

测试用例

一、 定义

测试用例（ Test Case ）是指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。

二、 测试用例的分类

根据测试过程中具体涉及到问题类型及测试需求，可将测试用例分为如下：

功能性测试用例

界面测试用例：适用于所有测试阶段中的界面测试

数据处理测试用例：适用于所有测试阶段中的数据处理测试

操作流程测试用例：适用于所有流程性的测试

安装测试用例：适用于所有安装测试

三、 测试用例管理

编写用例：测试工程师根据需求规约、概要设计、详细设计等文档编写测试用例。

用例评审：原则上用例象程序一样，要经过多次的修改才可以通过，实际 工作中通常进行一次。

用例修改：评审结束后，您需要根据评审意见进行修改，修改后通常不再进行评审。

使用用例：执行测试用例，并记录到测试用例执行报告中。

用例升级 / 维护：随着软件产品不断修改、升级，对应的用例也需要升级维护。针对同一个项目，可以根据需求的变更不断进行维护； 如果是产品，用例的维护更加重要， 要达到用例和产品的版本一一对应。

四、 测试用例的编制及使用

1 、设计测试用例

每个具体测试用例都将包括下列详细信息：编制人、审定人、编制日期、版本、用例类型、设计说明书编号、用例编号、用例名称、输入说明、期望结果（含判断标准）、环境要求、备注等。

测试用例

编制人

审定人

编制日期

版本

测试用例类型

设计说明书编号

测试用例编号

测试用例名称

输入说明（列出选用的输入项，覆盖正常、异常情况）：

期望结果（逐条与输入项对应，列出预期输出）：

环境要求（测试要求的软、硬件、网络要求）：

备注：

测试用例名称 ”可以是不涉及到具体模块的功能描述，如 “日期格式 ”， “非空检验 ”等。

输入说明 ”是功能模块接受的数据或各种操作描述，如 “输入非法的日期格式 ”等。

期望结果 ”是模块接受输入后应有的正常输出描述，如 “提示用户修改 ”等，期望结果应与输入说明一一对应。

测试用例用于指导执行操作，但某些意外操作也可导致程序错误，这些操作称为非预

期性操作，可以先有执行报告，再后补用例。

测试用例的设计应考虑通用性和简洁明了。

2 、 执行测试用例

此报告用于记录执行上一步设计的测试用例的过程及结果。

步骤 ”应填入详细的操作，如 “点增加 -> 输入日期 -> 保存 ”。 ‘输入数据 ’填入具体数据，如 “ 2002/12/12 。”

期望输出 ’即测试用例中的 ‘期望结果 ’，但描述应更具体，如 ‘弹出提示对话框，提示用户日期格式错误 ’。

实际输出 ’是操作的真实结果，必须详细、清晰，便于开发人员理解。

如 ‘实际输出 ’与 ‘期望输出 ’不符，则结果为 F (False), 若相符则结果为 T(True) 。

3 、 用例模板

软件功能性测试用例模板

一、功能检查

- 1 、 功能是否齐全，例如：增加、删除、修改
- 2 、 功能是否多余
- 3 、 功能是否可以合并
- 4 、 功能是否可以再细分
- 5 、 软件流程与实际业务流程是否一致
- 6 、 软件流程能否顺利完成
- 7 、 各个操作之间的逻辑关系是否清晰
- 8 、 各个流程数据传递是否正确
- 9 、 模块功能是否与需求分析及概要设计相符

二、面向用户的考虑

- 1 、 操作方便性，如：按键次数是否最少
- 2 、 易用性，面对用户的操作是否简单易学
- 3 、 智能化考虑
- 4 、 提示信息是否模糊不清或有误导作用
- 5 、 要求用户进行的操作是否多余，能否由系统替代
- 6 、 能否记忆操作的初始环境，无需用户每次都进行初始化设置
- 7 、 是否不经确认就对系统或数据进行重大修改
- 8 、 能否及时反映或显示用户操作结果
- 9 、 操作是否符合用户习惯，比如：热键
- 10 、 各种选项的可用及禁用是否及时合理
- 11 、 某些相似的操作能否做成通用模块

软件数据处理测试用例模板

一、输入数据

- 1 、 边界值
- 2 、 大于边界值
- 3 、 小于边界值
- 4 、 最大个数
- 5 、 最大个数加 1
- 6 、 最小个数
- 7 、 最小个数减 1
- 8 、 空值、空表

- 9 、 极限值
- 10 、 0 值
- 11 、 负数
- 12 、 非法字符
- 13 、 日期、时间控制
- 14 、 跨年度数据
- 15 、 数据格式

二、数据处理

- 1 、 处理速度
- 2 、 处理能力
- 3 、 数据处理正确率
- 4 、 计算方式

三、输出结果

- 1 、 正确率
- 2 、 输出格式
- 3 、 预期结果
- 4 、 实际结果

软件流程测试用例模板

- 1 、 反流程操作
- 2 、 反逻辑操作
- 3 、 重复操作
- 4 、 反业务流程操作

软件安装测试用例模板

项目名称：

项目版本号：

软件的安装 / 卸载流程能否正确顺利地进行

软件的安装 / 卸载是否简单、易学、易用

安装过程中的文字及提示有否错字、别字，提示信息是否完备

安装过程中的各选项是否有效，合理

安装完成后生成的快捷图标及菜单是否正确，路径是否有效

安装文件夹的个数及所包含的内容是否正确无误码

INI 文件及配置文件是否正确

生成的系统备份文件是否正确

动态库及主程序的个数、内容是否正确

运行程序，软件各项功能是否能正常运行，如果有修改，安装后的内容是否最新

系统固定数据、数据库是否正确

附注：用例编码规则

功能 — 以字母 U 开头后跟数字编码

界面 — 以字母 I 开头后跟数字编码

数据 — 以字母 D 开头后跟数字编码

流程 — 以字母 F 开头后跟数字编码

安装 — 以字母 S 开头后跟数字编码

测试用例编写规范

一、测试用例编写准备

从配置管理员处申请软件配置： 《需求规格说明书》和 《设计说明书》 ；根据需求规格说明书和设计说明书， 详细理解用户的真正需求， 并且对软件所实现的功能已经准确理解， 然后着手制订测试用例。

二、测试用例制定的原则

测试用例要包括欲测试的功能、 应输入的数据和预期的输出结果。 测试数据应该选用少量、高效的测试数据进行尽可能完备的测试； 基本目标是： 设计一组发现某个错误或某类错误的测试数据，测试用例应覆盖方面：

1、正确性测试：输入用户实际数据以验证系统是满足需求规格说明书的要求； 测试用例中的测试点应首先保证要至少覆盖需求规格说明书中的各项功能，并且正常。

2、容错性（健壮性）测试：程序能够接收正确数据输入并且产生正确（预期）的输出，输入非法数据（非法类型、不符合要求的数据、溢出数据等） ，程序应能给出提示 并进行相应处理。把自己想象成一名对产品操作一点也不懂的客户，在进行任意操作。

3、完整（安全）性测试：对未经授权的人使用软件系统或数据的企图，系统能够控制的程度，程序的数据处理能够保持外部信息（数据库或文件）的完整。

4、接口间测试：测试各个模块相互间的协调和通信情况，数据输入输出的一致性和正确性。

5、数据库测试：依据数据库设计规范对软件系统的数据库结构、数据表及其之间的数据调用关系进行测试。

6、边界值分析法：确定边界情况（刚好等于、稍小于和稍大于和刚刚大于等价类边界值），针对我们的系统在测试过程中主要输入一些合法数据 /非法数据， 主要在边界值附近选取。

7、压力测试：输入 10 条记录运行各个功能， 输入 30 条记录运行，输入 50 条记录运行。。进行测试。

8、等价划分：将所有可能的输入数据（有效的和无效的）划分成若干个等价类。

9、错误推测：主要是根据测试经验和直觉，参照以往的软件系统出现错误之处。

10、效率：完成预定的功能，系统的运行时间（主要是针对数据库而言） 。

11、可理解（操作）性：理解和使用该系统的难易程度（界面友好性） 。

12、可移植性：在不同操作系统及硬件配置情况下的运行性。

13、回归测试：按照测试用例将所有的测试点测试完毕， 测试中发现的问题开发人员 已经解决，进行下一轮的测试。

14、比较测试：将已经发版的类似产品或原有的老产品与测试的产品同时运行比较， 或与已往的测试结果比较 。

说明：针对不同的测试类型和测试阶段，测试用例编写的侧重点有所不同。

1、其中第 1、2、6、8、9、13 项为模块（组件、控件）测试、组合（集成）测试、系统测试都涉及并重点测试的方面。

2、单元（模块）测试（组件、控件）测试：重点测试第 5 项。

3、组合（集成）测试：重点进行接口间数据输入及逻辑的测试，即第 4 项。

4、系统测试：重点测试第 3、7、10、11、12、14 项。

5、其中压力测试和可移植性测试如果是公司的系列产品，可以选用其中有代表性的产品进行一次代表性测试即可。

6、GMPS 基础测试用例设计完成后，其他的测试项目只编写设计与之不同部分的测试用例。

7、对于每个测试项目测试的测试用例不是一成不变的，随着测试经验的积累或在测试其他项目发现有测试不充分的测试点时，可以不断的补充完善测试项目的测试用例。

三、测试用例的填写

一个软件系统或项目共用一套完整的测试用例，整个系统测试过程测试完毕，将实际测试结果填写到测试用例中，操作步骤应尽可能的详细，测试结论是指最终的测试结果（结论为：通过或不通过）。