测　试　用　例

联系凡是：babyxiaoji@pub.ss.pku.edu.cn

性能测试、压力测试、负载测试、强度测试、稳定性测试、健壮性测试、功能测试、系统测试、集成测试、接口测试… …,这么些眼花缭乱的测试类型名称，估计很少有有人能准确的区分和说出定义来，对应的测试用例如何编写和执行，就更不容易进行了。

如果问测试工程师测试用例如何编写，就好象是问程序员如何编写代码一样，每个人都会给出不同的方法，但是实用的测试用例却象优秀的程序一样困难。

本章针对上面的问题，主要讲解在企业实际工作中，如何有效划分测试种类和编写对应的测试用例，使测试工作更加合理、高效率的运行。

本章主要以测试用例的编写和管理为核心，讲述下面内容：

- 用例的分类
- 用例程度的把握
- 用例的执行
- 用例的评审
- 用例的升级、管理、维护

. . . . . .

事实上，完全可以把测试用例看成是测试工程师编写的程序：这个“ 程序 ”是为了辅助测试工作的进行而进行，目的是为了发现软件问题，同时“ 顺便 ”证明软件功能是否符合要求。

1.1 测试种类和阶段

1.1.1 测试种类

对于测试种类的说法多种多样，最多的能有 30 多种测试类型。而实际工作中很多测试是互相包含的。按照企业中实际工作需要，测试主要包含下面的类型：

功能测试：功能测试主要针对产品需求说明书的测试，主要是验证功能是否否合需求，包括原定功能的检验、是否有冗余功能、遗漏功能。这类测试应由测试员做，这并不意味着程序员在发布前不必检查他们的代码能否工作(自然他能用于测试的各个阶段)。

健壮性测试（容错能力/恢复能力测试）：侧重于程序容错能力的测试。本测试在单元测试阶段和系统测试阶段都要进行。如数据边界测试、非法数据测试、异常中断测试等等，主要是验证程序对各种异常情况是否进行正确处理。为了执行方便，建议健壮性的大部分测试用例尽量编写在功能测试用例中。

接口测试：程序员对各个模块进行系统联调的测试，包含程序内接口和程序外接口测试。这个测试，在单元测试阶段进行了一部分工作，而大部分都是在集成测试阶段完成的。由开发人员进行。

强度测试 强度测试检查程序对异常情况的抵抗能力。强度测试总是迫使系统在异常的资源配置下运行。例如，①当中断的正常频率为每秒一至两个时，运行每秒产生十个中断的测试用例；②定量地增长数据输入率，检查输入子功能的反映能力；③运行需要最大存储空间（或其他资源）的测试用例；④运行可能导致虚存操作系统崩溃或磁盘数据剧烈抖动的测试用例，等等。

压力测试：对系统不断施加压力的测试，是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。例如测试一个 Web 站点在大量的负荷下，何时系统的响应会退化或失败。

性能测试：在交替进行负荷和强迫测试时常用的术语。性能测试关注的是系统的整体。它和通常所说的强度、压力/负载测试测试有密切关系。所以压力和强度测试应该于性能测试一同进行。

举例说明：针对一个网站进行测试，模拟 10 到 50 个用户就是在进行常规性能测试，用户增加到1000乃至上万就变成了压力/负载测试。如果同时对系统进行大量的数据查询操作，就包含了强度测试。

压力测试注重的是外界不断施压，强度测试注重的是极限或者异常情况下系统的测试。

用户界面测试：对系统的界面进行测试，测试用户界面是否友好、是否方便易用、设计是否合理、位置是否正确等一系列界面问题

安全测试：主要是测试系统在没有授权的内部或者外部用户对系统进行攻击或者恶意破坏时如何进行处理，是否仍能保证数据的安全。测试人员可以学习一些黑客技术，来对系统进行攻击。

可靠性测试：这里是比较狭义的可靠性测试，它主要是对系统能否稳定运行进行一个统计，在实际工作中如果没有条件可以不必特意去做。重点做好与之紧密相关的功能测试、健壮性测试就可以了。

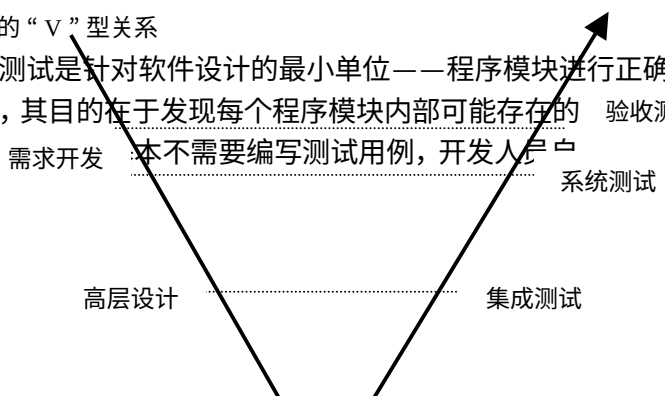
安装/反安装测试：安装测试主要检验软件是否可以正确安装，安装文件的各项设置是否有效，安装后能否影响原系统；反安装是逆过程，测试是否删除干净，是否给影响原系统等。

文档测试：主要测试开发过程中针对用户的文档，以需求、用户手册、安装手册等为主，检验文档是否和实际应用存在差别。文档测试不需要编写测试用例。

1.1.2 测试阶段

和开发过程相对应，测试主要按照时间顺序经历单元测试、集成测试、系统测试、验收测试四个阶段。对应关系如图 4-1 所示，

图 4-1 开发与测试的“ V ”型关系



单元测试：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，由开发人员进行，其目的在于发现每个程序模块内部可能存在的 [illegible] 程序员编码过程中已经进行了。本不需要编写测试用例，开发人[illegible]、符合设计要求

就可以了。

集成测试：集成测试是将模块按照设计要求组装起来进行测试，主要目标是发现与接口有关的问题，由于在产品提交到测试部门前，产品开发小组都要进行联合调试，所以大部分企业是由开发人员来完成集成测试的，但也可以到了测试部门后再次进行集成测试。主要测试模块之间数据传输是否正确、模块集成后的功能是否实现、模块接口功能与设计需求是否一致。集成测试紧接在单元测试之后，当单元测试通过后，便可开始配置集成测试环境。**集成测试是最关键的一步，如果问题较多就把产品送到测试部，会造成反复测试，从而浪费人力、物力资源，延误了工期。**

系统测试：系统测试是在集成测试通过后进行，目的是充分运行系统，验证各子系统是否都能正常工作并完成设计的要求。主要由测试部门进行，是测试部门最大最重要的一个测试，对产品的质量有重大的影响。系统测试的主要内容有：功能测试、健壮性测试、性能－效率测试、用户界面测试、安全性测试、压力测试、可靠性测试、安装/反安装测试等。这个测试需要编写大量的测试用例，投入大量的资源来完成。

验收测试：根据需求阶段的《需求规格说明书》为验收标准，测试时要求模拟实际运行环境。对于实际项目可以和客户共同进行，对于产品实际就是最后一次的系统测试。测试内容为对功能模块的全面测试，尤其要进行文档测试。

1.1.3 测试种类、阶段和用例关系

在合适的阶段编写不同的测试用例和执行测试才可以提高效率。为了便于在实际工作中提高效率，同时方便测试用例的编写和执行，可以把上面的提到的测试类型对应的测试用例进行合并。测试用例主要有如下几种：
功能测试用例：包含功能测试、健壮性测试、可靠性测试，
性能测试用例：包含性能测试、压力测试、强度测试
集成测试用例：包含接口测试、健壮性测试、可靠性测试
安全测试用例：安全测试用例
用户界面测试用例：用户界面测试用例、少量功能测试用例。
安装/反安装测试用例：安装/反安装测试用例

测试种类、阶段和用例关系具体的关系如表 4-1 所示。

测试阶段	测试类型	执行人员
单元测试	模块功能测试，包含部分接口测试、路径测试	开发人员
集成测试	接口测试、路径测试，含部分功能测试	开发人员，如果测试人员水平较高可以由测试人员执行
系统测试	功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、压力测试、可靠性测试、安装/反安装测试	测试人员
验收测试	对于实际项目基本同上，并包含文档测试；对于软件产品主要测试相	测试人员，可能包含用户

	关技术文档。	

表 4-1 测试的种类、阶段和用例的关系

1.2 用例编写方案

测试工作和开发通常一同进行，所以在完成测试计划编写后，就可以进行用例的编写工作。测试和开发的对应关系如表 4-2：

开发阶段	依据文档	编写的用例
需求分析结束后	需求文档	系统测试对应的用例
概要设计阶段结束后	概要设计、体系设计	集成测试对应的用例
详细设计阶段	详细设计文档	单元测试对应的用例

表 4-2 用例编写的时间安排

上面的单元测试在大多数企业工作中，由程序员在开发过程中进行，基本不编写测试用例，因此本书中重点论述集成测试用例的编写。下面按照编写的顺序说明各个测试用例的编写思路和方法。

各个用例的编写参考模板见附录。

对于用例，一个基本的思想就是：“ 一点多例 ”，就是针对一个测试点或者功能点，编写多个测试用例，从多个方面进行测试。各个部分的用例编写的都贯穿着这一基本思想。

1.2.1 功能测试用例

功能用例主要是等价类划分。

1.2.2 性能测试用例

为了便于用例的执行与编写，这里的性能测试综合了性能、强度、压力、负载等多方面的测试要求，主要包含内容有：用户并发性能测试、疲劳强度测试、大数据量测试和速度测试、网络、服务器等方面的内容。具体的用例编写模板参照附录二的性能测试用例。

性能测试不同的系统有不同的要求，编写方法要根据实际要求进行编写，本节提出一个参考方案。

下面介绍各个部分性能测试用例包含的内容：

1.2.2.1 预期性能指标测试**用例**

通常系统在设计前都会提出一些性能指标，这些指标是性能测试要完成的首要工作中一。针对每个指标都要编写多个测试用例来验证是否达到要求，根据测试结果来改进系统的性能。指标中通常以单用户为主，如果遇到并发用户的情况，可以归到并发用户测试用例中。

本部分的用例可以使用下面的模板编写，也可以根据实际要求进行编写。

用例编号：	001
性能描述：	
用例目的：	
前提条件：	
特殊的规程说明：	
用例间的依赖关系：	

步骤	输入/动作	期望的性能（平均值）	实际性能（平均值）	回归测试
	示例：典型值…			
	示例：边界值…			
	示例：异常值…			
	…			
	…			
	…			

1.2.2.2 **用户并发性能测试用例**

用户并发测试是性能测试最主要部分。它的过程是一个负载测试和压力测试的过程，主要是逐渐增加用户数量来加重系统负担，直到出现不能接收的性能点或者瓶颈。一般要测试正常情况下和异常情况下两种情况。

并发用户测试要求对系统的核心功能和重要部分进行测试，要以真实的业务数据作为输入，选择有代表性的、关键的业务操作来设计测试用例，更有效的评测系统性能。主要编写如下几个方面的用例：

核心模块的测试（可以理解为“ 单元性能测试 ”）：针对核心功能模块进行并发用户测试，测试系统是否能够稳定运行。例如对于电信计费软件，每月 20 左右是交费的高峰，这时候上千的用户都要查询，然后交费，系统修改用户的存款情况。所以交费管理这一功能要进行并发测试。通过测试可以知道数据库服务器、操作系统、网络设备等是否能够承受住考验，同时可以对瓶颈进行分析。（本部分用例与前一部分重复的可以略过。）

功能	
目的	
方法	

并发用户数与事务执行情况						
并发用户数	事务平均响应时间	事务最大响应时间	平均每秒处理事务数	事务成功率	每秒点击率	平均流量（字节/秒）
20						
25						
30						
35						
40						
45						
50						

并发用户数与数据库主机					
并发用户数	CPU 利用率	MEM 利用率	磁盘 I/O 情况	DB 参数 1	其它参数
20					
25					
30					
35					
40					

45					
50					

并发用户数与应用服务器的关系表			
并发用户数	CPU 利用率	MEM 利用率	磁盘 I/O 情况
20			
25			
30			
35			
40			
45			
50			

业务组合性能测试（可以理解为“ 集成性能测试 ”）：所有的用户不会只使用核心模块，每个功能都可能被使用到，所有既要要模拟多用户的“ 相同 ”操作，又要模拟多用户的不同操作，对多个业务进行组合性能测试。

功能	
目的	
方法	

并发用户数与事务执行情况														
并发用户数	事务平均响应时间			事务最大响应时间			平均每秒事务数			事务成功率			每秒点击率	平均流量（字节/秒）
	业务1	业务2	业务3	业务1	业务2	业务3	业务1	业务2	业务3	业务1	业务2	业务3		
20														
25														
30														
35														
40														
45														
50														

并发用户数与数据库主机					
并发用户数	CPU 利用率	MEM 利用率	磁盘 I/O 情况	DB 参数 1	其它参数
20					
25					
30					
35					
40					
45					
50					

并发用户数与应用服务器的关系表

并发用户数	CPU 利用率	MEM 利用率	磁盘 I/O 情况
20			
25			
30			
35			
40			
45			
50			

1.2.2.3 疲劳强度与大数据量测试

疲劳强度测试是在系统稳定运行下模拟最大用户数目、长时间运行系统，通过综合分析执行指标和资源监控来确定系统处理最大工作量的性能。

编写用例的格式如下：

极限名称 A	例如“ 最大并发用户数量 ”		
前提条件			
运行时间			
输入/动作		输出/响应	是否能正常运行
例如 10 个用户并发操作			
例如 20 个用户并发操作			
…			
故障发生的时刻		故障描述	
……			
任务 A 无故障运行的平均时间间隔		（CPU 小时）	
任务 A 无故障运行的最小时间间隔		（CPU 小时）	
任务 A 无故障运行的最大时间间隔		（CPU 小时）	

大数据量测试分为两种：一个是针对某些系统存储、传输、统计查询等业务进行大数据量的测试；另一个是与前面并发测试相结合的综合数据测试。编写用例时主要编写前一部分，后一部分尽量放在并发测试中。

编写用例的格式如下：

功能						
目的						
方法						
并发用户数与事务执行情况						
输入说明	事务平均响应时间	事务最大响应时间	平均每秒处理事务数	事务成功率	每秒点击率	平均流量（字节/秒）

1.2.2.4 网络性能测试

网络性能测试主要是为了准确展示带宽、延迟、负载和端口的变化是如何影响用户的响应时间的。在实际的软件项目中，主要是测试用户数目与网络带宽的关系。

编写用例的格式如下：

目的	测试广域网网络资源在不同并发用户条件下的使用情况		
方法	在不同的广域网带宽下（64K、128K、256K…）使用 LoadRunner 录制的日常业务的应用脚本，以不同的并发数进行并发性测试，记录各种用户连接数下，不同并发请求的性能变化；同时记录路由器端口的流量和其他数据。		
运行时间			
用户并发数	事务响应时间	端口流量	丢报率

本部分可以独立测试，也可以和用户并发性能测试、疲劳强度与大数据量性能测试结合起来，在原有的基础上采用工具来调整网络设置，从而达到监视网络性能的目的。

1.2.2.5 服务器性能测试

本部分的测试用例不必独立编写，也可以根据实际需要编写少量的测试用例，建议这部分和前两部分的用例编写结合起来，在用户并发性能测试、疲劳强度与大数据量性能测试时完成对服务器性能的监控。例如可以针对 UNIX 测试下面的内容：

监控指标	**描述**
平均负载	系统正常状态下，最后 60 秒同步进程的平均个数
冲突率	在以太网上监测到的每秒冲突数
进程/线程交换率	进程和线程之间每秒交换次数
CPU 利用率	CPU 占用率（%）
磁盘交换率	磁盘交换速率
接收包错误率	接收以太网数据包时每秒错误数
包输入率	每秒输入的以太网数据包数目
中断速率	CPU 每秒处理的中断数
输出包错误率	发送以太网数据包时每秒错误数
包输入率	每秒输出的以太网数据包数目
读入内存页速率	物理内存中每秒读入内存页的数目
写出内存页速率	每秒从物理内存中写到页文件中的内存页数目或者从物理内存中删掉的内存页数目
内存页交换速率	每秒写入内存页和从物理内存中读出页的个数

进程入交换率	交换区输入的进程数目
进程出交换率	交换区输出的进程数目
系统 CPU 利用率	系统的 CPU 占用率（%）
用户 CPU 利用率	用户模式下的 CPU 占用率（%）
……	……

1.2.3 安全测试用例

1.2.4 界面测试用例

1.2.5 集成测试用例

1.3 用例管理

您完全可以把测试用例看成程序——测试工程师编写的程序，这个程序也要经过“设计”、“开发”、“测试”、“版本管理”、“发布”、“维护”等一系列操作。

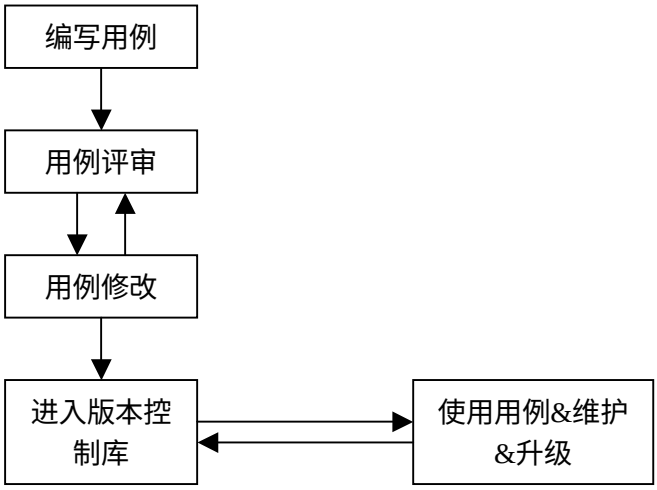
1.3.1 用例评审

用例评审在比较正规的公司更容易实施，同时您的软件开发团队必须在实际工作中对测试给

予足够的重视，才可以把这项工作做好，否则只是走走形式，有效的用例评审通常由下面两种形式组成：
测试部门外部评审——主要是由开发部、项目实施部、甚至销售人员参加的评审，目的主要是查找测试工程师编写的用例是否缺少内容。建议采用非正式评审的形式进行，因为您很难把开发人员组织在一起，通常他们开发进度通常很大，他们能够抽出时间看您的文档已经是“很给面子了”。当然不统一进行会耽误评审的进度，所以在实际工作中如果时间紧迫可以提前启动测试，待评审完成后进行用例的修改工作。通常测试工作进行一段时间评审就会结束，这个时候测试执行人员可以在工作中对测试用例的内容进行动态的调整，再次执行已经执行过的并且进行修改的部分用例。（如果能够采用正式评审效果肯定更好。）
测试部门内部评审——部门内部同行对测试策略的评审，中心是测试策略和用例编制思路是否正确，以次来保证测试用例的有效性。可以组织正式的评审，由用例的设计人员进行讲解，然后大家共同评审；也可以把文档发给部门的同事进行评审。内部评审有些象开发人员在单元测试中交叉测试。
其中的外部评审最重要，因为开发人员很容易发现您的用例遗漏了什么内容没有编制进去，同时还可以发现错误的用例——因为您可能对需求理解存在偏差。用例外部评审可以理解为开发人员在查找您的“程序”的缺陷。
通常情况下先执行内部评审，然后执行外部评审。很多时候，内部评审会被忽略，建议要进行内部评审。这样至少有两个好处：集思广益和提高测试小组输出文档的质量。
附录四为您准备了一个评审记录参考模板，您可以对她进行修改，使之适合您的组织的要求。

1.3.2 管理用例

版本管理是用例管理的核心部分，建议采用工具（例如 Visual SourceSafe）对用例进行控制。建议你的用例参照下面示意图进行管理。



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：4.3.1 小结说明了用例的评审，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。建议如果您的在时间和人力资源比较充裕的情况下对用例的评审要象您测试开发部门的产品一样，要经过反复的评审和修改，然后正式投入使用，因为每次评审您可能都有新的发现。

使用用例：在执行任务时版本控制库取出用例，执行用例时建议您直接在用例上记录测试的结果，这样做给您带来两个好处：首先是下次测试时可以看见上次测试结果，可以起一个提醒的作用，其次您可以统一把发现的缺席输入到数据中，在输入时您可以进行分析，同时避免输入重复的缺陷。每次使用后送入版本控制库，进行版本的管理。
用例升级/维护：随着您的软件产品不断修改、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。

说明：作者设计的用例模板包含了执行的相应记录位置，可以直接进行记录。

（本书是作者有关性能测试用例的一部分，欢迎大家来信交流。）