

测试目的

a) 验证软件是否满足软件开发合同或项目开发计划、系统 / 子系统设计文档、软件需求规格说明、软件设计说明和软件产品说明等规定的软件质量要求；
b) 通过测试，发现软件缺陷；
c) 为软件产品的质量测量和评价提供依据。

测试类别a) 单元测试；b) 集成测试；c) 配置项测试（也称软件合格性测试或确认测试）；d) 系统测试；e) 验收测试。
可根据软件的规模、类型、完整性级别选择执行测试类别。回归测试可出现在上述每个测试类别中，并贯穿于整个软件生存周期。

软件测试过程一般包括四项活动：测试策划、测试设计、测试执行、测试总结。计划与控制、分析与设计、实施与执行、评估出口准则和报告、测试结束活动

测试用例设计原则a) 基于测试需求的原则.b) 基于测试方法的原则.c) 兼顾测试充分性和效率的原则。d) 测试执行的可再现性原则。

测试的准入/准出条件如下：0

a) **准入条件** 1) 具有测试合同（或项目计划）；2) 具有软件测试所需的各种文档；3) 所提交的被测软件受控；4) 软件源代码正确通过编译或汇编。

b) **准出条件** 1) 已按要求完成了受控（或项目计划）所规定的软件测试任务；2) 实际测试过程遵循了原定的软件测试计划和软件测试说明；3) 客观、详细地记录了软件测试过程和软件测试中发现的所有问题；4) 软件测试文档齐全、符合规范；5) 软件测试的全过程自始至终在控制下进行；6) 软件测试中的问题或异常有合理解释或正确有效的处理；7) 软件测试工作通过了测试评审；8) 全部测试软件、被测软件、测试支持软件和评审结果已纳入配置管理。

评审测试计划和测试说明。评审的具体内容和要求应包括：

a) 评审测试文档内容的完整性、正确性和规范性；b) 通过比较测试环境与软件真实运行的软件、硬件环境的差异，评审测试环境要求是否正确合理，满足测试要求；c) 评审测试活动的独立性；d) 评审测试用例选择的完整性和合理性；e) 评审测试用例的可行性、正确性和充分性。
在测试完成后，评审**测试过程**和**测试结果**的有效性，确定是否达到测试目的。主要对测试记录、测试报告进行评审，其具体内容和要求应包括：
a) 评审文档和记录内容的完整性、正确性和规范性；b) 评审测试活动的独立性和有效性；c) 评审测试环境是否符合测试要求；d) 评审测试记录、测试数据以及测试报告内容与实际测试过程结果的一致性；e) 评审实际测试过程与测试计划和测试说明的一致性；f) 评审未测试项和新增测试项的合理性；g) 评审测试结果的真实性和正确性；h) 评审对测试过程中出现的异常进行处理的正确性。

测试策略：（1）测试从模块开始，然后扩大延伸到整个基于计算机的系统集合中。（2）不同的测试技术适用于不同的时间点。（3）测试是由软件的开发人员和（对于大型系统而言）独立的测试组来管理的。4) 测试和调试不是不同的活动，但是调试必须能够适应任何测试策略。

测试充分性标准：单调性、非复合性、非分解性、复杂性、回报递减率

单调性如果 一个软件系统在一个测试数据集合上的测试是充分的，那么再多测试一些数据也应该是充分的

非复合性即使对软件所有成分都进行了充分的测试，也并不表明整个软件的测试已经充分了

非分解性即使对软件系统整体的测试是充分的，也并不意味软件系统中各个成分都已经充分得到了测试
软件测试的充分性应该与软件的需求和软件的实现都相关。

复杂性软件越复杂，需要的测试数据就越多。
回报递减率测试过得越多，进一步测试所能得到的充分性增长就越少。

系统测试

是将通过确认测试的软件，作为整个基于计算机系统的的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其它系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的测试。

为什么要进行系统测试？

由于软件只是计算机系统中的—个组成部分，软件开发完成之后，最终还要和系统中的硬件系统、某些支持软件、数据信息等其他部分配套运行。因此，在投入运行前要完成系统测试，以保证各组成部分不仅能单独地得到检验，而且在系统各部分协调工作的环境下也能正常工作。

测试对象系统测试的对象是完整的、集成的计算机系统。重点是新开发的软件配置项的集合。

测试目的在于通过与系统的需求定义作比较，发现软件与系统的定义不符合或与之矛盾的地方。系统测试的测试用例应根据需求分析规格说明来设计，并在实际使用环境下来运行。

系统测试意义保证各组成部分不仅能单独地受到检验，而且在系统各部分协调工作的环境下也能正常工作。

1) 软件在系统中占有重要的位置，软件测试的好坏与能否顺利、成功地完成系统测试关系极大。

2) 系统测试实际上是针对系统中各个组成部分进行的综合性检验。尽管每一个检验有着特定的目标，然而所有的检测工作都要验证系统中每个部分均已得到正确的集成，并能完成指定的功能。

测试内容主要从：功能性、可靠性、易用性、效率、维护性、可移植性和依从性等方面（可负载）来考虑。

响应时间：指系统为完成—项规定任务所需的时间；**平均响应时间：**指系统执行若干并行任务所需的平均时间；**响应极限时间：**指在最大负载条件下，系统完成某项任务需要时间的极限；

吞吐量：指在给定的时间周期内系统能成功完成任务数量；**平均吞吐量：**指在一个单位时间内系统能处理并发任务的平均数；**极限吞吐量：**指在最大负载条件下，在给定的时间周期内，系统能处理的最多并发任务数；**周转时间：**指从发出一条指令开始到—组相关的任务完成的时间；**平均周转时间：**指在一个特定的负载条件下，对一些并发任务，从发出请求到任务完成所需要的平均时间；**周转时间极限：**指在最大负载条件下，系统完成—项任务所需要时间的极限。

系统测试完成后形成的文档一般应有：a) 系统测试计划；b) 系统测试说明；c) 系统测试报告；d) 系统测试记录和 / 或测试日志；e) 系统测试问题报告，总结报告
附录 A
静态测试方法：代码审查、代码走查、静态分析（控制流分析、数据流分析、接口分析、表达式分析）
动态测试方法：黑盒白盒，单元测试一般用白盒，配置项和系统测试用黑盒。

RUSP 可以是一种打包出售给对其特征和其他质量没有任何影响的需方的软件产品。
产品说明的内容: 1)标识和标示 2)功能性陈述 3)可靠性陈述 4)易用性陈述 5)效率陈述 6)可维护性陈述 7)可移植性陈述 8) 使用质量陈述

用户文档集要求

编写原则完备性、正确性、一致性、易理解性、易学性、可操作性

软件质量要求 1. 功能性 2. 可靠性 3. 易用性 4. 效率 5. 维护性 6. 可移植性 7. 使用质量

测试文档包括测试计划、测试说明（需要时进一步细分为测试设计说明、测试用例说明和测试规程说明）、测试项传递报告、测试日志、测试记录、测试问题报告和测试总结报告

面向对象

开发前期 面向对象分析的测试 面向对象设计的测试 面向对象编程的测试

开发后期的面向对象测试 3.1 面向对象的单元测试 3.2 面向对象的集成测试 3.3 面向对象的确认测试 3.4 面向对象的系统测试

面向对象=对象+类+继承+通信(Coad 和 Yourdon 定义) 新特点：封装、继承、多态

面向对象和传统测试区别

1）基本功能模块

系统的基本构造单元不再是传统的功能模块，而是类和对象。在测试过程中，不能仅检查输入数据产生的输出结果是否与预期结果相吻合，还要考虑对象的状态变化，方法间的相互影响等。

2）系统的功能实现

系统的功能体现在对象间的协作上，而不再是简单的过程调用。原有集成测试所要求的逐步将开发的模块搭建在一起进行测试的方法已成为不可能。

3）封装对测试的影响

封装使对象的内部状态隐蔽，如果类中未提供足够的存取函数来表明对象的实现方式和内部状态，则类的信息隐蔽机制将给测试带来困难。

4）继承对测试的影响

继承削弱了封装性，产生了类似于非面向对象语言中全局数据的错误风险。由于继承的作用，一个函数可能被封装在具有继承关系的多个类中，子类中还可以对继承的特征进行覆盖或重定义。若一个类得到了充分的测试，当其被子类继承后，继承的方法在子类的环境中的行为特征需要重新测试。

5）多态对测试的影响

多态依赖于不规则的类层次的动态绑定，可能产生非预期的结果。某些绑定能正确的工作但不能保证所有的绑定都能正确地运行。以后绑定的对象可能很容易将消息发送给错误的类，执行错误的功能，还可能导致一些与信息序列和状态相关的错误。

OOA 测试的重点在其完整性和冗余性。OOA 对问题领域分析抽象的不完整，最终会影响软件的功能实现，导致软件开发后期大量不可避免的修补工作；而一些冗余的对象或结构会影响类的选定、程序的整体结构或增加程序员不必要的工作量。因此，OOA 测试的重点在其完整性和冗余性。

OOA 阶段的测试划分为以下五个方面：1) 对认定的类的测试 2) 对认定的结构的测试 3) 对认定的主题的测试 4) 对定义的属性和实例关联的测试 5) 对定义的服务和信息关联的测试
认定的结构分为两种：分类结构和组装结构。分类结构，又称泛化结构，体现了问题空间中，对象的一般与特殊的关系。组装（复合）结构体现了问题空间中，对象的整体与局部的关系。

对 OOD 的测试，应从以下三方面考虑： • 对认定的类的测试 • 对构造的类层次结构的测试 • 对类库的支持的测试（复用性测试）

面向对象编程的测试面向对象程序是把功能的表现分布在类中 with 某种设计功能相关的一组对象，通过对对象提供的服务和对象之间的消息传递，共同协作来实现这个功能。这种面向对象程序风格，可将出现的错误精确定位在某一个具体的对象上。

因此，在面向对象编程(OOP)阶段，忽略类功能实现的细则，将测试的目光集中在类功能的实现和相应的面向对象程序风格，主要体现为以下两个方面，数据成员是否满足数据封装的要求、类是否实现了要求的功能

面向对象的集成测试

为什么集成测试目的发现与接口有关的问题

集成策略：一次性组装(先分散测试，再集中起来一次完成集成测试

• 传统的集成测试，有两种方式通过集成完成的功能模块进行测试：自顶向下集成、自底向上集成

• 当开发面向对象系统时，集成的层次并不明显。而当一组对象类通过组合行为提供一组服务时，则需将它们一起测试，这就是簇测试。此时不存在自底向上和自顶向下的集成。

面向对象系统的集成测试 3

1) 类层次测试 2) 基于线程的测试 3) 对象交互测试

好测试用例的特点：完整、准确、清晰简洁、可维护、适当、可复用、设计测试用例基本准则：代表性、可判定性、可再现性

黑盒测试分为：

功能测试(等价类划分、边界值分析、因果图法、判定表法、场景法、正交实验法，随机测试法、错误推测法)、非功能测试(配置/安装测试、兼容性测试、互操作性测试、文档和帮助测试、性能测试、可靠性测试、易用性测试和界面测试)

黑盒测试特点测试人员不需了解程序内部的代码和实、测试用例设计不依赖于系统内部的设计与实现、测试用例设计基于功能的定义和需求说明书、从用户角度出发进行测试，能较容易知道用户的需求和用户会遇到的问题、关注于测试数据的选择和测试结果的分析、对测试人员的编程技术要求不高、在自动化测试时较为方便、不易发现代码部分的缺陷

白盒(结构或逻辑驱动测试(语句覆盖、判定覆盖(支覆盖)、条件覆盖、判定/条件覆盖、条件组合覆盖、基本路径测试))
白盒测试的优缺点

1. **优点**迫使测试人员去仔细思考软件的实现、可以检测代码中的每条分支和路径、揭示隐藏在代码中的错误、对代码的测试比较彻底 2. **缺点**昂贵、无法检测代码中遗漏的路径和数据敏感性错误、不验证规格的正确性
怎样做好文档测试?文档测试：是检查用户文档(如需求、用户手册、安装手册等)的正确性、清晰性和精确性，检验文档是否和实际应用存在的差别。对用户文档中所使用的例子必须在测试中一一试过，确保叙述正确无误。

对文档的测试**正确性、完备性和易理解性**几个方面，具体包括以下内容：（1）检查产品说明书属性、（2）检查是否完整、（3）检查是否准确、（4）检查是否精确、（5）检查是否一致、（6）检查是否贴切、（7）检查是否合理、（8）检查代码无关、（9）检查可测试性。

V 模型：

需求分析	验收测试
概要测试 (开发之后测试)	系统测试
详细设计	集成测试
编码	单元测试

V 模型价值：非常明确标明测试过程中的不同级别,并且描述了这些测试阶段和开发过程各阶段的对应关系

V 模型问题：测试是开发之后的一个阶段、测试的对象是程序本身、易导致需求阶段的错误一直到最后系统测试阶段才被发现

W 模型

是 V 模型测试的改进，在概要设计、详细设计和编码每个步骤都要进行测试。主要思想：尽早地 and 不断地进行软件测试。

优点：强调了测试计划等工作的先行和对系统需求和设计的测试。

缺点：没有对软件测试流程予以说明

H 模型

H 模型将测试作为一个独立流程，贯穿整个开发周期，与其他流程并行，同时测试准备和测试执行分离。

特性：1 测试不仅仅指测试的执行，还包括许多其他活动；2 测试是一个独立流程，贯穿产品整个生命周期；与其它流程并行 3 测试要尽早准备，尽早执行 4 测试是根据被测对象的不同而分层进行。

意义：1 测试准备和测试执行分离，有利于资源调配，降低成本，提高效率；2 充分体现测试过程（不是技术）的复杂性；3 有组织、有结构化的独立流程，有助于跟踪测试投入的流向

自动化测试的意义：软件测试工作量大，重复性高

自动化测试的原理和方法：软件测试自动化实现的基础、软件测试自动化的原理和方法（主要有：直接对代码进行静态和动态分析、测试过程的捕获和回放、测试脚本技术、虚拟用户技术和测试管理技术。）

1) **代码分析：**代码分析类似于高级语言编译系统，一般针对不同的高级语言去构造分析工具，在工具中定义类、对象、函数、变量等定义规则、语法规则；在分析时对代码进行语法扫描，找出不符合编码规范的地方；根据某种质量模型评价代码质量，生成系统的调用关系图等。

2) **捕获和回放**（代码分析是一种白盒测试的自动化方法，捕获和回放则是一种黑盒测试的自动化方法）：**捕获**时将用户每一步操作都记录下来所有的记录转换为一种脚本语言所描述的过程，以模拟用户的操作；**回放**时，将脚本语言所描述的过程转换为屏幕上的操作，然后将被测系统的输出记录下来同预先给定的标准转换比较。

3) **脚本技术：**脚本是一组测试工具执行的指令集，也是计算机程序的一种形式。可通过录制测试的操作产生，然后再做修改，这样可减少脚本编程的工作量。包括：线性脚本、结构化脚本、共享脚本、数据驱动脚本、关键字驱动脚本。

4) **自动比较：**1)静态比较(测试过程中不比较，将结果存入数据库)和动态比较 2) 简单比较(允许误差)和复杂比较

17. **自动化测试存在的一些问题和限制：**(1)不能取代手工测试 (2)手工测试比自动测试发现的缺陷更多 (3)对测试质量的依赖性极大 (4)测试自动化不能提高有效性 (5)测试自动化可能会制约软件开发。(6)工具本身并无想象力。

静态测试：只对被程序进行特性分析(文档评审、代码检查、代码度量)

包括检查单和静态分析方法 测试的内容与选择的测试方法有关

性能测试， 连接速度测试、负载测试、压力测试

调试与测试的区别：目的不同：（1）软件测试的目的是发现错误，至于找出错误的原因和错误发生的地方是调试的任务；调试的目的是为了证明程序的正确。（2）任务不同：测试贯穿于整个开发过程，而调试是编码活动的一部分，主要任务就是排错。（3）指导原则和方法不同。（4）操作者不同

单元测试

对软件中的最小可测试单元进行检查和验证。

单元测试的考虑

模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试、功能、性能、内存使用

回归测试：已进行过测试的某些子集再重新进行一遍，以保证上述改变不传播无法预料的副作用或引发新问题、

回归测试集包括三种类型的测试用例：

1)能够测试软件的所有功能(的代表性测试用例 2)专门针对可能会被修改而影响软件功能的附加测试

3)针对修改过的软件成分的测试

试。应避免此方式)、增值组装(自顶向下、自底向上、混合增值式)

确认测试：验证软件功能和性能及其特性是否与用户要求一致对象：包括有效性测试和软件配置审查

有效性测试(功能测试)：验证被测软件是否满足需求规格说明列出的需求

验收测试(交付测试,UC 根据需求分析规格说明设计,并在实际使用环境下来运行)：是否满足软件合同中的确认标准**对象：**

是完整的、集成的计算机系统**步骤:**确定测试计划;建立测试环境;准备测试数据;执行测试用例;记录测试结果;分析测试结果。
验测与系统测试区别：组织机构、测试地点、覆盖范围、实施人员

常用测试方法：文档测试、配置测试、外国语言测试、易用性测试、网站测试、对 Web 进行压力测试、测试和 β 测试、Client/Server 测试、实时系统测试、面向对象的软件测试。

质量保证和软测试关系：软件测试能够找出软件缺陷，确保软件产品满足需求。测试可以查找错误并进行修改，从而提高软件产品的质量。软件质量保证测试避免错误以求高质量，并且还有其他方面的措施以保证质量因素。