

# 软件测试项目的追踪与控制方法

于 蕾, 晏海华, 赫建营

(北京航空航天大学软件工程研究所, 北京 100083)

**摘要:** 如何有效追踪和控制软件测试项目是一个亟待解决的问题。该文分析了项目追踪与控制在测试过程管理领域的实施方法, 探讨了测试项目中计划、追踪与控制的定义以及三者之间的关系, 并结合已经实现的软件平台分析了测试项目追踪与控制的实施策略, 包括测试项目计划的实施细则、测试项目追踪过程中的信息采集状况以及测试项目的具体控制方法。

**关键词:** 软件测试; 测试项目追踪; 测试项目控制

## Tracking and Control of Software Testing Project

YU Lei, YAN Hai-hua, HE Jian-ying

(Software Engineering Institute, Beihang University, Beijing 100083)

**【Abstract】** How to track and control software testing projects is urgent. This paper analyses the method of tracking and control in software testing process. It studies the relationship among testing plan, tracking and control, then presents the method of tracking and control based on the platform which is implemented, including the details of the testing plan, the status of the information gathered in testing process and the ways of project control.

**【Key words】** software testing; testing project tracking; testing project control

随着CMM、RUP等软件过程相关理论和模型的提出, 迭代增量式的软件开发方法已被广泛接受, “测试自需求始”也被国外著名的软件企业和组织采纳。然而, 如何保证软件测试的有效进行、如何对整个测试项目进行控制, 仍是一个亟待解决和深入研究的课题。实践表明, 测试项目的有效追踪和控制要有经验丰富的项目管理人员及先进的技术, 关键是采用工程化、规范化的方法对软件测试过程中涉及的资源、费用、效率、工作量、文档、产品、过程的质量及流程、活动的执行状态等诸多情况进行全面的控制和管理, 还要有相应的测试项目追踪与控制平台支持。基于这种需求, 本文从工程实践中总结出一套实用的测试项目追踪与控制方法。

### 1 测试项目中计划、追踪与控制的关系

**定义1** 测试项目计划: 描述软件测试的范围、需要测试的功能、测试执行的方法、测试所需资源和进行测试的计划时间表等<sup>[1]</sup>。测试计划通常来自测试合同或软件需求规格说明书。测试人员根据测试合同条款或软件需求规格说明书制定一份详细的项目测试计划, 确定测试范围、质量标准、测试类型、测试进度、测试人员、测试成本等几个方面的内容。

**定义2** 测试项目追踪: 追踪测试计划在实施过程中的各种参数, 如进度、工作量、人工消耗、变更信息, 获得项目的实际进展情况并与测试项目计划进行比较, 记录当前项目状态<sup>[2]</sup>。

**定义3** 测试项目控制: 为了保证测试项目按照测试计划中规定的预期目标进行, 对测试项目的运行情况进行控制, 收集分析各种项目的进展信息, 并与预期的项目目标进行比较。在出现偏差时及时分析其原因, 制定有效的纠正预防措施并落实<sup>[3]</sup>。

由追踪与控制的内容可以看出, 测试计划是测试项目追踪的基础, 而测试项目追踪结果又是测试项目控制的依据。

项目的计划、追踪与控制是可记录的、相互依赖的, 三者的关系如图1所示。一个项目的测试过程就是一个获取测试需求、制定测试计划, 并依据测试计划对测试实施过程(通常包括测试设计及执行过程)进行追踪和控制的过程, 在实际工作中, 它是一个计划、追踪、控制多次迭代的过程。

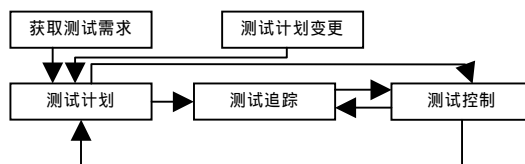


图1 计划、追踪与控制的关系

### 2 测试项目追踪与控制实施方法

#### 2.1 测试项目计划

若想对测试项目的测试过程进行追踪与控制, 首先需要制定测试计划。为了便于对计划进行追踪和控制, 制定测试计划时, 需要考虑以下几点:

- (1) 确定待测试软件的测试范围和测试任务。设定测试任务、测试需求, 细化测试项。
- (2) 明确测试项目的质量目标, 即完成软件测试工作后软件必须达到的性能要求、技术指标、质量要求等。
- (3) 指定测试类型。可按质量特性分类或按传统分类<sup>[4]</sup>。
- (4) 指派测试人员及估算测试人工成本。根据软件的代码规模和测试任务等估算出测试工作量, 根据测试工作量和工

**基金项目:** 国家自然科学基金资助项目“基于统计学的软件使用测试及其分析方法研究”(60373016)

**作者简介:** 于蕾(1980-), 女, 硕士研究生, 主研方向: 软件测试; 晏海华, 副教授; 赫建营, 博士研究生

**收稿日期:** 2007-01-08 **E-mail:** yu@cse.buaa.edu.cn

期确定实际的测试人员。

(5)制定进度计划表。这是测试计划中最重要的一个部分。把工作量分配到各个测试任务,并确定每个测试任务的起、止日期及相应测试人员,从而将估算的工作量分布于计划好的项目持续期内。测试任务的状态将随着时间的流逝不断变化。

测试计划中的测试类型、进度和成本是围绕测试范围和质量标准展开的,因此,在实施测试项目前,最重要的工作是确定测试范围和质量标准、明确测试目标。

## 2.2 测试项目追踪

制定测试计划后就可以依据测试计划实施测试工作。在测试实施过程中需要持续追踪测试项目的状况,并与测试计划比较,发现偏差,分析其原因及趋势,及时采取纠正、预防措施,随时解决测试过程中出现的问题,以确保测试工作顺利地完成。

测试项目追踪主要采集测试项目进展状况、测试规模信息、软件问题统计信息、测试变更信息以及人工消耗信息。

### (1)测试项目进展状况

掌握测试项目的进展状况是保证测试工作持续有效开展的一个关键因素。测试项目进展包括阶段进展状况、任务进展状况和活动进展状况。1)阶段进展状况。一个测试项目通常有测试策划、测试设计、测试执行及测试总结等阶段<sup>[4]</sup>。每个阶段都有预计开始时间和结束时间,以及阶段开始和结束的约束条件。随着时间的推移,阶段进展状况记录以计划为基线的测试过程中各个阶段的实际进展情况,以及在整个测试项目中的完成情况等。2)任务进展状况。根据测试目标,一个测试项目可以定义若干个主测试任务,每个主测试任务又可以分解成若干个子测试任务。任务进展状况记录主测试任务及其子测试任务的实际执行情况。3)活动进展状况。在每个测试任务下,可以定义若干个关键活动和非关键活动。关键活动的执行情况决定了测试任务能否按测试计划完成,也间接决定了测试工作能否按计划完成。活动进展状况记录关键活动的实际执行状况。

通过跟踪测试项目进展状况,测试人员可以轻松了解测试项目的当前状态。一旦测试项目的实际进展状况与测试计划发生偏差,测试人员可以根据阶段、任务、活动的进展状况作相应调整。

### (2)测试规模信息

测试规模信息主要记录测试用例和软件问题报告的编写情况。编写测试用例的目的是发现软件问题。测试用例记录每个测试任务或测试活动下测试用例以及软件问题报告的数目。通过查看测试规模信息,测试人员可以轻松了解测试项目中哪些测试任务进行了充分测试,哪些测试任务尚未完全执行、有待进一步测试。

### (3)软件问题统计信息

测试的主要目的是为了发现软件问题,而软件问题的数量是软件质量是否过关的唯一标准,因此,分析软件问题的状态和趋势能有效追踪当前软件的质量状况,如果统计数据显示软件质量不过关,则测试工作需要进一步进行。软件问题统计信息<sup>[1]</sup>的描述如下:1)发现软件问题的数量和修复软件问题的数量的比率(find/fix ratio)。每天发现的软件问题数量如果大于每天修复的软件问题数量,即发现率大于纠错率,证明软件质量不过关。2)所有软件问题数量的时间走势。用曲线图绘制软件问题的走势。如果软件问题的总数在上升,

证明软件源代码还不稳定,软件质量不过关。3)未修复的软件问题数量的时间走势。在软件项目开发过程中,每天有多少软件问题被查出,并且没有被修复,这个度量值表现了开发团队的纠错速度和效率。如果未修复软件问题始终较多,证明软件还不能发行。4)所有软件问题按照严重程度的统计。软件问题按严重程度一般分为3个级别:S1, S2, S3,其中,S1代表最严重。这个度量值统计在软件中共有多少严重的软件问题和轻微的软件问题。5)未修复的软件问题按照严重程度的统计。这个度量值统计在软件中还有多少不同严重程度的软件问题,特别是S1严重程度的软件问题。如果有太多严重的软件问题没有修复,证明软件质量不过关。

(4)项目人工消耗信息。测试项目人工消耗是对测试人工成本的度量。人工消耗信息记录人员加入和离开测试项目的时间、测试项目各个阶段的人工消耗总和。测试人员可以根据人工消耗信息掌握测试项目的人工成本。

(5)测试变更信息。测试变更是测试控制的结果。测试项目实施过程中,有时依据实际情况需要对测试项目计划中某些项进行变更,如测试需求发生变化、测试人员发生流动、测试任务被取消等。测试变更信息主要记录测试需求、测试任务、测试人员以及测试项目状态的变更情况。测试变更次数越高,说明测试项目过程越不稳定,越难按照测试计划完成测试任务。

对测试项目进行追踪,能够从客观上避免原本合理的计划在实施过程中落空,防止按照不合理的计划行事,同时可以将追踪过程中产生的数据保存起来,为持续的过程改进提供有价值的信息。

## 2.3 测试项目控制

在测试实施过程中,测试项目的实际进展可能与测试计划存在偏差,这种偏差就是测试偏差。测试偏差可以分为零偏差、正偏差和负偏差。零偏差意味着没有偏差;正偏差说明测试进度比进度计划有所超前,或当前耗费人工成本少于计划中的预算成本;负偏差则相反。测试人员根据项目当前状况判断是否出现偏差,偏差是在合理的、可接受的范围内,还是应当尽快纠正。通过把项目实施过程中的进展信息与测试计划进行对比,及时发现测试实施情况与预期结果之间的差距,进一步了解情况,分析产生偏差的原因。对某个特定的偏差,测试人员还可以进一步分析在执行哪个测试任务时出现了偏差。

在工程项目中总结了如下一些偏差原因:

(1)内部原因。1)原来制定的测试计划不合理,过于保守的测试计划造成了正偏差;过于乐观的测试计划导致负偏差。2)方法革新、工具革新提高了效率造成了正偏差;资源不足、低效率、故障、人员离职导致负偏差。3)抽调技术人员造成进度的负偏差;测试需求分析不够清楚、测试方案有问题导致进度和成本的负偏差。

(2)外部原因。1)软件开发组提供的测试版本不稳定导致负偏差。2)软件开发组没有及时修复软件问题致使测试周期延长导致负偏差。3)测试工作量大、时间紧、人员不足导致负偏差。

确定了偏差产生的原因,就可以制定适当的纠正或预防措施。测试人员可以与开发人员或其他部门相关人员协调解决偏差产生的外部原因。对于内部原因,测试人员需要进行适当的测试项目控制,一般从测试需求控制、测试任务控制、测试项目状态控制3个方面解决偏差问题。

(1)测试需求控制。测试需求的状态分为草稿、审阅中、审批中、审阅完毕、审批完毕通过、审批完毕不通过、定稿 7 种,测试人员执行提交审阅、提交审批、审阅、审批、定稿 5 种操作,各种状态与操作之间的转移关系如图 2 所示(其中,椭圆代表状态;箭头代表操作)。一个测试需求的初始状态为草稿,之后可以提交审阅或审批,进行审阅审批工作。一旦针对某个测试需求的测试工作延误,测试人员可以追踪测试需求状态,迅速找出延误原因。譬如,如果一个测试需求的审阅审批工作出现多次审批不通过的状态,那么测试人员必须查验阻碍审批通过的原因,并尽快解决。

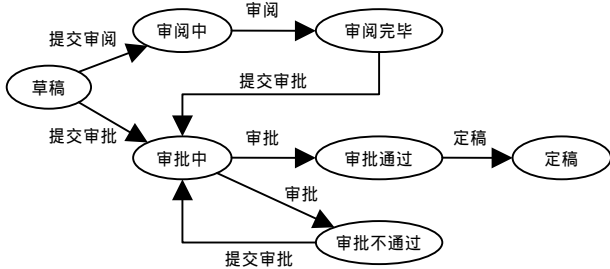


图 2 测试需求状态转移图

(2)测试任务控制。测试任务的状态可以分为生成、执行中、完成、废弃 4 种。在测试计划中,指定测试任务,初始状态为生成。在测试实施过程中,如果发生偏差的原因是由某一测试任务延误导致的,测试人员可以对测试任务下各个测试活动进行调整,甚至废弃该测试任务。只有测试任务下的所有测试活动都执行完毕后,该测试任务的状态才能变为完成,否则认为测试任务未执行完毕。

(3)测试项目状态控制。当测试项目出现较大负偏差时,仅调节测试需求、测试人员、测试任务是不足的,有时甚至直接修改项目状态。测试项目的状态可以分为未启动、启动、暂停、终止、结束 5 种,测试人员执行启动、暂停、恢复、终止、结束 5 种操作,各种状态与操作之间的转移关系如图

3 所示,其中,椭圆代表状态;箭头代表操作。

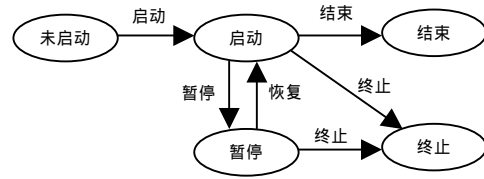


图 3 测试项目状态转移图

一个测试项目创建之初,它的状态是“未启动”,测试项目管理人员只有启动项目后,才能开展所有测试活动。在测试实施过程中,通过测试项目追踪了解项目当前状况,如果测试项目出现较大偏差,测试人员可以调整项目资源(如增减测试人员或调整测试时间等),也可以根据实现情况(如需要实施另一个更高优先级的测试项目)暂停项目,适时再恢复该项目,或终止测试项目,项目一旦终止就不能再实施任何测试活动。一个运行良好的测试项目完成时,测试人员可以结束该项目。

### 3 结束语

测试项目实施追踪与控制可以确保测试过程对测试人员可见、增强测试项目的可控性、提高测试效率、降低盲目测试的概率、节省测试成本。本文的测试项目追踪与控制方法已经在北京航空航天大学软件工程研究所研发的测试过程管理工具 QESuite2.0 中实现,并在一些测试项目中得到了良好的应用,但测试项目追踪与控制模型还有待进一步的完善,这也是下一步的主要工作。

#### 参考文献

- 1 栾 跃. 软件开发项目管理[M]. 上海: 上海交大出版社, 2005.
- 2 Shandilya R T. Software Project Management[M]. 北京: 科学出版社, 2002.
- 3 Patton R. Software Testing[M]. 北京: 机械工业出版社, 2005.
- 4 GJB/Z 141-2004 军用软件测试指南[S]. 2004.
- 5 McConnell S. Rapid Development: Timing Wild Software Schedules [M]. 北京: 电子工业出版社, 2002.

(上接第 67 页)

### 5 结束语

本文结合原有的概率数据库技术,以元组匹配所产生的聚类为基础,提出了基于聚类的非一致性数据的概率方法。这种方法扩展了以前关于概率数据库和 CQA 的工作 给出了基于可信聚类的基本查询重写技术。但需要在将来的工作中改进查询性能,扩展多关系连接查询重写及聚集问题。

#### 参考文献

- 1 Arenas M, Bertossi L, Chomicki J. Consistent Query Answers in Inconsistent Databases[C]//Proc. of ACM Symposium on Principles of Database Systems, Philadelphia. 1999.
- 2 Ananthkrishna R, Chaudhuri S. Eliminating Fuzzy Duplicates in Data Warehouses[C]//Proc. of International Conference on Very Large Databases, Hongkong. 2002: 586-597.
- 3 Fuhr N, Rolke T. A Probabilistic Relational Algebra for the

- Integration of Information Retrieval and Database Systems[J]. ACM Trans. of Inf. Sys., 1997, 15(1): 32-66.
- 4 Fuxman A, Fazli E. Efficient Management of Inconsistent Databases[C]//Proc. of ACM International Conference on Management of Data, Baltimore. 2005: 155-166.
- 5 Dalvi N, Suciu D. Efficient Query Evaluation on Probabilistic Databases[C]//Proc. of International Conference on Very Large Databases, Toronto. 2004.
- 6 Andritsos P, Fuxman A. Clean Answers over Dirty Databases: A Probabilistic Approach[C]//Proc. of International Conference on Data Engineering, Atlanta. 2006.
- 7 Transaction Processing Performance Council. TPC-H Standard Specification[Z]. <http://www.tpc.org/tpch>.
- 8 Hernandez M A, Stolfo S J. UIS Database Generator[EB/OL]. (2002-06). <http://www.cs.utexas.edu/users/ml/riddle/data.html>.