陷，需要测试人员检查缺陷复现的步骤

软件缺陷来源列表

缺陷来源	描述
需求说明书	需求说明书的错误、或不清楚引起的问题
设计文档	设计文档描述不准确、和需求说明书不一致的问题
系统集成接口	系统各模块参数不匹配、开发组之间缺乏协调引起的缺陷
数据流(库)	由于数据字典、数据库中的错误引起的缺陷
程序代码	纯粹在编码中的问题所引起的缺陷

软件缺陷根源列表

缺陷根源	描述
测试策略	错误的测试范围，误解了测试目标，超越测试能力等
过程，工具和方法	无效的需求收集过程，过时的风险管理过程，不适用的项目管理方法，没有估算规程，无效的变更控制过程等。
团队	项目团队职责交叉，缺乏培训，没有经验等。
组织和通讯	缺乏用户参与，职责不明确，管理失败等。
硬件	硬件配置不对、缺乏，或处理器缺陷导致算术精度丢失，内存溢出等
软件	软件设置不对、缺乏，或操作系统错误导致无法释放资源，工具软件的错误，编译器的错误， 2000 千年虫问题等。
工作环境	组织机构调整，预算改变，工作环境恶劣，如噪音过大。

缺陷生命周期

一个缺陷从它产生到终结的过程，称之为缺陷生命周期。

每个缺陷都具有7种生命状态：

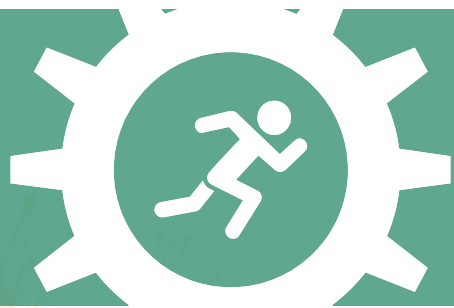
缺陷初始状态、缺陷分配状态、缺陷重新分配状态、
缺陷修复状态，缺陷验证状态、缺陷重新打开状态、
缺陷关闭状态。

缺陷引入和移除活动

开发阶段	缺陷引入阶段	缺陷移除阶段
需求	需求说明过程及需求规格说明书开发	需求分析和评审
高层设计	设计工作	高层设计审查
详细设计	设计工作	详细设计审查
实现	编码	代码复查
测试	不正确的修复	测试

缺陷移除效率计算公式:

$$\text{缺陷移除效果} = \frac{\text{该阶段中移除的缺陷数量}}{\text{该阶段开始存在的缺陷数} + \text{该阶段中引入的缺陷数}} * 100\%$$



软件测试行业

04

软件测试职业

技术方向

初级软件测试工程师

中级软件测试工程师

高级软件测试工程师

管理方向

测试负责人（4-6年）1-3人

测试经理（10年）8+人

测试思维方式

逆向思维方式

组合思维方式

全局思维方式

两极思维方式

比较思维方式

发散思维方式

测试认识的误区

误区一

- 使用了测试工具，就是进行了有效测试

误区二

- 存在太多无法测试的东西

误区三

- 软件开发完成后进行软件测试

误区四

- 测试发布后发现质量问题，是测试人员的问题

测试认识的误区

误区五

- 软件测试要求不高，随便找个人就可以

误区六

- 软件测试是测试人员的事情，与程序员无关

误区七

- 项目紧的时候少测试，时间多的时候多测试

误区八

- 软件测试是低级工作，软件开发才是高级工作

THANK YOU

