

## 第5章

# 软件测试基础

1983年,IEEE提出的软件工程标准术语中对软件测试定义如下:“使用人工和自动手段来运行或测试某个系统的过程,其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。软件测试是一个包含许多不同活动的过程,包括测试计划和监控、测试分析、测试设计、测试实施、测试执行、报告测试进度和结果,以及评估测试对象的质量等活动。

本章将系统介绍软件测试的基础知识,包括软件测试的目的、原则和软件测试过程,为学习本书后续内容做准备。



视频讲解

## 5.1 目的和原则

软件测试希望以最少的时间和人力,系统地找出软件中潜在的各种错误和缺陷。如果成功地实施了测试,就能够发现软件中的错误。测试能够证明软件的功能和性能与需求说明相符合。实施测试收集到的测试结果数据为可靠性分析提供了依据。

软件测试并不仅仅是为了找出错误,通过分析错误产生的原因和错误的分布特征,可以帮助项目管理者发现当前所采用的软件过程的缺陷,以便改进。同时,通过分析也有助于设计出有针对性的检测方法,改善测试的有效性。

### 5.1.1 软件测试的目的

基于不同的立场,存在着两种完全不同的测试目的。从用户的角度出发,普遍希望通过软件测试暴露软件中隐藏的错误和缺陷,以考虑是否可接受该产品。从软件开发者的角度出发,则希望测试成为表明软件产品中不存在错误的过程,验证该软件已正确地实现了用户的要求,确立人们对软件质量的信心。

Grenford J. Myers 曾对软件测试的目的提出以下观点。

- 测试是为了发现程序中的错误而执行程序的过程。
- 好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案。
- 成功的测试是发现了至今为止尚未发现的错误的测试。
- 测试并不仅仅是为了找出错误。通过分析错误产生的原因和错误的发生趋势,可以帮助项目管理者发现当前软件开发过程中的缺陷,以便及时改进。
- 测试分析能帮助测试人员设计出有针对性的测试方法,改善测试的效率和有效性。
- 没有发现错误的测试也是有价值的,完整的测试是评定软件质量的一种方法。
- 根据测试目的的不同,还有回归测试、压力测试、性能测试等,分别为了检验修改或优化过程是否引发新的问题、软件所能达到的处理能力和是否达到预期的处理能力等。

软件测试并不等于程序测试。软件测试应贯穿于软件定义与开发的整个期间。需求分析、概要设计、详细设计以及程序编码等各阶段所得到的文档,包括需求规格说明、概要设计说明、详细设计说明以及源程序,都应成为软件测试的对象。

对于给定的任何项目,其测试目标可以包括以下内容:评估工作产品,例如需求、用户故事、设计和代码等;验证是否实现了所有指定的需求;确认测试对象是否完成,并按照用户和其他干系人期望的那样工作;建立对被测对象质量级别的信心;预防缺陷;发现失效和缺陷;降低软件质量不足所带来的风险,例如运行软件时,发现了之前未发现的失效;遵守合同、法律或法规要求或标准和(或)验证测试对象是否符合这些要求或标准。

根据被测组件或系统的环境、测试级别和软件开发生命周期模型的不同,测试目标会有所变化。例如,在组件测试时,目标是尽可能多地发现失效,以便尽早识别和修复潜在的缺陷,而另一个目标可能是增加组件测试时的代码覆盖率;在验收测试时,目标是确认系统能够按照预期工作并且满足用户需求,而另一个目标可能是为干系人提供关于在给定时间发布系统的风险信息。

总之,达到已定义测试目标有助于整个软件开发和维护的成功。对组件和系统及其相关文档进行严格的测试,有助于降低软件运行过程中出现失效的风险。当发现缺陷并加以修复时,有助于提高组件或系统的质量。此外,进行软件测试是为了满足合同或法律法规或行业具体标准的要求。

测试的目的不仅仅是为了发现软件缺陷与错误,也是对软件质量进行度量和评估,以提高软件的质量。因此,需要遵守软件测试的原则。

### 5.1.2 软件测试的原则

软件测试的原则是指帮助测试团队有效地利用他们的时间和精力来发现测试项目的隐藏 Bug 的指导方针。

**原则 1: 测试说明缺陷的存在,而不能说明缺陷不存在。**

测试只能证明软件中存在缺陷,但不能证明软件中不存在缺陷。软件测试是为了降低存在缺陷的可能性,即便是没有找到缺陷,也不能证明软件是完美的。

### 原则 2: 穷尽测试是不可能的。

现在软件的规模越来越大,复杂度越来越高,能做到完全性的测试是不可能的。软件测试应适时终止。此外,应避免冗余测试。在测试阶段,测试人员可以根据风险和优先级来进行集中和高强度的测试,从而保证软件的质量。

### 原则 3: 测试的尽早介入可以节省时间和成本。

为了尽早发现缺陷,应该在软件开发生命周期中尽早启动静态和动态测试活动。测试的尽早介入有时被称为测试的左移。在软件开发生命周期的早期进行测试有助于减少或消除代价高昂的变更。

尽早开展测试准备工作使测试人员能够在早期了解到测试的难度,预测测试的风险,有利于制订出完善的计划和方案,提高软件测试及开发的效率,规避测试中存在的风险。尽早开展测试工作,有利于测试人员尽早发现软件中的缺陷,大大降低错误修复的成本。测试工作进行得越早,越有利于提高软件的质量,这是预防性测试的基本原则。

### 原则 4: 缺陷的群集效应。

软件测试中存在 Pareto 原则: 80% 的缺陷发现在 20% 的模块中。缺陷集群性表明,小部分模块包含大部分的缺陷。一个功能模块发现的缺陷越多,那么存在的未被发现的缺陷也越多,因此发现的缺陷与未发现的缺陷成正比。

### 原则 5: 杀虫剂悖论。

反复使用相同的杀虫剂会导致害虫对杀虫剂产生免疫而无法杀死害虫。软件测试也一样,如果一直使用相同的测试方法或手段,可能无法发现新的缺陷。为了解决这个问题,测试用例应当定期修订和评审,增加新的或不同的测试用例以帮助发现更多的缺陷。但是,在某些情况下,杀虫剂悖论也有好处,例如在自动化回归测试中,发现的回归缺陷数量相对较少。

### 原则 6: 测试活动依赖于测试内容和情境。

软件测试的活动开展依赖于所测试的内容。根据业务的不同分为不同的行业,如游戏行业、电商行业、金融行业。不同的行业,测试活动的开展都有所不同,如测试技术、测试工具的选择、测试流程都不尽相同。不同的测试情境,测试活动不同,如在敏捷模式项目中进行的测试活动不同于在瀑布模型项目中进行的测试。

### 原则 7: 不存在缺陷的谬论。

软件测试不仅是找出缺陷,同时也需要确认软件是否满足需求。如果对错误的需求进行了彻底的测试,有可能 99% 没有 bug 的软件也是不能使用的。如果开发出来的产品不满足用户的需求,即便找到和修复了缺陷作用也不大。

软件测试是一项极富创造性、极具智力挑战性的工作。为了尽可能发现软件中的错误,提高软件产品的质量,在软件测试的实践过程中还有很多其他需要遵守的测试原则。

- 测试应基于用户需求,所有的测试标准应建立在满足客户需求的基础上。从用户角度看,最严重的错误是那些导致程序无法满足需求的错误。应依照用户的需求配置环境并且依照用户的使用习惯进行测试并评价结果。
- 做好软件测试计划是做好软件测试工作的关键。软件测试是有组织、有计划、有步

骤的活动,因此测试必须要有组织有计划,并且要严格执行测试计划,避免测试的随意性。

- 测试前必须明确定义好产品的质量标准。只有建立了质量标准,才能根据测试的结果,对产品的质量进行分析和评估。
- 测试设计决定了测试的有效性和效率。测试工具只能提高测试效率而非万能。根据测试的目的,采用相应的方法去设计测试用例,从而提高测试的效率,更多地发现错误,提高程序的可靠性。
- 测试用例应由测试输入数据和对应的预期输出结果这两部分组成。除了检查程序是否做了应该做的事外,还要看程序是否做了不该做的事;另外,测试用例的编写不仅应当根据有效和预料的输入情况,也需要根据无效和未预料的输入情况。
- 妥善保存测试计划、测试用例、出错统计和最终分析报告,为维护提供方便。
- 程序员应避免检查自己的程序。由于心理因素的影响或者程序员本身错误地理解了需求或者规范导致程序中存在错误,应避免程序员或者编写软件的组织测试自己的软件。一般要求由专门的测试人员进行测试,并且还要求用户参与,特别是验收测试阶段,用户是主要的参与者。

## 5.2 测试过程

影响组织测试过程的环境因素是多方面的,没有统一的软件测试过程,但是有一些常见的测试活动,这些测试活动就组成了一个测试过程。影响测试过程的因素,测试过程中涉及的测试活动,如何实施这些测试活动以及何时开始这些测试活动,都将在本章中进行讨论。

影响测试过程的因素,通常包括以下几个方面:使用的软件开发生命周期模型及项目方法;考虑的测试级别和测试类型;产品风险和项目风险;业务领域;运行限制,例如预算和资源、时间、复杂度、合同和法规要求等;组织方针和实践;所需的内部和外部标准。

测试过程主要由测试计划和监控、测试分析、测试设计、测试实施、测试执行、测试评估和报告、测试结束等主要的活动组组成。每个活动组都是由多个活动构成,每个活动组中的每个活动都可能由多个单独的任务组成,这些任务可能因项目或发布而不同。表 5.1 呈现了从测试小组成立到测试成果归档全测试过程中每个阶段的输入、主要工作人员及成果输出。每个测试过程的活动和工作产品,将在下面的内容中加以详细说明。

表 5.1 测试过程

输入	测试阶段(活动)	输出	主要工作人员
需求规格说明书	测试计划	系统测试计划	测试组长
需求规格说明书 项目原型 系统测试计划	测试分析 测试设计	系统测试用例	测试工程师
测试用例 开发版本	测试实施	阶段测试报告 缺陷报告	测试实施工程师
需求规格说明书 缺陷报表	测试执行 (缺陷跟踪)	缺陷报告 缺陷状态跟踪报告	测试工程师 测试实施工程师



视频讲解

续表

输入	测试阶段(活动)	输出	主要工作人员
测试计划 阶段测试报告	测试评估和报告	阶段测试报告 系统测试报告	测试组长 测试经理
测试计划 系统测试报告	测试工作总结(结束)	测试工作总结	测试经理

虽然这些活动组中的许多活动在逻辑上看起来是顺序的,但它们通常是迭代实现的。即使在顺序开发中,活动的阶梯式逻辑顺序也会涉及重叠、组合、并发或省略,所以必须根据系统和项目的情境来对测试活动进行适当裁剪。

软件测试过程与软件开发流程类似,也包括计划、设计、开发、执行和评估 5 个阶段,图 5.1 展示了软件项目实施过程中软件开发生命周期和软件测试生命周期的主要活动。软件项目一启动,软件测试也就开始了。由于软件的复杂性和抽象性,在软件开发生命周期各阶段都可能产生错误,所以不应把软件测试仅仅看作是软件开发的一个独立阶段,而应当把它贯穿到软件开发的各个阶段中。

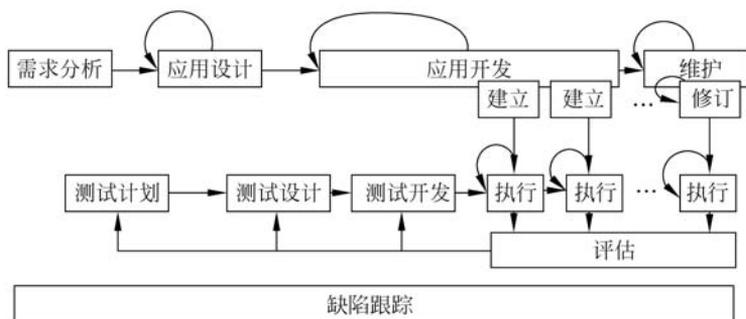


图 5.1 软件开发生命周期和测试生命周期的主要活动

在需求分析和应用设计阶段就应开始进行测试工作,编写相应的测试计划及测试设计文档;应用开发阶段进行相应的测试开发活动,如编写测试脚本、开发自动化测试工具等;在开发各个阶段执行包括组件测试、集成测试、系统测试和验收测试等各个级别的测试任务,评估软件产品是否实现开发目标;当软件项目进入维护阶段,测试活动仍需在改正性维护、适应性维护、完善性维护和预防性维护等活动中持续进行。同时,对于缺陷的跟踪,要贯穿于整个开发生命周期和测试生命周期。坚持在开发各阶段进行技术评审和验证,这样才能尽早发现和预防错误,杜绝某些缺陷和错误,提高软件质量。

软件测试和软件开发一样,都遵循软件工程原理,遵循管理学原理。测试专家通过实践总结出很多很好的测试模型,软件测试模型将在 6.1 节介绍。这些模型将测试活动进行了抽象,明确了测试与开发之间的关系,是测试管理的重要参考依据。

### 5.2.1 测试计划和监控

测试计划包括的活动有定义测试目标,在环境因素限制下指定适当的测试技术和方法,

并制定满足截止期限的测试进度表。根据测试监控活动的反馈,可以重新审议测试计划。制订测试计划的目的是收集和整理测试计划信息,并且创建测试计划。制订计划受组织的测试方针和测试策略、使用的开发生命周期和方法、测试范围、目标、风险、约束、关键性、可测试性和资源可用性的影响。随着项目和测试计划的进展,测试计划中包含的内容便越来越多,越来越详细。测试计划是一项持续的活动,贯穿于整个产品的生命周期。前期的测试计划通常以需求规格说明书和测试计划作为输入。

测试计划工作流程如图 5.2 所示。根据需求工件集收集和整理测试需求信息,确定测试需求;制定测试策略,针对测试需求定义测试类型、测试方法以及需要的测试工具等;建立测试通过准则,根据项目实际情况为每一个层次的测试建立通过准则;确定资源和进度,确定测试需要的软硬件资源、人力资源以及测试进度;评审测试计划,根据同行评审规范对测试计划进行同行评审。

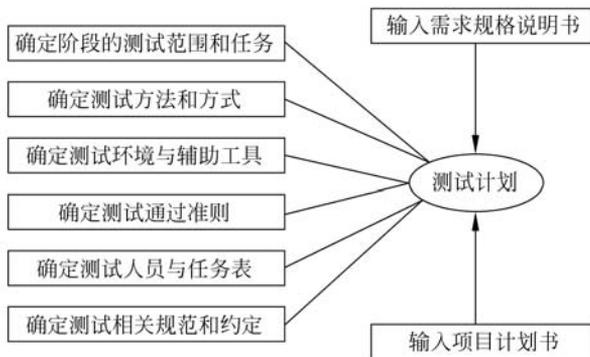


图 5.2 测试计划工作流程

测试监控包括使用测试计划中定义的测试监控度量,对实际进度与测试计划进行持续的比较,还包括采取必要的措施来满足测试计划的目标。例如,当已识别的风险发生时,重新确定测试优先级。又如,由于测试环境或其他资源的可用性或不可用性而更改测试进度表。

测试计划的工作产品通常包括一个或多个测试计划,测试计划包括关于测试依据与其他测试工作产品之间可追溯性的相关信息。测试监控的工作产品通常包括各种类型的测试报告,包括测试进度报告和测试总结报告。

测试计划作为软件项目计划的子计划,在项目启动初期是必须规划的,是测试工作得以顺利开展的基础。在越来越多公司的软件开发中,软件质量日益受到重视,测试过程也从一个相对独立的阶段越来越紧密地嵌套在软件整个生命周期中。

具体地,一份测试计划应体现以下内容,详见表 5.2。

表 5.2 测试计划内容

测试内容	说明
目标	必须明确定义每个测试阶段的目标
通过准则	设计准则来判断每个测试阶段的完成标准

续表

测试内容	说明
进度	每个阶段都需要日程安排,指出何时设计、编写、执行测试用例
职责	每个阶段必须有负责识别设计、编写、执行和验证测试用例的人员,以及修订被发现的错误的人员。在大型项目中,可能会引起有些测试结果是否是错误的争论,所以还需要识别仲裁人
测试用例库和标准	在一个大型项目中,必须要有系统的关于识别、编写、存储测试用例的方法
工具	识别缺陷所需的测试工具,包括谁去开发或去获取工具,工具将如何被使用、何时使用等
计算机时间	这是关于每个测试阶段所需的计算机时间的总量的计划,包括编译应用程序的服务器、安装测试的桌面机、Web应用的Web服务器、网络设备等
硬件配置	如果需要特殊的硬件配置或设备,需要一个计划来描述这种需求,它们如何满足、何时需要
集成	系统集成计划定义了集成的次序,定义程序如何结合在一起。一个包含大量子系统或程序的系统可以增量(从上到下或从下到上)地结合起来。构造块是程序或子系统
跟踪过程	制定跟踪机制来跟踪测试的全过程,包括倾向于错误的模块的定位、计划、资源、完成准则等各方面进展的估计
调试过程	定义了机制来报告检测到的错误,跟踪纠正的进展,将纠正好的添加到系统中。计划、职责、工具、计算机时间、资源都是调试计划的组成部分
回归测试	做了功能改进或对程序修订后,需要执行回归测试。目的是确定改变是否已经回归了程序的其他方面。一般是通过允许重新执行程序的测试用例的子集来确认。回归测试的重要性在于变更和纠错倾向于产生更多的错误,所以一份回归测试的计划是有必要的

在工程实践中,一份好的测试计划可以使测试工作和整个开发工作融合起来,资源和变更也成为可控制的风险。测试计划中的测试阶段的划分尤为重要。对于采用“瀑布型”开发方式的项目,各个阶段的主要活动很清晰,易于操作。然而,在制订测试计划时,往往把测试单纯地简化为系统测试,或者把各种类型的测试设计(如测试用例编写和测试数据准备)全部放入生命周期的“测试阶段”。这样,一方面浪费了开发阶段可以并行的项目日程,另一方面造成测试不足。

合理的测试阶段的建议划分方法,如表 5.3 所示。横向代表开发的几个典型阶段,包括需求分析、设计、编码、组件测试、集成测试、系统测试和验收测试阶段;纵向代表测试级别,从下至上依次为组件测试、集成测试、系统测试及验收测试。相应阶段可以同步进行相应的测试计划编制,而测试设计也可以结合开发过程而实现并行,测试执行则可以在开发之后实施。组件测试和集成测试往往由开发人员承担,因此这部分的阶段划分可能会安排在开发计划而不是测试计划中。例如,验收测试阶段,验收测试的测试计划在需求分析阶段即开始制订,而在设计、编码直到系统测试阶段,都可以并行进行测试设计,而当项目进入验收测试阶段,才开始执行验收测试任务。

表 5.3 测试阶段划分

测试	阶 段						
	需求分析	设计	编码	组件测试	集成测试	系统测试	验收测试
验收测试	计划	设计					执行
系统测试	计划	设计				执行	
集成测试		计划	设计		执行		
组件测试			计划/设计	执行			

### 5.2.2 测试分析

在测试分析过程中,根据可测量的覆盖标准来确定“测试什么”。测试分析主要包括以下活动。

(1) 分析相应测试级别的测试依据。在软件工程生命周期的各个阶段,将输出多项工作产品。例如,需求规格说明,如业务需求、功能需求、系统需求、用例或类似的工作产品,其中指定了所需的功能和非功能的单元或系统行为;设计和实现信息,如系统或软件架构图或文档、设计规格说明、调用流、模型图(如 UML 或实体关系图 ERD)、接口规格说明或类似的工作产品,其中指定了单元或系统的结构;单元或系统本身的实现,包括代码、数据库元数据、查询和接口;风险分析报告,其考虑到了单元或系统的功能、非功能及结构问题。以上工作产品都是进行测试分析的重要依据。

(2) 评估测试依据和测试项,以识别各种类型的缺陷。例如,在需求规格说明中可能存在以下缺陷:功能描述语言的歧义或不准确,功能的遗漏,性能指标要求的不一致等,根据测试级别将识别出的缺陷列入测试项。

(3) 识别被测特性和特性集。

(4) 根据对测试依据的分析,并考虑到功能、非功能和结构特征、其他业务和技术因素以及风险级别,界定每个特征的测试条件并确定其优先次序。

(5) 在测试依据的每个元素与相关测试条件之间获取双向可追溯性。

进行有效的测试分析,往往采用以下三种方法。

(1) 质量模型分析法。针对每个功能使用软件质量模型进行分析,分析应测特性,确认各功能的测试点以及测试。

(2) 功能交互分析法。针对不同的功能确认各功能之间的交互操作,分析各功能交互时的测试特性,测试注意点,确认测试项。

(3) 用户场景分析法。针对所有功能,站在用户的角度考虑用户会怎么操作和使用这个功能,分析确认测试点以及测试项。

在测试分析过程中,使用黑盒测试技术、白盒测试技术及基于经验的测试技术,可以减少遗漏重要测试条件的可能性,并且能更精确和准确地定义测试条件。

测试分析工作产品包括已定义的和按优先级排序的测试条件,理想情况下每一个测试条件都可以双向追溯到它所覆盖的测试依据的特定元素。

### 5.2.3 测试设计

测试分析回答了“测试什么”的问题,而测试设计回答“如何测试”的问题。测试设计为每一个测试需求确定测试用例集,并且确定执行测试用例的测试过程。设计测试用例,对每一个测试需求确定其所需的测试用例,对每一个测试用例确定其输入及预期结果;确定测试用例的测试环境配置、需要的驱动界面或稳定桩;编写测试用例文档。开发测试过程,根据界面原型为每一个测试用例定义详细的测试步骤,为每一个测试步骤定义详细的测试结果验证方法;为测试用例准备输入数据;编写测试过程文档;在实施测试时对测试过程进行更改。设计驱动程序或稳定桩,设计组件测试和集成测试需要的驱动程序和稳定桩。

测试设计包括以下主要活动。

- (1) 设计测试用例和测试用例集,并确定其优先级。
- (2) 识别所需的测试数据,以支持测试条件和测试用例。
- (3) 设计测试环境并识别所需的基础设施和工具。
- (4) 擷取测试依据、测试条件、测试用例和测试规程之间的双向追溯性。

图 5.3 显示了测试设计的准备工作和过程。项目启动初期进行了系统划分及风险分析,测试计划阶段也对项目可能风险进行了进一步的识别和分析,明确了测试策略,为选择恰当的测试设计技术做好准备工作;项目需求分析阶段明确了项目的设计需求,制订了测试计划,并进行测试分析输出了测试条件和依据;最后,以测试依据为输入,运用选定的测试设计技术,完成测试设计。测试设计生成测试用例和测试用例集,以覆盖测试分析中定义的测试条件。在测试设计时,将测试条件细化成概要测试用例、概要测试用例集和其他测试件。

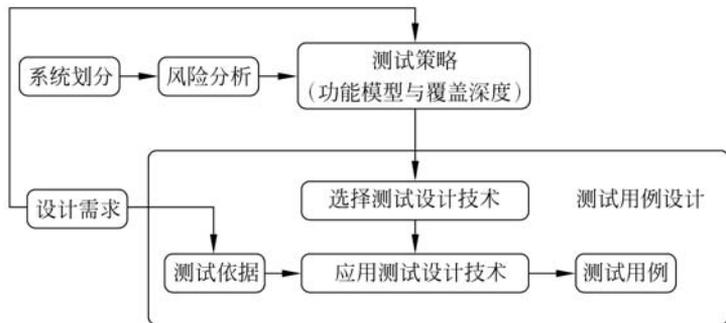


图 5.3 测试设计的准备工作和过程

测试设计是整个软件测试过程中的一个重要活动,其输出质量(无论是文档化的工作产品,还是存在于测试人员头脑中的想法)将会直接影响后续测试活动的效率和有效性,进而影响软件产品的最终质量。优质的测试设计可有效地减少测试用例数目,避免测试用例之间的冗余,满足客户对产品的不同质量要求,应付紧迫的测试时间和有限的测试资源,适应需求的不完善和变更,快速获得产品的质量信息等。

测试设计的主要活动是测试用例的设计,应用恰当的测试设计技术才能输出全面且无冗余的测试用例。测试设计技术包括静态测试技术和动态测试技术,将分别在第7章和第8章详细介绍。

#### 5.2.4 测试实施

测试设计回答了“如何测试”的问题,而测试实施则要回答“我们现在是否已经有了运行测试所需的一切条件”。测试设计和测试实施任务通常是结合在一起的。在测试实施期间,创建和(或)完成测试执行所需的测试件,包括将测试用例排序为测试规程。测试实施包括以下主要活动。

- (1) 开发并确定测试规程的优先级,如有可能,同时创建自动化测试脚本。
- (2) 根据测试规程和自动测试脚本(如果已创建)来创建测试套件。
- (3) 在测试执行进度表中,以促进有效测试执行的方式安排测试套件。
- (4) 构建测试环境,包括可能的测试工具、服务虚拟化、模拟器和其他基础设施项目等,并验证一切所需条件都已正确设置。
- (5) 准备测试数据并确保在测试环境中正确地加载。
- (6) 确认并更新测试依据、测试条件、测试用例、测试规程和测试套件之间的双向可追溯性。

测试实施工作产品包括测试规程以及这些测试规程的顺序、测试套件和测试执行进度表等。

#### 5.2.5 测试执行

在测试执行期间,测试套件按照测试执行进度表运行。测试执行包括以下主要活动。

- (1) 记录测试项或测试对象、测试工具及测试件的 ID 和版本。
- (2) 手工或者使用测试执行工具执行测试。
- (3) 将实际结果与预期结果进行比较。
- (4) 分析异常现象,以确定它们可能发生的原因。例如,出现失效可能是由于代码中的缺陷,但也可能出现误报。
- (5) 根据实际观察到的失效报告缺陷。
- (6) 记录测试执行的结果。例如,通过、失败、阻塞等结果。
- (7) 作为对异常现象采取行动的结果,或作为计划要测试的一部分,重复测试活动。例如,执行修正后的测试、确认测试和(或)回归测试。
- (8) 确认并更新测试依据、测试条件、测试用例、测试规程和测试结果之间的双向可追溯性。

测试执行工作产品包括记录各单个测试用例或测试规程状态的文档,缺陷报告,测试过程中包含测试项、测试对象、测试工具及测试件的文档。

### 5.2.6 测试评估和报告

测试评估是结合量化的测试覆盖域及缺陷跟踪报告,对应用软件的质量和开发团队的工作进度及工作效率进行综合评价。创建测试总结报告,并将信息传达给干系人。例如,由需求产生的测试缺陷,设计人员将根据测试评估报告的数据修正设计;若测试缺陷由实现产生,开发人员将根据本测试评估报告的数据修正代码。

测试评估和报告的工作产品包括测试评估报告和测试总结报告等文档。测试评估报告通常从软件能力、缺陷情况和建议三方面对软件进行评价。将软件的测试结果与功能需求做比较,评价软件能力达到需求规格说明书规定的的能力要求的情况;对软件测试结果中的缺陷加以总结,汇总操作中发现较大问题的功能,准备下一步的开发和测试计划;通过测试,对软件测试欠缺的方面加以总结,比如本次测试虽然完成了大部分的功能测试,但由于操作方式多变,所以建议使用更多测试样例来测试该软件的可靠性。

### 5.2.7 测试结束活动

测试结束活动从已完成的测试活动中收集数据,以强化经验、测试件以及任何其他相关信息。测试结束活动出现在项目里程碑点,例如,当软件系统发布时、当测试项目完成或取消时、当敏捷项目的迭代完成时、当测试级别完成时或当维护版本完成时。

测试结束包括以下主要活动。

(1) 检查是否所有的缺陷报告已关闭;测试执行结束时仍未解决的缺陷,是否已创建需求变更或产品待办事项。

(2) 最后确定并归档测试环境、测试数据、测试基础设施及其他相关测试件,以便以后重复使用。

(3) 将测试件移交给维护部门、其他项目团队和(或)其他可以从使用测试件中获益的干系人。

(4) 从已完成的测试活动中,分析所获得的经验教训来确定以后迭代、版本和项目所需的变更。

(5) 使用收集到的信息来改进测试过程的成熟度。

测试结束的工作产品包括改进后续项目或迭代的行动项、变更请求或产品待办事项以及最终的测试件。



视频讲解

## 5.3 案例：测试工作流程

软件测试的质量从根本上是由软件测试流程决定的。预防缺陷在软件生命周期的早期,需要有专人对测试流程各环节负责。本节是一个测试团队的日常工作规范示例,主要侧重测试工作流程的控制,明确软件工程的各阶段测试团队应完成的工作。

测试团队共同协同完成测试工作,在人力资源有限的情况下,一个团队成员可能会同时承担多个角色。测试团队中的角色划分及其主要责任,详见表 5.4。

表 5.4 测试团队角色划分及其主要责任

角 色	主 要 责 任
测试经理	组建测试组
	协调测试组内部的沟通,代表测试组与其他角色组进行沟通; 统筹安排测试人员,管理各项目测试进度,并对测试提供支持; 制定和改进测试规范,考核测试组人员; 组织测试人员交流培训; 测试报告汇总分析
测试组长	编写测试计划; 管理测试进度和报告遇到的问题; 测试报告分析; 辅导测试工程师和测试实施工程师
自动化测试工程师	编写测试用例,指导测试工程师和测试实施工程师; 开发测试工具,编写和执行自动化测试脚本
测试工程师	编写测试用例; 完成测试阶段报告; 协助完成系统测试报告
测试实施工程师	实施测试用例,执行测试; Bug 申报,缺陷跟踪
软件维护工程师	编写产品手册、用户使用说明; 负责对用户进行使用培训,反馈用户使用情况

测试工作流程及规范,如图 5.4 所示。图中显示了从测试组成立到测试工作总结的完整测试工作过程,包括需求理解、测试计划、测试用例设计、测试执行、缺陷跟踪、测试报告、测试工作总结及归档。图中展示了对应阶段的输入依据、输出工作产品及主要责任人,为测试过程的实施提供了规范。

接下来对测试计划阶段、测试分析与设计阶段、测试实施和执行阶段、测试评估和报告阶段及测试结束等核心阶段的具体过程进行详细介绍。

### 1. 测试计划阶段

在项目组成立的同时,测试组也将成立。成立测试团队时,测试经理负责为测试组任命一名测试组长,同时确定测试组的构成人选。

在制订测试计划之前,项目经理、测试组长和开发组长会依据初始需求开发原型召开需求理解会议。会上,测试人员从流程逻辑、边界定义等角度理解需求,并提出建议,帮助确认软件需求的边界;统一项目组的软件目标和测试的工作重点。当开发组和测试组对需求的理解基本一致时,会议结束。

需求分析文档确立后,测试组需要编写测试计划文档,为后续的测试工作提供直接的指导。编写测试计划的输入条件、工作内容、退出标准和责任人,如表 5.5 所示。

### 2. 测试分析与设计阶段

在实际的测试中,测试用例将是重要的实施标准。测试用例设计的具体任务和责任人如表 5.6 所示。

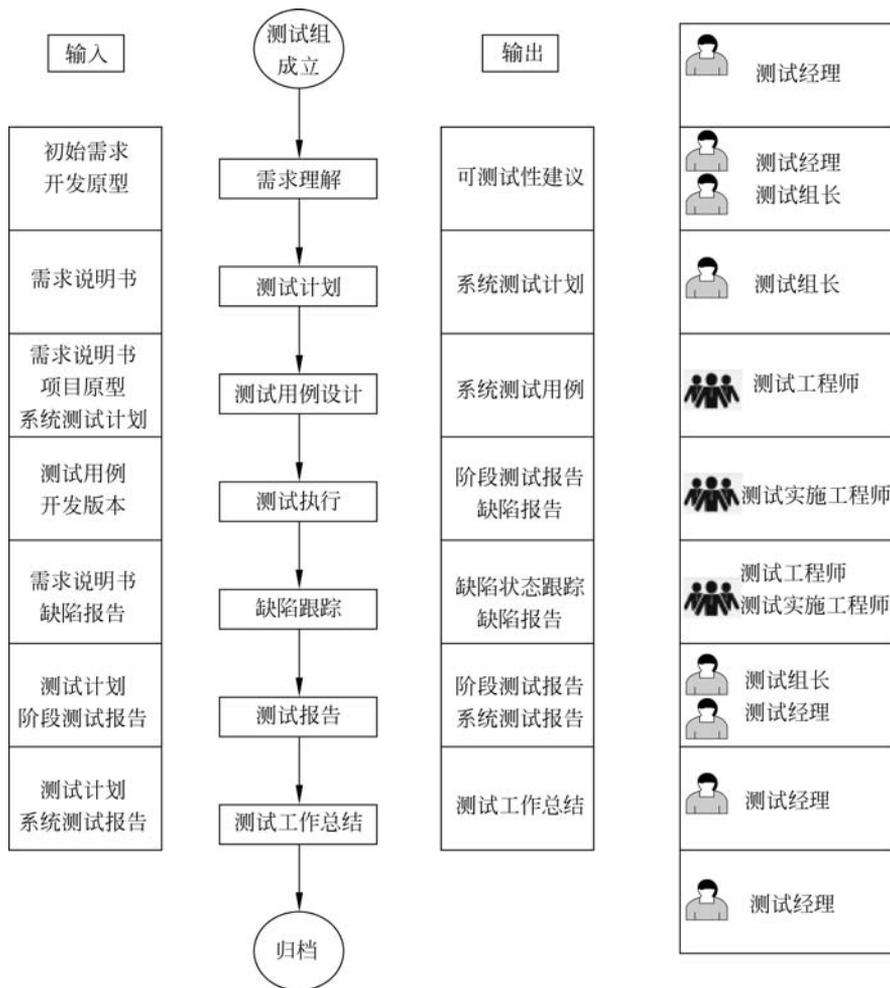


图 5.4 测试工作流程及规范

表 5.5 编写测试计划过程

过程要点	详细说明
输入条件	项目需求分析文档确立,原型完成
工作内容	<p>根据项目的需求文档,按照测试计划文档模板编写测试计划。测试计划中应该至少包括以下关键内容。</p> <ul style="list-style-type: none"> <li>测试需求:需要测试组测试的范围,估算出测试所花费的人力资源和各个测试需求的测试优先级。</li> <li>测试方案:整体测试的测试方法和每个测试需求的测试方法。</li> <li>测试资源:本次测试所需要用到的人力、硬件、软件的资源。</li> <li>测试组角色:明确测试组内各个成员的角色和相关责任。</li> <li>里程碑:明确标准项目过程中测试组应该关注的里程碑。</li> <li>可交付工件:在测试组的工作中必须向项目组提交的产物,包括测试计划、测试报告等。</li> </ul> <p>测试计划编写完毕后,必须交由项目组中各个角色组联合评审</p>

续表

过程要点	详细 说明
退出标准	《测试计划》由项目组评审通过。 在项目开发过程中,要适时地对测试计划进行跟踪和必要的改进,在项目结束时还要最后评估一下测试计划的质量
责任人	测试组长

表 5.6 测试用例设计过程

过程要点	详细 说明
输入条件	测试需求明确,测试计划明确
工作内容	根据测试需求和测试计划定义的测试范围编写全部的测试用例
退出标准	测试用例需要覆盖所有的测试需求,提交项目组评审
责任人	测试工程师

测试用例草稿完成后,项目组组长和专家组主持召开测试用例评审会议,对《测试用例》进行评审。评审内容包括:用例覆盖所有需求定义的范围;用例预期结果与需求预期效果匹配程度;用例测试的重点与测试计划匹配程度;根据会议结果,测试组修改和补充《测试用例》。《测试用例》通过评审,与开发同时发布测试用例版本。

### 3. 测试实施和执行阶段

开发和测试都发布了对应的版本后,由开发组提出,项目经理确认,测试申请转测试经理,测试组长和测试实施工程师作为主要负责人正式实施测试。具体如表 5.7 所示。

表 5.7 实施测试用例过程

过程要点	详细 说明
输入条件	测试经理提前得到测试申请,确认测试计划和可用的测试用例
工作内容	测试组长明确测试计划,管理测试进度,报告缺陷和遇到的问题; 测试实施工程师根据测试计划和测试用例,实施相应的测试用例,并将记录实施用例的结果
退出标准	测试用例中的所有任务被执行,结果被记录,所有发现的 Bug 被报告
责任人	测试组长,测试实施工程师

在每个测试周期完成之后(包括回归测试周期),测试组长需要总结此次的测试结果,编写测试报告。提交测试阶段报告的过程,详见表 5.8。

表 5.8 提交测试阶段报告过程

过程要点	详细 说明
输入条件	测试组完成了预定周期的测试任务
工作内容	根据此轮测试的结果,编写测试阶段汇总报告,主要包含以下内容。 <ul style="list-style-type: none"> <li>• 测试报告的版本,测试的人员和时间。</li> <li>• 测试所覆盖的缺陷:测试组在这轮测试中所有处理的缺陷,包括实施工程师验证的缺陷。</li> <li>• 测试新发现的缺陷数量。</li> <li>• 经过此轮测试,所有活动缺陷的数量及其状态分类。</li> <li>• 亟待解决的问题:写明当前项目组中面临的优先级最高的问题</li> </ul>

续表

过程要点	详细说明
退出标准	在每轮测试结束之后应尽快将符合标准的阶段测试报告发给全项目组
责任人	测试组长

在每轮测试结束之后,由开发组发布修改后的最新版本,提出测试申请,测试组开始回归测试。回归测试的过程如 5.9 所示。

表 5.9 回归测试过程

过程要点	详细说明
输入条件	新需求完成了功能实现,测试组确认测试申请
工作内容	测试组将按照测试计划中对于回归测试的策略对产品进行回归测试;回归测试的用例属于测试用例的一部分或者是全部; 对于新功能的实现,要求先补充新的测试用例,再进行测试
退出标准	回归测试全部通过;新需求用例测试全部执行
责任人	测试工程师,测试实施工程师

#### 4. 测试评估和报告阶段

测试工作即将结束时,测试组就要开始着手准备进行评估和报告的工作。在回归测试结束后,测试组长要编写测试总结报告,对测试进行总结,并且提交给全体项目组,为产品的后续工作提供重要的信息支持。编写测试总结报告的过程,详见表 5.10。

表 5.10 编写测试总结报告

过程要点	详细说明
输入条件	回归测试完成,或测试达到项目组要求的程度
工作内容	<p>测试组长根据测试的结果,按照测试总结报告的文档模板编写测试总结报告,报告必须包含以下重要内容。</p> <ul style="list-style-type: none"> <li>• 测试资源概述:多少人、多长时间。</li> <li>• 测试结果摘要:分别描述各个测试需求的测试结果,产品实现了哪些功能点,哪些还没有实现。</li> <li>• 测试需求覆盖率:原先列举的测试需求的测试覆盖率。</li> <li>• 缺陷分析:按照缺陷的属性分类进行分析。</li> <li>• 测试评估:从总体对项目质量进行评估。</li> <li>• 测试组建议:从测试组的角度为项目组提出工作建议</li> </ul>
退出标准	测试组长完成了符合标准的测试报告,发送给全项目组
责任人	测试组长

#### 5. 测试结束

测试验收工作是在以上工作全部结束后,对测试的过程、效果进行验收,宣布测试结束。测试验收的过程,如表 5.11 所示。

表 5.11 测试验收过程

过程要点	详细说明
输入条件	测试组完成了所有的测试实施工作,测试组长提交了测试总结报告
工作内容	由验收组成员,对本测试进行验收,验收内容如下。 测试效果验收:测试是否达到预期目的。 测试文档验收:测试过程文档是否齐全、可信、符合标准。 测试评估:从总体对测试的质量进行评估。 测试建议:对本次测试工作指出不足,需要在以后工作中改进的地方。 宣布测试结束:测试验收组成员签字宣布本次测试结束
退出标准	签发测试验收报告
责任人	项目经理

测试总结工作是在以上的工作全部结束以后,它的目的是评估本次测试工作,总结经验,使下一次的工作做得更好。测试总结报告提交测试经理后,测试经理根据项目的测试总结报告,按照测试总结的文档模板编写测试总结,测试经理最后将测试总结发送给全测试组。

在结束测试后,对测试过程中涉及的各种标准文档进行归类、存档,完成测试归档。测试归档由测试经理负责,归类、存档测试过程涉及的文档主要包括《测试计划》《测试用例》《测试报告》《测试验收》和《测试总结》。全部文档归类完毕,版本号封存。

## 5.4 本章小结

测试是软件项目开发过程中的重要组成部分,其主要职责是多方面的:在项目的前期、需求文档确立基线前对需求文档进行走查,从用户体验和测试的角度提出可测试性建议;编写合理的测试计划,并与项目整体计划有机结合;确定项目的测试重点,安排人员、设备、时间;编写覆盖率高的测试用例,包括功能测试、压力测试、性能测试、兼容性测试等;针对测试需求进行相关测试技术的研究,包括开发技术、自动化测试技术等;认真仔细地实施测试工作,并及时提交测试报告供项目组参考;进行缺陷跟踪与分析。

软件测试是一项极富创造性、极具智力挑战性的工作,为了尽可能发现软件中的错误,提高软件产品的质量,在软件测试的实践过程中还有很多需要遵守的测试原则。这些软件测试的原则能够帮助测试团队有效地利用时间和精力来发现测试项目的隐藏 Bug。

影响组织测试过程的环境因素是多方面的,没有统一的软件测试过程,但是有一些常见的测试活动,这些测试活动就组成了一个测试过程。测试过程主要由测试计划和监控、测试分析、测试设计、测试实施、测试执行、测试评估和报告、测试结束等主要的活动组组成。每个活动组都是由多个活动构成,每个活动组中的每个活动都可能由多个单独的任务组成,这些任务可能因项目或发布而不同。