

测试用例编写规范

1	目的	2
2	范围	2
3	术语解释	2
4	测试用例原则	2
4.1	系统性	2
4.2	连贯性	2
4.3	全面性	2
4.4	正确性	2
4.5	符合正常业务惯例	3
4.6	仿真性	3
4.7	可操作性	3
5	测试用例主要元素	3
6	测试用例编写规范	3
6.1	常规的测试用例:	3
6.2	初始化的测试用例:	4
6.3	边界的测试用例	4
6.4	空值的测试用例:	4
6.5	格式错误的测试用例:	4
6.6	溢出的测试用例:	4
6.7	关联的测试用例:	4
6.8	唯一值的测试用例:	4
6.9	权限不足的测试用例:	4
6.10	角色权限的测试用例:	4
7	测试用例编写细则	5
7.1	测试用例命名规则	5
7.2	测试用例编号规则	5
8	测试用例编写方法	5
8.1	测试用例编写准备	5
8.2	测试用例编写方法	5

1 目的

统一测试用例编写的规范，为测试设计人员提供测试用例编写的指导，提高编写的测试用例的可读性、可执行性、合理性。为测试执行人员更好执行测试，提高测试效率，最终提高公司整个产品的质量。

2 范围

适用于集成测试用例和系统测试用例的编写，现在编写用例的辅助工具为 TestDirector 8.0。

3 术语解释

集成测试：

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。

系统测试：

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。

4 测试用例原则

4.1 系统性

1. 对于系统业务流程要能够完整说明整个系统的业务需求、系统由几个子系统组成以及它们之间的关系；
2. 对于模块业务流程要能够说明清楚子系统内部功能、重要功能点以及它们之间的关系；

4.2 连贯性

1. 对于系统业务流程来说，各个子系统之间是如何连接在一起，如果需要接口，各个子系统之间是否有正确的接口；如果是依靠页面链接，页面链接是否正确；
2. 对于模块业务流程来说，同级模块以及上下级模块是如何构成一个子系统，其内部功能接口是否连贯；

4.3 全面性

1. 应尽可能覆盖程序的各种路径
2. 应尽可能覆盖系统的各个业务
3. 应考虑存在跨年、跨月的数据
4. 大量数据并发测试的准备

4.4 正确性

1. 输入界面后的数据应与测试文档所记录的数据一致
2. 预期结果应与测试数据发生的业务吻合

4.5 符合正常业务惯例

1. 测试数据应符合用户实际工作业务流程
2. 兼顾各种业务变化的可能
3. 要符合当前业务行业法律，法规。

4.6 仿真性

人名、地名、电话号码等应具有模拟功能，符合一般的命名惯例；不允许出现与知名人士、小说中人物名等雷同情况。

4.7 可操作性

测试用例中应写清测试的操作步骤，不同的操作步骤相对应的操作结果。

5 测试用例主要元素

标准规范中包含的主要元素如下：

- l 测试名称(**Test Name**)：测试用例编号和测试用例名称。
- l 创建日期(**Creation Date**)：测试用例创建时间，系统自动产生。
- l 设计人员(**Designer**)：测试用例设计人员
- l 状态(**Status**)：测试用例状态
- l 描述 (**Description**)：测试用例详细描述
- l 步骤名称 (**Step Name**)：测试步骤名称
- l 步骤描述 (**Step Description**)：测试步骤详细描述。
- l 预期结果 (**Expected Result**)：测试预期结果。

6 测试用例编写规范

1. 对于每个功能，从类型 1 至类型 N 依次撰写相应用例
2. 对于不满足要求的非常规类型，可以不写相应的用例
3. 对于边界、空值、格式错误、溢出这几个类型，一个功能如有多个数据项测试类型相同，则可以放在一个用例里
4. 测试用例均为最小的用例覆盖要求；对于没有提及的用例类型，视业务需求情况，撰写相应用例
5. 在测试过程中，输入数据可在测试用例规定的范围内做一定变化

6.1 常规的测试用例：

1. 对于一个功能一个模块（页面），每个数据项输入或选中典型的取值，生成一个用例
2. 对于一个功能多个模块（页面），多个模块（页面）一起生成一个用例
3. 对于多个功能一个模块（页面），每个功能生成一个用例
4. 每个功能操作需覆盖，如删除对话框点击确定、取消分别生成 2 个用例步骤。
5. 输入框测试，在允许范围内尽可能覆盖多的字符类别，如中文、英文、数字等
6. 对于每个功能点，必须通过一组（一个或多个）用例满足其业务覆盖：对于某条记录的每个状态，对于能进行的每个操作，都生成一个用例（即对业务功能流程中的每个角色，每个功能操作，生成一个用例）

6.2 初始化的测试用例：

进入功能模块（页面）后，某些控件会初始化填入数据，生成一个用例确保所有的初始数据正确

6.3 边界的测试用例

1. 每个数据项，生成一个边界用例（含最大、最小两个边界值）
2. 字符串数据以字符串长度为计量单位
3. 布尔值数据的所有取值都需测试
4. 多个复选框一组时，需测同时都被选中及都不被选中
5. 下拉菜单、列表框、单选按钮组为最大、最小的 2 个取值

6.4 空值的测试用例：

对于每个必填数据项，都生成一个用例（不提供空值的除外，比如无空值的下拉框、有缺省值的单选按钮组），则预期结果提示该数据项为空

6.5 格式错误的测试用例：

对于输入框数据项，都生成一个用例，预期结果提示该数据项格式错误

- l 日期输入框
- l 数字输入框
- l 字符串输入框：Email、邮编、用户名等带格式要求的

6.6 溢出的测试用例：

对于输入框数据项，都生成一个取值范围外的测试用例，预期结果提示该数据项超出范围日期输入框

- l 范围的日期输入框，需添加上边界日期小于下边界日期的用例
- l 数字输入框（如‘金额’一般为正整数，填入一个负数）
- l 字符串输入框：超出规定长度的字符串

6.7 关联的测试用例：

对于相互关联的两个或多个数据项，生成一个用例，确保当一个数据项改变时，其他数据项的变化正确

6.8 唯一值的测试用例：

某些业务的数据字段要求是唯一的，生成一或两个用例（新建、编辑），使得输入数据与原有数据在该字段重复，预期结果为页面返回该数据已存在的提示

6.9 权限不足的测试用例：

对于功能模块，生成一个用例，以没有权限的用户身份访问，预期结果为提示权限不足

6.10 角色权限的测试用例：

业务功能流程涉及一到多个角色，对于每个角色，都生成一个用例，预期结果为用户以这个角色登陆时，他仅能执行权限允许的操作

7 测试用例编写细则

7.1 测试用例命名规则

由于项目的实际需求和测试的工作需要，分以下几个等级来规范测试用例的命名

1. 一级目录使用各项目的顶级菜单名称来命名，如维护、业务、查询三大类；
2. 二级目录使用顶级菜单下的二级菜单名称类命名，用户可根据名字判别该用例是测试哪个模块的；
3. 各用例根据各用例的功能来命名，尽量做到简洁明了。同一个目录下的用例名字字数最好相同；

7.2 测试用例编号规则

每个测试用例都有自己唯一的编号。根据工作的实际需要，我们规定在每个用例名称前面必须写上用例编号，用例编号的定义分以下几大类：

- 1、根据需求编写测试用例：

需求编号+用例一级目录号+用例二级目录号+用例号
R001 + 01 + 01 01

- 2、根据功能编写测试用例：

用例一级目录号+用例二级目录号+用例号
F001 + 001 + 001

在编写测试用例时，我们会根据系统模块的具体情况从不同的角度去考虑测试用例的编写，有些是通过操作步骤来编写，有些则是根据功能条件来编写，更有可能是根据测试目的来编写，为了区分这些用例，我们规定在每种用例前写上对应的编码。具体见下表：

需求	功能	业务	性能
R(Requirement)	F(Function)	B(Business)	P(Performance)

8 测试用例编写方法

8.1 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。

8.2 测试用例编写方法

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，测试中发现问题开发人员
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。