

第5章



软件测试管理

在当前软件开发规模不断增加、复杂程度迅速提高的社会,寻找软件中存在问题为最终目的的测试工作将变得愈发艰难。但是,为了更好地寻找出程序中存在的相关问题,形成更高质量的软件产品,应当强化对于测试工作的组织和管理工作。在项目管理领域,项目管理的理论体系很多,对项目管理的理解也各不相同,各种组织的最佳实践模型更是数不胜数。本章将讲述对一个具体的软件测试项目而言,需要哪些管理工作才能让项目可控,并且朝着成功的方向前进。

本章要点

- 软件测试项目的基本特性
- 软件测试项目管理的特性和原则
- 软件测试管理计划制订
- 主要的软件测试文档
- 软件测试过程控制

5.1 软件测试管理概述

5.1.1 软件测试项目

软件测试项目是在一定的组织机构内,利用有限的人力和财力等资源,在指定的环境和要求下对特定软件完成特定测试目标的阶段性任务,该任务应满足一定质量、数量和技术指标等要求。软件测试项目一般具有以下基本特性。

1. 项目的独特性

每个测试项目都有属于自己的一个或多个目标,都有明确的时间期限、费用、质量和技术指标等方面的要求。

2. 项目的不确定性

软件测试项目目标不明确,软件质量标准定义不准确,任务边界模糊,找不到严重的缺陷并不代表软件不存在严重的缺陷以及难以确定什么时候软件测试可以结束。还有在测试项目过程中遇见的技术、规模等方面的因素,这些都可能导致软件测试项目的失败。

3. 智力、劳动密集性

软件测试项目具有智力密集、劳动密集的特点,受人力资源影响最大,项目成员的结构、责任心、能力和稳定性对测试执行、产品质量有很大的影响。因此,软件测试项目要求人力资源十分稳定,这不仅是一个技术工作,而且要求测试人员对产品的功能、特性非常了解。

4. 测试项目的困难性

由于每个公司的测试人员水平、工作内容、工作时间、组织结构不同以及软件模块重要性有差异,测试任务的分配有一定难度。而且,软件测试项目在变化控制和预警分析方面要求较高。

5. 测试项目的目标冲突性

每个测试项目都会在实施的范围、时间、成本等方面受到一定的制约,这些制约称为“三约”。为了取得测试项目的成功,必须同时考虑这3个主要因素。而这些目标并不总是一致的,往往会出现冲突,如何取得彼此之间的平衡,也是影响测试能否成功完成的重要因素。

正是由于软件测试项目存在以上特点,尤其是存在不确定性和困难性,因此,优秀的测试人员和科学的管理是测试项目成功的关键。

5.1.2 软件测试项目管理

软件测试项目管理是以测试项目为管理对象,通过测试人员运用专门的软件测试知识、技能、工具和方法,对测试项目进行计划、组织、执行和控制,并在时间成本、软件测试质量等方面进行分析和管理的。测试项目管理贯穿测试项目启动阶段、计划阶段、实施阶段、收尾阶段的整个测试项目生命周期,是对测试项目的全过程进行管理。

软件测试项目管理虽然涉及诸多因素,如成本、质量、时间、资源等,但实际问题可以归结于人员、问题和过程,如图5-1所示。

软件测试项目管理具有以下一些基本特性。

(1) 系统工程的思想贯穿软件测试项目管理的全过程。软件测试项目管理将测试项目看作一个完整的系统,可以将软件系统测试分散成多个阶段,每个阶段有不同的任务、特点和方法,需要分别按要求完成,任何阶段或部分任务的失败都有可能对整个测试项目的结果产生影响。为此,软件测试项目管理需有相应的管理策略。

(2) 软件测试项目管理的技术手段具有先进性。软件测试项目管理采用科学、先进的管

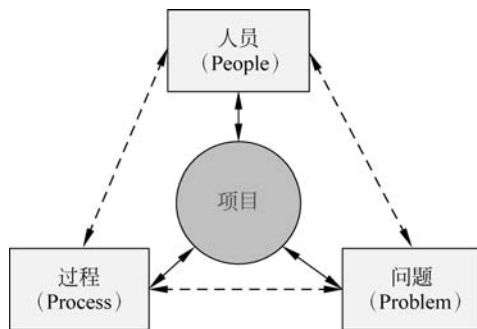


图 5-1 软件测试项目管理的因素

理理论和方法,如采用目标管理、全面质量管理、技术经济分析、先进的测试工具、测试综合跟踪数据库等进行目标和成本控制。

(3) 需要保持能使测试工作顺利进行的环境。软件测试项目管理的要点之一是创造和保持一个使测试工作顺利进行的环境,使置身于这个环境的人员能在集体中协调工作以完成预定的目标。项目组内环境、项目所处的组织环境以及整个开发流程所控制的全局环境,这3个环境要素直接关系到软件项目的可控性。

软件测试项目管理应先于任何测试活动之前开始,并且贯穿于整个测试项目的过程中。为了保证成功管理测试项目,需要坚持下列软件测试项目管理基本原则。

(1) 坚持测试计划先行。测试计划中需要清楚地描述测试目标、测试范围、测试风险、测试手段、测试环境等,需要制订测试工作的日程表。由于现实中系统需求的不断变化,导致项目进度、系统设计、程序代码和相关文档都在不停变化和修改,很容易造成测试时间被严重压缩。因此,应当为改错、再测试、变更留出足够时间。

(2) 建立客观的评价标准。做好测试工作的根本就是正确理解需求定义,建立客观的评价标准,然后验证被测软件是否符合该标准。测试人员在充分理解了软件的需求定义后,需要建立纸面上或默认的客观评价标准/规范,后续才能制订好测试策略和合理的时间表。

(3) 建立独立的测试环境。无论测试环境的大小,都需要和有关人员审查测试环境的软硬件配置,测试环境必须保证其“独立性”。在这个环境中只能做测试,不做其他任何工作。尤其不能使用开发人员的计算机进行测试工作,否则混乱的环境势必会影响最终的测试结果。

5.1.3 软件测试项目范围管理

软件测试项目范围管理就是界定测试项目所必须包含且只需要包含的全部工作,并对其他的测试项目管理工作起指导作用,以确保测试工作的顺利完成。

项目目标确定后,下一步工作就是确定需要执行哪些工作来完成该项目目标。这一过程通常有两种方法:一种是让测试小组通过开展头脑风暴,根据经验总结并集思广益,这种方法比较适合小型测试项目;另一种是对更大更复杂的项目建立一个工作分解结构(Work Breakdown Structure, WBS)和任务一览表。

工作分解结构是进行范围规划时重要的工具和技术之一。它是以项目的可交付结果为导向而对测试项目任务进行的分组,它把软件测试项目整体任务分解成较小的、易于管理和控制的若干子任务或工作单元,并由此组织和定义整个项目的工作范围;未列入工作分解结构的工作将排除在项目范围之外,不属于项目团队的工作。工作分解结构的每一个细分层次表示对项目可交付结果更细致的定义和描述。通过工作分解结构,项目团队得到完成项目的工作清单,这将为所有项目管理人员提供一个一致的基准,即使产生人员变动,也有一个可以相互理解和交流的平台。

进行工作分解是十分重要的工作,它在很大程度上决定了测试项目能否成功。具体来说有以下作用。

(1) 把复杂的事情简单化,使项目的任务执行起来更加容易。

(2) 通过 WBS 得到完成项目的任务清单,从而界定出测试项目的工作范围。

(3) 把测试项目要做的所有工作都清楚地展示出来,不至于漏掉任何重要的事情以及需要项目组完成的任务。

(4) 容易对每项分解出的活动估计所需时间、所需成本,便于制订完善的进度、成本预算等项目计划。

(5) 通过工作分解,可以确定完成测试项目所需要的技术、所需要的人力及其资源。

(6) 便于将任务落实到责任部门或个人,有利于界定职责和权限,也便于各方面就测试项目的工作进行沟通。

(7) 使测试项目团队成员更清楚地理解任务的性质及其努力的方向。

(8) 能够对测试项目进行有效的跟踪、控制和反馈。

5.2 软件测试管理计划

5.2.1 软件测试计划制订

测试计划是一个叙述了预定的测试活动的范围、途径、资源和进度安排的文档,是实现全生命周期测试管理的基础。它确认了测试项、被测特征、测试任务、人员安排以及任何偶发事件的风险,包括被测试项目的背景、目标、范围、方式、资源、人员、进度安排、测试组织以及与测试有关的风险等方面。

在制订测试计划时考虑了测试采用的模式、方法、步骤、问题和风险等方面,因此能够使软件测试工作进行更顺利。有了测试计划,测试人员之间以及测试人员与产品开发小组之间可以更好地沟通,能够减少工作中的无序和浪费,并且让软件测试工作更易于管理和控制。

制订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订测试计划工作。

- (1) 制订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。

制订测试计划时,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,制订测试计划时,使用正规化文档通常更好。可以参考 GB/T 8567—2006《计算机软件文档编制规范》,也可以参考 IEEE829 软件测试文档标准,包括 IEEE 829—2008、IEEE 829—1998、IEEE 829—1983 等版本。下面以 IEEE 829—1998 中的目录模板为例进行介绍。

- (1) 测试计划标识符。
- (2) 介绍。
- (3) 测试项。
- (4) 需要测试的功能。
- (5) 不需要测试的功能。
- (6) 方法(策略)。

- (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。

根据 IEEE 829—1998 软件测试文档编制标准的建议,测试计划包含了以上 16 个大纲要项,简要说明如下。

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。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(包含哪些阶段的测试)。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。具体要点包括功能的测试、设计的测试以及整体测试(数据流)。IEEE 标准中指出,可以参考以下文档完成测试项:需求规格说明、用户指南、操作指南、安装指南以及其他与测试项相关的事件报告。

在需要测试的功能这一部分,从用户的角度列出了待测的功能(测试项是从开发者或程序管理者的角度),而当测试落后于进度表时可以将那些具有相对低风险的功能改列在不需要测试的功能中。方法(策略)这部分内容是测试计划的核心所在,测试策略主要描述如何进行测试,以及解释测试成功与否起决定作用的所有相关问题,描述了测试小组用于测试整体和每个阶段的方法。测试项通过/失败的标准这一部分给出了“测试项”中描述的每个测试项通过/失败的标准,由通过/失败的测试用例、缺陷数量、类型、严重性和位置、可靠性或稳定性等描述。下面是通过/失败的标准的一些例子。

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

常用的测试中断标准如下。

- (1) 关键路径上的未完成任务。
- (2) 大量的缺陷。
- (3) 严重的缺陷。
- (4) 不完整的测试环境。
- (5) 资源短缺。

而在中断产生后,通过重新设计、修改错误、替代等操作可以恢复测试。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和工具,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。测试任务中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,通常与“测试人员的工作分配”一起描述。

环境需求是确定实现测试策略必备条件的过程,如人员、设备、办公室和实验室空间、软件,以及其他所需资源。测试人员的工作职责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。复杂的任务可能有多个执行者,或者由多人共同负责,可以利用表格组织测试人员的工作职责。人员安排与培训需求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,以及他们需要掌握的技能等。这个实际责任表会更加详细,确保软件的每部分都有人进行测试。

测试进度表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)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。软件测试人员要在潜在的问题和风险部分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。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,如下面的一些方面。

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

在审批部分,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。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。

要制订好软件测试计划,需要准确理解被测软件的功能特征、应用行业的知识和软件测试技术,在需要测试的内容中突出关键部分,可以列出关键及风险内容、属性、场景或测试技术。对测试过程的阶段划分、文档管理、缺陷管理、进度管理给出切实可行的方法。在完成后再需要经过测试团队、开发团队和项目管理层的复查,或者由项目经理、开发经理、测试经理、市场经理等组成的评审委员会审阅。

5.2.2 软件测试计划执行

测试计划审核通过后,需要通过测试设计阶段、测试执行阶段以及测试评估阶段完成该测试计划。

1. 测试设计

测试设计阶段是使用各种测试用例设计方法进行用例设计。在设计测试用例之前,首

先应该根据测试计划中每个需求点编写包括需求点简介、测试思路 and 详细测试方法 3 部分的测试方案,测试方案编写完成后也需要进行评审。在测试方案编写阶段,测试人员对整个系统需求有了详细的理解,这时开始编写用例才能保证用例的可执行和对需求的覆盖。测试用例包括测试用例编号、测试标题、用例级别、预置条件、测试输入、操作步骤和预期结果。其中操作步骤和预期结果需要编写详细且明确。测试用例应该覆盖测试方案,而测试方案又覆盖了测试需求点,同样,测试用例也需要评审。

在设计测试用例时,首先要对测试用例的组织结构进行划分,如通过软件的功能模块对测试用例进行分类。然后根据功能点划分,利用等价类划分、边界值分析等测试方法设计测试用例。测试用例设计是不断迭代的过程,每执行完一轮测试,都要对测试用例进行补充和整理。一是因为测试过程中会发现设计测试用例时考虑不周需要完善的地方;二是由于软件自身的新增功能以及软件版本的更新;三是在软件交付使用后如果反馈了软件缺陷,这些缺陷又是因测试用例存在漏洞造成的。

2. 测试执行

测试执行阶段是满足开始执行条件后(测试用例通过评审以及已提交可测试的系统),按照测试计划和测试用例搭建相应的软硬件测试环境,结合所需工具开始测试执行,以及对修改的 Bug 进行回归测试的过程。

在测试执行阶段需要特别注意用例的前提条件和特殊说明,如要测试某个软件的登录功能,在测试前就应该创建用户并为用户分配一定的权限。在过程中需要确保所有的测试用例都被执行,对于在执行用例时偶然出现的错误不要忽视,应该多测试几次,尽可能准确地找出问题的原因。当测试执行结果出现预期与实际不一致的情况,要从多个角度多测试几次,尽量详细地定位软件出错的位置和原因,并测试这个错误会不会导致更严重的错误,最后把详细的输入和实际的输出以及对问题的详细描述记录下来。当发现比较复杂、难定位的问题,应当在测试用例执行后记录相关的软件运行日志、用户操作日志等文件,作为测试过程记录,便于开发人员快速定位问题。对于发现的正确、可复现、证据充分的缺陷,测试人员应精简、中立并且准确地编写缺陷报告并提交缺陷管理平台,以便开发人员进行修复,并对这些缺陷不断地进行测试和跟踪。

3. 测试评估

测试评估阶段是对整个测试的过程和版本质量做一个详细的评估。所有的测试用例都至少被执行一遍并且存在的问题已得到合理的处理后,需要对测试过程和测试结果进行分析和总结,确认测试计划是否得到完整履行,确认测试覆盖率、缺陷率以及其他各项指标是否达到预定的质量标准要求,并最终在报告中给出测试和产品质量的评估结论。

5.3 软件测试文档

测试文档是对要执行的软件测试及测试的结果进行描述、定义、规定和报告的任何书面或图示信息。由于软件测试是一个非常复杂却对保证软件的质量和软件的正常运行有重要意义的工作,因此必须把对软件测试的要求、规划、测试过程等有关信息和测试的结果以及对测试结果的分析、评价等内容以正式的文档形式给出。这些测试文档应该在软件开发初

期的需求分析阶段就开始着手准备,在编制过程中最好能够让用户参与,这将有利于他们对开发的应用系统有更好的理解。设计阶段的一些设计方案也应在测试文档中得到体现,以利于设计的检验。测试文档对于测试阶段工作的组织、规划和管理具有重要作用,能够很好地支持各项测试工作顺利开展。除此之外,在已开发的软件投入运行的维护阶段,通常还要进行再测试或回归测试,这时还会用到测试文档。测试文档的编写是测试管理的一个重要组成部分。

5.3.1 软件测试文档的作用

测试文档主要有以下几方面的作用。

1. 有利于管理测试项目

测试文档作为质量标准化的一项例行基本工作,可为项目管理者提供项目计划、预算、进度等各方面的信息,为组织、规划和管理测试项目提供结构。

2. 便于项目组成员交流沟通

测试文档是测试人员之间以及测试人员与产品开发小组之间相互沟通的基础和依据,基于此,人员之间的交流可以事半功倍。一个软件的开发过程需要大量人员共同参与,若没有将一些标准认定工作文档化,测试人员很难对软件整体(包括系统功能、问题原因和解决方法等)完全了解。项目组成员之间的交流沟通以文档的形式进行交接是方便、省时、省力的方式。

3. 印证测试的有效性

测试执行完成后,将测试结果记录于文档中,这能够对分析测试的有效性,甚至整个软件的可用性提供必要的依据。通过这个文档化的测试结果,可以分析软件系统是否达到预定的质量标准要求。

4. 为测试资源的检验提供标准

测试文档不仅用文档的形式把测试过程中涉及的相关描述、定义以及要完成的任务规定下来,还列出了测试工作不可或缺的资源。对照测试文档可以检查这些资源能否得到,如果测试计划即将实施,但所需资源尚未全部落实,那就必须及早解决。

5. 方便后期再测试

由于现在经常采用迭代的方法进行软件开发,每次迭代都包括了定义、需求分析、设计、实现与测试,前期测试过程与结果的记录文档对后期进行重复测试意义重大;并且测试文档中的部分内容在后期的维护阶段往往由于各种原因需要进行修改完善,修改后的内容需要进行重新测试。因此,测试文档在整个过程中对于管理测试和复用测试非常重要。

6. 为测试工作的总结和评价提供依据

软件测试的最终目的是保证软件产品的质量可信,在软件开发的过程中,需要对软件产品进行质量控制。对测试数据进行翔实记录,并根据测试情况撰写测试报告,有助于测试人员对测试工作进行总结,并识别出软件的局限性和失效性。测试完成后,将测试结果与预期结果进行比较,便可对测试软件提出评价意见。

7. 更好地防范项目风险

在测试文档中列出测试任务可能的风险有助于测试小组对潜在的、可能出现的问题事先做好思想上和技术上的准备。

测试文档可以完整地记录测试的全部过程和测试的结果,文档是测试过程中必不可少的组成部分,测试文档的编写也是测试工作规范化的重要组成部分,应该按照软件系统文档标准编写和使用测试文档,为高质量地开展各项测试工作提供便利。

5.3.2 主要的软件测试文档

软件测试文档标准是保证文档质量的基础,根据一定的标准编写文档可以使测试工作更加流程化、规范化,让测试工作能够更好地开展。这里依然参照 IEEE 829—1998 标准,给出所有软件测试文档的目录模板,在实际工作中可以根据实际情况对模板进行部分增加、删除和修改。

IEEE 829—1998 标准中给出了 8 个软件测试文档的目录模板,分别是测试计划、测试设计规格说明书、测试用例规格说明书、测试过程规格说明书、测试项记录报告、测试日志、测试缺陷报告和测试总结报告。其中测试计划的目录已经在 5.2.1 节中介绍过,下面分别介绍其余 7 个文档目录模板。

1. 测试设计规格说明书

测试设计规格说明书细化了测试计划中的测试方法,并且给出了需要通过这些方法测试的功能点范围。

- (1) 测试设计规格说明书标识符。
- (2) 待测试功能点。
- (3) 测试方法细化。
- (4) 测试标识符。
- (5) 功能通过/失败标准。

2. 测试用例规格说明书

测试用例规格说明书运用测试设计规格说明书定义了测试用例,描述了测试用例的标准。

- (1) 测试用例规格说明书标识符。
- (2) 测试项。
- (3) 输入规格说明。
- (4) 输出规格说明。
- (5) 环境要求。
- (6) 过程中的特殊需求。
- (7) 依赖关系。

3. 测试过程规格说明书

测试过程规格说明书指定了执行一组测试用例的步骤,或者说,用于分析软件项目以评估一组功能的步骤。

- (1) 测试过程规格说明书标识符。
- (2) 目的。
- (3) 特殊需求。
- (4) 测试过程步骤,包括以下几个步骤。

- 记录
- 准备
- 开始
- 进行
- 度量
- 中止
- 重新开始
- 停止
- 描述恢复环境所需的活动
- 应急措施

4. 测试项记录报告

测试项记录报告用于记录测试项在测试过程中的传递过程,它包括了测试项的负责人、物理位置以及状态,任何对测试项现有要求和设计上的修改都需要在这个报告中注明。

- (1) 测试项记录报告标识符。
- (2) 所记录的测试项。
- (3) 物理位置。
- (4) 状态。
- (5) 审批。

5. 测试日志

测试日志记录了测试的执行情况,提供了关于测试执行详细的时序信息。

- (1) 测试日志标识符。
- (2) 描述。
- (3) 活动和事件信息。
 - 执行情况描述
 - 执行结果
 - 环境信息
 - 异常事件
 - 缺陷报告标识符

6. 测试缺陷报告

测试缺陷报告用来描述那些出现在测试过程中并且需要调查原因的异常情况,这些异常情况可能存在于需求、设计、代码、文档或测试用例中。

- (1) 测试缺陷报告标识符。
- (2) 缺陷总结。
- (3) 缺陷描述。

- 输入
- 预期结果
- 实际结果
- 异常情况
- 日期和时间
- 测试步骤
- 测试环境
- 再现测试
- 测试人员
- 观察人员

(4) 影响。

7. 测试总结报告

测试总结报告总结了测试项目的完成结果,并且基于这些结果给出了关于测试的评价。

(1) 测试总结报告标识符。

(2) 总结。

(3) 差异。

(4) 综合评估。

(5) 结果总结。

(6) 评价。

(7) 活动总结。

(8) 审批。

5.4 测试组织和人员管理

测试人员能力与素质的高低以及能否将他们有效地组织起来是测试项目能否顺利完成的关键因素之一。测试组织和人员管理是测试项目管理职能中最难的部分,会直接影响测试工作的效率和软件的产品质量。测试组织和人员管理是对测试项目相关人员在组织形式、人员组成以及职责方面所做的规划和安排。测试组织和人员管理的任务包括:选择适合测试项目的组织结构模式;确定项目组人员的组织形式;合理安排人员,明确各自分工和责任;有效管理项目成员,充分发挥大家的主观能动性,密切配合。在测试组织和人员管理过程中应做到尽快落实责任,减少沟通成本以及明确责任,以便高效地实现项目目标。

5.4.1 测试人员和组织结构

高素质的测试人员是测试项目成功不可或缺的要素,什么样的人员才是高素质的测试人员呢?下面列出一些测试人员应该具备的能力。

(1) 一般性能力:包括沟通表达能力、创新能力、自我督促不断学习的能力、质量意识、责任意识以及对软件工程中计算机网络、操作系统、数据库、编程技能等专业知识的掌握等。

(2) 测试的专业技能:包括测试的基本概念和整体流程、测试策略、测试方法、测试工

具、测试标准等。

(3) 测试设计规划能力：包括业务需求分析、测试目标及规划、测试计划和设计的评审方法、风险分析及防范等。

(4) 测试执行能力：包括自动化测试工具、测试数据/脚本/用例、缺陷记录和处理、测试结果分析和比较等。

(5) 测试分析改进能力：包括测试报告撰写、统计分析、测试度量、过程监测分析和持续改进等。

而作为测试工作的组织管理者,除了需要对测试工作有充分的了解,还需要具备以下能力。

(1) 了解软件测试相关政策、标准、过程、工具、度量。

(2) 领导一个独立自主的测试组织的能力,包括领导力、沟通力、控制力等。

(3) 对优秀人才的吸引和培养能力。

(4) 快速提出和执行有效解决问题方案的能力。

(5) 对测试时间、质量和成本的把控能力。

而如何将测试人员通过一定的模式对责任和关系进行分配安排,使大家能够通过这种组织结构充分发挥自己的能力与作用,需要在人员组织结构设计时充分考虑以下因素。

(1) 集中还是分散：测试人员可以集中管理,也可以分散于各个业务组。分散于业务组有利于了解业务需求;集中管理有利于保持测试的独立性。

(2) 功能还是项目：测试组织可以面向功能,也可以面向项目。

(3) 垂直还是扁平化：垂直的组织结构是从上级管理者到下面的测试人员之间呈金字塔状,下级人员只接受一个上级的指令,各级主管负责人对下属的一切问题负责,其优点是结构比较简单,责任分明,命令统一;扁平化的方式减少了管理层次,在测试工作中效率较高。

测试的组织结构形式有很多种,并没有正确或错误的分别,根据企业文化、管理水平、成员的能力水平以及软件产品的不同可以选择合适的测试组织结构形式。目前常见的测试组织结构包括独立的测试小组以及与开发同属一个部门这两种形式。

1. 独立的测试小组

成立独立的测试小组方便接收各项目的测试平台或测试框架需求,部署统一的开发并提供统一测试服务,为各项目的测试工作提供统一的规范及指导,方便协调测试资源,包括人力、硬件、环境等。将测试人员和研发人员分开,这有利于保持测试人员的独立性,有助于让测试流程和规范很好地得到执行。此外,测试组长与开发组长保持同级平等关系,测试人员和研发人员分别向不同的领导汇报可以赋予测试人员更多的话语权,使测试策略和测试计划可以顺利执行。但这种组织架构也存在一些缺点,如测试人员对于产品业务知识的理解很难深入,测试和研发之间有距离感以及测试人员对公司文化的认可度可能会受到影响。

2. 与开发同属一个部门

测试与开发同属一个部门的组织形式是当测试团队的成熟度到达一定程度后,开发和测试同属一个团队,现在,这种组织结构越来越多地被一些软件公司采用。这种组织形式的优劣势恰恰与第1种方式相反,测试人员与软件开发人员并肩作战,一起工作,可

以减少软件开发人员与测试人员合作时的不利因素,会极大地方便交流和沟通。但同时测试人员的话语权有可能受到研发人员的压制,所以当开发和测试人员融合后,这两个部门公共的领导对于质量和进度的权衡和把握尤为重要,将影响测试人员在公司的地位和话语权。

5.4.2 测试人员的沟通和激励

良好的沟通是保证测试项目顺利进行的基础,一个项目小组成员之间及时又高效的沟通形式通常是以下多种交流方式的有机组合。

- (1) 正式非个人方式:如正式会议等。
- (2) 正式个人之间交流:如成员之间的正式讨论等。
- (3) 非正式个人之间交流:如个人之间的自由交流等。
- (4) 电子通信方式:如 E-mail、QQ、微信、钉钉等。

激励机制就是通过特定的方法和管理体系,将员工对组织和工作的承诺最大化的过程。激励机制在测试组织过程中非常重要,有效的激励机制才能调动测试人员的工作积极性,将潜力充分体现出来。激励机制是多元化的,可以从目标激励、示范激励、尊重激励、参与激励、荣誉激励、关心激励、竞争激励、物质激励、信息激励、文化激励以及自我激励等多方面加以考虑。测试人员管理中激励机制的关键点如下。

- (1) 激励员工从结果均等转移到机会均等,并努力创造公平竞争环境。
- (2) 激励要把握最佳时机,如需要在目标任务下达前激励的要提前激励,在员工遇到困难时应及时激励。
- (3) 激励要有足够的力度,并且在工作获得认可后尽快兑现,而对于已经满足的需要很可能不再成为激励因素。
- (4) 激励要公平准确、奖罚分明,需要健全、完善的绩效考核制度,克服有亲有疏的人情风。
- (5) 物质奖励与精神奖励相结合,奖励与惩罚相结合,注意采取卓有成效的非货币形式的激励措施。
- (6) 过多行使权力、资金或处罚手段很可能导致项目失败。

每个员工的思想、性格、学识、教养、道德水准不同,在建立激励机制时充分考虑员工的个体差异,实行差别激励的原则,以人为本,真正做到关心人、尊重人,并建立企业与员工的全方位的激励沟通机制。作为测试人员,也应遵循下列测试工作的 7 项效率原则。

- (1) 主动思考,积极行动,尽早参与项目,做好前期准备,“有备”才能“无患”。
- (2) 一开始就牢记目标,不迷失方向,要牢记什么时间点完成某个测试项目。
- (3) 重要的事情放在首位(但常常将紧急的事情放在首位),学会时间管理。
- (4) 先理解人,后被人理解,测试是发现缺陷,让产品更完美,而不是故意找茬。
- (5) 寻求双赢,积极配合开发人员工作。例如,帮助他发现问题规律,努力赢得开发人员支持;哪些地方可能会有问题,需要加强测试。
- (6) 互相合作,追求 $1+1>2$,测试团队人员密切配合,促进测试整体进度。
- (7) 不断学习,自我更新,不断进步。

5.4.3 测试人员的培训

从测试管理的角度来看,为了高效地完成测试工作制定的目标,除了前面提到的人员素质、组织结构、沟通形式、激励机制,还需要通过培训不断地帮助测试人员进行知识的更新和技术能力的提升。

1. 制订测试人员培训计划

测试人员培训计划是测试计划中的一个重要组成部分,培训计划必须满足测试项目和员工两方面的需求,兼顾项目组织资源条件和员工素质基础,并充分考虑培训的超前性和培训结果的不确定性。

(1) 尽可能多地得到公司高级管理层和各部门主管的承诺以及足够的资源支持各项具体培训计划,尤其是学员培训时间上的承诺。

(2) 培训计划制订前必须进行充分的培训需求调查和分析。

(3) 尽量将培训活动安排在测试任务开始之前。

(4) 培训计划必须首先从实用角度出发,“好看”更要“有用”。

(5) 争取更广泛的培训参加人员,更多的人参与,将获得更多的支持。

(6) 在计划制订过程中,应考虑设计不同的学习方式,以适应员工的需要和个体差异。

(7) 要采取一些提高培训效率的积极性的措施。

(8) 注重培训细节。

(9) 注重培训内容的实效性。

(10) 鼓励合作学习,团队共同参与。

(11) 对培训效果要及时评价,对发现的不足进行反思和改进。

2. 软件测试培训内容

软件测试培训主要包括以下内容。

(1) 测试基础知识和技能培训。

(2) 测试设计规划和测试工具培训。

(3) 所测试的软件产品培训。

(4) 测试过程、测试管理、测试环境等培训。

5.5 软件测试过程管理

成功的软件测试项目除了采用先进的标准、方法、工具和高素质的测试人员,还需要对测试过程进行管理和控制。软件的测试工作与软件开发一样,是软件工程项目的重要组成部分。测试的过程管理和控制是软件项目成功的重要保证,是保证测试过程质量、控制和减少测试风险的重要手段。

5.5.1 测试项目的过程管理

软件测试活动与项目同时展开,项目一旦启动,测试工作也就相应地开始了。在软件开

发周期的每个阶段都有相应的测试阶段。在需求分析阶段,在需求分析、产品功能设计的同时,测试人员就可以阅读、审查需求分析的结果,创建测试的准则。在系统设计阶段,测试人员能够了解系统是基于什么平台、是如何实现的,这样可以更好地设计系统的测试方案和测试计划,并事先准备系统的测试环境。在详细设计阶段,测试人员可以参与设计,对设计进行评审,同时设计功能测试用例,完善测试计划。在编码的同时可以进行单元测试,从而尽快找出程序中的错误,充分提高程序质量,减少成本。

软件测试项目的过程管理主要集中在测试项目启动、测试计划、测试设计、测试执行、测试结果分析和测试过程管理工具的使用上。

1. 测试项目启动阶段

测试项目的启动,首先需要确定测试项目组长,确定了组长才可以组建整个测试小组,从而可以和开发小组一起开展工作,共同参与项目分析、设计等相关会议,获得必要的项目需求、设计文档以及相关技术知识的培训等。

2. 测试计划阶段

测试计划涵盖了测试活动的范围、途径、资源和进度安排等信息,不可能一蹴而就,而是要经过长期的起草、讨论、编制和审查等阶段才能将测试计划制订好。软件测试项目过程管理的基础就是软件测试计划。软件测试计划中描述了如何实施和管理软件的测试过程,在批准生效后,将被用来作为对测试过程跟踪分析的依据。通过选定测试的某个时刻,将实际测试的工作量、投入、进度、风险等与计划进行对比,若计划未完成,则可以采用相应的纠正措施,如调整测试计划、重新安排测试进度以及提高测试效率等方法。

3. 测试设计阶段

软件测试设计中,要充分考虑所设计的测试技术方案是否可行、所设计的测试用例是否完善、所设计的测试环境是否接近于用户的真实环境等问题。测试设计阶段的关键是将开发设计人员已经掌握的产品相关知识传递给测试人员,同时也要做好开发设计人员对测试用例的审查工作。

4. 测试执行阶段

测试执行阶段需要建立相关的测试环境,准备测试数据,执行测试用例,并对发现的缺陷进行分析和跟踪,这一阶段是测试的基础,需要做好缺陷的报告和管理,这些工作直接关系到测试的可靠性和准确性,对软件产品的质量产生重大影响。

5. 测试结果分析阶段

测试结果分析阶段是对测试结果进行综合分析,以确定软件产品的质量状态,为产品的改进和发布提供依据。从管理上讲,要做好测试结果的审查和分析会议,做好测试报告的编写和审查。具体包括以下内容。

(1) 审查测试整体过程:对软件测试项目全过程、全方位地进行审查,检查测试计划、测试用例是否得到执行,检查测试过程中是否有疏漏。

(2) 对当前状态的审查:检查现阶段是否还存在尚未解决的问题,对产品存在的缺陷进行逐一分析,了解每个缺陷对产品质量影响的程度。

(3) 总结评估:对审查后的当前状态进行评估,如果产品质量、测试标准都已经达到要

求,则对项目进行中的问题分析总结,获取项目成功经验。

6. 测试过程管理工具的使用

在整个软件测试项目的过程管理中,可以采用周报、日报、例会、评审会等多种形式的管理工具了解测试项目的进展状况,对项目进行跟踪和监控。通过这些方式建立、收集和分析项目的可靠实际状态数据,可以让管理者更好地做出明智的决策,从而达到项目有效管理的目的。

在整个测试项目的各个阶段都需要项目管理人员充分认识项目的状态,监控项目的发展趋势,发现潜在的问题,从而更好地控制成本,降低风险,提高测试工作质量。

5.5.2 软件测试的配置管理

软件测试的配置管理是在团队开发中标识、控制和管理软件测试变更的一种管理,能够记录演化过程,确保软件测试人员在软件测试生命周期中各个阶段都能得到精确的产品配置。配置管理随着软件系统的日益复杂化逐渐成为软件生命周期中的重要控制过程,可以建立和维护起软件产品的完整性、一致性和可追溯性。软件测试过程的配置管理和软件开发过程的配置管理一样,包括以下 6 个基本活动。

1. 配置管理的准备

在配置管理具体实施之前,需要配置管理员制订配置管理计划以及创建项目的配置管理环境。其中,配置管理计划的主要内容包括配置管理软硬件资源、配置项计划、基线计划、交付计划、备份计划等,该计划需要测试主管领导或项目变更控制委员会审核批准。

2. 配置项的标识

配置项的标识主要是标识测试样品、测试标准、测试工具、测试相关文档、测试报告等配置项的名称和类型。配置项的标识是配置管理的基础,通过将所有配置项都按照模板统一生成、统一编号,并在文档中的规定章节记录对象的标识信息(包括各配置项的所有者、基准化时间以及存储位置),从而让测试人员更方便地知道每个配置项的内容和状态。

3. 版本控制

版本控制是配置管理的核心功能。在整个测试项目过程中,绝大部分的配置项都要经过多次的修改才能最终确定下来。对配置项的任何修改都将产生新的版本。由于不能保证新版本一定比老版本“好”,所以不能抛弃老版本。版本控制的目的是按照一定的规则保存配置项的所有版本,避免发生版本丢失或混淆等现象,并且可以快速、准确地查找到配置项的任何版本。配置项的状态有草稿、正式发布和正在修改 3 种,根据制订的配置项状态变迁与版本号规则对配置项状态和版本号进行更改。

4. 变更控制

在整个测试项目过程中,配置项发生变更几乎是不可避免的。变更控制的目的并不是控制和限制变更的发生,而是对变更进行有效的管理,防止配置项被随意修改而导致混乱。变更的起源包括功能变更和缺陷修补,其中功能变更是为了增加或删除某些功能,缺陷修补是对存在的缺陷进行修补。修改处于草稿状态的配置项不算变更,无须批准,修改者按照

版本控制规则执行即可。当配置项的状态成为正式发布,或被冻结后,此时任何人都不能随意修改,必须依据“申请→审批→执行变更→再评审→结束”的规则执行。

5. 配置报告

配置报告是根据配置库中的数据操作记录,向管理者汇报测试工作进展情况。配置报告一般是定期编写的,根据数据库中的客观数据真实反映各配置项的情况,尤其是当前基线配置项的状态,以作为测试进度报告的参考。基线在 IEEE 中被定义为“已经正式通过审核标准的某规约或产品,它因此可以作为进一步开发的基础,并且只能通过正式的变化控制过程来改变”。配置报告应该包括以下主要内容。

- (1) 定义配置报告形式、内容和提交方式。
- (2) 确认测试过程记录和跟踪问题报告、配置项更改请求、配置项更改次序等。
- (3) 确定测试报告提交的时间与方式。

6. 配置审计

配置审计是为了保证所有人员(包括项目成员、配置管理员和项目变更控制委员会)都遵守配置管理规范而定期执行的过程质量检查活动,是质量保证人员的工作职责之一。通过配置审计确保所有配置管理规范已被切实地执行和实施,配置审计主要包括以下内容。

- (1) 确定审计执行人员和执行时间。
- (2) 确定审计的范围、内容和方式。
- (3) 确定发现问题的处理方法。

配置管理是管理和调整变更的关键,尤其是对于参与人员较多、变更较多的项目。软件测试项目配置管理能够跟踪每个变更的创造者、时间和原因,为测试项目管理提供了多种跟踪测试项目进展的角度,为更好地了解测试项目进度提供了保证。良好的配置管理能使软件测试过程有更好的可预测性、可重复性,使用户和主管部门对软件质量有更强的信心。

5.5.3 软件测试的风险管理

软件测试风险指的是软件测试过程中出现的或潜在的问题,这些问题会给软件测试工作带来损失。风险产生的原因主要是测试计划的不充分、测试方法或测试过程的偏离等。软件测试风险管理主要是对测试计划执行的风险进行分析与评估,以及制定应急措施,防止和降低软件测试产生的风险造成的危害。在软件测试过程中常见的风险主要有以下 7 类。

1. 测试时间进度风险

用户需求有可能发生重大变更,开发进度可能发生延误,这些原因都可能导致测试时间的大幅调整压缩,测试人员、测试环境、测试资源的不能准时到位以及可能存在的难以修复的缺陷也会对测试进度造成影响。

2. 测试质量目标风险

测试的质量目标不清晰,如易用性测试、用户文档的测试目标存在见仁见智的问题。

3. 测试范围认知风险

对产品质量需求或产品特性理解不准确,造成测试范围分析误差,出现测试盲区或验证

标准错误。

4. 测试人员风险

测试开始后,测试人员、技术支持人员可能出现被紧急项目调用,无法按照计划安排参与项目的情况。

5. 测试充分性风险

在设计测试用例时,部分测试用例可能会忽视边界条件和深层次的逻辑关系;在执行测试用例时,部分测试用例可能会被测试人员有意无意地忽略执行,这都将导致测试的不充分。

6. 测试环境风险

测试环境无法与生产环境完全一致,致使性能测试的结果存在误差。

7. 测试工具风险

能否及时准备相关测试工具,测试人员对新工具无法熟练运用等情况也时有发生。

对风险的评估主要依据风险描述、风险概率和风险影响3个因素,从成本、进度以及性能3个方面进行评估。首先,需要进行风险识别,可以测试组成员头脑风暴、找资深专家进行访谈以及通过风险项目检查表,按风险内容进行分项检查,逐项检查风险。然后,对识别出来的风险进行分析,主要从风险概率分析(如很高、较高、中等、较低、很低等)和风险带来的后果入手,从风险的表现、范围、时间确定风险评估的正确性,根据损失(影响)和风险概率的乘积,确定风险的优先级别,从而有针对性地制定风险应对措施。

基于风险评估的结果还需要进行风险的控制,对于那些可以避免的风险,要采取措施来避免;对于那些可能带来非常严重后果的风险,看能否通过一些方法将它转移为不会引起严重后果的低风险;对于无法避免的风险,尽量采取措施降低风险。为了避免、转移以及降低风险,需要提前做好风险管理计划以及应急处理方案。除了这些工作,还可以依据以下策略进行风险控制。

(1) 在做计划时,对资源、时间、预算等的估算要留有余地,避免风险发生时没有相应的资源及时支持应急方案。

(2) 在项目开始前,把测试环境、测试工具等有变化和难以控制的因素列入风险管理计划中。

(3) 通过培训提高测试人员的综合素质,培养后备人员,做好人员流动的准备。

(4) 对所有工作多进行相互审查,及时发现问题。

(5) 对所有过程做好日常跟踪,并进行完善的文档管理。

风险管理的重点不在于分析产生的原因,而是根据风险评估的结果进行风险控制和提前制定应急措施以应对风险的发生。当测试计划风险发生时,可以采用缩小范围、增加资源、减少过程以及延长时间等措施。测试风险管理的目标在于提高测试积极事件的概率和影响,降低测试消极事件的概率和影响。对于已识别出的风险,分析其发生概率和影响程度,并进行优先级排序,优先处理高概率和高影响风险。软件测试始终存在着风险,如果提前重视风险,并且有所防范,就可以最大限度地减少风险的发生以及减小风险的影响。

5.5.4 软件测试的成本管理

软件测试项目的成本管理就是根据企业的情况和软件测试项目的具体要求,利用公司既定的资源,在保证软件测试项目的进度、质量达到客户满意的情况下,对软件测试项目成本进行有效的组织、实施、控制、跟踪、分析和考核等一系列管理活动,最大限度地降低软件测试项目成本,提高项目利润。成本管理的过程主要包括以下4方面。

1. 资源计划

资源计划包括决定为实施这一软件测试项目需要使用什么资源(包括人力资源、设备和物资等)以及每种资源的用量。这里的主要输出是资源需求的清单。

2. 成本估算

成本估算包括估计完成软件测试项目所需要的资源成本的近似值。这里的主要输出是成本管理计划。

3. 成本预算

成本预算包括将整个成本估算配置到各单项工作,以建立一个衡量绩效的基准计划。这里的主要输出是成本基准计划。

4. 成本控制

成本控制包括控制测试项目预算的变化,这里的主要输出是修正的成本估算、更新预算、纠正成本使用方式以及取得的教训。

在实际的软件测试中,时间、人力、资金等各种资源都是有限的,想要找出所有的缺陷是不可能的。“太少的测试是犯罪,而太多的测试是浪费”,测试工作的目标是使测试产能最大化,也就是要使通过测试找出错误的能力最大化,而检测次数最小化。测试的成本控制目标是使测试开发成本、测试实施成本和测试维护成本最小化。在软件产品测试过程中,测试实施成本主要包括测试准备成本、测试执行成本和测试结束成本。测试是一种带有风险性的管理活动,可以使企业减少因为软件产品质量低劣而花费不必要的成本。

质量成本要素主要包括一致性成本和非一致性成本。一致性成本是指用于保证软件质量的支出,包括预防成本和测试预算,如测试计划、测试开发、测试实施费用。非一致性成本是由出现的软件错误和测试过程故障引起的,这些问题会导致返工、补测、延迟。追加测试时间和资金就是一种由于内部故障引起的非一致性成本,非一致性成本还包括外部故障(软件遗留错误影响客户)引起部分。

缺陷探测率是一个衡量测试工作效率的软件质量成本的指标。 $\text{缺陷探测率} = \frac{\text{测试发现的软件缺陷数}}{(\text{测试发现的软件缺陷数} + \text{客户发现并反馈给技术支持人员进行修复的软件缺陷数})}$ 。缺陷探测率越高,就意味着测试发现的缺陷多,发布后用户发现的缺陷少,达到了节约总成本的目的,可以获得更高的测试投资回报率($\text{投资回报率} = \frac{\text{利润}}{\text{测试投资}} \times 100\%$)。

测试项目的开展不可避免地会遇到各种不确定的事件,测试项目的管理者都要在存在不确定因素的环境下管理项目,通过采取一些措施和办法可以帮助测试项目的管理者更好地进行项目成本管理,实现测试项目生命周期内的成本度量和控制。以下是软件测试项目成本的控制原则。

- (1) 坚持成本最低化原则。
- (2) 坚持全面成本控制原则。
- (3) 坚持动态控制原则。
- (4) 坚持项目目标管理原则。
- (5) 坚持责任、权力以及奖惩相结合的原则。

在软件测试项目成本的控制过程中,软件测试项目负责人是项目成本管理的第一责任人,全面组织软件测试项目的成本管理工作,应该及时掌握项目盈亏状况,并迅速采取有效措施;技术人员应尽可能采取先进技术、先进工艺,以降低项目成本;财务人员应及时分析财务收支情况,合理安排资金,对于人工费、材料费、工具费和间接费等其他费用及时分析,加强控制。

软件测试项目成本管理的目的就是确保在批准的预算范围内完成测试项目所需的各项过程。在软件实际测试过程中,应当有针对性地加强软件测试项目的成本管理,进一步节约成本,提高经济效益。

5.6 本章小结

本章主要介绍了软件测试项目的基本特性、软件测试项目管理的特性和原则,以及如何制订和实施软件测试管理计划和主要的软件测试文档。然后从测试人员管理、测试过程管理、测试配置管理、测试风险管理和测试成本管理等多角度进行了测试组织与管理的介绍。

软件测试项目具有独特性、不确定性、智力/劳动密集性、困难性以及目标冲突性,因此,优秀的测试人员和科学的管理是测试项目成功的关键。

测试人员应该具备沟通表达、创新、不断学习、质量意识、责任意识以及对软件工程专业知识的掌握等一般性能力,同时还需要具有一定的测试专业技能、测试设计规划能力、测试执行能力以及测试分析改进能力。

将软件测试项目的过程管理、配置管理、风险管理、成本管理与人员管理有效地结合起来,能够实现软件测试的可控管理、高效组织。



拓展练习

习题 5

- (1) 主要的软件测试文档有哪些?
- (2) 测试人员应该具备哪些能力?
- (3) 软件测试配置管理包括哪些基本活动?
- (4) 软件测试过程中常见的风险分为哪几类?
- (5) 质量成本要素包括哪些?