
XXX 项目

软件系统测试计划

拟制：_____

审核：_____

批准：_____

XX股份有限公司

文件更改记录

编号:

序号:

[illegible]

目录

1	概述.....	4
1.1	目的.....	4
1.2	内容和范围.....	4
1.3	术语定义.....	4
2	测试需求.....	4
3	测试约束.....	4
4	测试类型和策略.....	5
4.1	数据和数据库完整性测试.....	5
4.2	功能测试.....	6
4.3	业务周期测试.....	7
4.4	用户界面测试.....	8
4.5	性能测试.....	9
4.5.1	性能评价.....	9
4.5.2	负载测试.....	10
4.5.3	强度测试.....	11
4.5.4	容量测试.....	12
4.6	安全性和访问控制测试.....	12
4.7	故障转移和恢复测试.....	13
4.8	配置测试.....	15
4.9	安装测试.....	16
5	测试时间安排.....	16
6	测试资源.....	17
6.1	测试环境.....	17
6.2	测试工具.....	17
6.3	人员及职责.....	18
7	测试输出.....	18
	附件：测试用例和记录格式.....	20

1 概述

1.1 目的

[提示：本文档适用于项目中各类测试之前的计划，是在项目计划阶段建立的测试计划（包含在《项目计划》中）的细化。

此处定义本次测试的目的。

如果是集成测试，可能如下：

测试产品各个模块之间是否能协同工作，交互顺畅，并符合需求的定义

如果是系统测试，可能如下：

测试产品的功能是否满足所有的需求，并可以实现需求中定义的所有业务流程

]

1.2 内容和范围

[提示：简要描述本次测试的内容和界限，如测试包含哪些功能模块、接口等，包含哪些测试类型（功能测试、性能测试、强度测试……），明确规定该次测试的包含范围和不包含范围。还可以简单陈述本次测试的过程。]

1.3 术语定义

[提示：定义所使用的术语。对于易混淆的客户常用语要有明确规定定义。]

2 测试需求

[提示：

在此处填写一个本次测试所涉及的测试需求的详细列表。可以引用《产品需求说明书》、《概要设计说明书》、《数据库设计说明书》、《详细设计说明书》等相关文档中的内容。

测试需求的主要来源为：系统主要功能描述，需求规格说明书，补充需求，系统关注点。

下面列出了那些已被确定为测试对象的条目（用例、功能性需求和非功能性需求），说明了测试的对象。测试的类型可参考“4 测试类型和策略”]

测试需求	描述	来源	类型

3 测试约束

[提示：描述本次测试的约束条件，例如测试的软件需要有一定的网络环境，但是本次测试

只测试软件，默认网络环境为正常。]

4 测试参考文档

下表列出了制定测试计划时所使用的文档，并标明了各文档的可用性：

[提示：以下表格中的文档项仅为示例，可根据项目情况适当地删除或添加文档项。]

文档	已创建或可用	已被接收或已经过复审	作者或来源	备注
可行性分析报告	是□ 否□	是□ 否□		
软件需求定义	是□ 否□	是□ 否□		
软件系统分析	是□ 否□	是□ 否□		
软件概要设计	是□ 否□	是□ 否□		
软件详细设计	是□ 否□	是□ 否□		
软件测试需求	是□ 否□	是□ 否□		
硬件可行性分析报告	是□ 否□	是□ 否□		
硬件需求定义	是□ 否□	是□ 否□		
硬件概要设计	是□ 否□	是□ 否□		
硬件原理图设计	是□ 否□	是□ 否□		
硬件结构设计	是□ 否□	是□ 否□		
硬件测试需求	是□ 否□	是□ 否□		
模块开发手册	是□ 否□	是□ 否□		
用户操作手册	是□ 否□	是□ 否□		
安装指南	是□ 否□	是□ 否□		

5 测试类型和策略

[提示：根据测试的具体内容对以下各项测试类型进行裁剪，只包括本次测试设计的测试类型即可，以下的斜体内容均为解释或示例]

5.1 数据和数据库完整性测试

数据库和数据库进程应作为项目的子系统来进行测试。

在测试这些子系统时，不应将测试对象的用户界面用作数据的接口。对于数据库管理系统 (DBMS)，还需要进行深入的研究，以确保数据库访问方法和进程正常运行，数据不会遭到损坏。

测试目标:	确保数据库访问方法和进程正常运行，数据不会遭到损坏。
方法:	<ul style="list-style-type: none"> 调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据或对数据的请求。 检查数据库，确保数据已按预期的方式填充，并且所有数据库事件都按正常方式出现；或者检查所返回的数据，确保为正当的理由检索到了正确的数据
完成标准:	所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。
需考虑的特殊事项:	<ul style="list-style-type: none"> 测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接输入或修改数据。 进程应该以手工方式调用。 应使用小型或最小的数据库（其中的记录数很有限）来使所有无法接受的事件具有更大的可见性。

5.2 接口测试

接口测试主要目的是确保所有硬件接口和软件接口调用的正确性。主要测试方法是通过记录各接口的输入输出数据，并核对其是否为预期值。

测试目标:	确保所有软件、硬件接口调用的正确性
方法:	记录输入输出数据
完成标准:	
需考虑的特殊事项:	接口的限制条件

5.3 集成测试

集成测试的主要目的是检测系统是否达到需求，对业务流程及数据流的处理是否符合标准，检测系统对业务流处理是否存在逻辑不严谨及错误，检测需求是否存在不合理的标准及要求。此类型测试基于完成的单元测试。

测试目标	检测需求中业务流程，数据流的正确性
方法:	测试需求中明确的业务流程，或组合不同功能模块而形成一个大的功能。利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以

	<p>下内容:</p> <p>在使用有效数据时得到预期的结果。</p> <p>在使用无效数据时显示相应的错误消息或警告消息。</p> <p>各业务规则都得到了正确的应用。]</p>
完成标准:	<p>[所计划的测试已全部执行。</p> <p>所发现的缺陷已全部解决。]</p>
需考虑的特殊事项:	<p>[确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）]</p>

5.4 功能测试

功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面 (GUI) 与应用程序交互并分析输出结果来验证应用程序及其内部进程。功能测试的一般方法是利用有效的和无效的数据来执行各个用例、用例流或功能，以核实是否达到预期结果。

测试目标:	<p>确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。</p>
方法:	<p>基于黑盒方法，即通过图形用户界面 (GUI) 与应用程序交互并分析输出结果来验证应用程序及其内部进程。利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容:</p> <ul style="list-style-type: none"> • 在使用有效数据时得到预期的结果。 • 在使用无效数据时显示相应的错误消息或警告消息。 • 各业务规则都得到了正确的应用。
完成标准:	<ul style="list-style-type: none"> • 所计划的测试已全部执行。 • 所发现的缺陷已全部解决。
需考虑的特殊事项:	<p>确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）</p>

5.5 业务周期测试

业务周期测试应模拟在一段时间内对项目执行的活动。应先确定一段时间（例如一年），然后执行将在该时段内发生的事务和活动。这种测试包括所有的每日、每周和每月的周期，以及所有与日期相关的事件（如备忘录）。

测试目标	确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。
方法：	<p>通过执行以下活动，测试将模拟若干个业务周期：</p> <ul style="list-style-type: none"> • 将修改或增强对测试对象进行的功能测试，以增加每项功能的执行次数，从而在指定的时段内模拟若干个不同的用户。 • 将使用有效的和无效的日期或时段来执行所有与时间或日期相关的功能。 • 将在适当的时候执行或启动所有周期性出现的功能。 • 在测试中还将使用有效的和无效的数据，以核实以下内容： • 在使用有效数据时得到预期的结果。 • 在使用无效数据时显示相应的错误消息或警告消息。 • 各业务规则都得到了正确的应用。
完成标准：	<ul style="list-style-type: none"> • 所计划的测试已全部执行。 • 所发现的缺陷已全部解决。
需考虑的特殊事项：	<ul style="list-style-type: none"> • 系统日期和事件可能需要特殊的支持活动 • 需要通过业务模型来确定相应的测试需求和测试过程。

5.6 用户界面测试

用户界面（UI）测试用于核实用户与软件之间的交互。UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。另外，UI 测试还可确保 UI 中的对象按照预期的方式运行，并符合公司或行业的标准。

测试目标：	<p>核实以下内容：</p> <ul style="list-style-type: none"> • 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab 键、鼠标移动和快捷键）的使用 • 窗口的对象和特征（例如：菜单、大小、位置、状态和 中心）都符合标准。
方法：	为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。
完成标准：	证实各个窗口都与基准版本保持一致，或符合可接受标准

需考虑的特殊事项:	并不是所有定制或第三方对象的特征都可访问。
-----------	-----------------------

5.7 性能测试

5.7.1 性能评价

性能评价是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。性能评价的目标是核实性能需求是否都已满足。实施和执行性能评价的目的是将测试对象的性能行为当作条件(例如工作量或硬件配置)的一种函数来进行评价和微调。

测试目标:	<p>核实所指定的事务或业务功能在以下情况下的性能行为:</p> <ul style="list-style-type: none"> • 正常的预期工作量 • 预期的最繁重工作量
方法:	<ul style="list-style-type: none"> • 使用为功能或业务周期测试制定的测试过程。 • 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代次数。 • 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多台客户机（虚拟的或实际的客户机，请参见下面的“需考虑的特殊事项”）上重复。
完成标准:	<ul style="list-style-type: none"> • 单个事务或单个用户：在每个事务所预期或要求的时间范围内成功地完成测试脚本，没有发生任何故障。 • 多个事务或多个用户：在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。

需考虑的特殊事项:	<p>综合的性能测试还包括在服务器上添加后台工作量。</p> <p>可采用多种方法来执行此操作，其中包括：</p> <ul style="list-style-type: none"> • 直接将“事务强行分配到”服务器上，这通常以“结构化查询语言”(SQL) 调用的形式来实现。 • 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机。此负载可通过“远程终端仿真”(Remote Terminal Emulation) 工具来实现。此技术还可用于在网络中加载“流量”。 • 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载。 <p>性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。</p> <p>性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。</p>
-----------	--

5.7.2 负载测试

负载测试是一种性能测试。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征，例如，响应时间、事务处理速率和其他与时间相关的方面。

测试目标:	核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。
方法:	<ul style="list-style-type: none"> • 使用为功能或业务周期测试制定的测试。 • 通过修改数据文件来增加事务数量，或通过修改测试来增加每项事务发生的次数。
完成标准:	多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有任何故障。
需考虑的特殊事项:	<ul style="list-style-type: none"> • 负载测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。 • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。

5.7.3 强度测试

强度测试是一种性能测试，实施和执行此类测试的目的是找出因资源不足或资源争用而导致的错误。如果内存或磁盘空间不足，测试对象就可能会表现出一些在正常条件下并不明显的缺陷。而其他缺陷则可能由于争用共享资源（如数据库锁或网络带宽）而造成的。强度测试还可用于确定测试对象能够处理的最大工作量。

测试目标:	<p>核实测试对象能够在以下强度条件下正常运行，不会出现任何错误：</p> <ul style="list-style-type: none">• 服务器上几乎没有或根本没有可用的内存（RAM 和 DASD）• 连接或模拟了最大实际（或实际可承受）数量的客户机• 多个用户对相同的数据/账户执行相同的事务• 最繁重的事务量或最差的事务组合（请参见上面的“性能测试”）。 <p>注：强度测试的目标还可表述为确定和记录那些使系统无法继续正常运行的情况或条件。</p>
方法:	<ul style="list-style-type: none">• 使用为性能评价或负载测试制定的测试。• 要对有限的资源进行测试，就应该在一台计算机上运行测试，而且应该减少或限制服务器上的 RAM 和 DASD。• 对于其他强度测试，应该使用多台客户机来运行相同的测试或互补的测试，以产生最繁重的事务量或最差的事务组合。
完成标准:	<p>所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。</p>
需考虑的特殊事项:	<ul style="list-style-type: none">• 如果要增加网络工作强度，可能会需要使用网络工具来给网络加载消息或信息包。• 应该暂时减少用于系统的 DASD，以限制数据库可用空间的增长。• 使多个客户机对相同的记录或数据账户同时进行的访问达到同步。

5.7.4 容量测试

容量测试使测试对象处理大量的数据，以确定是否达到了将使软件发生故障的极限。容量测试还将确定测试对象在给定时间内是否能够持续处理的最大负载或工作量。例如，如果测试对象正在为生成一份报表而处理一组数据库记录，那么容量测试就会使用一个大型的测试数据库，检验该软件是否正常运行并生成了正确的报表。

测试目标:	<p>[核实测试对象在以下大容量条件下能否正常运行:</p> <ul style="list-style-type: none">• 连接（或模拟了）最大（实际或实际可承受）数量的客户机，所有客户机在长时间内执行相同的、且情况（性能）最差的业务功能。• 已达到最大的数据库大小（实际的或按比例缩放的），而且同时执行了多个查询或报表事务。
方法:	<ul style="list-style-type: none">• 使用为性能评价或负载测试制定的测试。• 应该使用多台客户机来运行相同的测试或互补的测试，以便在长时间内产生最繁重的事务量或最差的事务组合（请参见上面的“强度测试”）。• 创建最大的数据库大小（实际的、按比例缩放的、或输入了代表性数据的数据库），并使用多台客户机在长时间内同时运行查询和报表事务。
完成标准:	<ul style="list-style-type: none">• 所计划的测试已全部执行，而且在达到或超出指定的系统限制时没有出现任何软件故障。
需考虑的特殊事项:	对于上述的大容量条件，哪个时段是可以接受的时间？

5.8 安全性和访问控制测试

安全性和访问控制测试侧重于安全性的两个关键方面:

- 应用程序级别的安全性，包括对数据或业务功能的访问
- 系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性可确保：在预期的安全性情况下，用户只能访问特定的功能或用例，或者只能访问有限的数据。

系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。

测试目标:	<ul style="list-style-type: none"> • 应用程序级别的安全性: 核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。 • 系统级别的安全性: 核实只有具备系统和应用程序访问权限的主角才能访问系统和应用程序。
方法:	<ul style="list-style-type: none"> • 应用程序级别的安全性: 确定并列出各用户类型及其被授权使用的功能或数据。 • 为各用户类型创建测试, 并通过创建各用户类型所特有的事务来核实其权限。 • 修改用户类型并为相同的用户重新运行测试。对于每种用户类型, 确保正确地提供或拒绝了这些附加的功能或数据。 • 系统级别的访问: 请参见下面的“需考虑的特殊事项”
完成标准:	各种已知的主角类型都可访问相应的功能或数据, 而且所有事务都按照预期的方式运行, 并在先前的应用程序功能测试中运行了所有的事务。
需考虑的特殊事项:	<p>必须与相应的网络或系统管理员一起对系统访问权进行检查和讨论。</p> <p>由于此测试可能是网络管理或系统管理的职能, 可能不需要执行此测试。</p>

5.9故障转移和恢复测试

故障转移和恢复测试可确保测试对象能成功完成故障转移, 并从硬件、软件或网络等方面的各种故障中进行恢复, 这些故障导致数据意外丢失或破坏了数据的完整性。

故障转移测试可确保: 对于必须始终保持运行状态的系统来说, 如果发生了故障, 那么备选或备份的系统就适当地将发生故障的系统“接管”过来, 而且不会丢失任何数据或事务。

恢复测试是一种相反的测试流程。其中, 将应用程序或系统置于极端的条件下 (或者是模仿的极端条件下), 以产生故障, 例如设备输入/输出 (I/O) 故障或无效的数据库指针和关键字。启用恢复流程后, 将监测和检查应用程序和系统, 以核实应用程序或系统是正确无误的, 或数据已得到了恢复。

<p>测试目标:</p>	<p>确保恢复进程（手工或自动）将数据库、应用程序和系统正确地恢复到了预期的已知状态。测试中将包括以下各种情况：</p> <ul style="list-style-type: none"> • 客户机断电 • 服务器断电 • 通过网络服务器产生的通信中断 • DASD 和/或 DASD 控制器被中断、断电或与 DASD 和/或 DASD 控制器的通信中断 • 周期未完成（数据过滤进程被中断，数据同步进程被中断）。 • 数据库指针或关键字无效 • 数据库中的数据元素无效或遭到破坏
<p>方法:</p>	<p>应该使用为功能和业务周期测试创建的测试来创建一系列的事务。一旦达到预期的测试起点，就应该分别执行或模拟以下操作：</p> <ul style="list-style-type: none"> • 客户机断电：关闭 PC 的电源。 • 服务器断电：模拟或启动服务器的断电过程。 • 通过网络服务器产生的中断：模拟或启动网络的通信中断（实际断开通信线路的连接或关闭网络服务器或路由器的电源）。 • DASD 和 DASD 控制器被中断、断电或与 DASD 和 DASD 控制器的通信中断：模拟与一个或多个 DASD 控制器或设备的通信，或实际取消这种通信。 <p>一旦实现了上述情况（或模拟情况），就应该执行其他事务。而且一旦达到第二个测试点状态，就应调用恢复过程。</p> <p>在测试不完整的周期时，所使用的方法与上述方法相同，只不过应异常终止或提前终止数据库进程本身。</p> <p>对以下情况的测试需要达到一个已知的数据库状态。当破坏若干个数据库字段、指针和关键字时，应该以手工方式在数据库中（通过数据库工具）直接进行。其他事务应该通过使用“应用程序功能测试”和“业务周期测试”中的测试来执行，并且应执行完整的周期。</p>

完成标准:	在所有上述情况中, 应用程序、数据库和系统应该在恢复过程完成时立即返回到一个已知的预期状态。此状态包括仅限于已知损坏的字段、指针或关键字范围内的数据损坏, 以及表明进程或事务因中断而未被完成的报表。
需考虑的特殊事项:	<ul style="list-style-type: none"> 恢复测试会给其他操作带来许多的麻烦。断开缆线连接的方法 (模拟断电或通信中断) 可能并不可取或不可行。所以, 可能会需要采用其他方法, 例如诊断性软件工具。 需要系统 (或计算机操作)、数据库和网络组中的资源。 这些测试应该在工作时间之外或在一台独立的计算机上运行。

5.10 配置测试

配置测试核实测试对象在不同的软件和硬件配置中的运行情况。在大多数生产环境中, 客户机工作站、网络连接和数据库服务器的具体硬件规格会有所不同。客户机工作站可能会安装不同的软件, 例如, 应用程序、驱动程序等。而且在任何时候, 都可能运行许多不同的软件组合, 从而占用不同的资源。

测试目标:	核实测试对象可在要求的硬件和软件配置中正常运行。
方法:	<ul style="list-style-type: none"> 使用功能测试脚本。 在测试过程中或在测试开始之前, 打开各种与非测试对象相关的软件 (例如 Microsoft 应用程序: Excel 和 Word), 然后将其关闭。 执行所选的事务, 以模拟主角与测试对象软件和非测试对象软件之间的交互。 重复上述步骤, 尽量减少客户机工作站上的常规可用内存。
完成标准:	对于测试对象软件和非测试对象软件的各种组合, 所有事务都成功完成, 没有出现任何故障。

需考虑的特殊事项:	<ul style="list-style-type: none"> • 需要、可以使用并可以通过桌面访问哪种非测试对象软件？ • 通常使用的是哪些应用程序？ • 应用程序正在运行什么数据？例如，在 Excel 中打开的大型电子表格，或是在 Word 中打开的 100 页文档。 • 作为此测试的一部分，应将整个系统、Netware、网络服务器、数据库等都记录下来。
-----------	---

5.11 安装测试

安装测试有两个目的。第一个目的是确保该软件能够在所有可能的配置下进行安装，例如，进行首次安装、升级、完整的或自定义的安装，以及在正常和异常情况下安装。异常情况包括磁盘空间不足、缺少目录创建权限等。第二个目的是核实软件在安装后可立即正常运行。

测试目标:	<p>核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中：</p> <ul style="list-style-type: none"> • 首次安装。以前从未安装过 <项目名称> 的新计算机 • 更新。以前安装过相同版本的 <项目名称> 的计算机 • 更新。以前安装过较早版本的 <项目名称> 的计算机
方法:	<ul style="list-style-type: none"> • 手工开发脚本或开发自动脚本，以验证目标计算机的状况——新 - <项目名称> 从未安装过；已安装 <项目名称> 相同或较早版本。 • 启动或执行安装。 • 使用预先确定的功能测试脚本子集来运行事务。
完成标准:	<项目名称> 事务成功执行，没有出现任何故障。
需考虑的特殊事项:	<p>应该选择 <项目名称> 的哪些事务才能准确地测试出 <项目名称> 应用程序已经成功安装，而且没有遗漏主要的软件构件？</p>

6 测试时间安排

[提示：测试可分多个时间进行，而且也可以分多轮进行。以下斜体滋补分为示例内容]

测试活动	开始时间	结束时间	工作量估计（人时）	输入	输出
活动 1				系统测试用例 1-50	系统测试记录
活动 2				系统测试用例 51-70	系统测试记录

7 测试资源

7.1 测试环境

测试的系统环境

软件环境（例如操作系统、应用软件、系统软件、开发软件、测试程序等）
1、
2、
3、
硬件环境（例如 计算机、服务器、网络、交换机等）
1、
2、
3、

7.2 测试工具

测试使用的工具：

工具名称	用途	生产厂商/自产	版本
	测试管理工具		
	缺陷跟踪工具		
	用于功能测试的 ASQ 工具		
	用于性能测试的 ASQ 工具		

	测试覆盖监测器或评价器		

7.3 人员及职责

[提示：例如项目经理、技术经理、开发人员、测试人员、技术支持人员、第三方代表等。]

测试所配备的人员及其角色职责：

人员	角色	具体职责或注释

8 测试输出

[提示：列出在测试阶段结束后，所有可提交的文档、工具和报告，及其创建人员、交付对象和交付时间。

可能的输出产品如：

- 1、测试模型报告：将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用工具来创建或引用。
- 2、测试记录：用来记录和报告测试结果和测试状态的方法和工具，如：《集成测试用例和记录》
- 3、缺陷报告：用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具，如：《测试报告》、《缺陷跟踪表》等。]

输出产品	创建人	交付对象	交付时间

附件：测试用例和记录格式

[提示：是每一个用例测试后记录结果的格式，可以是一个表格，或是一行记录，参见《集成测试用例和记录》或《系统测试用例和记录》]