第3章 chapter 3

软件测试过程

本章介绍了软件测试过程。软件测试也存在着生命周期的概念，即软件测试生命周期，它包括测试计划，测试设计，实施测试以及测试评估等几个部分。

3.1 概　　述

在统一软件开发过程(Rational Unified Process，RUP)中，测试生命周期分为测试计划、测试设计、测试开发、测试执行、缺陷跟踪和评估测试等，如图 3-1 所示。

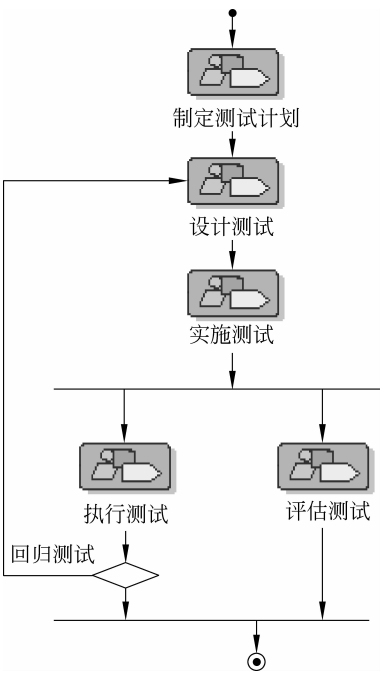

图 3-1　软件测试过程

软件测试过程中必需的基本测试活动及其产生的结果：

1. 拟定软件测试计划(plans)

定义测试项目的阶段，便于对项目进行适当的评估与控制，测试计划包括测试需求，测试策略、测试资源和测试进度。

2. 编制软件测试大纲(outlines)

测试大纲主要说明要测试的内容、参考资料、测试的阶段(内部测试、Alpha 测试、Beta 测试)、要进行哪些类型的测试(功能测试、性能测试、界面测试)、进度安排、人员和资源的需求等。

3. 设计和生成测试用例(test case generation)

设计测试用例及测试过程阶段，是验证测试需求被测试到的最有效方法。

4. 实施测试(execution)

实施测试包含测试的执行过程(如单元测试、集成测试、系统测试等)，是对测试设计

阶段已被定义的测试进行创建或修正的阶段。

5. 生成软件测试报告(software testing reports)

对被测软件进行一系列测试并记录结果,分析测试结果并判断测试的标准是否被满足,一般生成软件问题报告(software problem report)和测试结果报告(test result reports)[6]。

3.2 软件测试计划

测试计划就是描述所有要完成的测试工作,包括被测试项目的背景、目标、范围、方式、资源、进度安排、测试组织,以及与测试有关的风险等方面。测试计划规定测试活动的范围、方法、资源和进度;明确正在测试的项目、要测试的特性,要执行的测试任务、每个任务的负责人,以及与计划相关的风险。测试计划采用的形式是书面文档。

3.2.1 制定测试计划的作用和原则

1. 制定软件测试计划的作用

(1) 使软件测试工作进行更顺利。
(2) 促进项目参加人员彼此的沟通。
(3) 及早发现和修正软件规格说明书的问题。
(4) 使软件测试工作更易于管理。

2. 测试计划制定原则

制定测试计划是软件测试中最有挑战性的一个工作。以下原则将有助于制定测试计划工作。

(1) 制定测试计划应尽早开始。
(2) 保持测试计划的灵活性。
(3) 保持测试计划简洁和易读。
(4) 尽量争取多渠道评审测试计划。
(5) 计算测试计划的投入。

3. 一份好的测试计划书应具备的特点

(1) 能有效地引导整个软件测试工作正常运行,并配合编程部门,保证软件质量,按时将产品推出。
(2) 能使测试高效地进行,即能在较短的时间内找出尽可能多的软件缺陷。
(3) 提供了明确的测试目标、测试策略、具体步骤及测试标准。
(4) 既强调测试重点,也重视测试的基本覆盖率。
(5) 尽可能充分利用公司现有的、可以提供给测试部门的人力和物力资源,而且是可

行的。

(6) 列举的所有数据都必须是准确的,如外部软硬件的兼容性所要求的数据、输入输出数据等。

(7) 对测试工作的安排有一定的灵活性,可以应付一些突然的变化情况,如当时间安排或产品出现的一些变化的时候。

3.2.2 测试计划的内容

制定测试计划时,由于各软件公司的背景不同,测试计划文档也略有差异。实践表明,制定测试计划时,使用正规化文档通常比较好。为了使用方便,在这里给出 IEEE 软件测试计划文档模板,如图 3-2 所示。

IEEE829-1998 软件测试文档编制标准
软件测试计划文档模板

目录
1. 测试计划标识符
2. 介绍
3. 测试项
4. 需要测试的功能
5. 方法(策略)
6. 不需要测试的功能
7. 测试项通过/失败的标准
8. 测试中断和恢复的规定
9. 测试完成所提交的材料
10. 测试任务
11. 环境需求
12. 测试人员的工作职责
13. 人员安排与培训需求
14. 进度表
15. 潜在的问题和风险
16. 审批

图 3-2 IEEE 软件测试计划文档模板

根据 IEEE829-1998 软件测试文档编制标准的建议,测试计划包含了 16 个大纲要项,简要说明如下。

1. 测试计划标识符

一个测试计划标识符是一个由公司生成的唯一值,它用于标识测试计划的版本、等级,以及与该测试计划相关的软件版本。

2. 介绍

在测试计划的介绍部分主要是测试软件基本情况的介绍和测试范围的概括性描述。

3. 测试项目

测试项部分主要是纲领性描述在测试范围内对哪些具体内容进行测试，确定一个包含所有测试项在内的一览表。有功能测试、设计测试、整体测试。

IEEE 标准中指出，可以参考需求规格说明、用户指南、操作指南、安装指南、与测试项相关的报告来完成测试项。

4. 需要测试的功能

列出待测对象的单项功能及组合功能。

5. 不需要测试的功能

列出不测试的单项功能及组合功能并说明不予测试的理由。

6. 测试方法（策略）

测试策略描述测试小组用于测试整体和每个阶段的方法。要描述如何公正、客观地开展测试，要考虑模块、功能、整体、系统、版本、压力、性能、配置和安装等各个因素的影响，要尽可能地考虑到细节，越详细越好，并制作测试记录文档的模板，为即将开始的测试做准备，主要完成以下事项。

（1）确定要使用的测试技术和工具。

（2）确定测试完成的标准。

（3）对于影响资源分配的特殊考虑，如测试与外部接口等。

7. 测试项通过/失败的标准

测试计划中这一部分给出了“测试项”中描述的每一个测试项通过/失败的标准。正如每个测试用例都需要一个预期的结果一样，每个测试项同样都需要一个预期的结果。

下面是通过/失败的标准的一些例子：

- 通过测试用例所占的百分比；
- 缺陷的数量、严重程度和分布情况；
- 测试用例覆盖；
- 用户测试的成功结论；
- 文档的完整性；
- 性能标准。

8. 测试中断和恢复的规定

测试计划中这一部分给出了测试中断和恢复的标准。常用的测试中断标准如下：

- 关键路径上的未完成任务；
- 大量的缺陷；
- 严重的缺陷；

- 不完整的测试环境；
- 资源短缺。

9. 测试完成所提交的材料

测试完成所提交的材料包含了测试工作开发设计的所有文档、工具等。例如，测试计划、测试设计规格说明、测试用例、测试日志、测试数据、自定义工具、测试缺陷报告和测试总结报告等。

10. 测试任务

测试计划中这一部分给出了测试工作所需完成的一系列任务。在这里还列举了所有任务之间的依赖关系和可能需要的特殊技能。

11. 测试所需的资源

测试所需的资源是实现测试策略所必需的。例如：

- 人员——人数、经验和专长。他们是全职、兼职、业余的工作人员还是学生？
- 设备——计算机、测试硬件、打印机、测试工具等。
- 办公室和实验室空间——在哪里？空间有多大？怎样排列？
- 软件——字处理程序、数据库程序和自定义工具等。
- 其他资源——软盘、电话、参考书、培训资料等。

12. 测试人员的工作职责

测试人员的工作职责是明确指出了测试任务和测试人员的工作责任。

有时测试需要定义的任务类型不容易分清，不像程序员所编写的程序那样明确。复杂的任务可能有多个执行者，或者由多人共同负责。

13. 人员安排与培训需求

前面讨论的测试人员的工作职责是指哪类人员（管理、测试和程序员等）负责哪些任务。人员安排与培训需求是指明确测试人员具体负责软件测试的哪些部分、哪些可测试性能，以及需掌握的技能等。实际责任表会更加详细，确保软件的每一部分都有人进行测试。每一个测试员都会清楚地知道自己应该负责什么，而且有足够的信息开始设计测试用例。

培训需求通常包括学习如何使用某个工具、测试方法、缺陷跟踪系统、配置管理，或者与被测试系统相关的业务基础知识。培训需求各个测试项目会各不相同，它取决于具体项目的情况。

14. 测试进度表

测试进度是围绕着包含在项目计划中的主要事件（如文档、模块的交付日期，接口的可用性等）来构造的。

作为测试计划的一部分，完成测试进度计划安排，可以为项目管理员提供信息，以便更好地安排整个项目的进度。表 3-1 给出了一个例子。

表 3-1 相对日期的测试进度

测试任务	开始日期	期限	测试任务	开始日期	期限
测试计划完成	说明书完成之后 7 天	4 周	第 2 阶段测试通过	Beta 构造	6 周
测试案例完成	测试计划完成	12 周	第 3 阶段测试通过	发布构造	4 周
第 1 阶段测试通过	代码计划完成	6 周			

进度安排会使测试过程容易管理。通常，项目管理员或者测试管理员最终负责进度安排，而测试人员参与安排自己的具体任务。

15. 风险及应急措施

软件测试人员要明确地指出计划过程中的风险，并与测试管理员和项目管理员交换意见。这些风险应该在测试计划中明确指出，在进度中予以考虑。有些风险是真正存在的，而有些最终证实是无所谓的，重要的是尽早明确指出，以免在项目晚期发现时感到惊慌。

一般而言，大多数测试小组都会发现自己的资源有限，不可能穷尽测试软件所有方面。如果能勾画出风险的轮廓，将有助于测试人员排定待测试项的优先顺序，并且有助于集中精力去关注那些极有可能发生失效的领域。下面列出一些潜在的问题和风险：

- 不现实的交付日期；
- 与其他系统的接口；
- 处理巨额现金的特征；
- 极其复杂的软件；
- 有过缺陷历史的模块；
- 发生过许多或者复杂变更的模块；
- 安全性、性能和可靠性问题；
- 难于变更或测试的特征。

16. 审批

审批人应该是有权宣布已经为转入下一个阶段做好准备的某个人或某几个人。测试计划审批部分一个重要的部件是签名页。审批人除了在适当的位置签署自己的名字和日期外，还应该签署表明他们是否建议通过评审的意见。

3.3 测试用例

Grenford J. Myers 在 *The Art of Software Testing* 一书中提出：一个好的测试用例是指很可能找到迄今为止尚未发现的错误的测试用例，由此可见测试用例设计工作在整个测试过程中的地位。

影响软件测试的因素很多，例如软件本身的复杂程度、开发人员（包括分析、设计、编程和测试的人员）的素质、测试方法和技术的运用等。因为有些因素是客观存在的，无法避免。有些因素则是波动的、不稳定的，例如开发队伍是流动的，有经验的走了，新人不断补充进来；某位工作人员的情绪受到影响等。如何保障软件测试质量的稳定？有了测试用例，无论是谁来测试，参照测试用例实施，都能保障测试的质量。可以把人为因素的影响减少到最小。即便最初的测试用例考虑不周全，随着测试的进行和软件版本更新，也将日趋完善。

因此，测试用例的设计和编制是软件测试活动中最重要的。测试用例是测试工作的指导，是软件测试的必须遵守的准则，更是软件测试质量稳定的根本保障。

3.3.1 测试用例定义

测试用例（test case）目前没有经典的定义。比较通常的说法是：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。

IEEE 610.12 给出测试用例的定义如下：

（1）测试用例是一组输入（运行前提条件）和为某特定的目标而生成的预期结果及与之相关的测试规程的一个特定集合。

（2）测试用例是一个详细说明测试的输入、期望输出和为一测试项所准备的一组执行条件。

其中，定义（1）给出了测试用例的实质，定义（2）是测试用例的存在方式。

3.3.2 测试用例在软件测试中的作用

1. 指导测试的实施

测试用例主要适用于集成测试、系统测试和回归测试。在实施测试时测试用例作为测试的标准，测试人员一定要按照测试用例严格按用例项目和测试步骤逐一实施测试，并将测试情况记录在测试用例管理软件中，以便自动生成测试结果文档。

根据测试用例的测试等级，集成测试应测试哪些用例，系统测试和回归测试又该测试哪些用例，在设计测试用例时都已作明确规定，实施测试时测试人员不能随意作变动。

2. 规划测试数据的准备

测试实践中测试数据是与测试用例相分离的。按照测试用例配套准备一组或若干组测试原始数据，以及标准测试结果。除正常数据之外，还必须根据测试用例设计大量边缘数据和错误数据。

3. 保证软件的可维护性和可复用性

软件功能模块的通用化和复用化使软件易于开发，只需要修改少部分的测试用例便可以开展测试，测试用例的反复使用提高了测试的效率，缩短了项目周期。

4. 评估测试结果的度量基准

完成测试实施后需要对测试结果进行评估，并且编制测试报告。判断软件测试是否完成、衡量测试质量的优劣需要一些量化的结果。例如，测试覆盖率是多少、测试合格率是多少、重要测试合格率是多少等。以前统计基准是软件模块或功能点，显得过于粗糙。采用测试用例作度量基准更加准确、有效。

5. 分析缺陷的标准

通过收集缺陷，对比测试用例和缺陷数据库，从而分析缺陷是漏测还是缺陷复现。漏测反映了测试用例的不完善，应立即补充相应测试用例，最终达到逐步完善软件质量。而已有相应测试用例，则反映实施测试或变更处理存在问题。

3.3.3 测试用例设计的基本原则

测试用例在设计时应遵循以下原则：

1. 测试用例的代表性

能够代表并覆盖各种合理的和不合理的、合法的和非法的、边界的和越界的以及极限的输入数据、操作和环境设置等。

2. 测试结果的可判定性

测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果。

3. 测试结果的可再现性

对同样的测试用例，系统的执行结果应当是相同的。

3.3.4 测试用例设计应注意的问题

软件测试用例是为了有效发现软件缺陷而编写的包含测试目的、测试步骤、期望测试结果的特定集合。正确认识和设计软件测试用例可以提高软件测试的有效性，便于测试质量的度量，增强测试过程的可管理性。在实际软件项目测试过程中，由于对软件测试用例的作用和设计方法的理解不同，测试人员（特别是刚从事软件测试的新人）对软件测试用例存在不少错误的认识，给实际软件测试带来了负面影响。以下几点是经常遇到的错误做法。

1. 把测试输入数据设计方法等同于测试用例设计方法

现在不少人认为测试用例设计就是如何确定测试的输入数据，从而掩盖了测试用例设计内容的丰富性和技术的复杂性。对于软件功能测试和性能测试，确定测试的输入数据很重要，它决定了测试的有效性和测试的效率。但是，测试用例中输入数据的确定方

法只是测试用例设计方法的一个子集，除了确定测试输入数据之外，测试用例的设计还包括如何根据测试需求、设计规格说明等文档确定测试用例的设计策略、设计用例的表示方法和组织管理形式等问题。

在设计测试用例时，需要综合考虑被测软件的功能、特性、组成元素、开发阶段（里程碑）、测试用例组织方法（是否采用测试用例的数据库管理）等内容。具体到设计每个测试用例而言，可以根据被测模块的最小目标，确定测试用例的测试目标；根据用户使用环境确定测试环境；根据被测软件的复杂程度和测试用例执行人员的技能确定测试用例的步骤；根据软件需求文档和设计规格说明确定测试用例期望的执行结果。

2. 强调测试用例设计得越详细越好

在确定测试用例设计目标时，一些项目管理人员强调测试用例“越详细越好”。具体表现在两个方面：一是尽可能设计足够多的设计用例，测试用例的数量越多越好；二是测试用例尽可能包括测试执行的详细步骤，达到“任何一个人都可以根据测试用例执行测试”，追求测试用例越详细越好。

这种做法和观点最大的危害就是耗费了很多的测试用例设计时间和资源，可能等到测试用例设计、评审完成后，留给实际执行测试的时间就所剩无几了。因为当前软件公司的项目团队在规划测试阶段，分配给测试的时间和人力资源是有限的，而软件项目的成功要坚持“质量、时间、成本”的最佳平衡，没有足够多的测试执行时间，就无法发现更多的软件缺陷，测试质量更无从谈起了。

编写测试用例的根本目的是有效地找出软件可能存在的缺陷，为了达到这个目的，需要分析被测试软件的特征，运用有效的测试用例设计方法，尽量使用较少的测试用例，同时满足合理的测试需求覆盖，从而达到“少花时间多办事”的效果。

3. 追求测试用例设计“一步到位”

一些人认为设计测试用例是一次性投入，测试用例设计一次就可以了，片面追求测试设计的“一步到位”。这种认识造成的危害性使设计出的测试用例缺乏实用性。

任何软件项目的开发过程都处于不断变化过程中，用户可能对软件的功能提出新需求，设计规格说明相应地更新，软件代码不断细化。设计软件测试用例与软件开发设计并行进行，必须根据软件设计的变化，对软件测试用例进行内容的调整，数量的增减，增加一些针对软件新增功能的测试用例，删除一些不再适用的测试用例，修改那些模块代码更新了的测试用例。

软件测试用例设计只是测试用例管理的一个过程，除此之外，还要对其进行评审、更新、维护，以便提高测试用例的“新鲜度”，保证“可用性”。因此，软件测试用例也要坚持与时俱进的原则。

4. 让测试新人设计测试用例

软件测试用例设计是软件测试的中高级技能，不是每个人（尤其是测试新人）都可以编写的，测试用例编写者不仅要掌握软件测试的技术和流程，而且要对被测软件的设计、

功能规格说明、用户试用场景以及程序/模块的结构都有比较透彻的理解。让测试新人设计测试用例是一种高风险的测试组织方式，它带来的不利后果是设计出的测试用例对软件功能和特性的测试覆盖性不高，编写效率低，审查和修改时间长，可重用性差。

因此，实际测试过程中，通常安排经验丰富的测试人员进行测试用例设计。测试新人可以从执行测试用例开始。随着项目的不断开展，测试人员对测试技术和被测软件的不断熟悉，可以积累测试用例的设计经验，编写测试用例。

3.3.5 测试用例的编写标准

在编写测试用例过程中，需要参考和规范一些基本的测试用例编写标准，在 ANSI/IEEE 829 标准中列出了和测试计划相关的测试用例编写规范和模板。标准模板中主要元素如下。

1. 标识符(identification)

每个测试用例应该有一个唯一的标识符，它将成为所有和测试用例相关的文档及表格引用和参考的基本元素，这些文档及表格包括设计规格说明书、测试日志表、测试报告等。

2. 测试项(test item)

测试用例应该准确地描述所需要测试的项及其特征，测试项应该比测试设计说明中所列出的特性描述更加具体，例如做 Windows 计算器应用程序的窗口测试，测试对象是整个的应用程序用户界面，其测试项就应该是应用程序的界面和特性要求，如窗口缩放测试、界面布局、菜单等。

3. 输入标准(input criteria)

用来执行测试用例的输入要求。这些输入可能包括数据、文件或者操作(如，鼠标单击，键盘按键处理等)，必要的时候，相关的数据库、文件也应被罗列。

4. 输出标准(output criteria)

用于标识按照指定的环境和输入标准得到的期望输出结果。尽可能提供适当的系统规格说明来证明期望的结果。

5. 测试用例之间的关联

用于标识该测试用例与其他的测试或测试用例之间的依赖关系。在测试的实际过程中，很多的测试用例并不是单独存在的，它们之间可能有某种依赖关系，例如，用例 A 需要在 B 测试结果正确的基础上才能进行，此时需要在 A 的测试用例中表明对 B 的依赖性，从而保证测试用例的严谨性。

表 3-2 所示是 ANSI/IEEE 829 标准给出的测试用例编写的表格形式，在编写测试用例时可以用来参考。