

软件测试基础教程

测试的基本概念

测试是软件生存周期中十分重要的一个过程，是产品发布、提交给最终用户前的稳定化阶段。

1、 测试的分类：

从测试方法的角度可以分为手工测试和自动化测试。

手工测试：不使用任何测试工具，根据事先设计好的测试用例来运行系统，测试各功能模块。

自动化测试：利用测试工具，通过编写测试脚本和输入测试数据，自动运行测试程序。目前最常用的自动化测试工具是基于 GUI 的自动化测试工具，基本原理都是录制、回放技术。

从整体的角度可以分为单元测试、集成测试、系统测试、确认测试。

单元测试：是针对软件设计的最小单位—程序模块，进行正确性检验的测试工作。一般包括逻辑检查、结构检查、接口检查、出错处理、代码注释、输入校验、边界值检查。

单元测试的依据是系统的详细设计；一般由项目组开发人员自己完成。

集成测试：在单元测试的基础上，将所有模块按照设计要求组装进行测试。一般包括逻辑关系检查、数据关系检查、业务关系检查、模块间接口检查、外部接口检查。

系统测试：系统测试是在所有单元、集成测试后，对系统的功能及性能的总体测试。

确认测试：模拟用户运行的业务环境，运用黑盒测试方法，验证软件系统是否满足用户需求或软件需求说明书中指明的软件特性（功能、非功能）上的。

从测试原理上分为：白盒测试、黑盒测试和灰盒测试。

白盒测试：是通过程序的源代码进行测试而不使用用户界面。这种类型的测试需要从代码句法发现内部代码在算法，溢出，路径，条件等等中的缺点或者错误，进而加以修正。

黑盒测试：是通过使用整个软件或某种软件功能来严格地测试，而并没有通过检查程序的源代码或者很清楚地了解该软件的源代码程序具体是怎样设计的。测试人员通过输入他们的数据然后看输出的结果从而了解软件怎样工作。在测试时，把程序看作一个不能打开的黑盒子，

在完全不考虑程序内部结构和内部

特性的情况下，测试者在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收和正确的输出。黑盒测试方法主要有等价类划分、边界值分析、因一果图、错误推测法。

等价类划分：

是把所有可能的输入数据,即程序的输入域划分成若干部分(子集),然后从每一个子集中选取少数具有代表性的数据作为测试用例.该方法是一种重要的,常用的黑盒测试用例设计方法.

1) 划分等价类：等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的.并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况:有效等价类和无效等价类.

有效等价类:是指对于程序的规格说明来说是合理的,有意义的输入数据构成的集合.利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能.

无效等价类:与有效等价类的定义恰巧相反.

设计测试用例时,要同时考虑这两种等价类.因为,软件不仅要能接收合理的数据,也要能经受意外的考验.这样的测试才能确保软件具有更高的可靠性.

边界值分析：

长期的测试工作经验告诉我们,大量的错误是发生在输入或输出范围的边界上,

而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以

查出更多的错误。

错误推测法：

基于经验和直觉推测程序中所有可能存在的各种错误,从而有针对性的设计测试用例的方法.错误推测方法的基本思想:列举出程序中所有可能有的错误和容易发生错误的特殊情况,根据他们选择测试用例.例如,在单元测试时曾列出的许多在模块中常见的错误.以前产品测试中曾经发现的错误等,这些就是经验的总结.还有,输入数据和输出数据为0的情况.输入表格为空格或输入表格只有一行.这些都是容易发生错误的情况.可选择这些情况下的例子作为测试用例。

灰盒测试：灰盒测试就像黑盒测试一样是通过用户界面测试，但是测试人员已经有所了解该软件或某种软件功能的源代码程序具体是怎样设计的。甚至于还读过部分源代码。因此测试人员可以有真对性地进行某种确定的条件/功能的测试。

从软件特性上分为功能测试和性能测试。

功能测试：是指为了确保软件系统功能实现的正确性，完整性和其他特性而进行的测试。

性能测试：是指为了评估软件系统的性能状况，和预测软件系统性能趋势而进行的测试和分析。

2、 BUG 的定义：

BUG：（小错误，缺陷，不足，过失 …）一个计算机 bug 指在计算机程序中存在的一个错误（error）、缺陷（flaw）、疏忽（mistake）或者故障（fault），这些 bug 使程序无法正确的运行。Bug 产生于程序的源代码或者程序设计阶段的疏忽或者错误。

Defect：（缺陷）在软件工程（Software Engineering）中，软件与它的需求（requirements）不一致，常常指软件无法正确完成需求所要求的功能，也称之为 bug。

Fault：（故障）被定义为存在于组件、设备或者子系统中异常的条件或者缺陷，常常会导致系统的失败。

Error：（错误）一个 error 是指编写错误的代码，通常是无意中造成的。一般有两类主要的错误，一是语法错误（syntax error），该类错误易于检测，因为代码在编译阶段无法解析而不能正常编译通过。另一个是逻辑错误（logical error），因为它与代码的实际执行密切相关所以不易发现。

二、 项目测试的规划

项目测试内容：

将项目测试分为项目开发阶段测试和项目完工验收测试两个部分。

开发阶段测试内容主要包括：模块功能测试、集成测试和文档检查。

模块功能测试：确保系统各功能模块能够正常运行，数据的 IPO 符合系统设计的要求。单元和模块功能满足需求定义。

集成测试：系统各模块组装后，根据业务流程的要求，能够正确地完成各业务功能，并且数据的处理和输出正确。

文档检查：在项目开发阶段，按照项目进度表，根据《项目文档测试规范与标准》，对提交的项目文档和记录（技术文档和管理文档）进行检查和验证，以符合公司质量管理体系和项目制度的要求，对于技术类文档的关键要素，验证是否能够达到通过标准。

完工验收测试内容主要包括：安装测试、功能验证、性能测试、需求验证、文档测试。完工验收测试实际上是项目在结项前的一个全面的检查和验证。可以作为项目结项的依据和放行条件。

需求测试：检查软件产品是否满足该项目的需求说明书中规定的功能需求，检查需求的完整性、一致性、最新性，该项测试重点是需求满足的完整性。

安装测试：根据项目提供的安装文档中的安装步骤，搭建系统运行环境，检查系统安装过程是否正确。可能包括数据库服务器的安装与配置、应用服务器、控件注册、客户端的安装与配置、应用程序的安装。

功能验证：按照需求说明书和系统概要设计，逐项检查各项功能（功能单元、功能模块）的可运行性和正确性。

文档测试：文档测试从项目立项时就开始了，实际上就是文档检查，包括规范性检查和有效性检查。目的是使项目相关的文档和记录既规范又有意义，不是为了应付的无用文件。对于技术文档如：需求说明书、概要设计、详细设计等，在技术评审时也进行了评测。用户文档，如安装手册、用户操作手册，根据文档检查规范进行。

性能测试：这部分测试的来源，严格来讲，取决于用户对软件特性的一些特定要求，另外，就是公司的开发部门对产品的一些基本的性能要求。若用户从业务的角度考虑，对软件产品本身有特定的非功能要求，则必须在软件需求说明书中加以说明，使之具有可度量和可测试性。对于一些多用户环境或数据处理能力和负载方面的测试，很难通过手工搭建测试环境来测试，所以可以参考使用一些专门的性能测试工具和手工测试相结合的方式。

项目测试的基本流程：

- 1、项目测试启动：项目立项后，在测试配置库中创建项目。

2、 测试计划：系统详细设计后，制定测试计划，准备测试资源。

3、 设计测试用例，主要是与业务相关的测试用例。

4、实施功能模块测试，搭建运行或开发环境，采用功能模块测试表的方式，开发人员在功能模块测试表中更新进度状态，测试人员在该表中描述测试进度。形成测试错误列表，该表对每个错误都有相应的测试记录与之链接，在测试记录中，详细描述错误的情况。在测试记录中还要包括修正信息和验证信息。

5、错误关闭后，测试人员维护测试记录表和更新测试用例库和问题库，作为经验积累。

6、项目在结项时，测试人员进行项目完工验收测试，填写项目测试报告。该测试报告可作为用户验收的输入工件。

三、 功能测试方法与内容

1、数据输入测试：向系统输入数据或输入数据库操作命令时，一般是测试系统对数据库中数据操作的过程。

数据类型测试：由于不同的数据库系统对数据类型要求的不同，在定义数据库表时，也规定了数据字段的数据类型。测试步骤和方法：在系统的数据维护功能界面上，录入或修改数据时，特意输入非系统设计的数据类型，检查系统是否可以接受，若不能接受则检查是否满足了系统在这方面的设计要求，如即刻清除非法内容、输入焦点不能到下一输入位置、出现系统自定义的提示信息、不允许出现开发工具的报错信息等。若系统可以接受并保存，则要看数据库表的字段类型设计是否与用户或习惯上不一致，并且要注意其他模块在调取该数据时，是否有特定要求。

边界值测试：根据数据取值范围的要求，输入符合取值范围的数据、取值范围的上、下限和超过取值范围的数据。注意，除要测试数据库系统本身数据类型取值范围外，还要根据软件系统设计中的一些特定要求，设计测试用例来测试。

数据合法性测试：测试人员除了要测试输入数据是否满足所使用数据库系统本身的数据类型和取值范围的要求外，还应该根据经验和软件系统和需求的特定要求检查输入数据的合法性。比如：日期合法性（出生年月、参保日期、发生时间、根据习惯和业务逻辑顺序对日期合理性的要求等）。工资、比例、率等，都要注意输入的合理、合法性。

单引号和双引号：不要忽略输入单引号和双引号可能引起的错误和数据问题。在功能录入界面上，在某字段的输入框输入了包括单引号和双引号的数据，以后在通过 Select 语句查询时可能会出问题。特别在基于 WEB 方式的系统，输入了单引号，在查询数据记录时，肯定会出现页面链接错误（页面无法链接或找不到或链接对象错误）。

空值测试：在测试数据录入或修改的功能界面时，若不输入任何东西，系统又没有设计成 NOT NULL，则这时，要非常注意其影响。因为数据可以正常保存，但数据表该字段是空值，那么所有与该字段有关的操作，如：查询(AND)、计算（累加、连乘）等，则可能出现数据问题（计算结果为 0，无记录返回）。对于测试人员首先要检查系统到底是作为空值，还是作为空串或空字符处理。另外对于允许不输入任何值的字段，在测试过程中，要检查是否在界面显示或打印报表时，这些字段作为了关键要素或标题等情况。

空格：在数据维护的功能界面上，输入数据时，要注意是否在输入位置有空格，首先看系统设计时，是怎么考虑的，若系统允许输入空格，则检查条件查询或作为调用参数时的数据返回情况；另外检查程序是否使用了去掉空格的函数。

数据校验的不一致：测试时，对于一些编号、编码、代码等主键或作为查询或调用条件的字段，要注意系统对他们的输入合法性检查与查询或调用条件的要求是否是一致的。特别是对于数据结构设计中没有特定约束，而由程序进行校验控制的情况。

分析：数据输入测试的主要目的是保证输入到系统中数据的合法、合理性。我觉得，数据输入过程的检查是非常重要的，若在编程过程中，不注重数据的校验功能，虽然看起来加快了开发进度，但给以后会带来一些不可预计的编程或维护工作量。

2、目录路径测试：测试系统中规定的路径要求，更改路径，检查系统的是否可以正确运行及系统的排错功能。测试时，根据系统设计说明书（详细设计）或通过程序源代码的熟悉，找出系统运行过程中指定的路径或在运行过程中，需要使用选择路径的地方。特意更改路径（选择正确的路径、选择另外的路径、输入不存在的路径）。检查系统是否具有路径上的容错性和灵活性。比如，原则上在程序中，最好不要写绝对路径，另外可以提供配置路径的对话框，若输入了非法路径，系统有无提示等。

3、 数据操作测试：包括数据操作测试和用户界面操作的测试。

修改、新增数据：对于新增和修改数据，要注重以下几个方面的测试。界面上，新增数据成功后，数据列表是否立即刷新，输入有错误时，是否清空错误的数据，输入焦点是否得以控制。在提示信息上，是否有保存成功的提示，输入有错误时，提示的错误信息是否准确，可读。数据方面，要通过 SQL 检查数据提交是否正确。删除数据：测试删除记录时，系统是否有确认提示，能否批量删除，根据系统详细设计，检查删除主表记录时，在业务上，其他相关表是否相应更改。

事物的提交与回滚：熟悉 C/S 模式开发或数据库应用系统开发的人都知道，数据库事物的概念。对于一个比较复杂的业务逻辑或业务上有数据一致和完整性要求时，尽量使用事物对数据进行提交，这样一旦由于意外原因引起系统或硬件故障时，可以回滚。根据系统的设计要求在测试时，可人为模拟意外故障，来测试系统的数据完整性和容错能力。

4、工具条和快捷键测试：在功能界面测试时，对系统菜单中定义的快捷键和菜单工具条中的工具按钮要测试。主要是有效性和一致性测试。有效性：检查是否

有效，界面有无反应。一致性：定义或提示的信息是否与实际完成的功能一致。

5、 操作顺序测试

按钮顺序测试：在功能界面上，不按照设计上或习惯上的操作顺序点击功能按钮，看系统有什么反应；多次、反复点击某一按钮，看系统有什么反应。主要是测试系统的控制、校验和容错能力。

业务逻辑顺序：不按照系统的正常业务逻辑、流程操作，来测试系统是否控制了业务流程的顺序。

6、按钮有效性控制测试：主要是测试当不具备条件或无实际意义的情况下，按钮的“Enabled”属性。比如：某一业务未处理，下一环节的功能按钮则应变灰（不可用）。逐条显示数据记录，当游标已经指到了最后一条时，“下一条”和“末记录”按钮则应变灰等。

7、同时刻操作测试：对于删除、修改、增加数据和一些业务功能，进行多客户端同时刻操作测试，看系统有什么反应。

8、附件压力测试：对于有发送、上传、下载、邮件等功能的系统，选取大的文件，进行测试，来检查系统的界面效果和稳定性，看是否会死机或长时间无任何反应等。

9、 数据输出测试：

数据处理输出测试：主要测试对数据的排序、条件查询是否按照输入的条件或要求输出了正确的数据。

打印输出：测试打印功能是否能够正常打印出报表，打印设置后，是否能按照设置的要求打印。

10、WEB 测试：基于 WEB 方式的应用，对于一些提交表单的页面，通过多次点击“back”键，来测试系统的处理情况。对于有保存数据功能的页面，多次点击

“保存”，来测试系统的处理情况。